Strathmore UNIVERSITY

SU+ @ Strathmore University Library

Electronic Theses and Dissertations

2024

A Location based security response mobile application: a case study of Nairobi CBD.

Maturu, Alan
School of Computing and Engineering Sciences
Strathmore University

Recommended Citation

Maturu, A. (2024). *A Location based security response mobile application: A case study of Nairobi CBD*

[Strathmore University]. https://hdl.handle.net/11071/16547

Follow this and additional works at: https://hdl.handle.net/11071/16547

This work is availed for free and open access by Strathmore University Library.
It has been accepted for digital distribution by an authorized administrator of SU+ @Strathmore University.
For more information, please contact library@strathmore.edu

2024

A LOCATION BASED SECURITY RESPONSE MOBILE APPLICATION: A CASE STUDY OF NAIROBI CBD

Alan Maturu

School of Computing and Engineering Sciences

Strathmore University

A Location Based Security Response Mobile Application: A Case Study of Nairobi CBD

Alan Maturu

060477

Submitted to the School of Computing and Engineering Sciences in Partial Fulfillment for the Requirements for the Degree of Master of Science in Information Technology at Strathmore University

School of Computing and Engineering Sciences

Strathmore University

Nairobi, Kenya

March 2024

Declaration and Approval

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

© No part of this thesis may be reproduced without the permission of the author and Strathmore University.

Student Name: Alan Maturu

Student Signature: .. …….. Date: 26th March 2024

The thesis of Alan Maturu was received and approved by the following:

Dr. Julius Butime

School of Computing and Engineering Sciences

Strathmore University

Supervisor Signature: …… …….. Date: 27th March 2024

Abstract

Access to real-time reporting and response to security incidences has been a major issue in Kenya, with Nairobi, as the capital city, facing significant challenges. Over the years, the use of various forms of technology, including location-based services and emergency response systems, has been deployed to curb and report criminal activities in Kenya. Upon the occurrence of a crime, a victim usually reports it to the nearest police station, where they are issuedwith an OB (Occurrence Book) number for follow-up. At times, this system poses challenges, as response times to ongoing crimes are slow, and some victims shun off from reporting criminal activity due to poor response times or lethargic follow-up procedures. Moreover, the existing 999-call center, representing the current state of the art in emergency reporting, also demonstrates limitations in efficiency and response timeliness.

This study aims to establish how location-based technology, specifically GPS (Global Positioning System), can be used to manage real-time reporting, response, and management of criminal activities within Nairobi. This research, therefore, proposed the development of a location-based mobile application that facilitates the easy reporting of crime to police stations and assignment of on-duty police officers to a criminal case. The application, developed using a hybrid approach to support both Android and iOS platforms, aims to facilitate real-time response to criminal activities, enhancing accountability, and transparency. The agile deployment methodology, particularly Scrum, will be used to ensure the inclusion of all stakeholders from the Product Owner to the Scrum Master and the End User throughout the development phase. This methodology also emphasizes iterations to ensure continuous improvement until the End-Release phase of the product. Data collection for the research is proposed to be achieved through purposive sampling of Kenyan citizens and residents who have been victims of crime, as well as through the utilization of the application by police officers handling criminal cases reported at police stations. Expected outcomes of implementing this application include improved response times to criminal activities, enhanced monitoring of cases by victims, and increased information accessibility for police officers, thereby significantly impacting security within Nairobi.

Table of Contents

DECLARATION AND APPROVAL ..... **III**

ABSTRACT ..... **IV**

LIST OF ABBREVIATIONS ..... **VII**

DEFINITION OF TERMS ..... **VIII**

ACKNOWLEDGEMENTS ..... **IX**

CHAPTER 1: INTRODUCTION **1**

1.1. Background of the Study 1
1.2. Problem Statement 2
1.3. Main Objective 3
1.4. Specific Objectives 3
1.5. Research Questions 3
1.6. Justification 4
1.7. Scope and Limitation 4

CHAPTER 2: LITERATURE REVIEW **5**

2.1. Introduction 5
2.2. Empirical Literature 5
2.3. Theoretical Framework 9
2.4. Location-based security response – why it's needed and its significance 10
2.5. Mobile Application Adoption in Law Enforcement 11
2.6. The Potential of Mobile Technology in Enhancing Security 12
2.7. Technologies and Approaches for Location-Based Security Response 14
2.7.1 Conventional Emergency Hotline System 14
2.7.2 Wearable Panic Buttons 14
2.7.3 Radio Frequency Identification Technology (RFID) 15
2.7.4 Cellular Networks 16
2.7.5 Global Positioning System (GPS) 16
2.7.6 Peer to Peer Communication 17
2.8. Impediments to Law Enforcement Utilizing Technology to Respond to Reported Crimes 18
2.9. Exploring Solutions for Enhancing Police Response to Crime Reports Made Through Mobile Applications 18
2.10. User Experience and Acceptance of Mobile Applications 19
2.11. Design and Development Requirements for a Location-Based Security Response MobileApplication. 20
2.12. Frameworks/Models 21
2.13. Algorithms 22
2.14. Existing Applications 23
2.15. Current Security Response Strategies: Effectiveness and their Challenges 25
2.16. Conceptual Framework 25
2.17. The Issue of false alert in a location-based system 26

CHAPTER 3: RESEARCH METHODOLOGY **27**

3.1. Introduction 27
3.2. Study Population 27
3.3. Fairness of the Selection and Recruitment Procedures 27
3.4. System Development Methodology 28
3.5. Requirements Phase 28
3.6. Data Collection and Analysis 28
3.7. Appropriateness of the research tools 29

3.8. Research Design and Development 29
3.9. System Testing 29
3.10. Risk of this Research 29
3.11. Research Quality 30
3.12. Risk/Benefit Analysis 30
3.13. Dissemination and Utilization of Results 31
3.14. Ethical Considerations 32

CHAPTER 4: DATA ANALYSIS 33

4.1. Introduction 33
4.2. Response Data and Rate 33
4.3. Owing a mobile device and the type of OS on the device 34
4.4. Modes of crime reporting that people are aware of within Nairobi CBD? 36
4.5. Reporting and responding to Criminal Emergencies 38
4.6. Willingness to report a Crime and willingness to respond to a Crime via a MobileApplication. 40
4.7. Factors to consider to rapidly respond to a criminal emergency that a crime victim reports 44

CHAPTER 5: SYSTEM DESIGN, IMPLEMENTATION AND TESTING 46

5.1. Introduction 46
5.2. System Architecture and Design 46
5.2.1 System Overview 46
5.2.2 System Architecture 47
5.2.3 User Interface and Experience Design 47
5.2.4 User Role: Normal User 47
5.2.5 User Role: Law Enforcement Officer 50
5.2.6 Database Design 52
5.2.7 Entity Relationship Diagram 53
5.2.8 Sequence Diagram 54
5.2.9 UML Diagram 55
5.2.10 Use Case Diagram 56
5.3. System Testing 58
5.3.1 Approach to Testing 58
5.4. Security Measures 58

CHAPTER 6: CONCLUSION AND RECOMMENDATION 60

Recommendations and Future Work 61

REFERENCE 62

APPENDIX A: PLAGIARISM CHECKER 66

APPENDIX B: ETHICS REVIEW BOARD APPROVAL 67

APPENDIX C: NACOSTI PERMIT 68

APPENDIX D: RESEARCH QUESTIONS 69

Law Enforcers 69
Crime Victims 72

APPENDIX E: SYSTEM ARCHITECTURE 75

APPENDIX F: APPLICATION USER INTERFACE 76

APPENDIX G: USER MANUAL 85

APPENDIX H: BACK END/SOURCE CODE 88

List of Abbreviations

CCTV	Closed-Circuit Television (CCTV),
DNA	De-oxy Ribonucleic Acid (DNA)
GPS	Global Positioning System
iOS	iPhone Operating System
IT	Information Technology
NAVSTAR	Navigation Satellite Timing and Ranging
OOP	Object Oriented Programming
P2P Communication	Peer to Peer Communication
RFID	Radio Frequency Identification
SECaaS	Security as a Service
SU-IERC	The Strathmore University Institutional Ethics Review Committee

Definition of Terms

GPS (Global Positioning System): A satellite-based navigation system developed by the US Department of Defense under its NAVSTAR (Navigation Satellite Timing and Ranging) satellite program (Grewal, Andrews and Bartone, 2013).

P2P Architecture (Peer-to-Peer Architecture): A way to structure a distributed application so that it consists of many identical software modules, each module running on a different computer. The different software modules communicate with each other to complete the processing required for the completion of the distributed application (Verma, 2004).

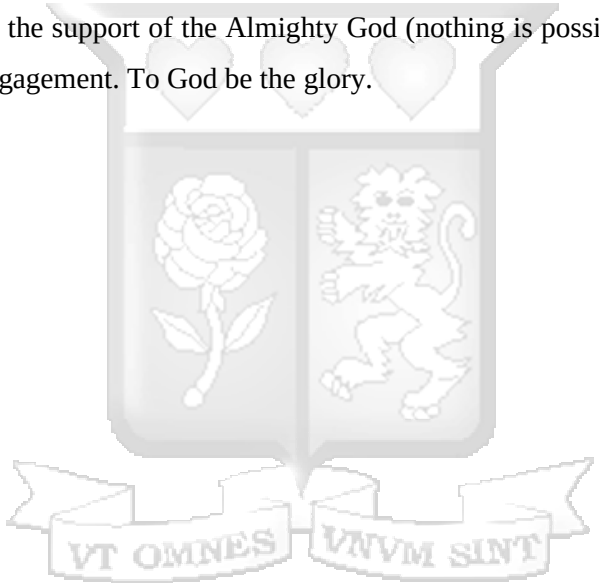
RFID (Radio Frequency Identity): A communication technology that uses radio waves for the storing and retrieval of data from identification chips (Briseno, Hirata, Lopez, Garcia, Cota and Hipolito, 2012).

Acknowledgements

I acknowledge the School of Computing and Engineering for granting me the opportunity to pursue a Masters in Information Technology course and thesis seminars that were guided by Dr. Julius Butime to help us to achieve quality work.

I greatly acknowledge the guidance from my colleagues who have been extremely instrumental and supportive towards this initiative. I laud their support as well as the encouragement accorded throughout my academic journey.

Last but not least, I acknowledge the support of the Almighty God (nothing is possible without Him) and my immediate family towards this engagement. To God be the glory.



Chapter 1: Introduction

1.1. Background of the Study

According to Citizen Digital on November 14th, 2022, the Cabinet Secretary of Interior and Coordination of National Government, Prof. Kindiki pointed out that the government has noticed an increase in crime rate in the Nairobi City (Citizen Digital, 2022). In accordance with Article 243 of the new Constitution of Kenya, Law Enforcement Agencies are tasked with the responsibility of protecting citizens, controlling crime, and maintaining public order (Kenya, 2013). This entails providing aid to the public when necessary, safeguarding peace, protecting lives and property of citizens, collecting intelligence, and investigating criminal activities. Over the years, Law Enforcement Agencies have been striving to meet these objectives, such as by responding toreported cases of crime as quickly as possible, inspecting a large number of cases, and preserving peace and protecting citizens to the best of their abilities. When a citizen identifies or observes an illegal activity, the standard procedure for reporting the offence involves capturing the details of the crime, then contacting the police through the designated emergency number to file the complaint (Oketch, 2018). If the crime does not necessitate immediate attention, the witness is expected to visit the nearest police station to submit a report. Subsequently, the police will strive to respond to the crime, which can be contingent on various factors such as the type of crime, proximity of the police station to the incident, availability of personnel and possibly other external considerations such as the presence of police vehicles. After making an initial report, there is a chance that the witness may be redirected to another division that focuses on the particular category of the crime, for instance, accident and gender-based violence section. It is evident from this process that the response time to thecrime can range from 30 minutes to an indefinite amount of time depending on the severity of the crime. It is logical to assess that this performance is suboptimal. Hypothetically, if a criminal act has been reported, the probability of its successful completion is high, particularly when the law enforcement officer is not in the immediate vicinity of the crime. Furthermore, it is likely that the perpetrator will evade apprehension without a trace.

Over the last decades, rapid technological changes have occurred in the security and policing sector. Koperet al., (2015) observed that the information technology (IT) has brought important development in mobile phone, video surveillance systems, license plates readers, De-oxy Ribonucleic Acid (DNA)

testing, Closed-Circuit Television (CCTV), among others which have far reaching effects on police agencies. The increasing advancement of technology has necessitated a revision of the conventional security and policing strategies. In particular, I believe that the integration of a location-based security response approach has the potential to expedite crime resolution and enhance the efficacy of law enforcement.

1.2. Problem Statement

Crime is a pervasive challenge that hampers economic growth and development in a nation, undermining overall wellbeing and creating a climate of fear and compromising human dignity. The traditional model of crime prevention is rooted in the perception that the primary issue is adherence to law and crime prevention policies and law enforcement agencies should direct their efforts towards addressing current and potential violations of the law. According to White (1996), the solution to crime prevention lies in reducing the opportunity for committing crimes and bolstering the probability of detecting them. At the basic level, these elements include police patrol, cross-circuit cameras, emergency and confidential hotlines, and citizen watch committees. Since the implementation of this model, law enforcement agencies have achieved partial success in managing crimes, although there is room for improvement in terms of response time and duration of closing a crime case.

Due to the increased crime, there has been adoption of various legacy strategies to mitigate the crime rate in Nairobi CBD. It is also paramount to note that the law enforcement agencies are spending large sums on technology in view of enhancing their efficiency and effectiveness. This is because fighting crime all over the world continues to pose great challenge to law enforcement officers in the recent years (Quarshie, 2014).

As technology continues to evolve, mobile phones have become an integral part of everyday life, transforming from mere 'technological objects' to key 'social objects' (Srivastava, 2005). The Communication Authority of Kenya (2021) has reported that, as of the beginning of 2022, there were at least 59 million mobile phone devices connected to the internet, with 33 million of those being featured phones. The implementation of technological services has earned substantial attention from governmental entities, with information technologybeing utilized for the purpose of criminal reporting and evidence collection. Although this approach has been successful in the identification and reporting of criminal activity, the response to such offences has been sluggish and requires improvement. Upon analyzing the factors which influence an individual's reportage of a crime, a case study of Githurai 45, Nairobi, concluded that failure to report major crimes to police was primarily caused by the perceived

laboriousness of the reporting process (Kimenju, 2015). If law enforcement agencies become more responsive to crime, the nation's criminal rate would decrease significantly, thus improving society's overall growth.
This thesis proposes the creation of a location-based security response mobile application to ensure an optimized physical security reporting, response and tracking within Nairobi CBD by pairing of a designated police officer within the area of crime and the victim of a crime.

1.3. Main Objective

The aim of this research is to develop a location-based security response mobile application for real-time physical security reporting, response and tracking in Nairobi CBD.

1.4. Specific Objectives

1. To assess the possibility of a crime victim using a mobile application to report a crime.
2. To assess the possibility of a law enforcer being notified and responding to a crime reportedvia a mobile application.
3. To assess the factors that a mobile application user would factor in when using a location-based securityresponse application.
4. To develop a location-based security response mobile application.
5. To test the proposed system.

1.5. Research Questions

1. What is the probability that a crime victim would report a crime using a mobile application?
2. What is the probability that a law enforcer would be notified and respond to a criminal emergency use a mobile application?
3. What are the factors that a mobile application user would factor in when using a location-based securityresponse mobile system?
4. How will the location-based security response mobile application be developed?
5. How will the location-based security response application be tested?

1.6. Justification

The objective of this study is to develop a mobile application system that is geared towards the initial real- time reporting of insecurity incidents from a geo-fenced location within the jurisdiction of the responding police officer(s). In addition to reporting, the application will also facilitate speedy assignment of law enforcement officers such as the NPLS (National Police Service) to aid in not only efficient response but also tracking and monitoring of reported cases.

As per research best practices aims to not only add to the ICT knowledge bank but to also to address a societal concern; this application will aid to facilitate satisfaction of the citizenry through speedy and efficient provision of security services as well as help the enforcement officers such as the NPLS to maintain accountability and transparency in their core mandate.

1.7. Scope and Limitation

This scope for this research is limited to Nairobi CBD as it is assumed that it has a high concentration of android smart phones as compared to other parts of the country. In addition, with the location being within the capital city of Kenya, it is assumed that the required infrastructure to support the use of the mobile application such as stable internet is present. In addition, there is crime prevalence in this area hence the focus which also factors in a limitation of time as far as this research is concerned.

It is also assumed that a majority of the citizenry in Nairobi use android mobile phones as compared to Apple iOS (iPhone Operating System) hence the limitation of android phones as compared to Apple iOS (iPhone Operating System). It is therefore safe to note that initially and for purposes of this research that users who will have phones that are not android in nature will be left out in the use of this application.

Chapter 2: Literature Review

2.1. Introduction

Security in Kenya, particularly in Nairobi, the capital city, has been a major concern for residents and the government alike. Crime and insecurity have been on the rise in recent years, affecting the lives of many citizens and hindering economic growth (Bett & Muturi, 2016). With the increasing number of terrorist attacks,armed robberies, and crime syndicates operating in the country, security has become a major issue (Bett & Muturi, 2016). In this appraisal, we present an analysis of the current state of security, particularly in Nairobi, Kenya. We evaluate precedent technologies and approaches that have been taken to eliminate or at least curb insecurities across Nairobi. We argue that though previous implementations have helped to a great extent in the prevention and elimination of crimes in the city, a location-based security mobile application would further improve security by providing real-time inputs on the current state of security across the towns in Nairobi.

2.2. Empirical Literature

In recent years, there have been various studies conducted in Kenya and other countries on the effectiveness of mobile applications in enhancing security. Omolo (2016) explores the development of a mobile and web-based application for the purpose of collecting security intelligence. The primary objective of the author was to create a platform that would assist security personnel in Kenya to acquire information and intelligence regarding potential security threats in real-time. The research entailed the development of a prototype mobile and web- based application, which enabled security agents to record and communicate incidents, share data, and collaborate on investigations. The application was designed to be user-friendly and accessible from any device with an internet connection. A case study was conducted to evaluate the effectiveness of the application in actual scenarios. The study revealed that the mobile and web-based application was a competent tool for obtaining security intelligence. It enabled security agents to capture and report incidents in real-time, share information, and collaborate on investigations. The application was also identified to be user-friendly, easy to use, and accessible from any device with an internet connection.

Other studies have focused on the effectiveness of location-based services in enhancing security. A study by PARK et al., (2018) on the impact of location-based services on crime prevention found that the use of these services helped to deter crime by providing real-time information on crime hotspots and enabling law enforcement agencies to respond promptly. Furthermore, a study by Huang et al., (2018) on the effectiveness of location-based services in enhancing public safety found that the use of these services led to a reduction in

crime rates and improved safety for citizens.

Apima (2018) discusses the potential of mobile phone tracking technology in thwarting crime in Nairobi City County, Kenya. The primary objective of the study was to explore the efficacy of employing mobile phone tracking technology as a tool for crime prevention in Nairobi City County. To gain insight into the public's opinion on the utilization of mobile phone tracking technology for crime prevention, the author also conducted a survey. The findings of the study suggest that mobile phone tracking technology could be a potent tool for the prevention of crime in Nairobi City County. Monitoring and prediction of criminal activity and tracking and apprehension of criminals can be facilitated by this technology. As well, the public in the area holds a favorable perception of the usage of mobile phone tracking technology for crime prevention. However, the study also highlighted certain impediments associated with this technology such as privacy concerns and potential misuse. Toaddress such issues, the author proposed the formulation of explicit policies and regulations pertaining to the implementation of mobile phone tracking technology for anti-crime endeavors in Nairobi City County.

Filippi et al., (2020) paper explores the utilization of various data sources to tackle urban security in informal areas, utilizing Kibera in Nairobi as an example. The primary aim of the authors was to probe how the amalgamation of varied data sources can better security strategies in informal urban regions. The inquiry entailed thegathering and study of disparate kinds of data, including satellite imagery, social media data, and data from mobile phone networks. The authors utilized these data sources to chart the urban setting of Kibera and pinpoint areas of criminal activity and insecurity. The study found that the integration of different data sources can provide amore holistic understanding of security matters in informal urban areas. The use of satellite imagery and social media data, in particular, was proven to be advantageous in mapping the urban environment of Kibera and locating areas of high crime and insecurity. The authors also determined that the proposed security protocols, based on the analysis of the data, were successful in subduing crime and augmenting security in Kibera. Nevertheless, the study also pinpointed certain difficulties connected with the integration of disparate datasources, such as problems with data quality and privacy worries.

Oketch (2018) paper investigates the utilization of a crime reporting and intelligence system in Buruburu Police Division in Nairobi, Kenya in order to evaluate its effectiveness in enhancing crime reporting and intelligence gathering. The research included the development and implementation of a web-based crime reporting and intelligence system that allowed the police to record and report incidents, exchange information, and cooperate on investigations. The results of the study indicated that the crime reporting and intelligence system was an effective tool for amplifying crime reporting and intelligence gathering in Buruburu Police Division. The system enabled the police to capture and report incidents in a timely manner, share information, and cooperate on investigations. Additionally, it was discovered to be user-friendly and accessible from any device with an internet connection. Nonetheless, some challenges associated with the implementation of the system were identified, such as those related to data quality and privacy concerns.

Mugi (2016) investigates the privacy-related issues associated with Location-Based Services (LBS) and presents a framework aiming to improve the privacy of users. The author's fundamental objective was to create a system that allows users to utilize LBS while protecting their privacy. The research involved the development of a conceptual framework comprising of various privacy-enhancing components, such as authentication, data anonymization and consent management. The results of the investigation revealed that the framework was successful in enhancing privacy of users when using LBS. The framework granted users the power to manage the amount and type of information shared with the service providers, whilst protecting the data from illegal use. Furthermore, the framework enabled users to withdraw their consent at any time, thus furthering their privacy. The study also identified some challenges associated with the implementation of the framework, such as issues with data quality and user acceptance.

Wasomi et al., (2021) paper investigates the acceptance and adoption of location-based services (LBS) among mobile phone users in Nairobi, Kenya. The primary objective of the study was to analyze the factors thatinfluence the uptake and usage of LBS in the area. Results indicated that the majority of mobile phone users in Nairobi were aware of LBS and had used them at least once. Navigation services were the most widely used LBS, followed by location-based social networking and tracking services. Further, perceived usefulness and ease of use were identified as the most significant factors influencing the adoption and usage of LBS. Barriers to the adoption and usage of LBS, such as concerns about privacy and security, lack of awareness of the benefits of LBS, and the cost of data, were also identified.

Vinarao et al., (2019) examine the efficacy of Athena, a mobile-based application designed to ameliorate the safety of women in Rizal Province, Philippines. The application offers features such as the ability to send distress signals to the police and emergency contacts as well as real-time GPS tracking of the user's location for improved assistance. Positive outcomes and results were observed, with the application being found to be efficient and user-friendly. Women participating in the study reported feeling safer with the application installed on their phones and many would recommend it to other women. The table below highlights a summary of related works:

Citation	**Paper Title**	**Technology Used**	**Gap (s)**
APIMA, M. B. (2019)	The Role of Mobile Phone Tracking Technology in Crime Prevention in Kenya: A Case of Nairobi City County	Mobile Phone Tracking Technology	The authors demonstrate the effectiveness of mobile tracking technology in enhancing crime reporting and law enforcement's response capabilities in Nairobi.
Babu, S. S., Yellepeddi Vijayalakshmi, D. A., & Kala, J. R. (2016)	Improving Performance and Privacy in Location Based Service Applications Using GPS	GPS	This research indicates the advancements in GPS technology to enhance performance and privacy for users of location-based services.

& Sabherwal, R. (2020).	the predictor of security perceptions.		security perceptions, emphasizing the importance of privacy measures in mobile app development for security applications.
de Guzman, J. B., & Ado, R. G. (2014).	Mobile emergency response application using geolocation for command centers.	Geolocation	They address the use of geolocation in mobile applications for enhancing emergency response capabilities of command centers, directly relevant to improving law enforcement's response to reported crimes.
Jang, S. H., & Lee, C. W. (2018).	The impact of location-based service factors on usage intentions for technology acceptance: The moderating effect of innovativeness.	Location-Based Services	Explores how location-based service factors influence the intention to use such technologies, considering the moderating role of user innovativeness, which can inform the design and implementation strategies for location-based security response applications.
Kaur, S., & Sharma, S. (2020).	Role of Location-Based Security Response Mobile Applications in Enhancing Security in the Society.	Location-Based Security Response Mobile Applications	Discusses the direct impact of location-based security applications on societal security, supporting the objective to assess the utility of such apps in crime reporting and law enforcement response enhancement.
Kaushal, A., & Anand, P. (2020).	Location Based Security Response Mobile Application: Need of the Hour.	Location-Based Security Response Mobile Applications	Argues for the critical need for location-based security apps, reinforcing the objectives related to assessing mobile app usage for crime reporting and enhancing law enforcement's response with technology.
Kumbuti, H. M. (2013).	Use of technology as a strategy by Kenya police in detection of crimes in Nairobi City.	Various Technologies in Crime Detection	Highlights the integration of technology in crime detection by the Kenya police, offering insights into the technological advancements that could be leveraged in the development and efficacy of a mobile crime reporting and response application.

Quarshie, H. O. (2014)	Using ICT to Fight Crime-A Case of Africa.	ICT in Crime Fighting	Discusses the broad use of Information and Communication Technology (ICT) in combating crime across Africa, providing a contextual backdrop for the deployment of a location-based security response application in Nairobi.
Cruz, E. N. (2019, October).	Athena: A Mobile Based Application for Women's Safety with GPS Tracking and Police Notification for Rizal Province.	GPS Tracking and Police Notification	Illustrates the development and implementation of a safety application specifically designed for women, utilizing GPS tracking and direct police notification. This directly supports the exploration of mobile applications for crime reporting and the facilitation of rapid law enforcement response.
Wambugu, P. M. (2017).	Application of Mobile Phone in Crime Prevention within Central Division, Nairobi City County	Mobile Phones in Crime Prevention	Highlights how mobile phone technology has been utilized for crime prevention in Nairobi, aligning with objectives related to assessing the use of mobile apps for crime reporting and enhancing law enforcement response.

Table 1: Summary of related works

2.3. Theoretical Framework

The Security Situation in Kenya

Statistics show that the number of reported incidents of violence and crime increased by 10% in the year 2019 compared to 2018. According to the Kenya National Bureau of Statistics (2021), the total number of crimes reported to the police in 2020 was 69,645, the command station Nairobi, reporting 5,844 cases. This is a worrying trend as it suggests that the level of security is increasing. A recent study by the International Crisis Group showed that the security situation in Nairobi has increased in recent years, with crime and violencelevels rising across the city (International Crisis Group, 2018).

Statistics show that incidents of robbery, carjacking, and home invasions have increased significantly in the

last few years. There has also been an increase in the number of gang related violence, as well as violent protests and demonstrations. The political environment in the city has been volatile, with frequent clashes between rival political parties (Kenya Travel Advisory, 2022). This has resulted in an increased level of insecurity and fear amongst citizens. According to the Kenya National Bureau of Statistics, there has been a 20% increase in reported cases of crime in Nairobi between 2019 and 2020 (Kenya National Bureau of Statistics, 2021).

The causes of insecurity in Nairobi are complex and multi-faceted. Factors such as poverty, unemployment, and political instability have contributed to the rise in crime (Atta-Asamoah, 2015). The lack of adequate policing and limited resources available to the law enforcement agencies have also contributed to the problem. The insecurity in Nairobi has affected the people and the country in numerous ways. Residents are now more cautious and fearful of crime, leading to a decrease in the quality of life (Natasha, 2007). Businesses, particularly those operating in areas with high crime rates, have also been affected, with some closing down due to security concerns.

It is difficult to predict the future state of security in Nairobi with certainty. However, the government has recently taken steps to address the problem, such as increasing the number of police personnel and investing in crime-prevention technology. The government has also implemented a number of initiatives aimed at improving security, such as the Nairobi Security Network and the Nairobi Security Centre. These efforts, along with increased community involvement and public-private partnerships, may help to reduce the incidence of crime in the future. In addition, with the advancement in technology, such as the development of GPS enabled smartphone devices, there is a probability that crime will reduce in future as it will be easier and more flexible toreport crimes, hotspot crime areas, while alerting the community and security personnel around the area.

2.4. Location-based security response – why it's needed and its significance.

The integration of mobile technology in addressing security concerns in urban areas has resulted in theemergence of location-based security response mobile applications, leveraging GPS technology for real- time information and swift response. Location-based security response mobile applications are becoming increasingly vital in contemporary society, affording users a convenient and secure means of accessing security services while on the move (Pitt & Wager, 2020). These applications provide users with the capability to monitortheir environment, solicit aid, and receive real-time updates on security threats in their vicinity. Kaushal and Anand (2020) claim that this heightened level of security is especially invaluable in situations where the user is travelling through unfamiliar regions or high-risk areas.

The utilization of location-based security response mobile applications offers a plethora of advantages. First, the user can be informed of potential threats in their vicinity, thus enabling them to take suitable measures. In addition, the user can request assistance from emergency services, such as fire or police departments, as required. Real-time information sharing and response coordination between various agencies and organizations lead to an optimized use of resources, resulting in a more comprehensive approach to security incidents (Hassan & Alashqar, 2020).

Location-based security response mobile applications encourage public engagement by providing real- time information and enabling citizens to report incidents, thereby fostering awareness and participation in securing urban areas. Moreover, the user can also gain access to updates pertaining to security threats, for example reports of criminal activity. This allows the user to remain informed and aware of their surroundings. Location-based security response mobile applications also provide a range of features designed to ensure the user's security. These features include an emergency alert system, which allows the user to quickly signal emergency services if they find themselves in a precarious situation (Kaur & Sharma, 2020). Additionally, the application can be configured to issue notifications when the user enters a high-risk area, thereby allowing them to take the necessary precautions.

The ability to access security services on the go is especially pertinent in areas with high levels of criminal activity. Through the use of location-based security response mobile applications, users can instantly and easily obtain assistance when required. This is particularly useful in regions where the user may not be familiar with the area or may not possess the required resources to access conventional security services. As well, the application can disseminate information on criminal activity in the area, allowing the user to remain informed and take the appropriate measures.

2.5. Mobile Application Adoption in Law Enforcement

Mobile application adoption in law enforcement is an area of growing interest due to the advances intechnology and the availability of cost-effective solutions. The integration of technology into society hasresulted in a transformative effect on the criminal justice system, with law enforcement agencies adoptingadvanced solutions to enhance their response capabilities towards security threats (Jang & Lee, 2018). One suchinnovation is the implementation of location-based security response mobile applications, with Nairobi CBDserving as a pertinent case study to examine the influence of these applications on the criminal justice landscape. In comparison to conventional methods of crime reporting, mobile applications provide a host of advantages. These applications furnish real-time data, allowing law enforcement to swiftly respond to incidents (Eneh & Orah, 2012). They also provide greater accessibility, enabling victims to report crimes from any location and at any time, eliminating the need for physical visits to police stations. Furthermore, these applications present a more secure platform for reporting crimes, as the information is encrypted and only accessible to authorized

personnel.

Despite the numerous benefits, some law enforcement agencies have been slow to adopt these applications, owing to a dearth of understanding regarding the technology and its advantages, as well as concerns regarding the costs associated with implementation and maintenance of the systems, insufficient personnel training, and potential inaccuracies or breaches in the collected data.

To ensure the successful implementation and adoption of location-based security response mobile applications in law enforcement, it is crucial to overcome these barriers through targeted training, education, and awareness programs. In addition, law enforcement agencies must collaborate with technology providers to design systems that are user-friendly and meet their specific needs.

The incorporation of location-based mobile applications in law enforcement is a novel approach, and it should gain momentum due to its potential to optimize the efficacy of law enforcement operations (O'Hara, 2018). The location-based mobile applications can be utilized to monitor the whereabouts of police officers, notify them of potential threats, and give them insights into the current security situation in the area. This can allow police officers to act quickly and effectively to the security challenges they face.

In Nairobi Metro, the adoption of location-based mobile applications has been implemented to facilitate the police response to security threats (Wambugu, 2017). The police officers are equipped with mobile devices that can be used to track their location and provide them with real-time intelligence about the security situation in the area. The mobile devices are also used to alert police officers about potential security hazards. Moreover, the application also permits police officers to exchange information with other police officers and to coordinate their efforts in responding to security threats.

The use of location-based mobile applications in Nairobi CBD has been demonstrated to be effective in addressing security threats. The application has enabled the police officers to act swiftly and effectively to the security threats they face, which has improved the safety of citizens. As well, the utilization of the application has allowed the police to better collate their efforts in responding to security threats, which has raised the efficiency of the response.

2.6. The Potential of Mobile Technology in Enhancing Security

The utilization of mobile technology has great potential in reinforcing the security of a given region, especially in urban settings. This is due to its ability to offer a variety of functions, such as access to real- time data, enhanced communication, and the capacity to observe and trace individuals. The integration of

mobile technology in security systems can yield better results when it comes to monitoring and addressing security threats, thereby ensuring more secure environments (Balapour et al., 2020). The advantages of mobile technology in security are numerous and include real-time information access, rapid response, data collection and analysis, and improved inter-agency communication and collaboration.

Mobile technology can allow security personnel to send and receive messages in real-time, which can be utilized to alert the public to any potential threats and furnish them with suitable safety measures (Volkoviy & Tikhov, 2012). Moreover, mobile technology can enable security personnel to observe and trace individuals in the area, providing them with data on their whereabouts and any suspicious activities they maybe engaging in. This can be utilized to identify any potential threats and provide a prompt response. Mobile technology can allow security personnel to access databases of information on potential threats, allowing them to monitor the situation and react appropriately. Mobile devices can also serve as a platform for location- based security response applications, providing real-time information on incidents and facilitating rapid response (Inukollu et al., 2014). This data can be used to educate the public of any potential threats, as well as toprovide security personnel with the necessary information to respond quickly and effectively. As well, it will improve security if the public get an access to data that shows crime hotspots at a given timeframe and this will generally improve security across the country.

Omolo (2016) suggests that a location-based security response mobile application can be used to bolster security in Nairobi and the application would allow security personnel to identify potential threats in the area and respond in a timely manner. The application would be capable of tracking the location of potential threats, aswell as furnishing security personnel with real-time data on the situation. Furthermore, the application could be used to inform the public of any potential threats and provide them with safety precautions.

Furthermore, the collection and analysis of data is a significant aspect of mobile technology in security. Security systems with mobile capabilities can store large amounts of data, which can be analyzed to uncover patterns and trends. This information can aid in the improvement of existing security systems and the development of new security strategies. Mobile technology promotes inter-agency communication and collaboration, enabling security personnel to share information and coordinate responses to security incidents, thus augmenting overall security.

2.7. Technologies and Approaches for Location-Based Security Response

With the accelerated expansion of mobile devices and the propagation of location-based tracking and navigating technologies, it has become viable to pinpoint security threats with greater speed and accuracy, thereby allowing for a timelier response. There are various technologies such as Radio-Frequency Identification(RFID), GPS, peer to peer computing, and cellular networks that can be utilized to construct location-based security response systems.

2.7.1 Conventional Emergency Hotline System

The earliest form of location-based security response technology was the conventional emergency hotline system. This system is based on manual reporting of an emergency via phone call and required the victim to provide accurate location information. However, this system is limited in its efficiency and relied on the ability of the victim to make the call.

The use of a conventional emergency hotline system to provide a location-based security response in a city is not an effective strategy due to several key drawbacks. First and foremost, a conventional emergency hotline system is designed to provide immediate assistance in the event of an emergency. However, in order to provide alocation-based security response, the system would need to be able to track the location of the caller in real time, which is not something that a conventional system is capable of. Secondly, even if a conventional emergency hotline system could track the caller's location, it would still be difficult to provide an appropriate security response in a timely manner.

While the system may be able to alert police or other authorities to the caller's location, it could take several minutes or even hours for those authorities to arrive on the scene. In the meantime, the person calling for help may be in an increasingly dire situation. Lastly, a conventional emergency hotline system is not designed to provide ongoing security. It is designed to provide immediate assistance in the event of an emergency. It is not equipped to provide ongoing support or protection to the caller. As such, it is not an appropriate strategy for providing a location-based security response in a city.

2.7.2 Wearable Panic Buttons

The development of wearable panic buttons represented another step in the evolution of location-based security response technologies (Tsai et al., 2010). These devices, in the form of wristbands or pendants, could be triggered by the wearer in the event of an emergency and transmitted the wearer's location

information to the authorities. This technology was particularly useful for individuals who may not have immediate access to a phone or may not be able to make a call.

The use of wearable panic buttons to provide a location-based security response in a city is an increasingly popular security measure. However, it is important to consider why it may not be an effective approach. Firstly, the implementation of panic buttons can be expensive and may not be a feasible option for many cities. This is particularly true for cities that are already strapped for funds and resources, for example, Nairobi. Furthermore, the implementation and maintenance of these devices will also require additional funds and personnel. Secondly, wearable panic buttons may not be effective in areas where crime is rampant. This is because criminals may be aware of the presence of the panic buttons and may be less likely to act if they are present. Additionally, criminals may also be able to circumvent security measures by using other undetectable methods of entering a building or area. Thirdly, the use of wearable panic buttons may be ineffective if there is a lack of aresponse from law enforcement. This is because even if a person presses the panic button, if there is no immediate response from police, the threat may become worse, or the criminal may have already left the area. Finally, the use of wearable panic buttons may not be an effective security solution because they create a false sense of security. People may become complacent and may not take other safety measures such as avoiding dangerous areas or carrying pepper spray.

2.7.3 Radio Frequency Identification Technology (RFID)

RFID is a technology that uses radio waves to recognize and monitor objects. It is based on the transmission of electromagnetism signals between electronic tags and readers. RFID tags are typically affixed to objects, such as people, vehicles, or goods, and can be utilized to trace the movements of these objects. RFID technology is implemented to construct location-based security response systems that can identify and track objects in a designated area (Lyon et al., 2005). For instance, RFID tags can be employed to identify and track people or vehicles in an area, thereby enabling security forces to respond to security threats without delay.

However, it is not the most effective strategy for providing a secure response in a city. RFID technology works by sending out radio signals that are picked up by receivers, which can then be used to track the location of a person or object. This technology has been used in the past to identify criminals, or provide security at public events. On the surface, it seems like an ideal solution for providing location-based security in a city. However, there are several drawbacks to RFID technology that make it less than ideal for providing a secure response in a city. For one, the signals sent out by RFID devices can be easily blocked, or even hacked.

This means that someone with malicious intent could easily access the secure data or use it to their advantage. Furthermore, the signals sent out by RFID devices are limited in range, which means that they are not suitable for providing secure coverage over a large area. In addition, the cost of installingand maintaining RFID technology is quite high, and the technology is not always reliable. This means that it maynot always be feasible to install and maintain RFID systems in a city.

2.7.4 Cellular Networks

Cellular networks can also be utilized to create location-based security response systems. Cellular networks are used to transmit data and voice signals over a cellular network. The signals are received by cellular towers, which are equipped with antennas that are used to triangulate the location of the cell phone. This technology is used to identify and track objects in a specified area, allowing for a more precise and prompt response.

The use of cellular networks to provide location-based security response in a city is not an effective strategy due to a variety of reasons. First, cellular networks are designed to provide communication services, not security. As such, they are not equipped to handle the complexities and security needs associated with such a system. Additionally, cellular networks are prone to interference, making it difficult to accurately track and respond to security threats in a timely manner. Another issue with using cellular networks for security purposes is the cost associated with such systems. The installation and upkeep of these systems can be very expensive, and it is not always possible for cities to afford such an investment. Furthermore, many cities lack the infrastructure necessary to provide cellular coverage over a large area, which would be necessary for a location-based security response system. Finally, cellular networks are not 100% secure, and cannot guarantee the same level of security that a dedicated security system would provide. This is due to the fact that cellular networks rely on unencrypted data transmission, which can be intercepted and manipulated. This makes it difficult to ensure the accuracy and reliability of the security response provided by such a system.

2.7.5 Global Positioning System (GPS)

The most recent advancement in location-based security response technology has been the development of mobile applications. An example is the Global Positioning System (GPS). GPS is a system of satellites that can provide location and time information in any weather conditions. These applications have greatly improved the ease of reporting an emergency and have enabled real-time location information to be shared with the authorities. Some of these applications also allow for real-time communication between the individual and the authorities, which can significantly improve response times and enhance the likelihood of a positive outcome.

GPS receivers are equipped with antennas that receive signals transmitted by GPS satellites in order to determine the location of the receiver. GPS is used to develop location-based security response systems that canaccurately identify and track objects in real-time (Babu et al., 2016). GPS technology is employed in numerous applications, such as navigation, tracking, and location-based security response. This is actually the best approach since virtually all smartphone devices have an in-built GPS system.

The use of Global Positioning System (GPS) technology in providing a location-based security response in a city is an innovative way to ensure the safety of its citizens. The major advantage of utilizing GPS for location- based security is that it can help to quickly identify and respond to security threats in a timely manner. GPS can be used to determine the exact location of a person or vehicle and provide detailed information about the security situation in that particular area. This information can then be used to quickly deploy security personnel to the location where they are most needed. In addition to providing location- based security, GPS can also be used to monitor the activities of people and vehicles in a city. This can be used to identify suspicious behavior and quickly respond to it before it escalates into something more serious. GPS-enabled cameras can also be used to monitor the activities of people and vehicles in the city, providing a more comprehensive security solution.

2.7.6 Peer to Peer Communication

The use of peer-to-peer (P2P) connections can be used to provide a more effective and efficient Location Based Security Response (LBSR). Peer-to-peer (P2P) connections allow for the sharing of information from one device to another without the need for a centralized server (Porras, et al., 2004). With LBSR, this means that the data can be shared in real time.

This is especially important for cities, as it allows for the quick response to any security threats that may arise. Since P2P connections can be established over any network, LBSR can be provided to any part of a city, regardless of its size or population. The use of P2P connections also allows for the more efficient use of resources. For example, by having multiple devices connected to each other in a P2P network, the data can be shared more quickly and efficiently. This means that more resources can be devoted to the actual security response, rather than the transfer of data. By using P2P connections, city officials can also be more proactive in their security response. For example, they can keep track of suspicious activity in a certain area, or even track the movement of a certain individual. This will allow for a more targeted response, which can help to ensure thatany potential problems are dealt with quickly and effectively.

2.8. Impediments to Law Enforcement Utilizing Technology to Respond to Reported Crimes

The advancement of technology has revolutionized the way police officers respond to reported crimes, allowing them to act with increased speed and accuracy. However, there are several impediments that hinder police officers from utilizing technology to respond to reported crimes, including a lack of training, inadequate resources, and obsolete technology.

a. Without adequate training, police officers may be uncertain as to how to effectively utilize the technology or may not be aware of the processes involved with responding to a reported crime. In addition, officers may not be able to recognize when a crime has been reported or understand the protocol for responding to the reported crime.

b. Second, inadequate resources can prevent police from responding to reported crimes via technology. Police departments may lack the necessary funds to acquire the technology, or they may not have the personnel capable of maintaining and operating the technology (Wambugu, 2017). Without the required resources, police departments may be unable to effectively utilize the technology to respond toreported crimes.

c. Third, technology that is not up to date can also impede police from responding to reported crimes via technology. As technology can become outdated quickly, without proper maintenance and updates, it may become ineffectual in responding to reported crimes. This can lead to police officers being unable to properly respond to reported crimes using the outdated technology.

2.9. Exploring Solutions for Enhancing Police Response to Crime Reports Made Through Mobile Applications

Police are a vital component of any community, charged with the responsibility to protect and serve. Yet, if they are not responding to crimes reported through an app by citizens in a city with a high rate of criminal activity, then this is an issue necessitating immediate consideration. To take on this challenge, it is necessary to identify the root cause of why the police are not responding. This could be due to a deficiency of resources, a lack of knowledge, or even inadequate communication between citizens and law enforcement. Once the source has been identified, it is important to take action to resolve the matter.

a. One way to handle this is to raise awareness and provide education about how to use the app, how to report a crime, and how to contact the police if they do not respond (Wambugu, 2017).

Moreover, campaigns could be organized to motivate citizens to report any offenses they observe or become a victim of.

b. Another solution is to increase the resources available to police. According to Wambugu (2017) this could involve increasing the number of police officers in the city, providing them with enhanced training and equipment, and equipping them with the necessary tools and resources to quickly respond to crimes reported via the app.
c. It is essential to facilitate better communication between the police and citizens. This could involve organizing regular meetings between the police and citizens to discuss the issue, as well as establishing a platform for citizens to share their concerns and feedback.

2.10. User Experience and Acceptance of Mobile Applications

The User Experience and Acceptance of Mobile Applications is a critical aspect in the development and implementation of location-based security response mobile applications. A user-centric approach to design and development ensures that the application is intuitive, easy to use, and delivers a positive user experience (ADAMS, 1992). The user experience is a key factor in determining the adoption and continued use of a mobile application.

Several studies have shown that user experience and perceived usefulness are the main factors influencing the adoption of new technologies (Venkatesh & Davis, 2000). For example, the technology acceptance model (TAM) posits that perceived usefulness and perceived ease of use are the key determinants of user acceptance. The model highlights the importance of understanding user attitudes and behavior towards new technologies. Inthe context of location-based security response mobile applications, user experience and acceptance are critical in ensuring the effectiveness and reliability of the system. A positive user experience leads to increased user satisfaction, which in turn leads to higher levels of user engagement and more frequent usage. Moreover, user acceptance of a mobile application is also influenced by its perceived security, privacy, and trust.

User experience and acceptance of mobile applications has been a topic of great interest in the IT industry due to the proliferation of mobile applications, employed daily by millions of people worldwide. The success of any mobile application is largely contingent upon its UX (User Experience) and acceptance. Therefore, it is imperative to comprehend the factors that influence UX and acceptance of mobile applications.

Usability plays an integral role in influencing UX and acceptance of mobile applications. Usability is essentially the application's ease of use, and the intuitiveness it offers users (Weichbroth, 2020). Usability

is affected by factors such as the intuitiveness of the interface, the accessibility of information, the responsiveness of the application to user input, and its speed of use.

User interface design is another key factor that can affects UX and acceptance of mobile applications. The UI (User Interface) should be designed to be intuitive, user-friendly, and aesthetically pleasing. This is essential as it impacts how users interact with the application, and their ability to locate the required information.

Security is an indispensable factor that must be taken into account for any mobile application, particularly for location-based security response applications. Security considerations include data protection, user authentication, malware protection, and data encryption. It is imperative to guarantee that the application is secure and that the user data is safeguarded.

Performance is another factor that influences UX and acceptance of mobile applications. Performance is affected by factors such as the speed of the application, the volume of data that can be processed, and the responsiveness of the application (Hasan et al., 2016). It is necessary to ensure that the application is expeditious and efficient, and delivers a good user experience.

Finally, user acceptance is a factor that affects UX and acceptance of mobile applications. User acceptance is affected by factors such as the ease of use of the application, its usefulness, how well it fulfils user needs, and the effectiveness of the application's marketing (Uğur & Turan, 2019). It is essential to ensure that the application is well-received by users and meets their needs.

2.11. Design and Development Requirements for a Location-Based Security Response MobileApplication.

The design and development requirements for a location-based security response mobile application are essential to ensure the application is efficient and effective. The primary purpose of such an application is to guarantee users that they can promptly report and respond to security incidents. Therefore, particular attention should be paid to the requirements for the UI, backend systems, security, scalability, and availability of the application.

The UI should be designed with ease of use and intuitiveness in mind. It should afford users the capability to quickly report security incidents and access information regarding the incident, such as its type and location. Further, it should provide users with the ability to contact the pertinent security authorities and view their responses. The mobile application should have a user-friendly interface that is attractive, and accessible for all users. The interface should be designed to make it easy for users to quickly find and access the information they need.

The application should be designed to provide real-time alerts and notifications to users, so that they can quickly respond to any security threats. The notifications should be customizable so that users can choose the type and frequency of alerts they receive. The application should use GPS and other location-based services to accurately determine the user's location. This information can then be used to provide relevant security information to the user, including alerts and notifications of any potential security threats. The application should have a reporting feature that allows users to quickly and easily report any security incidents they witness. The reporting feature should be integrated with the user's location data, so that the security response team can quickly respond to the incident. The application should be designed to be scalable, so that it can be used by a large number of users and can easily be adapted to different security needs and environments.

The backend systems must be robust and secure, with a database that allows users to query the data stored, a server that can efficiently handle the requests from the application, and networking components that enable the application to communicate with other systems, such as the security authority's systems. Furthermore, the security of the application should be of paramount importance and must include data encryption, authentication and authorization, and the capability to detect and respond to malicious attacks. In addition, the application should be designed to be scalable and available at all times. This includes being able to handle large numbers of requests and respond in a timely manner, as well as not being affected by network outages or other disruptions.

2.12. Frameworks/Models

Various models are relevant in this study, which include Crime Prevention through Environmental Design, Rational Choice Theory, Social Learning Theory, and Routine Activity Theory.

a. Crime Prevention through Environmental Design - Crime Prevention Through Environmental Design (CPTED) is an established concept aimed at lowering the likelihood of criminal activity by designing the physical environment (Minnery & Lim, 2005). This strategy has been widely accepted by urban planners, architects and law enforcement professionals, as it has proven to be an efficient approach to diminish criminal activities. By employing strategic design of the physical environment, it is possible to prevent or reduce crime. The CPTED approach focuses on four areas: natural surveillance, access control, territoriality,and maintenance. Natural surveillance is the ability to monitor and be observed by others, which can be achieved by designing an environment that permits clear visibility and encourages people to be conscious oftheir surroundings. Access control is the concept of limiting access to an area through the utilization of gates, fences, or other physical obstacles. Territoriality is the notion of generating a feeling of ownership or control over a certain area, usually accomplished through design elements such as landscaping or lighting.

Finally, maintenance is the concept of guaranteeing that the physicalenvironment is kept in good condition, which can also help to create a sense of ownership and deter

criminal activity. When these principles are combined, they can create a setting that is visually appealing and also less vulnerable to crime. CPTED is an effective way to create a safe, secure, andwelcoming environment for all.

b. Rational Choice Theory - This theoretical model proposes that criminals are rational actors who weigh the costs and benefits of committing a crime. Rational choice theory suggests that by increasing the cost of crime (e.g., through increased surveillance, deterrence, and punishment) and reducing the benefits of crime (e.g., through economic incentives), crime rates can be reduced (Green, 2002).
c. Social Learning Theory - This model suggests that individuals learn criminal behavior through their interactions with others and through observing and imitating the behavior of those around them (Akers & Jennings, 2015). Social learning theory suggests that by providing positive social norms and role models, crime rates can be reduced.
d. Routine Activity Theory - This theoretical model proposes that crime occurs when three elements converge: a motivated offender, a suitable target, and the absence of a capable guardian (Miró, 2014). Routine activity theory suggests that by manipulating these elements (e.g., by increasing the presence of capable guardians), crime rates can be reduced.

2.13. Algorithms

Algorithms are a fundamental component in the development of secure systems, including mobile applications. Different approaches, such as clustering, classification, event detection, and response, have been utilized to create algorithms used in location-based security applications. These algorithms have been investigated in a range of contexts, including urban environments and regions with high levels of criminal activity.

Clustering is a widely utilized algorithm in location-based security systems, which involves the grouping of data points based on their similarities. This algorithm has been successfully employed in the past to classify and analyze security-related incidents within a given area (Yuki et al., 2019). In this paper, the application of clustering can be exploited to analyze crime trends in various locales of Nairobi. By leveraging historical crime data, the algorithm can detect areas predisposed to particular types of criminal activity, such as robbery or burglary. This information can then be leveraged to generate location-based security notifications for users of the mobile application. Additionally, clustering can be applied to the data accumulated from the mobile application. By clustering comparable incidents, the algorithm can detect nascent crime patterns which may not be immediately discernible. This data can then be exploited to develop more efficacious strategies for crime prevention.

Classification algorithms have been extensively utilized in lieu of location-based security applications for the purpose of detecting anomalous behavior in real-time data (Yuki et al., 2019). This research proposes the use of such algorithms to identify and classify security threats based on their severity and location. For instance, the algorithm can be trained to recognize events such as armed robbery, burglary, or kidnapping and categorize them in terms of their level of danger. A noteworthy benefit of using classification algorithms is their capacity toacquire knowledge from historical data and create precise forecasts in response to that data. In the case of location-based security, this implies that the algorithm can draw on past events to estimate the likelihood and severity of future events. This information can then be utilized by security personnel or law enforcement agencies to allocate resources and respond more efficiently to security threats. Moreover, classification algorithms can be deployed to automate the decision-making process, thereby reducing the time and effort expended for analyzing and decoding voluminous data. This can be especially useful in the context of location- based security applications where data is generated in real-time and necessitates prompt action.

ToppiReddy et al. (2018) proposed a novel methodology for utilizing crime data to forecast future occurrences and monitor current activities. By constructing a spatial analysis algorithm, the authors were able to detect criminal hot spots and investigate the correlation between crime and its geographic environment. Various data sources, including demographic data from the US Census Bureau and crime data from the FBI, were utilized to implement the algorithm and identify criminal hot spots. Furthermore, the algorithm was used to identify criminal networks and forecast crime trends. The results of the study indicated that the algorithm was successful in anticipating criminal activity and recognizing regions of heightened risk. This approach could be utilized to advise police strategies and aid law enforcement agencies in safeguarding citizens more effectively.

Supervised learning algorithms are employed to create models that learn from labeled data. These algorithms are utilized to identify potential criminal activity by analyzing past data of criminal behavior (Jenga et al., 2023). Predictive policing is one such application of supervised learning algorithms, allowing law enforcement agencies to allocate resources more efficiently by anticipating crime in specific areas. On the other hand, unsupervised learning algorithms are utilized to process data without labels. Through this method, patterns of criminal activity that are not initially visible may be detected. This enables law enforcement agencies to better target their efforts to regions where criminal activity is more probable.

2.14. Existing Applications

Today, Kenya is considered a leading hub for ICT across Africa, especially on innovation. Various software and initiatives have been developed to address the issue of security in the country and across the world. Examples of these software include Ushahidi, Sisi ni Amani, KaaRadah, Najua, Crime Stoppers. Etc.

I. **Ushahidi -** The online system was set in Kenya days after the 2007 national election to help citizens report incidents of violence. The platform allowed citizens to tell their stories via SMS, and social media, and such stories included the geographical location where the incident took place and the type of incident that occurred.

II. **Sisi ni Amani** – This initiative was launched in Kenya in an attempt to improve public security where citizens would report cases of insecurity via SMS. While this technology improved security, it was only utilized for a short period of time around the general election in 2013.

III. **KaaRadah** – The application was developed in memory of the Garissa attack that happened in Kenya, which led to the killing of 147 students from the Garissa University. The term is a Kiswahili word meaning "stay alert" and the application was developed to allow users report any suspicious attacks around them. The app allows users to upload pictures, and also have an access to the Kenya Police contact lines. In addition, it also provides location to the nearest police stations. Unfortunately, not so many people use the app, because the response is usually static and police officers can take a lot of time responding to reported crimes.

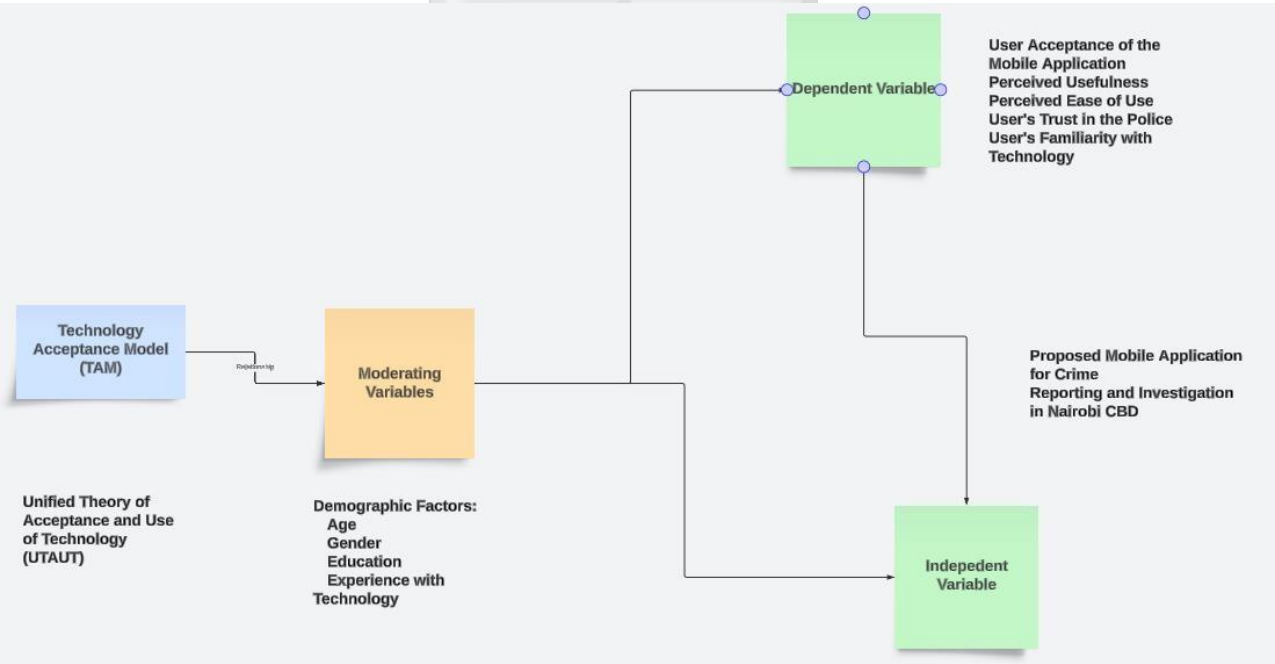
IV. **Najua** – The Kenyan-based system allows users to submit crimes in their regions. The data goes directly to police handsets on the police stations. Basically, the users select the type of crime, i.e., corruption, terror, robbery, takes a picture, describe the crime and the location where the crime is taking place. This approach is effective but it's not reactive, i.e., although the police will react to the reported crime, the crime will already have been committed.

VT OMNES VNVM SINT

2.15. Current Security Response Strategies: Effectiveness and their Challenges

Kumbuti's (2013) study demonstrated the application of technology in crime prevention, with the utilization of walkie-talkies and mobile phones as the primary communication equipment. Further assessment uncovered that police forces have failed to exploit technology to optimize efficacy in crime detection. Walkie-talkies are the most practical solution for the police, although they possess certain limitations such as unreliability of radio channels, proximity issues and disruption of radio signals due to barriers. Mobile phones have allowed police officers to contact each other and request assistance in the event of a crime. However, communication via mobile phones is contingent upon the caller's prepaid credit. This calls for a structural reconfiguration of software and technological hardware at the police force, in order to ensure the prompt transmission of data regarding crime and ultimately enhance the effectiveness of crime prevention and management.

2.16. Conceptual Framework



Orodho (2009) poses that the purpose of a conceptual framework is to explain the correlation between various research variables. These variables can be either dependent or independent; the former is determined by another variable, while the latter regulates the dependent variable. Kaplan (2002) contends that a conceptual framework serves as an expert's opinion on a particular issue, providing guidance to the investigation by presenting a proposed model outlining the concepts and their relationship. According to Mugenda and Mugenda (2012), a conceptual framework allows the reader to comprehend the proposed connection between the dependent and

independent variables. The conceptual framework for this study is based on the Technology Acceptance Model (TAM) and the Unified Theory of Acceptance and Use of Technology (UTAUT). TAM proposes that user acceptance of technology is determined by two key factors: perceived usefulness and perceived ease of use. UTAUT extends TAM by including four additional factors: social influence, facilitating conditions, performance expectancy, and effort expectancy. In this study, the independent variable is the proposed mobile application, which is designed to enhance crime reporting and investigation in Nairobi CBD. The dependent variable is user acceptanceof the mobile application, which is operationalized through the following dimensions: the user's perception of the usefulness and ease of use of the mobile application, the user's trust in the police, and the user's level of familiarity with technology. The moderating variables in this study are demographic factors, such as age, gender, education, and experience with technology. These factors may affect the relationship between the independent and dependent variables.

2.17. The Issue of false alert in a location-based system

Regardless of the immense advantages such an innovation brings to society, there is an issue of false alert which can arise, and this may result in increased crime activities instead of reducing them. This is an incumbent issue which has been addressed by other authors. A security company, Alert Media highlights the noteworthiness of wellness checks, communication templates, event pages, and integrated threat intelligence to ensure the efficiency of emergency notification systems (AlertMedia, 2021). Following a similar approach, this study will integrate various strategies including provide a comprehensive template for rapid response, integrating threat intelligence for accurate alerts, and leveraging emergency communication systems with location-based alerts to differentiate between genuine and false alarms. In addition, every user will be mandated to create an account before reporting a crime, so the template will stipulate consequences of reporting false alerts, which will be severe, and this will hopefully deter such activities.

Chapter 3: Research Methodology

3.1. Introduction

This particular chapter outlines the research methodology in use in this particular research as well as the application development methodology in use with respect to the development of the proposed mobile application. Other key facets such as the data collection source, requirements, user design, development, transition and testing are also explored.

The methodology section also touches on quality assurance and ethical consideration in relation to the research.

3.2. Study Population

This research aims to target potential victims of crime and general members of the public within Kenya, with a particular focus on those located in the Nairobi Central Business District (CBD). A purposive sampling technique will be utilized to acquire a sample size of both victims of crime and law enforcement officers, particularly police officers, who respond to criminal incidences. This non-probability sampling technique is specifically designed to target the area of study. The study population will consist of individuals aged 18 years and above, who reside, work, or spend time in Nairobi CBD, including residents, commuters, and visitors. Additionally, police officers who are involved in responding to criminal incidences within the area will also be included. To ensure that the findings are representative of the population, a diverse range of participants regarding age, gender, occupation, and socioeconomic status will be included. The research will guarantee that the participants' privacy and confidentiality are safeguarded, and their informed consent will be obtained prior tothe commencement of the study. Lastly, no participant will be discriminated against based on their race, ethnicity, religion, or any other personal characteristic.

3.3. Fairness of the Selection and Recruitment Procedures

A non-probability, purposive sampling technique was adopted to ensure fairness in the selection and recruitment of potential respondents for this study. The targeted population comprised of victims of crime and law enforcement officers within the Nairobi CBD area. To identify suitable participants, the researcher coordinated with the police department in Nairobi CBD, as well as utilized online platforms to reach out to potential victims of crime. Inclusion criteria for the study stipulated that the potential respondents had to be at least 18 years of age and had to have been within the Nairobi CBD area at the time of the crime. Prior to providing their consent,

participants were apprised of the purpose of the study, the data collection process, and were further assured of the voluntary and confidential nature of their participation, with the privilege to withdraw from the study at any point without consequences.

3.4. System Development Methodology

According to Cobb (2011), scrum is an agile software development model based on multiple small teams working in an intensive and interdependent manner. Scrum deployment methodology which is a type of agile methodology will be utilized in the development of this mobile application. This is because agile methodology utilizes various iterations to ensure a speedy deployment of a product which can be incremented up to the final release. In addition, scrum also adequately captures the requirements through user stories and epics that are captured from the user by the product owner. The same is communicated to the Scrum Master who collaboratively works with the whole team including the developers to ensure a steady development of the application.

3.5. Requirements Phase

This research aims to collect both functional (business requirements) and non-functional requirements for the development of the application. The functional requirements will be collected from the product owner through user stories from the end user whereas the non-functional requirements will be collated from the technical team including the Quality Assurance team. The functional requirements will be collected through the use of online questionnaires.

3.6. Data Collection and Analysis

The data collected through questionnaires was thoroughly analyzed using descriptive and inferential statistics. An online questionnaire, namely a Google Sheet was utilized, for the data to be efficiently processed through SPSS and Qlik. This further enabled the illustration of the results through graphs and charts for a clear interpretation. Furthermore, Pearson's correlation coefficient was applied to determine the relationship between the variables, while t-tests and ANOVA were employed to test the significance of the differences between groups. Regression analysis was conducted to identify the factors that would influence the probability of a crimevictim submitting a report through a mobile application. Furthermore, content analysis was employed to analyze the open-ended responses to the survey questions, which enabled the recognition of emerging themes and patterns in the data. This approach facilitated us in gaining a deeper insight into the respondents' perceptions and experiences related to the use of a location- based security response mobile application.

3.7. Appropriateness of the research tools

The questionnaire was deemed the most suitable research tool for this study for a variety of reasons. Primarily, it enabled the collection of both quantitative and qualitative data, which is indispensable in comprehending the manifold facets of crime reporting and how the mobile application can be incorporated into the process. Moreover, it was economical and effective in achieving a vast sample size within a limited time-frame. Finally, it allowed for anonymity and confidentiality, which is essential in procuring sincere and precise responses from the study population. Thus, considering its capacity to acquire both quantitative and qualitative data, cost-effectiveness, efficiency, and confidentiality, the questionnaire was elected as the primary research tool for this study.

3.8. Research Design and Development

The following research proposes the development of an android application that uses object-oriented programming as it involves the use of both mainly flutter in the frontend and node.js in the backend. According to Herron (2020), Node.js is an exciting new platform for developing web applications, application servers, any sort of network server or client, and general-purpose programming. It is designed for extreme scalability in networked applications through an ingenious combination of server-side JavaScript, asynchronous I/O, and asynchronous programming. Node.js was used to create the backend of the system. Postgres database was used as the database management system to handle storing and processing of data as requested by the user. Geo-location has been effected through Google map API's which has been incorporated to handle real-time sharing of data.

3.9. System Testing

The application sanity will be tested through the use of various rounds of technical tests to ensure that the mobile application does not have any bugs. Debugging of the code will be done to ensure that the final compilation is seamless. In addition, functional testing will be done by super users to ensure the functionalityof the application as per the functional (business) requirements.

3.10. Risk of this Research

Privacy risks: The development and implementation of a location-based security response mobile application poses a possible risk of privacy infringement. This threat is derived from the collection and storage of personal data, such as location information, which may be utilized for purposes beyond crime reporting and response. Tocounteract this risk, it is essential to put in place explicit privacy policies and data protection measures.

Security risks: The use of a Location-Based Security Response Mobile Application (LBSRA) presents a potential risk of security vulnerabilities. If appropriate security measures are not taken, it can be exposed to malicious hacking and compromise of sensitive personal data. To mitigate such risks, the application should be developed with robust security protocols such as encryption, authentication, and access control. The implementation of these measures ought to be carefully considered and the app should be regularly monitored forany suspicious activity.

Accuracy risks: A location-based security response mobile application is dependent on precise location data for its proper functioning. However, due to poor GPS coverage in certain areas, this data can be inaccurate, leading to potential peril for users if the application is developed with a reliance on this data. To avoid any untoward consequences, the application should be designed with redundancy measures and additional reporting methods as a guard against any potential inaccuracies.

Social risks: The potential for social ramifications associated with the deployment of a location-based security response mobile application warrants consideration. To limit the potential for negative repercussions, it is critical that the application be developed with an awareness of the range of cultural and social contexts that could be affected. Moreover, the application should not be based on suppositions or stereotypes regarding crime or safety.

3.11. Research Quality

To ensure an enhanced research quality, the research was reviewed among peers as well as being submitted to the assigned supervisor for guidance and necessary recommendations for amendment or enhancement.

3.12. Risk/Benefit Analysis

The potential advantages of a location-based security response mobile application are manifold. By leveraging the power of mobile technology, the police department can achieve faster response times and improved accuracy when locating crime scenes. Moreover, the application's ability to allow for convenient and accessible reporting of criminal acts can facilitate a rise in the number of reported crimes, thus enhancing public safety. Moreover, the real-time information provided by the application can help users make more informed decisions about their safety. All of these factors can contribute to an increased sense of security and safety in the community.

However, there exists a couple of risks. First, the development and implementation of a location-based security response mobile application poses a possible risk of privacy infringement. This threat is derived from the collection and storage of personal data, such as location information, which may be utilized for purposes beyond crime reporting and response. To counteract this risk, it is essential to put in place explicit privacy policies and data protection measures.

Second, the use of a Location-Based Security Response Mobile Application (LBSRA) presents a potential risk of security vulnerabilities. If appropriate security measures are not taken, it can be exposed to malicious hacking and compromise of sensitive personal data. To mitigate such risks, the application should be developedwith

robust security protocols such as encryption, authentication, and access control. The implementation of these measures ought to be carefully considered and the app should be regularly monitored for any suspicious activity. Third, a location-based security response mobile application is dependent on precise location data for its proper functioning. However, due to poor GPS coverage in certain areas, this data can be inaccurate, leading to potential peril for users if the application is developed with a reliance on this data. To avoid any untoward consequences, the application should be designed with redundancy measures and additional reporting methods asa guard against any potential inaccuracies.

Fourth, the potential for social ramifications associated with the deployment of a location-based security response mobile application warrants consideration. To limit the potential for negative repercussions, it is critical that the application be developed with an awareness of the range of cultural and social contexts that could be affected. Moreover, the application should not be based on suppositions or stereotypes regardingcrime or safety.

3.13. Dissemination and Utilization of Results

Developing a dissemination plan ensures that research findings are communicated to pertinent stakeholders, including the academic milieu, policymakers, and the general public (Rodgers, 1998). Furthermore, a utilization plan serves to illustrate how the research results will be utilized practically, such as contributing to policy decisions or creating interventions.

The results of this study will be communicated to a broad audience, including scholars, policymakers, law enforcement organizations and the general public, via multiple channels. To begin with, our findings will be published in peer-reviewed academic journals and presented at related conferences and symposiums in the domain of criminology and technology, with the aim of making them available to other researchers, thus fostering the growth of crime prevention and response strategies. Subsequently, our research will be shared with pertinent stakeholders such as law enforcement authorities, victim assistance programs and governmental bodies responsible for public welfare. This will be done through tailored presentations, workshops and reports, which will be catered to their individual necessities and interests.

Ultimately, our aim is to disseminate our research findings to the general public through various media outlets, including social media, press releases, and community outreach programs. This will serve to increase the public's awareness of the potential benefits of using mobile applications for crime reporting and response, and motivate them to interact with these technologies.

In terms of practical implications, our research is intended to inform the development of mobile applications for crime reporting and response, as well as improve existing practices. We hope that our findings will have a substantial impact on policy and decision-making in the area of crime prevention and response, and help shape public perceptions and attitudes towards these technologies.

3.14. Ethical Considerations

In regard to maintaining the standard code of ethics, the researcher used citations to reference context adequately captured in this research thus ensuring compliance of the highest level of ethics. In addition, according to the set procedures and guidelines set by Strathmore University, the researcher submitted his research to the Strathmore University Institutional Ethics and Review Committee for review and validation against ethics compliance.

VT OMNES VNVM SINT

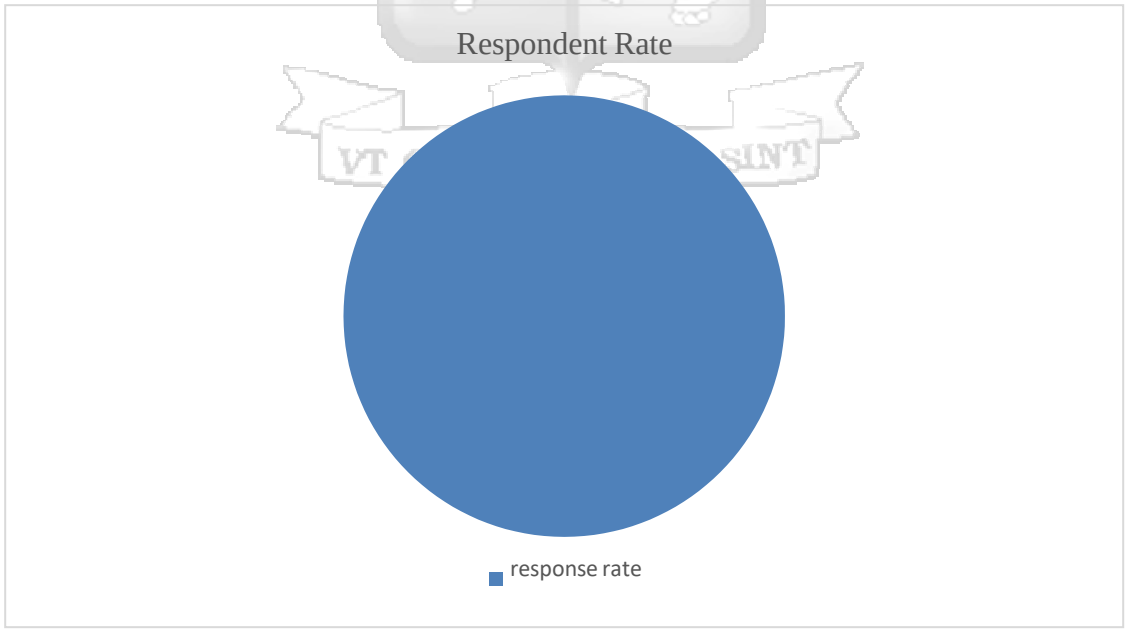
Chapter 4: Data Analysis

4.1. Introduction

This chapter provides an analysis of the research data, research findings, and their interpretation. This section will comprehensively understand the collected data and its implications. The research findings are conveyed using a combination of frequency tables, charts, and graphs to visualize the results. Furthermore, exploring the respondents' data and its alignment with the research objectives is undertaken to offer insights and explanations regarding the observed outcomes.

4.2. Response Data and Rate

A questionnaire was provided to various law enforcement offices and potential victims in Nairobi. The questionnaire was administered online via google forms, and respondents were required to respond to the questionnaires and submit their responses online. Since a purposive sampling technique was employed, the expected respondent number was at least 50 crime victims and 50 law enforcement officers. 12 questions were administered to law enforcement officers and Nairobi Metro Residents, from which 53 officers and 53 residents responded, translating to a 100% response rate. Curtis and Keeves (2000) regarded "an overall response rate forthe Course Experience Questionnaire (CEQ)" of at least 70% to be achievable and desirable which was achieved inthis research.



Crime Victims

4.3. Owing a mobile device and the type of OS on the device

Mobile phone adoption has brought about a significant revolution in governance and crime prevention. The widespread use of mobile phones has transformed how governments operate and provided new opportunities for enhancing public safety. The study aimed to establish the extent to which respondents (both law enforcement officers and Nairobi residents) owned a mobile phone and whether owning the device affected reporting or responding to reported crimes in Nairobi. The study found that most respondents (Nairobi resident at 100% and law enforcement at 98%) owned a mobile phone.

			Frequency	Valid Percent
	Valid	Yes	50	100.0

Law Enforcement

			Frequency	Percent
	Valid	Blank	1	1.9
		Yes	51	98.1
		Total	52	100.0

Nairobi CBD Residents

The type of operating system on mobile devices plays a role in shaping how residents report crimes and the likelihood of law enforcement officers responding to those reports. The study reveals important insights into the prevalence of different operating systems among law enforcement officers and respondents, namely Android and iOS.

Most law enforcement officers (90%) indicated ownership of Android devices. This finding suggests a higher adoption rate of Android-based mobile devices within the law enforcement community. The prevalence of Android devices among officers could be attributed to various factors, such as affordability, availability, and the specific features and functionalities offered by Android devices that cater to their professional needs.

In contrast, a small percentage (2%) of law enforcement officers reported owning devices with other operating systems. This could include devices operating on operating systems such as Windows, BlackBerry, or other less common platforms. The limited ownership of these alternative operating systems among law enforcement

officers could be due to compatibility, personal preferences, or organizational restrictions.

Turning to the Nairobi residents, the study found that the majority (90.4%) owned Android devices. This aligns with the prevalence of Android devices among law enforcement officers and reflects the widespread use of Android-based smartphones within the community. Android's popularity could be attributed to affordability, a wide range of device options, and the open-source nature of the Android operating system that allows for customization and flexibility.

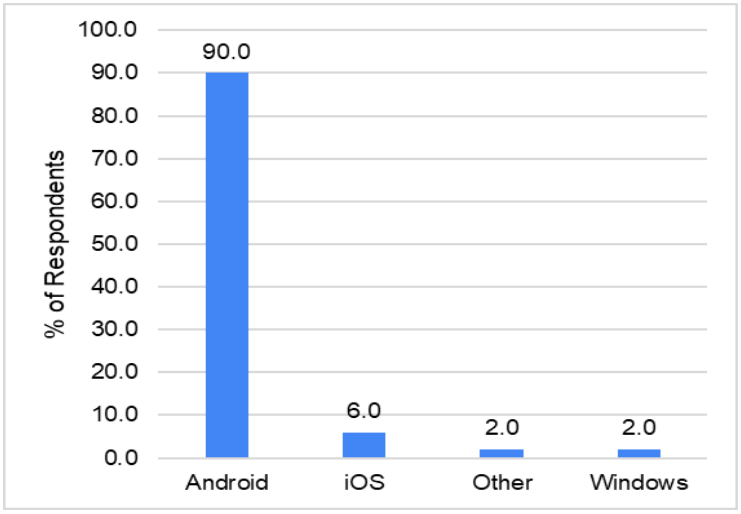
While Android devices dominate among residents, a smaller percentage (9%) indicated ownership of iOS devices. iOS, the operating system used by Apple devices, is recognized for its user-friendly interface, strong security measures, and integration with other Apple products and services. The presence of iOS devices among respondents indicates a notable user base within the community who prefer or have access to Apple devices.

The study also revealed that only a tiny percentage (0.06%) of respondents owned devices with other operating systems. This suggests that alternative operating systems have limited penetration within the studied population, with Android and iOS being the primary choices for mobile devices.

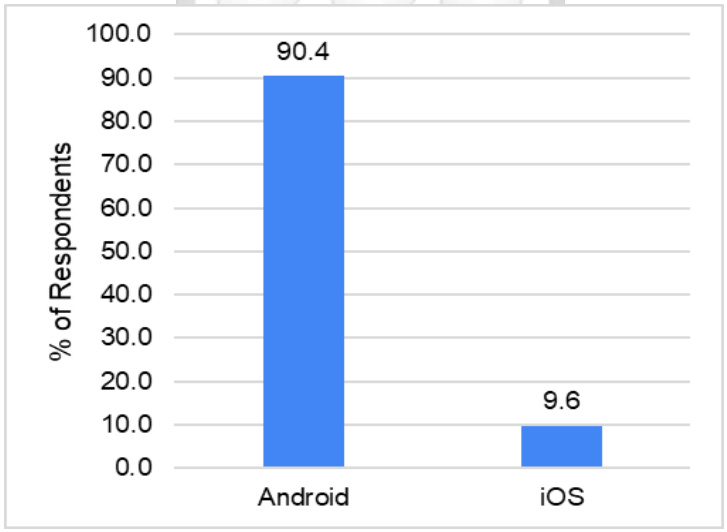
The findings regarding the prevalence of Android and iOS devices among law enforcement officers and respondents provide valuable insights into the technological landscape and preferences within the community. Understanding the distribution of operating systems can inform the development and optimization of mobile applications and crime reporting platforms, ensuring compatibility, accessibility, and tailored user experiences across the dominant operating systems.

The study's hypotheses sought to understand if the choice of operating system impacts a victim's ability to report a crime. Analyzing correlations, a weak negative correlation (Pearson Correlation: -0.247) surfaced. This implies that as the preference for one OS swells, the ability to report a crime moves in the opposite direction. Though this weak correlation was not statistically significant ($p > 0.05$), implying that other factors might dictate a victim's ability to report a crime.

The insignificant impact was further validated by the conducted ANOVA ($F=3.262$, p-value = 0.077), failing to refute the null hypothesis: "The OS used by the crime victim does not affect their ability to report a crime." These results thus imply that factors apart from the OS a victim uses may affect their ability to report crimes. It opens doors for future research to delve into these unidentified influential factors.



Law Enforcement



Nairobi CBD Resident

4.4. Modes of crime reporting that people are aware of within Nairobi CBD?

The assessment of various modes of crime reporting, including mobile applications, social media, television, telephone calls, and physical visits to the police station, provides valuable insights into the current landscape of reporting practices. Specific reporting modes are more well-known and widely used than others among law enforcement officers and Nairobi residents.

Among law enforcement officers, the mode of reporting that stands out as the most known is 'physical reporting to the police station,' with a remarkable 100% awareness. This finding suggests that law enforcement officers

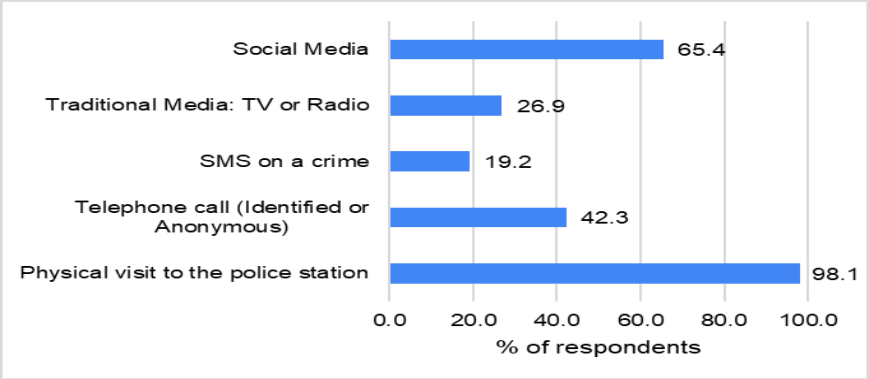
are well-acquainted with traditional reporting methods, where individuals visit the police station to report crimes directly. This could be attributed to the long-standing practice of physical reporting and officers' familiarity with handling such cases.

Telephone calls also emerge as a widely recognized reporting mode among law enforcement officers, with 92% indicating awareness. Telephone calls have been a standard method of crime reporting for many years, and officers are still quite familiar with this means of communication.

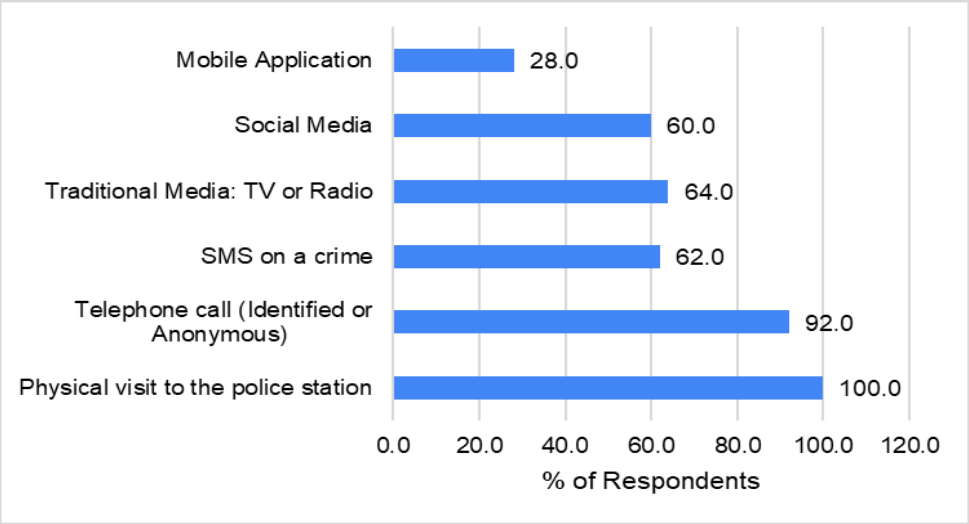
Interestingly, while traditional media, such as television, is known to a considerable extent by 64% of officers, other modern forms of reporting, such as mobile applications and social media, show lower levels of awareness. Only 28% of officers are familiar with mobile application reporting, and 60% are aware of reporting through social media. This finding suggests that there may be room for increased awareness and training on the benefits and functionalities of mobile reporting. Shifting the focus to Nairobi residents, the mode of 'physical reporting to the police station' also stands out as the most known, with a high awareness rate of 98.1%. This aligns with the findings among law enforcement officers and reflects the enduring popularity of visiting the police station to reportcrimes among the general population.

Social media is Nairobi residents' second most known reporting mode, with 65.4% indicating awareness. The relatively high percentage suggests that social media platforms are becoming increasingly recognized as a means of reporting crimes within the community.

Telephone calls are also well-known among Nairobi residents, with 42.3% indicating awareness, followed by traditional media at 26.9%. However, similar to the findings among law enforcement officers, mobile application reporting and SMS-based reporting show relatively lower levels of awareness at 19.2% and 19.2%, respectively.



Nairobi CBD Residents



Law Enforcement Officer

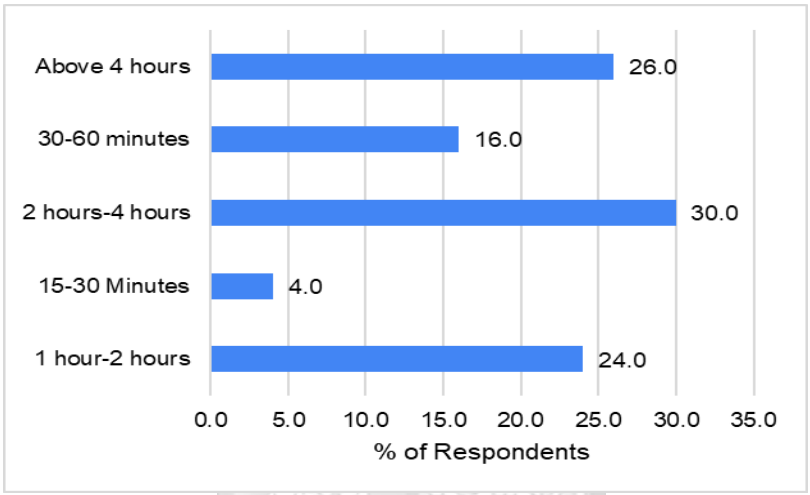
4.5. Reporting and responding to Criminal Emergencies

The rate of crime reporting among citizens was above average, where 65.3% of crime victims indicated that they reported a criminal emergency. In comparison, 36.5% indicated they have never responded to a crime in Nairobi. On average, 30% (majority of the respondents) of the law enforcement officers responded to a crime in 1-2 hrs, 26% responded above 4hrs, 24% responded in 1-2 hrs, and only 4% responded within 15 – 30 minutes, which is a meager response rate. On average, 30% of the respondents who reported a crime received a response/assistance in 4hrs and more, 3% received a response in 15-30 minutes, 12.1% in 1-2hrs, 3% in 0-15 minutes, 18.2% in 30 -60 minutes, and 12.1% victims indicated that they never received a response/ assistance from the law enforcement. Investigating the impact of time taken to receive assistance on the victims' willingness to report a crime via a mobile application, the null hypothesis presumed that the assistance response time does not influence thewillingness to use a mobile application for crime reporting. The alternative hypothesis, on the contrary, proposed that it does.

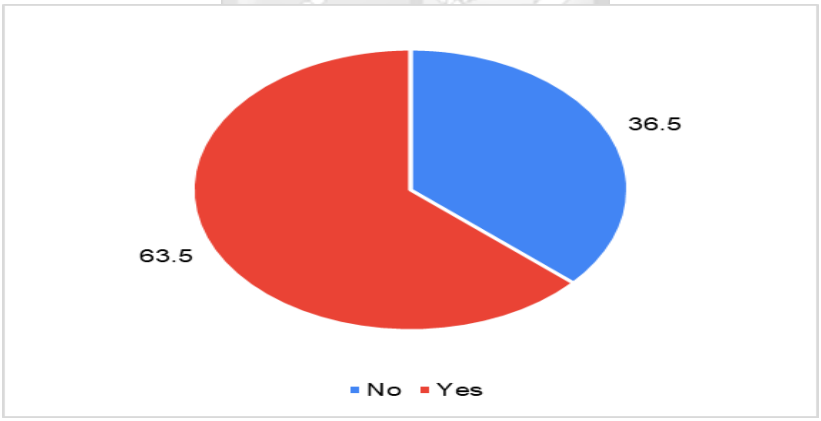
Correlation analysis showed a weak positive correlation (Pearson Correlation: 0.296) between the duration it took a victim to receive assistance after reporting a crime and their willingness to report a crime through a mobile application. A positive correlation in this context indicates a tendency that, as the response time increases, so does the willingness to utilize a mobile application for crime reporting. However, this correlation wasn't substantial, and it wasn't statistically significant (p-value: 0.084 > 0.05), implying that other factors may have a more significant role in this regard.

Similar conclusions were drawn from the ANOVA, where the impact of the time taken to receive assistance on the willingness to report a crime using a mobile application was not statistically significant (F = 3.171, p-value:

0.084). Given the results of both tests, we maintained the null hypothesis concluding, "The time taken for the crime victims to receive assistance after reporting a crime does not influence their willingness to report a crime using a mobile application." Hence, it's implied that factors other than the time taken to receive assistance upon reporting a crime, influence the willingness to adopt mobile applications for crime reporting. Future research can focus on understanding these alternative factors in greater detail.



Reporting Criminal Emergencies



Responding to Criminal Emergencies

4.6. Willingness to report a Crime and willingness to respond to a Crime via a Mobile Application.

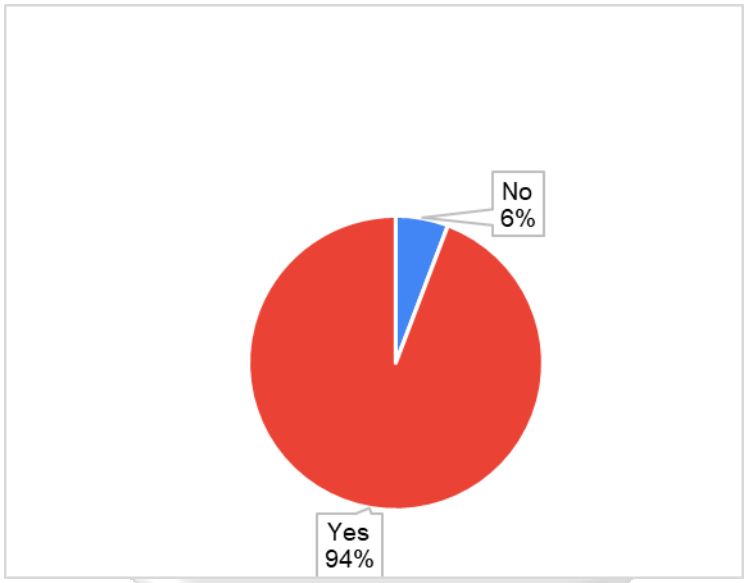
A high number of crime victims (94%) indicated that they would be willing to report a crime via a mobile application, and only 6% indicated that they would be not willing to report a crime via a mobile application. All the law enforcement officers indicated they would be willing to respond to crimes reported via a mobile application. The high willingness of crime victims to report crimes via a mobile application signifies the potential of such technology in enhancing crime reporting. This positive response suggests that victims recognize the convenience, accessibility, and potential benefits mobile applications offer for reporting crimes.

The inferential statistical analysis aimed to understand the relationship between the time taken for crime victims to receive support after reporting a crime and their willingness to report crimes through mobile applications.

The correlation analysis illustrated a weak positive correlation (Pearson Correlation: 0.296, p-value: 0.084, N = 35). Here, a positive correlation suggests that as the time taken to offer support or assistance increases, so does the victim's willingness to report a crime using a mobile application. However, the correlation was weak and not statistically significant (p-value > 0.05), indicating that other factors might have a more substantial influence on the victim's decision to use the mobile application for crime reporting.

The Analysis of Variance (ANOVA) concurred with the conclusion drawn from the correlation analysis. The influence of support timing on the willingness to report a crime through a mobile application was found not to be statistically significant (F=3.171, p-value = 0.084).

Given the results of both analyses, the hypothesis stating "the time to receive assistance following a crime report does not impact the willingness to report a crime via a mobile application" cannot be rejected. This signals that the willingness to report crimes using mobile applications may be influenced by factors other than the time taken to receive assistance post-reporting. Such factors warrant more detailed exploration in future research.



Willingness to respond to a Crime

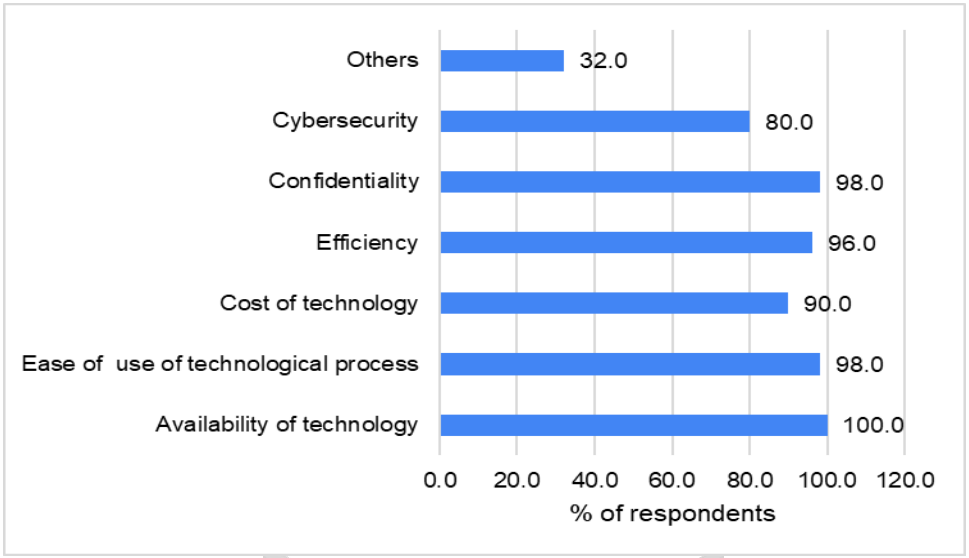
4.1 Factors to consider when reporting and responding to an ongoing Criminal Emergency through Technology

The responses obtained from law enforcement officers and Nairobi residents shed light on the factors they prioritize when reporting crimes through a mobile application. Law enforcement officers unanimously identified the availability of technology as the leading factor when responding to a crime rate using a mobile application. This highlights the critical role that technology accessibility plays in facilitating their response efforts. The high percentage of officers considering availability indicates a firm reliance on the presence and reliability of mobile technology for effective crime response.

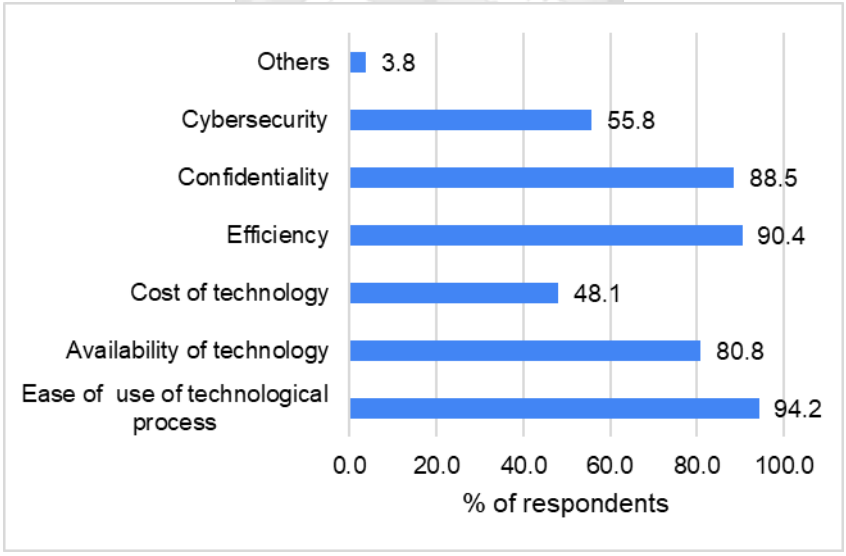
Furthermore, when examining the average responses across all factors, it becomes evident that the officers consistently prioritize specific considerations. Confidentiality, efficiency, cost of technology and ease of use of the technology process were identified by at least 90% of the officers as essential factors in their decision- making process. This emphasizes the significance of streamlined and user-friendly technology solutions that ensure efficient communication and reporting while maintaining the confidentiality of sensitive information. The officers'focus on these factors reflects their desire for an adequate, streamlined technological infrastructure to aid their crime response efforts.

Remarkably, cybersecurity received less consideration from law enforcement officers, with only 80% indicating its importance. This finding suggests that while officers recognize the significance of cybersecurity in crime reporting through mobile applications, there may be room for improvement in prioritizing and

implementing robust cybersecurity measures. Given the sensitive nature of crime-related information, addressing potential vulnerabilities and ensuring the integrity and security of data transmitted through these platforms is crucial.
Shifting the focus to Nairobi residents, the leading factor for residents was the ease of use of the technology process, with a substantial 94.2% indicating its importance. This underscores the importance of user-friendly interfaces and intuitive design in mobile crime reporting applications. Technology solutions can empower residents to report crimes promptly and accurately by prioritizing ease of use, leading to more effective law enforcement responses.
Efficiency was the second most considered factor among Nairobi residents, with 90% emphasizing its significance. This finding aligns with the desire for swift and effective crime response, indicating that residents value prompt action from law enforcement agencies when reporting criminal emergencies. Confidentiality and theavailability of technology were also highly considered factors, at 80.8%. This suggests that residents prioritize the availability of mobile technology infrastructure to ensure widespread access and convenience when reporting crimes.
However, it is noteworthy that only 55.8% of Nairobi residents considered cybersecurity as a factor when reporting criminal emergencies through technology. This finding suggests a need for increased awareness and education regarding the importance of cybersecurity in protecting sensitive information and ensuring the privacy ofcrime reports.
Investigating the correlation between a crime victim's operating system (OS) and their ability to report a crime, the data demonstrated a weak negative correlation (Pearson Correlation: -0.247, p-value: 0.077, N = 52). A negative correlation suggests that as one value increases, the other decreases. Despite this, the correlation was not significant ($p > 0.05$), indicating that factors other than the crime victim's OS might hold more sway over their ability to report crimes.
Statistical insignificance was also observed in the Analysis of Variance (ANOVA) ($F=3.262$, p-value = 0.077). Given both tests' results, we retain the null hypothesis, concluding: "The operating system used by the crime victim does not affect their ability to report a crime." Therefore, other factors influencing the victims' ability to report crimes using a mobile application warrant further exploration.



Law Enforcement Officer



Nairobi CBD Residents

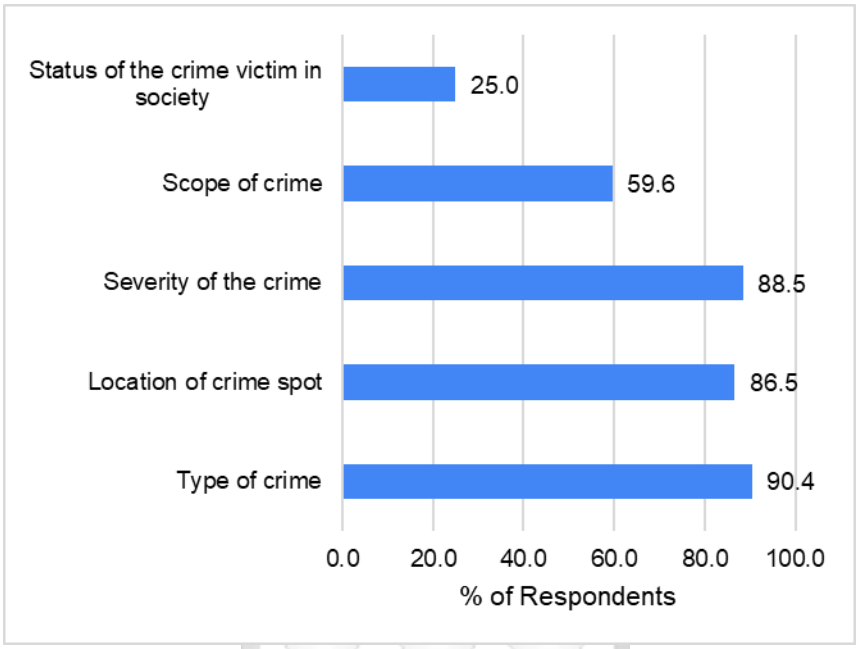
4.7. Factors to consider to rapidly respond to a criminal emergency that a crime victim reports

All the law enforcement officers who responded to the questionnaire indicated that factors that should be most considered to respond rapidly to a criminal emergency using a mobile application include the type of crime, location of the crime spot, severity of the crime, and scope. At least 80% of the law officers indicated that the status of the crime should also be considered, and only 42% indicated that other factors should be considered. Different types of crimes require varying levels of urgency and response. The location of the crime spot is crucial information for law enforcement officers to respond rapidly. Crimes that threaten life, such as physical violence or an ongoing dangerous situation, require a swift and decisive response. Understanding the scope of a reported crimehelps law enforcement officers determine the resources required for an effective response. The null hypothesis tested was that the rank of the police officers has no significant impact on the response time to a crime, whereas the alternative hypothesis proposed that their rank influences the response time to a crime.

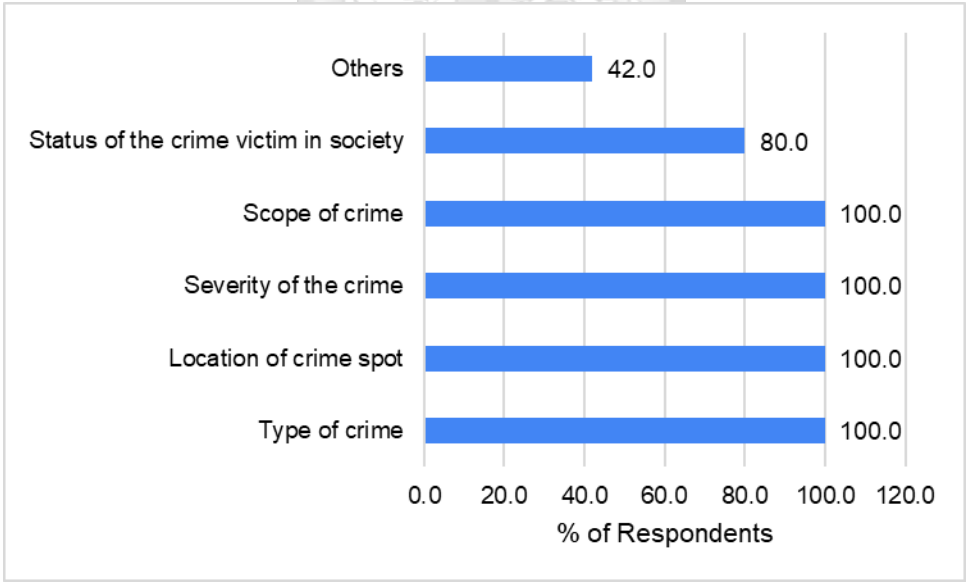
Correlational analysis demonstrated a weak positive correlation between the officers' ranks and their response times to crimes (Pearson Correlation = 0.248, p-value = 0.093, N = 47). A positive correlation indicates that as therank of an officer increases, the response time to a crime also tendentially increases. However, the correlation was weak, hinting that other factors might exert a more substantial influence on the response time. Moreover, the correlation was not statistically significant (p-value > 0.05), providing limited support for the alternative hypothesis.

Further support for the null hypothesis was provided by ANOVA analysis, which again demonstrated that the influence of the officer rank on the response time to a crime was not statistically significant (F = 2.940, p-value = 0.093).

Thus, the null hypothesis that "police officers' ranks do not affect the response time to a crime" could not be rejected based on these results. This implies that factors other than the police officer’s rank need to be considered when aiming to reduce the response time to reported crimes.



Nairobi CBD Resident



Law Enforcement Officer

Chapter 5: System Design, Implementation and Testing

5.1. Introduction

This chapter explores the design, creation, and testing phases of the Geolocation Security Alert Mobile Application. It details the different steps and procedures taken to make the app a reality. The process of building this application included stages such as crafting the initial concept, designing the app, developing it into aworking system, and conducting tests to ensure it functioned as planned. At each stage, we followed specific standards to create an app that is both efficient and user-friendly.

First, we'll discuss the app's design, detailing how it's built and how its database is organized. Then we'll describe how we implemented the system, highlighting which programming languages and tools were used to build and operate the app. Finally, we'll dive into the testing phase, when the app was tested thoroughly to check its reliability, performance, and ease of use. We paid particular attention to the security features of the app to protect users' sensitive data, such as location details and personal information.

In the sections that follow, we'll shed more light on various features of the system that make it a reliable tool for reporting crime and providing real-time response options for law enforcement officials. Understanding these design elements, implementation processes, and testing standards can help users and future developers understand the app's capabilities and potential to improve security in Nairobi's Central Business District.

5.2. System Architecture and Design

5.2.1 System Overview

The Location Based Security Response Mobile Application is fundamentally designed to provide an accessible, effortless crime reporting system that enhances real-time response for law enforcement agencies. The system's architecture is centered around delivering user-friendly interfaces for two main types of users - ordinary users (affected victim) and law enforcement officials. Each user type has unique access permissions and functionalities aimed at simplifying crime reporting and resolution processes.

The platform offers a secure channel for ordinary users to swiftly and easily report crimes. The application comes with integrated location services, enabling accurate geotagging of crime incidents, thus facilitating a more effective response by officers. In addition, users have access to a continuous record of their reported cases, enabling them to revisit their reports when necessary.

Conversely, for law enforcement officers, the application provides real-time access to reported crimes, allowing them to monitor case status and oversee the entire incident resolution phase.

The backend of the application revolves around a Node.js server, which oversees API management and facilitates the connection between the application's frontend and the PostgreSQL database. The Node.js and PostgreSQL

containers independently operate on the Docker host machine, communicating through a Docker Network established between them to ensure smooth data management and exchange.

Subsequent sections will delve deeper into the architectural details of the system, illuminating the intricate design choices and demonstrating how the various components are harmonized to deliver a seamless user experience.

5.2.2 System Architecture

The envisaged user interaction flow is as follows:

- Users enter the mobile application and based on their role, are directed to the appropriate user interface customized for their specific tasks.
- Users submit information following the step-by-step guide. For instance, ordinary users may detail a crime in progress while law enforcement officers may accept incoming crime reports.
- Upon submission, the information reaches the Application Logic Layer, consisting of the Flutter-built app, where data is processed depending upon the user's action. If an ordinary user reports a crime, the app's logic interprets this data and saves it in the database.
- Conversely, if an officer accepts an incoming crime report, the application identifies this action, assigns the report to said officer and subsequently updates the report's status in the database.
- The Data Access Layer oversees all these interactions, functioning as a channel facilitating smooth communication between the application's logic and its database. All data transactions, including storage, retrieval, and updates, occur through this layer. Each piece of information sent to the database is conserved in the PostgreSQL container, where it is securely stored and can be accessed when needed.

5.2.3 User Interface and Experience Design

Creating an intuitive user interface was an integral part of developing the Location Based Security Response Mobile Application. Ensuring a positive user experience sets the tone for user interaction and aids the effectiveness of the platform. Each user role has unique needs, and the interface design was done considering these distinct requirements.

5.2.4 User Role: Normal User

The interface for normal users is designed to be straightforward and easy to use, focusing on delivering a seamless user experience even during intense situations such as reporting a crime.

Login and Sign-Up Process

The registration process for regular users starts when the user selects the 'Sign Up' option on the login screen. The

user needs to provide an email address, which is subsequently verified through a One-Time Password (OTP) sent to the user. Once the email has been verified, the user has to provide their phone number, also verified by a distinct OTP. This dual verification procedure ensures the security and authenticity of the user's contact information. Following this, the user needs to enter personal details, including First Name, Middle Name, Last Name, and ID Number. Next, the user has to create a password that aligns with the provided security standards to ensure its strength, such as a minimum length and a combination of upper- and lower-case letters, as well as numbers and symbols. Users must re-enter the password in the 'Confirm Password' field to ensure accuracy.

After all details are successfully filled and verified, the user chooses 'Normal User' as their role and clicks on the 'Submit' button to finalize the registration process.

For users returning to the app, there is a 'Login' option. They are required to enter the email address and password registered during sign-up to access the application. In case of forgotten credentials, there is an option to reset the password using the registered email. The login and sign-up interface is designed to be straightforward and swift, allowing users to engage with the app without unnecessary delay, which is a critical feature for an application designed to rapidly report crimes.

User Front Page & Service Tab

The User Front Page and Service Tab form the cornerstone of our normal user's interaction with the Location Based Security Response Mobile Application.

User Front Page

After successful login, users are welcomed with the User Home Page, a clear and focused portal designed to offer immediate access to the application's primary feature: reporting a crime. There is a prominently displayed and easily recognizable button for users to quickly proceed to the crime reporting form.

The form is designed to be simple and intuitive, requesting key details like crime type, exact location (which can either be auto-detected or manually entered), and an optional section for additional incident details. The swift and efficient method of reporting a crime caters to the immediacy usual in such situations, thereby promoting prompt reporting. Once a report has been submitted, users can track the status of their reports in real-time directly from the home page, which emphasizes the application's dedication to transparency and ongoing communication.

Service Tab

The Service Tab has been designed to serve as the user’s personal command center over their reported incidents. Here, users can track the progression of their submitted crime reports in chronological order. The list is neatly categorized by incident status - whether 'Reported,' 'In-Progress,' or 'Resolved'.

Clicking on an individual report opens up a detailed view of the crime report. This includes granular details such as responding law enforcement officer's name, real-time whereabouts (if the officer is en route), progress updates when available, and ultimate resolution details. This feature underscores the app's commitment to accountability and visibility, ensuring users are well-informed about the status and progress of their reports. Together, the User Front Page and Service Tab represent the cornerstone of the application's user experience, driving real-time, actionable, transparent reporting and tracking of criminal activities.

Service History

Service History provides users with a comprehensive and traceable record of their past crime reports. It maintains an organized, chronological catalog of the user's reports, showing the most recently concluded reports at the very top to facilitate ease of access. Each corresponding entry is more than just a brief overview of the type, date, and location of the crime reported. When selected, it unfolds to a detailed page carrying minute specifics of the case
- initial descriptions, updates provided by the user, action steps by the responding officer, and the final resolution status. Service History offers a complete audit trail of every crime reporting activity undertaken by the user, fortifying transparency and responsibility while enabling users to track and retrieve their reporting history whenever needed.

User Account Settings

The User Account Settings is where users can manage their personal and app-usage information. It offers several customization options to enhance user experience and ensure control over personal data. From updating basic profile details, such as name and email address, to changing a password, the control remains with the user. Advanced settings let users define their interaction with the app: Enable or disable location services to determine thelevel of location accuracy while reporting a crime or manage notification preferences to get alerts and updates aboutthe status of their reports.

Importantly, the User Account Settings also houses an option to delete the user account for users who wish to discontinue using the app. A prompt confirmation is required to validate their decision, ensuring it's intentional. Upon confirmation, the app permanently erases their data, respecting user decisions pertaining to data privacy and security. The User Account Settings takes personalization and data control to the next level, offering a versatile setting platform that aligns the app usage closely with users' preferences, thus bolstering the user-centric approach of the Location Based Security Response Mobile Application.

5.2.5 User Role: Law Enforcement Officer

5.2.5.1 Login and Sign-Up Process

The sign-up and login processes for law enforcement officers have been designed to be robust, secure, and straightforward. Similar to normal users, in the sign-up process, officers must provide both an email address and a phone number which are then validated through a two-step OTP (One-Time Password) verification process for security and contact validity. Post successful verification, officers must fill in their personal details, including First, Middle and Last names, as well as their official ID number. Subsequently, they're required to set a secure password for their account and confirm it by re-entering in the next step.

The main difference in the officer's sign-up process is the selection of the user role. They must select 'Law Officer' before proceeding. This categorizes them in the system's database as law enforcement officers, giving them access to the specific departmental interface and functionalities upon login. For returning officers, the app provides a simple login form asking for their registered email or username and their set password. The system offers a password reset option via the registered email if they forget their login credentials. This careful blend of simple steps and security measures ensures that the sign-up and login processes are accessible yet secure for our dedicatedlaw enforcement users.

Law Officer Dashboard

The Law Officer Dashboard is the central hub of activity for the law enforcement officers on the application. Thoughtfully designed to cater to the unique needs of officers, it serves as their go-to platform to monitor reported crimes and manage responses.

Right on the main screen of the dashboard, officers are presented with real-time crime reports submitted by users. Each report on the dashboard features crucial details about the incident such as the type of crime, location, and the time it was reported, supplementing officers with information necessary for their initial assessment.

Moreover, officers can view the status of each report ranging from 'Reported,' 'In Progress,' to 'Resolved.' They can accept incoming crime reports directly from the dashboard, and upon acceptance, these reports move into the 'In Progress' category, enabling officers to track the reports they're actively working on. Officers can also mark reports as 'Resolved' once the case is concluded, assisting in efficient management of their caseload.

One unique feature of the Law Officer Dashboard is the direct communication line it establishes between officers and the reporting users. Officers can view additional information provided by the users, communicate updates or ask questions through a dedicated in-app chat, and alert the user when they've arrived at the crime location. This interactive feature significantly improves incident response and resolution times, enhancing the overall efficiency of the system. The Officer Dashboard configures a multitude of functionalities to fit within an easy- to-use interface, making crime report management more efficient and effective for our law enforcement officers.

Crime History Review

The Crime History Review module is an essential feature for law enforcement officers using the Location Based Security Response Mobile Application. It serves essentially as a digital archive of all the crime reports that the officer has responded to in the past, neatly organized in chronological order from the most recent to the oldest.

This feature maintains a record of each crime's specifics, such as the type of crime, exact location, its status, and the date and time of report and resolution. Upon selecting a specific entry from the list, officers can view a detailed page delineating every critical aspect of the incident. Moreover, this detailed view provides a historical trail of status changes, officer comments, responses, and any additional notes made during the incident's resolution. This includes the direct chat history with the reporting user, enabling officers to retrospectively review their communication, the reported details, and the actions taken.

The Crime History module aids officers in revisiting previously resolved cases for reference or investigative purposes in the future. By having seamless access to their professional history of crime resolution right at their fingertips, law enforcement officers can enhance their decision-making, improve response strategies, and track their performance over time. This feature is an instrumental tool for any law enforcement officer centered on constant improvement and effectiveness in their duty.

Law Enforcement Officer Account Settings

The Account Settings section offers law enforcement officers the ability to manage their profile and customize their application experience. It maintains a keen balance of personal details, app preferences, and provisions relating to data visibility and control.

Profile Management: Officers can view and edit their profile, including their names, profile picture, email address, and contact number. They can also choose to change their password from this section, promoting accountsecurity. Furthermore, officers can view their assigned police station and jurisdiction but are unable to change these details themselves for data accuracy.

App Preferences: Account settings also allow officers to manage app-specific settings, such as notification preferences. Officers can toggle notifications on or off based on their convenience, ensuring they don't miss any important updates about reported crimes in their jurisdiction.

Data Visibility and Control: Like regular users, law enforcement officers also have the option to delete their account from the Account Settings. Triggering this action again prompts a confirmation to prevent accidental triggers. Post-confirmation, the app permanently erases the officer's data, adhering strictly to the guidelines of data privacy.

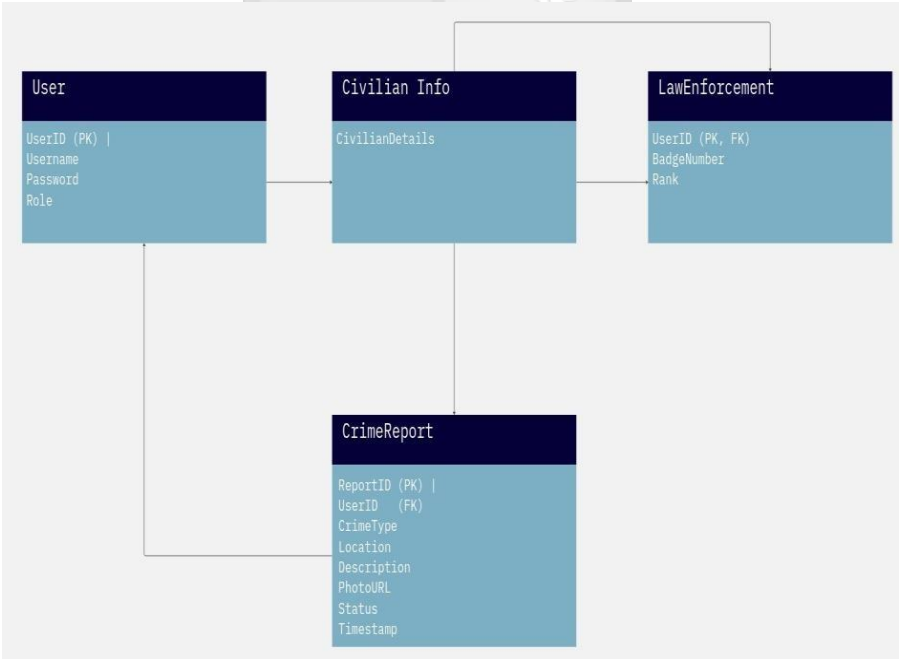
5.2.6 Database Design

Database Schema and Data Flow

Starting with a user logging into their account, any crime reported by the user creates a new entry or row in the 'Crime Reports' Table, capturing the crime details, user's location, and setting the status as 'Reported.' These reports become available to law enforcement officers in their dashboard. If an officer accepts the responsibility of addressing the crime, the database updates the corresponding entry in the 'Crime Reports' Table by marking the status 'In Progress' and adding the officer's ID. As the officer provides updates or changes the status, the database records these modifications, and the 'Notifications' Table triggers an update to the user.



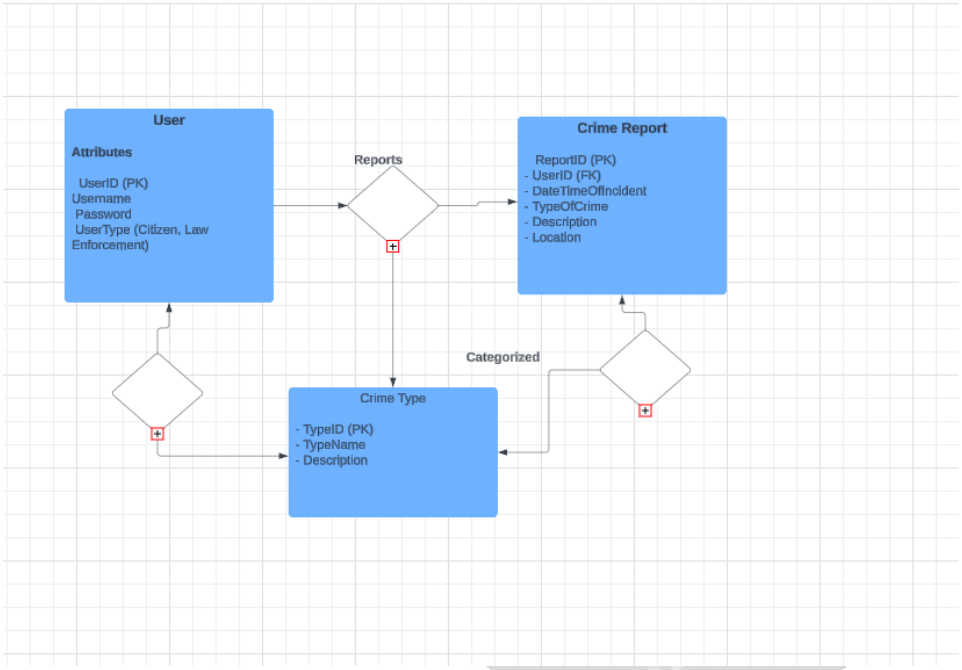
Database Data Flow



Database Schema

This meticulously designed database schema and logical data flow help ensure a seamless, responsive application experience, enabling rapid reporting and resolution of criminal activities. The efficient traffic of data contributes to the application's primary objective of assisting users and aiding law enforcement agencies effectively.

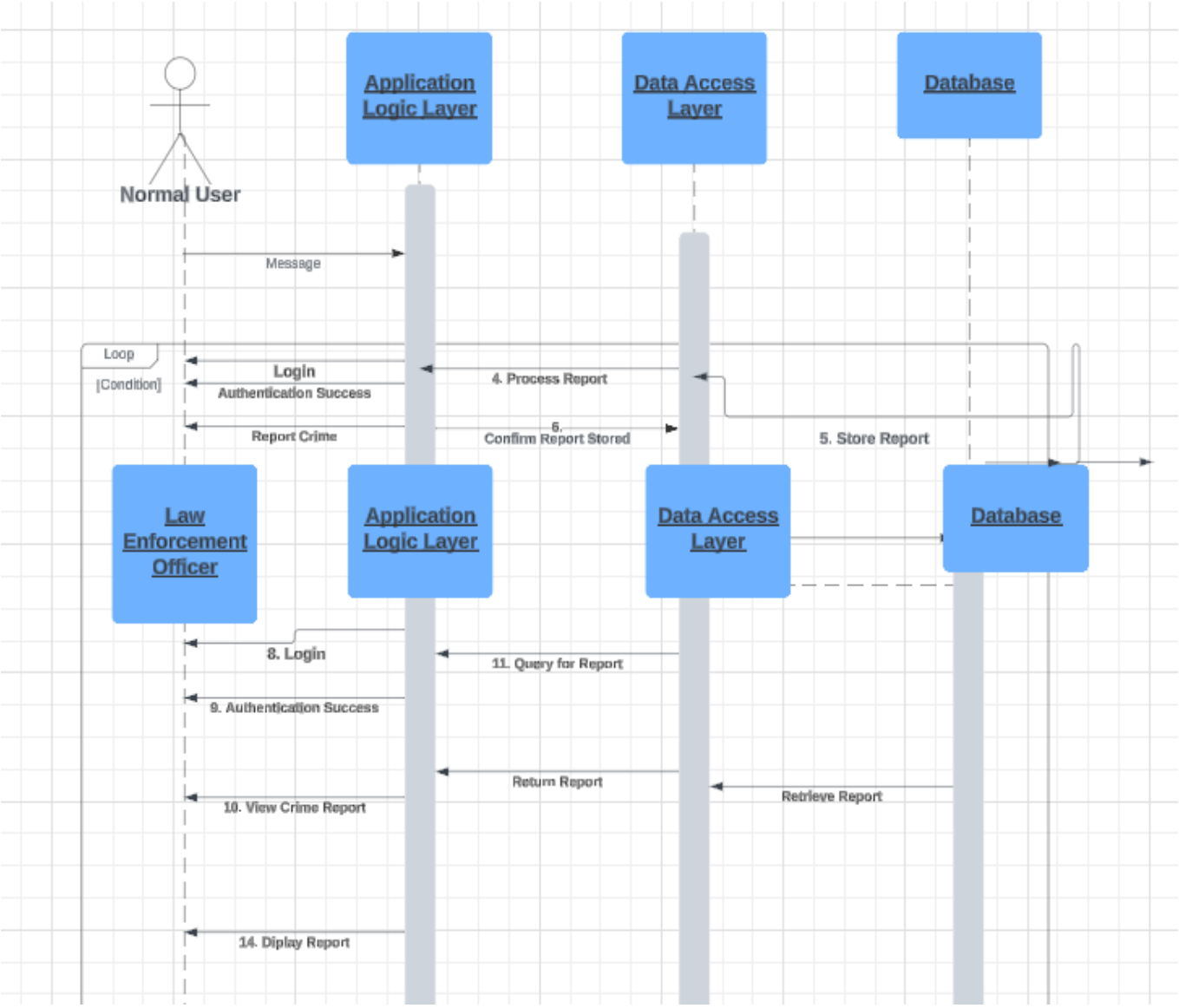
5.2.7 Entity Relationship Diagram



In the conceptual blueprint of our crime reporting application, we delineate a trio of pivotal entities—"User", "CrimeReport", and "CrimeType"—each interwoven through a network of relationships that mirror the real-world dynamics of crime reporting. The User entity embodies both the citizens, who report crimes, and law enforcement officers, who review these reports. Each user is uniquely identified and distinguished by their role within the system, facilitating a tailored interaction experience.

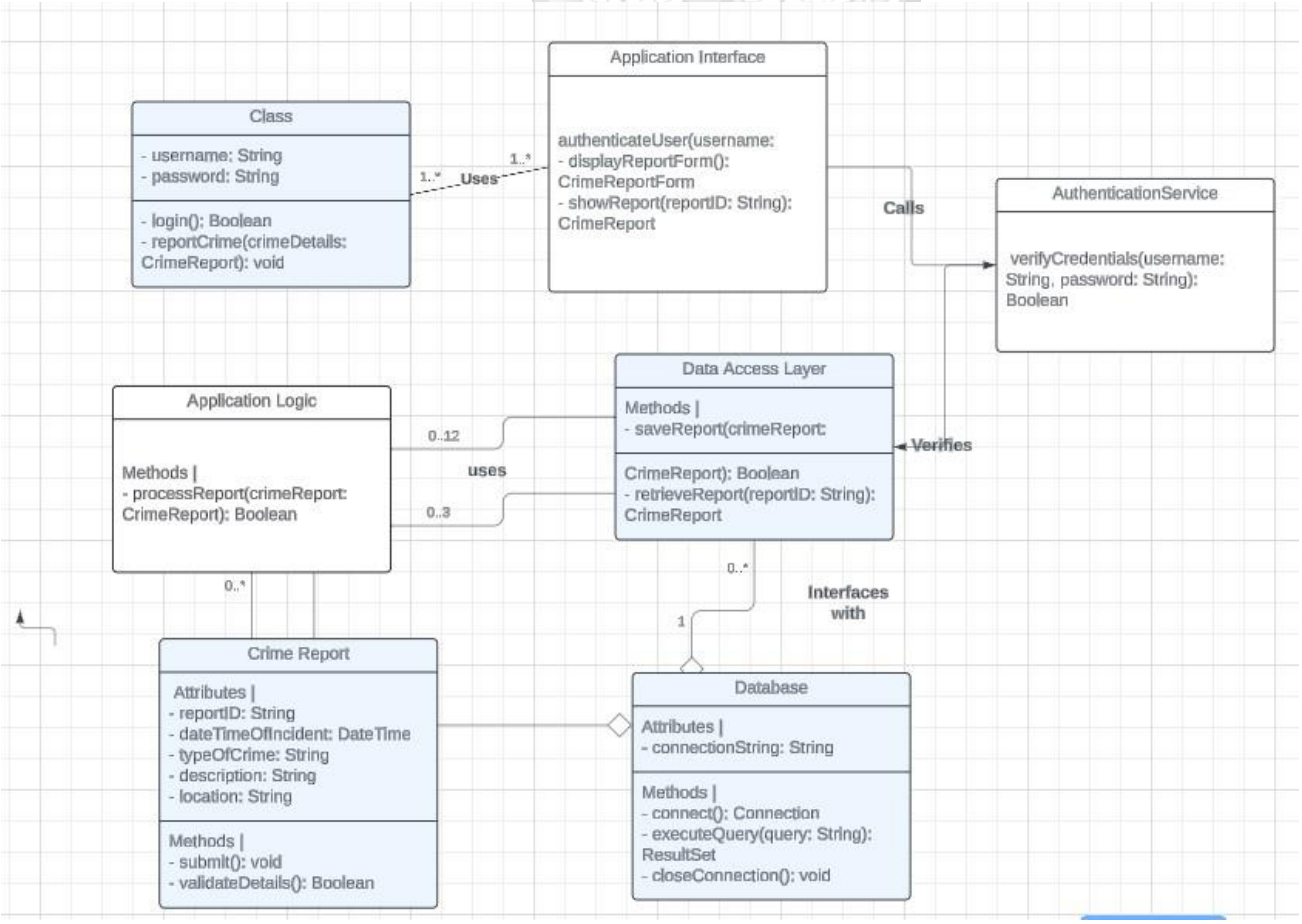
Branching from the User entity is the "CrimeReport" entity, a repository of detailed crime narratives. This entity captures not only the temporal and spatial specifics of the crime but also categorizes the nature of the offense, linking back to the reporting user. This connection manifests a one-to-many relationship, underscoring the reality that a single user may report multiple incidents over time, yet each report is singular in its origin.

5.2.8 Sequence Diagram



The sequence diagram for the crime reporting application delineates the interaction between users, the application logic layer, the data access layer, and the database. Initially, a normal user logs into the application, upon which the application authenticates the user and grants access (steps 1 and 2). Once authenticated, the user can report a crime through the application, triggering the application logic layer to process the report (steps 3 and 4). The processed data is then sent to the data access layer, which is responsible for storing the crime report in the database (steps 5 and 6). The data access layer confirms the storage back to the application logic layer, which in turn confirms the report submission to the normal user (steps 7). Separately, a law enforcement officer logs into the application through a different access point, receives authentication, and requests to view a specific crime report (steps 8 to 10). The application logic layer queries the data access layer, which retrieves the requested report from the database (steps 11 and 12). The retrieved report is sent back through the layers to the application logic layer, and ultimately, the report is displayed to the law enforcement officer (steps 13 and 14). This sequence of actions ensures that crime reports are securely submitted and accessed by authorized parties in a systematic andtraceable manner.

5.2.9 UML Diagram



In the heart of the crime reporting system's design lies a meticulously crafted structure that orchestrates the flow of data and control across several key components. At the forefront, the Citizen class encapsulates the essential attributes and behaviors of users, empowering them with functionalities to log in and report crimes through intuitive methods. This class interacts directly with the Application Interface, serving as the primary conduit through which users engage with the system. The Application Interface, in turn, liaises with the Authentication Service to authenticate user credentials, ensuring a secure entry point into the system.

Once authenticated, the journey of a crime report begins with the Citizen class leveraging the Application Interface to initiate a new report, encapsulated within the Crime Report class. This class holds detailed information about the crime, such as its type, description, and location, and provides methods for submission and validation, ensuring the integrity and completeness of the report.

The Application Interface further acts as a bridge to the Application Logic, where the essence of the report undergoes processing. This vital component of the system leverages the Data Access Layer (DAL) to persist the report into the Database, marking a seamless transition from the user's input to secure storage. The DAL, with its robust methods for saving and retrieving reports, interacts closely with the Database, a foundational layer that manages the connection and execution of queries, embodying the system's capacity for data persistence and retrieval.

Furthermore, law enforcement officers, through a similar interaction pattern with the Application Interface, can query and view reports, completing the cycle of crime reporting and investigation support. This structured approach not only delineates clear roles and responsibilities across the system's architecture but also ensures a cohesive and secure workflow from the moment a report is initiated to the point it becomes accessible for law enforcement review.

5.2.10 Use Case Diagram

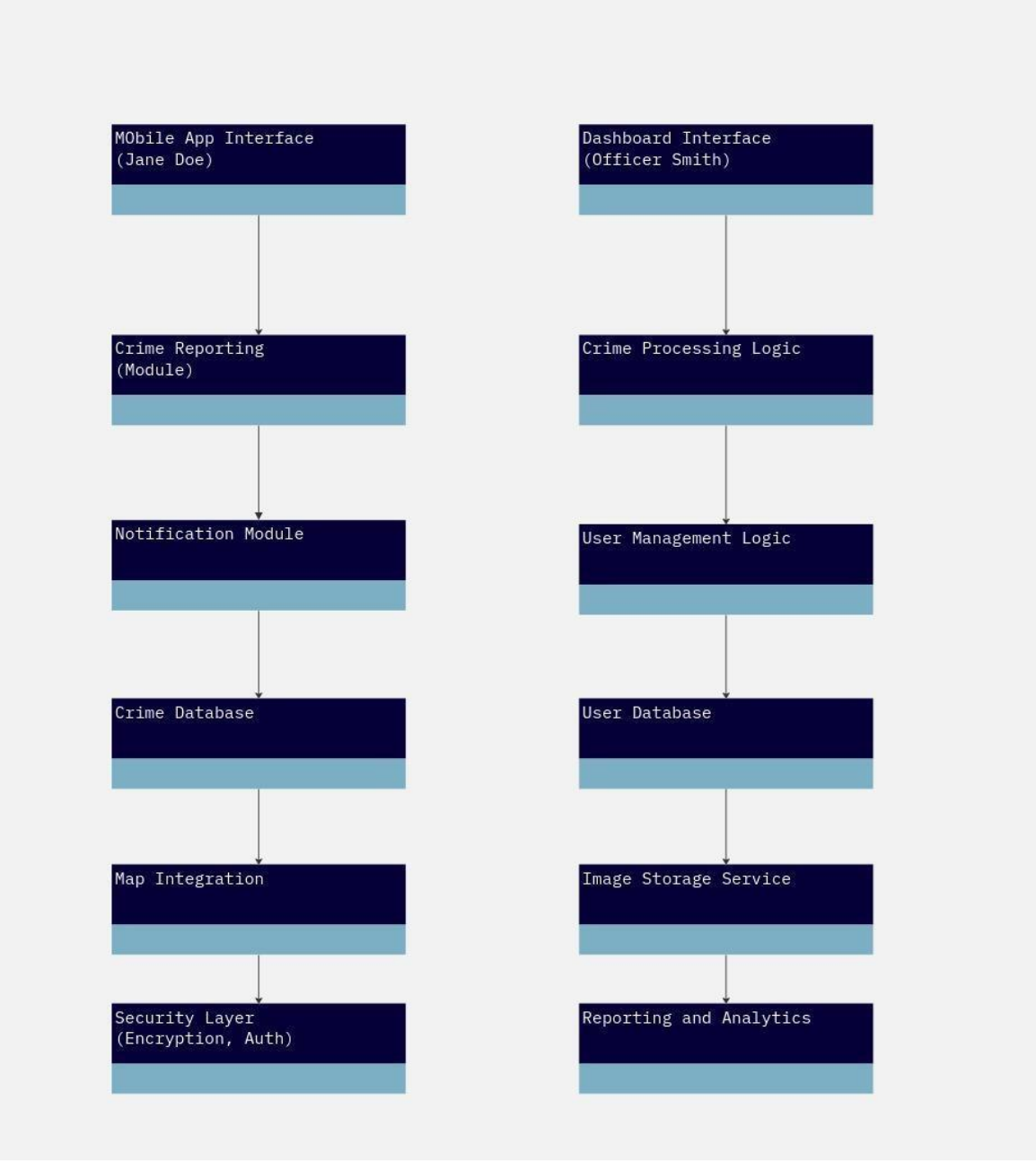
Below are some examples of hypothetical scenario that simulates a real-life use case of the application: Suppose a user, Jane Doe, witnesses a robbery at a cafe near her place. She logs into the Crime Reporting App, goes to 'Report Crime,' fills in the required fields with the crime type, location, description, and uploads a relevant photo. Upon clicking 'Submit,' the data gets pushed to the database, and a new crime report is successfully created.

On the other end, a law enforcement officer, Officer Smith, sees this new report in his dashboard. He reviews the information, accepts the report, and chooses to navigate to the crime scene. Acceptance of this crime report changes its status to 'In Progress' on Jane Doe's view as well as in the database.

After reaching the location and ensuring it's secure, Officer Smith updates the report status to 'Resolved' from his dashboard. This status update is reflected in Jane Doe's account, and she is notified about the update. Both Jane Doe and Officer Smith can review this crime report's details and status changes in their 'History Tab' at any time.

The sequence of successful interactions in this scenario suggests a positive result from our system testing. The

application's components worked seamlessly together, the database consistently recorded changes, and both user categories interacted successfully with the application. Conclusively, this affirms that the Location Based Security Response Mobile Application meets the set requirements and ready for real-time application. The figure below shows a system architecture summarizing this process.



5.3. System Testing

System testing forms a critical inspection point in the lifecycle of app development, facilitating the assessment of integrated components, overall functionality, and performance.

5.3.1 Approach to Testing

The testing protocol embraced a multi-tier approach, undertaking several layers of tests to ensure comprehensivecoverage:

Unit Testing: Began with unit testing, where individual components (like login, report generation, status update,etc.) were tested in isolation to ensure they function correctly.

Integration Testing: Subsequently, we tested how efficiently these units interact with each other and the backendservers, confirming seamless data flow and logic execution.

Functional Testing: This tested the functionality of the app as a whole and its alignment with the specified requirements. It ensured that the app functions as expected and provided the defined features and services to the users.

Load Testing: We also simulated various use case scenarios, ranging from minimal use to intense simultaneous usage to test the app's endurance and responsiveness under different loads.

5.4. Security Measures

Data security and privacy are paramount in any mobile application handling user data, and the Location Based Security Response Mobile Application strictly adheres to this principle, implementing stringent security mechanisms to protect user information and communication.

User Information Protection: All user data - be it personal information of the regular user or law enforcement officer - stored in the application's database, is encrypted. Encryption ensures that the data, if intercepted during a potential breach, remains unintelligible and useless to the interceptor. Moreover, certain sensitive information, such as passwords, is hashed. A hashed password means even the application system cannot retrieve the actual password, ensuring an additional layer of security.

Privacy Feature: User privacy has been a central focus in the design of the application. Users have full control over what personal information they want to share. They can also control notification and location settings per their comfort and needs.

Location Service Control: Location information is especially sensitive as it can reveal real-time movement of theusers. The user has the option to control if and when they want to share their location. The real-time location is only accessed when a crime is being reported, ensuring that the user's location stays confidential during other times. The application asks for necessary permissions from the user before accessing location

services.

Data Sharing Policy: Strict guidelines are in place regarding sharing of data. Without the explicit consent of users, their personal information is not shared with other parties. This applies to both normal users and law enforcement officers.

These high-grade security and privacy measures ensure users can use the application without concerns about privacy or data breach. The protection of user data and maintaining the confidentiality of crime reports is fundamental to the overall credibility and effectiveness of the application. These measures also comply with data protection regulations and underline our commitment to user privacy and safety.

Chapter 6: Conclusion and Recommendation

The high prevalence of mobile phone ownership among law enforcement officers and residents underscores the potential of mobile technology in enhancing crime reporting and law enforcement efforts. Android devices emerged as the dominant operating system among respondents, followed by iOS, highlighting the need to develop mobile applications that cater to these preferences. Traditional modes of crime reporting, such as physical visits to police stations and telephone calls, remain widely known and used among law enforcement officers and residents. However, there is a need to increase awareness and training regarding the benefits of modern reporting methods, such as mobile application reporting and social media, which showed lower levels of awareness but offered convenience and accessibility. While a significant percentage of crime victims reported crimes, the response rate of law enforcement officers within a reasonable timeframe was relatively low. This emphasizes the importance of improving response times and considering factors such as the reported crime's type, severity, and location when allocating resources for rapid response.

Mobile applications can be vital in streamlining the reporting and response process, enhancing communication between law enforcement and residents. The willingness of crime victims to report crimes via mobile applications signifies the potential of technology to enhance crime reporting practices. This positive response highlights the convenience, accessibility,and benefits mobile applications offer for reporting crimes promptly and accurately. Leveraging this willingnesscan lead to more efficient and effective crime reporting practices. Factors prioritized by both law enforcement officers and residents when reporting crimes through technology include availability, efficiency, ease of use, confidentiality, and cybersecurity. Addressing these factors is crucial in developing user-friendly interfaces, efficient communication channels, and robust cybersecurity measures in mobile crime reporting applications.

This positive response suggests that both residents and law enforcement officers recognize the convenience, accessibility, and potential benefits offered by mobile applications for reporting crimes. The widespread use of mobile phones in Nairobi, coupled with the high ownership rate among both groups, indicates a readiness to embrace technology to enhance crime reporting practices.

Recommendations and Future Work

The research underscores the imperative need for enhanced awareness and training among law enforcement officers and Nairobi residents regarding mobile application reporting. The gap in awareness necessitates the organization of awareness campaigns, workshops, seminars, and online tutorials to educate both law enforcement personnel and the public on the benefits, functionalities, and confidentiality aspects of the crime reporting application. Elevating awareness is anticipated to drive greater adoption and utilization of this technological solution.

Second, in parallel, the significance of developing user-friendly interfaces emerges as a priority, especially since ease of use is a critical factor for Nairobi residents in reporting crimes via a mobile application. The design of intuitive and easy-to-navigate interfaces should be paramount in the application development process. Through conducting usability testing and gathering feedback from potential users, developers can pinpoint areas needing enhancement, thereby ensuring the application aligns with user needs and expectations.

Third, another critical area highlighted by the research is the need to bolster cybersecurity measures. Given the lessened priority that law enforcement officers and Nairobi residents place on cybersecurity, it is vital to instill trust and confidence in the application's use. Implementing stringent cybersecurity protocols, such as encryption, secure data storage, and multi-factor authentication, is essential in protecting sensitive information and preserving the privacy of crime reports.

Fourth, the study reveals a pressing need to optimize response times by law enforcement to reported crimes. With a mere 4% of responses occurring within 15-30 minutes, there is a clear imperative to enhance the efficiency of crime response. Improvements can be achieved through better coordination, strategic resource allocation based on the severity and scope of reported crimes and leveraging real-time data to prioritize emergencies effectively.

Lastly, promoting community engagement stands out as a critical factor for the successful implementation of a mobile crime reporting application. Law enforcement agencies should seek collaboration with local community leaders, NGOs, and other stakeholders to foster the application's use and cultivate a trustful relationship between residents and law enforcement. By engaging with the community and addressing their concerns, a more robust partnership in ensuring public safety can be established, thereby enhancing the overall impact of the crime reporting application on community well-being.

Reference

Adams, D. A. (1992). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS quarterly*, *13*(3), 319-339.

Akers, R. L., & Jennings, W. G. (2015). Social learning theory. The handbook of criminological theory,230-240.

APIMA, M. B. (2019). THE ROLE OF MOBILE PHONE TRACKING TECHNOLOGY IN CRIME PREVENTION IN KENYA: A CASE OF NAIROBI CITY COUNTY.

AlertMedia. (2022, March 18). Active shooter planning: Prevention, preparedness, and response. https://www.alertmedia.com/blog/active-shooter-response-plan/

Atta-Asamoah, A. (2015). East Africa Report. *The Nature and Drivers of Insecurity in Kenya*.

Babu, S. S., Yellepeddi Vijayalakshmi, D. A., & Kala, J. R. (2016) Improving Performance and Privacyin Location Based Service Applications Using GPS.

Balapour, A., Nikkhah, H. R., & Sabherwal, R. (2020). Mobile application security: Role of perceivedprivacy as the predictor of security perceptions. *International Journal of Information Management*, *52*, 102063.

Bhardwaj P. (2019). Types of sampling in research J Pract Cardiovasc Sci 2019; 5:157-63).

Bjame Stroustrup, (1991), What is "Object Oriented Programming" *Object Oriented Programming*, *6*.

Bett, A., & Muturi, W. (2016). Does Economic Crimes Affect Kenya's Economic Growth? A Cointegration Approach. International Journal of Economics, Commerce and Management, 4(11), 231-250.

PARK, S. H., PARK, K. R., & KIM, J. Y. (2018). MODELING OF CRIME PREVENTION SYSTEM USING LOCATION-BASED SERVICE. Journal of Theoretical and Applied Information Technology, 96(22).

Huang, H., Gartner, G., Krisp, J. M., Raubal, M., & Van de Weghe, N. (2018). Location based services: ongoing evolution and research agenda. Journal of Location Based Services, 12(2), 63-93.

Citizen Digital (2022, November 14). Police shake-up in Nairobi as gov't moves to contain crime in the capital. Citizen Digital. https://www.citizen.digital/news/police-shake-up-in-nairobi-as-govt-moves-to-contain-crime-in-the-capital-n309405

Communications Authority of Kenya. (2021, December 16). 59 million mobile phone devices connected to mobile networks as at September 2021, CA report shows. https://www.ca.go.ke/59-million- mobile-phone-devices-connected-to-mobile-networks-as-at-september-2021-ca-report-shows/

de Guzman, J. B., & Ado, R. G. (2014). Mobile emergency response application using geolocation for command centers. International Journal of Computer and Communication Engineering, 3(4),235.

Eneh, J. N., & Orah, H. O. (2012). Location Based Service (LBS) Provision: Concept, Benefits and Future Challenges.

Filippi, F. D., Cocina, G. G., & Martinuzzi, C. (2020). Integrating Different Data Sources to Address Urban Security in Informal Areas. The Case Study of Kibera, Nairobi. Sustainability, 12, 2437.

Green, S. L. (2002, May). Rational choice theory: An overview. In Baylor University Faculty development seminar on rational choice theory (pp. 1-72).

Guha, S., Jain, M., & Padmanabhan, V. N. (2012, April). Koi: A Location-Privacy Platform for Smartphone Apps. In NSDI (Vol. 12, p. 14).

Hasan, B., Rajski, E., Gómez, J. M., & Kurzhöfer, J. (2016). A proposed model for user acceptance of mobile security measures–business context. In *Mobile and Wireless Technologies 2016* (pp. 97-108). Springer Singapore.

Hassan, F., & Alashqar, M. (2020). The Impact of Location-Based Mobile Security Response Applications on Security Management. International Journal of Mobile Communications, 18(5),558-573.

Information & Security, *28*(1), 146.

International Crisis Group. (2018). Kenya: Nairobi's Insecurity Crisis. Retrieved from https://www.crisisgroup.org/africa/horn-africa/kenya/kenya-nairobis-insecurity-crisis

Inukollu, V. N., Keshamoni, D. D., Kang, T., & Inukollu, M. (2014). Factors influencing quality ofmobile apps: Role of mobile app development life cycle. *arXiv preprint arXiv:1410.4537*.

Jang, S. H., & Lee, C. W. (2018). The impact of location-based service factors on usage intentions for technology acceptance: The moderating effect of innovativeness. Sustainability, 10(6), 1876.

Jenga, K., Catal, C., & Kar, G. (2023). Machine learning in crime prediction. Journal of Ambient Intelligence and Humanized Computing, 14(3), 2887-2913.

Kaplan, A. M., & Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of social media. Business horizons, 53(1), 59-68.

Kaur, S., & Sharma, S. (2020). Role of Location-Based Security Response Mobile Applications in Enhancing Security in the Society. International Journal of Computer Applications, 177(9), 1-5.

Kaushal, A., & Anand, P. (2020). Location Based Security Response Mobile Application: Need of the Hour. International Journal of Computer Science and Mobile Computing, 9(9), 52-59.

Kenya National Bureau of Statistics. (2021). Economic Survey 2021. Kenya National Bureau of Statistics, Nairobi, Kenya. https://www.knbs.or.ke/wp-content/uploads/2021/09/Economic-Survey-2021.pdf

Kenya Travel Advisory. (2022). Kenya travels advisory - Exercise Increased Caution. Travel. https://travel.state.gov/content/travel/en/traveladvisories/traveladvisories/kenya-travel-advisory.html

Kenya, L. O. (2013). The constitution of Kenya: 2010. Chief Registrar of the Judiciary.

Kimenju, G. N. (2015). Determinants of Reporting or Failing to Report A Crime to Police: A Case Study of Githurai 45, Nairobi (Doctoral dissertation, University of Nairobi).

Koper, C. S., Lum, C., & Willis, J. J. (2014). Optimizing the use of technology in policing: Results and implications from a multi-site study of the social, organizational, and behavioral aspects of implementing police

technologies. *Policing: A Journal of Policy and Practice*, *8*(2), 212-221.

Koper, C. S., Lum, C., Willis, J. J., Woods, D. J., & Hibdon, J. (2015, December). Realizing the potential of technology in policing: A multi-site study of the social, organizational, and behavioral aspects of implementing policing technologies. In Report to the National Institute of Justice, US Department of Justice. Fairfax, VA: Center for Evidence-Based Crime Policy, George Mason University and Police Executive Research Forum.

Kumbuti, H. M. (2013). Use of technology as a strategy by Kenya police in detection of crimes in Nairobi City (Doctoral dissertation, University Of Nairobi). Literature. International Journal of Mobile Communications, 18(5), 543-557.

Lyon, D., Marmura, S., & Peroff, P. (2005). *Location Technologies: Mobility, surveillance and privacy*.

Minnery, J. R., & Lim, B. (2005). Measuring crime prevention through environmental design. Journal of architectural and planning research, 330-341.

Miró, F. (2014). Routine activity theory. The encyclopedia of theoretical criminology, 1-7.

Mugenda, A. G., & Mugenda, A. G. (2012). Research methods dictionary. Nairobi, Kenya: Applied Research & Training Services.

Mugi, J. W. (2016). A framework for Enhancing Privacy in Location Based Services (Doctoral dissertation, Jomo Kenyatta University of Agriculture and Technology).

Natasha, E. I. (2007). Insecurity situation in Kenya: its economic and social costs on businesses within Nairobi's central business district (Doctoral dissertation, University of Nairobi).

O'Hara, K. (2018). Use of Mobile Applications in Law Enforcement: An Analysis. Journal of Criminal Law and Criminology, 108(3), 563-591.

Oketch, P. O. (2018). Crime Reporting and Intelligence System: Case of BuruBuru Police Division-Nairobi (Doctoral dissertation, United States International University-Africa).

Omolo, F. O. (2016). A Mobile and web-based application for security intelligence gathering-a case study of Nairobi County (Doctoral dissertation, Strathmore University).

Orodho, J. A. (2009). Elements of education and social science research. Maseno: Kanezja Publishers.

Pitt, R., & Wager, E. A. (2020). Location-based Mobile Security Response Applications: A Review of the

Porras, J., Hiirsalmi, P., & Valtaoja, A. (2004, January). Peer-to-peer communication approach for a mobile environment. In 37th Annual Hawaii International Conference on System Sciences, 2004.

Proceedings of the (pp. 7-pp). IEEE.

Quarshie, H. O. (2014). Using ICT to Fight Crime-A Case of Africa. *Journal of Emerging Trends in Computing and Information Sciences*, *5*(1), 21-24.

Rodgers, S. (1998). The dissemination and utilization of research. Research and development in clinical nursing practice, 247-270.

Srivastava, L. (2005). Mobile phones and the evolution of social behavior. *Behavior & Information Technology*, *24*(2), 111-129.

ToppiReddy, H. K. R., Saini, B., & Mahajan, G. (2018). Crime prediction & monitoring framework based on spatial analysis. Procedia computer science, 132, 696-705.

Tsai, J. Y., Kelley, P. G., Cranor, L. F., & Sadeh, N. (2010). Location-sharing technologies: Privacy risks and controls. *Isjlp*, *6*, 119.

Uğur, N. G., & Turan, A. H. (2019). Mobile applications acceptance: a theoretical model proposal and empirical test. *International Journal of E-Adoption (IJEA)*, *11*(2), 13-30.

Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management science*, *46*(2), 186-204.

Vinarao, E. D. G., De Guzman, M. N. B., Fernandez, E. A., Quije, D. J. V., Gorres, R. C., Francisco, E. D., ... & Cruz, E. N. (2019, October). Athena: A Mobile Based Application for Women's Safety with GPS Tracking and Police Notification for Rizal Province. In 2019 IEEE Student Conference on Research and Development (SCOReD) (pp. 117-122). IEEE.

Volkoviy, A., & Tikhov, I. (2012). Use of Mobile Applications in Security and Safety Systems. Wambugu, P. M. (2017). Application of Mobile Phone in Crime Prevention within Central Division, Nairobi City County

Wasomi, C. B., Waithaka, E. H., Gachari, M. K., & Kuria, D. N. (2021). Location Based Services Acceptance and Adoption Evaluation: An Investigation on Mobile Phone Users' Response to Awareness and Utilization of LBS Technology in Nairobi Kenya.

Weichbroth, P. (2020). Usability of mobile applications: a systematic literature study. *Ieee Access*, *8*, 55563-55577.

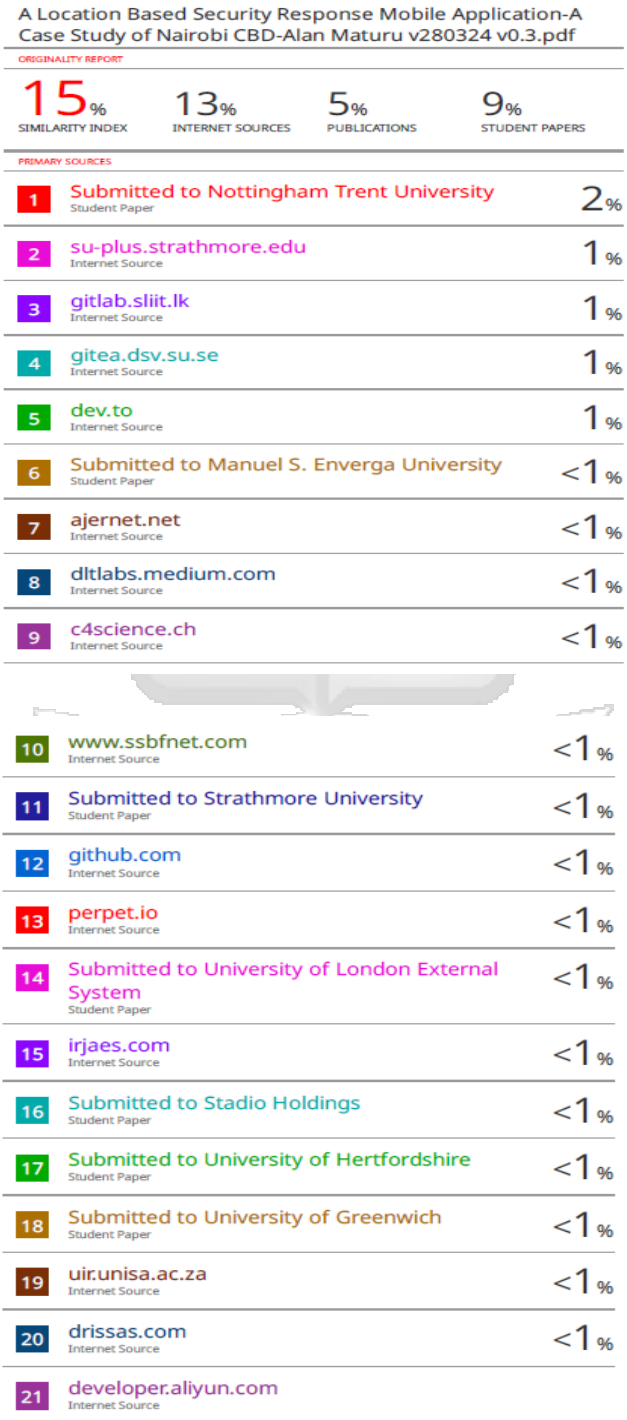
White, R. (1996). Situating crime prevention: Models, methods and political perspectives. Crime Prevention Studies, 5, 97-113.

Yuki, J. Q., Sakib, M. M. Q., Zamal, Z., Habibullah, K. M., & Das, A. K. (2019, July). Predicting crime using time and location data. In Proceedings of the 7th International Conference on Computer and Communications Management (pp. 124-128).

Curtis, D. D., & Keeves, J. P. (2000). The Course Experience Questionnaire as an institutionalperformance indicator.

Appendix A: Plagiarism Checker

As highlighted in Turnitin Similarity Checker, the plagiarism check returned a similarity report of 15% which is within the allowable threshold allowed by the Strathmore University School of Graduate School.



A Location Based Security Response Mobile Application-A Case Study of Nairobi CBD-Alan Maturu v280324 v0.3.pdf

ORIGINALITY REPORT

15%	13%	5%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Nottingham Trent University Student Paper	2%
2	su-plus.strathmore.edu Internet Source	1%
3	gitlab.sliit.lk Internet Source	1%
4	gitea.dsv.su.se Internet Source	1%
5	dev.to Internet Source	1%
6	Submitted to Manuel S. Enverga University Student Paper	<1%
7	ajernet.net Internet Source	<1%
8	dltlabs.medium.com Internet Source	<1%
9	c4science.ch Internet Source	<1%
10	www.ssbfnet.com Internet Source	<1%
11	Submitted to Strathmore University Student Paper	<1%
12	github.com Internet Source	<1%
13	perpet.io Internet Source	<1%
14	Submitted to University of London External System Student Paper	<1%
15	irjaes.com Internet Source	<1%
16	Submitted to Stadio Holdings Student Paper	<1%
17	Submitted to University of Hertfordshire Student Paper	<1%
18	Submitted to University of Greenwich Student Paper	<1%
19	uir.unisa.ac.za Internet Source	<1%
20	drissas.com Internet Source	<1%
21	developer.aliyun.com Internet Source	

Appendix B: Ethics Review Board Approval

The following highlights the Ethics Review Board Approval for this research paper:

Strathmore UNIVERSITY

5th May 2023

Mr Maturu Alan,
alan.maturu@gmail.com

Dear Mr Maturu,

RE: A Location-Based Security Response Mobile Application-A Case Study of Nairobi CBD

This is to inform you that SU-ISERC has reviewed and **approved** your above SU-masters research proposal. Your application reference number is **SU-ISERC1709/23**. The approval period is from **5th May 2023 to 4th May 2024.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used.
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-ISERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-ISERC within 72 hours of notification.
iv. Any changes anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-ISERC within 72 hours.
v. Clearance for the export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to the expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days of completion of the study to SU-ISERC.

Before commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and obtain other clearances needed.

Yours sincerely,

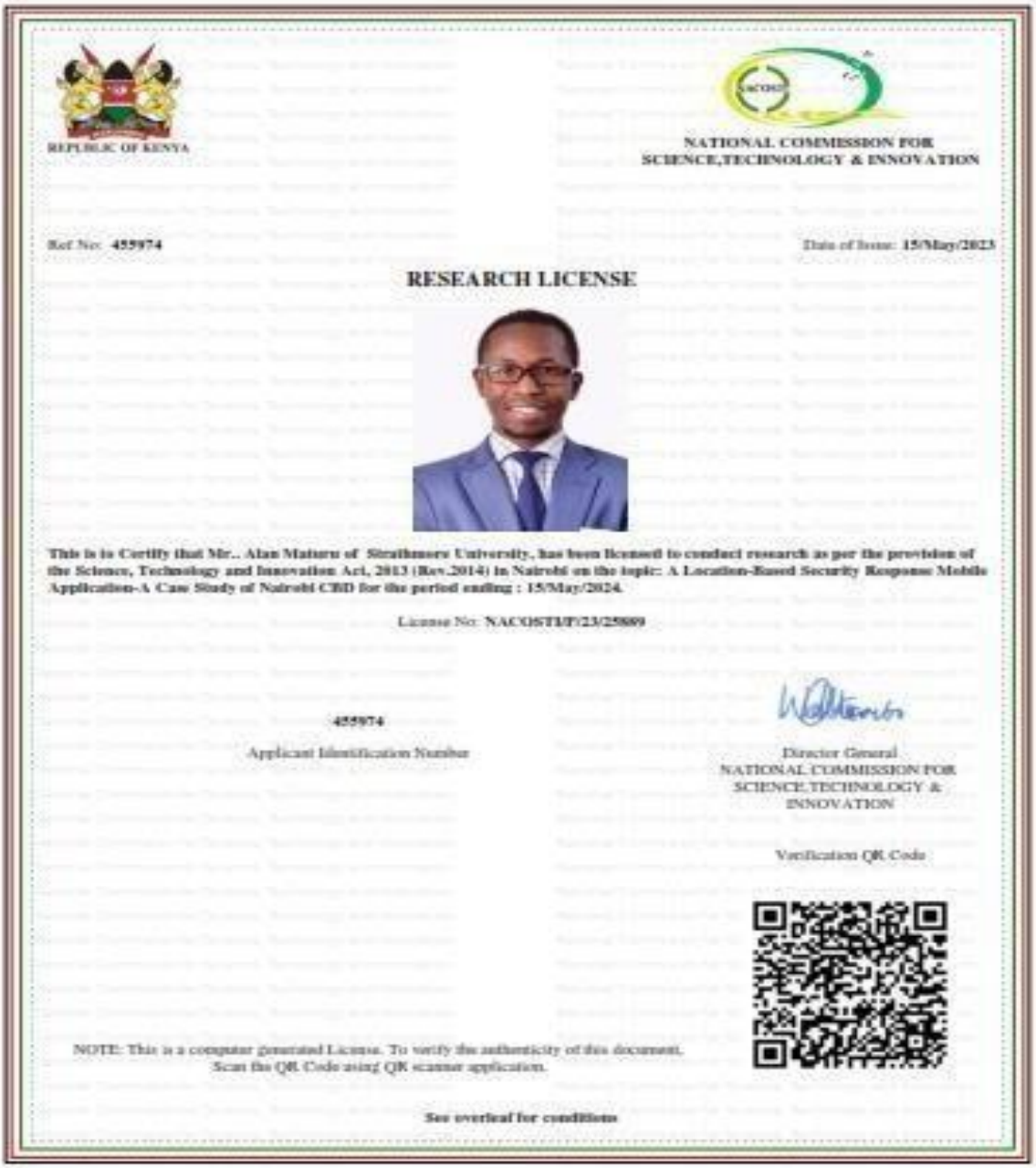
for: **Mr Ambrose Rachier,**
Chairperson; SU-ISERC

STRATHMORE UNIVERSITY INSTITUTIONAL SCIENTIFIC AND ETHICAL REVIEW COMMITTEE
(SU-ISERC)
05-May-2023
Email:ethicsreview@strathmore.edu
P.O BOX 59857-00200
NAIROBI-KENYA

Ole Sangale Rd, Madaraka Estate. PO Box 59857-00200, Nairobi, Kenya. Tel +254 (0)703 034000
Email admissions@strathmore.edu www.strathmore.edu

Appendix C: NACOSTI Permit

The following highlights the NACOSTI permit for this research paper:



REPUBLIC OF KENYA

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Ref No: **455974**

Date of Issue: **15/May/2023**

RESEARCH LICENSE

This is to Certify that Mr.. Alan Maiuru of Strathmore University, has been licensed to conduct research as per the provision of the Science, Technology and Innovation Act, 2013 (Rev.2014) in Nairobi on the topic: A Location-Based Security Response Mobile Application-A Case Study of Nairobi CBD for the period ending : 15/May/2024.

License No: **NACOSTI/P/23/25889**

455974

Applicant Identification Number

Director General
NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Verification QR Code

NOTE: This is a computer generated License. To verify the authenticity of this document, Scan the QR Code using QR scanner application.

See overleaf for conditions

Appendix D: Research Questions

The following highlights the research questions filled in by 2 sets of respondents i.e. both the Law Enforcers and Crime Victims:

Law Enforcers

A Location Based Security Response Mobile Application-A Case Study of Nairobi CBD

Alan Maturu
Faculty of Information Technology,
Strathmore University.

Dear Respondent,

My name is Alan Maturu and I am a student at Strathmore University finalizing a Master of Science in Information Technology with a specialization in Information Management Systems. As a requirement to finalize on the degree program, I am conducting research on A Location Based Security Response Mobile Application-A Case Study of Nairobi CBD.

The information collected will go a long way in assessing the possibility of a crime victim reporting a criminal emergency through a mobile application as well as a law enforcer being notified and responding to a criminal emergency via the same mobile application.

This will form the basis for developing a mobile application that will address rapid response to criminal emergencies by law enforcers after crime victims report criminal emergencies through the software platform. The mobile application will incorporate the use of google maps to map a crime victim to the nearest law enforcer.

The information in this questionnaire will be kept confidential and names of respondents will be concealed when compiling the report.

I will be glad to offer any clarification and can be reached on alan.maturu@gmail.com. I will highly appreciateyour participation.

Sincerely,

Alan Maturu

By filling in this form, I hereby confirm that I have had this study elaborated to me and that I have understood all that pertains with it. In addition, I certify that I am voluntarily taking part in the research and agree to have my filled in questionnaire to be stored for future data analysis.

1. Which category below denotes your response to this research (tick only one option)?

Law Enforcement Officer
Crime Victim

2. Do you own a mobile device (tick only one option)?

Yes
No

3. What O/S does your mobile phone use (tick only one option)?

Android
iOS
Windows
Other

4. Do you have access to the following on your mobile phone (tick all that apply)?

Internet Access
Google Maps
A Digital Mark

5. Which are the modes of crime reporting that you are aware of within Nairobi CBD (tick all that apply)?

Physical visit t
Telephone call
SMS on a crime
Traditional Me
Social media
Mobile Applica

6. Have you ever responded to an ongoing criminal emergency (tick only one option)?

Yes
No

7. How long did it take you to respond to the crime (tick only one option)?

0-15 Minutes
15-30 Minutes
30-60 minutes
1 hour-2 hour
2 hours-4 hours
Above 4 hours
Did Not Respond
Not Applicable

8. Would you be willing to respond to an ongoing criminal emergency through a mobile alert or technological advancement as opposed to traditional means of crime notification and response (tick only one option)?

- Yes
- No

9. Please tick all the factors that you would consider when using technology (e.g. mobile application) to respond quickly to a crime using (tick all that apply)?

- Ease of use
- Availability of
- Cost of technology
- Efficiency
- Confidentiality
- Cybersecurity
- Others

10. Please tick all the applicable factors that you would consider so as to respond rapidly to a criminal emergency (tick all that apply)?

- Type of crime
- Location of crime
- Severity of the crime
- Scope of crime
- Status of the victim in society
- Others

11. Your Full Names?

12. Your Rank?

13. Your Email Address?

14. Today's Date?

Day, month, year

Kindly note that checking the box below is equivalent to a handwritten signature

Signature

VT OMNES VNVM SINT

Crime Victims

A Location Based Security Response Mobile Application-A Case Study of Nairobi CBD

Alan Maturu
Faculty of Information Technology,
Strathmore University.

Dear Respondent,

My name is Alan Maturu and I am a student at Strathmore University finalizing a Master of Science in Information Technology with a specialization in Information Management Systems. As a requirement to finalize on the degree program, I am conducting research on A Location Based Security Response Mobile Application-A Case Study of Nairobi CBD.

The information collected will go a long way in assessing the possibility of a crime victim reporting a criminal emergency through a mobile application as well as a law enforcer being notified and responding to a criminal emergency via the same mobile application.

This will form the basis for developing a mobile application that will address rapid response to criminal emergencies by law enforcers after crime victims report criminal emergencies through the software platform. The mobile application will incorporate the use of google maps to map a crime victim to the nearest law enforcer.

The information in this questionnaire will be kept confidential and names of respondents will be concealed when compiling the report.

I will be glad to offer any clarification and can be reached on alan.maturu@gmail.com. I will highly appreciate your participation.

Sincerely,

Alan Maturu

By filling in this form, I hereby confirm that I have had this study elaborated to me and that I have understood all that pertains with it. In addition, I certify that I am voluntarily taking part in the research and agree to have my filled in questionnaire to be stored for future data analysis.

1. Which category below denotes your response to this research (tick only one option)?

Crime Victim
Law Enforcement Officer

2. Do you own a mobile device (tick only one option)?

Yes
No

3. What O/S does your mobile phone use (tick only one option)?

Android
iOS
Windows
Other

4. Do you have access to the following on your mobile phone (tick all that apply)?

Internet Access
Google Maps
A Digital Mark

5. Which are the modes of crime reporting that you are aware of within Nairobi CBD (tick all that apply)?

Physical visit to the police station
Telephone call
SMS on a crime
Traditional Media: Tv or Radio
Social media
Mobile Application

6. Have you ever reported a criminal emergency (tick only one option)?

Yes
No

7. How long did it take you to get support/assistance after reporting the crime (tick only one option)?

0-15 Minutes
15-30 Minutes
30-60 minutes
1 hour-2 hour

VT OMNES VNVM SINT

2 hours-4 hours
Above 4 hours
Never Responded
Not Applicable

8. Would you be willing to report an ongoing criminal emergency through a mobile alert or technological advancement as opposed to traditional means of crime notification and response (tick only one option)?

Yes
No

9. Please tick all the factors that you would consider when reporting an ongoing criminal emergency through technology using technology (tick all that apply)?

Ease of use
Availability of
Cost of technology
Efficiency
Confidentiality
Cybersecurity
Others

10. Please tick all the applicable factors that you believe a law enforcer should consider in order for him/her to rapidly respond to a criminal emergency that a crime victim reports (tick all that apply)?

Type of crime
Location of crime
Severity of the crime
Scope of crime
Status of the victim in society

11. Your Full Names?

12. Your Email Address?

14. Today's Date?

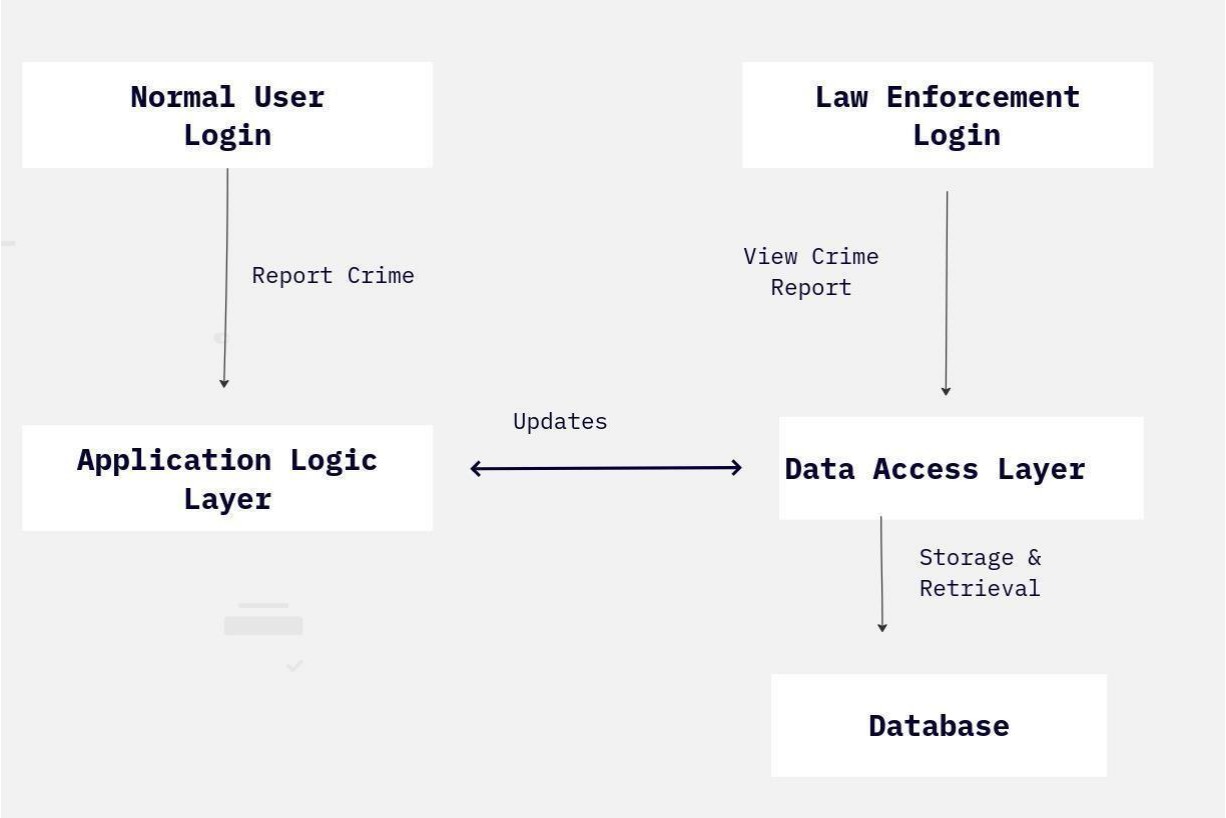
Day, month, year

15. Kindly note that checking the box below is equivalent to a handwritten signature

Signature

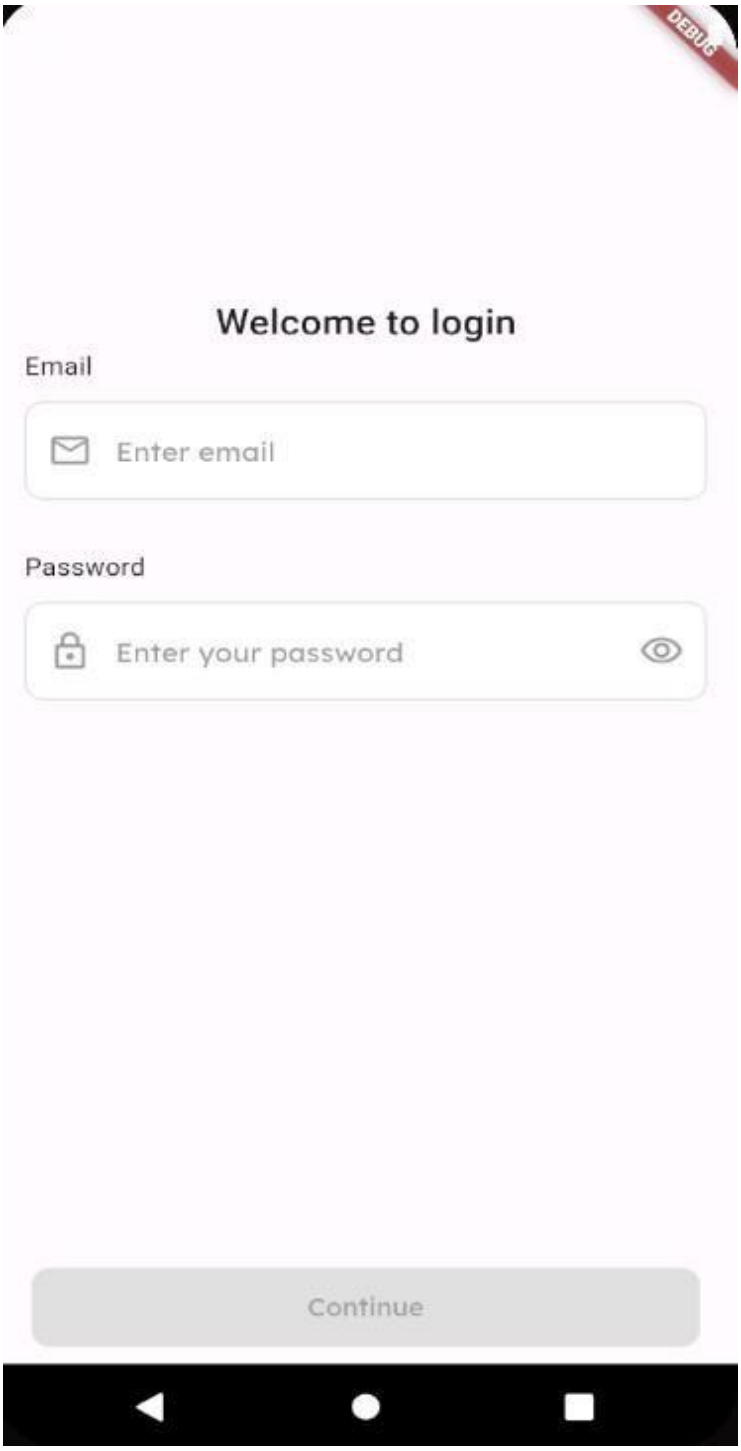
Appendix E: System Architecture

Front page

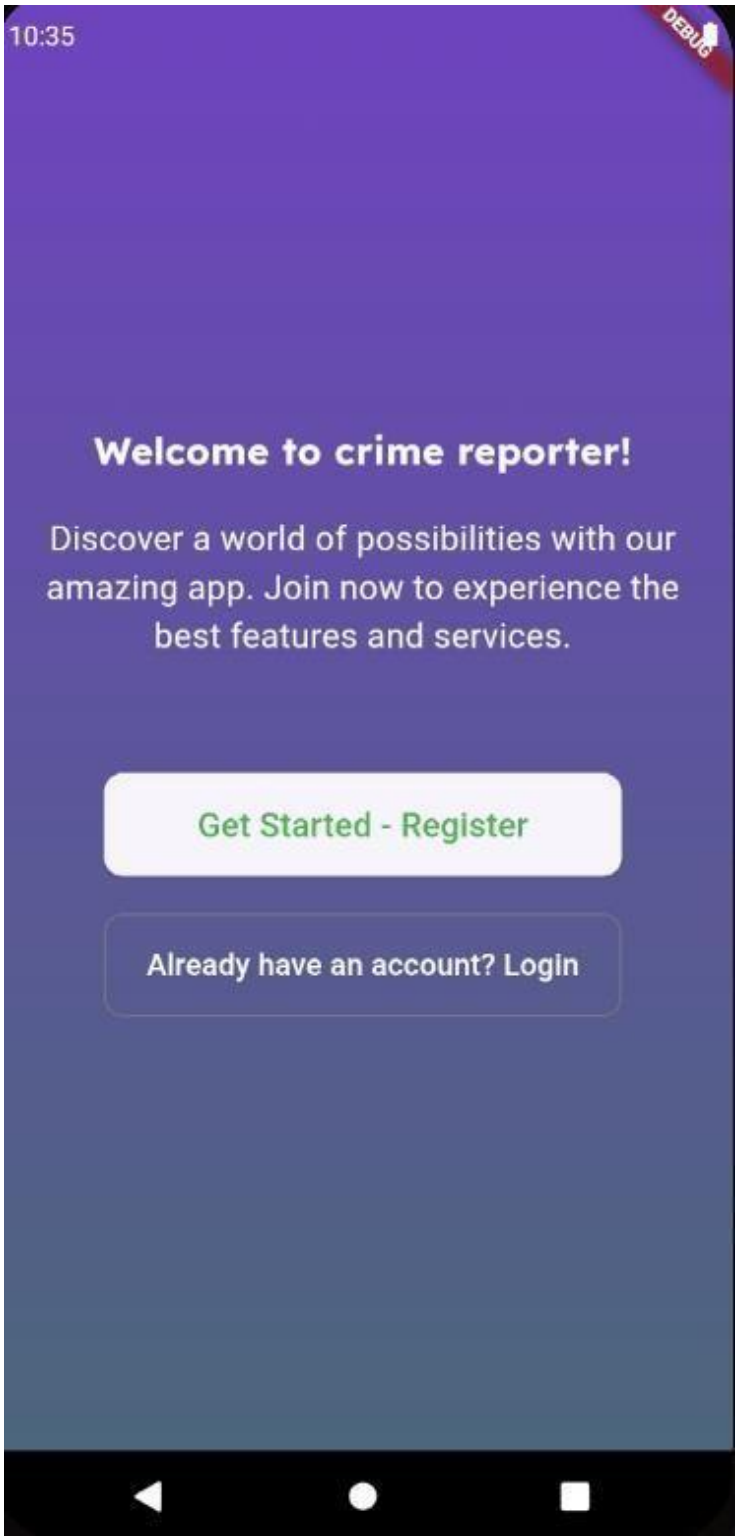


Appendix F: Application User Interface

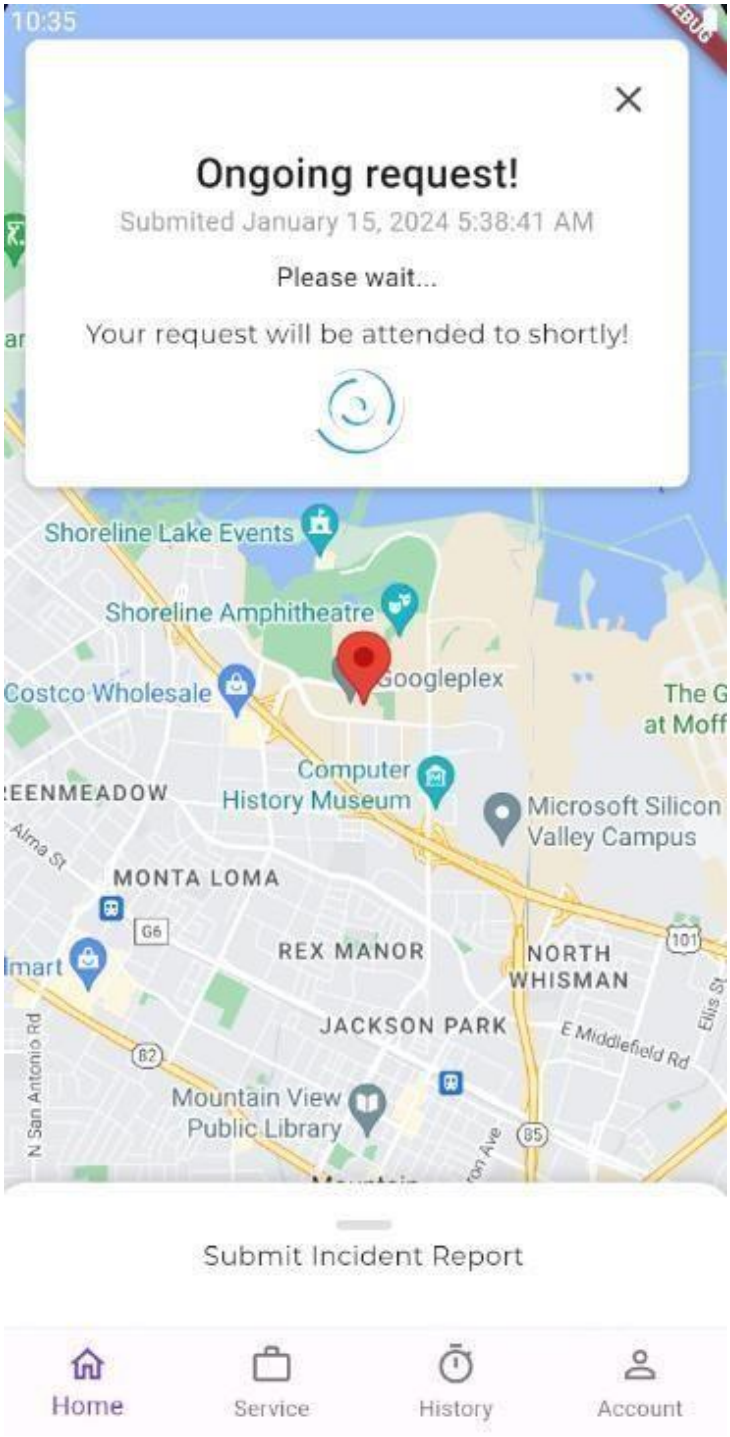
Login Page



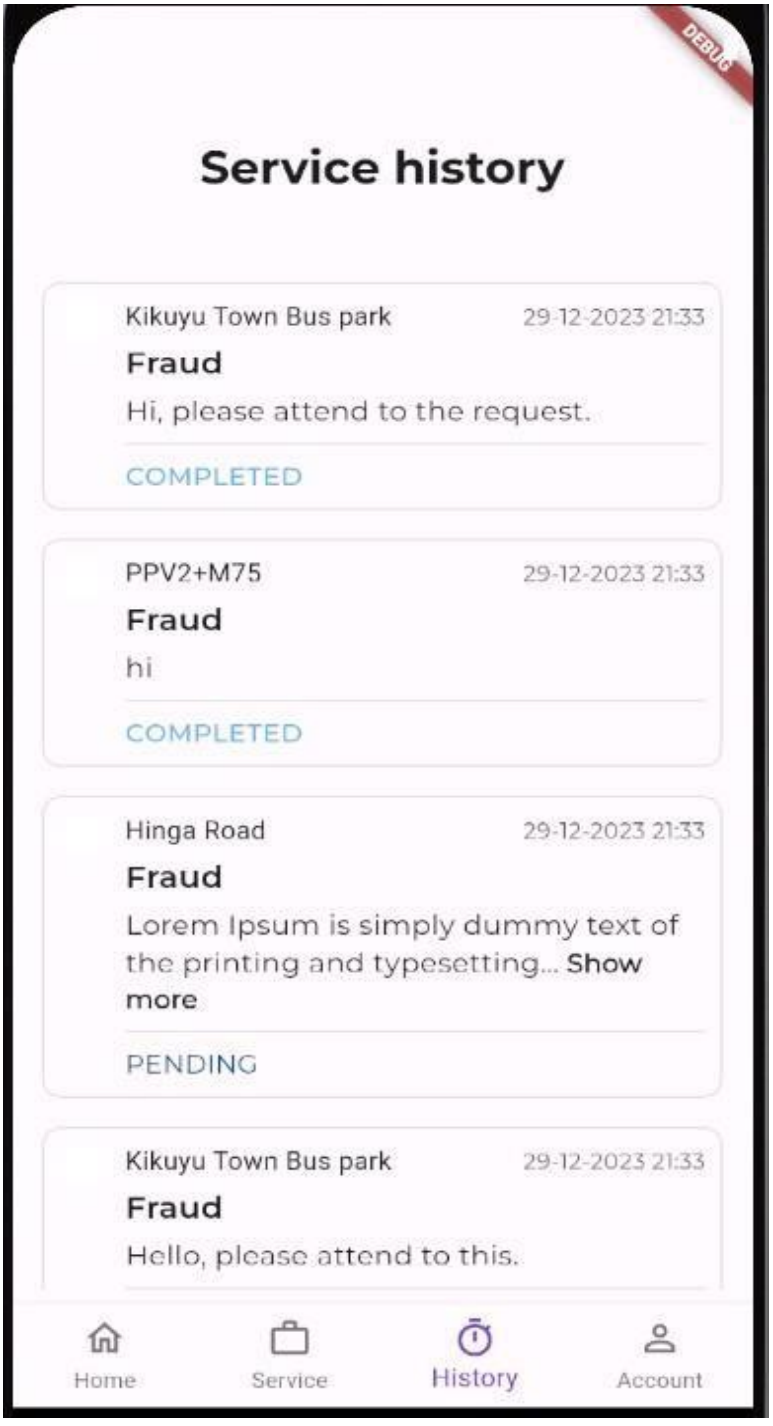
Landing Page



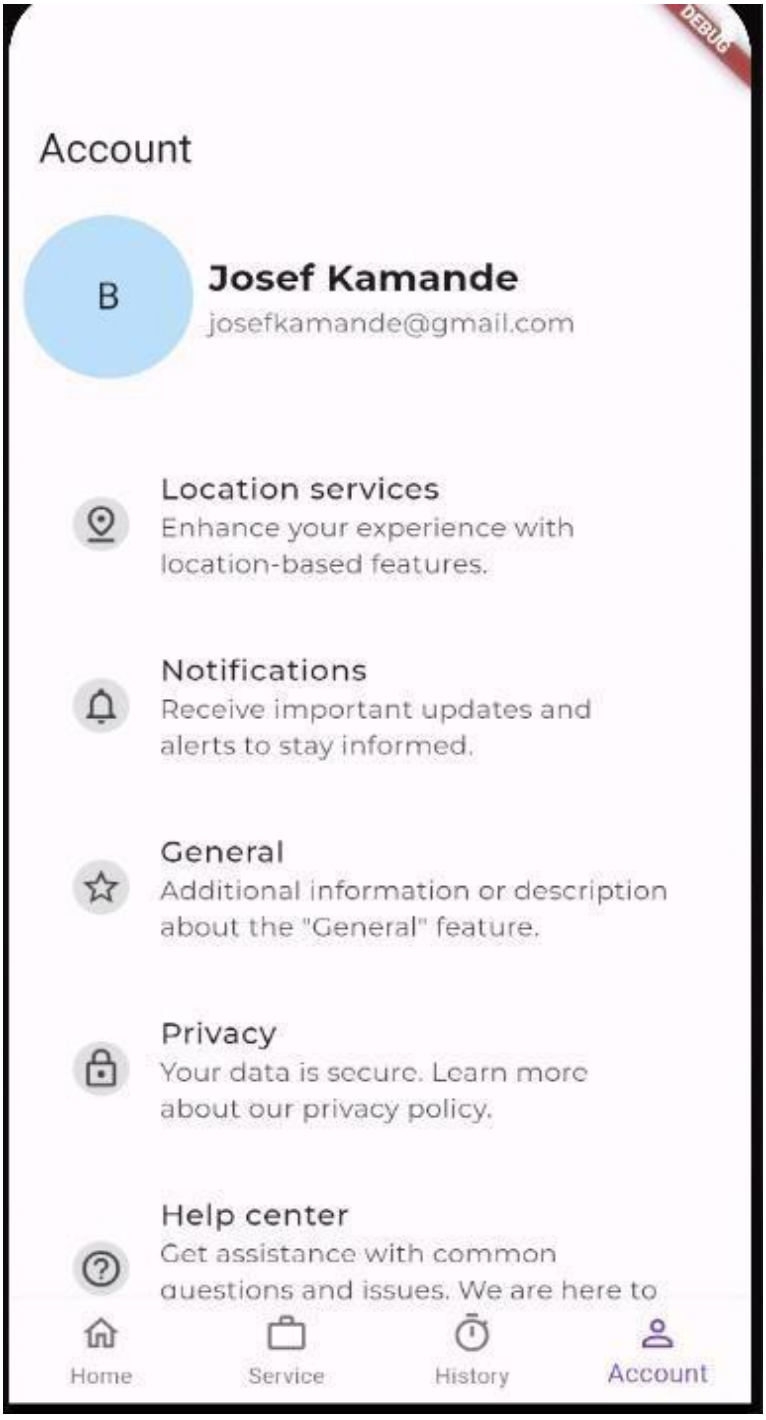
Ongoing Services – User submitting a Crime Request



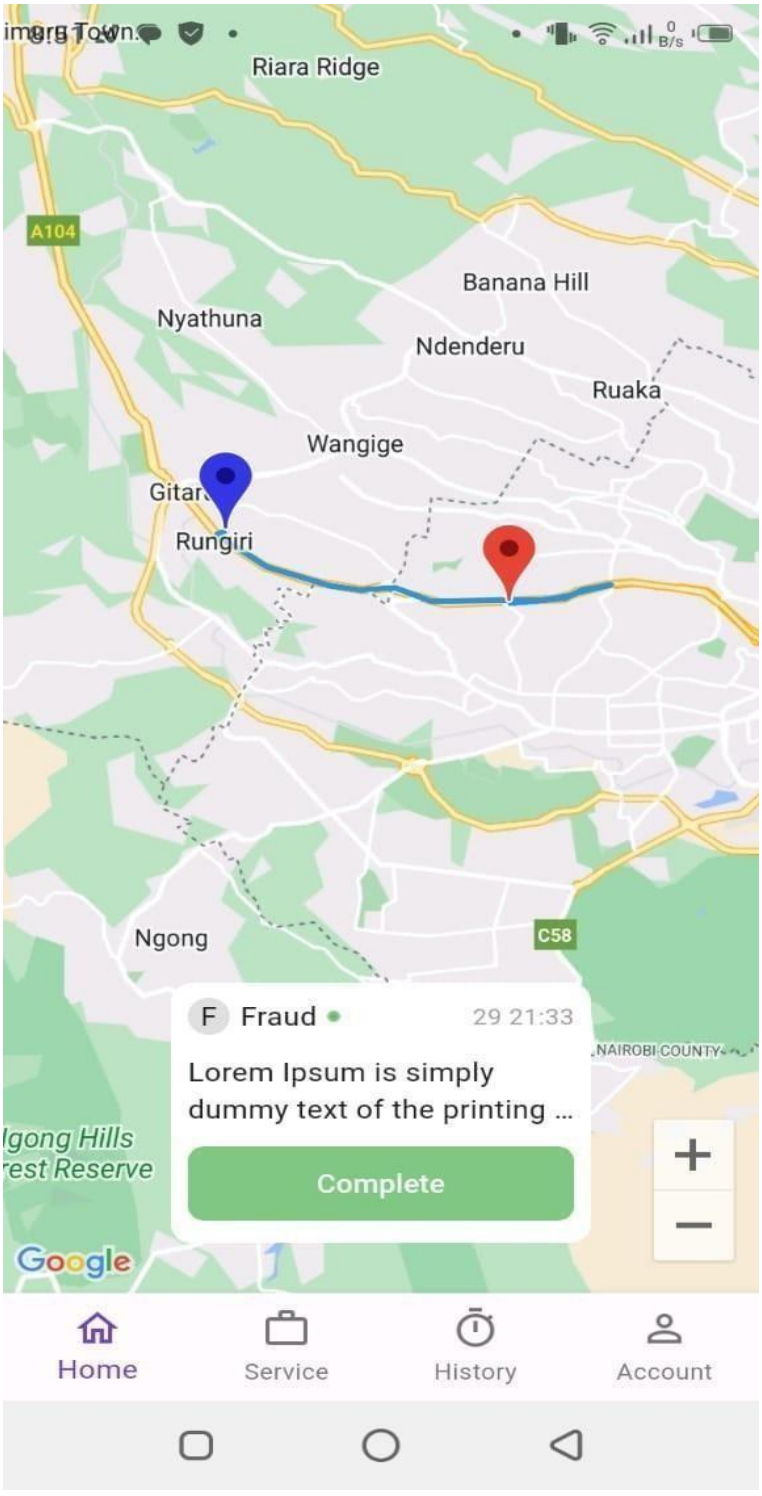
End User History Tab



End User Account



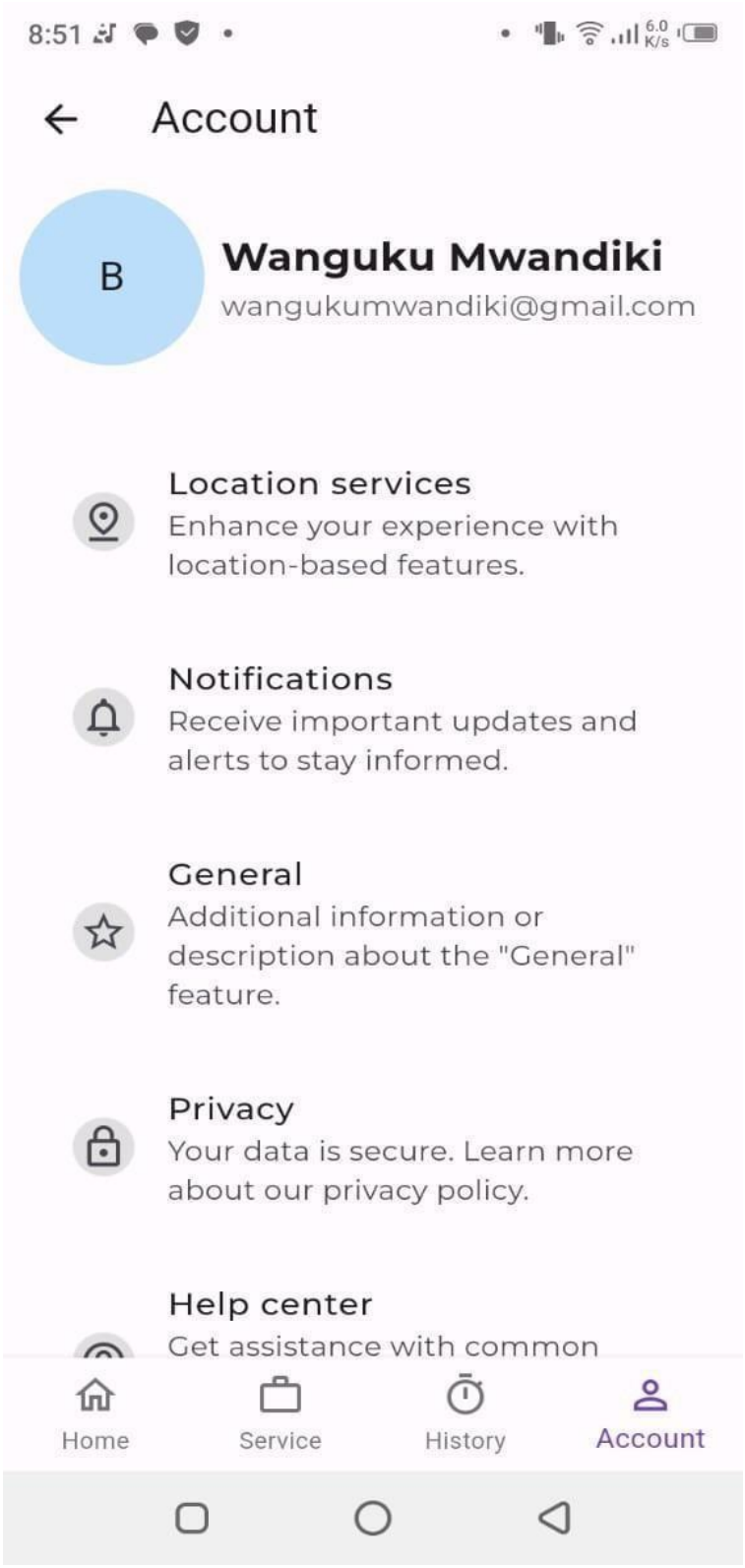
Law Enforcement Homepage



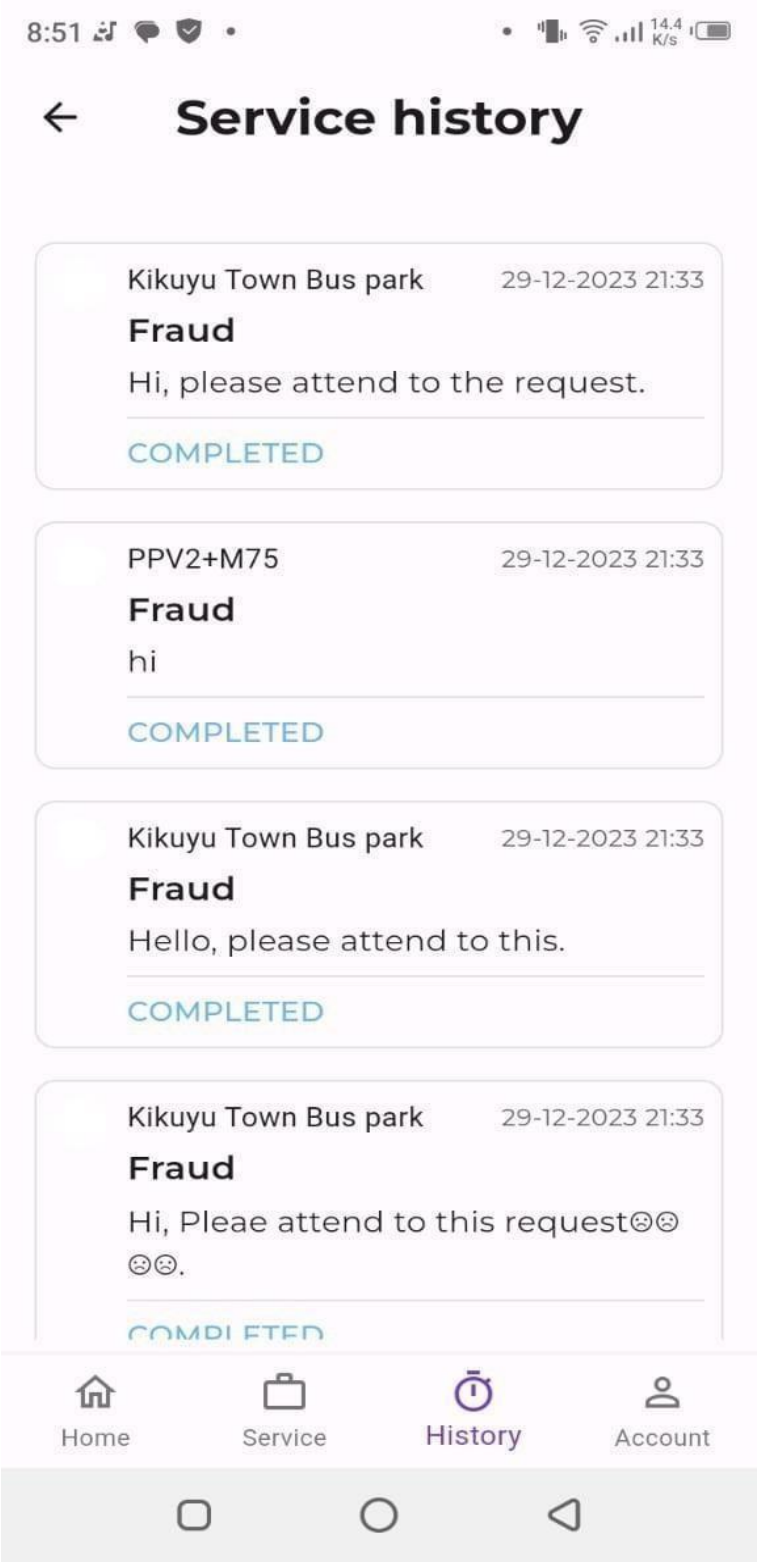
Law Enforcement Service Tab



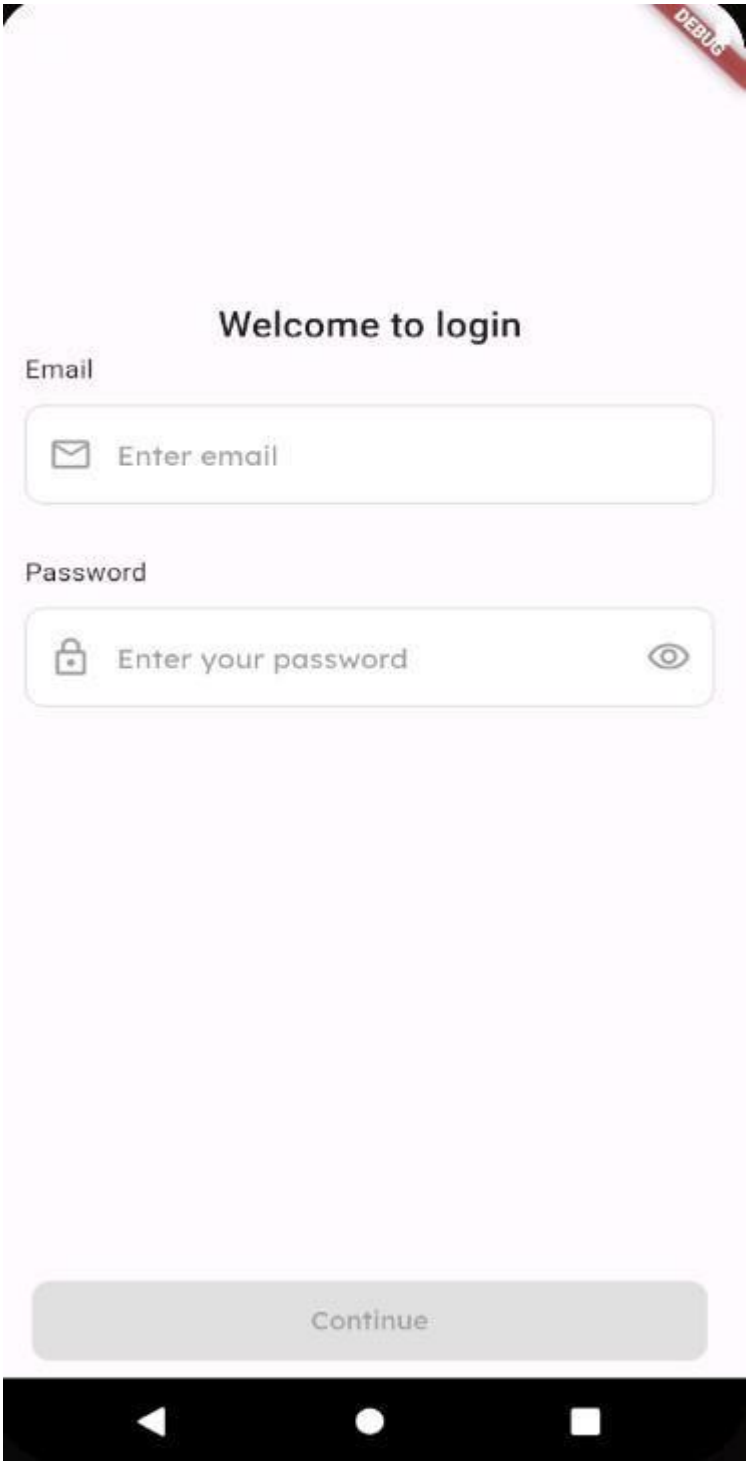
Law Enforcement Account



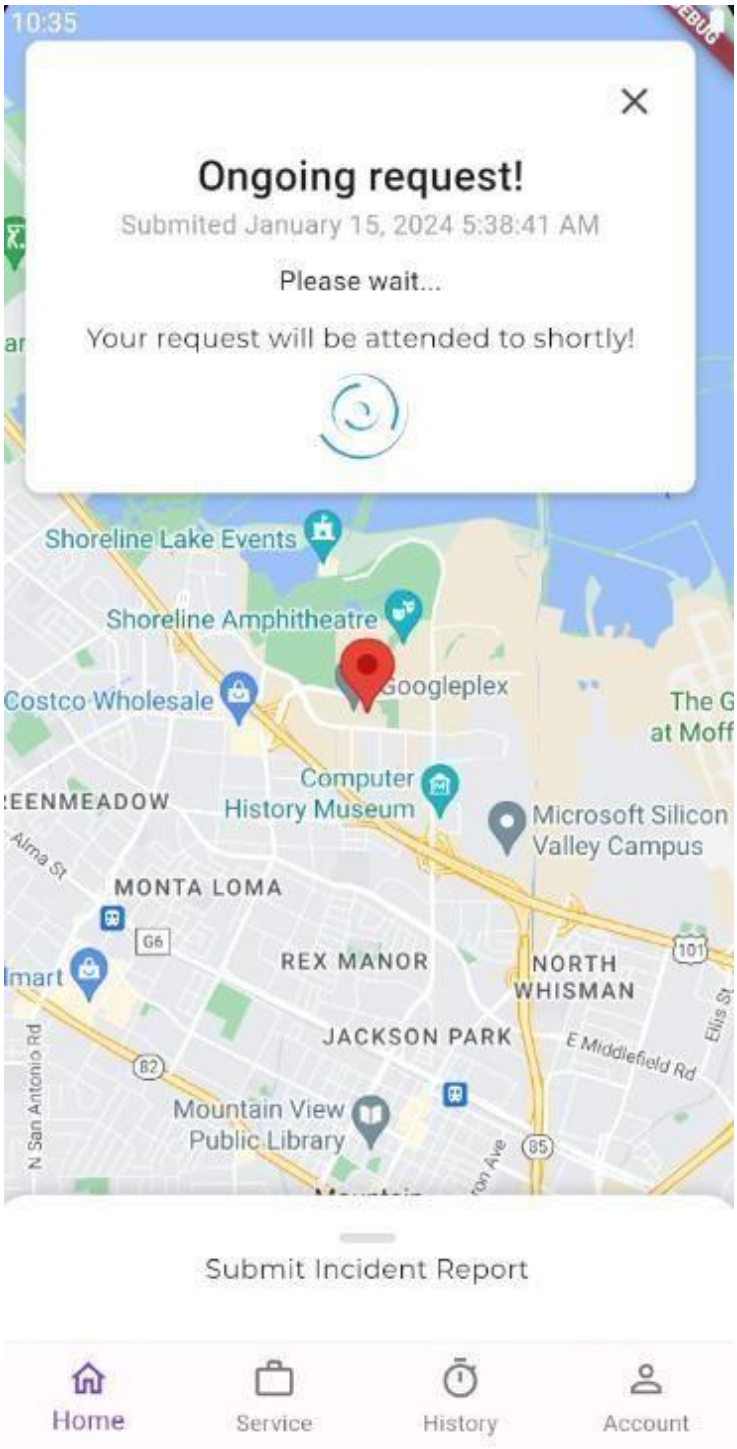
Law Officer History Tab



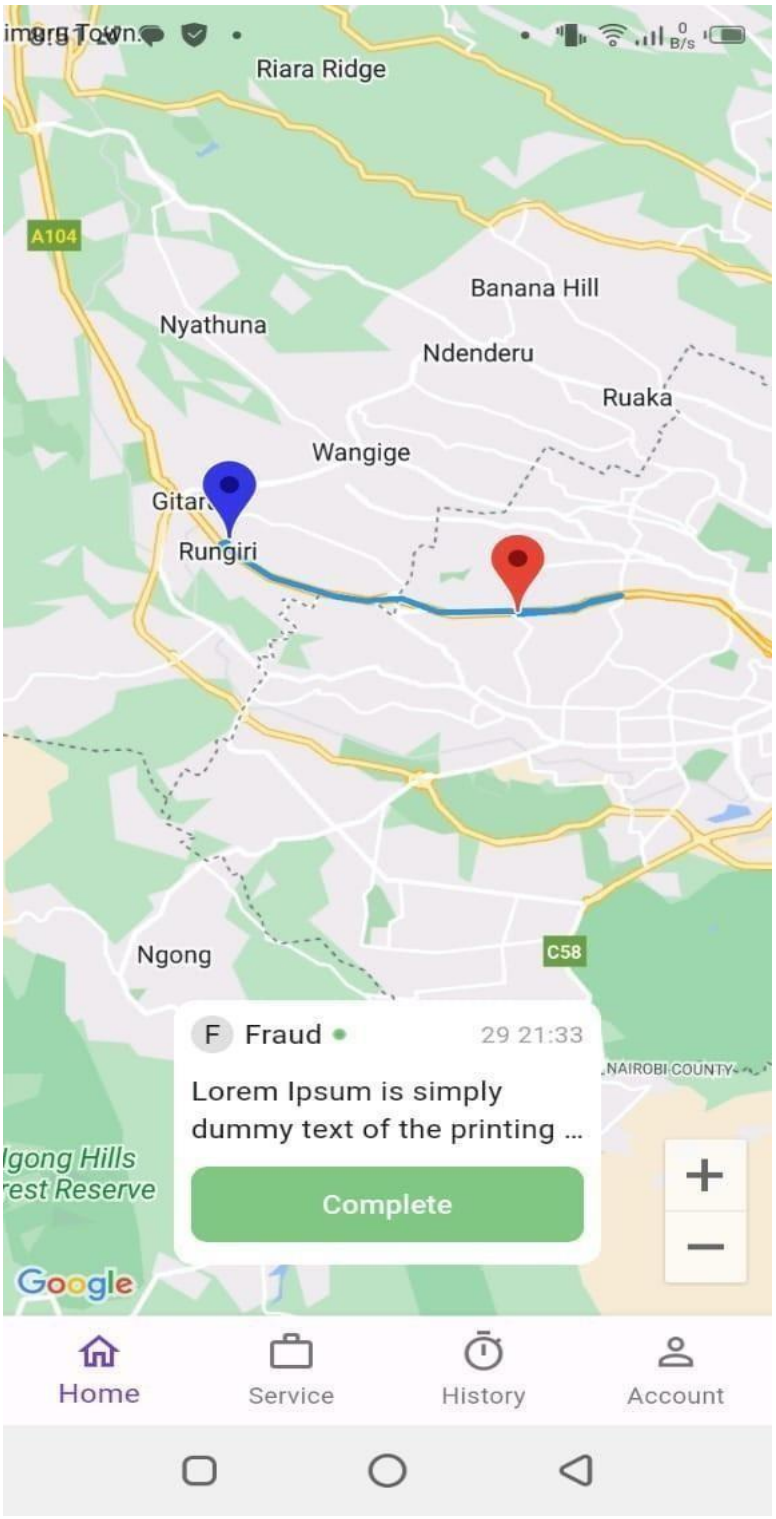
Appendix G: User Manual



A law enforcement officer and/or a reporter create an account



A user reports the crime, and it appears on the law enforcement dashboard



A law enforcement officer sees the crime and responds to it

Appendix H: Back End/Source Code

End user Home Page

```
import 'dart:async'; import 'dart:convert';
import 'package:crimereporter/apiservices/apiservices.dart';import
'package:crimereporter/model/crime_types.dart'; import 'package:crimereporter/model/officer.dart';
import 'package:crimereporter/pages/enduser/enduserbottomnav.dart';import
'package:geocoding/geocoding.dart' as geocoding; import 'package:google_fonts/google_fonts.dart';
import 'package:google_places_flutter/google_places_flutter.dart';import
'package:google_places_flutter/model/prediction.dart'; import 'package:flutter/material.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';import 'package:intl/intl.dart';
import 'package:location/location.dart';
import 'package:shared_preferences/shared_preferences.dart';import
'package:sliding_up_panel/sliding_up_panel.dart'; import 'package:uuid/uuid.dart';
import 'package:flutter_spinkit/flutter_spinkit.dart';import 'package:http/http.dart' as http;

class EndUserHome extends StatefulWidget {@override
    _EndUserHomeState createState() => _EndUserHomeState();
}
```

```
class _EndUserHomeState extends State<EndUserHome> {bool isPanelOpen = false;
    var uuid = const Uuid(); var location =
    Location();
    final Completer<GoogleMapController> _mapController =Completer<GoogleMapController>();


    LatLng? _currentPosition;@override

    void initState() {
        super.initState();getLocation();
        fetchUserData(); setState(() {});
    }

    Future<void> getLocation() async { bool serviceEnabled;
        PermissionStatus permissionGranted;

        serviceEnabled = await location.serviceEnabled();if (!serviceEnabled) {
            serviceEnabled = await location.requestService();if (!serviceEnabled) {
                return;
            }
        }


        permissionGranted = await location.hasPermission();if (permissionGranted ==
        PermissionStatus.denied) {
            permissionGranted = await
```

```
location.requestPermission();
            if (permissionGranted != PermissionStatus.granted) {return;
            }
        }


        // return location.getLocation();
        location.onLocationChanged.listen(
            (LocationData currentLocation) {
                if (currentLocation.altitude != null &&currentLocation.longitude != null) {
                    setState(() {
                        _currentPosition = LatLng(currentLocation.latitude!,
currentLocation.longitude!);
                        cameraPorision(_currentPosition!);
                        // print(_currentPosition);
                    });
                }
            },
        );
    }


    Future<void> cameraPorision(LatLng pos) async {final GoogleMapController
        controller = await
_mapController.future;
        CameraPosition newCameraPosition =CameraPosition(target:
pos, zoom: 13);
        await controller.animateCamera( CameraUpdate.newCameraPosition(newCameraPosition),
        );
    }
```

```
    ScrollController? controller;
    PanelController panelController = PanelController();

    final TextEditingController searcLocation =TextEditingController();
    final TextEditingController _incidentDescription =TextEditingController();
    DateFormat dateFormat = DateFormat();String? latitudeVal;
    String? longitudeVal;
    ApiServices apiServices = ApiServices();

    String? status;
    bool hideStatus = true;

    late Future<List<CrimeTypes>> crimeTypes; Future<List<Officer>> userReported =
    Future.value([]);Officer? userReportedDetails;

    String? userId;String?
    token;
    String crimeTypeId = ""; void fetchUserData()
    async {
        SharedPreferences prefs = await
SharedPreferences.getInstance();
        String jsonDataString = prefs.getString('userData') ??
";
        Map<String, dynamic> jsonData =
json.decode(jsonDataString);setState(() {
            userId = jsonData['data']['id'];
```

```
            token = jsonData['token'];
        });
        crimeTypes = apiServices.fetchCrimesType(token.toString());
        Timer.periodic(const Duration(seconds: 3), (timer) async
{
                                apiServices.getUserReported(token.toString(),
userId.toString());
            setState(() { userReported =
                    apiServices.getUserReported(token.toString(),userId.toString());
            });
            List<Officer> result = (await userReported)
                    .where((item) => item.status == 'Pending')
                    .toList();
            if (result.isNotEmpty && result != []) {setState(() {
                    status = result.last.status; userReportedDetails = result.last;
                });
                print('Status!!!!!!!!!!!! $status');
            } else { setState((){
                    status = 'Completed';
                });
                SharedPreferences prefs = await
SharedPreferences.getInstance();
                prefs.remove('submittedCrime');
              }
            });
    }
```

```
    bool isLoading = false;

    @override
    Widget build(BuildContext context) {return Scaffold(
            appBar: isPanelOpen
                    ? AppBar(
                            automaticallyImplyLeading: false,title: Row(
                        children: [
                            Expanded(
                                child: GooglePlaceAutoCompleteTextField(textEditingController:
                                    searcLocation, googleAPIKey:
"AIzaSyA7zmasenUJGF1QDQrmPdrOzLxBL4864s0",
                                    inputDecoration: const
InputDecoration(
                                        enabledBorder: UnderlineInputBorder(borderSide:
                                            BorderSide.none,
                                        ),
                                        focusedBorder: UnderlineInputBorder(borderSide:
                                            BorderSide.none,
                                        ),
                                        hintText: 'Search your location',hintStyle: TextStyle(
                                            color: Colors.grey,
                                        ),
                                        prefixIcon: Icon( Icons.search, color:
                                            Colors.grey,
                                          ),
```

```
                                    ),
                                    debounceTime: 800, countries: const ["ke"],
                                    isLatLngRequired:
                                        true, // if you requiredcoordinates from place detail
                                    getPlaceDetailWithLatLng: (Prediction
prediction) {
                                        setState(() { latitudeVal =
prediction.lat.toString();
                                            longitudeVal =
prediction.lng.toString();


                                        });
                                    },
                                    itemClick: (Prediction prediction) {searcLocation.text =
prediction.description.toString();
                                        searcLocation.selection =
TextSelection.fromPosition(
                                            TextPosition(offset:
prediction.description!.length),
                                        );
                                    },
                                    itemBuilder: (context, index,Prediction prediction) {
                                        return Container(
                                            padding: const EdgeInsets.all(10),child: Row(
                                                children: [
                                                    const Icon(Icons.location_on),const SizedBox(
```

```
                                                    width: 7,
                                                ),
                                                Expanded(child:
Text(prediction.description ?? ""),
                                                )
                                            ],
                                        ),
                                    );
                                },
                                seperatedBuilder: const Divider(),isCrossBtnShown: true,
                            ),
                        ),
                    ],
                ),
                backgroundColor: Colors.white,elevation: 0,
                toolbarHeight: 80.0,
            )
        : null,
    body: _currentPosition == null
        ? const Center(
            child:      SpinKitRipple(      color:
                Colors.blue,size: 50.0,
            ),
        )
        : SlidingUpPanel(
            // minHeight: MediaQuery.of(context).size.height *
0.07,
            minHeight: MediaQuery.of(context).size.height
```

```
* 0.1,

* 0.6,                    maxHeight: MediaQuery.of(context).size.heightborderRadius:
                                const BorderRadius.vertical(top:


Radius.circular(18)),
                          controller: panelController,parallaxEnabled: true,
                          parallaxOffset: 0.5, onPanelOpened: () {
                              setState(() { isPanelOpen = true;
                              });
                          },
                          onPanelClosed: () {setState(() {
                                isPanelOpen = false;
                              });
                          },
                          body: buildMap(),
                          panelBuilder: (controller) => Stack(children: [
                                ListView(
                                    controller: controller, padding:
                                    EdgeInsets.zero,children: [
                                        const SizedBox(height: 20,
                                        ),
                                        Visibility(
                                            visible: isPanelOpen == false,child: Column(
```

```
                                        children: [Center(
                                            child: GestureDetector(onTap: () {
                                                panelController.isPanelOpe

                                                    ?
n
                                                    :

panelController.close()


panelController.open();

                                                },
                                                child: Container(height: 5,
                                                    width: 30,
                                                    decoration: BoxDecoration(color:

                                                        borderRadius:
Colors.grey.shade300, BorderRadius.circular(12)),

                                                ),
                                            ),
                                        ),
                                        const SizedBox(height: 4,
                                        ),
                                        Center( child: Text(
                                            'Submit Incident Report',style:

                                            ),
GoogleFonts.montserrat(),                       ),
```

```
                                    const SizedBox(height: 50,
                                      ),
                                  ],
                                ),
                              ),
                            Padding( padding:const
EdgeInsets.symmetric(horizontal: 15),
                                child: Column( crossAxisAlignment:
CrossAxisAlignment.start,
                                    children: [Text(
                                            'Pick incident type',
                                            style: GoogleFonts.montserrat(textStyle: const
                                                TextStyle(
                                                    fontWeight:
FontWeight.bold,
                                                    fontSize: 17,
                                                ),
                                            ),
                                        ),
                                        const SizedBox(height: 8,
                                        ),
                                        // DropdownButtonFormField(
                                        //         decoration:
InputDecoration(
                                        //             filled: true,
                                        //             fillColor:
Colors.grey.shade100,
```

```
//                 contentPadding: const
EdgeInsets.all(5),
//                 enabledBorder:
OutlineInputBorder(
//                     borderSide:
//                         BorderSide(color:
Colors.grey.shade200), BorderRadius.circular(10.0),
//                     borderRadius:
//                 ),
//                 focusedBorder:
OutlineInputBorder(
//                     borderSide:
//                         BorderSide(color:
Colors.grey.shade200), BorderRadius.circular(10.0),
//                     borderRadius:
//                 ),
//                 hintText: 'Pick type of
crime', GoogleFonts.montserrat(
//                 hintStyle:
//                     textStyle:
//                         const
TextStyle(color: Colors.grey),
//                 ),
//             ),
//             items:
criminalTypes.map((String value) {
//                 return
DropdownMenuItem<String>(
//                     value: value,
```

```
//                     child: Text(value),
//                 );
//             }).toList(),
//             onChanged: (value) {},
//             isExpanded: true,
// ),


FutureBuilder( future: crimeTypes,
    builder: (context, snapshot) {if
        (snapshot.connectionState

==

                ConnectionState.waiting)

{

            return const Center(child:

CircularProgressIndicator(),

            );
        } else if

(snapshot.hasError) {

            return Center(child: Text(
                    'Error:

${snapshot.error.toString()}'),


            );
        } else if (snapshot.hasData

&&

                snapshot.data != null) {var crimeTypesList =

snapshot.data!;

            return Container(height: 100,
```

```
                                                                    width:
t).size.width * 0.9, ListView.separated(                                MediaQuery.of(contexchild:

Axis.horizontal, crimeTypesList.length,                             scrollDirection: itemCount:

(context, index) {                                                  separatorBuilder:

                                                                        return Container(width: 5,
                                                                        );
                                                                    },
                                                                    itemBuilder: (context,var crime =
index) { crimeTypesList[index];

                                                                        return Stack( clipBehavior:

Clip.none,                                                                  children: [ GestureDetector(
                                                                                onTap: () { setState(()


                                                                                    crimeTypeI

{                                                                                   });
                                                                                    print(crimeT
d = crime.id;

ypeId);
```

```
                                                                    },
                                                                    child:
Container(
                                                                        height: 80,
                                                                        width: 80,
                                                                        decoration:
BoxDecoration( Colors.green.shade50,
                                                                            color: borderRadi
us: Radius.circular(8),
                                                                                        Border
Column(
                                                                        ),
                                                                        child:
ignment:
                                                                            mainAxisAl
isAlignment.center,[
                                                                                    MainAx
r(
                                                                            children:
const Icon( s.security_outlined,
                                                                                Containechild:
                                                                                        Icon
                                                                                    ),
                                                                                ),
                                                                                Text(
                                                                                    crime.
```

```
crime_name, ign:
                                                                                textAl
xtAlign.center,
                                                                                    Te
                                                                                ),
                                                                            ],
                                                                        ),
                                                                    ),
                                                                ),
                                                            crimeTypeId ==
crime.id Positioned(
                                                                ?
4,
                                                                                right: -
GestureDetector(() {                                                        top: -2,child:
                                                                                onTap:
tate(() { imeTypeId = "";
                                                                                    setS
                                                                                    cr
Container(ration:
                                                                                });
                                                                            },
                                                                            child:
                                                                                deco
```

```
BoxDecoration(
                                                                    co
lor: Colors
    .blue.shade50,
                                                                    sh
ape:
    BoxShape.circle,
                                                                    bo
xShadow: [
BoxShadow(
    color: Colors.grey
                .withOpacity(0.4),
    spreadRadius: 3,
    blurRadius: 5,offset:
                    const Offset(
        0,
        3,
```

```
      ),

),
                                                                                                    ],
                                                                                                ),
                                                                                                chil
d: const Icon( ons.close_outlined,
                                                                                                    Icco
lor: Colors.grey,
                                                                                                ),
                                                                                            ),
                                                                                        ),
                                                                                    )
Container()                                                                                 :

                                                                    ],
                                                                );
                                                            },
                                                        ),
                                                    );
                                                } else {

                                                    return Text('No data
available');
                                                }
                                            },
                                        ),
                                    ],
                                ),
                            ),
                                const SizedBox(
```

```
                                        height: 25,
                                    ),
                                    Padding( padding: const
EdgeInsets.symmetric(horizontal: 15),
                                        child: Column( crossAxisAlignment:
CrossAxisAlignment.start,
                                            children: [Text(
                                                'Describe incident',
                                                style: GoogleFonts.montserrat(textStyle: const
                                                    TextStyle(
                                                        fontWeight:
FontWeight.bold,
                                                        fontSize: 17,
                                                    ),
                                                ),
                                            ),
                                            const SizedBox(height: 8,
                                            ),
                                            TextFormField(controller:
_incidentDescription,
                                                maxLines: 5,
                                                decoration: InputDecoration(filled: true,
                                                    fillColor: contentPadding: const
Colors.grey.shade100,
EdgeInsets.all(5),
```

```
                                                    enabledBorder:
OutlineInputBorder(
                                                        borderSide:
                                                            BorderSide(color:
Colors.grey.shade200), BorderRadius.circular(10.0),
                                                        borderRadius:
                                                    ),
                                                    focusedBorder:
OutlineInputBorder(
                                                        borderSide:
                                                            BorderSide(color:
Colors.grey.shade200), BorderRadius.circular(10.0),
                                                        borderRadius:
                                                    ),
                                                    hintText: 'Describe crimehintStyle:
here(optional)', GoogleFonts.montserrat(
                                                        textStyle:
                                                            const TextStyle(color:
Colors.grey),
                                                      ),
                                                  ),
                                              ),
                                          ],
                                      ),
                                  )
                              ],
                          ),
                              const SizedBox(height: 25),
```

```
                                    crimeTypeId == "" ||
_incidentDescription.text.isEmpty
                                ? Padding(
                                    padding: const
EdgeInsets.symmetric(vertical: 25),
                                    child: Align(alignment:
Alignment.bottomCenter,
                                        child: Container(width:
MediaQuery.of(context).size.width * 0.9,
                                            height: 60,
                                            child: OutlinedButton(onPressed: () {},
                                                style: ButtonStyle(
                                                    shape:
MaterialStateProperty.all(
                                                        RoundedRectangleBorder(borderRadius:

BorderRadius.circular(                                                          10.0,
                                                        ),
                                                    ),
                                                ),
                                                backgroundColor:
MaterialStateProperty.all(
                                                    Colors.grey.shade200,
                                                ),
                                                side:

MaterialStateProperty.all(                                                      BorderSide(color:


Colors.grey.shade200),
```

```
                                                    ),
                                                ),
                                                child: Text('HELP',
                                                    style:
GoogleFonts.montserrat(TextStyle(),
                                                        textStyle: const

                                                    ),
                                                ),
                                            ),
                                        ),
                                    ),
                                )
                            : Padding(
                                    padding: const
EdgeInsets.symmetric(vertical: 25),
                                    child: Align(alignment:
Alignment.bottomCenter,
                                        child: Container(width:
MediaQuery.of(context).size.width * 0.9,
                                            height: 60,
                                            child: OutlinedButton( onPressed: () async {
                                                setState(() { isLoading = true;
                                                });
                                                FocusScope.of(context).unf
ocus();
                                                var result = await
```

```
apiServices.subnmitCrime(
                                                    userId.toString(), crimeTypeId.toString()

,                                                   _incidentDescription.t

ext,                                                latitudeVal.toString()

,                                                   longitudeVal.toString(

),                                                  token.toString());
                                                // ignore:
use_build_context_synchronously
                                                Navigator.push( context,
                                                    MaterialPageRoute(
                                                        builder: (context) =>

EndUserBottomNav(),                                 ),
                                                );


                                                const snackBar = SnackBar(content:
                                                    Text('Updated
successfully'), Duration(seconds: 3),
                                                    duration:

                                                );

                                                // ignore: ScaffoldMessenger.of(conte

use_build_context_synchronouslyxt)
```

```
                        .showSnackBar(snackBar
);
                    setState(() { isLoading = false;
                    });
                },
                style: ButtonStyle(shape:
MaterialStateProperty.all(
                        RoundedRectangleBorder(borderRadius:
BorderRadius.circular(
                                10.0,
                            ),
                        ),
                    ),
                    backgroundColor:
MaterialStateProperty.all(
                        Colors.blue.shade400,
                    ),
                    side:
MaterialStateProperty.all(
                        BorderSide(color:
Colors.grey.shade200),
                    ),

                ),
                child: Text( isLoading == false
                            ? 'HELP'
                            : 'Please wait...',style:
GoogleFonts.montserrat(
```

```
                                                    textStyle: const
TextStyle(
                                                        color: Colors.white,fontWeight:

FontWeight.bold),

                                                    ),
                                                ),
                                            ),
                                        ),
                                    ),
                                )
                            ],
                        ),
                    ),
            );
    }


    Widget buildMap() => Stack(children: [
                    GoogleMap(
                        mapType: MapType.normal, initialCameraPosition: CameraPosition(
                            target: _currentPosition!,zoom: 13,
                        ),
                        markers: {Marker(
                                markerId: const
MarkerId('_currentPosition'),
                                position: _currentPosition!,
                                icon: BitmapDescriptor.defaultMarker,
                            ),
```

```
                        },
                        onMapCreated: ((GoogleMapController controller)
=>
                                _mapController.complete(controller)),
                    ),
                    Positioned(top: 35,
                        left: 16,
                        child: status == 'Pending' && hideStatus == true
                                ? GestureDetector( onTap: () {}, child:
                                        Container(
                                            width:
MediaQuery.of(context).size.width * 0.9,
                                            // height: 80,
                                            padding: const EdgeInsets.all(8),decoration: BoxDecoration(
                                                color: Colors.white,borderRadius:
BorderRadius.circular(8),
                                                boxShadow: [
                                                    BoxShadow(
                                                        color:
Colors.black.withOpacity(0.2),
                                                        blurRadius: 4,
                                                        offset: const Offset(0, 2),
                                                    ),
                                                ],
                                            ),
                                            child: Column( crossAxisAlignment:
CrossAxisAlignment.center,
```

```
                                        children: [Row(
                                                mainAxisAlignment:
MainAxisAlignment.end,
                                                children: [ IconButton(
                                                        onPressed: () { hideStatus = false;
                                                        },
                                                        icon: const
Icon(Icons.close),
                                                    ),
                                                ],
                                            ),
                                            const Text(
                                                'Ongoing request!',style: TextStyle(
                                                        fontSize: 22, fontWeight:
FontWeight.w500),

                                            ),
                                            Text(
                                                'Submited
${dateFormat.format(userReportedDetails!.createdAt!)}',
                                                style: const TextStyle(color:
Colors.grey),
                                            ),
                                            const SizedBox(height: 10,
                                            ),
                                            userReportedDetails!.responded_to_
by != null
                                                    ? Row(
```

```
                                                    mainAxisAlignment:
MainAxisAlignment.center,
                                                    children: [Text(
                                                            'Your request was
accepted',
                                                            style: TextStyle(color:
Colors.blue.shade600,                                                       fontWeight: fontSize:16),
FontWeight.w500,
                                                        ),
                                                        const SizedBox(width: 8,
                                                        ),
                                                        Icon(
                                                            Icons.check_circle,color:
Colors.blue.shade600,                                                       size: 17,
                                                        )
                                                    ],
                                                )
                                            :
userReportedDetails!.cancelled_by != null
                                                ? SizedBox(
                                                        width: 220,child:
                                                        Text(
                                                            'Cancelled but will
be attended to shortly.Please wait...',
                                                            style: TextStyle(
```

```
                                                    color:
Colors.red.shade600,
                                                        ),
                                                    ),
                                                )
                                            : Text('Please wait...'),const SizedBox(
                                    height: 10,
                                ),
                                Column( children: [
                                    Text(
                                        userReportedDetails!.respond

ed_to_by != null                                    ? 'This disappers once

you have been served!' attended to shortly!',       : 'Your request will bestyle:

GoogleFonts.montserrat(),
                                    ),
                                    const SizedBox(height: 4,
                                    ),
                                    const SpinKitSpinningLines(color:    Colors.blue,
                                        size: 50.0,
                                    ),
                                    const SizedBox(height: 10,
                                    ),
                                ],
```

```
                                                )
                                            ],
                                        ),
                                    ),
                                )
                            : status == 'ongoing'
                                ? GestureDetector( onTap: () {}, child:
                                    Container(
                                        width:
MediaQuery.of(context).size.width * 0.9,
                                        height: 80,
                                        padding: const EdgeInsets.all(8),decoration: BoxDecoration(
                                            color: Colors.white,borderRadius:
BorderRadius.circular(8),
                                            boxShadow: [
                                                BoxShadow(
                                                    color:
Colors.black.withOpacity(0.2),
                                                    blurRadius: 4,
                                                    offset: const Offset(0, 2),
                                                ),
                                            ],
                                        ),
                                        child: Column( crossAxisAlignment:
CrossAxisAlignment.start,
                                            children: [Row(
                                                    mainAxisAlignment:
```

```
                                    MainAxisAlignment.spaceB
etween,
                                children:[Row(
                                    children: [Container(
                                        height: 25,
                                        width: 25,
                                        decoration:

BoxDecoration( Colors.blue.shade200,

                                            color:

                                            shape:

BoxShape.circle, Border.all( Colors.white,

                                            border:

width: 2),

                                                color:

                                        ),
                                        child: const Center(child: Text(
TextStyle(
                                            'J',
                                            style:

Colors.white,

                                                color: fontSize: 12,

                                            ),
                                        ),
                                    ),
                                ),
                                const SizedBox(
```

```
                                        width: 3,
                                    ),
                                    Row(
                                        children: [Text(
                                                'Joe Biden',style:
GoogleFonts.montserrat(const TextStyle(                                 textStyle:
ht: Weight.bold)),                                                          fontWeig
                                                                                Font
                                        ),
                                        const SizedBox(width: 4,
                                        ),
                                        Text(
GoogleFonts.montserrat(const TextStyle(                                     '2 min ago',style:
Colors.grey)),                                                          textStyle:
                                                                            color:
                                        )
                                    ],
                                )
                            ],
                        ),
                        GestureDetector(
```

```
                                    onTap: () {
                                        setState(() { hideStatus = false;
                                        });
                                    },
                                    child: const Icon( Icons.close, color:
                                        Colors.grey,
                                    ),
                                )
                        ],
                    ),
                    const SizedBox(height: 10,
                    ),
                    Row(
                        children:[Text(
                                'You have currentstyle:

ongoing service', GoogleFonts.montserrat(),
                            ),
                            const SizedBox(width: 4,
                            ),
                            const SpinKitRipple( color:
                                Colors.blue,size: 20.0,
                             )
                         ],
                     )
```

```
                                                    ],
                                                ),
                                            ),
                                        )
                                    : Container(),
                            ),
                        ],
                    );
}
```

l user home page.

```
import 'dart:async'; import 'dart:convert';
import 'package:crimereporter/apiservices/apiservices.dart';import
'package:crimereporter/model/crime_types.dart'; import 'package:crimereporter/model/officer.dart';
import 'package:crimereporter/pages/enduser/enduserbottomnav.dart';import
'package:geocoding/geocoding.dart' as geocoding; import 'package:google_fonts/google_fonts.dart';
import 'package:google_places_flutter/google_places_flutter.dart';import
'package:google_places_flutter/model/prediction.dart'; import 'package:flutter/material.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';import 'package:intl/intl.dart';
import 'package:location/location.dart';
import 'package:shared_preferences/shared_preferences.dart';
```

```
import 'package:sliding_up_panel/sliding_up_panel.dart';import 'package:uuid/uuid.dart';
import 'package:flutter_spinkit/flutter_spinkit.dart';import 'package:http/http.dart' as http;

class EndUserHome extends StatefulWidget {@override
    _EndUserHomeState createState() => _EndUserHomeState();
}

class _EndUserHomeState extends State<EndUserHome> {bool isPanelOpen = false;
    var uuid = const Uuid(); var location =
    Location();
    final Completer<GoogleMapController> _mapController =Completer<GoogleMapController>();


    LatLng? _currentPosition;@override
    void initState() {
        super.initState();getLocation();
        fetchUserData(); setState(() {});
    }

    Future<void> getLocation() async { bool serviceEnabled;
        PermissionStatus permissionGranted;

        serviceEnabled = await location.serviceEnabled();
```

```
        if (!serviceEnabled) {
            serviceEnabled = await location.requestService();if (!serviceEnabled) {
                return;
            }
        }

        permissionGranted = await location.hasPermission();if (permissionGranted ==
        PermissionStatus.denied) {
            permissionGranted = await
location.requestPermission();
            if (permissionGranted != PermissionStatus.granted) {return;
            }
        }

        // return location.getLocation();
        location.onLocationChanged.listen(
            (LocationData currentLocation) {
                if (currentLocation.altitude != null &&currentLocation.longitude != null) {
                    setState(() {
                        _currentPosition = LatLng(currentLocation.latitude!,
currentLocation.longitude!);
                        cameraPorision(_currentPosition!);
                        // print(_currentPosition);
                    });
                }
            },
        );
    }
```

```
    Future<void> cameraPorision(LatLng pos) async {final GoogleMapController
        controller = await
_mapController.future;
        CameraPosition newCameraPosition =CameraPosition(target:
pos, zoom: 13);
        await controller.animateCamera( CameraUpdate.newCameraPosition(newCameraPosition),
        );
    }


    ScrollController? controller;
    PanelController panelController = PanelController();

    final TextEditingController searcLocation =TextEditingController();
    final TextEditingController _incidentDescription =TextEditingController();
    DateFormat dateFormat = DateFormat();String? latitudeVal;
    String? longitudeVal;
    ApiServices apiServices = ApiServices();

    String? status;
    bool hideStatus = true;

    late Future<List<CrimeTypes>> crimeTypes; Future<List<Officer>> userReported =
    Future.value([]);Officer? userReportedDetails;

    String? userId;String?
    token;
```

```
    String crimeTypeId = ""; void fetchUserData()
    async {
        SharedPreferences prefs = await
SharedPreferences.getInstance();
        String jsonDataString = prefs.getString('userData') ??
'';
        Map<String, dynamic> jsonData =
json.decode(jsonDataString);setState(() {
            userId = jsonData['data']['id'];token = jsonData['token'];
        });
        crimeTypes = apiServices.fetchCrimesType(token.toString());
        Timer.periodic(const Duration(seconds: 3), (timer) async
{
                                apiServices.getUserReported(token.toString(),
userId.toString());
            setState(() { userReported =
                    apiServices.getUserReported(token.toString(),userId.toString());
            });
            List<Officer> result = (await userReported)
                    .where((item) => item.status == 'Pending')
                    .toList();
            if (result.isNotEmpty && result != []) {setState(() {
                    status = result.last.status; userReportedDetails = result.last;
                });
                print('Status!!!!!!!!!!!! $status');
```

```
            } else { setState((){
                    status = 'Completed';
                });
                SharedPreferences prefs = await
SharedPreferences.getInstance();
                prefs.remove('submittedCrime');
              }
            });
    }


    bool isLoading = false;

    @override
    Widget build(BuildContext context) {return Scaffold(
            appBar: isPanelOpen
                    ? AppBar(
                            automaticallyImplyLeading: false,title: Row(
                        children: [
                            Expanded(
                                child: GooglePlaceAutoCompleteTextField(textEditingController:
                                    searcLocation, googleAPIKey:
"AIzaSyA7zmasenUJGF1QDQrmPdrOzLxBL4864s0",
                                    inputDecoration: const
InputDecoration(
                                        enabledBorder: UnderlineInputBorder(borderSide:
                                            BorderSide.none,
                                        ),
                                        focusedBorder: UnderlineInputBorder(
```

```
                                        borderSide: BorderSide.none,
                                    ),
                                hintText: 'Search your location',hintStyle: TextStyle(
                                    color: Colors.grey,
                                ),
                                prefixIcon: Icon( Icons.search, color:
                                    Colors.grey,
                                ),
                            ),
                            debounceTime: 800, countries: const ["ke"],
                            isLatLngRequired:
                                true, // if you requiredcoordinates from place detail
                            getPlaceDetailWithLatLng: (Prediction
prediction) {
                                setState(() { latitudeVal =
prediction.lat.toString();
                                    longitudeVal =
prediction.lng.toString();


                                });
                            },
                            itemClick: (Prediction prediction) {searcLocation.text =
prediction.description.toString();
                                searcLocation.selection =
TextSelection.fromPosition(
                                    TextPosition(offset:
prediction.description!.length),
```

```
                                        );
                                    },
                                    itemBuilder: (context, index,Prediction prediction) {
                                        return Container(
                                            padding: const EdgeInsets.all(10),child: Row(
                                                children: [
                                                    const Icon(Icons.location_on),const SizedBox(
                                                        width: 7,
                                                    ),
                                                    Expanded(child:
Text(prediction.description ?? ""),
                                                    )
                                                ],
                                            ),
                                        );
                                    },
                                    seperatedBuilder: const Divider(),isCrossBtnShown: true,
                                ),
                            ),
                        ],
                    ),
                    backgroundColor: Colors.white,elevation: 0,
                    toolbarHeight: 80.0,
                )
            : null,
        body: _currentPosition == null
```

```
                    ? const Center(
                            child:          SpinKitRipple(        color:
                                Colors.blue,size: 50.0,
                            ),
                        )
                    : SlidingUpPanel(
                            // minHeight: MediaQuery.of(context).size.height *
0.07,
                            minHeight: MediaQuery.of(context).size.height
* 0.1,

* 0.6,                       maxHeight: MediaQuery.of(context).size.heightborderRadius:
                                    const BorderRadius.vertical(top:


Radius.circular(18)),
                            controller: panelController,parallaxEnabled: true,
                            parallaxOffset: 0.5, onPanelOpened: () {
                                setState(() { isPanelOpen = true;
                                });
                            },
                            onPanelClosed: () {setState(() {
                                    isPanelOpen = false;
                                });
                            },
                            body: buildMap(),
                            panelBuilder: (controller) => Stack(children: [
```

```
                                ListView(
                                    controller: controller, padding:
                                    EdgeInsets.zero,children: [
                                        const SizedBox(height: 20,
                                        ),
                                        Visibility(
                                            visible: isPanelOpen == false,child: Column(
                                                children: [Center(
                                                        child: GestureDetector(onTap: () {
                                                                panelController.isPanelOpe
n
                                                                        ?
            panelController.close()
                                                                        :
            panelController.open();
                                                            },
                                                        child: Container(height: 5,
                                                            width: 30,
                                                            decoration: BoxDecoration(color:

                                                                    borderRadius:
Colors.grey.shade300, BorderRadius.circular(12)),

                                                        ),
                                                ),
                                        ),
```

```
                                    const SizedBox(height: 4,
                                    ),
                                    Center( child: Text(
                                            'Submit Incident Report',style:

                                        ),
GoogleFonts.montserrat(),                   ),
                                    const SizedBox(height: 50,
                                    ),
                                ],


                            ),
                        ),

                        Padding( padding: const
EdgeInsets.symmetric(horizontal: 15),
                            child: Column( crossAxisAlignment:
CrossAxisAlignment.start,
                                children: [Text(
                                        'Pick incident type',
                                        style: GoogleFonts.montserrat(textStyle: const
                                            TextStyle(
                                                fontWeight:
FontWeight.bold,
                                                fontSize: 17,
                                            ),
                                        ),
```

```
                                                ),
                                                const SizedBox(height: 8,
                                                ),
                                                // DropdownButtonFormField(
                                                //           decoration:
InputDecoration(
                                                //                filled: true,
                                                //                fillColor:
Colors.grey.shade100,
                                                //                contentPadding: const
EdgeInsets.all(5), OutlineInputBorder(
                                                //                enabledBorder:
Colors.grey.shade200), BorderRadius.circular(10.0),
                                                //                     borderSide:
                                                //                          BorderSide(color:
OutlineInputBorder(
                                                //                     borderRadius:
                                                //                ),
Colors.grey.shade200), BorderRadius.circular(10.0),
                                                //                focusedBorder:
                                                //                     borderSide:
                                                //                          BorderSide(color:
crime', GoogleFonts.montserrat(
                                                //                     borderRadius:
                                                //                ),
                                                //                hintText: 'Pick type of
                                                //                hintStyle:
```

```
                                                    //                     textStyle:
                                                    //                         const
TextStyle(color: Colors.grey),
                                                    //                 ),
                                                    //             ),
                                                    //             items:
criminalTypes.map((String value) {
                                                    //                 return
DropdownMenuItem<String>(
                                                    //                     value: value,
                                                    //                     child: Text(value),
                                                    //                 );
                                                    //             }).toList(),
                                                    //             onChanged: (value) {},
                                                    //             isExpanded: true,
                                                    // ),

                                                    FutureBuilder( future: crimeTypes,
                                                        builder: (context, snapshot) {if
                                                            (snapshot.connectionState
==
                                                                    ConnectionState.waiting)return const
{
                                                                Center(
                                                                    child:
CircularProgressIndicator(),
                                                                );
                                                            } else if
(snapshot.hasError) {
                                                                return Center(child: Text(
```

```
                                                        'Error:
${snapshot.error.toString()}'),
                                                    );

                                                } else if (snapshot.hasData
&&
                                                        snapshot.data != null) {var crimeTypesList =

snapshot.data!;
                                                    return Container(height: 100,
                                                        width:
                                                            MediaQuery.of(contexchild:

t).size.width * 0.9, ListView.separated(
                                                            scrollDirection: itemCount:

Axis.horizontal, crimeTypesList.length,
                                                            separatorBuilder:

(context, index) {
                                                                return Container(width: 5,
                                                                );
                                                            },
                                                            itemBuilder: (context,var crime =

                                                                return Stack( clipBehavior:
index) { crimeTypesList[index];

Clip.none,
```

```
                                                                    children: [ GestureDetector(
                                                                        onTap: () { setState(()

                                                                                crimeTypeI
{
                                                                            });
d = crime.id;                                                               print(crimeT

ypeId);                                                                 },
                                                                        child:

Container(                                                                  height: 80,
                                                                            width: 80,
                                                                            decoration:

BoxDecoration( Colors.green.shade50,                                            color: borderRadi

us: Radius.circular(8),                                                                 Border

                                                                            ),
Column(                                                                     child:

ignment:                                                                            mainAxisAl

isAlignment.center,                                                                  MainAx

[                                                                               children:
```

```
                                                
```

```
tate(() { imeTypeId = "";
                                                                                cr

                                                                            });
Container(ration:                                                         },
                                                                          child:

                                                                            deco
BoxDecoration(
                                                                                co
lor: Colors

    .blue.shade50,

                                                                                sh
ape:

    BoxShape.circle,
                                                                                bo
xShadow: [

BoxShadow(

    color: Colors.grey

            .withOpacity(0.4),

    spreadRadius: 3,
```

```
    blurRadius: 5,offset:

            const Offset(

        0,

        3,

    ),

),
                                                                                ],
                                                                            ),
                                                                            chil
d: const Icon( ons.close_outlined,
                                                                                Icco
lor: Colors.grey,
                                                                            ),
                                                                        ),
                                                                    ),
                                                                )
Container()                                                             :
                                                                ],
                                                            );
                                                        },
                                                    ),
                                                );
```

```
                                                } else {
                                                    return Text('No data
available');

                                                }
                                            },
                                        ),
                                    ],
                                ),
                            ),
                        const SizedBox(height: 25,
                        ),
                        Padding( padding: const
EdgeInsets.symmetric(horizontal: 15),
                            child: Column( crossAxisAlignment:
CrossAxisAlignment.start,
                                children: [Text(
                                        'Describe incident',
                                        style: GoogleFonts.montserrat(textStyle: const
                                            TextStyle(
                                                fontWeight:
FontWeight.bold,
                                                fontSize: 17,
                                            ),
                                        ),
                                    ),
                                const SizedBox(height: 8,
                                ),
```

```
TextFormField(controller:

_incidentDescription,

maxLines: 5,
decoration: InputDecoration(filled: true,
    fillColor: contentPadding: const

Colors.grey.shade100,

    enabledBorder:

EdgeInsets.all(5), OutlineInputBorder(

        borderSide:
            BorderSide(color:

Colors.grey.shade200),

        borderRadius:

BorderRadius.circular(10.0),

    ),
    focusedBorder:

OutlineInputBorder(

        borderSide:
            BorderSide(color:

Colors.grey.shade200), BorderRadius.circular(10.0),

        borderRadius:

    ),
    hintText: 'Describe crimehintStyle:

here(optional)', GoogleFonts.montserrat(

        textStyle:
            const TextStyle(color:
```

```
      Colors.grey),
                                                    ),
                                                ),
                                            ),
                                        ],
                                    ),
                                )
                            ],
                        ),
                        const SizedBox(height: 25),crimeTypeId == "" ||
_incidentDescription.text.isEmpty
                            ? Padding(
                                padding: const
EdgeInsets.symmetric(vertical: 25),
                                child: Align(alignment:
Alignment.bottomCenter,
                                    child: Container(width:
MediaQuery.of(context).size.width * 0.9,
                                        height: 60,
                                        child: OutlinedButton(onPressed: () {},
                                            style: ButtonStyle(
                                                shape:
MaterialStateProperty.all(
                                                    RoundedRectangleBorder(borderRadius:
BorderRadius.circular(
                                                        10.0,
                                                    ),
```

```
                                                    ),
                                                ),
MaterialStateProperty.all(                          backgroundColor:
                                                        Colors.grey.shade200,
                                                ),
MaterialStateProperty.all(                          side:
                                                        BorderSide(color:
Colors.grey.shade200),
                                                    ),
                                                ),
                                                child: Text('HELP',
                                                    style:
GoogleFonts.montserrat(TextStyle(),
                                                        textStyle: const
                                                    ),
                                                ),
                                            ),
                                        ),
                                    ),
                                )
                            : Padding(
                                    padding: const
EdgeInsets.symmetric(vertical: 25),
                                    child: Align(alignment:
Alignment.bottomCenter,
                                        child: Container(width:
```

```
MediaQuery.of(context).size.width * 0.9,
                                                    height: 60,
                                                    child: OutlinedButton( onPressed: () async {
                                                            setState(() { isLoading = true;
                                                            });
                                                            FocusScope.of(context).unf
ocus(); apiServices.subnmitCrime(
                                                            var result = await
                                                                    userId.toString(), crimeTypeId.toString()
,
                                                                    _incidentDescription.t
ext,
                                                                    latitudeVal.toString()
,
                                                                    longitudeVal.toString(
),
                                                                    token.toString());
                                                            // ignore:
use_build_context_synchronously
                                                            Navigator.push( context,
                                                                MaterialPageRoute(
                                                                    builder: (context) =>
EndUserBottomNav(),
                                                                ),
                                                            );
                                                            const snackBar = SnackBar(
```

```
                content: Text('Updatedduration:
successfully'), Duration(seconds: 3),
            );


// ignore: ScaffoldMessenger.of(conte
use_build_context_synchronouslyxt)
                .showSnackBar(snackBar
);
            setState(() { isLoading = false;
            });
        },
        style: ButtonStyle(shape:
MaterialStateProperty.all(
            RoundedRectangleBorder(borderRadius:
BorderRadius.circular(
                    10.0,
                ),
            ),
        ),
        backgroundColor:
MaterialStateProperty.all(
            Colors.blue.shade400,
        ),
        side:
MaterialStateProperty.all(
```

```
                                                    BorderSide(color:
Colors.grey.shade200),
                                                ),
                                            ),
                                            child: Text( isLoading == false
                                                    ? 'HELP'
                                                    : 'Please wait...',style:

GoogleFonts.montserrat(TextStyle(                   textStyle: const

                                                        color: Colors.white,fontWeight:

FontWeight.bold),

                                                ),
                                            ),
                                        ),
                                    ),
                                ),
                            ),
                        )
                    ],
                ),
            ),
        );
    }

    Widget buildMap() => Stack(children: [
                GoogleMap(
                    mapType: MapType.normal, initialCameraPosition: CameraPosition(
```

```
                            target: _currentPosition!,zoom: 13,
                        ),
                        markers: {Marker(
                                markerId: const
MarkerId('_currentPosition'),
                                position: _currentPosition!,
                                icon: BitmapDescriptor.defaultMarker,
                            ),
                        },
                        onMapCreated: ((GoogleMapController controller)
=>
                                _mapController.complete(controller)),
                    ),
                    Positioned(top: 35,
                        left: 16,
                        child: status == 'Pending' && hideStatus == true
                                ? GestureDetector( onTap: () {}, child:
                                    Container(
                                        width:
MediaQuery.of(context).size.width * 0.9,
                                        // height: 80,
                                        padding: const EdgeInsets.all(8),decoration: BoxDecoration(
                                            color: Colors.white,borderRadius:
BorderRadius.circular(8),
                                            boxShadow: [
                                                BoxShadow(
```

```
                                        color:
Colors.black.withOpacity(0.2),
                                        blurRadius: 4,
                                        offset: const Offset(0, 2),
                                    ),
                                ],
                            ),
                            child: Column( crossAxisAlignment:
CrossAxisAlignment.center,
                                children: [Row(
                                        mainAxisAlignment:
MainAxisAlignment.end,
                                        children: [ IconButton(
                                                onPressed: () { hideStatus = false;
                                                },
                                                icon: const
Icon(Icons.close),
                                            ),
                                        ],
                                    ),
                                    const Text(
                                        'Ongoing request!',style: TextStyle(
                                                fontSize: 22, fontWeight:
FontWeight.w500),

                                    ),
                                    Text(
                                        'Submited
```

```
${dateFormat.format(userReportedDetails!.createdAt!)}',
                                                    style: const TextStyle(color:
Colors.grey),
                                                ),
                                                const SizedBox(height: 10,
                                                ),
                                                userReportedDetails!.responded_to_
by != null
                                                        ? Row(
                                                            mainAxisAlignment:
MainAxisAlignment.center,
                                                            children: [Text(
                                                                'Your request was
accepted',
                                                                style: TextStyle(color:
Colors.blue.shade600,
                                                                        fontWeight: fontSize: 16),
FontWeight.w500,
                                                            ),
                                                            const SizedBox(width: 8,
                                                            ),
                                                            Icon(
                                                                Icons.check_circle,color:
Colors.blue.shade600,
                                                                size: 17,
                                                            )
```

```
                                                    ],
                                                )
                                            :
userReportedDetails!.cancelled_by != null
                                                ? SizedBox(
                                                        width: 220,child:
                                                        Text(
                                                            'Cancelled but will
be attended to shortly.Please wait...',
                                                            style: TextStyle(color:
Colors.red.shade600,
                                                            ),
                                                        ),
                                                    )
                                                : Text('Please wait...'),const SizedBox(
                                        height: 10,
                                    ),
                                    Column( children: [
                                            Text(
                                                userReportedDetails!.respond
ed_to_by != null
                                                    ? 'This disappers once
you have been served!' attended to shortly!',
                                                    : 'Your request will bestyle:
GoogleFonts.montserrat(),
                                            ),
                                            const SizedBox(
```

```
                                                height: 4,
                                            ),
                                            const SpinKitSpinningLines(color:    Colors.blue,
                                                size: 50.0,
                                            ),
                                            const SizedBox(height: 10,
                                            ),
                                        ],
                                    )
                                ],
                            ),
                        ),
                    )
                : status == 'ongoing'
                    ? GestureDetector( onTap: () {}, child:
                        Container(
                            width:
MediaQuery.of(context).size.width * 0.9,
                            height: 80,
                            padding: const EdgeInsets.all(8),decoration: BoxDecoration(
                                color: Colors.white,borderRadius:
BorderRadius.circular(8),
                                boxShadow: [
                                    BoxShadow(
                                        color:
Colors.black.withOpacity(0.2),
                                        blurRadius: 4,
```

```
                                                    offset: const Offset(0, 2),
                                                ),
                                            ],
                                        ),
                                        child: Column( crossAxisAlignment:

CrossAxisAlignment.start,

                                            children: [Row(
                                                    mainAxisAlignment:
                                                            MainAxisAlignment.spaceB


etween,

                                                    children:[Row(
                                                            children: [Container(
                                                                    height: 25,
                                                                    width: 25,
                                                                    decoration:


                                                                        color:

BoxDecoration( Colors.blue.shade200,

                                                                        shape:

BoxShape.circle, Border.all( Colors.white,

                                                                        border:

width: 2),

                                                                                color:

                                                                    ),
                                                                    child: const Center(child: Text(
                                                                            'J',
```

```
                                                    style:
TextStyle(

Colors.white,                                                   color: fontSize: 12,
                                                    ),
                                                ),
                                            ),
                                        ),
                                        const SizedBox(width: 3,
                                        ),
                                        Row(
                                            children: [Text(
                                                'Joe Biden',style:

                                                        textStyle:
GoogleFonts.montserrat(const TextStyle(                                                                 fontWeig

ht: Weight.bold)),                                                                                      Font

                                                ),
                                                const SizedBox(width: 4,
                                                ),
                                                Text(
                                                    '2 min ago',style:
GoogleFonts.montserrat(
```

```
                                                                                textStyle:
const TextStyle(
                                                                                    color:
Colors.grey)),
                                                                        )
                                                                    ],
                                                                )
                                                            ],
                                                        ),
                                                        GestureDetector(onTap: () {
                                                                setState(() { hideStatus = false;
                                                                });
                                                            },
                                                            child: const Icon( Icons.close, color:
                                                                Colors.grey,
                                                            ),
                                                        )
                                                    ],
                                                ),
                                                const SizedBox(height: 10,
                                                ),
                                                Row(
                                                    children: [Text(
                                                            'You have currentstyle:


ongoing service', GoogleFonts.montserrat(),
```

```
                                                    ),
                                                    const SizedBox(width: 4,
                                                    ),
                                                    const SpinKitRipple( color:
                                                        Colors.blue,size: 20.0,
                                                    )
                                                ],
                                            )
                                        ],
                                    ),
                                ),
                            )
                            : Container(),
                    ),
                ],
            );
}
```

l user home page.

```
import 'dart:async'; import 'dart:convert';
import 'package:crimereporter/apiservices/apiservices.dart';import
'package:crimereporter/model/crime_types.dart'; import 'package:crimereporter/model/officer.dart';
import 'package:crimereporter/pages/enduser/enduserbottomnav.dart';
```

```
import 'package:geocoding/geocoding.dart' as geocoding;import
'package:google_fonts/google_fonts.dart';
import 'package:google_places_flutter/google_places_flutter.dart';import
'package:google_places_flutter/model/prediction.dart'; import 'package:flutter/material.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';import 'package:intl/intl.dart';
import 'package:location/location.dart';
import 'package:shared_preferences/shared_preferences.dart';import
'package:sliding_up_panel/sliding_up_panel.dart'; import 'package:uuid/uuid.dart';
import 'package:flutter_spinkit/flutter_spinkit.dart';import 'package:http/http.dart' as http;

class EndUserHome extends StatefulWidget {@override
    _EndUserHomeState createState() => _EndUserHomeState();
}

class _EndUserHomeState extends State<EndUserHome> {bool isPanelOpen = false;
    var uuid = const Uuid(); var location =
    Location();
    final Completer<GoogleMapController> _mapController =Completer<GoogleMapController>();


    LatLng? _currentPosition;@override
```

```
    void initState() { super.initState();
        getLocation(); fetchUserData();
        setState(() {});
    }

    Future<void> getLocation() async { bool serviceEnabled;
        PermissionStatus permissionGranted;

        serviceEnabled = await location.serviceEnabled();if (!serviceEnabled) {
            serviceEnabled = await location.requestService();if (!serviceEnabled) {
                return;
            }
        }


        permissionGranted = await location.hasPermission();if (permissionGranted ==
        PermissionStatus.denied) {
            permissionGranted = await
location.requestPermission();
            if (permissionGranted != PermissionStatus.granted) {return;
            }
        }


        // return location.getLocation();
        location.onLocationChanged.listen(
            (LocationData currentLocation) {
                if (currentLocation.altitude != null &&
```

```
                        currentLocation.longitude != null) {setState(() {
                        _currentPosition = LatLng(currentLocation.latitude!,
currentLocation.longitude!);
                        cameraPorision(_currentPosition!);
                        // print(_currentPosition);
                    });
                }
            },
        );
    }


    Future<void> cameraPorision(LatLng pos) async {final GoogleMapController
        controller = await
_mapController.future;
        CameraPosition newCameraPosition =CameraPosition(target:
pos, zoom: 13);
        await controller.animateCamera( CameraUpdate.newCameraPosition(newCameraPosition),
        );
    }


    ScrollController? controller;
    PanelController panelController = PanelController();

    final TextEditingController searcLocation =TextEditingController();
    final TextEditingController _incidentDescription =TextEditingController();
    DateFormat dateFormat = DateFormat();String? latitudeVal;
```

```
    String? longitudeVal;
    ApiServices apiServices = ApiServices();

    String? status;
    bool hideStatus = true;

    late Future<List<CrimeTypes>> crimeTypes; Future<List<Officer>> userReported =
    Future.value([]);Officer? userReportedDetails;

    String? userId;String?
    token;
    String crimeTypeId = ""; void fetchUserData()
    async {
        SharedPreferences prefs = await
SharedPreferences.getInstance();
        String jsonDataString = prefs.getString('userData') ??
'';
        Map<String, dynamic> jsonData =
json.decode(jsonDataString);setState(() {
            userId = jsonData['data']['id'];token = jsonData['token'];
        });
        crimeTypes = apiServices.fetchCrimesType(token.toString());
        Timer.periodic(const Duration(seconds: 3), (timer) async
{
                                apiServices.getUserReported(token.toString(),
userId.toString());
            setState(() { userReported =
```

```
                        apiServices.getUserReported(token.toString(),userId.toString());
            });
            List<Officer> result = (await userReported)
                    .where((item) => item.status == 'Pending')
                    .toList();
            if (result.isNotEmpty && result != []) {setState(() {
                    status = result.last.status; userReportedDetails = result.last;
                });
                print('Status!!!!!!!!!!!! $status');
            } else { setState((){
                    status = 'Completed';
                });
                SharedPreferences prefs = await
SharedPreferences.getInstance();
                prefs.remove('submittedCrime');
             }
            });
    }

    bool isLoading = false;

    @override
    Widget build(BuildContext context) {return Scaffold(
            appBar: isPanelOpen
                    ? AppBar(
                            automaticallyImplyLeading: false,title: Row(
```

```
                        children: [
                            Expanded(
                                child: GooglePlaceAutoCompleteTextField(textEditingController:
                                    searcLocation, googleAPIKey:
"AIzaSyA7zmasenUJGF1QDQrmPdrOzLxBL4864s0",
                                    inputDecoration: const
InputDecoration(
                                        enabledBorder: UnderlineInputBorder(borderSide:
                                            BorderSide.none,
                                        ),
                                        focusedBorder: UnderlineInputBorder(borderSide:
                                            BorderSide.none,
                                        ),
                                        hintText: 'Search your location',hintStyle: TextStyle(
                                            color: Colors.grey,
                                        ),
                                        prefixIcon: Icon( Icons.search, color:
                                            Colors.grey,
                                        ),
                                    ),
                                    debounceTime: 800, countries: const ["ke"],
                                    isLatLngRequired:
                                            true, // if you requiredcoordinates from place detail
                                    getPlaceDetailWithLatLng: (Prediction
prediction) {
                                        setState(() { latitudeVal =
```

```
prediction.lat.toString();
                    longitudeVal =
prediction.lng.toString();

                });
            },
            itemClick: (Prediction prediction) {searcLocation.text =
prediction.description.toString();
                searcLocation.selection =
TextSelection.fromPosition(
                    TextPosition(offset:
prediction.description!.length),
                );
            },
            itemBuilder: (context, index,Prediction prediction) {
                return Container(
                    padding: const EdgeInsets.all(10),child: Row(
                        children: [
                            const Icon(Icons.location_on),const SizedBox(
                                width: 7,
                            ),
                            Expanded(child:
Text(prediction.description ?? ""),
                            )
                        ],
                    ),
                );
            },
```

```
                                        seperatedBuilder: const Divider(),isCrossBtnShown: true,
                                    ),
                                ),
                            ],
                        ),
                        backgroundColor: Colors.white,elevation: 0,
                        toolbarHeight: 80.0,
                    )
                : null,
            body: _currentPosition == null
                ? const Center(
                    child:          SpinKitRipple(        color:
                        Colors.blue,size: 50.0,
                    ),
                )
                : SlidingUpPanel(
                    // minHeight: MediaQuery.of(context).size.height *
0.07,
                    minHeight: MediaQuery.of(context).size.height
* 0.1,

* 0.6,                    maxHeight: MediaQuery.of(context).size.heightborderRadius:
                            const BorderRadius.vertical(top:


Radius.circular(18)),
                    controller: panelController,parallaxEnabled: true,
                    parallaxOffset: 0.5, onPanelOpened: () {
```

```
                    setState(() { isPanelOpen = true;
                    });
                },
                onPanelClosed: () {setState(() {
                        isPanelOpen = false;
                    });
                },
                body: buildMap(),
                panelBuilder: (controller) => Stack(children: [
                        ListView(
                            controller: controller, padding:
                            EdgeInsets.zero,children: [
                                const SizedBox(height: 20,
                                ),
                                Visibility(
                                    visible: isPanelOpen == false,child: Column(
                                        children: [Center(
                                                child: GestureDetector(onTap: () {
                                                        panelController.isPanelOpe
n
                                                                ?
        panelController.close()
                                                                :
        panelController.open();
```

```
                                        },
                                        child: Container(height: 5,
                                            width: 30,
                                            decoration: BoxDecoration(color:

                                                    borderRadius:
Colors.grey.shade300, BorderRadius.circular(12)),

                                        ),
                                    ),
                                ),
                                const SizedBox(height: 4,
                                ),
                                Center( child: Text(
                                        'Submit Incident Report',style:

GoogleFonts.montserrat(),                                    ),
                                ),
                                const SizedBox(height: 50,
                                ),
                            ],

                        ),
                    ),

                    Padding( padding: const
EdgeInsets.symmetric(horizontal: 15),
                        child: Column(
```

```
                                        crossAxisAlignment:
CrossAxisAlignment.start,
                                        children: [Text(
                                                'Pick incident type',
                                                style: GoogleFonts.montserrat(textStyle: const
                                                    TextStyle(
                                                        fontWeight:
FontWeight.bold,
                                                        fontSize: 17,
                                                    ),
                                                ),
                                            ),
                                        const SizedBox(height: 8,
                                        ),
                                        // DropdownButtonFormField(
                                        //         decoration:
InputDecoration(
                                        //             filled: true,
                                        //             fillColor:
Colors.grey.shade100,
                                        //             contentPadding: const
EdgeInsets.all(5), OutlineInputBorder(
                                        //             enabledBorder:
                                        //                 borderSide:
Colors.grey.shade200), BorderRadius.circular(10.0),
                                        //                     BorderSide(color:
                                        //                 borderRadius:
                                        //             ),
```

```
                                                    //                 focusedBorder:
OutlineInputBorder(
                                                    //                     borderSide:
                                                    //                         BorderSide(color:
Colors.grey.shade200), BorderRadius.circular(10.0),
                                                    //                     borderRadius:

crime', GoogleFonts.montserrat(                     //                 ),
                                                    //                 hintText: 'Pick type of

                                                    //                 hintStyle:

                                                    //                     textStyle:
                                                    //                         const
TextStyle(color: Colors.grey),
                                                    //                 ),
                                                    //             ),
                                                    //             items:
criminalTypes.map((String value) {
                                                    //                 return
DropdownMenuItem<String>(
                                                    //                     value: value,
                                                    //                     child: Text(value),
                                                    //                 );
                                                    //             }).toList(),
                                                    //             onChanged: (value) {},
                                                    //             isExpanded: true,
                                                    // ),

                                                    FutureBuilder( future: crimeTypes,
                                                        builder: (context, snapshot) {
```

```
==
{
CircularProgressIndicator(),
(snapshot.hasError) {
${snapshot.error.toString()}'),
&&
snapshot.data!;
t).size.width * 0.9, ListView.separated(
Axis.horizontal, crimeTypesList.length,
if (snapshot.connectionState
        ConnectionState.waiting)
    return const Center(child:
    );
} else if
    return Center(child: Text(
            'Error:
    );
} else if (snapshot.hasData
        snapshot.data != null) {var crimeTypesList =
    return Container(height: 100,
        width:
            MediaQuery.of(contexchild:
            scrollDirection:
            itemCount: separatorBuilder:
```

```
(context, index) {
                                                    return Container(width: 5,
                                                    );
                                                },
                                                itemBuilder: (context,var crime =
index) { crimeTypesList[index];
                                                    return Stack( clipBehavior:
                                                        children: [ GestureDetector(
Clip.none,
                                                                onTap: () { setState(()
                                                                    crimeTypeI
{                                                               });
                                                                print(crimeT
d = crime.id;
ypeId);
                                                            },
                                                            child:
Container(
                                                                height: 80,
                                                                width: 80,
                                                                decoration:
BoxDecoration( Colors.green.shade50,
                                                                    color: borderRadi
```

```
us: Radius.circular(8),
                                                                    Border

                                                                ),
Column(                                                         child:

                                                                    mainAxisAl
ignment: isAlignment.center,[
                                                                    MainAx
r(
                                                                children:
const Icon( s.security_outlined,
                                                                    Containechild:

                                                                        Icon
crime_name,ign:                                                     ),
                                                                ),
xtAlign.center,                                                 Text(
                                                                    crime.

                                                                    textAl

                                                                        Te
                                                                ),
                                                            ],
                                                        ),
                                                    ),
                                                ),
```

```
                                                                crimeTypeId ==
crime.id Positioned(
                                                                    ?
4,
                                                                        right: -
                                                                top: -2,child:
GestureDetector(() {
                                                                        onTap:
tate(() { imeTypeId = "";
                                                                            setS
                                                                                cr
Container(ration:
                                                                        });
                                                                    },
                                                                    child:
                                                                        deco
BoxDecoration(
                                                                            co
lor: Colors
    .blue.shade50,
                                                                            sh
ape:
    BoxShape.circle,
                                                                            bo
```

```
    xShadow: [
   BoxShadow(
    color: Colors.grey
                .withOpacity(0.4),
    spreadRadius: 3,
    blurRadius: 5,offset:
                const Offset(
        0,
        3,
    ),
),
                                                                                ],
                                                                            ),
                                                                            chil
d: const Icon( ons.close_outlined,
                                                                                Ic
                                                                                co
```

```
lor: Colors.grey,
                                                                                                    ),
                                                                                                ),
                                                                                            ),
                                                                                        )
Container()                                                                                 :
                                                                                ],
                                                                            );
                                                                        },
                                                                    ),
                                                                );
                                                            } else {
available');                                                             return Text('No data

                                                        }
                                                    },
                                                ),
                                            ],
                                        ),
                                    ),
                                    const SizedBox(height: 25,
                                    ),
                                    Padding( padding: const
EdgeInsets.symmetric(horizontal: 15),
                                        child: Column( crossAxisAlignment:
CrossAxisAlignment.start,
                                            children: [Text(
```

```
                                    'Describe incident',
                                    style: GoogleFonts.montserrat(textStyle: const
                                        TextStyle(
                                            fontWeight:
FontWeight.bold,
                                            fontSize: 17,
                                        ),
                                    ),
                                ),
                                const SizedBox(height: 8,
                                ),
                                TextFormField(controller:
_incidentDescription,
                                    maxLines: 5,
                                    decoration: InputDecoration(filled: true,
                                        fillColor: contentPadding: const
Colors.grey.shade100,
                                        enabledBorder:
EdgeInsets.all(5), OutlineInputBorder(
                                            borderSide:
                                                BorderSide(color:
Colors.grey.shade200),
                                            borderRadius:
BorderRadius.circular(10.0),
                                        ),
                                        focusedBorder:
OutlineInputBorder(
```

```
                                                    borderSide:
                                                        BorderSide(color:
Colors.grey.shade200), BorderRadius.circular(10.0),
                                                    borderRadius:

                                                ),
                                                hintText: 'Describe crimehintStyle:
here(optional)', GoogleFonts.montserrat(
                                                    textStyle:
                                                        const TextStyle(color:
Colors.grey),


                                                ),
                                            ),
                                        ),
                                    ],
                                ),
                            )
                        ],
                    ),
                    const SizedBox(height: 25),crimeTypeId == "" ||
_incidentDescription.text.isEmpty
                        ? Padding(
                            padding: const
EdgeInsets.symmetric(vertical: 25),
                            child: Align(alignment:
Alignment.bottomCenter,
                                child: Container(width:
```

```
MediaQuery.of(context).size.width * 0.9,
                                                    height: 60,
                                                    child: OutlinedButton(onPressed: () {},
                                                        style: ButtonStyle(
                                                            shape:
MaterialStateProperty.all(
                                                                RoundedRectangleBorder(borderRadius:
BorderRadius.circular(
                                                                    10.0,
                                                                ),
                                                            ),
                                                        ),
                                                            backgroundColor:
MaterialStateProperty.all(
                                                                Colors.grey.shade200,
                                                            ),
                                                            side:
MaterialStateProperty.all(
                                                                BorderSide(color:
Colors.grey.shade200),
                                                            ),
                                                        ),
                                                        child: Text('HELP',
                                                            style:
GoogleFonts.montserrat(TextStyle(),
                                                                textStyle: const
                                                            ),
                                                      ),
```

```
                                                    ),
                                                ),
                                            ),
                                        )
                                    : Padding(
                                            padding: const
EdgeInsets.symmetric(vertical: 25),
                                            child: Align(alignment:
Alignment.bottomCenter,
                                                child: Container(width:
MediaQuery.of(context).size.width * 0.9,
                                                    height: 60,
                                                    child: OutlinedButton( onPressed: () async {
                                                            setState(() { isLoading = true;
                                                            });
                                                            FocusScope.of(context).unf
ocus(); apiServices.subnmitCrime(
                                                            var result = await
                                                                    userId.toString(), crimeTypeId.toString()
,
                                                                    _incidentDescription.t
ext,
                                                                    latitudeVal.toString()
,
                                                                    longitudeVal.toString(
),
```

```
                                                        token.toString());
                                                    // ignore:
use_build_context_synchronously

                                                    Navigator.push( context,
                                                        MaterialPageRoute(
                                                            builder: (context) =>

EndUserBottomNav(),
                                                        ),
                                                    );


                                                    const snackBar = SnackBar(content:
                                                        Text('Updated
successfully'), Duration(seconds: 3),
                                                        duration:

                                                    );


                                                    // ignore: ScaffoldMessenger.of(conte
use_build_context_synchronouslyxt)

                                                            .showSnackBar(snackBar
);
                                                    setState(() { isLoading = false;
                                                    });
                                                },
                                                style: ButtonStyle(shape:

MaterialStateProperty.all(
                                                        RoundedRectangleBorder(
```

```
                                    borderRadius:
BorderRadius.circular(
                                        10.0,
                                    ),
                                ),
                            ),
                            backgroundColor:
MaterialStateProperty.all(
                                Colors.blue.shade400,
                            ),
                            side:
MaterialStateProperty.all(
                                BorderSide(color:
Colors.grey.shade200),
                            ),
                        ),
                    ),
                    child: Text( isLoading == false
                            ? 'HELP'
                            : 'Please wait...',style:
                        textStyle: const
GoogleFonts.montserrat(TextStyle(
                            color: Colors.white,fontWeight:
FontWeight.bold),
                        ),
                    ),
                ),
            ),
        ),
    ),
```

```
                                                    )
                                        ],
                                    ),
                                ),
        );
    }

    Widget buildMap() => Stack(children: [
                        GoogleMap(
                            mapType: MapType.normal, initialCameraPosition: CameraPosition(
                                target: _currentPosition!,zoom: 13,
                            ),
                            markers: {Marker(
                                    markerId: const
MarkerId('_currentPosition'),
                                    position: _currentPosition!,
                                    icon: BitmapDescriptor.defaultMarker,
                                ),
                            },
                            onMapCreated: ((GoogleMapController controller)
=>
                                    _mapController.complete(controller)),
                        ),
                        Positioned(top: 35,
                            left: 16,
                            child: status == 'Pending' && hideStatus == true
                                    ? GestureDetector(
```

```
                                    onTap: () {}, child: Container(
                                        width:
MediaQuery.of(context).size.width * 0.9,
                                        // height: 80,
                                        padding: const EdgeInsets.all(8),decoration: BoxDecoration(
                                            color: Colors.white,borderRadius:
BorderRadius.circular(8),
                                            boxShadow: [
                                                BoxShadow(
                                                    color:
Colors.black.withOpacity(0.2),
                                                    blurRadius: 4,
                                                    offset: const Offset(0, 2),
                                                ),
                                            ],
                                        ),
                                        child: Column( crossAxisAlignment:
CrossAxisAlignment.center,
                                            children: [Row(
                                                    mainAxisAlignment:
MainAxisAlignment.end,
                                                    children: [ IconButton(
                                                            onPressed: () { hideStatus = false;
                                                            },
                                                            icon: const
```

```
Icon(Icons.close),
                                                    ),
                                                ],
                                            ),
                                            const Text(
                                                'Ongoing request!',style: TextStyle(
                                                        fontSize: 22, fontWeight:
FontWeight.w500),

                                            ),
                                            Text(
                                                'Submited
${dateFormat.format(userReportedDetails!.createdAt!)}',
                                                style: const TextStyle(color:
Colors.grey),
                                            ),
                                            const SizedBox(height: 10,
                                            ),
                                            userReportedDetails!.responded_to_

by != null

                                                    ? Row(
                                                            mainAxisAlignment:

MainAxisAlignment.center,

                                                            children: [Text(
                                                                    'Your request was


accepted',

                                                                    style: TextStyle(color:

                                                                            fontWeight:

Colors.blue.shade600,
```

```
FontWeight.w500,
                                                                                fontSize: 16),
                                                                        ),
                                                                        const SizedBox(width: 8,
                                                                        ),
                                                                        Icon(
                                                                            Icons.check_circle,color:

Colors.blue.shade600,                                                       size: 17,

                                                                        )
                                                                    ],
                                                                )
                                                            :
userReportedDetails!.cancelled_by != null
                                                                    ? SizedBox(
                                                                            width: 220,child:
                                                                            Text(
                                                                                'Cancelled but will
be attended to shortly.Please wait...',
                                                                                style: TextStyle(color:

Colors.red.shade600,                                                            ),

                                                                             ),
                                                                        )
                                                                    : Text('Please wait...'),const SizedBox(
                                                        height: 10,
                                                    ),
                                                    Column(
```

```
                                                children: [Text(
                                                        userReportedDetails!.respond

ed_to_by != null                                                    ? 'This disappers once

you have been served!' attended to shortly!',                       : 'Your request will bestyle:

GoogleFonts.montserrat(),
                                                    ),
                                                    const SizedBox(height: 4,
                                                    ),
                                                    const SpinKitSpinningLines(color:    Colors.blue,
                                                        size: 50.0,
                                                    ),
                                                    const SizedBox(height: 10,
                                                    ),
                                                ],


                                                )
                                            ],
                                        ),
                                    ),

                                )
                            : status == 'ongoing'
                                    ? GestureDetector( onTap: () {}, child:
                                            Container(
                                                width:
```

```
MediaQuery.of(context).size.width * 0.9,
                                                    height: 80,
                                                    padding: const EdgeInsets.all(8),decoration: BoxDecoration(
                                                        color: Colors.white,borderRadius:
BorderRadius.circular(8),
                                                        boxShadow: [
                                                            BoxShadow(
                                                                color:
Colors.black.withOpacity(0.2),
                                                                blurRadius: 4,
                                                                offset: const Offset(0, 2),
                                                            ),
                                                        ],
                                                    ),
                                                    child: Column( crossAxisAlignment:
CrossAxisAlignment.start,
                                                        children: [Row(
                                                                mainAxisAlignment:
                                                                        MainAxisAlignment.spaceB
etween,
                                                                children:[Row(
                                                                        children: [Container(
                                                                                height: 25,
                                                                                width: 25,
                                                                                decoration:
BoxDecoration(
```

```
                                                    color:
Colors.blue.shade200, BoxShape.circle,
                                                    shape:
Border.all( Colors.white, width: 2),
                                                    border:
                                                        color:
                                                ),
                                                child: const Center(child: Text(
TextStyle(                                              'J',
Colors.white,                                           style:
                                                            color: fontSize: 12,
                                                        ),
                                                    ),
                                                ),
                                            ),
                                            const SizedBox(width: 3,
                                            ),
                                            Row(
                                                children: [Text(
                                                        'Joe Biden',style:
GoogleFonts.montserrat(const TextStyle(                             textStyle:
```

```
                                                                                fontWeig
ht: Weight.bold)),
                                                                                    Font

                                                    ),
                                                    const SizedBox(width: 4,
                                                    ),
                                                    Text(
                                                        '2 min ago',style:
GoogleFonts.montserrat(
                                                                            textStyle:

                                                                                color:
const TextStyle(


Colors.grey)),

                                                                ]
                                                            )
                                                        ],
                                                    ),
                                                GestureDetector(onTap: () {
                                                        setState(() { hideStatus = false;
                                                        });
                                                    },
                                                    child: const Icon( Icons.close, color:
                                                        Colors.grey,
                                                    ),
                                                )
```

```
                                                ],
                                            ),
                                            const SizedBox(height: 10,
                                            ),
                                            Row(
                                                children: [Text(
                                                        'You have currentstyle:
ongoing service', GoogleFonts.montserrat(),
                                                    ),
                                                    const SizedBox(width: 4,
                                                    ),
                                                    const SpinKitRipple( color:
                                                        Colors.blue,size: 20.0,
                                                    )
                                                ],
                                            )
                                        ],
                                    ),
                                ),
                            )
                            : Container(),
                    ),
                ],
            );
}
```

r user home page.

```
import 'dart:async'; import 'dart:convert';
import 'package:crimereporter/apiservices/apiservices.dart';import
'package:crimereporter/model/crime_types.dart'; import 'package:crimereporter/model/officer.dart';
import 'package:crimereporter/pages/enduser/enduserbottomnav.dart';import
'package:geocoding/geocoding.dart' as geocoding; import 'package:google_fonts/google_fonts.dart';
import 'package:google_places_flutter/google_places_flutter.dart';import
'package:google_places_flutter/model/prediction.dart'; import 'package:flutter/material.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';import 'package:intl/intl.dart';
import 'package:location/location.dart';
import 'package:shared_preferences/shared_preferences.dart';import
'package:sliding_up_panel/sliding_up_panel.dart'; import 'package:uuid/uuid.dart';
import 'package:flutter_spinkit/flutter_spinkit.dart';import 'package:http/http.dart' as http;

class EndUserHome extends StatefulWidget {@override
    _EndUserHomeState createState() => _EndUserHomeState();
}
```

```
class _EndUserHomeState extends State<EndUserHome> {bool isPanelOpen = false;
    var uuid = const Uuid(); var location =
    Location();
    final Completer<GoogleMapController> _mapController =Completer<GoogleMapController>();


    LatLng? _currentPosition;@override

    void initState() {
        super.initState();getLocation();
        fetchUserData(); setState(() {});
    }

    Future<void> getLocation() async { bool serviceEnabled;
        PermissionStatus permissionGranted;

        serviceEnabled = await location.serviceEnabled();if (!serviceEnabled) {
            serviceEnabled = await location.requestService();if (!serviceEnabled) {
                return;
            }
        }


        permissionGranted = await location.hasPermission();if (permissionGranted ==
        PermissionStatus.denied) {
```

```
            permissionGranted = await
location.requestPermission();
            if (permissionGranted != PermissionStatus.granted) {return;
            }
        }


        // return location.getLocation();
        location.onLocationChanged.listen(
            (LocationData currentLocation) {
                if (currentLocation.altitude != null &&currentLocation.longitude != null) {
                    setState(() {
                        _currentPosition = LatLng(currentLocation.latitude!,
currentLocation.longitude!);
                        cameraPorision(_currentPosition!);
                        // print(_currentPosition);
                    });
                }
            },
        );
    }


    Future<void> cameraPorision(LatLng pos) async {final GoogleMapController
        controller = await
_mapController.future;
        CameraPosition newCameraPosition =CameraPosition(target:
pos, zoom: 13);
        await controller.animateCamera( CameraUpdate.newCameraPosition(newCameraPosition),
        );
```

```
    }

    ScrollController? controller;
    PanelController panelController = PanelController();

    final TextEditingController searcLocation =TextEditingController();
    final TextEditingController _incidentDescription =TextEditingController();
    DateFormat dateFormat = DateFormat();String? latitudeVal;
    String? longitudeVal;
    ApiServices apiServices = ApiServices();

    String? status;
    bool hideStatus = true;

    late Future<List<CrimeTypes>> crimeTypes; Future<List<Officer>> userReported =
    Future.value([]);Officer? userReportedDetails;

    String? userId;String?
    token;
    String crimeTypeId = ""; void fetchUserData()
    async {
        SharedPreferences prefs = await
SharedPreferences.getInstance();
        String jsonDataString = prefs.getString('userData') ??
'';
        Map<String, dynamic> jsonData =
json.decode(jsonDataString);setState(() {
```

```
            userId = jsonData['data']['id'];token = jsonData['token'];
        });
        crimeTypes = apiServices.fetchCrimesType(token.toString());
        Timer.periodic(const Duration(seconds: 3), (timer) async
{
                                apiServices.getUserReported(token.toString(),
userId.toString());
            setState(() { userReported =
                    apiServices.getUserReported(token.toString(),userId.toString());
            });
            List<Officer> result = (await userReported)
                    .where((item) => item.status == 'Pending')
                    .toList();
            if (result.isNotEmpty && result != []) {setState(() {
                    status = result.last.status; userReportedDetails = result.last;
                });
                print('Status!!!!!!!!!!!! $status');
            } else { setState(() {
                    status = 'Completed';
                });
                SharedPreferences prefs = await
SharedPreferences.getInstance();
                prefs.remove('submittedCrime');
              }
            });
```

```
    }


    bool isLoading = false;@override
    Widget build(BuildContext context) {return Scaffold(
            appBar: isPanelOpen
                    ? AppBar(
                            automaticallyImplyLeading: false,title: Row(
                        children: [
                            Expanded(
                                child: GooglePlaceAutoCompleteTextField(textEditingController:
                                    searcLocation, googleAPIKey:
"AIzaSyA7zmasenUJGF1QDQrmPdrOzLxBL4864s0",
                                    inputDecoration: const
InputDecoration(
                                        enabledBorder: UnderlineInputBorder(borderSide:
                                            BorderSide.none,
                                        ),
                                        focusedBorder: UnderlineInputBorder(borderSide:
                                            BorderSide.none,
                                        ),
                                        hintText: 'Search your location',hintStyle: TextStyle(
                                            color: Colors.grey,
                                        ),
                                        prefixIcon: Icon( Icons.search, color:
                                            Colors.grey,
```

```
                                        ),
                                    ),
                                    debounceTime: 800, countries: const ["ke"],
                                    isLatLngRequired:
                                        true, // if you requiredcoordinates from place detail
                                    getPlaceDetailWithLatLng: (Prediction
prediction) {
                                        setState(() { latitudeVal =
prediction.lat.toString();
                                            longitudeVal =
prediction.lng.toString();


                                        });
                                    },
                                    itemClick: (Prediction prediction) {searcLocation.text =
prediction.description.toString();
                                        searcLocation.selection =
TextSelection.fromPosition(
                                            TextPosition(offset:
prediction.description!.length),
                                        );
                                    },
                                    itemBuilder: (context, index,Prediction prediction) {
                                        return Container(
                                            padding: const EdgeInsets.all(10),child: Row(
                                                children: [
                                                    const Icon(Icons.location_on),
```

```
                                                const SizedBox(width: 7,
                                                ),
                                                Expanded(child:
Text(prediction.description ?? ""),
                                                )
                                            ],
                                        ),
                                    );
                                },
                                seperatedBuilder: const Divider(),isCrossBtnShown: true,
                            ),
                        ),
                    ],
                ),
                backgroundColor: Colors.white,elevation: 0,
                toolbarHeight: 80.0,
            )
        : null,
    body: _currentPosition == null
        ? const Center(
            child:      SpinKitRipple(      color:
                Colors.blue,size: 50.0,
            ),
        )
        : SlidingUpPanel(
            // minHeight: MediaQuery.of(context).size.height *
0.07,
```

```
                        minHeight: MediaQuery.of(context).size.height
* 0.1,

* 0.6,                  maxHeight: MediaQuery.of(context).size.heightborderRadius:
                              const BorderRadius.vertical(top:
Radius.circular(18)),
                        controller: panelController,parallaxEnabled: true,
                        parallaxOffset: 0.5, onPanelOpened: () {
                            setState(() { isPanelOpen = true;
                            });
                        },
                        onPanelClosed: () {setState(() {
                              isPanelOpen = false;
                            });
                        },
                        body: buildMap(),
                        panelBuilder: (controller) => Stack(children: [
                              ListView(
                                  controller: controller, padding:
                                  EdgeInsets.zero,children: [
                                      const SizedBox(height: 20,
                                      ),
                                      Visibility(
                                          visible: isPanelOpen == false,
```

```
                                        child: Column(children: [
                                            Center(
                                              child: GestureDetector(onTap: () {
                                                    panelController.isPanelOpe

                                                                ?
n
                                                                :
panelController.close()


panelController.open();

                                                  },
                                                  child: Container(height: 5,
                                                      width: 30,
                                                      decoration: BoxDecoration(color:

                                                                borderRadius:
Colors.grey.shade300, BorderRadius.circular(12)),

                                                  ),
                                              ),
                                          ),
                                          const SizedBox(height: 4,
                                          ),
                                          Center( child: Text(
                                                  'Submit Incident Report',style:

                                              ),
GoogleFonts.montserrat(),
```

```
                                    ),
                                    const SizedBox(height: 50,
                                      ),
                                ],
                            ),
                        ),
                    Padding( padding: const
EdgeInsets.symmetric(horizontal: 15),
                        child: Column( crossAxisAlignment:
CrossAxisAlignment.start,
                            children: [Text(
                                'Pick incident type',
                                style: GoogleFonts.montserrat(textStyle: const
                                    TextStyle(
                                        fontWeight:
FontWeight.bold,
                                        fontSize: 17,
                                    ),
                                ),
                            ),
                            const SizedBox(height: 8,
                            ),
                            // DropdownButtonFormField(
                            //         decoration:
InputDecoration(
                            //             filled: true,
                            //             fillColor:
```

```
Colors.grey.shade100,
                                                    //                 contentPadding: const
EdgeInsets.all(5), OutlineInputBorder(
                                                    //                 enabledBorder:
                                                    //                     borderSide:
Colors.grey.shade200), BorderRadius.circular(10.0),//                         BorderSide(color:
                                                    //                     borderRadius:
OutlineInputBorder(
                                                    //                 ),
                                                    //                 focusedBorder:
Colors.grey.shade200), BorderRadius.circular(10.0),
                                                    //                     borderSide:
                                                    //                         BorderSide(color:
crime', GoogleFonts.montserrat(
                                                    //                     borderRadius:
                                                    //                 ),
                                                    //                 hintText: 'Pick type of
                                                    //                 hintStyle:
                                                    //                     textStyle:
                                                    //                         const
TextStyle(color: Colors.grey),
                                                    //                 ),
                                                    //             ),
                                                    //             items:
criminalTypes.map((String value) {
                                                    //                 return
DropdownMenuItem<String>(
```

```
//                     value: value,
//                     child: Text(value),
//                 );
//             }).toList(),
//             onChanged: (value) {},
//             isExpanded: true,
// ),


FutureBuilder( future: crimeTypes,
    builder: (context, snapshot) {if
        (snapshot.connectionState
==
                ConnectionState.waiting)
{
            return const Center(child:
CircularProgressIndicator(),
            );
        } else if
(snapshot.hasError) {
            return Center(child: Text(
                    'Error:
${snapshot.error.toString()}'),
            );
        } else if (snapshot.hasData
&&
                snapshot.data != null) {var crimeTypesList =
snapshot.data!;
            return Container(
```

```
t).size.width * 0.9, ListView.separated(

Axis.horizontal, crimeTypesList.length,

(context, index) {



index) { crimeTypesList[index];


Clip.none,



{

d = crime.id;
```

```
height: 100,width:
        MediaQuery.of(contexchild:

    scrollDirection: itemCount:

    separatorBuilder:

        return Container(width: 5,
        );
    },
    itemBuilder: (context,var crime =

        return Stack( clipBehavior:

            children: [   GestureDetector(
                    onTap: () { setState(()

                        crimeTypeI

                    });
                    print(crimeT
```

```
ypeId);

Container(

BoxDecoration( Colors.green.shade50,
us: Radius.circular(8),

Column(
ignment:
isAlignment.center,[
r(
const Icon( s.security_outlined,
},
child:
height: 80,
width: 80,
decoration:
color: borderRadi
Border
),
child:
mainAxisAl
MainAx
children:
Containechild:
Icon
),
),
Text(
```

```
                                                
```

```
BoxDecoration(
                                                                    co
lor: Colors
    .blue.shade50,
                                                                    sh
ape:
    BoxShape.circle,
                                                                    bo
xShadow: [
BoxShadow(
    color: Colors.grey
                .withOpacity(0.4),
    spreadRadius: 3,
    blurRadius: 5,offset:
                    const Offset(
        0,
        3,
```

```
      ),

),
                                                                                ],
                                                                            ),
                                                                            chil
d: const Icon( ons.close_outlined,
                                                                                Icco
lor: Colors.grey,
                                                                            ),
                                                                        ),
                                                                    ),
                                                                )
Container()                                                                 :

                                                                ],
                                                            );
                                                        },
                                                    ),
                                                );
                                            } else {

                                                return Text('No data
available');
                                            }
                                        },
                                    ),
                                ],
                            ),
                        ),
```

```
                                    const SizedBox(height: 25,
                                    ),
                                    Padding( padding: const
EdgeInsets.symmetric(horizontal: 15),
                                        child: Column( crossAxisAlignment:
CrossAxisAlignment.start,
                                            children: [Text(
                                                'Describe incident',
                                                style: GoogleFonts.montserrat(textStyle: const
                                                    TextStyle(
                                                        fontWeight:
FontWeight.bold,
                                                        fontSize: 17,
                                                    ),
                                                ),
                                            ),
                                            const SizedBox(height: 8,
                                            ),
                                            TextFormField(controller:
_incidentDescription,
                                                maxLines: 5,
                                                decoration: InputDecoration(filled: true,
                                                    fillColor:
Colors.grey.shade100,
                                                    contentPadding: const
```

```
EdgeInsets.all(5),

OutlineInputBorder(

Colors.grey.shade200), BorderRadius.circular(10.0),

OutlineInputBorder(

Colors.grey.shade200), BorderRadius.circular(10.0),

here(optional)', GoogleFonts.montserrat(

Colors.grey),

                                                    ),
                                                ),
                                            ),
                                        ],
                                    ),
                                )
                            ],
                        ),

enabledBorder:

    borderSide:
        BorderSide(color:

    borderRadius:

),
focusedBorder:

    borderSide:
        BorderSide(color:

    borderRadius:

),
hintText: 'Describe crimehintStyle:

    textStyle:
        const TextStyle(color:
```

```
                                const SizedBox(height: 25),crimeTypeId == "" ||
_incidentDescription.text.isEmpty
                                        ? Padding(
                                            padding: const
EdgeInsets.symmetric(vertical: 25),
                                            child: Align(alignment:
Alignment.bottomCenter,
                                                child: Container(width:
MediaQuery.of(context).size.width * 0.9,
                                                    height: 60,
                                                    child: OutlinedButton(onPressed: () {},
                                                        style: ButtonStyle(
                                                            shape:
MaterialStateProperty.all(
                                                                RoundedRectangleBorder(borderRadius:
BorderRadius.circular(
                                                                    10.0,
                                                                ),
                                                            ),
                                                            backgroundColor:
MaterialStateProperty.all(
                                                                Colors.grey.shade200,
                                                            ),
                                                            side:
MaterialStateProperty.all(
                                                                BorderSide(color:
```

```
            Colors.grey.shade200),
                                                                    ),
                                                                ),
                                                            child: Text('HELP',
                                                                style:
GoogleFonts.montserrat(TextStyle(),                                 textStyle: const

                                                                ),
                                                            ),
                                                        ),
                                                    ),
                                                ),
                                            )
                                        : Padding(
                                            padding: const
EdgeInsets.symmetric(vertical: 25),
                                            child: Align(alignment:
Alignment.bottomCenter,
                                                child: Container(width:
MediaQuery.of(context).size.width * 0.9,
                                                    height: 60,
                                                    child: OutlinedButton( onPressed: () async {
                                                            setState(() { isLoading = true;
                                                            });
                                                            FocusScope.of(context).unf
ocus();
```

```
apiServices.subnmitCrime(

,

ext,

,

),

use_build_context_synchronously



EndUserBottomNav(),



successfully'), Duration(seconds: 3),



use_build_context_synchronously
```

```
var result = await

        userId.toString(), crimeTypeId.toString()

        _incidentDescription.t

        latitudeVal.toString()

        longitudeVal.toString(

        token.toString());
// ignore:

Navigator.push( context,
    MaterialPageRoute(
        builder: (context) =>

    ),
);


const snackBar = SnackBar(content:
    Text('Updated

    duration:

);

// ignore:

ScaffoldMessenger.of(conte
```

```
xt)

);                                                    .showSnackBar(snackBarsetState(() {
                                                        isLoading = false;
                                                    });
                                                },
                                                style: ButtonStyle(shape:

MaterialStateProperty.all(                              RoundedRectangleBorder(borderRadius:

                                                                10.0,
BorderRadius.circular(                                        ),
                                                        ),
                                                    ),
                                                    backgroundColor:

MaterialStateProperty.all(                              Colors.blue.shade400,
                                                    ),
                                                    side:
MaterialStateProperty.all(                              BorderSide(color:

Colors.grey.shade200),                              ),


                                                ),
                                                child: Text( isLoading == false
                                                            ? 'HELP'
                                                            : 'Please wait...',style:
```

```
GoogleFonts.montserrat(TextStyle(
                                                    textStyle: const

                                                        color: Colors.white,fontWeight:
FontWeight.bold),

                                                                ),
                                                            ),
                                                        ),
                                                    ),
                                                ),
                                            )
                                    ],
                                ),
                            ),
                    );
            }

        Widget buildMap() => Stack(children: [
                        GoogleMap(
                            mapType: MapType.normal, initialCameraPosition: CameraPosition(
                                target: _currentPosition!,zoom: 13,
                            ),
                            markers: {Marker(
                                    markerId: const
MarkerId('_currentPosition'),
                                    position: _currentPosition!,
                                    icon: BitmapDescriptor.defaultMarker,
```

```
                                ),
                            },
                            onMapCreated: ((GoogleMapController controller)
=>
                                    _mapController.complete(controller)),
                        ),
                        Positioned(top: 35,
                            left: 16,
                            child: status == 'Pending' && hideStatus == true
                                    ? GestureDetector( onTap: () {}, child:
                                        Container(
                                            width:
MediaQuery.of(context).size.width * 0.9,
                                            // height: 80,
                                            padding: const EdgeInsets.all(8),decoration: BoxDecoration(
                                                color: Colors.white,borderRadius:
BorderRadius.circular(8),
                                                boxShadow: [
                                                    BoxShadow(
                                                        color:
Colors.black.withOpacity(0.2),
                                                        blurRadius: 4,
                                                        offset: const Offset(0, 2),
                                                    ),
                                                ],
                                            ),
                                            child: Column( crossAxisAlignment:
```

```
CrossAxisAlignment.center,
                                        children: [Row(
                                            mainAxisAlignment:
MainAxisAlignment.end,
                                            children: [ IconButton(
                                                onPressed: () { hideStatus = false;
                                                },
                                                icon: const
Icon(Icons.close),
                                            ),
                                        ],
                                    ),
                                    const Text(
                                        'Ongoing request!',style: TextStyle(
                                                fontSize: 22, fontWeight:
FontWeight.w500),

                                    ),
                                    Text(
                                        'Submited
${dateFormat.format(userReportedDetails!.createdAt!)}',
                                        style: const TextStyle(color:
Colors.grey),
                                    ),
                                    const SizedBox(height: 10,
                                    ),
                                    userReportedDetails!.responded_to_

by != null
```

```
                                                ? Row(
                                                    mainAxisAlignment:
MainAxisAlignment.center,
                                                    children: [Text(
                                                        'Your request was
accepted',
                                                        style: TextStyle(color:
Colors.blue.shade600,
                                                            fontWeight: fontSize: 16),
FontWeight.w500,
                                                    ),
                                                    const SizedBox(width: 8,
                                                    ),
                                                    Icon(
                                                        Icons.check_circle,color:
Colors.blue.shade600,
                                                        size: 17,
                                                    )
                                                  ],
                                                )
                                                :
userReportedDetails!.cancelled_by != null
                                                    ? SizedBox(
                                                        width: 220,child:
                                                        Text(
                                                            'Cancelled but will
be attended to shortly.Please wait...',
```

```
                                                    style: TextStyle(color:

Colors.red.shade600,                                                    ),
                                                ),
                                            )
                                        : Text('Please wait...'),const SizedBox(
                                    height: 10,
                                ),
                                Column( children: [
                                        Text(
                                            userReportedDetails!.respond

ed_to_by != null                                    ? 'This disappers once

you have been served!' attended to shortly!',       : 'Your request will bestyle:

GoogleFonts.montserrat(),                           ),
                                        const SizedBox(height: 4,
                                        ),
                                        const SpinKitSpinningLines(color:    Colors.blue,
                                            size: 50.0,
                                        ),
                                        const SizedBox(height: 10,
                                        ),
```

```
                                                ],
                                            )
                                        ],
                                    ),
                                ),
                            )
                        : status == 'ongoing'
                            ? GestureDetector( onTap: () {}, child:
                                Container(
                                    width:
MediaQuery.of(context).size.width * 0.9,
                                    height: 80,
                                    padding: const EdgeInsets.all(8),decoration: BoxDecoration(
                                        color: Colors.white,borderRadius:
BorderRadius.circular(8),
                                        boxShadow: [
                                            BoxShadow(
                                                color:
Colors.black.withOpacity(0.2),
                                                blurRadius: 4,
                                                offset: const Offset(0, 2),
                                            ),
                                        ],
                                    ),
                                    child: Column( crossAxisAlignment:
CrossAxisAlignment.start,
                                        children: [Row(
```

```
                                                    mainAxisAlignment:
                                                            MainAxisAlignment.spaceB
etween,
                                                    children:[Row(
                                                            children: [Container(
                                                                    height: 25,
                                                                    width: 25,
                                                                    decoration:

BoxDecoration( Colors.blue.shade200,

                                                                        color:

BoxShape.circle, Border.all( Colors.white,

                                                                        shape:

width: 2),

                                                                        border:

                                                                                color:


                                                                    ),
                                                                    child: const Center(child: Text(
TextStyle(
                                                                            'J',
                                                                            style:
Colors.white,

                                                                                color: fontSize:12,
                                                                            ),
                                                                    ),
                                                            ),
                                                    ),
```

```
                                                    const SizedBox(width: 3,
                                                    ),
                                                    Row(
                                                        children: [Text(
                                                                'Joe Biden',style:

                                                                        textStyle:
GoogleFonts.montserrat(const TextStyle(
                                                                            fontWeig
ht: Weight.bold)),
                                                                                Font

                                                            ),
                                                            const SizedBox(width: 4,
                                                            ),
                                                            Text(
                                                                '2 min ago',style:
GoogleFonts.montserrat(const TextStyle(
                                                                        textStyle:
Colors.grey)),
                                                                            color:
                                                            )
                                                        ],
                                                    )
                                                ],
                                            ),
```

```
                                    GestureDetector(onTap: () {
                                            setState(() { hideStatus = false;
                                            });
                                        },
                                        child: const Icon( Icons.close, color:
                                            Colors.grey,
                                        ),
                                    )
                                ],
                            ),
                            const SizedBox(height: 10,
                            ),
                            Row(
                                children:[Text(
                                        'You have currentstyle:

ongoing service', GoogleFonts.montserrat(),
                                    ),
                                    const SizedBox(width: 4,
                                    ),
                                    const SpinKitRipple( color:
                                        Colors.blue,size: 20.0,
                                    )
                                ],
```

```
                                                    )
                                                ],
                                            ),
                                        ),
                                    )
                                : Container(),
                        ),
                    ],
                );
}
```

r home page.

```
import 'dart:async'; import 'dart:convert';
import 'package:crimereporter/apiservices/apiservices.dart';import
'package:crimereporter/model/crime_types.dart'; import 'package:crimereporter/model/officer.dart';
import 'package:crimereporter/pages/enduser/enduserbottomnav.dart';import
'package:geocoding/geocoding.dart' as geocoding; import 'package:google_fonts/google_fonts.dart';
import 'package:google_places_flutter/google_places_flutter.dart';import
'package:google_places_flutter/model/prediction.dart'; import 'package:flutter/material.dart';
import
```

```
'package:google_maps_flutter/google_maps_flutter.dart';import 'package:intl/intl.dart';
import 'package:location/location.dart';
import 'package:shared_preferences/shared_preferences.dart';import
'package:sliding_up_panel/sliding_up_panel.dart'; import 'package:uuid/uuid.dart';
import 'package:flutter_spinkit/flutter_spinkit.dart';import 'package:http/http.dart' as http;

class EndUserHome extends StatefulWidget {@override
    _EndUserHomeState createState() => _EndUserHomeState();
}

class _EndUserHomeState extends State<EndUserHome> {bool isPanelOpen = false;
    var uuid = const Uuid(); var location =
    Location();
    final Completer<GoogleMapController> _mapController =Completer<GoogleMapController>();


    LatLng? _currentPosition;@override

    void initState() {
        super.initState();getLocation();
        fetchUserData(); setState(() {});
    }

    Future<void> getLocation() async {
```

```
        bool serviceEnabled; PermissionStatus permissionGranted;

        serviceEnabled = await location.serviceEnabled();if (!serviceEnabled) {
            serviceEnabled = await location.requestService();if (!serviceEnabled) {
                return;
            }
        }


        permissionGranted = await location.hasPermission();if (permissionGranted ==
        PermissionStatus.denied) {
            permissionGranted = await
location.requestPermission();
            if (permissionGranted != PermissionStatus.granted) {return;
            }
        }


        // return location.getLocation();
        location.onLocationChanged.listen(
            (LocationData currentLocation) {
                if (currentLocation.altitude != null &&currentLocation.longitude != null) {
                    setState(() {
                        _currentPosition = LatLng(currentLocation.latitude!,
currentLocation.longitude!);
                        cameraPorision(_currentPosition!);
                        // print(_currentPosition);
                    });
```

```
                    }
                },
            );
        }


    Future<void> cameraPorision(LatLng pos) async {final GoogleMapController
        controller = await
_mapController.future;
        CameraPosition newCameraPosition =CameraPosition(target:
pos, zoom: 13);
        await controller.animateCamera( CameraUpdate.newCameraPosition(newCameraPosition),
        );
    }


    ScrollController? controller;
    PanelController panelController = PanelController();

    final TextEditingController searcLocation =TextEditingController();
    final TextEditingController _incidentDescription =TextEditingController();
    DateFormat dateFormat = DateFormat();String? latitudeVal;
    String? longitudeVal;
    ApiServices apiServices = ApiServices();

    String? status;
    bool hideStatus = true;

    late Future<List<CrimeTypes>> crimeTypes; Future<List<Officer>> userReported = Future.value([]);
```

```
    Officer? userReportedDetails;

    String? userId;String?
    token;
    String crimeTypeId = ""; void fetchUserData()
    async {
        SharedPreferences prefs = await
SharedPreferences.getInstance();
        String jsonDataString = prefs.getString('userData') ??
'';
        Map<String, dynamic> jsonData =
json.decode(jsonDataString);setState(() {
            userId = jsonData['data']['id'];token = jsonData['token'];
        });
        crimeTypes = apiServices.fetchCrimesType(token.toString());
        Timer.periodic(const Duration(seconds: 3), (timer) async
{
                                apiServices.getUserReported(token.toString(),
userId.toString());
            setState(() { userReported =
                    apiServices.getUserReported(token.toString(),userId.toString());
            });
            List<Officer> result = (await userReported)
                .where((item) => item.status == 'Pending')
                .toList();
            if (result.isNotEmpty && result != []) {setState(() {
```

```
                    status = result.last.status; userReportedDetails = result.last;
                });
                print('Status!!!!!!!!!!!! $status');
            } else { setState(() {
                    status = 'Completed';
                });
                SharedPreferences prefs = await
SharedPreferences.getInstance();
                prefs.remove('submittedCrime');
              }
            });
    }

    bool isLoading = false;

    @override
    Widget build(BuildContext context) {return Scaffold(
            appBar: isPanelOpen
                    ? AppBar(
                            automaticallyImplyLeading: false,title: Row(
                        children: [
                            Expanded(
                                child: GooglePlaceAutoCompleteTextField(textEditingController:
                                    searcLocation, googleAPIKey:
"AIzaSyA7zmasenUJGF1QDQrmPdrOzLxBL4864s0",
                                    inputDecoration: const
InputDecoration(
```

```
                                        enabledBorder: UnderlineInputBorder(borderSide:
                                            BorderSide.none,
                                        ),
                                        focusedBorder: UnderlineInputBorder(borderSide:
                                            BorderSide.none,
                                        ),
                                        hintText: 'Search your location',hintStyle: TextStyle(
                                            color: Colors.grey,
                                        ),
                                        prefixIcon: Icon( Icons.search, color:
                                            Colors.grey,
                                        ),
                                    ),
                                    debounceTime: 800, countries: const ["ke"],
                                    isLatLngRequired:
                                            true, // if you requiredcoordinates from place detail
                                    getPlaceDetailWithLatLng: (Prediction
prediction) {
                                        setState(() { latitudeVal =
prediction.lat.toString();
                                            longitudeVal =
prediction.lng.toString();


                                        });
                                    },
                                    itemClick: (Prediction prediction) {searcLocation.text =
prediction.description.toString();
```

```
                                        searcLocation.selection =
TextSelection.fromPosition(
                                            TextPosition(offset:
prediction.description!.length),
                                        );
                                    },
                                    itemBuilder: (context, index,Prediction prediction) {
                                        return Container(
                                            padding: const EdgeInsets.all(10),child: Row(
                                                children: [
                                                    const Icon(Icons.location_on),const SizedBox(
                                                        width: 7,
                                                    ),
                                                    Expanded(child:
Text(prediction.description ?? ""),
                                                    )
                                                ],
                                            ),
                                        );
                                    },
                                    seperatedBuilder: const Divider(),isCrossBtnShown: true,
                                ),
                            ),
                        ],
                    ),
                    backgroundColor: Colors.white,elevation: 0,
```

```
                        toolbarHeight: 80.0,
                    )
                : null,
            body: _currentPosition == null
                ? const Center(
                    child:          SpinKitRipple(         color:
                        Colors.blue,size: 50.0,
                    ),
                )
                : SlidingUpPanel(
                    // minHeight: MediaQuery.of(context).size.height *
0.07,
                    minHeight: MediaQuery.of(context).size.height
* 0.1,

* 0.6,                    maxHeight: MediaQuery.of(context).size.heightborderRadius:
                        const BorderRadius.vertical(top:


Radius.circular(18)),
                    controller: panelController,parallaxEnabled: true,
                    parallaxOffset: 0.5, onPanelOpened: () {
                        setState(() { isPanelOpen = true;
                        });
                    },
                    onPanelClosed: () {setState(() {
                        isPanelOpen = false;
                        });
```

```
                    },
                    body: buildMap(),
                    panelBuilder: (controller) => Stack(children: [
                            ListView(
                                controller: controller, padding:
                                EdgeInsets.zero,children: [
                                    const SizedBox(height: 20,
                                    ),
                                    Visibility(
                                        visible: isPanelOpen == false,child: Column(
                                            children: [Center(
                                                    child: GestureDetector(onTap: () {
                                                            panelController.isPanelOpe
n
                                                                    ?
            panelController.close()
                                                                    :
            panelController.open();
                                                        },
                                                    child: Container(height: 5,
                                                        width: 30,
                                                        decoration: BoxDecoration(color:

Colors.grey.shade300,                                                   borderRadius:
```

```
            BorderRadius.circular(12)),
                                                    ),
                                                ),
                                            ),
                                            const SizedBox(height: 4,
                                            ),
                                            Center( child: Text(
                                                    'Submit Incident Report',style:


GoogleFonts.montserrat(),

                                                ),
                                            ),
                                            const SizedBox(height: 50,
                                            ),
                                        ],
                                    ),
                                ),
                                Padding( padding: const
EdgeInsets.symmetric(horizontal: 15),
                                    child: Column( crossAxisAlignment:
CrossAxisAlignment.start,
                                        children: [Text(
                                                'Pick incident type',
                                                style: GoogleFonts.montserrat(textStyle: const
                                                    TextStyle(
                                                        fontWeight:
```

```
FontWeight.bold,
                                    fontSize: 17,
                                ),
                            ),
                        ),
                        const SizedBox(height: 8,
                        ),
                        // DropdownButtonFormField(
                        //         decoration:
InputDecoration(
                        //             filled: true,
                        //             fillColor:
Colors.grey.shade100,
                        //             contentPadding: const
EdgeInsets.all(5), OutlineInputBorder(
                        //             enabledBorder:
                        //                 borderSide:
Colors.grey.shade200), BorderRadius.circular(10.0),
                        //                         BorderSide(color:
                        //             borderRadius:
OutlineInputBorder(
                        //             ),
                        //             focusedBorder:
Colors.grey.shade200), BorderRadius.circular(10.0),
                        //                 borderSide:
                        //                         BorderSide(color:
                        //             borderRadius:
                        //             ),
```

```
                                                    //                 hintText: 'Pick type of
crime', GoogleFonts.montserrat(
                                                    //                 hintStyle:

                                                    //                     textStyle:
                                                    //                         const
TextStyle(color: Colors.grey),
                                                    //                 ),
                                                    //             ),
                                                    //             items:
criminalTypes.map((String value) {
                                                    //                 return
DropdownMenuItem<String>(
                                                    //                     value: value,
                                                    //                     child: Text(value),
                                                    //                 );
                                                    //             }).toList(),
                                                    //             onChanged: (value) {},
                                                    //             isExpanded: true,
                                                    // ),

                                                    FutureBuilder( future: crimeTypes,
                                                        builder: (context, snapshot) {if
                                                            (snapshot.connectionState
==

{
                                                                    ConnectionState.waiting)return const


                                                                Center(
                                                                    child:
                CircularProgressIndicator(),
                                                                );
```

```
(snapshot.hasError) {

${snapshot.error.toString()}'),

&&

snapshot.data!;

t).size.width * 0.9, ListView.separated(

Axis.horizontal, crimeTypesList.length,

(context, index) {

index) {
```

```
} else if

    return Center(child: Text(
                'Error:

    );
            } else if (snapshot.hasData

        snapshot.data != null) {var crimeTypesList =

    return Container(height: 100,
        width:
                MediaQuery.of(contexchild:

            scrollDirection: itemCount:

            separatorBuilder:

                return Container(width: 5,
                );
            },
            itemBuilder: (context,

                    var crime =
```

```
crimeTypesList[index];

Clip.none,


{
d = crime.id;

ypeId);

Container(


BoxDecoration( Colors.green.shade50,
us: Radius.circular(8),


Column(

ignment:
```

```
return Stack( clipBehavior:

    children: [ GestureDetector(
            onTap: () { setState(()

                        crimeTypeI

            });
            print(crimeT

        },
        child:

            height: 80,
            width: 80,
            decoration:

                color: borderRadi

                        Border

            ),
            child:

                    mainAxisAl
```

```
                                                                    MainAx
isAlignment.center,[
                                                                children:
r(
                                                                    Containechild:
const Icon( s.security_outlined,
                                                                        Icon


                                                                    ),
                                                                ),
crime_name,ign:                                                 Text(
                                                                    crime.
xtAlign.center,
                                                                    textAl

                                                                        Te
                                                                ),
                                                            ],
                                                        ),
                                                    ),
                                                ),
                                            crimeTypeId ==
crime.id Positioned(
                                                ?
4,
                                                    right: -

                                                    top: -2,child:
```

```
GestureDetector(() {                                                    onTap:
tate(() { imeTypeId = "";                                                 setS
                                                                            cr
Container(ration:                                                       });
                                                                      },
                                                                      child:
                                                                        deco
BoxDecoration(
                                                                          co
lor: Colors
    .blue.shade50,
                                                                          sh
ape:
    BoxShape.circle,
                                                                          bo
xShadow: [
BoxShadow(
    color: Colors.grey
            .withOpacity(
```

```
                        0.4),
    spreadRadius: 3,
    blurRadius: 5,offset:
                    const Offset(
        0,
        3,
    ),

),
                                                                    ],
                                                                ),
                                                                chil
d: const Icon( ons.close_outlined,
                                                                    Ic
lor: Colors.grey,
                                                                    co

                                                                ),
                                                            ),
Container()                                                     ),
                                                        )
                                                    :

                                                ],
```

```
                                                                    );
                                                                },
                                                            ),
                                                        );
                                                    } else {
                                                        return Text('No data
available');
                                                    }
                                                },
                                            ),
                                        ],
                                    ),
                                ),
                            const SizedBox(height: 25,
                            ),
                            Padding( padding: const
EdgeInsets.symmetric(horizontal: 15),
                                child: Column( crossAxisAlignment:
CrossAxisAlignment.start,
                                    children: [Text(
                                        'Describe incident',
                                        style: GoogleFonts.montserrat(textStyle: const
                                            TextStyle(
                                                fontWeight:
FontWeight.bold,
                                                fontSize: 17,
                                            ),
                                        ),
```

```
),
const SizedBox(height: 8,
),
TextFormField(controller:

_incidentDescription,

maxLines: 5,
decoration: InputDecoration(filled: true,
fillColor: contentPadding: const

Colors.grey.shade100,

enabledBorder:

EdgeInsets.all(5), OutlineInputBorder(

borderSide:
BorderSide(color:

Colors.grey.shade200),

borderRadius:

BorderRadius.circular(10.0),

),
focusedBorder:

OutlineInputBorder(

borderSide:
BorderSide(color:

Colors.grey.shade200), BorderRadius.circular(10.0),

borderRadius:

),
hintText: 'Describe crime

here(optional)',
```

```
                                                    hintStyle:
GoogleFonts.montserrat(
                                                        textStyle:
                                                            const TextStyle(color:
Colors.grey),

                                                    ),
                                                ),
                                            ),
                                        ],
                                    ),
                                )
                            ],
                        ),
                    const SizedBox(height: 25),crimeTypeId == "" ||
_incidentDescription.text.isEmpty
                            ? Padding(
                                padding: const
EdgeInsets.symmetric(vertical: 25),
                                child: Align(alignment:
Alignment.bottomCenter,
                                    child: Container(width:
MediaQuery.of(context).size.width * 0.9,
                                        height: 60,
                                        child: OutlinedButton(onPressed: () {},
                                            style: ButtonStyle(
                                                shape:
MaterialStateProperty.all(
                                                    RoundedRectangleBorder(
```

```
                                                    borderRadius:
BorderRadius.circular(
                                                        10.0,
                                                    ),
                                                ),
                                            ),
                                            backgroundColor:
MaterialStateProperty.all(
                                                Colors.grey.shade200,
                                            ),
                                            side:
MaterialStateProperty.all(
                                                BorderSide(color:
Colors.grey.shade200),
                                            ),
                                        ),
                                        child: Text('HELP',
                                            style:
GoogleFonts.montserrat(TextStyle(),
                                                textStyle: const
                                            ),
                                        ),
                                    ),
                                ),
                            ),
                        ),
                    )
                : Padding(
                        padding: const
EdgeInsets.symmetric(vertical: 25),
                        child: Align(
```

```
                                                    alignment:
Alignment.bottomCenter,
                                                    child: Container(width:
MediaQuery.of(context).size.width * 0.9,
                                                        height: 60,
                                                        child: OutlinedButton( onPressed: () async {
                                                                setState(() { isLoading = true;
                                                                });
                                                                FocusScope.of(context).unf
ocus(); apiServices.subnmitCrime(
                                                                var result = await
                                                                        userId.toString(), crimeTypeId.toString()
,
                                                                        _incidentDescription.t
ext,
                                                                        latitudeVal.toString()
,
                                                                        longitudeVal.toString(
),
                                                                        token.toString());
use_build_context_synchronously
                                                                // ignore:
                                                                Navigator.push( context,
                                                                    MaterialPageRoute(
EndUserBottomNav(),                                                     builder: (context) =>
```

```
successfully'), Duration(seconds: 3),

use_build_context_synchronouslyxt)

);

MaterialStateProperty.all(

BorderRadius.circular(

MaterialStateProperty.all(
```

```
        ),
      );

      const snackBar = SnackBar(content:
          Text('Updated

          duration:

      );

      // ignore: ScaffoldMessenger.of(conte

              .showSnackBar(snackBar

      setState(() { isLoading = false;
      });
  },
  style: ButtonStyle(shape:

          RoundedRectangleBorder(borderRadius:

                  10.0,
              ),
          ),
      ),
      backgroundColor:
```

```
                                                    Colors.blue.shade400,
                                                ),
                                                side:
MaterialStateProperty.all(
                                                    BorderSide(color:
Colors.grey.shade200),
                                                ),
                                            ),
                                            child: Text( isLoading == false
                                                    ? 'HELP'
                                                    : 'Please wait...',style:
GoogleFonts.montserrat(TextStyle(
                                                    textStyle: const
                                                        color: Colors.white,fontWeight:
FontWeight.bold),
                                                ),
                                            ),
                                        ),
                                    ),
                                ),
                            ),
                        ],
                    ),
                ),
            );
        }

        Widget buildMap() => Stack(
```

```
                children: [
                    GoogleMap(
                        mapType: MapType.normal, initialCameraPosition: CameraPosition(
                            target: _currentPosition!,zoom: 13,
                        ),
                        markers: {Marker(
                                markerId: const
MarkerId('_currentPosition'),
                                position: _currentPosition!,
                                icon: BitmapDescriptor.defaultMarker,
                            ),
                        },
                        onMapCreated: ((GoogleMapController controller)
=>
                                _mapController.complete(controller)),
                    ),
                    Positioned(top: 35,
                        left: 16,
                        child: status == 'Pending' && hideStatus == true
                                ? GestureDetector( onTap: () {}, child:
                                    Container(
                                        width:
MediaQuery.of(context).size.width * 0.9,
                                        // height: 80,
                                        padding: const EdgeInsets.all(8),decoration: BoxDecoration(
                                            color: Colors.white,
```

```
                                        borderRadius:
BorderRadius.circular(8),
                                        boxShadow: [
                                            BoxShadow(
                                                color:
Colors.black.withOpacity(0.2),
                                                blurRadius: 4,
                                                offset: const Offset(0, 2),
                                            ),
                                        ],
                                    ),
                                    child: Column( crossAxisAlignment:
CrossAxisAlignment.center,
                                        children: [Row(
                                                mainAxisAlignment:
MainAxisAlignment.end,
                                                children: [ IconButton(
                                                        onPressed: () { hideStatus = false;
                                                        },
                                                        icon: const
Icon(Icons.close),
                                                    ),
                                                ],
                                            ),
                                            const Text(
                                                'Ongoing request!',style: TextStyle(
                                                        fontSize: 22, fontWeight:
```

```
FontWeight.w500),
                                                    ),
                                                    Text(
                                                        'Submited
${dateFormat.format(userReportedDetails!.createdAt!)}',
                                                        style: const TextStyle(color:
Colors.grey),
                                                    ),
                                                    const SizedBox(height: 10,
                                                    ),
                                                    userReportedDetails!.responded_to_
by != null
                                                            ? Row(
                                                                mainAxisAlignment:
MainAxisAlignment.center,
                                                                children: [Text(
                                                                    'Your request was

accepted',                                                                  style: TextStyle(color:

Colors.blue.shade600,                                                                       fontWeight: fontSize: 16),

FontWeight.w500,


                                                                ),
                                                                const SizedBox(width: 8,
                                                                ),
                                                                Icon(
                                                                    Icons.check_circle,
```

```
                                                color:
Colors.blue.shade600,
                                                size: 17,
                                            )
                                        ],
                                    )
                                :
userReportedDetails!.cancelled_by != null
                                    ? SizedBox(
                                        width: 220,child:
                                        Text(
                                            'Cancelled but will
be attended to shortly.Please wait...',
                                            style: TextStyle(color:
Colors.red.shade600,
                                            ),
                                        ),
                                    )
                                    : Text('Please wait...'),const SizedBox(
                                height: 10,
                            ),
                            Column( children: [
                                Text(
                                    userReportedDetails!.respond
ed_to_by != null
                                        ? 'This disappers once
you have been served!'attended to
shortly!',
```

```
                                                    style:
GoogleFonts.montserrat(),
                                                ),
                                                const SizedBox(height: 4,
                                                ),
                                                const SpinKitSpinningLines(color:    Colors.blue,
                                                    size: 50.0,
                                                ),
                                                const SizedBox(height: 10,
                                                ),
                                            ],


                                        )
                                    ],
                                ),
                            ),

                        )
                    : status == 'ongoing'
                        ? GestureDetector( onTap: () {}, child:
                            Container(
                                width:
MediaQuery.of(context).size.width * 0.9,
                                height: 80,
                                padding: const EdgeInsets.all(8),decoration: BoxDecoration(
                                    color: Colors.white,borderRadius:
            BorderRadius.circular(8),
                                        boxShadow: [
```

```
                                                    BoxShadow(
                                                        color:
Colors.black.withOpacity(0.2),
                                                        blurRadius: 4,
                                                        offset: const Offset(0, 2),
                                                    ),
                                                ],
                                            ),
                                        child: Column( crossAxisAlignment:

CrossAxisAlignment.start,                       children: [Row(
                                                        mainAxisAlignment:
                                                            MainAxisAlignment.spaceB

                                                        children:[Row(
etween,                                                     children: [Container(
                                                                height: 25,
                                                                width:          25,
                                                                decoration:

                                                                    color:  shape:

BoxDecoration( Colors.blue.shade200,
                                                                    border:

BoxShape.circle, Border.all(
                                                                        color:

Colors.white, width: 2),
```

```
                                                                    ),
                                                                    child: const Center(child: Text(
                                                                            'J',
                                                                            style:
TextStyle(

                                                                                    color: fontSize: 12,
Colors.white,                                                               ),
                                                                        ),
                                                                    ),
                                                                ),
                                                                const SizedBox(width: 3,
                                                                ),
                                                                Row(
                                                                    children: [Text(
                                                                            'Joe Biden',style:

                                                                                    textStyle:
GoogleFonts.montserrat(const TextStyle(                                                                fontWeig

ht: Weight.bold)),                                                                                        Font

                                                                        ),
                                                                        const SizedBox(width: 4,
                                                                        ),
```

```
                                                    Text(
                                                        '2 min ago',style:
GoogleFonts.montserrat(
                                                                    textStyle:
                                                                        color:
const TextStyle(

Colors.grey)),
                                                                )
                                                            ],
                                                        )
                                                    ],
                                                ),
                                                GestureDetector(onTap: () {
                                                        setState(() { hideStatus = false;
                                                        });
                                                    },
                                                    child: const Icon( Icons.close, color:
                                                        Colors.grey,
                                                    ),
                                                )
                                            ],
                                        ),
                                        const SizedBox(height: 10,
                                        ),
                                        Row(
                                            children: [Text(
```

```
                                                    'You have currentstyle:
ongoing service', GoogleFonts.montserrat(),
                                                ),
                                                const SizedBox(width: 4,
                                                ),
                                                const SpinKitRipple( color:
                                                    Colors.blue,size: 20.0,
                                                )
                                            ],
                                        )
                                    ],
                                ),
                            ),
                        )
                    : Container(),
            ),
        ],
    );
}
```

l user home page.

```
import 'dart:async'; import 'dart:convert';
import 'package:crimereporter/apiservices/apiservices.dart';import
'package:crimereporter/model/crime_types.dart'; import 'package:crimereporter/model/officer.dart';
```

```
import 'package:crimereporter/pages/enduser/enduserbottomnav.dart';import
'package:geocoding/geocoding.dart' as geocoding; import 'package:google_fonts/google_fonts.dart';
import 'package:google_places_flutter/google_places_flutter.dart';import
'package:google_places_flutter/model/prediction.dart'; import 'package:flutter/material.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';import 'package:intl/intl.dart';
import 'package:location/location.dart';
import 'package:shared_preferences/shared_preferences.dart';import
'package:sliding_up_panel/sliding_up_panel.dart'; import 'package:uuid/uuid.dart';
import 'package:flutter_spinkit/flutter_spinkit.dart';import 'package:http/http.dart' as http;

class EndUserHome extends StatefulWidget {@override
    _EndUserHomeState createState() => _EndUserHomeState();
}

class _EndUserHomeState extends State<EndUserHome> {bool isPanelOpen = false;
    var uuid = const Uuid(); var location =
    Location();
    final Completer<GoogleMapController> _mapController =Completer<GoogleMapController>();

    LatLng? _currentPosition;
```

```
    @override
    void initState() { super.initState();
        getLocation(); fetchUserData();
        setState(() {});
    }

    Future<void> getLocation() async { bool serviceEnabled;
        PermissionStatus permissionGranted;

        serviceEnabled = await location.serviceEnabled();if (!serviceEnabled) {
            serviceEnabled = await location.requestService();if (!serviceEnabled) {
                return;
            }
        }


        permissionGranted = await location.hasPermission();if (permissionGranted ==
        PermissionStatus.denied) {
            permissionGranted = await
location.requestPermission();
            if (permissionGranted != PermissionStatus.granted) {return;
            }
        }


        // return location.getLocation();
        location.onLocationChanged.listen(
```

```
            (LocationData currentLocation) {
                if (currentLocation.altitude != null &&currentLocation.longitude != null) {
                    setState(() {
                        _currentPosition = LatLng(currentLocation.latitude!,
currentLocation.longitude!);
                        cameraPorision(_currentPosition!);
                        // print(_currentPosition);
                    });
                }
            },
        );
    }


    Future<void> cameraPorision(LatLng pos) async {final GoogleMapController
        controller = await
_mapController.future;
        CameraPosition newCameraPosition =CameraPosition(target:
pos, zoom: 13);
        await controller.animateCamera( CameraUpdate.newCameraPosition(newCameraPosition),
        );
    }


    ScrollController? controller;
    PanelController panelController = PanelController();

    final TextEditingController searcLocation =TextEditingController();
    final TextEditingController _incidentDescription =TextEditingController();
```

```
    DateFormat dateFormat = DateFormat();String? latitudeVal;
    String? longitudeVal;
    ApiServices apiServices = ApiServices();

    String? status;
    bool hideStatus = true;

    late Future<List<CrimeTypes>> crimeTypes; Future<List<Officer>> userReported =
    Future.value([]);Officer? userReportedDetails;

    String? userId;String?
    token;
    String crimeTypeId = ""; void fetchUserData()
    async {
        SharedPreferences prefs = await
SharedPreferences.getInstance();
        String jsonDataString = prefs.getString('userData') ??
'';
        Map<String, dynamic> jsonData =
json.decode(jsonDataString);setState(() {
            userId = jsonData['data']['id'];token = jsonData['token'];
        });
        crimeTypes = apiServices.fetchCrimesType(token.toString());
        Timer.periodic(const Duration(seconds: 3), (timer) async
{
                                apiServices.getUserReported(token.toString(),
userId.toString());
```

```
            setState(() { userReported =
                        apiServices.getUserReported(token.toString(),userId.toString());
            });
            List<Officer> result = (await userReported)
                        .where((item) => item.status == 'Pending')
                        .toList();
            if (result.isNotEmpty && result != []) {setState(() {
                        status = result.last.status; userReportedDetails = result.last;
                    });
                    print('Status!!!!!!!!!!!!! $status');
            } else { setState(() {
                        status = 'Completed';
                    });
                    SharedPreferences prefs = await
SharedPreferences.getInstance();
                    prefs.remove('submittedCrime');
                }
            });
    }


    bool isLoading = false;

    @override
    Widget build(BuildContext context) {return Scaffold(
            appBar: isPanelOpen
                        ? AppBar(
```

```
                        automaticallyImplyLeading: false,title: Row(
                            children: [
                                Expanded(
                                    child: GooglePlaceAutoCompleteTextField(textEditingController:
                                        searcLocation, googleAPIKey:
"AIzaSyA7zmasenUJGF1QDQrmPdrOzLxBL4864s0",
                                        inputDecoration: const
InputDecoration(
                                            enabledBorder: UnderlineInputBorder(borderSide:
                                                BorderSide.none,
                                            ),
                                            focusedBorder: UnderlineInputBorder(borderSide:
                                                BorderSide.none,
                                            ),
                                            hintText: 'Search your location',hintStyle: TextStyle(
                                                color: Colors.grey,
                                            ),
                                            prefixIcon: Icon( Icons.search, color:
                                                Colors.grey,
                                            ),
                                        ),
                                        debounceTime: 800, countries: const ["ke"],
                                        isLatLngRequired:
                                                true, // if you requiredcoordinates from place detail
                                        getPlaceDetailWithLatLng: (Prediction
prediction) {
```

```
                                        setState(() { latitudeVal =
prediction.lat.toString();
                                            longitudeVal =
prediction.lng.toString();

                                        });
                                    },
                                    itemClick: (Prediction prediction) {searcLocation.text =
prediction.description.toString();
                                        searcLocation.selection =
TextSelection.fromPosition(
                                            TextPosition(offset:
prediction.description!.length),
                                        );
                                    },
                                    itemBuilder: (context, index,Prediction prediction) {
                                        return Container(
                                            padding: const EdgeInsets.all(10),child: Row(
                                                children: [
                                                    const Icon(Icons.location_on),const SizedBox(
                                                        width: 7,
                                                    ),
                                                    Expanded(child:
Text(prediction.description ?? ""),
                                                    )
                                                ],
                                            ),
```

```
                                        );
                                    },
                                    seperatedBuilder: const Divider(),isCrossBtnShown: true,
                                ),
                            ),
                        ],
                    ),
                    backgroundColor: Colors.white,elevation: 0,
                    toolbarHeight: 80.0,
                )
            : null,
        body: _currentPosition == null
            ? const Center(
                child:          SpinKitRipple(         color:
                    Colors.blue,size: 50.0,
                ),
            )
            : SlidingUpPanel(
                // minHeight: MediaQuery.of(context).size.height *
0.07,
                minHeight: MediaQuery.of(context).size.height
* 0.1,

* 0.6,                maxHeight: MediaQuery.of(context).size.heightborderRadius:
                    const BorderRadius.vertical(top:


Radius.circular(18)),
                controller: panelController,parallaxEnabled: true,
```

```
                    parallaxOffset: 0.5,onPanelOpened:
                    () {
                        setState(() { isPanelOpen = true;
                        });
                    },
                    onPanelClosed: () {setState(() {
                            isPanelOpen = false;
                        });
                    },
                    body: buildMap(),
                    panelBuilder: (controller) => Stack(children: [
                            ListView(
                                controller: controller, padding:
                                EdgeInsets.zero,children: [
                                    const SizedBox(height: 20,
                                    ),
                                    Visibility(
                                        visible: isPanelOpen == false,child: Column(
                                            children: [Center(
                                                    child: GestureDetector(onTap: () {
                                                            panelController.isPanelOpe
n
                                                                    ?
panelController.close()
```

```
                                                                    :
            panelController.open();
                                                                    },
                                                                    child: Container(height: 5,
                                                                        width: 30,
                                                                        decoration: BoxDecoration(color:

                                                                                borderRadius:
Colors.grey.shade300, BorderRadius.circular(12)),

                                                                    ),
                                                                ),
                                                            ),
                                                            const SizedBox(height: 4,
                                                            ),
                                                            Center( child: Text(
                                                                    'Submit Incident Report',style:

                                                                ),
GoogleFonts.montserrat(),
                                                            ),
                                                            const SizedBox(height: 50,
                                                            ),
                                                        ],

                                                    ),
                                                ),

                                                Padding( padding: const
```

```
EdgeInsets.symmetric(horizontal: 15),
                                        child: Column( crossAxisAlignment:
CrossAxisAlignment.start,
                                            children: [Text(
                                                    'Pick incident type',
                                                    style: GoogleFonts.montserrat(textStyle: const
                                                        TextStyle(
                                                            fontWeight:
FontWeight.bold,
                                                            fontSize: 17,
                                                        ),
                                                    ),
                                                ),
                                                const SizedBox(height: 8,
                                                ),
                                                // DropdownButtonFormField(
                                                //         decoration:
InputDecoration(
                                                //             filled: true,
                                                //             fillColor:
Colors.grey.shade100,
                                                //             contentPadding: const
EdgeInsets.all(5), OutlineInputBorder(
                                                //             enabledBorder:
Colors.grey.shade200),                          //                 borderSide:
                                                //                         BorderSide(color:
                                                //                 borderRad
```



THE END

(PAGE LEFT BLANK INTENTIONALLY)