2022

A Framework to secure data transmission in wearable heart-rate monitors using Elliptic Curve Cryptography (ECC).

Onyango, Oscar Omondi
School of Computing and Engineering Sciences
Strathmore University

A Framework to Secure Data Transmission in Wearable Heart-Rate Monitors Using Elliptic Curve Cryptography (ECC)

Oscar Omondi Onyango

September 2021

Submitted to the School of Computing and Engineering Sciences in partial fulfillment of the requirement of Degree of Master of Science in Information System Security

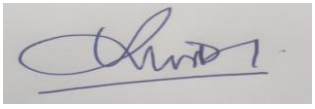
STRATHMORE UNIVERSITY

NAIROBI, KENYA

DECLARATION

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.

Oscar Omondi Onyango



3nd September 2021.

Approval

The thesis of Oscar Omondi Onyango was reviewed and approved by the following:

Dr. Vincent Omwenga

Research Director,

School of Computing and Engineering Sciences,

Strathmore University

Dr. Julius Butime,

Dean, School of Computing and Engineering Sciences,

Strathmore University

Dr. Bernard Shibwabo,

Director of Graduate Studies,

Strathmore University

ACKNOWLEDGMENT

I wish to thank God for having given me the strength and guidance in this dissertation right from the beginning.

Special thanks to my supervisor Dr. Vincent Omwenga, together with Dr. Joseph Sevilla and Strathmore University staff, especially those at @iLab Africa, for the great support I got from them and good relations throughout the time I have been a student of this great university.

I want to thank the IoT Department (@iLab Africa) under the leadership of Leonard Mabele and Joseph Shitote for playing a key role in this dissertation proposal by allowing me access to information and data that was valuable and useful in this proposal from start to completion.

I wish to acknowledge my parents, Francis Onyango and Ruth Onyango, for their constant financial, moral, and any other form. I also wish to thank my siblings, Pamela, David, and Hezborn, for their support and patience during my time at the university. Special thanks to Sharon Awuor for her continued support.

Thank you, and May God bless you all

ABSTRACT

The wearable technology refers to biological sensors which are conveniently attached to the patient's body to collect data about their heart rates, body temperature, oxygen levels, and physical activities. They mostly include smart watches. Unfortunately, maintaining data security in terms of integrity, confidentiality, and authenticity of the data during transfer in these wearables is becoming a challenge. Since cyber-criminals are always looking for new avenues to exploit, particularly in a sensitive field like healthcare, wearables can become their next big targets.

This study designs and implements an application-based security framework that uses Elliptic Curve Cryptography (ECC) to secure patient data during transmission from wearable heart-rate monitors. The study used integrative and methodological reviews to understand wearable technology by considering the technologies that support patient data sharing, techniques used to abstract data to enhance security during data transmission in wearable devices, and suitable public key encryption algorithms that can be implemented to ensure data security. It then proceeds to apply the Elliptic-Curve Cryptography (ECC) to develop the encryption application-based framework.

The results showed that Wi-Fi, Bluetooth, Global Positioning System (GPS), and Cellular Communication are the primary technologies supporting the wearables' data sharing. Data abstraction in wearables is achieved through differences in data models, data names, and counters. ECC was suitable for the implementation because it has smaller keys and can be computed substantially faster. The system's provision of authentication, confidentiality, and integrity was tested and validated through user tests. It was noted that data in the wearable devices regarding the heart-rate measurement were saved in an encrypted format using the user-generated cryptographic keys. Thus, an unauthorized person could not have access to the data. The passwords, keys, and usernames the user-created were stored using the SHA-hash algorithm in the server. The encrypted were uploaded to the server and could only be viewed or modified after decryption to ensure integrity.

Keywords- Internet of Things (IoT), wearable technology, cybersecurity, encryption algorithms, wearable heart-rate monitors

TABLE OF CONTENTS

DECLARATION ii

Approval iii

LIST OF FIGURES xi

LIST OF TABLES xiii

CHAPTER ONE: INTRODUCTION 1

1.1. Background to the Study 1

1.1.1. The Emerging Landscape of the Wearable Biosensor Technology in Healthcare 2

1.1.2. The Emerging Threat of Cybersecurity-related Issues in Wearable Monitors 3

1.2. Problem Statement 4

1.3. Objectives 5

1.3.1. General Objective 5

1.3.2. Specific Objectives 5

1.4. Research Questions 5

1.5. Significance of the Study 6

1.6. Scope and Limitations 6

CHAPTER TWO: LITERATURE REVIEW 7

2.1. Introduction 7

2.2. Wearable Technology 7

2.2.1. Understanding Wearables 7

2.2.2. Overview of Development of Wearable Technology 8

2.2.3. Wearable Technology in Healthcare 9

2.3. Technologies and Sensors that Support Patient Data Sharing in Wearables 11

2.3.1. Technologies that Support Patient Data Sharing in Wearables 11

2.3.2. Sensors that Support Patient Data Sharing in Wearables 13

Optical Sensor 14

Development of PPG Sensors 14

Components of the Photoplethysmography (PPG) Sensor 14

Metrics Obtained from PPG Sensors 16
Classification of Optical Sensors 17
Electrocardiography 18
Blood Pressure Measurement 19
Phonocardiography 20
2.4. Wearable Technology Challenges: Security and Privacy Issues 20
2.4.1. Unsecure Data Transmission via Bluetooth to Local Storage Devices 21
2.4.2. Software Communication to the Cloud via Wi-Fi Network and Cellular 21
2.4.3. Insecure Data Storage in Cloud 22
2.4.4. Lack of Authorization and Authentication 22
2.4.5. Lack of Physical Security Controls 22
2.5. Data Collection and Transfer from the Wearables 22
2.6. Data Abstraction in the Wearables 25
2.6.1. Levels of Data Abstraction 25
2.6.2. Abstraction Techniques 25
2.7. An Overview of the Existing Approaches used to Secure Communications and Data in Wearable Devices 26
2.7.1. Secure Communication Protocols 26
2.7.2. SRAM PUF 28
2.7.3. Blockchain 28
2.7.4. Data Tokenization 28
2.7.5. Pseudonymisation 29
2.7.6. Cryptography 29
2.8. Public Key Encryption 30
2.8.1. RSA Algorithm (Rivest-Shamir-Adleman) 32
2.8.2. Diffie-Hellman Public-Key 32
2.8.3. SHA-512 Algorithm 33
2.8.4. Elliptic-Curve Cryptography (ECC) 34
2.9. Conclusion 37

CHAPTER THREE: RESEARCH METHODOLOGY 38
3.1. Introduction 38
3.2. Research Approach for Objectives 1 and 2 38
3.3. Research Approach for Objectives 3 and 4 39
3.3.1. Planning 40
3.3.2 Requirement Gathering and Analysis 41
3.3.3. Design 41
3.3.3.1. Research Design 41
3.3.3.2. System Design 46
3.4.3. System Implementation 47
Implementation Tools 47
3.4.4. Testing and Deployment 47
Testing 47
Deployment 48
3.5. Validation 48
3.6. Research Quality Aspects 49
3.7. Ethical Considerations and Approval 49
CHAPTER FOUR: SYSTEM ANALYSIS, DESIGN, AND ARCHITECTURE 51
4.1. Introduction 51
4.2. Requirement Gathering and Analysis 51
4.3. System Analysis 53
4.3.1. Functional Requirements 53
4.3.2. Non-Functional Requirements 54
4.4. System Architecture 54
4.5.1. System Diagrams 56
4.5.1.1. Use Case Diagram 56
4.5.1.2. Sequence Diagram 58
4.5.1.3. Class Diagram 58
4.5.1.4. Data Flow Diagram (DFD) 59

4.5.1.5. Wireframes 60
4.5.2. Network and Security Designs 62
4.7. Chapter Summary 62
CHAPTER FIVE: SYSTEM IMPLEMENTATION AND TESTING 63
5.1. Introduction 63
5.2. System Implementation 63
5.2.1. Hardware Requirements 63
5.2.2. Software Requirements 63
5.2.3. System Development Segments 63
5.3. System Testing 72
5.3.2. Non-Functional Requirement Tests 73
5.3.3. Threat Alert System 74
5.4. System Validation 75
CHAPTER SIX: DISCUSSION 76
6.1. Introduction 76
6.2. The Technologies that Support Patient Data Sharing in the Wearable Device 76
6.3. Techniques Used to Abstract Data 78
6.4. The Encryption Framework Implemented to Secure Patient Data Transfer in Wearable Heart-rate Monitors Using Encryption Algorithms 78
6.5. Validation of the Implementation of the Security Framework 79
CHAPTER SEVEN: CONCLUSIONS AND RECOMMENDATIONS 81
7.1. CONCLUSIONS 81
7.2. RECOMMENDATIONS 81
7.3. FUTURE WORKS 82
REFERENCES 83
Appendix 1: Survey Questionnaire 87
Appendix 2: Survey Results 87
Section II: Heart-Rate Wearable Device 91
Appendix 3: Turnitin Report 93

Appendix 4: Ouriginal Report 93
Appendix 5: Ethical Review 94
Appendix 6: Encryption Code 94
Appendix 7: Arduino Code to Send Data from NodeMCU to MQTT Serve 98

LIST OF FIGURES

Figure 2.1: Physical Activities Monitored by Wearable……..9

Figure 2.2.: Wearable Estimate by 2023……...10

Figure 2.3: ZigBee Technology……13

Figure 2.4: PPG Sensor Development……...14

Figure 2.5: PPG Sensor Components……....15

Figure 2.6: PPG Signal Summary……..17

Figure 2.7: Activity Record……...17

Figure 2.8: Heart Rate Measurement Methods……..18

Figure 2.9: Generic Data Acquisition Architecture in Wearable Technology……..…21

Figure 2.10: Data Collection Systems in Wearables……..…23

Figure 2.11: Public Key Encryption Process……31

Figure 2.12: SHA-512 Iteration of Message Processing……..34

Figure 2.13: ECC Elliptic Curve……...…….35

Figure 3.1: Agile Software Development……..40

Figure 3.2: A Snippet of Email Confirmation from Respondents……44

Figure 4.1: System Architecture……....55

Figure 4.2: Use Case Diagram for the Wearable Device……...56

Figure 4.3: System Sequence Diagram……..…57

Figure 4.4: Class Diagram…….....58

Figure 4.5: Data Flow Diagram…….....59

Figure 4.6: Wireframes……..60

Figure 4.7: Wireframes……..61

Figure 5.1: Hardware Prototype…….64

Figure 5.2: Cloudmqtt Server……65

Figure 5.3: Live Data Streaming into the Server……...…65

Figure 5.4: Application Implementation Simulator……...66

Figure 5.5: Account Registration and Login Screens……...67

Figure 5.6: Application Home Screen……..….68

Figure 5.7: Key Generation……...68

Figure 5.8: Encryption Process……..69

Figure 5.9: Decryption Process……..70

Figure 5.10: Registered Users……71

Figure 5.11: Encryption and Decryption Database…..……...……71

LIST OF TABLES

Table 2.1: Systems and Options Available in Wearables…….…….....24

Table 2.2: Examples of Insecure Network Protocols and their Secure Alternatives……..27

Table 2.3: Components of Public Key Encryption……31

Table 2.4: Classes of Public Key Algorithms……....32

Table 3.1: Summary of Respondent Classification……43

Table 3.2: Data Summary……..45

Table 5.1: Functional Requirements Tests Summary…….…..….72

Table 5.2: Non-Functional Requirements Tests Summary……...….73

Table 5.3: System Response to Threats……74

Table 5.4: Functionality Tests and Validation……..…….75

Table 6.1: Summary of Data Transfer Technologies…….....77

CHAPTER ONE: INTRODUCTION

1.1. Background to the Study

Cardiovascular diseases, particularly heart attacks, are among the leading cause of disabilities and death worldwide, accounting for at least 17.6 million deaths every year (World Health Organization, n.d.). Heart attack is an abrupt and sometimes very deadly incident of coronary thrombosis, which typically causes the death of a section of the heart muscles (Chowdhury et al., 2019). As of 2012, the World Health Organization recorded that there are at least 712 deaths per 100,000 people, all related to heart problems (Kaptoge et al., 2019). World Health Organization's report further indicates that four are due to heart attacks in every five cardiovascular disease deaths, with most victims being over forty years old (Gallacher, 2019).

In Kenya, heart attacks account for 27 percent of the recorded deaths every year. In 2018, the Ministry of Education (MoH), while working with the World Heart Federation (WHF) and Kenya Cardiac Society (KCS), launched the National Guidelines for the Management of Cardiovascular Diseases (World Heart Federation, 2019). This was the ministry's attempt to equip health workers nationwide to help prevent, treat, and manage cardiovascular diseases and the associated risks. The goal was to improve the overall health outcomes of people suffering from heart-related diseases.

A year later, MoH, KCS, and WHF initiated the Access Accelerated Program, which involved utilizing innovative solutions and partnerships to improve the overall health welfare of patients with cardiovascular diseases (World Heart Federation, 2019). Initially, the project was intended to reach five counties in Kenya- Nyandarua, Isiolo, Kitui, Kisumu, and Nakuru (World Heart Federation, 2019). As of March 2019, the program had united more than 60 national stakeholders and medical experts. Their objective was to harness strategies to improve monitoring and evaluating patients' heart rates using modern technologies. The experts then advocated for utilizing real-time wearable technologies to monitor heart rates for patients as a pre and post-diagnosis tool.

In other words, a few years ago, electronic wearable was a completely new concept and was mainly used when it was necessary. Today, the technology has become very popular and is

widely used by everyone, including those that do not understand the concept of their operations and security-related issues. Furthermore, as Ometov et al. (2021) report in their study, the wearable market is expected to expand by at least 20 percent every year. As a result, the wearables have opened an extensive avenue of cyber-security exploits.

1.1.1. The Emerging Landscape of the Wearable Biosensor Technology in Healthcare

Although wearable technology has existed since the 1950s, it was not until 2013 when its craze exploded worldwide. In 2013, Google Glass was introduced into the market, and two years later, the Apple Watch was launched ("The History of Wearable Technology," 2018). In healthcare, wearable biosensor technology is used to extract clinical information based on heart rates, skin temperature, body motion, respiratory rates, and blood pressure. Currently, companies involved in the production of healthcare wearables include Apple, HTC Corporation, Google, Fitbit, Microsoft, Huawei, and Garmin.

According to Guk et al. (2019), wearable devices are real-time and apply non-invasive devices that allow them to monitor individuals continually and thus generate enough information that health experts can use to determine the person's health status or offer a preliminary diagnosis. Additionally, the sensors can be used to monitor the patient's physiological characteristics after treatments or therapeutics.

Wearable biosensor technology refers to biological sensors, including smart watches, glasses, rings, contact lenses, and clothing (Parlak et al., 2020). The devices are conveniently attached to the patient's body, which makes them easier to use. In the last few decades, advancements in electronics have accelerated the development of biocompatible and nanomaterials that achieve diagnosis and prognosis using small sensors, thus improving the quality and efficiency of health care. The first wearable device was the pacemaker which was developed for patients suffering from arrhythmia in 1958 (Guk et al., 2019). Since then, more advanced, flexible, and stretchable wearables have been developed. The optical heart-rates monitors are the most common wearable technology currently used to monitor patients' heart activity.

1.1.2. The Emerging Threat of Cybersecurity-related Issues in Wearable Monitors

Wearable technology is evolving as a category of the Internet of Things (IoT), offering life-challenging issues through applications like optical heart-rates monitors that are increasingly changing healthcare. The development of mobile networks, miniaturized microprocessors, and high-speed data transfer has further accelerated the usage of this technology (Olet, 2016). However, with its growth come the emergent issues of cyber-related crimes.

As a vulnerability, the wearable monitors lack encryption and operate in insecure wireless connectivity. The monitor devices connect wirelessly using Bluetooth, Near-field Communication (NFC), or WiFi protocols. According to Olet (2016), many of them operate using their operating systems and applications. They are driven or controlled by their self-built receiver, have a signal processor, and are power using small batteries. They operate as microcomputers which allow for various forms of connections, including information collection, processing, communication, and power supply. As mentioned above, they usually connect to other smart devices or data centers using near-field communication (NFC), infrared, Bluetooth, WiFi, and Radio-frequency Identification (RFID) which have facilitated remote patient-monitoring in health, which was previously not possible (McFarland & Olatunbosun, 2019). Due to their forms of connectivity and lack of encryption, wearable devices are increasingly becoming easy targets for hackers. They have numerous potential threats to the user.

First, the wearables lack hardware and software standards. According to Blow, Hu, and Hoppa (2020), the hardware and software used to manufacture these devices are proprietary. This means that all the wearable users depend solely on the security that individual manufacturers provide to them. If hackers successfully gain control of the manufacturers' system and corporate network, the wearable may also become vulnerable (Ching & Singh, 2016). This is because most wearables have a system that allows them to ship built-in security.

Secondly, the devices do not provide the basic security that most computers have. The majority do not require pin or password login, biometric access, or any other form of authentication. Once powered up, they begin their operations (Marrington, Kerr, & Gammack, 2016). In this case, if

the devices get in the wrong person's hands, the individual can easily access sensitive data stored or transmitted from the device.

Moreover, the devices facilitate extensive geographical tracking. They have geolocation data that updates every minute and uploads the same data to the cloud or another application. Unfortunately, this communication happens through an unsecured channel. Since IoT devices have a good history of poor security, wearable devices which fall under the same category continually expose users to similar risks (Marrington, Kerr, & Gammack, 2016).

Lastly, the wearables are not entirely classified as part of the Mobile Device Management Systems (MDM systems). MDM systems were created to manage the Bring your own device (BYOD) trend. BYOD is a trend that allows workers to connect to their organizational network using their personal devices and access confidential information or work-related systems. Some of the devices include tablets, USB drivers, smartphones, and computers (Padyab & Habibipour, 2021). Unfortunately, wearable devices are not classified under the BYOD trends because they work differently from the other devices mentioned above.

1.2. Problem Statement

The optical heart-rate sensors used in the wearable monitors for heart attacks lack embedded security for data transfer. Maintaining integrity, confidentiality, and authenticity of the data during transfer in these wearables is the next big challenge.

Cyber criminals are always looking for new avenues to exploit, particularly in a sensitive field like healthcare. The lack of embedded security in wearable monitors presents a promising opportunity for exploitation. A study by Bent et. (2020) shows that cybercrime on these wearable devices could compromise data integrity by allowing attackers to interfere with cause data accuracy, completeness, and consistency if the problem is not solved in time. Besides, hackers can compromise data confidentiality errors during transfer by injecting bugs, viruses, and malware into the data. The study also proves that most wearable devices require user authentication or have poor password practices, allowing hackers to overpass easily. Since the heart-rate wearables record and monitor sensitive data about the patient’s overall well-being, it is necessary to safeguard this information by ensuring it cannot be compromised during the transfer from the devices to a listening port.

1.3. Objectives

1.3.1. General Objective

The purpose of this study is to design and implement an application-based framework using elliptic curve cryptography to secure patient data from heart-rate wearable monitors to ensure data integrity, confidentiality, and authentication.

1.3.2. Specific Objectives

i. To determine the technologies that support patient data sharing in the wearable device by analyzing patient data transmission from the sensor to the listening port.

ii. To review techniques used to abstract data and approaches used to secure communications in wearable devices.

iii. To design, develop, and test an application framework to secure patient data transfer in wearable heart-rate monitors using a using elliptic curve cryptography

iv. To validate the application framework to ensure it provides integrity, confidentiality, and data authentication during the transfer from the wearable heart-rate monitors.

1.4. Research Questions

i. What technologies are used to support patient data sharing in wearable heart-rate monitors?

ii. What are the techniques used to abstract data to enhance security during data transmission in wearable devices?

iii. What encryption algorithms can be used to design, develop and test a framework to secure patient data during transfer in the wearable heart-rate monitors?

iv. What indicators show that the proposed framework offers data integrity, confidentiality, and authentication during transmission in wearable heart-rate monitors?

1.5. Significance of the Study

The study will provide data security during transmission in optical heart-rates wearable monitors by improving data integrity, confidentiality, and authentication. In other words, the proposed framework will prevent the disclosure of data to unauthorized persons, limit modification or deletion of data in an unauthorized manner, and ensure that the individual that accesses the data is the actual person. Through this, it will provide a continuous and efficient method of data transmission in the wearable device. The continuous and secure monitoring system will provide effective, efficient, and fast health care service to patients at risk of heart problems, even if the doctor and relatives are not near the patient. The study will envision to inspire more research related to securing patient data from wearable devices and other IoT-based systems using in modern healthcare.

1.6. Scope and Limitations

The study focuses on Wearable Wireless Sensor System (WWSS) used in healthcare in Kenya to monitor patients with heart problems. The system includes an optic sensor for heart attacks, methods of connectivity, and a listening device.

The anticipated limitation of the study is handling large amounts of data from the optical sensors. The study will mitigate this by using a cloud proxy extractor to activate anonymous IPs to avoid blocking automatically. The proposed framework will be implemented on a Windows Operating System using Android Studio.

CHAPTER TWO: LITERATURE REVIEW

2.1. Introduction

The study will apply for a majorly integrative review with a few elements of methodological review. The review's goal will be to criticize and synthesize representative literature on a topic in an integrated way such that new frameworks and perspectives on the topic are generated. In particular, it will pay attention to wearable wireless sensor systems (WWSS) used in healthcare in Kenya to monitor patients with heart problems. In order to give empirical evidence, the study will measure data behaviors and the factors that influence data-related security breaches. The importance of this measurement will be to conceptualize Dr. H. Hames Harrington phenomenon that states that, “Measurement is the first step that leads to control and eventually to improvement. If you cannot measure something, you cannot understand it. If you cannot understand it, you cannot control it. If you cannot control it, you cannot improve it” (Harrington, 1991). Also included in the section will be a review of technologies that support data transfer in wearables, data abstraction techniques during transmission, and frameworks for securing data during transfer.

2.2. Wearable Technology

2.2.1. Understanding Wearables

According to Casey (2020), “Wearables are electronic devices incorporated into daily use products that are comfortably worn on a body with added enhancements for tracking much big data information on a real-time basis.” Wearables are hands-free devices designed for practical daily activities and can be used to monitor pulse rate and keep track of the body’s activities since they are fitted with microprocessors. They have been enhanced and have the power to send and receive information through the internet. The devices are integrated with sensors that help monitor and collect different data based on the device's functionality (Casey, 2020). For example, smart watches and jewelry can monitor heart activities and communicate alerts to a physician if the individual experience a sudden shock or other medical condition.

2.2.2. Overview of Development of Wearable Technology

Wearable devices like iBeat, Hotler, and Apple Watch have become very common in the recent past. People are increasingly becoming conscious of their help. So, they have become closer to these devices more than ever. The younger generation, for example, has adapted the technology and continues to improve as the technology also improves. A question one would ask is how did wearable technology come into sudden play in human life?

The first invention of wearables was in the 13th century, when eyeglasses were created. Three hundred years later, China designed the first-ever smart wristwatch. However, before that, they were several inventions that were made. In 1955, Sony created the first radio transistor named "The Sony TR-55", which became the template for future related inventions ("The History of Wearable Technology," 2018). iPods and Game Boy are also based on the TR-55 technology. In 1961, Claude Shannon and Edward Thorp developed a new version of the wearables. They fitted a computer into a shoe. They used the technology as a predictive tool to help them cheat in the roulette games since they could tell where the ball would land. In the 1970s, wearable technology went mainstream. The first calculator wristwatch was developed in 1975. The first calculator was hugely recognized in the 70s and the 80s (Berglund, Duvall, & Dunne, 2016). Later in the 1980s, the Walkman was launched, and it became the people's choice of music platform. The Walkman wearable technology revolutionized the music industry as it popularized different musical genres around the world. As of 1987, more than 200 million units of the devices had been sold globally (Berglund, Duvall, & Dunne, 2016).

In 1994, a Canadian Researcher by the name of Steve Mann designed and developed a wearable wireless webcam (Berglund, Duvall, & Dunne, 2016). Although his device was bulky, it successfully exploited the IoT technologies, creating a good groundwork for other related inventions in IoT. After that, stakeholders in the tech world began to hold wearable technology conferences to find a way to popularise the technology. As a result, in the 2000s, wearable technology found its breakpoint when the Bluetooth headsets, Fitbits, and Nike iPods, among other devices, were created. Later in 2013, the technology exploded and became the norm of the day (Loncar-Turukalo et al., 2019). This craze was further facilitated when Google Glass came into the market in 2013, and Apple Watch joined two years later. In 2016, the Oculus Rift

Headset was also introduced into the market (Berglund, Duvall, & Dunne, 2016). Today, tech designers are increasingly developing more wearable technology devices.

2.2.3. Wearable Technology in Healthcare

While these inventions were taking place in other sectors, healthcare was also experiencing related improvements. Wearable Technologies are considered innovative solutions to various healthcare problems. In healthcare, the devices have been designed to monitor and maintain health like monitoring physical activity, weight control, and heart-rate monitoring (Wu, Li, Cheng, & Lin, 2016). As shown in Figure 2.1, the devices are used to measure data that include but are not limited to heart rate, blood pressure, oxygen saturation in the blood, posture, body temperature, and physical activities through the ballistocardiogram and the electrocardiogram, among other devices (Wu, Li, Cheng, & Lin, 2016).

Figure 2.1: Physical Activities Monitored by Wearables (Source: Wu, Li, Cheng, & Lin, 2016)

A study by IBM Watson proved that, on average, an individual generates at least one million gigabytes of health data during their lifetime (Strickland, 2019). Over the years, the increasing use of wearable health monitors has resulted in the health sector collecting a lot of health data. The challenge has been developing a seamless data-driven methodology to increase the maximum utilization of this data and minimize the associated risks. According to the University

of Illinois Chicago (2020), by 2023, the IDTechEx market research projects that wearable technology among adults will have reached S100 billion, as indicated in Figures 2.2.

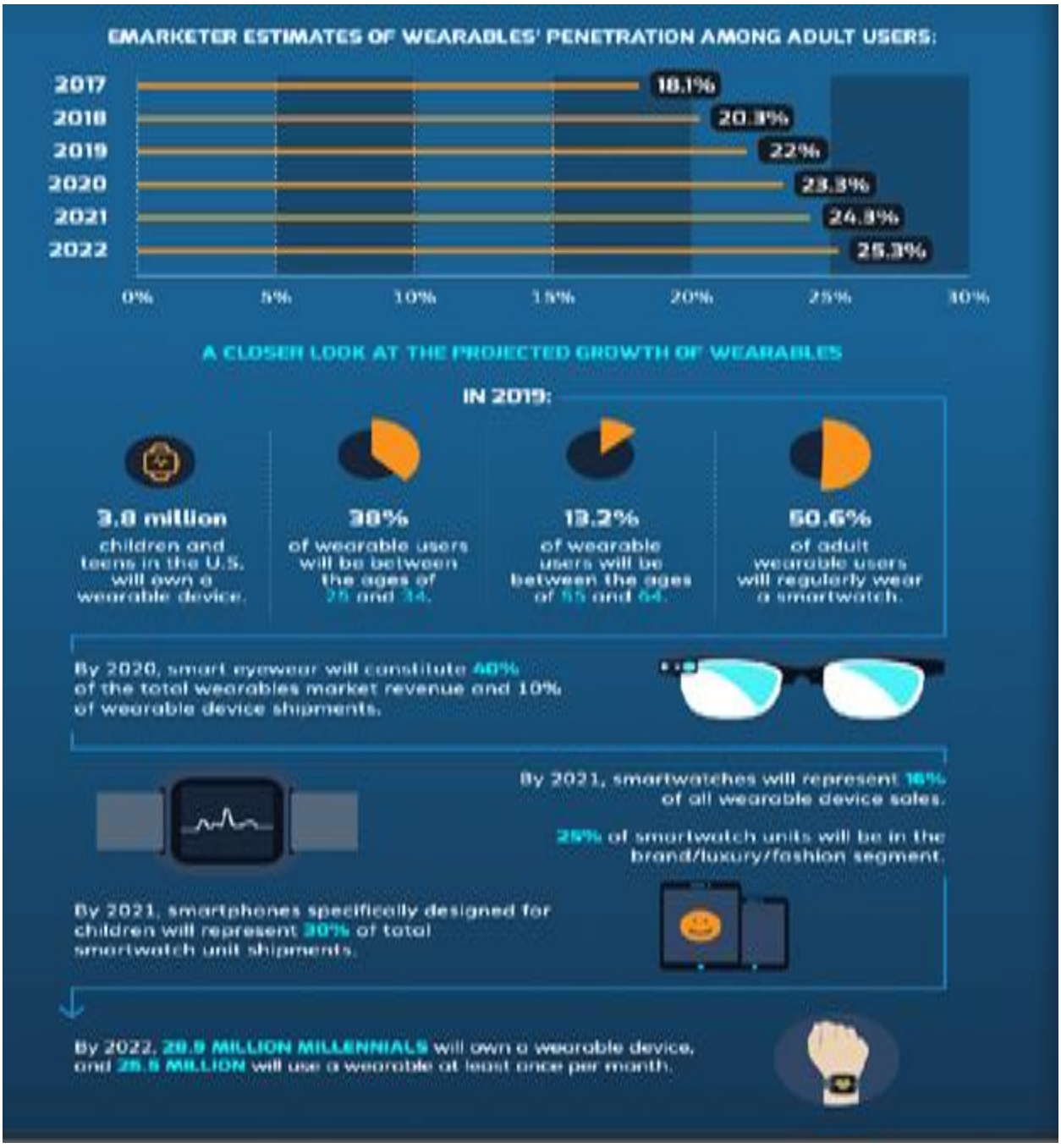


Figure 2.2: Wearable Estimate by 2023 (Source: University of Illinois Chicago, 2020)

2.3. Technologies and Sensors that Support Patient Data Sharing in Wearables

2.3.1. Technologies that Support Patient Data Sharing in Wearables

Wearables use wireless communication technologies to transmit patient data from devices to the server. Wireless communication was first used in the 19th century and has developed over the years to become complex but efficient (Ghamari, Arora, Sherratt, & Harwin, 2015). Through wireless technologies, data is transmitted through the air without the use of any cables or other electronic conductors. Wearables use radio, Wi-Fi, Bluetooth, GPS, ZigBee, satellite, WiMax, and Cellular Communication technologies.

Radio Communication: It is one of the oldest forms of wireless technologies, which is still used to date. The technology has portable multi-channel radios that allow radio users to communicate over short distances. It also has citizen bands and marine radios that facilitate long-distance communication used by sailors and truckers. Radio communication technologies broadcast through waves. They have a transmitter that sends the data in the signal form to the receiver (antenna) located in a distant location (Ghamari, Arora, Sherratt, & Harwin, 2015).

WiFi: This low-cost wireless communication technology consists of a wireless router functioning as the communication hub. The router is linked to other portable devices that have internet connections. Feng, Yan, and Liu (2019) state that Wi-Fi facilitates the internet connection of several devices depending on the router's configuration. However, it has a limited network range due to low power transmission, requiring uses to have proximity if they want to use the network. Wi-Fi is common among portable devices, and domestic and commercial service centers can be used to transmit large quantities of data, depending on the internet package the organization has acquired (Feng, Yan, & Liu, 2019). Wi-Fi is the most commonly used form of wireless technology because of its ease of integration and convenience, mobility, and expandability. For instance, it is easy to adjust Wi-Fi bandwidth to accommodate the increasing number of clients.

Cellular: A cellular technology (also called cellular networkElect) uses encrypted radio links that have been modulated to allow several users to communicate by transmitted and receiving data across a single frequency band (Atanasov, 2013). Since a person’s handset lacks a

significant broadcasting power, the system often relies on the cellular network, triangulating any signal's origin before handling the reception duties to the most preferred receive antennae (Atanasov, 2013). Data transmission through cellular technology is possible in most countries due to 4G network systems that offer enough speeds. However, countries like South Africa have since adopted the 5G technology is faster and efficient.

Bluetooth: Bluetooth is a wireless technology that allows the user to connect different electronic devices to share or transfer data with each other. Most smart devices have Bluetooth technology and can connect to other computers or handsets wirelessly to transmit information from one device to the other. It is the most commonly used form of wireless technology as it does require any internet connections. Bluetooth uses radio waves to allow communication between devices within a range of 15 to 50 feet (Feng, Yan, & Liu, 2019). It is a low-power signal technology that provides a maximum range of 50 feet with enough, allowing efficient data transmission between the connected devices. Moreover, the pairing process for the devices is easy and is configured to prevent any external interference from non-paired devices in the loop. Unfortunately, Bluetooth can also be used for a single device when the speed is relatively low compared to other wireless technologies (Feng, Yan, & Liu, 2019).

ZigBee: It is a wireless communication standard that was designed to address the unique requirement of low-cost wireless sensors, low-power, and control networks. According to Feng, Yan, and Liu (2019), the technology can be used almost everywhere globally because it is easy to implement. It also requires little power to operate as the technology has been designed to meet the demanding communication of transmitting data with the simplest structure (Feng, Yan, & Liu, 2019). Usually, ZigBee is used in commercial applications which involve sensors and monitoring applications. It is a suitable technology because of its flexibility that allows it to provide reliable wireless performance and power management. Figure 2.3 shows a picture of the ZigBee technology that has been deployed in much wireless communication mediums.

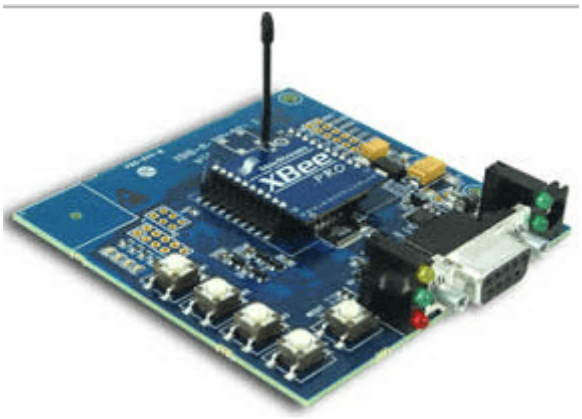


Figure 2.3: ZigBee Technology (Source: Feng, Yan, & Liu, 2019)

Worldwide Interoperability for Microwave Access (WiMAX): WiMAX is a wireless broadband system that provides Web surfing without any connection through cables or DSL. It is a family of wireless communication technology that apply the IEEE 802.16 set of standards. The standards provide different physical layers (PHY) and Media Access Control (MAC) options (Brain & Grabianowski, 2020). The technology can transmit data at a speed of more than thirty megabits per second. However, most providers offer an average of zero data rates of six megabits per second, making WiMAX significantly slower than hard-wired broadband. Besides, the cost of data while using WiMAX also depends on the versions of 4G wireless available on the handset. Regardless, if WiMAX is effectively implemented, it can transmit information faster and more accurately (Brain & Grabianowski, 2020). WiMAX can be used anywhere at any time. In operation, WiMAX is similar to Wi-Fi. The only difference is that it offers faster speed and can work for long distances and with many users compared to Wi-Fi. The WiMAX consists of a tower and receiver. The tower provides a wide area coverage of up to 8000 square kilometers, while the receiver is an antenna that could be a PCMCIA card or small box (Brain & Grabianowski, 2020).

2.3.2. Sensors that Support Patient Data Sharing in Wearables

Various techniques can be used to measure heart rate: Photoelectric pulse waves, photoplethysmography, electrocardiogram, and blood pressure measurement.

Optical Sensor

Most wearables like Fitbit, Garmin vivosmart, polar H10, and Xiaomi Mi band use photoplethysmography (PPG) sensors to measure the user's heart rate. The photoplethysmography (PPG) sensors are based on the fact that light that enters the body is spread predictably and is dependent on the changes in the blood flow dynamics like the changes in the volume of blood (cardiac output) and the changes in the pulses in the blood (heart rate) (Shirzadfar, Ghaziasgar, MPiri, & Khanahmadi, 2018).

Development of PPG Sensors

In the last five years, the development of PPG sensors has focused on better consumer applications as supported by wearable technology. Figure 2.4 below gives a brief history of the PPG sensors.

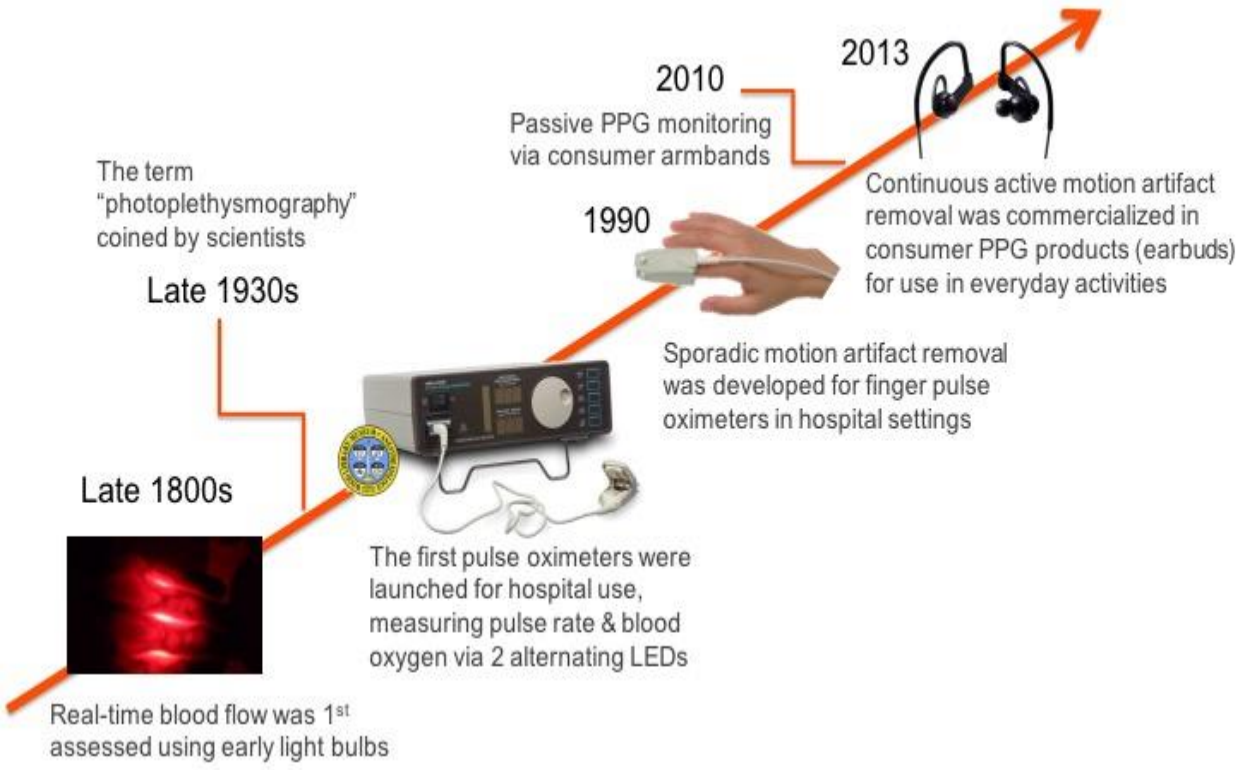


Figure 2.4: PPG sensor development (Source: Vallencell, 2020)

Components of the Photoplethysmography (PPG) Sensor

The photoplethysmography (PPG) has two sensors. The first sensor is used to detect light, and the other is used for determining motion or vibrations. PPG sensors apply four technical

components which it uses to measure the heart rate (Shirzadfar et al., 2018). An overview of the PPG sensor components are given in Figure 2.5 below;

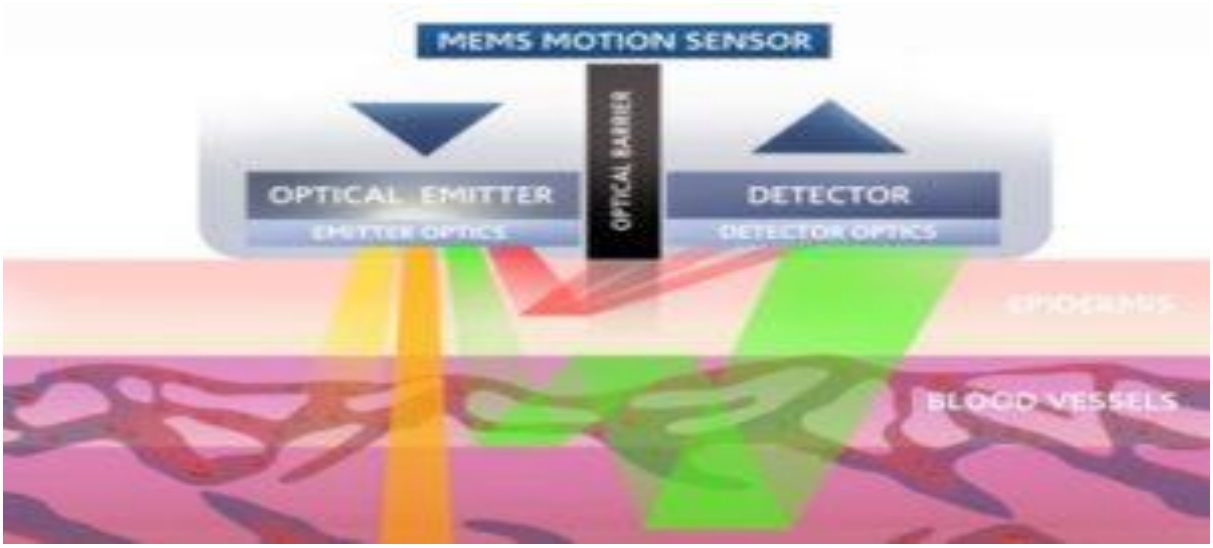


Figure 2.5: PPG Sensor Components (Source: Vallencell, 2020)

Optical Emitter: It comprises at least two light-emitting diodes (LEDs) that send direct light waves to the body's skin. Usually, because of the differences in skin thickness, tone, and morphology, the optical sensor is designed to have several LEDs fitted inside to allow the sensor to interact differently with different skins and tissues it come in contact with during its operation (Vallencell, 2020).

Digital Signal Processor (DSP): It captures the light refracted from the device user and translates the signals into ones and zeros. The ones are zeros that can be calculated to meaningful data about the heart rate.

Accelerator- It measures the motion of the user. It is used together with the DSP as inputs to the PPG algorithms. According to Tung (2020), most heart-rate wearables are fitted with a 3-axis accelerator that enables them to track the number of steps the user takes, distance traveled, number of floors climbed, and calories burned. The accelerometer uses electromechanical techniques to measure acceleration forces. For example, the Fitbit accelerometer is designed to sense vibrations and movements. The more sensitive the accelerometer is, the more powerful it can sensor more movements.

Algorithm- It processes the DSP signals and the movements data from the accelerometer into motion-tolerant heart rate data. The algorithm can also calculate other biometrics information like blood metabolic concentrations, blood pressure, heart rate variability, and oxygen levels in the blood, among others. For example, Fitbit and Garmin vivosmart software are designed to apply a special type of algorithms that it uses to change raw data from the accelerometer into usable information. According to Chandler (2020), the algorithms are secretes, and every firm makes improvements on their algorithms every year by conducting several experiments and tests to improve the device's accuracy compared to other hear-rate wearables in the field (Chandler, 2020). For instance, Fitbit has mastered tack and covert energy expenditures from the accelerometer through trial-and-error by using the information from the quantity of calories an individual's body burns. Similarly, the algorithm has been improved to calculate the number of steps one makes as they move. Lastly, Fitbit has also improved its algorithm to allow the devices to track the user's sleeping techniques (Chandler, 2020).

Metrics Obtained from PPG Sensors

Heart-rate wearables employ a high-quality PPG signal which allows them to monitor several biometrics, among them being;

i. Breathing rates- The number of breath the individual takes every 60 seconds
ii. VO_2 max- the volume of oxygen the patient uses. It determines the aerobic endurance of the individuals
iii. Oxygen saturation- The indication of the oxygen levels and concentration in the blood
iv. Heart-rate variability- The intervals between the blood pulses. It is also called the ECG beats. It is used to indicate stress levels and related cardiac issues.
v. Blood pressure- The indicator of cardiovascular health
vi. Cardiac Efficiency- Measures how efficiently the heart is working.

Figure 2.6 gives a simplified PPG signal that shows where each biometrics above is measured within the signal, while Figure 2.7 shows an example of an activity record obtained from a Fitbit wearable.

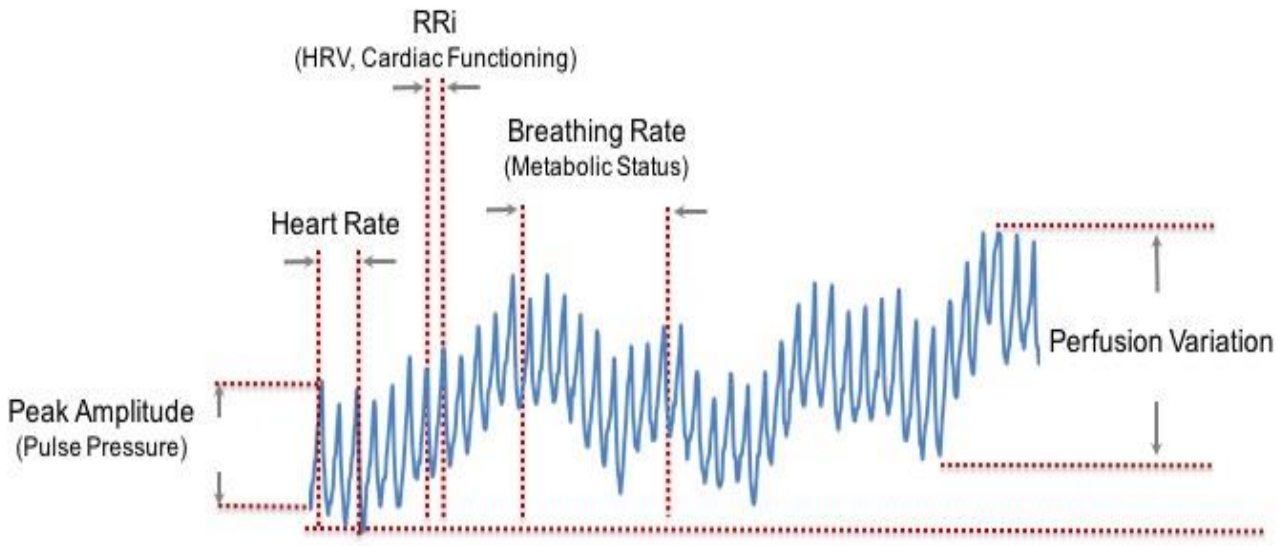


Figure 2.6: PPG Signal Summary (Source: Vallencell, 2020)

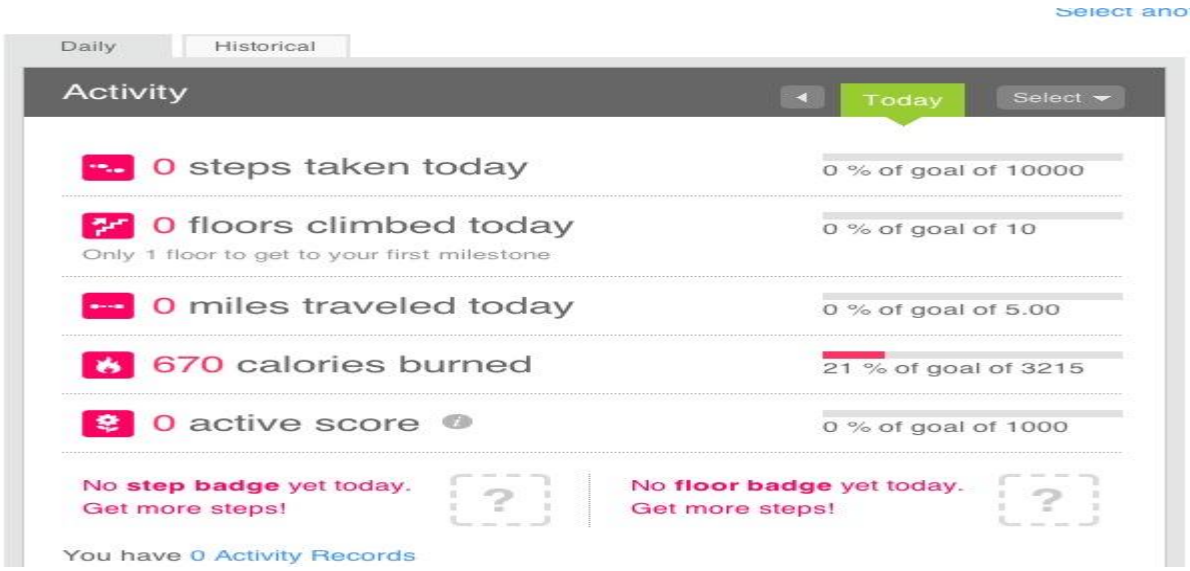


Figure 2.7: Activity Record (Source: Fitbit, 2020)

Classification of Optical Sensors

Pulse sensors apply the photoelectric method to measure heart rates. They are further classified into two depending on the method of measure, i.e., Reflection and transmission.

Transmission Pulse Sensors*:* Measure waves by emitting infrared or red light from the body surface. They also detect the changes in blood flow as the heartbeat changes depending on the

amount of light that the body receives. However, the transmission method can only be used where light can effectively penetrate the surface, like the earlobe or the fingertips ("Pulse Sensor," 2020).

The Reflection Pulse Sensor: Emits green light, red light, or infrared to the body. It then uses a phototransistor or photodiode to measure the amount of light that the body reflects. Usually, oxygenated hemoglobin in the arteries absorbs the incident light ("Pulse Sensor," 2020). Therefore, the sensor can measure the pulse wave signal by sensing the rate of blood flow characterized by the changes in the volume of the blood vessel. Figure 2.8 gives a summary of heart rate measurement methods using the pulse sensors.

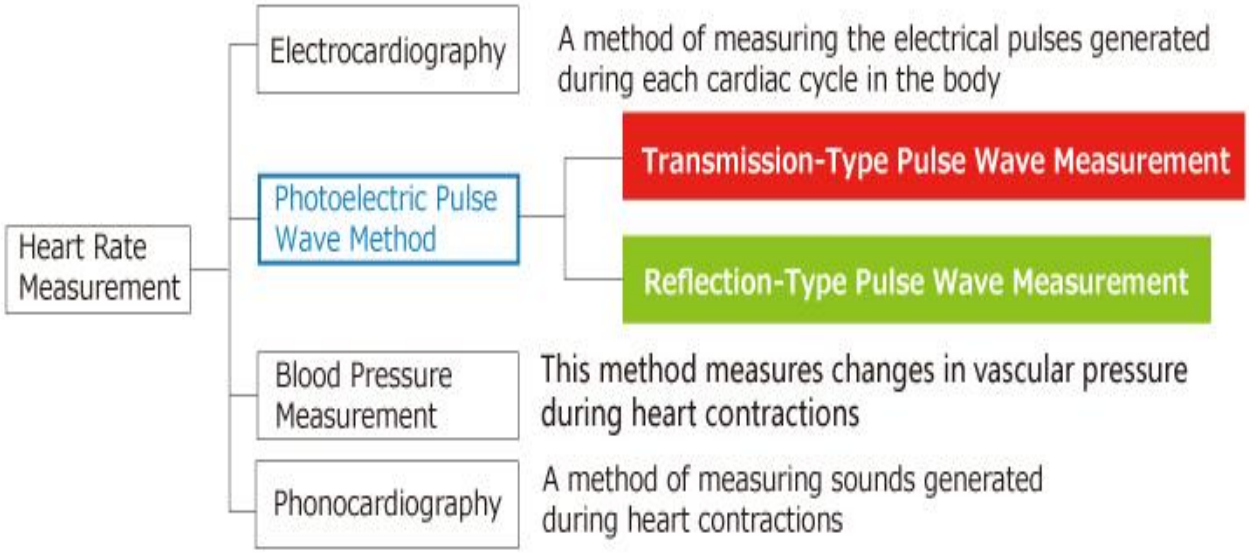


Figure 2.8: Heart rate measurement methods (Source: RIHM Semiconductor, 2020)

Electrocardiography

Electrocardiography (ECG) detects cardiac problems by measuring the electrical activities that the heart generates during contraction ("ECG Test," 2020). The electrocardiograph is the machine that doctors use to make a record of the patient's ECG. The electrocardiograph works by recording all the electrical activities of the heart muscles. It then displays the data through a screen or a piece of paper. Qualified medical practitioners can only interpret such data who understand the patient's dynamics ("ECG Test," 2020). Usually, doctors recommend the

electrocardiograph if the patient has or is experiencing symptoms related to fainting, dizziness, chain pain, irregular heartbeat, and shortness of breath. The method works best for patients who have been diagnosed with heart problems in the past or have a family issue related to heart diseases ("ECG Test," 2020). There are three types of ECGs;

i. Resting ECG: It requires the patient to lie on the device and minimize any form of movement. Electrical impulses generated by the muscles of the heart are recorded. The test may take between five to ten minutes ("ECG Test," 2020). However, the device is not portable, which disqualifies it as a wearable.

ii. Ambulatory ECG: A common example in the Holter ECG is commonly used for patients with heart-related issues. Holter is a wearable that allows the patient to go about their daily routine while the monitor is attached to their body taking recordings of their heart activities. The ambulatory ECGs are suitable for patients who have symptoms of intermittent heart issues or patients who are recovering from a recent heart attack, and thus, their hearts are not functioning normally ("ECG Test," 2020).

iii. Exercise Stress Test (EST): These are ECG wearables that take heart records when an individual is engaged in physical exercise like cycling, walking up a cliff, or walking on a treadmill.

Heart issues that ECGs can diagnose include heart enlargement, the poor blood supply to the heart, abnormal positioning of the heart, congenital heart problems, and abnormal heart rhythm ("ECG Test," 2020).

Blood Pressure Measurement

SeismoWatch and HeartGuide are examples of wearable devices that are used to measure blood pressure. According to Lazazzera, Belhaj, and Carrault (2019), the wearable is compatible with other apps like HeartAdvisors which allow the doctor to monitor and analyze from the wearable device. For example, HeartGuide is the latest ad more advanced technology that allows medics to track and manage blood pressure much easier compared to other technologies. The device proactively monitors the heart activity, thereby determining its overall health. It provides real-time data about the heart when the patient is engaged in different activities which affect their blood pressure.

Phonocardiography

It is a diagnostic method that creates graphic records of the sounds that are generated by the contracting heart. The phonocardiogram data may be generated from a chest microphone fitted with a miniature sensor at the tip of a tubular instrument that the doctor inserts into the blood vessel and the heart chambers. According to Goodman (2004), the phonocardiogram supplements information medics obtain from using other devices like the stethoscope.

2.4. Wearable Technology Challenges: Security and Privacy Issues

Wearable technology has several challenges, especially communication capacity, power consumption, design problems, and security and privacy issues. Security and privacy issues being the focus of this study. These can be categorized into security attacks, vulnerabilities, and solutions. For starters, security vulnerabilities in the wearables can be exploited by passive and active attacks on the devices (Ching & Singh, 2016). Usually, passive attacks focus on obtaining the user's credentials (passwords) and other sensitive information without interfering with the system.

On the other hand, active attacks involve altering the system to obtain information. Security vulnerabilities affect confidentiality, availability, authenticity, and integrity of data. Privacy attacks can be classified as data integrity and user identity attacks. Sometimes, they can be location or time-based attacks (Ching & Singh, 2016). As shown in Figure 2.9, the most popular security vulnerabilities present in wearable devices from the attack surfaces are insecure data transmission using Bluetooth for local device storage, lack of authentication and authorization, lack of physical security controls which creates enough opportunities for attacks, and software communication in the Cloud through Wi-Fi network and cellular (Ching & Singh, 2016).

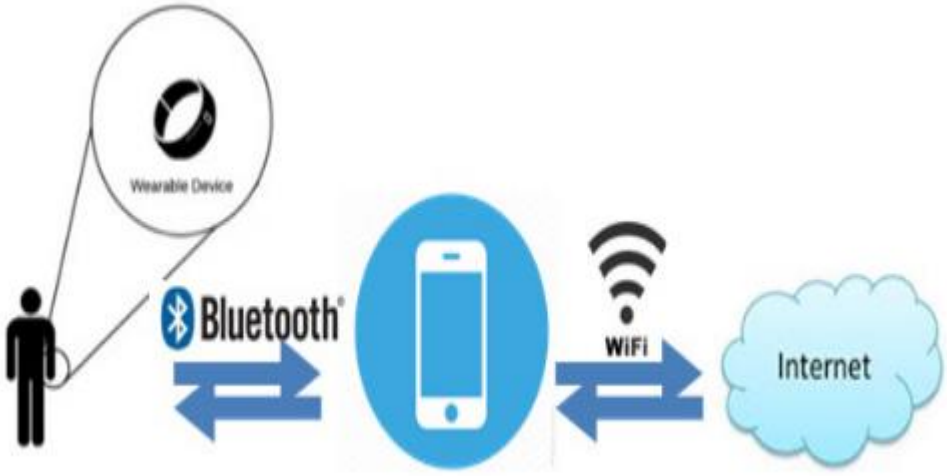


Figure 2.9: Generic Data Acquisition Architecture in Wearable Technology (Source: Ching & Singh, 2016)

2.4.1. Unsecure Data Transmission via Bluetooth to Local Storage Devices

Wearables depend on Bluetooth to transmit the data they collect from embedded sensors to integrated devices like smartphones. This presents an excellent opportunity for attackers to exploit a bug present within the devices to extract sensitive data locally stored, especially health records (Ching & Singh, 2016). For instance, an attacker can use a sniffer to steal data by detecting the broadcast signals when the wearable device communicates to an integration device using Bluetooth. As a result, the user will experience a loss which could be in terms of monetary, safety, and health losses.

2.4.2. Software Communication to the Cloud via Wi-Fi Network and Cellular

Software communication to the cloud over network or cellular is more risk-prone compared to Bluetooth communication discussed above. As Ching and Singh (2016) state, wireless communication via Wi-Fi or cellular creates an excellent opportunity for attackers to steal personal and sensitive information which are locally being transmitted from the local storage to cloud application. The data can be a combination of personally identifiable information like the name, emails, location, and phone numbers of the users, increasing the intensity of the risks (Ching & Singh, 2016). For example, the attacks can exploit security vulnerabilities like man-in-

the-middle attacks and redirection attacks. Such attacks cause data to be transmitted to the wrong server. Arguably, the potential loss is more severe in this form of communication.

2.4.3. Insecure Data Storage in Cloud

Cloud can be described as a public space that exists on the transmission lines between endpoints of the transfer. It provides better data or file accessibility. However, it could be the most vulnerable component in the wearable environment because of the amount of personally identifiable information that it contains. Data in the cloud is exposed to several threats, including denial of service attacks (DDoS), back door attacks, and SQL attacks. It requires a highly skilled attacker to steal data stored in the Cloud (Ghaffari, Gharaee, & Arabsorkhi, 2019).

2.4.4. Lack of Authorization and Authentication

Wearables do not often come with in-built security mechanisms like user authentication. Besides, they usually stored data in their original states, without any form of encryption whatsoever. Moreover, wearables devices demand high communication security about data encryption, integrity, and confidentiality because they operate over uncontrolled wireless networks like Wi-Fi, Bluetooth, and cellular to transfer their data (Ching & Singh, 2016). Besides, it is difficult to apply better security measures on these devices due to their small sizes and low bandwidths. A study by HP determined that at least thirty percent of smartwatches are vulnerable to account harvesting, an attack that allows an unauthorized user to gain access to the device and its data by exploited weak passwords and user enumeration (Ching & Singh, 2016).

2.4.5. Lack of Physical Security Controls

Loss of the devices itself is another challenge in securing their data. Given that most wearables are small, they can be easily stolen or seemingly misplaced by the user. Stolen or lost devices risk exposure to the stored personal information, thereby compromising its integrity, confidentiality, and availability (Ching & Singh, 2016).

2.5. Data Collection and Transfer from the Wearables

Wearables allow various methods in which other systems can collect data from the sensors. As shown in Figure 2.9 below, two types of systems can be used to collect data from the

wearables: proprietary systems and third-party systems. First, proprietary systems are found in the wearable and include apps used in smartphones, computers, and cloud services. Proprietary systems are delivered and maintained by the wearable vendors (Arriba-Pérez, Caeiro-Rodríguez, & Santos-Gago, 2016). The vendors use these systems to collect the user's information, perform analytics or avail the information to be used for analytics by the authorized third party. On the other hand, third-party systems include apps that are developed and maintained by external entities to provide specific functionalities.

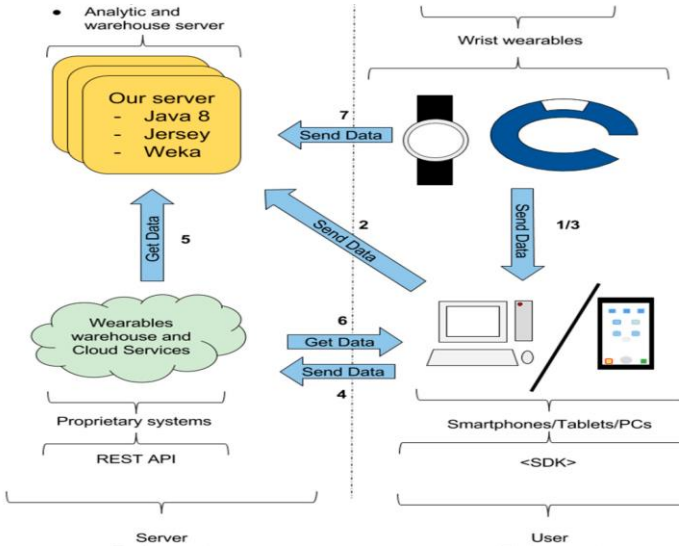


Figure 2.10: Data Collection Systems in Wearables (Source: Arriba-Pérez et al., 2016)

Data the wearables collect can be transferred to another computer as an intermediate step before it is eventually transferred to permanent data storage. Data transfer is produced through third-party apps or proprietary solutions or programs (Arriba-Pérez et al., 2016). In other instances, data from the wearables can be transferred directly to permanent data storage. Some wearable developers provide a software development kit (SDKs) to allow third parties to develop applications for the wearables to collect and send data directly. Proprietary warehouses and cloud providers always use a REST API that allows third parties to access the users' data (Arriba-Pérez et al., 2016). The following case examples illustrate data collection and transfer in different wearables;

i. Apple Health uses warehouse and cloud services, an app, and a health SDK. The cloud services maintain data collected from the users and facilitate other analytical

processes. The apple app can synchronize data present in the Apple Health server and allow the user to view the data. The SDK further allows for the development of other apps for iPhones, iPhones, and iWatch to give users access to the data available at the Apple Health Warehouse. It, however, lacks REST API to be used for third-party systems (Arriba-Pérez et al., 2016).

ii. Jawbone providers SDK for app developers, an SDK warehouse, and a REST API for third-party systems. Like Apple Health, it also stores data and gives access from connected Jawbone devices.

iii. Google Fit offers a complete solution of a warehouse, cloud services, SDK, and REST API for third-party systems. An android app transfers data from the wearable to the smartphone before being transferred to the warehouse. The app is fitted with artificial intelligence, which filters data from the best sensors (Arriba-Pérez et al., 2016).

iv. S-Health has a smartphone app that synchronizes data with a Samsung server. It has an SDK to allow third-party app development. However, it lacks REST API to be used for third-party systems (Arriba-Pérez et al., 2016).

v. Fitbit only has warehouse REST API for third-party systems. The platform can synchronize data from specific Fitbit brands and allow third-party developers to obtain such data using their REST API.

vi. Microsoft Health offers SDK and REST API for app development and third-party systems. Table 2.1 gives a summary of data collection and transfer in different wearables.

Table 2.1: Systems and Options Available in Wearables

Platforms	SDK-Sensors	SDK-Warehouse	REST API
Apple Health	-	X	-
Google fit	X	X	X
S-Health	-	X	-
Fitbit	-	-	X
Jawbone	-	X	X
Microsoft Health	X	-	X

Using the previous description, two modes can be used to transfer data from the wearables to third-party servers; Warehouse and Wearable data transfer. The wearable data transfer takes data directly from the sensor to the third-party server, while warehouse data transfer takes data from proprietary warehouses to the servers.

2.6. Data Abstraction in the Wearables

2.6.1. Levels of Data Abstraction

There are three levels of data abstraction in the wearables; internal, conceptual, and external levels. The internal level of data abstraction defines the physical storage structure of the database. It is a low representation of the whole database and contains multiple internal records (Muthumanickam, 2019). The level helps keep information about the actual representation of the whole database. It outlines the type of data store and how it is stored. Next, the conceptual level describes the structure of the whole database for the community users. It hides data about the physical storage structure and focuses on data type, relationships, and entities. It is also described as the logical level between the physical storage view and the user. Lastly, the external level describes the portion of the database which is user-specific. It hides other unrelated particulars about the database. Data abstraction is used in the wearables to ensure the user can access the same data and, at the same time, see what has been customized. It also ensures the user does not deal directly with details about the physical storage (Muthumanickam, 2019).

2.6.2. Abstraction Techniques

Differences in Data Models: Data from the wearables are organized in temporal segments using different models. The models are categorized using the elements and attributes in the key-value

pairs as in JavaScript Object Notation (JSON) and extensible Markup Language (XML) (Arriba-Pérez et al., 2016). For example, Microsoft manages to sleep, and heart rate under different segments called sleep and heart-rate segments.

The difference in Data Names: Each wearable uses different vocabularies to refer to similar parameters. For example, Microsoft managers sleep types using words like Doze, Awake, Sleep, and Snooze, while Google Fit uses Sleep lightly, deep, Rapid Eye Movement, and awake to describe different levels of sleep.

Temporal Discrepancies: Different wearables have a different methods of identifying traces. For example, Microsoft identifies various heart-rate periods using a single identifier per day while identifier each heart-rate period using a different identifier.

Differences in Counters: Each Wearable vendor follows a specific approach to count events produced by the devices. For example, Microsoft takes counts of all items produced like number of steps, distance, calories, and heart rate.

2.7. An Overview of the Existing Approaches used to Secure Communications and Data in Wearable Devices

2.7.1. Secure Communication Protocols

The majority of wearables communicate with internet servers for data storage and analysis. As a result, they require a secure communication protocol to ensure secure data. Since many of them have WiFi network interfaces, they can communicate with the internet. To enhance secure communication, it is important to have integrated security protocols. Some of the available communication protocols that ensure the security of data in wearables can apply to include Virtual Private Networks (VPNs), Secure Sockets Layer (SSL), File Transfer Protocol Secure (FTPS), and Hyper Text Transport Protocol Secure (HTTPS) (Phoenix Nap, 2019).

Table 2.2: Examples of Insecure Network Protocols and their Secure Alternatives

	Instead of...	Use...
Web Access	HTTP	HTTPS
File transfer	FTP, RCP	FTPS, SFTP, SCP, WebDAV over HTTPS
Remote Shell	telnet	SSH2 terminal
Remote desktop	VNC	radmin, RDP

Private and Virtual Private Networks (VPNs) are software-based. Private and Virtual Private Networks (VPNs) are closed to the outside world, which means they are not easily accessible to attacks from malicious uses. According to Phoenix Nap (2019), they isolate the communication between servers and clients and allow several servers under the same account to exchange data without being exposed to the public (Phoenix Nap, 2019). Next, Hyper Text Transport Protocol Secure (HTTPS) is a secure connection protocol used to navigate the World Wide Web (WWW). According to Ometov et al. (2020), HTTPS provides secure communication across a network since data is encrypted using the Transport Layer Security, formerly known as Secure Sockets Layer (SSL). HTTPS applies a secure certificated (SSL certificate) provided by a third-party vendor to secure the connections by verifying a legitimate site. The certificate enhances security by encrypting data and offering an extra layer of security for sensitive information. Lastly, File Transfer Protocol Secure (FTPS) is a technique that encrypts data files and ensures authentication of the information. FTPS employs command and data channels. The user is required to encrypt both channels (Phoenix Nap, 2019). The method only protects data in transit. It is an extension of the File Transfer Protocol that provides additional support for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL).

2.7.2. SRAM PUF

It is known as the SRAM Physical Unclonable Function (PUF). It creates a “device-unique, unclonable fingerprint.” Through this, it authenticates wearable devices. It allows secure sensitive health information contained or generated by the wearable devices during operation. The SRAM Physical Unclonable Function (PUF) thus limits identity theft for wearable users. It operates as a suitable technique for key generation and storage and protections counterfeiting the wearable technologies through cloning and software reverse engineering (Schrijen, 2021).

2.7.3. Blockchain

Wearables have attempted to exploit blockchain technology. Blockchain technology provides a stable data storage method that prevents an unauthorized user from stealing and altering the medical records captured by the wearable. According to Schrijen (2021), when wearable medical users want to share their records with other users, doctors, or organizations, they can do so securely because some of the devices come fitted with blockchain-chained based data security systems. The blockchain technology secures data by hashing it before storing it in blocks that are interconnected and unchangeable. It is often quite complex to breach the security of data stored using blockchain technology, even if the copies are shared through insecure channels (Schrijen, 2021).

2.7.4. Data Tokenization

It involves substituting data with randomly generated values known as tokens for a cleartext value. The process involves having a token vault (lookup table) that is securely saved. The cleartext value is then mapped to the corresponding token. Usually, each token's length and data type is similar to the cleartext value so that it is easy to retrieve them from the lookup table. The

process is also reversible, which makes it an excellent approach in protecting individual data fields from wearables (Tang & Shi, 2021).

2.7.5. Pseudonymisation

According to Hintze and El Emam (2018), data pseudonymization is a data security technique supported in The General Data Protection Regulation (GDPR). It increases data security and privacy for individual wearable users. Hintze and El Emam (2018) state that the method works mainly with larger data sets stripping identifying information and removing them from the data snippets. For example, it may replace the names of wearable users with randomly generated strings to hide their identities (Hintze & El Emam, 2018). In either case, the data is still useful because the send and recipient both understand the random string and can use it to categorize and examine the data.

2.7.6. Cryptography

Cryptography has several functions like authentication, encryption, and data protection. Data protection secures wearable data by masking it. It makes it difficult for cyber criminals to access, compromise, or exploit the data. Authentication protects the wearable data from forgery or modification by validating the data exchange devices (Senthil & Perumal, 2020). The devices also validate and confirm the identities of their users through authentication codes and digital signatures. Lastly, encryption obscures the wearable data so that it becomes impossible for unauthorized users to read and understand it while in the encrypted format. According to Senthil and Perumal (2020), cryptography defines a key as a piece of information containing scrambled data that appears in random, very large, and contains a mixture of numbers, letters, and symbols. The unencrypted data is called plaintext. Cryptography is divided into public and private key cryptography. Public key encryption is a method of encrypting data using two keys; public and

private keys. Data encrypted using a public can only be decrypted using the matching private key, and encrypted using a private key can only be decrypted using the corresponding public key (Senthil & Perumal, 2020). On the other hand, Private Key cryptography uses only private keys to encrypt and decrypt the data. It is the cryptography approach that this study is based on by exploiting public-key cryptography, as discussed in the next section.

2.8. Public Key Encryption

It is a method of encrypting data using two different keys; public and private keys. Data encrypted using a public can only be decrypted using the matching private key, and encrypted using a private key can only be decrypted using the corresponding public key. According to Knipp et al. (2002), cryptography defines a key as a piece of information containing scrambled data that appears in random, is very large, and contains a mixture of numbers, letters, and symbols. The unencrypted data is called the plaintext (Knipp et al., 2002).

Encryption is the process of converting plaintext to ciphertext. The process includes a key and algorithm. The key value is not dependent on the plaintext. After generating the ciphertext, it is transmitted to the receiver. The security of any conventional encryption relies on two factors; the secrecy of the key and the encryption algorithm. The algorithm used generates different outputs depending on the particular key that is being used at the time. After receiving the ciphertext, it can be converted back to the initial plaintext in a decryption process (Knipp et al., 2002). Public key or asymmetric encryption is a type of cryptosystem where encryption and decryption are initialized using different private keys and public keys. The private key is a secret key known only by a single user, while the public key is known to everyone (Knipp et al., 2002). The keys are provided by the certification authorities to allow uses to encrypt their data and safety share it with another party who will decrypt it, as illustrated in Figure 2

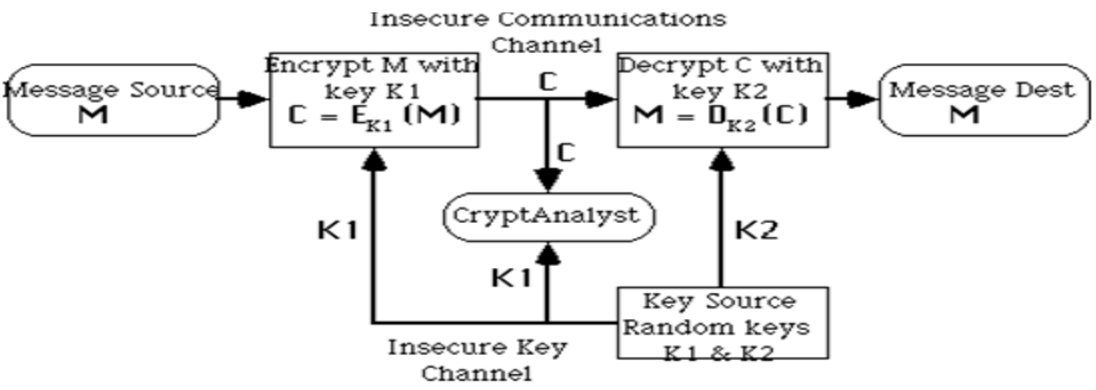


Figure 2.11: Public Key Encryption Process (Source: Knipp et al., 2002)

Table 2.2 summarises the various components of public-key encryption and its functionalities, while 2.3 provides an overview of public-key algorithms' classes.

Table 2.3: Components of Public Key Encryption (Source: Knipp et al., 2002)

Component	Description
Plain Text	The message which is readable or understandable. This message is given to the Encryption algorithm as an input.
Cipher Text	The cipher text is produced as an output of Encryption algorithm. We cannot simply understand this message
Encryption Algorithm	The encryption algorithm is used to convert plain text into cipher text
Decryption Algorithm	It accepts the cipher text as input and the matching key (Private Key or Public key) and produces the original plain text
Public and Private Key	One key either Private key (Secret key) or Public Key (known to everyone) is used for encryption and other is used for decryption

Table 2.4: Classes of Public Key Algorithms (Source: Knipp et al., 2002)

Class	Details
Public-Key Distribution Schemes (PKDS)	Used to securely exchange a single piece of information Value depends on the two parties, but cannot be set Value is normally used as a session key for a private-key scheme
Public Key Encryption (PKE)	Used to encrypt any arbitrary message Anyone can use the public-key to encrypt a message Owner uses the private-key to decrypt the messages Any public-key encryption scheme can be used as a PKDS by using the session key as the message Many public-key encryption schemes are also signature schemes (provided encryption & decryption can be done in either order)
Signature Schemes	Used to create a digital signature for some message Owner uses private-key to **signs** (create) the signature Anyone can use the public-key to verify the signature

2.8.1. RSA Algorithm (Rivest-Shamir-Adleman)

RSA is a cryptographic algorithm that uses used secure sensitive data, especially when the data is being sent over an insecure network. The algorithm was first described in 1977 by Leonard Adleman, Adi Shamir, and Ron Rivest. RSA uses public and private keys to encrypt and decrypt the transmitted message (Bellare, Boldyreva, Desai, & Pointcheval, 2001). It is the most widely used asymmetric algorithm because it assures confidentiality, integrity, authenticity, and non-repudiation of data. The security of RSA depends on the computational difficulties of factoring large integers. The encryption strength depends on the key size, and thus, doubling the key size can result in an exponential increase in the encryption strength. RSA keys are typically 1024 or 2048-bits long (Bellare et al., 2001). Experts argue that 1024-bit long keys have become insecure against attacks.

2.8.2. Diffie-Hellman Public-Key

It is the first public-key scheme ever proposed by Whitfield Diffie and Martin Hellman in 1976 (Ahmed et al., 2012). It is used to exchange cryptographic keys over a public communication medium securely. Unlike in RSA, in Diffie-Hellman Public-Key, keys are not

exchange but are joined derived. Thus, it uses temporary public keys. One can verify the authenticity of the server's temporary key by examining the signature on the key used. Usually, since the public keys are impermanent, a compromise of the server's long-term signing key cannot endanger the privacy of any of the past sessions. As a result, it is commonly known as Perfect Forward Secrecy (PFS) (Ahmed et al., 2012). Since Diffie-Hellman uses a set of private-public keys, it is an asymmetric algorithm that is used to create a shared secrete for a symmetric key algorithm. Although it is still used today, certain precautions must be taken in the building blocks.

2.8.3. SHA-512 Algorithm

It is a hashing algorithm that is used to perform the function on a given data. The algorithm is mostly used in internal security, block chain, and digital security. It is based on 64-bit unsigned integers. JavaScript does not support the integers since they are more complex (Khaishangi, 2019). Hash functions take data input and generate the output, which is called a hash digest. The hash digest has a fixed length depending on the input data. The output must meet certain conditions. First, it must have uniform distribution to ensure that it obtains a fixed length. Next, the input must have collision resistance. It is not feasible to have two different inputs to the hash function have the same hash digest (Khaishangi, 2019).

SHA-512 operation stages include formatting the input, initialization of the hash buffer, message processing, and output generation. The message input cannot be any size because SHA-512 has a size limit. Thus, the entire message has to be formatted to include three sections: initial message, padding bits, and size of the initial message. When combined, the whole message should be 1024 bits because it was processed as blocks of 1024 bits Sumagita, Riadi, & Warungboto, 2018). Thus, each block must be 1024. Once the message is taken, some padding bits are appended to it to obtain the required length. The paddings bits are zeros with a leading one (1000…000). After that, the initial message given to the algorithm is also appended, and the size value must be represented in 128 bits. This is the reason SHA-512 has an input size limitation. The message is processed by taking one block of 1024 bits at a time (Sumagita, Riadi, & Warungboto, 2018). Continuous processing takes the 1024 bit block and the outcome of the previous message processing. Several rounds might be performed during the process, as shown in Figure 4. After the phases are completed, the final 512 bit Hash value is obtained from the initial message.

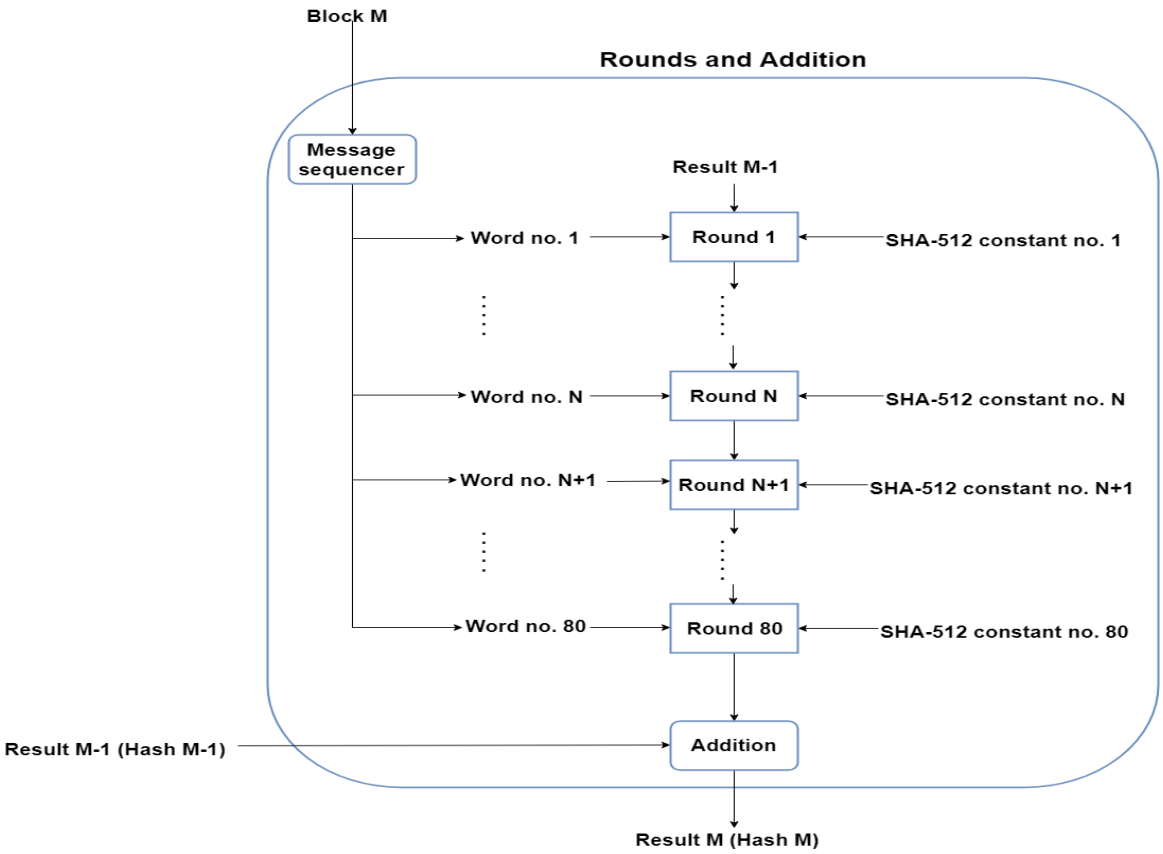


Figure 2.12: SHA-512 Iteration of Message Processing

2.8.4. Elliptic-Curve Cryptography (ECC)

Elliptic-Curve Cryptography (ECC) is considered a modern family of public-key cryptography. It is based on the algebraic structures of the elliptic curves over finite fields and on the difficulty of the Elliptic Curve Discrete Logarithm Problem (ECDLP). The algorithm is designed to implement all major public key/asymmetric cryptosystems capabilities, namely key exchange, encryption, and signatures. "The ECC cryptography is considered a natural modern successor of the RSA cryptosystem because ECC uses smaller keys and signatures than RSA for the same level of security and provides very fast key generation, fast key agreement, and fast signatures" (Hankerson, Menezes, & Vanstone, 2006).

Elliptic-curve cryptography (ECC) offers several algorithms based on the math of the elliptic curves over finite fields. All the algorithms utilize a curve behind (like secp256k1, curve25519, or p521) for the calculations. They also depend on the difficulty of the elliptic curve discrete logarithm problem (ECDLP). Additionally, the algorithms use public or private key pairs. Here, the private key exists as an integer, while the public key exists as a point on the

elliptic curve (EC point) (Hankerson, Menezes, & Vanstone, 2006). The process of key generation in ECC cryptography is simple and secure. Usually, any number within the curve's field size range, normally 256-bit integers, is a valid ECC private key. The algorithms are as follows;

i. ECC digital signature algorithms like ECDSA (for classical curves) and EdDSA (for twisted Edwards curves).
ii. ECC encryption algorithms and hybrid encryption schemes like the ECIES integrated encryption scheme and EEECC (EC-based ElGamal) (Hankerson, Menezes, & Vanstone, 2006).
iii. ECC key agreement algorithms like ECDH, X25519, and FHMQV.

Figure 2.13 gives an elliptical representation of Improved ECC.

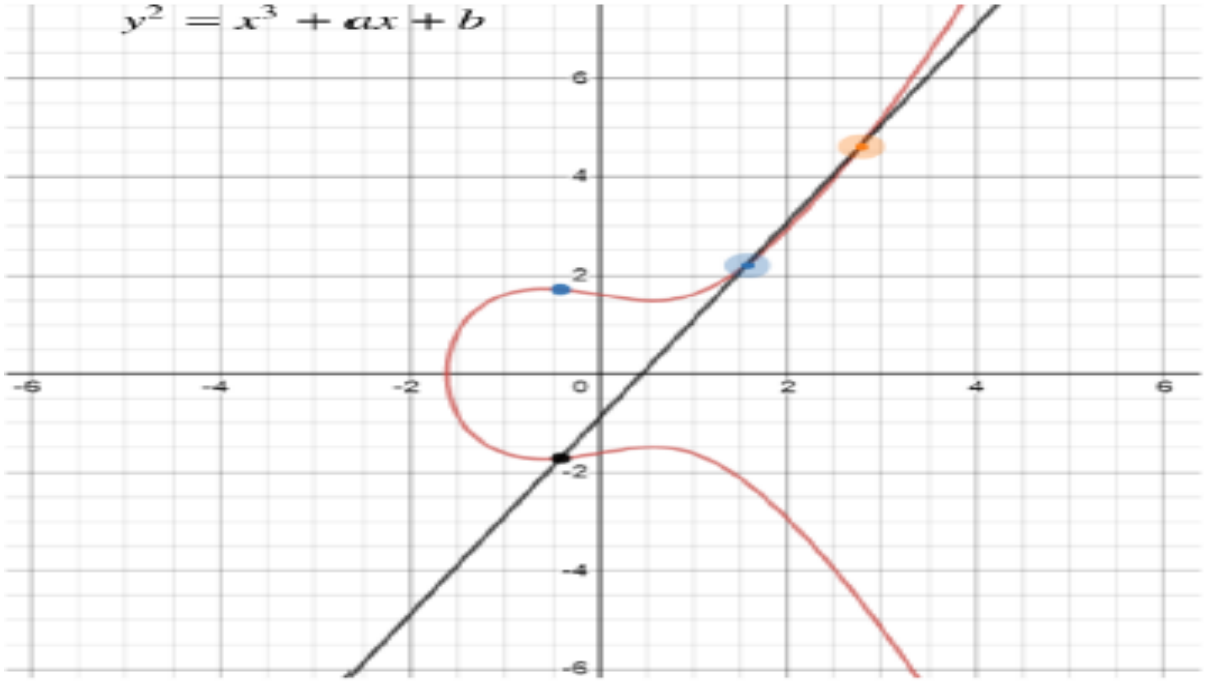


Figure 2.13: IECC Elliptic Curve

Equations (1) to (6) provide the mathematical illustrations of the ECC.

Consider the curve equation

$Y^2 = X^3 + ax + b$ ----------*eq (1)*

(a and b are integers)

The encryption reliability depends on the technique employed in key generation during the cryptographic process. As stated before, three different keys must be generated in the implemented system. The first key is the public key which is generated for data encryption. After that, a private key is generated for data decryption. Finally, a secret key is generated from the public, private, and pints on the elliptic curve. Let B_s be the curve's base point. If you select a random number between 0 and n-1, it can be used to generate the private key Pr_k. The corresponding public key Pu_k will be generated using equation (2).

$$Pu_k = Pr_k * B_s \quad \text{----------eq (2)}$$

This can be rewritten into

$$Pu_k = \pi(Pr_k, B_s) \quad \text{----------eq (3)}$$

To generate the secret key S_k, private key (Pr_k), public key (Pu_k), and point on the elliptical curve (B_s) are added as indicated in equation (4).

$$S_k = \sum (Pr_k, Pu_k, B_s) \quad \text{----------eq (4)}$$

Once the secret is obtained, it becomes easy to encrypt the data from the wearable sensor. The encrypted data has mathematically expressed ciphertexts, as shown in equations (5) and (6).

$$C_1 = (S_1 * B_s) + S_k \quad \text{----------eq (5)}$$

$$C_2 + M + (S_1 * Pu_k) + S_k \quad \text{----------eq (6)}$$

Where S_1 is the random number between 1 and n-1 and C_1 and C_2 are the ciphertext

Decryption gives the original information. In this case, the secret key obtained during decryption is substituted from the normal equation as indicated in equation (7).

$$M = ((C_2 - Pr_k) * C_1) - S_k \quad \text{----------eq (7)}$$

2.9. Conclusion

The discussion above shows that wearable technologies have not fully exploited security mechanisms for data, especially those in transit. Using either proprietary systems or third-party systems, the devices can collect a wide range of data from their users using their inbuilt sensors. The chapter also shows that different wearables have different data collection and transfer techniques as part of the discussion. Since this data is critical for the user's health, it is necessary to protect it from unauthorized access. Public key encryption thus presents an excellent framework to secure wearable data because it reduces the chances of cyber criminals discovering or learning a person's secret key during data transmission. Public key cryptography is the most secure protocol compared to private key cryptography. Users are not obligated to transmit or reveal their private keys to anyone, which minimizes the probability of cybercriminals discovering a person’s secret key during data transmission. From the three public key algorithms discussed, the SHA-512 algorithm and Elliptic-Curve Cryptography (ECC) will be suitable for developing the proposed framework because it has smaller keys and can be computed substantially faster.

CHAPTER THREE: RESEARCH METHODOLOGY

3.1. Introduction

Research methodology is a set of procedures applied to identify, select, and analyze data related to a specific topic. In a research paper, research methodology allows the reader to effectively analyze the study's overall reliability and validity, especially the sections that provide a systematic approach the study applied to answer the research questions. Thus, this chapter will focus on the methodology that will be used to carry out the overall research. The chapter includes several quantitative and qualitative methods that will be used to gather data about usage about the status of data transit and security in wearables. The chapter explains the application of experiments to demonstrate a prototype that was developed using low-cost electronic components and can remotely be used to solve the research questions. Additionally, it gives details about the secondary sources on various technologies used or documented in this study.

3.2. Research Approach for Objectives 1 and 2

This phase of the study focused on addressing the first two research objectives. The first objective seeks to determine the technologies that support patient data sharing in the wearable device by analyzing patient data transmission from optical sensor to listening device. A literature review of wearable devices was used to address this question. The study analyzed the manufacturer's manual of the devices to understand their system architecture and general operation. The study also evaluated the datasheet for each sensor to determine pin configuration. The study identified the most appropriate method to reconfigure and reprogram the devices to provide new encryption that can offer authentication, confidentiality, and data integrity. Lastly, an online questionnaire was sent to various manufactures and users of the wearables to understand the efficiency that is currently used. These two also served as the sample population due to their ease of accessibility. The questionnaire was sent to least five manufacturers and ten users and contained a set of recorded personal information, highlighting how long they had used the wearables and the manufacturers' statement on security concerns during data transfer from the wearables.

The second research question reviews techniques used to abstract data to enhance security during data transmission in wearable devices. Similarly, a literature review of the existing techniques

was employed to identify and analyze the existing security protocols already in use to secure data during transmission. In the literature review above, some research approaches identified are secure file transfer protocol, secure sockets layer certificates, private networks and VPNs, and Secure Shell (SSH) connection. Through this, the identified the limitations of the existing solutions, which validated the need for the proposed frameworks

3.3. Research Approach for Objectives 3 and 4

To achieve the last two objectives, an encryption tool for enhancing integrity, confidentiality, and authenticity of the data during transfer in the wearable devices was designed, developed, and validated to test the effectiveness of the proposed solution to the problem statement. The process was accomplished by implementing the Agile Software methodology process. It is an approach based on iterative processes. The method breaks down a given task into smaller iterations, and it does not include long-term planning. In this method, the researcher lays down the project scope and requirements at the start of the development process. The study also lays down the plan about the number of iterations that will take place during the design process. In agile development, each part of the design development is a short time frame that should last for one to four weeks. Dividing the whole project into smaller parts helps the developer to reduce project risks and, at the same time, ensure the project takes the minimal time possible.

Thus, the method was selected for the study because it provides a realistic methodology to software development, promotes cross-training and teamwork, is possible to develop and demonstrate the functionality, and minimizes resources requirements. The iterations involve planning, requirement analysis, design, programming, testing, and validation.

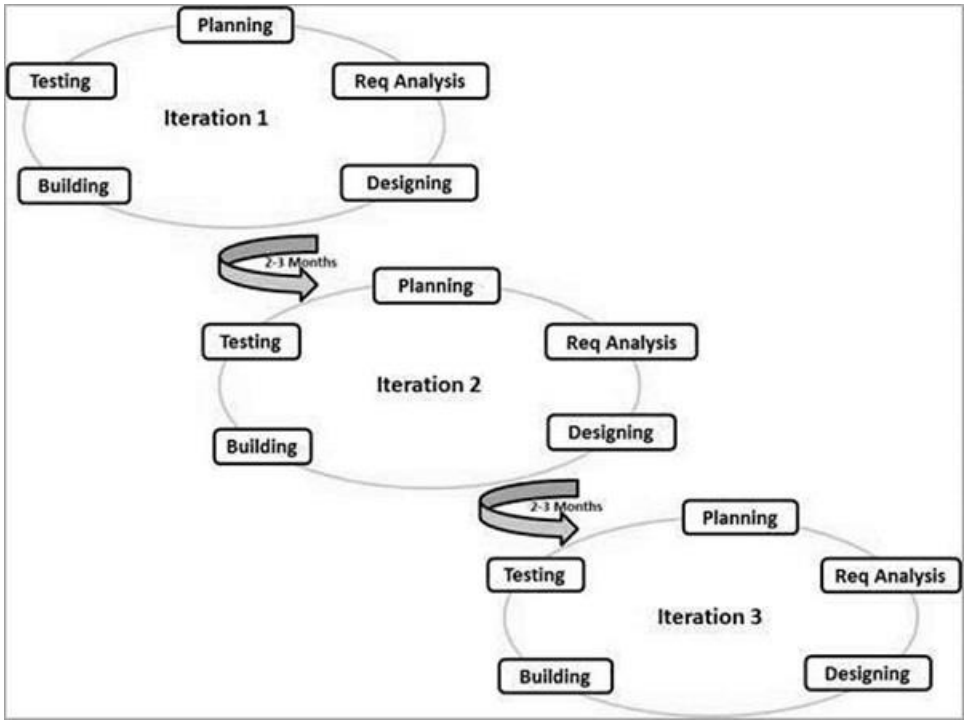


Figure 3.1: Agile Software Development

3.3.1. Planning

The planning was divided into project initiation, implementation planning, risk analysis, final reviews. First, at the project initiation, there was a preliminary review of the proposal. This was followed by consultative meetings with the supervisor, technical advisors, and other stakeholders who may benefit from the study's final output. Consultations are necessary to receive analytical reviews and suggestions which can be included in the framework design. The initiation planning helped outline the phases of the study and the roles of each individual or company that will be identified as a participant in the study. The study consulted through discussions with the different stakeholders to determine the views on the study's success by building a backlog.

Next, in the implementation planning, the project identified the cost and sources of the proposed implementation tools. As described under the implementation stage, the tools included both software and hardware. Through planning, it was possible to identify and contact local dealers for purchases. The next phase of the planning was a risk analysis, where the study also estimated the risks and development milestones for every phase of the research. Through risk analysis and literature reviews, the study identified risk mitigation strategies to avoid delays during the design and implementation of the proposed framework. Finally, the planning was completed through a

final review that involves counterchecking the system requirements to ensure everything required for design, implementation, tests, and validations were available or could be accessed.

3.3.2 Requirement Gathering and Analysis

The phase was used to gather the requirements that were necessary to develop the framework. It involved identifying the requirements for the design and development of the encryption tool based on the analysis of the various gaps identified after the analysis of the existing solutions. The outcome of this stage determined or defined the necessary technical approaches that were followed to develop the tool. Moreover, successive iterative changes were made to the tool requirements to ensure the objectives were achieved. Requirements gathering was performed through literature review, observations, application review, and questionnaire. For starters, the literature review focused on the IoT framework to understand data transmission from the sensor to the cloud and methods of encryption that could be applied. It was achieved by analyzing both public and private key encryption and methods of system development. The idea was to understand the operation of various encryption algorithms and determine how the platform could be built and operated. Thus, the literature review facilitated the development process and helped understand security solutions in the encryption environment. Application reviews and observation were achieved by analyzing various encryption tools' documentation to understand their development and operation. Some of the tools considered were AppLock, Samsung Knox, Mobilflage, and CryAll. Finally, questionnaires were used to determine user habits and how the proposed solutions could be designed to meet their tastes and preferences. A questionnaire was created, and the responses received were analyzed using google form tools, graphs and charts as presented in Appendix 3. Other elements that were included during requirement gathering and analysis included;

3.3.3. Design

3.3.3.1. Research Design

The research employed a combination of descriptive and experimental research designs. For starters, the descriptive methodology was used to provide answers to where what, who, and how questions as directly linked with the research questions. The approach assisted the study in evaluating the research problem's current status with respect to the study objectives (Sacred

Heart University Library, n.d.) On the other hand, the experimental research methodology was employed as a blueprint of the whole study. It enabled the study to maintain control of the problem statement and propose viable solutions using information security culture. Since consistency is needed during the study, the experimental research method became very appropriate. Thus, the researchers ensured control, manipulation, and randomization to ensure true experiments are conducted during the study.

Location and Target Population

It was an in-house study that was conducted in the IoT lab at @iLab Africa Strathmore University. IoT Lab is a research Unit of @iLab Africa which is a research center in Strathmore University. IoT Lab is involved in the research and development of smart, automated solutions which can be applied in various sectors of the economy. It also offers training and consultancy services that allow businesses to adopt and utilize IoT Technology. Some of the previous research and projects the lab has worked on under the leadership of Manager Leonard Mabele include Precision Farming and Livestock Monitoring, Air Quality Monitoring System, TV White Space, and Disaster Prediction and Management Project. Thus, based on its pre-existing research and development activities, the IoT lab provides enough resources like sensors, microcontrollers, and technical expertise (consultants) to study various wearable devices currently being used to monitor heart rates.

The research was based on the medical recommendations from Strathmore Health Center. During analysis of the wearables, it was necessary to get information that would help explain the gaps which data exploiters commonly exploit while undertaking an attack. Strathmore Health Center was contacted to help in data interpretation from the wearables so as to understand the possible methods of attacks cybercriminals can use against data in transit. The study also identified students and other professionals within the university who have used the wearables to share their experiences with various types of wearables currently in use. As a prerequisite to understanding the wearables in terms of their cost, operation, user preference, and daily usage, among other factors in real-life scenarios, the study aimed to question at least fifty participants during data collection about the wearables. The participants that are expected to be included wearable users (students and lecturers), doctors, and data analysts. Wearable manufacturers

whose devices were analyzed and studied during the investigation included FitBit, JawBone, Samsung, Microsoft, and Google, as they are the most commonly used in the market today. Only twenty-eight responded to the questionnaire, as shown in Appendix 2.

Sampling Techniques and Sample Size

The study utilized probabilistic sampling, where the researcher used randomization to ensure that all elements of the population (manufacturers, users who include students and lecturers, doctors, and data analysts were given equal chances of being part of the selected sample. The stratified sampling technique was used because it offered a better coverage of the population. The technique involved dividing the elements of the targeted population into sub-groups, categorizing them as either manufacturers or users. The study included a total of five different types of wearables to understand technologies used and methods of data transmission and abstraction in different devices. Through this, it estimated that at least fifty people were to be interviewed during data collection because the data generated was enough to gather enough information needed for analysis (Sigh, 2018). The respondents were obtained through online research panels. First, the respondents that were needed for the study were identified and classified as shown in Table 3.1

Table 3.1: Summary of Respondent Classification

	Class	Name	Number of Respondents Obtained
1	Manufacturer	Manufacturer	1
2	Within Strathmore	Students	1
		Lecturers	1
3	Specialist 1	Data analyst	1
4	Outside Strathmore	Other Users	27
5	Specialist 2	Doctors	1

After that, an online research panel was created, especially since the research was conducted online due to the prevailing pandemic at the time of this study. An email requesting their participation was sent to a series of people. A snippet of the email confirmation from the respondents is shown in Figure 3 below.

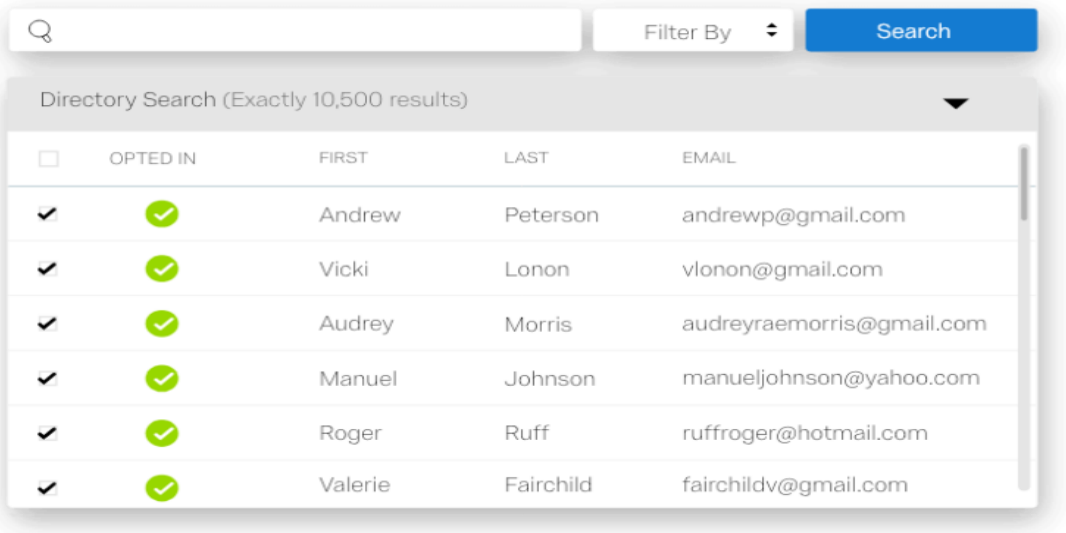


Figure 3.2: A Snippet of Email Confirmation from Respondents

Data Collection/Research Instruments

Both primary and secondary data were collected during the study. The data collection took three to four months. The study applied the primary data collection that included both quantitative and qualitative research methods. Online questionnaires were the most used as they were efficient, easy to use, and was more convenient given that the study was conducted during the covid-19 pandemic and physical contacts were limited. Secondary data sources included journals, books, newspapers, and websites, which were retrieved from a literature review. A summary of the data obtained is provided in Table 3.2 below.

Table 3.2: Data Summary

Used Wearables	User Experience	Data Transfer	User Concerns
The most used wearables are wearable health and fitness (67.9%), wearable smartphone (28.6%), smart clothing (14.3%), GPS tracker (25%), Smart glasses (17.9%), and Smart Watch (3.6%).	• Users recorded good experience with the wearables because the devices effectively recorded their; ❖ Interaction with other physical devices ❖ Activities ❖ Environment	Bluetooth (25%), Wi-Fi (45%), and Celluar (30%)	• Unsecure data transmission via Bluetooth to local storage devices • Software communication to the cloud via Wi-Fi network and cellular • Authorization and Authentication • Most users do not use physical controls • Price • Relevance • Size

Data Analysis

The study applied the diagnostic analysis method. The purpose was to dive deeper during the research and find out the major causes of the research problem. The study used a diagnostic analysis to create a better connection between data. This way, it was easy to identify the pattern of behavior and isolate possible causes of data breaches during transit. It created detailed information such that when the problem arose, which was related to the problem under study, it was likely that some data had been collected about the same (Friese, 2019). Thus, by applying this analysis approach, the study most likely covered future loops that are related or may arise from the research problem. The data analysis tools that the study used were Matlab, java, python, and SQL. There were six phases in the data analysis process; the gathering of data requirement, collection of data, cleaning of data, analysis of data, interpretation of data, and visualization of

data (Guru99, 2019). By the end of the analysis, it was expected that the information collected should assist in developing the appropriate framework that could be included in the wearables to provide confidentiality, integrity, and authenticity of data during transmission from the wearable sensor to the listening device or portal.

3.3.3.2. System Design

The system employed a combination of visual design and architectural structures in system design. The architecture part involved creating layouts and structures based on sensors, controllers, and other physical devices that the design process will include. On the other hand, visual or graphic design will involve the use of Adobe Illustrator to create the proposed framework. The Adobe Illustrator that was used is mentioned below in the implementation stage. The selected design was suitable for this study because it allowed for other applications like real-time systems, simulation systems, and implementations of systems with dynamic and complex entities, which may be impossible to represent in traditional methods (Borysowish, 2018).

Other design tools that were used will include flowcharts, data flow diagrams, use case diagrams, class diagrams, and wireframes. First, data flow diagrams were used to provide graphical representations of how the data flowed from the sensor to the listening device and finally to the cloud. Using the data flow diagrams, it was possible to depict the income, stored, and outgoing data flow. Secondly, flowcharts were used to give a more detailed representation of the system. Flowcharts break down the system into smaller functional modules used to describe the functions and sub-functions of the various parts of the system, thereby giving more details than data flow diagrams. Lastly, use case diagrams and system sequence diagrams were used to represent the hierarchy of various modules created in the software system. The system sequence diagram was essential as it provided a high-level view of the system and its functions. In other words, it decomposed the functions into sub-functions in a categorized manner so that it was easy to document the role of every function within the system ("Software Analysis and Design Tools," 2020). The sub-sections included context diagrams, partial domain model, database schema, and entity-relationship diagrams. The diagrams assist in understanding the structure of the software by creating a logical view of the whole database. The diagrams expressed the interactions of various external elements within the system so that it was possible to realize the use case. Lastly, they identified the input and output of all the automated uses cases within the system.

3.4.3. System Implementation

The system was implemented using a phased approach. This involved bringing in the new system one step at a time. Some of the tools that were used during this process were flowcharts and prototypes.

Implementation Tools

The prototype constituted both hardware and software parts. The hardware included the fabricated sensors and microcontrollers, while the software included the client's web interface and server endpoints. The programming integrated development environment (IDE) that was used include Arduino IDE and Andriod Studio for programming and Visual Studio Code for software development of the server-side scripts and hardware. The Programming languages used are python and C/C++ to program the nodemcu board. On the server-side, a VPS storage provided by Microsoft Azure Cloud will be utilized. The conceptual modeling was created using draw.io, while the wireframe and architecture design tool was used as Balsamiq. Lastly, the hardware parts of the system included Arduino Microcontroller, a mini-solar panel and 4.5V DC battery, PPG/ECG Combo Wearable Biosensor Module, and connectors for the GPIO pins ("Software Analysis and Design Tools," 2020).

MySQL was utilized as a backend tool during the design and management of the encryption tool. It was used in establishing a connection between the wearable device and the server. MySQL was for development because it is flexible and scalable, making it easy to handle embedded applications and large volumes of data. MySQL also facilitates easy customization of the requirements of the design as per the specific functionalities. The encryption application was hosted on an online Apache HTTP Server because PHP is an Open Source Platform with various security layers, allowing it to prevent malicious attacks ("Software Analysis and Design Tools," 2020).

3.4.4. Testing and Deployment

Testing

The system was tested through end-user demonstrations through a selected group of participants. The parameters of the test were to determine if the system provided security, usability, and

compatibility. For starters, security testing was employed on cloud-based platforms. A non-functional software testing approach was used to determine if the system and data were protected from external interference ("Software testing methodologies," n.d.). The goal of this test was to find loopholes and any existing security risks within the system. The principle of security testing was integrity, availability, confidentiality, non-repudiation, authentication, and authorization. Secondly, usability testing was used to measure the application's ease of use from the end user's perspective. Its goal was to determine if the design met the expected standards. Finally, compatibility testing was used to measure how the proposed ideas or solutions worked in different environments ("Software testing methodologies," n.d.). It helped to check the software and hardware functionality of the whole process.

Deployment

The system was deployed within the IoT laboratory at @iLab Africa, located in Student Center at Strathmore University. It adopted the source code control deployment method. This method used the source code control server to maintain a centralized master copy of the deployment image. Test stations or clients could sync up with the source code control server to use or test the framework. The source code control server was network-based which means it was possible to download the deployment images from the local copies to other computers should there be any network failure. The method also allowed for easy software upgrades because it was connected to the source code control server. Clients could easily switch to new or older versions depending on their wanted utilities ("Software Deployment Strategies," n.d.).

3.5. Validation

The goal was to prove that the system provides confidentiality, integrity, and authentication. Validation would be conducted in two phases. The first stage was content validation that was conducted through a questionnaire designed to explore whether the prototype meets the research objectives (Heale & Twycross, 2015). The next method validation was conducted by sharing the developed framework with five people in the IoT laboratory to share their experience if it provides the three parameters confidentiality, integrity, and authentication. The validation was done at the IoT laboratory at @iLab Africa, located in Student Center at Strathmore University.

3.6. Research Quality Aspects

The quality of the study was ensured through reliability and validity. Validity refers to "the appropriateness of the inferences made about the results of an assessment. Inferences being conclusions derived from empirical evidence bearing on score meaning" (Bruin, 2010). Reliability, on the other hand, shows consistency and replicability over a period. Moreover, reliability is seen as "the degree to which a test is free from measurement errors since the more measurement errors occur, the less reliable the test" (Bruin, 2010).

Reliability was ensured by ensuring applying the methodology appropriately. The researcher planned the methods systematically so that each phase of the study proceeded with limited or without difficulties. Secondly, reliability was achieved by standardizing the conditions of the research (Middleton, 2019). For instance, the researcher ensured they minimized the number and rate of external influences during data collection because these factors could create variations in the final results.

In contrast, validity was ensured by choosing suitable methods of measurement. The researchers ensured the methods and measurement techniques are high quality, and they targeted to measure only what is necessary for the study. The researcher was also conducted thorough research based on research objectives to assist them in building on the existing knowledge. Secondly, validity was achieved by using suitable sampling methods based on the needs of the study (Paul, Rajiv, Dana, & Carrie, 2019). As proposed above, the stratified sampling technique was applied to help the study achieve all the research objectives. The researchers ensured they clearly defined their population in terms of range, geographical location, and profession.

3.7. Ethical Considerations and Approval

For starters, the study required institutional approval from Strathmore University since it was in-house research. The study was conducted within the IoT laboratory at @iLab Africa, located in Student Center at Strathmore University. Institutional approval was necessary to certify the study and the results that were obtained. Strathmore University is an already accredited university that can offer a certificate of ethical clearance of the research.

The important ethical issues during the studies were confidentiality, validity, voluntary participation and consent, risk of harm, and research methods. Therefore, to ensure that all ethical considerations are taken care of during the study, the researchers;

i. They developed an agreement of trust. It was used between the researcher and the participants. The two parties must give informed and explicit consent to the requirements of the study.
ii. Employed the third ethics principle. This was derived from the Economic and Social Research Council (ESRC), and it states that "The confidentiality of the information supplied by research subjects and the anonymity of respondents must be respected" (Smill, 2003). In cases where confidentiality was limited, anonymity was encouraged.
iii. Ensured safety and security of all participants during the study by providing safety gear, training, or consultancy services if needed.
iv. Followed informed and consent rules. These are guidelines developed and protected in Kenya's laws through National Commission for Science, Technology, and Innovation (NACOSTI).
v. The data collection exercise was subjected to the Data Protection Act of Kenya

CHAPTER FOUR: SYSTEM ANALYSIS, DESIGN, AND ARCHITECTURE

4.1. Introduction

The chapter focuses on the detailed structure of the proposed algorithm under this study. To achieve this, the chapter discusses requirement gathering, technical and framework requirements, including functional and non-functional requirements, the system architecture, system diagrams, and the wireframe developed and used during research and design phases.

4.2. Requirement Gathering and Analysis

It was the second phase of the development life cycle. Requirement gathering is the method used to collect data used to understand the wearables in preparation for the system design and development of the framework. Requirements are blueprints from which everyone involved in the study can work from. Requirement gathering was necessary to prevent delays in developing and testing the final designs (Silhavy, Silhary, & Prokopova, 2011). It is important to mention that the study was conducted during the period of the Covid-19 pandemic, and thus, physical encounters were minimized as much as possible. As Tiwari and Rathore (2017) suggest, the approaches used during this phase included group interviews, questionnaires, analysis of existing documents, and Prototyping.

Group interviews were locally organized and conducted by interviewing members of the IoT Lab, which constituted eight members. The study determined that of the eight members, only two were using a wearable device at the study time. The rest indicated that although they were aware of the devices, they had not used any of them. Xiaomi and Haylou are the two wearable versions that the two members were using. They recorded satisfaction with the devices, indicating that the devices provided accurate heart-rate measures, oxygen levels distance covered per day, and location. They also recorded a lack of satisfaction with the devices' security system and indicated they did not provide any form of authentication or data being transmitted to their mobile application for visualization. Cellular, WiFi, and Bluetooth were the three modes of communication supported by the devices.

Secondly, a questionnaire was developed to collect information from people. Although fifty respondents were targeted, only twenty people and their feedback were recorded as indicated in Appendix 2. The results obtained indicated that the number of professionals (72.7%) exceeds the number of students (27.3%) using wearable devices. At the time of the study, only 56% of the participants owned wearable devices, with 61% owning a smartwatch, 11% fitness devices. 43% of those who owned the devices used them daily, while 4% used them once a week. The respondents also indicated that their purchasing of the devices was highly influenced by priced (56%), data security (28%), and privacy (3%). They also acquired the devices to get information about their health (84%), dietary updates (60%), and exercise information (48%).

Analysis of existing documents through literature review was also used to gather information about the information essential to the study. Secondary data sources like journals, books, newspapers, and websites were used during the process. A summary of the results obtained is provided in Table 3.1 above.

Finally, as Eid (2015) suggests, prototyping was also used as an iterative process to understand the functionalities and operation of the wearables. A sample wearable device was acquired, and its components were studied. The areas of concern were sensor used, power module, communication modules, data storage, and user preferences. It was discovered that the devices used the Max30100 Heart rate sensor, GPS, and GSM. For communication, the devices used Wi-Fi and Bluetooth. The visualization application could be downloaded from Google Play Store. It did not have local data storage and thus, the data collected we lost as soon they were transmitted to the visualization application. Based on its rating (3.4/5) and review on the Google Play store, it was evident that the haylou wearable was widely accepted. The Max30100 sensor allowed the devices to collect heart rate and oxygen levels from the user. The android could be charged from a 240V ac supply. The charger was fitted with a step-down transformer. Based on this information, the study could design and develop a prototype of the same device used as the data source in this study.

4.3. System Analysis

The system requirements give descriptions of what the system will do and its operational constraints. They reflect the user's needs by showing that the proposed framework will secure their data during transfer from the wearable device to storage devices or the cloud. This study gathered system requirements through an online survey and physical implementation at IoT Lab, @iLab Africa (Strathmore University), where a prototype system was developed and implemented. The prototype included both hardware and software.

4.3.1. Functional Requirements

Functional requirements are statements of services that the system will provide and how it will react to various inputs. Basically, functional requirements are information, processes, and interactions of the system. They define the study's desired functionality, including the interactions between the system and the environment (Emoghene & Nonyelum, 2017). In this study, the functional requirements for the data interface included;

i. User registration- it was the first stage for the users. A user could register on web-platform using their Gmail and phone numbers

ii. Login/logout- it allowed the user to access the platform at their convenience

iii. Creation of Cryptography Keys- the functionality enabled the user to create cryptography keys they need to encrypt and decrypt the data

iv. Data views- the functionality allowed the user to view stored data after and before transmission

v. Data encryption and uploading- the functionality allowed the user to add a valid encryption key to the data to encrypt it before uploading it for storage in the cloud.

vi. Data download and decryption- the functionality allowed the user to access the files stored in the system and use the valid key to decrypt them.

vii. System logout- the user was allowed to log out of the system.

4.3.2. Non-Functional Requirements

Non-functional requirements, from the name, are non-functional features that the study uses to address technical and operational requirements. According to Angel and MesiaDhas (2017), they are not directly concerned with the design's particular services to its users. Rather, they relate to emerging characteristics like store occupancy, response time, hosting, environment, business continuity, disaster recovery, and reliability. Besides, they can define some constraints the study experiences during system implementation, like capabilities of sensors, input/output devices, and data representation interfaces (Angel & MesiaDhas, 2017). The non-functional requirements covered the whole ecosystem in this study, from hardware, connectivity, and user interface. They included;

i. Usability- the system should be easy to use/user-friendly.

ii. Performance and reliability- the encryption framework deployed guarantees secure and continuous data transfer from the wearable device

iii. Compatibility- the system should run in various android-based operating systems

iv. Stability- the system should allow the addition of other or new features without affecting its performance, services, and other functionalities.

4.4. System Architecture

This section shows the relationship between the various components used to develop the system. It was concerned with understanding how the system was organized to make it easy to design the overall system. The architecture provided an important link between design and the engineering requirements by identifying the system's primary structural components and how they related to each (Tiwari & Rathore, 2017). The architecture thus acted as a conceptual representation that was used to describe the behavior, structure, and views of the whole system. Since the system was developed using a three-tier architecture. It consisted of device, application, and cloud layers, as shown in Figure 4.1.

The device layer- It acts as the client layer. The user interacts with the system using the device layer. The encryption application is designed to allow the user to perform various functions like login, logout, key generation, encryption, and decryption.

The encryption application layer acts as the middle tier between the device (client) layer and the cloud server. The application has all the logistics that implement all data processing and access rules. In particular, the application implements data encryption and decryption which begins by key generation and then proceeds to encrypt the data before it is transmitted. Upon download, the data can then be decrypted using the key generated.

Database level- It acts as the resource manager as it stores the encrypted or unencrypted data. The application layer protects the database from direct access by the clients. The layer comprises of MySQL relational database, which was protected from direct user access.

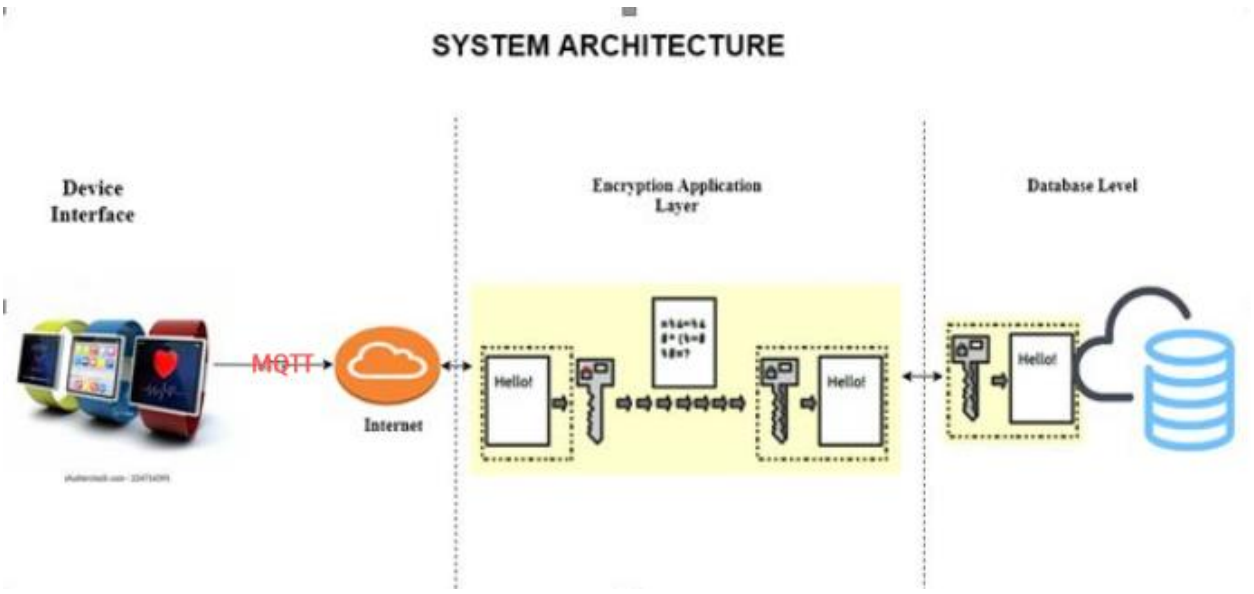


Figure 4.1: System Architecture

4.5. Device Prototype

A prototype of wearable devices was created using the circuit diagram represented in Figure 4.2 below. It was used to provide live heart-rate data, which was encrypted using the system developed. DipTrace software was used to design and implement the circuit below.

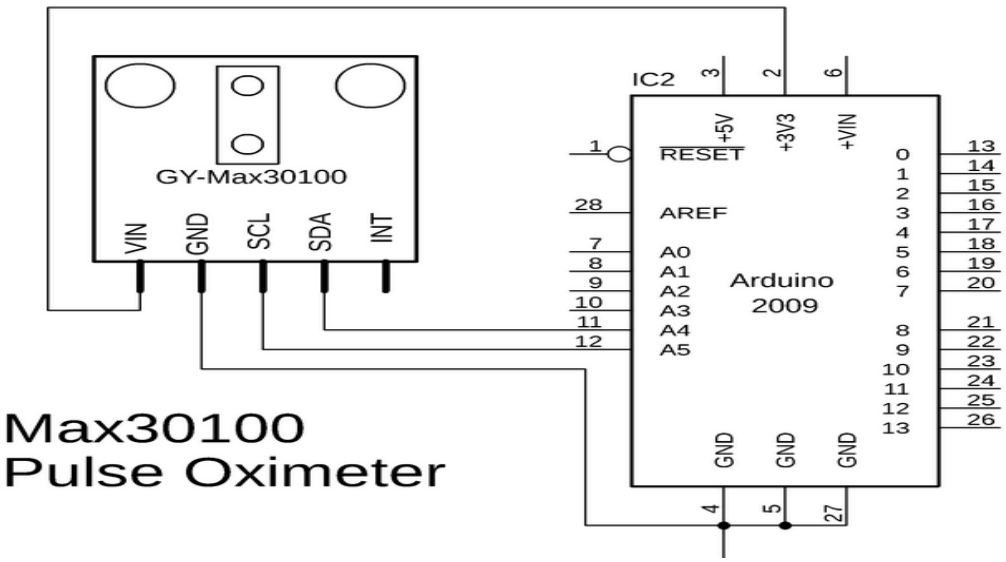


Figure 4.2: Circuit Diagram

4.6. System Designs

4.5.1. System Diagrams

The system diagrams were used to model the conceptual structure and understand the behavior of the encryption tool. The diagrams thus give a logical representation of the system. They also give detailed insights into the actual implementation of the system. Based on the Agile Methodology, the system diagrams that were considered were a use case diagram, class diagram, system sequence diagram, data flow diagram (DFD), and wireframes.

4.5.1.1. Use Case Diagram

The system was broken down to develop processes and data models for the networked sensor module by systematically outlining a set of activities (use cases). The final use case diagram is shown in Figure 4.2. The user is the primary actor in the system.

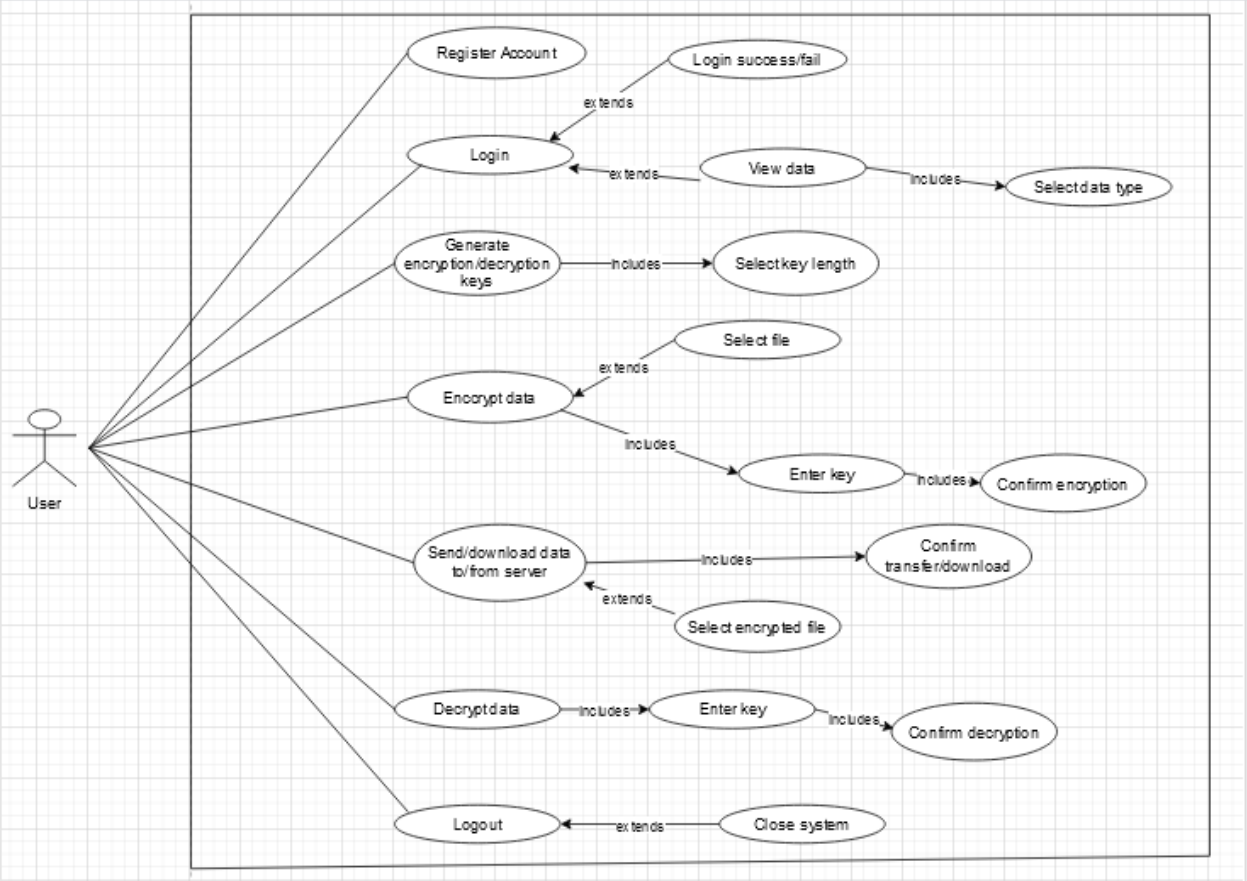


Figure 4.3: Use Case Diagram for the Wearable Device

Use Case Descriptions

i. Send data- The use case describes the process of moving data from the wearable to cloud

ii. Login- The ability of the user or administrator to access the system using their credentials

iii. Register user- The use case is an administrative role of registering new users and giving them access to the system.

iv. Encrypt or decrypt data- secure data (plaintext to ciphertext) using encryption keys

v. Generate key- the use case involves the user using the system to generate encryption keys needed to encrypt or decrypt the ciphertext.

vi. View data- The use case allows the user to view data stored in the wearable device through the system

4.5.1.2. Sequence Diagram

As the name suggests, a system sequence diagram shows all the events initiated by actors from outside the system. In other words, the diagram shows the interactions of individual actors to the system. The diagram was used to represent the progression of events for specific objects identified while developing the partial domain model.

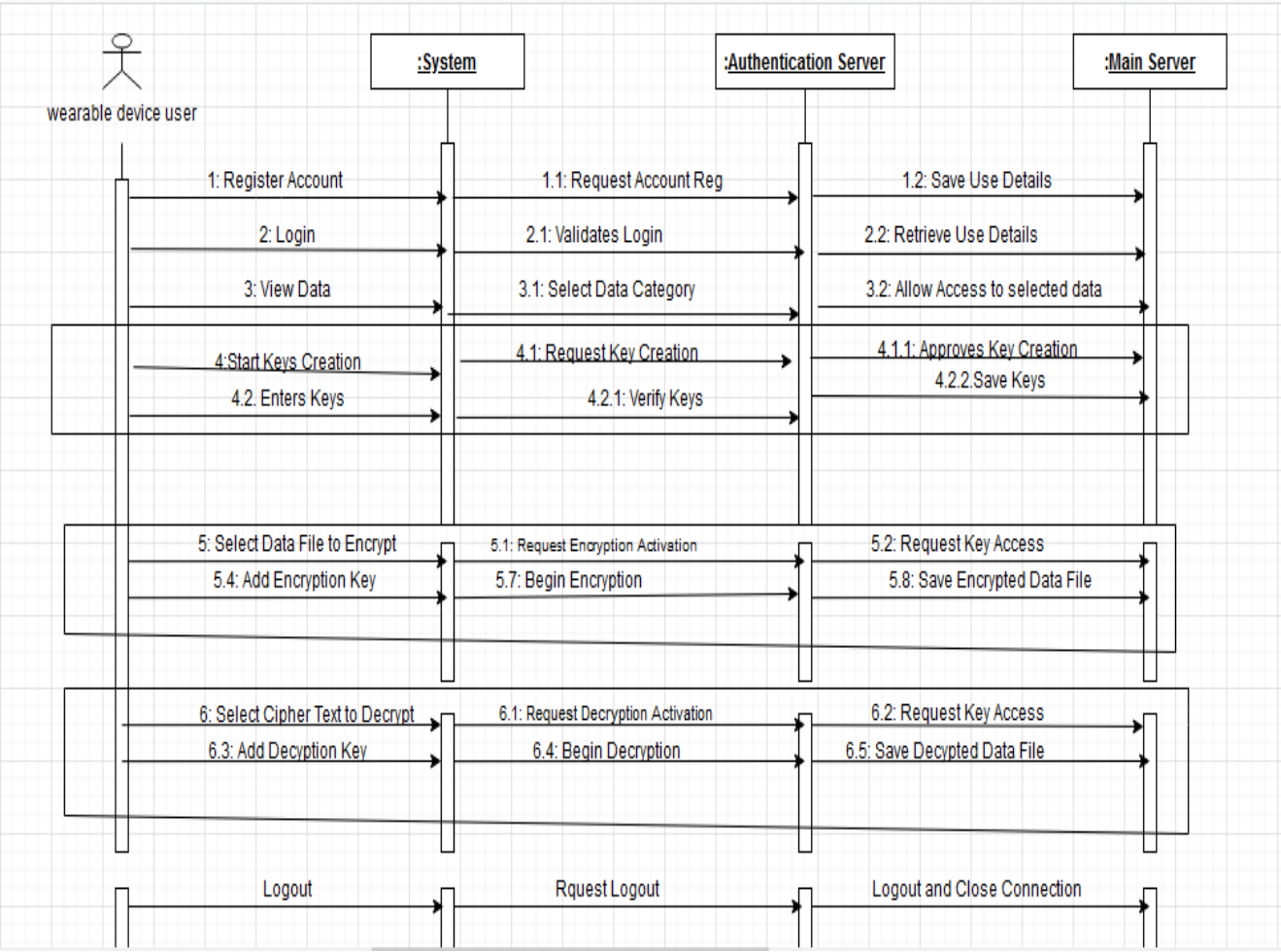


Figure 4.4: System Sequence Diagram

4.5.1.3. Class Diagram

As a building block in object-oriented modeling, a class diagram is used to conceptual model the application structure. It provides detailed modeling, which translates models into programming code. It can also be used for data modeling. Figure 4.3 provides a class diagram for the cryptographic tool that was developed in this study.

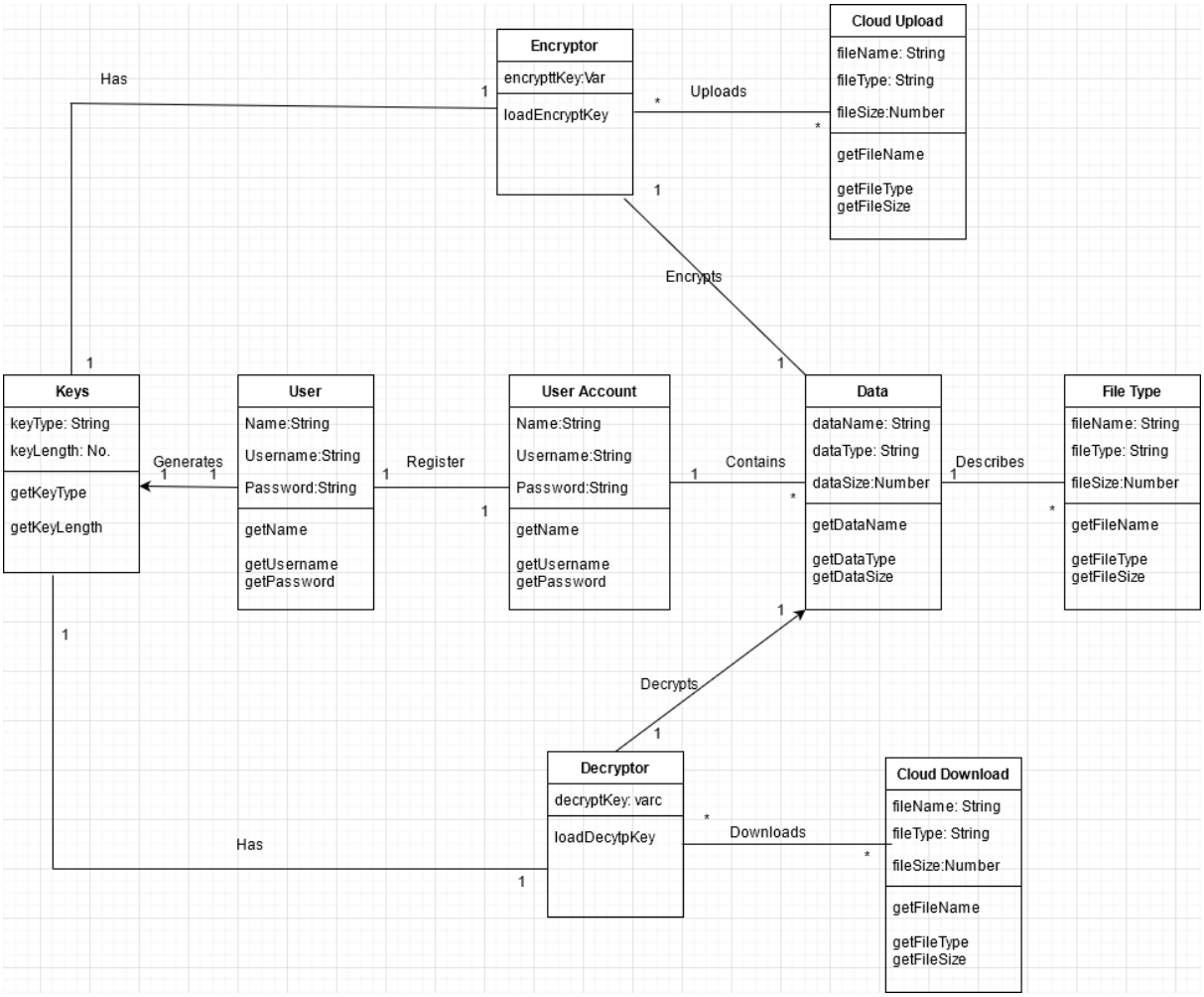


Figure 4.5: Class Diagram

4.5.1.4. Data Flow Diagram (DFD)

As the name suggests, a data flow diagram (DFD) shows the way data flows through a system. It categorizes the input and output data, storage, and other processes that the system performs. A data flow diagram is created using symbols and notations, which are used to describe the different entities within the system and their relationship with each other. Figure 4.4 gives the data flow diagram that was created during this study.

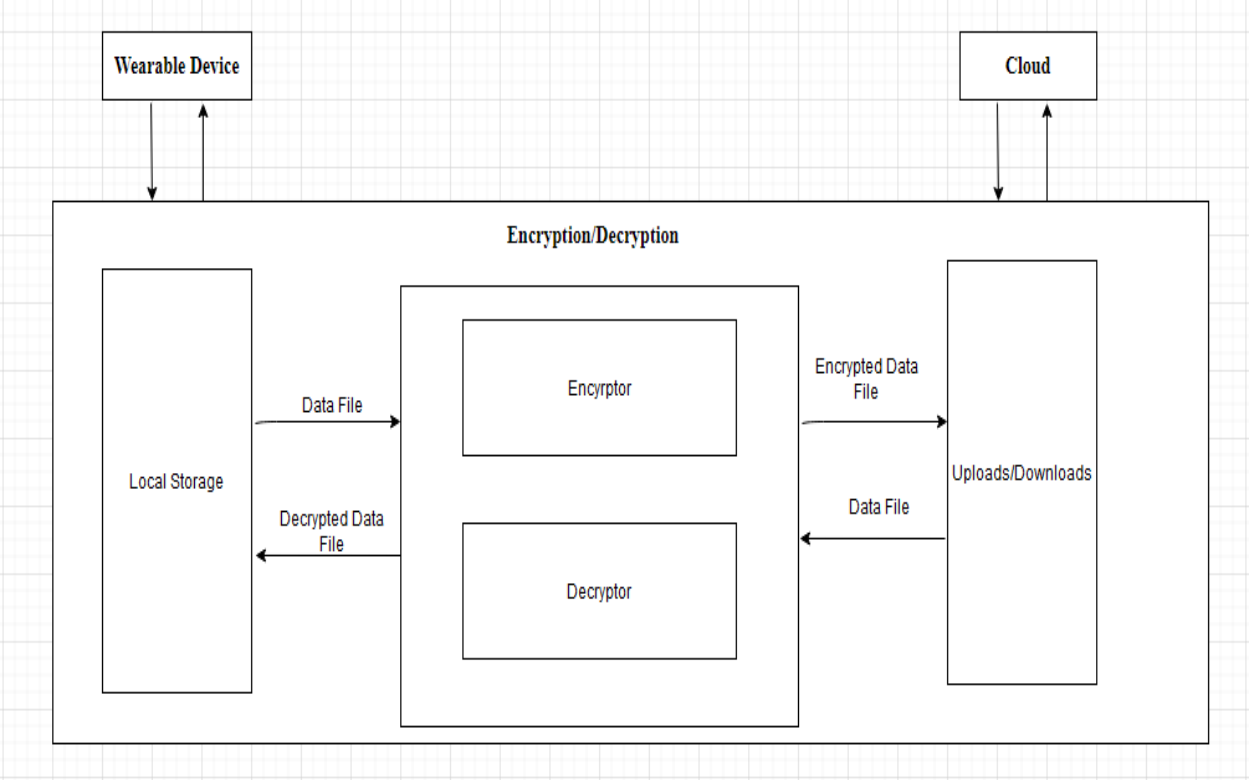


Figure 4.6: Data Flow Diagram

4.5.1.5. Wireframes

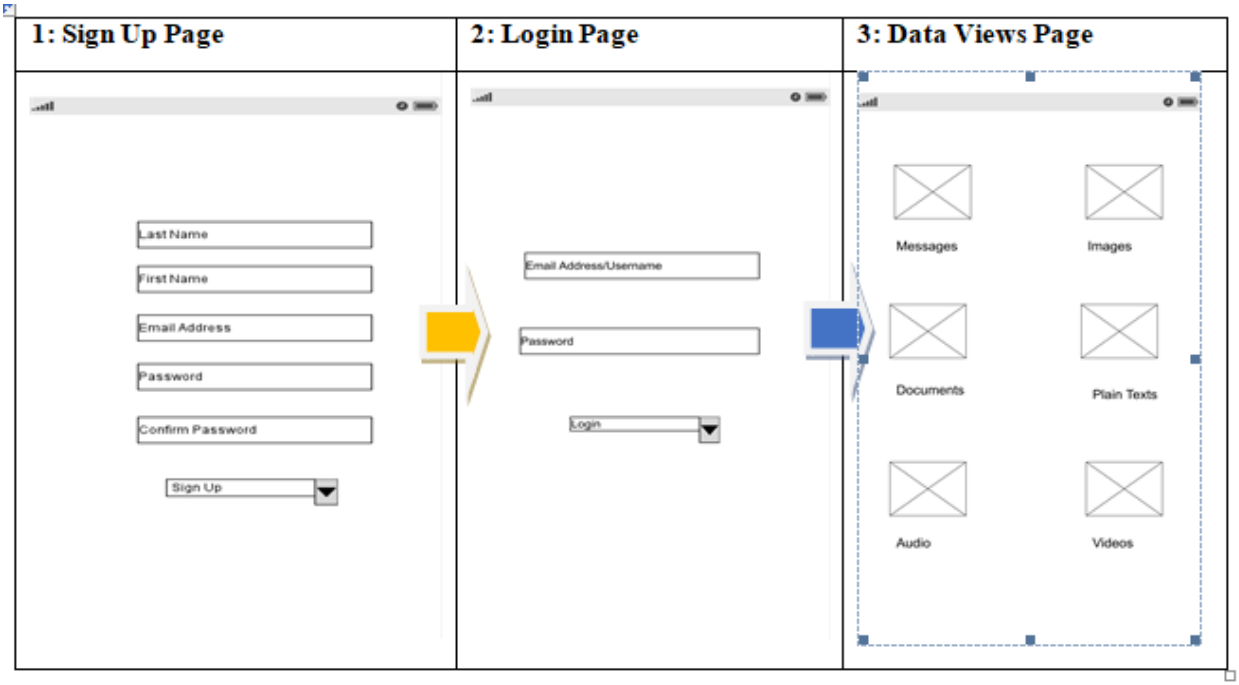


Figure 4.7: Wireframes

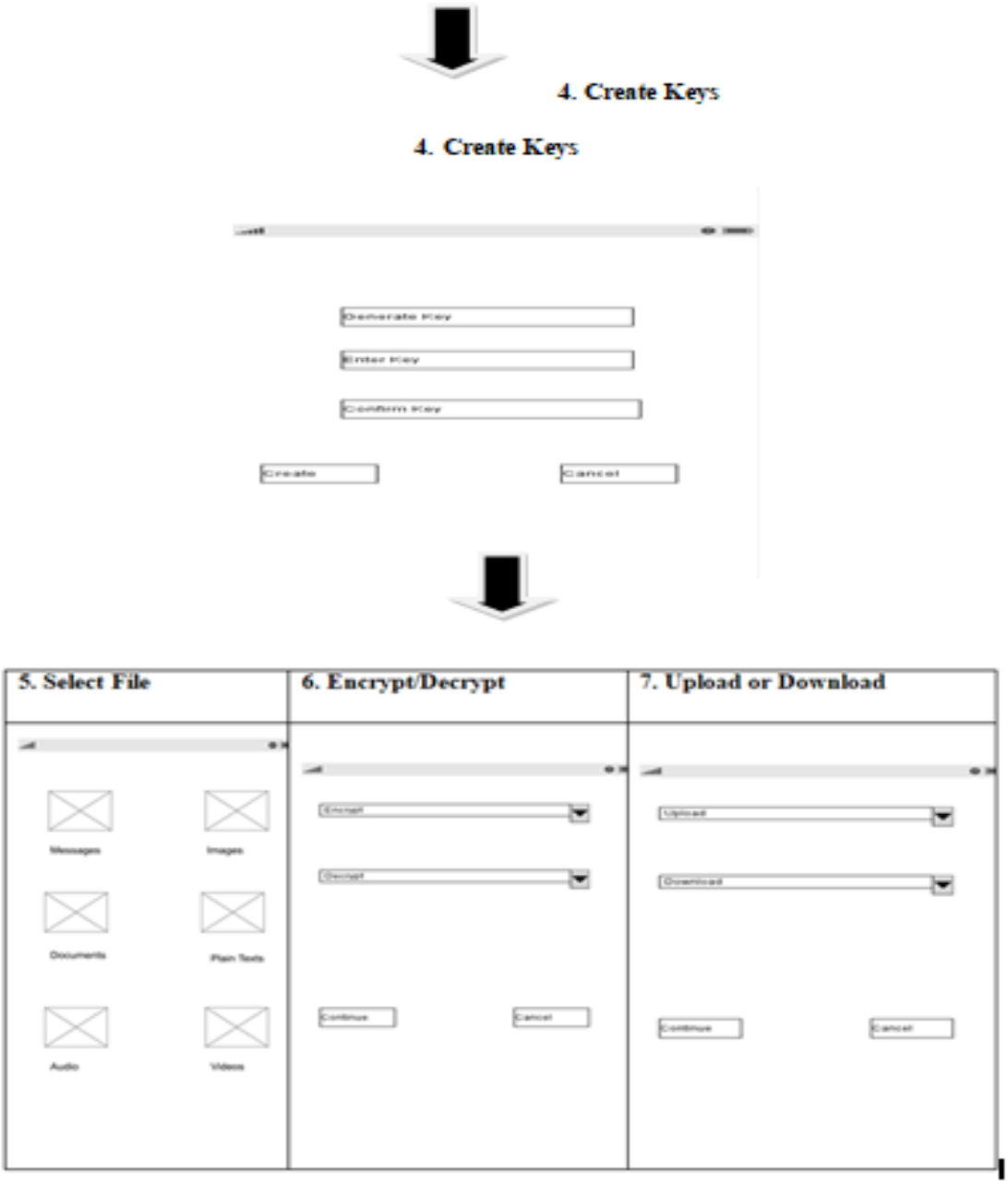


Figure 4.8: Wireframe

4.5.2. Network and Security Designs

As mentioned before, the system utilized client-server-based architecture. Thus, a server-based network is used. The user connected the wearable device with the tool to the network through WiFi to facilitate data transfer from the device to the cloud. Data requests to and from the server were sent through the MQTT and HTTPS protocols, securing the communication from external threats. Regarding security, the system was designed to have access controls that required the user to log in before initiating any activity, including data transfer. Thus, the system provides authentication as an initial security requirement for the user. It then allows the user to encrypt the data to provide confidentiality and integrity before and after transfer. Login credentials and encryption keys were transferred over a secure HTTPS network and stored in the server. The keys were generated using the Elliptic-Curve Cryptography (ECC) algorithm.

4.7. Chapter Summary

The section focuses on the design aspect by giving an overview of the requirements gathered from the users and the components needed for the implementation. Thus, the chapter has outlined the study's hardware and software requirements by detailing the software's design models. These included the system diagrams that were necessary for the attainment of the value of the proposed framework, and as such, meeting the requirement of objective three of the study.

CHAPTER FIVE: SYSTEM IMPLEMENTATION AND TESTING

5.1. Introduction

The chapter describes the implementation and testing of the tool that was developed during this study. The implementation constituted both hardware and software techniques that were used to realize the proposed framework for securing data from wearable devices. Thus, the chapter focuses on detailing the steps and algorithms that were used during the study.

5.2. System Implementation

5.2.1. Hardware Requirements

The encryption system was implemented on a 64-bit computer running a Windows operating system and tested on a smartwatch with an android operating system. Other specifications of the computing environment used to build the system included; Core i7 processor with a 2.4GHz speed, 12GB RAM, and 500 GB Hard Drive. The prototype hardware components included jumper wires, NodeMCU, Max30100 heart rate sensor, USB cable, and breadboard.

5.2.2. Software Requirements

The development was divided into frontend and backend. For the backend development, PHP v7.1.1.1 and MySQL v5.6.4.3 were used. The program's tools used to create the application were Java 8, Android Studio, and Gradle 3.6.0. The encryption code is shown in Appendix 6

5.2.3. System Development Segments

This was based on the wireframes developed in the previous chapter. It was divided into five segments as described below.

Data Source Prototype

The circuit in Figure 4.2 was used to develop a prototype of the wearable device. To obtain live data, the user was required to place their fingers on top of the Max30100 heart sensor. The code used to program the NodeMCU board is found in Appendix 7. The data was transmitted to the MQTT server, whose implementation is described in the next section. To ensure the client was

sending the data to the server, the code was created to specify the Server link, Username, Password, and Port which were securely protected. These details are shown in Figure 5.2

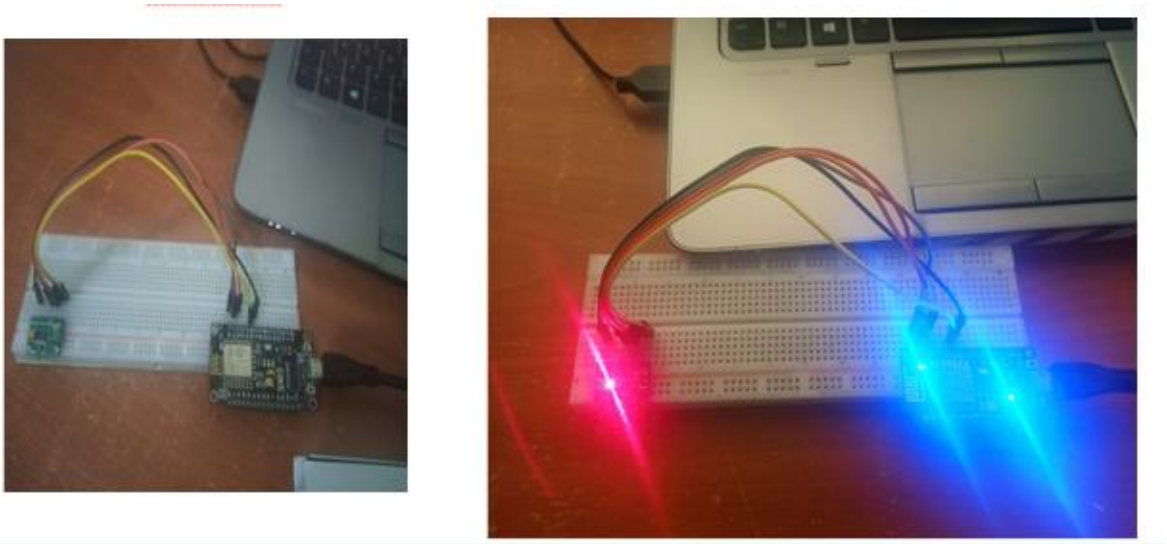


Figure 5.1: Hardware Prototype

Data Transfer with MQTT, NodeMCU, and Application

The MQTT was chosen for data transmission because it is lightweight and can facilitate communications with low bandwidth. The MQTT broker acted as the server while the nodemuc and application were the clients sending and receiving data. To create the MQTT broker, the user must sign up on the cloud MQTT server. After successful login, the user can access the server's details, which are needed to program the nodemcu to receive data. The cloud MQTT thus acted as the intermediary between the sensor node and encryption application. It uses SSL/TLS, which secures data before it reaches the encryption application. A Paho MQTT plugin was included in the app.gradle file in Andriod Studio to facilitate the data transmission to the encryption application. The paho-MQTT client implemented connection, subscribe, messaging, and other functions between the client and MQTT broker. In the main activity, three main functions are included; Connect ( ), Publish ( ) and Subscribe ( ). The connect function is used to connect with the Cloud MQTT Broker by stating the “tcp://yourserver-URL: port.” The server could connect up to twenty-five clients (users). However, this could be adjusted by increasing the package subscription on the cloud MQTT server. Figure 5.2 shows the server details, while Figure 5.3

shows live data steaming into the server from the hardware prototype that was developed during the implementation.

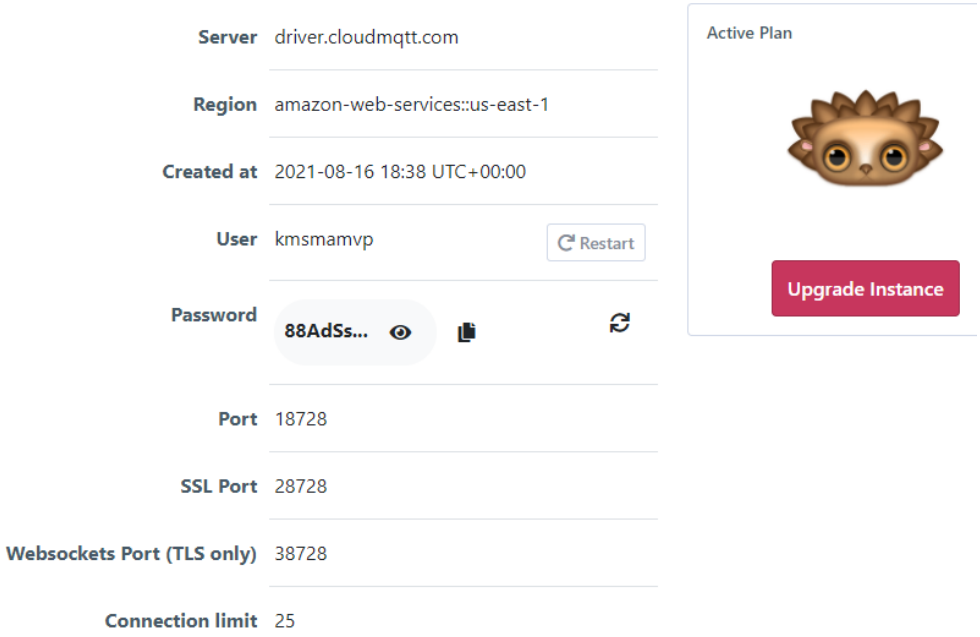


Figure 5.2: Cloudmqtt Server

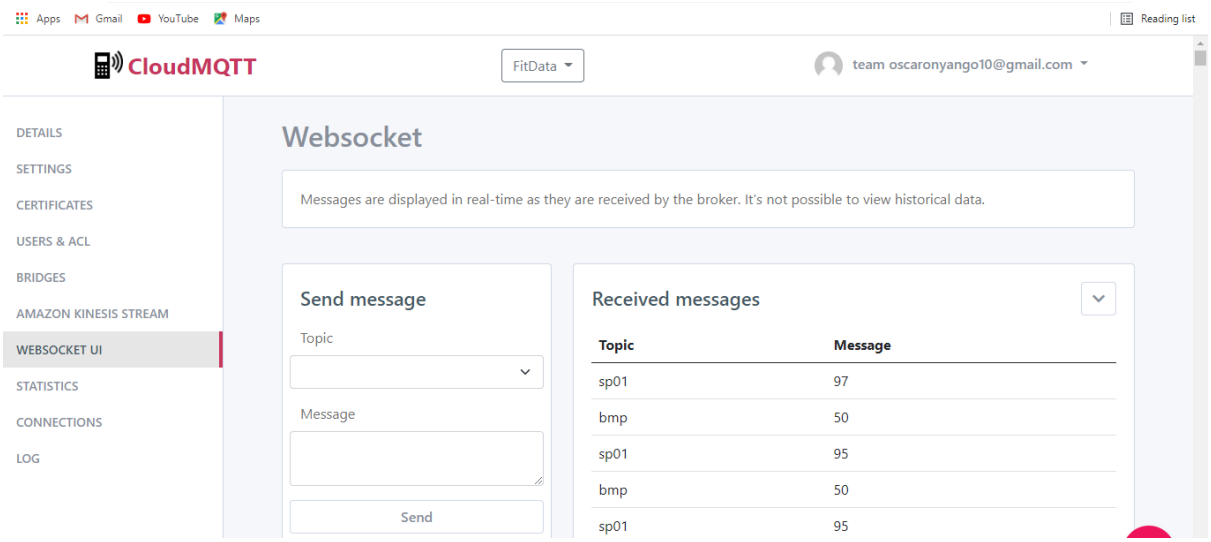


Figure 5.3: Live Data Streaming into the Server

Application Development and Hosting

The application was developed using Android Studio software. It was hosted using cPanel Server. The ECC library used for encryption and decryption was forked from GitHub (https://github.com/cossacklabs/themis). ECC was suitable for this framework because it provided shorter encryption keys with less memory and CPU resources. This was necessary since most wearables come with less memory and low CPU capacity. It is also computationally complex, which makes it quite secure. This is because it is based on the elliptic curve. Interestingly, it also computes very fast, which makes the framework rapidly fast in operation. Figure 5.2 gives a snippet of the simulator during the implementation of the application.

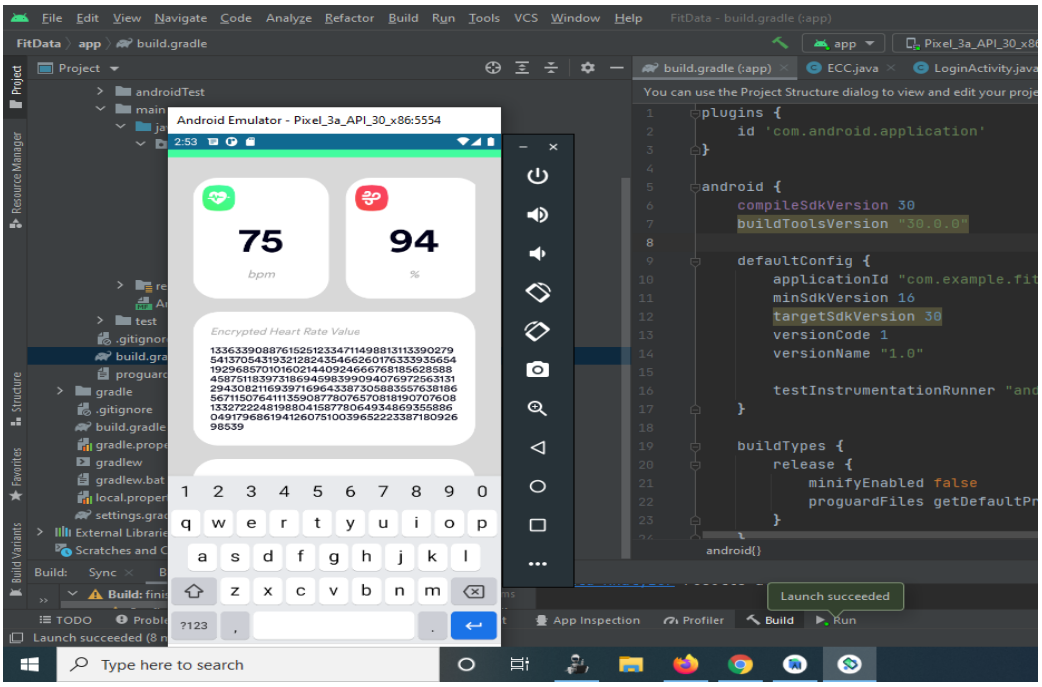


Figure 5.4: Application Implementation Simulator

Application User Interfaces

The sections describe the various interfaces user can access in the system and the operation of each interface

i. Account Registration (Sign Up) and Sign In

Once the user installs the application on the smartwatch wearable device, they must sign up to create an account that would allow them to log in and access the system utilities. During

registration, the user has to provide valid email addresses and passwords. The user has to confirm the password. The account registration and login page are shown in the screenshots presented in figure 5.1 below.

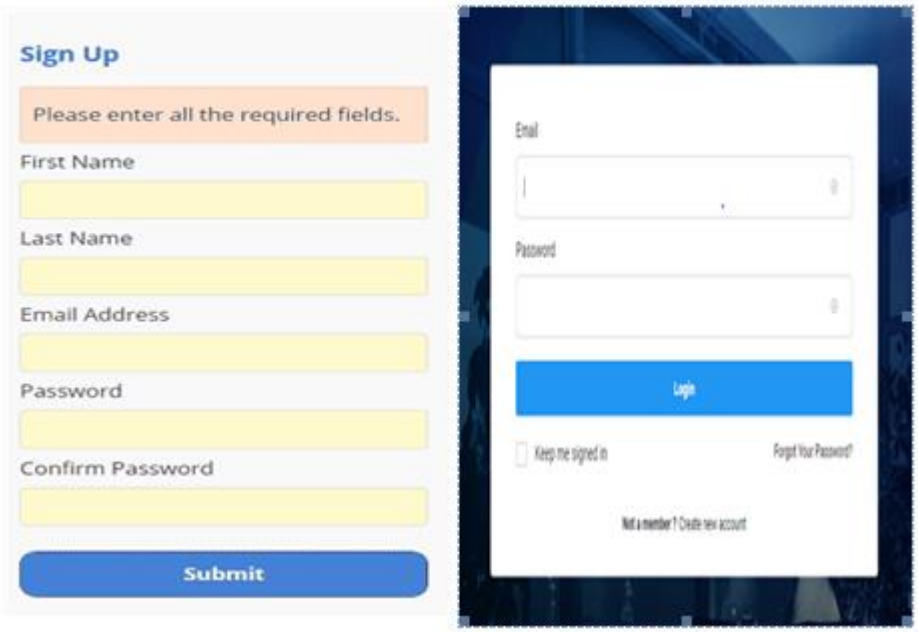


Figure 5.5: Account Registration and Login Screens

ii. Home Screen

This is the first page that the user accesses once they login into the system. It shows all the files contained in the wearable device. The user must confirm permission to give the application access to the file system. There is a menu bar on the bottom page where the user can select which operation they intend to do. Figure 5.2 presents the home page of the developed application.

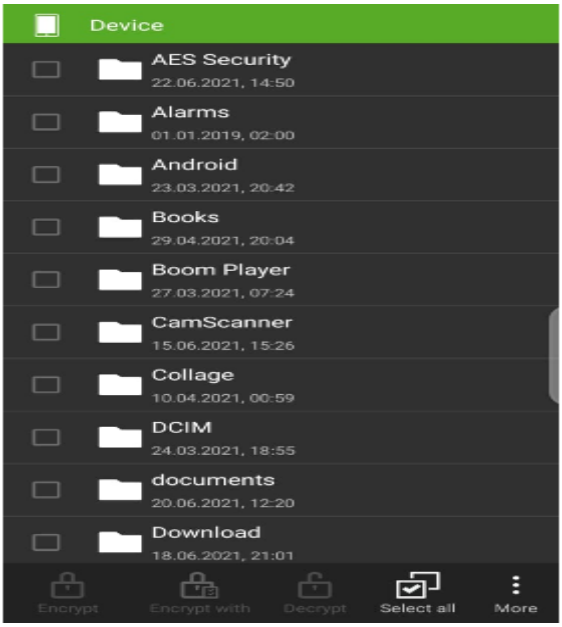


Figure 5.6: Application Home Screen

iii. Cryptographic Key Creation

In this segment, the user already has access to the system after login in. The user can therefore create the cryptographic keys needed for encryption of data. The keys as based on public-key encryption using the ECC algorithm. The user must confirm the valid key length to continue. Otherwise, the system will not access it. Figure 5.2 shows a screenshot of the cryptographic key creation process.

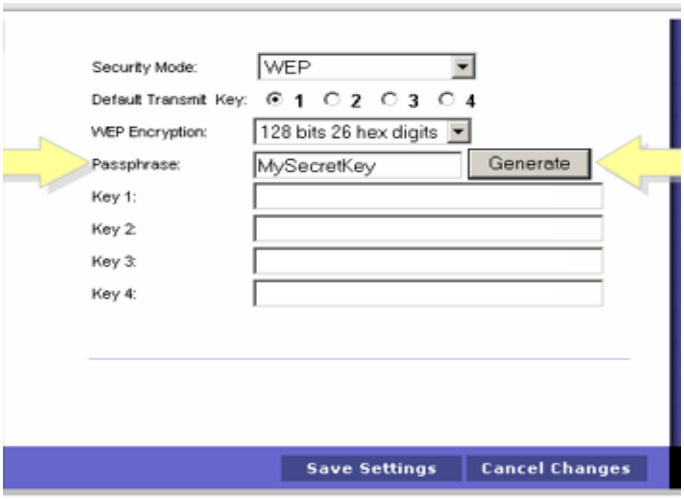


Figure 5.7: Key Generation

iv. Encryption

This three-step approach allows the user to identify the file type they want to encrypt and secure it before transfer. Naturally, the data collected by the wearable device is stored in the local storage. The user can create a file from any of the folders shown in figure 5.2. Within the folders, there are different file types that the user can proceed to encrypt before sending. The application prompts the user to provide their password to validate the encryption. Once encrypted, the data is automatically transmitted to the cloud for storage. Figure 5.4 shows the successful image of the encryption process. A data file named Oscar Onyango-Deliverable 3 was encrypted in this case as an example.

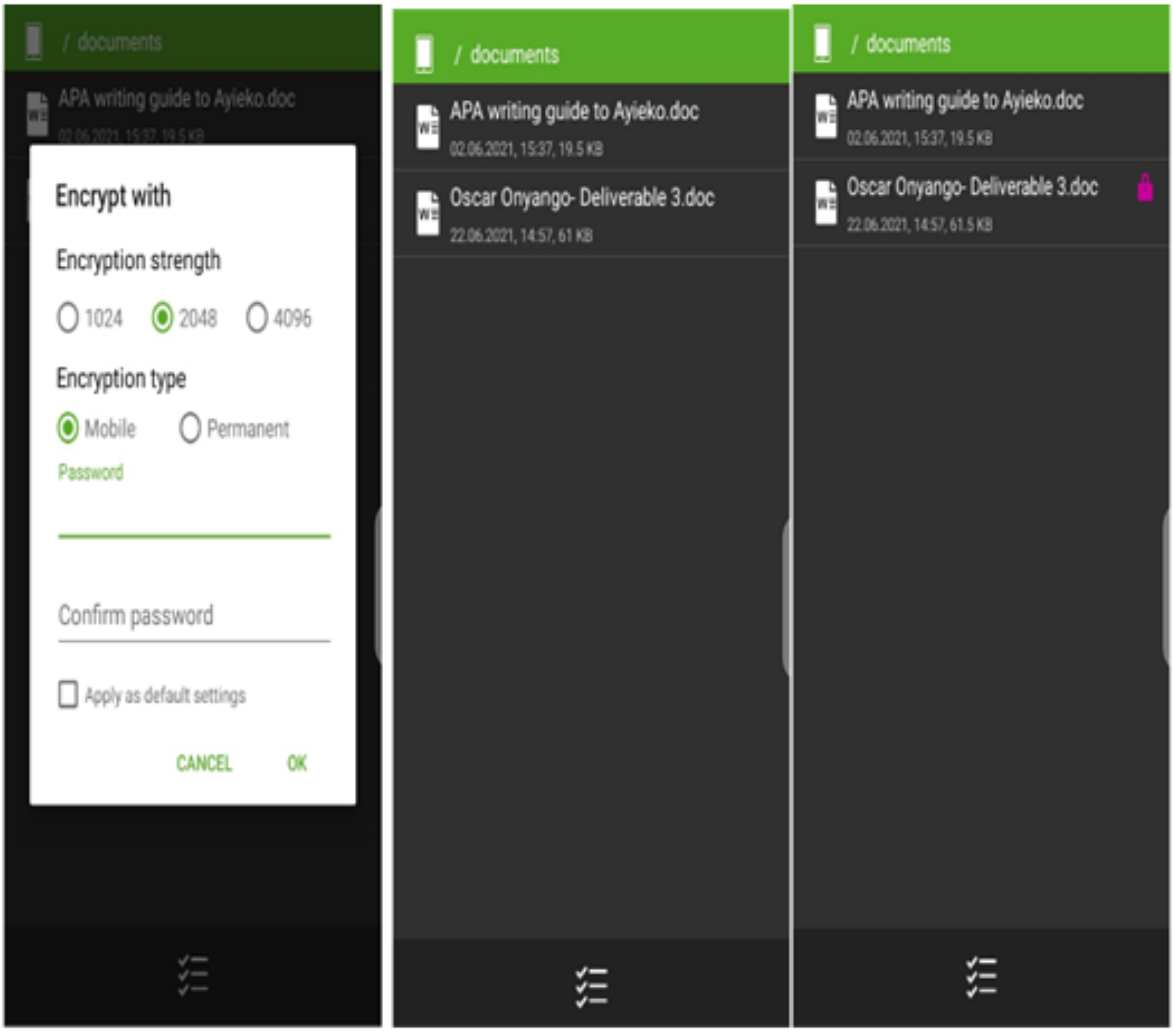


Figure 5.8: Encryption Process

v. Decryption

In this category, the user is only allowed to access the data by decrypting the file. It begins with the user downloading the file. The file is stored in the home view. From here, the user selects the file to decrypt and begins the decryption process. Again, the user is required to validate the decryption by providing a password. The password must match the one that was used to encrypt the file. Otherwise, the decryption will fail. Figure 5.5 shows the successful image of the decryption process. A data file named Oscar Onyango-Deliverable 3 was decrypted in this case.

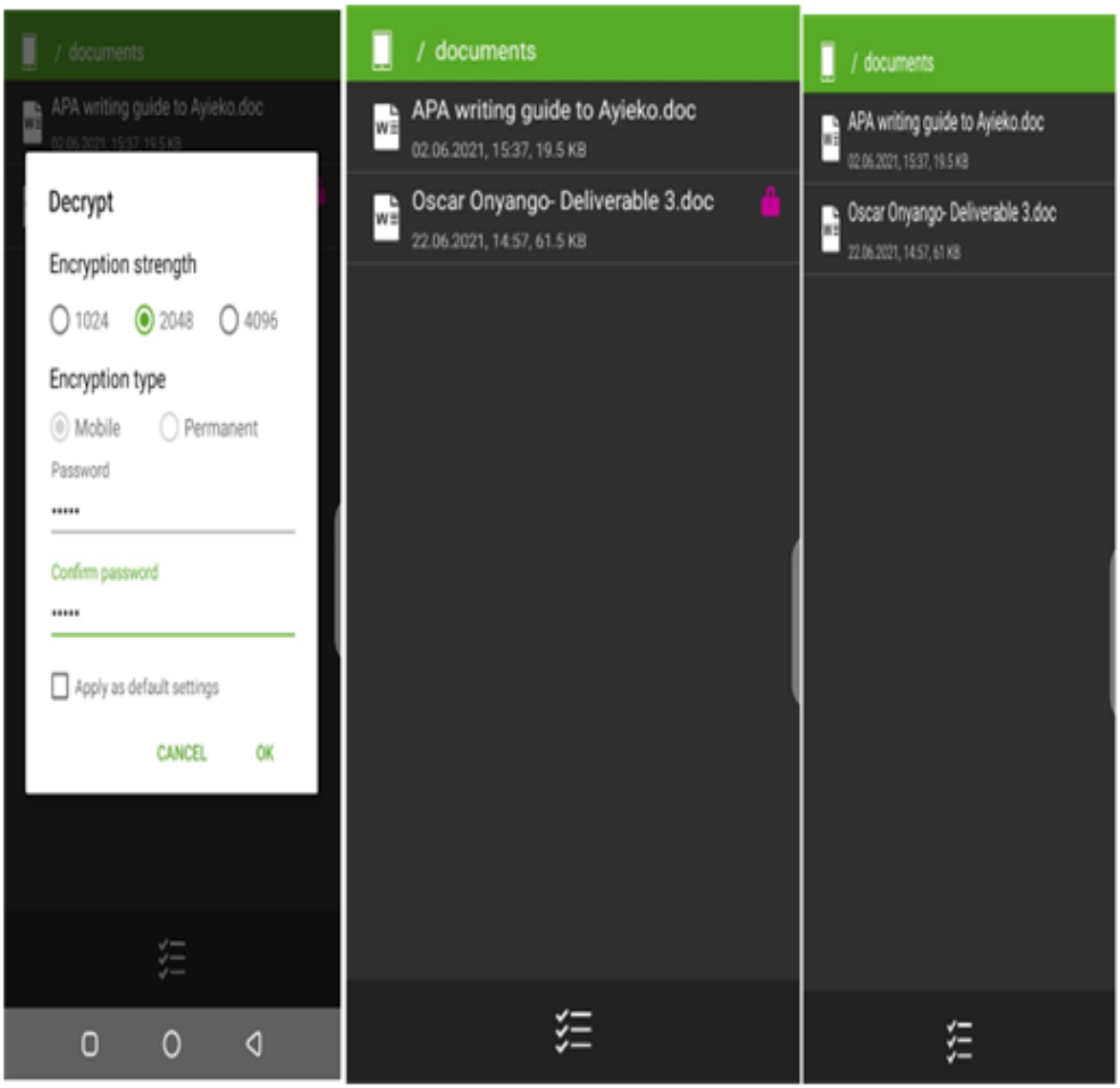


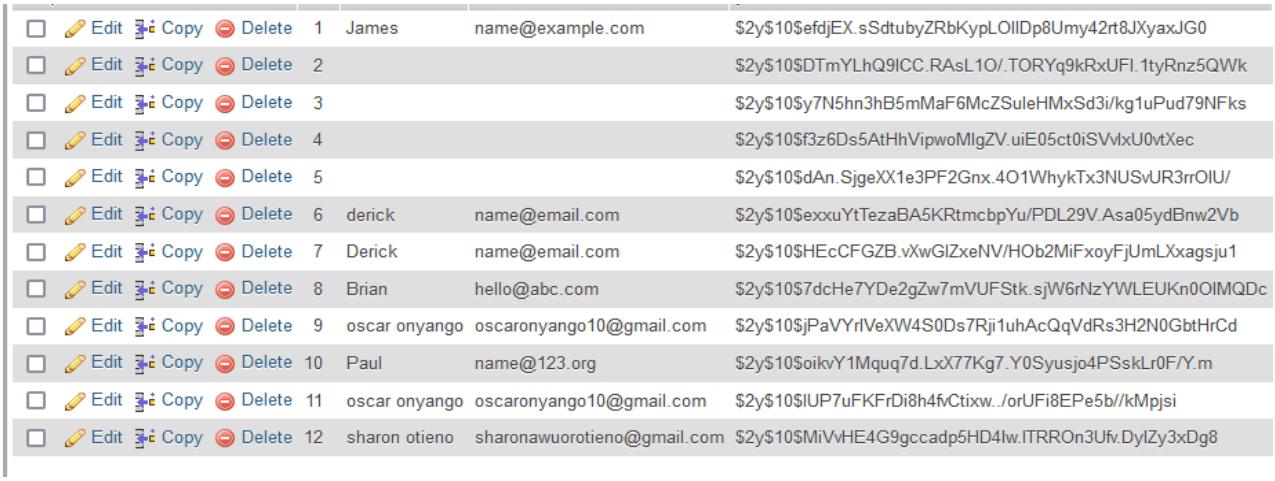
Figure 5.9: Decryption Process

Database Server

The database was created using cPanel MySQL. MySQL database allows the user to create, manage, and delete databases. A dataset name can have a maximum of sixty-four characters. The database is divided into two as described below;

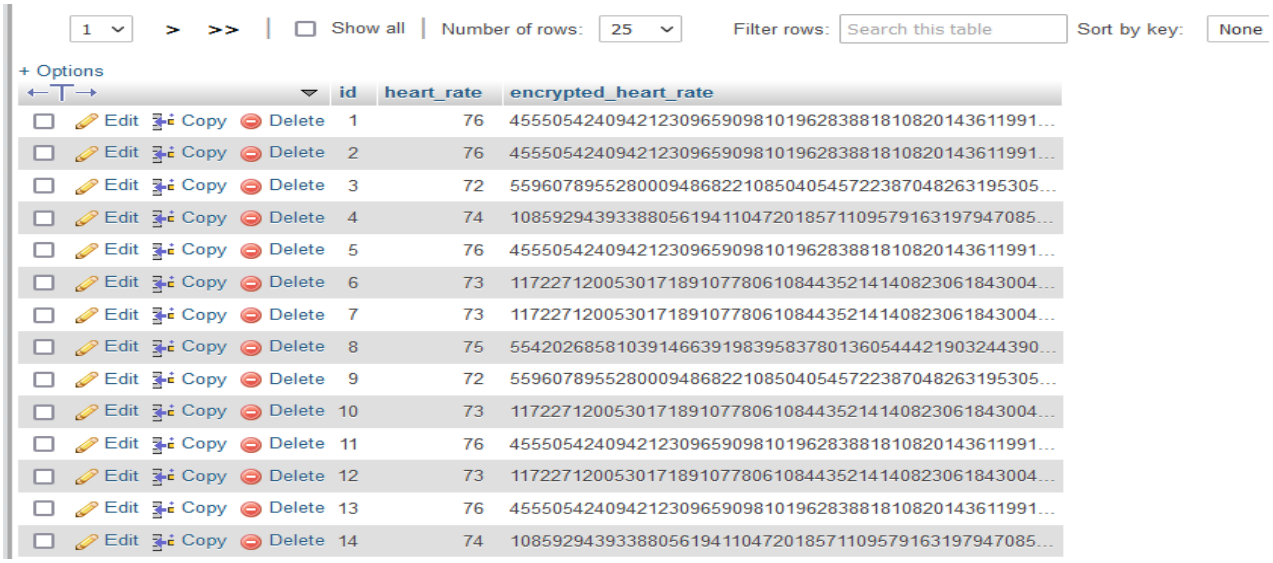
i. Registered Users

Once the user registers an account with the system, a cloud server database stores all the credentials (username and password). Figure 5.6 below provides a snippet of the database that was used to record several users who registered into the system.



Edit Copy Delete	1	James	name@example.com	$2y$10$efdjEX.sSdtubyZRbKypLOllDp8Umy42rt8JXyaxJG0
Edit Copy Delete	2			$2y$10$DTmYLhQ9ICC.RAsL1O/.TORYq9kRxUFI.1tyRnz5QWk
Edit Copy Delete	3			$2y$10$y7N5hn3hB5mMaF6McZSuleHMxSd3i/kg1uPud79NFks
Edit Copy Delete	4			$2y$10$f3z6Ds5AtHhVipwoMlgZV.uiE05ct0iSVvlxU0vtXec
Edit Copy Delete	5			$2y$10$dAn.SjgeXX1e3PF2Gnx.4O1WhykTx3NUSvUR3rrOlU/
Edit Copy Delete	6	derick	name@email.com	$2y$10$exxuYtTezaBA5KRtmcbpYu/PDL29V.Asa05ydBnw2Vb
Edit Copy Delete	7	Derick	name@email.com	$2y$10$HEcCFGZB.vXwGlZxeNV/HOb2MiFxoyFjUmLXxagsju1
Edit Copy Delete	8	Brian	hello@abc.com	$2y$10$7dcHe7YDe2gZw7mVUFStk.sjW6rNzYWLEUKn0OlMQDc
Edit Copy Delete	9	oscar onyango	oscaronyango10@gmail.com	$2y$10$jPaVYrlVeXW4S0Ds7Rji1uhAcQqVdRs3H2N0GbtHrCd
Edit Copy Delete	10	Paul	name@123.org	$2y$10$oikvY1Mquq7d.LxX77Kg7.Y0Syusjo4PSskLr0F/Y.m
Edit Copy Delete	11	oscar onyango	oscaronyango10@gmail.com	$2y$10$IUP7uFKFrDi8h4fvCtixw../orUFi8EPe5b//kMpjsi
Edit Copy Delete	12	sharon otieno	sharonawuorotieno@gmail.com	$2y$10$MiVvHE4G9gccadp5HD4Iw.ITRROn3Ufv.DylZy3xDg8

Figure 5.10: Registered UsersEncrypted and Decrypted Data



1 > >> | Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	id	heart_rate	encrypted_heart_rate
Edit Copy Delete	1	76	455505424094212309659098101962838818108201436119911...
Edit Copy Delete	2	76	455505424094212309659098101962838818108201436119911...
Edit Copy Delete	3	72	559607895528000948682210850405457223870482631953055...
Edit Copy Delete	4	74	108592943933880561941104720185711095791631979470855...
Edit Copy Delete	5	76	455505424094212309659098101962838818108201436119911...
Edit Copy Delete	6	73	117227120053017189107780610844352141408230618430045...
Edit Copy Delete	7	73	117227120053017189107780610844352141408230618430045...
Edit Copy Delete	8	75	554202685810391466391983958378013605444219032443905...
Edit Copy Delete	9	72	559607895528000948682210850405457223870482631953055...
Edit Copy Delete	10	73	117227120053017189107780610844352141408230618430045...
Edit Copy Delete	11	76	455505424094212309659098101962838818108201436119911...
Edit Copy Delete	12	73	117227120053017189107780610844352141408230618430045...
Edit Copy Delete	13	76	455505424094212309659098101962838818108201436119911...
Edit Copy Delete	14	74	108592943933880561941104720185711095791631979470855...

Figure 5.11: Encryption and Decryption Database

5.3. System Testing

The section describes the functionality tests that were performed to verify if the system met the intended objectives.

5.3.1. Functional Requirements Tests

These tests were based on the five segments described in section 5.2.2. Table 5.1 below gives a summary of the results of the tests. They were achieved through user tests which determined whether the respective segment was a success or failure. The results from the tests generally indicate that the tests were successful, and thus, the system met the intended objectives of encrypted data, thus securing it during transmission from the device to the cloud.

Table 5.1: Functional Requirements Tests Summary

Segment	User Test Description	Results
Account Registration	User required to create an account by providing and confirming their credentials (Name, Email, and Password)	Test passed
Account Login and Logout	Users are required to login and logout of the system using their personal credentials	Test passed
Data Views	User required to access the main menu and view various data files available in the device	Test passed
Key Generation	User required to create encryption and decryption keys of valid length	Test passed
File Encryption and Upload	User required to encrypt a data	Test passed

	file and upload it to the cloud	
File Download and Decryption	User required to download the data file from cloud and decrypt	Test passed

5.3.2. Non-Functional Requirement Tests

The non-functional requirement tests were used to determine the general outlook of the system in terms of its usability, performance, compatibility, and stability. Again, user tests were applied, and the results indicated that the system was stable, compatible with android version 8 and above, recording better performance speed, and was achieved its usability. Table 5.2 gives a summary of the tests processes and results.

Table 5.2: Non-Functional Requirements Tests Summary

Segment	User Test Description	Results
Usability and robustness	User was required to install the application in their devices and navigate through while responding to the questionnaire	User recorded satisfaction with the system usability
Performance and accuracy	User was required to install the application in their devices and navigate through while evaluating its performance The user was required to test the measurement and transmission of heart-rate data. The user was required to test	User recorded satisfaction with the system performance in terms of speed, interface, encryption, and decryption processes.

	encryption and decryption speed, accuracy, and precision.	
Compatibility and Stability	The user was required to install the application on devices with different android versions	Successful installation in devices with android version 8, 8.1, 9, and 10. Its operation was also stable.

5.3.3. Threat Alert System

Table 5.3: System Response to Threats

Threat	System Response
Login attempts	The system locks after three unsuccessful login attempts
Data decryption attempt	The user was required to input their password to decrypt data. The system locks if there are three failed password attempts
Data transmission attempt	Only login users can transmit data through their registered emails. The system would display an email error if a different email were used.
DDoS attack on the system	The system was fitted with inbuilt firewalls which alert the administrator in case of a denial of service attack or traffic sniffing

5.4. System Validation

The first stage was content validation that was conducted through a questionnaire designed to explore whether the tool met the research objectives. This was conducted by sharing the developed tool with five people in the IoT laboratory to share their experience if it provided the three parameters confidentiality, integrity, and authentication. The results of the survey were as follows.

Functionality: The selected users tested the system functionality by determining that it provided the three study parameters, confidentiality, integrity, and authentication, and recorded satisfaction with the results.

Table 5.4: Functionality Tests and Validation

	Functionality Tests and Validation		
	Authentication	**Confidentiality**	**Integrity**
Test User 1	Yes	Yes	Yes
Test User 2	Yes	Yes	Did not know
Test User 3	Yes	Yes	Yes
Test User 4	Yes	Yes	Yes
Test User 5	Yes	Did not know	Yes

Acceptability: Four out of the five test users accepted the system and were willing to share the application with their friends and family as a security framework in their wearable devices to assist secure their data during transfer from these devices.

CHAPTER SIX: DISCUSSION

6.1. Introduction

The chapter discusses the outcome of the study, giving a clear analysis of the recorded information during the process. The discussion is objective specific, as it considers the outcome of each of the study objectives.

6.2. The Technologies that Support Patient Data Sharing in the Wearable Device

The study analyzed patient data transmission from sensor to listening device through a literature review to determine the technologies that support data sharing in these devices. The study determined that Wi-Fi, Bluetooth, Global Positioning System (GPS), and Cellular Communication are the primary technologies that support data sharing from the wearables.

Traditionally, wearables devices mostly utilized short-range Bluetooth technology to share data. The Bluetooth chips are small and fit nicely on the wearables. Bluetooth is the most energy-efficient data transfer technology compared to Wi-Fi and cellular communication. Unfortunately, it can only transmit low volumes of data which makes it less efficient compared to Wi-Fi.

If the wearable needs to transmit a lot of data with little lag, Wi-Fi seemed to be the best option. In other words, Wi-Fi is suitable when streaming large amounts of data. Unfortunately, it has high power consumption, requiring that the wearable device's battery must be charged daily. Wi-Fi connects directly with the internet through the Wi-Fi Access point. It provides a better user experience, especially since there is an increasing demand for its usage in most medical facilities. Wi-Fi capabilities allow the devices to roam through the network and receive notifications, make and receive calls in addition to transmitted data. It has become possible to offload more data on the Wi-Fi network. In 2016, Cisco predicted that it would be possible to offload more data traffic through Wi-Fi, and this has since been achieved.

Also, wearable devices use cellular radio to link directly to cellular networks. Although cellular communication may be convenient because it connects the device to the cloud, it is highly

inconvenient. Cellular communication is increasingly becoming a faster method than Bluetooth, especially with the introduction of 5G and 6G networks. In Kenya, 5G is constantly helping with the evolution of the Internet of Things (IoT), the framework through which most wearable devices operate. It allows the wearables to hold more sensors which allows them to collect more data. 5G allows the wearables to have an all-inclusive communication platform where they can deliver low-end, mid-end, and high-end wearable application requirements.

Lastly, Global Positioning System (GPS) is a location-based technology that determines the device's exact location through a process referred to as triangulation. Triangulation involves calculating the difference between the time the running wearable device receives a GPS signal and the time the signal is sent to the device. By determining the difference between the receiving and sending of the GPS signal, it is possible to determine the satellite's location. Wearable devices, especially fitness watches, measure several location metrics like distance, pace, and speed. Through this, the user can have real-time insight into their body performance.

In short, wearables devices had managed to combine various data transfer technologies giving users the option of selecting which technique is convenient to them. For example, Apple Watch has Bluetooth Classic, Wi-Fi, and NFC that supports mobile payments.

Table 6.1: Summary of Data Transfer Technologies

Bluetooth	Wi-Fi	Cellular Communication	GPS
It is suitable when streaming low amounts of data **Less efficient**	Wi-Fi is suitable when streaming large amounts of data High power consumption Better user experience	Faster method compared to Bluetooth Allows the wearables to hold more sensors which allows them to collect more data Highly inconvenient because of high power consumption	Determine location through triangulation Give the user can have real-time insight into their body performance

6.3. Techniques Used to Abstract Data

Data abstraction is a technique used to enhance security during data transmission in wearable devices. The process hides unnecessary information from the users and helps reduce programming complexities. It is object-oriented programming that solves issues at the design level. It allows developers to group various related objects in classes.

The study employed literature to analyze various abstraction techniques. It determined that wearable devices use four techniques in data abstraction. The first technique is differences in data models, where data from the wearables are organized in temporal segments using different models. The models are categorized using the elements and attributes in the key-value pairs as in JavaScript Object Notation (JSON) and extensible Markup Language. For example, Microsoft manages to sleep, and heart rate under different segments called sleep and heart-rate segments. Next is a difference in data names technique where each wearable uses different vocabularies to refer to similar parameters. For example, Microsoft managers sleep types using words like Doze, Awake, Sleep, and Snooze, while Google Fit uses Sleep lightly, deep, Rapid Eye Movement, and awake to describe different levels of sleep. Another technique is temporal discrepancies. In this approach, different wearables have a different methods of identifying traces. For example, Microsoft identifies various heart-rate periods using a single identifier per day while identifier each heart-rate period using a different identifier. The last technique is differences in counters, where each wearable vendor follows a specific approach to count events produced by the devices. For example, Microsoft takes counts of all items produced like number of steps, distance, calories, and heart rate.

6.4. The Encryption Framework Implemented to Secure Patient Data Transfer in Wearable Heart-rate Monitors Using Encryption Algorithms

The data was encrypted using an improved ECC encryption method. Improved ECC (IECC) is a curve-based approach with specific base points generated from prime numbers' functions. ECC was selected because it requires shorter encryption keys which uses less memory and CPU resources. This was necessary since most wearables come with less memory and low CPU

capacity. ECC involves creating two keys (private and public). The private is added to the encryption formula and subtracted from the decryption formula, as shown in the mathematical illustration in Chapter 2 of the study.

Using the ECC library that applies the mathematical implementation shown in Chapter 2, the study developed an application-based framework that is installed in the wearable device. The application generates the keys needed to encrypt and decrypt the data. Upon encryption, the data can then be transmitted through wireless means, after which it can be decrypted when it reaches the destination using the same application. The password and cryptography keys are saved in the server, which the client cannot access for security reasons. Also, the connection to the server was implanted using the middle-tier application, thus creating a direct link between the client and the server. This way, it was easy to access the data when necessary from any Android-based wearable device.

6.5. Validation of the Implementation of the Security Framework

Based on the questionnaire's user experience, the study validated the framework and confirmed that it provides authentication, integrity, and confidentiality. For starters, authentication was achieved by requiring all users to register before using the application. Thus, before using the system, the user has to confirm their identity by providing accurate credentials, including their usernames and passwords, failure to which the system locks them out. The application also has assigned roles and privileges to the user. For example, they cannot delete or edit data. Their roles are limited to key creation, encryption, decryption, and data transfer. Generally, the application achieved authentication through the following approach;

i. The user is required to register on the application using their email address and password
ii. The application is designed to save the user credentials in the database
iii. After that, the application sends a verification email to the user to allow them to validate the registration
iv. On complete registration, the user must key incorrect credentials for logging in;
v. The user is allowed to access particular resources on successful authentication

vi. Finally, the user state is maintained using sessions.

The system's provision of confidentiality and integrity were tested and validated through user tests. It was noted that all confidential data in the wearable devices regarding the heart-rate measurement were saved in an encrypted format by using the cryptographic keys that the user-generated. Thus, an unauthorized person could not have access to the data. The passwords, keys, and usernames the user-created were stored in the server using the SHA-hash algorithm. The encrypted were uploaded to the server and could only be viewed or modified after decryption to ensure integrity.

CHAPTER SEVEN: CONCLUSIONS AND RECOMMENDATIONS

7.1. CONCLUSIONS

The study proposed and implemented an encryption framework to secure data in the heart-rate wearable devices during transfer. A successful implementation determined that wearable technologies have not fully exploited security mechanisms for data, especially those in transit. Using either proprietary systems or third-party systems, the devices can collect a wide range of data from their users using their inbuilt sensors. Since this data is critical for the user's health, it is necessary to protect it from unauthorized access. Public key encryption thus presents an excellent framework to secure wearable data because it reduces the chances of cybercriminals discovering or learning a person's secret key during data transmission. Elliptic-Curve Cryptography (ECC) was suitable for developing the application framework because it has smaller keys and can be computed substantially faster. Although ECC provides the same cryptographic strength as the RSA system, it is three times faster than AES. Elliptic-Curve Cryptography (ECC) can also run on mobiles with really less computing power and provides an equivalent encryption strength. By implanting the developed framework, data security is enhanced in wearable devices. Also, the study confirmed that the framework ensures that authentication, confidentiality, and integrity are achieved in the heart-rate wearables.

7.2. RECOMMENDATIONS

Wearable technology is becoming a huge deal in healthcare. The developed framework in this study is low-cost, user-friendly, and implanted on android platforms. However, the application should be scaled for production deployment to accommodate the iOS operating since several of the wearable devices are based on it.

Hackers are continuously evolving. The proposed approach ensures data security by storing both files and encryption keys in the same server. However, it would be convenient to create different servers for encryption keys and files to make the security more complex to attackers. This way, hackers will have to gain access to both servers to retrieve the data. In such a case, the system becomes more secure.

7.3. FUTURE WORKS

The study utilized ECC. Compared to other cryptographic algorithms like RSA and AES, ECC is narrowly applied. The algorithms have existed for around fifteen years, implying that only a few studies have been performed on them. The study has determined that ECC is more efficient since it requires smaller key sizes, offers better security, improves efficiency, and ensures perfect secrecy. Thus, future studies should focus on exploiting ECC and applications in IoT, Artificial Intelligence, and Big Data as a security framework.

Also, advanced studies on the security of heart-rate wearable devices should consider the integration of blockchain technology. This is because blockchain can increase trust, transparency, security, and data traceability that is transmitted across a network.

REFERENCES

Alsop, T. (2020, October 28). Fitbit: Statistics and facts. *Statista*. https://www.statista.com/topics/2595/fitbit/#dossierSummary__chapter1

Angel, T. S., & MesiaDhas, J. T. (2017). Requirement Engineering Model for Web Applications. *Indian Journal of Science and Technology*, *10*(11), 1-4.

Bent, B., Goldstein, B. A., Kibbe, W. A., & Dunn, J. P. (2020). Investigating sources of inaccuracy in wearable optical heart rate sensors. *NPJ digital medicine*, *3*(1), 1-9. https://www.nature.com/articles/s41746-020-0226-6

Berglund, M. E., Duvall, J., & Dunne, L. E. (2016, September). A survey of the historical scope and current trends of wearable technology applications. In *Proceedings of the 2016 ACM international symposium on wearable computers* (pp. 40-43).

Bhatia, V., Singhal, A., Bansal, A., & Prabhakar, N. (2019). A Review of Software Testing Approaches in Object-Oriented and Aspect-Oriented Systems. In *Software Engineering* (pp. 487-496). Springer, Singapore.

Blow, F., Hu, Y. H., & Hoppa, M. (2020, July). A Study on Vulnerabilities and Threats to Wearable Devices. In Journal of The Colloquium for Information Systems Security Education (Vol. 7, No. 1, pp. 7-7).

Borysowish, C. (2018). Systems design: Techniques for designing software structure. *ToolBox Tech*. Retrieved from https://it.toolbox.com/blogs/craigborysowich/systems-design-techniques-for-designing-software-structure-083111

Bruin, I. (2010). Exploring how objects used in a picture vocabulary test influence validity: Chapter three. The *University of Pretoria*. Retrieved from https://repository.up.ac.za/bitstream/handle/2263/25218/00front.pdf

Chandler, N. (2020). How Fitbit works. *How StuffWorks*. https://electronics.howstuffworks.com/gadgets/fitness/fitbit.htm

Chapter 2: Abstraction. Retrieved from http://web.engr.oregonstate.edu/~budd/Books/oopintro3e/info/chap02.pdf

Chau, K. Y., Lam, M. H. S., Cheung, M. L., Tso, E. K. H., Flint, S. W., Broom, D. R., ... & Lee, K. Y. (2019). Smart technology for healthcare: Exploring the antecedents of adoption intention of healthcare wearable technology. *Health psychology research*, *7*(1).

Ching, K. W., & Singh, M. M. (2016). Wearable technology devices security and privacy vulnerability analysis. *International Journal of Network Security & Its Applications*, *8*(3), 19-30.

Ching, K. W., & Singh, M. M. (2016). Wearable technology devices security and privacy vulnerability analysis. International Journal of Network Security & Its Applications, 8(3), 19-30.

Chowdhury, M. E., Alzoubi, K., Khandakar, A., Khalifa, R., Abouhasera, R., Koubaa, S., ... & Hasan, A. (2019). Wearable real-time heart attack detection and warning system to reduce road accidents. *Sensors*, *19*(12), 2780. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6632021/

Comte, A. (2019). *A General View of Positivism*. GENERAL PRESS.

ECG test. (2020). *Better Health Channel*. Retrieved from https://www.betterhealth.vic.gov.au/health/conditionsandtreatments/ecg-test

Eid, M. (2015). Requirement gathering methods. University of Missouri-St. Louis. https://www.umsl.edu/~sauterv/analysis/F2015/Requirement%20Gathering%20Methods.html.htm

Emoghene, O., & Nonyelum, O. F. (2017). Information Gathering Methods and Tools: A Comparative Study. *IUP Journal of Information Technology*, *13*(4), 51-62.

Fitbit. (2020). Compatible apps. Retrieved from https://www.fitbit.com/global/us/technology/partnership

Friese, S. (2019). *Qualitative data analysis with ATLAS. Ti*. SAGE Publications Limited.

Gallacher, J. (2019). World Health Organization cardiovascular disease risk prediction charts: revised models to estimate risk in 21 global regions. *The Lancet Global Health*.

Ghaffari, F., Gharaee, H., & Arabsorkhi, A. (2019, April). Cloud security issues based on people, process and technology model: a survey. In 2019 5th International Conference on Web Research (ICWR) (pp. 196-202). IEEE.

Goodman, G. (2004). Cardiovascular techniques and technology. In *Clinical engineering handbook* (pp. 417-420). Academic Press.

Guk, K., Han, G., Lim, J., Jeong, K., Kang, T., Lim, E. K., & Jung, J. (2019). Evolution of wearable devices with real-time disease monitoring for personalized healthcare. *Nanomaterials*, *9*(6), 813. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6631918/

Guru99. (2019). What is data analysis? Types, processes, methods, and techniques. Retrieved from https://www.guru99.com/what-is-data-analysis.html

Hankerson, D., Menezes, A. J., & Vanstone, S. (2006). *Guide to elliptic curve cryptography*. Springer Science & Business Media.

Heale, R., & Twycross, A. (2015). Validity and reliability in quantitative studies. *Evidence-based nursing*, *18*(3), 66-67.

Hintze, M., & El Emam, K. (2018). Comparing the benefits of pseudonymisation and anonymisation under the GDPR. Journal of Data Protection & Privacy, 2(2), 145-158.

HTTP overview. (n.d.). *Tutorials Point*. https://www.tutorialspoint.com/http/http_overview.htm

Kaptoge, S., Pennells, L., De Bacquer, D., Cooney, M. T., Kavousi, M., Stevens, G., ... & Amouyel, P. (2019). World Health Organization cardiovascular disease risk charts: revised models to estimate risk in 21 global regions. *The Lancet Global Health*, *7*(10), e1332-e1345.

Khabbazi, M. R., Wikander, J., Onori, M., & Maffei, A. (2018). Object-oriented design of product assembly features data requirements in advanced assembly planning. *Assembly Automation*, *38*(1), 97-112.

Khaishangi, Z. (2019, January 21). Cryptography: Explaining SHA-512. *Medium*. Retrieved from https://medium.com/@zaid960928/cryptography-explaining-sha-512-ad896365a0c1

Loncar-Turukalo, T., Zdravevski, E., da Silva, J. M., Chouvarda, I., & Trajkovik, V. (2019). Literature on wearable technology for connected health: Scoping review of research trends, advances, and barriers. *Journal of medical Internet research*, *21*(9), e14017.

Marrington, A., Kerr, D., & Gammack, J. (Eds.). (2016). Managing security issues and the hidden dangers of wearable technologies. IGI Global.

McFarland, R. J., & Olatunbosun, S. B. (2019). An exploratory study on the use of Internet_of_Medical_Things (IoMT) In the healthcare industry and their associated cybersecurity risks. In *Proceedings on the International Conference on Internet Computing (ICOMP)* (pp. 115-121). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldCom). https://csce.ucmss.com/cr/books/2019/LFS/CSREA2019/ICM2519.pdf

Mills, G. E., & Gay, L. R. (2019). *Educational research: Competencies for analysis and applications*. Pearson. One Lake Street, Upper Saddle River, New Jersey 07458.

Olet, M. (2016). 7 potentual security concerns for wearables. In CSO Online. Retrieved from https://www.csoonline.com/article/3054584/7-potential-security-concerns-for-wearables.html

Olugbenga, A. M. Wearable Technology for Enhanced Security. *Communications*, *5*, 7-12.

Ometov, A., Shubina, V., Klus, L., Skibińska, J., Saafi, S., Pascacio, P., ... & Lohan, E. S. (2021). A survey on wearable technology: History, state-of-the-art and current challenges. Computer Networks, 193, 108074.

Padyab, A., & Habibipour, A. (2021). Issues and Adoption Barriers in Wearable Technologies. International Journal of Technology Diffusion (IJTD), 12(1), 75-89.

Paul, C. P., Rajiv, J. A. C., Dana, C. L., & Carrie. , C. (2019). Reliability and Validity of Measurement. *PressBooks* Retrieved from https://opentext.wsu.edu/carriecuttler/chapter/reliability-and-validity-of-measurement/

Phoenix Nap. (2019, April 20). Secure server connectivity. Retrieved from https://phoenixnap.com/kb/server-security-tips

Pulse sensor. (2020). *ROHM Semiconductor*. Retrieved from https://www.rohm.com/electronics-basics/sensor/pulse-sensor

Roos, D. (2020). How to leverage an API for conferencing. *How Stuff Works*. https://money.howstuffworks.com/business-communications/how-to-leverage-an-api-for-conferencing1.htm

Sacred Heart University Library. (n.d.). Organizing academic research papers: Types of research design. Retrieved from https://library.sacredheart.edu/c.php?g=29803&p=185902

Schrijen, G.J. (2021). Basics of SCRAM PUF and how to deploy it for IoT security. Embedded Systems. https://www.embedded.com/basics-of-sram-puf-and-how-to-deploy-it-for-iot-security/

Shirzadfar, H., Ghaziasgar, M. S., Piri, Z., & Khanahmadi, M. (2018). Heartbeat rate monitoring using optical sensors. *International Journal of Biosensors & Bioelectronics*, *4*(2). https://medcraveonline.com/IJBSBE/heart-beat-rate-monitoring-using-optical-sensors.html

Sigh, S. (2018). Sampling techniques. *Towards Data Science*. Retrieved from https://towardsdatascience.com/sampling-techniques-a4e34111d808

Silhavy, R., Silhary, P., & Prokopova, Z. (2011). Requirements gathering methods in system engineering. *Recent Researches in Automatic Control*, 105-110.

Smill, D. (2003). Five principles for research ethics. *American Psychological Association*. Retrieved from https://www.apa.org/monitor/jan03/principles

Software analysis and design tools. (2020). *Tutorials Point*. https://www.tutorialspoint.com/software_engineering/software_analysis_design_tools.htm

Software deployment strategies. (2019, March 5). NI. Engineer https://www.ni.com/en-za/innovations/white-papers/06/software-deployment-strategies.html

Sumagita, M., Riadi, I., Sh, J. P. D. S., & Warungboto, U. (2018). Analysis of secure hash algorithm (SHA) 512 for encryption process on a web-based application. *International Journal of Cyber-Security and Digital Forensics (IJCSDF)*, *7*(4), 373-381.

Tang, S., & Shi, K. (2021). Data privacy protection technology of wearable-devices. Journal of Intelligent & Fuzzy Systems, (Preprint), 1-8.

The history of wearable technology. (2018, September 14). *Condeco Group Limited*. Retrieved from https://www.condecosoftware.com/blog/the-history-of-wearable-technology/

The University of Illinois Chicago. (n.d.). Big Data and wearable health monitors: harnessing the benefits and overcoming challenges. Retrieved from https://healthinformatics.uic.edu/blog/big-data-and-wearable-health-monitors-harnessing-the-benefits-and-overcoming-challenges/

Tiwari, S., & Rathore, S. S. (2017). A methodology for the selection of requirement elicitation techniques. *arXiv preprint arXiv:1709.08481*.

Tung, (2020). How it works: Fitbit. Jameco Electronics. https://www.jameco.com/Jameco/workshop/Howitworks/how-it-works-fitbit.html?__cf_chl_jschl_tk__=e777be3fd9c69ec33fe95547c508544773d8bc78-1606408757-0-ARcmy3qjzujlwyJYtiFZsAlgaL0gnFISucv9gJwLx-9MneZrEb9DFERFQyINvhKqMhlu6wcREHi_PxWF35vQmVNJOJv4oQ2fnGaPB0DiMOZYbbpZF6PnXH267u9XgrsIKSIo73BSj1D84B6zTNjp4nhV1mFNIX0_78GVE1pyX8O9PmtN3RdEUx0L97h1eJHIdxN8YJ8YARA_A23TY1U3f8N2FYRviAWtPtnt2Gd2gmyZ2_epRS8qrWqQsqXQ_6Yq8WN_xchCi4gOgyaWrNWxWhOUjYIa9j3n2FTyVfb2c9Lcq4yDuxCNS6s5xWkEnTpAHpnocunoGNXhgYQy6VnrcPG_BBOkduCjImEO6LGeaXcENWhR30D8NdOsWKhtkNN-iQ

Vallencell. (2020). Optical heart rate monitoring. What you need to know. Retrieved from https://valencell.com/blog/optical-heart-rate-monitoring-what-you-need-to-know/

World Health Organization. (n.d.). Cardiovascular diseases. Retrieved from https://www.who.int/health-topics/cardiovascular-diseases/#tab=tab_1

World Heart Federation. (2019, May 10). Advancing heart health in Kenya. Retrieved from https://www.world-heart-federation.org/news/advancing-heart-health-in-kenya/

Wu, J., Li, H., Cheng, S., & Lin, Z. (2016). The promising future of healthcare services: When big data analytics meets wearable technology. *Information & Management*, *53*(8), 1020-1033.

APPENDICES

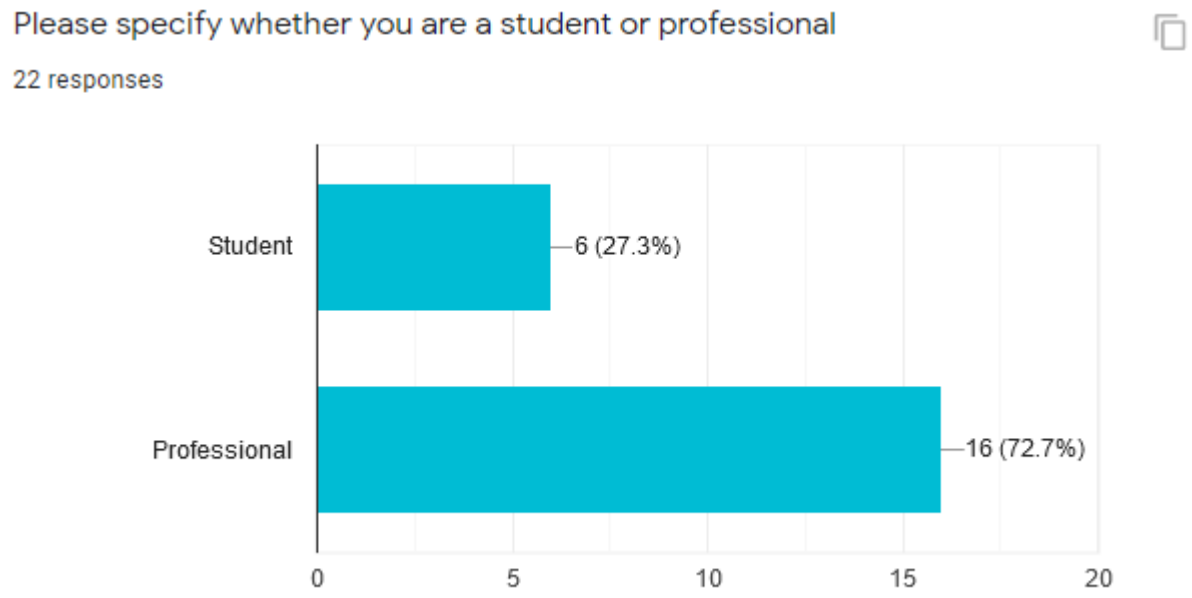
Appendix 1: Survey Questionnaire

The link below provides the Survey Questionnaire on the usage of Wearable Device Technology.

https://docs.google.com/forms/d/e/1FAIpQLScHnAiPqlCUbX4Y7Vlerq03kEQ_861IBVFryXQODCr5P3xyHA/viewform?usp=sf_link

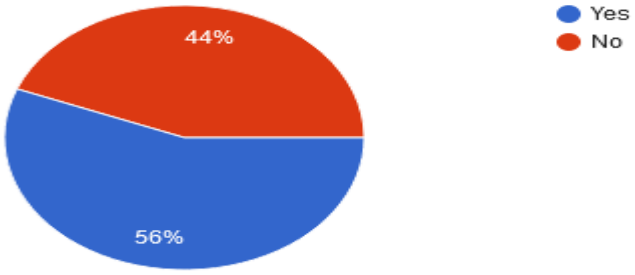
Appendix 2: Survey Results

Sex	Number of Respondents
Male	18
Female	10



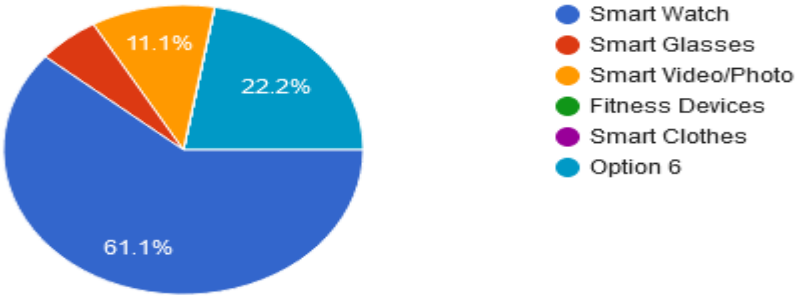
Do you own a wearable device

25 responses



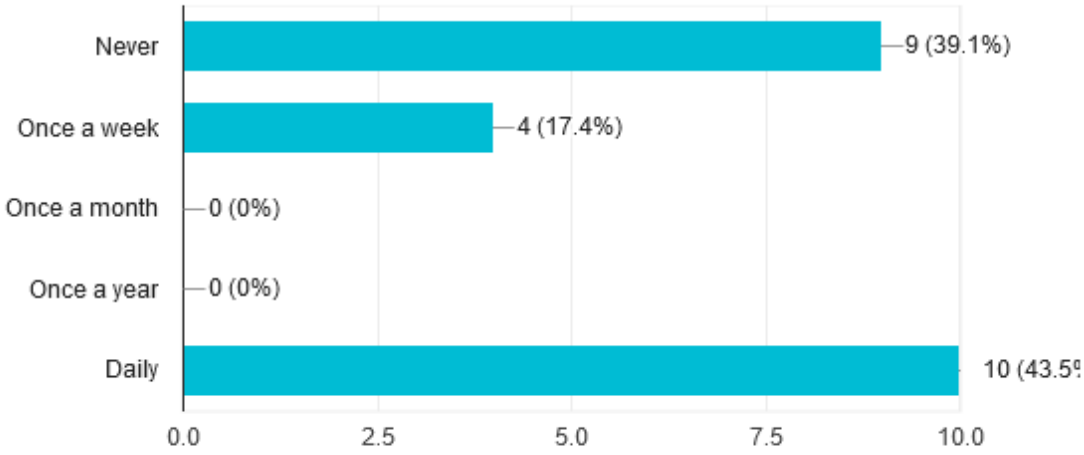
If yes,what wearable device do you own?

18 responses



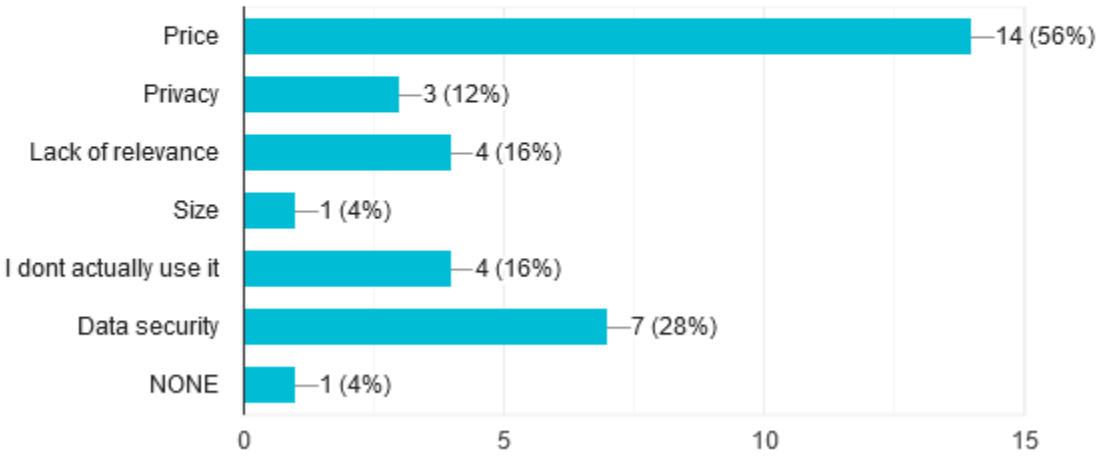
How often do you use the wearable device?

23 responses



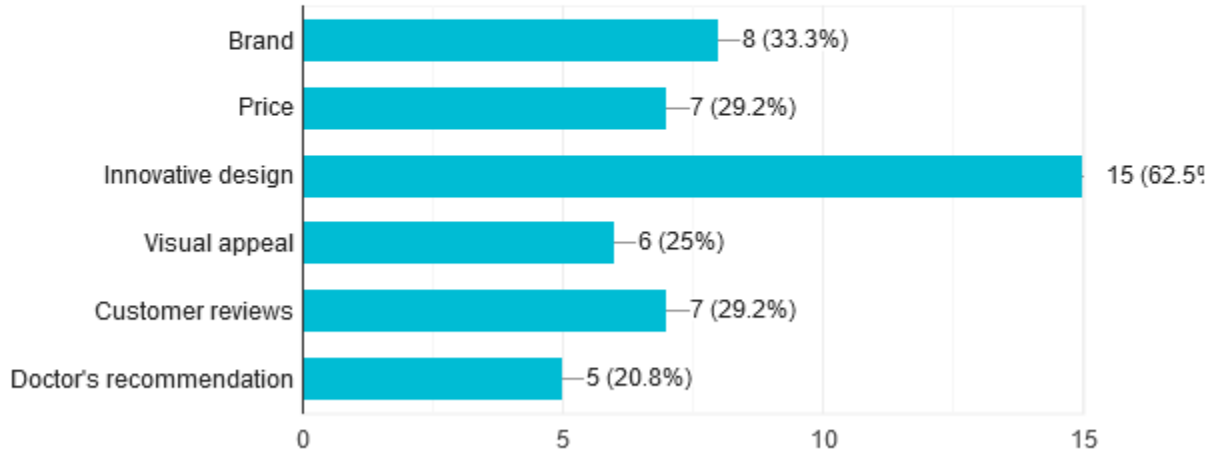
What are your biggest hestitations with regards to purchasing wearable technology? Please select 2

25 responses



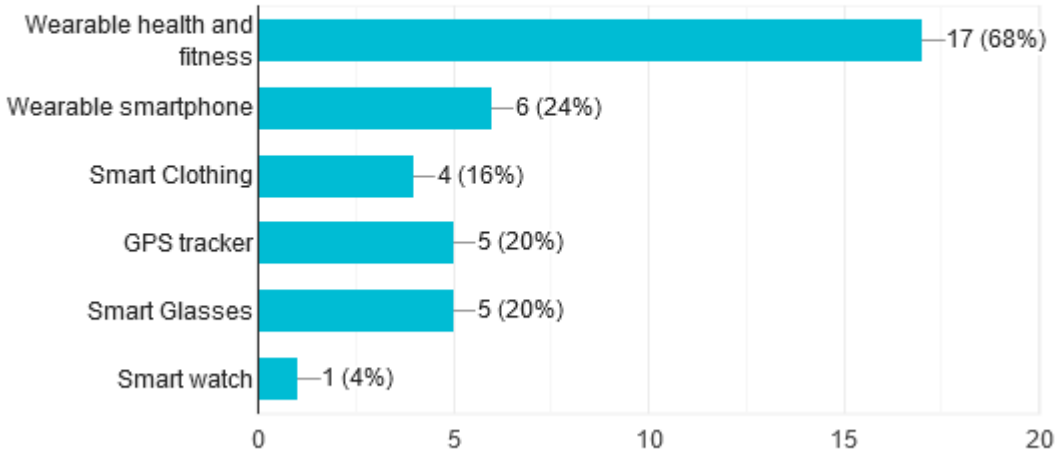
What features of wearable technology are most important to you? (You can select more than 2)

24 responses



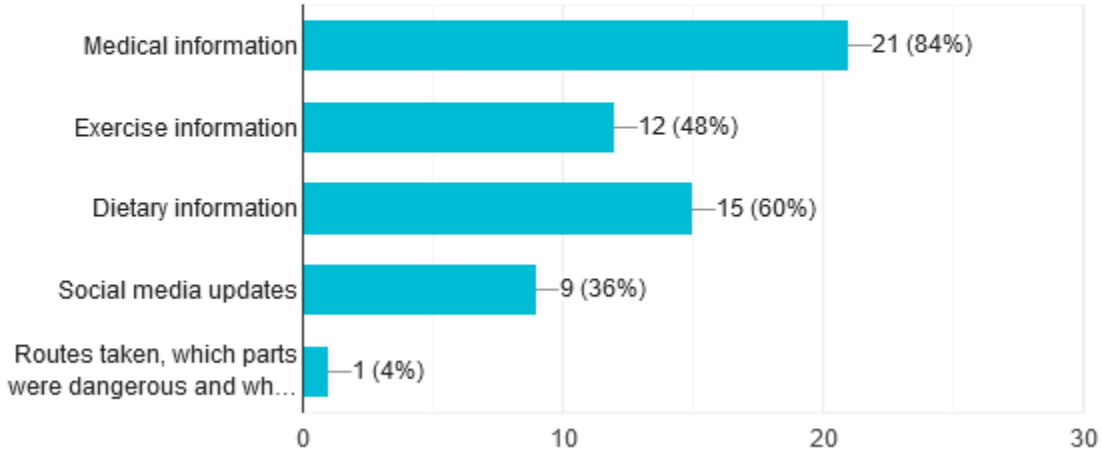
Which of the wearable devices would you be most interested in?

25 responses



What information would you personally want your wearable device to tell you? Please specify 3 options maximum

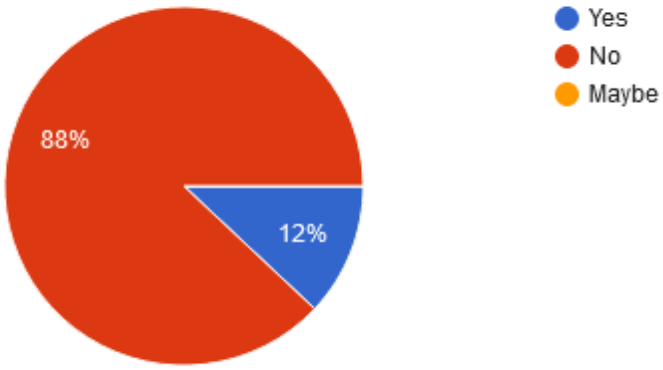
25 responses



Section II: Heart-Rate Wearable Device

Have you used a heart-rate wearable monitor before?

25 responses



If yes, specify which one

4 responses

N/A

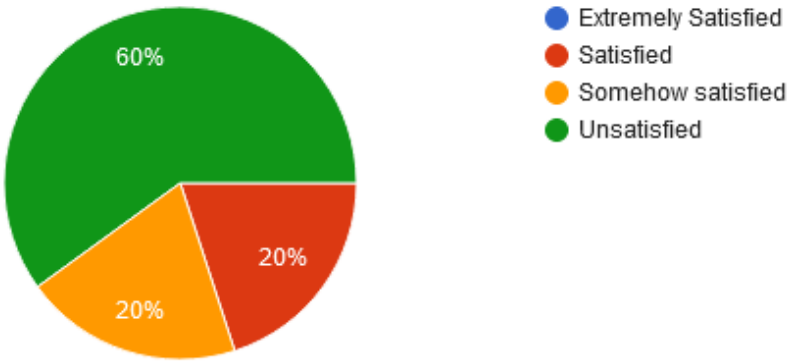
W26

smart watch

Health watch

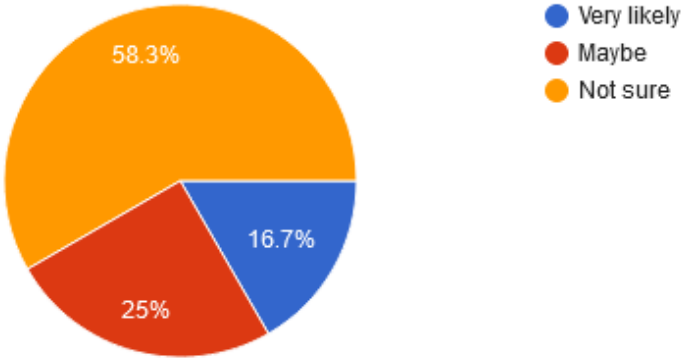
If yes, what was your overall experience with the device?

5 responses

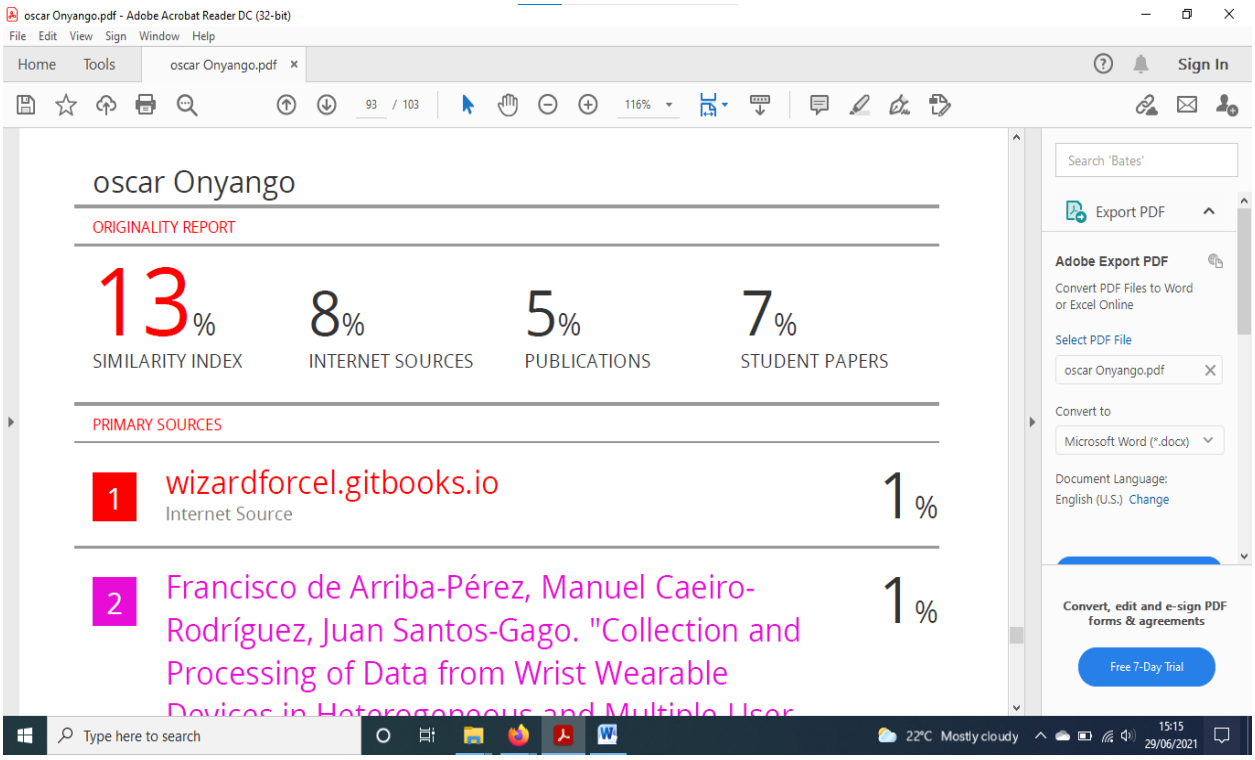


If no, how likely are you to purchase the following wearables in the next 12 months?

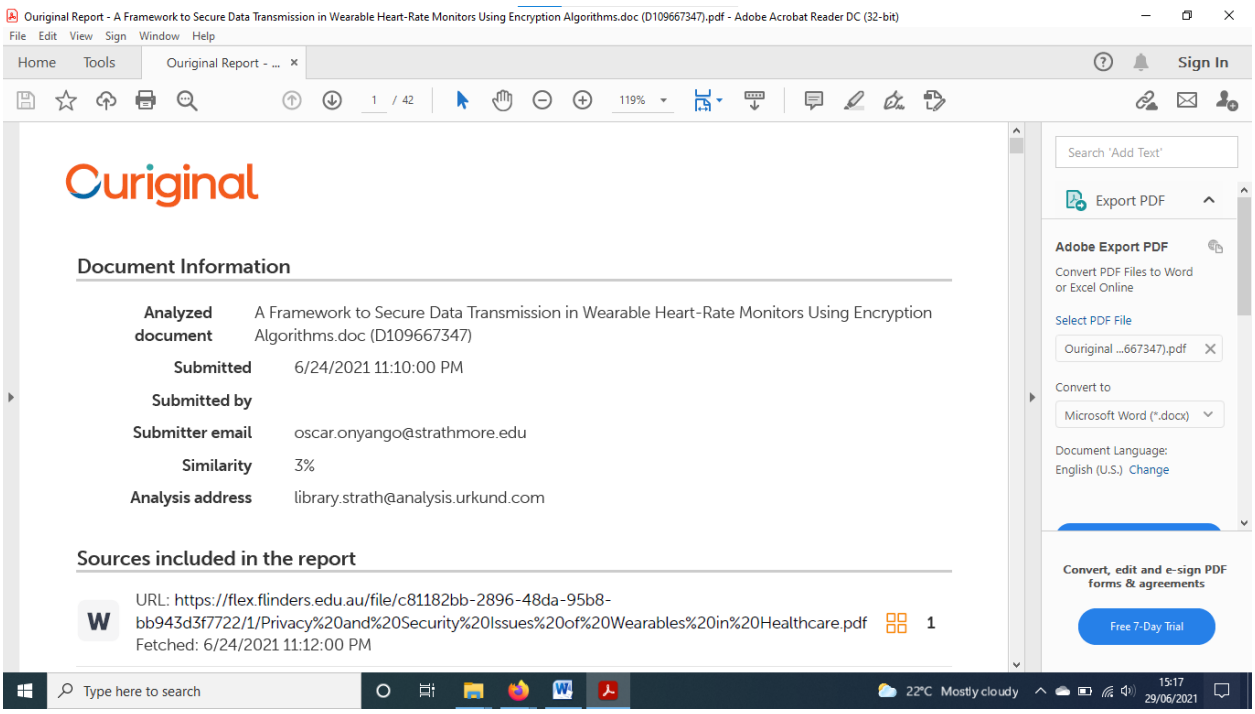
24 responses



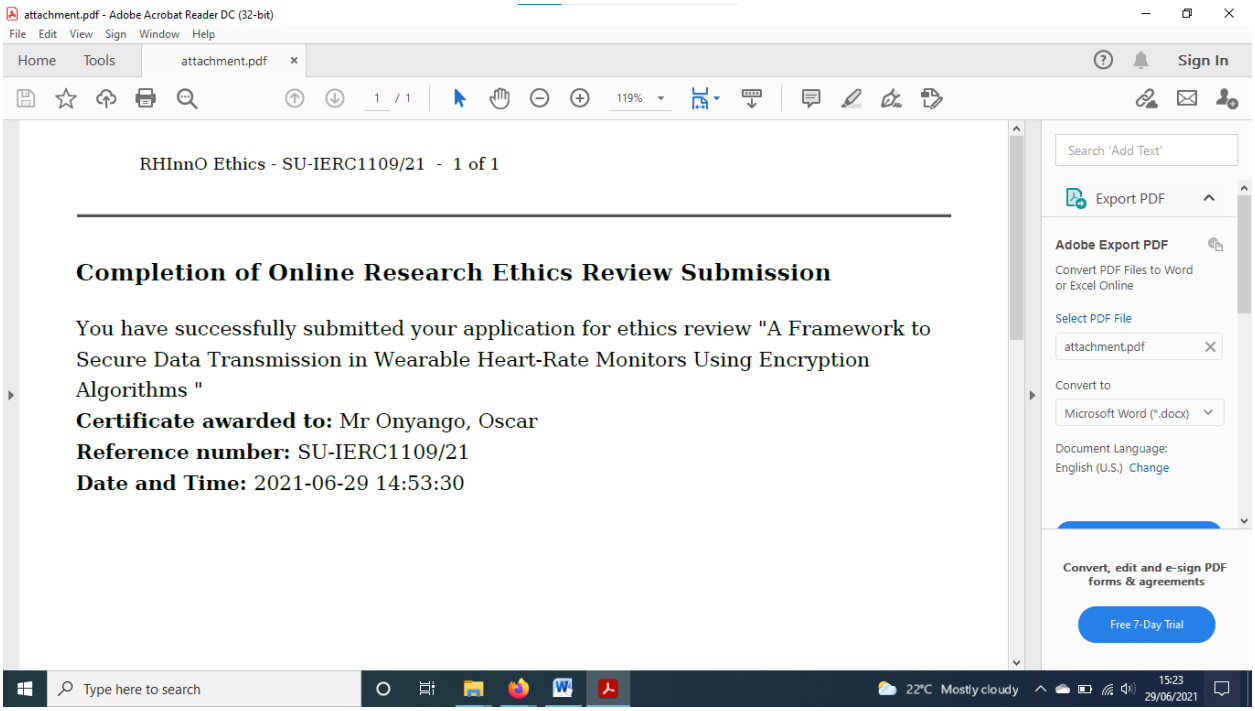
Appendix 3: Turnitin Report



Appendix 4: Ouriginal Report



Appendix 5: Ethical Review



attachment.pdf - Adobe Acrobat Reader DC (32-bit)

RHInnO Ethics - SU-IERC1109/21 - 1 of 1

Completion of Online Research Ethics Review Submission

You have successfully submitted your application for ethics review "A Framework to Secure Data Transmission in Wearable Heart-Rate Monitors Using Encryption Algorithms "

Certificate awarded to: Mr Onyango, Oscar
Reference number: SU-IERC1109/21
Date and Time: 2021-06-29 14:53:30

Appendix 6: Encryption Code

```
package android.kaisms;

import java.security.Key;

import java.util.ArrayList;

import javax.crypto.Cipher;

import javax.crypto.spec.SecretKeySpec;

import android.app.Activity;

import android.os.Bundle;

import android.telephony.SmsManager;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

public class EncDecSMSActivity extends Activity {
```

```
/** Called when the activity is first created. */
EditText recNum;
EditText secretKey;
EditText msgContent;
Button send;
Button cancel;
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
recNum = (EditText) findViewById(R.id.recNum);
secretKey = (EditText) findViewById(R.id.secretKey);
msgContent = (EditText) findViewById(R.id.msgContent);
send = (Button) findViewById(R.id.Send);
cancel = (Button) findViewById(R.id.cancel);
// finish the activity when click Cancel button
cancel.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {
finish();
}
});
// encrypt the message and send when click Send button
send.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {
String recNumString = recNum.getText().toString();
String secretKeyString = secretKey.getText().toString();
String msgContentString = msgContent.getText().toString();
// check for the validity of the user input
```

```
// key length should be 16 characters as defined by AES-128-bit
if (recNumString.length() > 0 && secretKeyString.length() > 0
&& msgContentString.length() > 0
&& secretKeyString.length() == 16) {
// encrypt the message
byte[] encryptedMsg = encryptSMS(secretKeyString,
msgContentString);
// convert the byte array to hex format in order for
// transmission
String msgString = byte2hex(encryptedMsg);
// send the message through SMS
sendSMS(recNumString, msgString);
// finish
finish();
} else
Toast.makeText(
getBaseContext(),
"Please enter phone number, secret key and the message. Secret key must be 16 characters!",
Toast.LENGTH_SHORT).show();
}

});
}
public static void sendSMS(String recNumString, String encryptedMsg) {
try {
// get a SmsManager
SmsManager smsManager = SmsManager.getDefault();
// Message may exceed 160 characters
// need to divide the message into multiples
```

```
ArrayList<String> parts = smsManager.divideMessage(encryptedMsg);
smsManager.sendMultipartTextMessage(recNumString, null, parts,
null, null);
} catch (Exception e) {
e.printStackTrace();
}
}
// utility function
public static String byte2hex(byte[] b) {
String hs = "";
String stmp = "";
for (int n = 0; n < b.length; n++) {
stmp = Integer.toHexString(b[n] & 0xFF);
if (stmp.length() == 1)
hs += ("0" + stmp);
else
hs += stmp;
}
return hs.toUpperCase();
}
// encryption function
public static byte[] encryptSMS(String secretKeyString,
String msgContentString) {
try {
byte[] returnArray;
// generate AES secret key from user input
Key key = generateKey(secretKeyString);
// specify the cipher algorithm using AES
Cipher c = Cipher.getInstance("AES");
// specify the encryption mode
```

```
c.init(Cipher.ENCRYPT_MODE, key);
// encrypt
returnArray = c.doFinal(msgContentString.getBytes());
return returnArray;
} catch (Exception e) {
e.printStackTrace();
byte[] returnArray = null;
return null;
}
}
private static Key generateKey(String secretKeyString) throws Exception {
// generate secret key from string
Key key = new SecretKeySpec(secretKeyString.getBytes(), "AES");
return key;
}
}
```

Appendix 7: Arduino Code to Send Data from NodeMCU to MQTT Serve

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <ESP8266WebServer.h>
#include "MAX30100_PulseOximeter.h"
const char* ssid = "SSD";
const char* password = "password";
const char* mqtt_server = "driver.cloudmqtt.com";
const char *mqtt_user = "kmsmamvp";
const char *mqtt_pass = "88AdSsQvLTk8";
```

```
PulseOximeter pox;
uint32_t tsLastReport = 0;

#define REPORTING_PERIOD_MS   3000
ESP8266WebServer server(80);
float BPM, SpO2;
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

void setup_wifi() {
  delay(100);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  server.on("/", handle_OnConnect);
  server.onNotFound(handle_NotFound);
  server.begin();
```

```
  Serial.println("HTTP server started");
  Serial.print("Initializing pulse oximeter..");
  if (!pox.begin()) {
    Serial.println("FAILED");
    for (;;);
  } else {
    Serial.println("SUCCESS");

  }

}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
        String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
       if (client.connect(clientId.c_str(), mqtt_user, mqtt_pass)) {
      Serial.println("connected");
        client.publish("outTopic", "hello world");
```

```
      client.subscribe("inTopic");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void setup() {
  pinMode(BUILTIN_LED, OUTPUT);     // Initialize the BUILTIN_LED pin as an output
  Serial.begin(9600);
  setup_wifi();
  client.setServer(mqtt_server, 18728);
  client.setCallback(callback);
  reconnect();
}



void loop() {

  if (!client.connected()) {
    reconnect();
  }

  client.loop();
  server.handleClient();
  pox.update();
```

```
  if (millis() - tsLastReport > REPORTING_PERIOD_MS) {

    BPM = pox.getHeartRate();
    SpO2 = pox.getSpO2();
    _pulseOximeterFunction(BPM, SpO2);
    Serial.print("BPM: ");
    Serial.println(BPM);

    Serial.print("SpO2: ");
    Serial.print(SpO2);
    Serial.println("%");

    Serial.println("*********************************");
    Serial.println();
    tsLastReport = millis();
  }

}

// Set up the PPulseometer to work
void _pulseOximeterFunction(float BPM, float SpO2 ){
  delay(1000);
//BPM
  Serial.print("Publish message: ");
  Serial.println(BPM);

   int numBPM = (int)BPM;
  char cstrBPM[16];
  itoa(numBPM, cstrBPM, 10);
```

```
//Sp02
  int numSpO2 = (int)SpO2;
   char cstrSpO2[16];
   itoa(numSpO2, cstrSpO2, 10);

  delay(1500);
  client.publish("bmp", cstrBPM);
  client.publish("sp01", cstrSpO2);
}

// Connectivity functions
void handle_OnConnect() {
  server.send(200, "text/html", SendHTML(BPM, SpO2));
}

void handle_NotFound() {
  server.send(404, "text/plain", "Not found");
}


// format and send the data recorded from the oximeter

String SendHTML(float BPM, float SpO2) {
  String ptr = "<!DOCTYPE html>";
  ptr += "<html>";
  ptr += "<head>";
  ptr += "<title>ESP8266 WebServer</title>";
  ptr += "<meta name='viewport' content='width=device-width, initial-scale=1.0'>";
  ptr += "<link rel='stylesheet' href='https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.7.2/css/all.min.css'>";
```

```
  ptr += "<link rel='stylesheet' type='text/css' href='styles.css'>";

  ptr += "<style>";

  ptr += "body { background-color: #fff; font-family: sans-serif; color: #333333; font: 14px Helvetica, sans-serif box-sizing: border-box;}";

  ptr += "#page { margin: 20px; background-color: #fff;}";

  ptr += ".container { height: inherit; padding-bottom: 20px;}";

  ptr += ".header { padding: 20px;}";

  ptr += ".header h1 { padding-bottom: 0.3em; color: #008080; font-size: 45px; font-weight: bold; font-family: Garmond, 'sans-serif'; text-align: center;}";

  ptr += "h2 { padding-bottom: 0.2em; border-bottom: 1px solid #eee; margin: 2px; text-align: left;}";

  ptr += ".header h3 { font-weight: bold; font-family: Arial, 'sans-serif'; font-size: 17px; color: #b6b6b6; text-align: center;}";

  ptr += ".box-full { padding: 20px; border 1px solid #ddd; border-radius: 1em 1em 1em 1em; box-shadow: 1px 7px 7px 1px rgba(0,0,0,0.4); background: #fff; margin: 20px; width: 300px;}";

  ptr += "@media (max-width: 494px) { #page { width: inherit; margin: 5px auto; } #content { padding: 1px;} .box-full { margin: 8px 8px 12px 8px; padding: 10px; width: inherit;; float: none; } }";

  ptr += "@media (min-width: 494px) and (max-width: 980px) { #page { width: 465px; margin 0 auto; } .box-full { width: 380px; } }";

  ptr += "@media (min-width: 980px) { #page { width: 930px; margin: auto; } }";

  ptr += ".sensor { margin: 12px 0px; font-size: 2.5rem;}";

  ptr += ".sensor-labels { font-size: 1rem; vertical-align: middle; padding-bottom: 15px;}";

  ptr += ".units { font-size: 1.2rem;}";

  ptr += "hr { height: 1px; color: #eee; background-color: #eee; border: none;}";

  ptr += "</style>";


  //Ajax Code Start

  ptr += "<script>\n";

  ptr += "setInterval(loadDoc,1000);\n";

  ptr += "function loadDoc() {\n";

  ptr += "var xhttp = new XMLHttpRequest();\n";

  ptr += "xhttp.onreadystatechange = function() {\n";

  ptr += "if (this.readyState == 4 && this.status == 200) {\n";
```

```
ptr += "document.body.innerHTML =this.responseText}\n";
ptr += "};\n";
ptr += "xhttp.open(\"GET\", \"/\", true);\n";
ptr += "xhttp.send();\n";
ptr += "}\n";
ptr += "</script>\n";
//Ajax Code END

ptr += "</head>";
ptr += "<body>";
ptr += "<div id='page'>";
ptr += "<div class='header'>";
ptr += "<h1>MAX30100 ESP8266 WebServer</h1>";
ptr += "<h3><a href='https://theiotprojects.com'>https://theiotprojects.com</a></h3>";
ptr += "</div>";
ptr += "<div id='content' align='center'>";
ptr += "<div class='box-full' align='left'>";
ptr += "<h2>Sensor Readings</h2>";
ptr += "<div class='sensors-container'>";

//For Heart Rate
ptr += "<p class='sensor'>";
ptr += "<i class='fas fa-heartbeat' style='color:#cc3300'></i>";
ptr += "<span class='sensor-labels'> Heart Rate </span>";
ptr += (int)BPM;
ptr += "<sup class='units'>BPM</sup>";
ptr += "</p>";
ptr += "<hr>";

//For Sp02
```

```
  ptr += "<p class='sensor'>";
  ptr += "<i class='fas fa-burn' style='color:#f7347a'></i>";
  ptr += "<span class='sensor-labels'> Sp02 </span>";
  ptr += (int)SpO2;
  ptr += "<sup class='units'>%</sup>";
  ptr += "</p>";
  ptr += "</div>";
  ptr += "</div>";
  ptr += "</div>";
  ptr += "</div>";
  ptr += "</div>";
  ptr += "</body>";
  ptr += "</html>";
  return ptr;
}
```