2020

Dynamic knowledge based authentication model for enhancing security of USSD banking transactions.

Njuguna, Michael Wanuna
Faculty of Information Technology
Strathmore University

Dynamic Knowledge Based Authentication Model for Enhancing Security of USSD Banking Transactions

By

Njuguna Michael

A Thesis Proposal Submitted in partial fulfilment of the requirements for the award of a Degree of Master of Science in Information Technology.

Faculty of Information Technology

Strathmore University

Nairobi, Kenya

July 2020

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.

Njuguna Michael

Signature_____

Date _____

Approval

The thesis of Njuguna Michael was reviewed and approved by the following:

Dr. Joseph Onderi Orero, PhD.

Dean, Faculty of Information Technology

Strathmore University

Dr. Bernard Shibwabo,

Director of Graduate Studies,

Strathmore University

Abstract

A large part of mobile banking transactions in Africa are facilitated by USSD technology. In authenticating customers, banks rely on a single security vector: a shared secret such as a six-digit PIN. This mechanism presents vulnerabilities that are commonly exploited to perpetuate fraud. In particular, this study focuses on insider threats, privacy leakage and social engineering attacks. To address these challenges, the study proposes a dynamic authentication model that poses diverse challenge questions based on available customer and transactional data. These challenge questions are unique to a given customer and variable over time making it difficult for anyone other than the legitimate user to deduce the correct response. A test-driven approach was used to guide development with the test scenario increasing in complexity after each iteration. Validation tests show the proposed scheme demonstrably provided enhanced security. The true acceptance score for legitimate users stood at 92.8 percent. As for guessing attacks by adversarial users, the probability of a correct guess was reduced to less than 0.08 percent. Performance-wise, the computational overhead increased by only 22 percent as compared to the classical method. This was sufficiently small as not to be noticeable by a user in real-world deployment. The study points to the feasibility of the model but recommends further research on challenge question generation for even greater security.

Keywords: **USSD, Banks, Mobile Network Operators, Privacy Leakage, Insider Threat, Social Engineering, Authentication, Mobile Money**.

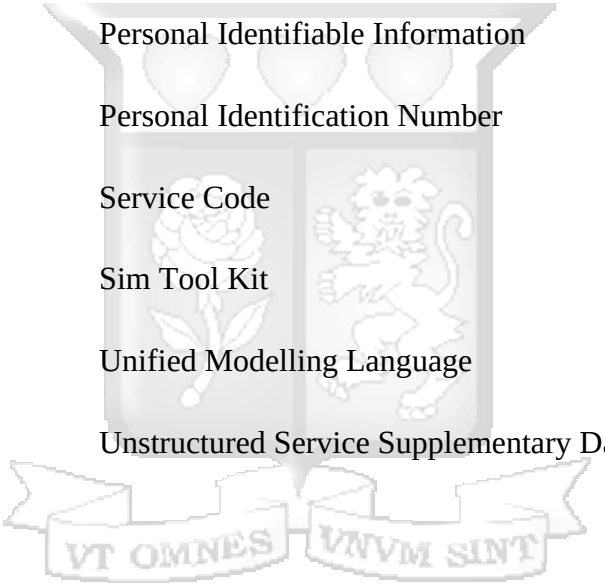
Table of Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

CRM	-	Customer Relationship Management
GSM	-	Global System for Mobile communication
KBA	-	Knowledge Based Authentication
MFIs	-	Micro Finance Institutions
MFSs	-	Mobile Financial Services
MNO	-	Mobile Network Operator
PII	-	Personal Identifiable Information
PIN	-	Personal Identification Number
SC	-	Service Code
STK	-	Sim Tool Kit
UML	-	Unified Modelling Language
USSD	-	Unstructured Service Supplementary Data



Definition of Terms

Bank-to-Customer Transaction – A transaction using a bank's mobi-services to transfer funds from the customer's bank account to a mobile money account (loose definition).

Knowledge-Based Authentication (KBA) - Techniques that test the knowledge of an individual in order to ascertain their claimed identity (Hastings & Dodson, 2004).

Dynamic Knowledge based authentication (Dynamic KBA) – A knowledge-based authentication mechanism that ascertains the identity of a user by testing his knowledge of information that applies only to him (IDology, 2019).

Mobile Money Account – A Sim Tool Kit Application that holds the digital funds of a customer. Funds can be 'deposited' to or 'withdrawn' from the account using the Mobile Network Operator's agent network (GSM Association, 2019).

Mobi-Services – A term used to refer to mobile-based banking services. In most cases, it refers to native mobile applications or USSD applications (GSM Association, 2019).

Personal Identifiable Information – Information that, when used alone or with other relevant data, can identify an individual (Ranchal, et al., 2010). It may be either sensitive (held in confidence) or non-sensitive (publicly accessible).

Privacy Leakage – The inadvertent revelation of confidential information resulting from a system's design or operation flaw (van Dijk & Haider, 2019).

Acknowledgements

I would like to express my sincerest appreciation to my research supervisor Dr. Joseph Orero, for his untiring support, encouragement and immense knowledge without which this research would not have been possible. My deepest thanks also go to Dr. Humphrey Njogu for his insightful comments and stimulating discussions.

Dedication

This work is dedicated to my family, a source of endless support.

Chapter 1: Introduction

1.1 Background of the Study

As of 2010, only 20% of Sub-Saharan Africans owned a bank account (Mlachila, Park, & Yabara, 2013). Prior to the remarkable growth of the African telecommunication sector, access to financial services for the majority remained beyond reach. However, as captured in the State of the Industry Report by GSM Association (2019), Africa has progressed to be the pace-setter in financial inclusivity. The report identifies convergence of key technologies and innovations on the part of Mobile Network Operators (MNOs) as key factors contributing to this success. In 2017 alone, mobile money transactions within the region were close to USD 20 billion. (GSM Association, 2018).

Growth of the mobile money services did not displace bank accounts. On the contrary, cooperation between banks and Mobile Network Operators led to the emergence of Mobile Financial Services (MFSs), enabling banks to reach even more customers through the convenience offered by mobile banking. The two channels facilitating mobile-based banking transactions are native applications and Unstructured Supplementary Service Data (USSD). GSM (2019) reports that over 90 per cent of transactions are driven by USSD.

Despite the best efforts of Mobile Network Operators and banks to secure Mobile Financial Service deployments, occurrences of exploits continue to persist. Occupational fraud, fraud committed against an organization by its own officers, is on the rise in the African banking sector. In ranking incidents of occupational fraud reported in 2018, the three largest mobile money ecosystems (South Africa, Nigeria and Kenya) accounted for nearly 95% of all the cases (Association of Certified Fraud Examiners, 2018). The risk posed by insiders varies depending on thier role within the organization. For example, a malicious server administrator, network engineer or privileged insider within the bank could ostensibly monitor access patterns to database records as USSD requests were being handled, and through traffic analysis reconstruct the database contents even in cases where the records are encrypted (Abdelraheem, Gehrmann, Andersson, & Glackin, 2018). In doing so, the malevolent insider would be able to retrieve sensitive Personal Identifiable Information (PII) such as customer phone numbers and mobile banking PINs.

Insider threats to mobile banking deployments are not limited to officers of a bank. USSD banking applications must also necessarily interface with the core GSM network of Mobile Network Operators; a consequence of which is the exposure of customer information to a third-party. As a mobile user interacts with a USSD code, any data he inputs is transparent to the Mobile Station (MS) and Base Station Subsystem (BSS) so that the USSD handler can service the request. Thus, it means that data passes through these entities in its original (plaintext) form and is not modified in any way (Taskin, 2012). It is entirely within the capability of privileged insiders within the Mobile Network Operator to retrieve, in plaintext, the data transmitted during USSD sessions and recover sensitive PII.

Besides insider threats, banks also cite social engineering attacks among the top five challenges that banks encounter with respect to fraud (KPMG, 2019). The scammer's aim is the acquisition of personal information from his victim using which he can perform an account take-over and access the victim's mobile money account or bank account.

1.2 Problem Statement

USSD applications, as used in mobile banking, implement what is known *static* knowledge-based authentication (KBA), which relies upon an unchanging piece of knowledge such as a password or PIN (Skračić, Pale, & Jeren, 2014). In this scheme, security is guaranteed only insofar as the piece of knowledge can remain exclusively secret to the rightful owner.

The first threat posed by the use of static KBA as the sole means of identity verification is privacy leakage. As discussed earlier, it is the unintended exposure of database information to external observers (Islam, Kuzu, & Kantarcioglu, 2012). Malevolent insiders stationed at either the bank or Mobile Network Operator may exploit their privileged positions to harvest customer personally identifiable information (PII) used for authentication in USSD sessions. Secondly, the rising number of social engineering cases where scammers induce customers to reveal their secret PIN such as by impersonating a person of authority from the bank or Network Operator is major cause for concern. In monetary terms, social engineering accounts for fifteen per cent (USD 21 million.) lost as a consequence of cyber-crime in Kenya (Serianu, 2017).

The above threats all exploit the same inherent vulnerability of static KBA: knowledge of a single piece of information is enough to be successfully authenticated. Several scholars such as Okamoto (2013) and Fatima, Siddiqui, Umar, & Khan (2019) have proposed a variety of techniques to improve on static KBA by making the "shared secret" dynamic or variable with time. Different from past studies, this research endeavours to automate the process of challenge question generation. Also, no attempt has been made to extend the approaches to a model that satisfies the real-time requirements of USSD-mediated banking transactions given that USSD sessions remain active for a maximum duration of only two minutes. The resulting research gap is what this study seeks to fill; and in doing so, offer a practical approach that would be beneficial to the industry.

1.3 General Objective

To design and prototype a two-factor authentication mechanism that implements both static and dynamic knowledge-based authentication for use by USSD applications in the banking sector to address the security concerns arising from privacy leakage, insider threat and social engineering attacks.

1.4 Specific Objectives

i. To assess the challenges of the present implementation of USSD within the Kenyan banking sector.
ii. To analyse the vulnerabilities of the present PIN-based authentication mechanism with respect to privacy leakage, insider threat and social engineering.
iii. To develop a dynamic, knowledge-based authentication mechanism that addresses vulnerabilities arising from privacy leakage, insider threat and social engineering.
iv. To test the efficacy of the proposed authentication mechanism.

1.5 Research Questions

i. What are the challenges arising from the use of USSD bank applications in Kenya?
ii. In what ways can privacy leakage, insider threat and social engineering compromise the secrecy of a customer's mobile banking PIN?
iii. Which techniques can be used to design an efficient, dynamic, knowledge-based authentication mechanism that addresses the described vulnerabilities?

iv. How can the effectiveness of the proposed technique be measured?

1.6 Justification

The study will be of use to banks operating within Kenya and Africa. It will assist the organizations combat the rising cases of fraud occasioned by social engineering attacks and unauthorized access to sensitive Personal Identifiable Information (PII) by privileged insiders or third parties. Additionally, in sending a message that securing financial transactions across all channels is a concern they take seriously, banks can expect increased customer confidence and reputation gains.

It will also certainly be of use to policy makers such as central banks, oversight bodies and banking associations in developing security standards pertaining to USSD-mediated banking transactions.

1.7 Scope of the Study

This study primarily focused on the banking sector in Kenya, where USSD accounts for 90% of mobile-based, bank-to-customer transactions. In designing the solution, the researcher took into consideration chiefly the requirements of the Kenyan ecosystem.

Owing to the sensitivity and proprietary nature of information relating to bank internal processes and records, the researcher was constrained to the use of dummy customer data that may not have fully represented the diverse characteristics of a real data set.

The rest of the document is structured as follows: chapter 2 reviews past studies that expound on the problem statement, chapter 3 describes the methodology selected to address the problem, chapter 4 outlines the proposed design of a prototype, chapter 5 covers the implementation and validation aspects of the protype, chapter 6 analyses key findings of the validation results and chapter 7 concludes the work in addition to providing recommendations.

Chapter 2: Literature Review

2.1 Overview

This chapter sought to the analyse the present state of USSD technology as used in banking so as to provide the necessary context informing the study. We explore the evolution of mobile money in Kenya, taking a look at the unique features of this ecosystem. The history of USSD technology is then discussed to make clear the reasons why it is in use today to facility mobile banking transactions. Lastly, we critically investigate the vulnerabilities posed by current USSD deployments and delve into how these can be exploited to commit fraud.

2.2 The Evolution of Mobile Money in Kenya

The term 'mobile money', is taken to mean access to financial services through the use of a mobile device (GSM Association, 2019). The three years following the launch of *Mpesa* were marked by a rapid rise in the number of mobile money account holders, who made use of Sim Tool Kit (STK) applications and the Mobile Network Operator's agent network to 'withdraw from' and 'deposit to' their mobile money wallets (Asongu, 2018).

At maturity of the technology, there was "buy-in" from banks who proceeded to partner with the Mobile Network Operators to facilitate customer-to-bank and bank-to-customer transactions using the mobile money infrastructure. The Communications Authority of Kenya (2018) reported that in the period covering July 2018 to September 2018, Kenya's mobile money transactions were valued at just over KES. 2 trillion (≃ USD 20 billion). Going by the industry report released by GSM (2019), the movement of close to USD 18 billion. (90%) was facilitated by USSD channels.

2.3 USSD Technology

A fundamental limiting factor on the use of mobile devices in the early 1990s was power storage or battery life. The GSM technology available at the time (3G network) established persistent connections between mobile devices and the Mobile Network Operator to facilitate transfer of network messages. Periodically, keep-alive messages (otherwise referred to as heart beat messages) were sent to prevent time-outs and maintain the active session.

This, besides clogging up the communication channel with heart beat messages, had the effect of rapidly depleting the battery life of handsets. Astrom Bo and Svennesonn Bjorn (1994) addressed these challenges by developing a scheme that utilized the voice channel on the GSM network to provide text-based interaction between the Mobile Network Operator and mobile device in short-lived, isolated sessions (US Patent No. US5752188A, 1994). Figure 2.1 illustrates the how USSD was first envisioned to interact with network elements. Much of the process flow has remained unchanged to date.

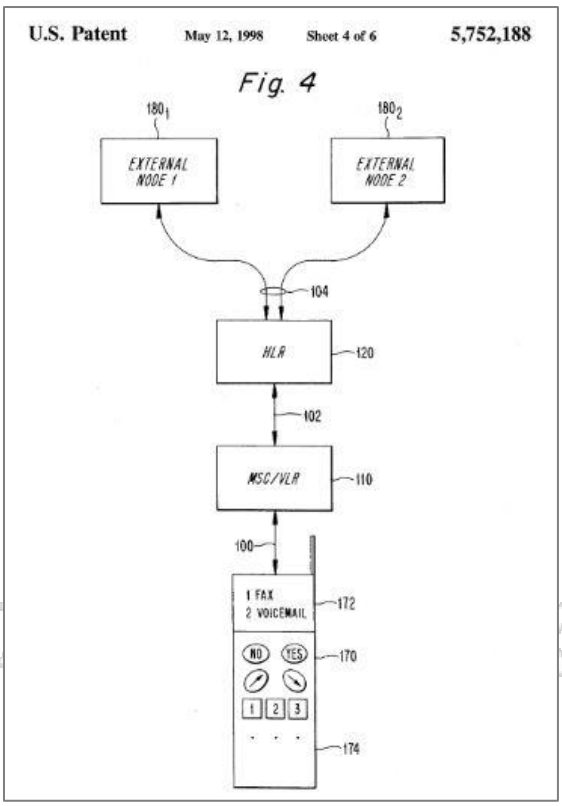


Figure 2.1: USSD Interaction with Network Elements (US Patent No. US5752188A, 1994).

2.3.1 USSD Architecture

Nyamtiga, Anael & Laizer (2013) summarized USSD as: a message routing scheme used to facilitate communication between GSM handsets, the Mobile Network Operator and any other external application in the form of session-based requests and responses.

Figure 2.2 depicts a high-level abstraction of how a USSD string travels from a mobile device, through the core GSM network to an external application server.

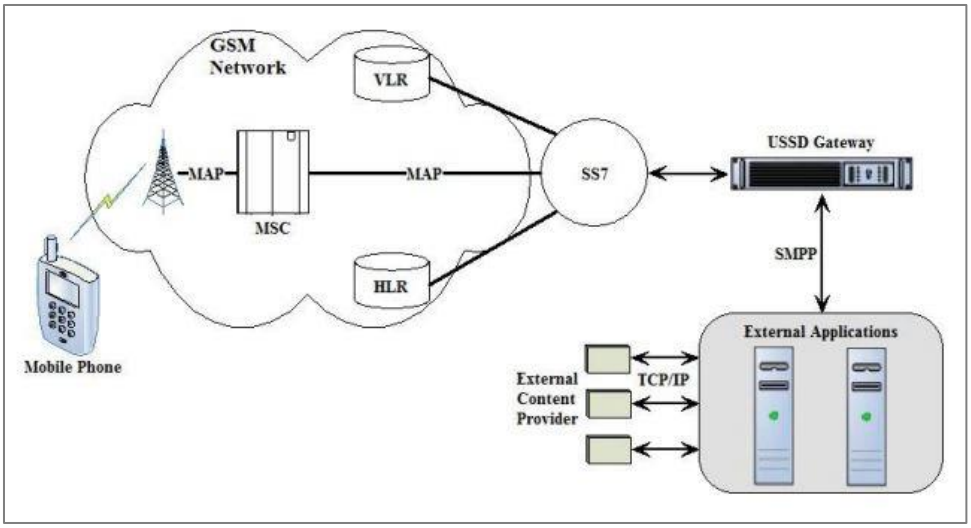


Figure 2.2: Abstracted USSD Operation (Nyamtiga, Anael , & Laizer, 2013).

As specified in US Patent No. US5752188A (1994), the operation of USSD is as follows: the Mobile Network Operator first generates a Service Code (SC) which users can use to access a USSD service. Upon dialling the SC on a GSM handset, it is interpreted as a USSD code and the contents of the message are passed via a signalling channel to the USSD application located at the Mobile Switching Centre (MSC) network element. At the MSC, a lookup if first performed on the Visitor Location Register (VLR) and if the USSD is unrecognized a subsequent look up is performed on the Home Location Register (HLR). The USSD application then performs the desired action(s) such as initiating a menu-based response to the user. Much of the dynamism in functionality afforded by USSD is attributable to the ability of the USSD handler to direct requests to either services defined in the HLR or, more pertinent to this study, to an external network node (such as a gateway to USSD banking applications).

2.3.2 USSD Usage in Kenya

A bank typically has a unique USSD code which users dial in order to access the bank's *mobi-services*. Practical considerations require that the user first registers his number with the bank (to facilitate user authentication at the start of each session) in order to enjoy the USSD service.

Upon dialling the USSD code of the bank, the customer is presented with a series of text-based menus (see Figure 2.3) to guide him complete a transaction.

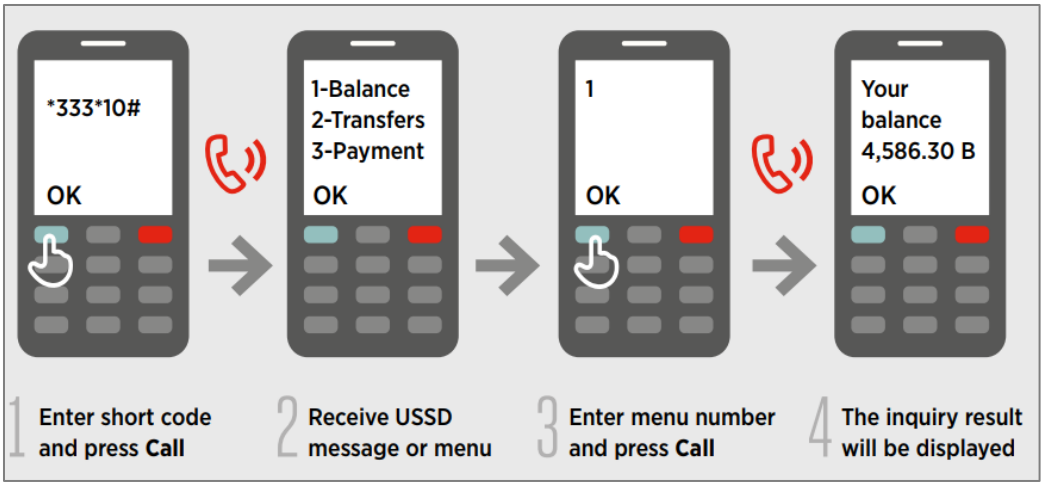


Figure 2.3: USSD Text-Based Menus (GSM Association, 2018).

Once authenticated (by providing their secret PIN), the user can access his mobile money account and perform actions such as balance enquiry, funds transfer and bill payment.

Table 2.1 contains the USSD codes for some of the largest financial institutions in Kenya.

Table 2.1: Example of Bank USSD Codes in Kenya (Safaricom PLC, 2020.).

Bank	USSD Code
KCB	*522#
Co-operative Bank	*667#
Standard Chartered Bank	*722#
Barclays Bank K LTD	*224#

2.3.3 Financial Grade Security and USSD

Though ubiquitous within Africa, USSD is a relatively recent entrant onto the banking scene. Owing to their relative novelty, bank-related USSD deployments have yet to undergo rigorous vulnerability testing or risk profiling - leaving them susceptible to the "less known" or scarcely researched exploits (Buku & Mazer, 2017).

The threat posed is further heightened by the fact that the banking sector is already vigorously contending with the menacing challenge of fraud, which is even more marked in ecosystems where Mobile Financial Services (MFSs) have matured. PWC (as cited in

Kiragu, 2015) found East African banks to be at a significantly higher risk of fraud in comparison its peers in neighbouring regions. In ranking various risks in order of severity, East African banks placed fraud at position 2 out of 25 as compared to the global industry average of 15 out of 25. The research points to an increase in the use of technology without sufficient risk management expertise as one of the significant causes.

2.4 Insider Threat

Insider threats, those caused by internal members of an organization, are perhaps the most damaging yet difficult to detect. The IBM X-Force threat intelligence index for 2018 established that 66% of compromised data resulted from inadvertent insiders (IBM, 2018). Even more alarming, IBM found that 60% of cyber threats were found be directly attributable to insider threats. The exploit may be occasioned inadvertently or deliberately, and while the latter may be rarer in occurrence it is certainly just as costly in effect.

A 2018 survey of over four hundred thousand cyber security professionals was compiled and found that 90% of organizations felt vulnerable to insider threat with the three top predisposing risk factors being: superfluous access privileges, many more devices holding sensitive data and the increased intricacy of the I.T landscape (Cyber Security Insiders & Crowd Research Partners, 2018). Figure 2.5 provides a summary of the biggest insider threats facing an organization.

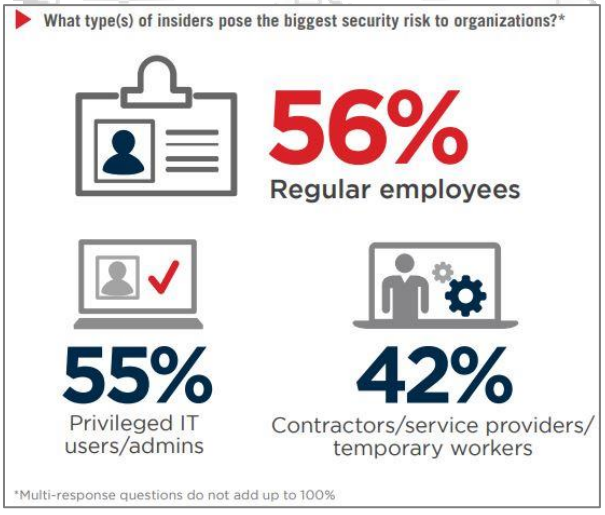


Figure 2.4: Organization Insider Threat Risks (Cyber Security Insiders & Crowd Research Partners, 2018).

Data is no longer just an IT asset, but a strategic tool. Disgruntled employees or insiders eyeing financial gain may maliciously access privileged information for fraudulent use at the organization's expense. The 2018 Insider Threat 2018 report by Cyber Security Insiders and Crowd Research Partners indicates that many organizations, being aware of the possible ramifications of insider threat, are beginning to take proactive steps to counter it. A combination of Data Loss Prevention (DLP), access management, encryption and log management techniques are being employed to develop a comprehensive security program that provides sufficient data protection mechanisms. As can be seen in Figure 2.6, sensitive personal information (PII) is targeted 49% of the time.

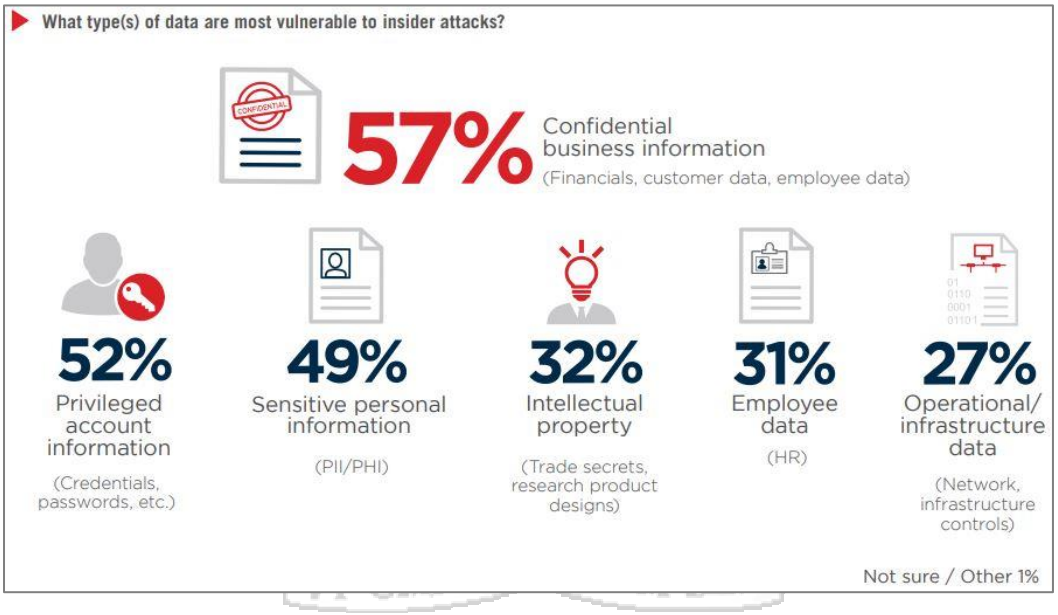


Figure 2.5: Data most vulnerable to Insider Attacks (Cyber Security Insiders & Crowd Research Partners, 2018).

Interestingly, the report indicated organizations attributed 55% of insider threat risk to privileged insiders such as I.T or server administrators (see Figure 2.5). Their unique position within the security matrix of an organization provides them an opportunity to attempt unauthorized access to sensitive resources, and in some cases cover their tracks.

As more and more organizations outsource their data to the cloud, there is a new type of insider: third-party cloud service providers. Their unfettered access to sensitive network and database resources and could ostensibly result in compromised confidentiality of data they are tasked to host.

2.5 Privacy Leakage

For banks to process USSD requests made by their clients, they employ the use of application servers that communicate with the Mobile Network Operator's USSD gateway (Nyamtiga, Anael , & Laizer, 2013). Usually, the accepted trusted boundary for such a set up extends to the server itself; that is, it is assumed that the deployment environment is secure and where computing has been outsourced, that the cloud server is curious but honest. By this we mean that though the set protocols will generally be followed by the server, it will still try to find out as much information as possible on data and its context based on received inputs. (Raykova, Zhao, & Bellovin, 2012)

In addressing the concern of privacy leakage, we shall narrow our focus to strong adversaries. As described by Van Djik & Heider (2019), hacks by strong adversaries are perpetuated by an insider with some arbitrary background knowledge of the server as well as some level of physical or remote access.

Even when the database is encrypted, which is typically the case for sensitive bank applications, strong adversaries can still infer non-trivial data from analysis of traffic. This has been demonstrated by scholars such as (Liu, Zhu, Wang, & Tan, 2014) and (Islam, Kuzu, & Kantarcioglu, 2012) who were able to derive user queries and secret keys solely by observing read access patters. Remote adversaries too pose a considerable threat as was correctly hypothesized by as van Dijk & Haider (2019). By montioring the network, a remote hacker does have the ability to "learn" an applications write-access patters and reconstruct non-trivial information from an encrypted database. John, Haider, Omar, & Van Dijk (2017) demonstrated such an exploit using the reknowned Montgomery's ladder technique. Using a compromised Direct Memory Access (DMA) attack where just the write-access patterns on hardware with compromised firmware are observed, a 512-bit secret key is correctly inferred in three and a half minutes. Several comparison of snapshots before and after write operations were compared revealing a precise pattern of memory access. It is entirely possible for both types of adverseries to succesfully reconstruct PII (such as customer phone numbers, hashed PINs, corresponsding salts etc.) by monitoring the read/write access patters on a bank's application server as it handles USSD requests from clients.

2.6 Cracking of Passwords and PINs

Salting of hashes was a technique developed in response to the rapid increase of processing power that significantly reduced the effective computation time of brute force attacks (Borde, Hebare, & Dhanedhar, 2017). Salting adds "randomness" to a hashed string which increases the difficulty of brute force attacks by several orders of magnitude. Today, it is standard practice for passwords and PINs to be hashed by a 32-bit string or larger. Unfortunately, in many database setups, the conceptual relation between hashed passwords/PINs and their salts lead them to being stored together; making it quite rare to know one and not the other.

With an adversary in possession of a hashed PIN and its corresponding salt by exploiting privacy leakage, it remains a matter of little difficulty to crack the hash and obtain the plaintext PIN. Open source applications such as *Jack the Ripper* or the more powerful *Hashcat* can be employed to this end. Given the fixed length and narrow domain of mobile banking PINs (6 digits), a brute force attack on thousands of records could be performed in just a matter of hours with the processing power of a current generation CPU/GPU desktop setup (Hashcat, 2019).

Figure 2.6 illustrates pseudocode performing a brute force attack on PINs hashed with the MD5 algorithm. Given the domain, the algorithm iterates through all possible values, either appending or pre-pending the salt and hashing the result to compare it to the retrieved, hashed password.

2.7 Social Engineering Attacks

Why are humans the weakest link in security? Cognitive biases, fear, curiosity, forgetfulness and docility to authority are some examples of human weaknesses that make us especially susceptible to social engineering attacks. (Mitnick & Simon, 2011). As tightened security controls make technical exploits more difficult to execute, criminals are turning to social engineering techniques in order to capitalize on human error and human weaknesses (Guangxuan , et al., 2015).

BRUTE FORCE ALGORITHM PSEUDOCODE

Algorithm: Check Match

Input: MaxRange: r, Type: t, Salt: s, Recovered hashed PIN: e.

Output: *True* if match to salted password is found. *False* otherwise.

```
for ($i = 0; $i <= $r; $i++) { //set domain (0 to r) of PINs
    $try = str_pad($i, 6, 0, STR_PAD_LEFT); //pad guesses of less than 6 digits with zeros on the left side
    if($t == "prepend") //if type t is prepend, add the salt in front of the guess
    {
        $salted_pin = $s.$try;
    }
    else //else, append the salt after the guess
    {
        $salted_pin = $try.$s;
    }
    $hash = md5($salted_pin); //hash the result of the guess and salt with an algorithm of choice
    if($salted_pin == $e)
    {
        return TRUE; //return true if there is a match, i.e. correct PIN has been guessed
    }
    else
    {
        return FALSE; //return false if no match
    }
}
```

Figure 2.6: Brute Force Attack Pseudo Code.

Even more alarming, the increased sharing of photos and personal information on social media provides the fraudster ample opportunity to reconnoitre his victim and piece together meaningful information based off publicly shared data (Khidzir, et al., 2016). The example below, drawn from templates developed by Mouton, Leenen and Venter (2016), illustrates how a social engineering exploit is typically performed.

Say *Earl*, a fraudster, wishes to perform an account takeover on his victim *Andrew*. On performing a cursory observation of *Andrew's* Twitter profile, *Earl* finds that the account is private and he can only glean that Andrew is a resident of **Mombasa**. *Earl* then creates a new twitter account, looks through the accounts that follow *Andrew* and chooses twenty of them to send "Follow Requests". Of these twenty, sixteen are accepted within the week. Next, *Earl* sends *Andrew* a "Follow Request". On seeing that *Earl* has some connections in common with him, *Andrew* accepts *Earl's* request. Going through *Andrew's* past tweets,

Earl notices that *Andrew* had tweeted to ***XYZ Bank*** a few days prior regarding a mobile banking transaction that was yet to reflect on his account. He also comes across a tweet in which Andrew advertises his new lorry business and provides his **mobile number** for enquiries. Given this information, *Earl* downloads the mobile banking app of *XYZ bank* proceeds to register an account in order to explore the available functionalities. He observes that whenever he tries to login onto the app with a different device (whether successful or not), he receives a text message from the bank informing him that the device associated to the account has changed and that in order to successfully authenticate it he needs to use the "Link" button to request for a One-Time Password (OTP).

Earl then purchases a new SIM card, and gets some of his friends with the *Truecaller* app to save the new number on their phones as "XYZ Bank, Fraud Department". The following Monday at around 8:30 am, when most people are battling morning traffic, he gives *Andrew* a call. It goes something like this:

> ***Earl***: Good Morning. This is Brian. I am calling from XYZ Bank. Am I speaking to Andrew?
>
> ***Andrew***: Yes, this is he. XYZ Bank you say? I have called XYZ Bank before; this is not the number they usually use.
>
> ***Earl***: You are right. I'm calling from the Fraud Prevention Department; we have different lines. Do you have a moment to speak?
>
> ***Andrew***: Fraud Prevention? I am actually on my way to work. Can you call again in a few minutes?
>
> ***Earl***: I am sorry Sir, but it is urgent. We've observed unusual activity from your mobile money account. Someone has been unsuccessfully trying to withdraw money from it all morning. We can see the offending device is in Nairobi.
>
> ***Andrew***: Withdrawal? Nairobi? But I'm in Mombasa!

Earl: Yes, I can see that from our records, which is why we have called. What we need to do now it disable your account to prevent withdrawals over the phone and you then can present yourself to any of our banking halls to sort this out.

Andrew: Go ahead and do that then. I'm on my way to work, but I will visit my nearest branch today.

Earl: Good. What I will do now is send an unlink request.

[Earl then attempts to login to the mobile banking app using Andrew's phone number. This results in Andrew receiving a text from XYZ bank informing him that his device has changed and that he needs to request an OTP.]

Earl: We've sent you a message. Have you received it?

[Andrew hears a beep and sees and incoming message from XYZ bank]

Andrew: Yes. I can see a message from the bank.

Earl: Good, the important thing now is to finish temporarily disabling your account and prevent withdrawals until you can get to the bank and sort this out. We require your approval to do so. Kindly confirm your registered phone number.

Andrew: It is this one that you have called. 0721 123123.

Earl: Okay. And your mobile banking PIN?

[Andrew knows he should not ordinarily share his banking PIN, but this is an emergency. His account is about to be emptied and the caller sounds like a legitimate bank employee. Truecaller showed his Caller ID as belonging to XYZ Bank and he even received a text message from the bank's Sender ID. Andrew decides to share his banking PIN this just this one time.]

Andrew: It is 454454.

Earl: Great. Your account has been successfully disabled. Make your way to the nearest branch for further assistance. Your reference number is TR20145. Do you have any other queries?

Andrew: I do, but I am in a bus. Let me get to the office where I'll be able to talk.

Earl: Alight Andrew. Good Day.

In the above example, the criminal was able to get useful information from seemingly disparate data that the victim had shared on his social media account. The criminal then employed a combination of impersonation (of an authority figure), fear and fake credibility in inducing the victim to reveal information that should remain secret to him. There are countless other anecdotes that can be given on criminals employing a variety of phycological techniques to beguile their victims.

2.8 SIM-Based Fraud

The security principle of unique possession can be defined as "that which you must have in order to access something" (Ricks, Ricks, & Dingle, 2014). In the security model adopted by the banks, the SIM card constitutes unique possession; and since the fraudster now has the victim's PIN, it now only remains acquire his mobile number.

The two main techniques criminal gangs have used to hijack mobile numbers are SIM swaps (which utilize on social engineering exploits) and SIM spoofing (which requires advanced technological expertise).

2.8.1 SIM Swap

In Africa, virtually all mobile money accounts are linked to a customer's phone number. This, doubtless, has been the motivating factor behind the increased efforts by fraudsters to attempt illegal sim swaps. In Kenya, it was reported that 57% of banking fraud was perpetuated on mobile money channels and that 90% of banks identified sim-swaps as an acute issue plaguing the sector (Myriad Connect, 2018).

In summary, it is performed when a fraudster acquires personal information about his victim, using which he is able to convince the network operator of his assumed identity and request that the victims phone number be ported to a new SIM card that the fraudster

possesses (Myriad Connect, 2018). As soon as this is done, the victims SIM card loses its connection to the GSM network and it may take hours for the victim to realise this. At the same time, any USSD requests made with the fraudsters SIM card will appear to the bank's USSD application server as coming from the legitimate customer. It will therefore allow transfer of funds out the the unsuspecting victim's bank account.

Should the bank also use a One Time Password (OTP) in addtion to the secret PIN, this message will be sent to the fraudsters new SIM card. In this way, all the safeguards put in place by the bank will have been successfully by-passed.

In response to the increasing number of SIM-swap related fraud, the MFSs industry devised means of combating the threat. In Kenya legislation was enacted mandating Mobile Network Operators to keep the particulars (names, national ID, birth certificate etc.) of each of its subscribers (Communications Authority of Kenya, 2015). To perform a SIM swap, a customer now has to present one of these documents as proof of identify. Market leader Safaricom, so as to reduce the additional customer service overhead and inefficiencies that the requirement introduced, adopted a voice biometric system dubbed *Jitambulishe* that provided customers the option of using their voice as a mode of authentication for services such as SIM swaps (Safaricom PLC, 2019).

Other countries have adopted more elaborate mechanisms. In Mozambique, for example, banks query a SIM-swap API each time a customer wants to perform a mobile banking transaction (Assolini & Tenreiro, 2019). The API flags SIM-cards that have been swapped within a pre-defined period; such as the preceding forty-eight hours. The bank, guided by its internal policies, then decides whether to proceed or abort a transaction initiated by a "suspect" sim-card. In most cases, the 48-hour window is enough time for the unsuspecting victim to become aware of the SIM-swap and take action. However, SIM-swaps in which face-to-face verification was performed (i.e. the customer physically presented himself) are not flagged as "suspect" (Sapovadia, 2018).

2.8.2 SIM Cloning

Given the increased controls on SIM-swaps, exploits based on the weaknesses of GSM are gaining increasing attention.

Authentication within the GSM network is based on a shared secret (otherwise known a cipher key: *kc*) between the network operator and SIM card (Kiran & Krishna, 2014). For security reasons, the cipher key never leaves the SIM. Instead, a challenge-response scheme is employed to facilitate authentication. The authentication server, that has all the cipher keys for SIM cards on the network, sends the SIM card a random challenge computed using the A5 and A8 hashing algorithms. With its cipher key, the SIM card computes a response also using the A5 and A8 hashing algorithms. If the response is as expected, the authentication server has verified the identity of the SIM card (Kiran & Krishna, 2014). The process is depicted in figure 2.7.

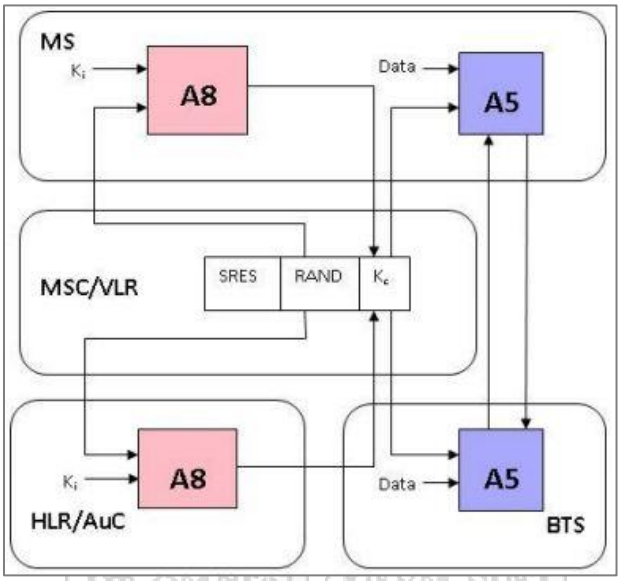


Figure 2.7: GSM Encryption Architecture, Kiran & Krishna (2014).

GSM is considered secure only so far as it is infeasible to obtain a SIM card's secret cipher. Edlund (2009), elaborates on some of the attack-resistance features embedded within industry-standard SIM cards such as the self-destruct feature that triggers when the SIM card is probbed. Therefore, from a physical security stand-point, security has been generally upheld given that is is both very difficult to extract the cipher key from a SIM card without destroying it and that the authentication server is in the safe custody the Mobile Network Operator.

However, other types of side-channel attacks have proved successful on SIM cards (Novak, 2003). One such example is the masquarading attack as described by Kiran & Krishna

(2018). The GSM protocol requires that GSM handsets attatch to the base station with the strongest signal. When a hacker introduces a rogue base station with a strong signal, all GSM handsets in the vicinity (~ 200 to 500 metres) attatch to it. While the real base station would require communication between itself and the GSM handset to be on the stronger A5/3 encryption algorithm, the rouge base station can induce the device to use the weaker, crackable A5/2 algorithm. This is enough to obtain the secret cipher key given that the same secret encryption key is used for both algorithms.

Barkan, Biham, & Keller (2008) demontrated such an exploit using an insant ciphertext-only attack on the A5/2 algorithm, succesfully recovering the cipher key in a matter of minutes using a two terabyte (2TB) rainbow table. So effective is this sort of attack that specially-designed devices known as *IMSI Catchers* are commerically available for just this purpose. In fact, Mjølsnes & Olimid (2017) illustrate how to assemble such a device using readily available tools and equipment, and without requiring any programming knowledge. Employing it, one is able to track GSM handsets within an area, capture their IMSI (International Mobile Subscriber Identity) and TIMSI (Temporary Mobile Subscriber Identity) numbers, block service to handsets and even intercept incoming SMSs and calls.

In studying the weaknesses of GSM, Toorani & Beheshti (2008) highlighted SIM cloning as one of the key vulnerabilities of the architecture. Knowledge of the victim's IMSI and SIM cipher key are sufficient for a hacker to clone a SIM card, make calls, send SMSs and perform USSD requests that appear indistinguishable from that of the legitimate customer. The sole challenge inconveniencing the hacker is that GSM allows only a single SIM to access the network at any given time. Meaning, should both the victim and hacker attempt to connect to the network while at a considerable distance from each other, the network will detect the duplicate SIM and disable the affected account.

While previously these types of attacks were thought merely academic, the present technology and computing resources at the disposal of any would-be attacker have necessitated serious research on means of hardening GSM authentication mechanisms.

2.9 Multi-Factor Authentication

Multi-factor authentication refers to the use of several separate pieces of evidence as proof of identity. Typically, the they belong to at least two of the following categories: unique knowledge (something they know), unique possession (something they have), and inherence (something they are) (Nwabueze, Obioha, & Onuoha, 2017). In realising the vulnerabilities of reliance solely on static KBA, banks have begun the adoption of two-factor authentication in the form of PINs and One-Time Passwords (OTPs).

However, the use of SMSs to deliver OTPs renders it vulnerable to interception as the message must traverse a channel over which the bank has no direct control. It has been demonstrated that GSM and even 3G networks cannot guarantee confidentiality of SMSs. Criminals have also adopted alternative means of capturing SMS, targeting the end-device rather than the network by use of mobile phone trojans designed to log SMSs in real-time (Mulliner, Borgaonkar, Stewin, & Seifert, 2019). For these reasons, OTP cannot be considered secure.

2.10 Dynamic Knowledge-Based Authentication

Authentication was borne out of the need for computers to associate entities with identity in order to distinguish between legitimate users and others. In designing an authentication mechanism, a balance must be struck between security and usability to effectively prove identity without requiring an inordinate amount of resources or effort (Smith, 2001).

Of the three main authentication schemes (i.e. knowledge-based, biometric and token-based) this study shall be particularly concerned with knowledge-based authentication (KBA) where a shared secret such as a PIN or password secures access to a system or resources. Our choice is due to its simplicity. Using either biometric or token-based authentication for USSD transactions would require specialized hardware in addition to a GSM handset.

Threats against knowledge-based authentication can be broadly categorized into two: user-side attacks and server-side attacks. (Katsini, Belk, Fidas, Avouris, & Samaras, 2016). In the former, the goal of the hacker is to steal or extract the authentication key from the user. This could be through the use of malware, social engineering attacks or observation

techniques such as shoulder surfing. The latter entails trying to gain access by bypassing the security mechanism of the server. Brute force attacks and cryptanalysis are the commonly employed techniques to achieve this.

Static KBA utilizes human knowledge that is held by an individual permanently or as long as the individual's memory allows (Kwon & Moon, 2007). Its weakness lies in the fact that acquisition of the single piece of knowledge is sufficient to impersonate the legitimate user. For this reason, several scholars have proposed a new type of KBA. Dynamic KBA diverges from static KBA in that the user does not share the secret information beforehand. Rather, the questions and answers are derived from various data collections on users' activity that are in the exclusive possession of the authenticating party (Skračić, Pale, & Jeren, 2014). The security of this method is a consequence of the unlikelihood that anyone other than the legitimate user can know the correct answers to the dynamically generated questions. The questions are designed to be easily rememberable to the customer and hard to guess by the hacker (LexisNexis, 2012).

Bypassing of static KBA requires an attacker be in possession of merely a single piece of information: his victim's PIN. However, for dynamic KBA, the attacker would be faced with the considerably more difficult task of obtaining the customer's complete data set (which is likely an aggregation of data from several other data sets); and even then, he would have to perform a range of queries on the fly in order to derive the correct answer when presented with a challenge question. As for insider threat, dynamic KBA frustrates the efforts of the fraudster by requiring him to have real-time, read-access to the bank's data warehouse. On the off-chance that the fraudster is somehow able to capture a snapshot of the complete customer data set, it would be useful to him for only a small window of time. Customer data sets are in a state of constant flux and the correct answer to a particular challenge question varies based on when it is asked. Lastly, social engineering exploits are rendered impractical by the utter variety of the challenge questions. Being infeasible to accurately predict what questions will be asked, the social engineer cannot attempt to extract the information from his victim before-hand.

In realising the potential in dynamic KBA, a number of organizations offering identity management solutions such as GB Group have begun to invest in commercializing

dynamic KBA solutions. Authentication services such as *IDology* ask a customer several "out-of-wallet" questions in order to ascertain his claimed identity. The out-of-wallet questions quiz the customer on information that cannot be gleaned from the contents of a wallet i.e. questions not relating to birthdays, spouses, identification numbers, residences etc. (IDology, 2019). To appeal to a wider clientele, these commercial solutions utilize diverse datasets that are legally obtainable and often sold for such purposes (e.g. credit reports and customer surveys) thereby making the authentication scheme potentially "crackable".

For the highest level of security, the source data set should be exclusive to the authenticating party. The difficulty then encountered is that of extracting diverse challenge questions solely from data in possession of the bank, and the answers to which can be easily recollected by the customer (LexisNexis, 2012).

The difference between static and dynamic based knowledge based authentication is illustrated in Figure 2.8.

2.11 Dynamic Programming for Question Generation

The usability and security of dynamic KBA is dependent upon the quality of challenge questions employed. Questions, the answers to which are too difficult to recollect, will only frustrate legitimate users, rendering the authentication process tedious and likely "lock out" a legitimate user. In contrast, challenge questions whose answers can be easily guessed or deduced offer no security; the result being a large number of spurious accesses. The style of challenge question significantly contributes to facilitating recall among various users with different recollection abilities. (Albayram, et al., 2015). This "state of imbalance" is illustrated in Figure 2.9.

Some of the key findings in the research carried out by Albayram et al. (2015), are that: users are likely to remember occurrences only within a maximum of the preceding thirty to forty-five days, repeated actions have greater *recall-ease* and locations frequented by a user are retained in their memory longer.

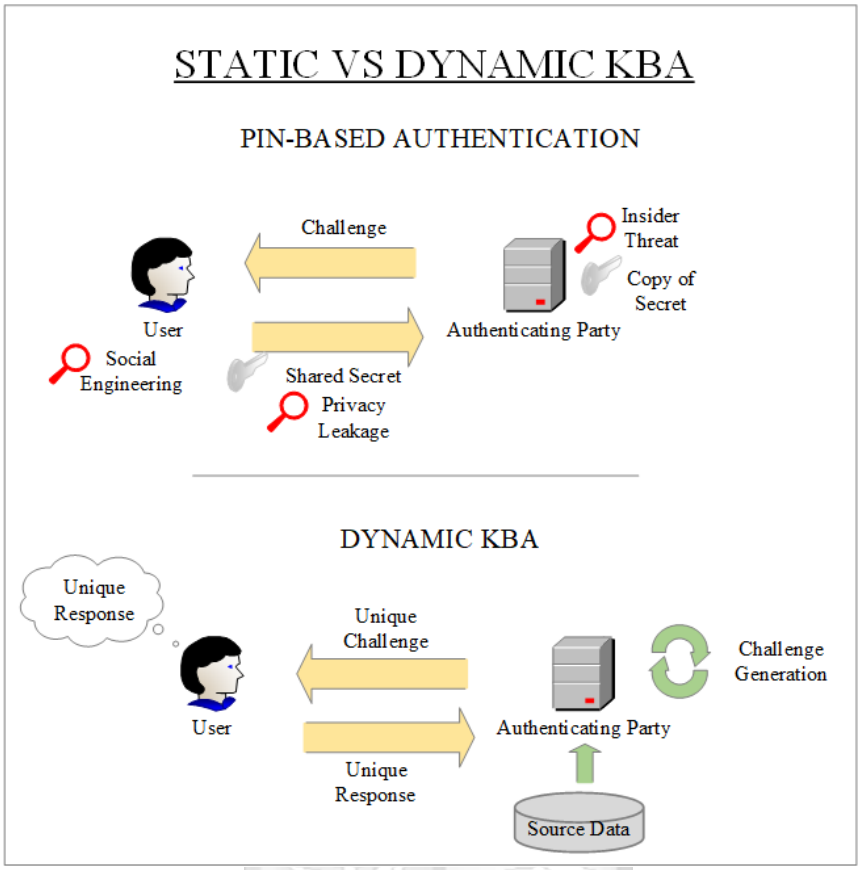


Figure 2.8: Static vs Dynamic Knowledge Based Authentication.

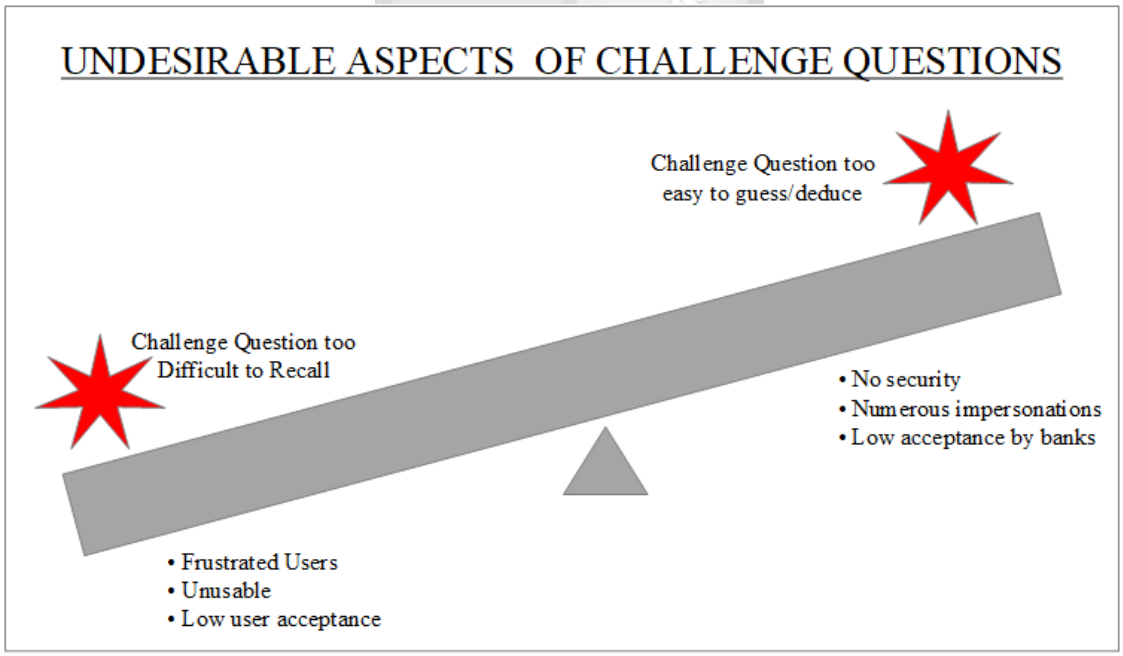


Figure 2.9: Undesirable Aspects of Challenge Questions.

When faced with a programming challenge that is comprised of smaller sub-challenges (the results of which can be used for other, future sub-problems) then dynamic programming algorithms may be used to optimize the solving of such an operation (Huang, Gu, Wang, Liu, & Chen , 2018).In particular, the results of the sub problems are stored for future reference in what is known as *memoization*; which reduces the time complexity of the problem from exponential to polynomial. In this study, challenge question generation is modelled around this algorithm for the reason that is it significantly "cheaper" to reference a stored value than recompute it.

Metatags of the underlying tables are analysed to generate the phrasing of a challenge question as well as the query needed to verify the response of a user. Although the correct response is unique to a particular user, it is possible optimize database lookups by temporarily storing common, intermediate values such as the syntax of the underlying dynamic query.

2.12 Research Gap

Though there exist works on dynamic KBA, none is yet to be applied in the context of USSD-mediated banking transactions. As USSD sessions are active for a maximum of two minutes and terminate when idle for longer than twenty seconds, meeting both the security and real-time requirements of USSD banking applications required improvements on existing techniques.

2.13 Conceptual Model

Figure 2.9 illustrates the conceptual model of the proposed solution. A two-factor authentication approach, incorporating both static and dynamic KBA, was used to verify the identity of bank customers. Questions were generated from a data set in exclusive possession of the bank and internally restricted.

The intention was to reinforce security using a combination of static and dynamic KBA based off an aggregation of the bank's customer data. To prevent criminals from becoming familiar with challenge questions (and recalibrate their social engineering strategy), security is reinforced by the retirement of questions that have attained a pre-defined

threshold of use; to be replaced by new ones that test a different element of customer knowledge.

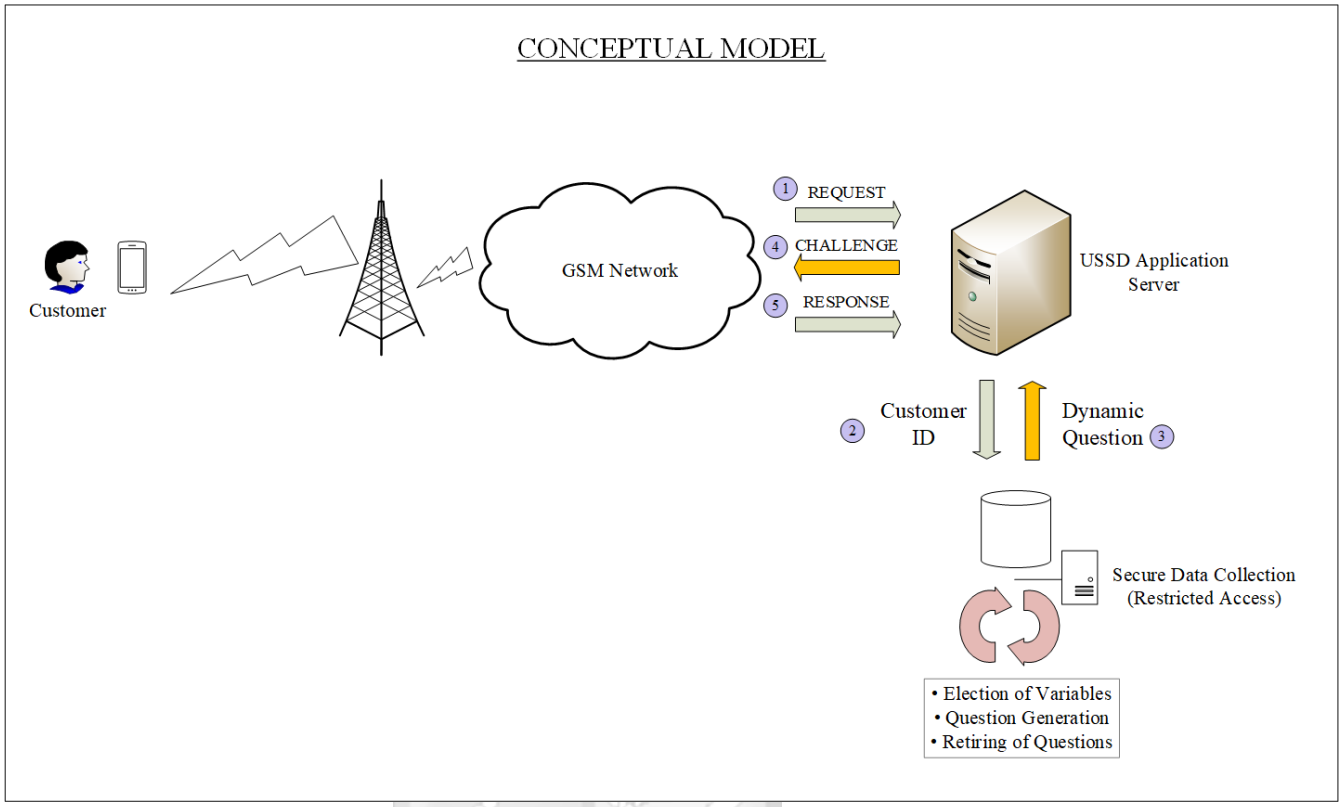


Figure 2.10: Conceptual Model of Proposed Solution.

Chapter 3: Research Methodology

3.1 Introduction

Research Methodology refers to a set of skills, techniques and tools commonly used in developing and testing new theories (Kumar, 2011). Within this chapter, we examine the research design to be adopted and select appropriate data collection methods. We then expound on the selected methodology, outlining the factors informing this choice. The closing section is sub-divided into these sections: choice of development tools, ethical concerns and measures taken to uphold the quality of research.

3.2 Research Design

The aim of the study was to design and prototype a two-factor authentication mechanism utilizing both static and dynamic KBA and that addresses the vulnerabilities of privacy leakage, insider threat and social engineering. To this end, a variety of techniques, tools and procedures were employed in the different research stages. By virtue of the fact that the study sought to address an unrequited need within the banking sector, and would have real-world application, it can be said to fall squarely within the class of applied research (Walliman, 2011).

The design science methodology was selected where the researcher sought to solve the problem by focussing on how the ideal solution ought to be. The conceptual model comprised various components that affected the performance (speed) and effectiveness of the authentication mechanism. These components were manipulated to achieve as close to "near real-time performance" as possible while also maintaining effective levels of security.

Testing of the proposed solution was with respect to performance metrics (the unit duration required to successfully present a challenge question for a user and verify the response) and security metrics (the likelihood that the answer to a question will be remembered by a legitimate user and difficult to guess by an attacker).

3.3 Data Collection

Data collection is the application of techniques to accurately collect empirical data for use in research (Tashakkori & Teddlie , 2010). Wireshark, a network sniffing tool, was used to

capture USSD communication between the test mobile device and the USSD application. Investigation of the packet timestamps formed the basis of comparison between the proposed and existing approach with regard to average session duration.

SQL database logs were also recorded to obtain data on the performance of question generation and replacement. This was used to approximate the performance overhead of the proposed technique and also compare it to the performance of the existing technique. Evaluation of the security offered by the proposed technique was performed by allowing test users to generate their own data and attempt to answer the challenge questions.

3.3.1 Location of the Study

The study focused on banks operating within Kenya (and more precisely Nairobi) for the reason that the Kenyan MFS ecosystem is arguably the most developed in Africa. Should the proposed solution be applicable in Kenya, it is ostensibly replicable anywhere else on the continent. Additionally, due to the financial and time constraints of the study, Nairobi was the most suitable choice owing to its proximity to the researcher.

3.3.2 Population and Sampling

A population refers the set of all things or cases which are the subject of a research (Atikan, 2016). To test the human-facing element of the proposed mechanism, a total of twenty test users were selected from among the researcher's colleagues using convenience sampling; a non-probabilistic method in which test subjects are selected merely because they are "convenient" sources of data for the researcher. Due to the unprecedented Covid-19 crisis, free movement of the researcher was curtailed thereby necessitating the chosen method. However, all of the twenty test users had previously interacted with USSD banking applications.

The test users were randomly divided into two equal-sized groups: the experimental group and the control group. The former generated their own test data (to simulate legitimate users) while the latter did not (to simulate adversarial users). Each of the test subjects then attempted to perform various mobile banking transactions requiring dynamic authentication.

In evaluating the performance of the proposed authentication method, Wireshark timestamp logs were recorded and analysed. Of interest were the time intervals between requests and responses. This was to aid in comparing the performance of the proposed mechanism and the contemporary method.

3.3.3 Data Generation

At a minimum, a Kenyan bank will typically have customer data aggregated from these channels: credit/debit cards, agent banking, over the counter transactions, mobile banking and online banking (Magutu, et al., 2011). The customer data set of the study was generated using these characteristics.

For ethical reasons, the data used in this study was not real. Instead, dummy data generated by the researcher was utilized. Database seeding using the Laravel framework was used to populate the database with test data matching the volume and characteristics required by the study (Laravel LLC., 2019).

3.4 Data Analysis

Analysis is the process of transforming data into information, and thereafter information into knowledge (Cuesta & Kumar, 2016). Performance and security effectiveness were the two main variables to be analysed with respect to the technical aspects of the proposed technique. Visualizations that lent themselves to easier interpretation were then relied upon to theorize based on the given data.

3.5 Design Phase

The University of Connecticut (2019.), under its information technology services, defines design as the process of "transforming identified requirements into a detailed system architecture that is feasible and which brings value". In performing this, Computer-Aided Software Engineering (CASE) Tools such as *StarUML* were utilized to produce high-level design documents such as use cases, entity relationship diagrams, sequence diagrams, class diagrams etc.

3.6 Prototype Development Methodology

Owing to novelty of the envisioned solution, development of a working prototype that achieved the objectives of the study required gradual progress. It was expected that the researcher would encounter challenges in achieving a working prototype that did not exceed the duration constraint of a USSD session. With this in mind, Agile Methodology was selected to guide development.

The key distinguishing feature of agile development is that it is responsive to rapid changes in an uncertain environment (Agile Alliance, n.d.). Several techniques exist under the agile umbrella, but of interest to the study was test driven programming. Agile Alliance (n.d.) defines it as a development technique in which three process are closely intertwined: coding, testing and design. Kent Beck (2002) outlines how this technique solves the seeming contradictory objective of developing clean code in the shortest time possible. He does this using yet another seemingly paradoxical approach: testing code before you write it.

Essentially, tests are first written, implemented in the simplest possible form using basic code and then tested. Once the code has passed the test, it can then be refactored for efficiency, maintainability and robustness. This process is illustrated in Figure 3.1.

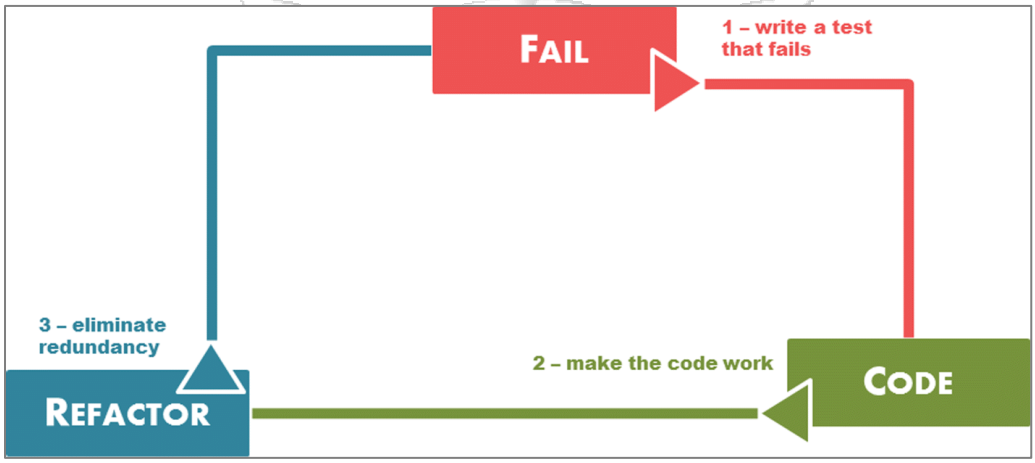


Figure 3.1: Test Driven Development Cycle (Ivo, Guerra, Porto, Choma, & Quiles, 2018).

3.7 Development Tools

Powered by a MYSQL database, the preferred programming language was PHP on account of its diverse array of open source libraries. An android handset was used interact with the USSD application. Wireshark was the tool of choice for monitoring of network traffic with *Navicat* recording the performance metrics for backend database operations driving the USSD application.

3.8 Research Quality

Quality research refers to the degree that the scientific process has been encompassed in all areas of the study. More specifically, it relates to the judgement on the match between questions and methods, measurement of variables and protection against various types of biases. (Boaz & Ashby, 2003; Lohr, 2004; Shavelson & Towne, 2002) as cited by the South West Educational Development Laboratory (2005).

The choice of past work to be studied and research instruments to be employed collectively contributed to a credible scientific method that enabled this study meet the standards of quality research. Data accuracy was achieved with the aid of detailed measurement tools.

3.9 Ethics and Research

Walliman (2011) argues that reliable of progress within a field of knowledge is dependent on the integrity of researchers; as results cannot be trusted if there is suspicion that the researcher acted in an otherwise manner.

Cognizant of the sensitivity of bank data, all testing was carried out in a simulated environment so as not to negatively impact the business or reputation of any bank. Test customer data was also be randomly generated, having no similarity to real data sets apart from that which was purely coincidental.

In adhering to ethical guidelines, the researcher correctly acknowledged and attributed all work that is not his own. Lastly, the researcher sought ethical approval from an independent review body in whose jurisdiction the study took place. As part of the researcher's ethical obligations, the findings of the study were made available to all parties and participants involved in the study. A copy of the research is publicly available on Strathmore

University's thesis repository. Beyond this, the researcher has also sought to disseminate the findings to a broader audience outside the research community; the target being the banking sector in which it is hoped that the study can have a positive impact on the fight against fraud.

Chapter 4: Design Analysis

4.1 Overview

This chapter addresses the set of requirements that the proposed solution intended to satisfy in order that the study may achieve its objectives. It is divided into these sub-headings: requirement analysis, system architecture and system design. Aided by UML diagrams, depictions of the overall system architecture, its components and their various interactions are provided.

4.2 Requirement Analysis

Review of literature highlighted the vulnerabilities the proposed model ought to address. In question form, they were: How can we make it more difficult for malevolent insiders to use personally identifiable information (PII)? How can we make data retrieved from privacy leakage exploits less dangerous? How can social engineers be stifled even if they manage to coax a user's PIN or swap their SIM card? The solution proposed was use of challenge questions with dynamic, time-variable responses.

Additionally, the desirable characteristics of the proposed authentication mechanism were discussed. Essentially, in order to be effective challenge questions must be easy to recollect yet difficult to guess. Informed by these considerations, requirement analysis was performed to identify the requisite characteristics that the proposed authentication system needed to have in order to achieve the objective of the study. Discussed in the next section are the actionable, measurable and testable requirements identified.

4.2.1 Functional Requirements

These refer to what the proposed solution should do i.e. its intended behaviour (Gupta, 2008). The following functional requirements were identified:

(i) The authentication system ought to automatically generate a variety of challenge questions without the intervention of the system administrator.

(ii) The authentication system should intelligently recognize a candidate challenge question that is similar to another existing (active or retired) challenge question. In such an instance, the candidate challenge question should **not** be added to the list of challenge questions.

(iii) While posing a challenge question to a user the authentication system ought to present the challenge question and its possible answers in a multiple-choice format consisting of at least three choices. The distractors (wrong-choices) should be plausible enough to make it difficult to correctly guess by elimination. Also, the correct answer should have a balanced (random) placement.

(iv) The authentication system ought to accept a response from the user after which it will determine the correctness of the answer, thereby authenticating the user.

(v) Available to the administrator, should be statistics on the ask-rate, failure-rate and success-rate for each of the challenge questions.

(vi) The authentication system ought to be able to keep track of the challenge questions posed to a particular user to prevent repetition of the same challenge question beyond the threshold set by the administrator.

(vii) Once a challenge question has been posed to users more than the threshold set by the administrator, it ought to be automatically retired by the authentication system.

(viii) For high-risk USSD operations (e.g. withdrawal transactions (above KES. fifty thousand) as defined by the administrator, the authentication system ought to provide added security by posing at least two challenge questions before allowing the transaction to complete. Low risk operations (e.g. balance checks) need not have a challenge question.

(ix) In the event that a challenge question is answered incorrectly by a user, a different challenge question ought to be posed. However, no more than three challenge questions should be posed within the same USSD session to prevent a session timeout as a consequence of exceeding the maximum session duration.

(x) Whereas a high-value transaction should **not** be completed without at least two challenge questions being answered correctly, a user should be flagged for 'suspicious activity' only if five challenge questions are answered incorrectly.

4.2.2 Non-Functional Requirements

These requirements describe desirable system attributes and served as constraints on how the system was to be designed. They were:

(i) Availability: the authentication system ought to work as intended, when required and without degradation of services for the duration of the project.

(ii) Response time: given the one hundred and twenty second upper-limit for USSD session duration, the authentication system ought to provide responses in as close to near real-time fashion as possible.

(iii) Usability: ease of use is determined by the extent to which the answers to the challenge questions are rememberable by the legitimate user.

(iv) Security: the strength of the authentication mechanism lies in the unlikelihood of anyone but the legitimate user providing the correct answer to a challenge question.

(v) Performance: the system ought to be able to handle the load requirement of a typical bank deployment (thousands of requests per hour).

Figure depicts a USSD session with challenge questions having dynamic responses. As described, the user is posed with a challenge question with a multiple-choice answer set containing the correct answer and plausible distractors.

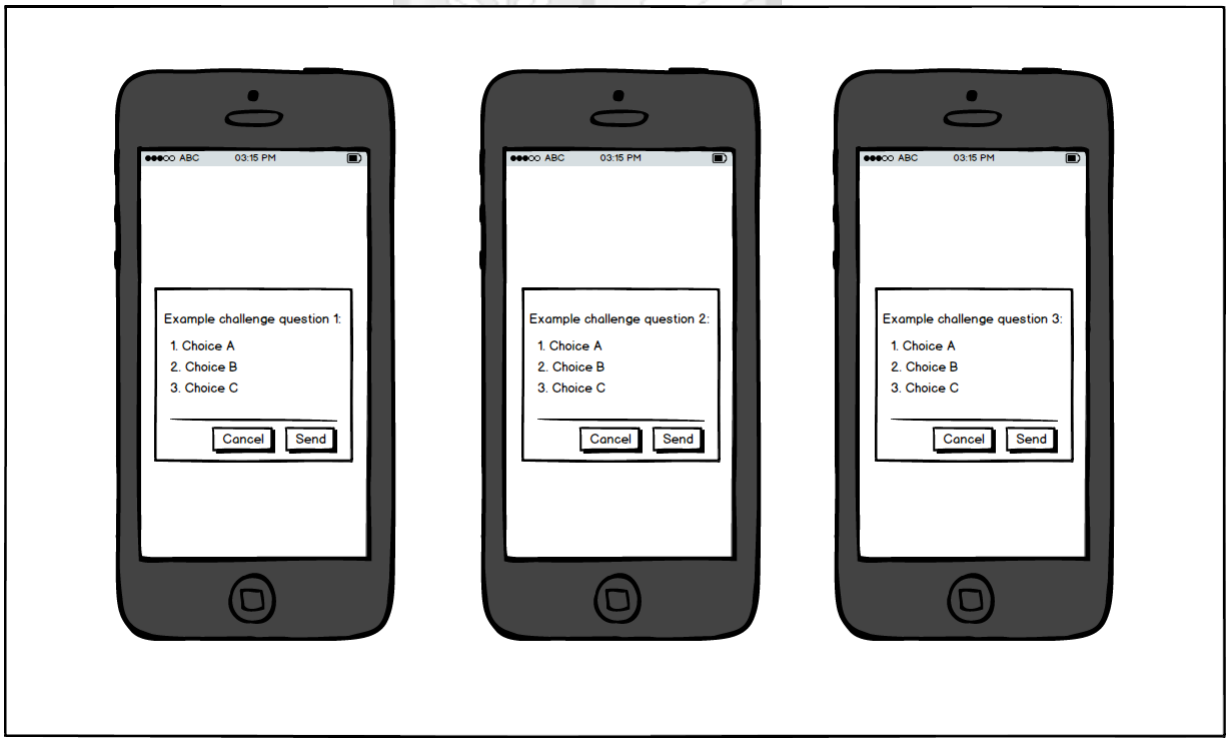


Figure 4.1: USSD Wireframes

Figure illustrates the administrator dashboard where the administrator is able to view various statistics on the challenge questions. Figure shows the administrator view to set the threshold of how many times a question should be posed to the same user.

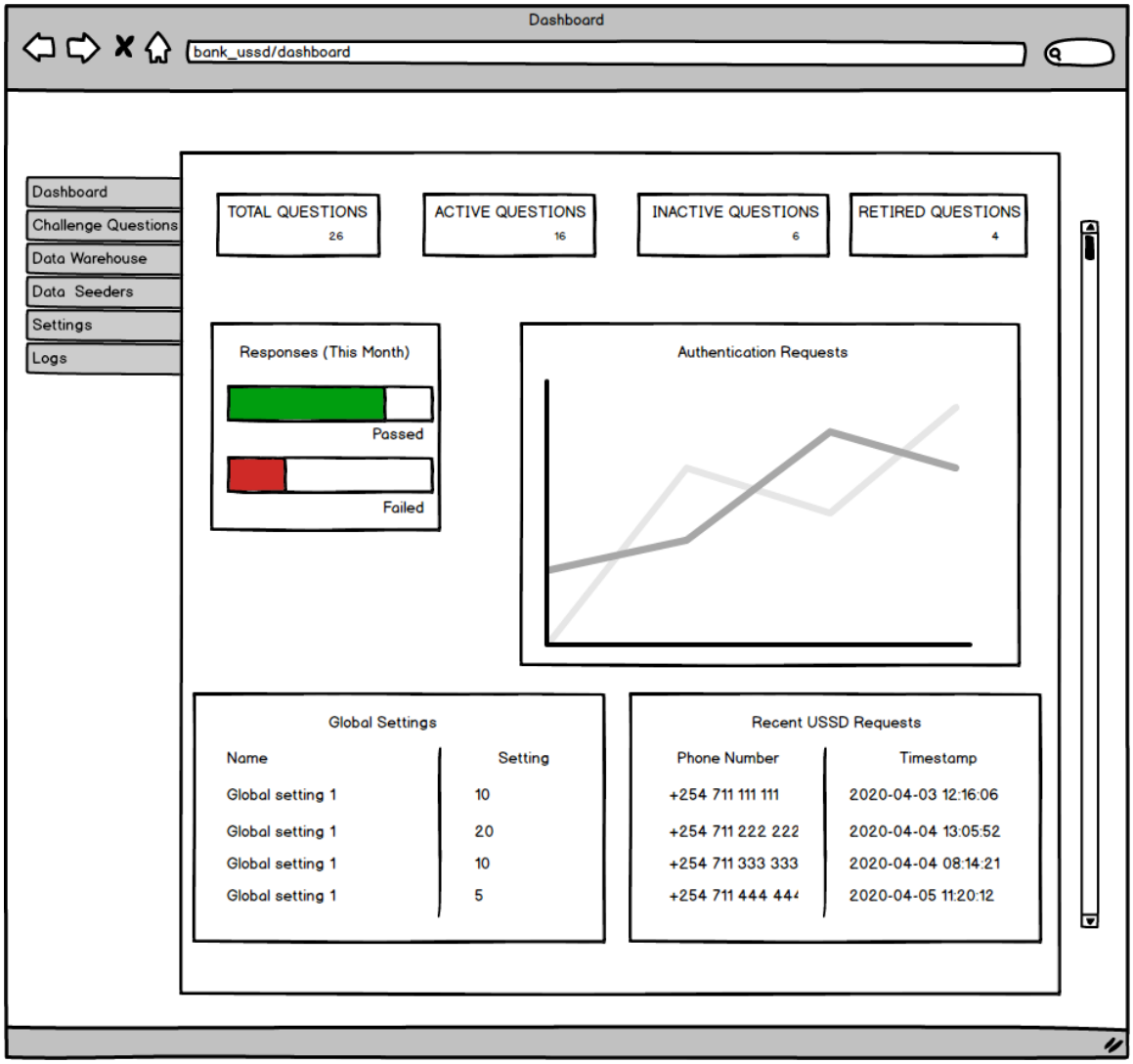


Figure 4.2: Administrator Dashboard Wireframe

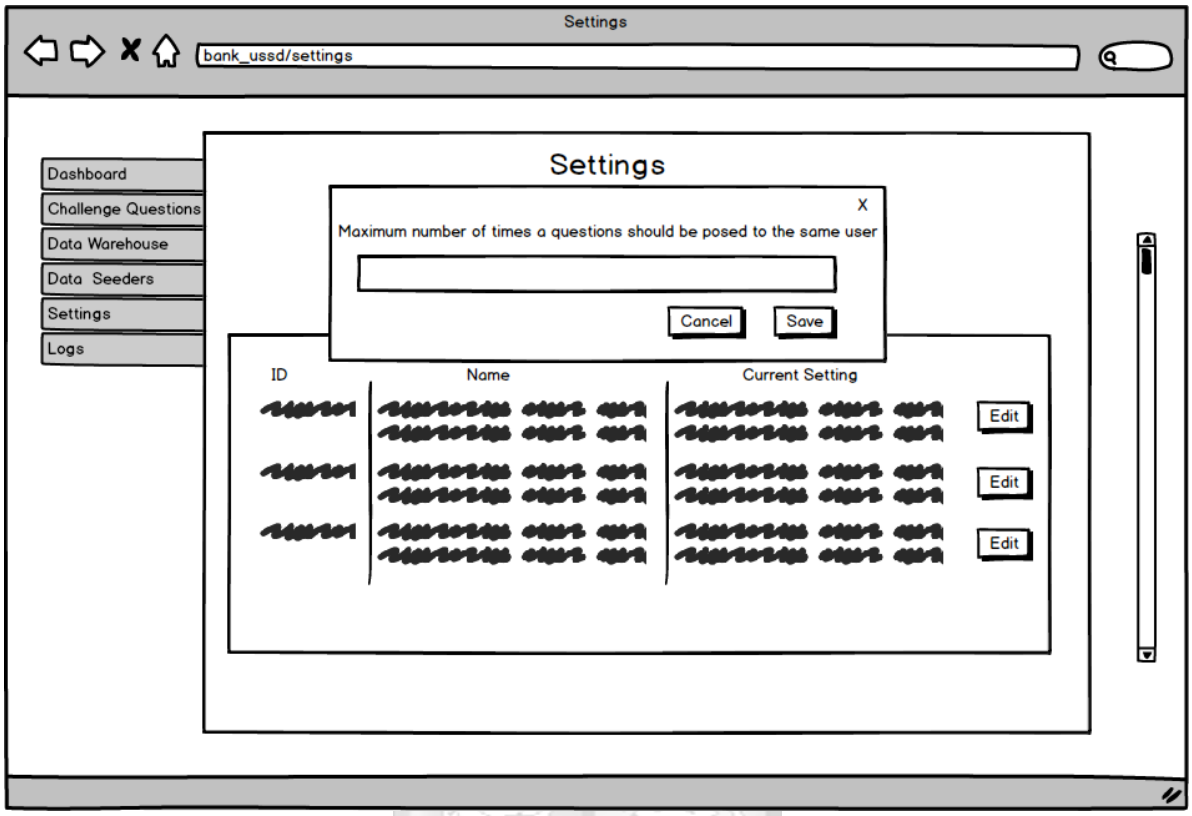


Figure 4.3: Administrator Global Settings Wireframe

The requirements were visualized in the following system design artefacts: use case diagram, entity relationship diagram, class diagram, data-flow diagram and sequence diagram.

4.3 System Architecture

Figure 4.1 illustrates the proposed architecture of the authentication system. At the lowest level are disparate databases containing customer data obtained from various banking channels such as ATM transactions, USSD applications, online banking platforms etc. To use the separate sources together, Extract-Transform-Load (ETL) techniques are first applied on the aggregate data for it to be rendered in a presentation-ready format. Kimball and Caserta (2011) remark that the aggregation process, if properly executed, can add real value to the data; allowing the owner to leverage its emergent properties.

Atop this layer is the application logic handling the generation and retirement of challenge questions. Generation of a challenge question and fetching of its multiple-choice answer set requires read access to the unified data set. Retirement of "aged" challenge questions is

performed automatically, where upon attainment of a preset threshold, the challenge question is no longer posed to users. Also performed within this layer is the maintenenace of logs relating to various aspects of performance.

Residing at the top layer is the USSD handler, whose role it is to mediate communication between the authentication system and the USSD application. It does so by parsing the requests and responses in formats interpratable by the respective parties.

4.4 System Design

Unified Modelling Language (UML) diagrams were utilized to further refine system requirements and to aid in the visualization of various implementation aspects. They include: a use case diagram, an entity relationship diagram, a class diagram, a data flow diagram and a sequence diagram.

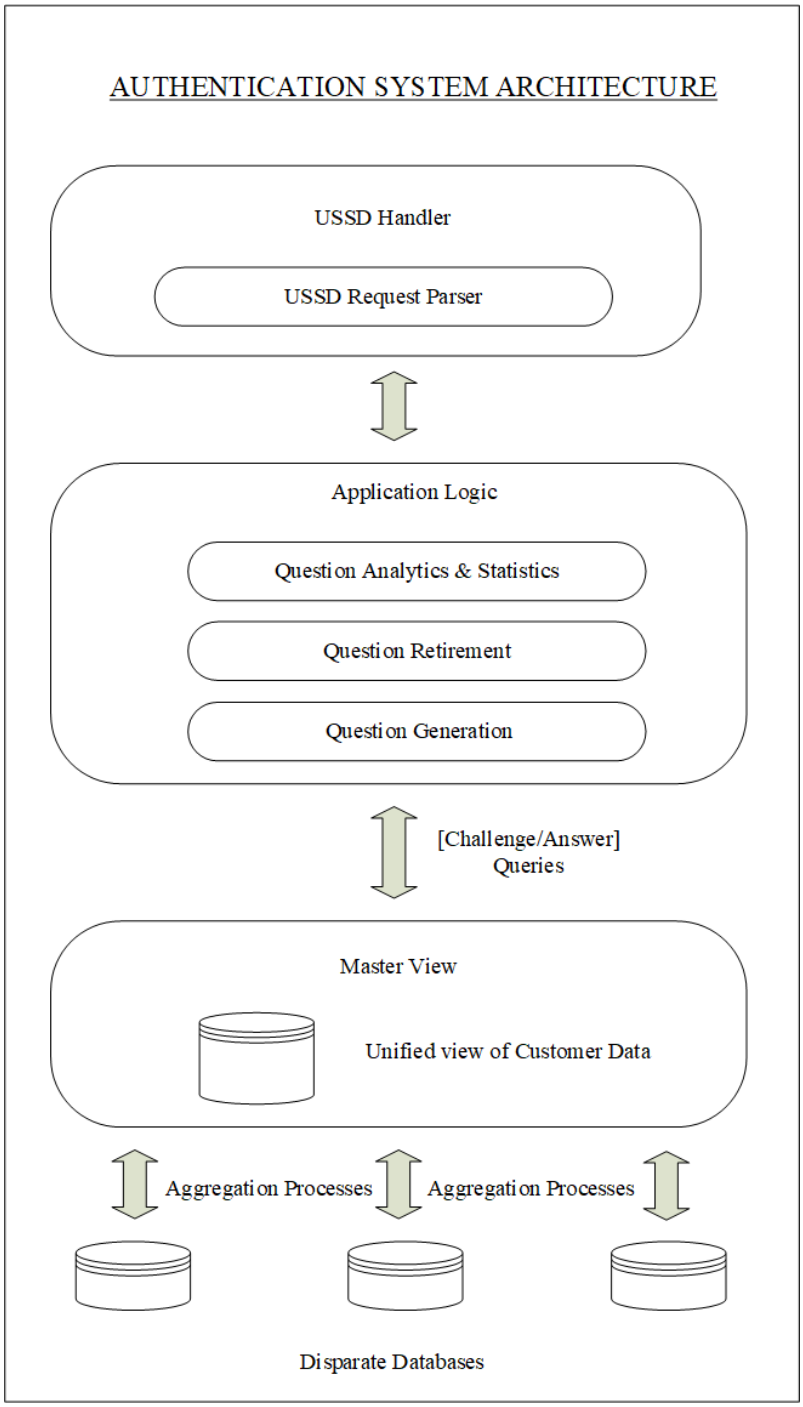


Figure 4.4: Authentication System Architecture.

4.4.1 Use Case Diagram

The simplicity of use cases makes them particularly useful in visualizing seemingly complex system interactions (Bittner & Spence, 2003). It does so by describing a sequence

of events that leads to a system doing something useful. Table 4.1 contains the basic flow for the main success scenario: the withdrawal of funds by a customer.

Table 4.1: Basic Flow for Main Success Scenario.

Actor Actions	System Response
1. The use case begins when the customer dials the bank's USSD code. 2. The customer provides his mobile banking PIN and attempts to withdraw funds to his mobile money account.	
	3. The Authentication System (AS) picks the customer's mobile number and uses is to fetch the customer's details. 4. The AS then fetches an active challenge question that meets the security threshold parameters set by the system administrator. 5. Given the chosen challenge question, the AS queries the unified data set for an appropriate multiple-choice set particular to that customer. 6. The system displays the challenge question and the multiple choices to the customer.
7. The customer, given his knowledge of his own account activity, selects one of the multiple choices.	

10. The customer completes the withdrawal request.	8. The AS validates the correctness of the response, adds a log entry and updates the usage statistics for that particular challenge question. 9. The AS allows the customer to proceed with the withdrawal request.

Figure 4.2 provides is a pictorial depiction of the basic functionalities of the system.

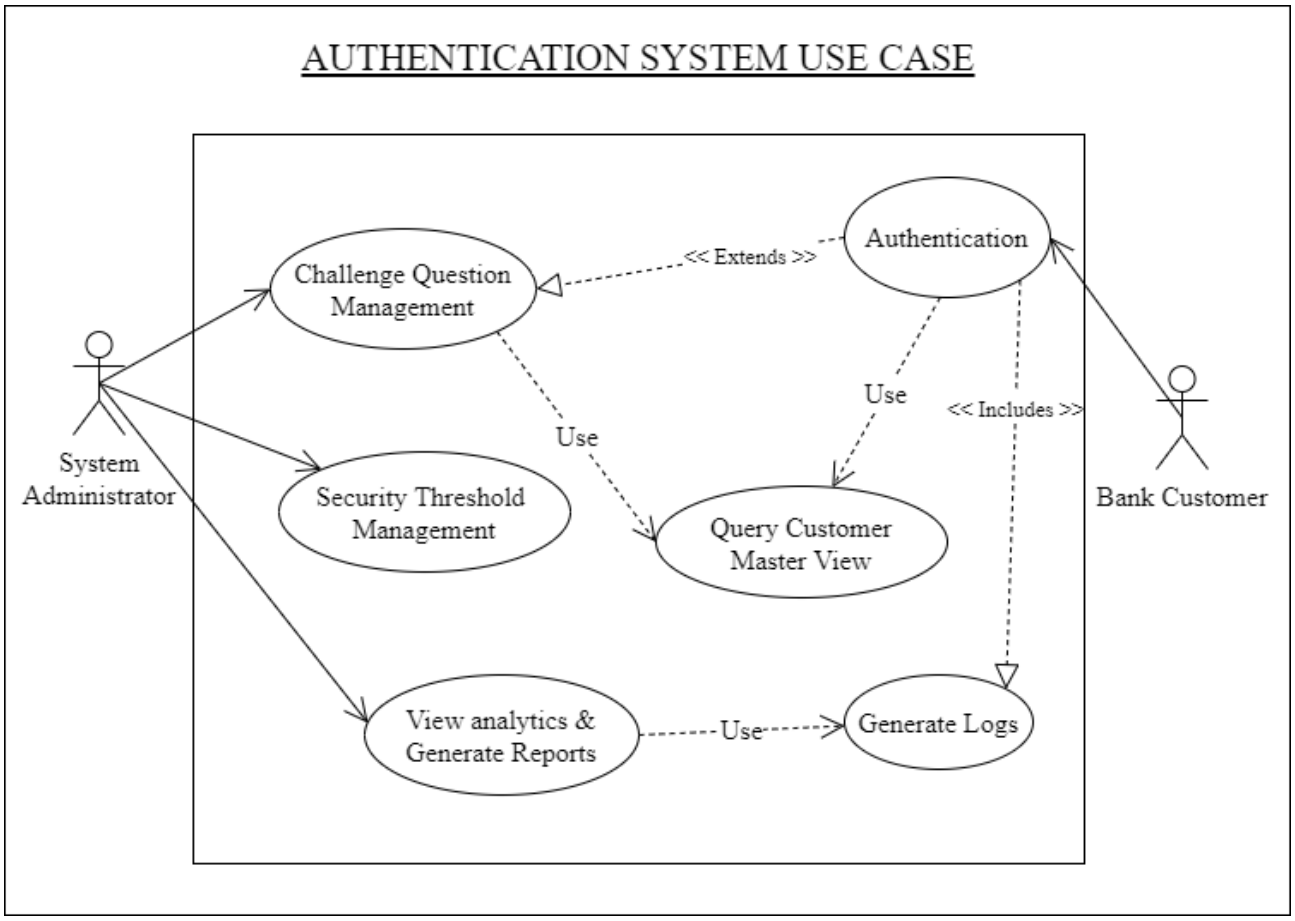


Figure 4.5: Authentication System Use Case.

4.4.2 Entity Relationship Diagram

The logical structure of the database was based on the objects captured in an Entity Relationship Diagram (ERD). MySQL, a relational database, was chosen to drive the authentication system. The tables, their attributes and the relationship among them are detailed in figure 4.3.

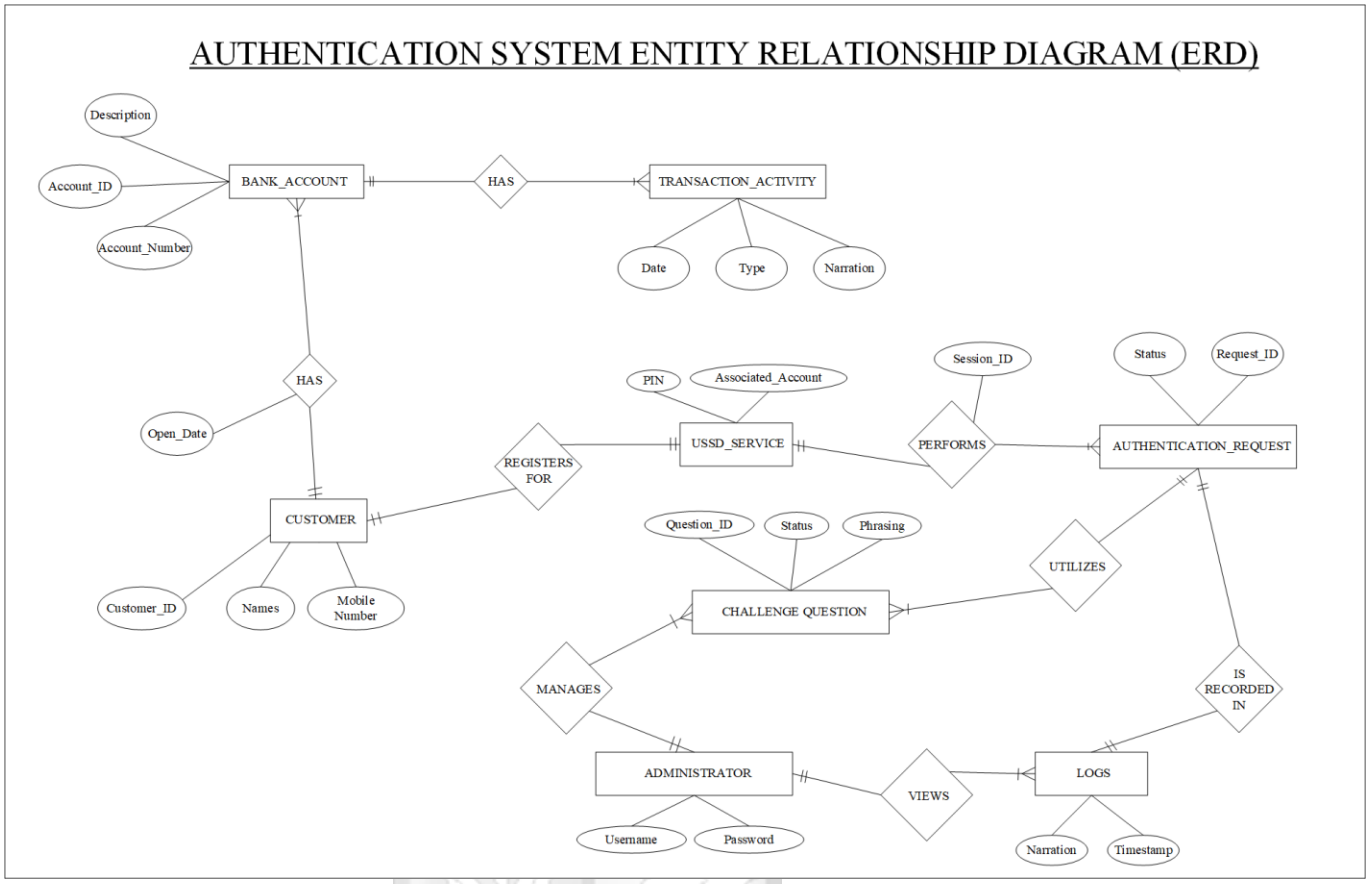


Figure 4.6: Authentication System Entity Relationship Diagram.

4.4.3 Class Diagram

The real-world objects and abstractions that form distinct entities in the proposed system were pictorially represented in the UML modelling tool known as a class diagram. With this design artefact, additional clarity is achieved by detailing the variables possessed by the class as well as the methods it requires in order to achieve its intended functionality. Figure 4.4 captures this.

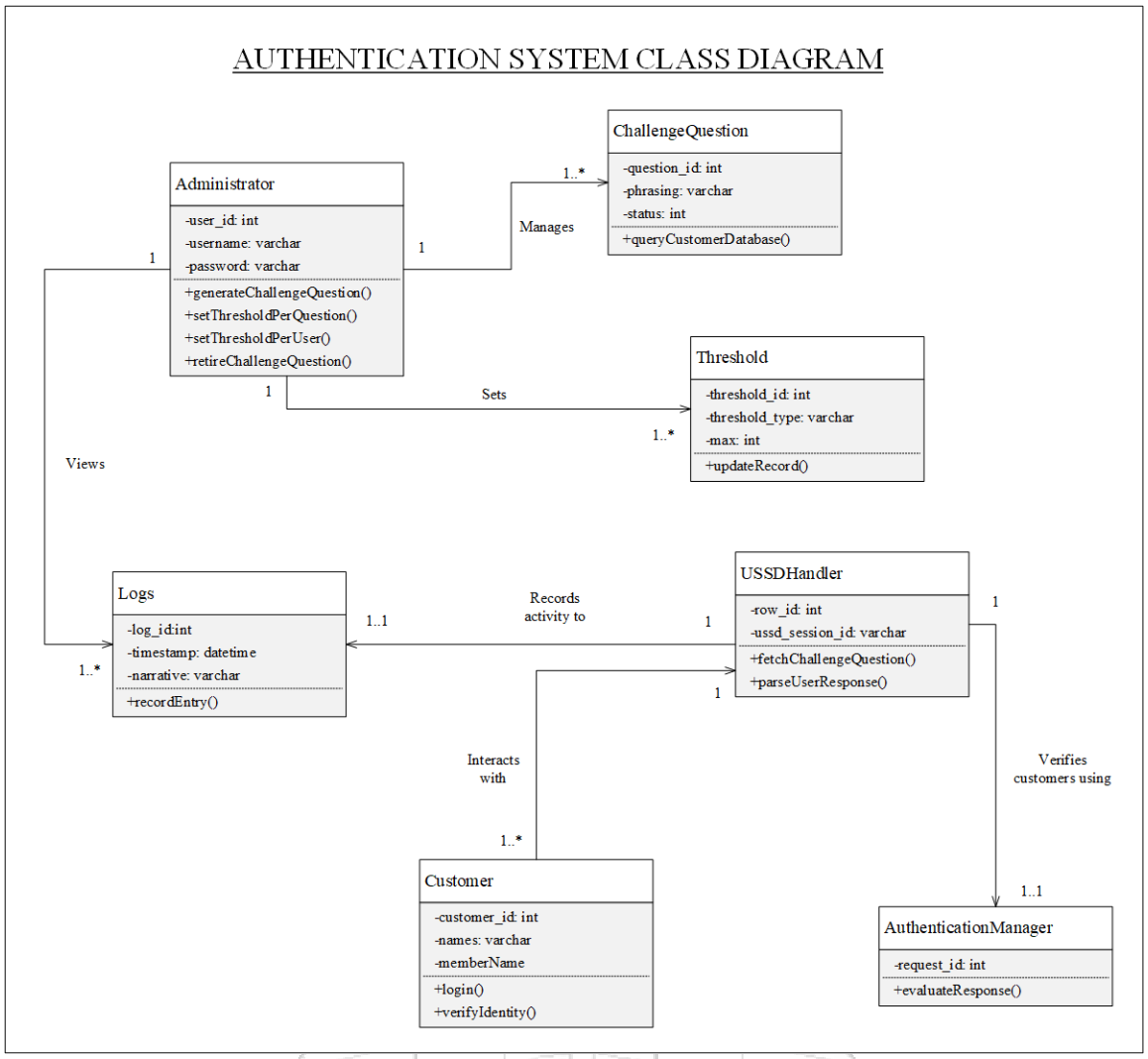


Figure 4.7: Authentication System Class Diagram.

4.4.4 Data Flow Diagram

Another way of depicting a system is through a Data Flow Diagram (DFD). Here, the system is understood as a collection of processes and actors with data flowing between them. DFDs are useful in breaking down a system into its logical sub-systems, each of which can be further decomposed to as detailed a level as necessary (Kaujalgi, 1994). The highest level of abstraction is the context diagram, captured in figure 4.5. This was further broken down to a level 0 DFD in figure 4.6.

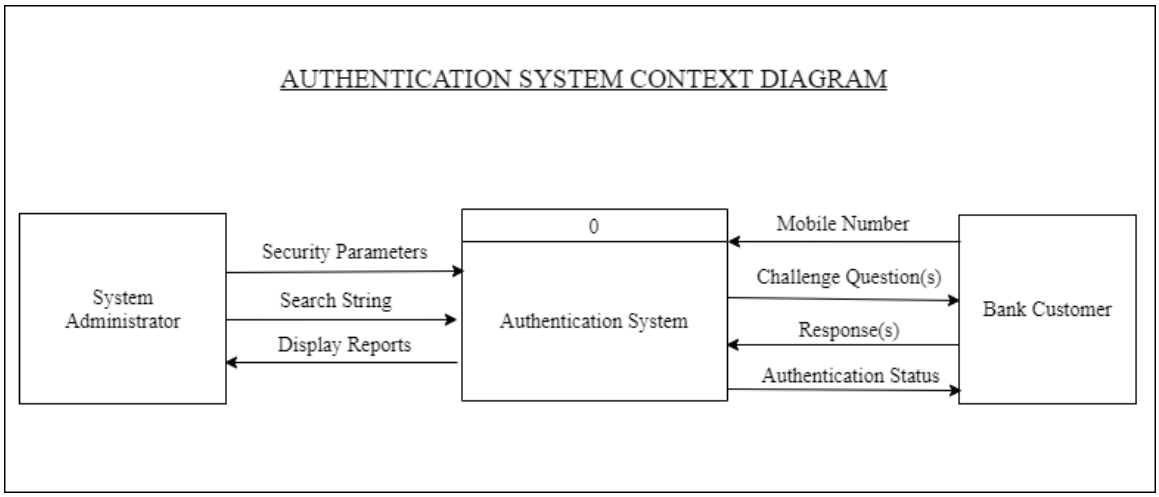


Figure 4.8: Authentication System Context Diagram.

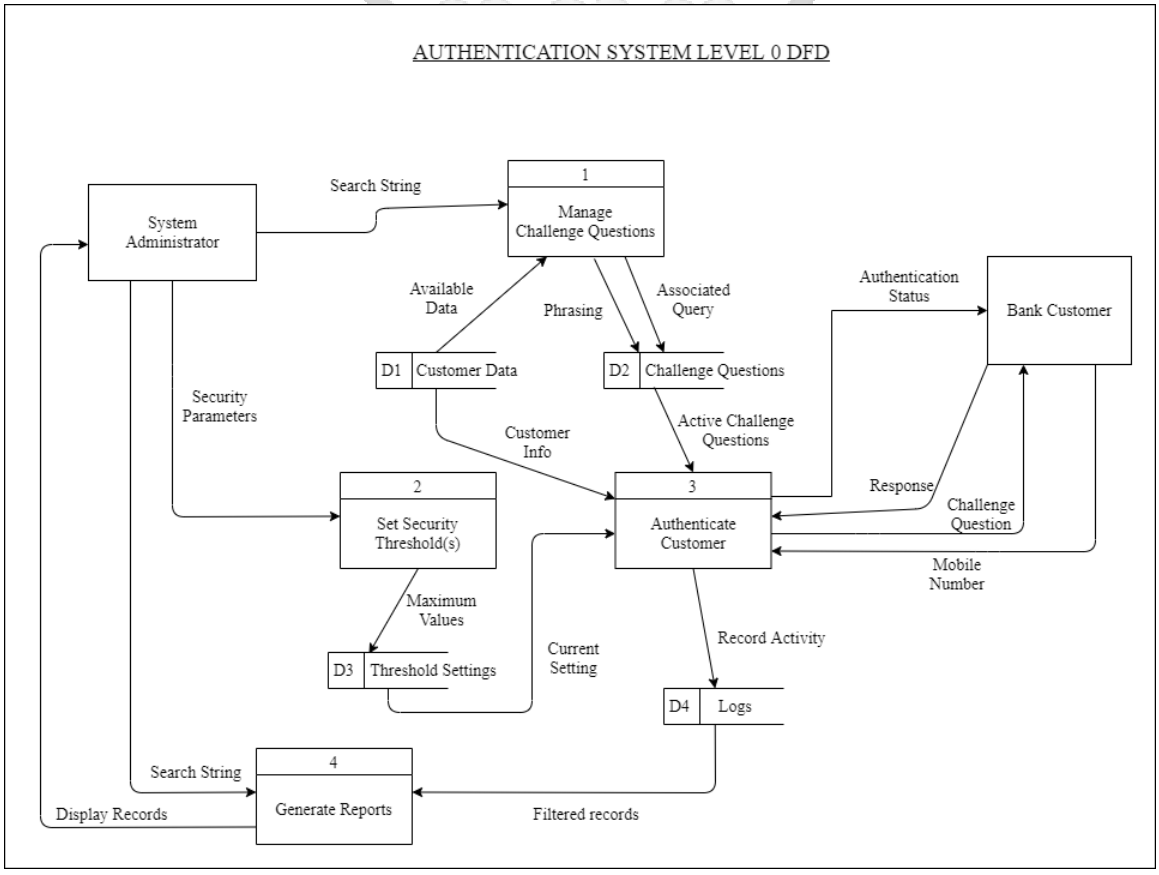


Figure 4.9: Authentication System Level 0 Data Flow Diagram.

4.4.5 Sequence Diagram

A sequence diagram takes a particular use case, and depicts the sequence of events necessary to usefully perform the desired action. Of particular interest, was the

authentication process of a bank customer after initiation of a USSD session. Figure 4.7 illustrates the classes involved in completing an authentication request, and the order of messages that must be passed between the customer and system for successful authentication.

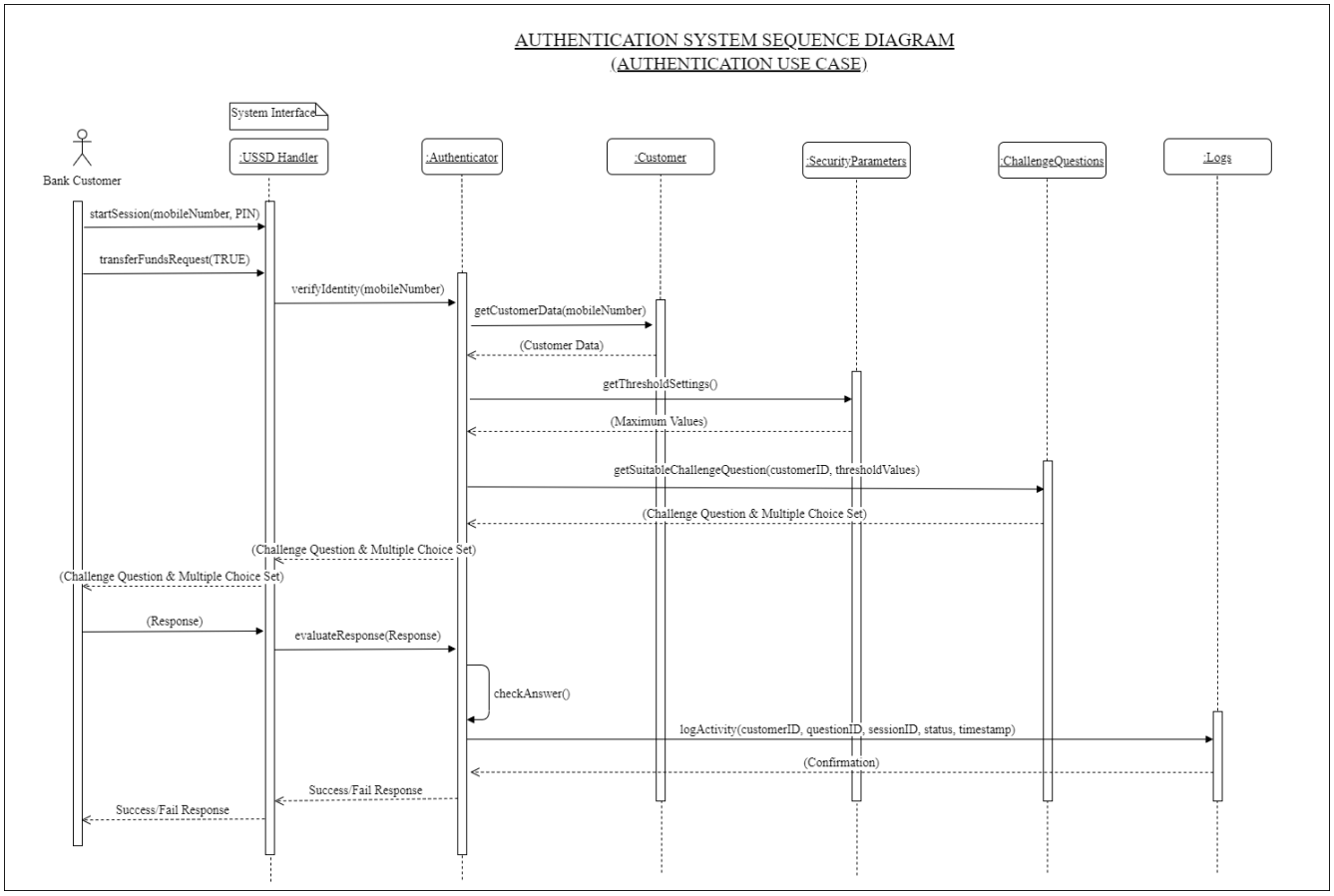


Figure 4.10: Authentication System Sequence Diagram.

Chapter 5: Prototype Implementation and Validation

5.1 Introduction

This Chapter covers the practical considerations of implementing the designs outlined in the previous heading. Thereafter, we examine the appropriateness of the prototype by testing its conformity to the identified requirements. Lastly, we compare the performance of the prototype to that of deployments utilizing the industry-standard practice.

5.2 Prototype Implementation

The prototype comprises three modules:

(i) Customer-facing module (USSD handler). See figure 5.1 for wireframe.

(ii) Administrator-facing module (Dashboard) See figure 5.2 for wireframe.

(iii) Authentication Manager (Back-end processes)

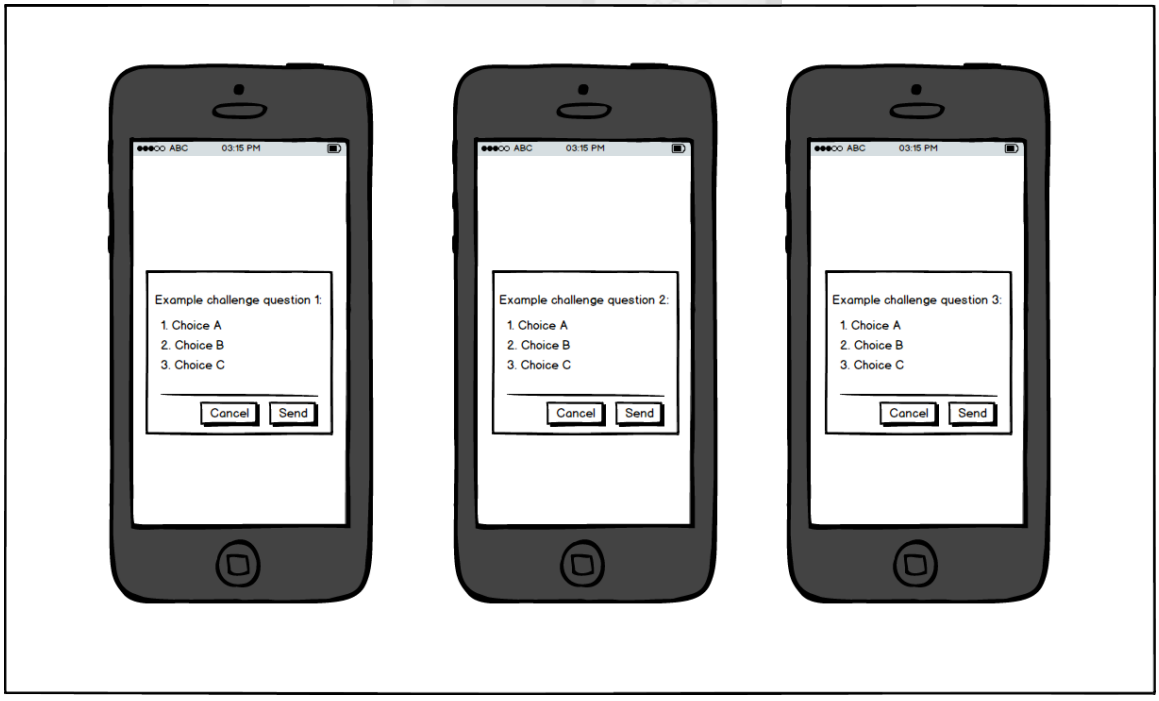


Figure 5.1: USSD Handler

Development was performed using the Laravel 5.5 framework and driven by MySQL database. USSD requests were handled using *Africa's Talking* API.

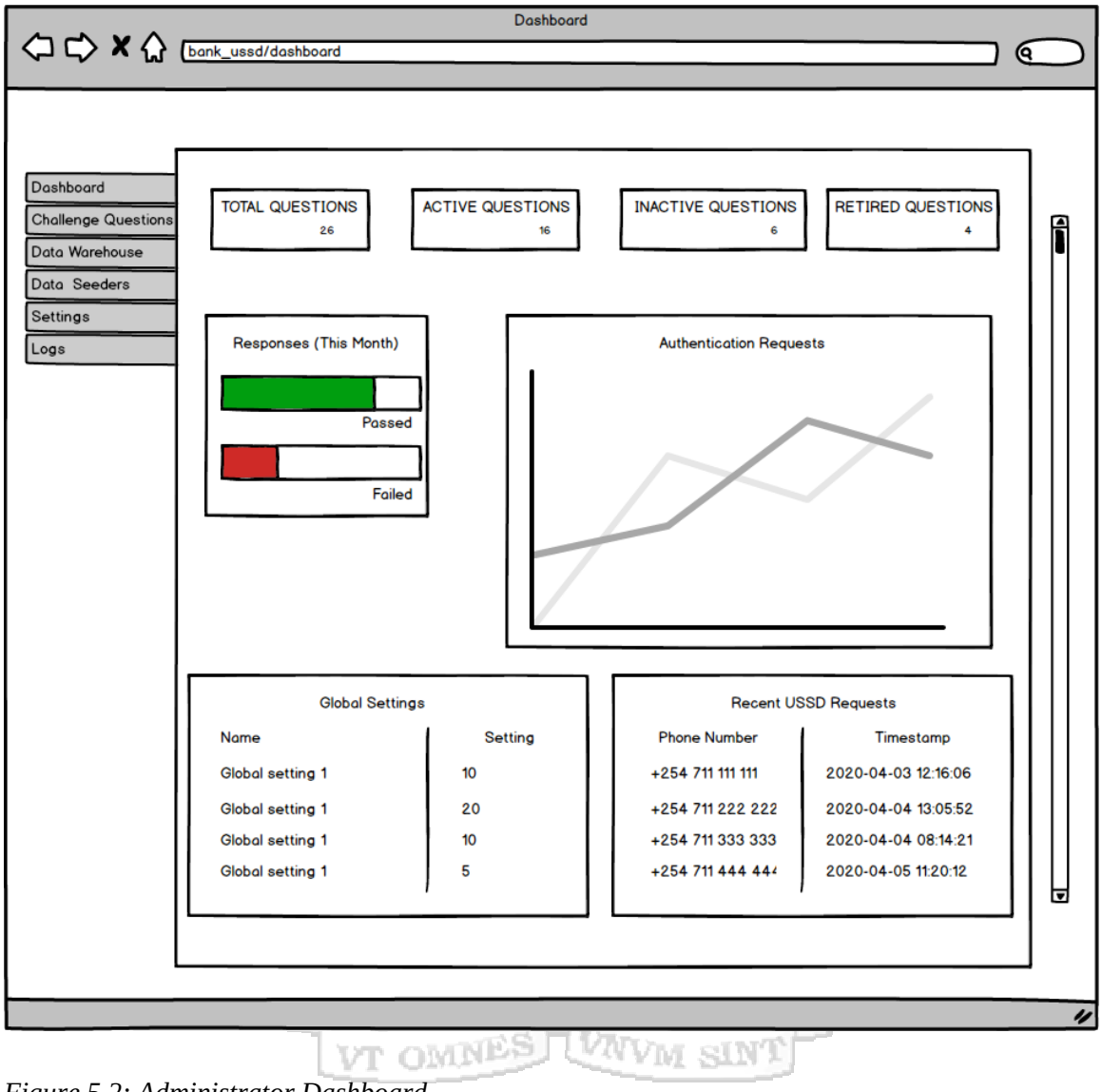


Figure 5.2: Administrator Dashboard.

5.2.1 Hardware Requirements

Pre-requisite to efficient operation of the prototype are the following hardware requirements:

Table 5.1: Prototype Hardware Requirements.

Hardware	**Minimum Requirement**	**Recommended**
Disk Space	500 Gigabytes (GB)	1 Terabyte (TB)
Memory	4 Gigabytes (GB)	8 Gigabytes (GB)
Internet Connectivity	Required	Required
Processor	1.8 Gigahertz (GHz), 64-bit	2.4 Gigahertz (GHz), 64-bit

5.2.2 Software Requirements

For proper functioning of the prototype, the deployment environment must be equipped with the following:

Table 5.2: Prototype Software Requirements.

Software	**Minimum**	**Recommended**
Operating System	Windows 7	CentOS / Ubuntu / Windows 10
Database Management System	MySQL 5.0.0.1	MySQL 8.0
PHP run-time environment	PHP 5.0	PHP 7.0

5.2.3 Prototype Development

Guided by Test Driven Development (TTD) of the Agile methodology, the three modules of the prototype were incrementally developed beginning with the core module: the authentication manager.

5.2.3.1 Authentication Manager

The Authentication Manager (AM) lies at the heart of the prototype, handling the concerns of challenge question generation, question retirement and security threshold enforcement. On receiving a request from a customer, the AM fetches an appropriate challenge question and then queries the unified data set to fetch a multiple-choice set.

The question generation algorithm analyses the metadata of the available tables to determine suitable candidates that challenge questions can be mined from. The algorithm then, concatenates pre-set strings to form the phrasing of the challenge question. This is stored alongside the query which will be used to fetch the correct response to the challenge question.

Figure 5.1 captures the code snippet used to fetch a challenge question for a given customer. In summary, the function fetches all active challenge questions (i.e. those that have not been retired). A look-up is then performed to get the set threshold for the

maximum number of times a challenge question can be posed to a user. Next, the *perUser* function iterates through each of the active challenge questions until one satisfying the threshold requirements is encountered. Once identified, the chosen challenge question is returned by the *fetchChallengeQuestion* function.

Figure 5.2 shows the workings of the *fetchAnswerSet* function. The IDs of the selected challenge question and customer are passed to the function. It then fetches the phrasing of the challenge question and query needed extract the correct answer from the unified data set. The correct answer, particular to the customer, is retrieved and the result is fed to the *generateMultipleChoices* function which returns an array containing both the phrasing of the *challenge question* and a its multiple-choice answer set.

```
235    public function fetch_challenge_question($customer_id)
236    {
237        //get active challenge questions
238        $questions = Question::where('status', 'active')->get();
239
240        //get threshold of the number of times a question can be repeated for the same customer
241        $threshold = Threshold::where('type', 'same_customer')->get();
242        $max = $threshold->maximum_allowable;
243
244        //check if the customer has been asked the question previously and if the threshold has been met
245        $selected_question;
246        foreach($questions as $q)
247        {
248            $question_id = $q->question_id;
249            $count = perUserCheck($question_id, $customer_id);
250
251            //if a given question has not met the threshold, then it is eligable to be used
252            if($count < $max)
253            {
254                $selected_question = $question_id;
255                break;
256            }
257        }
258        return $selected_question;
259    }
```

Figure 5.3: FetchChallengeQuestion Function.

```php
public function fetch_answer_set($question_id, $customer_id)
{
    //get phrasing of the challenge question
    $question = Question::find($question_id);

    //get the query needed to fetch the appropriate answer
    $query = $question->lookup_query;
    $query = $query." where customer_id = '".$customer_id."'";

    //get the *correct* answer to the question for the given customer
    $answer = DB::select($query, [1]);

    //generate multiple choice set of 4 choices with 3 distractors
    $set = array();
    $set = generateMultipleChoices($answer);

    return set();
}
```

Figure 5.4: FetchAnswerSet Function.

5.2.3.2 Administrator Dashboard

The administrator dashboard provides an interface using which the generated challenge questions can be managed, thresholds set and various reports viewed. Figure 5.3 depicts the landing page where, at a glance, the administrator is presented with various summaries and charts relating to the challenge questions and authentication process. In figure 5.4, we see the interface using which challenge questions can be managed. Various aspects of a particular challenge question can be viewed such as the query supporting it and the phrasing that will be presented to the customer. This phrasing can be modified by the system administrator who may also completely disable the challenge question if unsatisfied with it.

This module also handles the setting of security thresholds determining the retirement of challenge questions as seen in figure 5.5. Here, the administrator sets the maximum number of times a question should be posed in its lifetime as well as the maximum number of times a question should be posed to the **same** customer.

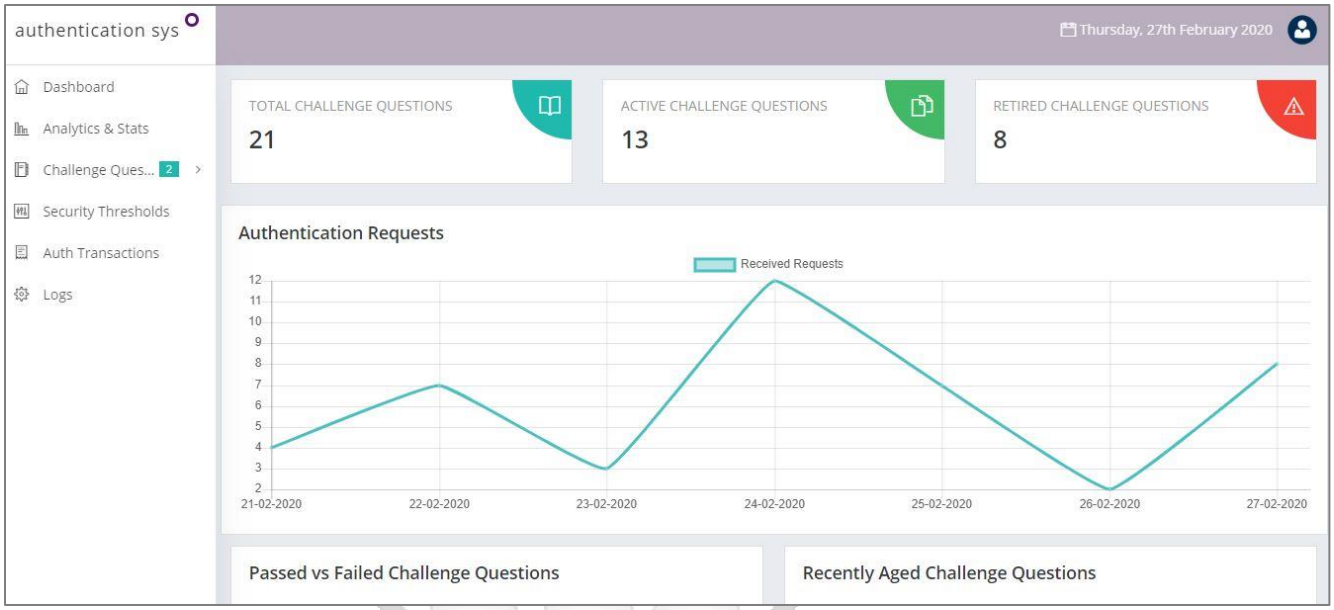


Figure 5.5: Administrator Dashboard.

5.2.3.3 USSD Handler

Transactions between the Authentication Manager (AM) and the USSD application are mediated by the USSD handler. Riding on *Africa's Talking* API, this component parses formatted responses to the user while also receiving user inputs using the same. Figure 5.6 provides a code snippet of how the USSD handler interacts with the API.

Screen captures of the USSD application as seen by the end user (bank customer) in the course of the authentication process are captured in Figure 5.7.

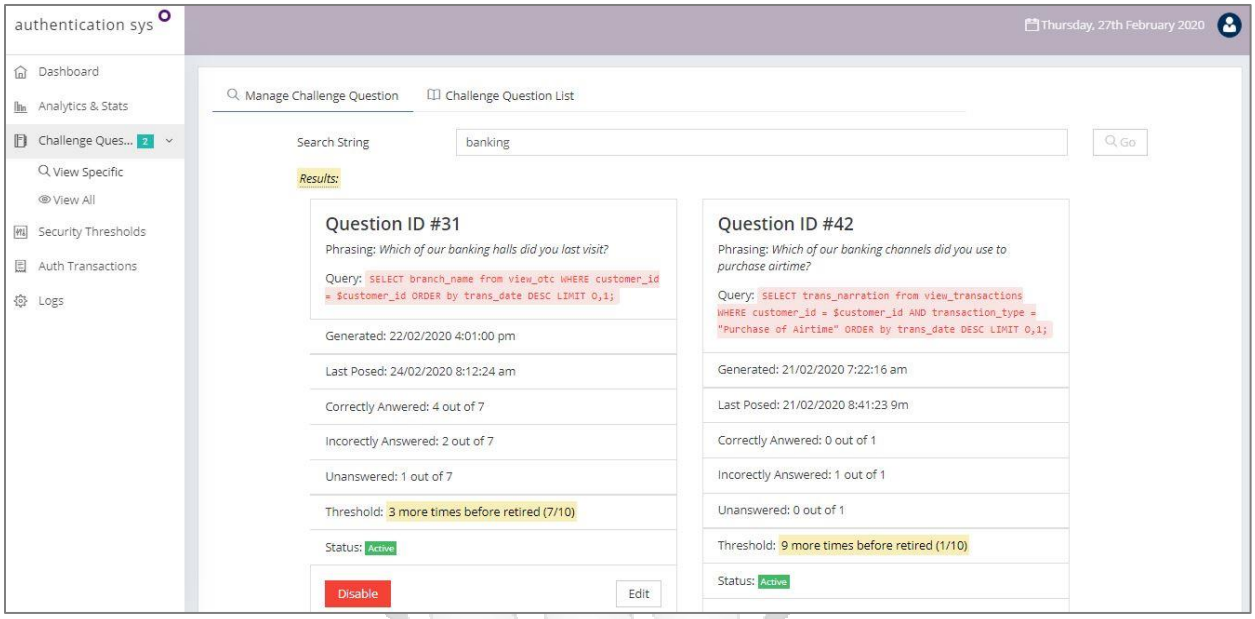


Figure 5.6: Manage Challenge Questions Interface.

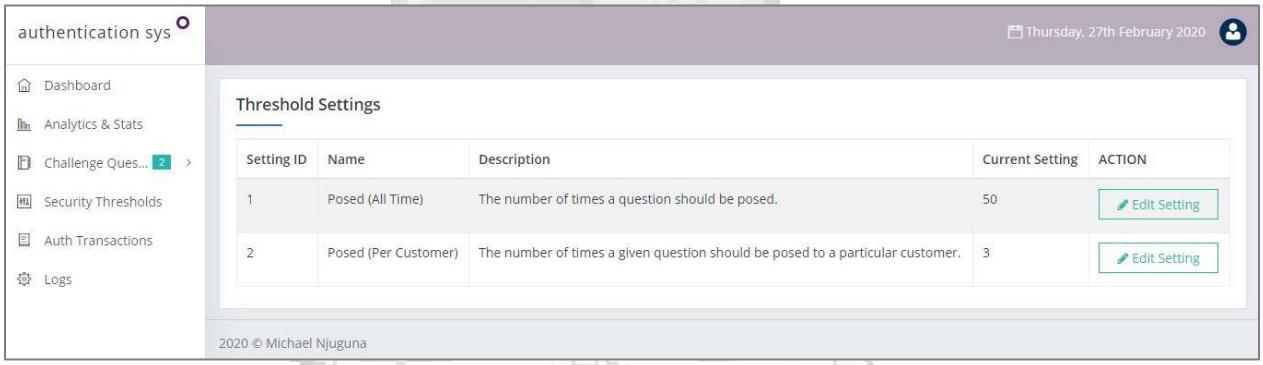


Figure 5.7: Manage Security Parameters Interface.

```php
// Reads the variables sent via POST from our Africa's talking gateway
$sessionId   = $_POST["sessionId"];
$serviceCode = $_POST["serviceCode"];
$phoneNumber = $_POST["phoneNumber"];
$text        = $_POST["text"];


//if user is attempting to withdraw cash
if ($text == "56421*2*3") {
    // Notify the user that they are about to be asked an authentication question
    $response  = "CON To proceed with this step, we need to confirm your identity
                  by asking some simple questions regarding your account. \n";
    $response .= "1. Proceed \n";
    $response .= "0. Go Back";

} else ($text == "1") {
    // Do nothing, dynamic authentication not required

}

// Echo the response back to the API
header('Content-type: text/plain');
echo $response;
```

Figure 5.8: USSD Handler Class.

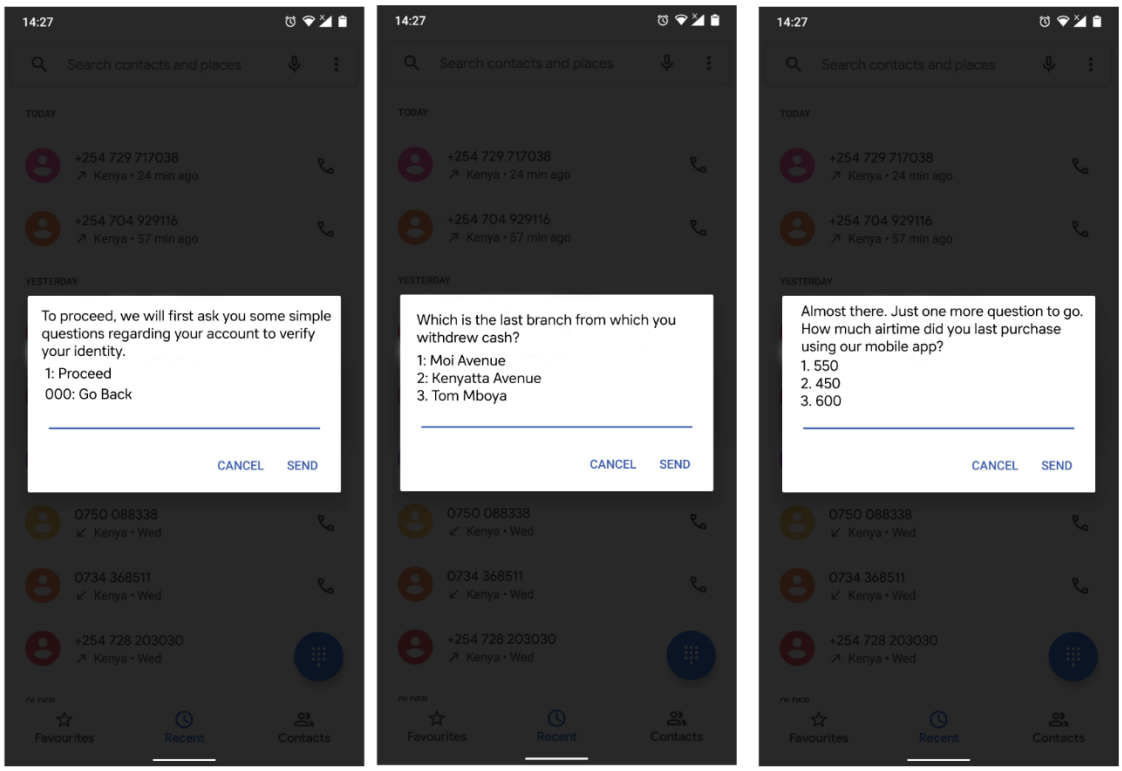


Figure 5.9: USSD Interfaces.

5.3 Prototype Testing

Agile methodology's test-driven approach is based on keeping initial iterations as simple as possible in order to satisfy requirements before proceeding to complete the system features in full (Beck, 2002). In a continual process, the prototype gradually took shape, passing all but one of the designed test scenarios. Table 5.3 details the test results.

Table 5.3: Prototype Test Case Report.

#	Test	Action	Expected Results	Verdict
1.	Automatic retirement of challenge questions meeting the set threshold.	Set the maximum number of times a given challenge question should be posed as five (5).	Logs should indicate that no challenge questions have been posed more than five times.	Pass
2.	Enforcement of challenge question	Set maximum number of times a given challenge	Logs should indicate that no customer has	Pass

	threshold per customer.	question should be posed to the same customer as two (2).	interacted with the **same** challenge question more than twice.	
3.	Prevent generation of duplicate challenge questions.	Examine and compare the semantics of the generated challenge questions for duplicates.	All challenge questions should be semantically unique.	Fail
4.	Posing of challenge questions **only** to fund transfer transactions meeting the defined threshold.	Attempt to transfer funds equal to or greater than the defined threshold.	Customer is prompted with a challenge question before being allowed to proceed transaction.	Pass.
5.	On submission of an incorrect response, presentation of a different challenge question.	Intentionally responding with an incorrect answer to a challenge question.	A different challenge question is posed.	Pass.

To validate the usability and security of the challenge questions, convenience sampling was used to identify subjects to test the USSD authentication prototype. The test subjects were divided into two categories: those that were allowed to generate their own test data ("legitimate" users) and those that were not ("adversarial" users).

With the test data available, eighteen challenge questions were selected and the posed to ten legitimate test users and ten adversarial test users. To visualize the results, a confusion matrix (Table 5.4) was used to analyse the outcomes.

Listed below are the meanings of the abbreviations in each of the quadrants of the confusion table:

i. True Acceptance [TA]: USSD sessions by legitimate users in which a correct response to the challenge question was given and authentication successful.

ii. False Rejection [FR]: USSD sessions by legitimate users in which a correct response to the challenge question was given and authentication **unsuccessful**.

iii. False Acceptance [FP]: USSD sessions by **adversarial** users in which an incorrect response to the challenge question was given and authentication **successful**.

iv. True Negative [TN]: USSD sessions by **adversarial** users in which an incorrect response to the challenge question was given and authentication unsuccessful.

Table 5.4: Confusion Table for Authentication Attempts.

		Authentication Attempt	
		Success	Fail
User Type	Legitimate	[TA] 167	[FR] 0
	Adversarial	[FA] 0	[TR] 124

**Correct guesses by adversarial users: 56* **Incorrect responses by legitimate users: 13*

In summary, legitimate test subjects were able to recall the correct responses 92.8 per cent of the time. On the other hand, adversarial users were able to correctly guess the appropriate response at a rate of 31.1 per cent.

The recall values of the posed challenge questions (the rate at which legitimate users were able to recall the correct answer) are captured in Table 5.5.

Table 5.5: Tested Challenge Questions and their Recall Value

	Challenge Question	Element Tested	Recall Rating
Q1.	Which branch did you last visit?	Place visited.	100%
Q2.	How much airtime did you last purchase using mobile banking?	Self-service action.	100%
Q3.	How much did you last deposit using mobile money?	Self-service action.	95%

Q4.	How much did you last withdraw using mobile banking?	Self-service action.	95%
Q5.	Which account did you last transfer money to?	Self-service action.	95%
Q6.	Which mobile banking transaction did you last perform?	Self-service action.	100%
Q7.	How much did you last borrow using our mobile banking services?	Self-service action.	100%
Q8.	Which bill did you last pay using our mobile banking services?	Self-service action.	90%
Q9.	Which ATM did you last withdraw funds from?	Place visited.	85%
Q10.	How many active accounts do you have with us?	Particular knowledge.	100%
Q11.	When was the last time you banked a cheque?	Particular knowledge.	95%
Q12.	What is the outstanding amount for the loan you have with us?	Particular knowledge.	100%
Q13.	How many debit cards do you have with us?	Particular knowledge.	100%
Q14.	How many credit cards do you have with us?	Particular knowledge.	100%
Q15.	Which asset did we last finance for you?	Particular knowledge.	95%
Q16.	What collateral was used for your last secured loan with us?	Particular knowledge.	95%
Q17.	When did you last request an account statement?	Self-service action.	100%
Q18.	Which of these areas did you last use one of our bank agents in?	Place visited.	90%

Figure 5.8 provides a breakdown of the fail rate for challenge questions categorized by the element tested. Challenge questions referencing a customer's particular knowledge scored the highest in recollection value, with those testing places visited by customers registering the lowest scores.

5.4 Performance Evaluation

A key metric in evaluating the effectiveness of the prototype was its ability to perform dynamic authentication within the duration allowed by USSD sessions. Responsiveness in fetching and presenting challenge questions formed the basis of performance evaluation. Wireshark logs were recorded to measure intervals between requests and responses within the USSD session while dynamic authentication was in progress.

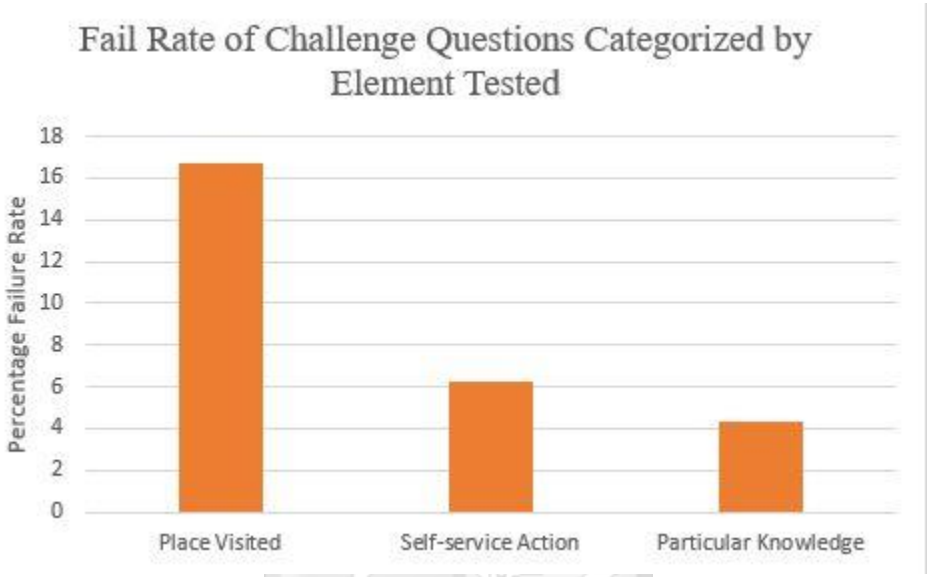


Figure 5.10: Fail Rate of Challenge Questions by Element Tested.

Figure 5.9 illustrates the average interval between requests and responses recorded during the presentation of various challenge questions. It can be seen that the longest wait time to be expected is four seconds.

Figure 5.10 provides a comparison of the "wait" intervals for USSD sessions utilizing dynamic authentication and those without dynamic authentication. Although our prototype performed comparatively slower than USSD sessions utilizing only PIN-based authentication, the difference (22%) was sufficiently inconsequential for use without intolerable delay. Moreover, as the performance of a USSD application is also influenced by the responsiveness of the server on which it resides, it remains possible to further reduce the performance difference by optimizing the deployment environment such as through the use of a hyperthreading, multi-core server.

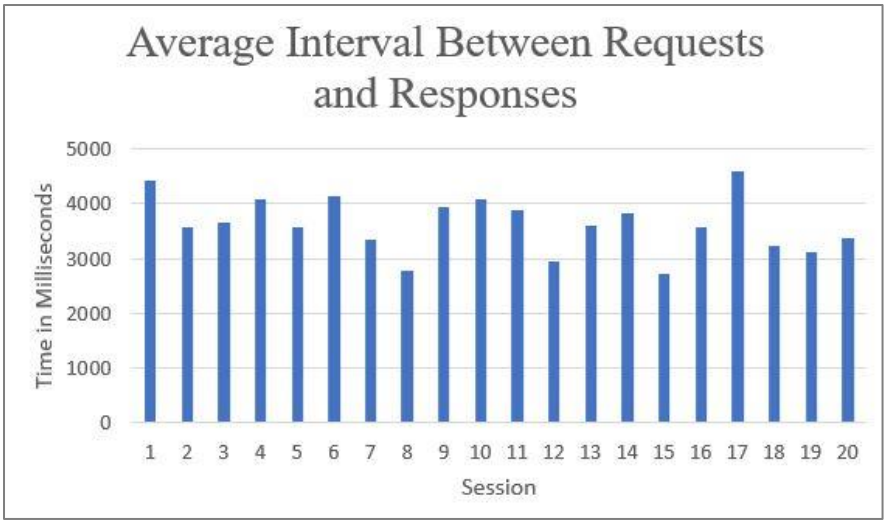


Figure 5.11: Average Interval Between Requests and Responses.

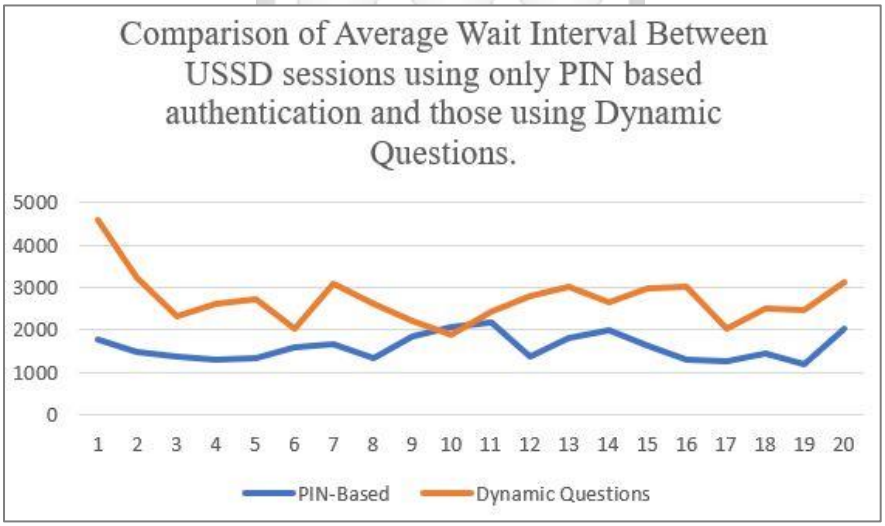


Figure 5.12: Comparison of Wait Intervals.

Chapter 6: Discussion of Key Findings

6.1 Overview

To arrive at a determination on whether this study has achieved its objectives, an interpretation of the test results is necessary. Whereas the validation tests appraise the degree to which system requirements have been satisfied, the prototype's applicability to real-world settings is dependent on its performance metrics. In this chapter, we examine key findings informing the conclusions in chapter 7.

6.2 Discussion

The study set out to achieve five objectives. In this sub-heading, we discuss the treatment each was accorded.

6.2.1 Assessment of the present implementation of USSD by Kenyan banks

Review of the state of the industry report on mobile money provided insight on the role played by USSD within the Mobile Financial Services (MFSs) sector. The report indicated that in Africa, 90% of the transactions were driven through USSD channels (GSM Association, 2019). PWC (as cited in Kiragu, 2015) examine the fraud landscape in the African banking sector and conclude ecosystems with matured mobile financial services are at the highest risk of fraud perpetuated by an organization's own employees. Further, it was observed that of the forty-four registered banks in Kenya, thirty possessed USSD codes for mobile baking (Safaricom PLC, 2020). Assessing the various deployments by Kenyan banks, it was noted that the common practice for customer authentication was use of a six-digit secret PIN. The vulnerabilities arising from reliance on this single authentication vector for identity verification informed the subsequent objective.

6.2.2 Analysis of vulnerabilities posed by the PIN-based authentication system with respect to privacy leakage, insider threat and social engineering.

The researcher sought to demonstrate the possibility of the shared secret being discovered and used by an illegitimate user. It was highlighted that bank USSD applications necessarily expose some level of data while communicating with the Mobile Network Operator's core network; data which includes sensitive PII. Privileged insiders working at the Mobile Network Operator could misuse their position to access plaintext USSD strings

containing customer mobile numbers and their mobile banking PINs (Taskin, 2012). Shifting focus to the bank itself, work by van Dijk and Haider (2019) illustrated privacy leakage exploits using which database contents (even when encrypted) were successfully reconstructed merely by observing database hits during read and write operations (something ostensibly within the reach of malicious system administrators). The study also examined the risk of sensitive PII being accessed by privileged insiders within the organization, which according to Cyber Security Insiders and Crowd Research Partners (2018) is targeted 52% of the time. Lastly, as technical exploits become more difficult thanks to tightened controls, the study gave due attention at the increasing prevalence of social engineering exploits (Guangxuan , et al., 2015).

6.2.3 Development of a dynamic, knowledge-based authentication mechanism that addresses the identified vulnerabilities

The study proposed a dynamic knowledge-based authentication mechanism to address the identified vulnerabilities. With the aid of reviewed literature, the system requirements were identified and a suitable methodology selected to guide methodology. The researcher then proceeded to design the system (producing the artefacts contained in chapter 4) and thereafter developed a working prototype as discussed in chapter 5.

6.2.4 Testing the efficacy of the proposed authentication mechanism.

Following development of the prototype, the researcher sought to evaluate its performance as well as the level to which it satisfied the identified system requirements. The performance of the prototype compared well-enough to that of current deployments relying solely on PIN-based authentication. The average "wait period" between requests and responses in the course of dynamic authentication was sufficiently small so as not to present usability challenges. With regard to validation testing, the prototype passed all but one of the test cases, with difficulties encountered in distinguishing between semantically different queries that returned the same values. However, all other requirements were satisfactorily met.

As for the security provided by the authentication method, it was observed that the generated challenge questions had a recall rate of 92.8 percent with the test users being able to correctly remember the answer to four out of every five challenge questions. The

recall rate of challenge questions was highest for those testing particular knowledge unique to the user (95.7 percent) and lowest for those testing places visited (83.8 percent). Although our test subjects did not *actually* visit any location and only generated their dummy data, this variance points to differences in the ease of recollection among various types of content.

Different from our test scenario (where subjects responded to challenge questions based on test-data that they inputted the very same day) real world users would face challenge questions referencing their account activity of up to the preceding forty-five days. It is therefore expected that the recall rate would be slightly lower in "live" environments. However, improvements on the quality of challenge questions would certainly aid in keeping the overall recall rate fairly high.

Also worthy of note is that "adversarial" test users were able to correctly guess the answer to challenge questions 31.1 per cent of the time. This figure is hardly surprising given that in the test scenarios, the multiple-choice set contained three options. To enhance the effectiveness of the scheme, either the number of multiple choices could be increased or the user could be presented with successive challenge questions. However, both of these options have practical limitations given that USSD sessions remain active for only two minutes and USSD menus can only fit a fixed amount text. For high value transactions (such as withdrawal or transfers of large sums) where identity verification is imperative, the practical upper limit for maximum security would be three challenge questions each with multiple-choice menus of five options. In this scenario, the likelihood of an adversarial user successfully authenticating himself would be reduced to only 0.8 per cent. Therefore, even if a hacker or scammer was able to obtain the mobile banking PIN of a bank customer, it is very unlikely that he would be able to transact with the account.

6.2.5 Analysis of the Merits and Demerits of the Prototype

The key highlight of the prototype was its enhanced resistance to the vulnerabilities afflicting the contemporary authentication method while at the same time maintaining an almost comparable level of performance. Social engineering attacks were made more difficult by the diversity of challenge questions, which were also retired upon attaining a threshold of repeated use. The social engineer would therefore have no way of reliably

predicting beforehand the unique element of customer knowledge would be tested in the USSD authentication process. As for privacy leakage exploits, this was made near-impossible given the length of time it would take to reconstruct a database of considerable size (weeks to months). Queries on a stale snapshot of the database would not be able to provide the correct (current) answers to challenge questions. Regarding insider threat, this was rendered more difficult as bypassing of the dynamic authentication mechanism would require the administrator to have real-time, unfettered access to the data warehouse in order to perform queries on the fly. Only one or two individuals (if any) would be assigned such a sensitive role in an organization's hierarchy.

However unlikely it may be, it is conceivable that a malevolent insider may be able to gain real-time, super administrator access to a data warehouse. In such an instance, the insider would be able to bypass the dynamic authentication mechanism. When presented with a challenge question, the hacker will merely query the data warehouse on the fly, retrieving the correct response before the USSD session times-out. Another drawback of the prototype is that the generated challenge questions are not all of the same "quality". For the highest-level security, it is desirable that the correct response to a challenge question be time-variable and prone to frequent change. As an example, a challenge question querying the last time a user used mobile banking services is of "high-quality". This is because the correct answer is likely to vary even within a short window of time. In contrast, a challenge question querying details on the most recent loan taken by a customer took is of "low-quality" for the reason that the correct response is likely to remain the same for weeks, months or even years. Lastly, the finite volume of available customer data combined with the limited recollection ability of the human mind means that there exists a practical limit on the number of challenge questions having an acceptable recall value. This, taken together with the fact that questions are retired after repeated use, means that eventually "fresh" challenge questions will be exhausted. At this point, the challenge questions may have to be repeated, albeit doing so would weaken the security of the mechanism.

Chapter 7: Conclusion and Recommendations

7.1 Conclusions

The study sought to enhance the security of USSD-mediated banking transactions through improvement of the authentication mechanism. To this end, the researcher reviewed various literature to gain a better understanding of the vulnerabilities posed by the present deployments. Most notably was the exposure of sensitive information to Mobile Network Operators during the course of USSD operation as described by Taskin (2012). Insider threats and social engineering exploits also proved to be effective means of extracting customer PII. In an eco-system already contending with rampant fraud, the weak, shared-secret authentication mechanism relied upon by bank USSD applications is insufficient to effectively safeguard transactions.

A dynamic knowledge-based authentication mechanism running atop a unified customer data set was designed and prototyped. The prototype effectively hardened the USSD application against social engineering attacks, insider threats and privacy leakage attacks. Testing was performed with respect to performance and validity, with the results indicating that the system requirements were satisfied and that responsiveness was adequate for real-world application. The strength of the proposed authentication mechanism lies in the quality and variety of the challenge questions, which are continually generated as new data sets become available. The proposed scheme demonstrably offers additional security without significantly affecting performance or usability of USSD applications.

7.2 Recommendations

The study has shown that is possible to enhance the security of bank USSD applications by incorporating dynamic authentication in the sessions. The researcher proposes several recommendations based on the findings made:

i. It was noted that the variety and quality of the generated challenge questions can be further improved by using data sets that have more varied characteristics. In the prototype, 27 challenge questions were generated from 7 candidate tables that held different types of transactional data (ATM, agent banking, mobile banking etc.). With additional candidate tables, more challenge questions can be generated.

ii. Due to the budgetary constraints of this research, a shared USSD code was used. However, the study recommends the use of a dedicated USSD code for optimal performance in servicing requests. The higher throughput would noticeably reduce wait-times between requests and responses especially when the application is under stress (high load).

7.3 Suggestions for Future Works

It was not possible for the study to examine the myriad factors contributing to the quality of challenge questions; certain aspects of which merit a study of their own. In particular, the study recommends further research in the area of Artificial Intelligence (AI) with respect to question generation. It was observed that the variety of the generated challenge questions was limited, partly because of the narrow characteristics of the data. AI implementations would therefore certainly prove useful in the extraction of a greater number of varied challenge questions meeting the dual requirements of "easy to remember but difficult to guess". Machine learning, for example, could help learn the sort of questions that are ideal for easy recollection while at the same time being difficult for anyone other than the legitimate user to deduce. The increased, varied and better-quality challenge questions would effectively harden the authentication mechanism.

Another area of interest for future work is further study on database query equivalence. In the validation testing, the prototype failed to distinguish between differently phrased queries that returned the same answer, though in different formats. This is as a result of the inability of MySQL database to accurately distinguish between semantically equivalent queries if they do not return exactly the same result. Therefore, without a means of discriminating query results, duplicate challenge questions would eventually be generated by the designed algorithm. The field of query equivalence therefore presents an exciting challenge whose fruits would be of use to this study, and countless others.

References

Abdelraheem, M. A., Gehrmann, C., Andersson, T., & Glackin, C. (2018). Practical Attacks on Relational Databases Protected via Searchable Encryption. *ICS.* Frankfurt.

Agile Alliance. (n.d.). *What is Agile?* Retrieved 04 22, 2019, from Agile Alliance: https://www.agilealliance.org/agile101/

Albayram, Y., Khan, M. M., Bamis, A., Kentros, S., Nguyen, N., & Jiang , R. (2015). Designing challenge questions for location-based authentication systems: a real-life study. *Human-centric Computing and Information Sciences, 5*(17). doi:https://doi.org/10.1186/s13673-015-0032-3

Asongu, S. A. (2018). Conditional Determinants of Mobile Phones Penetration and Mobile Banking in Sub-Saharan Africa. *Journal of the Knowledge Economy, 9*(1), 81-135.

Association of Certified Fraud Examiners. (2018). *2018 Global Study On Occupational Fraud and Abuse.* Texas: ACFE. Retrieved 11 16, 2019, from https://www.acfe.com/uploadedFiles/ACFE_Website/Content/rttn/2018/RTTN-Sub-Saharan-Africa-Edition.pdf

Assolini, F., & Tenreiro, A. (2019). *Large-scale SIM swap fraud: The enemy in your pocket.* Midrand: Kaspersky. Retrieved 03 31, 2020, from https://securelist.com/large-scale-sim-swap-fraud/90353/

Atikan, I. (2016). Comparison of Convenience Sampling and Purposive Sampling. *American Journal of Theoretical and Applied Statistics, 5*(1), 1-4.

Barkan, E., Biham, E., & Keller, N. (2008). Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication. *Journal of Cryptology, 21*(3), 392-429.

Beck, K. (2002). *Test-driven Development: By Example* (1st ed.). Bboston: Addison-Wesley Professional.

Bittner, K., & Spence, I. (2003). *Use Case Modeling.* Boston: Addison-Wesley Professional.

Borde, D. S., Hebare, P. A., & Dhanedhar, P. D. (2017). Overview Of Web Password Hashing Using Salt Technique. *International Research Journal of Engineering and Technology (IRJET), 4*(11), 152-154. Retrieved 04 21, 2019, from https://www.irjet.net/archives/V4/i11/IRJET-V4I1126.pdf

Buku, M. W., & Mazer, R. (2017). *Fraud in Mobile Financial Services: Protecting Consumers, Providers, and the System.* Washington: CGAP. Retrieved 11 17, 2019, from https://www.cgap.org/sites/default/files/Brief-Fraud-in-Mobile-Financial-Services-April-2017.pdf

Communications Authority of Kenya. (2015). Registration of SIM Cards Regulations. *The Kenya Information and Communications Act*.

Communications Authority of Kenya. (2018). *First Quarter Statistics Report for the Financial Year 2018/2019 (July-September 2018).* Nairobi: CA. Retrieved 03 28, 2019, from https://ca.go.ke/wp-content/uploads/2018/12/Sector-Statistics-Report-Q1-2018-2019.pdf

Cuesta, H., & Kumar, S. (2016). *Practical Data Analysis* (2nd ed.). Birmingham: Packt Publishing.

Cyber Security Insiders & Crowd Research Partners. (2018). *Insider Threat 2018 Report.* Retrieved 04 17, 2019, from http://crowdresearchpartners.com/wp-content/uploads/2017/07/Insider-Threat-Report-2018.pdf

Edlund, L. (2009). *Secure and Confidential Applications on UICC (Masters Thesis).* KTH Royal Institute of Technology, Stockholm. Retrieved 07 16, 2019, from https://www.nada.kth.se/utbildning/grukth/exjobb/rapportlistor/2009/rapporter09/edlund_lasse_09084.pdf

Fatima, R., Siddiqui, N., Umar, M. S., & Khan, M. H. (2019). A Novel Text-Based User Authentication Scheme using Pseudo Dynamic Password. *Information and*

Communication Technology for Competitive Strategies, 177-186. doi:https://doi.org/10.1007/978-981-13-0586-3_18

Grbich, C. (2012). *Qualitative Data Analysis: An Introduction* (2nd ed.). Thousand Oaks: Sage Publications.

GSM Association. (2018). *2017 State of the Industry Report on Mobile Money.* World: GSMA. Retrieved 03 27, 2019, from https://www.gsma.com/mobilefordevelopment/wp-content/uploads/2018/05/GSMA_2017_State_of_the_Industry_Report_on_Mobile_Money_Full_Report.pdf

GSM Association. (2018, 05 1). *Mobile for Development.* Retrieved 04 01, 2019, from GSMA: https://www.gsma.com/mobilefordevelopment/wp-content/uploads/2018/05/Start-ups-using-USSD-to-tackle-SDGs.pdf

GSM Association. (2019, April 7). *Mobile Money Glossary*. Retrieved from GSMA: https://www.gsma.com/mobilefordevelopment/mobile-money/glossary/

GSM Association. (2019). *State of the Industry Report on Mobile Money.* Retrieved June 11, 2019, from https://www.gsma.com/mobilefordevelopment/wp-content/uploads/2019/02/2018-State-of-the-Industry-Report-on-Mobile-Money.pdf

Guangxuan , C., Guomin, Z., Zhoujie, M., Qiang, L., Ziwan , Z., Guangxiao , C., & Panke , Q. (2015). Research of Social Engineering Attacks in Telecommunications Fraud. *International Conference on Social Science, Education Management and Sports Education* (p. 4). Beijing: Atlantis Press. doi:https://doi.org/10.2991/ssemse-15.2015.477

Gupta, P. (2008). *System Analysis and Design* (3rd ed.). New Delhi: Firewall Media.

Hashcat. (2019). *Hashcat Features*. Retrieved 04 21, 2019, from Hashcat: https://hashcat.net/hashcat/

Hastings, N. E., & Dodson, D. F. (2004). Quantifying Assurance of Knowledge Based. *3rd European Conference on Information Warfare and Security* (pp. 109-116). London, UK : Typeset by Academic Conferences Limited .

Huang, D., Gu, Y., Wang, H., Liu, Z., & Chen , J. (2018). An Incentive Dynamic Programming Method for the Optimization of Scholarship Assignment. *Discrete Dynamics in Nature and Society, 2018*. doi:https://doi.org/10.1155/2018/5206131

IBM. (2018). *IBM X-Force Threat Intelligence Index 2018.* Retrieved 04 17, 2019, from https://www.ibm.com/downloads/cas/MKJOL3DG

IDology. (2019, 11 18). *Dynamic KBA: Out-of-wallet questions to deter fraud.* Retrieved from IDology: https://www.idology.com/dynamic-kba/

Islam, M., Kuzu, M., & Kantarcioglu, M. (2012). Access Pattern disclosure on Searchable Encryption: Ramification, Attack and Mitigation. *NDSS*, (p. 15). Retrieved 04 20, 2019, from https://pdfs.semanticscholar.org/9614/87973d4b33f96406fddbfcf1235dc587571f.pdf

Ivo, A. A., Guerra, E. M., Porto, S. M., Choma, J., & Quiles, M. G. (2018). An approach for applying Test-Driven Development (TDD) in the development of randomized algorithms. *Journal of Software Engineering Research and Development, 6*(9). Retrieved 02 24, 2020, from https://link.springer.com/article/10.1186/s40411-018-0053-5

John, T. M., Haider, S. K., Omar, H., & Van Dijk, M. (2017). Connecting the Dots: Privacy Leakage via Write-Access Patterns to the Main Memory. *IEEE Transactions on Dependable and Secure Computing*. Retrieved 04 20, 2019, from https://arxiv.org/pdf/1702.03965.pdf

Johnson, B. R., & Onwuegbuzie, A. J. (2004). Mixed Methods Research: A Research Paradigm Whose Time Has Come. *Educational Researcher, 33*(7), 14-26.

Katsini, C., Belk, M., Fidas, C., Avouris, N., & Samaras, G. (2016). Security and Usability in Knowledge-based User Authentication: A Review. *20th Pan-Hellenic Conference on Informatics.* Patras, Greece. Retrieved 11 18, 2019, from http://mbelk.info/files/conferences/PCI2016.pdf

Kaujalgi, V. B. (1994). *Structured Systems Analysis and Design: Data Flow Approach.* Hyderabad, India: Orient Blackswan.

Khidzir, N. Z., Ismail, A. R., Daud, K. A., Afendi, M. S., Ghani, A., & Ibrahim, M. A. (2016). Critical Cybersecurity Risk Factors in Digital Social Media: Analysis of Information Security Requirements. *Lecture Notes on Information Theory, 4*(1), 18-24. doi:10.18178/lnit.4.1.18-24

Kimball, R., & Caserta, J. (2011). *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data.* Indianapolis: John Wiley & Sons.

Kiragu, D. N. (2015). Bank Size and Occupational Fraud Risk: Eemperical Evidence from Commercial Banks in Kenya. *European Journal of Business Management, 2*(1), 189-203. Retrieved 04 01, 2019, from http://repository.dkut.ac.ke:8080/xmlui/bitstream/handle/123456789/395/David_Ndungu_Kiragu-Journal-2.pdf?sequence=3

Kiran, P. R., & Krishna, Y. K. (2014). A Study Report on Authentication Protocols in GSM. *International Journal of Engineering Research and Development, 10*(6), 42-48. Retrieved 07 16, 2019, from https://pdfs.semanticscholar.org/0108/ec77ee0efddc6281e73939daf6411e394fe4.pdf

KPMG. (2019). *Global Banking Fraud Survey.* KPMG International Cooperative. Retrieved 11 16, 2019, from https://assets.kpmg/content/dam/kpmg/xx/pdf/2019/05/global-banking-fraud-survey.pdf

Kumar, R. (2011). *Research Methodology: A step-by-step guide for beginners* (3rd ed.). London: Sage Publishers.

Kwon, T., & Moon, H. (2007). Knowledge-Based User Authentication Associated with Biometrics. *Universal Acess in Human Computer Interaction. Coping with Diversity, 4554*. doi:https://doi.org/10.1007/978-3-540-73279-2_46

Laravel LLC. (2019, 11 20). *Database: Seeding*. Retrieved from Laravel: https://laravel.com/docs/master/seeding

LexisNexis. (2012). *White Paper: Financial Services Identity Management in an era of tell-all and technology overload.* New York: LexisNexis. Retrieved 11 20, 2019, from http://lexisnexis.com/risk/downloads/idm/financial-services-identity-management.pdf

Liu, C., Zhu, L., Wang, M., & Tan, Y.-A. (2014). Search pattern leakage in searchable encryption: Attacks and new construction. *Information Sciences, 265*(1), 176-188. Retrieved 04 20, 2019, from https://eprint.iacr.org/2013/163.pdf

Magutu, P. O., Mwangi, M., Nyaoga, R. B., Ondimu, G. M., Kagu, M., Mutai, K., . . . Nthenya, P. (2011). E-Commerce Products and Services in the Banking Industry: The Adoption and Usage in Commercial Banks in Kenya. *Journal of Electronic Banking Systems* , *2011*(2011). Retrieved 02 24, 2019, from https://ibimapublishing.com/articles/JEBS/2011/678961/678961.pdf

Mitnick, K. D., & Simon, W. L. (2011). *The Art of Deception: Controlling the Human Element of Security.* New Jersey: John Wiley & Sons.

Mjølsnes, S. F., & Olimid, R. F. (2017). Easy 4G/LTE IMSI Catchers for Non-Programmers. *7th International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security*, (pp. 235-246). Warsaw. Retrieved 07 16, 2019, from https://link.springer.com/chapter/10.1007/978-3-319-65127-9_19

Mlachila, M., Park, S. G., & Yabara, M. (2013). *Banking in Sub-Saharan Africa.* Washington DC: International Monetary Fund. Retrieved 04 22, 2019, from https://www.imf.org/external/pubs/ft/dp/2013/afr1303.pdf

Mouton, F., Leenen, L., & Venter, H. (2016). Social engineering attack examples, templates and scenarios. *Computers & Security, 59,* 186-209. doi:10.1016/j.cose.2016.03.004

Mulliner, C., Borgaonkar, R., Stewin, P., & Seifert, J.-P. (2019). SMS-Based One-Time Passwords: Attacks and Defense. *10th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (pp. 150-159). Berlin: Springer. Retrieved 11 18, 2019, from https://link.springer.com/chapter/10.1007/978-3-642-39235-1_9

Myriad Connect. (2018). *Kenya Fraud Report: Digital & Mobile Financial Transaction Fraud 2018 (White Paper).* Retrieved July 16, 2019, from https://connect.myriadgroup.com/resources/kenya-digital-mobile-transaction-fraud-report-2018/

Novak, R. (2003). Side-Channel Attack on Substitution Blocks. *Applied Cryptography and Network Security*, 307-318. Retrieved 03 31, 2020, from https://link.springer.com/chapter/10.1007/978-3-540-45203-4_24

Nwabueze, E. E., Obioha, I., & Onuoha, O. (2017). Enhancing Multi-Factor Authentication in Modern Computing. *Scientific Research Publishing: Communications and Network, 9,* 172-178. Retrieved 11 18, 2019, from https://www.scirp.org/pdf/CN_2017080814282111.pdf

Nyamtiga, B. W., Anael , S., & Laizer, L. S. (2013). Security Perspectives For USSD Versus SMS In Conducting Mobile Transactions: A Case Study Of Tanzania. *International Journal of Technology Enhancements and Emerging Engineering Research, 1*(3), 38-43.

Okamoto, M. (2013). Knowledge-Based Authentication Using Twitter: Can We Have Lunch Menus and Passwords? *International Journal of Network Security & Its Applications, 5*(5). doi:10.5121/ijnsa.2013.5501

Ranchal, R., Bhargava, B., Othmane, L. B., Lilien, L., Kim, A., Kang, M., & Linderman, M. (2010). Protection of Identity Information in Cloud Computing without Trusted Third Party. *29th IEEE Symposium on Reliable Distributed Systems* (pp. 368 - 372). Delhi: IEEE. doi:10.1109/SRDS.2010.57

Raykova, M., Zhao, H., & Bellovin, S. M. (2012). Privacy Enhanced Access Control for Outsourced Data Sharing. *15th International Confrence for Financial Cryptography*, (pp. 223-238). Kralendijk, Bonaire. Retrieved 04 19, 2019, from http://www.cs.columbia.edu/~mariana/papers/ac-cloud.pdf

Ricks, T. A., Ricks, B. E., & Dingle, J. (2014). *Physical Security and Safety: A Field Guide for the Practitioner.* Florida: CRC Press .

Safaricom PLC. (2019, July 16). *Safaricom Indroduces Voice Biometrics to Enhance Customer Experience: Press Release*. Retrieved from Safaricom: https://www.safaricom.co.ke/about/media-center/publications/press-release/release/408

Safaricom PLC. (2020). *Mobile Banking Codes / Paybills / Bank to Mpesa*. Retrieved 02 27, 2020, from Safaricom: https://www.safaricom.co.ke/images/Downloads/Mobile_Banking_Codes_Paybill.pdf

Sapovadia, V. (2018). Financial Inclusion, Digital Currency, and Mobile Technology. *Handbook of Blockchain, Digital Finance, and Inclusion, 2*(1), 361-385. doi:https://doi.org/10.1016/B978-0-12-812282-2.00014-0

Serianu. (2017). *Kenya Cyber Security Report: 2017.* Nairobi: Serianu. Retrieved 03 24, 20, from https://www.serianu.com/downloads/KenyaCyberSecurityReport2017.pdf

Skračić, K., Pale, P., & Jeren, B. (2014). Knowledge based authentication requirements. *2013 36th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1422-1427). Opatija, Croatia: IEEE.

Smith, R. E. (2001). *Authentication: from passwords to public keys.* Boston: Addison-Wesley Longman Publishing Co., Inc.

Southwest Educational Development Laboratory. (2005). *What Are the Standards for Quality Research?* Technical Brief. Retrieved 04 22, 2019, from https://ktdrr.org/ktlibrary/articles_pubs/ncddrwork/focus/focus9/Focus9.pdf

Susmaga, R. (2004). Confusion Matrix Visualization. *Intelligent Information Processing and Web Mining, 25*, 107-116.

Svennesson , B. A., & Bo , A. V. (1994). *US Patent No. US5752188A.*

Tashakkori, A., & Teddlie , C. (2010). *Handbook of Mixed Methods in Social & Behavioral Research* (2nd ed.). Thousand Oaks: Sage Publications.

Taskin, E. (2012). *GSM MSC/VLR Unstructured Supplementary Service Data (USSD) Service.* Uppsala: Uppsala University. Retrieved 11 16, 2019, from http://uu.diva-portal.org/smash/get/diva2:587744/FULLTEXT01.pdf

Toorani , M., & Beheshti, A. A. (2008). Solutions to the GSM Security Weaknesses. *The Second International Conference on Next Generation Mobile Applications, Services, and Technologies* (pp. 576 - 581). Cardiff: IEEE.

University of Conneticut. (2019). *Activity 3 – Requirements Analysis*. Retrieved 04 22, 2019, from UCONN: https://sdlc.uconn.edu/activity-3-requirements-analysis/

van Dijk, M., & Haider, S. K. (2019). Flat ORAM: A Simplified Write-Only Oblivious RAM Construction for Secure Processors. *MPDI Cryptography, 3*(1), 25. doi:https://doi.org/10.3390/cryptography3010010

Walliman, N. (2011). *Research Methods: The Basics.* London: Routledge.

Appendix A: Budget

Research Budget

Listed in the table below are cost approximations for expenditures to be incurred in performance of the research.

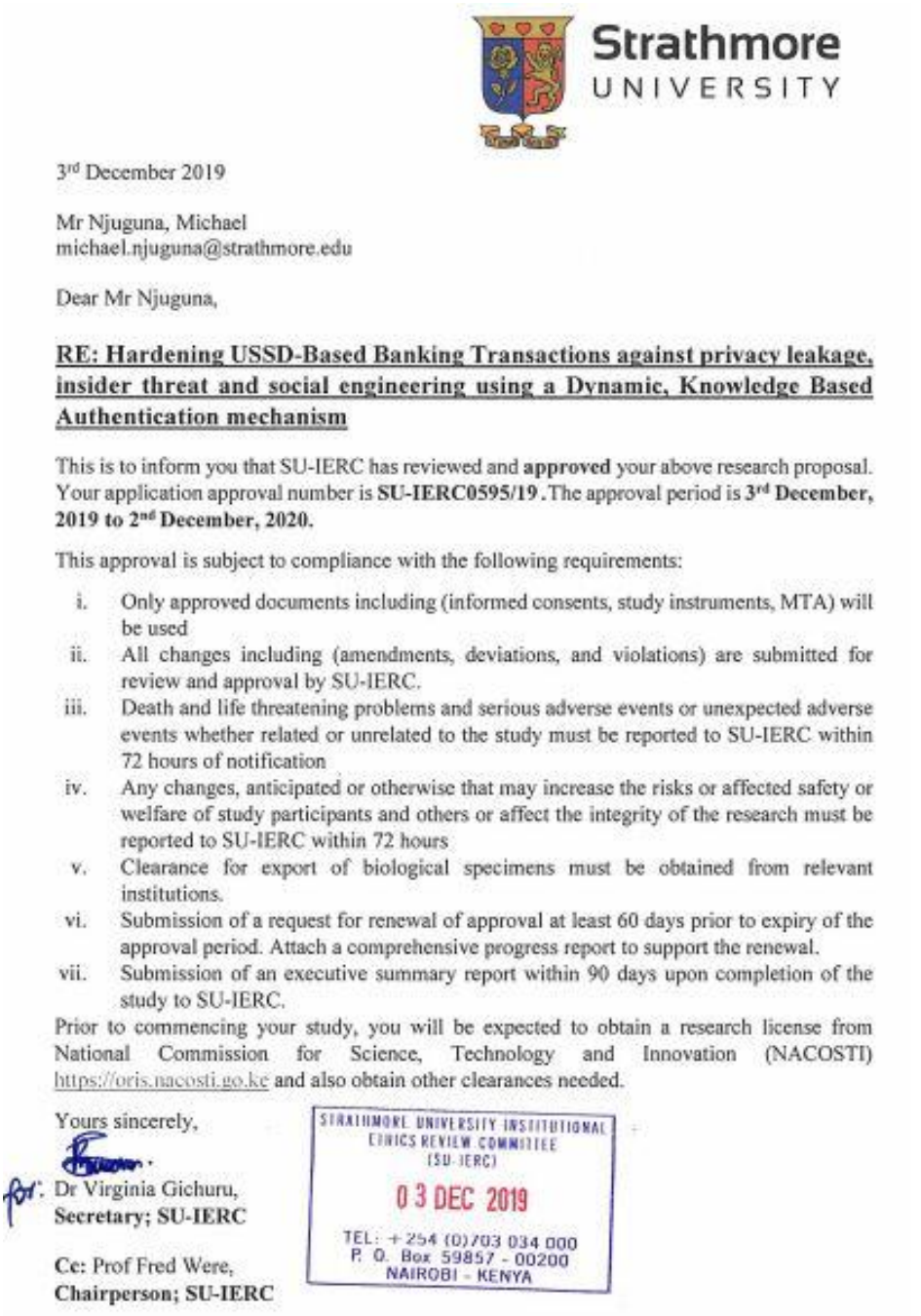
Item	**Cost Type**	**Units**	**Cost (KES.)**	**Duration Required**	**Total**
Shared USSD code from Africa's Talking	Monthly	N/A	5,800	4 Months	23,200
Domain	Annual	N/A	1,500	4 Months	1,500
Hosting	Annual	N/A	5,000	4 Months	5,000
Contingencies (10% of overall cost)	N/A	N/A	N/A	N/A	2,970
				Grand Total	**32,670**

Budget Notes

1. The "active" development and testing period will span four months (December 2019 to March 2020).
2. Domain and hosting charges must be paid annually and cannot be pro-rated.
3. The shared USSD code is billed monthly and cannot be pro-rated.
4. The allocation for contingencies is funds set aside to cover unexpected costs during the research process. This money is on reserve and not allocated to any specific budget item.

Appendix B: Ethical Approval

The researcher obtained approval to carry out the study from Strathmore University Institutional Ethics Review Committee (SU-IERC).



Strathmore UNIVERSITY

3rd December 2019

Mr Njuguna, Michael
michael.njuguna@strathmore.edu

Dear Mr Njuguna,

RE: Hardening USSD-Based Banking Transactions against privacy leakage, insider threat and social engineering using a Dynamic, Knowledge Based Authentication mechanism

This is to inform you that SU-IERC has reviewed and **approved** your above research proposal. Your application approval number is **SU-IERC0595/19**. The approval period is **3rd December, 2019 to 2nd December, 2020.**

This approval is subject to compliance with the following requirements:

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

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

Yours sincerely,

for: Dr Virginia Gichuru,
Secretary; SU-IERC

STRATHMORE UNIVERSITY INSTITUTIONAL ETHICS REVIEW COMMITTEE (SU-IERC)
03 DEC 2019
TEL: +254 (0)703 034 000
P. O. Box 59857 - 00200
NAIROBI - KENYA

Cc: Prof Fred Were,
Chairperson; SU-IERC

Appendix C: Originality Report

The similarity score (the percentage of the paper that matches to other sources) of the study as calculated by *TurnitIn* was 7 percent.