# ADDITIONAL SECURITY COUNTER MEASURES FOR MOBILE BANKING APPLICATIONS AND OTHER DEVICES

UNDERGRADUATE THESIS REPORT

**Computer Security and Forensics BSc (Hons)**

**University of Bedfordshire**
**Department of Computer Science & Technology**

MAY 5, 2017

AUTHOR: JUTEN AHMED
STUDENT NO: 1306929
SUPERVISOR: DR ALI MANSOUR

# Abstract

Throughout the last 20 years' technology has advanced tremendously. This has had its pros and cons, however having more technology has increased the amount of cybercrime taking place. This means there is an increase in security countermeasures which need to be put in place to stop any cybercrime from being a success.

In this document, there is information regarding the security countermeasures for mobile banking applications and any other devices. Majority of mobile banking applications use a one-step authentication system. This can easily be intercepted by using social engineering for example shoulder surfing to obtain the information of pin codes or memorable words.

In this project, the outcome is an artefact which involves a two-step security system to counter the main type of social engineering techniques of which are tail gaiting and shoulder surfing. This is done by creating a randomized numeric keypad system and a standard memorable word system on a graphical user interface.

In this thesis, there is market research completed in-depth to see what other companies use for their security countermeasure on mobile banking, furthermore, the creation and development the actual product for demonstration of a working prototype.

# Acknowledgements

I would like to thank Dr Ali Mansour for providing excellent resources and support to gain a better understanding of this new idea of authentication, furthermore, all the support provided through communication.

I would like to thank the University of Bedfordshire for allowing me to use their online resources and library to gain access to information to further my research.

# Dedication

*"Change your thought process from "I hope…" to "**I WILL**…!" and "**I ALREADY HAVE**…!".*
*If you do this, I promise, you will reach your goals in a matter of no time. Just don't give up."*
**– Juten Ahmed Zamindar**

I have had tremendous amount of support from my immediate family, whilst going through the hardship of my current situation.

My immediate family have always been there to support me with advice and guidance to help me get through my undergraduate degree and generally in life.

I am grateful for you all.

I appreciate you all.

Therefore, this thesis is dedicated to my loving, patient and caring parents and my smart, fun and awesome brothers and sister.

In addition, a special thanks to my eldest brother who motivated me to go university and study a computer science related course.

*"People are literally dying to come and study in the UK. You were born and brought up in the UK. How can you not study to at least a degree level, whilst knowing so many people would give up so much to be in your place?"*
**– R Ahmed**

# Table of Contents

Abstract ..... 2
Acknowledgements ..... 3
Chapter 1: Introduction ..... 8
1.0 Background of the Project ..... 9
1.1 Research Question ..... 9
1.2 Aims and Objectives of the research ..... 9
1.3 Research methods ..... 10
1.4 Scope and limitations of the research/project ..... 11
1.5 Thesis Structure (Summary of each chapter) ..... 11
Chapter 2: Literature Review ..... 13
2.1 Related work in Gaze Based Authentication ..... 13
2.2 Related work in area of Graphical Based Authentication ..... 15
2.3 Related work in area of Colour Light Based Authentication ..... 16
2.4 Related work in area of the Banking Industry ..... 17
2.5 SteganoPIN ..... 20
2.6 Biometrics ..... 22
2.7 Summary ..... 23
Chapter 3: Design & System Implementation of Authentication System ..... 24
3.1 Design ..... 24
3.2 System Implementation ..... 25
3.2.1 Registration Page ..... 25
3.2.2 Login Page ..... 26
3.2.3 Memorable Word Authentication Page ..... 27
3.2.4 Randomised PIN Authentication Page ..... 28
3.2.5 Success Page ..... 30
3.3 AES Encryption and Local Storage ..... 30
3.4 Integrity ..... 32
3.5 Coding ..... 33
3.6 Summary ..... 33
Chapter 4: Testing of Authentication System ..... 34
4.1 Testing Strategy ..... 34
4.2 Testing of Registration Page ..... 34
4.3 Testing of Login Page ..... 39
4.4 Testing of Memorable Word Page ..... 40

4.5 Testing of Randomized PIN Page ..... 42
4.6 Testing of Success Page ..... 47
4.7 Summary ..... 48
Chapter 5: Analysis and Evaluation of Test Results ..... 49
5.1 The Artefact ..... 49
5.2 Testing JA Zamindar Authentication with the Public. ..... 49
5.3 Suggested Improvements ..... 51
5.4 Advantages of the Artefact ..... 51
5.5 Disadvantages of the Artefact ..... 51
5.6 Summary ..... 52
Chapter 6: Comparison of Work to Previous Published Work ..... 53
6.1 UK Patent Application: GB0210322.4 ..... 53
6.2 UK Patent Application: GB0623944.6 ..... 54
6.3 Gaze Based Authentication ..... 55
6.4 Colour Light Based Authentication ..... 56
6.5 Banking Industry ..... 56
6.6 SteganoPIN ..... 57
6.7 Biometrics ..... 57
6.8 Summary ..... 58
Chapter 7: Conclusions and Future Work ..... 59
7.1 Conclusions ..... 59
7.2 Summary of contributions ..... 59
7.3 Personal Reflection ..... 60
7.4 Future Work ..... 60
7.4.1 Username Being Between a Certain Amount of Characters ..... 60
7.4.2 Username Can Include All Types of Characters ..... 61
7.4.3 Memorable Word to Include Any Characters ..... 61
7.4.4 A Longer PIN ..... 61
7.4.5 Memorable Word Authentication Page Can Require More Than 2 Characters ..... 61
7.4.6 Memorable Word Cannot Have 3 Repetitive Letters ..... 61
Bibliography ..... 62
Appendix A - Introduction ..... 1
A.1 Gantt Chart ..... 1
Appendix B - Design and System Implementation ..... 1
B.1 Code ..... 1
B.1.1 GUI Login Page: ..... 1

B.1.2 GUI PIN Page: 2
B.1.3 GUI Memorable Word Page: 5
B.1.4 GUI Registration Page: 6
B.1.5 GUI Success Page: 8
B.1.6 PIN Page: 9
B.1.7 AES Encryption 256 bit: 12
B.1.8 Encryption Password: 14
B.1.9 Store to Local Device (local storage): 14
B.1.10 Login Page for user: 17
B.1.11 Memorable Word Page: 18
Appendix C – Comparison of Work to Previous Published Work 1
C.1 Patent Application No: 0210322.4 1
C.2 Patent Application No: 0623944.6 26
Appendix D – Survey for Market Research 0
Appendix E – Thesis Poster 0

# **Chapter 1:** Introduction

Computer Security is a vital part of this day and age due to the vast amount of cyber-crime happening worldwide. This thesis report has been created to establish the current market and a detailed report into the final artefact. The artefact is based on a new concept of authentication. Authenticating using a 2-step verification system. This can be implemented into banking applications to decrease the success rate of personal identification numbers being stolen.

This project has been created to develop and help counter social engineering techniques used by anyone trying to commit cyber-crime which involves withdrawing sensitive data from the lawbreakers' target. The main goal of this project is to help increase the security of mobile banking applications and any mobile device by using a two-step authentication process to access any data or information displayed on the user's mobile devices.

Cybercrime has been on an increase within the last 10 years (Wealth & Finance International, 2015), therefore the increase has caused many types of cybercrime to take place. The two-step authentication process will allow any mobile banking or device users to detour anyone using social engineering techniques for example tailgating or shoulder surfing due to the fact of the nature of a randomized password matrix. By creating a randomized PIN keypad system, the difficulty of a keen observer or someone with malicious intent to withdraw the information which has been inputted into the mobile device is substantially difficult as the observer would generally think the keypad is the same as a calculator.

The overall objectives of this project are to create a randomized PIN keypad with an additional memorable word menu to input to increase the security by making this a two-step authentication process. The final artefact must allow the user to input the information they have been provided, which then enters a new page to allow the user to know they have gained access to their account.

To realise this artefact there must be a working prototype which is tangible. This artefact has been created as an android application to allow the user to be able to input their information to test the security countermeasures, and to test the integrity of the concept of detouring tailgating and shoulder surfing techniques. Furthermore, the prototype must be able to meet all the requirements to demonstrate how it differs from other authentication methods which are in place.

## 1.0 Background of the Project

This project is based on a 2-step authentication process which helps deter social engineering techniques i.e. shoulder surfing. This project uses a memorable word authentication process and then a new method of randomized numbers to login using a 3x4 matrix. The numbers 0-9 are randomized to make it much more difficult for someone whom is looking to steal the password. Furthermore, an additional aspect is having 2 blank numbers which also completes the 3x4 matrix making it more difficult for the social engineer to extract the passwords.

## 1.1 Research Question

Within this thesis, the main concept is to discuss this new two-step authentication process. Does this new authentication process increase data security and how? Will this new authentication process increase data security?

This thesis covers all aspects of how the two-step authentication process helps increase data security and how the whole artefact works.

## 1.2 Aims and Objectives of the research

The aims of this research are to create a secure authentication system of which banks can implement into their current applications. Furthermore, a new authentication process which increases the security of mobile banking application due to the fact there is very sensitive data within banking and personal banking. This new authentication process helps minimize the risk of shoulder surfers from gaining the PIN code.

Objectives of this project are as follows:

- Conducting thorough research into the cyber security sector regarding authentication and what is currently on the market.
- To create an authentication process which would be very difficult to break using a 2-step authentication system and to create a successfully working prototype for demonstration.

- To create a fully working GUI which can demonstrate a login screen transitioning into a generic “bank account” screen which tells the customer the login was successful with a welcome message.
- Creating a randomized PIN keypad as a 3x4 matrix as one of the authentication steps and creating a memorable word system where the user inputs 1 character to access the second stage of the authentication process.
- To test all the validations of the prototype.

## 1.3 Research methods

Within this project, the methodology used was RAD (Tutorials Point, 2015), Rapid Application Development and aspects of project management skills (PRINCE2) which included creating a Gantt Chart to help manage this project. [**Appendix A** - A.1 Figure 1: Gantt Chart]

The structural process of rapid application development is:

**Business Modelling** which involves doing some market research to find out if the product is out there or not.

**Data Modelling** which is all about designing the product to the specifications of what the business modelling phase has provided.

**Process Modelling** involves any changes that needs to be made to help the flow of the business the product is made for.

**Application Generation** involves creating and coding the product to get onto the final stage.

Testing and Turnover which is all about vetting the fact the code works every time, there are no bugs. However, in RAD methodology the testing time is reduced as throughout the process the coding is to be tested during every iteration.

Using Prince2 helped organize workload and manage the whole task to meet the final deadline.

Rapid application development applied to the artefact due to the fact it is flexible so changes can be made if necessary. Furthermore, it is a methodology where the main aim was to create a fully working prototype as fast as possible which is the plan to due to the fact there is deadline.

## 1.4 Scope and limitations of the research/project

The scope of this project is to create a two-step authentication process that will allow social engineering techniques risks to be minimized. Therefore, reducing the success rate of cybercrime within the authentication sector.

Limitations of the research project are the fact this artefact is only for demonstration purposes, therefore, an actual login to a database was not created. In addition, this might affect the coding when implementing it within a banking application due to the fact there are more protocols involved when logging into an actual bank account. However, logging into a phone or using this authentication process for mobile devices is to be valid.

## 1.5 Thesis Structure (Summary of each chapter)

Throughout this thesis there are in total another **7 chapters**.

**Chapter 1: Introduction**

This chapter provides detail about the new 2-step authentication process which helps add additional security for mobile banking applications and other devices. This chapter also includes the background of how mobile banking is now, a problem statement, aims and objectives of this thesis, the research methods used, scopes and limitations of the research/project and finally the whole thesis structure.

**Chapter 2: Literature Review**

This chapter includes a varied amount of research regarding other types of authentication processes used in mobile banking and other devices. This chapter enables the reader to gain a better understanding of the current market. This chapter also allows the reader to

understand any types of problems within this sector of authentication to determine if this authentication process is a viable option or not.

**Chapter 3: Design & System Implementation of Authentication System**

This chapter includes the design of the artefact including images drawn up for the GUI and most importantly the coding from android studios. This is to help the reader understand exactly how the application looks, without a physical demonstration, furthermore, allows the reader to replicate the artefact if necessary. Finally, this chapter displays the system applications final versions to allow the reader to see how the artefact looks once gone through to the logged in page.

**Chapter 4: Testing of Authentication System**

This chapter demonstrates reliability whilst testing results to check if the application has any issues or bugs. Furthermore, all the validations on the design and application are gone over to see if it meets the criteria and standard set by the author.

**Chapter 5: Analysis and Evaluation of Test Results**

In this chapter analysis and evaluation is completed in regards to the testing results of the application, to see if the application is randomizing the numbers every time the application is reloaded. This chapter also covers any detail of what might go wrong and if the standards and criteria set by the author has really been met or not.

**Chapter 6: Comparison of Work to Previous Published Work**

This chapter compares the artefact against any other previously published work that is related to this authentication process.

**Chapter 7: Conclusion and Future Work**

This is the final chapter, where the author concludes and summarizes his work in regards to the findings of meeting his objectives of the project. Furthermore, suggested future improvements that can be taken to improve this authentication process. Finally, any future work the author is going to do.

# **Chapter 2:** Literature Review

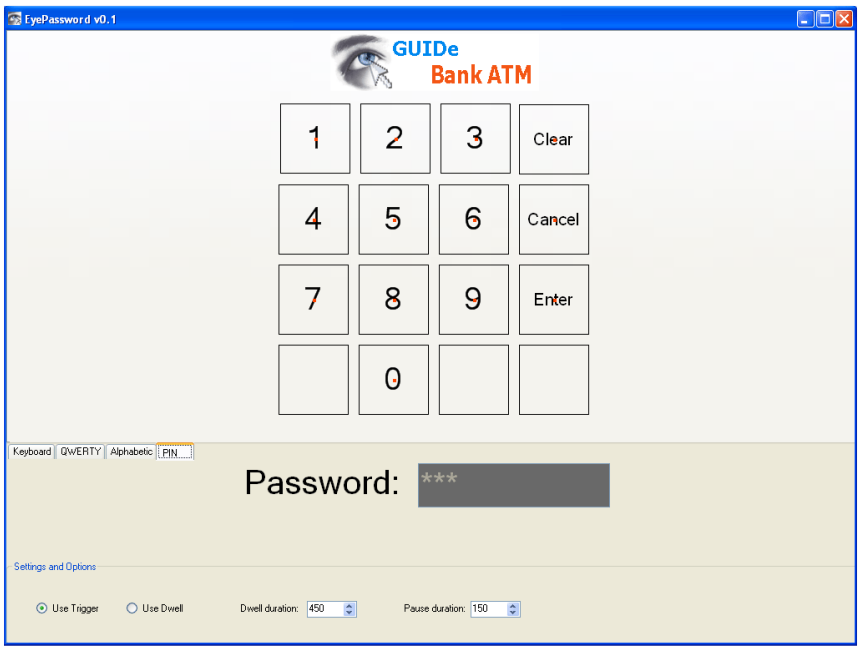
In this section the methodologies of research will be used to compare the weaknesses in comparison to the new randomized PIN keypad and memorable word authentication process. This section will include research journals and various banking websites which currently have mobile banking applications and the authentication process they currently use.

## 2.1 Related work in Gaze Based Authentication

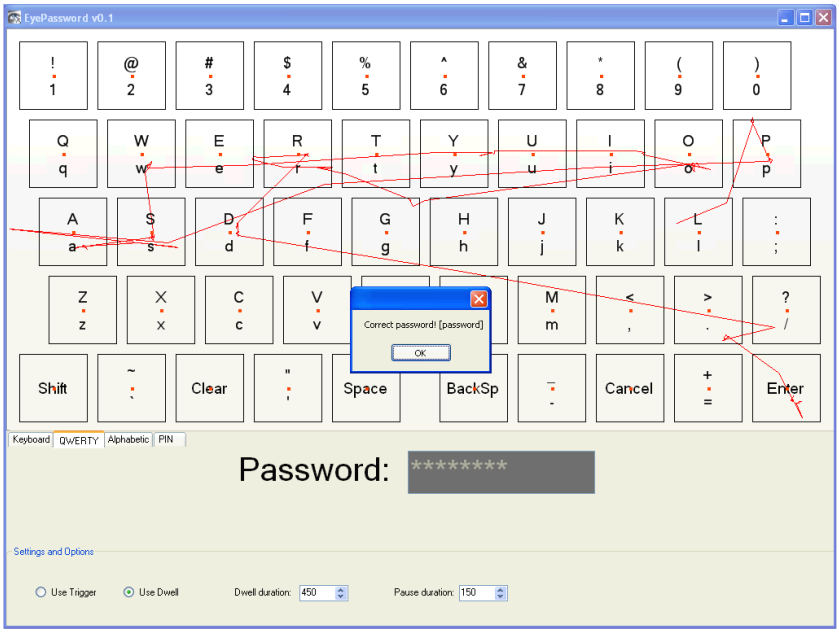
Social engineering techniques of shoulder surfing and tailgating are ways to obtain information without the victim knowing their information is being stolen. Similar studies have been taken place, which use the principles of social engineering of shoulder surfing. Kumar et al. (2007), researchers from Stanford University, have created a journal regarding 'Reducing Shoulder-surfing by Using Gaze-based Password Entry'. These researchers have created a countermeasure to shoulder surfing by using retinas. The concept of this countermeasure is to input passwords by the retina moving towards the characters which gets identified by their software which then inputs the characters into the password slot.

This then allows the user to login to the account or information the user is trying to get access to. Similarly, the randomized PIN keypad with the two-factor authentication process, which this project will deliver, also deters the social engineering technique of shoulder surfing. Using the randomized PIN is much more reliable than this retina concept due to the fact the ethnic issues of a user not having an eye or not having biometrics which fit the requirements of using this product would make the product obsolete. Therefore, would not generate a mass target audience like the randomized PIN keypad system would. Furthermore, there is no two-step authentication process. Another weakness in the 'Gaze-based Entry' system is if there was a hacker which wanted to withdraw the information of the password, the hacker could potentially hack into the device and switch the camera on to then identify where the retina is looking to try and imitate the password.

In comparison to the randomized PIN keypad, there would be no way of figuring out the PIN unless the shoulder surfer sees every keystroke and memorizes the numbers inputted. The only other way of trying to obtain the password is by hacking into the device and injecting malicious software that could log the keystrokes allowing the hacker to gain knowledge of the password used.




[*Figure 2.1* – Gaze PIN]




[*Figure 2.2* – Gaze Keyboard]

## 2.2 Related work in area of Graphical Based Authentication

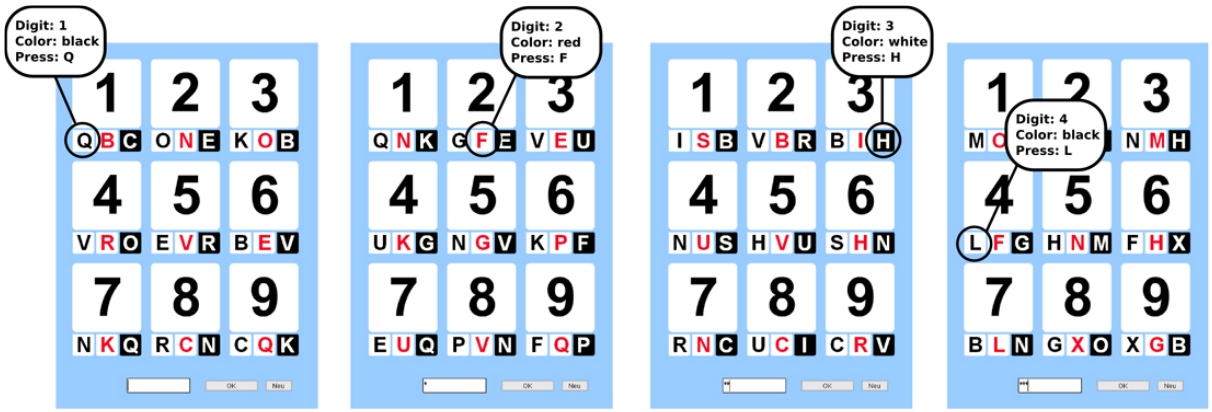
Li et al., (2005) researchers from the National University of Singapore have conducted research regarding using graphical based passwords to deter social engineering attacks of shoulder surfing. Li et al., (2005) publication titled 'An association-based graphical password resistant to shoulder-surfing attack' is the research paper submitted. This research journal states using graphical interfaces allows the user to remember the passwords by images much easier. Recent research on "Learning Recall Related to Type of Presentation" (Changingminds.org, 2016) has shown 80% still remember the image after 3 hours and 65% after 3 days. However, when regarding a password 72% of normal humans remembers the password after 3 hours however after 3 days a mere 10% remember the password.

This graphical interface concept allows the user to click on a specific picture out of 4 images. Once the user has picked the first step of authentication by clicking on the right image it then takes them onto the second step of authentication, which then allows the user to click on a specific colour of the picture the user previously chose. The next step asks the user to pick a group image of the specified image the user picked which then finally asks the user to pick a specific part of an image. For example, a table or painting in a picture of a room. This is a four-step authentication process, which is very secure, however as stated above, "Learning Recall Related to Type of Presentation" (Changingminds.org, 2016) states 80% of normal humans with generic IQ remember the photo sequence after 3 hours and then reduces to 65% after 3 days.

This overall means the potential for an observer using social engineering of shoulder surfing would have a higher chance of a remembering the password. In comparison to the randomized PIN keypad and memorable word authentication process; this deters any users from knowing the normal keystrokes a consumer would use. For example, due to the nature of the keypad being randomized, the digits would not be in the same place meaning where 1 would normally be, by random there might be a 5 which deters the shoulder surfer and creates confusion for the shoulder surfer which overall makes the shoulder surfer think you have pressed number 1 when in realism it was number 5.

## 2.3 Related work in area of Colour Light Based Authentication

Another research journal discusses the use of using letters and colour to allow the user to enter in their PIN for ATM machines and any other type of mobile devices for authentication. De Luca, Hertzschuch, and Hussmann. (2010), researchers have created a new method of inputting PIN numbers; this research journal is from the University of Munich; Media Informatics Group. The main concept of using this method would display three random letters stated below the standard PIN numbers on an ATM or Mobile Banking Application format. The user then would be told to click on the "red" letter. All the letters are coloured differently, which then allows the user to click on the right colour, which would then input the number of which the user has tried to input.



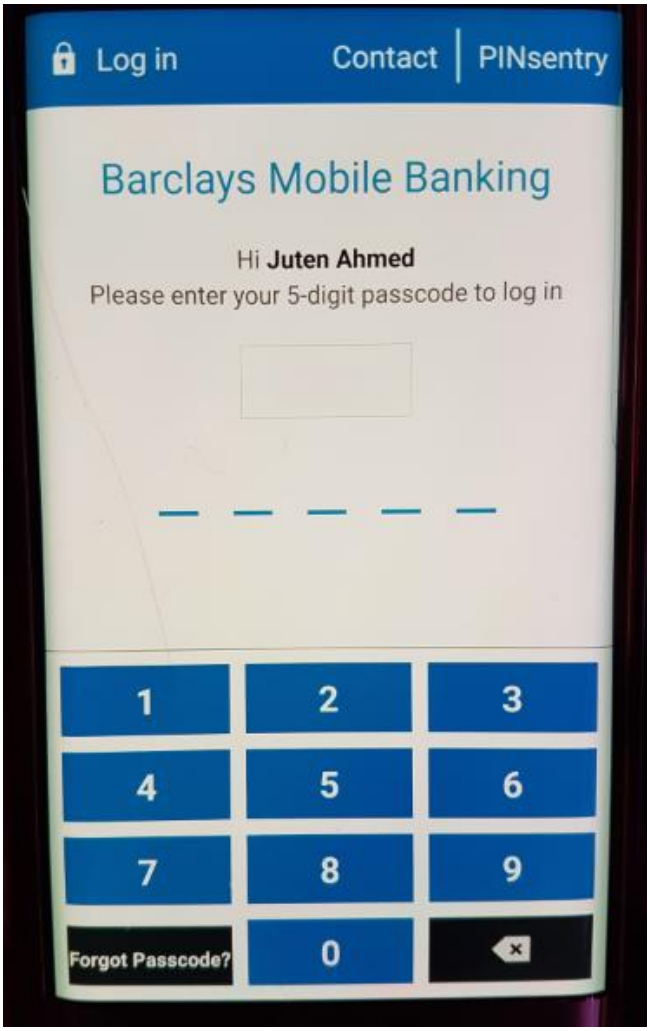

[*Figure 2.3* – ColorPIN Concept]

A flaw in this would be if a user / consumer is colour-blind. This would not be usable by such consumers. In comparison to the randomized PIN keypad and memorable word authentication process, the authentication process would still be use to consumers whom are colour-blind. The only restriction would be to anyone who does not have any fingers or any method of clicking on the chosen PIN created by the user. Furthermore, ColorPIN would take much longer to process for customer to input the correct letters which increases the time of figuring out then inputting the correct letter; this then gives the observer looking for the PIN using social engineering techniques extra time to have a look at what the user is inputting.

In comparison to the randomized PIN keypad authentication process; the randomized PIN keypad process is still faster, and much more reliable due to the fact the numbers are all already given to the consumer in regards to the consumer having to decipher what the letter and colour is before allowing the PIN to be inputted into the device.

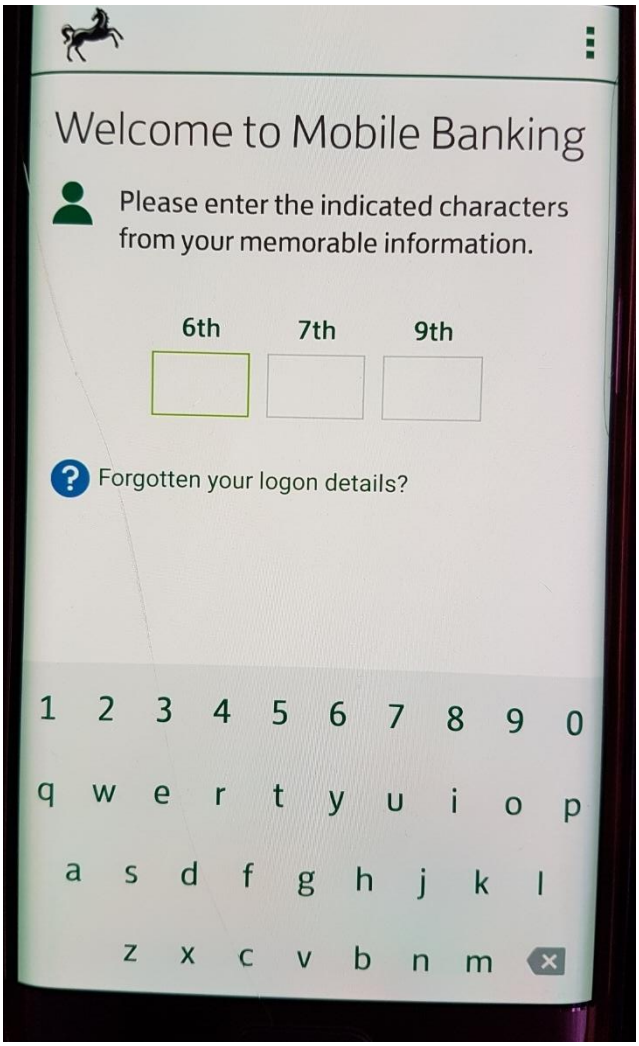
## 2.4 Related work in area of the Banking Industry

Banking is the main sector this randomized PIN keypad is necessary. Currently in the market Barclays Bank is the inspiration for this idea due to the fact Barclays Bank (2017) are currently only use a 5-digit PIN to enter the user's bank account. In comparison to the randomized PIN keypad the Barclays Bank Mobile Banking application does not deter social engineering due to the 4 by 3 matrix it currently has. This allows any observer using tailgating or shoulder surfing techniques to easily see where the user's fingers tap to easily obtain the PIN. The randomized PIN keypad will minimize the risk of the PIN getting into the observer's hands. Furthermore, by having a two-step authentication using both the memorable word and PIN allows a secure connection.




[*Figure 2.4* – Barclays Bank]

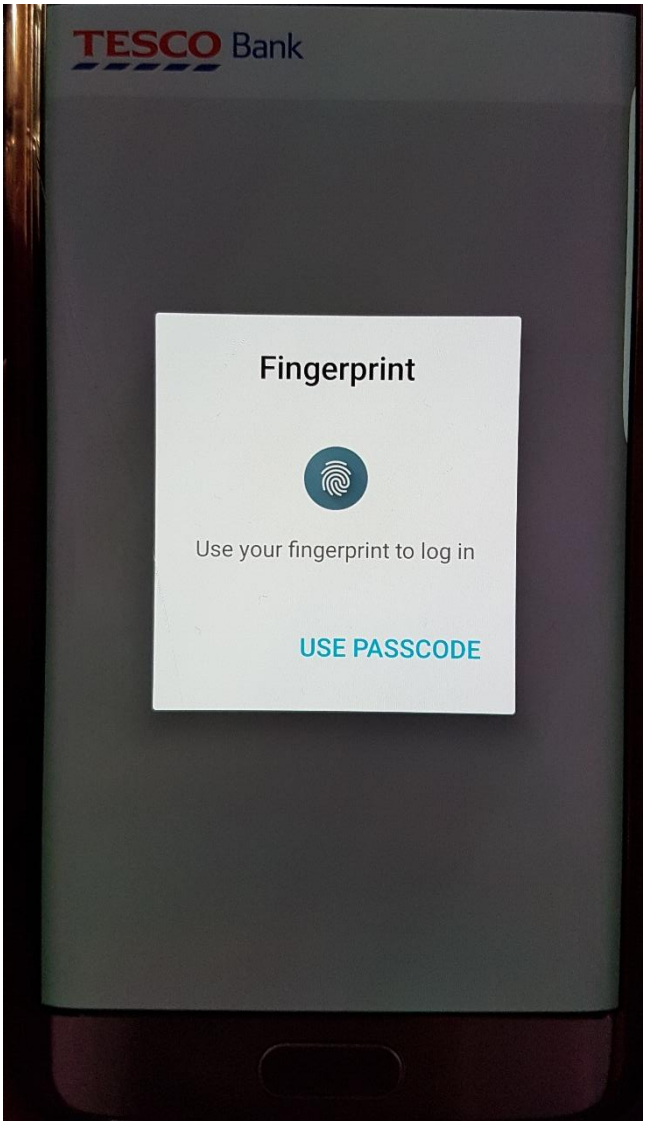
Lloyds Bank (2017), are currently using only a memorable word authentication process which asks the user to input 3 characters of the memorable word they have chosen. This allows users to input 3 different characters which are very hard to figure out however in comparison to the randomized PIN keypad authentication system, due to the fact it is a two-step authentication process, this not only makes it secure; however, increases the security due to the fact it deters social engineering techniques.



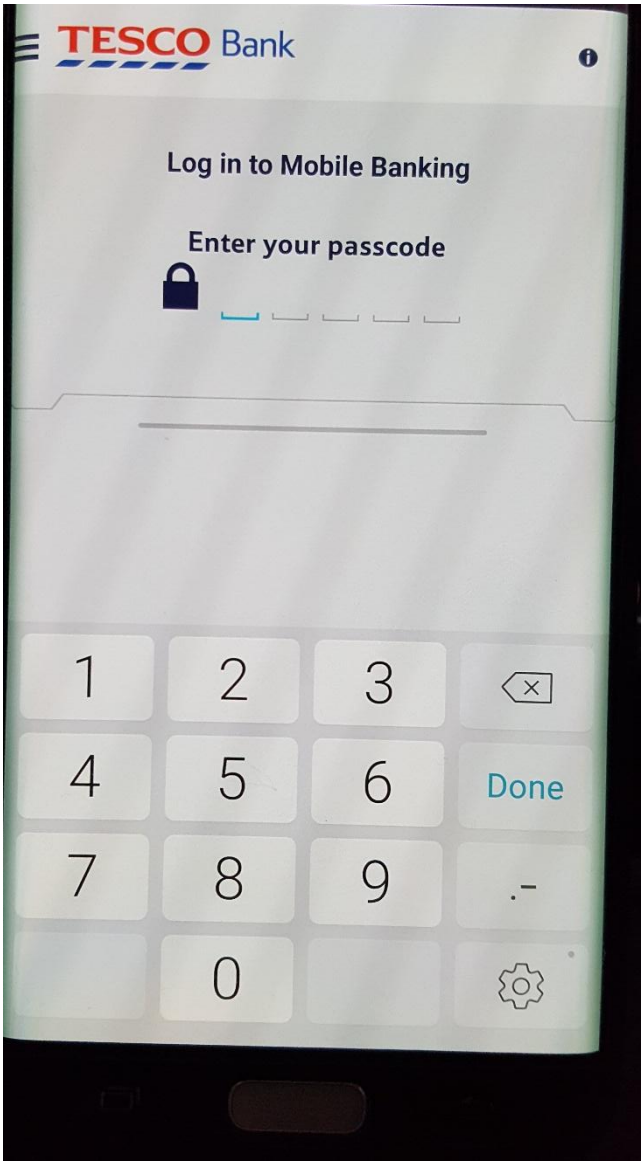

[*Figure 2.5* – Lloyds Bank]

These are the two types of authentication processes majority of banks use. Either one of each. However, in regards to the artefact of two-step authentication including the randomized PIN. No other banks use this authentication design and process.

Finally, Tesco PLC (2017) have started to use biometrics for the authentication process with a 5-digit PIN like Barclays bank as a back-up in case the fingerprint is not recognised. This makes the whole process of authentication much more difficult for someone trying to obtain the passcode or PIN as there is not one displaying. This is also a very convenient way for a user to login to the users account. Due to the fact Tesco uses both biometrics and a PIN as a back-up, this essentially makes this authentication process a 2-part authentication process, however both are not required to login.




[*Figure 2.6* – Tesco Biometrics]



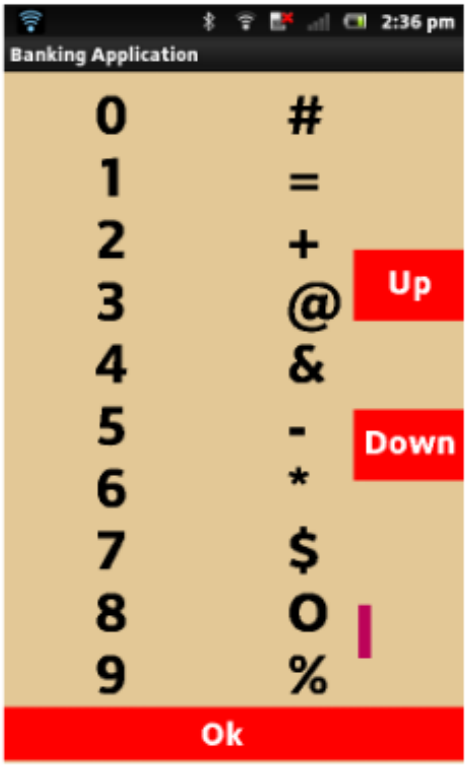

[*Figure 2.7* – Tesco PIN]

## 2.5 SteganoPIN

Kiruthika et al. (2016) had created SteganoPIN. SteganoPIN had been created to reduce the amount of attacks which are occurred by social engineering techniques like shoulder surfing. There are a few concepts behind SteganoPIN as it not just one authentication process but many different prototypes under one name.

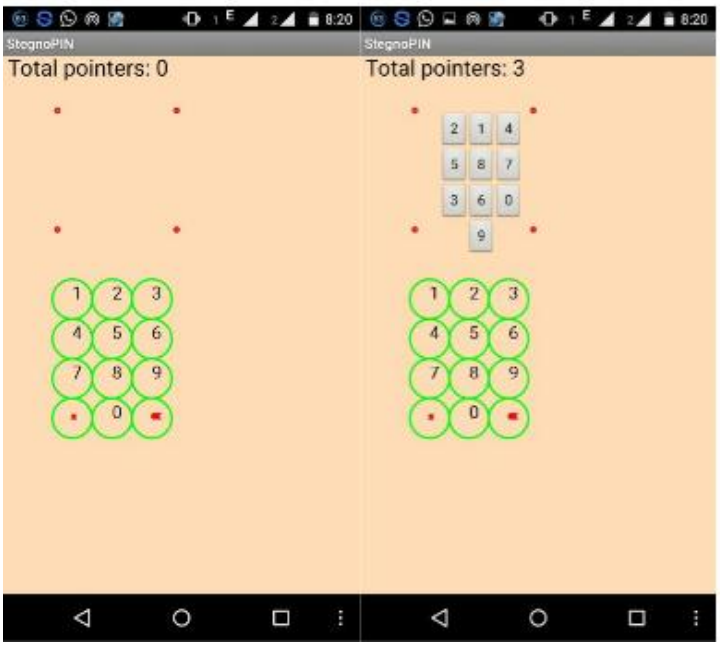
Firstly, entering a password then authenticating that and finally entering in the user's PIN. However, the design of the PIN keypad is the interesting part.

As seen on *figure 2.8* SteganoPIN first allows the user to input the users PIN with number ranging from 0-9 and then the user can also use several punctuation characters.




[*Figure 2.8* – SteganoPIN 1st Authentication]

Next the user must input the users PIN with the user's hand, however not just clicking on the number itself but the positioning of the page on the application. Therefore, each position has a different value as seen on *figure 2.9.*

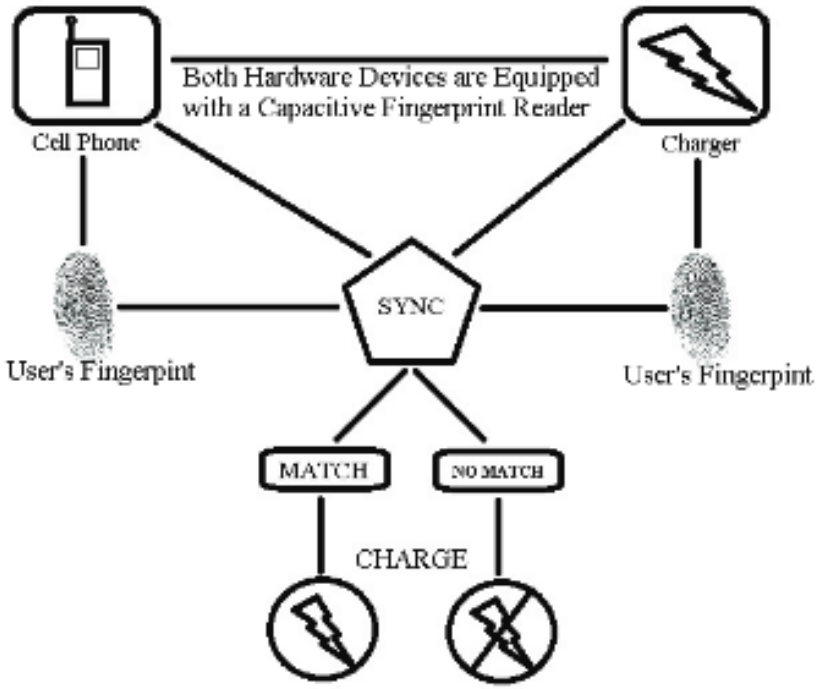



As displayed on *figure 2.9* the numbers are placed on the right-hand side with randomized digits. However, when the user clicks on a specified section of the PIN a different value comes out.

[*Figure 2.9* – SteganoPIN 2nd Authentication]

## 2.6 Biometrics

Ohana, Phillips, Chen. (2013) conducted research into using fingerprints as an authentication system for reducing the risk of items being stolen and then authenticated by a third party. For example, if a phone was stolen the thief would be able to withdraw all the information from the mobile phone using a USB.




[*Figure 2.10* – Biometrics Authentication IEEE, 2017 cited in Ohana, Phillips, Chen, 2013, p.3.]

*Figure 2.10* above demonstrates how this authentication would work. When the user is purchasing the users mobile phone or any sort of device or item, the manufacturer would need to use the purchasers fingerprint to register the user onto the company's database. The company would do the same with the charger the user has purchased with the phone. So, in future if the phone was stolen and the thief wanted to withdraw all the sensitive and private data on the phone. The first instinct of the thief would be to plug the phone into a computer or laptop. This would then ask for the fingerprint to be re-entered, thus preventing the thief from entering the phone.

## 2.7 Summary

In summary, this chapter has covered many different variations of authentication processes which are like the artefact displayed within this thesis. All these authentications processes have the idea of 1 step authentication, however do not follow the idea of having a two-step authentication process.

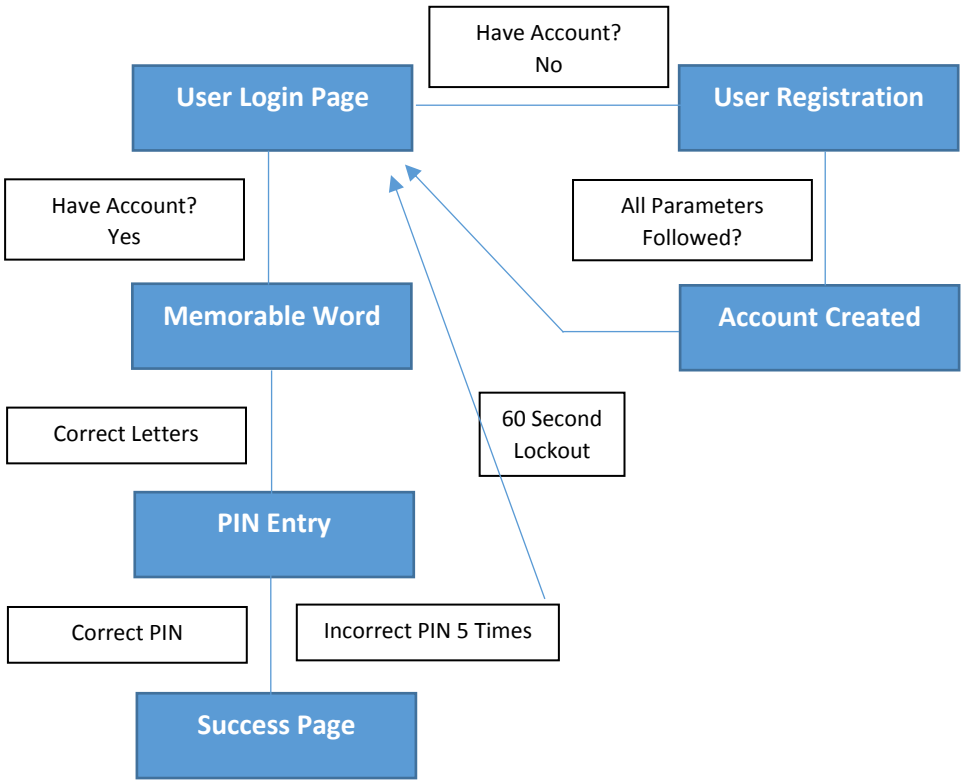
The next chapter is regarding the design and system implementation so will be revolved around the design of the artefact and how the artefacts final version should come out. Furthermore, the system implementation in regards to having a working artefact.

# **Chapter 3:** Design & System Implementation of Authentication System

This chapter entails all the design work needed to create this artefact, furthermore the system implementation and how the design will work in the real world.

## 3.1 Design

These steps allow the system to be viable and in working order. This is the design of how the artefact should work.




[*Figure 3.1* – Design Plan]

*Figure 3.1* demonstrates if the user does not have an account the user can then the user can go onto the registration page to create an account. If all the parameters of the memorable word and PIN are correct, the user would have successfully created an account.

Figure 3.1 also demonstrates, if the user already has an account the user can go straight onto the memorable word page. If the user inputs the correct letters the user will go onto the PIN entry page. However, if the user does not input the correct letters the user will be stuck on the memorable word page. Once on the PIN entry page the user must enter a 5-digit PIN. IF the PIN is correct the user will

reach the final page of the success page. However, if the user inputs an invalid PIN 5 times in a row the user will get locked out of the account for 60 seconds.

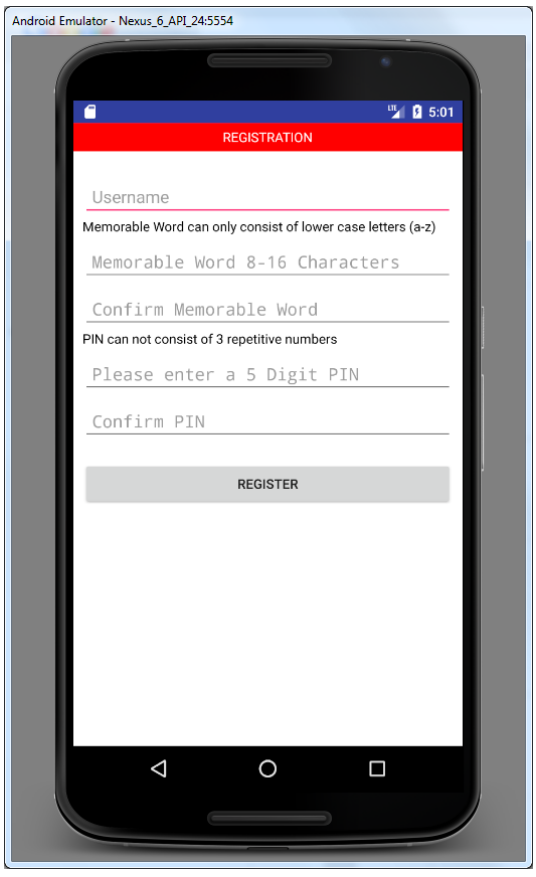
## 3.2 System Implementation

These are the final designs of how the working porotype looks.

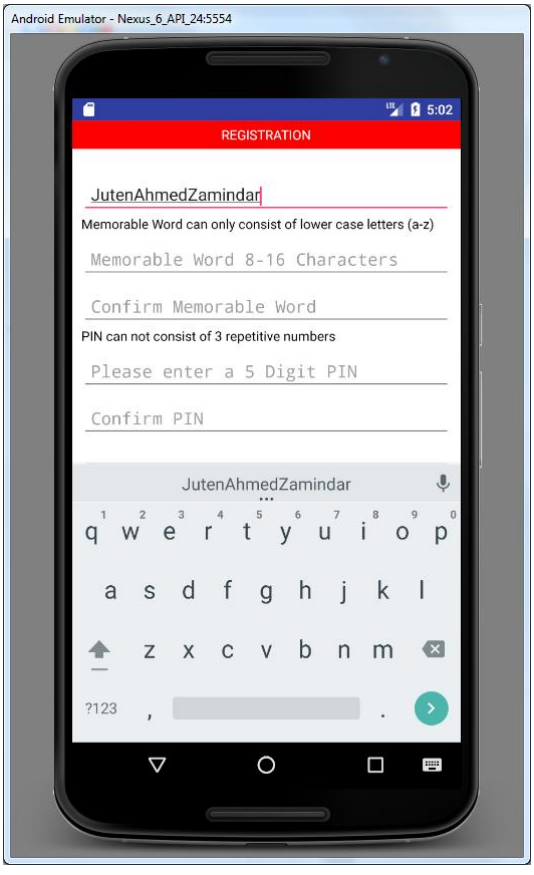
There are 5 pages in total:

1. Registration Page
2. Login Page
3. Memorable Word Authentication Page
4. Randomised PIN Authentication Page
5. Success Page

### 3.2.1 Registration Page




[*Figure 3.2* – Registration Page]



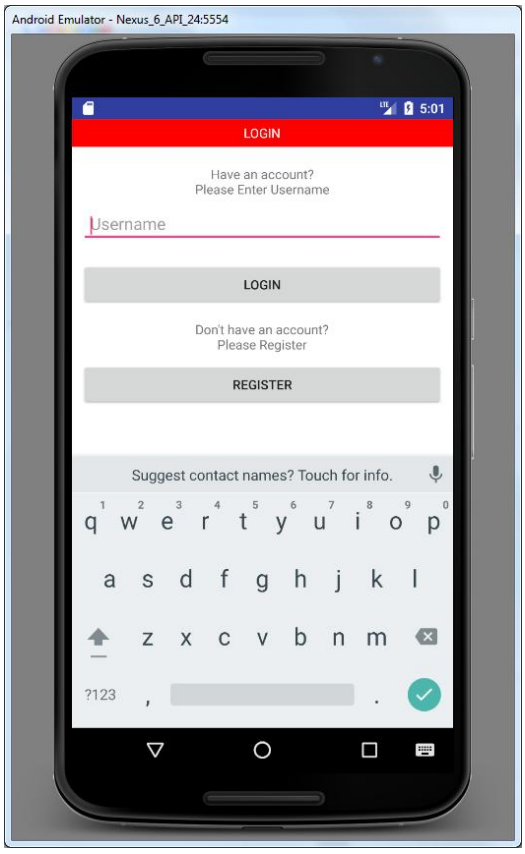

[*Figure 3.3* – Registration Page with Keyboard]

This is the layout of the registration page. As you can see in *figure 3.2* and *figure 3.3*, there is the title of the page “REGISTRATION” so the user knows which page the user is on. The user must then enter

his or her desired “Username” using the keyboard which pops up once the username section is clicked [*Figure 3.3*]. Then enter a “Memorable Word” which must be between 8-16 characters and only using lower case letters between “a-z”. Once the user has input their memorable word, the user must confirm the memorable word they had chosen in case the user misspelt the word the user was trying to spell. Finally, the user must input a 5-digit PIN which can be between numbers 0-9 and must not have 3 consecutive numbers. Again, once confirming the PIN the user may register their account.

### 3.2.2 Login Page


[*Figure 3.4* – Login Page]

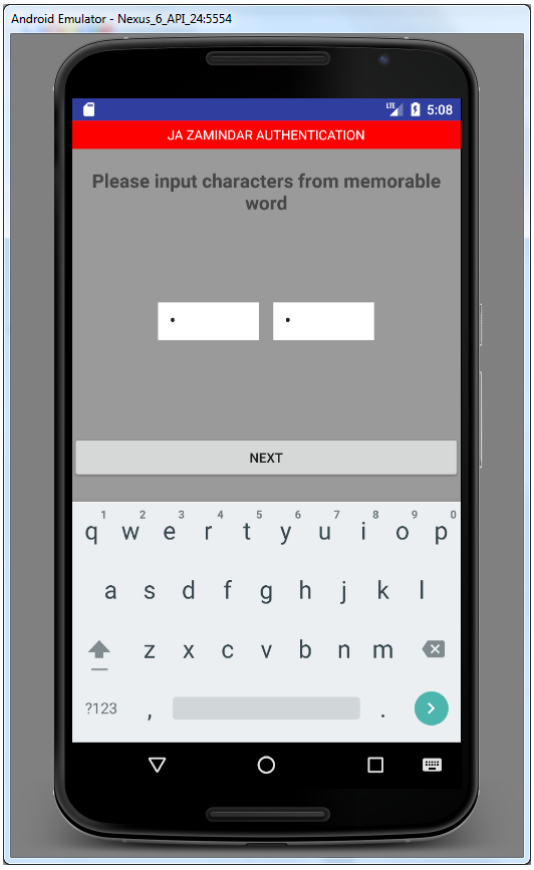



[*Figure 3.5* – Login Page with Keyboard]

This is the layout of the login page. On the top of the login page as displayed on *figure 3.4* and *figure 3.5*, is again the title “LOGIN” for the benefit of the user to understand where they are within the application. The user must enter the username he/she had created within the registration page using the keyboard which pops up once the username section is clicked [*Figure 3.5*]. This will then take them onto the memorable word page.

### 3.2.3 Memorable Word Authentication Page

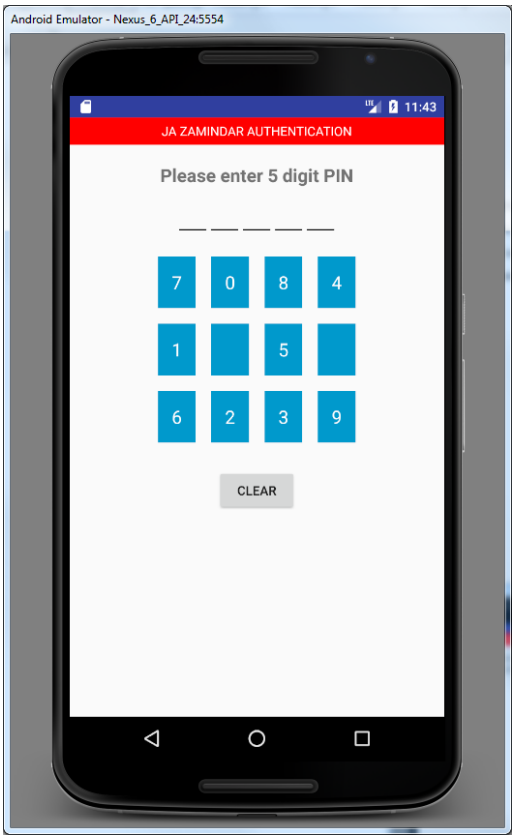

[*Figure 3.6* – Memorable Word




[*Figure 3.7* – Memorable Word Page with Keyboard &

This is the memorable word page. As displayed on *figure 3.6* and *figure 3.7*, due to the fact now the application has entered sensitive "secret" data which needs to be protected, there is now no title but just the application name "JA ZAMINDAR AUTHENTICATION". As displayed on figure 1 the application randomizes which letter out of the 8-16 character memorable word the user must input. *Figure 3.7* displays the characters already inputted by the user by using the keyboard. The characters inputted is displayed as an asterisk to hide the characters.

### 3.2.4 Randomised PIN Authentication Page




*[Figure 3.8* – PIN Entry

This page is the PIN entry page where all the digits are randomized as displayed on *figure 3.8*. The user must input the 5 digit PIN the user had created on the registration page which will then finally take the user to the success page.

This is the coding created to make the randomized PIN.

```
//return true of its a blank spot else false
private boolean showBlankKey(String str){
    //10 and 11 are blank spots within app
    return (str.compareTo("10")==0 || str.compareTo("11")==0);
}
```

As seen above, the code here is for the blank sections on the 3x4 matrix. It will display 2 blank keys which would not do anything when the user presses the key. Furthermore, as displayed there are two more slots added on the code. 10 and 11. This is because 0-9 are keys which will display the numbers 0-9 whereas 10 and 11 will only display blank spots.

```
    //loop through 0 to 11 .... digits 10 and 11 are blank spots
    for (int a=0;a<=11;a++){
        //initialize random class to get always random digit
        Random rnd=new Random();
        //get random digit to display everytime
```

```
        String digitToDisplay=String.valueOf(rnd.nextInt(12));
        //check if its already displayed
        while(isDigitDisplayed(digitToDisplay)){
            digitToDisplay=String.valueOf(rnd.nextInt(12));
        }
        //if digit is unique then add to collection
        digitsDisplayed.add(digitToDisplay);
```

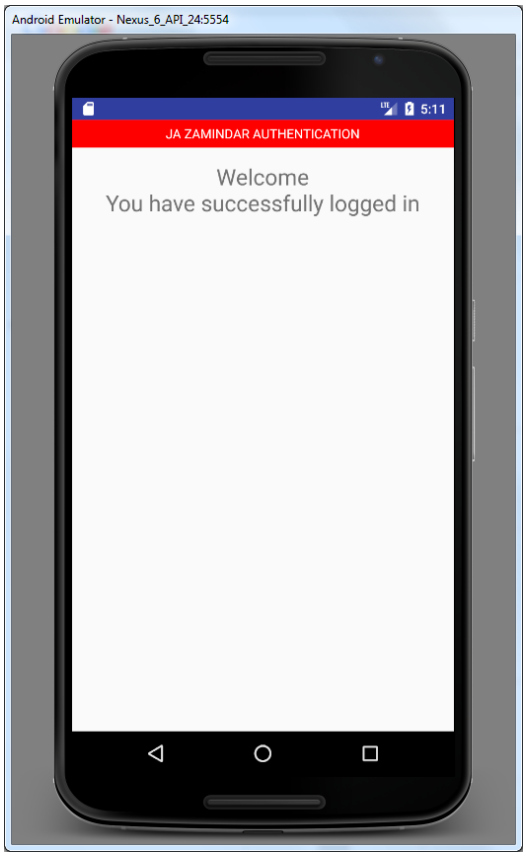
As seen above, this is the algorithm used to create the loop from numbers 0-9 and then finally the 2 blank spots (0-11). “Random rnd=**new** Random();” randomizes the numbers when displayed on the 3x4 matrix until every digit has been placed on the matrix. All digits must be unique to be added onto the 3x4 keypad and finally the 2 blank spots once all the numbers from 0-9 have been added onto the keypad.

```
        //if invalid it will increase the invalid counter
        AppCache.invalidPinCount++;
        //check how many tries are remaining
        int diff=AppCache.INVALID_PIN_LIMIT-AppCache.invalidPinCount;
        //if there are still tries show message
        if (diff>0) {
            UIHelper.msbox("Error", "Invalid: " + String.valueOf(diff)+" tries
remaining", PinActivity.this);
            clearClick(null);
            this.initializePinKeypad();
        }else { //if there are no more tries
            //lock the account
            LocalStorage ls = new LocalStorage(getApplicationContext());
            ls.setLocked();
            //get remaining seconds for unlock
            long remain=ls.getUnlockRemain();
            //show how many seconds are remaining to unlock
            UIHelper.makeLongToast("Your account is now locked. You can login after
"+String.valueOf(remain)+" seconds. Thank You for using JA Zamindar
Authentication", PinActivity.this);
            //move back to login screen as account is locked now
            Intent i = new Intent(PinActivity.this,LoginActivity.class);
            startActivity(i);
            finish();
        }
        return;
```

As seen above, if there are any invalid PIN’s inputted into the application there will be an error message displaying how many tries are remaining. The user has 5 tries to input the correct PIN. If the PIN is incorrect 5 times, the user will get locked out with the message displaying how many seconds and automatically directed onto the login page. If the user, then tries to login again with the username the user will receive a UIHelper message informing the user of how many seconds are left before the lockout has ended.

### 3.2.5 Success Page

This is the final page for the application. The success page once all authentication processes have been verified. The user then enters the page which informs the user they have successfully loggin in as displayed on *figure 3.9*.



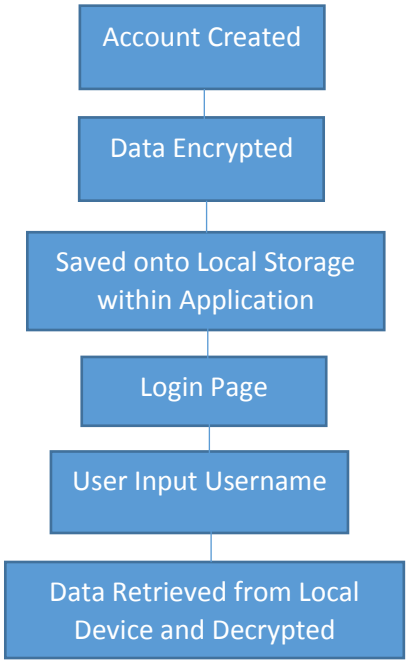

[*Figure 3.9* – Success

## 3.3 AES Encryption and Local Storage

The encryption method used to encrypt this is Advanced Encryption Standard (AES) using SHA-256 and hexadecimals to create the AES encryption. Due to the fact AES requires a secure and random key SHA-256. IV (initialization vector) is used for the whole process whilst using AES to encrypt. IV is initialized with hexadecimal digits.

All the data created throughout the registration process is then saved onto the devices local storage due to the fact this application was made for demonstration purposes. There is no file within the application when trying to root. All data is saved within the application and is encrypted using AES once created. When trying to log in using the same credentials the information is then decrypted using the cache.

So how does it work?




[*Figure 3.10* – AES Encryption]

*Figure 3.10* demonstrates how the encryption works. Once the account has been created the data from the registration page the user inputted gets encrypted and saved onto the local device within the application. This means there is no .txt file with the information if a hacker did try and get into the devices local storage. Once the user tries to login using the username the user has created, the data is retrieved from the local device storage and decrypted. This then is now visible within the application memory to put all the parameters in place regarding authenticating to the next step of the application.

This is the important part of the code which displays encrypting and decrypting the information.

The code below is for creating the account. When the account is being created, the information needs to be encrypted. So, when the user wants to login to the account the user can automatically decrypt the user's information when authenticating.

```
        try{
              //store values in local storage
              LocalStorage ls=new LocalStorage(getApplicationContext());
              ls.setUsername(usr.username); //store username in local storage
              //store encrypted secret word in local storage

ls.setWord(AESEncryption.encrypt(AppCache.passwordToEncrypt,usr.secretWord));
              //store encrypted pin in local storage

ls.setPassword(AESEncryption.encrypt(AppCache.passwordToEncrypt,usr.pin));
```

```
            //call unlock to reset lock
            ls.unlock();

            //account created message
            UIHelper.msbox("","You have successfully created an account",new
DialogInterface.OnClickListener(){
```

This code below is for the logging in step where the user needs to have the valid username information to withdraw the data from the local storage. The password is encrypted so there is another page connecting the encryption where there is a master password. As demonstrated below, every section of the authentication has to be decrypted so for example "swu.**secretWord** = AESEncryption.*decrypt*(AppCache.*passwordToEncrypt*, ls.getWord());" decrypts the password by taking data from the AppCache by using the encryption password key.

```
try {
    //if entered username matches the username in the storage than its correct
login
    if (susername.compareTo(username) == 0) {
        //create user for cache in app to be accessed by different screens
        SecretWordUser swu = new SecretWordUser();
        //set username
        swu.username = ls.getUsername();
        //set (memorable) word
        swu.secretWord = AESEncryption.decrypt(AppCache.passwordToEncrypt,
ls.getWord());
        //set PIN
        swu.pin =
AESEncryption.decrypt(AppCache.passwordToEncrypt,ls.getPassword());
        AppCache.currentUser = swu;
        //reset invalid pin counter as its a new login
        AppCache.invalidPinCount = 0;
        //open screen to enter memorable word
        Intent i = new Intent(this, SecretWordActivity.class);
        startActivity(i);
        finish();
```

The code above demonstrates that once the user had a valid username the user can then go into each step by decrypting the pages using the correct information at every page.

## 3.4 Integrity

Can the data be intercepted while inputting the information?

The only way the information can be intercepted is if the device had already been compromised. If the device already has a malicious software installed which allows the hacker to log all the keystrokes on every application, then answer to this question would be yes. There is the possibility of the information being intercepted. However, if the user is aware of keeping the user's devices up to date and regularly checking to make sure there are no malicious software on the user's device then the only way of decrypting the information is by going into the application and inputting the correct username to withdraw all the information from that account.

Furthermore, all the data is hashed using SHA-256 algorithms to maintain the integrity of the information so if the metadata has been tampered with, the passwords and PIN's would not work anymore.

## 3.5 Coding

Please view [**Appendix B** – B.1 Code] for all code regarding re-creating this application.

## 3.6 Summary

In summary, the whole design consists of 5 pages. Registration Page, Login Page, Memorable Word Authentication Page, Randomised PIN Authentication Page, Success Page. All these pages allow the user to use the application for a 2-step authentication process.

The whole system works properly and the design tries to tackle any bugs or issues that can come into place. In the next chapter the testing of the authentication system will be represented in detail to clarify if all the validations created were met. Furthermore, discussing which testing strategy had been used.

# Chapter 4: Testing of Authentication System

This chapter discusses what testing strategy was used and what test validations have taken place to make sure the quality and standards have been met in terms of the project proposal. Furthermore, to see if the artefact works as expected.

## 4.1 Testing Strategy

The testing strategy used when developing and testing the artefact was a proactive testing strategy (Tutorials Point, 2017). This approach is the best option for this type of artefact due to the fact there might be many errors that occur and cannot rely on a finished product before testing. Whilst RAD and PRINCE2 methodology were being used in the development and overall management of the project, there were two main testing approaches used. Dynamic and heuristic approach as there was much needed to be learnt while developing the artefact and finally a methodical approach as this application has been created by trial and error due to testing code to see the different outcomes to achieve the final product.

## 4.2 Testing of Registration Page

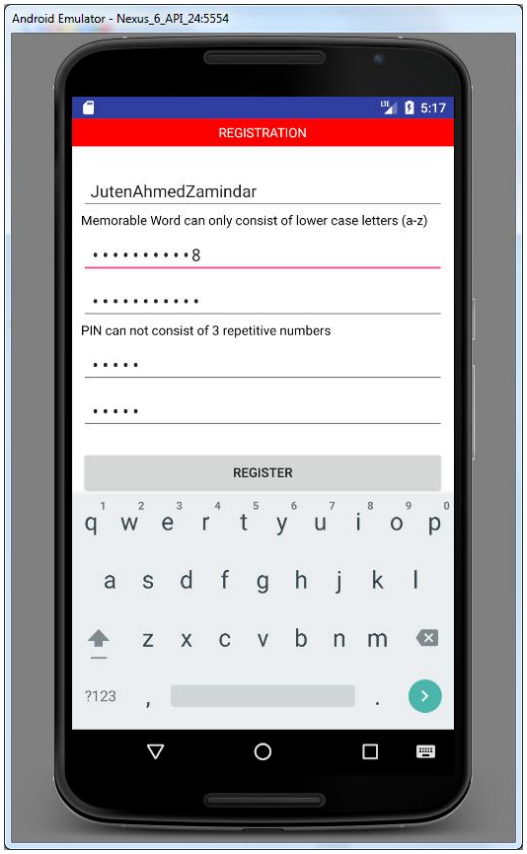

[*Figure 4.1* –Memorable Word - Asterisks

The Registration page has many validations that must be met.

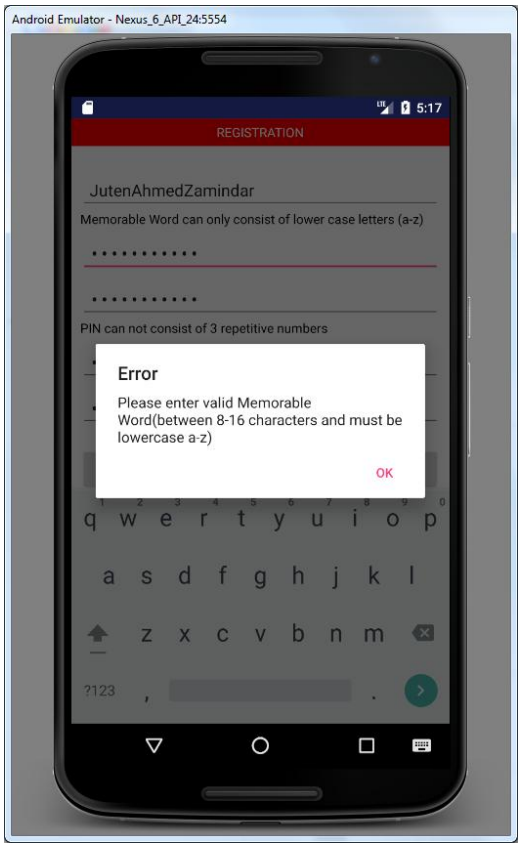
1. Users must be able to input a username.
2. The username can include any characters including numbers and punctuation.
3. User must input a minimum of 8 characters and a maximum of 16 characters to create the memorable word.
4. The memorable word can only be made with letters of made up of a-z.
5. The memorable word can not include any capital letters. All letters must be lower case.
6. The memorable word cannot have any numbers or punctuation included in the passphrase.
7. The PIN must be made up of 5 digits.
8. The PIN must not include 3 consecutive numbers.
9. The PIN must be made up of numbers between 0-9

10. All characters must show an asterisk (*) to protect confidentiality of the passphrase and PIN.
11. If a PIN or Memorable Word is invalid a message should come up informing the user of the information being invalid.
12. If the information does not match a user cannot be created.

To make sure all these validations are implemented to meet the requirements and standards set there was many tests that went through. All the screenshots displayed will explain and demonstrate the testing procedure. As demonstrated on *figure 4.1* the username has been inputted whilst the testing of the memorable word is taking place. In addition, as demonstrated on *figure 4.1* the characters inputted as the memorable word are all displayed as asterisks to maintain confidentiality.



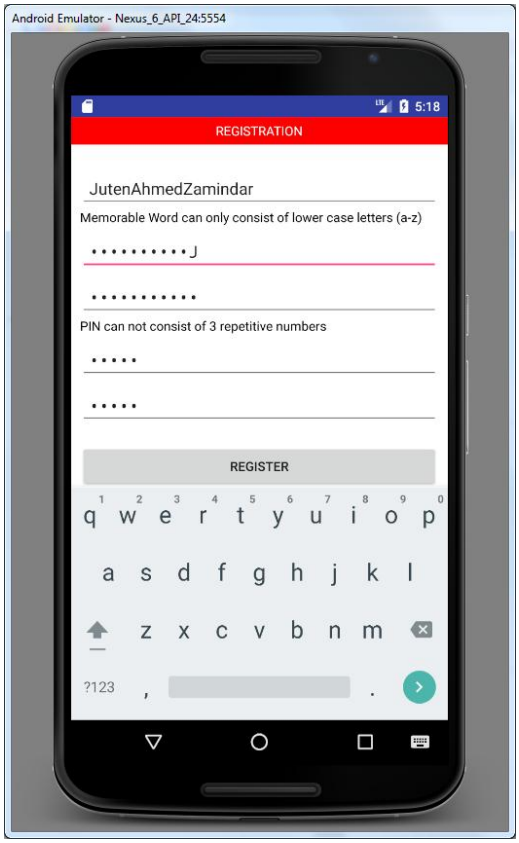

[*Figure 4.2* – Memorable Word - Number




[*Figure 4.3* – Memorable Word - ERROR

As demonstrated on *figure 4.2* the memorable word went through tests to check if a number would be valid when creating the memorable word.

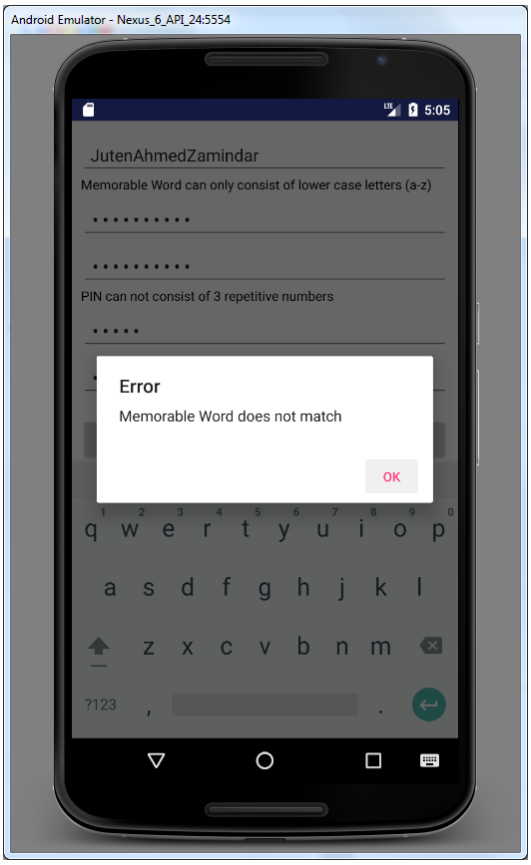
However, *Figure 4.3* demonstrates a system message coming up displaying as an “Error” and informing the user that the memorable word is not valid and must be between 8-16 characters, furthermore, must be lowercase letters between a-z.



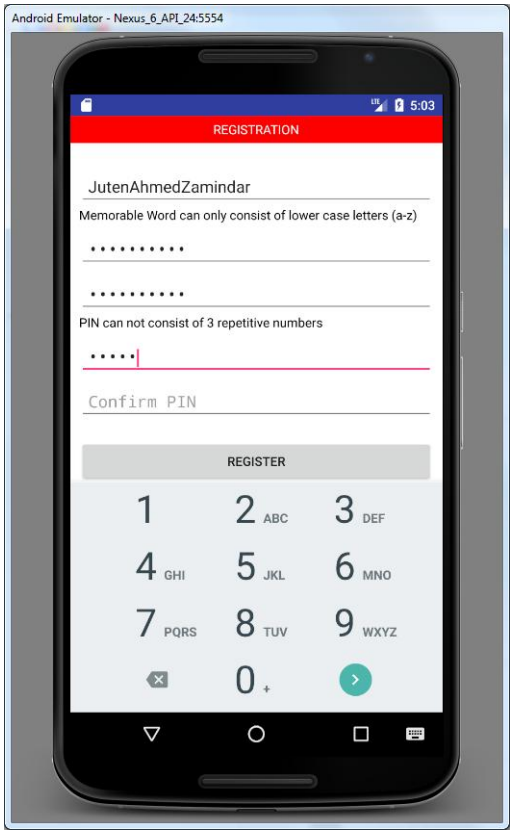

[*Figure 4.4* – Memorable Word – Upper

*Figure 4.4* demonstrates inputting an uppercase letter to test if the validation works. Where *figure 4.3* once again demonstrates the validation in place again providing the message of and "Error" and then informing the user that the memorable word is not valid and must be between 8-16 characters, furthermore, must be lowercase letters between a-z.

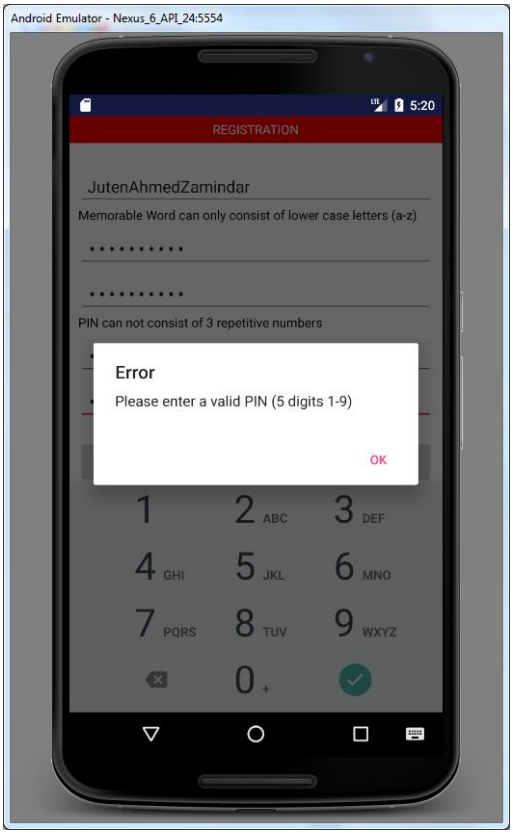
*Figure 4.5* demonstrates the process of a memorable word not matching. A "Error" box comes up informing the user the memorable did not match. Therefore, the user can re-type their memorable word.




[*Figure 4.5* – Memorable Word – DOES NOT MATCH]

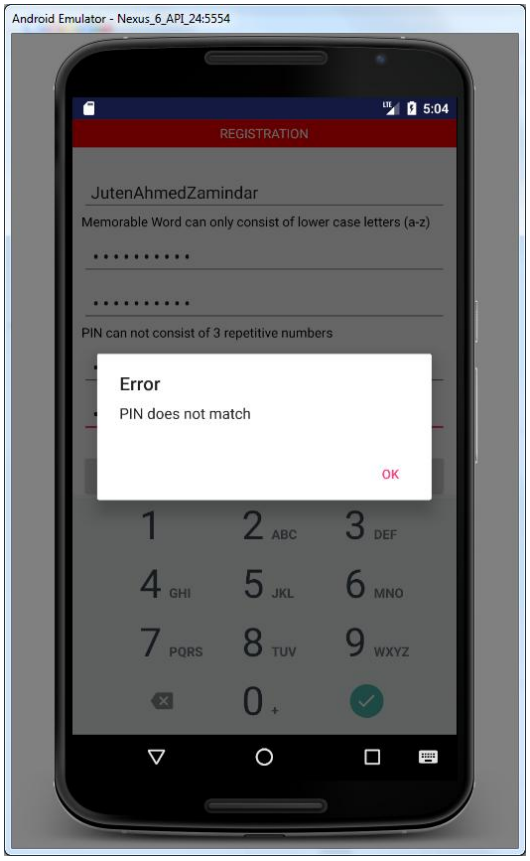



[*Figure 4.6* – PIN entry – Asterisks]



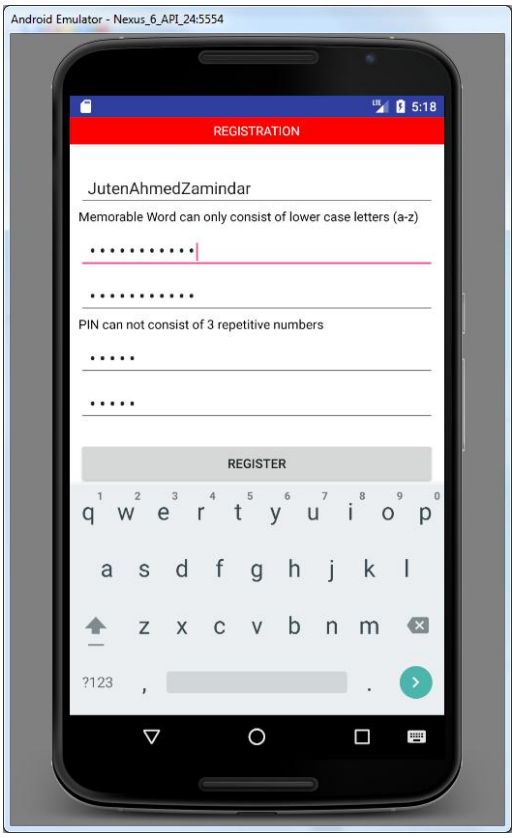

[*Figure 4.7* – PIN Entry – 3 Repetitive Numbers]

As demonstrated in *figure 4.6* the PIN entry section also makes all the numbers inputted into an asterisk to keep the data confidential. Finally, on *figure 4.7* there is an "Error" which demonstrates the validation of when a user inputs three repetitive numbers the user must then re-enter a valid PIN. As seen on *figure 4.6* the user can read the fact the "PIN can not consist of 3 repetitive numbers" to clarify to the customer the requirements of creating the PIN. This message is located before inputting the PIN.

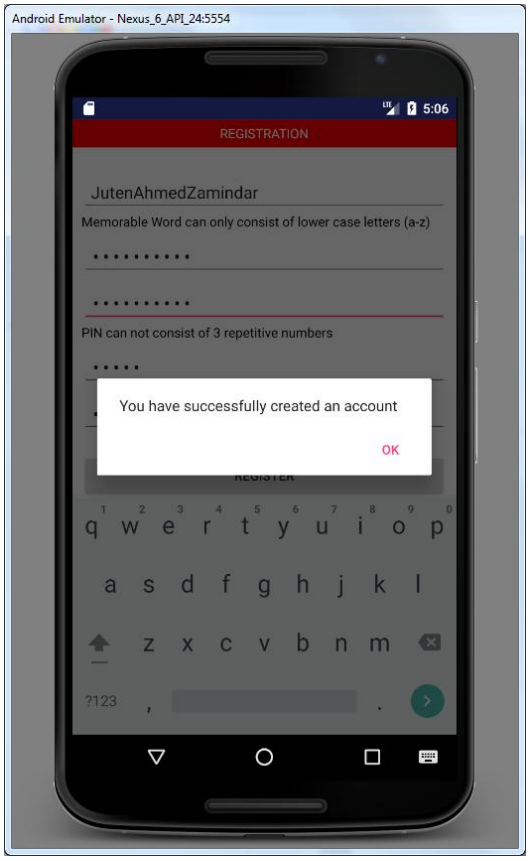



[*Figure 4.8* – PIN Entry – No Match]

*Figure 4.8* demonstrates when a user does not enter the correct PIN in both sections "Please enter a 5-digit PIN" and in the "Confirm PIN" the user will receive an "Error" window which will inform the user the PIN does not match. Therefore, the user can re-enter a new PIN which is valid for the registration process.




[*Figure 4.9* – Registration – Full




Both *figure 4.9* and *figure 4.10* are part of the completion process. *Figure 4.9* displays the whole registration section complete and *figure 4.10* demonstrates the user being informed that they have "successfully created an account".

[*Figure 4.10* – Registration – Success]

## 4.3 Testing of Login Page

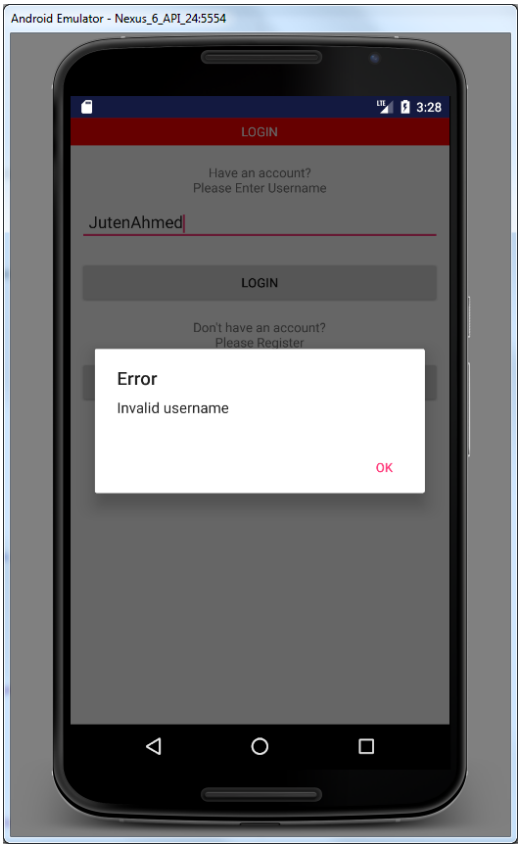

[*Figure 4.11* – Login – Registered

The login page has many validations that must be met.

1. Users must be able to input a username.
2. The username information must be from a registered account.
3. The user must be able to register if the user does not already have a registered account.
4. The user must be able to login if the registered username is valid.

*Figure 4.11* demonstrates the user "JutenAhmedZamindar" trying to login. As seen in *Chapter 4* sub-title 4.1 Testing of Registration Page, *figure 4.10 displays* the user has created an account. This will allow the user to proceed into the next page.

However, *figure 4.12* demonstrates when a user has not registered, the user "JutenAhmed" will not be able to login and proceed onto the next page. Furthermore, an "Error" message comes up to allow the user to know the user has entered an invalid username.




[*Figure 4.12* – Login – Unregistered

## 4.4 Testing of Memorable Word Page

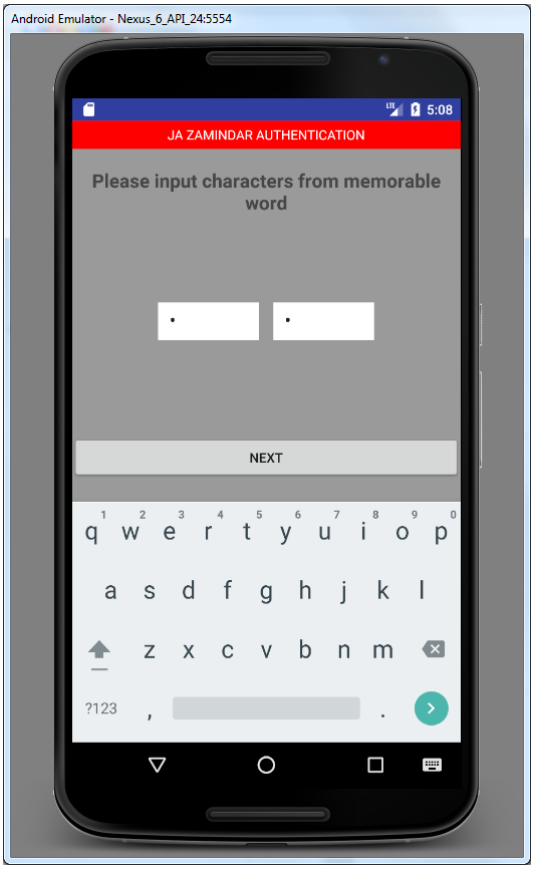

[*Figure 4.13* – Memorable Word – Letter " "]

The memorable word page has many validations that must be met.

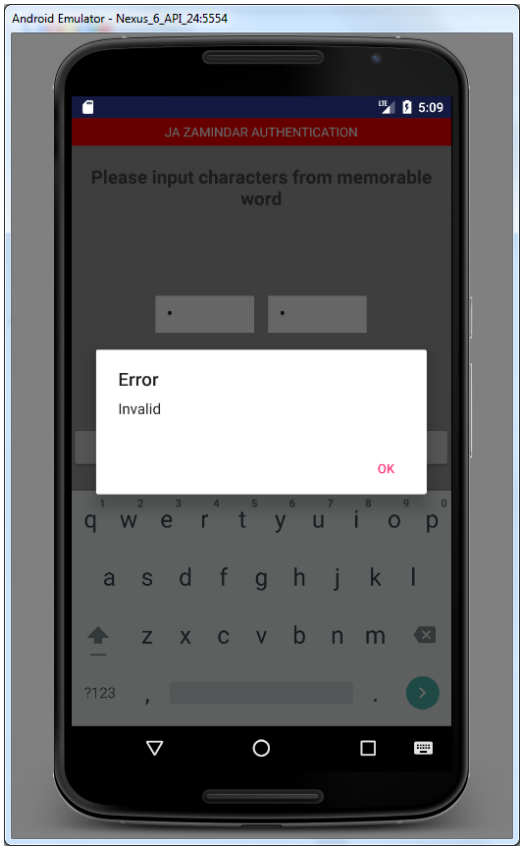
1. The user must input the letter that is asked for in each box.
2. The memorable word page must have 2 boxes which allow letters to be inputted.
3. The letters inputted into the box must be displayed as asterisks.
4. The memorable word must be for the account registered.
5. The memorable word must randomize every time the user logs into this page. For example, at one login the letters asked for could be 7 and 4. The next login must ask for a different set of letters i.e. 5 and 8.

*Figure 4.13* demonstrates the GUI of the memorable word page. The page asks the user to input the 7th letter and the 4th letter of the user's memorable word. This will then allow the user to enter the PIN entry page.

*Figure 4.14* demonstrates the asterisks which are put in place once the user enters the letter to maintain confidentiality of the letter.




[*Figure 4.14* – Memorable Word – Asterisks]



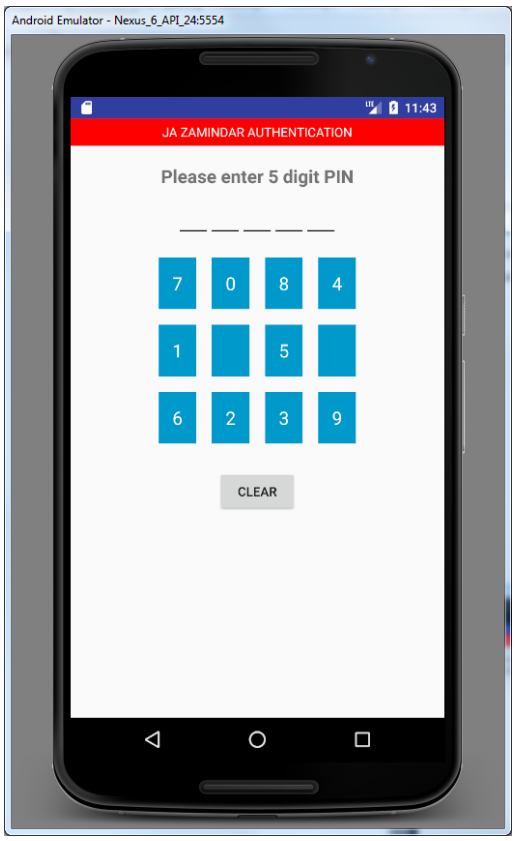

[*Figure 4.15* – Memorable Word – Error]

*Figure 4.15* demonstrates when a user enters the wrong information an error appears. This informs the user the memorable word letters entered were invalid therefore, allowing the user to re-enter the user's information. The letters asked on this occasion will not change to a different set of letters.

*Figure 4.16* demonstrates the fact the letters asked for are random when trying to log into the account on a different occasion. This time asking for letter 1 and letter 4.


[*Figure 4.16* – Memorable Word – Random Letter]

## 4.5 Testing of Randomized PIN Page




[*Figure 4.17* – PIN Entry – Randomized PIN Keypad]

The PIN entry page has many validations that must be met.

1. The user must be able to click on the PIN numbers the user has created.
2. The matrix must display as a 3x4 matrix.
3. The matrix must include 2 blank spaces.
4. The PIN once displayed must be shown as asterisks.
5. The PIN keypad must randomize each time the user logs into the application.
6. The user can only attempt 5 tries before being locked out.
7. Each time the user fails to input the correct PIN, the keypad must randomize before the user retries the PIN entry.
8. Once locked out the user can only re-login to the account after waiting 60 seconds.
9. If the user tries to login before 60 seconds has passed a message will come up to inform the user of how many seconds are left before the user can retry.
10. The user must be able to clear the keypad if they wish to.

*Figure 4.17* demonstrates the PIN keypad not being in the norm order of:

**1 2 3**

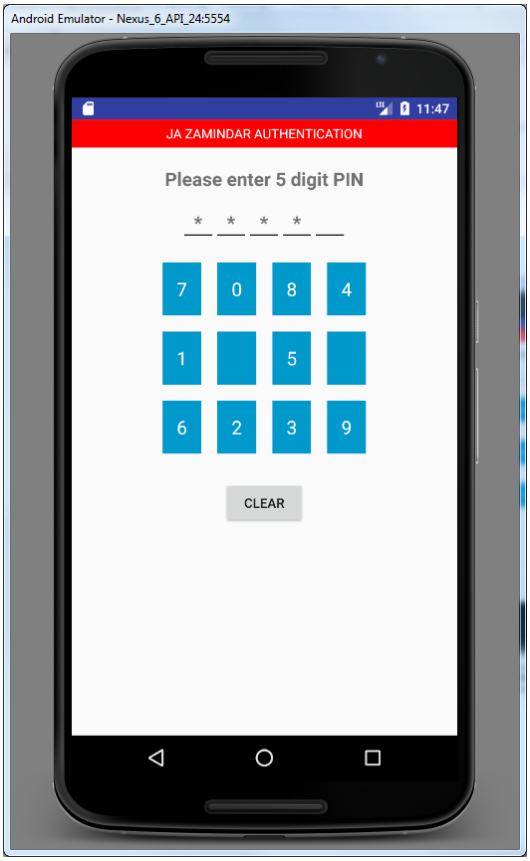
**4 5 6**

**7 8 9**

**0**

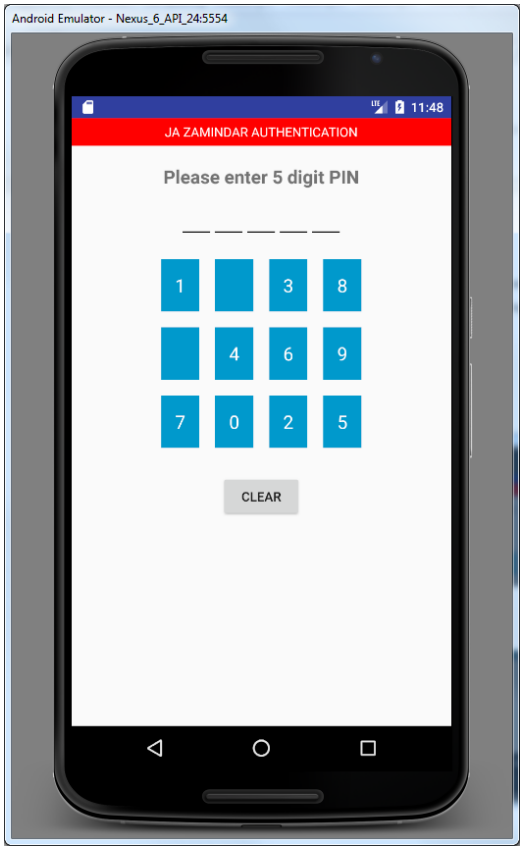
**CLEAR**

Instead the PIN Keypad is now randomized. The keypad has numbers between 0-9 displayed however in a random order. In addition, 2 blank entry spots which do nothing when clicked.



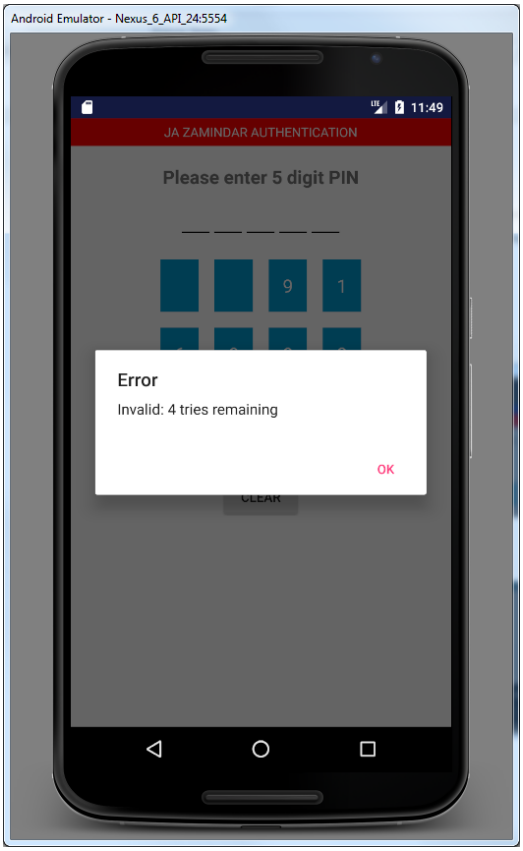

[*Figure 4.18* – PIN Entry – Asterisks]

*Figure 4.18* demonstrates once the user has inputted the users PIN the number inputted displays as an asterisk to maintain confidentiality of the user's PIN.

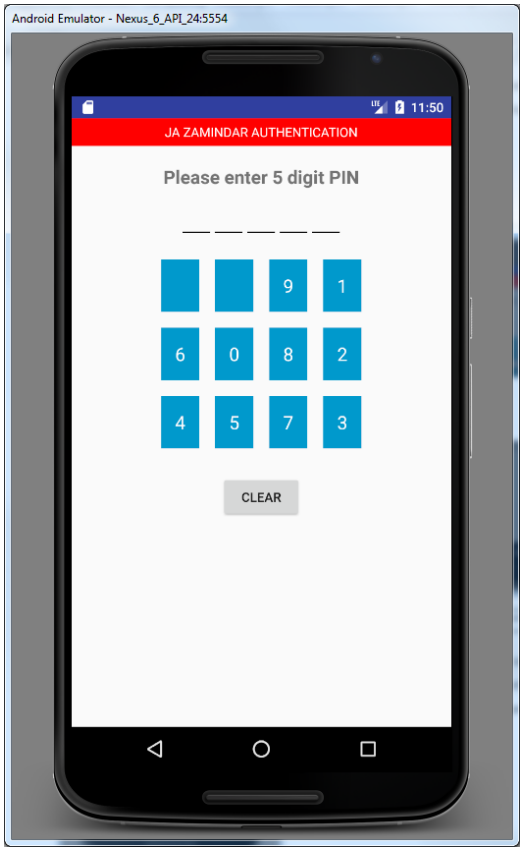



[*Figure 4.19* – PIN Entry – Randomized After re-entering application]

*Figure 4.19* demonstrates the application randomizes the PIN keypad after every attempt of reloading the application or a new login account.

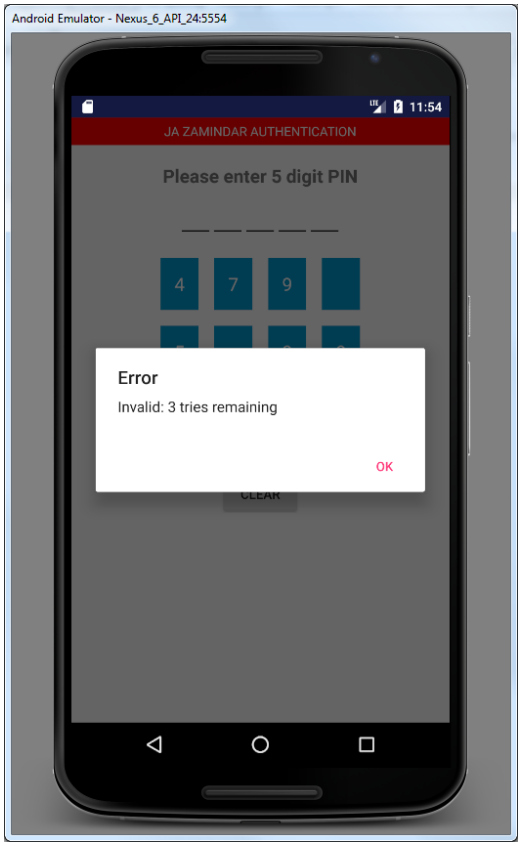



[*Figure 4.20* – PIN Entry – 4 tries remaining]



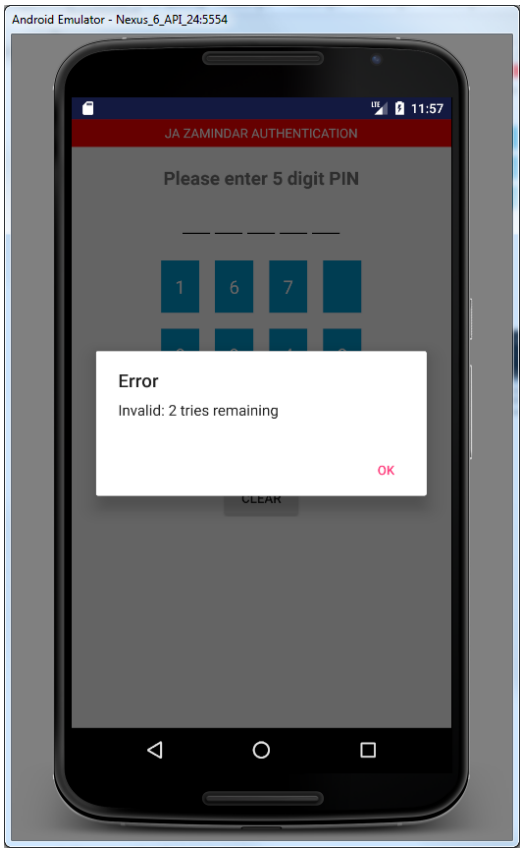

Images starting from *figure 4.21 to 4.27,* all demonstrate the user has 5 attempts to input the correct PIN before it locks the user out of the account for a specified time limit. Furthermore, each time the user inputs an invalid PIN the keypad randomizes.

[*Figure 4.21* – PIN Entry – Randomized 1st Attempt]



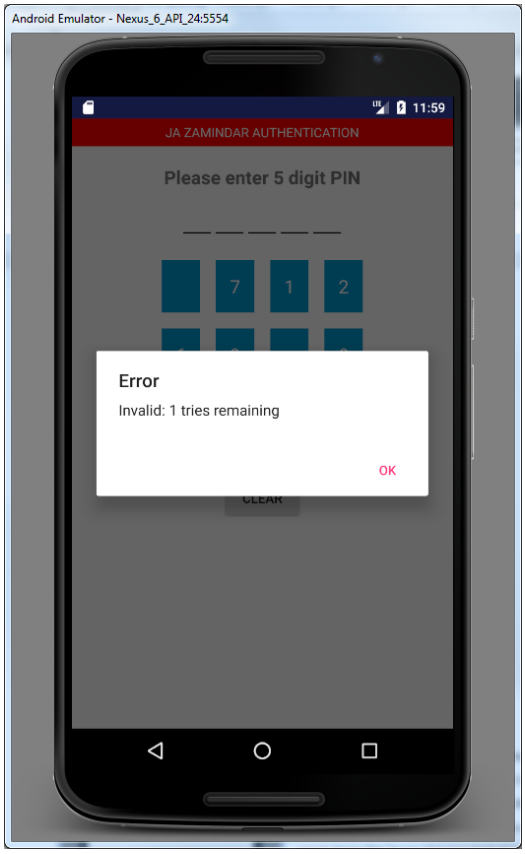

[*Figure 4.22* – PIN Entry – 3 tries remaining]


[*Figure 4.23* – PIN Entry – Randomized 2nd Attempt]




[*Figure 4.24* – PIN Entry – 2 tries remaining]


[*Figure 4.25* – PIN Entry – Randomized 3rd Attempt]




[*Figure 4.26* – PIN Entry – 1 tries remaining]


[*Figure 4.27* – PIN Entry – Randomized 4th Attempt]

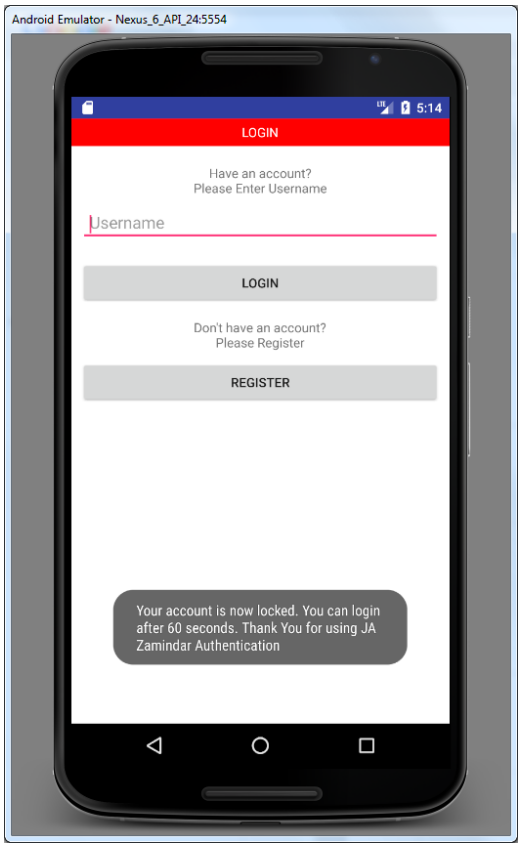



*Figure 4.28* demonstrates the lock out validation when a user attempts to login with the wrong PIN 5 times. The user automatically gets taken to the "LOGIN" page and a message is available for a short period to allow the user to know the "account is now locked." And the user can login after 60 seconds. There is finally a thank you message "Thank you for using JA Zamindar Authentication".

[*Figure 4.28* – PIN Entry – Locked Out 60 Seconds After 5th Invalid

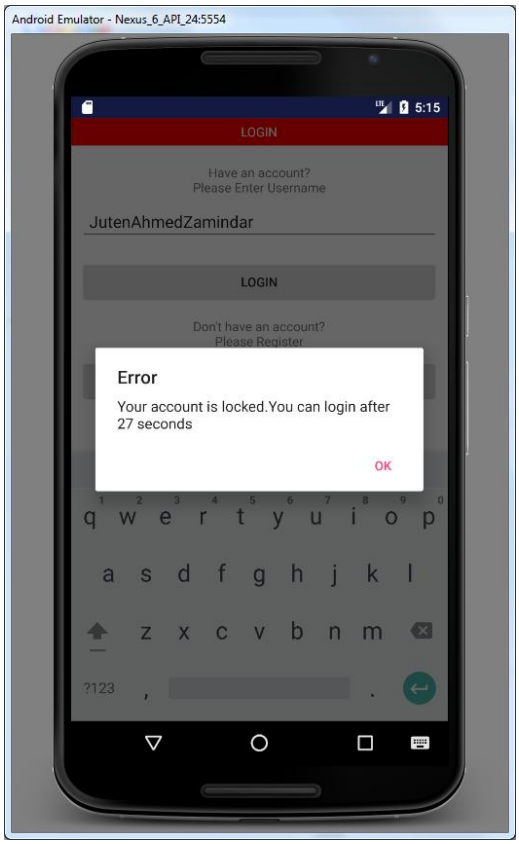
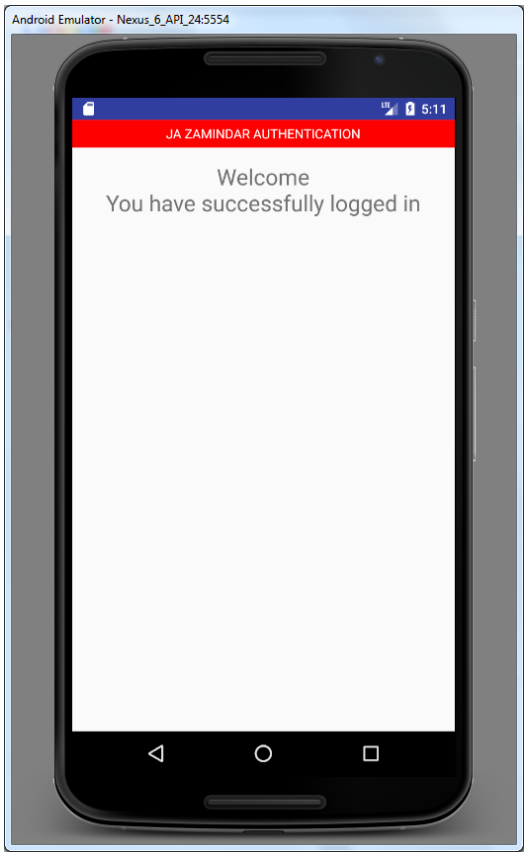



Figure 4.29 validates the user is not able to login to the account until the full 60 seconds has passed. If the user tries to log in before the 60 seconds has passed an error message shows informing the user that the user's account is locked and informs the user how many seconds, the user has remaining.

[*Figure 4.29* – PIN Entry – Lockout Time Remaining]

## 4.6 Testing of Success Page

The success pages' job is to allow the user know that the user has successfully logged in. *Figure 4.30* demonstrates the message on the screen which is informing the user that the user has successfully logged in with a welcome message.




[*Figure 4.30* – Success – SUCCESS!]

## 4.7 Summary

Overall within this chapter all testing regarding the process of the application is covered. Furthermore, validations of the application working. Every page has its specified validations of what the page's duty is.

In the next chapter, there will be an in-depth analysis and evaluation of the test results to see if there are any issues.

# **Chapter 5:** Analysis and Evaluation of Test Results

This chapter consists of the full analysis and evaluation of the artefact and a full analysis and evaluation of the test results the artefact has provided.

## 5.1 The Artefact

The artefact is based on authentication. This artefact allows users to create an account which is then saved on the user's local storage. The main purpose of this application is for demonstration purposes for the banking industry to implement within the banks IT security department as a new authentication method. This new authentication method is to be implemented on top of company data. For example, currently Barclays bank uses a standard PIN entry keypad requiring the user to input 5 digits of the user's PIN. The concept behind the artefact is to replace that screen excluding the registration section as the registration section would differ from all other companies depending on the company's requirements. The mandatory sections are the memorable word and PIN. Thus, allowing the application to work to its full potential.

When creating an account with this application, the user has many variables the user must input. Beginning with the user deciding a username as any authentication process would start with. Then creating a memorable word between 8 letters and 16 letters. The user then must confirm the memorable word to verify it was the same in case of any misspells or validation mistakes. Similarly, the user must create a PIN of 5 digits. The user must confirm the PIN and only then if all the rules of passphrase and PIN are followed can the account be created. Only to be taken to the next step.

The next step involves the user in validating the users account to make sure the testing of the application works. All aspects for example, the memorable word randomising which letter to input at every reload. In addition, the randomization of the PIN entry keypad. Once all the information has been inputted correctly in comparison to the registration process, only then will the user have been able to login successfully onto the final page of the account.

## 5.2 Testing JA Zamindar Authentication with the Public.

Testing the artefact with the public was 1$^{st}$ hand market research taken place. 50 individuals were asked to use the application to see the ease of use and to see if the public as users would benefit from this. A survey was given to each individual to complete with 5 questions to rate on a scale of which had different possible outcomes.

Findings from this research suggest each user believed the application was very easy to use. Each user thought it was a great adaptation of what is currently on the market. Furthermore, each user felt this authentication process was much more secure than what the users are currently using.

The questions included:

1. Would you like to see this authentication process in the current market?
2. Do you believe this authentication process is better than your current banks authentication process?
3. Do you feel this authentication process is more secure?
4. How easily did you feel this two-step authentication process was to use?
5. Are there any improvements you would recommend?

After full analysis of the whole pool of 50 public member's results, statistics show:

- 80% of the public would like to see this authentication process in the current market as 80% had rated an 8 or above on question 1.
- 70% of the public believes this authentication process is better than what the individuals currently uses as 70% had rated an 8 or above on question 2.
- 90% of the public believe this authentication process is more secure than what the individual is currently using as 90% had rated a 10 on question 3.
- 95% of the public found this authentication process average to very easy to use as 95% had rated between 1-5 on question 4.

The results prove that the artefact makes the 90% of the public feel secure when using this authentication process. Furthermore, 75% of the public from this pool believe it is better than the individuals current banks authentication process. Most importantly 80% of the public would like to see this authentication process in the current market. This could be a deciding factor for public users to change banks providing banks whom implement this authentication process more business. In addition, 95% did not believe this authentication process was difficult to use which is one of the main factors when selling to customers. Customer ease of use and satisfaction are major factors which determine whether a bank will have customers or not.

However, in reflection the reason the results did not come out as 100% as every question are due to many factors. One being that customers are not aware of how important security is and how to add additional countermeasures. Customers must be able to know all the countermeasures to prevent social engineering techniques like shoulder surfing to reduce the risk of the customer's personal data

being stolen. Furthermore, some of the answers might have been rushed, putting the integrity of the survey at risk.

Please view [***Appendix E*** – *A.9*] for an example of the survey provided.

There were some suggestions provided by the members of the public.

## 5.3 Suggested Improvements

Throughout the market research conducted, there was only two improvements two individuals suggested.

1. Creating bigger buttons for the randomized PIN keypad.
2. Making the application more colourful.

In conclusion, the reason for this could be the fact each of the individuals did not have a security background, instead a customer orientated background, therefore wanting customer ease rather than actual security concepts to help improve the authentication process.

## 5.4 Advantages of the Artefact

Advantages of this artefact is that there is a two-step authentication process. The criminal trying to commit cyber-crime must go through both parts of the authentication process. Even if the criminal knew the memorable word by using social engineering techniques and withdrawing personal information from the target for example, child's names, birthdays etc. The hacker would then need to be able to get through the second stage of a randomized PIN entry. This cannot be easily seen by someone shoulder suffering due to the fact the keys are not in the normal place where everyone would generally know.

## 5.5 Disadvantages of the Artefact

A disadvantage of this artefact is that there are still many improvements needed in prospective to creating an account. For example, having a limit on the amount of characters a username can have. Or what type of letters a username can have whether it be only upper case, only lower case or both. Including all aspects of characters i.e. being able to use punctuation and numbers. However, this artefact at the end of the day is just a prototype and the aim is to only show the functionality of the authentication method. Unfortunately, the application is not a full application linked to a corporate

bank database to finally disclose information of the users account. In addition, for the use of the public, users may not care about the security aspect as the users only seek customer ease.

## 5.6 Summary

The artefact has its flaws and strengths. However, overall the strengths outweigh any of the flaws. Regarding the cyber security aspect, everything this application provides works to the standards set out at the beginning.

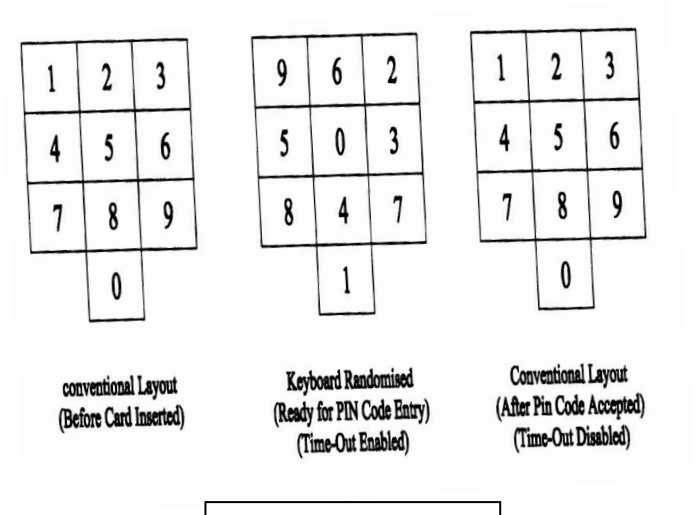
In the next chapter, there will be a comparison between this artefact and any previously published work.

# Chapter 6: Comparison of Work to Previous Published Work

This chapter consists of comparisons between previous published work and the artefact that has been created within this project.

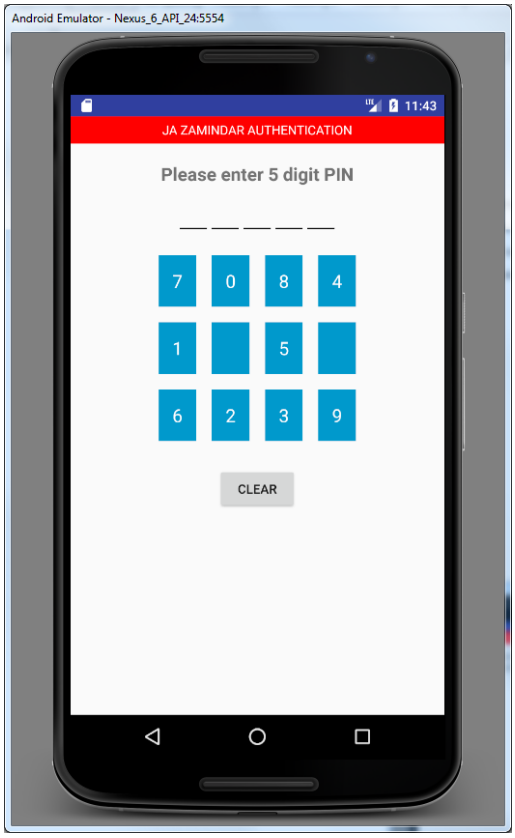
## 6.1 UK Patent Application: GB0210322.4

There have been patents already published in regards to using a randomized PIN keypad. After some thorough research the patents all have their own unique selling points. This patent is regarding a "keypad for generating code with scrambled displayed key sequences". However, none of these patents have been accepted. Due to the fact in the UK it is very difficult to obtain a patent in regards to an idea or concept. What is patentable is the design of a project which is different from all others out there in the current market.




[*Figure 6.1* – The Publication]

The idea of this patent was to create a randomized PIN keypad for ATMs.




[*Figure 6.2* – The Artefact]

The process would be, before entering a customer's card the layout of the keypad would stay as the norm of 1-9. However, when the customer as inputted the card to withdraw money, the keypad would randomize. This would include a timing system which allows the ATM to time out so the customers card would be given back to the customer. If successfully inputting the correct PIN, the PIN keypad would return to the normal state of 1-9.

The PIN Keypad system would also work for locked doors and any other authentication scenarios. The PIN keypad would work using emitted light which would randomize the numbers using an algorithm designed by the author.

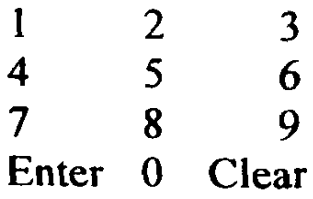
Please view [***Appendix C*** – *C.1*] for whole patent information.

In comparison to the artefact there are a lot of similarities in the sense both are randomized PIN keypads. However, in this case the artefact within this thesis is mainly for the banking mobile industry and other mobile device operating systems for example, android and IOS. With these differences the artefact also comes with a 2-step authentication process which would add an additional level of security for the criminal trying to commit cyber-crime. Thus, making it much more difficult to obtain the sensitive data. Furthermore, the additional 2 blank spots make it much more difficult for the social engineer to withdraw the information entered.

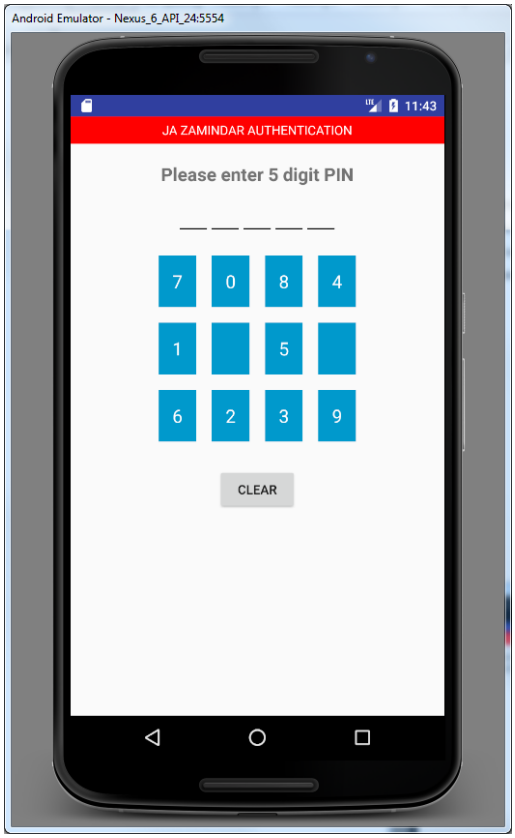
## 6.2 UK Patent Application: GB0623944.6

Another failed patent application which was not granted due to the fact of not being able to patent an idea. However, this patent was very like the artefact within this document, however, once again the patentee wanted to use it for ATM machines.

This patentee also tried to create a LCD monitor which would allow the customer to input the PIN at the ATM. This patent in the end was declined due to not providing enough information once the examiner requested.




[*Figure 6.3* – The Publication]




[*Figure 6.4* – The Artefact]

Please view [***Appendix D*** – *A.8*] for whole patent information.

In comparison, the artefact is still much more secure due to the fact of having a two-step authentication process and the two blank spaces. Even though this patent application is similar, the design of the artefact is much different, in addition to being used for a different platform.

## 6.3 Gaze Based Authentication

Gaze based authentication is a unique way to authenticate. Gaze based authentication has its pros and cons. The advantages of using gaze based authentication is the fact no one would know what the user is inputting as the user's password. The only vulnerability here is if there was a hacker which hacked the device and switched the front camera on the device to record what the user is doing. Then the hacker must be able to identify what characters the user is looking at by programming a software to do so. This would be a lengthy process.

A disadvantage to the gaze based authentication is the fact the amount of times the user would have to input the PIN in an event of the eye slipping. It is not 100% accurate due to the fact the iris on a user must be captured through the front camera of the device. Even though *figure 2.2* displayed in the previous chapter shows the password being correct there were many slips where the eye nearly went onto other letters, therefore in the testing stage, this suggests there would be mistakes occurring often. In addition, a user must know the exact location of the QWERTY keyboard to be able to move to the letter promptly. Furthermore, ethically if a user did not have an eye it would cause issues for the user to use. Therefore, the whole concept being obsolete.

In comparison to the artefact of using a randomized keypad, the randomized keypad is open for a mass audience. Ethically, an issue would only arise if there was someone whom does not have hands. But in this case the user would not be using mobile banking or a device in any case.

As displayed on *figure 6.4* the randomized keypad is unique and much more reliable than the gaze based authentication system due to the fact the margin of errors that would occur are much lower than the gaze based authentication system.

Consumers want the ease of use while being able to maintain the user's security and privacy. Using the artefacts method of authentication allows the user to get through a 2-step authentication process furthermore, a difficult authentication process in case of shoulder surfers trying to take the victims details.

## 6.4 Colour Light Based Authentication

The research conducted by De Luca, Hertzschuch, and Hussmann. (2010), is very difficult to use. This method is also a long process for the average mind to figure out what colour, and number the user has registered when registering. The complexity of ColorPIN is displayed in the previous chapter *figure 2.3*.

In comparison to the artefact, the ease of use is the main advantage here. Furthermore, the complexity of the ColorPIN makes it very difficult for the user to use as well due to the visual aspects. Ethically, if a user was colour blind, the application would be obsolete for users with that disability.

## 6.5 Banking Industry

Barclays authentication system is very similar to the artefact as demonstrated in the previous chapter *figure 2.4 and 6.4*. Barclays are currently using a 5-digit PIN authentication system which does not use a randomized keypad. In comparison to the artefact, the artefact is an improved version of this keypad design. Adding additional blank spaces and randomizing the keypad increases the integrity of the user inputting the password to verify it was the actual user due to the fact the PIN is much more difficult to guess.

In comparison, to Lloyds authentication process demonstrated in the previous chapter *figure 2.5. Processes are very alike, as the memorable word authentication process was* an addition to the whole authentication process for the artefact. However, the difference between Lloyds authentication process and the artefact is the fact the artefact includes both a memorable word authentication page and a randomized keypad demonstrated in the previous chapter *figure 3.6 and figure 3.8* whereas Lloyds only offers the memorable word system. On the other hand, Lloyds memorable word authentication page is more secure due to the fact Lloyds want users to input 3 characters rather than 2 required in the artefact.

In comparison to Tesco's banking authentication system. Tesco has the feature of using biometrics demonstrated in the previous chapter *figure 2.6* as a 1 step authentication with an additional backup of using a 5-digit PIN demonstrated in the previous chapter *figure 2.7*. Tesco has done well in this case as the fingerprint is unique to every person. However, ethically if a person(s) did not have a finger print due to medication or the loss of a hand or finger, this would be obsolete. Hence, having a backup process is very smart.

On the other hand, the artefact does not have to deal with any issues of these kinds due to the fact it is a much more secure authentication process as the user must go through 2 steps. This overall does increase the time to login to the account however the user must understand the importance of data security.

## 6.6 SteganoPIN

SteganoPIN have been able to create a stronger authentication system as seen in the previous chapter *figure 2.8*. This system has an advantage of not only allowing the user to use digits ranging from 0-9 but another 10 punctuation symbols. This provides a better statistical advantage for the user to being protected regarding a hacker trying to break the passcode. However, the issue here is the fact that it still does not really deter social engineering techniques like shoulder surfing as the user's passcode may still get stolen.

The 2nd authentication process the researchers have developed is very interesting. As if the user did input the correct PIN using the touch point system as displayed in the previous chapter *figure 2.9* anyone using social engineering techniques would still find it difficult to obtain the PIN. This PIN is also randomized which makes it very difficult to extract information from.

In comparison to the artefact, having a 2-step authentication process is very similar to this SteganoPIN publication. However, this publication is using 2 processes within one authentication page, for example the touch point and randomization within one page. This does make it stronger in the aspect of keeping the PIN hidden from shoulder surfers. On the other hand, the artefact does the same job, whilst creating a memorable word which would make it more difficult for the social engineer to withdraw the information as not only does the social engineer must try and obtain the full word of the memorable word, due to the fact the memorable word page on the artefact asks for randomized letters, but the engineer must also find the PIN. After 5 invalid tries the account would also get locked out.

## 6.7 Biometrics

Ohana, Phillips, Chen. (2013) research is only relevant to theft authentication rather than authenticating within the banking industry or mobile devices alone.

The concept is very good. As *figure 2.10* in the previous chapter demonstrates, the whole concept of this authentication process is revolved around fingerprint biometrics to help protect data if there

was an event of theft. This authentication process suggests to configure a fingerprint to both objects, the example used in the publication is a cell phone. Embedding the users fingerprint to the cell phone to decrypt the device and using the fingerprint authentication when charging the device. Thus, not allowing the thief to withdraw any information from the mobile device as the information is generally withdrawn through a USB port in a computer. In this occasion the fingerprint authentication would be required.

However, the disadvantage of this is the fact this whole concept is very costly for the companies that try and make the products. This is a very big disadvantage as a company would not want to employ more staff to try and manage the databases. Furthermore, having such sensitive data on a database like a fingerprint gives the identity of a human away.

In addition, there are many other protocols in place like deleting all the information remotely from a mobile phone. In comparison to the artefact, the artefact is less costly to deploy and very secure by nature of having a two-step authentication process to use as a login.

## 6.8 Summary

Overall there are examples of patents which have tried to go through the vigorous patent office in the UK but have had unsuccessful patents which are like this projects artefact. However, due to the fact this projects artefact is improved with not only the platform the artefact is targeted for but the concept of having a two-step authentication process for the mobile banking industry and other mobile devices.

In the next chapter the conclusions of the whole thesis will take place and any information regarding future work. Furthermore, the summary of contributions regarding the objectives of the project being met, and finally a personal reflection to the whole process.

# **Chapter 7:** Conclusions and Future Work

This chapter concludes the whole Thesis including a summary of contributions, recommendations and finally discussions of future work to advance the 2-step authentication PIN artefact.

## 7.1 Conclusions

In conclusion, this artefact created does what it designed for and has been created to demonstrate the authentication process which can be implemented into the banking industry for mobile banking applications currently in use.

This artefact is a cross between authentication processes built into one application to allow the users to be able to gain a more secure, reliable and complex login process. Overall increasing customer's satisfaction due to the fact of having the knowledge of having more security generally means the less risk the client has on being a victim of any type of cybercrime. This was the main aim, to minimize the risk of cyber criminals from being successful when committing cyber-crime.

Using this artefact, in theory will decrease the rate of cyber-crime in regards to theft of information as it makes the whole process of stealing the information much more difficult.

## 7.2 Summary of contributions

The main objectives of this project are to:

Create a fully working GUI which can demonstrate a login screen transitioning into a generic "bank account" screen which tells the customer the login was successful with a welcome message.

Creating a two-step authentication process

Creating a randomized PIN keypad as a 3x4 matrix as one of the authentication steps

Creating a memorable word system where the user inputs 1 character to access the second stage of the authentication process.

All four of these objectives have been met, with an addition to creating a registration page, login page and creating a memorable word page where the user can input two random characters rather than one. Furthermore, using local storage to contain all the user's information whilst leaving all the information encrypted using AES encryption. Furthermore, decrypting using the password set in the code "jutenahmedauthenticate" to decrypt the data.

## 7.3 Personal Reflection

Throughout this project, I had many struggles regarding time management and the actual development of the project. I had many personal issues which had affected my time management and mental stability which I had to go to the doctors for. This overall played a big part in the end result, however, I was able to try and overcome my personal issues to try and finish this project on time.

Whilst developing this project I had issues with learning android studios however, soon discovered it was pretty simple when creating the GUI. Majority of the code was pre-set and there were many helpful tips within the software which auto filled to help get what you needed. I did struggle with coding the PIN entry, memorable word page and registration as this was new to me. However, whilst doing research and playing around I was able to get the file to compile.

The best thing about using android studio is the fact you can use a VM mobile phone. So once any changes are made you can review it straight away rather than creating an .apk file to install on your phone.

Whilst creating the thesis I had difficulties finding many references for my work due to the fact majority of any comparisons or the literature review were from journals or other reputable websites. However, on other sections within my thesis there was not much to reference as it was all regarding analysis and evaluating my artefact.

Overall I am very happy with the end result, as I believe I have reached the target I wanted. Creating this application has given me some self-satisfaction and is an achievement for me. I am looking forward to working on this application in the future to improve and to develop further to maybe take to a new company or business to implement into their authentication processes.

## 7.4 Future Work

There can be many more features added to this artefact for example, creating more processes to increase the security and complexity of the authentication process.

### 7.4.1 Username Being Between a Certain Amount of Characters

This new process would allow the artefact to be able have a certain limit to the characters to create a username for any banking industry to be able to manage their clients. Currently the registration process does not have any limitations on how many characters the username can be. The username can even be 1 letter which is not a good countermeasure for hackers.

### 7.4.2 Username Can Include All Types of Characters

Currently the username can include all types of characters including numbers, letters and punctuation. Again, for future use and manging company databases, it would be advisable to create a certain amount of characters that can be used.

### 7.4.3 Memorable Word to Include Any Characters

Currently, the memorable word section on the registration page does not allow you to create a memorable word with any characters. It is limited to only being lowercase letters. In the future, the memorable word should be able to use any characters including numbers, letters and punctuation to make the memorable word that much difficult to guess. This would be a perfect countermeasure to anyone using social engineering techniques to guess what someone's memorable word could be.

### 7.4.4 A Longer PIN

Currently the PIN is of only 5 digits. This is the norm within the banking industry. However, in future, this can be increased to increase the amount of combinations the password could be causing the password to be much more difficult to decrypt or hack.

### 7.4.5 Memorable Word Authentication Page Can Require More Than 2 Characters

Currently the memorable word authentication page only requires two letters being inputted depending on what letter has been asked for. In the future, to improve the artefact, there can be more than two letters being required to authenticate. For example, having 3-4 letters would increase the security and make it much more difficult to login to the PIN entry page.

### 7.4.6 Memorable Word Cannot Have 3 Repetitive Letters

Currently the memorable word can have any letters as long as it is in lowercase and within 8-16 characters. A minimum of 8 letters and a maximum of 16 letters. In the future, it is advisable to make sure the user cannot create a memorable word with more than 2 repetitive letters due to the fact the letters might be too common when trying to get through that authentication process on the memorable word page within the application.

# Bibliography

Alexander De Luca, Katja Hertzschuch, Heinrich Hussmann. (2010). *ColorPIN – Securing PIN Entry through Indirect Input.* Available: https://www.medien.ifi.lmu.de/pubdb/publications/pub/deluca2010chi/deluca2010chi.pdf. Last accessed 5th March 2017.

Barclays Banking Group. (1999). *Banking Security Advise.* Available: http://www.barclays.co.uk/Helpsupport/Onlinebankingsecurity/P1242657728006 . Last accessed 11th March 2017.

Barclays Banking Group. (1999). *Mobile Banking.* Available: http://www.barclays.co.uk/BarclaysMobileBanking/MobileBankingapp/P1242609123821 . Last accessed 11th March 2017.

Changingminds.org. (2016). *Learning Recall Related to Type of Presentation.* Available: http://changingminds.org/explanations/learning/active_learning.htm. Last accessed 23rd April 2017.

Department for Culture, Media & Sport and National Cyber Security Centre. (2017). *Nearly seven in ten large companies identified a breach or attack, new Government statistics reveal.* Available: https://www.gov.uk/government/news/almost-half-of-uk-firms-hit-by-cyber-breach-or-attack-in-the-past-year. Last accessed 8th April 2017.

Donny Jacob Ohana, Liza Phillips, Lei Chen. (2013). *Preventing Cell Phone Intrusion and Theft using Biometrics.* Available: http://ieeexplore.ieee.org/document/6565247/. Last accessed 18th April 2017.

IEEE. (2017). *PIN Authentication.* Available: http://ieeexplore.ieee.org/search/searchresult.jsp?queryText=PIN%20authentication&newsearch=true. Last accessed 15th April 2017.

K.Kiruthika, D.Jennifer, K.Sangeetha, Jackulin.C, R.Shalini. (2016). *A Secure Pin Authentication Method against Shoulder Surfing Attacks.* Available: https://www.ijecs.in/issue/v5-i10/62%20ijecs.pdf. Last accessed 16th April 2017.

Lloyds Bank Group. (1999). *The Freedom of Mobile Banking.* Available: https://www.lloydsbank.com/online-banking/mobile-banking.asp. Last accessed 10th March 2017.

Manu Kumar, Tal Garfinkel, Dan Boneh, Terry Winograd. (2007). *Reducing Shoulder-surfing by Using Gaze-based Password Entry.* Available: http://hci.stanford.edu/cstr/reports/2007-05.pdf. Last accessed 8th March 2017.

Tesco PLC. (2017). *Banking.* Available: http://www.tescobank.com/credit-cards/. Last accessed 15 April 2017.

Tutorials Point. (2015). *SDLC - RAD Model.* Available: https://www.tutorialspoint.com/sdlc/sdlc_rad_model.htm. Last accessed 20th March 2017.

Tutorials Point. (2017). *Test Strategy.* Available: https://www.tutorialspoint.com/software_testing_dictionary/test_strategy.htm. Last accessed 29th April 2017.

Watson, Tim and Harris, Vicky and Harper, Gavin. (10th Jan 2007). *Random generated PIN keypad.* Available: https://www.ipo.gov.uk/p-pj?startYear=2006&startMonth=December&startDay=27th+-+6136&endYear=2017&endMonth=April&endDay=12th+-+6673&filter=randomized+keypad&perPage=10&sort=Publication+Date. Last accessed 24th March 2017.

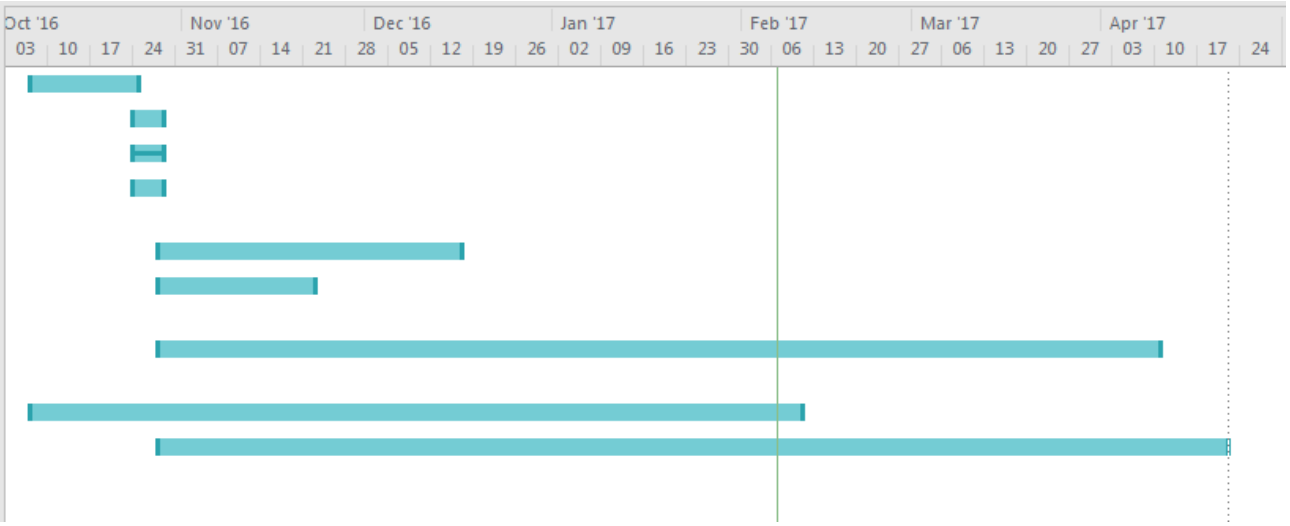
Wealth & Finance International. (2015). *Cybercrime Incidents on the Rise.* Available: http://www.wealthandfinance-intl.com/cybercrime-incidents-on-the-rise. Last accessed 8th April 2017.

Zhi Li, Qibin Sun, Yong Lian, D.D. Giusto. (2005). *An Association-Based Graphical Password Design Resistant to Shoulder-Surfing Attack.* Available: http://ieeexplore.ieee.org/document/1521406/. Last accessed 15th March 2017.

# **Appendix A** - Introduction

## A.1 Gantt Chart

| Task Mode | Task Name | Duration | Start | Finish | Predecesso |
|---|---|---|---|---|---|
| | Final Project Idea | 12 days | Fri 07/10/16 | Mon 24/10/1 | |
| | Draft Proposal | 5 days | Mon 24/10/1 | Fri 28/10/16 | |
| | Ethics Form | 5 days | Mon 24/10/1 | Fri 28/10/16 | |
| | Final Proposal with Ethics Form | 5 days | Mon 24/10/16 | Fri 28/10/16 | |
| | Contextual Report | 36 days | Fri 28/10/16 | Fri 16/12/16 | |
| | Planning the Artefact | 18 days | Fri 28/10/16 | Tue 22/11/16 | |
| | Developing the Artefact | 117 days | Fri 28/10/16 | Mon 10/04/17 | |
| | Reflective Report | 91 days | Fri 07/10/16 | Fri 10/02/17 | |
| | Thesis with Final Submission of Whole Project | 126 days | Fri 28/10/16 | Fri 21/04/17 | |



# **Appendix B** - Design and System Implementation

## B.1 Code

### B.1.1 GUI Login Page:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    tools:context="com.secretword.SignupActivity">

    <LinearLayout android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:layout_centerInParent="true">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="30dp"
            android:background="#FF0000">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerInParent="true"
                android:text="LOGIN"
                android:textColor="#FFFFFF"
                />

        </RelativeLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:orientation="vertical"
            android:layout_marginTop="20dp"
            android:paddingBottom="20dp">

            <TextView
                android:id="@+id/textView"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center_horizontal"
                android:gravity="center_horizontal"
                android:text="Have an account?\nPlease Enter Username" />

            <EditText
                android:id="@+id/txtusername"
                android:layout_width="match_parent"
                android:layout_height="40dp"
                android:layout_marginTop="10dp"
                android:hint="Username"
                android:inputType="textPersonName"
                android:padding="10dp"></EditText>

            <Button
```

```
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:text="LOGIN"
            android:onClick="loginClick"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:layout_marginTop="15dp"
            android:gravity="center_horizontal"
            android:text="Don't have an account?\nPlease Register" />

        <Button
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:text="REGISTER"
            android:onClick="registerClick"/>

    </LinearLayout>

</LinearLayout>

</RelativeLayout>
```

### B.1.2 GUI PIN Page:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_pin"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.secretword.PinActivity">

    <RelativeLayout
        android:id="@+id/layouttop"
        android:layout_width="match_parent"
        android:layout_height="30dp"
        android:background="#FF0000">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerInParent="true"
            android:text="JA ZAMINDAR AUTHENTICATION"
            android:textColor="#FFFFFF"
            />

    </RelativeLayout>

    <TextView
        android:id="@+id/txtpintext"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_centerHorizontal="true"
        android:layout_below="@id/layouttop"
        android:textStyle="bold"
        android:textSize="20sp"
```

```
        android:text="Please enter 5 digit PIN" />

    <LinearLayout
        android:id="@+id/pincontainer"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_below="@id/txtpintext"
        android:gravity="center_horizontal"
        android:orientation="horizontal">

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <TextView
                android:id="@+id/txtpin1"
                android:layout_width="30dp"
                android:layout_height="wrap_content"
                android:gravity="center_horizontal"
                android:textStyle="bold"
                android:textSize="20sp"
                android:text="" />

            <include layout="@layout/pin_line"></include>


        </LinearLayout>

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:orientation="vertical">

            <TextView
                android:id="@+id/txtpin2"
                android:layout_width="30dp"
                android:layout_height="wrap_content"
                android:gravity="center_horizontal"
                android:textStyle="bold"
                android:textSize="20sp"
                android:text="" />

            <include layout="@layout/pin_line"></include>


        </LinearLayout>

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:orientation="vertical">

            <TextView
                android:id="@+id/txtpin3"
                android:layout_width="30dp"
                android:layout_height="wrap_content"
                android:gravity="center_horizontal"
                android:textStyle="bold"
                android:textSize="20sp"
                android:text="" />

            <include layout="@layout/pin_line"></include>
```

```
        </LinearLayout>

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:orientation="vertical">

            <TextView
                android:id="@+id/txtpin4"
                android:layout_width="30dp"
                android:layout_height="wrap_content"
                android:gravity="center_horizontal"
                android:textStyle="bold"
                android:textSize="20sp"
                android:text="" />

            <include layout="@layout/pin_line"></include>


        </LinearLayout>

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:orientation="vertical">

            <TextView
                android:id="@+id/txtpin5"
                android:layout_width="30dp"
                android:layout_height="wrap_content"
                android:gravity="center_horizontal"
                android:textStyle="bold"
                android:textSize="20sp"
                android:text="" />

            <include layout="@layout/pin_line"></include>


        </LinearLayout>

    </LinearLayout>


    <LinearLayout
        android:id="@+id/pindigits"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_below="@id/pincontainer"
        android:orientation="vertical">

    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:orientation="horizontal"
        android:layout_centerHorizontal="true"
        android:layout_below="@id/pindigits">

        <Button android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="clearClick"
            android:text="CLEAR"></Button>
```

```
        <!--<Button android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="loginClick"
            android:text="LOGIN"></Button>-->


    </LinearLayout>


</RelativeLayout>
```

### B.1.3 GUI Memorable Word Page:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_secret_word"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.secretword.SecretWordActivity"
    android:background="#9a9a9a">

    <RelativeLayout
        android:id="@+id/layouttop"
        android:layout_width="match_parent"
        android:layout_height="30dp"
        android:background="#FF0000">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerInParent="true"
            android:text="JA ZAMINDAR AUTHENTICATION"
            android:textColor="#FFFFFF"
            />

    </RelativeLayout>

    <TextView
        android:id="@+id/txtpintext"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_below="@id/layouttop"
        android:layout_centerHorizontal="true"
        android:textStyle="bold"
        android:textSize="20sp"
        android:gravity="center_horizontal"
        android:text="Please input characters from memorable word" />

    <LinearLayout
        android:id="@+id/randomcontainer"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="93dp"
        android:orientation="horizontal"
        android:layout_below="@+id/txtpintext"
        android:layout_centerHorizontal="true">

        <EditText android:id="@+id/txtrandom1"
            android:layout_width="wrap_content"
```

```
            android:layout_height="40dp"
            android:padding="10dp"
            android:inputType="textPassword"
            android:maxLength="1"
            android:background="#FFFFFF"
            android:hint=""></EditText>

        <EditText
            android:id="@+id/txtrandom2"
            android:layout_width="wrap_content"
            android:layout_height="40dp"
            android:layout_below="@id/txtrandom1"
            android:layout_marginLeft="15dp"
            android:background="#FFFFFF"
            android:hint=""
            android:inputType="textPassword"
            android:maxLength="1"
            android:padding="10dp"></EditText>

    </LinearLayout>

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="100dp"
        android:text="NEXT"
        android:maxLength="16"
        android:onClick="loginClick"
        android:layout_below="@+id/randomcontainer"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

</RelativeLayout>
```

### B.1.4 GUI Registration Page:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    tools:context="com.secretword.SignupActivity">

    <LinearLayout android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="30dp"
            android:background="#FF0000">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerInParent="true"
                android:text="REGISTRATION"
                android:textColor="#FFFFFF"
```

```
            />

        </RelativeLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:orientation="vertical"
            android:layout_marginTop="20dp"
            android:paddingBottom="20dp">

            <EditText android:id="@+id/txtusername"
                android:layout_width="match_parent"
                android:layout_height="40dp"
                android:layout_marginTop="10dp"
                android:padding="10dp"
                android:inputType="textPersonName"
                android:hint="Username"></EditText>

            <TextView
                android:id="@+id/textView2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="left"
                android:text="Memorable Word can only consist of lower case letters
(a-z)"
                android:textColor="#000000" />

            <EditText android:id="@+id/txtsecretword"
                android:layout_width="match_parent"
                android:layout_height="40dp"
                android:layout_marginTop="10dp"
                android:padding="10dp"
                android:inputType="textPassword"
                android:maxLength="16"
                android:hint="Memorable Word 8-16 Characters"></EditText>

            <EditText android:id="@+id/txtconfirmsecretword"
                android:layout_width="match_parent"
                android:layout_height="40dp"
                android:layout_marginTop="10dp"
                android:padding="10dp"
                android:inputType="textPassword"
                android:maxLength="16"
                android:hint="Confirm Memorable Word"></EditText>

            <TextView
                android:id="@+id/textView3"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="left"
                android:text="PIN can not consist of 3 repetitive numbers"
                android:textColor="#000000" />

            <EditText android:id="@+id/txtpin"
                android:layout_width="match_parent"
                android:layout_height="40dp"
                android:layout_marginTop="10dp"
                android:padding="10dp"
                android:inputType="numberPassword"
                android:maxLength="5"
                android:hint="Please enter a 5 Digit PIN"></EditText>

            <EditText android:id="@+id/txtconfirmpin"
                android:layout_width="match_parent"
                android:layout_height="40dp"
```

```
                android:layout_marginTop="10dp"
                android:padding="10dp"
                android:inputType="numberPassword"
                android:maxLength="5"
                android:hint="Confirm PIN"></EditText>

            <Button
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="20dp"
                android:text="REGISTER"
                android:onClick="loginClick"/>

        </LinearLayout>

    </LinearLayout>

</RelativeLayout>
```

### B.1.5 GUI Success Page:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_welcome"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.secretword.WelcomeActivity">

    <RelativeLayout
        android:id="@+id/layouttop"
        android:layout_width="match_parent"
        android:layout_height="30dp"
        android:background="#FF0000">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerInParent="true"
            android:text="JA ZAMINDAR AUTHENTICATION"
            android:textColor="#FFFFFF"
            />

    </RelativeLayout>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:textSize="24sp"
        android:layout_below="@id/layouttop"
        android:layout_centerHorizontal="true"
        android:gravity="center_horizontal"
        android:text="Welcome\nYou have successfully logged in" />

</RelativeLayout>
```

## B.1.6 PIN Page:

```
package com.secretword;

import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.LinearLayout;
import android.widget.TextView;

import com.jutenahmed.android.UIHelper;

import java.util.Random;
import java.util.Vector;

public class PinActivity extends CommonActivity {

    private static final String TAG=PinActivity.class.getName();

    private ViewGroup layoutPinDigits;
    //private TextView txtpin;
    private String pin="";
    private Vector<String> pinDigits=new Vector<String>();
    private Vector<String> digitsDisplayed=new Vector<String>();


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pin);
        layoutPinDigits=(ViewGroup)findViewById(R.id.pindigits);
        initializePinKeypad();
    }

    private boolean isDigitDisplayed(String str){
        //check if digit is already displayed on screen
        //return (str.compareTo("0")==0 || digitsDisplayed.contains(str));
        return (digitsDisplayed.contains(str));
    }

    //return true of its a blank spot else false
    private boolean showBlankKey(String str){
        //10 and 11 are blank spots within app
        return (str.compareTo("10")==0 || str.compareTo("11")==0);
    }

    private void initializePinKeypad(){
        Log.d(TAG,"initializePinKeypad");
        layoutPinDigits.removeAllViews();
        LinearLayout layoutPinRow=null;
        LinearLayout.LayoutParams lps=new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT,ViewGroup.LayoutParam
s.WRAP_CONTENT);
        lps.setMargins(30,30,30,30);
        LayoutInflater inflater=getLayoutInflater();
        digitsDisplayed.clear();
        pinDigits.clear();
        pin="";
        //loop through 0 to 11 .... digits 10 and 11 are blank spots
        for (int a=0;a<=11;a++){
            //initialize random class to get always random digit
            Random rnd=new Random();
```

```
            //get random digit to display everytime
            String digitToDisplay=String.valueOf(rnd.nextInt(12));
            //check if its already displayed
            while(isDigitDisplayed(digitToDisplay)){
                digitToDisplay=String.valueOf(rnd.nextInt(12));
            }
            //if digit is unique then add to collection
            digitsDisplayed.add(digitToDisplay);

            //if it hasn't initialized the digits row then this code will create a
new one
            //if (layoutPinRow==null){
            if (a%4==0){

layoutPinRow=(LinearLayout)inflater.inflate(R.layout.layout_pindigitrow,null);
                layoutPinDigits.addView(layoutPinRow);
            }
            //create layout for separate digit
            ViewGroup
vgPin=(ViewGroup)inflater.inflate(R.layout.layout_pindigit,null);
            vgPin.setLayoutParams(lps);
            TextView txt=(TextView)vgPin.findViewById(R.id.txtdigit);
            //if key is not blank spot
            if (!showBlankKey(digitToDisplay)) {

//vgPin.setBackgroundColor(getResources().getColor(android.R.color.holo_blue_dark))
;
                txt.setText(digitToDisplay);
            }else{
                //vgPin.setBackgroundColor(Color.BLACK);
                txt.setText("6");

txt.setTextColor(getResources().getColor(android.R.color.holo_blue_dark));
            }
            //create cookie object to attach to each digit
            PinState ps=new PinState();
            ps.value=digitToDisplay;
            ps.pressed=false;
            vgPin.setTag(ps);

vgPin.setBackgroundColor(getResources().getColor(android.R.color.holo_blue_dark));

            layoutPinRow.addView(vgPin);


        }
    }

    public void clearClick(View v){
        Log.d(TAG,"clearClick");
        //clear all pin data entered
        pin="";
        setPinText("");
        pinDigits.clear();
    }

    public void loginClick(View v){
        Log.d(TAG, "loginClick");

        //check if pin entered is the user's pin
        if (pin.compareTo(AppCache.currentUser.pin)!=0) {
            //if invalid it will increase the invalid counter
            AppCache.invalidPinCount++;
            //check how many tries are remaining
            int diff=AppCache.INVALID_PIN_LIMIT-AppCache.invalidPinCount;
            //if there are still tries show message
            if (diff>0) {
                UIHelper.msbox("Error", "Invalid: " + String.valueOf(diff)+" tries
```

```
remaining", PinActivity.this);
                clearClick(null);
                this.initializePinKeypad();
            }else { //if there are no more tries
                //lock the account
                LocalStorage ls = new LocalStorage(getApplicationContext());
                ls.setLocked();
                //get remaining seconds for unlock
                long remain=ls.getUnlockRemain();
                //show how many seconds are remaining to unlock
                UIHelper.makeLongToast("Your account is now locked. You can login
after "+String.valueOf(remain)+" seconds. Thank You for using JA Zamindar
Authentication", PinActivity.this);
                //move back to login screen as account is locked now
                Intent i = new Intent(PinActivity.this,LoginActivity.class);
                startActivity(i);
                finish();
            }
            return;
        }

        //if pin is valid

        //reset the invalid pin counter
        AppCache.invalidPinCount=0;
        //start welcome screen
        Intent i=new Intent(PinActivity.this,WelcomeActivity.class);
        startActivity(i);
        finish();
    }

    public void digitClicked(View v){
        Log.d(TAG,"digitClicked");
        //get cookie object of clicked counter
        PinState ps=(PinState)v.getTag();
        Log.d(TAG,ps.value);

        //if its a blank spot do nothing
        if (showBlankKey(ps.value)){
            Log.d(TAG,"yes blank");
            return;
        }

        ps.pressed=!ps.pressed;

        if (pinDigits.size()>=5){
            return;
        }

        //add the clicked pin to the collection of already displayed pin
        pinDigits.add(ps.value);
        //set * in place of all pins entered
        setPinText("*");
        pin="";
        for (int a=0;a<pinDigits.size();a++){
            pin+=pinDigits.get(a);
        }

        //if 5 digits are entered then login
        if (pinDigits.size()==5){
            loginClick(null);
        }
    }

    private void setPinText(String tx){
        for (int a=0;a<pinDigits.size();a++){
            TextView txt=null;
            if (a==0){
```

```
                txt=(TextView)findViewById(R.id.txtpin1);
            }else if (a==1){
                txt=(TextView)findViewById(R.id.txtpin2);
            }else if (a==2){
                txt=(TextView)findViewById(R.id.txtpin3);
            }else if (a==3){
                txt=(TextView)findViewById(R.id.txtpin4);
            }else if (a==4){
                txt=(TextView)findViewById(R.id.txtpin5);
            }

            if (txt!=null){
                txt.setText(tx);
            }
        }
    }
}
```

### B.1.7 AES Encryption 256 bit:

```
package com.secretword;

import android.util.Base64;
import android.util.Log;

import java.io.UnsupportedEncodingException;
import java.security.GeneralSecurityException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

import javax.crypto.Cipher;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;


public final class AESEncryption {

    private static final String TAG = "AESEncryption";

    //AESEncryption uses CBC and PKCS7Padding
    private static final String AES_MODE = "AES/CBC/PKCS7Padding";
    private static final String CHARSET = "UTF-8";

    //AESEncryption uses SHA-256 (and so a 256-bit key)
    private static final String HASH_ALGORITHM = "SHA-256";

    private static final byte[] ivBytes = {0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};

    public static boolean DEBUG_LOG_ENABLED = false;


    // Generates a SHA256 hash of the password which is used as key

    private static SecretKeySpec generateKey(final String password) throws
NoSuchAlgorithmException, UnsupportedEncodingException {
        final MessageDigest digest = MessageDigest.getInstance(HASH_ALGORITHM);
        byte[] bytes = password.getBytes("UTF-8");
        digest.update(bytes, 0, bytes.length);
        byte[] key = digest.digest();
        SecretKeySpec = new SecretKeySpec(key, "AES");
        return secretKeySpec;
```

```
    }


    // Encrypt and encode the messages using 256-bit AES with key generated from
password.

    public static String encrypt(final String password, String message)
            throws GeneralSecurityException {

        try {
            //generate key to start encryption using the password
            final SecretKeySpec key = generateKey(password);
            log("message", message);
            //encrypt data using key , initializationVector and bytes of message
            byte[] cipherText = encrypt(key, ivBytes, message.getBytes(CHARSET));
            //NO_WRAP is important as was getting \n at the end
            //encode to base64
            String encoded = Base64.encodeToString(cipherText, Base64.NO_WRAP);
            log("Base64.NO_WRAP", encoded);
            return encoded;
        } catch (UnsupportedEncodingException e) {
            if (DEBUG_LOG_ENABLED)
                Log.e(TAG, "UnsupportedEncodingException ", e);
            throw new GeneralSecurityException(e);
        }
    }



    public static byte[] encrypt(final SecretKeySpec key, final byte[] iv, final
byte[] message)
            throws GeneralSecurityException {
        //create java's cipher class for AES encryption
        final Cipher = Cipher.getInstance(AES_MODE);
        //initialize parameters with initializationVector
        IvParameterSpec ivSpec = new IvParameterSpec(iv);
        //initialize cipher class
        cipher.init(Cipher.ENCRYPT_MODE, key, ivSpec);
        //encrypt the data
        byte[] cipherText = cipher.doFinal(message);
        //return encrypted data
        return cipherText;
    }


    //Decrypt and decode ciphertext using 256-bit AES with key generated from
password

    public static String decrypt(final String password, String
base64EncodedCipherText)
            throws GeneralSecurityException {

        try {
            //generate key using the password
            final SecretKeySpec key = generateKey(password);

            log("base64EncodedCipherText", base64EncodedCipherText);
            //decode from base64
            byte[] decodedCipherText = Base64.decode(base64EncodedCipherText,
Base64.NO_WRAP);
            //decode data using key and initializationVector
            byte[] decryptedBytes = decrypt(key, ivBytes, decodedCipherText);
            //convert bytes to string
            String message = new String(decryptedBytes, CHARSET);
            log("message", message);
            //return string
            return message;
        } catch (UnsupportedEncodingException e) {
```

```
            if (DEBUG_LOG_ENABLED)
                Log.e(TAG, "UnsupportedEncodingException ", e);

            throw new GeneralSecurityException(e);
        }
    }


    public static byte[] decrypt(final SecretKeySpec key, final byte[] iv, final
byte[] decodedCipherText)
            throws GeneralSecurityException {
            //get java's Cipher class for AES encryption
            final Cipher = Cipher.getInstance(AES_MODE);
            //initialize parameters with InitializationVector
            IvParameterSpec ivSpec = new IvParameterSpec(iv);
            //initialize Cipher class
            cipher.init(Cipher.DECRYPT_MODE, key, ivSpec);
            //decrypt the message and receive bytes
            byte[] decryptedBytes = cipher.doFinal(decodedCipherText);
            //return decryptedData
            return decryptedBytes;
    }

    private static void log(String what, String value) {
        //check if logging is enabled
        if (DEBUG_LOG_ENABLED)
            //log to android studio console
            Log.d(TAG, what + "[" + value.length() + "] [" + value + "]");
    }

    private AESEncryption() {
    }
}
```

### B.1.8 Encryption Password:

```
package com.secretword;


public class AppCache {

    public static final int INVALID_PIN_LIMIT=5;

    public static SecretWordUser currentUser=null;
    public static int invalidPinCount=0;

    public static String passwordToEncrypt="jutenahmedauthenticate";

}
```

### B.1.9 Store to Local Device (local storage):

```
package com.secretword;

import android.content.Context;
import android.content.SharedPreferences;
import android.preference.PreferenceManager;
import android.util.Log;
```

```
public class LocalStorage {

   private static final String TAG=LocalStorage.class.getName();

   private SharedPreferences prefs;
   private Context;

    //construct local storage
   public LocalStorage(Context ctx){
        //get preferences to store
      prefs=PreferenceManager.getDefaultSharedPreferences(ctx);
      Log.d(TAG, "Got prefs");
        //set context
      context=ctx;
   }

    //clear all the current local storage data
   public void clear(){
        //get editor
      SharedPreferences.Editor et=prefs.edit();
        //clear
      et.clear();
        //commit
      et.commit();
   }

    //save boolean value
   private void save(String key,boolean value){
      SharedPreferences.Editor et=prefs.edit();
      et.putBoolean(key, value);
      et.commit();
   }

    //restore boolean value using default value
   private boolean restore(String key,boolean defVal){
      return prefs.getBoolean(key,defVal);
   }

    //save string value
   private void save(String key,String value){
      SharedPreferences.Editor et=prefs.edit();
      et.putString(key, value);
      et.commit();
   }

    //restore string value with default value
   private String restore(String key,String defVal){
      return prefs.getString(key, defVal);
   }

    //save long value
   private void saveL(String key,long value){
      SharedPreferences.Editor et=prefs.edit();
      et.putLong(key, value);
      et.commit();
   }

    //restore long value
   private long restoreL(String key,long defVal){
      return prefs.getLong(key, defVal);
   }

    //store username in local storage
   public void setUsername(String uname){
      save("username", uname);
   }
```

```
 //retrieve username from local storage ...and null if its not present
public String getUsername(){
   return restore("username", null);
}

 //store password in local storage
public void setPassword(String uname){
   save("password",uname);
}

 //retrieve password from local storage ... and null if its not present
public String getPassword(){
   return restore("password",null);
}

 //store secret word in local storage
public void setWord(String uname){
   save("word",uname);
}

 //retrieve secret word from local storage .. and null if its not present
public String getWord(){
   return restore("word",null);
}

 //set locked and set timestamp to current
public void setLocked(){
   long ltstamp=new java.util.Date().getTime();
   saveL("locktime",ltstamp);
}

 //unlock user and set lock time to 0 which means not locked
 public void unlock(){
     saveL("locktime",0);
 }

//check if it is locked
public boolean isLocked(){
   //get time it was locked ... defaults to 0
   long lval=restoreL("locktime",0);
   //if its 0 then it means it is not locked so return false
   if (lval==0){
      return false;
   }
   //get current timestamp
   long ltstamp=new java.util.Date().getTime();
   //get difference from lock time and current timestamp
     long diff=ltstamp-lval;
   //convert it into seconds
     diff=diff/1000;
   //if seconds are less than 60 (1 minute) return its locked so true
     if (diff<60){
         return true;
     }
   //it's not locked so return false
     return false;
}

//get time remaining regarding unlock process
 public long getUnlockRemain(){
   //get time it was locked ... defaults to 0
     long lval=restoreL("locktime",0);
     if (lval==0){
         return 0;
     }
   //get current timestamp
     long ltstamp=new java.util.Date().getTime();
   //get difference from lock time and current timestamp
```

```
            long diff=ltstamp-lval;
          //convert it into seconds
            diff=diff/1000;
          //convert it into minutes
            diff=60-diff;
          //return remaining minutes
            return diff;
        }
}
```

### B.1.10 Login Page for user:

```
package com.secretword;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;

import com.jutenahmed.android.UIHelper;

public class LoginActivity extends CommonActivity {

    private static final String TAG=LoginActivity.class.getName();

    private EditText txtusername;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        txtusername=(EditText)findViewById(R.id.txtusername);
    }

    public void registerClick(View v){
        Log.d(TAG,"registerClick");
        //the user has asked to register which intents the user to open the screen
        Intent i=new Intent(this,SignupActivity.class);
        startActivity(i);
    }

    public void loginClick(View v){
        Log.d(TAG,"loginClick");
        String username=txtusername.getText().toString();
        //check username in localstorage if user has already created profile on
this app
        LocalStorage ls=new LocalStorage(getApplicationContext());
        String susername=ls.getUsername();
        //if the username is not in the storage it means user has not created
account yet
        if (susername==null){
            UIHelper.msbox("Error","Invalid username",LoginActivity.this);
            return;
        }

        //check if account is locked
        if (ls.isLocked()){
            //get number of seconds remaining in unlock period
            long remain=ls.getUnlockRemain();
            //if seconds are greater than 0 show this message
            if (remain>0) {
```

```
                UIHelper.msbox("Error", "Your account is locked.You can login after
"+String.valueOf(remain)+" seconds", LoginActivity.this);
            }
            return;
        }

        try {
            //if entered username matches the username in the storage than its
correct login
            if (susername.compareTo(username) == 0) {
                //create user for cache in app to be accessed by different screens
                SecretWordUser swu = new SecretWordUser();
                //set username
                swu.username = ls.getUsername();
                //set (memorable) word
                swu.secretWord = AESEncryption.decrypt(AppCache.passwordToEncrypt,
ls.getWord());
                //set PIN
                swu.pin =
AESEncryption.decrypt(AppCache.passwordToEncrypt,ls.getPassword());
                AppCache.currentUser = swu;
                //reset invalid pin counter as its a new login
                AppCache.invalidPinCount = 0;
                //open screen to enter memorable word
                Intent i = new Intent(this, SecretWordActivity.class);
                startActivity(i);
                finish();
            }
        }catch(Exception ex){
            UIHelper.msbox("Error","Exception:
"+ex.getMessage(),LoginActivity.this);
        }
    }
}
```

### B.1.11 Memorable Word Page:

```
package com.secretword;

import android.content.Intent;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.view.View;
import android.widget.EditText;

import com.jutenahmed.android.UIHelper;

import java.util.Random;

public class SecretWordActivity extends CommonActivity {

    private static final String TAG=SecretWordActivity.class.getName();
    private Random rnd=new Random();
    private int random1Index,random2Index;

    private EditText txtrandom1,txtrandom2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_secret_word);
        txtrandom1=(EditText)findViewById(R.id.txtrandom1); //get box for random
character1
        txtrandom2=(EditText)findViewById(R.id.txtrandom2); //get box for random
character2
        this.createRandomIndexes();

        txtrandom1.addTextChangedListener(new TextWatcher(){
            @Override
            public void beforeTextChanged(CharSequence, int i, int i1, int i2) {
            }

            @Override
            public void onTextChanged(CharSequence charSequence, int i, int i1, int
i2) {
            }

            @Override
            public void afterTextChanged(Editable editable) {
                Log.d(TAG,"Editable: "+editable.toString());
                if (editable.toString().length()>=1){
                    txtrandom2.requestFocus();
                }
            }
        });
    }

    private void createRandomIndexes(){
        int randomMax=AppCache.currentUser.secretWord.length(); //get length of
memorable word to determine the max size of random
        random1Index=rnd.nextInt(randomMax); //get first random letter
        random2Index=rnd.nextInt(randomMax); //get second random letter
        while(random2Index==random1Index){ //until both are unique
            random2Index=rnd.nextInt(randomMax);
        }
        //set letters in textboxes
        txtrandom1.setHint("Letter "+String.valueOf(random1Index+1));
        txtrandom2.setHint("Letter "+String.valueOf(random2Index+1));
    }

    public void loginClick(View v){
        Log.d(TAG, "loginClick");

        Log.d(TAG,"Char1: "+AppCache.currentUser.secretWord.charAt(random1Index));
        Log.d(TAG,"Char2: "+AppCache.currentUser.secretWord.charAt(random2Index));

        //if both characters entered match
        if (txtrandom1.getText().toString().length()==1 &&
txtrandom2.getText().toString().length()==1){
            if
(txtrandom1.getText().toString().charAt(0)==AppCache.currentUser.secretWord.charAt(
random1Index)
                    &&
txtrandom2.getText().toString().charAt(0)==AppCache.currentUser.secretWord.charAt(r
andom2Index)){

                //start pin entry screen
                Intent i=new Intent(SecretWordActivity.this,PinActivity.class);
                startActivity(i);
                finish();
                return;
            }else {
                UIHelper.msbox("Error", "Invalid", SecretWordActivity.this);
                return;
            }
        }else{
            UIHelper.msbox("Error","Invalid",SecretWordActivity.this);
            return;
```

```
            }

        }
}
```

# **Appendix C** – Comparison of Work to Previous Published Work

## C.1 Patent Application No: 0210322.4

(12) UK Patent Application (19) GB (11) 2 388 229 (13) A

(43) Date of A Publication 05.11.2003

(21) Application No: 0210322.4

(22) Date of Filing: 04.05.2002

(71) Applicant(s):
**Robert MacAlonan**
**101 Brompton Farm Road, Strood,**
**ROCHESTER, Kent, ME21 3RF,**
**United Kingdom**

(72) Inventor(s):
**Robert MacAlonan**

(74) Agent and/or Address for Service:
**Raworth Moss & Cook**
**Raworth House, 36 Sydenham Road,**
**CROYDON, Surrey, CR0 2EF,**
**United Kingdom**

(51) INT CL[7]:
**G07C 9/00 , G07F 7/10 , H03M 11/00**

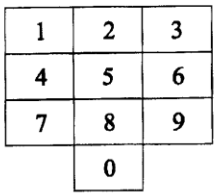
(52) UK CL (Edition V ):
**G4H** HKK HTG H1A H14A H2L H2P H2T

(56) Documents Cited:
**EP 1047837 A2** **WO 1998/027518 A1**
**US 4502048 A** **US 4333090 A**

(58) Field of Search:
UK CL (Edition T ) **G4H**
**Other**
**Online: WPI, EPODOC, PAJ**

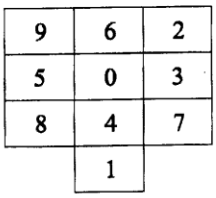
(54) Abstract Title: **Keypad for generating code with scrambled displayed key sequence**

(57) A keypad or keyboard (20, Fig. 4) is used to enter a code to enable an action to be effected, e.g. a cash withdrawal from an ATM or unlocking of a door locked by a security device. Each key contains a 7-segment type LED displaying a number, each driven by a multiplexer (22, Fig. 4) which receives signals from an EPROM (23, Fig. 4) in which random numbers are stored in the form of Data Blocks. Initially the LEDs show a conventional key layout, Fig. 6. In an ATM embodiment, after the user has entered a valid card, the key sequence displayed on the keypad is randomised, Fig. 7. After the user has successfully entered their PIN, the displayed key sequence can be returned to the conventional layout, Fig. 8, to facilitate keying for cash withdrawal. It is thus made more difficult for an unscrupulous person to observe and remember a code entered via the keypad.



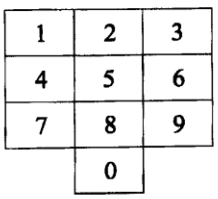

conventional Layout
(Before Card Inserted)

FIG. 6




Keyboard Randomised
(Ready for PIN Code Entry)
(Time-Out Enabled)

FIG. 7



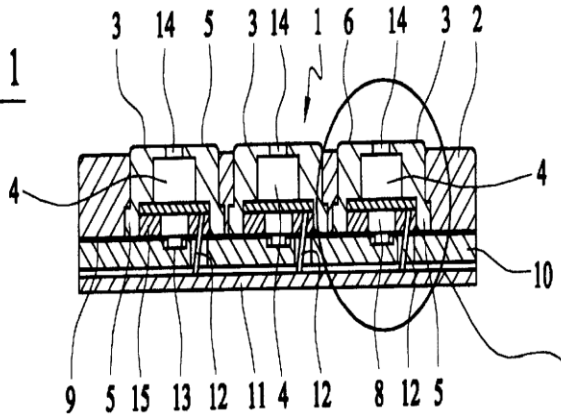

Conventional Layout
(After Pin Code Accepted)
(Time-Out Disabled)

FIG. 8

GB 2 388 229 A

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

FIG. 1



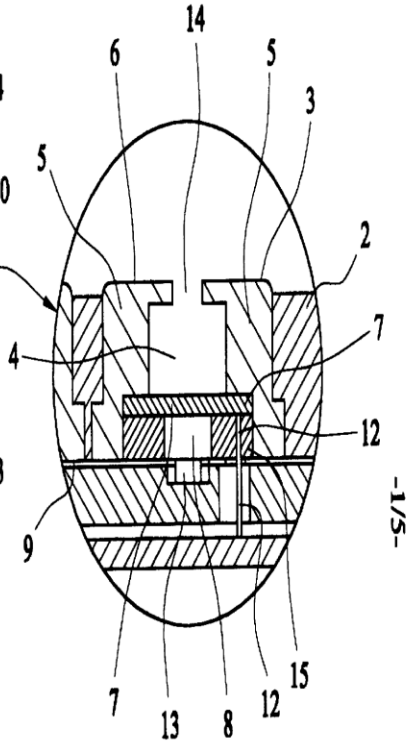




FIG. 2

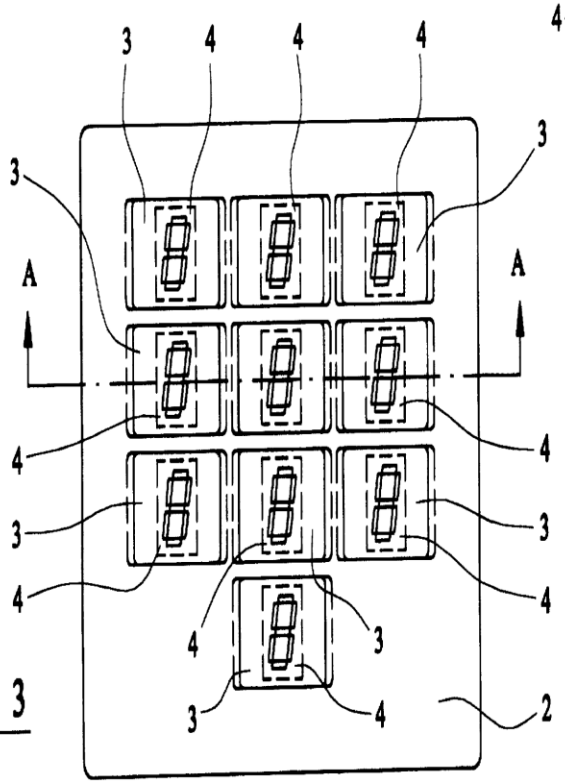



FIG. 3

-1/5-

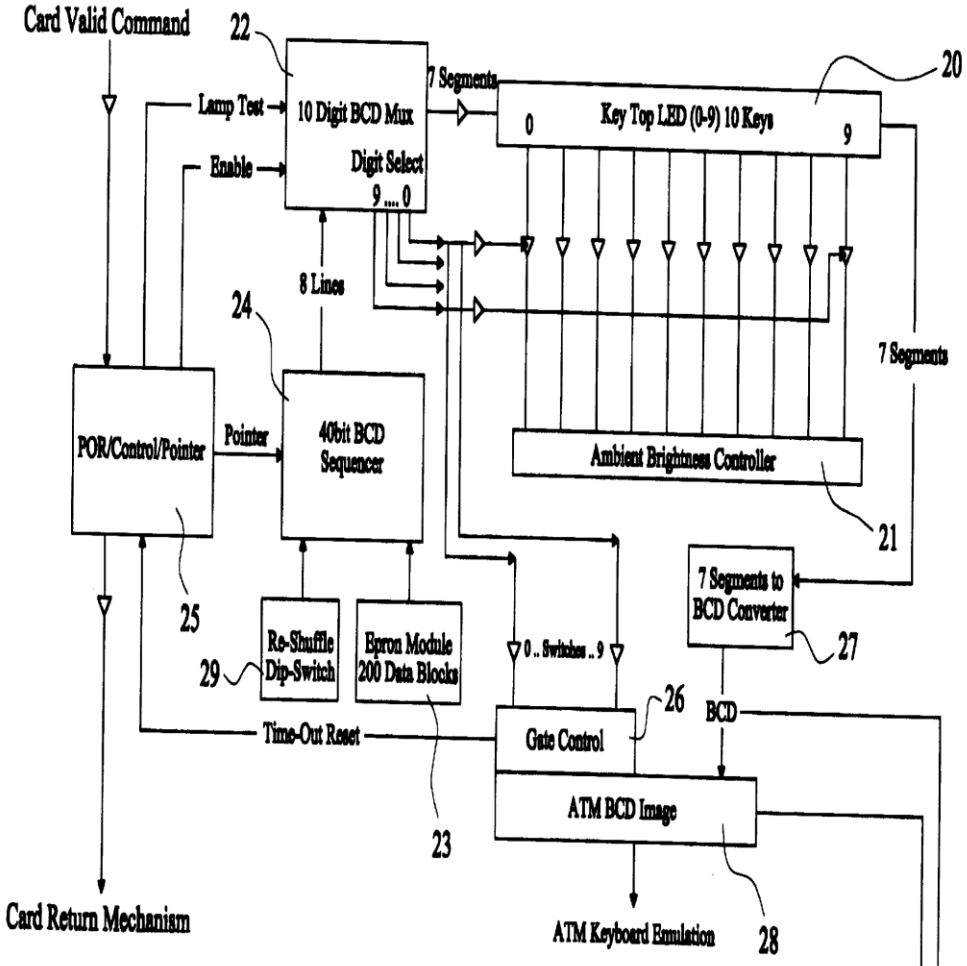



FIG. 4

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Numeric Display | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
| ASCII Format (Hex) | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | After Image Conversion |
| Binary/BCD Codes | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | Before Image Conversion |

Example of Binary Code for Hex Value 36 representing Numeric Digit 6

| 3 | 6 | Hex Code |
|---|---|---|
| 0011 | 0110 | Binary Code |

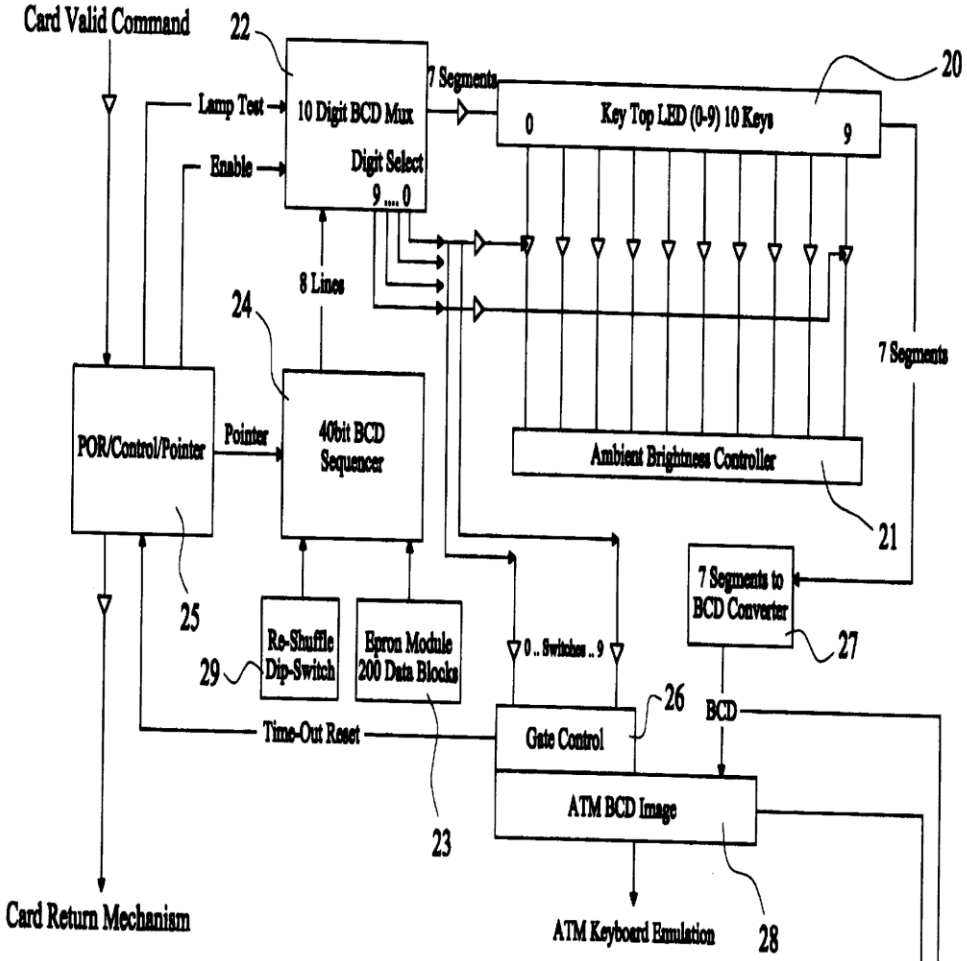


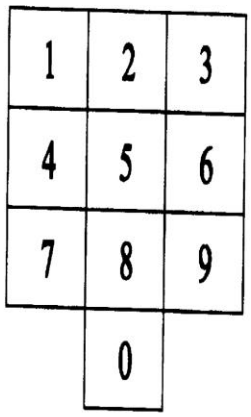

FIG. 4

| Numeric Display | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ASCII Format (Hex) | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | After Image Conversion |
| Binary/BCD Codes | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | Before Image Conversion |

Example of Binary Code for Hex Value 36 representing Numeric Digit 6

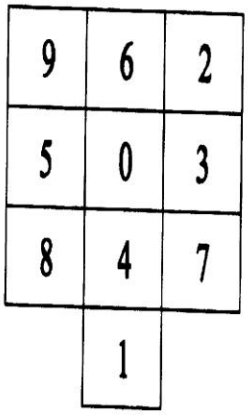
| 3 | 6 | Hex Code |
|---|---|---|
| 0011 | 0110 | Binary Code |

-2/5-



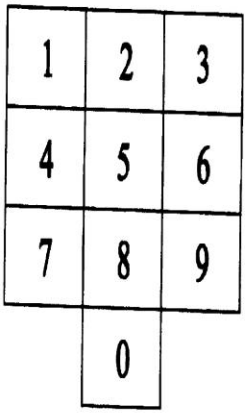

conventional Layout
(Before Card Inserted)

FIG. 6




Keyboard Randomised
(Ready for PIN Code Entry)
(Time-Out Enabled)

FIG. 7




Conventional Layout
(After Pin Code Accepted)
(Time-Out Disabled)

FIG. 8

-4/5-

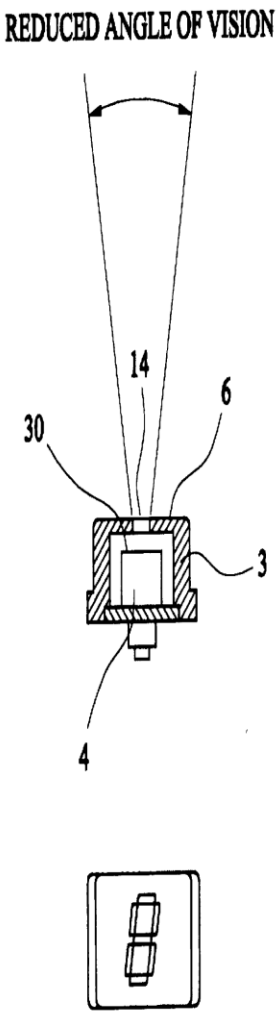



FIG. 9

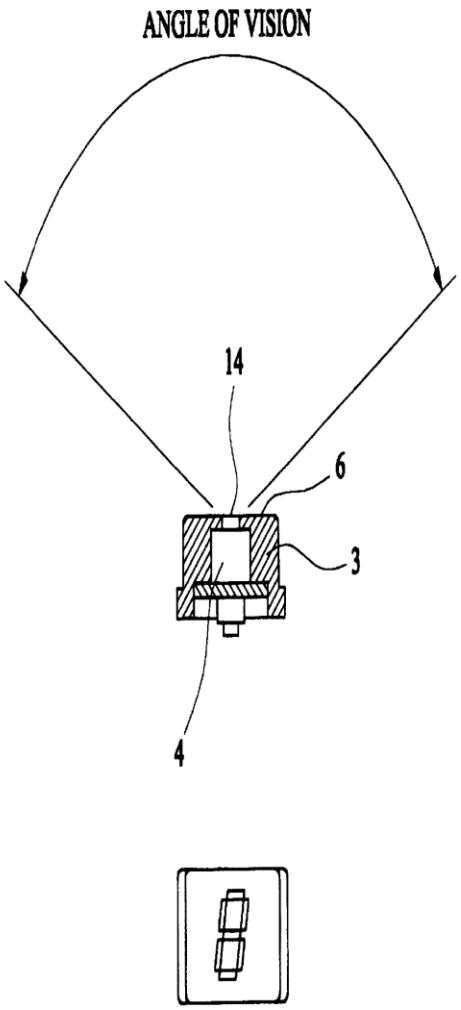



FIG. 10

2388229

# IMPROVEMENTS IN GENERATING A CODE

The present invention relates to improvements in generating a code for permitting an action to be effected particularly but not necessarily exclusively a cash-withdrawal from a cash withdrawal or automatic telling machine (ATM) or device attached to for instance a bank or building society but the invention also relates for example to the unlocking of a door locked by a security device.

Such actions are usually implemented with a keypad comprising a plurality of keys or push-buttons displaying symbols which may be numbers, letters or other symbols or combinations of same (alpha-numeric). The most usual combination is of course the numbers 0 to 9. The keys which may be of the mechanical button or touch type are depressed in a predefined sequence forming a code usually of four or more numbers e.g. a banking personal identification number (PIN) to implement a specific action such as withdrawing cash or unlocking a door to permit entry. The term "Keypad" includes other types of key operated devices such as keyboards. ATMs also normally include an "ENTER" button to enable the transaction, but also "CANCEL" and "CLEAR" buttons for known purposes.

For ATMs of course, before the PIN can be entered a card must be inserted into the machine and validated, the card bearing the user's details such as name, bank account number etc. Once the card has been inserted and validated the PIN can be entered by pressing the appropriate numerical keys and then the "Enter" key and after the PIN has been verified the amount of cash to be

withdrawn can be entered and withdrawn. Finally the card is returned to the customer.

It is quite easy for an unscrupulous person to loiter and watch ATM customers or other persons enter their PINs by noting the position of the keys pressed on the keypad and then remembering the sequence of numbers. This is particularly true of ATMs located in the street (so called Hole in the Wall machines). Customers may then be mugged or pick-pocketed after leaving the machine so as to steal their cards and these may be used together with the unauthorised PIN to withdraw cash illegally. This will continue until either the card holder informs his/her respective bank or building society or the customer's account is emptied rendering further transactions void.

In shops and stores at present in the U.K. the purchase of goods by credit or debit cards requires a verified signature from the purchaser thus making fraudulent activity more difficult. In mainland Europe, however, such transactions are conducted solely by a PIN entered by the customer from a keypad and it is planned to introduce this system into the U.K. soon. This will inevitably lead to a greater incidence of fraud.

It is therefore an object of this invention to make it more difficult for an unscrupulous person to observe and remember a code entered by way of a keypad or other key-bearing device.

According to one aspect of the present invention, there is provided a method for generating a code for use in permitting an action to be effected, the method

comprising actuating selected keys from a sequence of keys contained in a key-pad to generate and enter the code and changing the sequence of keys.

In one embodiment of the invention, the sequence of keys is changed after the code has been generated and entered and before a further code is generated.

Preferably each key displays a symbol so that a sequence of symbols is formed by the keys and the sequence of keys is changed by changing the sequence of the symbols displayed by the keys.

Suitably each symbol is formed by light emitting diodes (LEDs).

Conveniently the code is a plurality of numbers.

Preferably the symbols are alpha-numeric.

Suitably the sequence is generated by a random number generator.

Conveniently the random number generator includes an EPROM.

Preferably the action to be effected is the withdrawal of cash from an automatic telling machine (ATM).

Suitably before the code can be generated a card bearing relevant information is inserted into the ATM and the information validated.

Conveniently the sequence of symbols is changed upon insertion of the card into the ATM.

Preferably the sequence of symbols is changed after the code has been entered and validated.

Suitably the sequence of symbols always reverts to a standard format when the code has been entered and validated to assist in enabling the required amount of cash to be entered.

Conveniently each LED is of the 7 segment type.

According to another aspect of the invention apparatus is provided for generating a code for use in permitting an action to be effected, the apparatus comprising a key-pad having a key sequence in which selected keys may be actuated to generate and enter the code and means is provided to change the key sequence.

In one embodiment of the invention, the means changes the key sequence after the code has been generated and before a further code is generated.

Preferably each key bears a symbol so that a sequence of symbols is formed by the keys and the means changes the sequence of symbols displayed by the keys.

Suitably each key is provided with light emitting diodes (LED) which forms the symbol.

Conveniently the code comprises a plurality of numbers.

Preferably the symbols are alpha-numeric.

Suitably the means is a random number generator.

Conveniently the random number generator includes an EPROM.

Preferably the EPROM provides random signals representative of particular numbers and a multiplexer is provided to supply each signal individually to each key in turn whereby to alter the number if the number represented by the signal is different to the number displayed by the key.

Suitably the action to be effected is the withdrawal of cash from an automatic telling machine (ATM).

Conveniently means are provided to permit the insertion and validation of a card bearing relevant information before the code can be generated.

Preferably after validation of the card the means is enabled to change the symbol sequence.

Suitably means are provided to validate the code and after validation of the code the means for changing the key sequence is enabled to change the sequence of keys on the key-pad to a standard format to assist in entering the appropriate amount of cash.

Conveniently each LED is of the 7 segment type.

According to a further aspect of the invention, there is provided a key-pad in which each key displays a symbol and on actuation of any key a signal is generated representative of a particular symbol displayed by the key and each key is also adapted to receive a signal representative of a symbol which may or may not be different from the symbol currently being displayed by the key and if the symbol is different to the symbol currently being displayed the signal being received changes the symbol being displayed to that represented by the signal being received.

Preferably each symbol is displayed by a light emitting diode (LED) incorporated in the key.

In one embodiment of the invention each key has a top wall and each LED has an upper surface which is spaced from the top wall.

Suitably the LEDs are each adapted for connection to a random number generator.

Conveniently the LEDs are connected to a circuit board.

Preferably the circuit board is connected to the random number generator.

Suitably the random number generator includes an EPROM.

Conveniently a multiplexer is connected to the circuit board and the EPROM is connected to the multiplexer to supply signals representative of random

numbers thereto whereby the multiplexer can supply each signal individually to the circuit board for location at each LED.

An embodiment of the present invention will now be particularly described with reference to the accompanying drawings in which:-

Figure 1 is a transverse cross-section along lines A-A of Figure 3 of a key-pad intended for an ATM;

Figure 2 is an amplified view of a particular key or push-button shown in Figure 1;

Figure 3 is a plan view of the key-pad shown in Figure 1;

Figure 4 is a block circuit diagram of the various components of the invention relating to an ATM;

Figure 5 is a flow sheet relating to the processes involved in the operation of the ATM;

Figures 6 to 8 show schematically possible numerical keyboard displays in use;

Figure 9 is a modified version of the key shown in Figure 1 and Figure 10 shows the Figure 1 version for comparison.

Referring to Figures 1 to 3, the key-pad 1 comprises an alloy frame 2 having a number of keys or buttons 3 incorporating a conventional 7-segment light-emitting-diode LED 4 forming an LED assembly of 10 LEDs as shown

in Figure 3. In this case, each LED 4 is capable of forming one of the numbers 0 to 9. The keys 3 are of conventional hollow shape with side walls 5 and top walls 6. The underside of the frame 2 is machined out to permit the introduction thereinto of each LED 4 into the key or button 3 and each LED 4 is mounted on a printed circuit board (PCB) 7 (Figure 2) with a microswitch 8 connected to each LED, each microswitch 8 being mounted on the underside of the PCB 7 beneath its respective LED 4.

A silicon rubber membrane 9 is attached to the base of the frame 2 to improve the spring action of the keys 3. The membrane 9 is sandwiched between the frame 2, keys 3 and a stainless steel striker plate 10. Beneath the striker plate 10 is mounted an electronic interface control board 11. Ten fly-lead signal cables 12, one for each LED 4 are connected at one end to the control board 11 and at the other end to the PCB 7, each cable 12 comprising a 10 core signal cable with 7 segment lines, a common bus and two switch lines.

The microswitch 8 has actuators 13, which are proud of the frame 2, the actuators 13 being located in recesses in the striker plate 1. Each LED 4 is visible via a slot 14 in the top wall 6 of the key 3 while the circuit board 7 rests on a pad 15.

The striker plate 10 and the membrane 9 have suitable apertures to permit the cables 12 to be connected to the boards.

The frame 2 locates all the assemblies into a uniform pitch. The membrane 9 and the striker plate 10

hold each switch 8 under compression and actuation of each switch 8 is achieved by pushing down on the top 6 of the key or button 3 causing the rubber membrane 9 to displace by a small amount e.g. 0.4 mm and the striker plate 10 to be actuated. The small movement of the membrane 9 provides sufficient movement (approximately 0.2 mm) for the switch to be closed. The switch 8 (or each switch 8) when closed sends a signal to the PCB 7 representative of the symbol e.g. the number to be displayed by the particular LED. The PCB 7 then supplies this as part of a code via the control board 11 to a code validation means (not shown) and an actuation device (not shown) to effect the action. These latter two items will not be described in detail as they are well known in the art.

While not shown the key-pad 1 described may have additional keys as conventional such as "Enter", "Clear", "Delete" or "Cancel" but these and their function are not shown or described as also being well known in the art.

The board 11 receives signals from a multiplex and EPROM (shown in Figure 4) serving as a random number generator. These signals find their way via the cables 12 and the PCB 7 to the LEDs 4 where they can alter the number currently displayed by the LED to another number if the signal represents a different number.

Referring to Figure 4 a code generator is shown comprising a key-pad or keyboard 20 of the type previously described having ten keys numbered 0 to 9. Each key contains an LED displaying a number and each LED comprises a conventional 7-segment diode as previously shown in Figure 3 from which any of the numbers 0 to 9

can be formed. The brightness of all LEDs may be varied by an ambient brightness control 21 which is connected to all LEDs and comprises a photo-transistor which varies the current through the LEDs proportionately to the level of ambient light.

Each LED is driven by a two by five (10) Digit BCD (Binary Coded Decimal) Multiplex Chip (Mux)22. This greatly reduces the amount of wiring needed to connect the Mux 22 to the board 11 as only 7 segment lines and 10 address lines (one line per digit)are required. The Mux 22 is wired to the board 11 and receives signals from a Erasable Programmable Read Only Memory (EPROM) 23 in which the random numbers are stored in the form of Data Blocks.

A typical EPROM can be selected to hold from 100 to 500,000 number sequences but in this case only 200 number sequences (2000 numbers) are held as an example. The number sequences (in blocks of ten numbers) as supplied by the EPROM 23 are passed as 40 bit BCD codes to a 40 bit BCD Sequencer/Memory block 24 for transfer to the Mux/display driver 22 in 8 bit blocks. The sequencer 24 reads each address in the EPROM 23 and transfers any data stored in the address into the LED display registers 20 via Mux 22.

The particular numeric combinations are held in addresses in the EPROM. There can be hundreds of these addresses and therefore hundreds of numerical combinations and sequences are available for the key-pad. The data i.e. unique numerical combinations or sequences are each uniquely located in one of these addresses. The address 000 is unique inasmuch as it holds the standard

or conventional key-pad layout for the ATM manufacturer as shown in figure 6.

The invention will from hereon in be described with reference to the operation of an ATM but it will be appreciated that it would be equally applicable to other systems where the generation of a personal or security code can be seen by a third party e.g. a code operated security lock on a door.

On power up of the system Pointer 25 is set to address 000 so that the number sequence corresponding to the standard keyboard layout is generated by the EPROM 23 and supplied to the LEDs 20 by the Mux 22. The customer then inserts their card into the machine and this is examined for validity. If the card is valid a "Card Valid Command" is issued, the card is "swiped" i.e. accepted and temporarily retained by the system and a signal is provided to the Pointer 25 which is incremented by +1 and stored in memory to locate a new address (001) in the EPROM 23 for the next and new sequence of numeric keys which after supply to the Mux 22 are after a brief delay written to the 7 segment LEDs in a random order. The keyboard might then look something like Figure 7 with the numbering sequence of the keys now being quite different to that of the standard keyboard in Figure 6.

The code is now entered and validated. If the code is validated the keyboard displays the standard format of Figure 6 or Figure 8 to permit the key-pad to be read more easily and therefore the actual amount of cash to be entered more accurately. If the code is not validated the card is returned for another attempt after which the card would as conventional be confiscated by the machine.

After entering the amount of cash, this is supplied to the customer and the card is returned.

On insertion of another card if the card is valid a further "Card Valid Command" is issued, the card is swiped and a further increment signal is provided to the Pointer 25 which is incremented to +2 to locate a new address (002) in the EPROM 23 for the next or new sequence of numeric keys which after supply to the Mux 22 are written to the LEDs. The cash withdrawal process is now repeated.

A Time-out (provider set) sequence is invoked after the card has been swiped and a key has been depressed. This is to reduce viewing time of the keyboard sequence and will change the keyboard layout back to the standard (Figure 6) appearance and the card will be ejected. A gate control 26 must be enabled within a pre-set period by signals from the Mux 22.

Every depression of a key is output as illuminated segments forming a key image in the form of a number. This image is converted via a 7 segment LED to BCD converter 27 to the original BCD code. The Numeric Data displayed relating to the key pressed via the MUX is gated to enable the BCD code to be transmitted to the converter 28. It is also displayed on the keyboard as an ATM BCD image 28 before conversion.

If an incorrect PIN code is entered then dependent on a Re-Shuffle Dip switch 29 setting one of the following actions occurs: Dip switch off - the Random Number Code remains the same. The user re-enters the pin

code. Dip switch on - the Random Number Code changes. The user re-enters the pin code.

A persistent incorrect entry of the pin code causes an ATM violation which is handled by the ATM provider.

The generator shown in and described with respect to Figure 4 utilises 74 series TTL logic devices and C-MOS (Complimentary Metal Oxide Semiconductor) logic devices.

Figure 5 shows the steps involved in the process of cash withdrawal using the present invention. It is believed that this is straight-forward and self-explanatory which in effect repeats the processes described with reference to Figure 4. However Figure 5 anticipates that a microprocessor driven version of the device would be produced operated by software in accordance with the steps of Figure 5.

While the key-pad shown in Figures 1 to 3 is of the mechanical push-button type it will be appreciated that the keys could be of the pressure sensitive touch-pad type or even of the touch-screen type.

While not shown before a number from one key is read on being depressed a key release signal must be detected as is conventional.

The keyboard may be fitted with a guard or shield round it to narrow the angle of visibility to a bystander to prevent overlooking and thereby further inhibit fraudulent observation of the PIN.

Some of the keys could be redundant i.e. not provide a signal but carry changeable symbols which cannot produce a signal.

While not described, the function keys "enter", "cancel", "clear" could also move around in the sequence of numbers on the key-pad.

Furthermore the sequence of numbers could automatically be changed by the pulse of a clock or oscillator instead of in response to the generation of a code.

Seven segment LED displays are primarily used to display Numeric Symbols in the range 0-9. Whilst they can be used to display some Alphabetic characters e.g. A, B, C etc. they are not ideally suited to the latter simply because of confusion within the limited character set. For example, the numeric character '6' can be mis-read as the letter "b".

A typical system example developed by MACOL was purely intended for emulation of an ATM which requires Numeric input only. For other applications such as security systems and encryption devices requiring a randomised complete Alpha-Numeric character set, Dot-Matrix and Starburst display systems can be utilised. The most practical device would be the Dot Matrix display.

Dot Matrix displays enable characters to be formed by illumination of individual LEDs, similar to newsprint, the symbol or character is formed by a series of dots within a matrix. In newsprint, this is the number of

dots per $cm^2$. The more dots available the better the resolution or clarity of the picture. As the resolution increases so the dot diameter and pitch reduce. In our application, the dot matrix is formed by illuminating individual LEDs to create the desired shape. A standard Dot Matrix display comprises 35 dots arranged in a 5 x 7 format. Higher resolution displays are available. The hardware for driving such displays would be more complex than the 7-segment type.

Referring to Figure 9 the Key 3 is similar in structure to that shown in Figures 1 and 2 except the Key 3 has a top wall 6 and the LED 4 has an upper surface 30 which is spaced from the top wall 6. In Figure 10 it will be seen that by comparison the upper surface 30 of the LED 4 is hard against the underside of the key top wall (fascia) 6. By distancing the upper surface 30 of the LED 4 from the top wall 6 of the Key 4 and therefore from the slot 14 through which the LED 4 is visible, the angle of vision of the LED 4 to an observer is reduced making it very difficult if not impossible for an unauthorised observer to discern the key numbers. On the other hand if the user is standing immediately above the Key 4 they will see the Key 4 as clearly as they would see the Key 4 in the Figure 10 version. This modification therefore further reduces the ability of an unauthorised bystander to discern the key-pad numbers.

CLAIMS:

1. A method for generating a code for use in permitting an action to be effected, the method comprising actuating selected keys from a sequence of keys contained in a key-pad to generate and enter the code and changing the sequence of keys before a further code is generated.

2. A method as claimed in claim 1 in which the sequence of keys is changed after the code has been generated and entered.

3. A method as claimed in claim 1 or claim 2 in which each key displays a symbol so that a sequence of symbols is formed by the keys and the sequence of keys is changed by changing the sequence of the symbols displayed by the keys.

4. A method as claimed in claim 3 in which each symbol is formed by a light emitting diode (LED).

5. A method as claimed in any of the preceding claims in which the code is a plurality of numbers.

6. A method as claimed in claim 5 in which the symbols are alpha-numeric.

7. A method as claimed in claim 6 in which the sequence is generated by a random number generator.

8. A method as claimed in 7 in which the random number generator includes an EPROM.

9. A method as claimed in any of the preceding claims in which the action to be effected is the withdrawal of cash from an automatic telling machine (ATM).

10. A method as claimed in claim 9 in which before the code can be generated a card bearing relevant information is inserted into the ATM and the information is validated.

11. A method as claimed in claim 10 in which the sequence of symbols is changed upon insertion of the card into the ATM.

12. A method as claimed in any of claims 3 to 10 in which the sequence of symbols is changed after the code has been entered and validated.

13. A method as claimed in either of claims 10 or 11 in which the sequence of symbols always reverts to a standard format when the code has been entered and validated to assist in enabling the required amount of cash to be entered.

14. A method as claimed in any of claims 4 to 13 in which each LED is of the 7 segment type.

15. Apparatus for generating a code for use in permitting an action to be effected, the apparatus comprising a key-pad having a key sequence in which selected keys may be actuated to generate and enter the code and means is provided to change the key sequence.

16. Apparatus as claimed in claim 15 in which the means changes the key sequence after the code has been generated and before a further code is generated.

17. Apparatus as claimed in claim 14 or claim 16 in which each key bears a symbol so that a sequence of symbols is formed by the keys and the means changes the sequence of symbols displayed by the keys.

18. Apparatus as claimed in claim 17 in which each key is provided with a light emitting diode (LED) which forms the symbol.

19. Apparatus as claimed in any of claims 15 to 18 in which the code comprises a plurality of numbers.

20. Apparatus as claimed in claim 19 in which the symbols are alpha-numeric.

21. Apparatus as claimed in any of claims 17 to 20 in which the means is a random number generator.

22. Apparatus as claimed in claim 21 in which the random number generator includes an EPROM.

23. Apparatus as claimed in claim 22 in which the EPROM provides random signals representative of particular numbers and a multiplexer is provided to supply each signal individually to each key in turn whereby to alter the number if the number represented by the signal is different to the number displayed by the key.

24. Apparatus as claimed in any of claims 17 to 23 in which the action to be effected is the withdrawal of cash from an automatic telling machine (ATM).

25. Apparatus as claimed in any of claims 17 to 24 in which means are provided to permit the insertion and validation of a card bearing relevant information before the code can be generated.

26. Apparatus as claimed in claim 25 in which after validation of the card the means is enabled to change the symbol sequence.

27. Apparatus as claimed in claim 26 in which means are provided to validate the code and after validation of the code the means is enabled to change the sequence to a standard format to assist in entering the appropriate amount of cash.

28. Apparatus as claimed in any of claims 23 to 27 in which each LED is of the 7 segment type.

29. A key-pad in which each key displays a symbol and on actuation of any key a signal is generated representative of a particular symbol displayed by the key and each key is also adapted to receive a signal representative of a symbol which may or may not be different from the symbol currently being displayed by the key and if the symbol is different to the symbol currently being displayed the signal being received changes the symbol being displayed to that represented by the signal being received.

30. A key-pad as claimed in claim 29 in which each symbol is displayed by a light emitting diode (LED) incorporated in each key.

31. A key-pad as claimed in claim 30 in which each key has a top wall and each LED has an upper surface which is spaced from the top wall.

32. A key-pad as claimed in claim 31 in which the LEDs are each adapted for connection to a random number generator.

33. A key-pad as claimed in claim 32 in which the LEDs are connected to a circuit board.

34. A key-pad as claimed in claim 33 in which the board is connected to the random number generator.

35. A key-pad as claimed in any of claims 32 to 34 in which the random number generator is an EPROM.

36. A key-pad as claimed in claim 35 in which a multiplexer is connected to the circuit board and the EPROM is connected to the multiplexer to supply signals representative of random numbers thereto whereby the multiplexer can supply each signal individually.

37. A method for generating a code substantially as hereinbefore described with reference to the drawings.

38. Apparatus for generating a code substantially as hereinbefore described with reference to the drawings.

39. A keypad substantially as hereinbefore described with reference to the drawings.


22


**Application No:** GB 0210322.4 **Examiner:** Melanie Gee
**Claims searched:** 1-39 **Date of search:** 16 December 2002

## Patents Act 1977 : Search Report under Section 17

**Documents considered to be relevant:**

| Category | Relevant to claims | Identity of document and passage or figure of particular relevance | |
|---|---|---|---|
| X | 1-7, 9, 10, 14-21, 24, 25, 28-34 | US 4333090 | (HIRSCH), see col. 5 line 61 - col. 6 line 52, and col. 7 line 62 - col. 8 line 47. |
| X | 1-10, 14-25, 28-35 | US 4502048 | (REHM), see whole document. |
| X | 1-7, 9, 10, 12, 14-21, 24, 25, 28-30 | EP 0147837 A2 | (OMRON TATEISI ELECTRONICS CO), see page 10 paragraph 2 - page 23 paragraph 1, and Fig. 5. |
| X | 1, 3, 5-7, 9-11, 15, 17, 19-21, 24-26, 29 | WO 98/27518 A1 | (SIAB ITALIA), see especially page 7 line 19 - page 8 line 11. |

Categories:

| | | | |
|---|---|---|---|
| X | Document indicating lack of novelty or inventive step | A | Document indicating technological background and/or state of the art. |
| Y | Document indicating lack of inventive step if combined with one or more other documents of same category. | P | Document published on or after the declared priority date but before the filing date of this invention. |
| & | Member of the same patent family | E | Patent document published on or after, but with priority date earlier than, the filing date of this application. |

**Field of Search:**

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC[T]:

G4H

Worldwide search of patent documents classified in the following areas of the IPC[7]:

The following online and other databases have been used in the preparation of this search report:

WPI, EPODOC, PAJ

## C.2 Patent Application No: 0623944.6

(12) UK Patent Application (19) GB (11) 2 444 285 (13) A

(43) Date of A Publication 04.06.2008

(21) Application No: 0623944.6

(22) Date of Filing: 30.11.2006

(71) Applicant(s):
**Tim Watson**
**18 Copthorn Gardens, HORNCHURCH, Essex, RH11 3DC, United Kingdom**

**Vicky Harris**
**137 Birdwell Drive, Great Sarkey, WARRINGTON, WA5 1XD, United Kingdom**

**Gavin Harper**
**46 Fairfield Avenue, UPMINSTER, Essex, RM14 3AY, United Kingdom**

(72) Inventor(s):
**Tim Watson**
**Vicky Harris**
**Gavin Harper**

(74) Agent and/or Address for Service:
**Tim Watson**
**18 Copthorn Gardens, HORNCHURCH, Essex, RH11 3DC, United Kingdom**

(51) INT CL:
***G06F 3/023*** (2006.01) ***G06F 21/00*** (2006.01)

(56) Documents Cited:
**GB 2402649 A** **EP 1280113 A2**
**WO 1993/011551 A1** **CA 002214190 A1**
**FR 002693815 A1** **US 6549194 B1**
**US 6434702 B1**

(58) Field of Search:
UK CL (Edition X ) **G4A**
INT CL **G06F**
Other: **Online: WPI, EPODOC**

(54) Abstract Title: **Keypad with random key layout**

(57) A keypad allows the physical arrangement of the keys to be rearranged by altering the legends on the keypad. This may be done either at the beginning of data entry or after every data item input. A film may be placed over the keys to restrict the angle at which the legends on the keys may be seen. The keypad may have push buttons with a display embedded in the button. The display may be an LCD. Alternatively, the keypad may be a touch screen. The keypad may also use coloured lights or other displays to provide information. The keypad may incorporate a microcontroller to control the display of the key legends or to decode key presses.

GB 2 444 285 A

**Original Printed on Recycled Paper**

2444285

**Random Generated PIN Keypad**

**Description**

Numeric keypads and Personal Identification (PIN) numbers are commonly used in the security industry to authenticate the identity of the user. These find many applications in security devices, and are commonly used in transactions as a "secure" form of verifying a users identity and credentials.

With the explosion of electronic based forms of commerce, PIN numbers are used more extensively than ever.

In a typical transaction, the user is required to provide a non-confidential user identifier or token – for example a credit card, and a confidential PIN to gain access to a system. The system compares the confidential PIN with the record stored for that identifier to authenticate the users identity.

A numeric keypad consists of an array of keys in a fixed order which allows numbers to be entered into a system. A numeric keypad commonly consists of the numbers 0-9 in a static array, and usually there are additional option keys such as "Enter" and "Clear" which are used to aid the data entry process.

Users are familiar with the arrangement of a 12-key numeric keypad. The numbers are arranged in a logical order in rows of three:

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |
| Enter | 0 | Clear |

Unfortunately, user familiarity with the layout of this keypad compromises its security if the user is under surveillance whilst entering a PIN number.

Even if a user tries to "shield" their PIN number from being seen, the fixed position of the keys means that an observer can judge the number being entered from positional data – seeing where the users finger is moving. It is therefore possible to deduce a PIN from the users approximate finger position, without being able to see the key legends clearly.

PINs are often 4-digit numbers in the range 0000-9999 resulting in 10,000 possible numbers, so that an attacker would need to guess an average of 5000 times to get the correct PIN. Many PIN verification systems allow three attempts, thereby giving a card thief a 1/3000 chance to guess the correct PIN before the PIN is blocked.

If the user has some prior knowledge of the user's finger movements, then this probability can be compromised significantly.

This invention consists of a dynamic keypad, where the legends of the keys are not static and fixed, but change places. This can occur after each and every key entry, or at the beginning of the data entry sequence.

The keypad can either be physical – implemented using individual buttons, or virtual, using a touch-screen type display input device.

By moving the key legends about using a random algorithm, it is possible to create higher security, as the users finger movements give little clue to the number being entered if some or all of the key's are obscured. At the moment, if a user presses the top left key of a numeric keypad, the probability of it being a 1 is 1/1. However, if the keys are randomised, then the probability of that finger movement indicating a 1 is only 1 in 12, 0.083 on each key entry occasion. Clearly security is improved if key legends denoting the position of a number cannot be seen, and even if some of the key legends can be seen, the remaining obscured legends provide some level of security.

Additionally, the legends could be shielded from unwanted overlookers by means of an optical film applied over the device which gives a restricted angle of view.

A specific implementation of this device is as follows:

A "Screen Key", is a momentary push-switch, with a programmable LCD display in place of a fixed legend. The display can be programmed to display any number, legend or character on a matrix of pixels. A keypad composed of "Screen Keys" is a keypad whereby an array of "Screen Keys" are linked together, with each having a separate programmable LCD legend. Additionally, an option available on a "Screen Key" is to have a backlight provided by means of Tri-Coloured Light Emitting Diode. The proportions of Red, Green and Blue can be adjusted to convey colour information to the user, providing additional information about the key's function.

A random number generator pad could comprise an array of 4x3 screen keys controlled by a microcontroller. The microcontroller receives key-press information from each key. The microcontroller can also write data to the LCD screens atop of the key. The microcontroller interfaces to the application device, for example, an ATM machine. This interface could take the form of a digital bus connection to the microcontroller, or to retrofit existing applications, the microcontroller could connect to the application using the interface of the static keypad it is replacing.

The software function of the microcontroller is as follows.

1. Create a lookup table of 12 key positions.
2. Randomly assign each of the "key legend values" to one of the positions in the lookup table.
3. Associate each physical key, with a corresponding position in the lookup tables.
4. Write the data corresponding to the "key legend values" to the LCD screen of each key, in accordance with the data in the lookup table.
5. Scan the keypad for user entry.
6. One user entry has been detected, lookup the position indicated by the physical key press in the lookup table and obtain a value for the corresponding "key legend value".
7. Output this key legend value to the application device.

The legends on the key could also be shielded from unwanted onlookers by means of an optical film over each key which gives a restricted angle of view.

**Claims**

1. The device comprises a keypad, where the physical arrangement of the key functions is rearranged by altering the legends on each key of the keypad electronically either:
   i) at the beginning of data entry
   ii) after every individual data item input

2. The device may also incorporate a film over each key which only allows the dynamically changeable legend to be viewed from a narrow field of view.

3. The device may achieve this function through the use of pushbuttons with display screens embedded within the device, allowing the key legend to be dynamically altered.

4. The device may also incorporate coloured lighting and or displays to provide additional information to the user.

5. The device may also encompass an interface which allows the "new keypad" to be retrofitted to older systems.

6. The device may contain an embedded microcontroller which provides interface functions to a system.

4

**Application No:** GB0623944.6 **Examiner:** Mark Simms

**Claims searched:** 1-6 **Date of search:** 12 March 2007

## Patents Act 1977: Search Report under Section 17

**Documents considered to be relevant:**

| Category | Relevant to claims | Identity of document and passage or figure of particular relevance |
|---|---|---|
| X,Y | X: 1,3,5,6; Y: 2 | FR 2693815 A1 (GEMPLUS) Whole document |
| X | 1,3,5,6 | WO 93/11551 A1 (MAURRAS) Whole document |
| X | 1,4,6 | US 6549194 B1 (MCINTYRE et al) Whole document |
| X | 1, 3 | GB 2402649 A (DAWSON) Whole document |
| X | 1,3 | US 6434702 B1 (MADDALOZZO et al) |
| Y | 2 | EP 1280113 A2 (BOSCH) Whole document |
| Y | 2 | CA 2214190 A1 (BLOM) Whole document |

Categories:

| | | | |
|---|---|---|---|
| X | Document indicating lack of novelty or inventive step | A | Document indicating technological background and/or state of the art |
| Y | Document indicating lack of inventive step if combined with one or more other documents of same category | P | Document published on or after the declared priority date but before the filing date of this invention |
| & | Member of the same patent family | E | Patent document published on or after, but with priority date earlier than, the filing date of this application. |

**Field of Search:**

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC[X] :

G4A

Worldwide search of patent documents classified in the following areas of the IPC

G06F

The following online and other databases have been used in the preparation of this search report

WPI, EPODOC

# **Appendix D** – Survey for Market Research

## JA Zamindar Authentication

**This survey is anonymous. You do not need to provide your personal details.**

**Please circle each question from 1 to 10 Please rate each question honestly.**

**Thank you.**

**Would you like to see this authentication process in the current market?**

1 2 3 4 5 6 7 8 9 10

**Unlikely** **No Opinion** **Likely**

**Do you believe this authentication process is better than your current banks authentication process?**

1 2 3 4 5 6 7 8 9 10

**No** **Slightly** **DEFINITELY!**

**Do you feel this authentication process is more secure?**

1 2 3 4 5 6 7 8 9 10

**No** **Slightly** **DEFINITELY!**

**How easy did you feel this two-step authentication process was to use?**

1 2 3 4 5 6 7 8 9 10

**Very Easy** **Average** **Very Difficult**

**Are there any improvements you would recommend?**

_____

_____

# ADDITIONAL SECURITY COUNTER MEASURES FOR MOBILE BANKING APPLICATIONS AND OTHER DEVICES

**STUDENT NAME: JUTEN AHMED** **STUDENT ID NO: 1306929**

**STUDENT SUPERVISOR: DR ALI MANSOUR** **COURSE: COMPUTER SECURITY AND FORENSICS**

## INTRODUCTION

Computer Security is a vital part of this day and age due to the vast amount of cyber-crime happening worldwide meaning cybercrime has been on an increase within the last 10 years (Wealth & Finance International, 2015). This thesis report has been created to establish the current market and a detailed report into the final artefact. The artefact is based on a new concept of authentication. Authenticating using a 2-step verification system. This can be implemented into banking applications to decrease the success rate of personal identification numbers being stolen.

## AIMS AND OBJECTIVES

### AIMS

The aims of this research are to create a secure authentication system of which banks can implement into their current applications. Furthermore, a new authentication process which increases the security of mobile banking application due to the fact there is very sensitive data within banking and personal banking. This new authentication process helps minimize the risk of shoulder surfers from gaining the PIN code.

### OBJECTIVES

- Conducting thorough research into the cyber security sector regarding authentication and what is currently on the market.
- To create an authentication process which would be very difficult to break using a 2-step authentication system and to create a successfully working prototype for demonstration.
- To create a fully working GUI which can demonstrate a login screen transitioning into a generic "bank account" screen which tells the customer the login was successful with a welcome message.
- Creating a randomized PIN keypad as a 3x4 matrix as one of the authentication steps and creating a memorable word system where the user inputs 1 character to access the second stage of the authentication process.
- To test all the validations of the prototype.

## METHODOLOGY USED

Within this project, the methodology used was RAD (Tutorials Point, 2015), Rapid Application Development and aspects of project management skills (PRINCE2) which included creating a Gantt Chart to help manage this project.

## TECHNICAL UTILITIES USED

Android Studios - To create an .apk file compatible for android devices for demonstration.
Java Programming Language - To code the application for interacting functions.
Adobe Photoshop CS6 - To create GUI's examples and Poster.

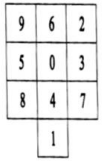
## PATENTS & THE CURRENT MARKET

There have been a few patents which have been approved. However, patents are all specified to design rather than the concept of randomization.

Currently in the market the banking industry are using PIN based authentication, memorable word based authentication and biometrics (fingerprint).

*Figure 1* - PIN authentication used by Barclays.
*Figure 2* - Concept design from parent application.


[*Figure 1*]




[*Figure 2*]

## INTRODUCING THE ARTEFACT

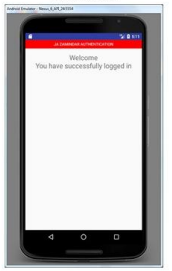
REGISTRATION

LOGIN

MEMORABLE WORD

RANDOMIZED PIN KEYPAD

SUCCESS



## CODING THE RANDOMIZED KEYPAD

```
//return true of its a blank spot else false
   private boolean showBlankKey(String str){
//10 and 11 are blank spots within app
   return (str.compareTo("10")==0 || str.compareTo("11")==0);
 //loop through 0 to 11 .... digits 10 and 11 are blank spots
     for (int a=0;a<=11;a++){
        //initialize random class to get always random digit
        Random rnd=new Random();
        //get random digit to display everytime
String digitToDisplay=String.valueOf(rnd.nextInt(12));
//check if its already displayed
while(isDigitDisplayed(digitToDisplay)){
   digitToDisplay=String.valueOf(rnd.nextInt(12));
}
//if digit is unique then add to collection
digitsDisplayed.add(digitToDisplay);
```

## COMPARISON OF WORK

Comparing the artefact to work published out there, overall the artefact is different to what is currently on the market, and any patents or journals published. This artefact does use similar concepts of authentication regarding published work in the current market however, this artefact is an improved concept which uses two factor authentication. Thus, increasing security and reducing the threat of shoulder surfers obtaining the users personal information or senstaive data.

## CONCLUSION

In conclusion, this artefact created does what it designed for and has been created to demonstrate the authentication process which can be implemented into the banking industry for mobile banking applications currently in use.

This artefact is a cross between authentication processes built into one application to allow the users to be able to gain a more secure, reliable and complex login process. Overall increasing customer's satisfaction due to the fact of having the knowledge of having more security generally means the less risk the client has on being a victim of any type of cybercrime. Using this artefact, in theory will decrease the rate of cyber-crime in regards to theft of information as it makes the whole process of stealing the information much more difficult.

## FUTURE WORK

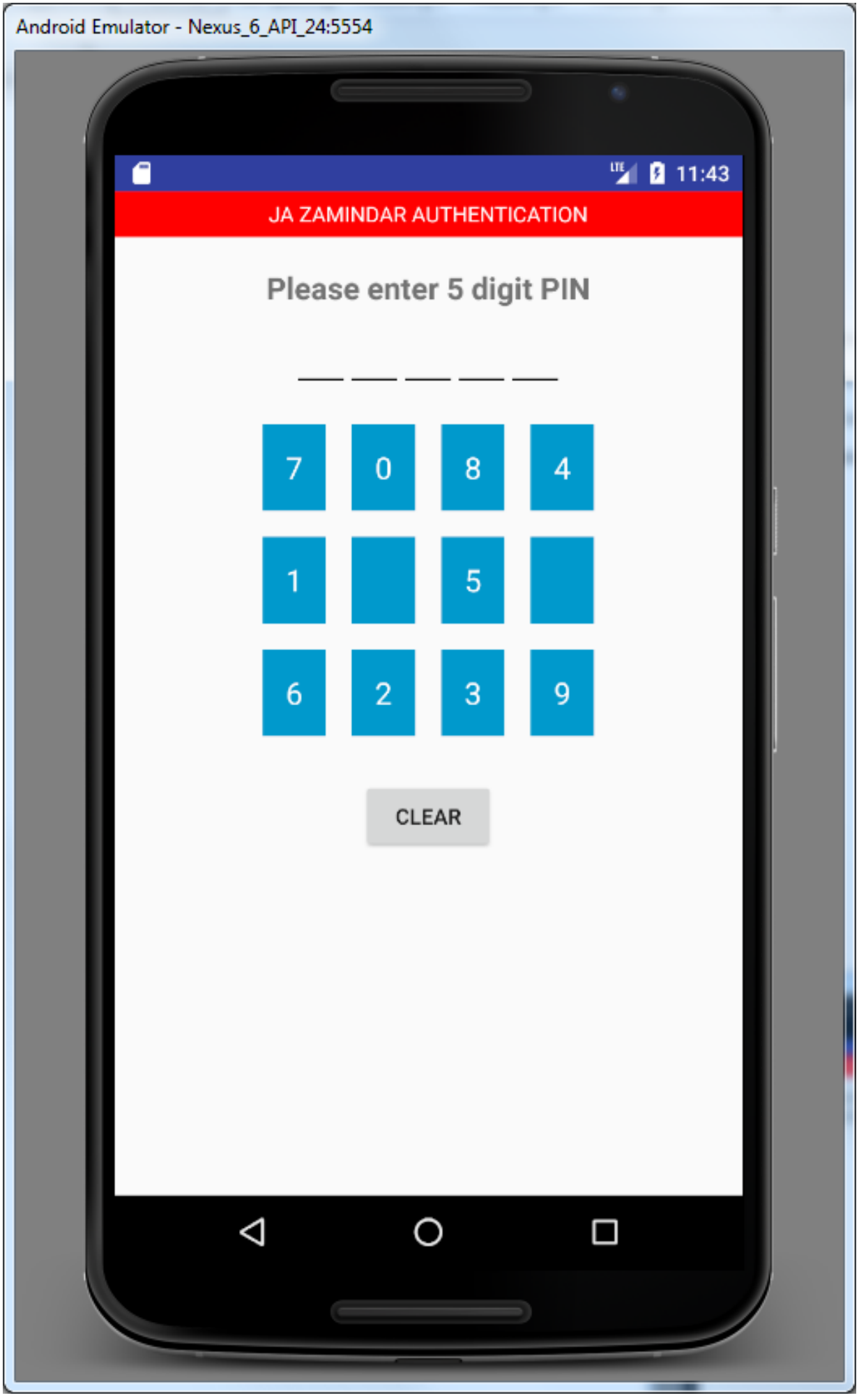
- Username Being Between a Certain Amount of Characters
- Username Can Include All Types of Characters
- Memorable Word to Include Any Characters
- A Longer PIN
- Memorable Word Page Requiring Addiotnal Character Input
- A Longer PIN
- Memorable Word Cannot Have 3 Repetitive Letters

## REFERENCES

Barclays Banking Group. (1999). Mobile Banking. Available: http://www.barclays.co.uk/BarclaysMobileBanking/MobileBankingapp/P1242609123821 . Last accessed 11th March 2017.

Tutorials Point. (2015). SDLC - RAD Model. Available: https ://www.tutorialspoint.com/sdlc/sdlc_rad_model.htm. Last accessed 20th March 2017.

Wealth & Finance International. (2015). Cybercrime Incidents on the Rise. Available: http://wealthandfinance-intl.com/cybercrime-incidents-on-the-rise. Last accessed 8th April 2017.


Android Emulator - Nexus_6_API_24:5554
11:43
JA ZAMINDAR AUTHENTICATION
Please enter 5 digit PIN
7
0
8
4
1
5
6
2
3
9
CLEAR