ANDROID PHONE KEY

MICHAEL WONG POH HONG

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons.) Electronic and Communication Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

September 2016

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature :

Name :

ID No. :

Date :

APPROVAL FOR SUBMISSION

I certify that this project report entitled **“ANDROID PHONE KEY”** was prepared by **MICHAEL WONG POH HONG** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons.) Electronic and Communication Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : _______________________

Supervisor : _______________________

Date : _______________________

Specially dedicated to

my beloved grandmother, mother and father

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Dr. Lo Fook Loong for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to my loving parents and friends who had helped me in different ways in terms of support, encouragement and also experience. This project wouldn’t be a success without them.

ANDROID PHONE KEY

ABSTRACT

The advancement of technology for the past decade has definitely triggered the need for developing more advanced devices. The traditional door lock system which comprises of a physical key and lock is slowly fading away. Although it is still considered efficient, a major key issue does occur which is the hassle of carrying a set of keys around for each lock to be unlocked. In addition to that, the distribution of keys in larger organisations could be costly. Thus, by developing a keyless system whereby only mobile phones are needed to unlock our doors, it would definitely reduce the cost tremendously and also increase flexibility in addition to providing ease of unlocking doors. In this modern society, majority of people in developed countries owns a smart phone as it has become an integral necessity which simplifies our lives. Without a doubt, utilising this important device in developing a keyless door locking system would provide an abundance of benefits because there is absolutely no need for carrying any extra objects but only to make use of existing devices which are our mobile phones.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

WAMP	Windows, Apache, MySQL, PHP
PHP	Hypertext Preprocessor

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

The capabilities of smartphones are never ending and its rate of growth is remarkably noticeable. There are tons of applications that are being used popularly in different categories for different purposes which greatly provide simplicity, comfort and also entertainment. This is primarily due to the powerful mobile operating systems such as Apple, Android, Windows and Blackberry which allow developers to create amazingly useful applications with high-end features.

However, this particular project will lean more towards Android as it will make use of the Android software development environment to develop the application. The Android operating system is a Linux-based free open source platform developed by Google which is designed mainly for touchscreen devices such as mobile phones and tablets. On this platform, applications can be developed by using Android software development environments such as Eclipse IDE or Android Studio. The main language in which the application would be written in is the Java programming language.

Currently, the official integrated development environment (IDE) for Android application is Android Studio. The Android Software Development Kit (SDK) has samples of source codes, emulator and necessary libraries to build applications on Android. The Android platform also provides a wide range of connectivity options such as Wi-Fi, Bluetooth or cellular data connection. Users or developers can also download and use external libraries to expand their application functionalities. Although JAVA already has most of the general and commonly used libraries, there are also functionalities that are not provided. So, external libraries are written by people to provide the necessary functionalities that JAVA does not offer. Moreover, Android Studio has a very user friendly and simple interface which allows developers to easily master the software hence bringing their application development to the next level.

1.2 Aims and Objectives

Besides the hassle of carrying a large amount of physical keys around, often there are also times where keys will be misplaced or people accidentally locking themselves outside of the door. The main purpose of creating a digital locking system is to deliver modern security features which can overcome the drawbacks of a typical physical door lock system. The main concept of this project is aimed at developing an android application written in JAVA (client side) which can interface and communicate with a centralised server (PC – server side) through Wi-Fi to unlock or lock specific doors at different locations depending on the user’s location. In this project, the smart locking system is targeted to be implemented in a University environment.

CHAPTER 2

LITERATURE REVIEW

2.1 Android-based Smart Locking System

Nowadays, the concepts of smart homes are fast emerging and it's effectively being integrated into society. In a paper written by Subhamay Sarker, Mithun Chakraborty and Anindita Banerjee (2014), the same concept of controlling home appliances was also being applied. The Android application communicates wirelessly using Bluetooth technology with the embedded devices. The proposed system makes use of Piconet network which is the linking of the android device with the controllers through Bluetooth modules attached to the individual microcontroller boards.

Since majority of the smart devices has the Bluetooth functionality, it is also another good alternative to use them for wireless communication. In a paper presented by Deepali Javale (2013), he recommended the use of android version 2.3.4 Gingerbread or 3.1 Honeycomb and above which has an accessory mode feature to allow interfacing the Arduino ADK with Android mobile phones. His research on smart home system caters more towards the disabled and senior citizens. Hao Shi has also done a research using the Android SDK, Java JDK and Android development tools to implement a home lighting system.

In addition, in a journal article written by Sneha Sahare (2016), she described a project that revolves around the concept of door opening automation whereby it incorporates the use of microcontroller with Arduino software. It also includes an additional feature whereby users are able to send short messaging service (sms) to unlock the door whenever they are out of the Wi-Fi coverage.

Moreover, a paper written by N.H.Ismail, Zarina Tukiran and N.N.Shamsuddin (2014) proposed an Android-based home locking system for the disabled peopled using Bluetooth connectivity. Their prototype involves the use of microcontrollers connected to the Arduino boards that can be remotely controlled via Bluetooth to lock or unlock electro-magnetic door locks. The electro-magnetic door lock is connected to a relay circuit which will be released once voltage is triggered by the Arduino board controller.

In another paper written by Hae-Duck J. Jeong, Jiyoung Lim and WooSeok Hyun (2014), it was mentioned that Bluetooth is easy to use and it is also capable of handling up to 7 device connections to a master device at a time in a piconet-based network. However, the downside is that a non-authorized person can unlock the door if the information that is being transmitted is hacked or intercepted.

Apart from the typical Bluetooth connectivity being incorporated in the Android-based home automation systems mentioned above, there are also other Android-based home automation systems that incorporate Wi-Fi as their means of peripherals interfacing. In a paper written by Shiu Kumar (2014), he designed an android application which is able to control home applications such as air-conditioner, lightings, fans, alarms and etc. His concept is to use the android application to communicate with an Arduino Ethernet based server through Wi-Fi or 3G or 4G networks. The main brain of this whole system would be the Arduino Microcontroller which will process requests from users and then carry out the necessary operations by sending specific control signals to the actuators connected. In his paper, he mentioned that the use of Personal Computers increases the cost of implementation and that it can be reduced by replacing them with microcontrollers.

Besides that, his proposed system also includes additional features whereby it notifies users of intrusion through email. Its functions can also be activated by voice commands using the Google Speech recognition which phases out the need for external voice recognition engines. In addition, a paper published by N.Hashim, N.F.A.M. Azmi, F. Idris and N.Rahim (2016) also showcased an Android-based locking system using Wi-Fi standard which is the IEEE 802.11. In their design, the Android smartphone will act as the transmitter to send signals through Wi-Fi to a Wi-Fi module connected to a PIC microcontroller which controls the relay that actuates the solenoid door lock.

All of the research analysed above has inspired this study to develop and implement an electronic keyless locking system for our University laboratories. For this particular project, it will focus mainly on unlocking doors using an Android mobile phone in a University environment. Thus, similar concept from the above journals will be adopted. In terms of wireless connectivity, Wi-Fi connection is preferred over Bluetooth because Java is very well-known for its networking capabilities hence giving us the advantage to utilize the Wi-Fi for data transfers.

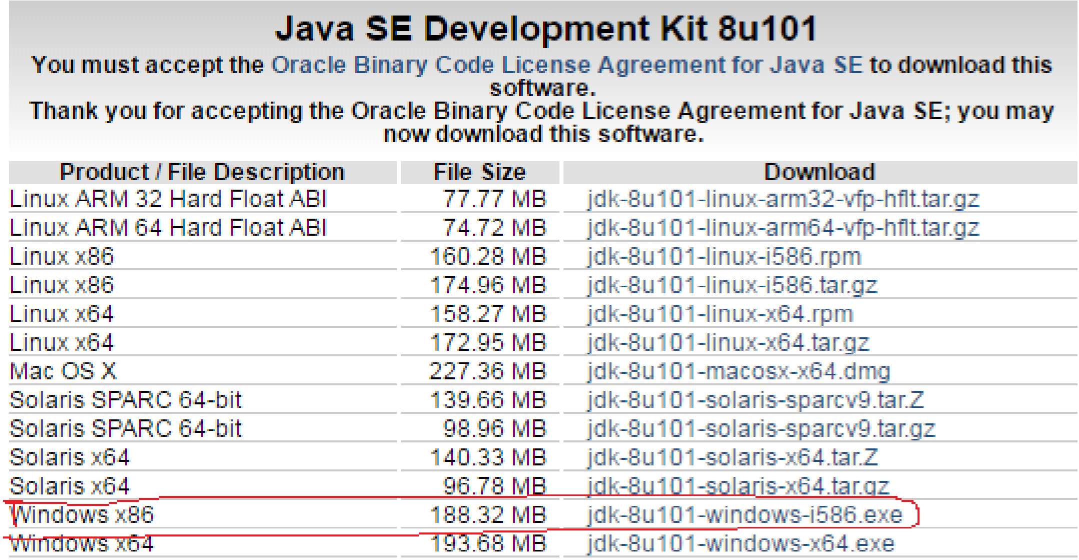
In this particular project, microcontroller system would not be implemented. Instead, its functionalities will be replaced with a personal computer (PC). This is favoured because there is no need for additional complicated circuit constructions as the research mainly focuses on writing the Android program. The main PC (server) will acts as a server which relays request signals from the client (Android phone) to the designated alternate PC which controls a specific lock to be unlocked. The MySQL database will be used to store all users' information.

CHAPTER 3

METHODOLOGY

3.1 Software Implementation

Before starting off the project, there are a few configuration steps to be executed. Firstly the latest Java JDK has to be installed. This is to make sure that the computer has the latest updated version of Java JDK. The Java JDK completes the whole package because it provides all the necessary programming tools such as the java compiler (javac), java debugger (jdb) and many other important utilities to ensure the proper performance of the Android integrated development environment. These will allow developers to fully establish a complete Android application on a Java platform.



Java SE Development Kit 8u101

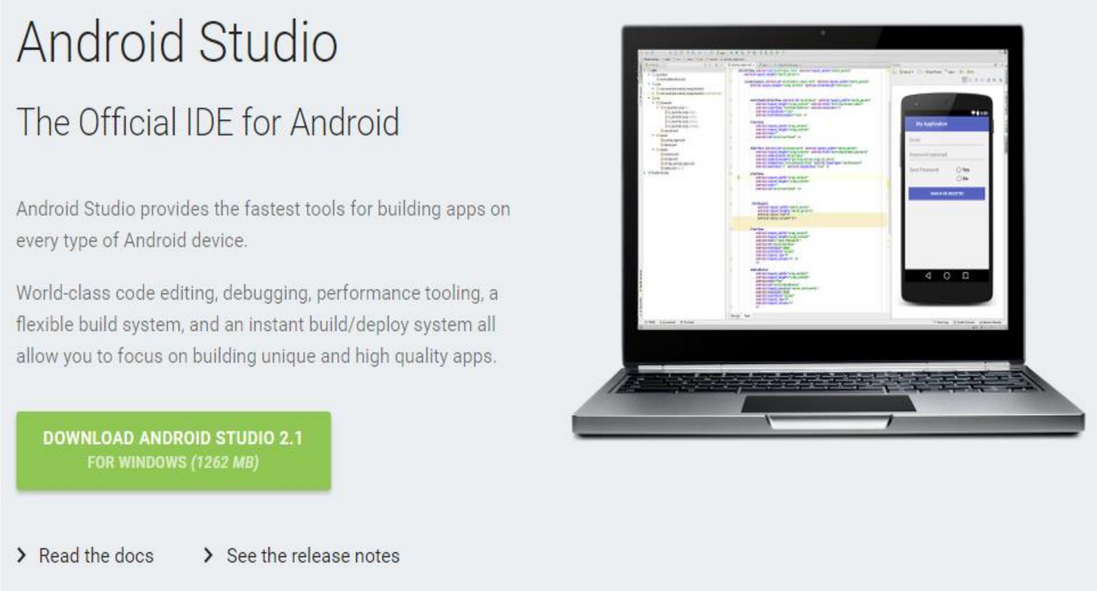
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.77 MB	jdk-8u101-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.72 MB	jdk-8u101-linux-arm64-vfp-hflt.tar.gz
Linux x86	160.28 MB	jdk-8u101-linux-i586.rpm
Linux x86	174.96 MB	jdk-8u101-linux-i586.tar.gz
Linux x64	158.27 MB	jdk-8u101-linux-x64.rpm
Linux x64	172.95 MB	jdk-8u101-linux-x64.tar.gz
Mac OS X	227.36 MB	jdk-8u101-macosx-x64.dmg
Solaris SPARC 64-bit	139.66 MB	jdk-8u101-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.96 MB	jdk-8u101-solaris-sparcv9.tar.gz
Solaris x64	140.33 MB	jdk-8u101-solaris-x64.tar.Z
Solaris x64	96.78 MB	jdk-8u101-solaris-x64.tar.gz
Windows x86	188.32 MB	jdk-8u101-windows-i586.exe
Windows x64	193.68 MB	jdk-8u101-windows-x64.exe

Figure 3.1: Java JDK installation

Once this is done, the main Android application development software which is the Android Studio is then installed. Android studio will be used primarily to program and write the client (application).



Android Studio

The Official IDE for Android

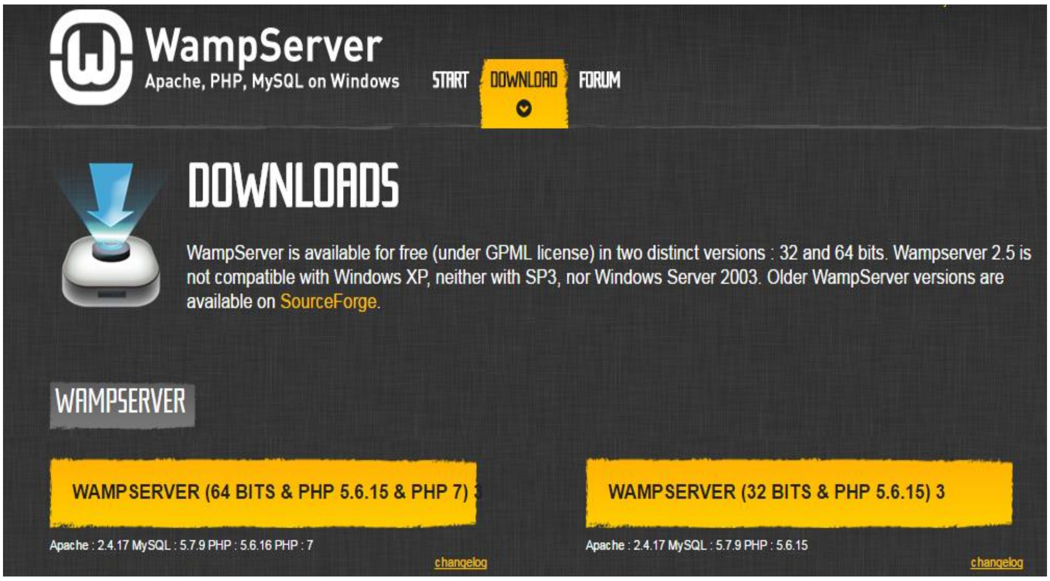
Android Studio provides the fastest tools for building apps on every type of Android device.

World-class code editing, debugging, performance tooling, a flexible build system, and an instant build/deploy system all allow you to focus on building unique and high quality apps.

Figure 3.2: Android Studio installation page

On the server side, MySQL database is used in conjunction with PHP as the server-side scripting. MySQL is a relational database management system whereby it allows users to create, update, delete and administer a database for storing important information in a form of a table. PHP is an abbreviation for Hypertext Preprocessor which serves as a bridge to send and retrieve data from the client (Android smartphone) to the MySQL database and vice versa.

PHP is also used to send and retrieve data from the server (PC) to the database as well as to store information. In order to achieve this, a web development environment called WAMP server has to be installed. WAMP is an abbreviation of Windows, Apache, MySQL and PHP. WAMP server is an easy to use application which integrates Apache, PHP and MySQL all into a single web tool. This allows developers to easily create and manage databases through a built-in function known as PhpMyAdmin.


WampServer
Apache, PHP, MySQL on Windows
START
DOWNLOAD
FORUM
DOWNLOADS
WAMPSERVER
WAMPSERVER (64 BITS & PHP 5.6.15 & PHP 7)
Apache : 2.4.17 MySQL : 5.7.9 PHP : 5.6.16 PHP : 7
changelog
WAMPSERVER (32 BITS & PHP 5.6.15) 3
Apache : 2.4.17 MySQL : 5.7.9 PHP : 5.6.15
changelog

WampServer is available for free (under GPML license) in two distinct versions : 32 and 64 bits. Wampserver 2.5 is not compatible with Windows XP, neither with SP3, nor Windows Server 2003. Older WampServer versions are available on SourceForge.

Figure 3.3: WampServer installation

On top of that, another separate IDE which is the Eclipse IDE for Java developers has to be installed. Eclipse will be the main building block to program and write the server which is also the central processing as well as the control unit of the whole system.

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration...

Figure 3.4: Eclipse IDE for Java Developers download.

3.2 Hardware Implementation

The whole hardware prototype will consist of an Android mobile phone, personal computers (PC) and an electro-magnetic door lock circuit. The reason why PCs are used is because in every laboratory, there would already be a PC sitting inside and thus this will help minimizes the cost of setting up any additional hardware connections. There will be a main PC which acts as a server that coordinates the request signals from users to unlock a specific door lock connected to their respective computers.

All of these signals will be transmitted through Wi-Fi and routed to their respective locks based on their assigned unique IP addresses. Each of the electro-magnetic lock circuits are connected to the PC via the serial port also known as the RS-232 port.

3.3 Process Flow

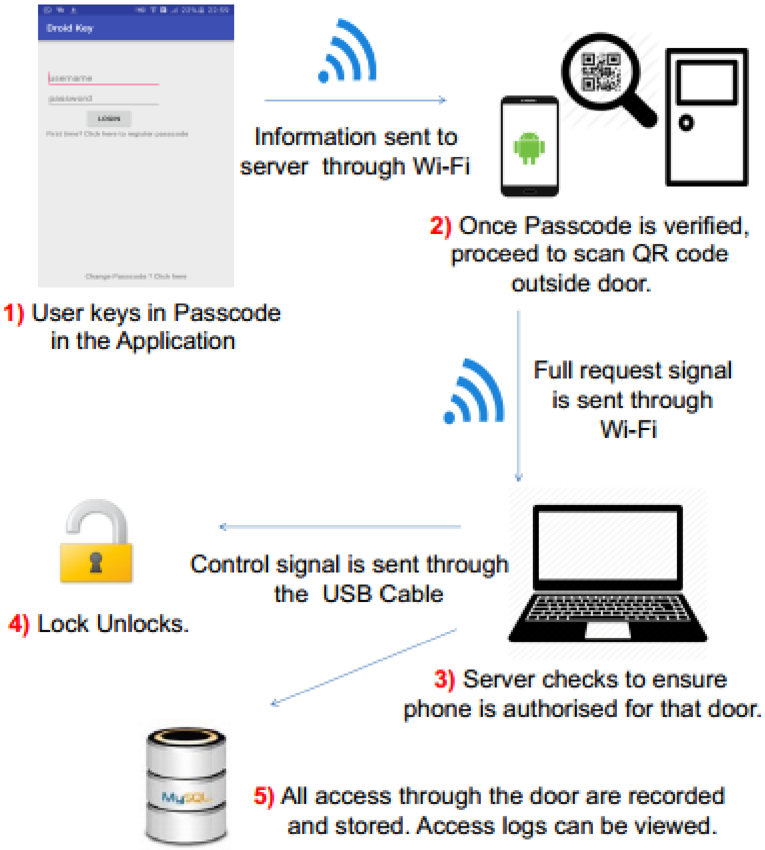

Droid Key
username
password
LOGIN
First time? Click here to register passcode
Change Passcode ? Click here
Information sent to server through Wi-Fi
2) Once Passcode is verified, proceed to scan QR code outside door.
1) User keys in Passcode in the Application
Full request signal is sent through Wi-Fi
Control signal is sent through the USB Cable
4) Lock Unlocks.
3) Server checks to ensure phone is authorised for that door.
MySQL
5) All access through the door are recorded and stored. Access logs can be viewed.

Figure 3.5: Process flow of operation

The targeted users of this application would be the university students. Before using the door-lock application, every student will be required to have their individual subscriber identity module (SIM) card number registered for their account into the database. This can be done by registering with the authorised technician in-charge of the Android-based door lock system in the university.

Once that is done, the application can now be used. At this point, the system would be able to recognize the users based on their SIM card number registered. Users will have to first register a password for their account which will be used to log in into the application. When users key in their passwords, the application would check with the database to see if the password matches. On top of that, the system would also verify if the SIM card serial number matches with the database. It means that only authorised phones will be allowed and non-authorised phone will have no access to the application. This is to ensure that only the right person with the right device can have access to a particular door.

Once the user has successfully logged into their account, they will be requested to scan the QR code which is located at every entrance of the laboratories. When a particular QR code is scanned, the application would know exactly which room the user is located at and the necessary control signal would be sent over across Wi-Fi to the central server PC which would then re-route the signal to the designated IP address of the PC to which the lock is connected to. The lock then unlocks to provide access.

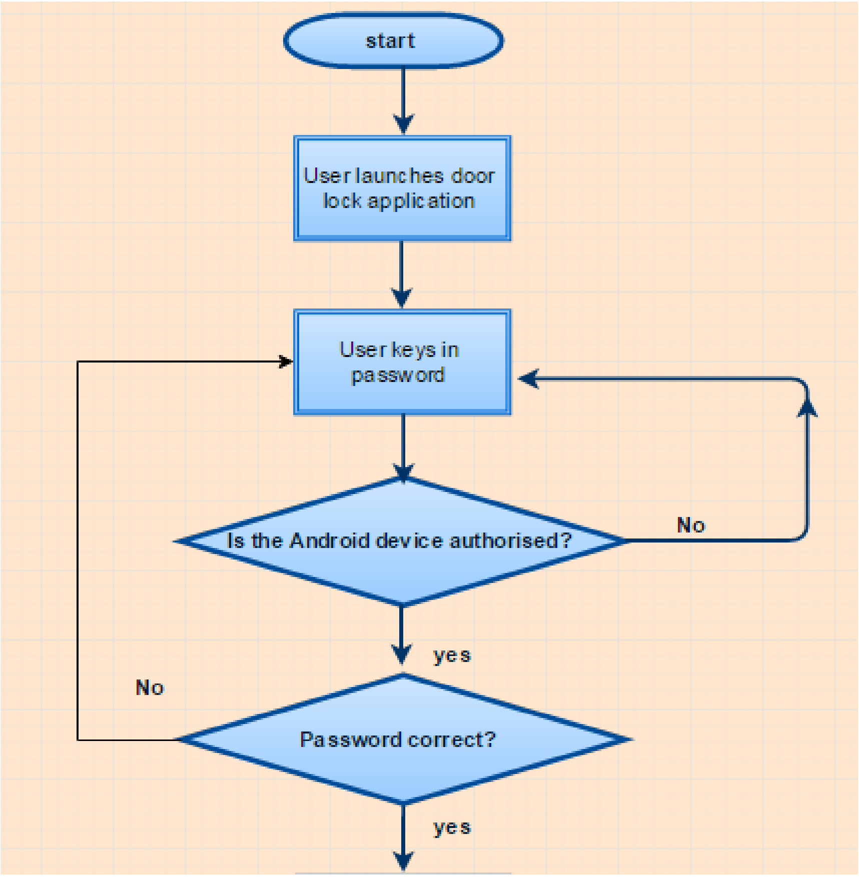
To tighten the security, all access through the doors will be recorded and stored in the MySQL database. In addition to that, the passwords will be hashed on the client-side before being sent over to the server for storage in the database. Hashing is also known as a one-way encryption whereby the encryption of a particular data cannot be decrypted to retrieve its original content. It also means that we will never know the true content of a hashed string.

A typical hashing algorithm takes in an input message and then converts it into a fixed size alphanumeric string which is also known as the hash value or message digest. This is a standard procedure whereby passwords are not encouraged to be stored as plaintext to protect user’s information from being stolen. Besides that, passwords are not being transmitted in plaintext but instead it is hashed before being sent over to the database. This helps to prevent hackers from intercepting the transmission mid-way and ultimately prevent them from stealing the information.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Detailed flowchart of operation


start
User launches door lock application
User keys in password
Is the Android device authorised?
No
yes
No
Password correct?
yes

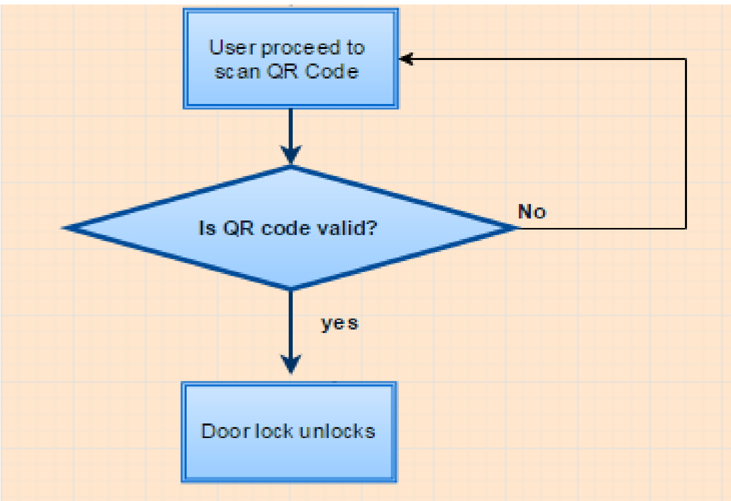


Figure 4.1: Detailed flowchart of operation

4.2 Android phone key application

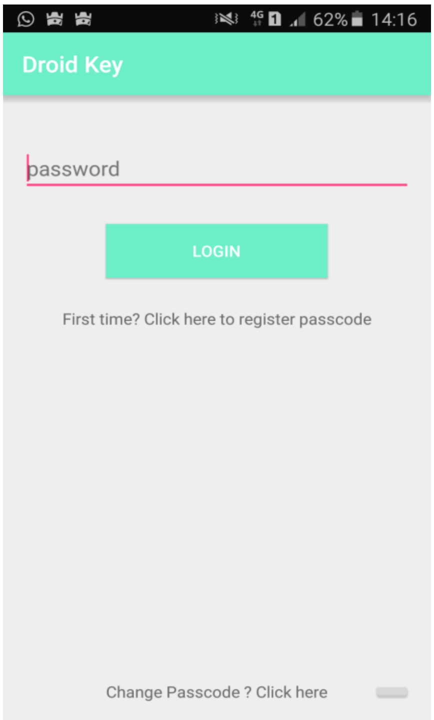
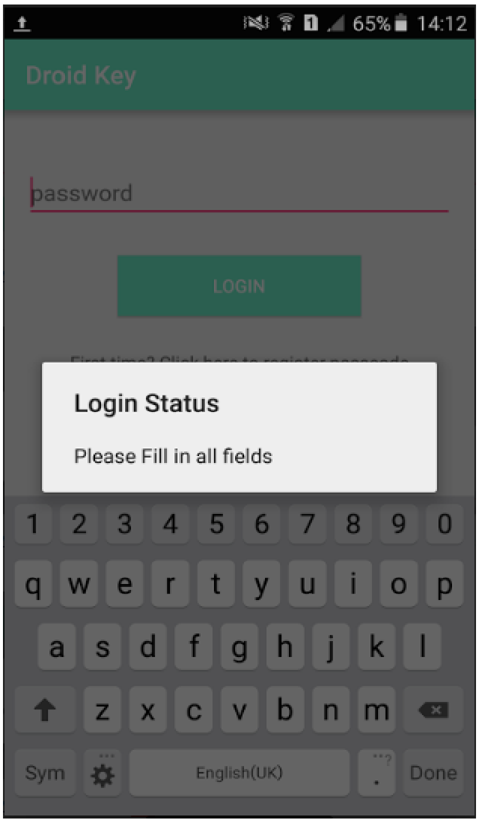

62% 14:16
Droid Key
password
LOGIN
First time? Click here to register passcode
Change Passcode ? Click here

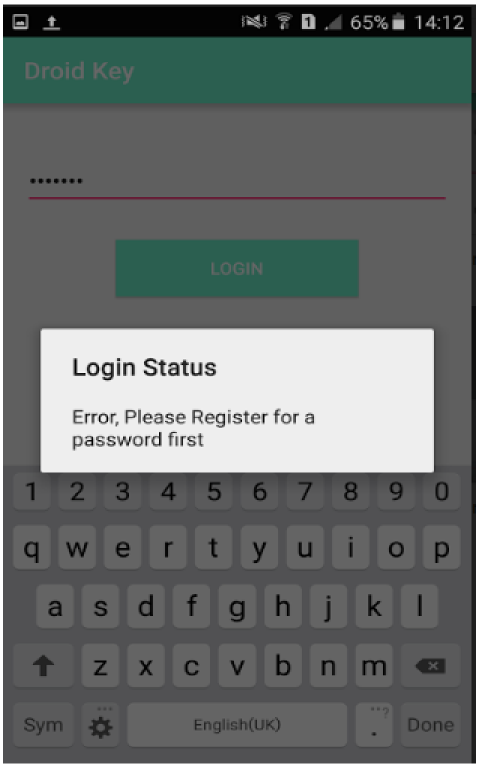
Figure 4.2: Login Screen



Login Status

Please Fill in all fields

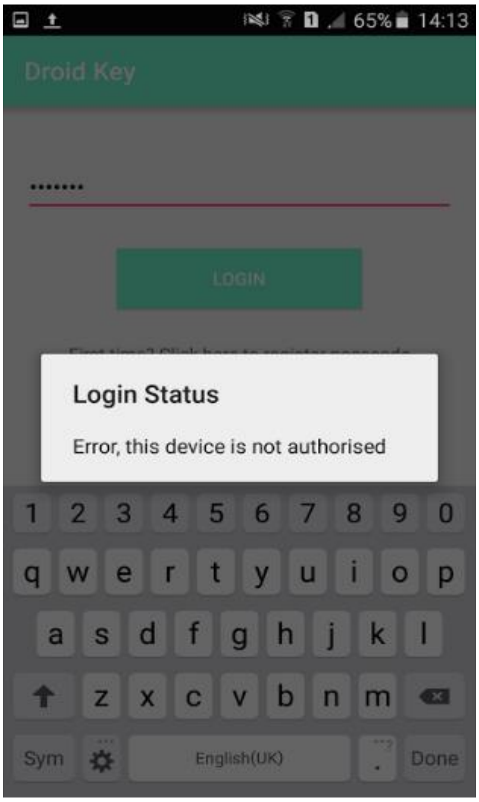
Figure 4.3: Login screen



Login Status

Error, Please Register for a password first

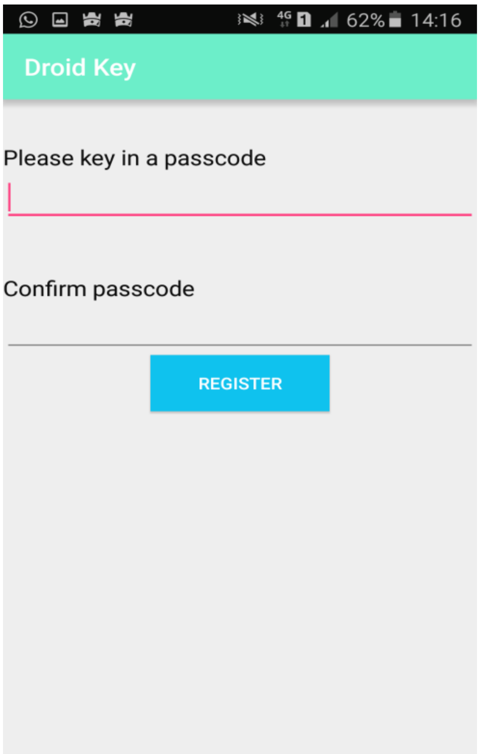
Figure 4.4: Login Screen



Login Status

Error, this device is not authorised

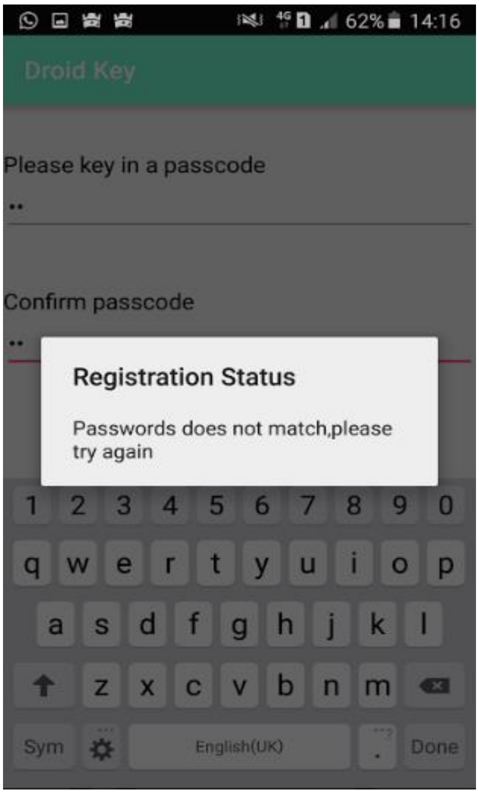
Figure 4.5: Login Screen



Please key in a passcode

Confirm passcode

Figure 4.6: Registration Screen



Please key in a passcode

••

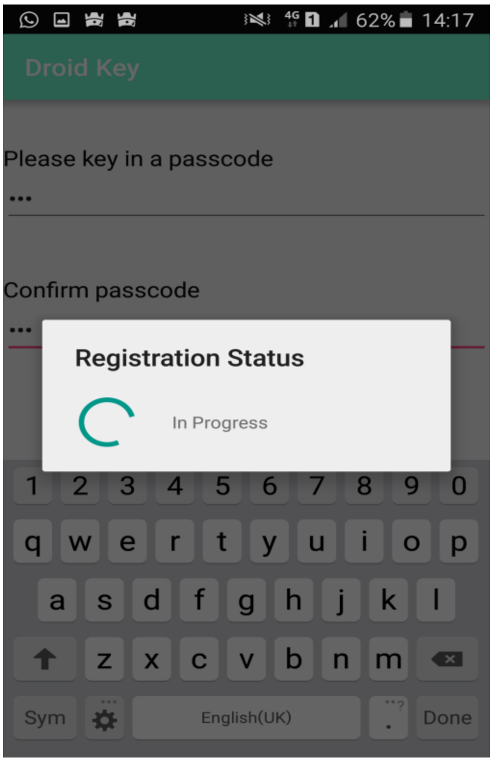
Confirm passcode

••

Registration Status

Passwords does not match,please try again

Figure 4.7: Registration Screen



Please key in a passcode

•••

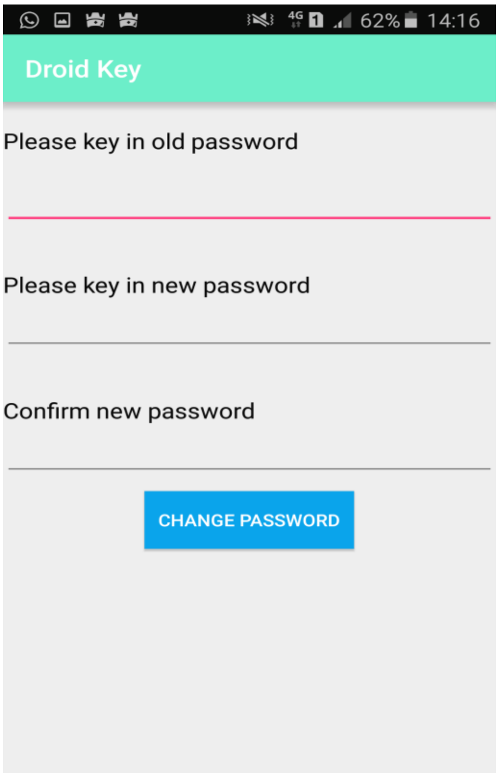
Confirm passcode

•••

Registration Status

In Progress

Figure 4.8: Registration Screen

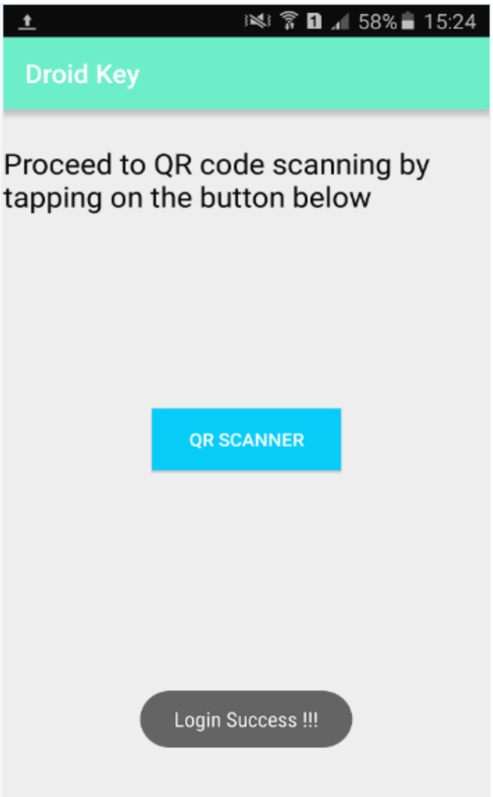


Please key in old password

Please key in new password

Confirm new password

Figure 4.9: Password change screen



Proceed to QR code scanning by tapping on the button below

Login Success !!!

Figure 4.10: QR code opening screen

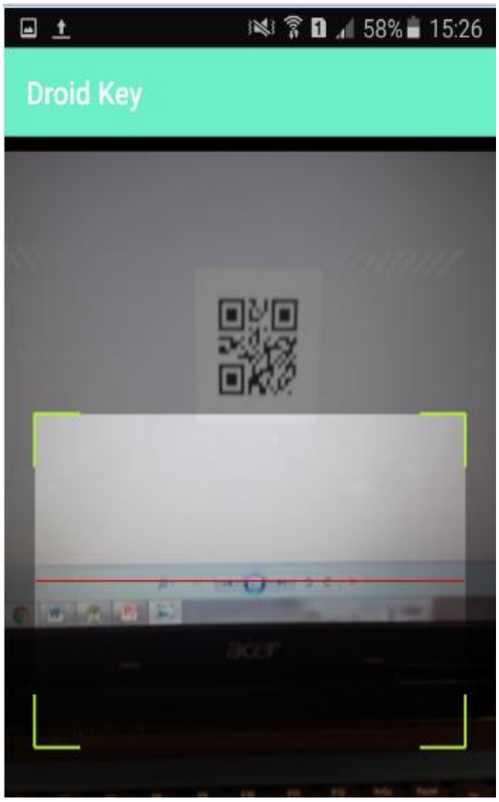


Figure 4.11: QR code scanning screen

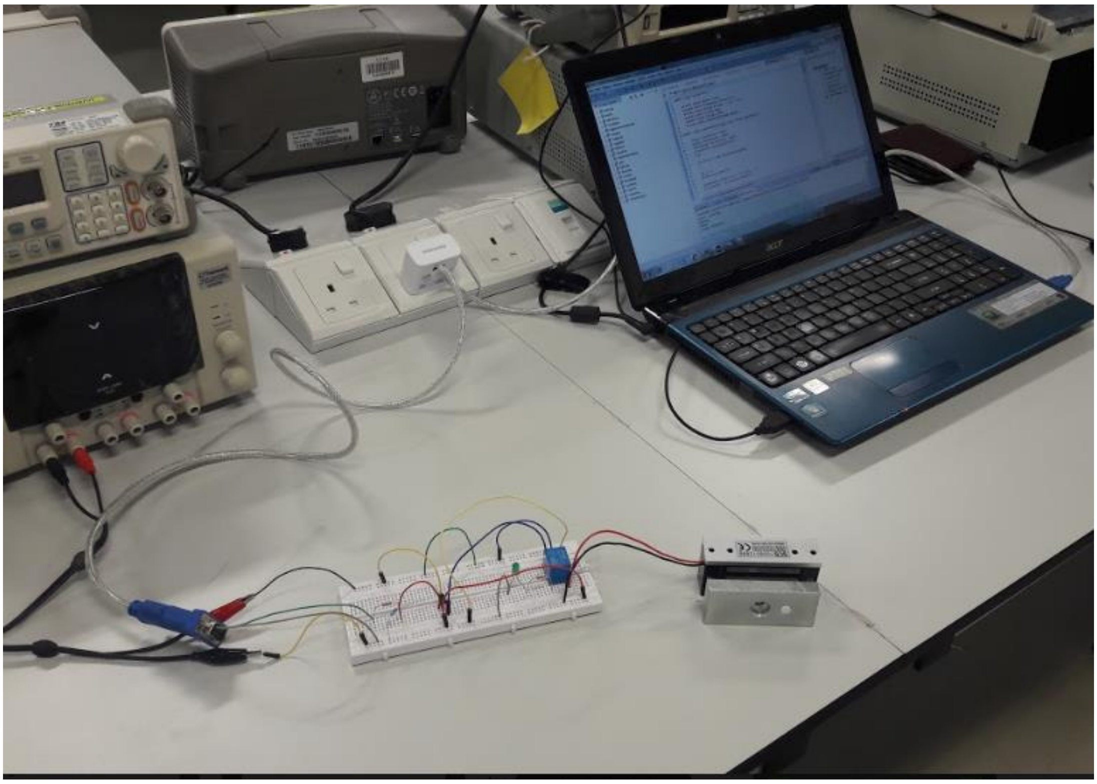
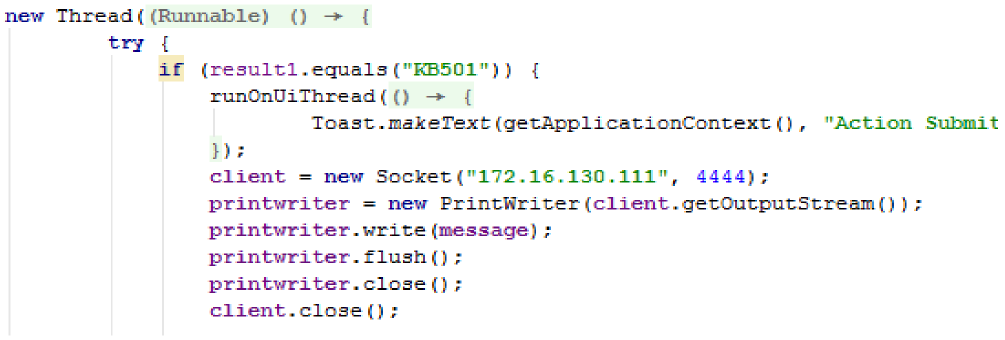


Figure 4.12: Android Phone Key system prototype

Figure 4.1 shows the detailed flowchart of the entire operation. Figure 4.2 shows the login screen of the application. Figure 4.3, figure 4.4 and figure 4.5 shows the errors that occur at the application’s login screen. Figures 4.6, 4.7 and 4.8 show the registration screen. Figure 4.9 shows the password change screen. Figure 4.10 and Figure 4.11 shows the QR code screens. Figure 4.12 illustrates the Android phone key system prototype.

4.3 Interfacing Android application with PC

In order for the Android application (client) to communicate with the PC (server), network programming has to be implemented. Two separate programs have been written whereby the application is written using Android Studio while the server program is written using Eclipse. Java has a java.net package which contains all the necessary classes to implement the necessary network programming specifically Socket programming in this case. In order for the communication to occur, both the devices (Android smartphone & Laptop) have to be connected to the same Wi-Fi network. The client will first create a socket and attempts to connect that socket to a server by specifying the targeted IP address and port number. At the same time, the server will be invoking the accept() method whereby it will be waiting until a client is connected to the server. Once the connection is established, both devices can send and receive data between each other through the input and output streams of the socket.

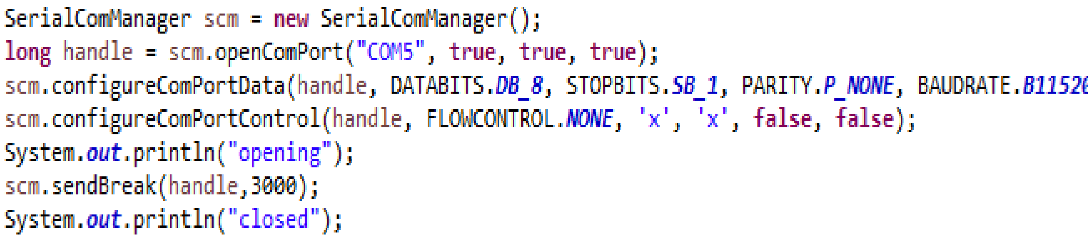


```
new Thread((Runnable) () → {
        try {
            if (result1.equals("KB501")) {
                runOnUiThread(() → {
                        Toast.makeText(getApplicationContext(), "Action Submit
                });
                client = new Socket("172.16.130.111", 4444);
                printwriter = new PrintWriter(client.getOutputStream());
                printwriter.write(message);
                printwriter.flush();
                printwriter.close();
                client.close();
```

Figure 4.13: Socket connection establishment code segment

4.4 Serial Port Interfacing between PC and Door Lock Circuit

The PC is connected to the door lock circuit via the RS232-USB converter cable. In order for the PC to communicate with the lock circuit through the serial port, an external library called ‘serial communication manager (scm)’ was used. Serial communication manager is a Java library for communication over serial port. This library is easy and simple to use as the documentation made by the author is very clear and concise. After importing this external library into the program, a simple program was written to output a pulse through the RS232 Tx pin 3 whenever a specific input from the Android application was received.



```
SerialComManager scm = new SerialComManager();
long handle = scm.openComPort("COM5", true, true, true);
scm.configureComPortData(handle, DATABITS.DB_8, STOPBITS.SB_1, PARITY.P_NONE, BAUDRATE.B11520
scm.configureComPortControl(handle, FLOWCONTROL.NONE, 'x', 'x', false, false);
System.out.println("opening");
scm.sendBreak(handle,3000);
System.out.println("closed");
```

Figure 4.14: Opening Serial COM port code segment

From the figure above, it can be seen that the pulse output was set to be held for a brief moment of 3000ms or 3 seconds. A typical RS232 port contains 9 pins. When the RS232 port pin was engaged, the voltage of the TXD pin (pin 3) with respect to ground (pin 5) was measured to be around 5V. The output voltage from Pin 3 of the RS232 serial port is used as the signal to control the door lock circuit which will be explained in detail further in this report.

4.5 Electro-magnetic lock circuit working principle

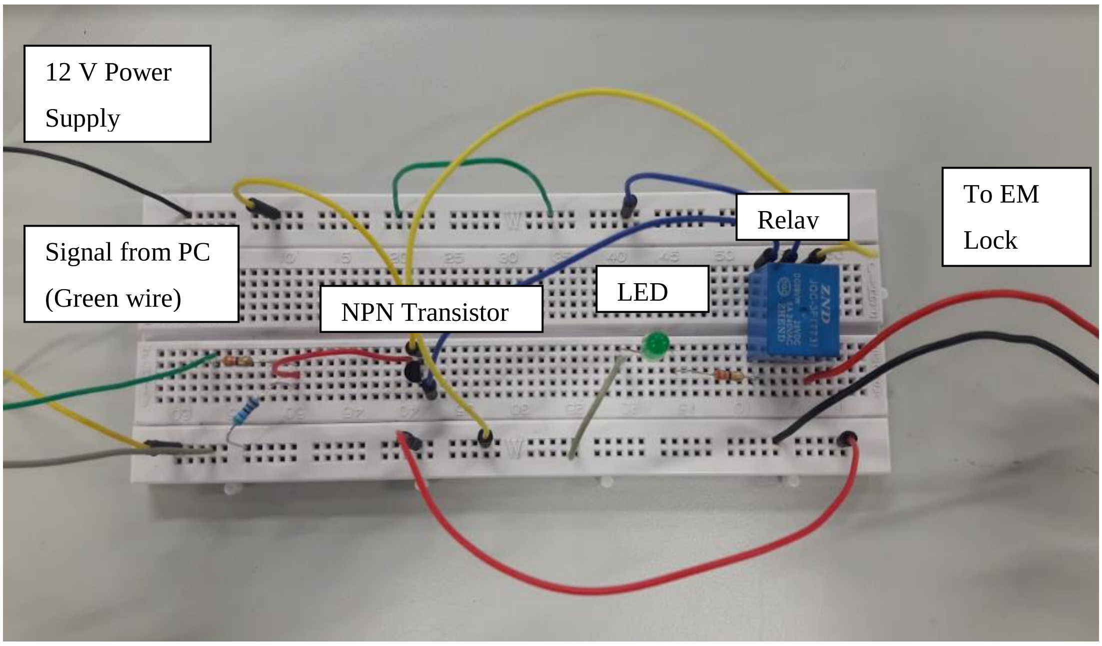


Figure 4.15: Door lock Circuit

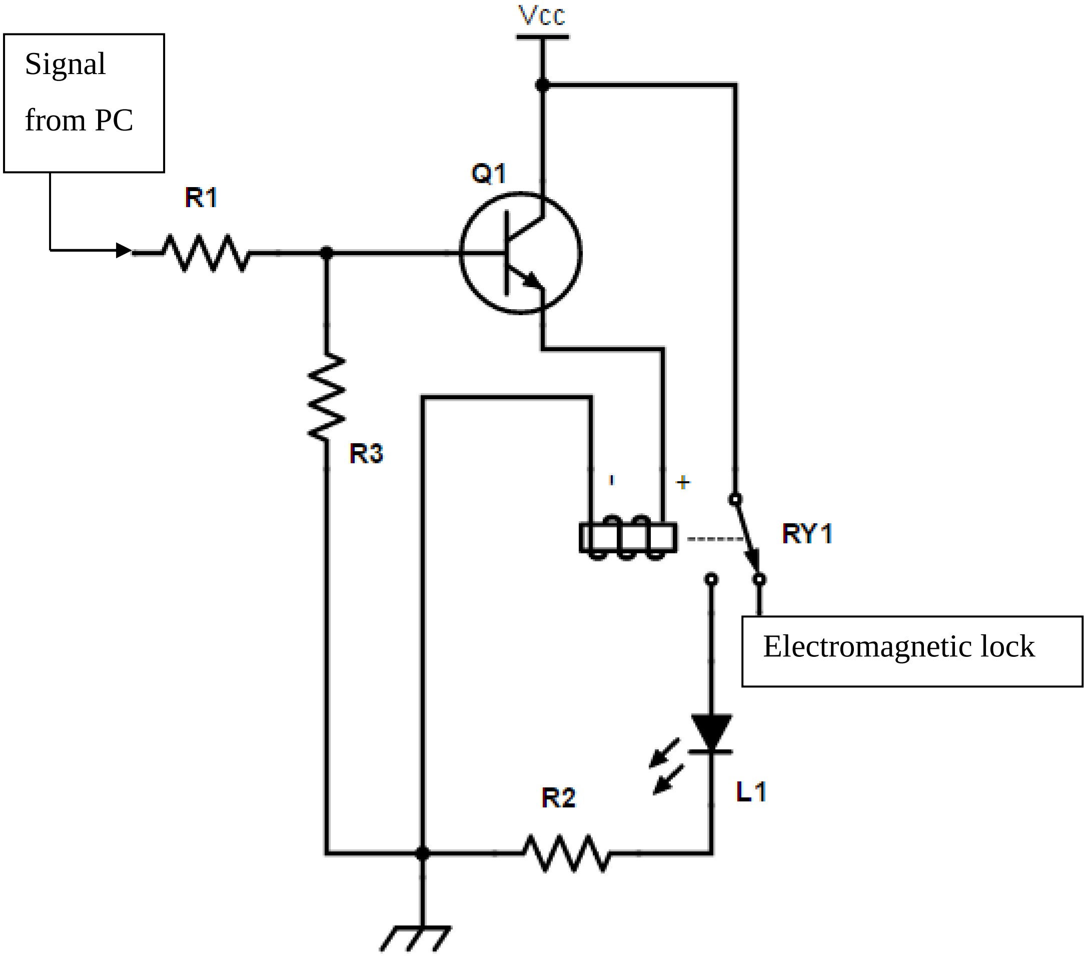


Figure 4.16: Door Lock circuit schematic

Figure 4.16 shows the schematic diagram of the lock circuit. The electro-magnetic lock circuit was constructed by using a combination of 12 V single pole double throw (SPDT) relay and an NPN transistor which acts as a switch that will only be triggered when a request signal is sent from the client (Android smartphone) to the server (PC). Once the request signal is received and processed, the server will then send control signals through the RS232-USB cable to the base junction of the NPN transistor which will then saturate the NPN transistor thus allowing current to flow through from the collector to the emitter junction of the transistor and magnetize the coil in the relay. Once the coil is magnetized, the switch will be pulled over to the other side hence disjointing the current flow from the voltage source to the electro-magnetic lock causing it to unlock the door. At the same time, the LED will light up to indicate that the lock is unlocked.

4.5.1 Single Pole Double Throw Relay Working Principle

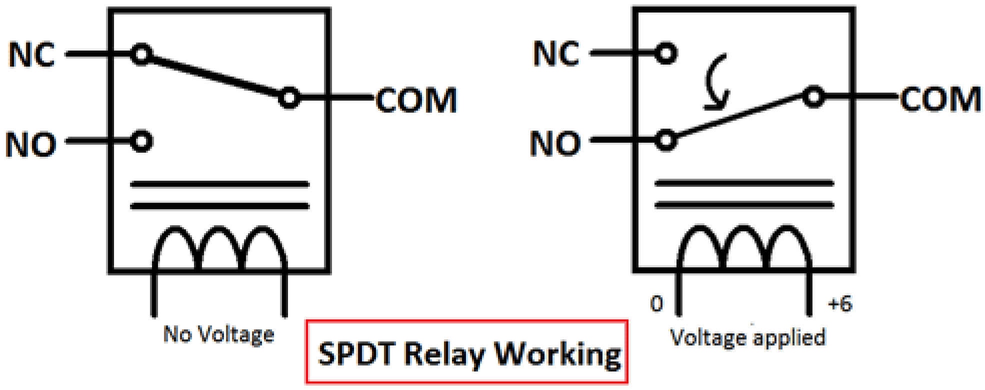


Figure 4.17: Single Pole Double Throw Relay Schematic

Figure 4.17 shows the schematic of a single pole double throw relay. The electromagnetic lock used in this project is a fail-safe magnetic lock which will magnetize when a voltage of 12 Volts is supplied to it. The lock is connected to the normally closed (NC) terminal of the relay with a voltage of 12 Volts supplied to the COM terminal. This enables the lock to stay magnetized hence keeping the door locked. When the necessary control signals are transmitted and received, current will be able to flow through and magnetize the coil in the relay hence pulling the switch over to the normally open (NO) terminal of the relay causing the current to flow towards the LED connected at the NO terminal. During this period, no current is able to flow through the electromagnetic lock causing the lock to de-magnetize and the door is unlocked.

4.5.2 NPN Transistor working principle

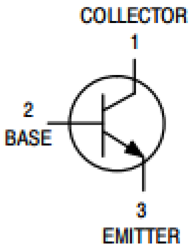
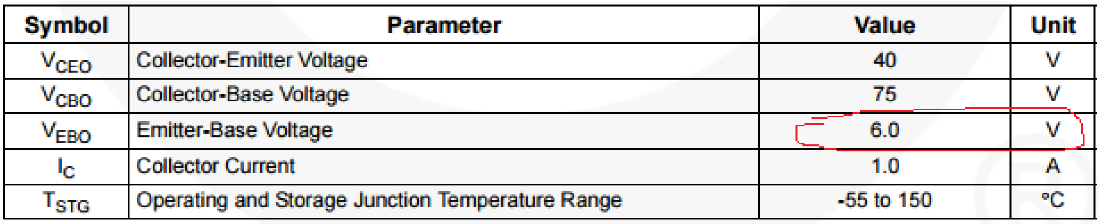


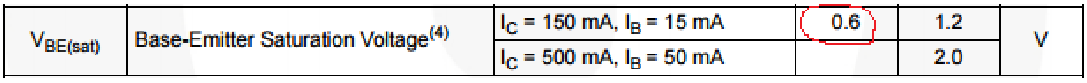
Figure 4.18: NPN transistor schematic

The transistor used in this project is a general purpose NPN transistor 2N2222. It is used as a switch to turn on a load (electromagnetic lock) which requires a larger voltage than a computer can handle. The electromagnetic lock is directly connected to the power supply whereas the NPN transistor acts as a switch that is grounded. The maximum voltage supplied from the RS232 port of the PC is only up to 5 Volts which is not enough to power up and magnetize the coil of the 12 V relay used. This would lead to the electromagnetic lock being magnetized continuously. Therefore, the utilization of an NPN transistor solves the issue by allowing the 5 Volts from the RS232 port to flow into the base junction of the transistor and drive it into saturation state. Once the transistor is saturated, the current will be able to flow through the collector-emitter junction and magnetize the relay coil which will then de-magnetize the magnetic lock hence unlocking the door. According to the datasheet of the NPN transistor:



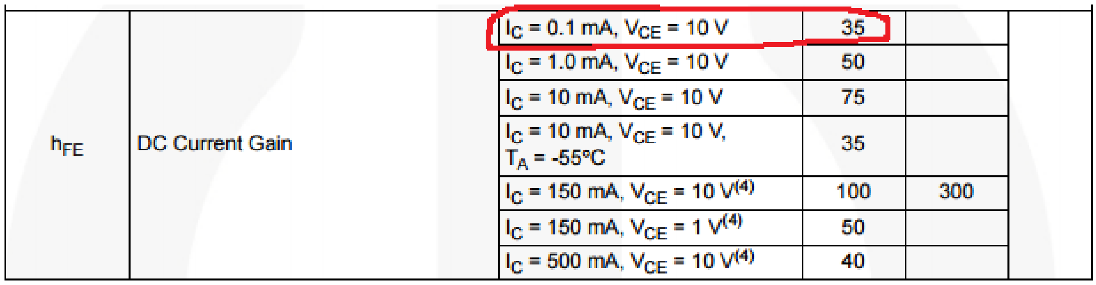
Symbol	Parameter	Value	Unit
V_{CEO}	Collector-Emitter Voltage	40	V
V_{CBO}	Collector-Base Voltage	75	V
V_{EBO}	Emitter-Base Voltage	6.0	V
I_C	Collector Current	1.0	A
T_{STG}	Operating and Storage Junction Temperature Range	-55 to 150	°C

Figure 4.19: 2N2222 transistor datasheet



$V_{BE(sat)}$	Base-Emitter Saturation Voltage[4]	I_C = 150 mA, I_B = 15 mA	0.6	1.2	V
		I_C = 500 mA, I_B = 50 mA		2.0	

Figure 4.20: V_{BE} saturation for 2N2222



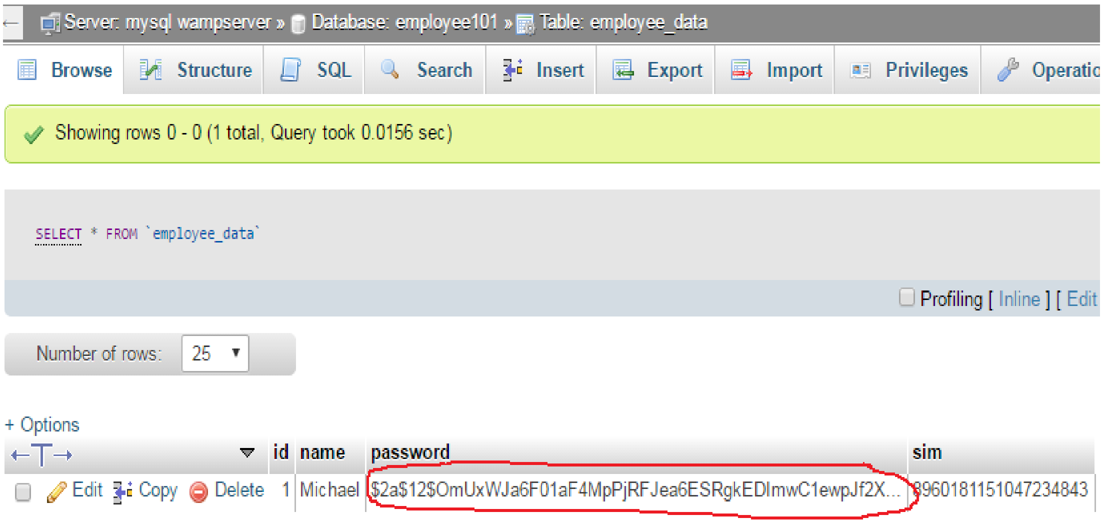
h_{FE}	DC Current Gain	I_C = 0.1 mA, V_{CE} = 10 V	35		
		I_C = 1.0 mA, V_{CE} = 10 V	50		
		I_C = 10 mA, V_{CE} = 10 V	75		
		I_C = 10 mA, V_{CE} = 10 V, T_A = -55°C	35		
		I_C = 150 mA, V_{CE} = 10 V[4]	100	300	
		I_C = 150 mA, V_{CE} = 1 V[4]	50		
		I_C = 500 mA, V_{CE} = 10 V[4]	40		

Figure 4.21: Current gain of 2N2222

The breakdown voltage at the base-emitter junction is 6V and the $V_{BE(sat)}$ is 0.6V. In addition to that, the relay coil current requires at least 33mA. Furthermore, the DC current gain of the transistor is at least 35. The transistor base current can be calculated by using the equation $I_B = I_C/h_{FE}$. I_B = 33m/35 = 0.94mA which can be rounded up to 1mA. By referring to Figure 4.16 with all this information, the resistor value R1 to be connected to the base junction of the NPN transistor can be calculated. Since the voltage supplied by the RS232 pins is around 5V, R1 = (5V – 0.6V)/1ma = 4400ohms. In this case, we used a 3.3k ohm resistor as we have some extra margin to ensure the transistor is driven hard into saturation. There is also a resistor R3 which is connected to the base junction of the transistor and it is grounded. It shunts the signal from the RS232 port into the ground to prevent any current or voltage spike from damaging the transistor. Besides that, the resistor R3 prevents the transistor from turning on due to any small random voltage that may appear at the RS232 pin by draining any additional charge out from the base of the transistor. As a rule of thumb, the resistor R3 value used is at least 10 times larger than the value of resistor R1. R2 is a 1k ohm resistor to limit the current through the LED thus preventing the LED from blowing.

4.6 Password hashing

All passwords are hashed on the client side (Android application) before being sent over to the server to be stored into the database. Initially, when a user registers for a password, the password will be hashed and sent over to the database for storage. When a user logs in using the password, the password will be hashed again and the resulting hash value will be compared to the hash value stored previously in the database. If the hash value matches, the user will be granted access. However, just by hashing alone is not sufficient to ensure reasonable protection of credentials. This can easily be cracked by hackers using a rainbow table. The reason behind this is because, for every similar input, similar password hashes will be generated. Since a rainbow table has a list of pre-computed hash functions, it is just a matter of time before the hacker gets to recover the plaintext password. Hence, in order to prevent the occurrence of this, salt has been prepended to the passwords before hashing them together to create a stronger hashing algorithm. Salt is a unique string that is concatenated on to the passwords before the hashing algorithm takes place. The reason why salted hashing is so useful is because it requires extra computational power in order to retrieve the plaintext password. In addition to that, to make it even better, random salt is used instead of a fixed salt thus making it more difficult for the hackers to use a rainbow table or dictionary attack to find a collision as it is not feasible to compute a table of all possible combinations of hashes. Besides that, even if the hacker gets a hold of the salt string, the database wouldn’t be compromised because random salt is used for every hashed password. This would result in a different output hash even if two same passwords are hashed with the same hashing algorithm ultimately making it more tedious for hackers to find a collision.



Showing rows 0 - 0 (1 total, Query took 0.0156 sec)

```
SELECT * FROM `employee_data`
```

id	name	password	sim
1	Michael	$2a$12$OmUxWJa6F01aF4MpPjRFJea6ESRgkEDImwC1ewpJf2X...	8960181151047234843

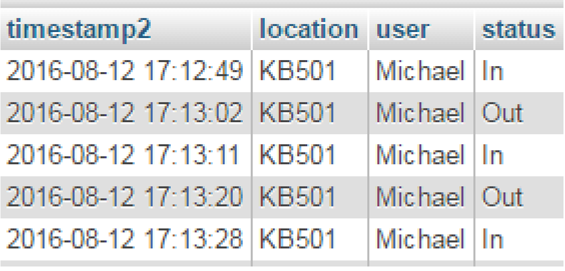
Figure 4.22: Hashed password

An external library called jBCrypt was used in order to perform the hashing algorithm above. jBCrypt is a Java implementation using Blowfish algorithm. In this algorithm, the salted hashing function is also iterated with proper number of rounds before the output hash is produced. This helps to increase the security of the passwords as the computation time taken to find a collision would increase exponentially.

4.7 Access Logs

When a request signal to unlock a particular door is received by the server (PC), the server will perform 2 tasks concurrently. It will send the necessary control signals to unlock the lock and at the same time it will record the access into the database through PHP scripting.

Every access through the door will be recorded and stored into the database. The access log contains information of the user, location, time, date, as well as every entry and exit.



timestamp2	location	user	status
2016-08-12 17:12:49	KB501	Michael	In
2016-08-12 17:13:02	KB501	Michael	Out
2016-08-12 17:13:11	KB501	Michael	In
2016-08-12 17:13:20	KB501	Michael	Out
2016-08-12 17:13:28	KB501	Michael	In

Figure 4.23: Access log

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

The Android-based smart locking system proposed has been developed and implemented successfully. At the end of this project, I was able to learn, experience and understand from the Android application developers' point of view when developing an application. It requires a great amount of dedication and critical thinking in order to come up with a working system. Every inch of the process has to be evaluated carefully before proceeding to write the application. Hence the process flow of the operation has to be laid out beforehand to clearly define the objectives and goals of the project. This project has been frustrating but at the same time enlightening as I have not done a project which involves such a huge amount of programming. The best part of the project was being able to establish communication between the devices (Android Smartphone, Laptop and Door Lock) and actually seeing them work and talk to each other. These moments were ground-breaking for me. In addition, I've come to a realization that no single module can function efficiently without proper communication between the devices. It's these little connections that come together to form bigger and greater things. Throughout this project, there were many problems that I've encountered. I've learned that most of the problems encountered were caused by misinterpretation of the programming language's syntax as well as very small and almost unnoticeable syntax errors such as a missing letter, semicolon, spelling, etc. PHP scripting was the one I had to pay extra attention to when coding because it does not notify the user when there's an occurrence of a syntax error. In the future, this proposed Android-based locking system can be enhanced and improved further by implementing additional biometric authentication such as thumbprint screening outside the doors. In case of any loss of mobile devices, authorised users would still be able to access the doors by scanning their thumbprints.

REFERENCES

Badrul Hisham, A. and Ishak, M. (2014). Bluetooth-Based Home Automation System Using an Android Phone. *Jurnal Teknologi*, 70(3).

F. Shawki. (2015). MICROCONTROLLER BASED SMART HOME WITH SECURITY USING GSM TECHNOLOGY', *International Journal of Research in Engineering and Technology*, 04(06), pp. 20–28.

Fedelin Jr., D. (2013). Controlled Access Cabinet Door Lock Via Android Phone. *IJMIE*, 3(7), pp.379-388.

Ghazal, B. and Al-Khatib, K. (2015). Smart Home Automation System for Elderly, and Handicapped People using XBee. *IJSH*, 9(4), pp.203-210.

Jeong, H., Lee, W., Lim, J. and Hyun, W. (2015). Utilizing a Bluetooth remote lock system for a smartphone. *Pervasive and Mobile Computing*, 24, pp.150-165.

K.S., V. and P.C., S. (2015). Automatic Tap Control System in the Smart Home using Android and Arduino. *International Journal of Computer Applications*, 127(8), pp.19-23.

Kumar, S. (2014). Ubiquitous Smart Home System Using Android Application. *IJCNC*, 6(1), pp.33-43.

N, Hashim. (2016). Smartphone Activated Door Lock Using Wifi. *ARPN Journal of Engineering and Applied Sciences*, 11(5), pp.3309-3311.

R., P. (2015). Home Automation using Android App and Bluetooth. *International Journal on Recent and Innovation Trends in Computing and Communication*, 3(2), pp.815-819.

Reza Khan, S. and Sultana Dristy, F. (2015). Android Based Security and Home Automation System. *IJASA*, 3(1), pp.15-24.

Sahare, S. (2016). Automated Door Control System Using Android Phone and GSM Modem. *International Journal of Advanced Research in Computer Science and Software Engineering*, 6(2), pp.616-618.

APPENDICES

APPENDIX A: PHP Scripts

conn.php

```
<?php

$db_name = "employee101";
$mysql_username = "root";
$mysql_password = "";
$server_name = "localhost";
$conn= mysqli_connect($server_name, $mysql_username, $mysql_password,
$db_name);

?>
```

login.php

```
<?php

require "conn.php";
$user_pass = $_POST["password"];
$sim = $_POST["sim_serial"];
$verify = "select * from employee_data where sim like '$sim' ";
$verify_result = mysqli_query($conn,$verify);
$vresult = mysqli_fetch_array($verify_result,MYSQLI_ASSOC);
$sql = "select * from employee_data where sim like '$sim' ";
$result = mysqli_query($conn ,$sql);

if(mysqli_num_rows($result)>0){
echo($vresult['password']);
}else{
	echo "Error, this device is not authorised";
}

?>
```

register1.php

```
<?php
require "conn.php";

$user_pass = $_POST["password"];
$sim = $_POST["sim_serial"];
$user_pass1 = $_POST["passcode"];

$mysql_query1 = "update employee_data set password='$user_pass' where sim like '$sim' ";
$sql = "select * from employee_data where sim like '$sim' ";
$sqll = "select * from employee_data where password like '$user_pass' ";
$sqlll = "select * from employee_data where sim like '$sim' ";
$result = mysqli_query($conn ,$sql);
$result1= mysqli_query($conn,$sqll);
$result2 = mysqli_query($conn,$sqlll);
$row = mysqli_fetch_array($result2,MYSQLI_ASSOC);
if(mysqli_num_rows($result)>0){
        if(mysqli_num_rows($result1)==0){
            if(empty($row['password'])){
                if(mysqli_query($conn,$mysql_query1)){
                    echo 'Registration Complete';
                }else{
                    echo 'Could Not Register, Please try again';
                }
            }else{
                echo ' Only one Passcode is allowed per device';
            }
        }else{
            echo ' Only one Passcode per device is allowed';
        }
}else{
    echo 'Error, this device is not authorised ';
}

?>
```

password.php

```
<?php
require "conn.php";

$pass_1 = $_POST["password1"];
$pass_2 = $_POST["password2"];
$pass_3 = $_POST["password3"];
$sim = $_POST["sim_serial"];
```

```
$mysql_query = "select * from employee_data where sim like '$sim' " ;
$result = mysqli_query($conn,$mysql_query);
$row = mysqli_fetch_array($result,MYSQLI_ASSOC);
$mysql_query1 = "update employee_data set password = '$pass_2' where sim like
'$sim' ";


if(password_verify($pass_1,$row['password'])){
        if(mysqli_query($conn,$mysql_query1)){
                echo 'Password change completed';
        }else{
                echo 'Error, please try again' ;
        }
}else{
echo 'Old password does not match, please key in again';
}

?>
```

APPENDIX B: Android Studio program code

MainActivity.java

```
package com.example.user.myapplication4;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.telephony.TelephonyManager;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    EditText  PasswordEt,NameEt;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        PasswordEt = (EditText)findViewById(R.id.etPassword);

    }


    public void OnLogin(View view) {
        String password = PasswordEt.getText().toString();
        String type = "Login";

        BackgroundWorker backgroundWorker = new BackgroundWorker(this);
        backgroundWorker.execute(type,password, getSimSerial());

    }


    public void passCode(View view){
        startActivity(new Intent(MainActivity.this,passcode.class));
    }

    public void OnPass(View view){
        startActivity(new Intent(MainActivity.this,ChangePass.class));
    }

    public String getSimSerial() {
        String Sim = "";
        TelephonyManager tm = (TelephonyManager)
getSystemService(Context.TELEPHONY_SERVICE);
        if (tm.getSimSerialNumber() != null) {
            Sim = tm.getSimSerialNumber();
        }
        return Sim;
    }


}
```

BackgroundWorker.java

```
package com.example.user.myapplication4;

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.telephony.TelephonyManager;
import android.widget.Button;
import android.view.View;
import android.content.Intent;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;


public class BackgroundWorker extends AsyncTask<String,Void,String> {

    Context context;
    AlertDialog alertDialog;
    ProgressDialog loading;
    BackgroundWorker (Context ctx) {
        context = ctx;
    }


    @Override
    protected String doInBackground(String... params) {
        String type = params[0];
        String password = params[1];
        String login_url = "http://192.168.0.104/webapp/login.php";

        if(!password.isEmpty()) {
                try {

                    String sim_serial = params[2];
                    String passwordh = BCrypt.hashpw(password,
BCrypt.gensalt(12));

                    URL url = new URL(login_url);
                    HttpURLConnection httpURLConnection =
(HttpURLConnection) url.openConnection();
                    httpURLConnection.setRequestMethod("POST");
                    httpURLConnection.setDoOutput(true);
                    httpURLConnection.setDoInput(true);
                    OutputStream outputStream =
httpURLConnection.getOutputStream();
                    BufferedWriter bufferedWriter = new BufferedWriter(new
OutputStreamWriter(outputStream, "UTF-8"));
                    String post_data = URLEncoder.encode("password", "UTF-
8") + "=" + URLEncoder.encode(passwordh, "UTF-8") + "&" +
URLEncoder.encode("sim_serial", "UTF-8") + "=" +
URLEncoder.encode(sim_serial, "UTF-8");
                    bufferedWriter.write(post_data);
                    bufferedWriter.flush();
                    bufferedWriter.close();
                    outputStream.close();
                    InputStream inputStream =
httpURLConnection.getInputStream();
                    BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(inputStream, "iso-8859-1"));
                    String result = "";
                    String line = "";
                    while ((line = bufferedReader.readLine()) != null) {
                        result += line;
                    }

                    bufferedReader.close();
                    inputStream.close();
                    httpURLConnection.disconnect();

                    if (result.equals("Error, this device is not
authorised")) {
                        return result;
                    }else if(result.isEmpty()){
                        return "Error, Please Register for a password first
";
                    }
                    else if(BCrypt.checkpw(password,result)) {

                        return "Login Success !!!";
                    }
                        return "Invalid Credentials, please try again";

                } catch (MalformedURLException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                }
        }
        return "Please Fill in all fields";
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        loading = ProgressDialog.show(context,"Processing","Please
Wait",true);
        alertDialog = new AlertDialog.Builder(context).create();
        alertDialog.setTitle("Login Status ");
    }


    @Override
    protected void onPostExecute(String result) {
        alertDialog.setMessage(result);
        alertDialog.show();


        if(loading!=null){
            loading.dismiss();
        }
        if(result.equals("Login Success !!!")) {
            alertDialog.dismiss();
            Toast.makeText(context,
                    result,
                    Toast.LENGTH_LONG).show();
            context.startActivity(new Intent(context, QrScanner.class));
        }


    }


    @Override
    protected void onProgressUpdate(Void... values) {
        super.onProgressUpdate(values);
    }

}
```

BackgroundWorker2.java

```
package com.example.user.myapplication4;

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;
import java.util.logging.Handler;

public class BackgroundWorker2 extends AsyncTask<String,Void,String> {

    Context context;
    AlertDialog alertDialog;
    ProgressDialog loading;
    String text = "Please fill in all fields";
    BackgroundWorker2 (Context ctx) {
        context = ctx;
    }
    @Override
    protected String doInBackground(String...params){
        String type = params[0];
        String password = params[1];
        String passcode = params[3];

        String login_url = "http://192.168.0.104/webapp/register1.php";
        if(!password.isEmpty()&&!passcode.isEmpty()){
            if(password.equals(passcode)) {

                try {
                    String sim_serial = params[2];
                    String passwordh = BCrypt.hashpw(password,
BCrypt.gensalt(12));
                    String passcodeh = BCrypt.hashpw(passcode,
BCrypt.gensalt(12));

                    URL url = new URL(login_url);
                    HttpURLConnection httpURLConnection =
(HttpURLConnection) url.openConnection();
                    httpURLConnection.setRequestMethod("POST");
                    httpURLConnection.setDoOutput(true);
                    httpURLConnection.setDoInput(true);
                    OutputStream outputStream =
httpURLConnection.getOutputStream();
                    BufferedWriter bufferedWriter = new BufferedWriter(new
OutputStreamWriter(outputStream, "UTF-8"));
                    String post_data = URLEncoder.encode("password", "UTF-
8") + "=" + URLEncoder.encode(passwordh, "UTF-8") + "&" +
URLEncoder.encode("sim_serial", "UTF-8") + "=" +
URLEncoder.encode(sim_serial, "UTF-8") + "&" + URLEncoder.encode("passcode",
"UTF-8") + "=" + URLEncoder.encode(passcodeh, "UTF-8");
                    bufferedWriter.write(post_data);
                    bufferedWriter.flush();
                    bufferedWriter.close();
                    outputStream.close();

                    InputStream inputStream =
httpURLConnection.getInputStream();
                    BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(inputStream, "iso-8859-1"));
                    String result = "";
                    String line = "";
                    while ((line = bufferedReader.readLine()) != null) {
                        result += line;
                    }
                    bufferedReader.close();
                    inputStream.close();
                    httpURLConnection.disconnect();
                    return result;
                } catch (MalformedURLException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
            return "Passwords does not match,please try again";
        }
        return "Please fill in all fields";
    }


    @Override
    protected void onPreExecute() {
        super.onPreExecute();

        loading = ProgressDialog.show(context, "Registration Status", "In
Progress", true);
        alertDialog = new AlertDialog.Builder(context).create();
        alertDialog.setTitle("Registration Status ");
    }

    protected void onPostExecute(String result) {
        alertDialog.setMessage(result);
        alertDialog.show();




        if(loading!=null){
            loading.dismiss();
        }



        if(result.equals("Registration Complete")) {
            alertDialog.dismiss();
            Toast.makeText(context,
                    result,
                    Toast.LENGTH_LONG).show();
            context.startActivity(new Intent(context, MainActivity.class));
        }

    }

    @Override
    protected void onProgressUpdate(Void... values) {
        super.onProgressUpdate(values);
    }
}
```

BackgroundWorker3.java

```
package com.example.user.myapplication4;

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;

public class BackgroundWorker3 extends AsyncTask<String,Void,String> {

    Context context;
    AlertDialog alertDialog;
    ProgressDialog loading;
    BackgroundWorker3 (Context ctx) {
        context = ctx;
    }
    @Override
    protected String doInBackground(String...params){
        String type = params[0];
        String password1 = params[1];
        String password2 = params[2];
        String password3 = params[4];
        String login_url = "http://192.168.0.104/webapp/password.php";

if(!password1.isEmpty()&&!password2.isEmpty()&&!password3.isEmpty()){
            if(password2.equals(password3)) {
                try {

                    String sim_serial = params[3];
                    String password2h = BCrypt.hashpw(password2,
BCrypt.gensalt(12));
                    String password3h = BCrypt.hashpw(password3,
BCrypt.gensalt(12));


                    URL url = new URL(login_url);
                    HttpURLConnection httpURLConnection =
(HttpURLConnection) url.openConnection();
                    httpURLConnection.setRequestMethod("POST");
                    httpURLConnection.setDoOutput(true);
                    httpURLConnection.setDoInput(true);
                    OutputStream outputStream =
httpURLConnection.getOutputStream();
                    BufferedWriter bufferedWriter = new BufferedWriter(new
OutputStreamWriter(outputStream, "UTF-8"));
                    String post_data = URLEncoder.encode("password1", "UTF-
8") + "=" + URLEncoder.encode(password1, "UTF-8") + "&" +
URLEncoder.encode("password2", "UTF-8") + "=" +
URLEncoder.encode(password2h, "UTF-8") + "&" +
URLEncoder.encode("sim_serial", "UTF-8") + "=" +
URLEncoder.encode(sim_serial, "UTF-8") + "&" +
URLEncoder.encode("password3", "UTF-8") + "=" +
URLEncoder.encode(password3h, "UTF-8");
                    bufferedWriter.write(post_data);
                    bufferedWriter.flush();
                    bufferedWriter.close();
                    outputStream.close();
                    InputStream inputStream =
httpURLConnection.getInputStream();
                    BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(inputStream, "iso-8859-1"));
                    String result = "";
                    String line = "";
                    while ((line = bufferedReader.readLine()) != null) {
                        result += line;
                    }
                    bufferedReader.close();
                    inputStream.close();
                    httpURLConnection.disconnect();
                    return result;
                } catch (MalformedURLException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
            return "New passwords does not match, please key in again";
        }
        return "Please Fill in all Fields";

    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        loading = ProgressDialog.show(context, "Executing...", "Please
Wait", true);
        alertDialog = new AlertDialog.Builder(context).create();
        alertDialog.setTitle("Processing.. ");
    }

    protected void onPostExecute(String result) {
        alertDialog.setMessage(result);
        alertDialog.show();


        if(loading!=null){
            loading.dismiss();
        }

        if(result.equals("Password change completed")) {
            alertDialog.dismiss();
            Toast.makeText(context,
                    result,
                    Toast.LENGTH_LONG).show();
            context.startActivity(new Intent(context, MainActivity.class));
        }


    }

    @Override
    protected void onProgressUpdate(Void... values) {
        super.onProgressUpdate(values);
    }
}
```

passcode.java

```
package com.example.user.myapplication4;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;


public class passcode extends MainActivity {

    EditText PasswordEt,PasscodeEt;

    @Override
    protected void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.passcode_new);
        PasswordEt = (EditText)findViewById(R.id.passcode);
        PasscodeEt = (EditText)findViewById(R.id.passcode1);
    }


    public void OnRegister(View view){
        String password = PasswordEt.getText().toString();
        String passcode = PasscodeEt.getText().toString();
        String type = "Register";

        BackgroundWorker2 backgroundWorker = new BackgroundWorker2(this);
        backgroundWorker.execute(type,password, getSimSerial(),passcode);
    }
}
```

ChangePass.java

```
package com.example.user.myapplication4;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.EditText;

public class ChangePass extends MainActivity{
    EditText Pass1,Pass2,Pass3;
    @Override
    protected void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.pass_change);
        Pass1 = (EditText)findViewById(R.id.pass1);
        Pass2 = (EditText)findViewById(R.id.pass2);
        Pass3 = (EditText)findViewById(R.id.pass3);
    }

    public void OnChange(View view){
        String password1 = Pass1.getText().toString();
        String password2 = Pass2.getText().toString();
        String password3 = Pass3.getText().toString();
        String type = "password";

        BackgroundWorker3 backgroundWorker = new BackgroundWorker3(this);

backgroundWorker.execute(type,password1,password2,getSimSerial(),password3);
    }
}
```

QrScanner.java

```
package com.example.user.myapplication4;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.telephony.TelephonyManager;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import com.google.zxing.Result;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.math.BigInteger;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.Socket;
import java.net.URL;
import java.net.URLEncoder;
import java.net.UnknownHostException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

import me.dm7.barcodescanner.zxing.ZXingScannerView;

public class QrScanner extends MainActivity implements
ZXingScannerView.ResultHandler {
    private ZXingScannerView mScannerView;
    private String message = "unlock";
    private String message1 = "unlock1";
    private String message2 = "exit";
    private String message3 = "exit1";
    private Socket client;
    private Socket client1;
    private PrintWriter printwriter;
    private DataOutputStream dataout;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.qr_code);

    }

    public void QrScanner(View view){

        mScannerView = new ZXingScannerView(this);   // Programmatically
initialize the scanner view
        setContentView(mScannerView);
        mScannerView.setResultHandler(this); // Register ourselves as a
handler for scan results.
        mScannerView.startCamera();         // Start camera



    }

    @Override
    public void handleResult(final Result rawResult) {
        // Do something with the result here
        mScannerView.stopCameraPreview();
        Log.e("handler", rawResult.getText()); // Prints scan results
        Log.e("handler", rawResult.getBarcodeFormat().toString()); // Prints
the scan format (qrcode)

        final String result1 = rawResult.getText().toString();

        AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(this);
        alertDialogBuilder.setMessage("Is this the correct room?");

        mScannerView.stopCameraPreview();
        alertDialogBuilder.setPositiveButton("yes", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface arg0, int arg1) {
                new Thread(new Runnable() {
                    @Override
                    public void run() {
                        try {
                            if (result1.equals("KB501")) {
                                runOnUiThread(new Runnable() {
                                    @Override
                                    public void run() {

Toast.makeText(getApplicationContext(), "Action Submitted",
Toast.LENGTH_LONG).show();
                                    }
                                });
                                client = new Socket("172.16.142.51", 4444);
                                printwriter = new
PrintWriter(client.getOutputStream());
                                printwriter.write(message);
                                printwriter.flush();
                                printwriter.close();
                                client.close();
                                try {
                                    String sim_serial = getSimSerial();
                                    String login_url =
"http://172.16.142.51/webapp/jdbc.php";
                                    URL url = new URL(login_url);
                                    HttpURLConnection httpURLConnection =
(HttpURLConnection) url.openConnection();

httpURLConnection.setRequestMethod("POST");
                                    httpURLConnection.setDoOutput(true);
                                    httpURLConnection.setDoInput(true);
                                    OutputStream outputStream =
httpURLConnection.getOutputStream();
                                    BufferedWriter bufferedWriter = new
BufferedWriter(new OutputStreamWriter(outputStream, "UTF-8"));
                                    String post_data =
URLEncoder.encode("sim_serial", "UTF-8") + "=" +
URLEncoder.encode(sim_serial, "UTF-8");
                                    bufferedWriter.write(post_data);
                                    bufferedWriter.flush();
                                    bufferedWriter.close();
                                    outputStream.close();
                                    InputStream inputStream =
httpURLConnection.getInputStream();
                                    BufferedReader bufferedReader = new
BufferedReader(new InputStreamReader(inputStream, "iso-8859-1"));
                                    String result = "";
                                    String line = "";
                                    while ((line =
bufferedReader.readLine()) != null) {
                                        result += line;
                                    }
                                    bufferedReader.close();
                                    inputStream.close();
                                    httpURLConnection.disconnect();
                                }catch(MalformedURLException e){
                                    e.printStackTrace();
                                }catch(IOException e){
                                    e.printStackTrace();
                                }



                            }
                            else if(result1.equals("KB502")){
                                runOnUiThread(new Runnable() {
                                    @Override
                                    public void run() {

Toast.makeText(getApplicationContext(), "Action Submitted",
Toast.LENGTH_LONG).show();
                                    }
                                });
                                client = new Socket("172.16.130.111", 4444);
                                printwriter = new
PrintWriter(client.getOutputStream());
                                printwriter.write(message1);
                                printwriter.flush();
                                printwriter.close();
                                client.close();

                                try {
                                    String sim_serial = getSimSerial();
                                    String login_url =
"http://172.16.130.111/webapp/jdbc.php";
                                    URL url = new URL(login_url);
                                    HttpURLConnection httpURLConnection =
(HttpURLConnection) url.openConnection();

httpURLConnection.setRequestMethod("POST");
                                    httpURLConnection.setDoOutput(true);
                                    httpURLConnection.setDoInput(true);
                                    OutputStream outputStream =
httpURLConnection.getOutputStream();
                                    BufferedWriter bufferedWriter = new
BufferedWriter(new OutputStreamWriter(outputStream, "UTF-8"));
                                    String post_data =
URLEncoder.encode("sim_serial", "UTF-8") + "=" +
URLEncoder.encode(sim_serial, "UTF-8");
                                    bufferedWriter.write(post_data);
                                    bufferedWriter.flush();
                                    bufferedWriter.close();
                                    outputStream.close();
                                    InputStream inputStream =
httpURLConnection.getInputStream();
                                    BufferedReader bufferedReader = new
BufferedReader(new InputStreamReader(inputStream, "iso-8859-1"));
                                    String result = "";
                                    String line = "";
                                    while ((line =
bufferedReader.readLine()) != null) {
                                        result += line;
                                    }
                                    bufferedReader.close();
                                    inputStream.close();
                                    httpURLConnection.disconnect();
                                }catch(MalformedURLException e){
                                    e.printStackTrace();
                                }catch(IOException e){
                                    e.printStackTrace();
                                }

                            }
                            else if(result1.equals("exit")){
                                runOnUiThread(new Runnable() {
                                    @Override
                                    public void run() {

Toast.makeText(getApplicationContext(),"Exiting room",Toast.LENGTH_LONG);
                                    }
                                });
                                client = new Socket("172.16.142.51", 4444);
                                printwriter = new
PrintWriter(client.getOutputStream());
                                printwriter.write(message2);
                                printwriter.flush();
                                printwriter.close();
                                client.close();

                                try {
                                    String sim_serial = getSimSerial();
                                    String login_url =
"http://172.16.142.51/webapp/jdbc.php";
                                    URL url = new URL(login_url);
                                    HttpURLConnection httpURLConnection =
(HttpURLConnection) url.openConnection();

httpURLConnection.setRequestMethod("POST");
                                    httpURLConnection.setDoOutput(true);
                                    httpURLConnection.setDoInput(true);
                                    OutputStream outputStream =
httpURLConnection.getOutputStream();
                                    BufferedWriter bufferedWriter = new
BufferedWriter(new OutputStreamWriter(outputStream, "UTF-8"));
                                    String post_data =
URLEncoder.encode("sim_serial", "UTF-8") + "=" +
URLEncoder.encode(sim_serial, "UTF-8");
                                    bufferedWriter.write(post_data);
                                    bufferedWriter.flush();
                                    bufferedWriter.close();
                                    outputStream.close();
                                    InputStream inputStream =
httpURLConnection.getInputStream();
                                    BufferedReader bufferedReader = new
BufferedReader(new InputStreamReader(inputStream, "iso-8859-1"));
                                    String result = "";
                                    String line = "";
                                    while ((line =
bufferedReader.readLine()) != null) {
                                        result += line;
                                    }
                                    bufferedReader.close();
                                    inputStream.close();
                                    httpURLConnection.disconnect();
                                }catch(MalformedURLException e){
                                    e.printStackTrace();
                                }catch(IOException e){
                                    e.printStackTrace();
                                }
                            }
                            else if(result1.equals("exit1")){
                                runOnUiThread(new Runnable() {
                                    @Override
                                    public void run() {

Toast.makeText(getApplicationContext(),"Exiting room",Toast.LENGTH_LONG);
                                    }
                                });
                                client = new Socket("192.168.0.127", 4444);
                                printwriter = new
PrintWriter(client.getOutputStream());
                                printwriter.write(message3);
                                printwriter.flush();
                                printwriter.close();
                                client.close();

                                try {
                                    String sim_serial = getSimSerial();
                                    String login_url =
"http://192.168.0.127/webapp/jdbc.php";
                                    URL url = new URL(login_url);
                                    HttpURLConnection httpURLConnection =
(HttpURLConnection) url.openConnection();

httpURLConnection.setRequestMethod("POST");
                                    httpURLConnection.setDoOutput(true);
                                    httpURLConnection.setDoInput(true);
                                    OutputStream outputStream =
httpURLConnection.getOutputStream();
                                    BufferedWriter bufferedWriter = new
BufferedWriter(new OutputStreamWriter(outputStream, "UTF-8"));
                                    String post_data =
URLEncoder.encode("sim_serial", "UTF-8") + "=" +
URLEncoder.encode(sim_serial, "UTF-8");
                                    bufferedWriter.write(post_data);
                                    bufferedWriter.flush();
                                    bufferedWriter.close();
                                    outputStream.close();
                                    InputStream inputStream =
httpURLConnection.getInputStream();
                                    BufferedReader bufferedReader = new
BufferedReader(new InputStreamReader(inputStream, "iso-8859-1"));
                                    String result = "";
                                    String line = "";
                                    while ((line =
bufferedReader.readLine()) != null) {
                                        result += line;
                                    }
                                    bufferedReader.close();
                                    inputStream.close();
                                    httpURLConnection.disconnect();
                                }catch(MalformedURLException e){
                                    e.printStackTrace();
                                }catch(IOException e){
                                    e.printStackTrace();
                                }
                            }
                            else{
                                runOnUiThread(new Runnable() {
                                    @Override
                                    public void run() {

Toast.makeText(getApplicationContext(), "Incorrect entry, please scan
again", Toast.LENGTH_LONG).show();
                                    }
                                });
                            }
                        } catch (UnknownHostException e) {
                            e.printStackTrace();
                        } catch (IOException e) {
                            e.printStackTrace();
                        }
                    }


                }).start();


            }

        });


        alertDialogBuilder.setNegativeButton("No", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                Toast.makeText(getApplicationContext(), "Please scan again",
Toast.LENGTH_SHORT).show();

            }
        });

        AlertDialog alertDialog = alertDialogBuilder.create();
        alertDialog.show();
        mScannerView.resumeCameraPreview(this);

    }
}
```

activity_main.xml

```
<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:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MainActivity"
    android:id="@+id/myrl">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:ems="10"
        android:id="@+id/etPassword"
        android:layout_marginTop="27dp"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:hint="password" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login"
        android:id="@+id/btnLogin"
        android:onClick="OnLogin"
        android:layout_marginTop="27dp"
        android:foregroundTint="#15318d"
        android:background="#6ceec9"
        android:textColor="#ffffff"
        android:layout_below="@+id/etPassword"
        android:layout_alignLeft="@+id/cPass"
        android:layout_alignStart="@+id/cPass"
        android:layout_alignRight="@+id/cPass"
        android:layout_alignEnd="@+id/cPass" />

    <TextView
        android:id="@+id/cPass"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Change Passcode ? Click here"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:onClick="OnPass"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="First time? Click here to register passcode"
        android:id="@+id/passcode"
        android:layout_marginTop="27dp"
        android:onClick="passCode"
        android:layout_below="@+id/btnLogin"
        android:layout_centerHorizontal="true" />

    <Button
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Emergency"
        android:id="@+id/button3"
        android:onClick="emerge"
        android:layout_alignTop="@+id/cPass"
        android:layout_toRightOf="@+id/passcode"
        android:layout_toEndOf="@+id/passcode" />

</RelativeLayout>
```

passcode_new.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="Please key in a passcode"
        android:layout_marginTop="37dp"
        android:id="@+id/textView6"
        android:textColor="#0d0d0d" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:ems="10"
        android:id="@+id/passcode" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="Confirm passcode"
        android:layout_marginTop="42dp"
        android:id="@+id/textView5"
        android:textColor="#0b0b0b" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:ems="10"
        android:id="@+id/passcode1" />

    <Button
        android:layout_width="136dp"
        android:layout_height="wrap_content"
        android:text="Register"
        android:id="@+id/register"
        android:onClick="OnRegister"
        android:layout_gravity="center_horizontal"
        android:background="#0fc2ee"
        android:textColor="#ffffff" />
</LinearLayout>
```

pass_change.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="Please key in old password"
        android:id="@+id/textView"
        android:layout_marginTop="20dp"
        android:textColor="#090909" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:ems="10"
        android:id="@+id/pass1"
        android:layout_marginTop="15dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="Please key in new password"
        android:id="@+id/textView2"
        android:layout_marginTop="35dp"
        android:textColor="#0a0a0a" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:ems="10"
        android:id="@+id/pass2" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="Confirm new password"
        android:id="@+id/textView3"
        android:layout_marginTop="35dp"
        android:textColor="#0a0a0a" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:ems="10"
        android:id="@+id/pass3" />

    <Button
        android:layout_width="153dp"
        android:layout_height="wrap_content"
        android:text="Change Password"
        android:id="@+id/button4"
        android:layout_gravity="center_horizontal"
        android:onClick="OnChange"
        android:layout_marginTop="10dp"
        android:background="#0aa4eb"
        android:textColor="#ffffff" />
</LinearLayout>
```

qr_code.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginTop="27dp"
        android:text="Proceed to QR code scanning by tapping on the button
below"
        android:id="@+id/textView7"
        android:textColor="#0a0b0a" />

    <Button
        android:layout_width="140dp"
        android:layout_height="wrap_content"
        android:text="QR Scanner"
        android:id="@+id/button"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="150dp"
        android:onClick="QrScanner"
        android:background="#08ccf8"
        android:textColor="#ffffff" />

</LinearLayout>
```

APPENDIX C: Eclipse program code

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.PrintWriter;
import java.math.BigInteger;
import java.net.ServerSocket;
import java.net.Socket;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Calendar;
import java.util.Scanner;
import java.util.logging.ConsoleHandler;
import java.util.logging.FileHandler;
import java.util.logging.Handler;
import java.util.logging.Level;
import java.util.logging.Logger;
import com.embeddedunveiled.serial.SerialComManager;
import com.embeddedunveiled.serial.SerialComManager.BAUDRATE;
import com.embeddedunveiled.serial.SerialComManager.DATABITS;
import com.embeddedunveiled.serial.SerialComManager.FLOWCONTROL;
import com.embeddedunveiled.serial.SerialComManager.PARITY;
import com.embeddedunveiled.serial.SerialComManager.STOPBITS;
import com.mysql.jdbc.Statement;

public class testServer1 {

        private static Socket client;
        private static PrintWriter printwriter;
        private static String message = "open sesame";
        private static String name;
        private static String simmie;
        private static DataOutputStream dataout;

public static void main(String[] args) throws IOException {

        Socket clientSocket = null;
        Socket clientSocket1 = null;
        ServerSocket serverSocket = null;
```

```
    ServerSocket serverSocket1=null;
    try{

    serverSocket = new ServerSocket(4444);

    int y = 1;
    for(int x=1;y<10;x++){

    System.out.println("server listening....");
    clientSocket = serverSocket.accept();

    Scanner in1 = new Scanner(clientSocket.getInputStream());

    String mes="";
    int mes2;

        if(mes!=null){
            if (in1.hasNext())
            {
            mes=in1.nextLine();

                if(mes.equals("unlock")){

                    try {

                        Connection myConn =
DriverManager.getConnection("jdbc:mysql://localhost/employee101","root","");
                        Statement myStmt = (Statement)
myConn.createStatement();
                        ResultSet myRs =
myStmt.executeQuery("SELECT * FROM serial WHERE id like ('"+x+"')");


                            SerialComManager scm = new
SerialComManager();
                            long handle =
scm.openComPort("COM5", true, true, true);

    scm.configureComPortData(handle, DATABITS.DB_8, STOPBITS.SB_1,
PARITY.P_NONE, BAUDRATE.B115200, 0);

    scm.configureComPortControl(handle, FLOWCONTROL.NONE, 'x', 'x',
false, false);
                            System.out.println("opening");
                            scm.sendBreak(handle,3000);
                            System.out.println("closed");

                            while(myRs.next()){

    simmie=(myRs.getString("simmie"));
```

```
                                                            break;}

                                                     ResultSet myRs1 =
myStmt.executeQuery("SELECT * FROM employee_data WHERE sim like
('"+simmie+"')");

                                                     while(myRs1.next()){

        name=(myRs1.getString("name"));
                                                     break;}

                                                     Calendar calendar =
Calendar.getInstance();
                                                     java.sql.Timestamp
ourJavaTimestampObject = new java.sql.Timestamp(calendar.getTime().getTime());
                                                     String
sqlTimestampInsertStatement = "INSERT INTO dates
(timestamp2,location,user,status) VALUES (?,?,?,?)";
                                                PreparedStatement
preparedStatement = myConn.prepareStatement(sqlTimestampInsertStatement);
                                                preparedStatement.setTimestamp(1,
ourJavaTimestampObject);
                                                preparedStatement.setString(2,
"KB501");
                                               preparedStatement.setString(3, name);
                                                preparedStatement.setString(4, "In");

                                                preparedStatement.executeUpdate();
                                                preparedStatement.close();

                                                scm.closeComPort(handle);

                                      } catch (Exception e) {
                                              e.printStackTrace();
                                      }

                              }else if(mes.equals("unlock1")){

                                      try{
                                      client = new Socket("192.168.0.107", 4444);
                                      System.out.println("opening lock 2");
                                      printwriter = new
PrintWriter(client.getOutputStream());
             printwriter.write(message);
             printwriter.flush();
             printwriter.close();
             client.close();
```

```
          Connection myConn = 
DriverManager.getConnection("jdbc:mysql://localhost/employee101","root","");
                              Statement myStmt = (Statement) 
myConn.createStatement();
                              ResultSet myRs = 
myStmt.executeQuery("SELECT * FROM serial WHERE id like ('"+x+"')");


                                    while(myRs.next()){

      simmie=(myRs.getString("simmie"));
                                          break;}

                                    ResultSet myRs1 = 
myStmt.executeQuery("SELECT * FROM employee_data WHERE sim like 
('"+simmie+"')");


                                    while(myRs1.next()){
                                    name=(myRs1.getString("name"));
                                    break;}



                                    Calendar calendar = 
Calendar.getInstance();
                                    java.sql.Timestamp 
ourJavaTimestampObject = new java.sql.Timestamp(calendar.getTime().getTime());
                                    String sqlTimestampInsertStatement = 
"INSERT INTO dates (timestamp2,location,user,status) VALUES (?,?,?,?)";
                                PreparedStatement preparedStatement = 
myConn.prepareStatement(sqlTimestampInsertStatement);
                                preparedStatement.setTimestamp(1, 
ourJavaTimestampObject);
                                preparedStatement.setString(2, "KB502");
                                preparedStatement.setString(3, name);
                                preparedStatement.setString(4, "In");

                                preparedStatement.executeUpdate();
                                preparedStatement.close();
          }catch(Exception e){
            e.printStackTrace();
          }
                        }else if(mes.equals("exit")){
                              try{
                                    SerialComManager scm = new 
SerialComManager();
                                    long handle = 
scm.openComPort("COM5", true, true, true);
```

```
                                                  scm.configureComPortData(handle,
DATABITS.DB_8, STOPBITS.SB_1, PARITY.P_NONE, BAUDRATE.B115200,
0);
                                                  scm.configureComPortControl(handle,
FLOWCONTROL.NONE, 'x', 'x', false, false);
                                                  System.out.println("opening");
                                                  scm.sendBreak(handle,3000);
                                                  System.out.println("closed");

                                                  Connection myConn =
DriverManager.getConnection("jdbc:mysql://localhost/employee101","root","");
                                                  Statement myStmt = (Statement)
myConn.createStatement();
                                                  ResultSet myRs =
myStmt.executeQuery("SELECT * FROM serial WHERE id like ('"+x+"')");


                                                          while(myRs.next()){

        simmie=(myRs.getString("simmie"));
                                                                  break;}

                                                          ResultSet myRs1 =
myStmt.executeQuery("SELECT * FROM employee_data WHERE sim like
('"+simmie+"')");


                                                          while(myRs1.next()){

        name=(myRs1.getString("name"));
                                                          break;}


                                                          Calendar calendar =
Calendar.getInstance();
                                                          java.sql.Timestamp
ourJavaTimestampObject = new java.sql.Timestamp(calendar.getTime().getTime());
                                                          String
sqlTimestampInsertStatement = "INSERT INTO dates
(timestamp2,location,user,status) VALUES (?,?,?,?)";
                                                    PreparedStatement
preparedStatement = myConn.prepareStatement(sqlTimestampInsertStatement);
                                                    preparedStatement.setTimestamp(1,
ourJavaTimestampObject);
                                                    preparedStatement.setString(2,
"KB501");
                                                   preparedStatement.setString(3, name);
                                                    preparedStatement.setString(4,
"Out");
```

```
                                                    preparedStatement.executeUpdate();
                                                    preparedStatement.close();

                                                    scm.closeComPort(handle);

                                         }catch(Exception e){
                                                 e.printStackTrace();
                                         }
                                 }
                                 else if(mes.equals("exit1")){
                                         try{

                                                 client = new Socket("192.168.0.107",
4444);
                                                 System.out.println("opening lock 2");
                                                 printwriter = new
PrintWriter(client.getOutputStream());
                      printwriter.write(message);
                      printwriter.flush();
                      printwriter.close();
                      client.close();

                                                 Connection myConn =
DriverManager.getConnection("jdbc:mysql://localhost/employee101","root","");
                                                 Statement myStmt = (Statement)
myConn.createStatement();
                                                 ResultSet myRs =
myStmt.executeQuery("SELECT * FROM serial WHERE id like ('"+x+"')");


                                                         while(myRs.next()){

        simmie=(myRs.getString("simmie"));
                                                                 break;}

                                                         ResultSet myRs1 =
myStmt.executeQuery("SELECT * FROM employee_data WHERE sim like
('"+simmie+"')");




                                                         while(myRs1.next()){

        name=(myRs1.getString("name"));
                                                         break;}



                                                         Calendar calendar =
Calendar.getInstance();
```

```
                            java.sql.Timestamp
ourJavaTimestampObject = new java.sql.Timestamp(calendar.getTime().getTime());
                            String
sqlTimestampInsertStatement = "INSERT INTO dates
(timestamp2,location,user,status) VALUES (?,?,?,?)";
                          PreparedStatement
preparedStatement = myConn.prepareStatement(sqlTimestampInsertStatement);
                          preparedStatement.setTimestamp(1,
ourJavaTimestampObject);
                          preparedStatement.setString(2,
"KB502");
                         preparedStatement.setString(3, name);
                          preparedStatement.setString(4,
"Out");

                          preparedStatement.executeUpdate();
                          preparedStatement.close();
                    }catch(Exception e){
                        e.printStackTrace();
                    }
                }
            }

        }
    }

    }catch(Exception e){
        e.printStackTrace();

    }

    }

}
```