SMART DOOR LOCKS
BLUETOOTH USING
MOBILE ANDROID APPLICATION

ABDUL MUIZZ BIN ABDUL MAJID

BACHELOR OF COMPUTER SCIENCE

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	: ABDUL MUIZZ BIN ABDUL MAJID
Date of Birth	: 1 OCTOBER 1996
Title	: SMART DOOR LOCKS BLUETOOTH USING MOBILE ANDROID APPLICATION
Academic Session	: 2018/2019

I declare that this thesis is classified as:

□ CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
□ RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☑ OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature)

ABDUL MUIZZ BIN ABDUL MAJID
961001-06-5125
Date: 7 JANUARY 2019

(Supervisor's Signature)

DR. SALWANA MOHAMAD@ASMARA
SENIOR LECTURER
FACULTY OF COMPUTER SYSTEMS
& SOFTWARE ENGINEERING
UNIVERSITI MALAYSIA PAHANG
26300 GAMBANG, KUANTAN,
PAHANG
TEL: [illegible] FAX: 09-549 2144

DR SALWANA BINTI
MOHAMAD @ ASMARA
Date: 7 JANUARY 2019

Universiti
Malaysia
PAHANG

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis/project and in my/our opinion, this thesis/project is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Network Engineering)

(Supervisor's Signature)

Full Name : DR SALWANA BINTI MOHAMAD @ ASMARA

Position :Supervisor

Date : 7 JANUARY 2019

Universiti
Malaysia
PAHANG

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

(Student's Signature)

Full Name : ABDUL MUIZZ BIN ABDUL MAJID
ID Number : CA15101
Date : 7 JANUARY 2019

SMART DOOR LOCKS
BLUETOOTH USING
MOBILE ANDROID APPLICATION

ABDUL MUIZZ BIN ABDUL MAJID

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Computer Science (Computer Systems & Networking) with Honours

Faculty of Computer Science & Software Engineering

UNIVERSITI MALAYSIA PAHANG

DECEMBER 2018

ACKNOWLEDGEMENTS

Firstly, I would like thanks to Allah S.W.T for giving me his blessing to successfully accomplish my final year project entitled “Smart door lock using Bluetooth”. I would like to give honor and gratefulness to my supervisor, Dr. Salwana Binti Mohamad @ Asmara for being my supervisor, for being guider and great helper.

Next, I would like to thanks my parent for always being great supporter and for never ending loves. Not forgetting my friends for being by my side through the good and hard times.

Lastly, I would like to extend my gratefulness to other person who directly or indirectly involved with my final year project.

ABSTRAK

Setiap orang memerlukan tempat perlindungan untuk berlindung dari cuaca buruk serta ancaman daripada binatang dan pencuri. Oleh itu, adalah penting bagi setiap kediaman untuk memastikan keselamatan yang terbaik untuk rumah mereka. Projek yang dibangunkan adalah bertumpu mengenai keselamatan pintu. Projek ini sesuai untuk orang yang selalu menghadapi masalah terlupa di mana kedudukan kunci pintu mereka. Selain daripada itu, masalah yang biasanya dihadapi adalah terlupa untuk membawa kunci apabila keluar. Sekiranya pintu perlu dimasukkan nombor pin untuk membuka pintu, masih ada kemungkinan bagi pemilik kediaman untuk terlupa nombor pin mereka. Objektif projek ini adalah untuk menghasilkan kaedah alternatif untuk melaksanakan keselamatan di pintu. Akhir sekali, untuk menilai tahap penerimaan pengguna. Penyelesaian masalah ini adalah dengan cadangan aplikasi. Tujuan projek ini adalah untuk membina aplikasi android untuk mengunci atau membuka kunci pintu menggunakan Bluetooth melalui peranti pintar. Projek ini bertujuan untuk membangunkan aplikasi android dan mengajar pengguna tentang cara mengendalikan aplikasi ini. Permohonan ini digunakan untuk kediaman manusia. Kaedah ‘Rapid Application Development’ (RAD) dipilih dan dilaksanakan semasa proses permohonan. RAD terdiri daripada 4 fasa iaitu perancangan, reka bentuk dari pengguna, proses pembinaan, dan perlaksaan. Bahasa yang digunakan dalam aplikasi ini adalah bahasa Inggeris.

ABSTRACT

Everyone need a shelter to cover from bad weather and threats from animals and thief. Thus, it is important in every residence to implement high security for their home. The project development is focus on security of door. This application is suitable for people who always faced problem misplaced their door key. Other than that, the problem that usually faced nowadays is forgot to bring the key when go outside. If the door system need to enter the pin number to unlock the door, there still a chance for the owner to forgot their pin number. The objective of this project is to design an alternative method to implement security on the door. Next, to develop alternative technique that can be used for the security. Lastly, to evaluate user acceptance of the application. The solution of the problems is with the proposed application. The purpose of this project is to develop a mobile android application to lock or unlock the door using Bluetooth via smart device. This project is aiming to develop an interactive android mobile application of smart door and teach user on how to operate the application. This application is targeted for human residence. Rapid Application Development (RAD) methodology is chosen and implemented during the process of the application. RAD consist 4 phase which is planning requirements, user design, rapid construction, and cutover. The language used in this application is English language.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

EEP	Electrically Erasable Programmable
RAM	Random Access Memory
ROM	Read Only Memory
MAC	Media Access Control
RAD	Rapid Application Development
DSDM	Dynamic System Development Method
USB	Universal Serial Bus
LCD	Light Crystal Display
LED	Light Emitting Diode
V	Voltage
UAT	User Acceptance Test

CHAPTER 1

INTRODUCTION

1.1 Background of study

Every home there must be a door as the path of its occupants to enter or leave the home. Thus it is important for the owner the of house to keep the house is safe from the intruders. One way to implement security to the home is by installing lock on the door of house. There are many ways to implement security on the door such as gas sensor (Irimia R & Gottschling M 2016).

This project proposed an alternative way to implement security on door. The proposed system will use technique using lock with Bluetooth. People would not need any key anymore. There also will be a problem to place the keys or to carry it if people have many doors at their house. Furthermore, people could misplace the door key as well. Nowadays, everyone will bring their smartphone together anywhere they would go. Thus, this technique is suitable to implement as security on the door.

1.2 Problem statement

i. People nowadays always faced problem of misplaced their door keys. There might be more than one door in the house. So, there are many key and sometimes peoples may forget where they place the key after using it. When the key is missing. The owner need to change the lock or need help of locksmitch.

ii. People always forgot to bring the key when go outside. In order to enter the house without having the key is only by break the door. This situation might cost the owner of the house to fix it.

iii. If the door need to enter pin number to unlock the door, there still a chance for the owner to forgot their pin number. This problem usually occurs after the owner renew the pin number after period of time.

1.3 Goal

The goal of this project is to develop the security system for door access by using Bluetooth that available on smart phone.

1.4 Objective

In order to achieve project goal, few subjective must be fulfilled. The objective of this project are:

i. To design a safer method to implement security on the door.

ii. To develop the best technique can be that can be used for the security.

iii. To evaluate user acceptance of the application.

1.5 Scope

The study is for target user:

i. Public human residence. Public human residence must have room or house that need to secure from thieve.

ii. Android user. This system uses an android application for the end-user.

1.6 Significance

This project focus on:

i. Prevent the house from burglary, break in and robbed.

ii. Help peoples to save money and space for key hanging places.

iii. The security of the residence is guaranteed.

1.7 Thesis organization

This thesis consists of five chapters.

i. Chapter 1 discuss on the introduction of the project. This chapter describe about the background, problem statement, objective and scope of study.

ii. Chapter 2 is about literature review. This chapter discuss about the comparison between existing application and proposed system.

iii. Chapter 3 is about methodology. This chapter will discuss about methodology will be used for development process.

iv. Chapter 4 will discuss implementation and result. This chapter will display the interface of propose system and result analysis on the application.

v. Chapter 5 is about conclusion. This chapter will come out with summary of the system proposed.

CHAPTER 2

LTERATURE REVIEW

2.1 Introduction

This chapter explains in detail about the existing applications that related to Bluetooth security door, which is suitable to be adapted in related development. The purpose of this chapter is to identify and analyze the feature and concept of the existing applications. Based on this comparison, it will be applied on the application that will be develop.

2.2 Review of existing application

This section explains on the analysis of three existing applications that can be obtained via online and application store. The chosen applications are Arduino Control Door Lock, Yale Bluetooth control door and Easy access. The finding feature, method and technology used to operate the application are all concluded in section 2.3.

2.2.1 Control Door Lock

Arduino Control Door Lock is an application that develop by MD Khairul, A, T. (2015). This application was published in September 23, 2015. The latest version of this application is version 1.0. Arduino Control Door Lock is an application that allow user to unlock a security door that using Bluetooth technique security. This application will be install in device such smart phones and performing as virtual key to access the door.

In this application, the user will be given several options in home screen which are pair device, connect device, exit app, enter password, open door and close door as shown in figure 2.1.

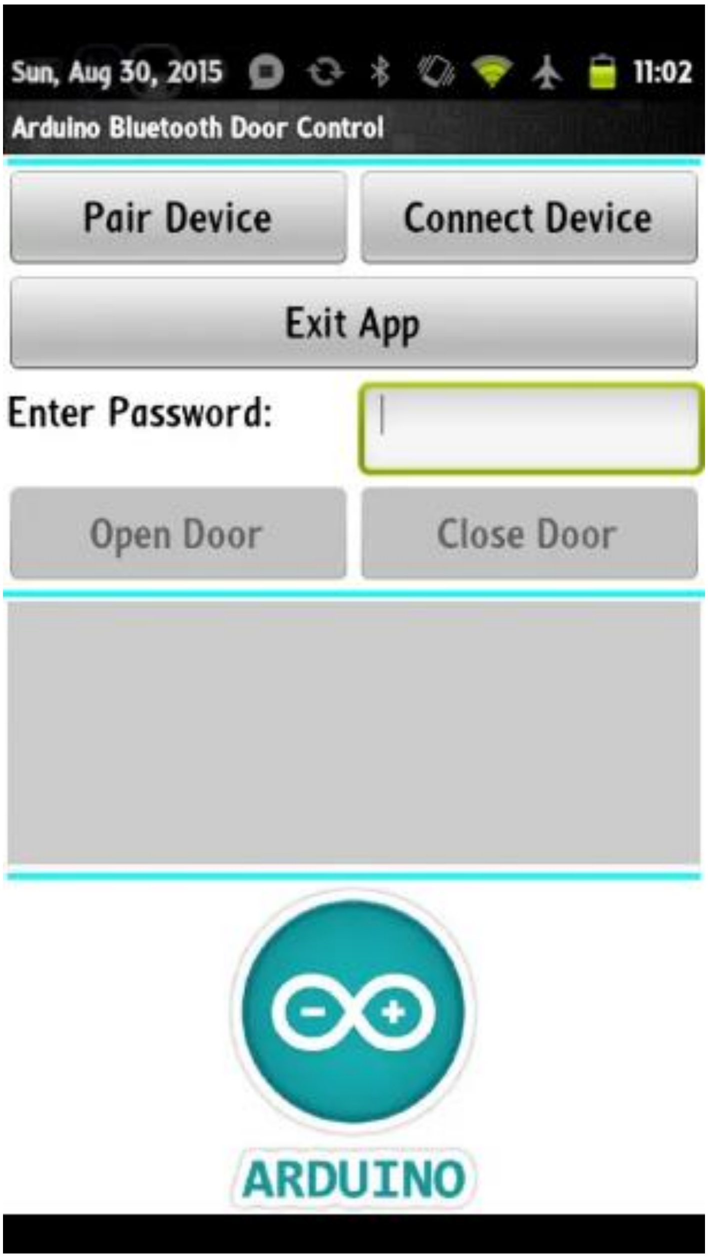
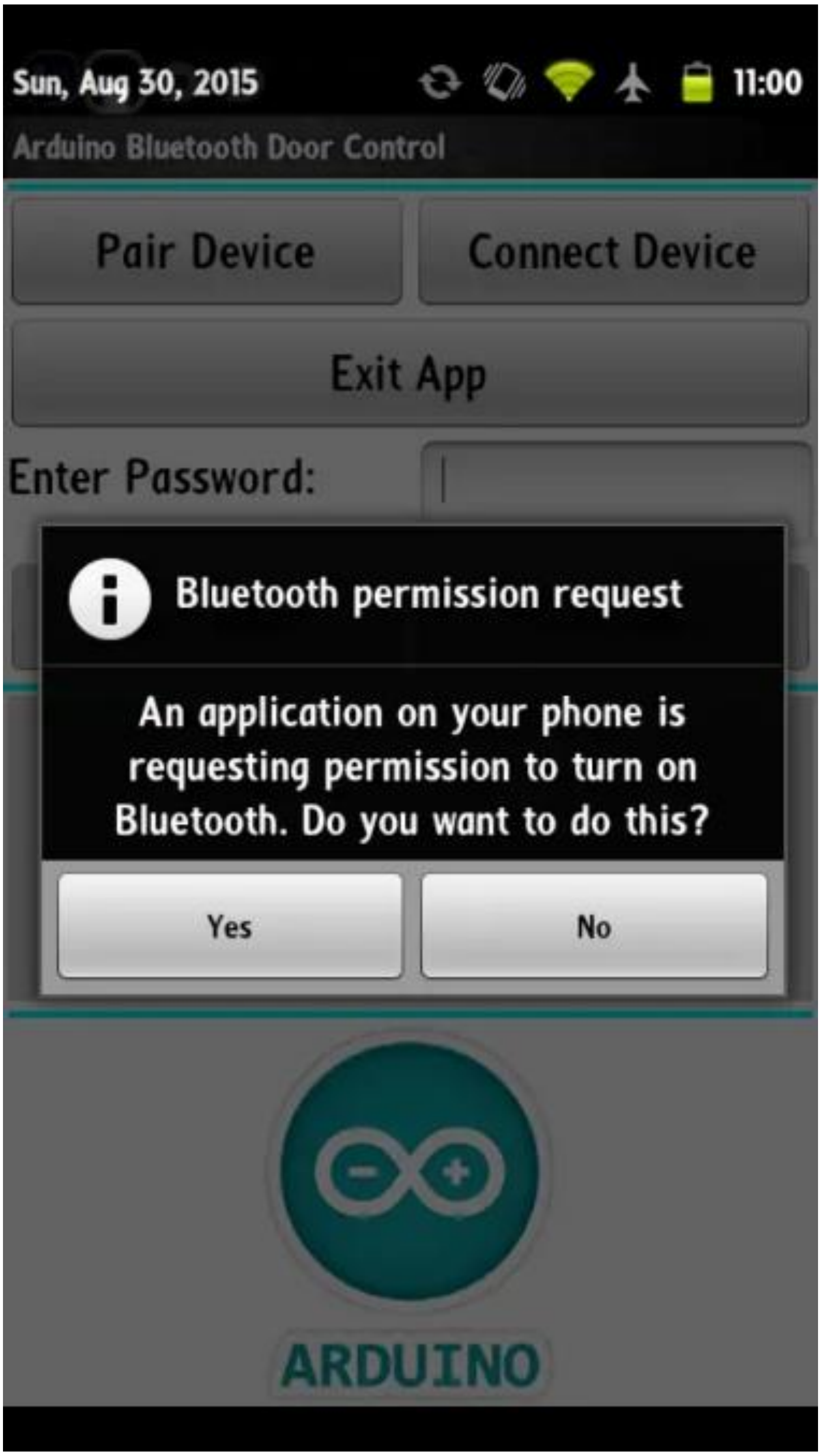


Figure 2.1: Home screen

In order the application start working, the device need to turned on Bluetooth first. If the Bluetooth is not turned on yet, a pop up massage will appear in the screen and request permission to turn on the Bluetooth as shown in figure 2.2.



Bluetooth permission request

An application on your phone is requesting permission to turn on Bluetooth. Do you want to do this?

Figure 2.2: request permission

In connect device button as shown in figure 2.1, it will connect the device and the door via Bluetooth. Without the connection between the device and the Bluetooth door, the application will not operate and functioning at all.

In the middle of main interface as shown in figure 2.3, there is navigation button that is exit app button. After the user done lock or unlock the door, the user can exit from the application by clicking the exit button.

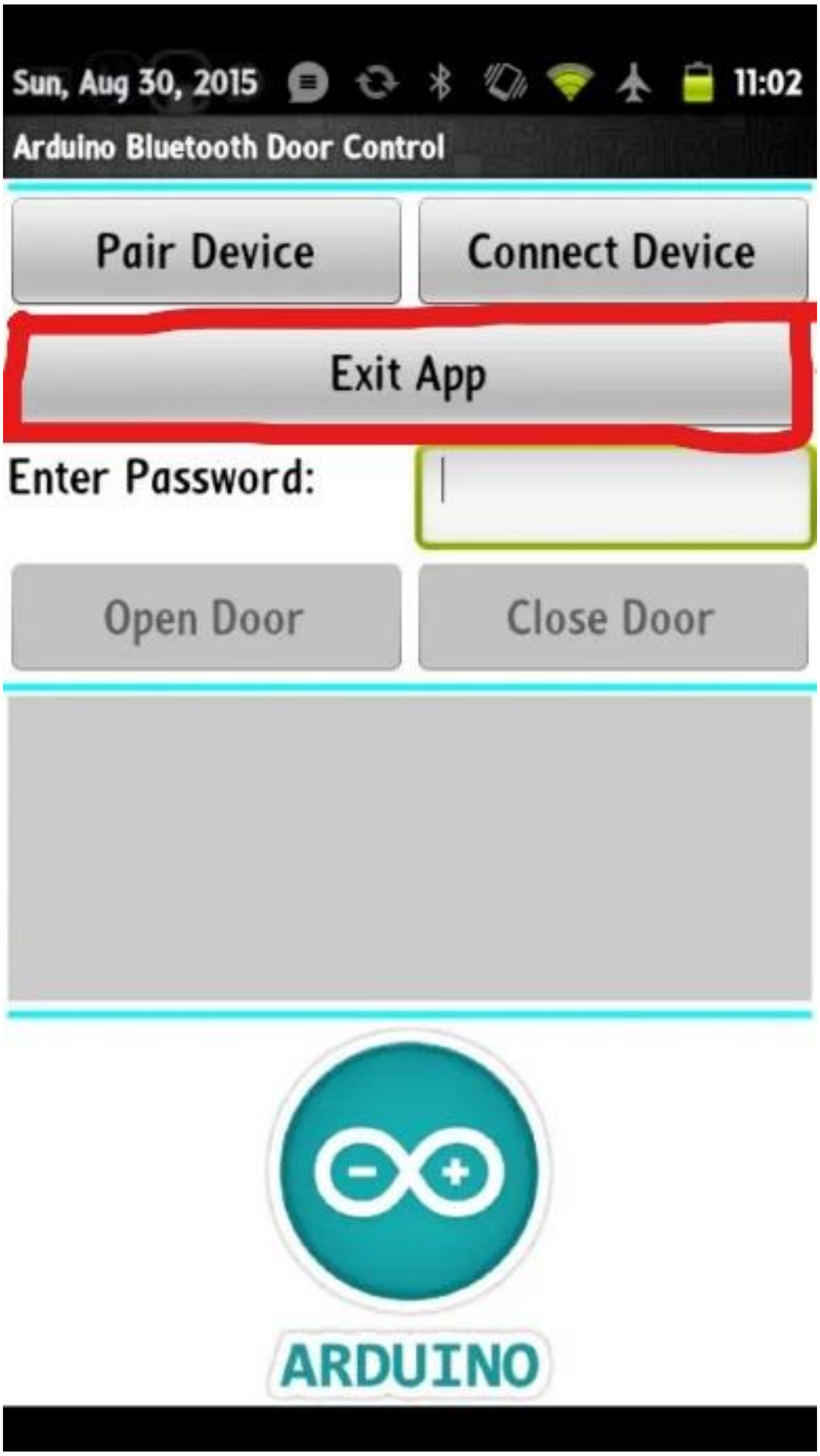


Figure 2.3: Exit App button

2.2.2 YALE Bluetooth Lock

YALE Bluetooth lock is another door lock using Bluetooth technique that available in google play. The developer of this application is ASSA ABLOY iRevo inc. this application was published in January 31, 2016. The latest version of this application is version 1.0.7. In this application, the number of door that wants to be lock or unlock is unlimited. User can use this app if there are many Bluetooth doors in the house.

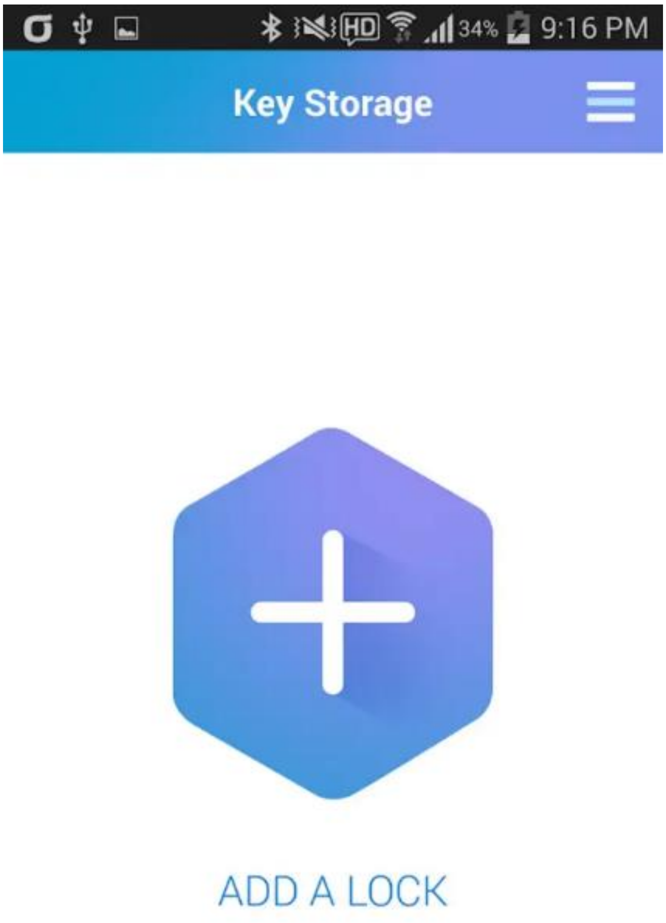
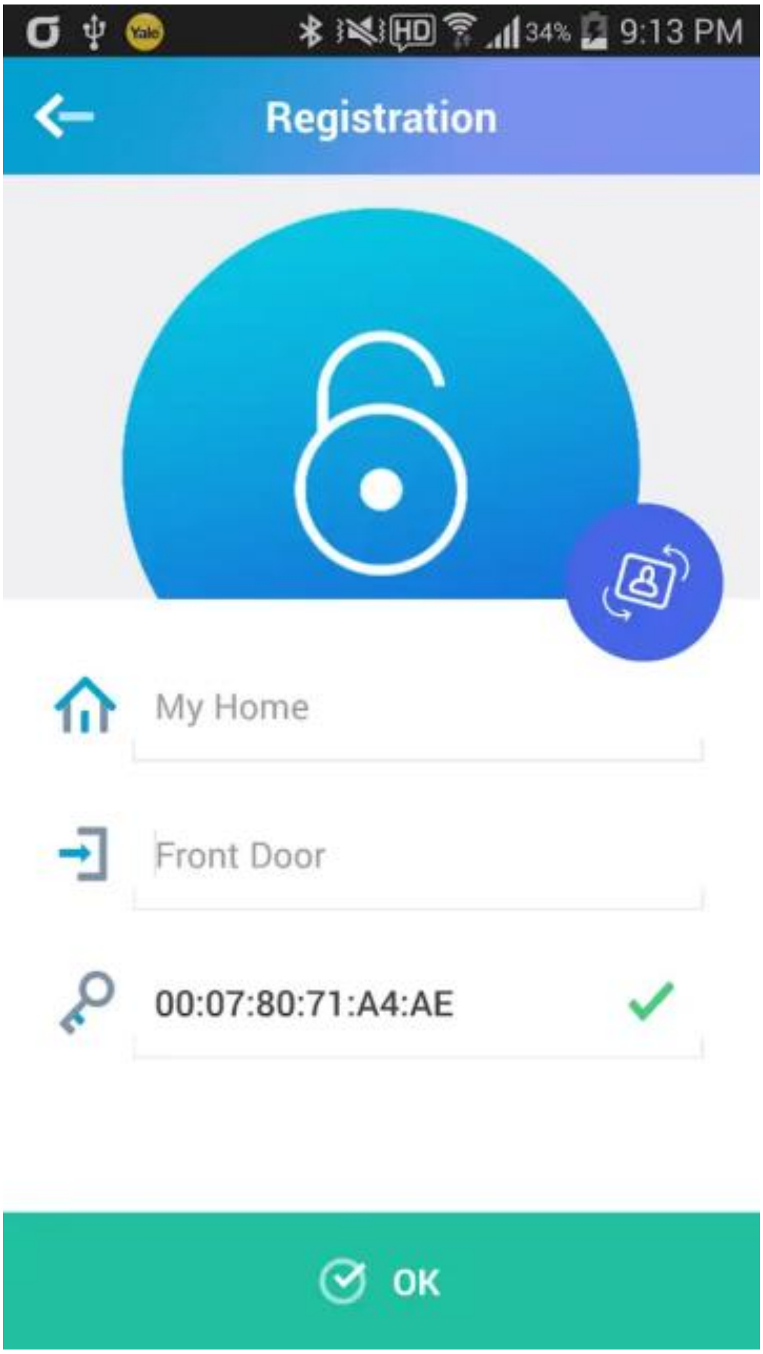


Figure 2.4: Home screen interface

Figure 2.4 shows home screen interface where user can add many lock in this application by touching the plus symbol on the middle of interface.



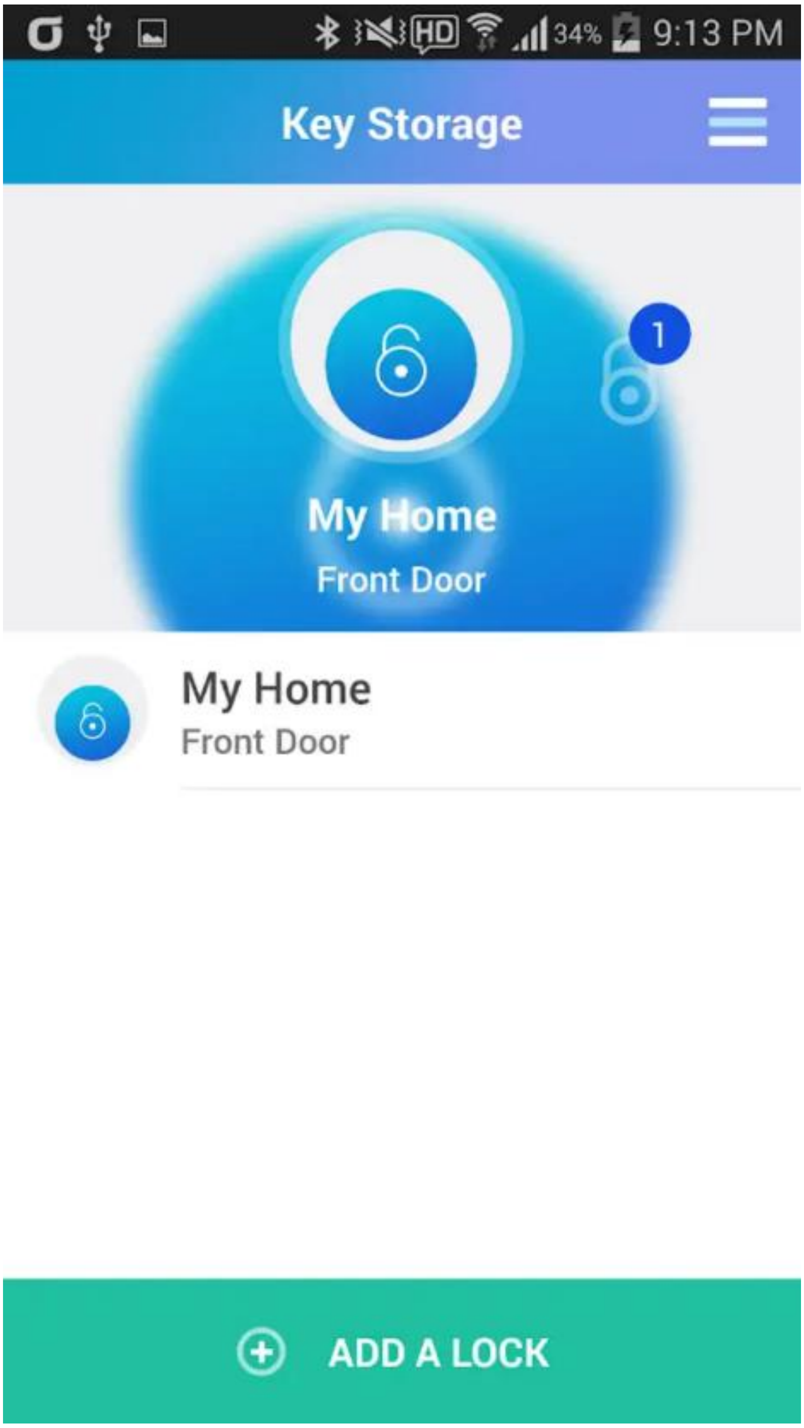
My Home

Front Door

00:07:80:71:A4:AE ✓

Figure 2.5: Door Name interface

YALE Bluetooth application also provide features to the user that enable user for naming each door as shows in figure 2.5.


9:13 PM

Key Storage

My Home

Front Door

My Home

Front Door

ADD A LOCK

Figure 2.6: Front Door interface

Figure 2.6 show the application save the changes made by the user after naming the door. This is the example of interface after the changes made.

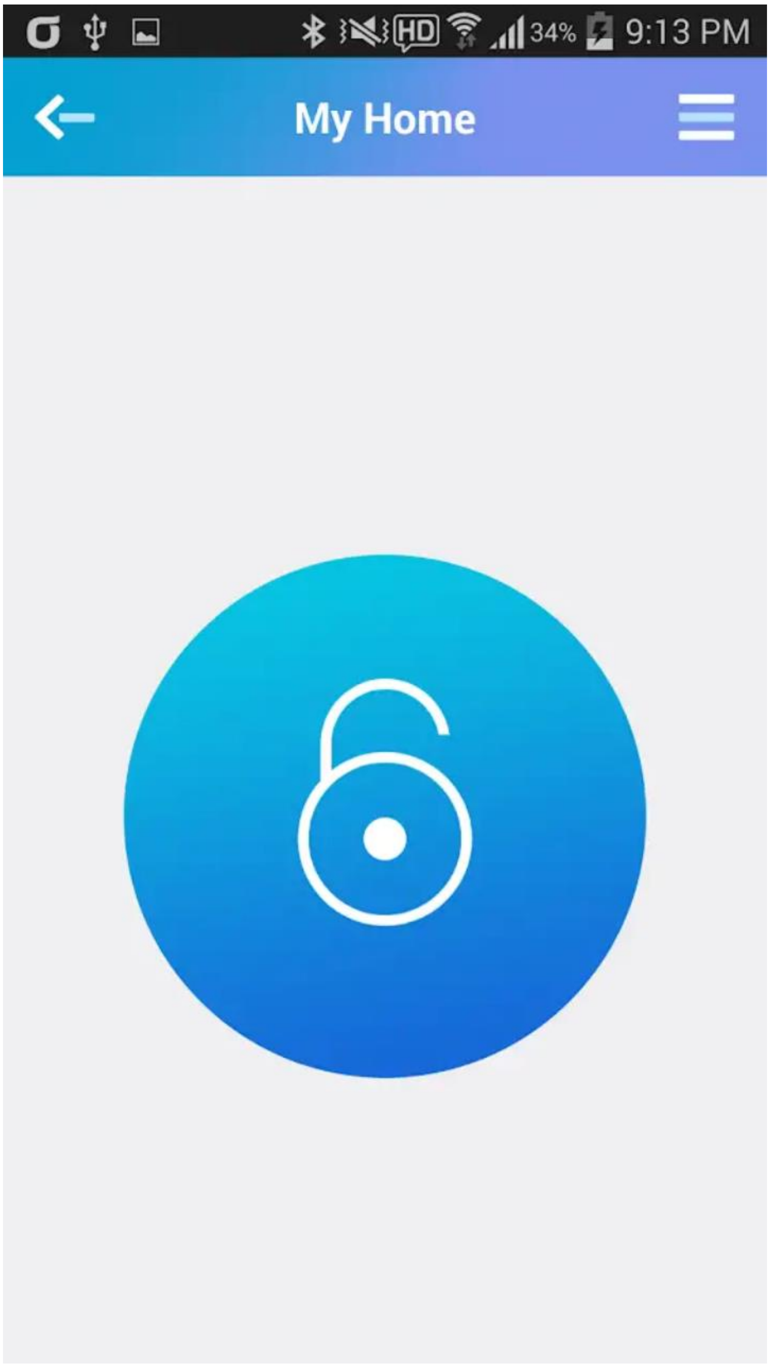


Figure 2.7: Unlock state interface

Figure 2.7 show when the door is in unlock state, the application will display blue lock. Thus, the users are aware with the state of the door without seeing the lock from the back of door. To lock the door, user just need to touch the lock symbol and the symbol will change from blue lock to red lock symbol.

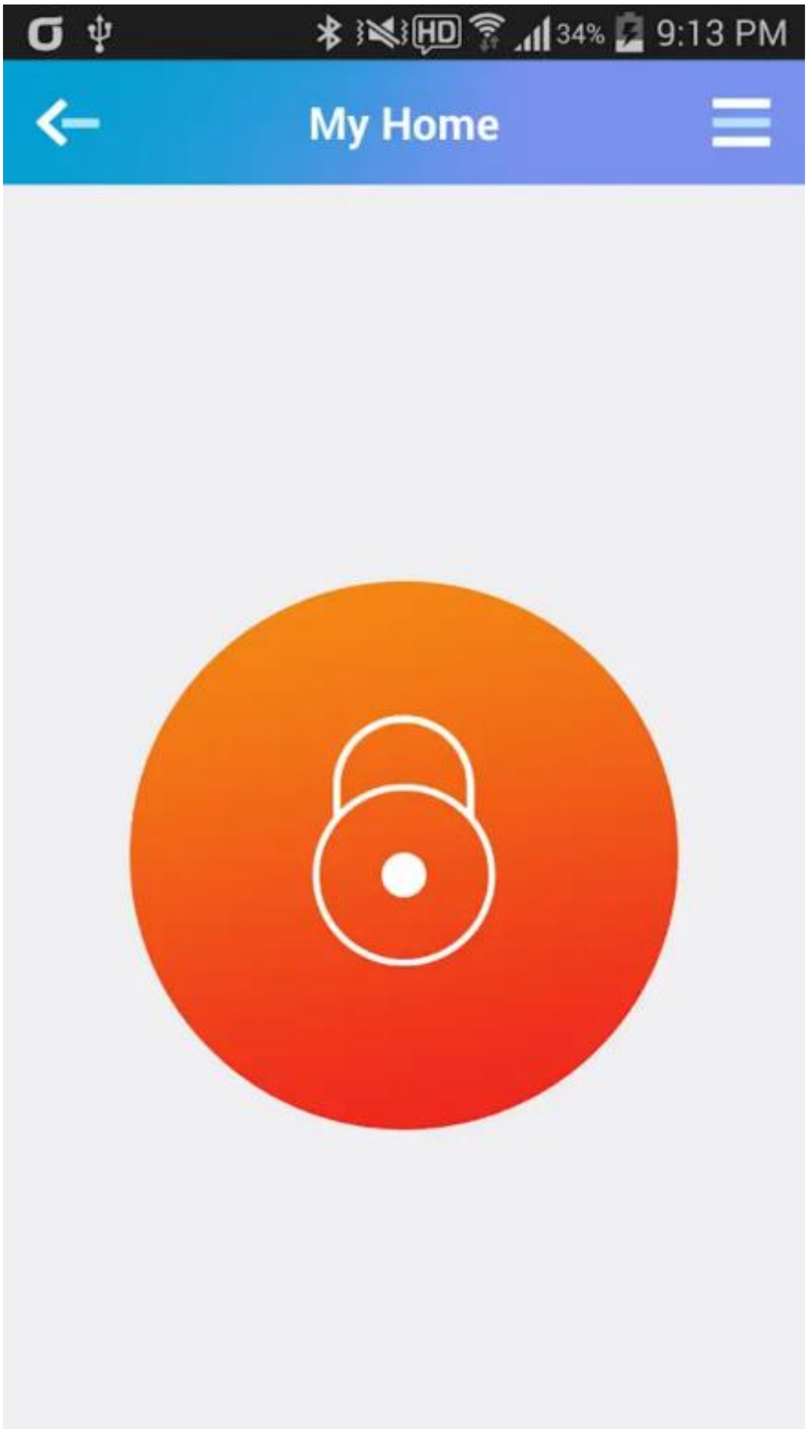
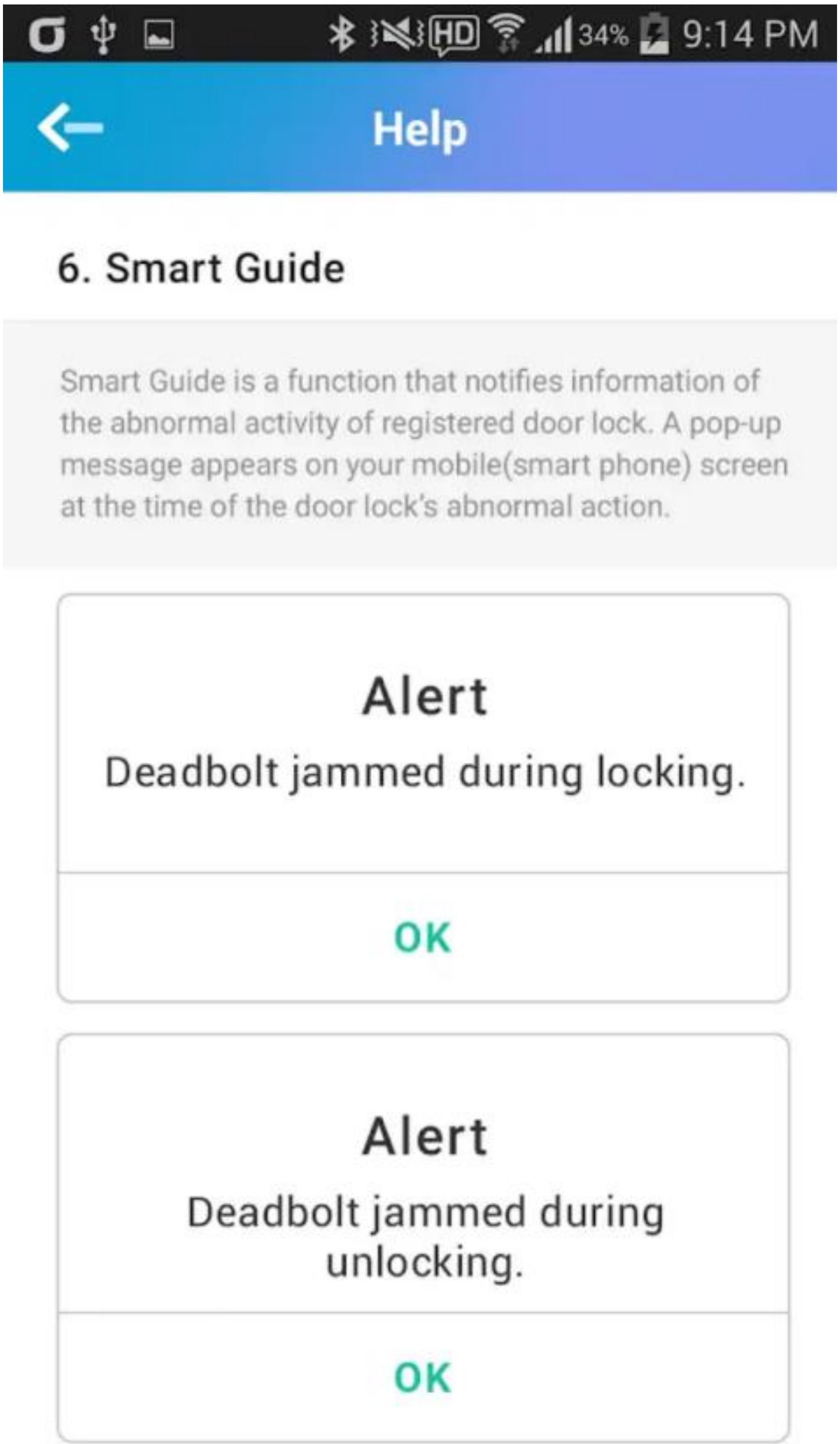


Figure 2.8: Lock state interface

Figure 2.8 shows the red lock symbol will appear in the interface when the door is in lock state as shown in figure 2.8.



6. Smart Guide

Smart Guide is a function that notifies information of the abnormal activity of registered door lock. A pop-up message appears on your mobile(smart phone) screen at the time of the door lock's abnormal action.

Alert

Deadbolt jammed during locking.

Alert

Deadbolt jammed during unlocking.

Figure 2.9: Alert massage interface

Figure 2.9 shows alert massage will pop up through application to notify the user if there are any door problem. By having this, users are aware with the current situation.

2.2.3 Easy key

Easy key is another technique to lock or unlock a door using application connected via Bluetooth. The developer of this application is Unicorn Devices Labs. This application was published in October 5, 2017. The information of latest version this application is not provided. Easy key application is used for gate of the house. The device need to pair with the car Bluetooth.

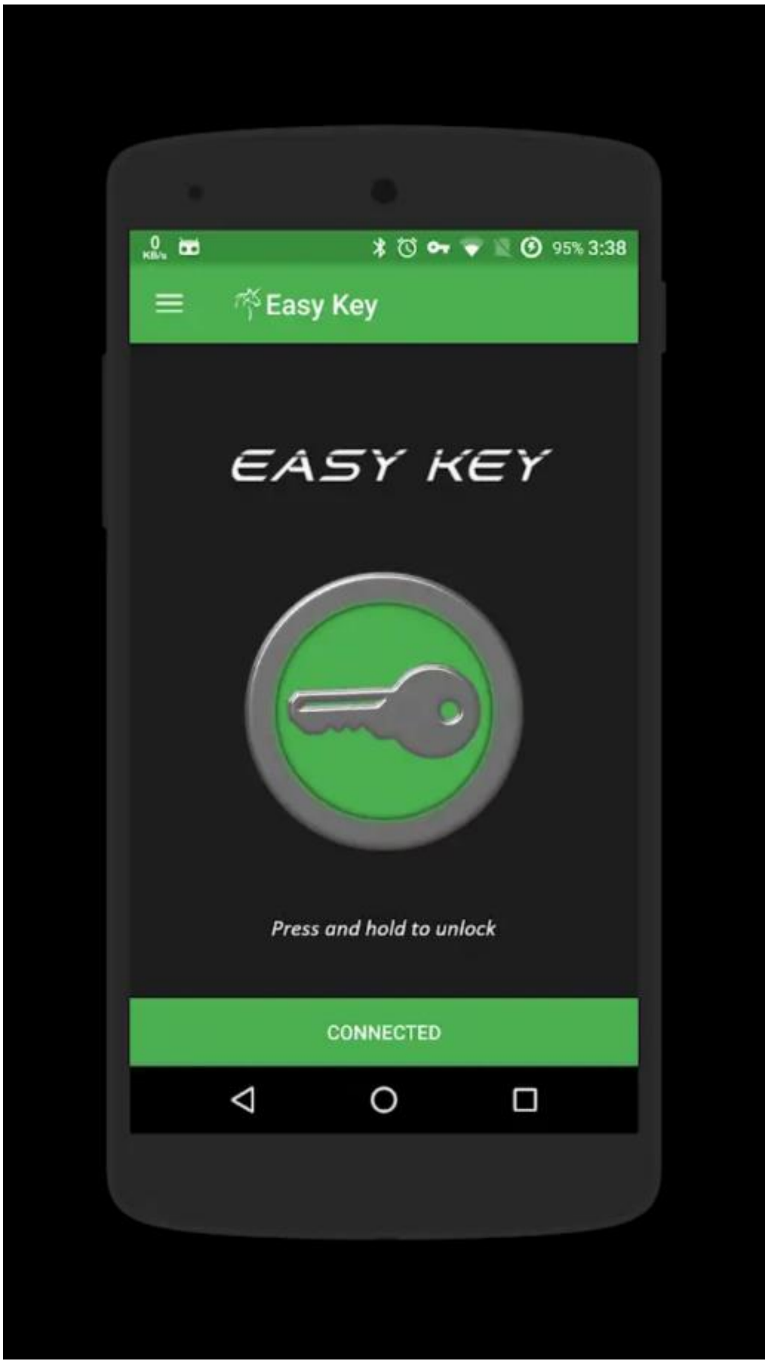


Figure 2.10: Main Interface

Figure 2.10 is the main interface of EASY KEY. To unlock the gate, user need to hold on key symbol for few second.

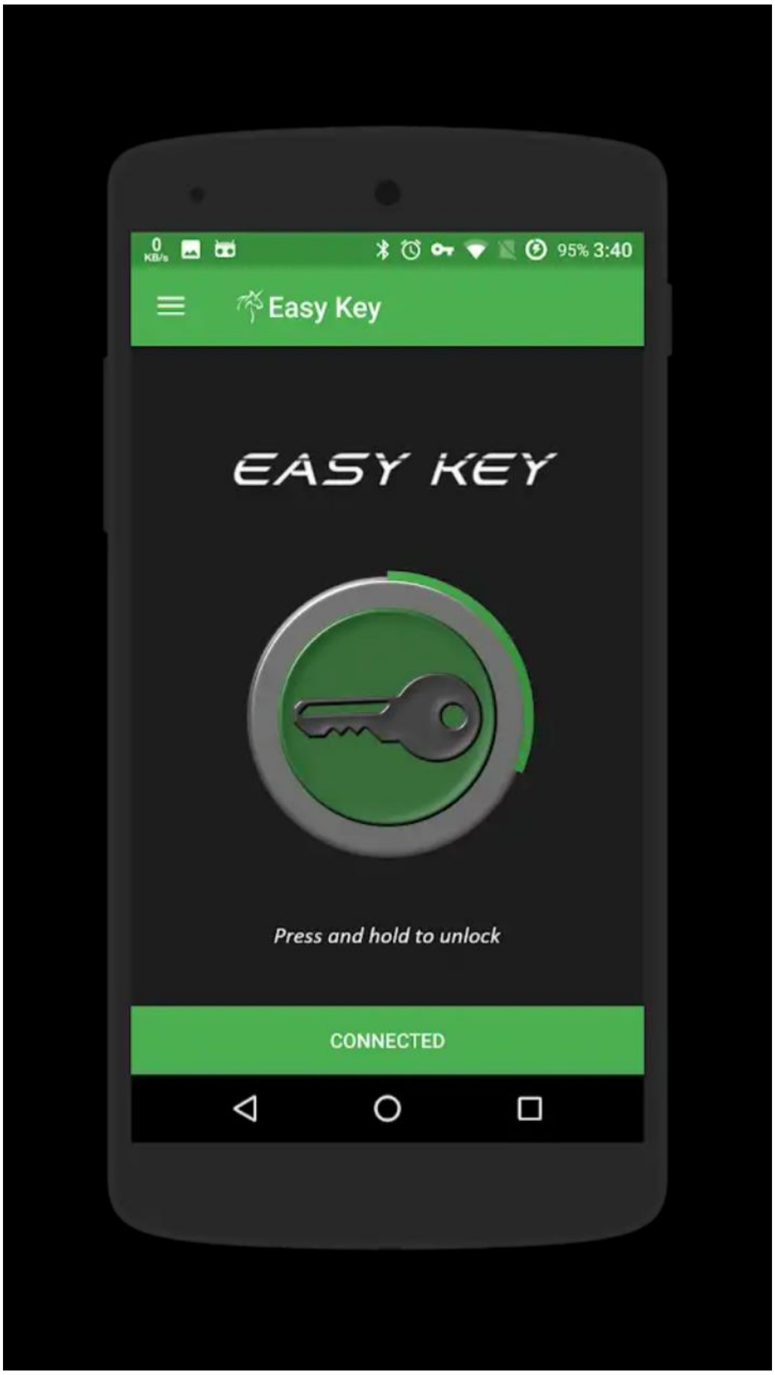
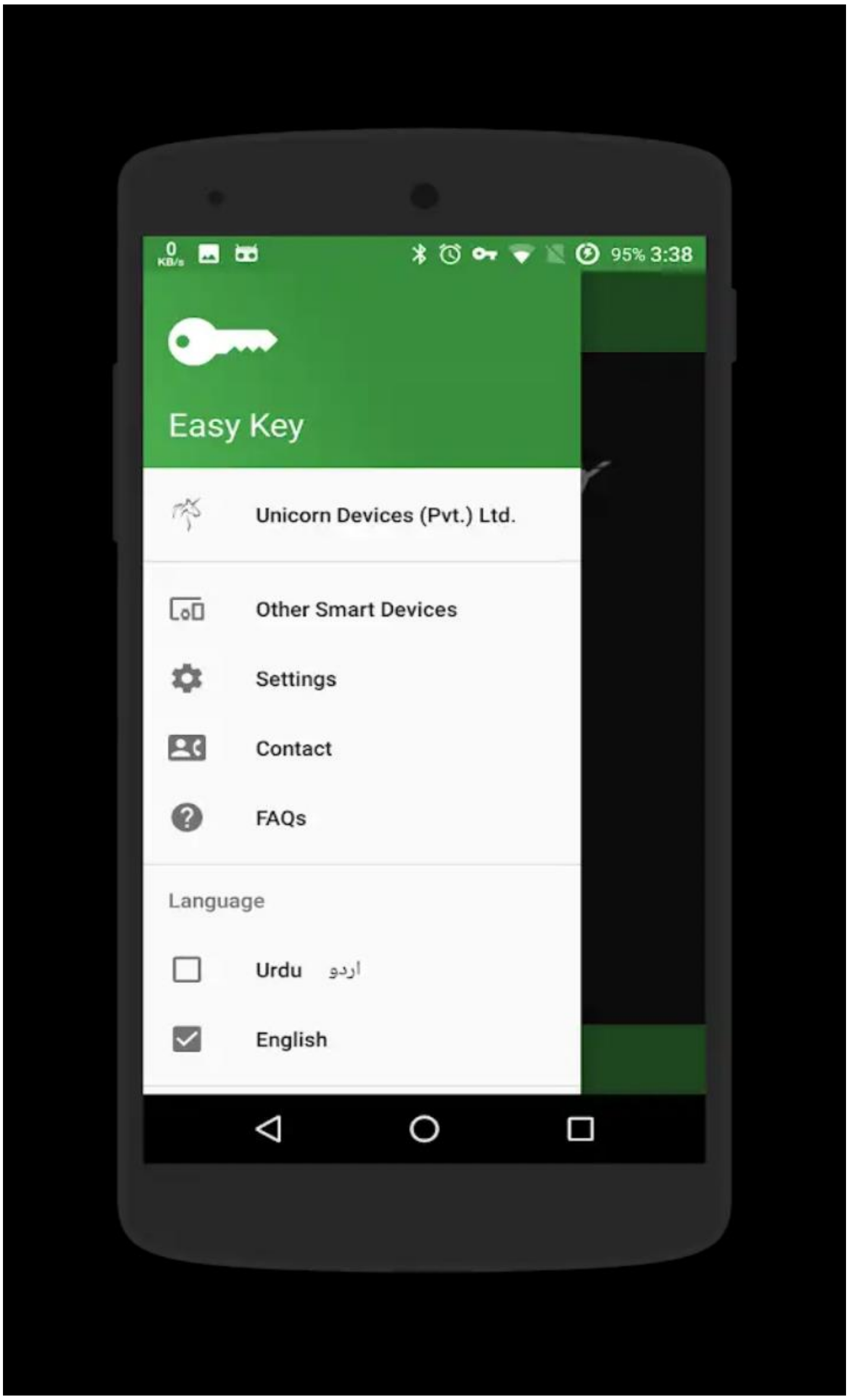


Figure 2.11: Lock gate interface

Figure 2.11 will appear when the user is holding on key symbol.



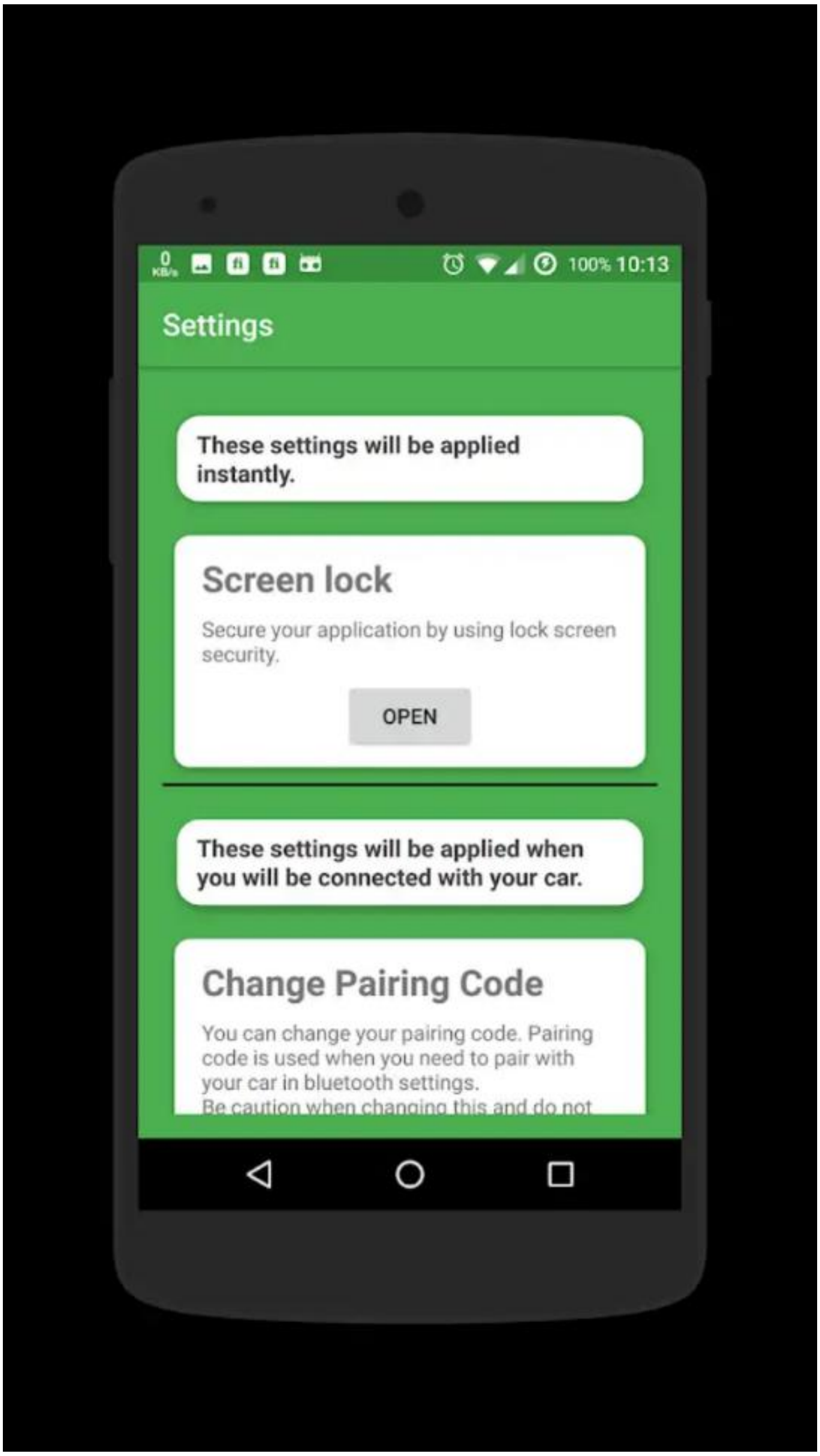
Language

☐ Urdu اردو

☑ English

Figure 2.12: setting interface

Figure 2.12 shows there is Hamburger icon for setting features on top left interface. In Hamburger icon, users are able to choose the language the user want such as English, Indonesia, and Arabic. The result is many users from different country with different language can use this application.



Settings

These settings will be applied instantly.

Screen lock

Secure your application by using lock screen security.

These settings will be applied when you will be connected with your car.

Change Pairing Code

You can change your pairing code. Pairing code is used when you need to pair with your car in bluetooth settings.
Be caution when changing this and do not

Figure 2.13: Extended setting interface

Figure 2.13 shows the extended setting available in the Easy Key application. The details of setting also provided in this interface.

2.3 Table

Table 2.1 shows the comparison of Arduino Control Door Lock, Yale Bluetooth key, Easy Key and proposed system.

Table 2.1 Comparison of existing application of door lock

Features	Arduino Control Door Lock	Yale Bluetooth key	Easy key	Proposed system
Graphical User Interface	Interface too simple and not interactive	Interface is simple but attractive	Interface is simple but attractive	Interface is simple and attractive
Navigation User Interface	Not provided	Good navigation system	Good navigation system	Good navigation system
Sources can be download	Google Play store	Google Play store	Google Play store	Google play store
Device/technology	Arduino, Bluetooth	Bluetooth	Bluetooth	Arduino, Bluetooth
Security	Password	None	None	Embedded MAC Address

Proposed system have simple and attractive graphical user interface. Besides that, proposed system also have good navigation system. The device that is being used for proposed system is Arduino and the technology used are Bluetooth. For security of the proposed application, correct MAC Address is embedded in the application. Thus, with the correct MAC Address will established connection between the device and application. All the applications have simple user interface. But, Yale Bluetooth key, Easy key have more attractive interface. Next, all the applications have good navigation button except for Arduino control door lock. All the applications can be download from the play store. Other than that, all the application is connected via Bluetooth. All the application has no security cover except for Arduino control door lock. Thus, proposed application have better security compare to others application.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, the methodology of developing mobile application for door lock security using Bluetooth technique is discussed. There are many types of methodology used to develop a system such as waterfall methodology, Dynamic System Development Method (DSDM) methodology and agile methodology. For this system, the Rapid Application Development (RAD) is chosen. The detail explanation on method, technique, hardware and software used are provide. The explanation on why Rapid Application Development (RAD) are being chose is discussed.

3.2 Methodology

This section describes the detail of the methodology that used in this project. Choose an accurate methodology is the important process because it shows the processes how the application is developed. RAD methodology is chooseen because, the development process is rapid. As the development is still ongoing, users are able to add new requirement as they start to visualize the project. Thus, the requirement can be added to the project before the development is completed.

3.3 Rapid Application Development (RAD)

RAD is one of a form Agile methodology. Unlike waterfall methodology, RAD method focus on working software and user feedback over requirements recording and planning. RAD method emphasizes on rapid prototyping and iterative delivery (Kelsie Anderson 2017). Difference with typical waterfall development which is focuses on planning and sequential design practices. There are four basic steps in RAD method.

Which is planning requirements, user design, rapid construction, and cutover (Andrew powell 2017).

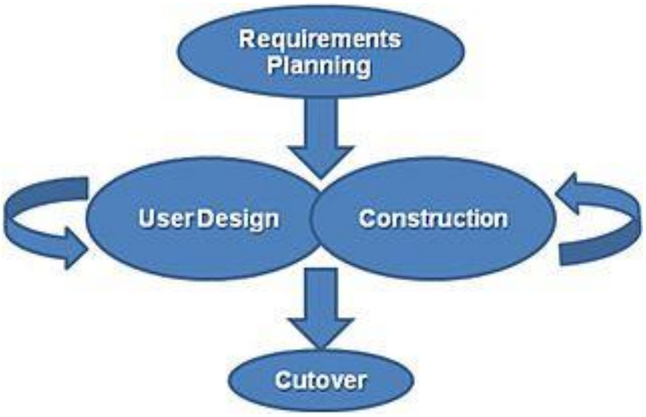


Figure 3.1: Phases in RAD method

3.3.1 Planning requirements

This is the initial step in RAD method. System analyst will meet users to get the user requirement on the proposed system. System analyst will identify and capture on every details that given by the customer. The details will come to the rough agreement on scope of system and application requirements. In this project, requirement needed for the application is gather by meeting with user.

3.3.2 User design

User feedback information is gathered and analyzed. Based on the analysis have been done, system architecture can be determined. Software and hardware needed for the proposed system also determined in this phase. This phase allows designers and engineer for created initial modelling and prototypes. After the user requirement is done, the planning to develop the application is start. Software and hardware that are needed are listed in order to build the application and start designing the interface of application.

3.3.3 Rapid construction

In this phase, prototype developed will be test and improvise in various of time. This is the phase where application coding was created and insert into the system and be test. Integration of the system will take place in this phase. The system will be upgraded until the system meet with the user requirements. New related requirements can be added by users as they have seen the prototype system is on the process of building. During the progress of building application, the progress of the project is shown to the user in order to ensure the application development follows user requirements.

3.3.4 Cutover

Cutover is the final stage in RAD method. Cutover stage allows development team to test the components into live production environment. This is necessary for full scale testing of proposed system. The system will be documented and instruction on how to use the system will design. After the application is complete in building, the application is testing for various number of time.

3.4 Use case diagram

Figure 3.2 is a use case diagram for proposed application. This use case diagram explains the functionality of the application. There are two actors in this application which is user and admin. User need to turn on Bluetooth and paired the device with the door. Once the connection is established, the application will notify user about the connection. If the connection is failed, admin can know the problem by checking the notification. Then, user able to lock or unlock the door.

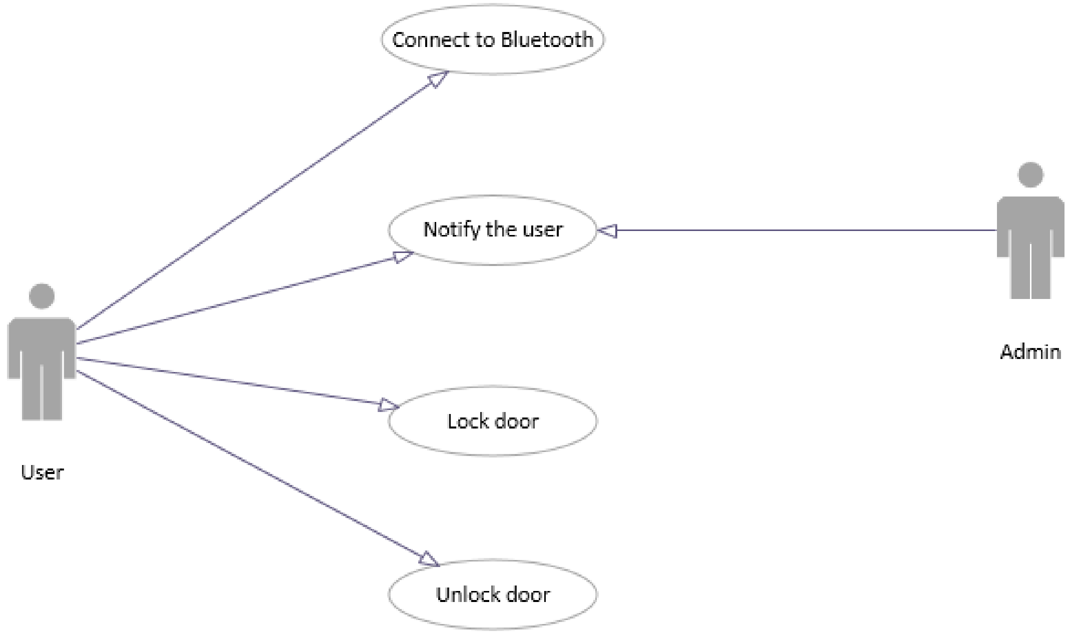


Figure 3.2: Use case diagram of proposed application

3.5 Sequence diagram

Figure 3.3 shows the sequence diagram for proposed system. This sequence diagram shows the interaction between the actor and the proposed application. It shows the environment of the application and relation between the actor and proposed application. When the user opens the application, the interface will display main interface to the user. User can choose from the menu to paired the device with the door. And the door will display the connectivity. When the connection is established, user can lock or unlock the door by pressing button on the interface. Every transition phase, the application will notify the user. From this notification, admin able to troubleshoot if any problem occurs with the apps.

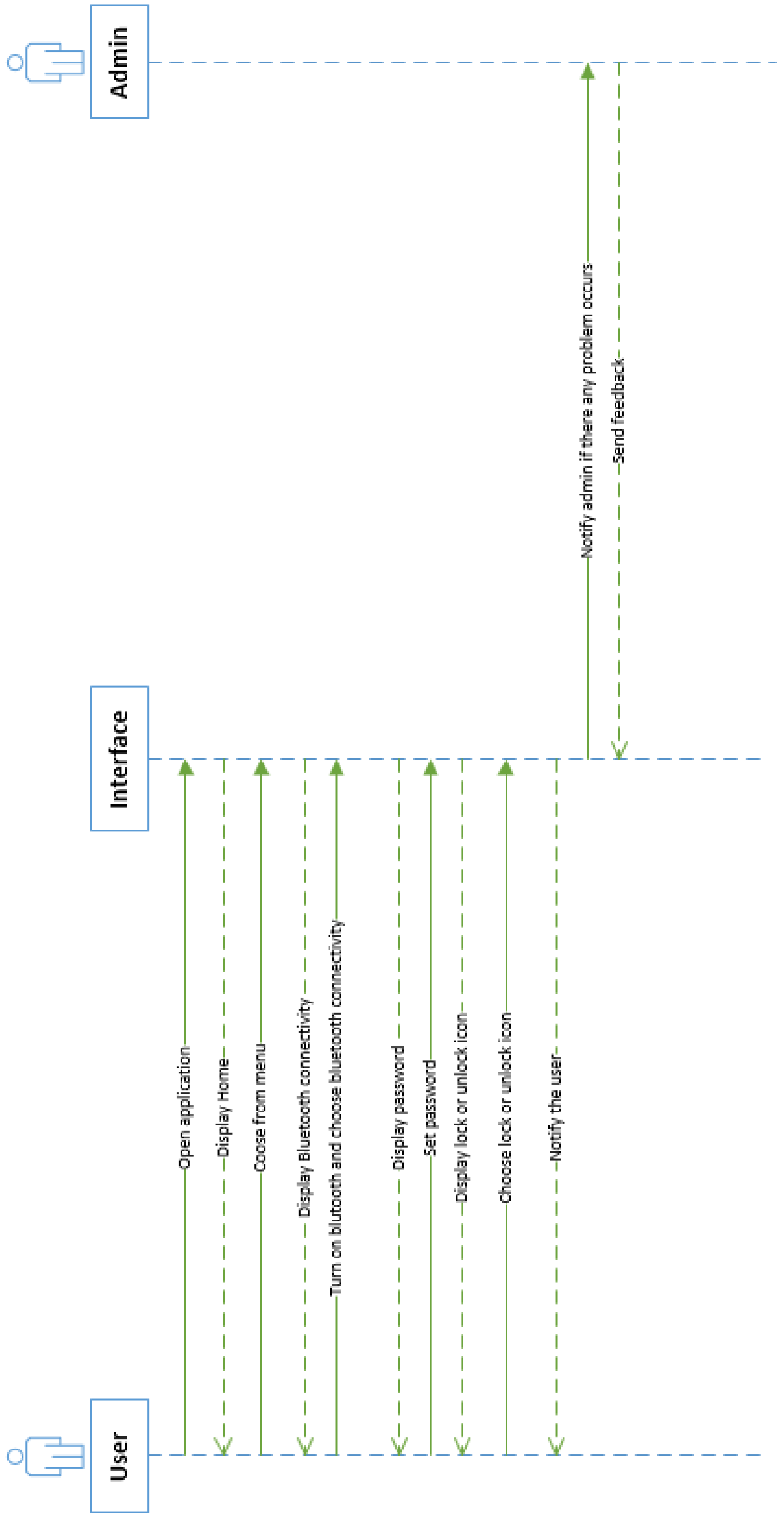

Admin
Interface
User
Open application
Display Home
Coose from menu
Display Bluetooth connectivity
Turn on blutooth and choose bluetooth connectivity
Display password
Set password
Display lock or unlock icon
Choose lock or unlock icon
Notify the user
Notify admin if there any problem occurs
Send feedback

Figure 3.3: Sequence diagram for proposed application

3.6 Flowchart diagram

The flowchart shows the overall happen in proposed application. This will help user to identify the flow chart of the system clearly. Figure 3.4 shows the flowchart for Bluetooth connection in proposed application.

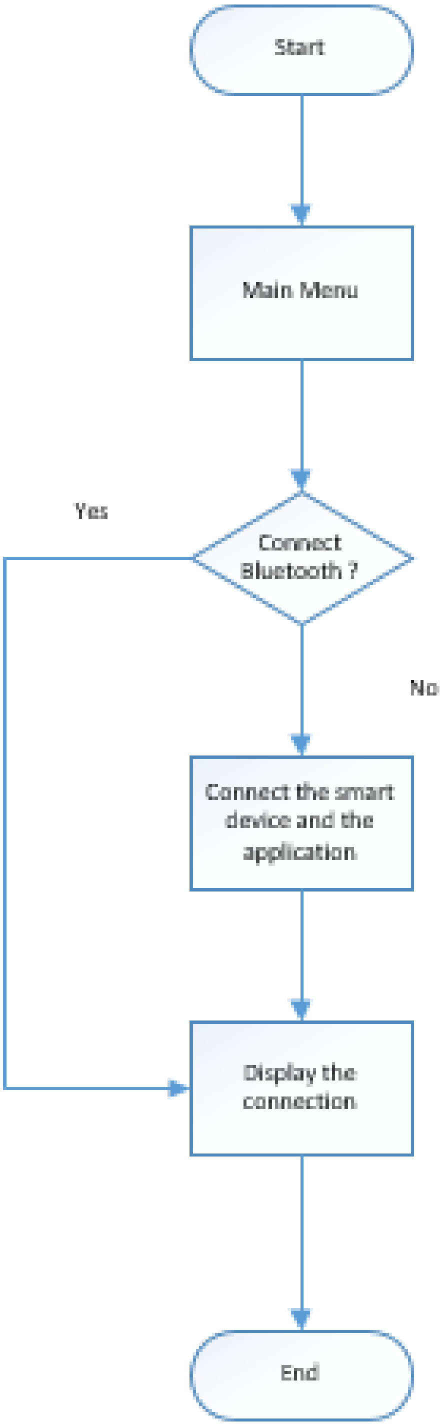


Figure 3.4: Flowchart for Bluetooth connection menu

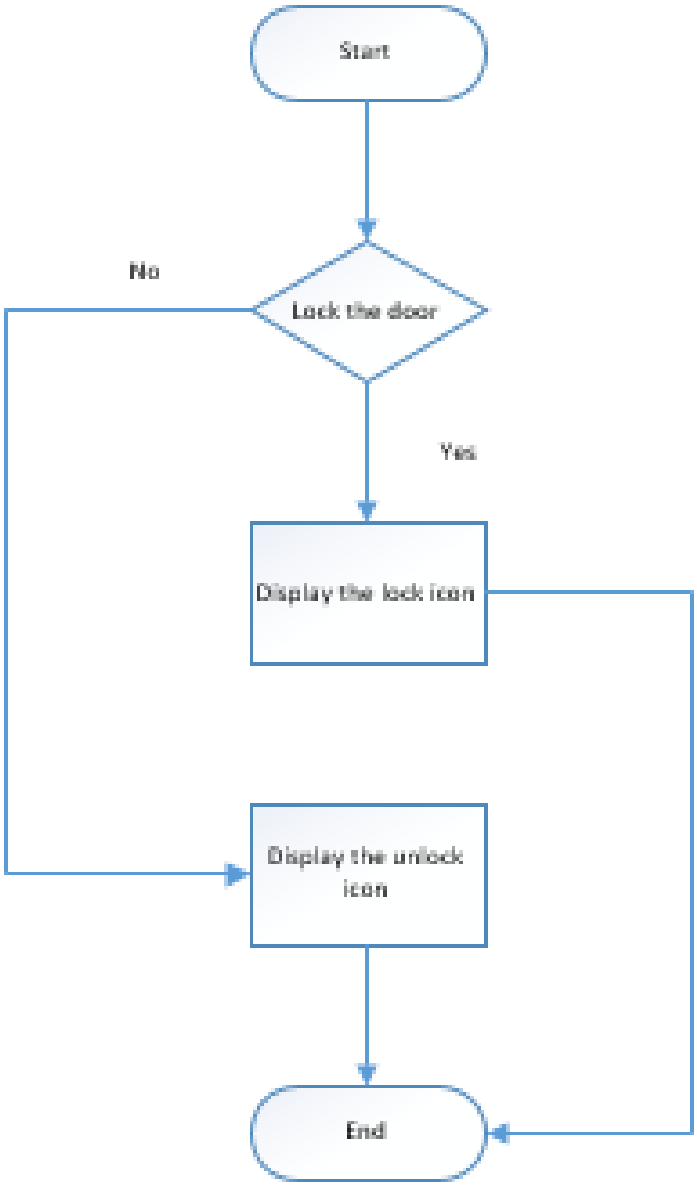


Figure 3.5: Flowchart for lock and unlock menu.

Figure 3.5 shows flowchart for lock and unlock menu. On the main interface, there is button for lock or unlock the door. Once the connection is established between the apps and the Bluetooth module, user can use this button to lock or unlock the door. If the user press the button to lock the door, state of the door will change to unlock. Unlock icon will appear in the main interface. Then, if the button is press again, the door will change its state from unlock to lock and the lock icon will appear in the main interface.

3.7 User interface design

Figure 3.6 shows lock icon on the main interface. To unlock the door, user need to press the lock icon. Then, the state of the door changed from lock to unlock. The icon will change to unlock icon.

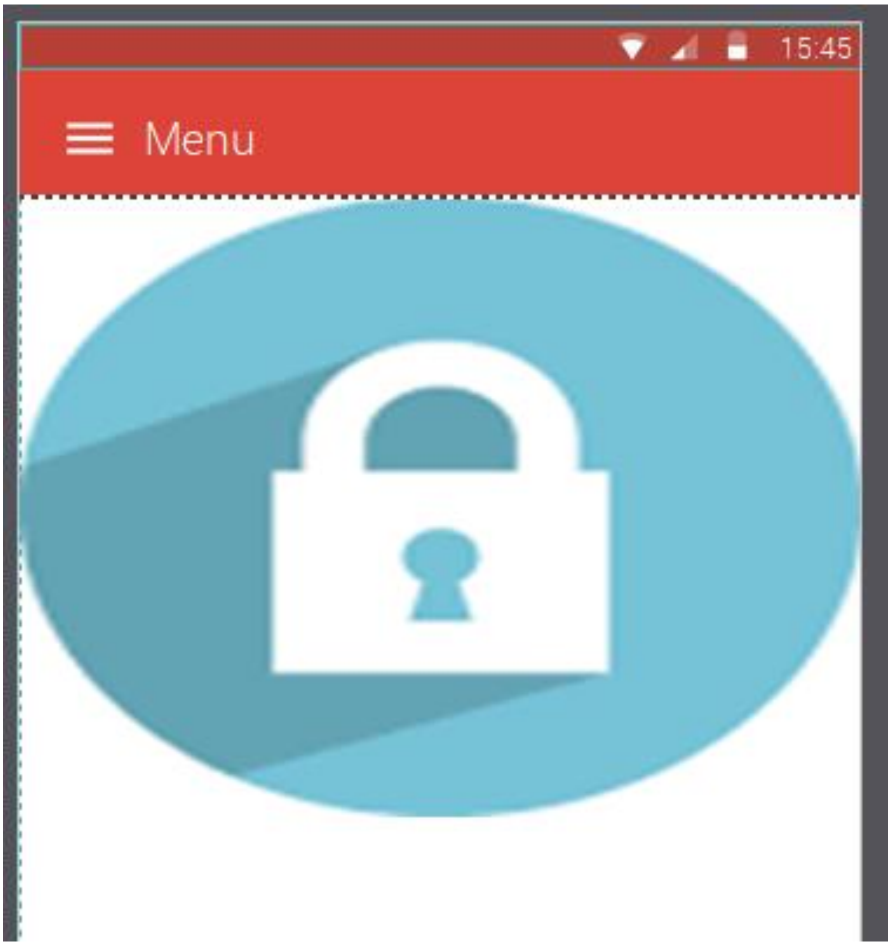


Figure 3.6: Lock interface design.

Figure 3.7 shows unlock icon on the middle of interface. To lock the door, user need to hold the unlock icon for few second. Then, the icon will change to lock icon.

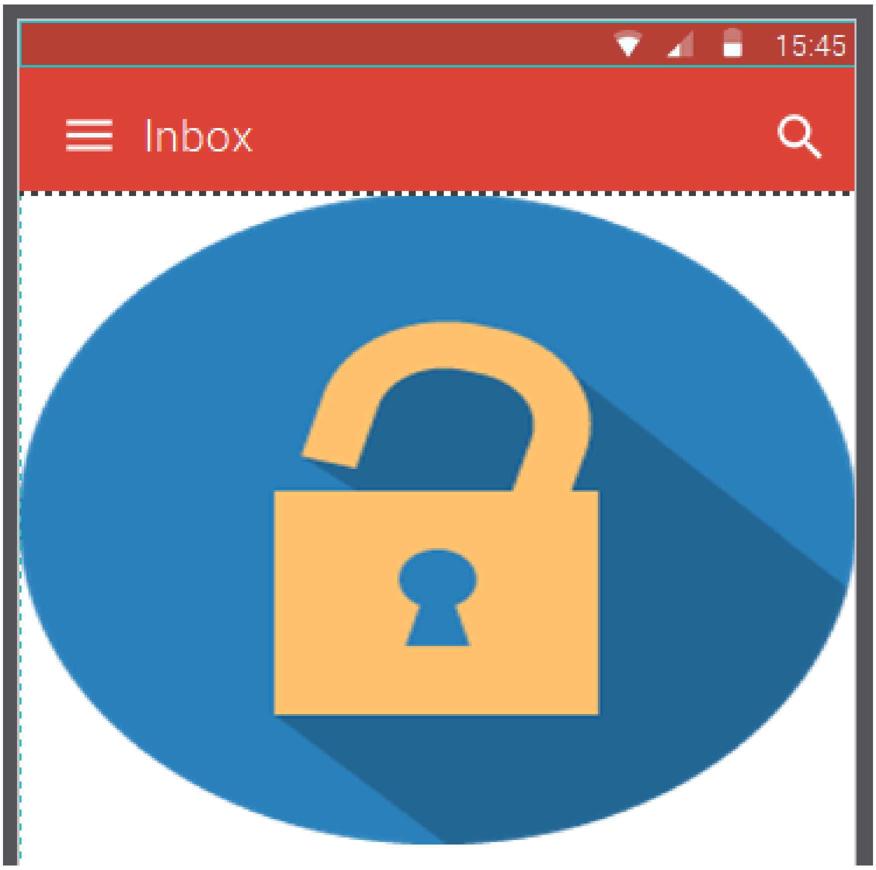


Figure 3.7: Unlock interface design.

3.8 Hardware and software

Selection of the correct software and hardware is an important part to ensure that all the progress in developing the proposed application run smoothly. The detail hardware and software used are explained in subsection below.

3.8.1 Documentation phase software

Table 3.1 shows the software items used throughout this documentation process in this project.

Table 3.1 Software used in the project

Software	Version	Description
Windows 10 professional	64-Bit operating system	As a platform to run the application.
Microsoft office word 2016	Version 2013	Used to create a report of proposed application.
Arduino	Version 2	Used to develop coding on Arduino Uno Board.
Android Studio	Version 4.0.3	Software used to develop an actual application.

3.8.2 Documentation phase hardware

Table 3.2 shows the hardware items used throughout the documentation process of this project.

Table 3.2: Hardware used in the project

Hardware	Specification	Description
Laptop	Model: Asus Processor: AMD A8-5550M	Use to create a report and also a platform to develop and run the application.

	System type: 64-bit operating system	
Printer	HP Deskjet Ink Advantage 2060	To print the report of proposed application.
USB Flash Drive	Kingston 4B	To duplicate the work done as a back-up as there is anything happen to the laptop.
Smart device	Samsung J7 2016 Android based	To connect with the Bluetooth.
Arduino Uno box	-	motherboard
Servo Motor	Tower pro SG 980 degrees servo	Rotate 180 degree to lock or unlock the door.
Bluetooth HCO six modules	-	Handle the Bluetooth communication between the android app and the Arduino.
Resistor	1 kilo ohm and 2 kilo ohm	For the Bluetooth modules voltage divider circuit.
Barrel door lock	-	To lock and unlock the door.
Wires	1 meter of wires	To connect between the Arduino and servo motor.
Paper clips	Few paper clips	Hold barrel door to lock and lock or unlock the door.
Miniature door	Made from wood	For testing the servo door lock for this project.

LCD	16x2 display	To display state of the door.
Red LED	-	Light up when the door is in unlock state.
Buzzer	Active buzzer	Make a sound when the door is in unlock state.
Breadboard	Long breadboard	Extension pin slot for Arduino Uno Board.
Potentiometer	10K	Adjust the brightness of LCD.

3.9 Conclusion

In this chapter the discussion has covered the project requirement, methodology used, hardware and software used in this development process. Rapid Application Development (RAD) methodology has been chosen. Flowchart, context diagram and use case have been included to show the flow of the application and interaction between user and application.

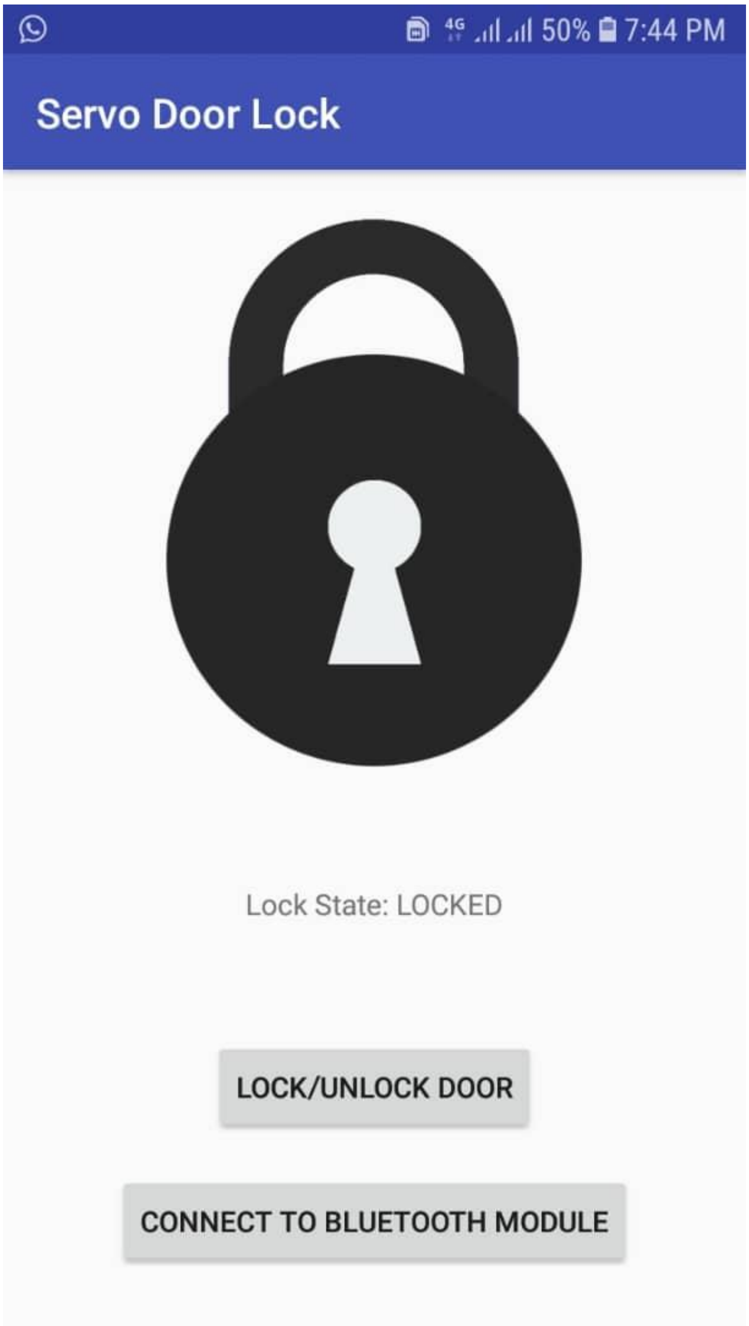
CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

In this chapter, the result implementation phase of Smart Door Lock Apps and testing phase is discussed. Smart Door Lock Apps is mobile android application that develops by using Android Studio. This app is connected with HC-06 Bluetooth module Arduino. In order to complete this project, connection in Arduino must be done before developing app in Android Studio.

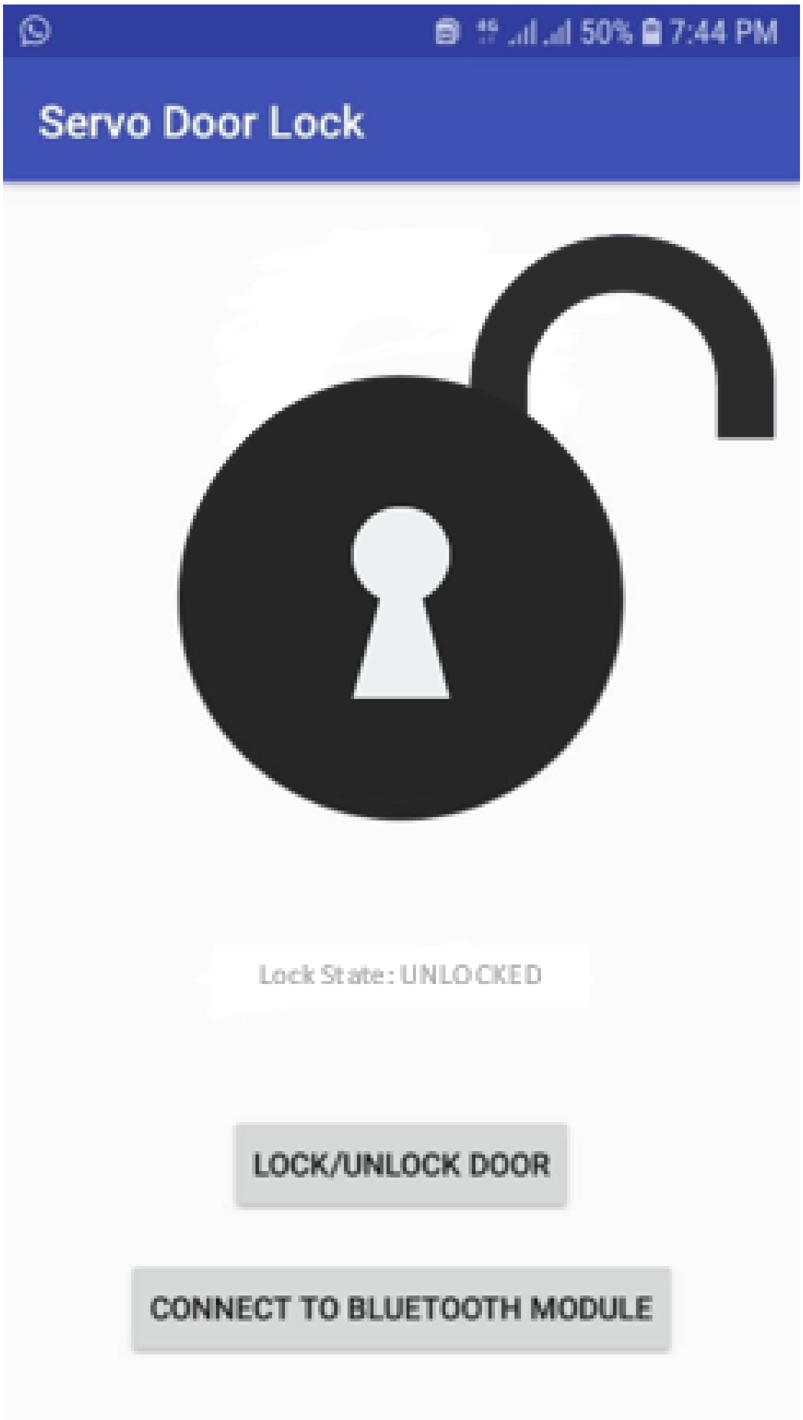
4.2 Implementation of Smart Door Lock Apps



Lock State: LOCKED

Figure 4.1: Locked State Interface

This is the main interface in the Smart Door Apps. In the main interface, there is icon to indicate user to know the state of the door. Below the icon, there are Lock State for the user. To use lock/unlock door button, user must connect the apps with Bluetooth module by turn on Bluetooth on android device. Once the connection is established, the user can lock or unlock the door by pressing lock/unlock door.



Lock State : UNLOCKED

Figure 4.2: Unlocked state interface.

This interface is shown in the apps when the state of door is unlocked. When the door is at unlocked state, buzzer will make sound. Red LED will light up to make the user become more alert with unlocked state of the door.

4.3 Arduino setup

This section discusses setup have been made in Arduino. Inside Arduino setup, there were few components are assembled together such as Arduino Uno Board, LCD, red LED, breadboard, servo motor and HC-06 Bluetooth module.

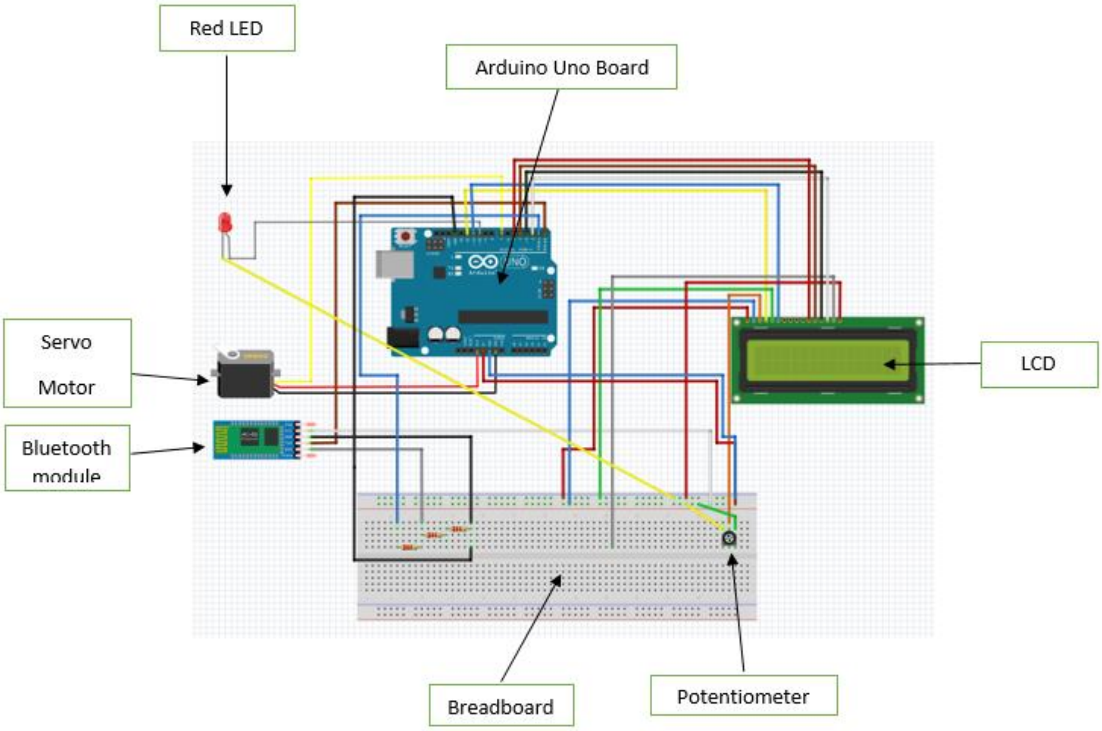


Figure 4.3: Schematic diagram of logical design of Arduino.

4.3.1 Bluetooth Module

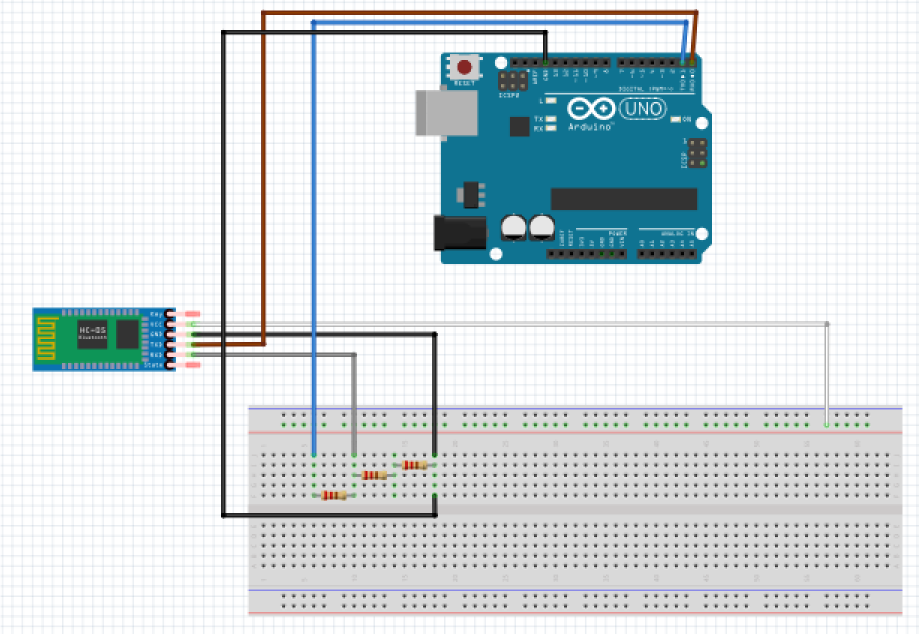


Figure 4.4: Schematic diagram Bluetooth Module

Bluetooth module that was used in the project is HC-06 module. HC-06 module was choose because this module is able to receive and transmit data between the app and Bluetooth module. Compare to HC-05 module, it’s feature only support receive data. Thus, HC-06 Bluetooth are the suitable to be used in this project.

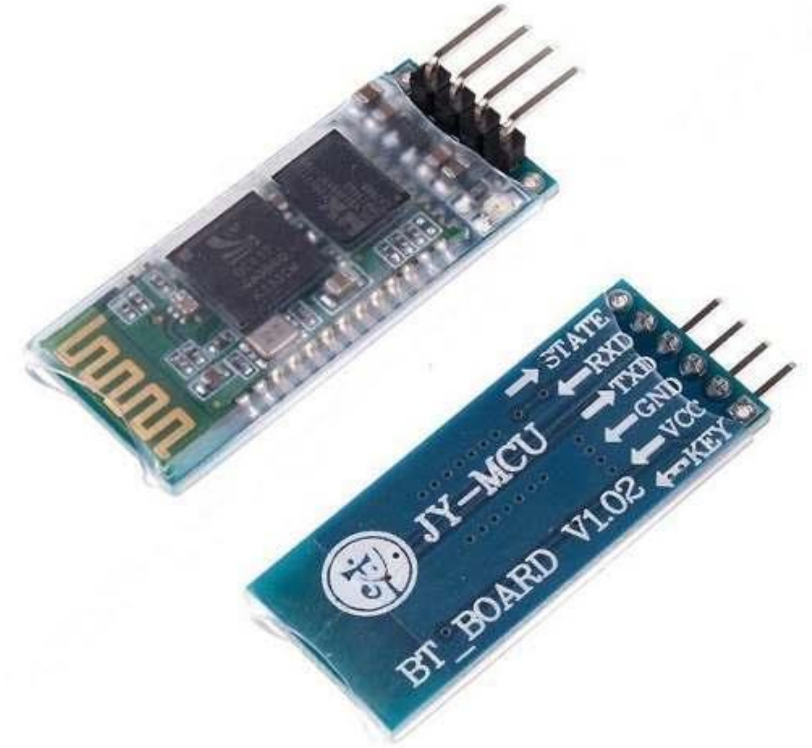


Figure 4.5: HC-06 Bluetooth module

HC-06 contains 4 pins. The pins were for receive data, transmit data, ground pin, and voltage pin to conduct electricity to power on the Bluetooth module. On the surface of HC-06 Bluetooth module, there is a small red LED. This small red LED is indicator for the user to know that connection between HC-06 Bluetooth module with the app is established. Transistor 0.2 ohm and 0.1 ohm was used in the breadboard to reduce the flow of voltage from Arduino Uno board which is 5V.

4.3.2 Tower Pro Servo motor SG90

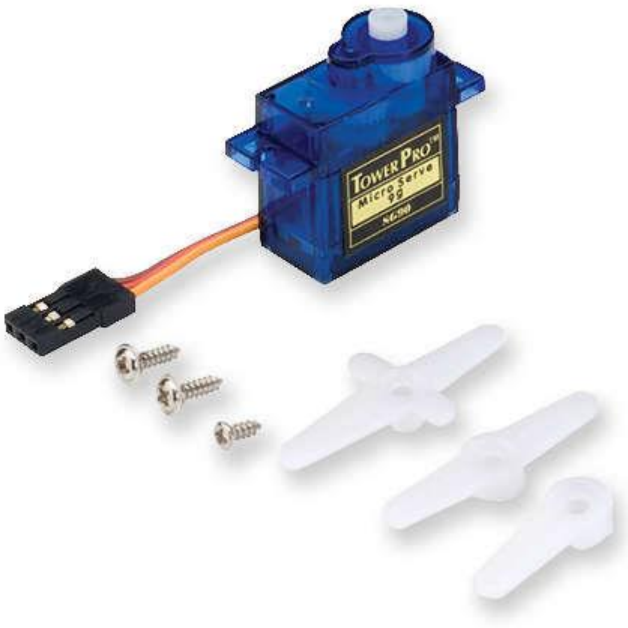


Figure 4.6: Tower Pro Servo Motor SG90

Tower Pro Servo Motor was choose because this type servo motor is able to make 180 degree of rotation. Servo motor SG90 is attached with paper clip that are bind together with the lock on the door. User will press lock or unlock button on the app. Then data is sent through HC-06 Bluetooth module. HC-06 Bluetooth module will send the information to the Servo Motor SG90 to do the action on the lock.

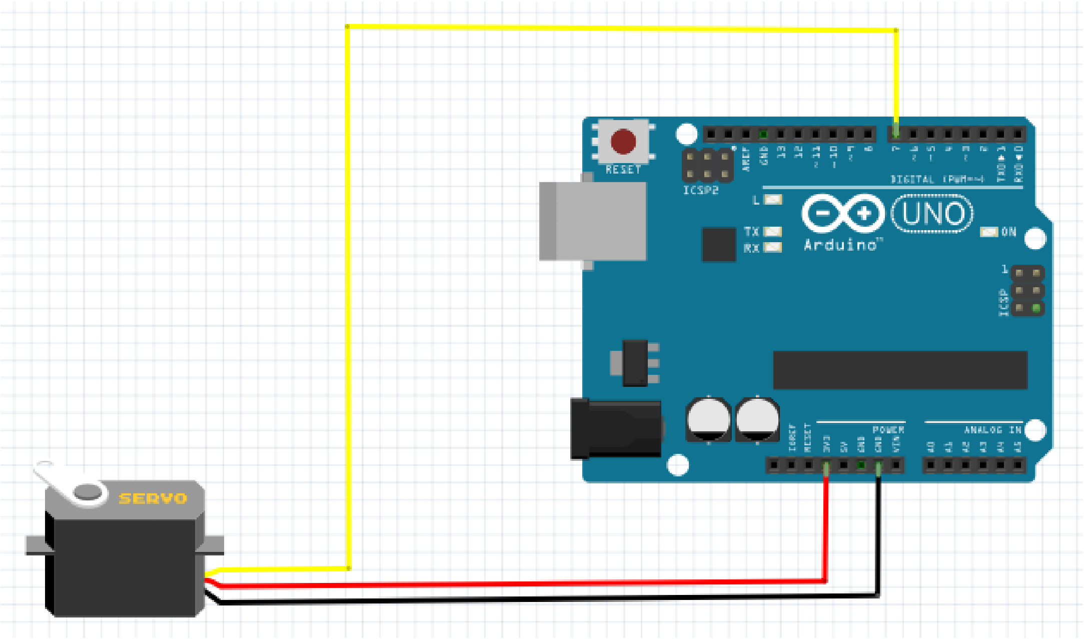


Figure 4.7: Schematic diagram Tower Pro Servo Motor SG90

Servo Motor tower pro SG90 need voltage 3.3V to operate. Thus, a pin from servo motor was connected to the pin 3.3V on Arduino Uno Board. One pin from servo motor is connected to pin number 7 on Arduino Uno Board for configuration purposed.

4.3.3 LCD

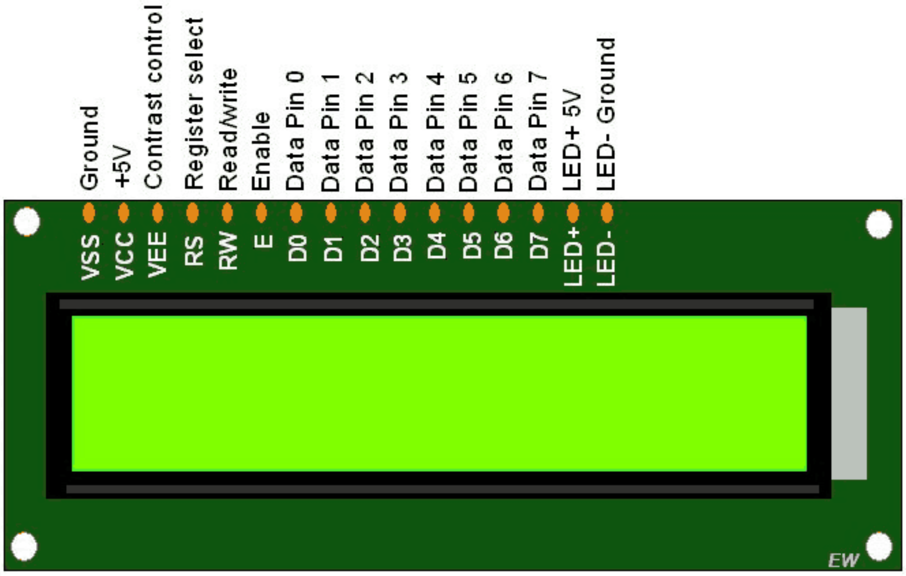


Figure 4.8: LCD 16x2 displays

LCD 16x2 display was implemented in front of the door. This LCD will display information about the state of the door for the user. If the door in lock state, the LCD will display “The door is locked!”. Meanwhile, if the door in unlock state LCD will display “The door is unlock!”.

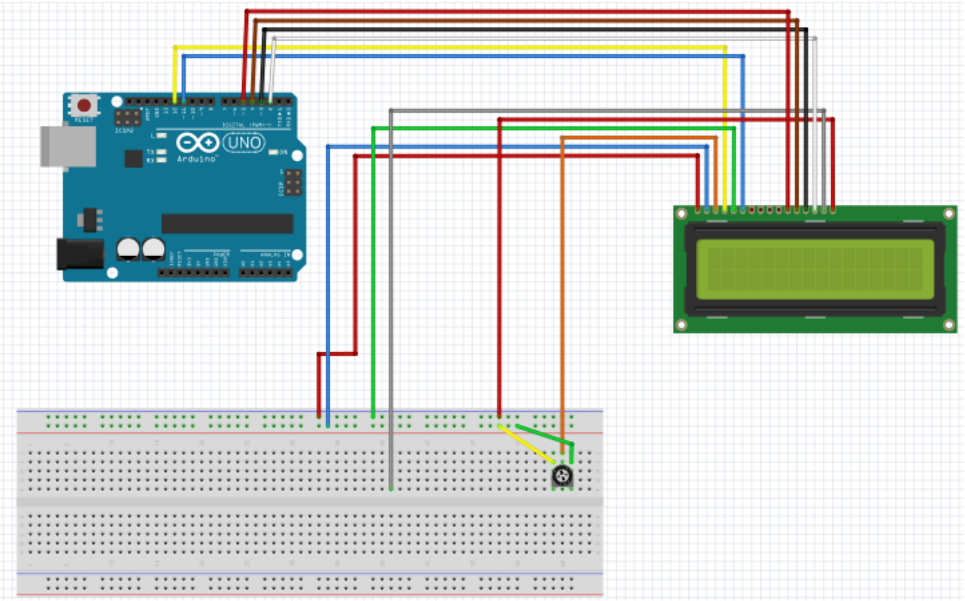


Figure 4.9: Schematic diagram LCD.

4.3.4 LED

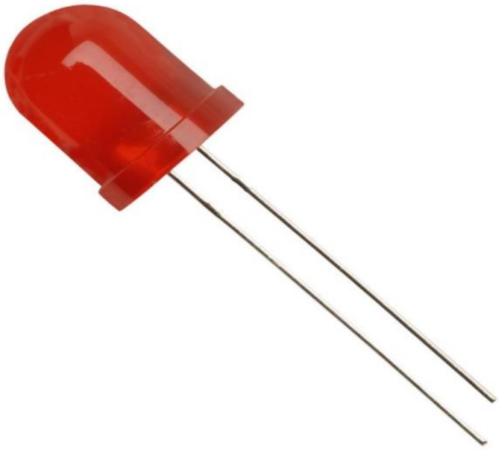


Figure 4.10: Red LED

Red LED is used in this project and was put in front of the door. This LED functions is to aid user to know the state of the door other than looking at the LCD. If the door is in locked state, the red LED will light up.

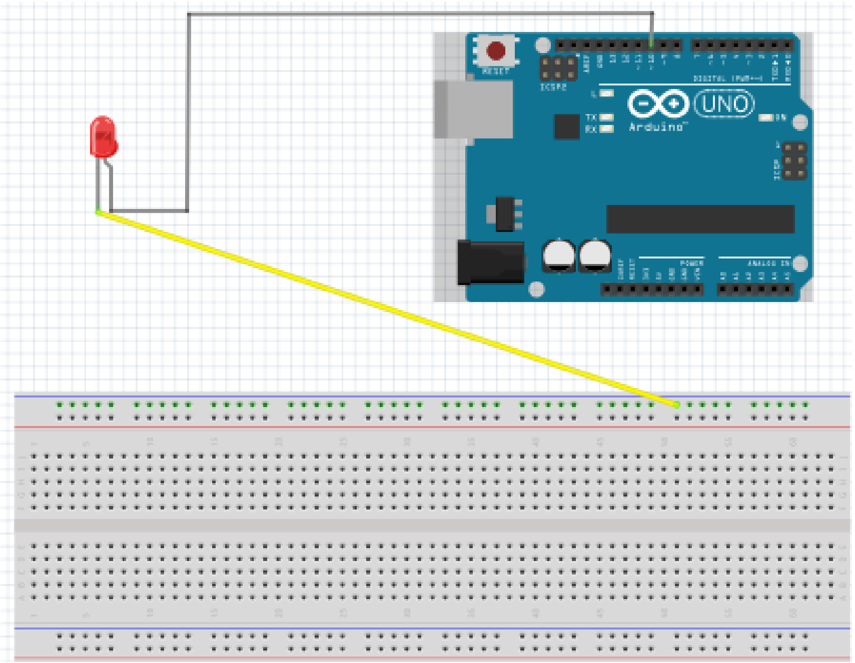


Figure 4.11: Red LED schematic diagram

4.3.5 Buzzer

Figure 4.12: Active Buzzer

Active buzzer is another aid for the user to know the state of the door. When the door is in unlocked state, the buzzer will be activated and make sounds as indication to the user to lock the door for security reminder.

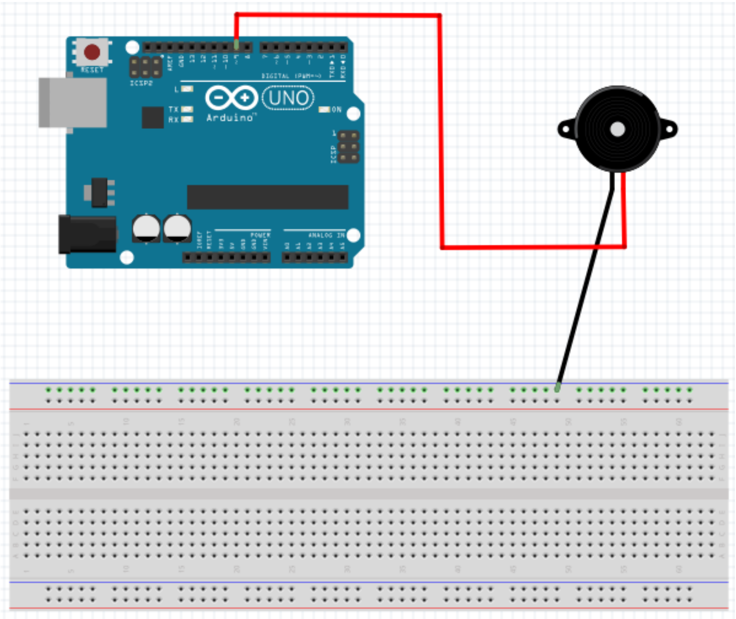


Figure 4.13: Buzzer Schematic diagram.

4.4 User Acceptance Test (UAT)

Smart Door Locks application is tested after the development is completed. The purpose of this test to check the functionality of the application. Please refer to appendix B for the result User Acceptance Test (UAT).

4.4.1 Connect to Bluetooth module button on the apps

Table 4.1 shows the User Acceptance Test for the Connect to Bluetooth module button on the Smart Door Locks apps.

Table 4.1: Connection Bluetooth module Test Case

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User connect to Bluetooth module	Mac Address Bluetooth Module	Connection established			
User not connected to Bluetooth module	No Mac Address Bluetooth Module	Failed to established the connection			

4.4.2 Lock/Unlock door button on the apps

Table 4.2 shows the User Acceptance Test for the lock/unlock door button on the Smart Door Locks apps.

Table 4.2: lock/unlock door button Test Case

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User press the button to lock the door	Press the button	The door state change to lock			
User press the button to unlock the door	Press the button	The door state change to unlock			

4.4.3 Bluetooth module HC-06 module on Arduino

Table 4.3 shows the User Acceptance Test for Bluetooth module HC-06 module on the Arduino.

Table 4.3: Bluetooth module HC-06 module connection Test Case

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
Servo Door Locks apps is paired with the Bluetooth module	Connection established	Small red LED will stop blinking			
Servo Door Locks apps is not paired with the Bluetooth module	Connection failed	Small red LED will continue blinking			

4.4.4 Red LED on Arduino

Table 4.4 shows the User Acceptance Test for red LED on the Arduino.

Table 4.4: red LED Test Case

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User locked the door	Door is lock	Red LED will not light up			
User unlocked the door	Door is unlock	Red LED will light up			

4.4.5 Buzzer on Arduino

Table 4.5 shows the User Acceptance Test for buzzer on the Arduino.

Table 4.5: buzzer Test Case

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User locked the door	Door is lock	Buzzer not produce sound			

User unlocked the door	Door is unlock	Buzzer will produce sound			

4.4.6 LCD on Arduino

Table 4.6 shows the User Acceptance Test for LCD on the Arduino.

Table 4.6: LCD Test Case

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User locked the door	Door is lock	LCD display “Door is locked”			
User unlocked the door	Door is unlock	LCD display “Door is unlocked”			

4.4.7 Tower Pro SG90 Servo Motor on Arduino

Table 4.7 shows the User Acceptance Test for Servo Motor on the Arduino.

Table 4.7: Servo Motor Test Case

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User locked the door	Door is lock	Servo motor will rotate 180 degree			
User unlocked the door	Door is unlock	Servo motor will rotate 180 degree			

As conclusion, all the test case will get pass result if the button lock or unlock on the apps works properly. But, the button do not work as expected. Thus, component Arduino such as buzzer, servo motor, red LED also do not run because cannot detect state of the door.

CHAPTER 5

CONCLUSION

5.1 INTRODUCTION

This chapter is discussed about the overall conclusion about the Smart Door Locks Bluetooth using mobile android application. This project consists of two major developments which is development of application and development of Arduino. This project is developed and proposed to help owner of human residence to keep their residence door or room secure.

5.2 CONSTRAINTS

There are few constraints during the development of this project such as:

i. Hard to get the Arduino component. Some of the Arduino component need to order and wait for the shipping about several weeks because the components were from other country such as China and Indonesia

ii. Limitation of Arduino knowledge. Steps to assemble Arduino components need refers to internet or someone that have knowledge about Arduino. If the components were not assembled correctly, the Arduino component would not work.

iii. Amount of money is needed to buy the Arduino components. The more Arduino components needed for the project, the more money will spend.

5.3 STRENGTH

Smart door locks Bluetooth using mobile android application have its own strength and weakness. They are:

i. Only applications paired with the correct Mac Address Bluetooth Module will be able to established the connection and use the apps. Thus, this application is more secure to use in term of security.

5.4 WEAKNESS

Smart door locks Bluetooth using mobile android application have its own weakness. They are:

i. Arduino Uno board need power source to operate. Without power source, Arduino Uno board would not work and the apps cannot be used.

ii. The connection between Bluetooth module and the Android Mobile only limited to radius 5 metre. If the Android mobile is exceeding from the radius Bluetooth module, the data cannot be received or transmit by Bluetooth module.

5.5 FUTURE WORKS

There are few upgrade can be done in this project. They are:

i. Change the feature project from Bluetooth to internet so that the user can lock the door from radius above 5 metre.

ii. Add Arduino components such as GSM module. GSM module functions is to send message to the user. When the user is away from home, GSM module can send message to the user about state of door as request by the user.

iii. Make the interface in Servo Door lock more attractive. For every transition state of door, the mobile will vibrate.

REFERENCES

Begun, A. L., Murphy, C., Bolt, D., Weinstein, B., Strodthoff, T., Short, L., & Shelley, G. (2003). *Characteristics of the Safe At Home Instrument for Assessing Readiness to Change Intimate Partner Violence*. Research on Social Work Practice, 13(1), 80-107.

What Is Rapid Application Development (RAD) and How Do You Use It? (2017, November 02). Retrieved from https://airbrake.io/blog/sdlc/rapid-application-development

What Is Rapid Application Development (RAD) and How Do You Use It? (2017, November 02). Retrieved from https://airbrake.io/blog/sdlc/rapid-application-development

MD Khairul, A, T. (2015) Play store [mobile software]. Malaysia.

ASSA ABLOY (2016) Play store [mobile software]. Australia.

Unicorn Devices Lab (2017) Play store [mobile software]. London.

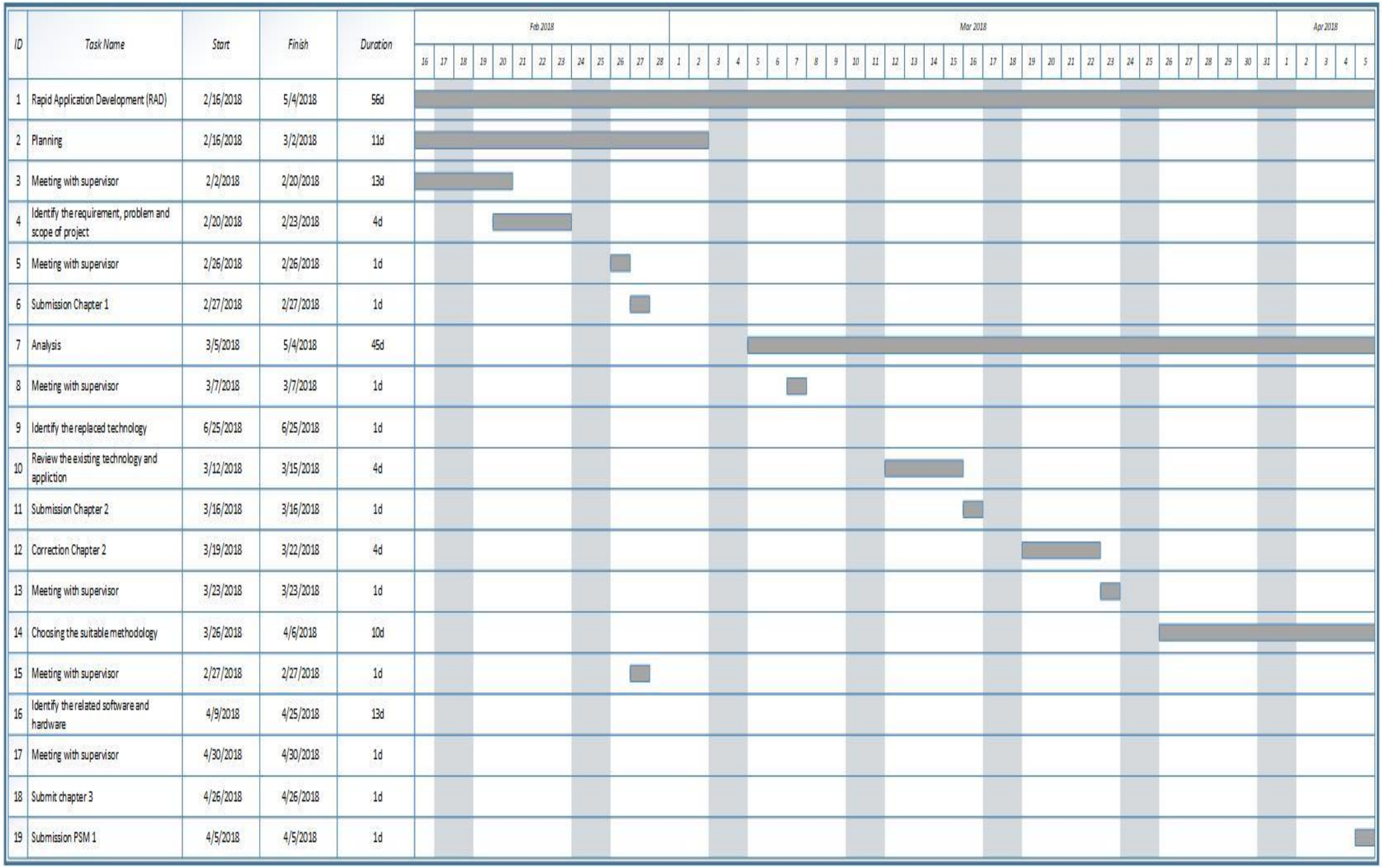
Basics, C. (2015, April 26). *How to Set Up and Program an LCD on the Arduino*. Retrieved from https://www.youtube.com/watch?v=Mr9FQKcrGpA

EEEnthusiast. (2014, January 01). *Arduino Tutorial #1 - Digital Inputs and Outputs* - Button & LED. Retrieved from https://www.youtube.com/watch?v=YWY_Is0L7fE

Education, S. M. (2015, April 10). *SunFounder Learning Kit Tutorial for Arduino* - Buzzer - Super Kit/Starter Kit/RFID Kit. Retrieved from https://www.youtube.com/watch?v=gRVwwB3ITgM

APPENDIX A

GANTT CHART



ID	Task Name	Start	Finish	Duration
1	Rapid Application Development (RAD)	2/16/2018	5/4/2018	56d
2	Planning	2/16/2018	3/2/2018	11d
3	Meeting with supervisor	2/2/2018	2/20/2018	13d
4	Identify the requirement, problem and scope of project	2/20/2018	2/23/2018	4d
5	Meeting with supervisor	2/26/2018	2/26/2018	1d
6	Submission Chapter 1	2/27/2018	2/27/2018	1d
7	Analysis	3/5/2018	5/4/2018	45d
8	Meeting with supervisor	3/7/2018	3/7/2018	1d
9	Identify the replaced technology	6/25/2018	6/25/2018	1d
10	Review the existing technology and appliction	3/12/2018	3/15/2018	4d
11	Submission Chapter 2	3/16/2018	3/16/2018	1d
12	Correction Chapter 2	3/19/2018	3/22/2018	4d
13	Meeting with supervisor	3/23/2018	3/23/2018	1d
14	Choosing the suitable methodology	3/26/2018	4/6/2018	10d
15	Meeting with supervisor	2/27/2018	2/27/2018	1d
16	Identify the related software and hardware	4/9/2018	4/25/2018	13d
17	Meeting with supervisor	4/30/2018	4/30/2018	1d
18	Submit chapter 3	4/26/2018	4/26/2018	1d
19	Submission PSM 1	4/5/2018	4/5/2018	1d

APPENDIX B

USER ACCEPTANCE TEST

Table 4.1: Connection Bluetooth module Test Case

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User connect to Bluetooth module	Mac Address Bluetooth Module	Connection established	connect	pass	
User not connected to Bluetooth module	No Mac Address Bluetooth Module	Failed to established the connection	Cannot connect	fail	

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User connect to Bluetooth module	Mac Address Bluetooth Module	Connection established	connect	pass	
User not connected to Bluetooth module	No Mac Address Bluetooth Module	Failed to established the connection	Cannot connect	fail	

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User connect to Bluetooth module	Mac Address Bluetooth Module	Connection established	connect	pass	
User not connected to Bluetooth module	No Mac Address Bluetooth Module	Failed to established the connection	Cannot connect	fail	

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**

User connect to Bluetooth module	Mac Address Bluetooth Module	Connection established	connect	pass	
User not connected to Bluetooth module	No Mac Address Bluetooth Module	Failed to established the connection	Cannot connect	fail	

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User connect to Bluetooth module	Mac Address Bluetooth Module	Connection established	connect	pass	
User not connected to Bluetooth module	No Mac Address Bluetooth Module	Failed to established the connection	Cannot connect	fail	

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User connect to Bluetooth module	Mac Address Bluetooth Module	Connection established	connect	pass	
User not connected to Bluetooth module	No Mac Address Bluetooth Module	Failed to established the connection	Cannot connect	fail	

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User connect to Bluetooth module	Mac Address Bluetooth Module	Connection established	connect	pass	
User not connected to Bluetooth module	No Mac Address Bluetooth Module	Failed to established the connection	Cannot connect	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User connect to Bluetooth module	Mac Address Bluetooth Module	Connection established	connect	pass	
User not connected to Bluetooth module	No Mac Address Bluetooth Module	Failed to established the connection	Cannot connect	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User connect to Bluetooth module	Mac Address Bluetooth Module	Connection established	connect	pass	
User not connected to Bluetooth module	No Mac Address Bluetooth Module	Failed to established the connection	Cannot connect	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User connect to Bluetooth module	Mac Address Bluetooth Module	Connection established	connect	pass	
User not connected to Bluetooth module	No Mac Address Bluetooth Module	Failed to established the connection	Cannot connect	fail	

Table 4.2: lock/unlock door button Test Case

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User press the button to lock the door	Press the button	The door state change to lock	The state change	fail	
User press the button to unlock the door	Press the button	The door state change to unlock	The state change	fail	

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User press the button to lock the door	Press the button	The door state change to lock	The state change	fail	
User press the button to unlock the door	Press the button	The door state change to unlock	The state change	fail	

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User press the button to lock the door	Press the button	The door state change to lock	The state change	fail	
User press the button to unlock the door	Press the button	The door state change to unlock	The state change	fail	

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User press the button to lock the door	Press the button	The door state change to lock	The state change	fail	
User press the button to unlock the door	Press the button	The door state change to unlock	The state change	fail	

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User press the button to lock the door	Press the button	The door state change to lock	The state change	fail	
User press the button to unlock the door	Press the button	The door state change to unlock	The state change	fail	

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User press the button to lock the door	Press the button	The door state change to lock	The state change	fail	
User press the button to unlock the door	Press the button	The door state change to unlock	The state change	fail	

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User press the button to lock the door	Press the button	The door state change to lock	The state change	fail	
User press the button to unlock the door	Press the button	The door state change to unlock	The state change	fail	

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User press the button to lock the door	Press the button	The door state change to lock	The state change	fail	
User press the button to unlock the door	Press the button	The door state change to unlock	The state change	fail	

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User press the button to lock the door	Press the button	The door state change to lock	The state change	fail	
User press the button to unlock the door	Press the button	The door state change to unlock	The state change	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User press the button to lock the door	Press the button	The door state change to lock	The state change	fail	
User press the button to unlock the door	Press the button	The door state change to unlock	The state change	fail	

Table 4.3: Bluetooth module HC-06 module connection Test Case

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Servo Door Locks apps is paired with the Bluetooth module	Connection established	Small red LED will stop blinking	Stop blinking	pass	
Servo Door Locks apps is not paired with the Bluetooth module	Connection failed	Small red LED will continue blinking	Continue blinking	pass	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Servo Door Locks apps is paired with the Bluetooth module	Connection established	Small red LED will stop blinking	Stop blinking	pass	
Servo Door Locks apps is not paired with the Bluetooth module	Connection failed	Small red LED will continue blinking	Continue blinking	pass	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Servo Door Locks apps is paired with the Bluetooth module	Connection established	Small red LED will stop blinking	Stop blinking	pass	
Servo Door Locks apps is not paired with the Bluetooth module	Connection failed	Small red LED will continue blinking	Continue blinking	pass	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Servo Door Locks apps is paired with the Bluetooth module	Connection established	Small red LED will stop blinking	Stop blinking	pass	
Servo Door Locks apps is not paired with the Bluetooth module	Connection failed	Small red LED will continue blinking	Continue blinking	pass	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Servo Door Locks apps is paired with the Bluetooth module	Connection established	Small red LED will stop blinking	Stop blinking	pass	
Servo Door Locks apps is not paired with the Bluetooth module	Connection failed	Small red LED will continue blinking	Continue blinking	pass	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Servo Door Locks apps is paired with the Bluetooth module	Connection established	Small red LED will stop blinking	Stop blinking	pass	
Servo Door Locks apps is not paired with the Bluetooth module	Connection failed	Small red LED will continue blinking	Continue blinking	pass	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Servo Door Locks apps is paired with the Bluetooth module	Connection established	Small red LED will stop blinking	Stop blinking	pass	
Servo Door Locks apps is not paired with the Bluetooth module	Connection failed	Small red LED will continue blinking	Continue blinking	pass	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Servo Door Locks apps is paired with the Bluetooth module	Connection established	Small red LED will stop blinking	Stop blinking	pass	
Servo Door Locks apps is not paired with the Bluetooth module	Connection failed	Small red LED will continue blinking	Continue blinking	pass	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Servo Door Locks apps is paired with the Bluetooth module	Connection established	Small red LED will stop blinking	Stop blinking	pass	
Servo Door Locks apps is not paired with the Bluetooth module	Connection failed	Small red LED will continue blinking	Continue blinking	pass	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Servo Door Locks apps is paired with the Bluetooth module	Connection established	Small red LED will stop blinking	Stop blinking	pass	
Servo Door Locks apps is not paired with the Bluetooth module	Connection failed	Small red LED will continue blinking	Continue blinking	pass	

Table 4.4: red LED Test Case

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Red LED will not light up	Not light up	fail	
User unlocked the door	Door is unlock	Red LED will light up	Light up	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Red LED will not light up	Not light up	fail	
User unlocked the door	Door is unlock	Red LED will light up	Light up	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Red LED will not light up	Not light up	fail	
User unlocked the door	Door is unlock	Red LED will light up	Light up	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Red LED will not light up	Not light up	fail	
User unlocked the door	Door is unlock	Red LED will light up	Light up	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Red LED will not light up	Not light up	fail	
User unlocked the door	Door is unlock	Red LED will light up	Light up	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Red LED will not light up	Not light up	fail	
User unlocked the door	Door is unlock	Red LED will light up	Light up	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Red LED will not light up	Not light up	fail	
User unlocked the door	Door is unlock	Red LED will light up	Light up	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Red LED will not light up	Not light up	fail	
User unlocked the door	Door is unlock	Red LED will light up	Light up	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Red LED will not light up	Not light up	fail	
User unlocked the door	Door is unlock	Red LED will light up	Light up	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Red LED will not light up	Not light up	fail	
User unlocked the door	Door is unlock	Red LED will light up	Light up	fail	

Table 4.5: buzzer Test Case

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Buzzer not produce sound	Not produce any sound	fail	
User unlocked the door	Door is unlock	Buzzer will produce sound	Produce sound	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Buzzer not produce sound	Not produce any sound	fail	
User unlocked the door	Door is unlock	Buzzer will produce sound	Produce sound	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Buzzer not produce sound	Not produce any sound	fail	
User unlocked the door	Door is unlock	Buzzer will produce sound	Produce sound	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Buzzer not produce sound	Not produce any sound	fail	
User unlocked the door	Door is unlock	Buzzer will produce sound	Produce sound	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Buzzer not produce sound	Not produce any sound	fail	
User unlocked the door	Door is unlock	Buzzer will produce sound	Produce sound	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Buzzer not produce sound	Not produce any sound	fail	
User unlocked the door	Door is unlock	Buzzer will produce sound	Produce sound	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Buzzer not produce sound	Not produce any sound	fail	
User unlocked the door	Door is unlock	Buzzer will produce sound	Produce sound	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Buzzer not produce sound	Not produce any sound	fail	
User unlocked the door	Door is unlock	Buzzer will produce sound	Produce sound	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Buzzer not produce sound	Not produce any sound	fail	
User unlocked the door	Door is unlock	Buzzer will produce sound	Produce sound	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Buzzer not produce sound	Not produce any sound	fail	
User unlocked the door	Door is unlock	Buzzer will produce sound	Produce sound	fail	

Table 4.6: LCD Test Case

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	LCD display “Door is locked”	display “Door is locked”	pass	
User unlocked the door	Door is unlock	LCD display “Door is unlocked”	display “Door is unlocked”	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	LCD display “Door is locked”	display “Door is locked”	pass	
User unlocked the door	Door is unlock	LCD display “Door is unlocked”	display “Door is unlocked”	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	LCD display “Door is locked”	display “Door is locked”	pass	
User unlocked the door	Door is unlock	LCD display “Door is unlocked”	display “Door is unlocked”	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	LCD display “Door is locked”	display “Door is locked”	pass	
User unlocked the door	Door is unlock	LCD display “Door is unlocked”	display “Door is unlocked”	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	LCD display “Door is locked”	display “Door is locked”	pass	
User unlocked the door	Door is unlock	LCD display “Door is unlocked”	display “Door is unlocked”	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	LCD display “Door is locked”	display “Door is locked”	pass	
User unlocked the door	Door is unlock	LCD display “Door is unlocked”	display “Door is unlocked”	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	LCD display “Door is locked”	display “Door is locked”	pass	
User unlocked the door	Door is unlock	LCD display “Door is unlocked”	display “Door is unlocked”	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	LCD display “Door is locked”	display “Door is locked”	pass	
User unlocked the door	Door is unlock	LCD display “Door is unlocked”	display “Door is unlocked”	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	LCD display “Door is locked”	display “Door is locked”	pass	
User unlocked the door	Door is unlock	LCD display “Door is unlocked”	display “Door is unlocked”	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	LCD display “Door is locked”	display “Door is locked”	pass	
User unlocked the door	Door is unlock	LCD display “Door is unlocked”	display “Door is unlocked”	fail	

Table 4.7: Servo Motor Test Case

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	
User unlocked the door	Door is unlock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	
User unlocked the door	Door is unlock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	
User unlocked the door	Door is unlock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	
User unlocked the door	Door is unlock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	
User unlocked the door	Door is unlock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	
User unlocked the door	Door is unlock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	
User unlocked the door	Door is unlock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	
User unlocked the door	Door is unlock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
User locked the door	Door is lock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	
User unlocked the door	Door is unlock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	

Test Case	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail**	**Comment**
User locked the door	Door is lock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	
User unlocked the door	Door is unlock	Servo motor will rotate 180 degree	Servo motor not rotate 180 degree	fail	

List of User that answer UAT:

User 1: Abdul murshid bin abdul majid

User 2: khairul nizam sukahli

User 3: amir waliyuddin

User 4: asyraf afnan jamluddin

User 5: fadhilah Abdullah

User 6: ahmad mustaqim

User 7: ilham affendi

User 8:hilfi aljefri

User 9: mimi maysarah mahadi

User 10: eidha athira

APPENDIX C

CODING IN ANDROID STUDIO

In main activity java:

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.view.MotionEvent;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.Set;
import java.util.UUID;
import android.os.Handler;


public class MainActivity extends AppCompatActivity {

    private final String DEVICE_ADDRESS = "00:18:E4:40:00:06"; //MAC Address
of Bluetooth Module
    private final UUID PORT_UUID = UUID.fromString("00001101-0000-1000-8000-
00805f9b34fb");

    private BluetoothDevice device;
    private BluetoothSocket socket;

    private OutputStream outputStream;
    private InputStream inputStream;

    Thread thread;
    byte buffer[];

    boolean stopThread;
    boolean connected = false;
    String command;

    Button lock_state_btn, bluetooth_connect_btn;

    TextView lock_state_text;

    ImageView lock_state_img;

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

        lock_state_btn = (Button) findViewById(R.id.lock_state_btn);
        bluetooth_connect_btn = (Button)
findViewById(R.id.bluetooth_connect_btn);

        lock_state_text = (TextView) findViewById(R.id.lock_state_text);

        lock_state_img = (ImageView) findViewById(R.id.lock_state_img);

        bluetooth_connect_btn.setOnClickListener(new View.OnClickListener() {
           @Override
            public void onClick(View v){

               if(BTinit())
               {
                   BTconnect();
                   beginListenForData();

                   // The code below sends the number 3 to the Arduino asking
it to send the current state of the door lock so the lock state icon can be
updated accordingly

                   command = "3";

                   try
                   {
                       outputStream.write(command.getBytes());
                   }
                   catch (IOException e)
                   {
                       e.printStackTrace();
                   }

               }
           }
        });

        lock_state_btn.setOnClickListener(new View.OnClickListener() {
           @Override
            public void onClick(View v){

            if(connected == false)
            {
                Toast.makeText(getApplicationContext(), "Please establish a
connection with the bluetooth servo door lock first",
Toast.LENGTH_SHORT).show();
            }
            else
            {
                command = "1";

                try
                {
                    outputStream.write(command.getBytes()); // Sends the
number 1 to the Arduino. For a detailed look at how the resulting command is
handled, please see the Arduino Source Code
                }
                catch (IOException e)
                {
                    e.printStackTrace();
                }
            }
           }
        });
    }

    void beginListenForData() // begins listening for any incoming data from
the Arduino
    {
        final Handler handler = new Handler();
        stopThread = false;
        buffer = new byte[1024];

        Thread thread = new Thread(new Runnable()
        {
            public void run()
            {
                while(!Thread.currentThread().isInterrupted() && !stopThread)
                {
                    try
                    {
                        int byteCount = inputStream.available();
                        if(byteCount > 0)
                        {
                            byte[] rawBytes = new byte[byteCount];
                            inputStream.read(rawBytes);
                            final String string = new String(rawBytes, "UTF-
8");

                            handler.post(new Runnable()
                            {
                               public void run()
                               {
                                    if(string.equals("3"))
                                    {
                                        lock_state_text.setText("Lock State:
LOCKED"); // Changes the lock state text

lock_state_img.setImageResource(R.drawable.locked_icon); //Changes the lock
state icon
                                    }
                                    else if(string.equals("4"))
                                    {
                                        lock_state_text.setText("Lock State:
UNLOCKED");

lock_state_img.setImageResource(R.drawable.unlocked_icon);
                                    }
                               }
                            });
                        }
                    }
                    catch (IOException ex)
                    {
                        stopThread = true;
                    }
                }
            }
        });

        thread.start();
    }

    //Initializes bluetooth module
    public boolean BTinit()
    {
        boolean found = false;

        BluetoothAdapter bluetoothAdapter =
BluetoothAdapter.getDefaultAdapter();

        if(bluetoothAdapter == null) //Checks if the device supports bluetooth
        {
            Toast.makeText(getApplicationContext(), "Device doesn't support
bluetooth", Toast.LENGTH_SHORT).show();
        }

        if(!bluetoothAdapter.isEnabled()) //Checks if bluetooth is enabled. If
not, the program will ask permission from the user to enable it
        {
            Intent enableAdapter = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(enableAdapter,0);

            try
            {
                Thread.sleep(1000);
            }
            catch(InterruptedException e)
            {
                e.printStackTrace();
            }
        }

        Set<BluetoothDevice> bondedDevices =
bluetoothAdapter.getBondedDevices();

        if(bondedDevices.isEmpty()) //Checks for paired bluetooth devices
        {
            Toast.makeText(getApplicationContext(), "Please pair the device
first", Toast.LENGTH_SHORT).show();
        }
        else
        {
            for(BluetoothDevice iterator : bondedDevices)
            {
                if(iterator.getAddress().equals(DEVICE_ADDRESS))
                {
                    device = iterator;
                    found = true;
                    break;
                }
            }
        }

        return found;
    }

    public boolean BTconnect()
    {

        try
        {
            socket = device.createRfcommSocketToServiceRecord(PORT_UUID);
//Creates a socket to handle the outgoing connection
            socket.connect();

            Toast.makeText(getApplicationContext(),
                    "Connection to bluetooth device successful",
Toast.LENGTH_LONG).show();
            connected = true;
        }
        catch(IOException e)
        {
            e.printStackTrace();
            connected = false;
        }

        if(connected)
        {
            try
            {
                outputStream = socket.getOutputStream(); //gets the output
stream of the socket
            }
            catch(IOException e)
            {
                e.printStackTrace();
            }

            try
            {
                inputStream = socket.getInputStream(); //gets the input stream
of the socket
            }
            catch (IOException e)
            {
                e.printStackTrace();
            }
        }

        return connected;
    }

    @Override
    protected void onStart()
    {
        super.onStart();
    }
}
```

In Android manifest:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.philipgo.servodoorlock">

    <uses-permission android:name = "android.permission.BLUETOOTH" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/logo_launch"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity" android:screenOrientation =
"portrait">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

APPENDIX D

CODING IN ARDUINO BOARD

```
#include <Servo.h>

#include <EEPROM.h>



Servo servo;

char state;



void setup() {

  // put your setup code here, to run once:

  servo.attach(7);



  if(EEPROM.read(0) == 1) // Reads the EEPROM value stored to know what state the door lock
was in before it was last turned off

  {                    // An EEPROM value of 1 means UNLOCKED and a value of 2 means LOCKED

    servo.write(70); // Rotates the servo to the unlocked position

    delay(200);

  }

  else if(EEPROM.read(0) == 2)
  {
    servo.write(120); // Rotates the servo to the locked position
    delay(200);
  }

  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  if(Serial.available() > 0)
  {
    char data;
    data = Serial.read(); // The variable data is used to store the value sent by the Android app

    switch(data)
  {
   case '1':
    if(EEPROM.read(0) == 1) //An EEPROM value of 1 means it is currently unlocked
    {
     EEPROM.write(0, 2); // Writes the number 2 to address 0 on the Arduino's EEPROM. This
value will be used by the Arduino to remember the last state the door lock was in
     Serial.print("3"); // Sends the number 3 to the Android app. To see what this does, please
see the Android Studio Project file

     for(int a = 70; a <= 120; a++) // Rotates the servo to the locked position
     {
      servo.write(a);
      delay(15);
      Serial.println(servo.read());
     }
    }
    else if(EEPROM.read(0) == 2) //An EEPROM value of 2 means it i currently locked
    {
     EEPROM.write(0, 1); // Writes the number 1 to address 0 on the Arduino's EEPROM. This value will be used by the Arduino to remember the last state the door lock was in

     Serial.print("4"); // Sends the number 4 to the Android app. The number sent will be used by the app to update the locked/unlocked icon

     for(int x = 120; x >= 70; x--) // Rotates the servo to the unlocked position

     {

      servo.write(x);

      delay(15);

     }

    }

    break;

   case '3':

   // This part of the switch case statement is used everytime the Arduino is turned on

   // Basically, the Android app sends a command asking the Arduino to send the current state of the lock in order to accordingly update the locked/unlocked icon in the app

    if(EEPROM.read(0) == 1)

    {

     Serial.print("4");
```

```
        }

        else if(EEPROM.read(0) == 2)

        {

          Serial.print("3");

        }

        break;

    }

  }



}
```