

**SRI LANKA INSTITUTE OF ADVANCE TECHNOLOGICAL EDUCATION  
HARDY ADVANCE TECHNOLOGICAL INSTITUTE -AMPARA**



## **PROJECT FINAL REPORT**

**Submitted in Partial Fulfilment of the Requirement  
OF  
HIGHER NATIONAL DIPLOMA IN INFORMATION TECHNOLOGY  
AMP/IT/2019/F/0004.**

## LOST – FOUND MOBILE APPLICATION.

### FINAL REPORT

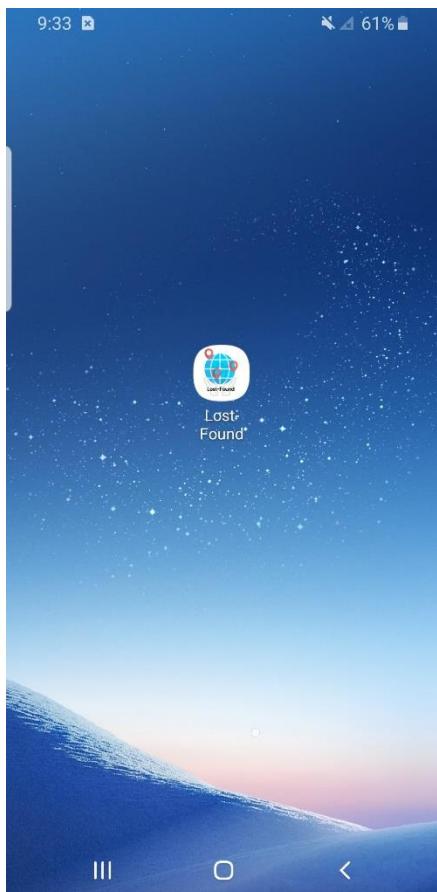


Figure 1: App icon

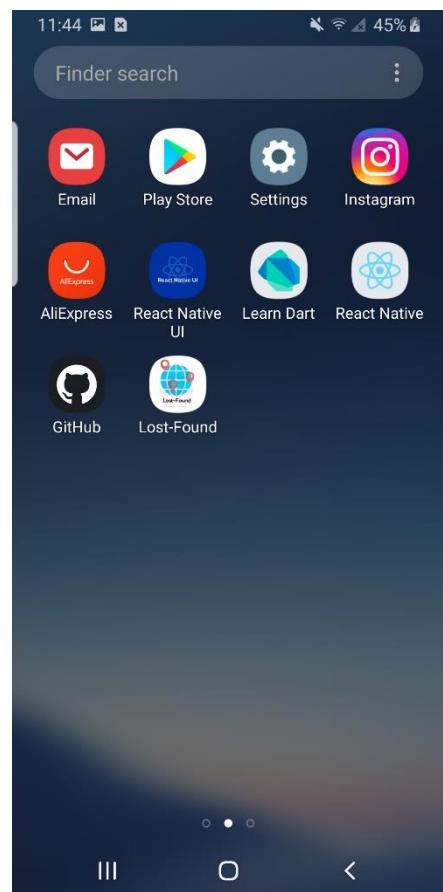


Figure 2: App icon

## **TITLE PAGE**

# **LOST – FOUND MOBILE APPLICATION.**

Name : M.A.M.Imran

Register No : AMP/IT/2019/F/0004

## **FINAL REPORT**

Submitted in Partial Fulfilment of the Requirement

Of

## **HIGHER NATIONAL DIPLOMA IN INFORMATION TECHNOLOGY**

2019 – 2022

Hardy Advanced Technological Institute

Ampara

-Approved-

.....

Mr. Aasik Abdulkassim.

The Supervisor,

Lecturer

Hardy Advanced Technological Institute

Ampara

Date.....

## PLAGIARISM

You Are Reminded That There Exist Regulations Concerning Plagiarism. Extracts from These Regulations Are Printed Overleaf. Please Sign Below to Say That You Have Read and Understand These Extracts.

.....

Signature

.....

Date

**For supervisor to complete: -**

Date Received : .....

Name of The Supervisor : .....

Marks Obtain : .....

Signature : .....

Date : .....

## **ACKNOWLEDGEMENT**

I take this opportunity to express our heartfelt gratitude and profound feeling to everyone who lend me their helping hand to make my project a success. A warm thank to our supervisor, Mr. Aasik Abdulcassim. (Lecturer of IT Department) for the constant support rendered to meet at all the stage of the project at all the times, right from the very beginning till the submission of the final report.

It is a great pleasure to be taking this opportunity to deliver our worth thanks to Miss.T.H. Dhinusha Darshani (HOD). It is our bond to be thanking Mr.F.A Inshaf , Mr.M.A. Arsath Areef (Demonstrators) and Mr.Neel Sampath (Demonstrators) to support for success of the project from them continue guidance.

We have taken efforts in this project. However, we would like to extend our sincere thanks to all of them.

Thank you.

M.A.M.Imran.

## **DECLARATION**

I hereby declare that the work reported in this project work on “**LOST – FOUND MOBILE APPLICATION.**” submitted to Hardy advance technological institute is my original work done in the form of partial requirement for the higher national diploma in Information Technology (HNDIT) under the supervision of Mr. Aasik Abdulkassim (Lecturer of IT Department) Hardy Advance Technological Institute. The material contained in the report has not been submitted to any University or Institution for the award of any higher national diploma.

## ABSTRACT

Today world is totally bases on electronic equipment. Nowadays everyone from waking up to telling the bed most of time spending with electronic equipment. If such important electronic equipment is lost there is a problem in locating, it and handing it over to the right person when someone else's electronic equipment is available therefore I here to solve this problem.

My main aim is to do this project for free of charge which can help the society. Oblivion and error are the common human beings. Oblivion of Electronic Equipment such as Mobile Phone, Camera, Laptops, Tablets, Smart Watches, Smart Gadgets, and other Electronic Equipment. This equipment is special and valuable in their lives sometimes unfortunately some of the situations are unexpected to lose their valuable equipment. If the valuable equipment lost, "It will learn to the suffering of the difficulty and enjoy the happiness when it gets back".

People with human being who never and ever love or like other's goods and thinks and they love to hand-over the goods/thinks to that's owner. But Major part of the process is difficult to hand-over the equipment to proper person. Therefore, My **LOST-FOUND** application is the solution for this problem.

**LOST – FOUND** Application mainly develop to engage the left behind property to owners and founders. If a person lost something on somewhere **LOST- FOUND** Application will completely help to get back the equipment with limited sorting plans and relations.

This System Has Developed for android users. I Have Done a Lot of Works for This Project. I Got Some Difficulties and Some Problems during This Project's Works. I Hope this application will be most helpful for all. I'll use this system for a long and test with critical usage after that I'll publish this application to Google Play Store for entire world use.

## CONTENTS

TITLE PAGE .....	3
PLAGIARISM .....	4
ACKNOWLEDGEMENT .....	5
DECLARATION .....	6
ABSTRACT.....	7
Chapter 01.	
Introduction.	
1.1. INTRODUCTION .....	12-13
1.2. AIM AND OBJECTIVES .....	14
1.3. INTENDED AUDIENCE OF THE PROJECT .....	15
1.4. SCOPE OF THE PROJECT .....	16
1.5. ASSUMPTION ON THIS PROJECT .....	17
1.6. KEY PROCESS /FUNCTIONS OF THE APPLICATION ..	Error! Bookmark not defined.18
1.7. PROJECT DEVELOPMENT AND METHODOLOGY .....	19
1.8. BRIEF DESCRIPTION OF THE SOFTWARE, HARDWARE SPECIFICATION TO DEVELOP THE APPLICATION .....	20-22
Chapter 02.	
Background Analysis.	
2.1. INRODUCTION.....	23
2.2. DETAILS AND REVIEW OF RELEVANT THEORY/ PREVIOUSLY DEVELOPED SYSTEM .....	24
2.3. BACKGROUND ANALYSIS FOR THE PROPOSED SYSTEM.....	25
Chapter 03	
Specification and design	
3.1 INTRODUCTION .....	26
3.2 DEVELOPMENT METHODOLOGY .....	26-27
3.3 TOP LEVEL ARCHITECTURE .....	28-29
3.4 USER INTERFACE DESIGN.....	30-35
Chapter 04	
Testing and Debugging	

4.1	INTRODUCTION .....	36
4.2	DIFFERENT TEST CASES .....	36
4.2.1	TEST CASE FOR REGISTER USER.....	36-38
4.2.2	TEST CASE FOR USER LOGIN .....	39-42
4.2.3	TEST CASE FOR RESET PASSWORD .....	43-44
4.2.4	TEST MENU .....	45
4.2.5	TEST CASE FOR LOST ACTIVITY .....	46
4.2.6	TEST CASE FOR FOUND ACTIVITY .....	47
4.2.7	TEST CASE FOR LOGOUT.....	48
4.3	ANDROID APP TESTING .....	49
4.4	ANDROID TESTING STRATEGIES .....	50-52
4.5	ANDROID APPLICATION DEBUGGING .....	53-56
Chapter 05		
Implementation		
5.1	INRODUCTION .....	57
5.2	CODE STRUCTURE.....	57-64
5.3	DOCUMENT STRUCTURE .....	65-66
5.4	REQUIREMENTS .....	67
5.4.1.	SOFTWARE REQUIREMENTS .....	67
5.4.2	HAREWARE REQUIREMENTS .....	67
5.4.3	OTHER REQUIREMENTS .....	67
5.5	PROBLEM ENCOUNTERED BY IMPLEMENTATION.....	67
6.	CONCLUSION .....	68
7.	LIMITATIONS .....	68
8.	REFERENCE .....	68-69

## TABLE OF FIGURES

Figure 1: Application Icon .....	2
Figure 2: Application Icon on menu .....	2
Figure 3: Application Logo .....	12
Figure 4: Android Studio IDE.....	13
Figure 5: Android Studio IDE.....	13
Figure 6: Android Studio IDE .....	13
Figure 7: Humanity.....	14
Figure 8: Sign In methods .....	15
Figure 9: Smart Gagtes .....	16
Figure 10: Waterfall Method .....	19
Figure 11: Chipmunk version .....	20
Figure 12: Android Studio Gradle version.....	20
Figure 13: IDE Interface.....	21
Figure 14: Emulator Memory consumed .....	21
Figure 15: Task Manager Hardware consumed .....	22
Figure 16: Methodology.....	26
Figure 17: IDE Logo .....	27
Figure 18: Firebase.....	27
Figure 19: Top Level Architecture .....	28
Figure 20: ER Diagram.....	29
Figure 21: UI login .....	30
Figure 22: UI register.....	31
Figure 23: UI reset password .....	32
Figure 24: UI Manu .....	33
Figure 25: Lost Activity.....	34
Figure 26: Found Activity .....	35
Figure 27: Test User register .....	37
Figure 28: Registration Success .....	38
Figure 29: Test User Login.....	39
Figure 30: Test Wrong User Login.....	40
Figure 31: Verification email.....	41
Figure 32: Verified Screen .....	42
Figure 33: Test User Reset Password .....	43
Figure 34: Reset Icon.....	43
Figure 35: Test Reset Email .....	44
Figure 36: Test Menu .....	45
Figure 37: Test Lost Activity .....	46
Figure 38: Test Found Activity.....	47
Figure 39: Logout Icon .....	48
Figure 40: Test Device Information.....	49
Figure 41: Testing Strategies.....	50
Figure 42: Unit Test.....	50

Figure 43: Unit Test Components .....	51
Figure 44: UI Integration Test .....	51
Figure 45: Debug Toolbar.....	53
Figure 46: Debug Devices.....	53
Figure 47: Debug Using USB.....	54
Figure 48: Wi-Fi Debugging.....	54
Figure 49: Wi-Fi Debugging Android 11.....	55
Figure 50: Debugger.....	56
Figure 51: XML Files .....	57
Figure 52: XML Code For Main Page.....	58
Figure 53: Java Files .....	59
Figure 54: Java Codes for Main Activity.....	60
Figure 55: Default App Icon .....	61
Figure 56: App Icon Change .....	61
Figure 57: Firebase Console .....	62
Figure 58: Firebase Authentication.....	62
Figure 59: Realtime Database .....	63
Figure 60: JSON Code File .....	64
Figure 61: Device Document.....	65
Figure 62: Project File .....	66

## Chapter 01

### 1.1. INTRODUCTION



Figure 3: Application Logo

The **LOST-FOUND** is an android application that run on Mobile Phones and Tablets which is above 5.1.1 – Lollipop android Operating System. It's an efficient way to return lost and founded item to their rightful owner. **LOST-FOUND** application, business that deal with the public easily manage reported lost item and customer claims all in one platform.

The **LOST – FOUND** Application mainly develop to engage the left behind property to owners and founders. If a person lost something on somewhere **LOST-FOUND** Application will completely help to get back the equipment with limited sorting plans and relations.

The **LOST-FOUND** have user registration system, verify account, reset password system, login system, report lost item system, report found item system, and etc.... we haven't asked any permissions from your phone for the security option, and we never miss use your information.

**LOST-FOUND** is developed with IntelliJ's Android Studio IDE and Google Firebase database therefore it's a well-developed and well-tested application which consume ram of your phone, battery of your phone, CPU of your phone, and other hardware components effectively and efficiently.

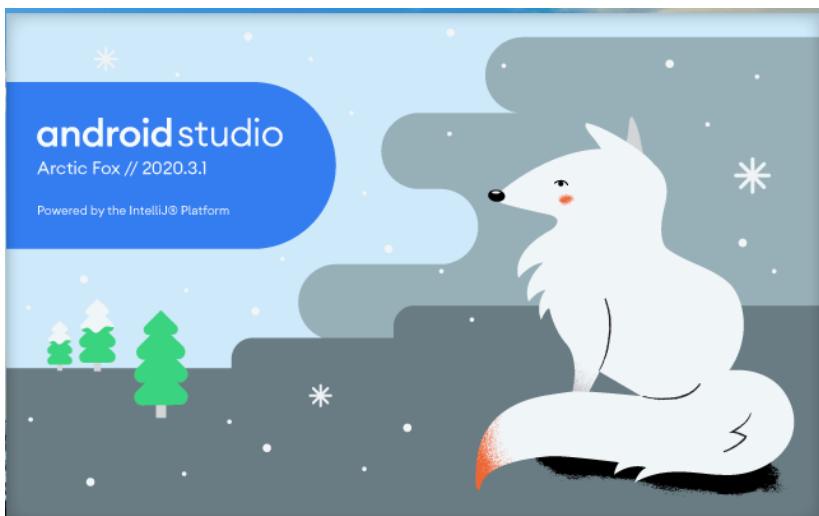


Figure 4: Android Studio IDE

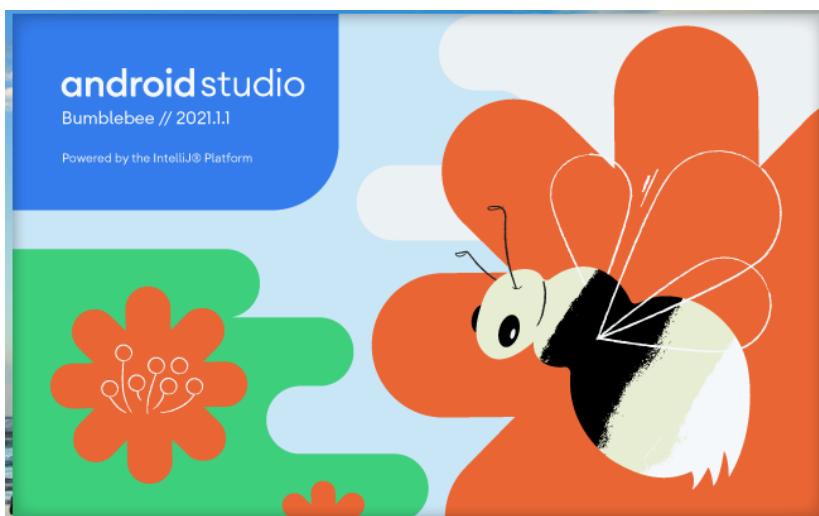


Figure 5: Android Studio IDE

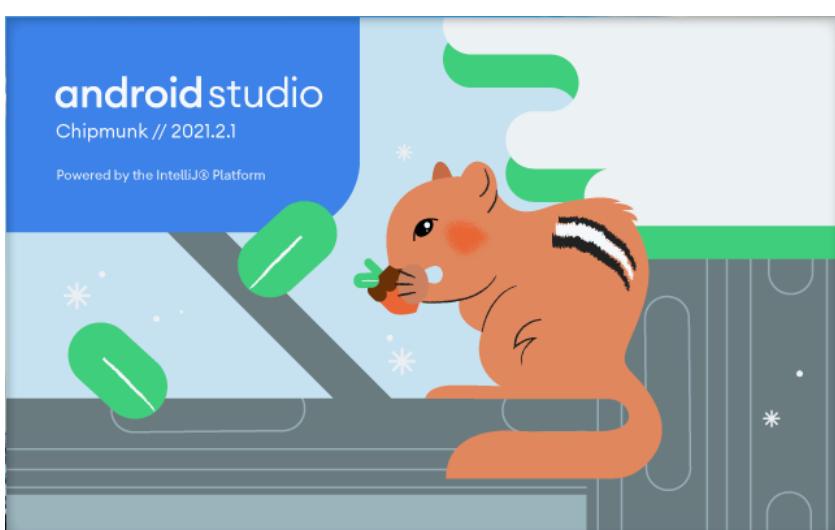


Figure 6: Android Studio IDE

## **1.2. AIMS AND OBJECTIVES**

### **AIMS**

To create a platform based on serial number which are electronic equipment that could be lost or found. Engage the left behind property to owners and founders of that electronic equipment.

### **OBJECTIVES**

- This project designed to meet requirements of electronic equipment looser and founder.
- The system gives reliable, easier, and faster action facilities.
- The system is very simple in design and to implement. The system requires very low system resources, and the system will work in almost all configurations it has got following features.
- It will ensure data accuracy.
- Record will be efficiently maintained by Firebase.
- Make connection between looser and founder.
- User can make request lost forms.
- Minimum time needed for the various processing.
- Users can communicate and our application will increase humanity.
- Login system, email interconnection



Figure 7: Humanity

### 1.3. INTENDED AUDIENCE OF THE PROJECT

The people who are interacted with the system can be defined as users. Users must register with email. User can use the own email such as google mail, yahoo mail, outlook, live, and whatever mail, but they want to verify theirs accounts via verify link which is send to their email. In this system, two types of users can be identified based on their access roles. They are namely equipment looser and equipment Founder. And I'm the (developer) will provide maintain all of bug and database and other remote services, I will accept all of my users inquires and I'll make them satisfy by fixing their inquiries, etc...

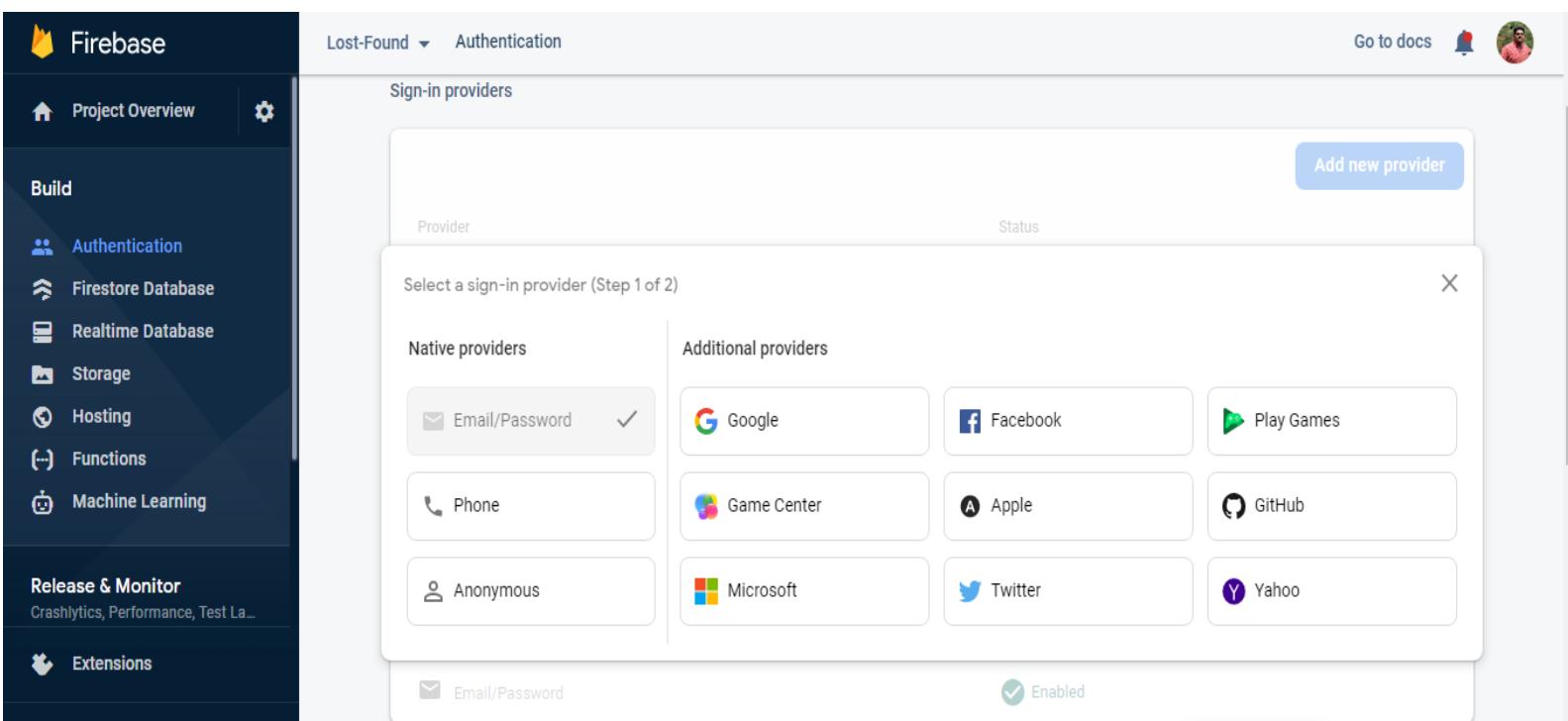


Figure 8: Sign in methods

## 1.4. SCOPE OF THE PROJECT

There is a no proper way for this problem to our country is a tourism country most of tourist are complaining they have oblivious their equipment while there's trip on that equipment they store their memories, travel photographs and videos and other stored information. Some of them saying "we don't want that electronic equipment we want only that stored information". Not only tourists our peoples also stack on this problem. My application is Simple solution for oblivion, mistake, and faults. By using our application ill Increasing Humanity.



Figure 9: Smart Gadgets

## **1.5. ASSUMPTION ON THIS PROJECT**

I avoid getting unnecessary permission from user Mobiles, due to the security reason and in case of attack or malware separation it will make user dissatisfaction therefore I avoid that permission such as, body sensor, Microphone, SMS, Telephone, and additional permissions. I have used some API for reduce workload. I avoid direct dialing system because of it make cost and its not work worldwide and I use mailing system for now.

## **1.6. KEY PROCESS /FUNCTIONS OF THE APPLICATION**

LOST – FOUND application activities and functions are user friendly, user understandable, and easy to access and use, Main functions are,

- i. User Login System.
  - User Registration System.
  - User account verify System
  - User password reset system
- ii. Lost Equipment System
  - Lost equipment searching system
  - Lost request inserting system
- iii. Found Equipment System
  - Found equipment information providing System.
  - Contacting equipment owner via email System.

I have decided to develop particular LOST – FOUND application. Because I hope this application will help to decrease thefts, Stoles, oblivion, error, and faults which of electronic equipment.

## 1.7. PROJECT DEVELOPMENT AND METHODOLOGY

The section describes the methods that will be used to gather requirements for the system. It covered data collection. Analysis, design tools and development etc.

The system development life cycle provides an overall framework for system development. There are many more concepts for help, including methodologies, models, tools, and techniques which will be discussed in detail. During the research review, there are no meaningful one-size-fits-all software development methods. The approach one takes to a very large project must include explicit forms of planning, coordination and control that would be out of place or even harmful in a small project. Methods used then, depend on such conditions as system size, reliability and safety requirements, cost constraints, implementation schedule and maintainability and expected system life spam (Gan Chun Hou, 2004).

The waterfall is the simplest model of the software development process to view its stages as successors to one another. The waterfall model gives a high-level view of the software life cycle. This model is a tried and tested problem-solving mechanism. 7

Documentation is an integral part of the process. This model has a various stage where the work of each stage is “signed off” before proceeding to the next phase. Waterfall model is sequential software development process, where progress flows steadily toward the conclusion (like waterfall) through the phase of a project.

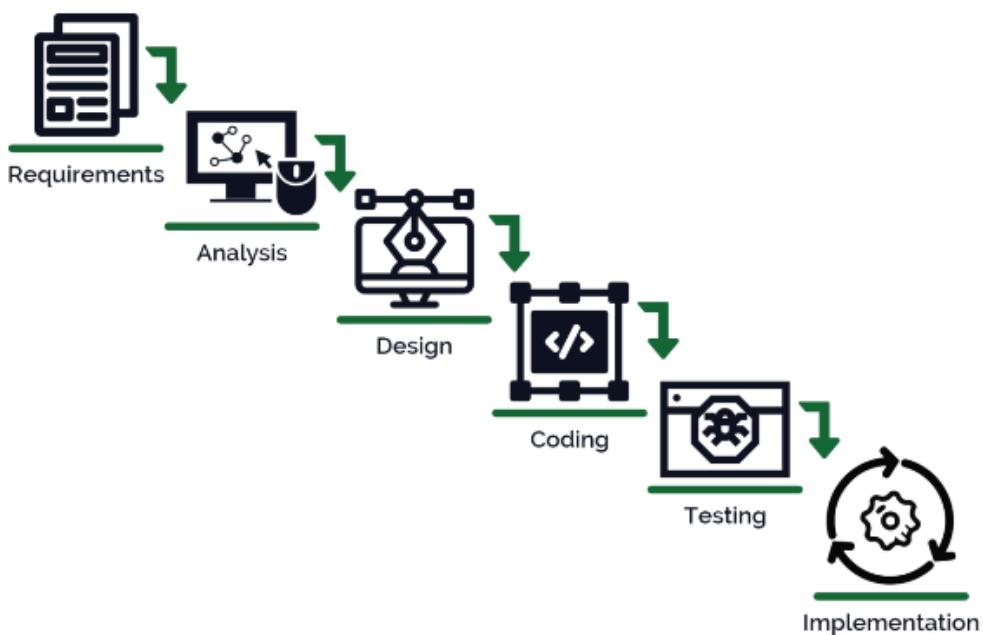


Figure 10: Waterfall Method

## **1.8. BRIEF DESCRIPTION OF THE SOFTWARE, HARDWARE SPECIFICATION TO DEVELOP THE APPLICATION.**

The recommended specification for developing mobile application, software system and hardware system, which is suitable with the system requirement and the project methodology as follows.

The team computer hardware is collection of physical parts of a computer system.

### **Software requirements**

System requirements are expressed in a software requirement documents. The software requirements specification (SRS) is the official statements of what is required of the system developers.

The operating system or platform will be Microsoft windows 10, then Android Studio IDE with updated SDK tool set, Java Language J2EE updated JDK, XML will be used for developing the front end of the Application because has easy to develop the lost found mobile application system. This mobile application including details of the electronic equipment and users' information were connected to the database on option Firebase Firestone and Firebase Realtime database and JSON for interconnection between different plate form data transaction and Java programming languages will be used to develop backend of the system.

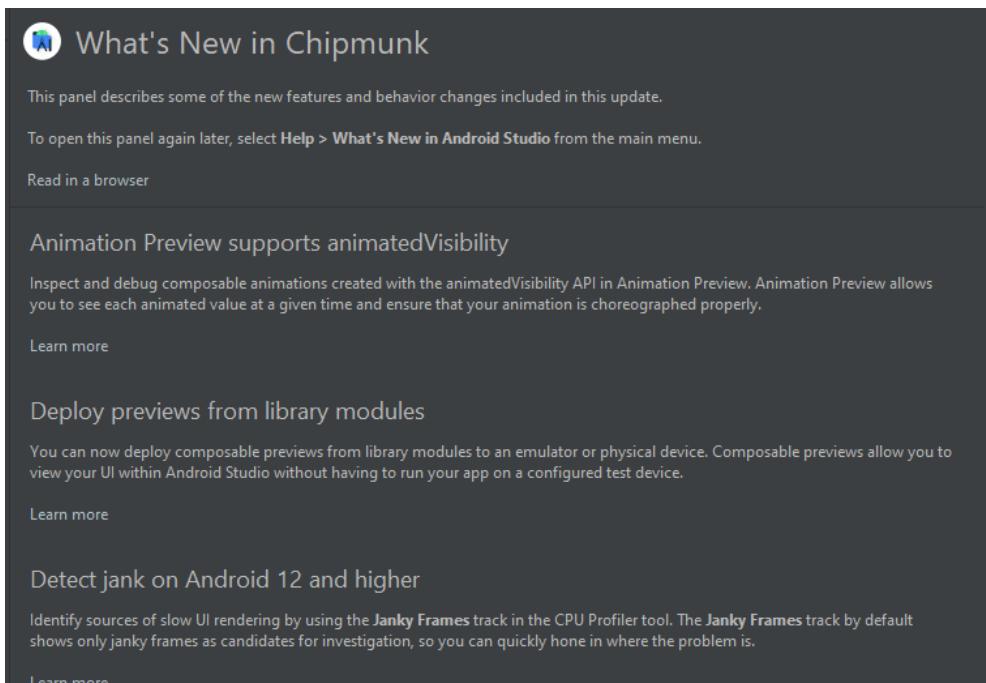


Figure 11: Chipmunk version

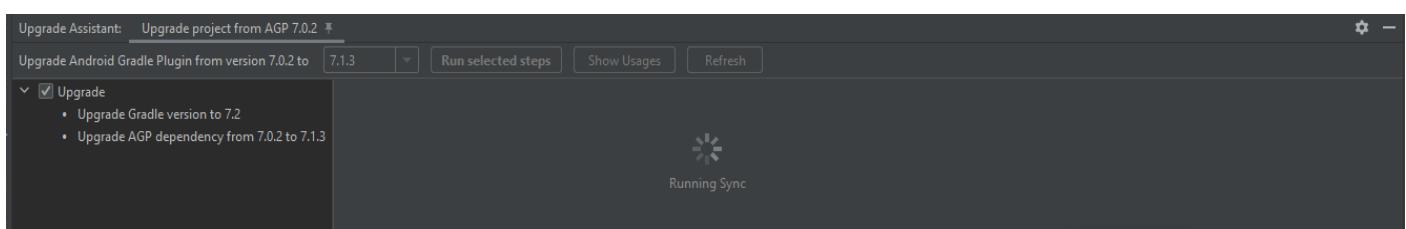


Figure 12: Android Studio Gradle version

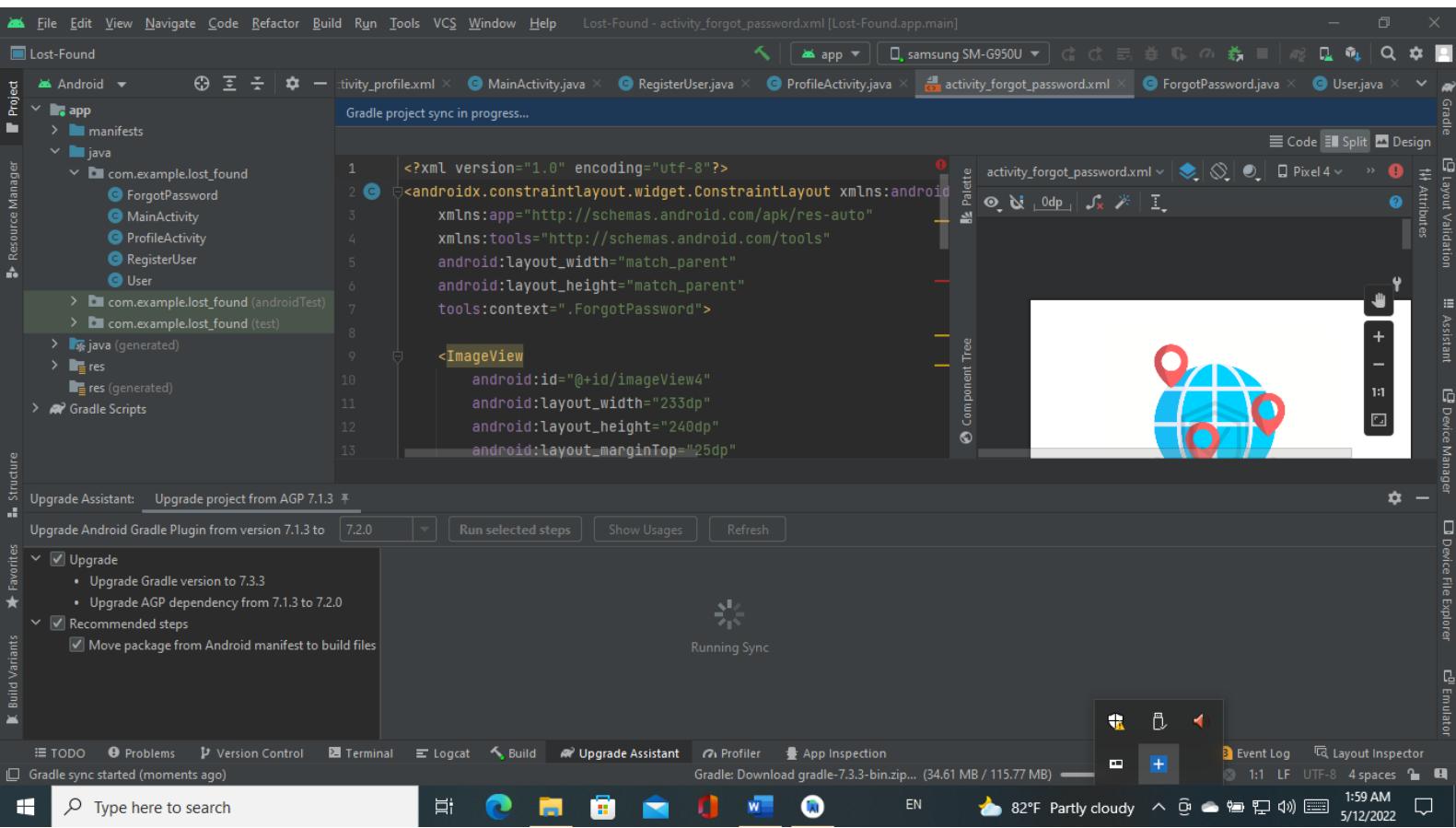


Figure 13: IDE Interface

## Hardware Requirement

The term computer hardware is the collection of physical parts of a computer system. This includes the monitor, keyboard, and mouse etc... In order to provide hardware to executing Gradle build run system store and enhance performance of the computer speed. Lost found mobile application system and it testing Emulator also including the hardware requirements for developing the new System. Such as Samsung CORE i3, 8GB - Ram will be used for the new system, 1TB Hard-disk space or high need for the system, 15.6" HD Monitor and Keyboard and Mouse.

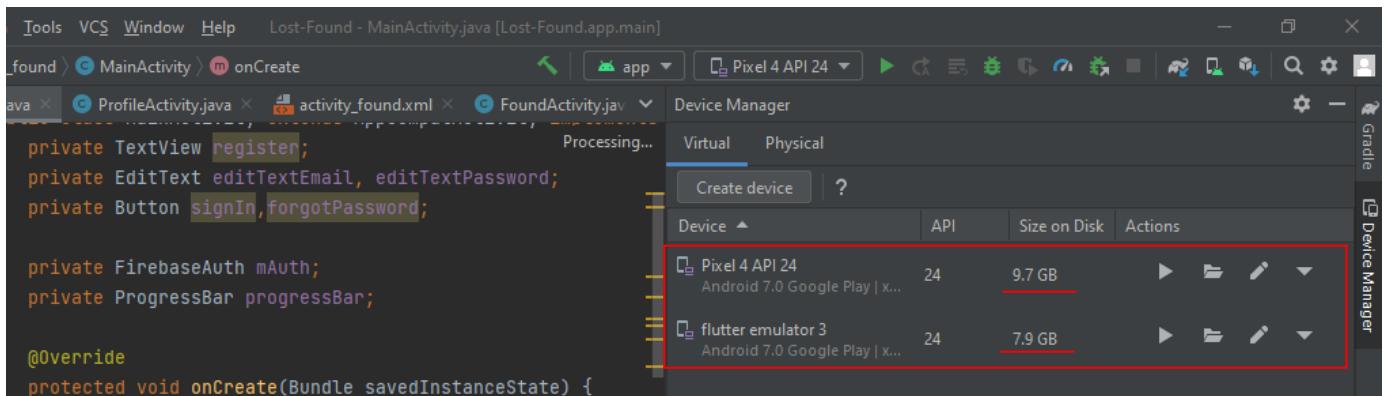


Figure 14: Emulator Memory consumed

Task Manager

File Options View

Processes Performance App history Startup Users Details Services

Name	Status	75% CPU	53% Memory	96% Disk	0% Network	Power usage	Power usage t...
<b>Apps (5)</b>							
>  Adobe Acrobat DC		0%	115.9 MB	0 MB/s	0 Mbps	Very low	Very low
>  Android Studio (4)		44.4%	705.4 MB	2.6 MB/s	0 Mbps	Very high	Moderate
>  Microsoft Word		0.1%	166.6 MB	0 MB/s	0 Mbps	Very low	Very low
>  Task Manager		0.8%	28.9 MB	0 MB/s	0 Mbps	Very low	Very low
>  Windows Explorer		0.6%	56.4 MB	0 MB/s	0 Mbps	Very low	Very low

Figure 15: Task Manager Hardware consumed

## Chapter 02

### **2.1. INRODUCTION**

This chapter describes the details and review of the existing system and the background analysis of the proposed of the lost - found Mobile application. It also comprises the functions of the proposed system through the background analysis and provides the solution to overcome the problems in the existing system. This new mobile application system will also enhance the efficiency of the existing system. It will broadly consider the technology environment and the advantages of the proposed computerized relevant system. The proposed mobile application system also provides a better solution for the equipment losers and founders in all over the world.

## **2.2. DETAILS AND REVIEW OF RELEVANT THEORY/ PREVIOUSLY DEVELOPED SYSTEM**

There are some of relevant software also available, but my application is unique from other relevant software because of mine is mobile application relevant software are desktop software system and web-based software system they don't have notifying Realtime I used Firebase Realtime database therefore it will work as a Realtime notifying application. User can collaborate at Realtime, and they can take immediate action to return or catch their contact details. This is a Mobile application that why portable feasibility also capable for my lost – found Mobile Application.

### **2.3. BACKGROUND ANALYSIS FOR THE PROPOSED SYSTEM**

This system will be developed as a Mobile application for free therefore anyone can use this application there are not any cost of usage. This application will be used to manage and deal with information about user information, lost equipment information and found equipment information. The loser and founder in dealing with particular equipment information and their collaboration. There is not any mobile application for do this job. In Sri Lanka near 1800 electronic equipment were missing by its owner oblivion at only registered hotels per year. Peoples are not only missing at hotels, not only in Sri Lanka, but electronic equipment usage is also common for entire world. Nowadays people getting up with help of electronic equipment and went sleep with also with an electronic equipment. So this Application is sinequanim for today's world.

## Chapter 03

### 3.1. INTRODUCTION

In this chapter describe what the system is required to do and development methodology of the system. Then display the top-level details of how the software system meets the requirement. It will also identify constraints on the software solution

### 3.2. DEVELOPMENT METHODOLOGY

The waterfall model is a classical model used in system development life cycle to create a system with a linear and sequential approach. It is termed as waterfall because the model develops systematically from one phase to another in downward fashion. The waterfall approach does not define the process to go back to the previous phase to handle changes in requirement. The waterfall approach is the earliest approach that was used for software development. This methodology is great for the development of lost – found application. I'm a single developer develop the entire application, so waterfall is best methodology therefore I choose this methodology.

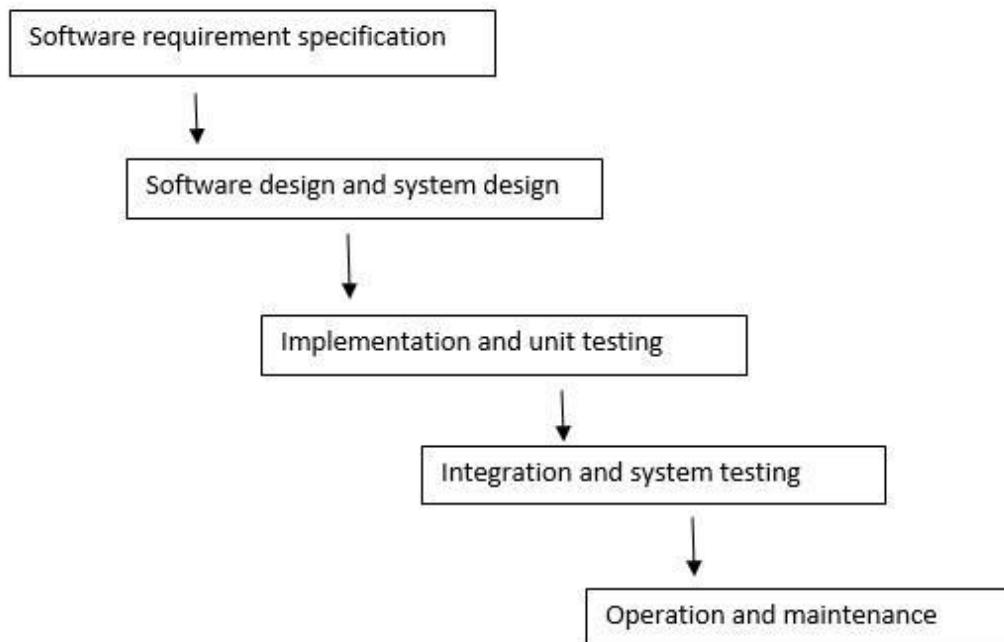


Figure 16: Methodology

## Development Environment

### Android Studio

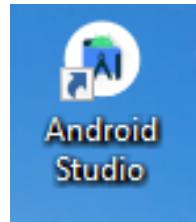


Figure 17: IDE Logo

## Development Languages

### Backend Development

- Java

### Frontend Development

- XML

### Database

- Firebase

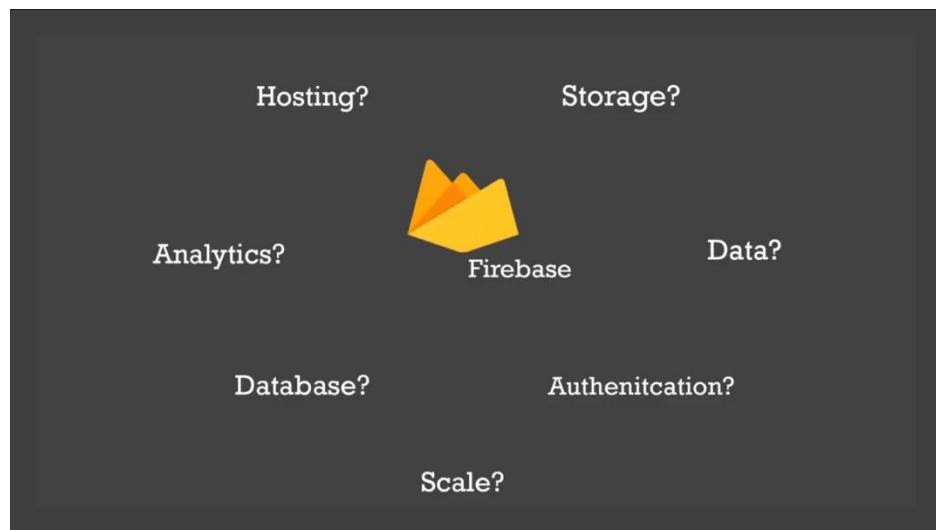


Figure 18: Firebase

### Cross Platform Data Access

- JSON

### 3.3. TOP LEVEL ARCHITECTURE

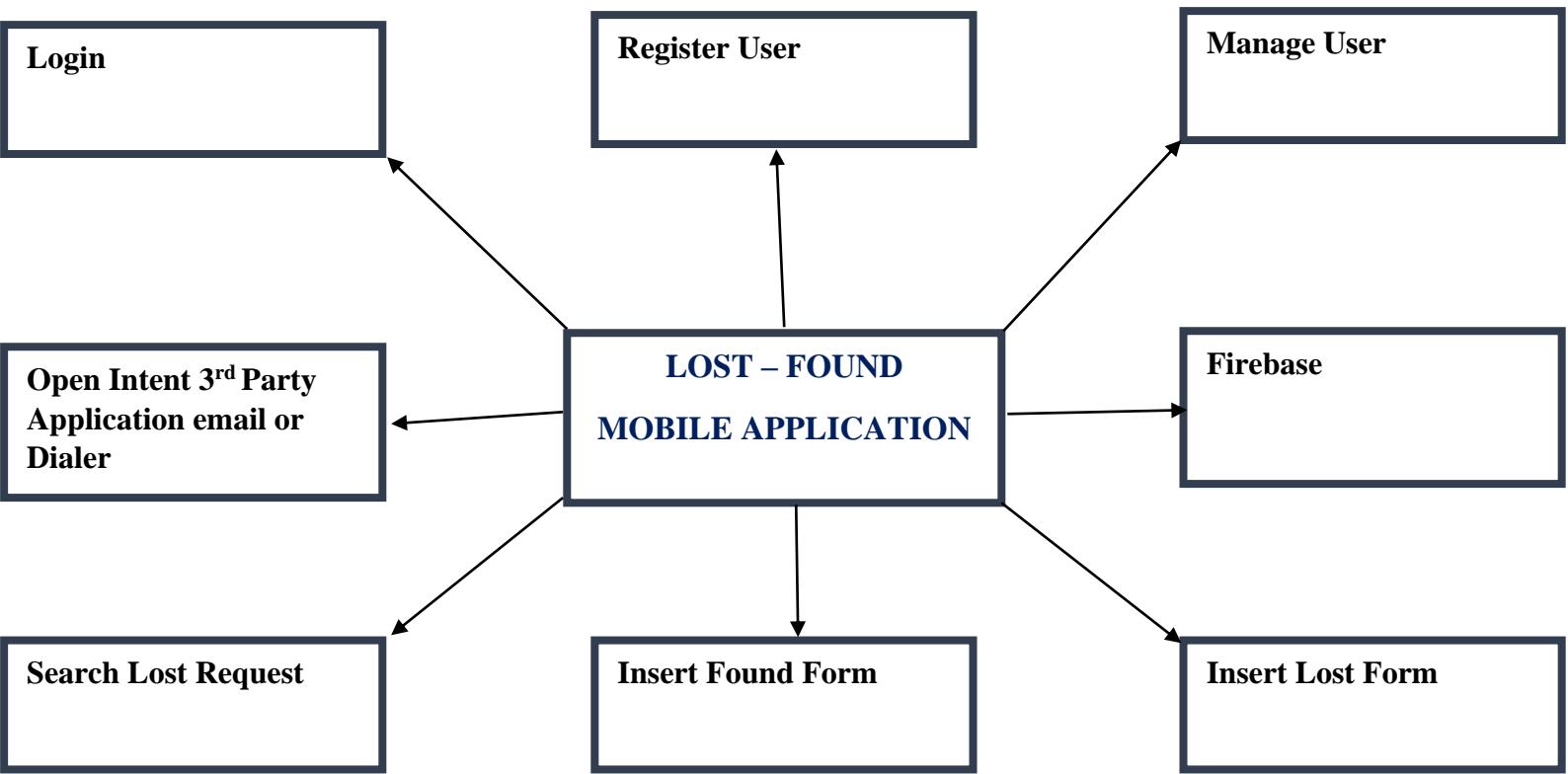


Figure 19: Top Level Architecture

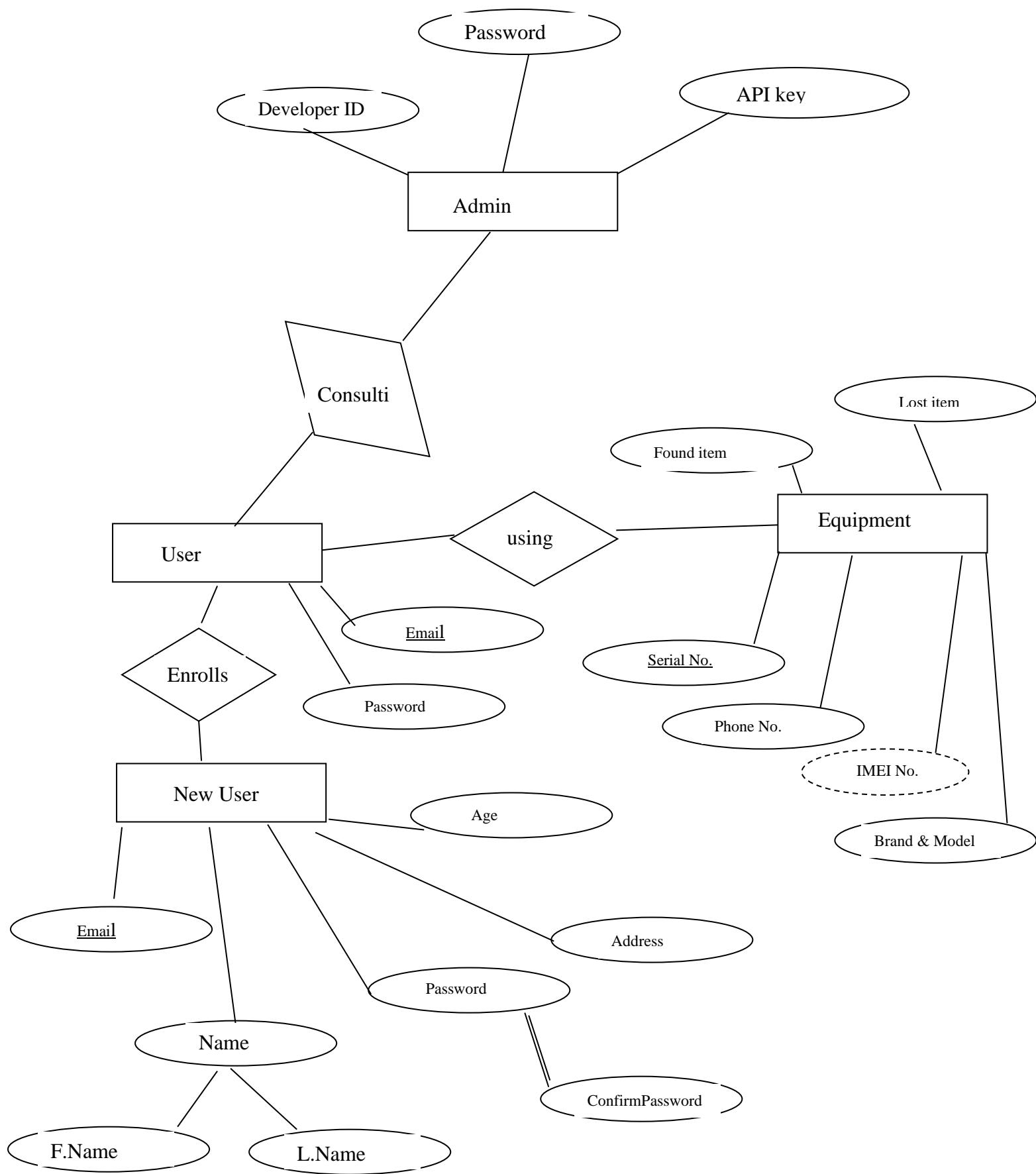


Figure 20: ER Diagram

### 3.4. USER INTERFACE DESIGN

User Interface of Application is attached below.

#### 1. Login Interface

- a. Top of the app UI I have used a banner
- b. I was added Application Logo under banner
- c. Here by inputting your pre created account you can access to the application other wish you need to create an account by hitting the **REGISTER** button
- d. Bottom of the UI I added my social media id for reference.

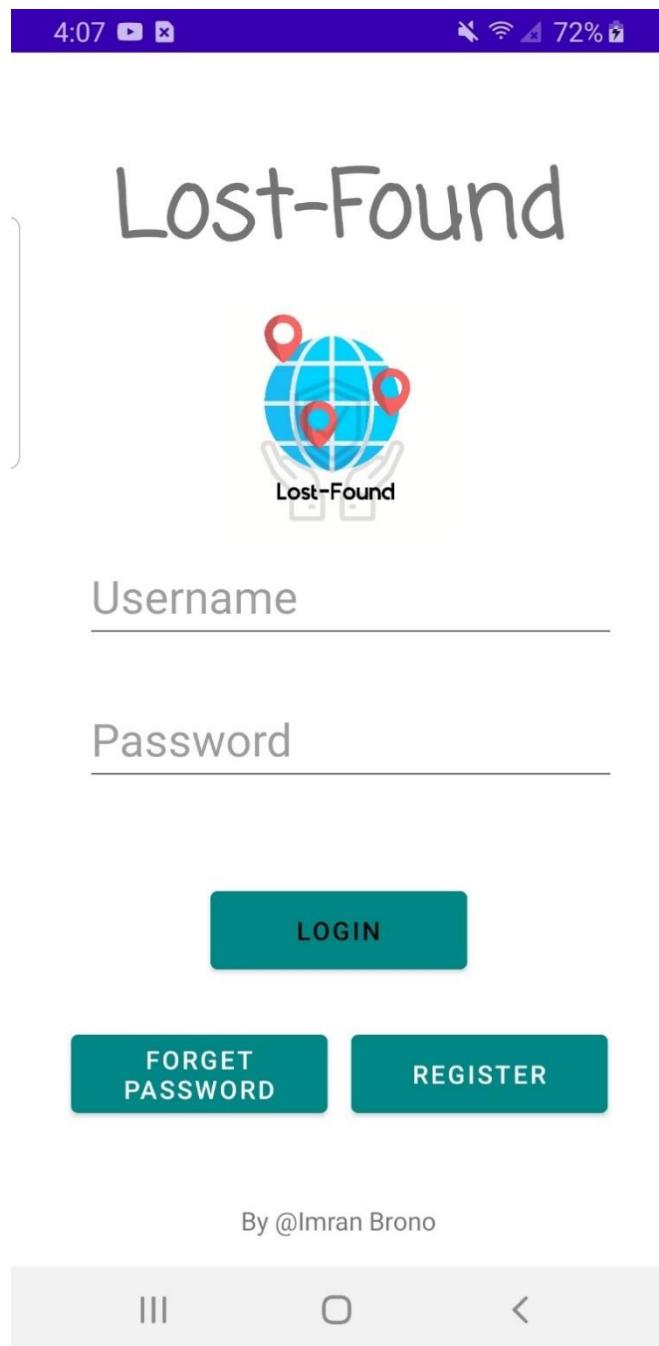


Figure 21: UI Login

## 2. Create Account Interface

- a. Here I attached app Logo with banner if we hit here application will go to login page without create an account.
- b. This area some requirements of users for reference and its store in Firebase database a special Realtime database is reserved for this.
- c. After Filling this form user want to hit register it will show the result user account was created or not if not try again else come to login page to verify your account.
- d. After creation the first attempt for login will consider as verify your account set up the Firebase server will send you a server massage link to verify your account.
- e. Using that link you can verify your account.
- f. Verification is only one time for life long. After verified you can use application normally.

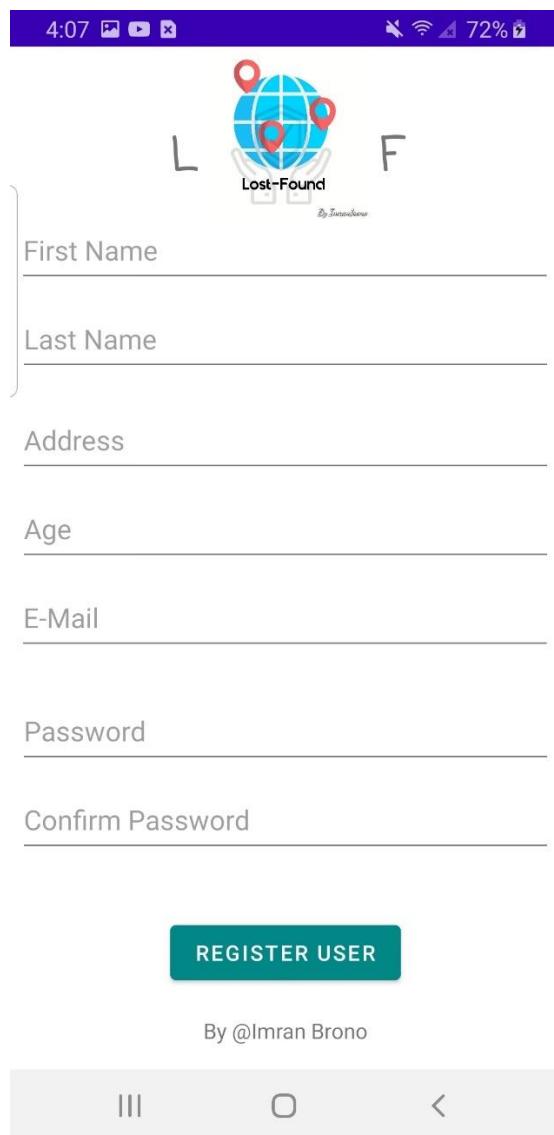


Figure 22: UI Register

### 3. Reset Password Interface

- a. Here also I attached app logo on top, under logo you can reset your password using your email.
- b. By entering your email on email requesting place, then hitting reset password icon the server will send you a link to reset your password.
- c. By entering to that link you can reset your password.

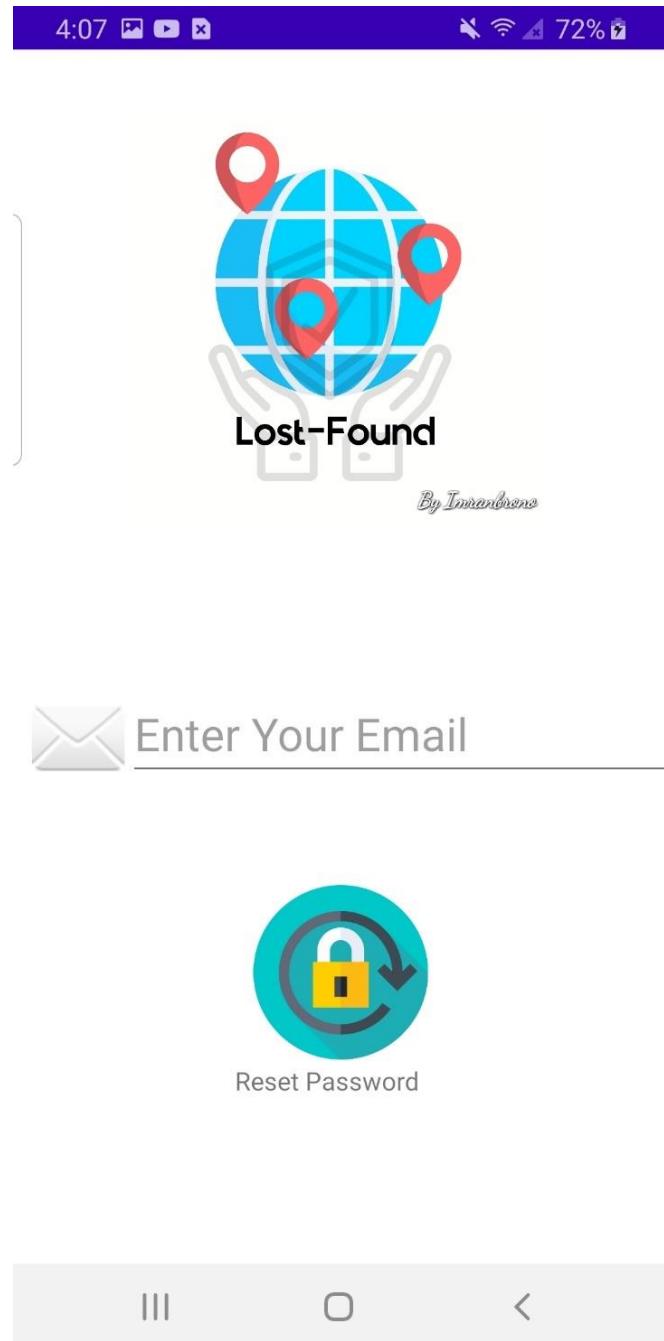


Figure 23: UI Reset password.

#### 4. After Login

- a. This is Menu page here I welcome my users first a fall than I given lost button and found button and one more button to log out that is image button.
- b. By hitting the lost button new intent will bring you to a new page that is place for search lost item and make lost request
- c. By hitting the found button there also same like above but here you only can insert information about founded item
- d. By touch on that logout button your logout from app.

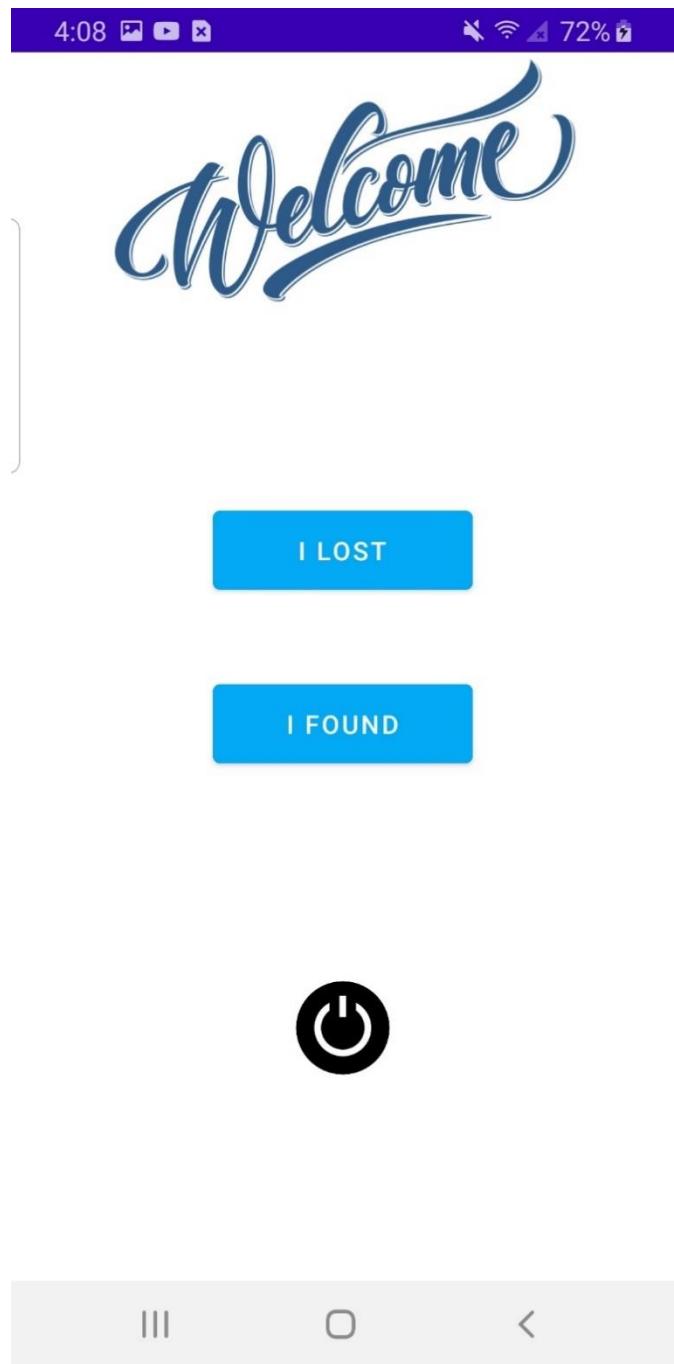


Figure 24: UI Menu.

## 5. Lost Activity

- a. In this page for the security purpose, I hide information and here only search bar with hint show to enter electronic equipment serial no:
- b. By entering serial no and touch the search icon if your item was add on database via found Activity than only the system will bring you to your default email system to send a manual email.



Figure 25: Lost Activity.

6. Found Activity

- a. In here there is a form to complete that requesting about that particular founded item.
- b. Fill the form and hitting upload will upload your form information to the firebase database.
- c. One more function also here to logout by hitting logout icon.



Item

---

Brand and Model

---

Serial Number

---

IMEI No (Optional)

---

Your Contact No: (Optional)

---

UPLOAD



Figure 26: Found Activity.

## Chapter 04

### 4.1. INTRODUCTION

In this chapter, I discussed about system Testing and Debugging. There are several methods available to test android applications. The Android App Testing – Test Your App capable to run your apps on real device with fire base test labs it's improve app quality with firebase test lab. Testing is a rich and important field, and many more distinctions could be drawn. Many developers are reluctant to spend time on testing, seeing it as time subtracted from "real" development, so each hour spent developing tests can amply pay back for it by finding defects from Sales Management System. Also it includes software testing strategies and debugging methods for the sales management system.

### 4.2. DIFFERENT TEST CASES

A test case has components that describe an input, action or event and an expected response, to determine if a feature of an application is working correctly. Test cases are often referred to as test scripts, particularly when written. Written test cases are usually collected into test suites.

#### 4.2.1. TEST CASE FOR REGISTER USER

Here I test all kind of Lost Activity Components.

The figure consists of three side-by-side screenshots of a mobile application interface. Each screenshot shows a registration form with various fields and validation messages. The top of each screenshot features a navigation bar with icons for back, forward, and other functions, along with a battery indicator at 100%.

**Screenshot 1:** Shows the first two fields of the registration form. The "First Name" field is empty, and a red error message "First Name is Required!" is displayed below it. The "Last Name" field contains the value "Imran".

First Name	Imran
Last Name	
Address	
Age	
E-Mail	
Password	
Confirm Password	

**Screenshot 2:** Shows the last three fields of the registration form. The "Last Name" field is empty, and a red error message "Last Name is Required!" is displayed below it. The "Address" field contains the value "Pottuvil".

First Name	Imran
Last Name	Brono
Address	Pottuvil
Age	
E-Mail	
Password	
Confirm Password	

**Screenshot 3:** Shows the final field of the registration form. The "Age" field is empty, and a red error message "Age is Required!" is displayed below it. The other fields are empty.

First Name	Imran
Last Name	Brono
Address	Pottuvil
Age	
E-Mail	
Password	
Confirm Password	

**Buttons:** Each screenshot shows a green "REGISTER USER" button at the bottom center. Below each button, the text "By @Imran Brono" is visible.

			
L	L	L	
F	F	F	
Imran	Imran	Imran	
Brono	Brono	Brono	
Pottuvil	Pottuvil	Pottuvil	
23	23	23	
E-Mail	imranbrono	imranbrono@gmail.com	
E-Mail is Compulsory!		Please provide valid Email!	
Password	Password	Password	>Password is Required!
Confirm Password	Confirm Password	Confirm Password	

**REGISTER USER**

By @Imran Brono

**REGISTER USER**

By @Imran Brono

**REGISTER USER**

By @Imran Brono

			
L	L	L	
F	F	F	
Imran	Imran	Imran	
Brono	Brono	Brono	
Pottuvil	Pottuvil	Pottuvil	
23	23	23	
imranbrono@gmail.com	imranbrono@gmail.com	imranbrono@gmail.com	
.....	.....	.....	
Confirm Passwo	Confirm Password	Both Password fields must be identic	
Password lenght should be 8 characters!		Confirm Password is Required!	
<b>REGISTER USER</b>	<b>REGISTER USER</b>	<b>REGISTER USER</b>	
By @Imran Brono		By @Imran Brono	

Figure 27: Test user register



Imran

Brono

Pottuvil

23

[imranbrono@outlook.com](mailto:imranbrono@outlook.com)

.....

.....

User has been registered successfully

REGISTER USER

By @Imran Brono



Figure 28: Registration Success

#### 4.2.2. TEST CASE FOR USER LOGIN

This is Login User Interface and its Main part of the application here email and password will check the firebase database and return your eligibility to enter the application. One more option I added here that is after the registration, 1<sup>st</sup> attempt to the login will request to verify the email after verifying you can enter the application.

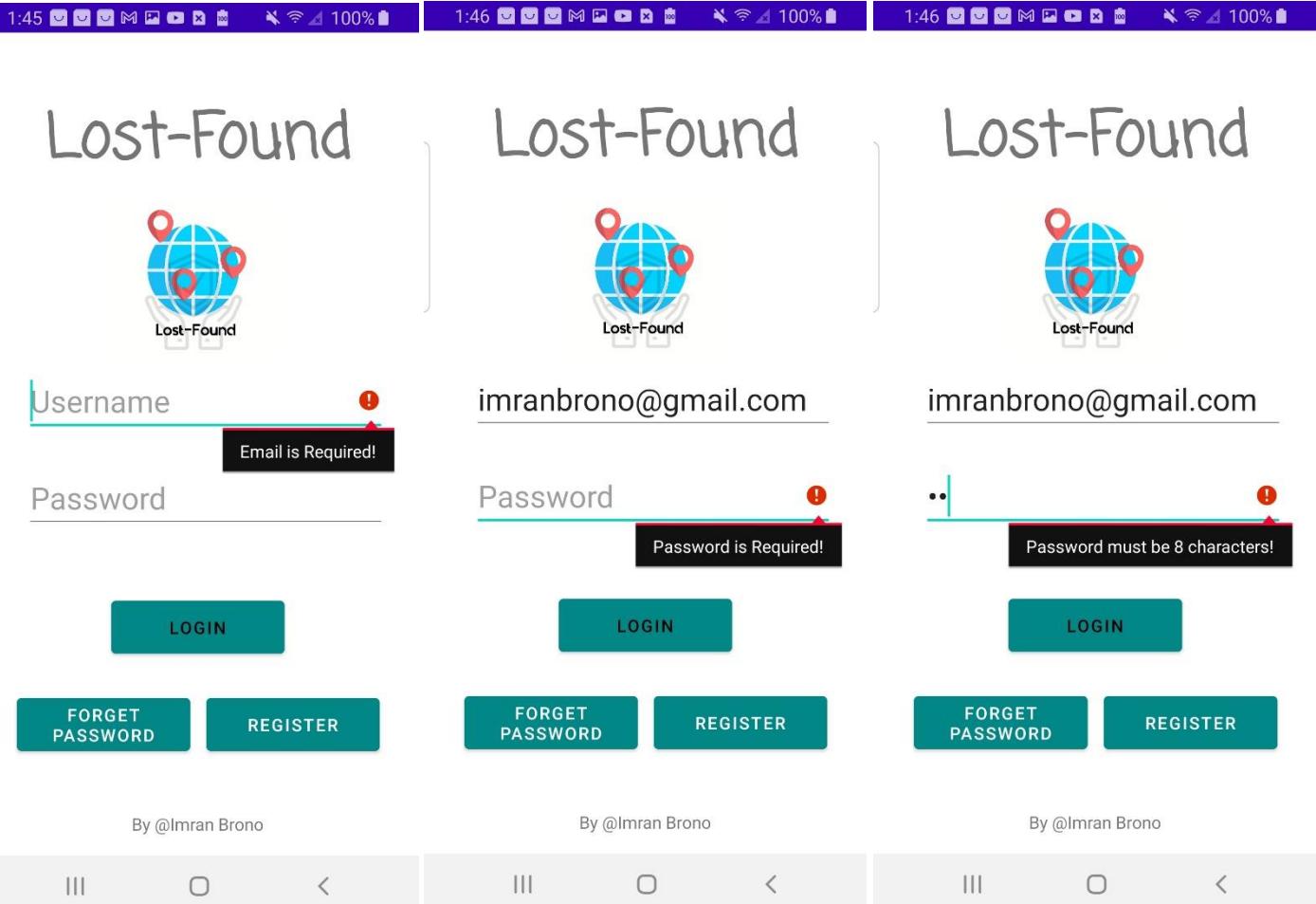


Figure 29: Test User Login

If user used any invalid email or wrong Password the bellow Toast Message “Failed to login! Please check your credential” will show to the valuable user else your can access to the application.

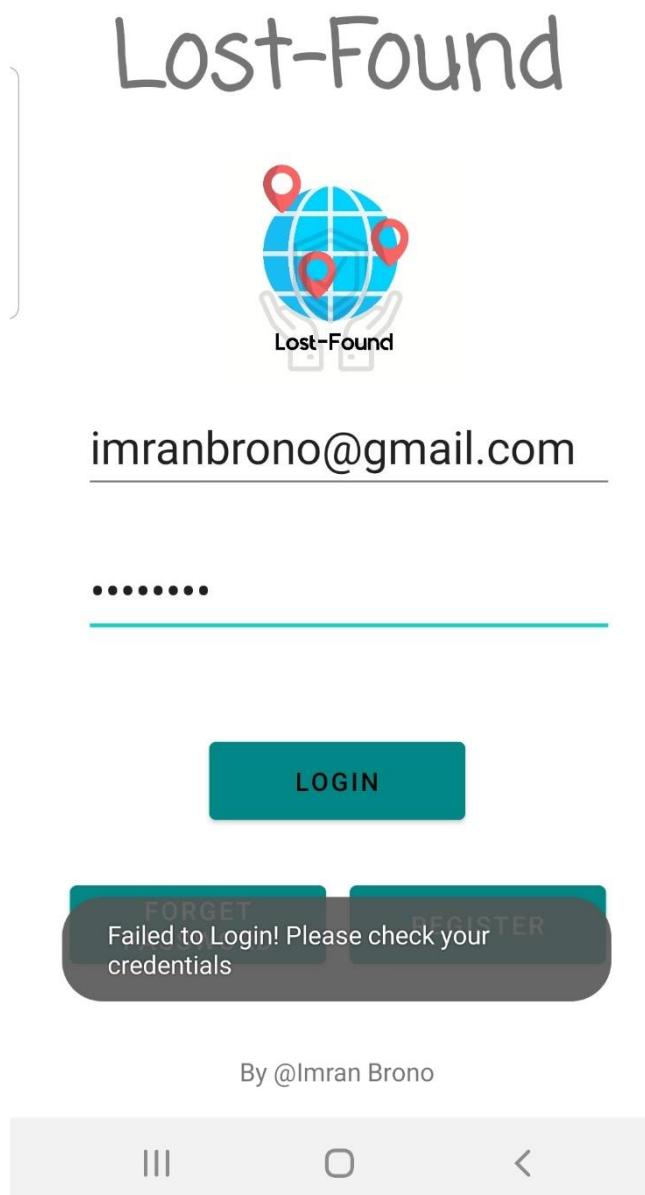
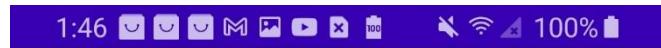


Figure 30: Test Wrong User

Here is a tested email verification sample was attached for reference this is Firebase Console prewritten default email for verify the account.

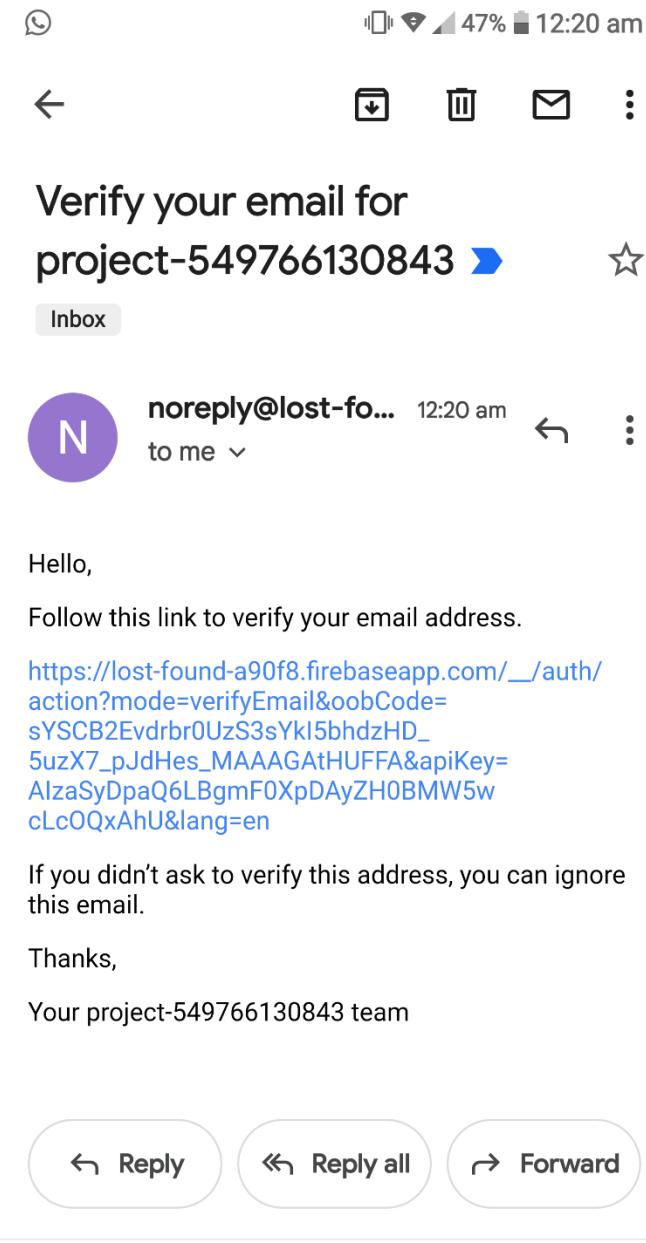


Figure 31: Verification email



← lost-found-a90f8.firebaseio.com ⋮

## Your email has been verified

You can now sign in with your new account

Figure 32: Verified Screen

#### 4.2.3. TEST CASE FOR RESET PASSWORD

This is interface for incase if user forgot theirs' password, they could reset using this interface. By hitting forget Password button on Main Activity it will bring user to this Activity here also I used Logo on Top and in middle requesting email in the edit text field by entering particular user email and clicking under the email reset password image button it will request to firebase database to reset their password by sending email to the user's email. They can reset their password with the help of that email.

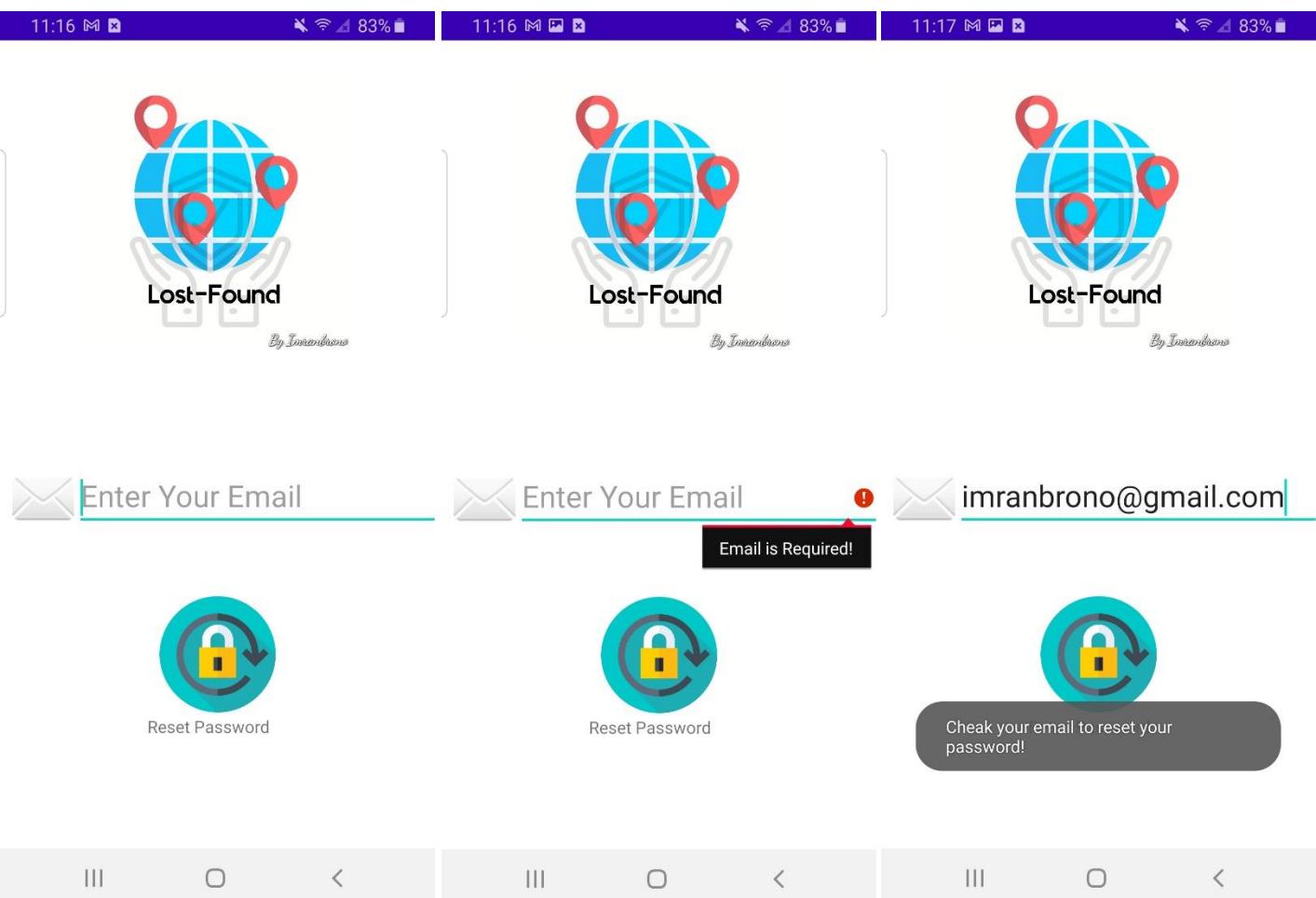


Figure 33: Test User Reset Password



Figure 34: Reset Icon

11:20

82%



## Reset your password for project-549766130843



Inbox

N

noreply@lost-f... 3 days ago  
to me ▾



Hello,

Follow this link to reset your project-549766130843  
password for your [latadathu@gmail.com](mailto:latadathu@gmail.com) account.

[https://lost-found-a90f8.firebaseio.com/\\_auth/  
action?mode=resetPassword&oobCode=\\_  
8o84fR09RIt6XnXxUJUeUkYhtACxPo  
t85BuedpPEmYAAAGAtRNPXQ&apiKey=  
AIzaSyDpaQ6LBgmF0XpDAyZH0BMW5w  
cLcOQxAhU&lang=en](https://lost-found-a90f8.firebaseio.com/_auth/action?mode=resetPassword&oobCode=_8o84fR09RIt6XnXxUJUeUkYhtACxPo t85BuedpPEmYAAAGAtRNPXQ&apiKey=AIzaSyDpaQ6LBgmF0XpDAyZH0BMW5w cLcOQxAhU&lang=en)

If you didn't ask to reset your password, you can  
ignore this email.

Thanks,

Your project-549766130843 team

Reply

Reply all

Forward

Figure 35: Test Reset Email

#### 4.2.4. TEST MENU

After the Login this menu Activity will be appear to the user in here top of the page there is a welcome Message and under the welcome message there are two button, I LOST, and I FOUND are holds to intent pages such as Lost Activity and Found Activity under those buttons there is a Logout icon also to logout from our app. By clicking on I LOST button user can access the Lost page and by clicking the I FOUND button user can access Found page.

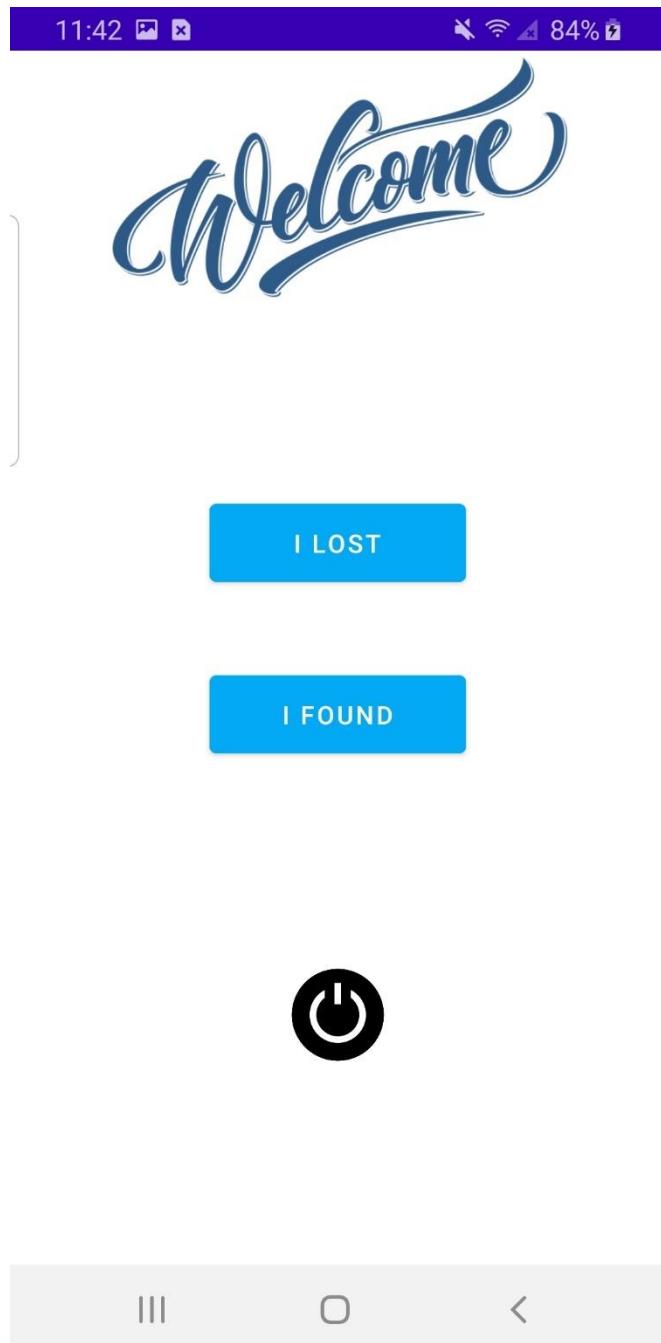


Figure 36: Test Menu

#### 4.2.5. TEST CASE FOR LOST ACTIVITY

Here search bar and a search icon are visible to the user by entering serial number on that particular search bar and clicking on the search icon would return if the item added than only it will bring user to send an email using user default email else it shows a toast message as “You item is not found”. Here for the security reason item serial number is important thing. Serial number must engrave in every electronic equipment, and it is visible for mentioner.

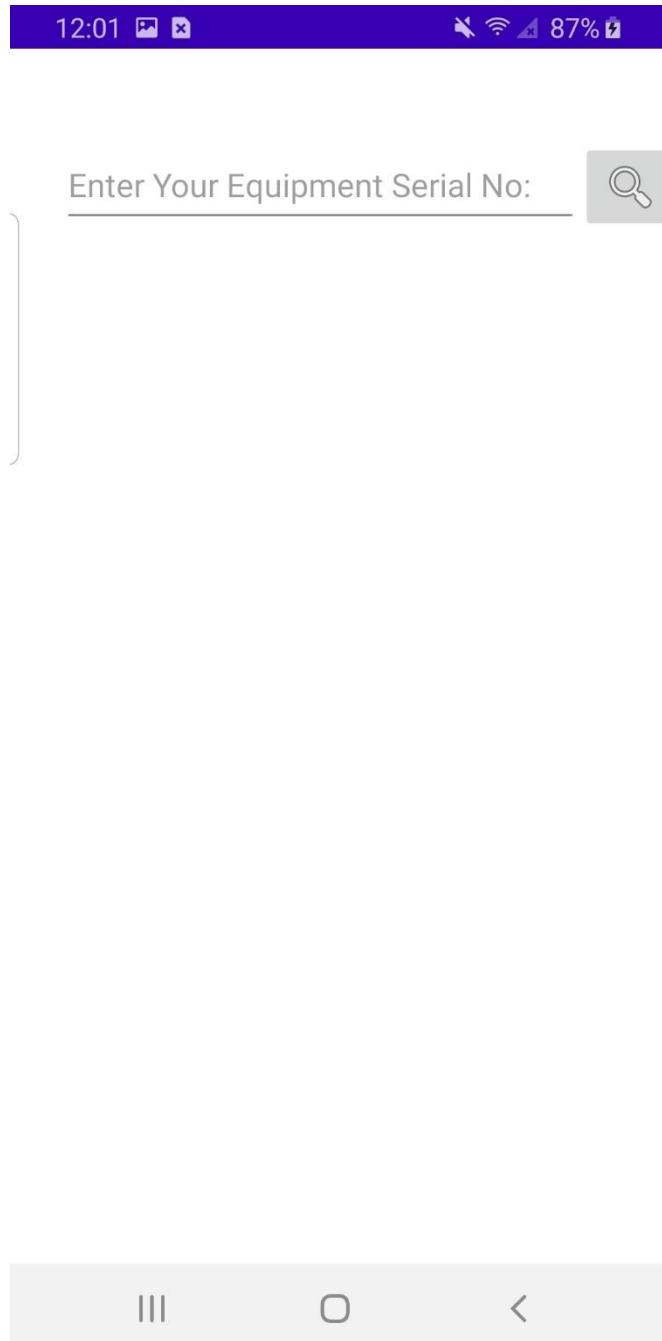


Figure 37: Test Lost Activity

#### 4.2.6. TEST CASE FOR FOUND ACTIVITY

By clicking on menu page I FOUND button user can access this page, here user want to fill some details about that found item such as, item, brand and model, serial number are compulsory and IMEI number and Your Contact Number are optional thing that user want to add and clicking on upload button will pack user input to the firebase database and the firebase database will only work with the primary id serial number for the looser information and it make interconnection with looser and founder. Bottom of the page there is a logout icon to logout.

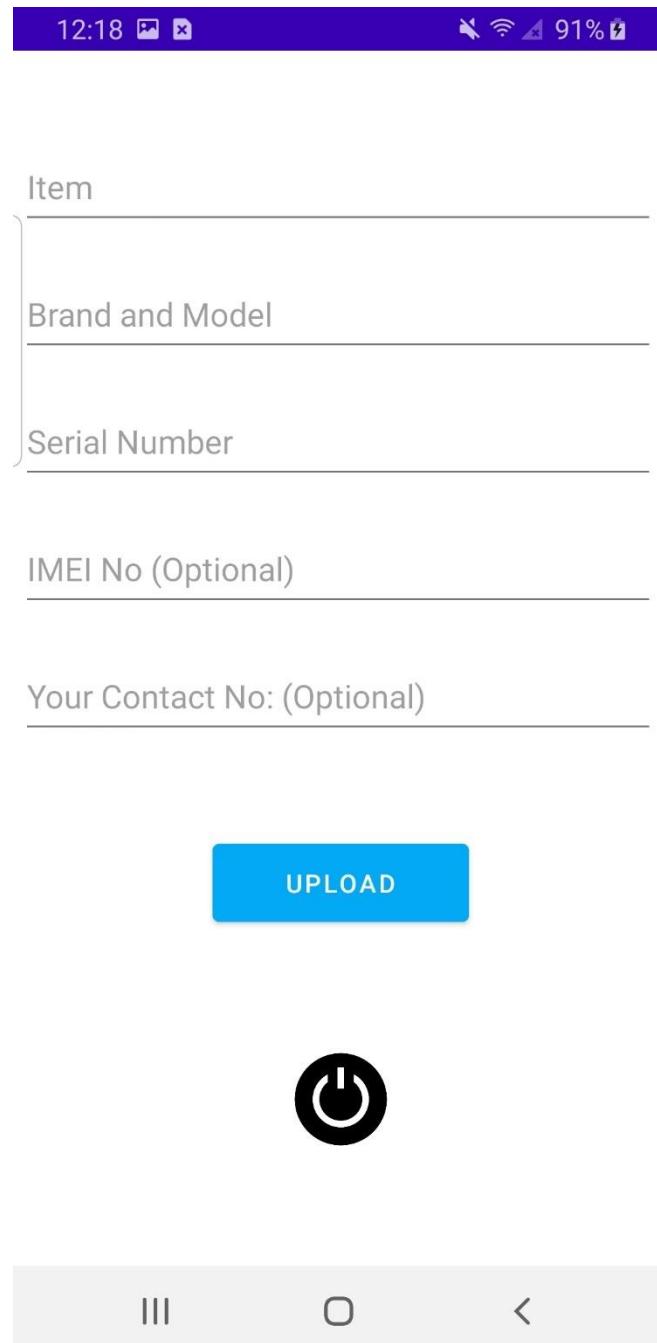


Figure 38: Test Found Activity

#### 4.2.7. TEST CASE FOR LOGOUT

I used a unique logout icon to logout I used this icon on menu page and found page its function is Sign-out the user from the Firebase.

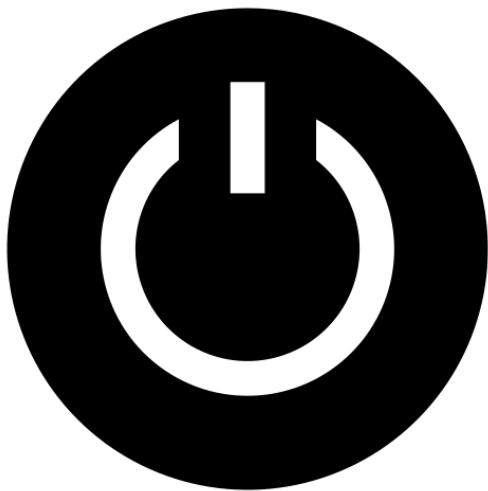


Figure 39: Logout Icon.

## 4.3. ANDROID APP TESTING

### Run your app on real devices

To ensure your app quality, Firebase Test Lab provides you with physical and virtual devices that allow you to run tests that simulate actual usage environments. I have used my own Samsung Galaxy S8 phone to test my application via connecting with a data cable and enabling developer mode. To turn on developer mode 1 Go to "Settings", then tap "About device" or "About phone". 2 Scroll down, then tap "Build number" seven times. Depending on your device and operating system, you may need to tap "Software information", then tap "Build number" seven times. 3 Enter your pattern, PIN, or password to enable the Developer options menu.

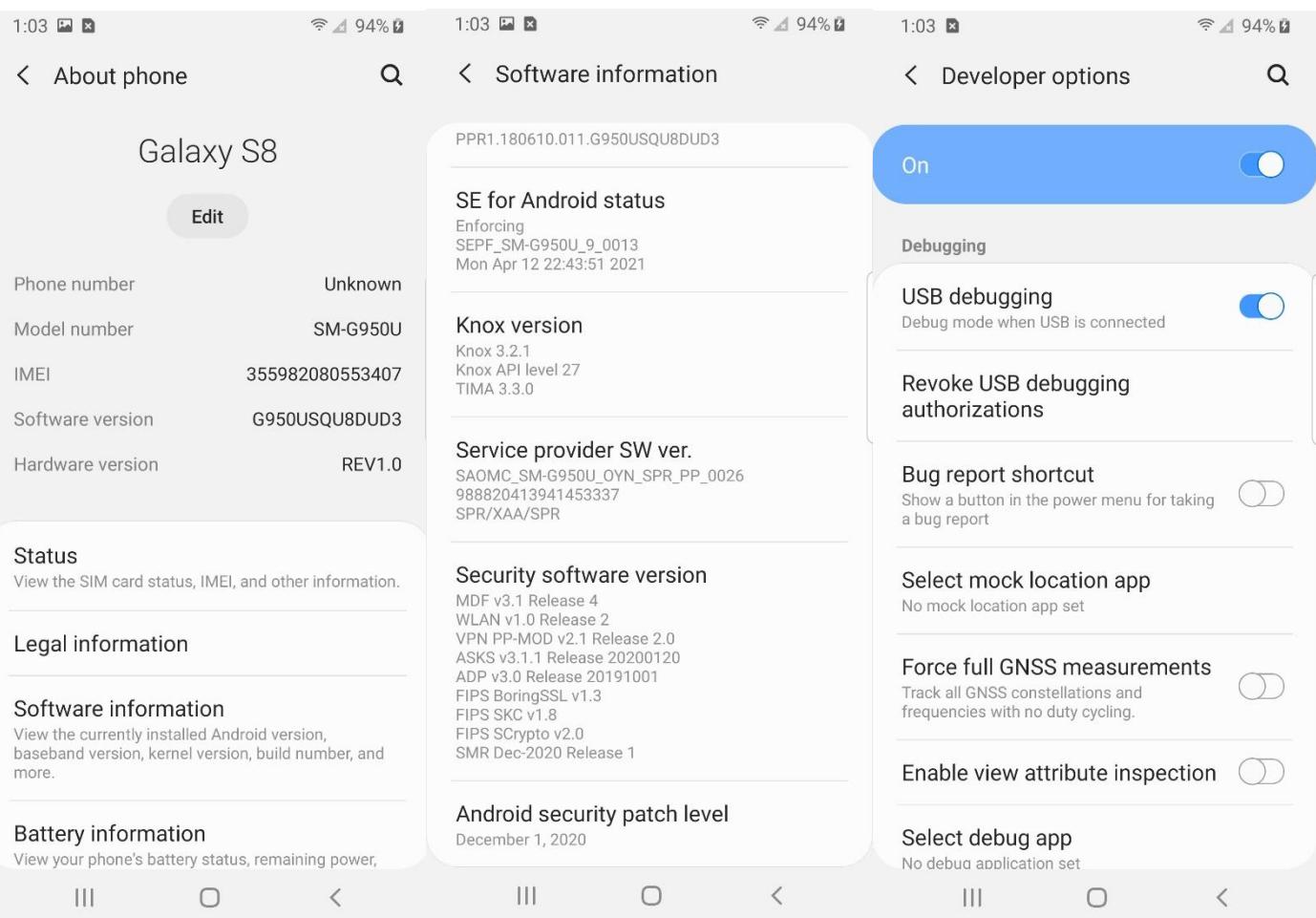


Figure 40: Test Device Information

## 4.4. ANDROID TESTING STRATEGIES

A correct android testing strategy should include the following

1. Unit Test
2. Integration Test
3. Operational Test
4. System Test

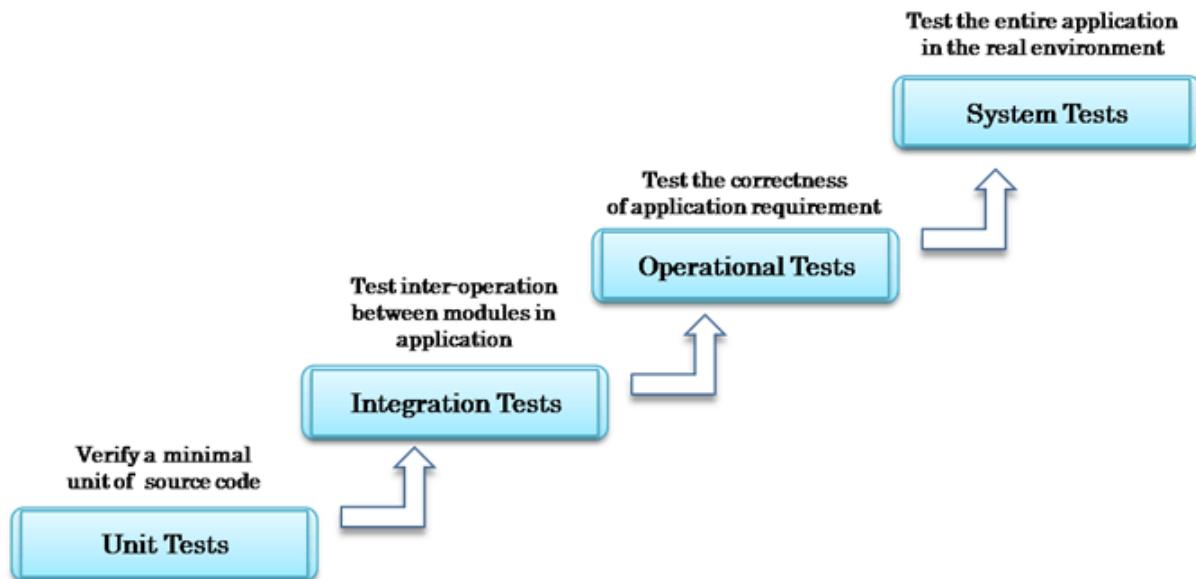


Figure 41: Testing Strategies

### Unit tests

Unit Tests include sets of one or more programs which are designed to verify an atomic unit of source code, such as a method or a class.

Android platform comes pre-integrated Junit 3.0 framework. It's open source framework for automating Unit Testing. Android Testing Framework is powerful tool for developer to write the effective unit test program.



Figure 42: Unit Test

The integration of Android and JUnit framework. An addition to Unit Testing is User Interface (UI) tests. These tests relate to UI components of your target application. UI tests ensure that your application return the correct UI output in response to sequence of user actions on device.

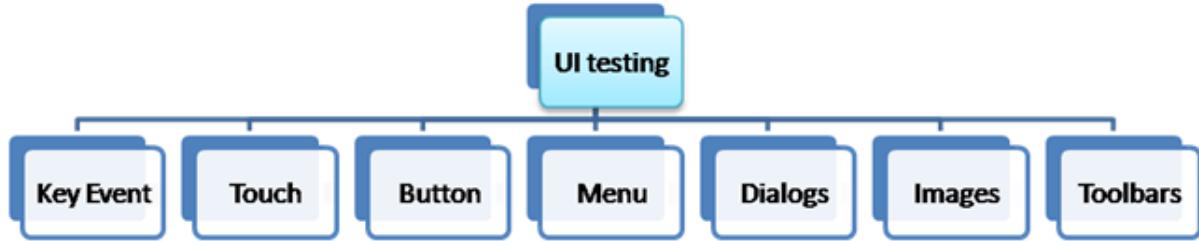


Figure 43: Unit Test components

Common user UI actions on application. The common way to performance UI tests on device is Android [Instrumentation](#). But this has performance issues. One of the best tools to conduct UI testing on Android is [Robotium](#).

## Integration tests

In [Integration Testing](#), all unit tested modules, are combined, and verified. In Android, integration tests often involve checking integration with Android components such as Service testing, Activity testing, Content Provider testing, etc

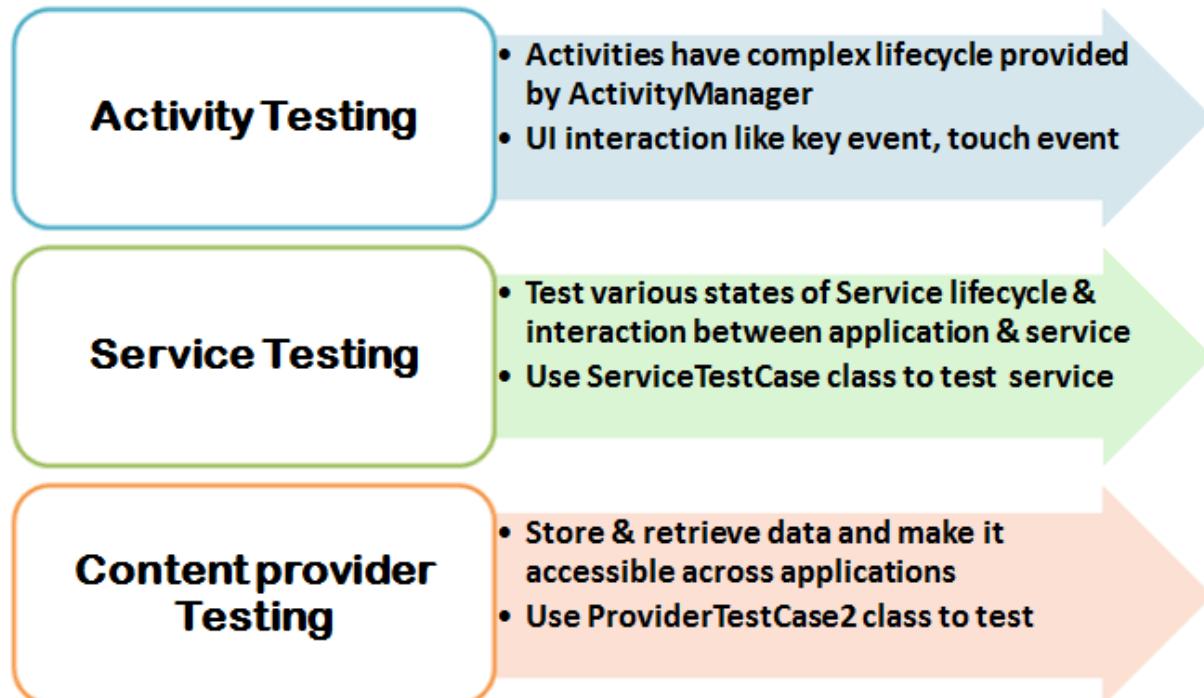


Figure 44: UI Integration Test

Types of integration test on Android. There's many testing frameworks are used to conduct integration test for Android such as Troyd, Robolectric, Robotium.

## Operational tests

Operational are also called Functional Tests or Acceptation Tests. They are high level tests designed to check the completeness and correctness of application.

In Android, [FitNesse](#) is open-source framework that makes it easy to conduct operational tests for target application.

## System tests

In [System Testing](#) the system is tested as a whole and the interaction between the components, software and hardware is checked. In Android, System Testing normally includes

- GUI tests
- Usability tests
- Performance tests
- Stress tests

In the above list, **Performance Testing** is given more focus. You can use tools like [Traceview](#) to conduct performance test on Android. This tool can help you debug your application and profile its performance.

## 4.5. ANDROID APPLICATION DEBUGGING

Android Studio provides a debugger that allows you to do the following and more:

- Select a device to debug your app on.
- Set breakpoints in your Java, Kotlin, and C/C++ code.
- Examine variables and evaluate expressions at runtime.

This page includes instructions for basic debugger operations. For more documentation, also see the [IntelliJ IDEA debugging docs](#).

- **Enable debugging on your device:**

If you're using the emulator, this is enabled by default. But for a connected device, you need to [enable debugging in the device developer options](#).

- **Run a debuggable build variant:**

You must use a [build variant](#) that includes [debuggable true](#) in the build configuration. Usually, you can just select the default "debug" variant that's included in every Android Studio project (even though it's not visible in the build.gradle file). But if you define new build types that should be debuggable, you must add `debuggable true` to the build type:

Start debugging

- Set some breakpoints in the app code.
- In the toolbar, select a device to debug your app on from the target device drop-down menu.



Figure 45: Debug Toolbar

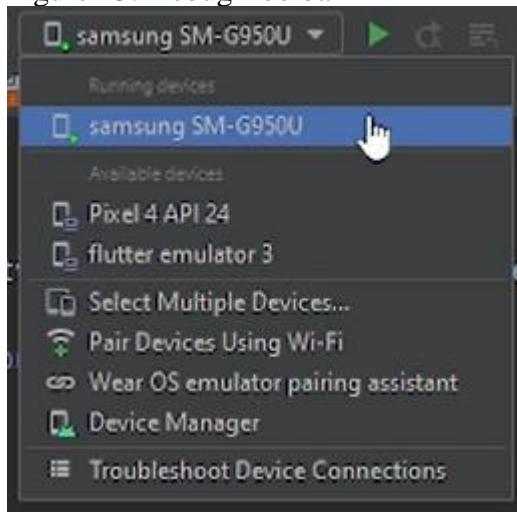


Figure 46: Debug Devices

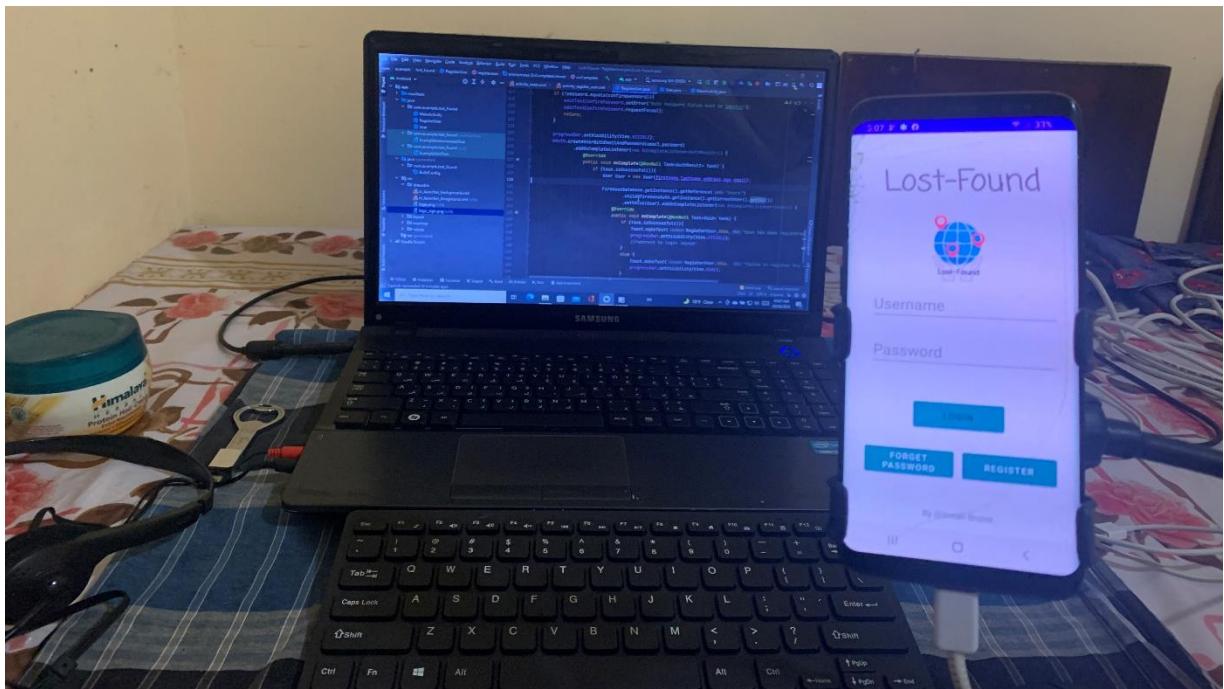
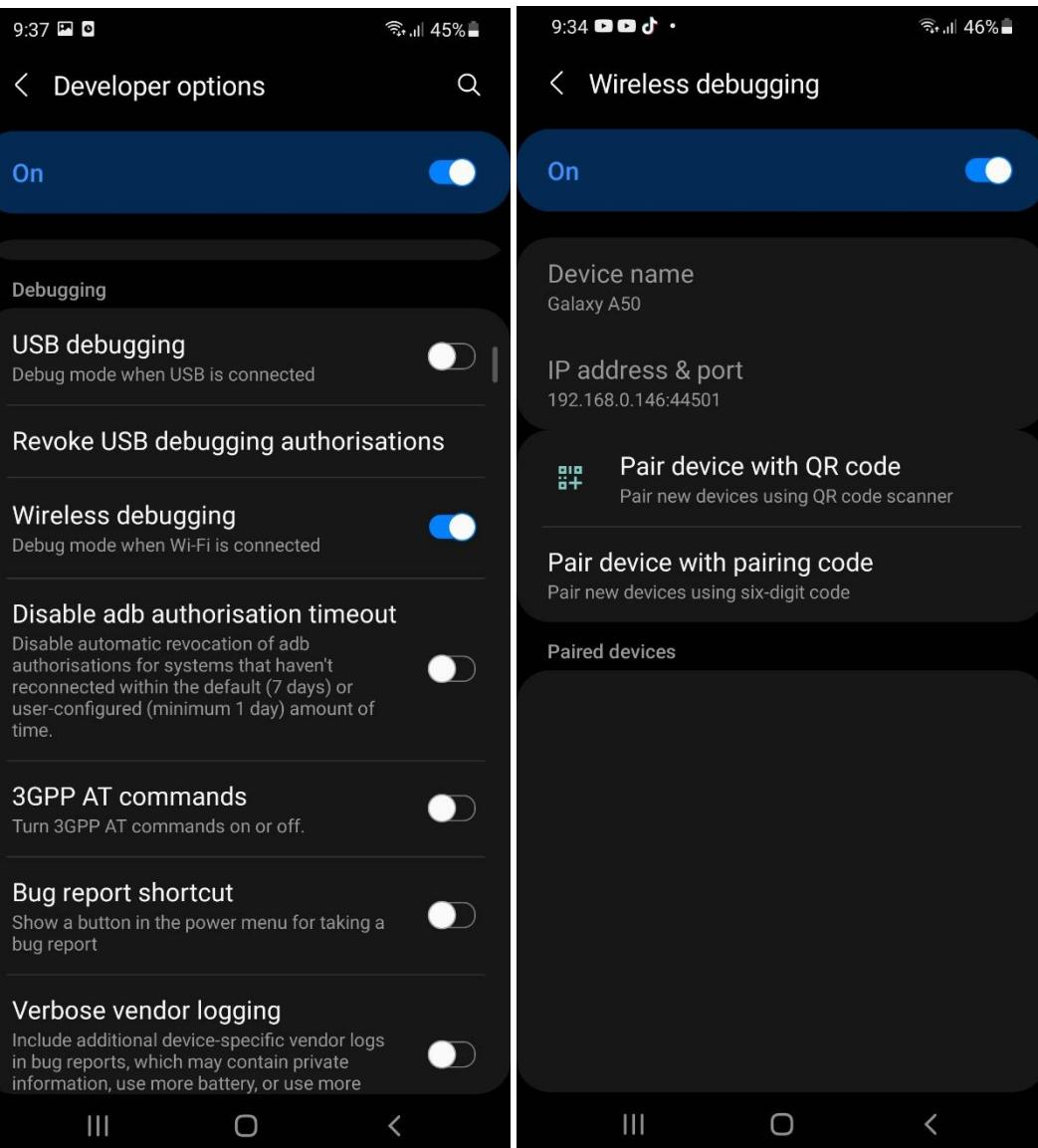


Figure 47: Debug Using USB.

- The above Figure 47 is representing debug using USB Debugging Method.



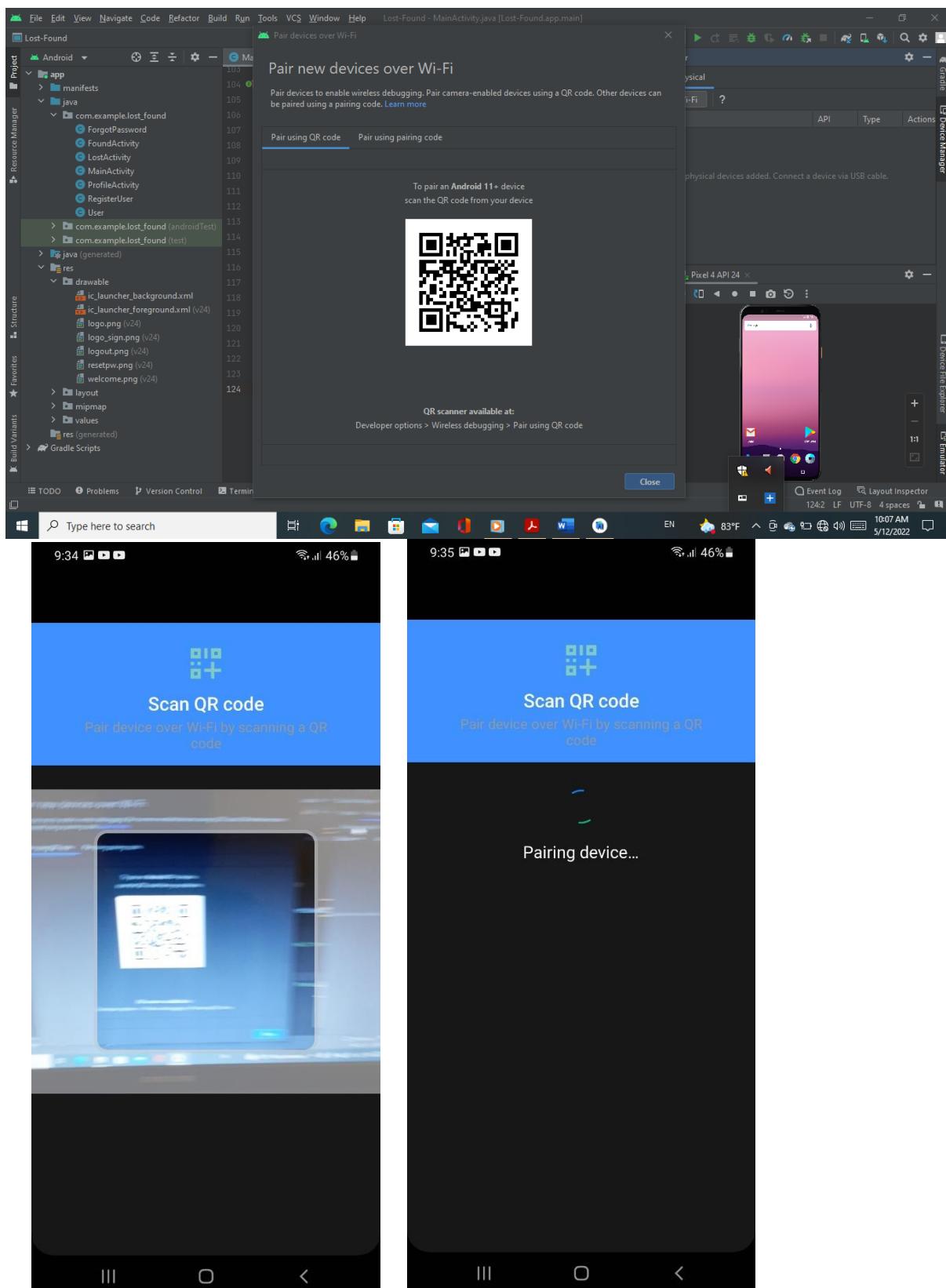


Figure 49: Wi- Fi Debugging Android 11.

- Above Figure 48 & Figure 49 representing Wi-Fi debugging Method and its only available for Android 11 and Android 12.

- If you see a dialog asking if you want to "switch from Run to Debug," that means your app is already running on the device and it will restart in order to begin debugging. If you'd rather keep the same instance of the app running, click Cancel Debug and instead attach the debugger to a running app.

Otherwise, Android Studio builds an APK, signs it with a debug key, installs it on your selected device, and runs it. If you add C and C++ code to your project, Android Studio also runs the LLDB debugger in the Debug window to debug your native code.

- If the Debug window is not open, select View > Tool Windows > Debug (or click Debug in the tool window bar), and then click the Debugger tab, as shown in figure 50.



Figure 50: Debugger

## Chapter 05

### 5.1. INRODUCTION

This chapter involves implementation, maintenance and outlines the user manual with screen shot. This also discuss about system conversion process for the proposed system in detail. Further it provides brief introduction code structure, document structure, requirements and problem encounter with the implementation, and it also outlines details on system maintenance and system maintenance process model.

### 5.2. CODE STRUCTURE

In this section, list down the main coding of LOST – FOUND Mobile Application and describe about the coding of Application UI XML code, Application Backend Java Code, Database connection, and Firebase to JSON Realtime code structure.

#### UI XML Codes

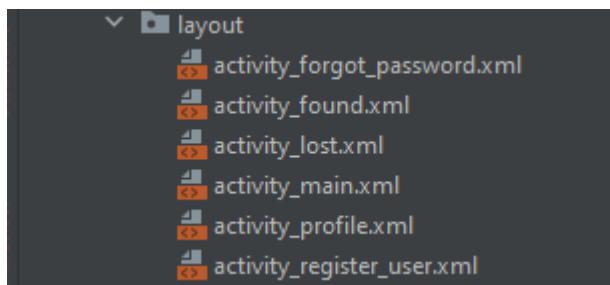


Figure 51: XML files

Here I attached only one XML code structure

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity">
8
9      <TextView
10         android:id="@+id/banner"
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:layout_margin="50dp"
14         android:fontFamily="casual"
15         android:text="Lost-Found"
16         android:textAlignment="center"
17         android:textSize="50sp"
18         android:textStyle="bold"
19         app:layout_constraintLeft_toLeftOf="parent"
20         app:layout_constraintRight_toRightOf="parent"
21         app:layout_constraintTop_toTopOf="parent" />
22
23      <ImageView
24         android:id="@+id/imageView"
25         android:layout_width="145dp"
         android:layout_height="138dp"
         android:layout_marginTop="8dp"
         app:layout_constraintEnd_toEndOf="parent"
         app:layout_constraintStart_toStartOf="parent"
         app:layout_constraintTop_toBottomOf="@+id/banner"
         app:srcCompat="@drawable/logo" />
```

```
75  
76  
77    android:layout_marginTop="24dp"  
78    android:layout_marginEnd="32dp"  
79    android:backgroundTint="@color/teal_700"  
80    android:text="Register"  
81    app:layout_constraintEnd_toEndOf="parent"  
82    app:layout_constraintHorizontal_bias="1.0"  
83    app:layout_constraintStart_toStartOf="parent"  
84    app:layout_constraintTop_toBottomOf="@+id/login" />  
85  
86  
87  
88  
89    <Button  
90        android:id="@+id/forgotpassword"  
91        android:layout_width="141dp"  
92        android:layout_height="55dp"  
93        android:layout_marginStart="32dp"  
94        android:backgroundTint="@color/teal_700"  
95        android:text="Forget Password"  
96        app:layout_constraintEnd_toStartOf="@+id/register"  
97        app:layout_constraintHorizontal_bias="0.075"  
98        app:layout_constraintStart_toStartOf="parent"  
99        app:layout_constraintTop_toTopOf="@+id/register" />  
100  
101  
102  
103  
104  
105    <TextView  
106        android:id="@+id/textView"  
107        android:layout_width="wrap_content"  
108        android:layout_height="wrap_content"  
109        android:layout_marginBottom="16dp"  
110        android:text="By @Imran Brono"  
111        android:textAppearance="@style/TextAppearance.AppCompat.Small"  
112        app:layout_constraintBottom_toBottomOf="parent"  
113        app:layout_constraintEnd_toEndOf="parent"  
114        app:layout_constraintStart_toStartOf="parent" />  
115  
116  
117  
118  
119    <ProgressBar  
120        android:id="@+id/progressBar"  
121        style="?android:attr/progressBarStyleLarge"  
122        android:layout_width="wrap_content"  
123        android:layout_height="wrap_content"  
124        android:layout_centerInParent="true"  
125        android:visibility="gone"  
126        app:layout_constraintBottom_toBottomOf="parent"  
127        app:layout_constraintEnd_toEndOf="parent"  
128        app:layout_constraintStart_toStartOf="parent"  
129        app:layout_constraintTop_toTopOf="parent"  
130        tools:ignore="MissingConstraints" />  
131  
132  
133    </androidx.constraintlayout.widget.ConstraintLayout>
```

Figure 52: XML Code of Main page

## Java Codes

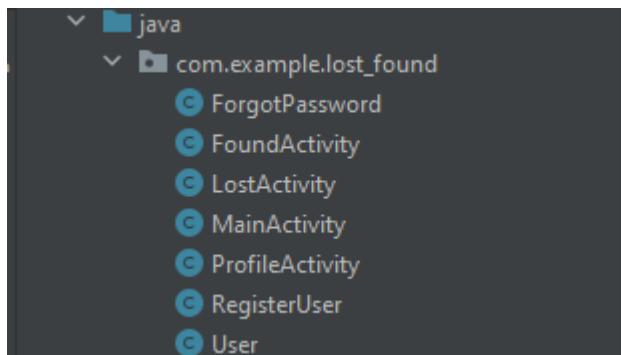


Figure 53: Java Files

```
1  package com.example.lost_found;
2
3  import ...
4
5
6  public class MainActivity extends AppCompatActivity implements View.OnClickListener{
7      private TextView register;
8      private EditText editTextEmail, editTextPassword;
9      private Button signIn,forgotPassword;
10
11     private FirebaseAuth mAuth;
12     private ProgressBar progressBar;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18
19         register =(TextView) findViewById(R.id.register);
20         register.setOnClickListener(this);
21
22         signIn = (Button) findViewById(R.id.login);
23         signIn.setOnClickListener(this);
24
25         editTextEmail = (EditText) findViewById(R.id.email);
26         editTextPassword = (EditText) findViewById(R.id.password);
27
28         progressBar = (ProgressBar) findViewById(R.id.progressBar);
29
30         mAuth = FirebaseAuth.getInstance();
31
32         forgotPassword = (Button) findViewById(R.id.forgotpassword);
33         forgotPassword.setOnClickListener(this);
34
35     }
36
37     @Override
38     public void onClick(View v) {
39         switch (v.getId()){
40             case R.id.register:
41                 startActivity(new Intent( packageContext: this, RegisterUser.class));
42                 break;
43
44             case R.id.login:
45                 userLogin();
46                 break;
47
48             case R.id.forgotpassword:
49                 startActivity(new Intent( packageContext: this, ForgotPassword.class));
50                 break;
51         }
52     }
53 }
```

```

70 }
71
72     private void userLogin() {
73         String email = editTextEmail.getText().toString().trim();
74         String password = editTextPassword.getText().toString().trim();
75
76         if(email.isEmpty()){
77             editTextEmail.setError("Email is Required!");
78             editTextEmail.requestFocus();
79             return;
80         }
81
82         if(!Patterns.EMAIL_ADDRESS.matcher(email).matches()){
83             editTextEmail.setError("Please Enter a Valid Email!");
84             editTextEmail.requestFocus();
85             return;
86         }
87
88         if(password.isEmpty()){
89             editTextPassword.setError("Password is Required!");
90             editTextPassword.requestFocus();
91             return;
92         }
93
94         if(password.length() < 8){
95             editTextPassword.setError("Password must be 8 characters!");
96             editTextPassword.requestFocus();
97             return;
98         }
99
100        progressBar.setVisibility(View.GONE);
101
102        mAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
103            @Override
104            public void onComplete(@NonNull Task<AuthResult> task) {
105
106                if (task.isSuccessful()){
107                    FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
108                    if (user.isEmailVerified()){
109                        //redirect to user profile
110                        startActivity(new Intent(getApplicationContext(), MainActivity.this, ProfileActivity.class));
111                    }
112                    else{
113                        user.sendEmailVerification();
114                        Toast.makeText(MainActivity.this, "Check your email to verify your account!", Toast.LENGTH_LONG).show();
115                    }
116                }
117                else{
118                    Toast.makeText(MainActivity.this, "Failed to Login! Please check your credentials", Toast.LENGTH_LONG).show();
119                }
120            }
121        });
122    }

```

Figure 54: Java Code of Main page

## App Icon Change From default Icon to my Logo

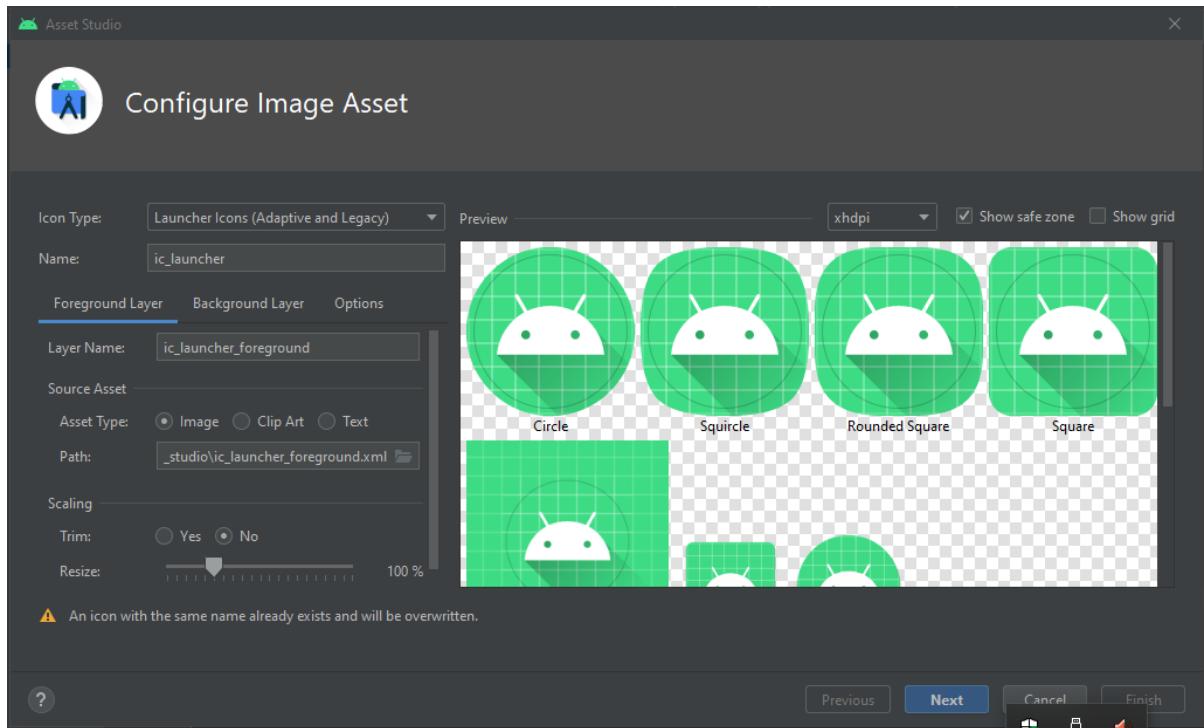


Figure 55: Default App Icon

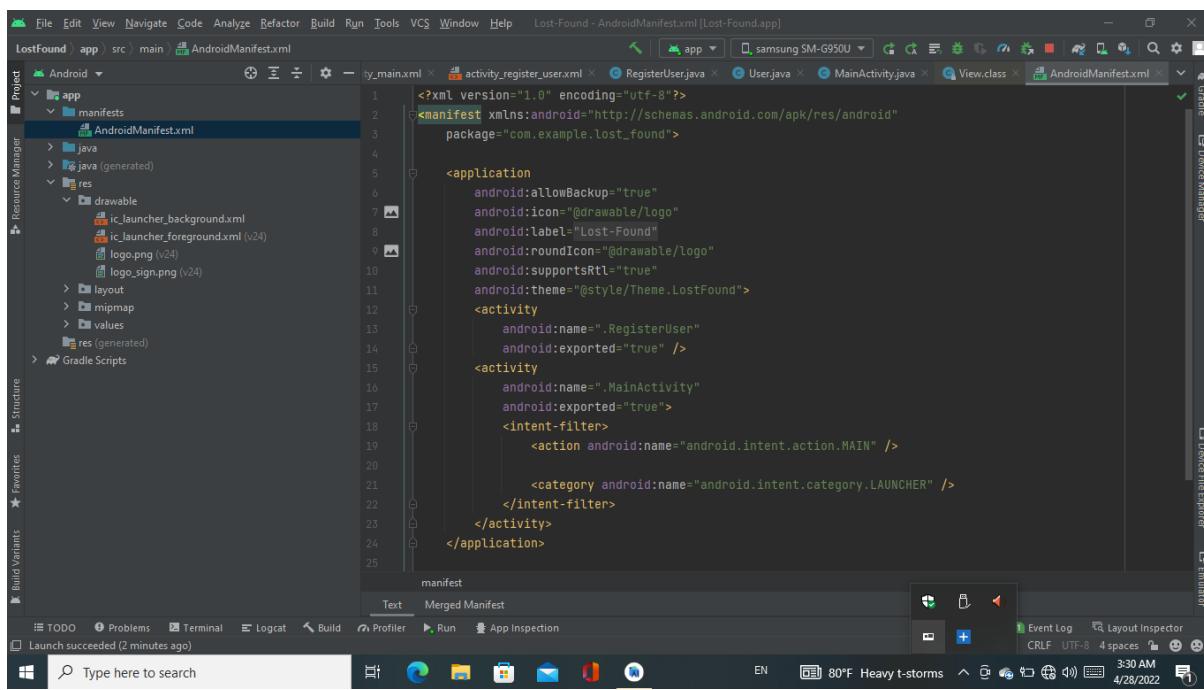


Figure 56: App Icon Change

## Firebase Console

The screenshot shows the Firebase Console homepage. At the top, there's a navigation bar with the Firebase logo, a 'Go to docs' link, and a user profile icon. Below the navigation is a large banner featuring a yellow character drawing and a person at a computer. To the left, there's a button labeled '+ Add project'. In the center, a project card for 'Lost-Found' (lost-found-a90f8) is displayed. Below the banner, there are two main sections: 'Explore a demo project' (with an 'iOS+' icon) and 'Firebase projects are containers for your apps' (with icons for Android, iOS+, and a code editor). A 'Learn more' button is also present.

Figure 57: Firebase Console

The screenshot shows the 'Authentication' section of the Firebase Console under the 'Lost-Found' project. The left sidebar includes 'Project Overview', 'Build' (with 'Authentication' selected), 'Release & Monitor', and 'Spark' (No-cost \$0/month). The main area shows the 'Users' tab of the Authentication interface. It features a search bar, an 'Add user' button, and a table listing user information. The table columns are 'Identifier', 'Providers', 'Created', 'Signed In', and 'User UID'. The data includes:

Identifier	Providers	Created	Signed In	User UID
akmalbinabousali@gmail.com	✉️	May 16, 2022	May 16, 2022	RzQBW4bQTSQOk2ry0cu6afbl6wr1
sajahanrja@gmail.com	✉️	May 16, 2022	May 16, 2022	ouQCRzcOrhPxbnlqzGyHyIXYgzt1
latadathu@gmail.com	✉️	May 16, 2022	May 16, 2022	d255YFCWlyvSbGrN4TXKeipwcJZf1
imranbrono@outlook.com	✉️	May 15, 2022	May 15, 2022	sDDZGZ1WRVg7irXyoX5dPNGeYLi2
sample@gmail.com	✉️	May 12, 2022	May 12, 2022	axNHXL5SRffNEyuqJZ8QYMuTEY...
imranbrono@gmail.com	✉️	May 11, 2022	May 15, 2022	6n1aP4z2WaNQWlxgWyFFkLqcsK...
asharbrono@gmail.com	✉️	May 11, 2022	May 11, 2022	cLgVJ48SSte3Om05NbP6v8JCh1...

Figure 58: Firebase Authentication.

## Firebase RealTime Database

The screenshot shows the Firebase Realtime Database console for the project 'Lost-Found'. The left sidebar includes sections for Authentication, App Check, Firestore Database, Realtime Database (selected), Extensions, Storage, Hosting, Functions, and Machine Learning. The main area displays the Realtime Database interface with tabs for Data, Rules, Backups, and Usage. A specific node under the root is selected, showing a user profile with fields: address ('akp'), age ('23'), email ('sajahanrija@gmail.com'), firstname ('sajahan'), and lastname ('Rija'). The database location is listed as Singapore (asia-southeast1). The browser address bar shows the URL for the database.

Figure 59: Realtime Database

## Firebase used JSON file Codes.

The screenshot shows the Visual Studio Code editor displaying the 'google-services.json' file. The file contains configuration for a Firebase project, including project info, client details for an Android app, OAuth clients, API keys, and services. Key parts of the JSON include:

```
1  "project_info": {  
2     "project_number": "549766130843",  
3     "firebase_url": "https://lost-found-a90f8-default-rtdb.firebaseio.com",  
4     "project_id": "lost-found-a90f8",  
5     "storage_bucket": "lost-found-a90f8.appspot.com"  
6   },  
7   "client": [  
8     {  
9       "client_info": {  
10          "mobilesdk_app_id": "1:549766130843:android:df35a43d633a0b584aea3e",  
11          "android_client_info": {  
12            "package_name": "com.example.lost_found"  
13          }  
14        },  
15        "oauth_client": [  
16          {  
17            "client_id": "549766130843-a4s06v6lana17qdrkltc4misjonmhdcn.apps.googleusercontent.com",  
18            "client_type": 3  
19          }  
20        ],  
21        "api_key": [  
22          {  
23            "current_key": "AIzaSyDpaQ6LBgmF0XpDAYZH0BMN5wcLcOQxAhU"  
24          }  
25        ],  
26        "services": {  
27          "appinvite_service": {  
28            "other_platform_oauth_client": [  
29              {  
30                "client_id": "549766130843-a4s06v6lana17qdrkltc4misjonmhdcn.apps.googleusercontent.com",  
31                "client_type": 3  
32              }  
33            ]  
34          }  
35        }  
36      }  
37    ]  
38  ]  
39 }
```

```
C:\Users\Imran Bruno\Desktop\Lost-Found> google-services.json ...  
10  
11     {  
12         "client_info": {  
13             "mobilesdk_app_id": "1:549766130843:android:df35a43d633a0b584aea3e",  
14             "android_client_info": {  
15                 "package_name": "com.example.lost_found"  
16             }  
17         },  
18         "oauth_client": [  
19             {  
20                 "client_id": "549766130843-a4s06v6lana17qdrkltc4misjonmhdcn.apps.googleusercontent.com",  
21                 "client_type": 3  
22             }  
23         ],  
24         "api_key": [  
25             {  
26                 "current_key": "AIzaSyDpaQ6LBgmF0XpDAyZH0BMW5wcLcOQxAhU"  
27             }  
28         ],  
29         "services": {  
30             "appinvite_service": {  
31                 "other_platform_oauth_client": [  
32                     {  
33                         "client_id": "549766130843-a4s06v6lana17qdrkltc4misjonmhdcn.apps.googleusercontent.com",  
34                         "client_type": 3  
35                     }  
36                 ]  
37             }  
38         },  
39     },  
40     "configuration_version": "1"  
41 }
```

X ⊗ 0 △ 0 Type here to search EN 81°F 5:26 AM 5/16/2022

Figure 60: JSON Code File

### 5.3. DOCUMENT STRUCTURE

In this section, there are two types of Document Structures are represented such as,

- Document Structure of Application Installed Device.

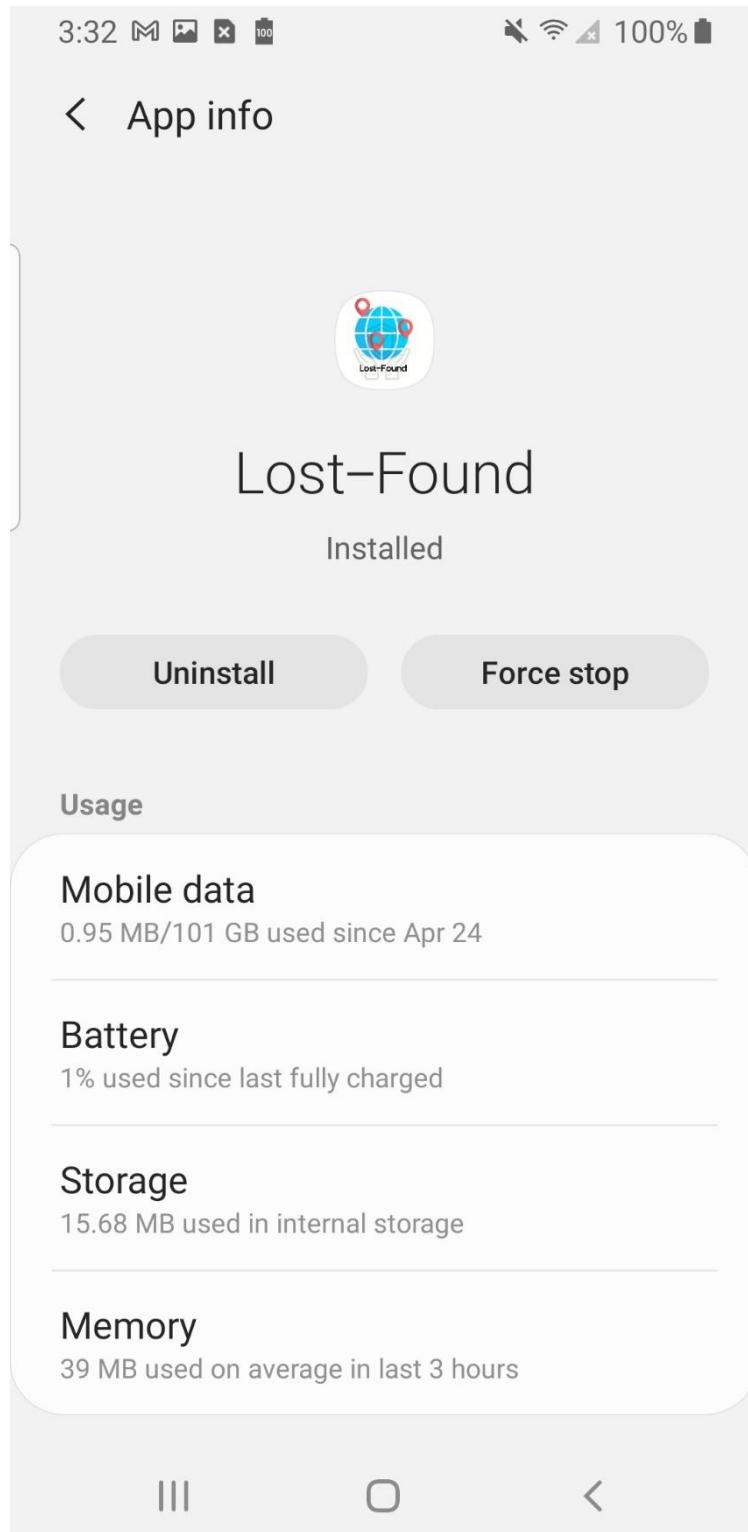


Figure 61: Device Document

- Document Structure of Project Files.

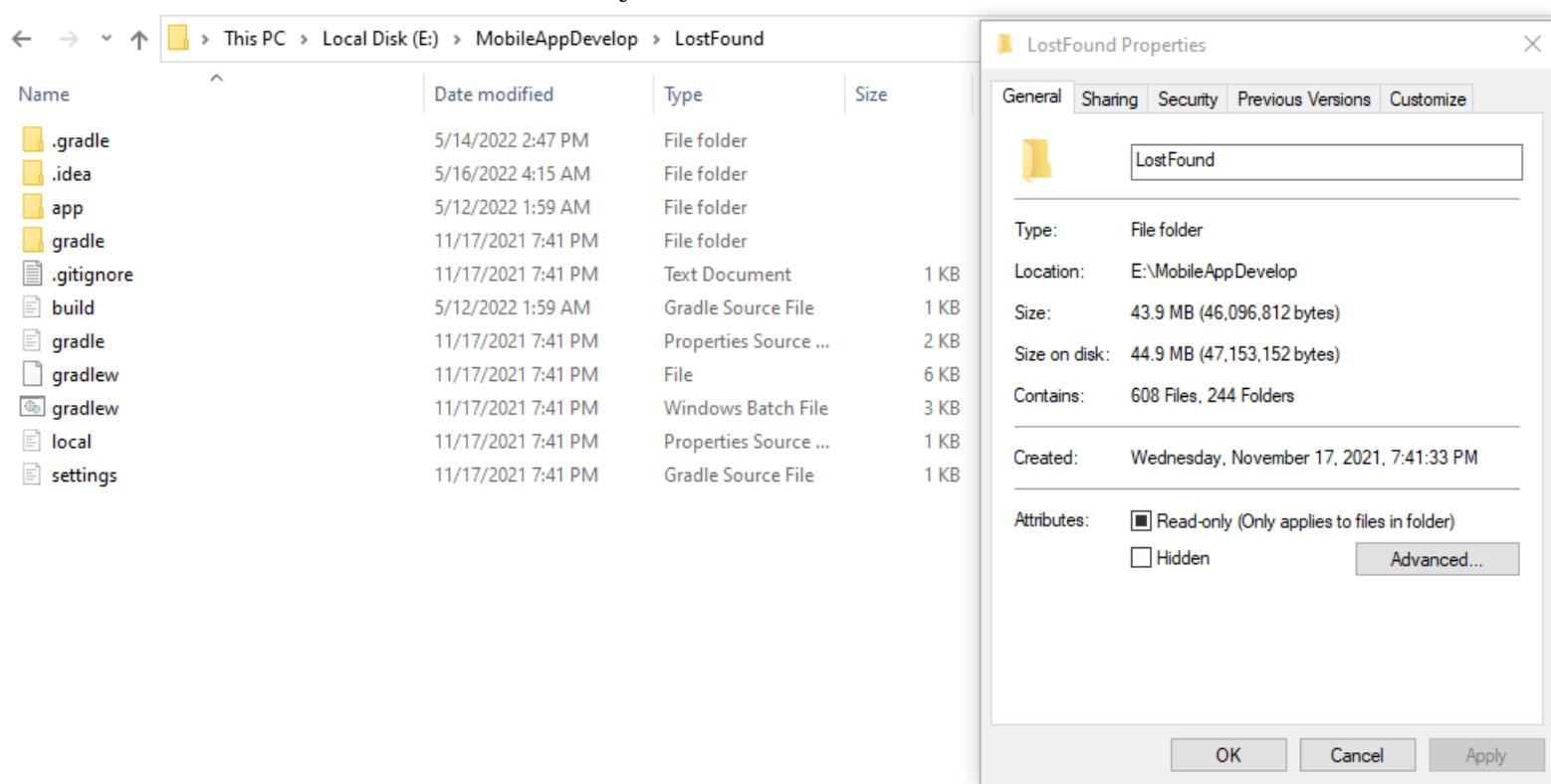


Figure 62: Project File

## **5.4. REQUIREMENTS**

### **5.4.1. SOFTWARE REQUIREMENTS**

The Software requirements of this Lost – Found Mobile Application to provide a better performance:

- Android Operating System above 5.1 capable with Android 12 latest version too.
- Operating System API 27 and above.

### **5.4.2. HAREWARE REQUIREMENTS**

The hardware requirements of this Lost – Found Mobile Application to provide a better performance

- An Android Mobile Device
- Minimum 1GB of ram.
- 4Gb of Storage.

### **5.4.3. OTHER REQUIREMENTS**

- Internet Facility.
- Email Facility

## **5.5. PROBLEM ENCOUNTERED BY IMPLEMENTATION**

The Application is Android Application, the users located in different geographical area. So, the system wants to publish in public domain. The domain register, web hosting and basic networking setting needed. There is no marketing for this application. Need a developer account to publish to the play Store.

## **6.CONCLUSION**

The project entitled Lost – Found Mobile Application was completed successfully. The Application has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The main purpose of developing this Application when all the Lost electronic equipment wants to return to the right person (It's Owners) and this application want be a solution for Oblivion, errors, and Fault.

I have worked more than 160 days for this project. I researched this problem and analysis this problem with different society, people and others, after that I make a great executable plan to this project, I design the entire project and Project Structure than I Start to code here I have faced so many problems such as IDE install, SDK tool establish, AVD manager creation, Insufficient Knowledge and insufficient work experience and so many. I hoped and I faced this problem myself and done this project successfully.

This project gave me the opportunity to get knowledge about many new technologies of Mobile Application Development, Android Development Technics, new programming languages (Java) and programming practices, etc.... I gained lots of knowledge about the programming and software development process and also about modeling methods, and improve my project management skills as well.

## **7. LIMITATIONS**

The Application I have developed some limitation. I am trying to fulfill all requirements. But some tasks are not done here. Real time services like dialling, voice, chatting and video conferencing are not done here. And language translator is not added here. The application is developed for only Losers and Founders. There is no Realtime notifying System. This Application is still not available in play store. Not allowed by google play protect. Only for Android Mobiles, Must be above than android operating system 5.1. email based application.

## **8.REFERENCE**

<https://developer.android.com/>

<https://io.google/2022/program/?q=96a8382e-68d1-40b1-8fe8-c32eafe45327>

<https://developer.android.com/games/guides>

<https://developer.android.com/studio>

[https://developer.android.com/studio?gclid=Cj0KCQjwyYKUBhDJARIAMj9lkFzBf8iC467KtrVZ8G7mAaS8rRkPh7H-zzZeUZZQy6-JeqQKhpAkS0aAq4VEALw\\_wcB&gclsrc=aw.ds](https://developer.android.com/studio?gclid=Cj0KCQjwyYKUBhDJARIAMj9lkFzBf8iC467KtrVZ8G7mAaS8rRkPh7H-zzZeUZZQy6-JeqQKhpAkS0aAq4VEALw_wcB&gclsrc=aw.ds)

<https://www.geeksforgeeks.org/android-tutorial/>

[https://www.jetbrains.com/space/solutions/software-teams/?source=google&medium=cpc&campaign=12683512779&term=%2Bsoftware%20%2Bdevelopment%20%2Btools&gclid=Cj0KCQjwyYKUBhDJARIAMj9lkFoI7Obg72UMCazR9CKVvz2CXTUjrulvnT1ZXMyRyMw6-jxvXDZ2IaAqQGEALw\\_wcB](https://www.jetbrains.com/space/solutions/software-teams/?source=google&medium=cpc&campaign=12683512779&term=%2Bsoftware%20%2Bdevelopment%20%2Btools&gclid=Cj0KCQjwyYKUBhDJARIAMj9lkFoI7Obg72UMCazR9CKVvz2CXTUjrulvnT1ZXMyRyMw6-jxvXDZ2IaAqQGEALw_wcB)

<https://www.jetbrains.com/>

<https://github.com/>

<https://github.com/Imranbrono>

<https://www.sololearn.com/>

<https://stackoverflow.com/>

<https://www.youtube.com/>

N. Bhatt, ""Emergence of Sociology in Europe UNIT 1 EMERGENCE OF SOCIOLOGY IN EUROPE," [Online]. Available:  
[http://www.academia.edu/8128794/Emergence\\_of\\_Sociology\\_in\\_Europe\\_UNIT\\_1\\_EMERGENE](http://www.academia.edu/8128794/Emergence_of_Sociology_in_Europe_UNIT_1_EMERGENE)

..  
M. Kompier, "The psychosocial work environment and health - what do we know and where should we go?," *Scandinavian Journal of Work, Environment & Health*, vol. 28, no. 1, pp. 1-4, 2002.

S. Hasani, "DIFFICULTIES FACING CHALLENGES KOSOVO BUSINESSES FOR INCREASING THEIR COMPETITIVENESS WITH OTHER BUSINESSES IN REGION AND ABROAD," *IJASOS- International E-journal of Advances in Social Sciences*, vol. 1, no. 2, p. 280, 2015.

Sommerville, ", Software Engineering, 9th ed. Addison-Wesley," 2010.

"Systems Analysis And Design Examination Question Papers - 12847," Kenyaplex.com, [Online]. Available: <https://www.kenyaplex.com/questionpapers/12847-systems-analysis-and-design-examination.aspx>.

"SDLC Software Prototype Model," [www.tutorialspoint.com](http://www.tutorialspoint.com/sdlc/sdlc_software_prototyping.htm), [Online]. Available:  
[https://www.tutorialspoint.com/sdlc/sdlc\\_software\\_prototyping.htm](http://www.tutorialspoint.com/sdlc/sdlc_software_prototyping.htm).

"2.127.2 IT Testing Process and Procedures | Internal Revenue Service," Irs.gov, [Online]. Available: [https://www.irs.gov/irm/part2/irm\\_02-127-002](https://www.irs.gov/irm/part2/irm_02-127-002).

"Opensource.com," 11 september 2006. [Online]. Available:  
<https://opensource.com/resources/what-open-source>.

Wiegers, Karl E. (2003). Software Requirements (2nd Ed.). Redmond, WA: Microsoft Press. ISBN 0-7356-1879-8, <http://www.processimpact.com>. 33

Tech-ict.com, 'Direct Implementation' 2015[online] Available: <http://techict.com>> methods.miniweb.

Systems Engineering Fundamentals. Defense Acquisition University Press, 2001.

Imran Ahamed, Software Quality Assurance (3rd Ed.), University of Malaysia, 2014.In Intl. Symp. On Information Visualization, 1995.

Jacobson, I., Christerson M., Jonsson P., Övergaard G., (1992). *Object-Oriented Software Engineering - A Use Case Driven Approach*, Addison-Wesley.

Jaksan M.A, A System development method, 101 Hamilton Terrace, London NW8 99X, England, 1982.

Joggerwalk,"Methodologyofthestudy,"|Geeks,<https://www.geeksunlimited.wordpress.com>, July 26, 2009.

S. Angelov, P. Grefen and D. Greefhorst, a Classification of Software Reference Architectures: Analyzing Their Success and Effectiveness, 2009.

Android™ Notes for Professionals book

<https://books.goalkicker.com/AndroidBook/AndroidNotesForProfessionals.pdf>