

# **SOCIAL MEDIA ANDROID APPLICATION**

BY

RADHE RAMAN TIWARI

ROLL NO. 17010115

Supervised by

Mr. Himangshu Sarma



THESIS

SUBMITTED TO

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, MANIPUR

Bachelor of Technology, Computer Science and Engineering

April, 2019

# **Abstract**

The thesis in my proposed project focuses on development of Android applications to be used on a handheld device running the Android Operating System(OS). It covers the basics such as setting up a development environment, downloading appropriate tools and add-ons. The thesis serves as a guideline to intermediate developers, seeking solutions to problems not discussed in available textbooks.

Android is a platform consisting of an Operating System(OS) and a Software Development Kit(SDK) for handheld devices. It originates from a small software company, acquired by Google and is now owned by Open handset Alliance(OHA), where Google is a member. A description of Android is included and it's advantages and disadvantages are discussed. Resources and recommendations on Android development and Android versioning are presented. Strategies on development are also suggested throughout this thesis.

The development for an Android device and it's data traffic characteristics is of interest, which is also included in this thesis. Connectivity and communication like connecting to Facebook and pulling data from an internet-connected web server, is discussed.

# **Declaration**

I declare that this submission represents my idea in my own words and where others' idea or words have been included, I have adequately cited and referenced the original source. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/sources in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from proper permission has not been taken when needed.

(Signature)

---

(Radhe Raman Tiwari)

Date:

(17010115)



## INDIAN INSTITUTE OF INFORMATION TECHNOLOGY MANIPUR

---

### Certificate

This is to certify that the project report entitled Social Media Android Application, submitted to the Department of Computer Science and Engineering, Indian Institute of Information Technology, Manipur, in partial fulfillment for the award of the degree of **Bachelor of Technology** in Computer Science and Engineering, is a record of bona-fide work carried out by **Mr. RADHE RAMAN TIWARI**, Roll No. 17010115, under my supervision and guidance.

No part of this report has been submitted elsewhere for award of any other degree.

**Mr. Himangshu Sarma**

Supervisor

Date :



## INDIAN INSTITUTE OF INFORMATION TECHNOLOGY MANIPUR

---

### Certificate

This is to certify that the project report entitled Social Media Android Application, submitted to the Department of Computer Science and Engineering, Indian Institute of Information Technology, Manipur, in partial fulfillment for the award of the degree of **Bachelor of Technology** in Computer Science and Engineering, is a record of bonafide work carried out by **Mr. RADHE RAMAN TIWARI**, Roll No.17010115.

No part of this report has been submitted elsewhere for award of any other degree.

Dr. Nongmeikapam Kishorjit Singh  
Professor & Head,  
Department of CSE  
IIIT, Manipur

Date :



## INDIAN INSTITUTE OF INFORMATION TECHNOLOGY MANIPUR

---

### Certificate

This is to certify that the project report entitled Social Media Android Application, submitted to the Department of Computer Science and Engineering, Indian Institute of Information Technology, Manipur, in partial fulfillment for the award of the degree of **Bachelor of Technology** in Computer Science and Engineering, is a record of bonafide work carried out by **Mr. RADHE RAMAN TIWARI**, Roll No. 17010115.

Examiner's Signature

Date :

# **Acknowledgement**

I extend my sincere thanks to the institute **Indian Institute of Information Technology Manipur** who has provided me an opportunity to do the project. I would like to express deep dept to Mr. Himangshu Sarma ,project guide for his vital suggestion ,meticulous guidance and constant motivation which went a long way in successful completion of the project. I cannot move forward without thank to Dr. Navnath Saharia, Dr. Nongmeikapam Kishorjit Singh and Dr. Prerna Mohit who has always been an inspiration to me. I would like to thanks to Dr.Kabita Tharojam, project co-ordinator who has been organizing the project presentation time to time which helped me to finish the project on time. On a moral personal note my deepest appreciation and gratitude to my beloved friends ,who has been an inspiration and have provided me a unrelenting encouragement and support

**RADHE RAMAN TIWARI**  
17010115

# Contents

|   |           |
|---|-----------|
| <b>Abstract</b>                                       | <b>2</b>  |
| <b>Declaration</b>                                    | <b>3</b>  |
| <b>Supervisor Certificate</b>                         | <b>4</b>  |
| <b>HOD Certificate</b>                                | <b>5</b>  |
| <b>Examiner Certificate</b>                           | <b>6</b>  |
| <b>Acknowledgement</b>                                | <b>7</b>  |
| <b>List of abbreviations</b>                          | <b>11</b> |
| <b>1 Introduction</b>                                 | <b>13</b> |
| 1.1 Outline of the report . . . . .                   | 13        |
| 1.2 Gantt chart . . . . .                             | 14        |
| <b>2 Existing System Study</b>                        | <b>15</b> |
| 2.1 A brief Introduction . . . . .                    | 16        |
| 2.2 Study methods . . . . .                           | 17        |
| 2.3 What is Android Operating System . . . . .        | 17        |
| 2.4 Software Requirements . . . . .                   | 17        |
| <b>3 System Analysis, Design &amp; Implementation</b> | <b>19</b> |
| 3.1 Introduction . . . . .                            | 20        |
| 3.2 System Analysis . . . . .                         | 20        |
| 3.2.1 System Objective . . . . .                      | 20        |
| 3.2.2 Relation to External Environment . . . . .      | 20        |
| 3.2.3 Design Considerations . . . . .                 | 20        |
| 3.2.4 System Architecture . . . . .                   | 21        |
| 3.2.5 Operational Concepts and Scenarios . . . . .    | 22        |

|                   |   |           |
|-------------------|---|-----------|
| 3.3               | Design . . . . .  | 23        |
| 3.3.1             | Flow Chart . . . . .  | 23        |
| 3.3.2             | Data Flow Diagram . . . . .                                 | 24        |
| 3.3.3             | UML Class Diagram . . . . .                                 | 27        |
| 3.3.4             | Database Design . . . . .                                   | 28        |
| 3.4               | Implementation . . . . .                                    | 30        |
| 3.4.1             | Functional Requirements . . . . .                           | 30        |
| 3.4.2             | Non Functional Requirements . . . . .                       | 31        |
| <b>4</b>          | <b>Result Analysis and Testing</b>                          | <b>32</b> |
| 4.1               | Introduction . . . . .                                      | 33        |
| 4.2               | Source Code . . . . .                                       | 33        |
| 4.2.1             | Screenshot . . . . .  | 33        |
| 4.3               | Result Analysis . . . . .                                   | 33        |
| 4.3.1             | Application Objectives :- . . . . .                         | 33        |
| 4.4               | Test Cases . . . . .  | 35        |
| 4.4.1             | Test for Home Page Activity . . . . .                       | 36        |
| 4.4.2             | Test for Profile Page Activity . . . . .                    | 37        |
| 4.4.3             | Test for Blog Post Page Activity . . . . .                  | 38        |
| <b>5</b>          | <b>Conclusion</b>   | <b>40</b> |
| 5.1               | Future direction and Scope . . . . .                        | 41        |
| <b>Appendix A</b> | <b>Screenshot and Description of the Implemented System</b> | <b>42</b> |
| A.1               | Screenshot . . . . .  | 42        |
| <b>Appendix B</b> | <b>User manual</b>  | <b>47</b> |
| B.1               | System Specification . . . . .                              | 47        |
| B.1.1             | Hardware requirement : . . . . .                            | 47        |
| B.1.2             | Software requirement : . . . . .                            | 47        |
| B.2               | Installation . . . . .                                      | 48        |

# List of Figures

|      |   |    |
|------|---|----|
| 1.1  | Gantt Chat . . . . .  | 14 |
| 3.1  | Flow Chart . . . . .  | 21 |
| 3.2  | Flow Diagram . . . . .  | 23 |
| 3.3  | User Case Table . . . . .   | 24 |
| 3.4  | Use Case Diagram of Authentication System . . . . .                 | 25 |
| 3.5  | Use Case Diagram of Contacts Form . . . . .                         | 25 |
| 3.6  | Use Case Diagram of Chat Form . . . . .                             | 26 |
| 3.7  | Use Case Diagram of Maintenance . . . . .                           | 26 |
| 3.8  | UML Class diagram . . . . .   | 27 |
| 3.9  | Firebase users database text messages list . . . . .                | 28 |
| 3.10 | Firebase users authentication list . . . . .                        | 29 |
| 3.11 | Firebase storage folders list . . . . .                             | 30 |
| 4.1  | Home page activity code which contains all home fragments . . . . . | 33 |
| 4.2  | Login activity code where user will login . . . . .                 | 34 |
| 4.3  | Register activity code where user will register . . . . .           | 34 |
| 4.4  | Script to convert firebase timestamp to real time . . . . .         | 35 |
| 4.5  | Home page activity contains four fragments. . . . .                 | 37 |
| 4.6  | User profile page activity. . . . .                                 | 38 |
| 4.7  | Blog post page activity. . . . .                                    | 39 |
| 4.8  | New Post activity. . . . .  | 39 |
| A.1  | Start Page Activity. . . . .  | 43 |
| A.2  | User Register Activity. . . . .                                     | 43 |
| A.3  | User Login Activity. . . . .  | 44 |
| A.4  | Popular News Page Activity. . . . .                                 | 44 |
| A.5  | Specific News Activity. . . . .                                     | 45 |
| A.6  | Other User Profile Activity. . . . .                                | 45 |
| A.7  | Individual Chat Activity. . . . .                                   | 46 |
| A.8  | Find Friends Activity. . . . .                                      | 46 |

# List of abbreviations

|      |  |
|------|--|
| 3G   | International Mobile Telecommunications-2000(IMT-2000) standards. The third generation standards family for Mobile Communication           |
| ADT  | Android Development Tool. A plug-in for Android Studio IDE   |
| API  | Application Programming Interface  |
| AVD  | Android Virtual Device. To be able to compile your solutions and run them as application on an emulator you need to setup at least one AVD |
| CPU  | Central Processing Unit  |
| CRM  | Customer Relationship Management system  |
| DBMS | Database Management System   |
| CSS  | Cascading Style Sheets   |
| IPA  | International Phonetics Alphabets  |

|      |   |
|------|---|
| DDMS | Dalvik Debug Monitor Server. Connects your IDE to your AVD emulator. Every Android application has its own Virtual Machine, not interfering with each other |
| HTML | Hyper Text Markup Language  |
| IDE  | Integrated Development Environment. In this thesis - Android Studio   |
| J2SE | Java Platform, Standard Edition   |
| LBS  | Location-Based Service. Often used with a handheld devices Global Position  |
| MiB  | Mebibyte.1 MiB is $2^{20}$ bytes equivalent to 1,048,576 bytes  |
| OHA  | Open Handset Alliance   |
| OS   | Operating System  |
| SD   | Secure Digital. A Memory Card used in handheld devices to increase storage  |
| IDE  | Software Development Kit. A software framework to use when developing application   |
| XML  | eXtended Mark-up Language   |

# **Chapter 1**

## **Introduction**

Teleconferencing or Chatting, is a method of using technology to bring people and ideas “together” despite of the geographical barriers. The technology has been available for years but the acceptance it was quite recent. Our project is an example of a chat server. It is made up of two applications the client application, which runs on the user's PC and server application, which runs on any PC on the network. To start chatting client should get connected to server where they can practice two kinds of chatting, public one (message is broadcasted to all connected users) and private one (between any two users only) and during the last one security measures were taken and as well as best recent popular news and their descriptions.

### **1.1 Outline of the report**

This report is organised around four main parts....

**Chapter 1** Introduction to the project

**Chapter 2** Describes the existing system analysis.

**Chapter 3** Describes the analysis, design, Implementation

**Chapter 4** Implementation, Result Analysis and Testing

**Chapter 5** Conclusion & Future Scope

## 1.2 Gantt chart

For the completion of any work we need to plan and divide the work in some time frame.I divided my work according to the following Gantt Chart.



Figure 1.1: Gantt Chat

The above chart describes the time frame of simultaneous phases of the project. The first phases is the analysis phase, in which I completed the analysis of the existing Social Media Apps like WhatsApp,Messanger,Instagram. The second phase is the design phase, in which I completed the database design, interface design of the application. The third phase is the implementation phase, in which I completed the implmentation part by writing the code in JavaScript, XML(eXtensible Markup Language) and JAVA Micro Edition(ME).

# Chapter 2

## Existing System Study

**Outline:** This chapter presents the following:

1. A brief Introduction
2. Study methods
3. What is Android Operating System
4. Software Requirements

## 2.1 A brief Introduction

continued to get smaller in size, using less power and performing more advanced calculations. In 2007 Apple released their iPhone to achieve the next goal in computing. This new type of communication tool, called Smartphone, is generally referred to as a phone, which is a poor labelling. A Smartphone is a handheld computer, which can place phone calls. Although the term Smartphone was first used in 1992, Apple was the first company to release a Smartphone to a wider audience. This evolution is led by computer manufacturers and software companies and not handset manufacturers, which have controlled the market thus far.

One competitor to Apple iPhone OS is the Android OS. Android originates from a small software company, acquired by Google and is now owned by Open Handset Alliance(OHA), where Google is a member. OHA has over a hundred member companies such as mobile operators, semiconductor companies, handset manufacturers, software companies and commercialisation companies. Driven through the Apache License, anyone can use the Android Software Development Kit(SDK) to develop applications to run on the Android OS. Especially interesting for Android is it's use of common non-proprietary techniques such as the Java programming language in combination with Extensible Markup Language(XML). This makes it open, simple and easy to use for a substantial part of the developer community.

The thesis reflects experiences of using XML, the Java Programming Language and Android SDK. Questions answered are the following:

- What is Android and how does it work using Linux Kernel, XML and Java Programming Language?
- How does a developer install an Android application on an Android device, without using the Android market?
- What does a developer do to connect the Android application to Facebook Connect?
- How does a developer connect, and synchronize data, with an internet-connected server?

This thesis focuses on development of Android applications to be used on a handheld device running on the Android OS. The thesis covers the basics such as setting up a development environment, downloading appropriate tools and add-ons and shows how to get developers working with Android application development. In essence this thesis serves as a guideline to intermediate developers, seeking solutions to practical problems, who might find current literature a bit too shallow on giving answers to these questions, such as “How about the warning messages I get when compiling Android applications” and “The AVD does not start when I use API level less than 16”.

## **2.2 Study methods**

This is an empirical qualitative study, based on reading above mentioned literature and testing their examples. Tests are made by programming according to books and online resources, with the explicit goal to find best practices and a more advanced understanding of Android. One use case is presented in this thesis as a “Hello World!” application, explaining what happens behind the scenes. The other use case is a developed application which is presented at a conceptual level.

## **2.3 What is Android Operating System**

“Android was built from the ground up with the explicit goal to be the first open, complete, and free platform created specifically for mobile devices.”

- Ableson F. et al, *Unlocking Android*, page 4.

Android is an open system, and is free to use by anyone. A handset manufacturer can use Android if they follow the agreement stated in the Software Development Kit. There are no restrictions or( requirement for the handset manufacturer to share their extensions with anyone else, as there are in other open source software, if they leave the Linux kernel as is. The Linux kernel is under a different and more restricted license than Android.

Android is a software environment and not a hardware platform, which includes an OS, built on Linux kernel-based OS hosting the Dalvik virtual machine. The Dalvik virtual machine runs Android applications as instances of the virtual machine. Android contains a rich user interface, application framework, Java Class libraries and multimedia support. Android also comes with built-in applications containing features such as short message service functionality messaging, phone capabilities and an address book(contacts).

## **2.4 Software Requirements**

This Project development requires the knowledge of following Tools (Open-Source) and Artificial Programming & Scripting Languages :—

1. Android Studio IDE
2. Text-Editor
3. Virtual Device (i.e emulator)
4. Firebase Server Side Database

- Authentication
  - Real Time Database
  - Cloud Storage
  - Storage
  - Cloud Functions
5. SQLite Client Side Database
  6. Node JS Environment
  7. JAVA ME, XML, JavaScript, Node JS for Firebase Functions & Others languages.
  8. Other basic requirements.

# **Chapter 3**

## **System Analysis, Design & Implementation**

**Outline:** This chapter presents the following:

1. Introduction
2. System Analysis
3. Design
4. Implementation

## **3.1 Introduction**

In my proposed project **Social Media Android Application**, User can create account, post images and writes their description, sent messages, check status of his or his friends, see popular news and share those popular news.

## **3.2 System Analysis**

System analysis consists following parts

### **3.2.1 System Objective**

Communication over a network is one field where this tool finds wide ranging application. Chat application establishes a connection between 2 or more systems connected over an intranet or ad-hoc. This tool can be used for large scale communication and conferencing in an organization or campus of vast size, thus increasing the standard of co-operation. In addition it converts the complex concept of sockets to a user friendly environment. This software can have further potentials, such as file transfer, video calling and voice chatting options that can be worked upon later.

### **3.2.2 Relation to External Environment**

This tool helps in two major aspects -

- Resolving the names of all the system connected in a network and enlisting them.
- Used for communication between multiple systems enlisted in the resolved list.

### **3.2.3 Design Considerations**

**Approach :**

The tool has been designed using XML & Android in-build interface.

**Methodology :**

The user interacts with the tool using a GUI

- The GUI operates in two forms, the List form & the chat form.
- The List form contains the names of all the systems connected to a network.

- The chat form makes the actual communication possible in the form of text and images

### 3.2.4 System Architecture

The chat application works in two forms..

**List Form :** In this form, all the names of the systems connected to a network are enlisted. These names can later be used for communication with the help of touch on display, or in simple language: texts or images.

**Chat form :** This form is called only when an element is selected from the List form. In this form, a connection is created between the host system and the selected system with the help of a socket.

**Flow Chart :**

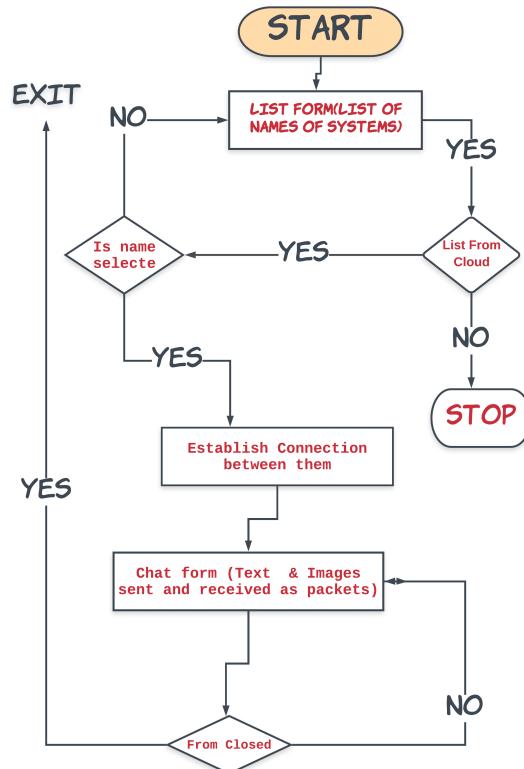


Figure 3.1: Flow Chart

### **3.2.5 Operational Concepts and Scenarios**

Operation of the application based on the inputs given by the user:

#### **List Form :**

- When initialized, returns a list containing the names of all the system connected in a network.
- Contains two sections Refresh and Connect
- When Refresh section is auto refreshes the list of names.
- When the Connect section is worked properly or a name is clicked, the chat form is initialized with a connection between the host and the client machine.
- Note: If no name is selected, and connect section is clicked an error box is displayed.

#### **Chat form :**

- Contains a rich text-view which cannot be edited but only displays the messages from one user to another, including the self sent message, as in any chat application.
- Contains a text-view for messages to be written that is sent across the network.
- Contains messages and images Send buttons.
- When the sent button is clicked, in the background, the text in the text-view is encoded and sent as a packet over the network to the client machine. Here this message is decoded and is shown in the rich text-view.
- To make it more realistic, the self sent message is shown in the rich text-view as well. Both the messages are differentiated with the help of the identifier name at the beginning of each message in the rich text-view

**Exit :** The user exits the software in two scenarios:

- Exits the chat form, the list form remains intact.
- Exits the list form, this is when the application is closed or user has been logout.

### 3.3 Design

My design phase includes :

1. Flow Chart
2. Data Flow Diagram
3. Unified Modeling Language(UML)
4. Database Design
5. Implementation

#### 3.3.1 Flow Chart

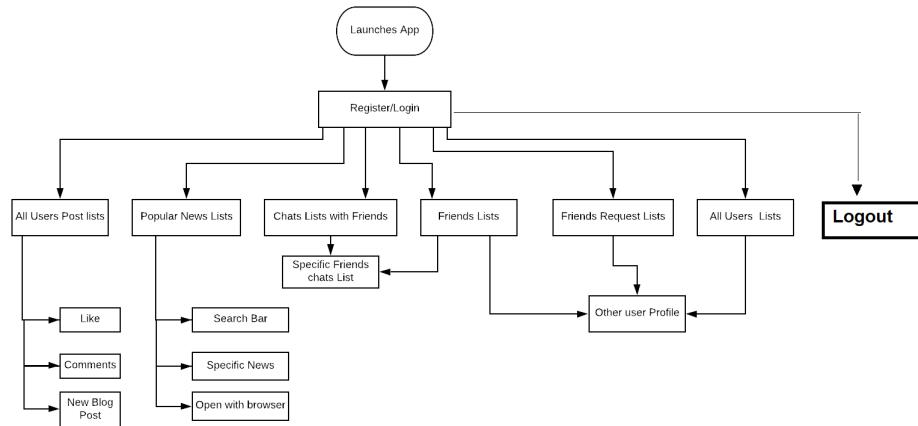


Figure 3.2: Flow Diagram

The above diagram is the Flow diagram of this thesis. Here are the steps explained :-

1. In the Above Diagram First of the user will enter its details then it will check for whether the provided details are correct or not if the user details are found correct then user will enter his or her home page activity and do the his or her required operations.
2. As I have mentioned some of the features here which are of the user as he or she can search news, send friend request, accept friend request, chat with his friends, see his friends profile, all users status, find user from all users list, see blog post, like on post, comments on posts or add new blog posts.

### **3.3.2 Data Flow Diagram**

#### **User Case Table :**

The below diagram is the User Case Table of this thesis. Here are the steps explained :-

1. In the below diagram level 1 explains the feature of my proposed application.
2. In the below diagram level 2 explains what does each sections of level 1.
3. In the below diagram level 3 explains who will manage each sections.
4. In the below diagram Admin means Database administrator

| Level 0  | Level1                   | Level 2   | Adiminstator   |
|--|--------------------------|---|----------------|
| Social<br>Media<br>Android<br>Application<br>& Popular<br>News | Authentication<br>System | Register, Login,Logout  | User and Admin |
|  | Contacts Form            | Friends List, Send Friend Request,Add Friend, Remove Friend             | User           |
|  | Chats Form               | Send Messages & images,view them,send blog posts,comments on posts,like | User           |
|  | Maintenace               | User's Profile  | Admin          |

Figure 3.3: User Case Table

#### **Authentication System :**

The below diagram is the authentication diagram of this thesis. Here are the steps explained :-

1. In my proposed application user can register with the help of unique valid email-id, Name and their secure password.

2. In this user can login with the help of unique email and their password.
3. In this user can also logout.

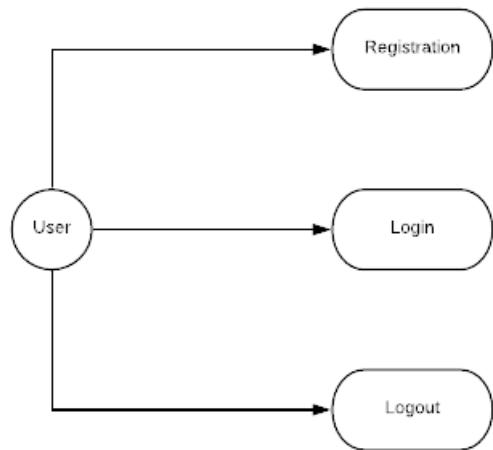


Figure 3.4: Use Case Diagram of Authentication System

### **Contacts Form :**

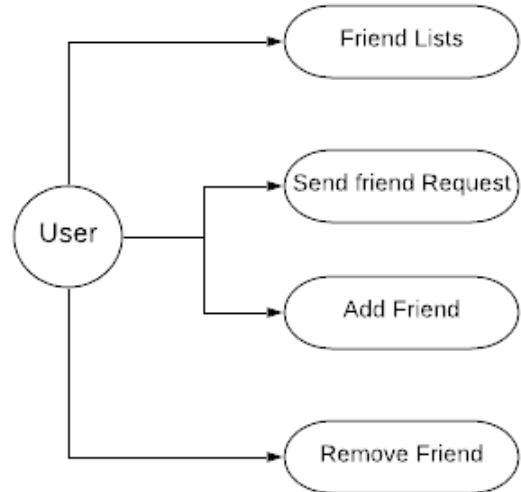


Figure 3.5: Use Case Diagram of Contacts Form

**Chat Form :**

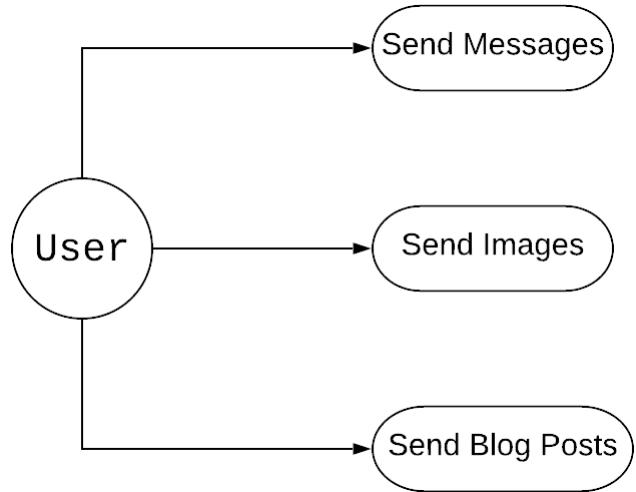


Figure 3.6: Use Case Diagram of Chat Form

**Maintenance :**

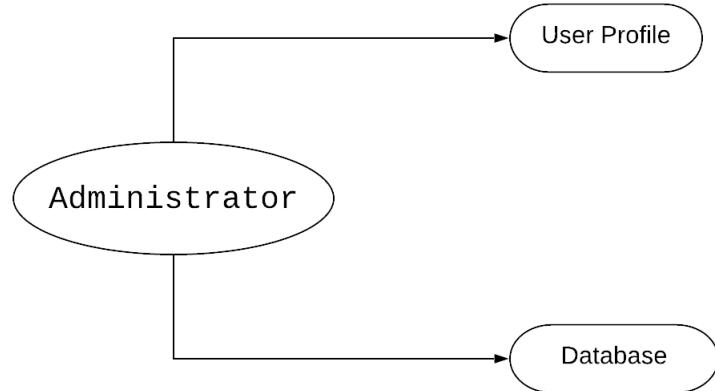


Figure 3.7: Use Case Diagram of Maintenance

### 3.3.3 UML Class Diagram

Here is a Unified Modeling Language(UML) class diagram for this thesis which consists four classes user class, Account class, Blog\_Post class and Individual\_chat class and shows composition of user class to their other classes

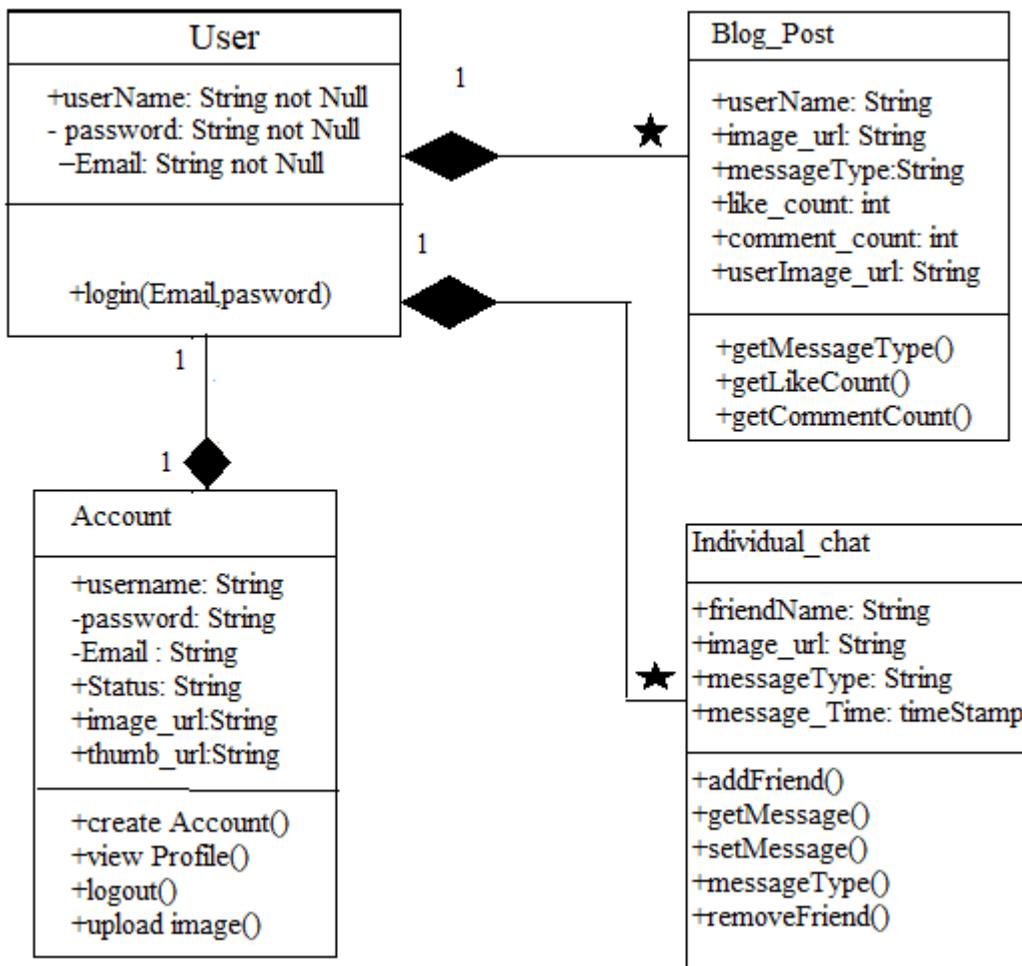


Figure 3.8: UML Class diagram

### 3.3.4 Database Design

Here is the screenshot of my existing database...

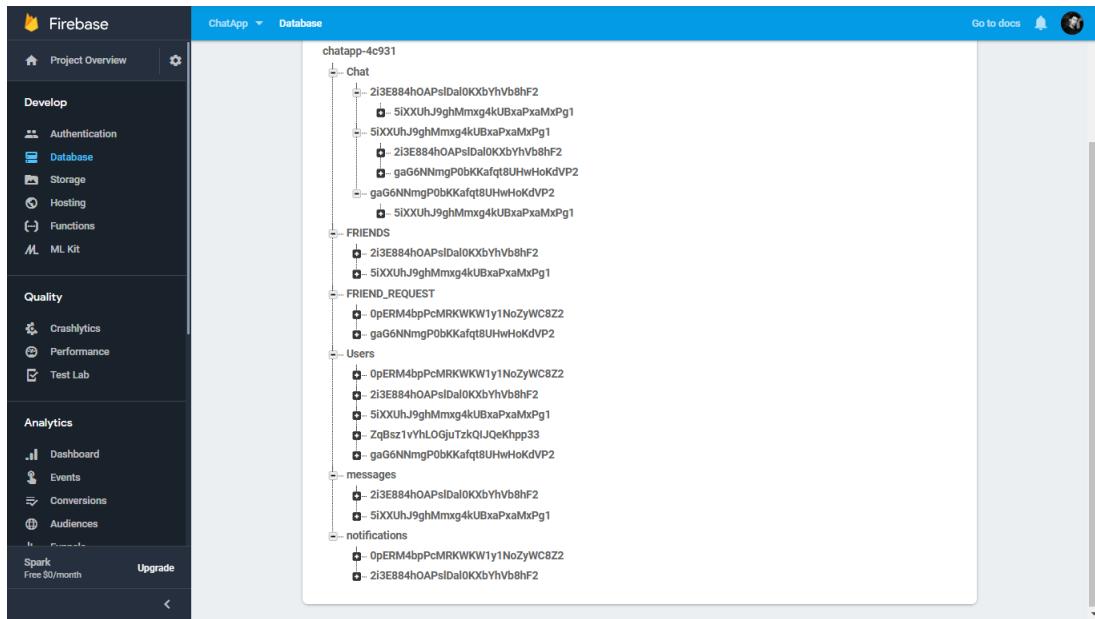


Figure 3.9: Firebase users database text messages list

The above diagram is the real time database storage diagram which contains following informations...

1. Users : Users Details like device token, users name, profile image URL, thumb-image URL, users status and his online status .
2. FRIENDS : It contains users friends list
3. FRIENDS\_REQUEST : It contains users friend request type sent or received.
4. message : It contains users messages list with his friends.

5. notifications : When user send friend request it saves informations about friend request.
6. Chat : It contains online or offline status of users.

The screenshot shows the Firebase console's Authentication section. On the left, there's a sidebar with 'Project Overview' and sections for 'Develop' (Authentication, Database, Storage, Hosting, Functions, ML Kit), 'Quality' (Crashlytics, Performance, Test Lab), and 'Analytics'. Below that, it says 'Spark Free \$0/month' and has an 'Upgrade' button. The main area is titled 'Authentication' and shows a table of users. The columns are 'Identifier', 'Providers', 'Created', 'Signed In', and 'User UID'. The data includes:

| Identifier       | Providers | Created     | Signed In   | User UID                      |
|------------------|-----------|-------------|-------------|-------------------------------|
| aregon@gmail.com | ✉️        | 11 Mar 2019 | 11 Mar 2019 | 0pERM4bpPcMRKWKW1y1NoZyW...   |
| elisa@gmail.com  | ✉️        | 11 Mar 2019 | 10 Apr 2019 | 2iSE884h0APsiDaI0KXbYhVb8hf2  |
| radhe@gmail.com  | ✉️        | 9 Apr 2019  | 10 Apr 2019 | 5iXXUhJ9ghMmxg4kUBxaPxamxP... |
| spyder@gmail.com | ✉️        | 11 Mar 2019 | 11 Mar 2019 | ZqBs21vYhLOGjuTzkQJQeKhpp33   |
| jacks@gmail.com  | ✉️        | 11 Mar 2019 | 16 Apr 2019 | gaG6NNmgP0bKKafqt8UHwHoKd...  |

At the bottom, there are buttons for 'Rows per page: 50' and '1-5 of 5'.

Figure 3.10: Firebase users authentication list

The above diagram is authentication diagram which contains following informations

1. Users valid email address.
2. Users password which should be greater than 5 characters.

The below diagram is storage diagram which contains following informations

1. message\_Images folder contains all chats related images .
2. Profile\_Images contains thumb-images of all users.
3. Picture\_images contains images of all users.
4. post\_images contains blog post images.

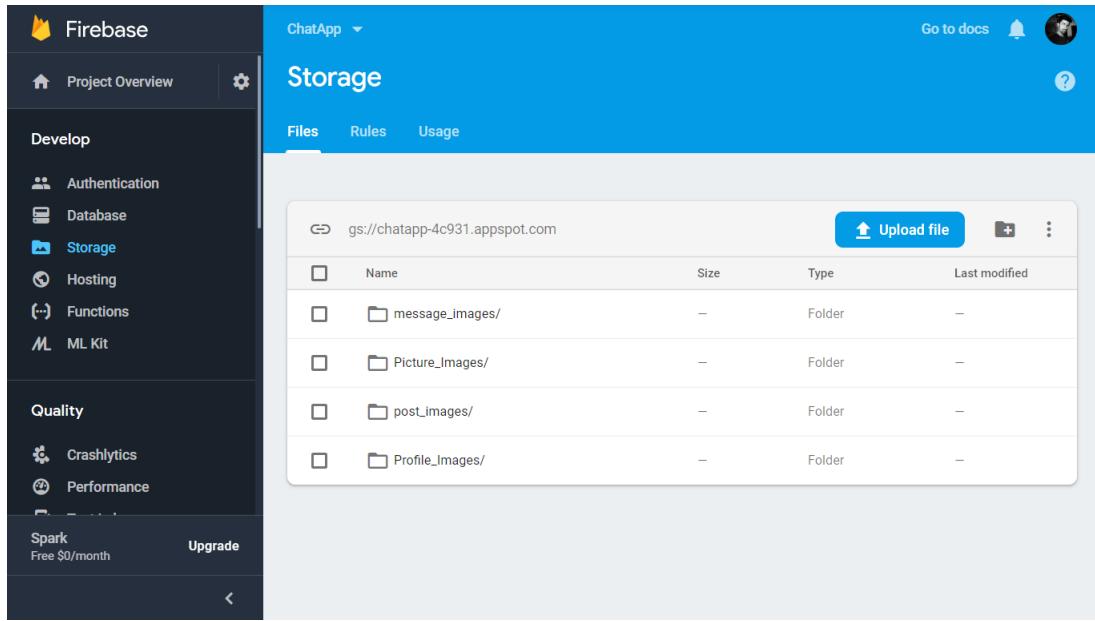


Figure 3.11: Firebase storage folders list

## 3.4 Implementation

### 3.4.1 Functional Requirements

#### 1. User Registration :

User must be able to register for the application through a valid Email Address. and password. On installing the application, user must be prompted to register with email, password and their name. If user skips this step, user will not be able to use this App. The users email will be the unique identifier of his/her account on this application.

#### 2. Send Message :

User should be able to send instant message to any his/her friends and user will get notify when his friends will online by one green color image.

#### 3. Broadcast Message :

User should be able to post images and their description. User should be able to broadcast post to all users.

#### 4. Message Status :

User must be able to know on when the message has come and how much old it is.

### **3.4.2 Non Functional Requirements**

- 1. Privacy :**

Messages shared between users should be encrypted to maintain privacy

- 2. Robustness :**

In case users App crashes, a backup of their chat history must be stored on remote database servers to enable recoverability.

- 3. Performance :**

Application must be lightweight and must send messages instantly.

# Chapter 4

## Result Analysis and Testing

**Outline:** This chapter presents the following:

1. Introduction
2. Source Code
3. Result Analysis
4. Test Cases

## 4.1 Introduction

After study of existing system, software requirements, system analysis and Implementation let us have a look optimum source code acceptable results analysis and their possible test cases.

## 4.2 Source Code

### 4.2.1 Screenshot



The screenshot shows the Java code for the `MainActivity` class. The code is annotated with several `Override` and `protected` keywords, indicating it's part of a larger framework. It includes imports for `AppCompatActivity`, `ViewPagerAdapter`, `AdapterSections`, `DatabaseReference`, and `FirebaseAuth`. The code handles the creation of the activity, retrieves user data from Firebase Database, and initializes UI components like the toolbar and fragments. It also includes logic for handling user authentication changes.

```
public class MainActivity extends AppCompatActivity {
    private FirebaseAuth mAuth;
    private FirebaseAnalytics mFirebaseAnalytics;
    private ViewPagerAdapter viewPagerAdapter;
    private AdapterSections section;
    private SectionsList sectionsList;
    private TableLayout tableLayout;
    private DatabaseReference mUserRef;

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

        mAuth = FirebaseAuth.getInstance();

        mainToolbar = (Toolbar) findViewById(R.id.main_page_toolbar);
        setSupportActionBar(mainToolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        if (mAuth.getCurrentUser() != null) {
            mUserRef = FirebaseDatabase.getInstance().getReference().child("Users").child(mAuth.getCurrentUser().getUid());
            viewPagerAdapter = new ViewPagerAdapter(findViewById(R.id.main_tabsViewPager));
            section = new AdapterSections();
            section.setSectionListener();
            viewPagerAdapter.setAdapterSections(section);
            tableLayout = (TableLayout) findViewById(R.id.main_table);
            tableLayout.setAdapterWithViewPager(viewPagerAdapter);
        }
    }

    @Override
    public void onStart() {
        super.onStart();
        // Check if user is signed in (non-null) and update UI accordingly.
    }
}
```

Figure 4.1: Home page activity code which contains all home fragments

## 4.3 Result Analysis

From a proper analysis of positive points and constraints on the component, it can be safely concluded that the product is a highly efficient GUI based component. This application is working properly and meeting to all user requirements. This component can be easily plugged in many other systems.

### 4.3.1 Application Objectives :-

#### 1. Improvement in control and performance :

The system is developed to cope up with the current issues and problems with all needed thing are situated to different-different places.

```

        setSupportActionBar(loginToolbar);
        getSupportActionBar().setTitle("Login Account");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        loginbtn.setOnClickListener(v -> {
            userMail=loginEmail.getText().toString();
            userPass=loginPass.getText().toString();
            if(!TextUtils.isEmpty(userMail) && !TextUtils.isEmpty(userPass)) {
                loginPro.setTitle("Login You");
                loginPro.setMessage("Wait for While");
                loginPro.setCanceledOnTouchOutside(false);
                loginPro.show();

                userLogin(userMail,userPass);
            }
        });
    }

    private void userLogin(String email,String password){
        mAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener((task) -> {
            if (task.isSuccessful()) {
                loginPro.dismiss();

                String current_user=mAuth.getCurrentUser().getUid();
                String token=FirebaseInstanceId.getInstance().getToken();
                mDatabase.child(current_user).child("device_token").setValue(token).addOnSuccessListener((OnSuccessListener) (eVoid) -> {
                    Intent MainIntent = new Intent(getApplicationContext(), LoginActivity.this, MainActivity.class);
                    MainIntent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK | Intent.FLAG_ACTIVITY_NEW_TASK);
                    startActivity(MainIntent);
                    finish();
                    Toast.makeText(context: LoginActivity.this, text: "Login Now", Toast.LENGTH_LONG).show();
                });
            } else {
                loginPro.hide();

                String errorMessage = task.getException().getMessage();
                Toast.makeText(context: LoginActivity.this, text: "Error : " + errorMessage, Toast.LENGTH_LONG).show();
            }
        });
    }
}

```

Figure 4.2: Login activity code where user will login

```

        regbtn.setOnClickListener(v -> {
            userName=rename.getText().toString();
            userEmail=reemail.getText().toString();
            userPass=repass.getText().toString();

            if(!TextUtils.isEmpty(userName) && !TextUtils.isEmpty(userEmail) && !TextUtils.isEmpty(userPass)) {
                regPro.setTitle("Registering You");
                regPro.setMessage("Wait for While");
                regPro.setCanceledOnTouchOutside(false);
                regPro.show();

                register_user(userName, userEmail, userPass);
            } else {
                Toast.makeText(context: RegisterActivity.this, text: "Check Input Fields",Toast.LENGTH_LONG).show();
            }
        });
    }

    private void register_user(final String Name, String email, String password){
        mAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener((task) -> {
            if(task.isSuccessful()){
                FirebaseUser user=FirebaseAuth.getInstance().getCurrentUser();
                String url=user.getUid();
                String token=FirebaseInstanceId.getInstance().getToken();

                myRef = FirebaseDatabase.getInstance().getReference().child("Users").child(url);

                Map< String, String> mapUser=new HashMap<>();
                mapUser.put("name",Name);
                mapUser.put("status","default");
                mapUser.put("image","default");
                mapUser.put("thumbImage","default");
                mapUser.put("device_token",token);
                myRef.setValue(mapUser).addOnCompleteListener((task) -> {

```

Figure 4.3: Register activity code where user will register

## 2. Save cost :

After computerized application is implemented less human force will be required to maintain user information like messages, blog post and most important popular

```

import ...

public class GetTimeAgo extends Application{

    private static final int SECOND_MILLIS = 1000;
    private static final int MINUTE_MILLIS = 60 * SECOND_MILLIS;
    private static final int HOUR_MILLIS = 60 * MINUTE_MILLIS;
    private static final int DAY_MILLIS = 24 * HOUR_MILLIS;

    public static String getTimeAgo(long time, Context ctx) {
        if (time < 100000000000L) {
            time+= 1000;
        }

        long now = System.currentTimeMillis();
        if (time > now || time < 0) {
            return null;
        }

        final long diff = now - time;
        if (diff < MINUTE_MILLIS) {
            return "just now";
        } else if (diff < 2 * MINUTE_MILLIS) {
            return "a minute ago";
        } else if (diff < 50 * MINUTE_MILLIS) {
            return diff / MINUTE_MILLIS + " minutes ago";
        } else if (diff < 90 * MINUTE_MILLIS) {
            return "an hour ago";
        } else if (diff < 24 * HOUR_MILLIS) {
            return diff / HOUR_MILLIS + " hours ago";
        } else if (diff < 48 * HOUR_MILLIS) {
            return "yesterday";
        } else {
            return diff / DAY_MILLIS + " days ago";
        }
    }
}

```

Figure 4.4: Script to convert firebase timestamp to real time

news thus reducing the overall cost.

### 3. Save time :

User is able to search popular news by using few clicks on screen and few search keywords thus saving his valuable time

### 4. Save effort :

In current time people has no such time to cope-up with all possible thing to go at all possible places so that all thing regarding current popular news,messages chat, blog post will come at my proposed project.

## 4.4 Test Cases

The aim of the application testing process was to determine all defects in our project .The program was subjected to a set of test inputs and various observations were made and based on these observations it will be decided whether the program behaves as expected or not

Items to be tested :-

1. create Account

2. Login in Account
3. Home Page
4. Home Page Fragments
5. User Profile
6. Chatting feature
7. Chat messages status
8. Blog Post
9. Like and Comments
10. All user or Find friends
11. User Status
12. User Online and Offline Status
13. Recent Popular News
14. Specific News
15. Share News
16. News open with browser

#### **4.4.1 Test for Home Page Activity**

There is following scenario come over user home page activity...

- When user gets login or register then user will come this home page activity
- This home page activity contains four fragments and one menu bar.
- Fragment CHATS contains list of all the chats with user friends and if user clicks any of his friend he or she will go to his chat list with friend.
- Fragment FRIENDS contains list of user friends and if user clicks any of his friend then he or she will find two options first chats and second their profile.
- Fragment REQUESTS contains list of friend request and their type sent or received and if user clicks any of them from list then he or she will go to their profile .

- When user clicks on POST fragment then he or she will be reached at blog post page.
- Menu Bar contains five things..
  1. All Users Post
  2. Popular News
  3. All Users List or Find Friends.
  4. User account setting
  5. Logout

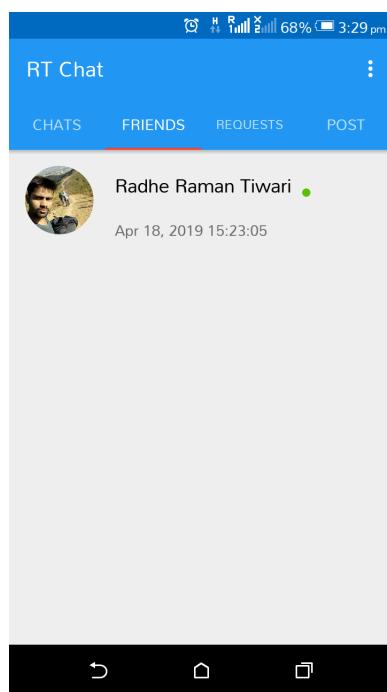


Figure 4.5: Home page activity contains four fragments.

#### 4.4.2 Test for Profile Page Activity

There is following scenario come over user profile page activity...

- It contains User profile image, user name, user status
- It also contains two button first button(i.e CHANGE IMAGE) for profile picture change and second button(i.e CHANGE STATUS) for user status change page activity.

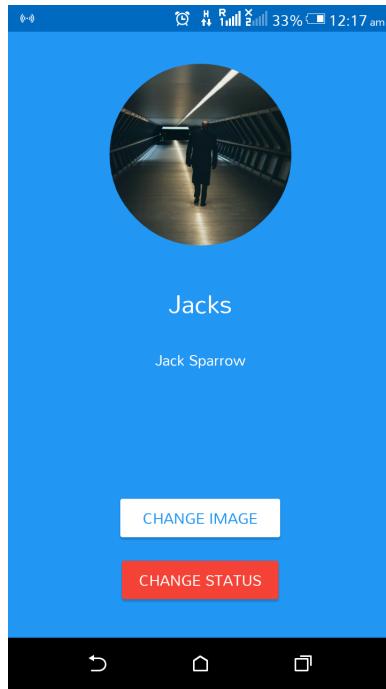


Figure 4.6: User profile page activity.

#### 4.4.3 Test for Blog Post Page Activity

There is following scenario come over blog post page activity...

- It contains three fragments first Home, second Notification and third Account fragment.
- In Home fragment, user will see his and his friends blog post images and their description.
- In home fragment user can also like and commands to his or her and his or her friends blog posts.
- In blog post activity, there is one plus image when user clicks on it then it goes to new blog post activity where can add new blog image and their description.
- When user clicks on Account fragment then it goes to user profile.

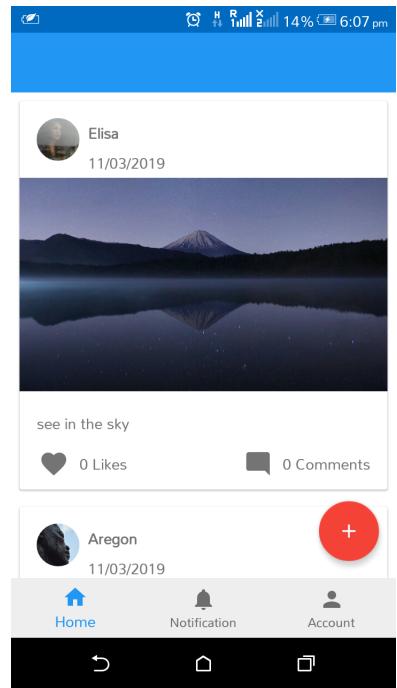


Figure 4.7: Blog post page activity.

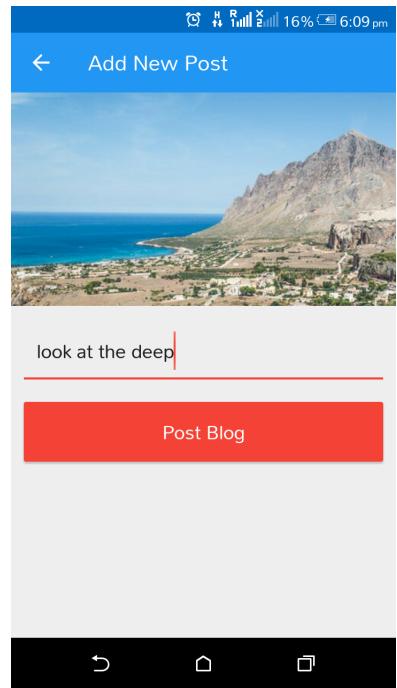


Figure 4.8: New Post activity.

# **Chapter 5**

## **Conclusion**

There is always a room for improvements in any apps. Right now we are just dealing with text and images communication as well as most popular recent news. There are several android apps which serve similar purpose as this project, but those apps were rather difficult to use and provide confusing interfaces and my proposed application has better user interface and time saving in terms of better user experience and make the thing together as well. A positive first impression is essential in human relationship as well as in human computer interaction. This project hopes to develop a chat service as well as news services Android app with high quality user interface.

This thesis has shown what Android is and how it works with XML in combination with Java. It has described how to setup a development environment and the emulator (AVD). It has showed what textbooks did not focus on, such as how the binding between XML and Java work on Android. Issues of versioning of Android is one of most common problem will not come with this thesis. The thesis has also discussed the advantages with Android software environment, and has lot of potential over Android world. The author's understanding is that Android and its SDK is a feature of the future. In part due to the fact that it will be fun for most phone users to be able to add, develop and equip their phones with new features and personal preferences, it will open a new era in sharing open source software components, such free utilities and games, for mobile devices. Furthermore, Android has most popular operating system, and the author is convinced that the development environment and tools will be improved and enhanced in the future, making the development process effective and efficient.

## 5.1 Future direction and Scope

This application has a lots of potential to grow in near future in today's life social media is very popular and it has lots of opportunities as well.

In future we may be extended to this application include features such as:

- File Transfer
- Voice Message
- Video Message
- Audio Call
- Video Call
- Group Messages
- Group Call

## **Appendix A**

### **Screenshot and Description of the Implemented System**

#### **A.1 Screenshot**

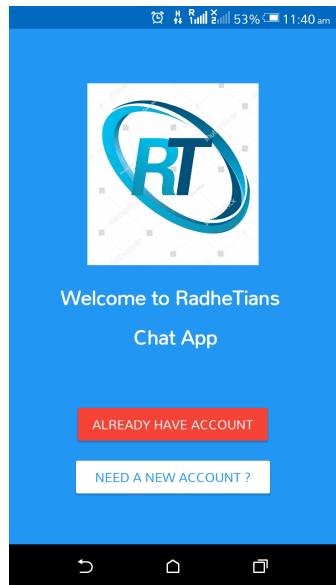


Figure A.1: Start Page Activity.

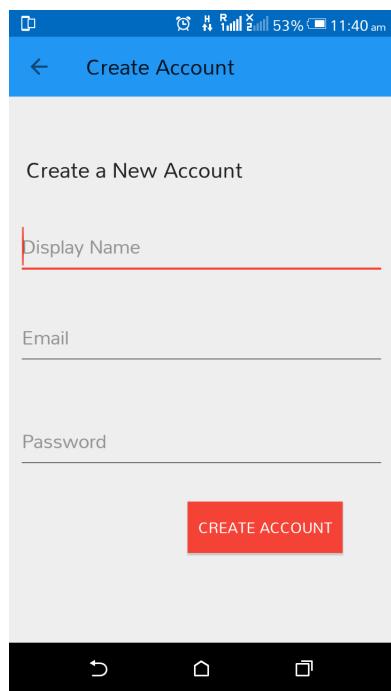


Figure A.2: User Register Activity.

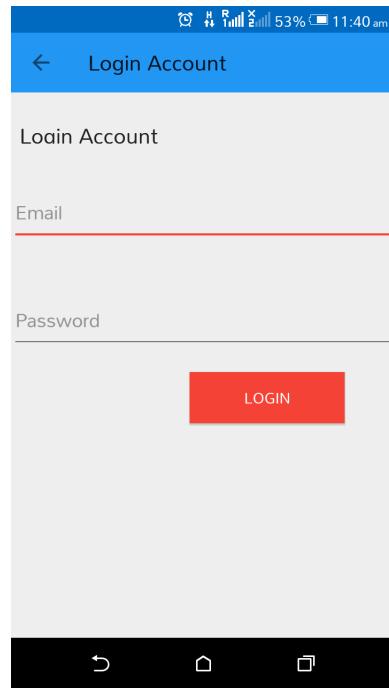


Figure A.3: User Login Activity.

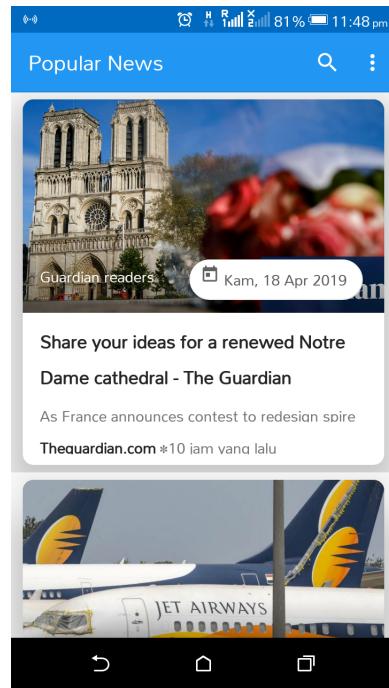


Figure A.4: Popular News Page Activity.



IPL 2019 LIVE SCORE, DC vs MI

Match at Feroz Shah Kotla: Rahul

Chahar's three derails Delhi Capitals -

Firstpost

Firstpost.comnull \* 7 iam yano lalu

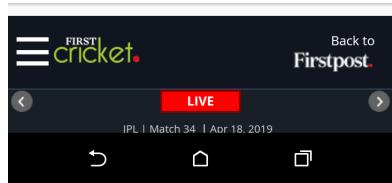


Figure A.5: Specific News Activity.



Figure A.6: Other User Profile Activity.

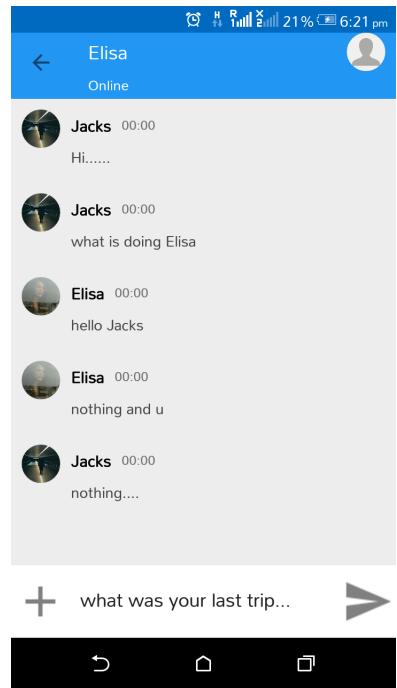


Figure A.7: Individual Chat Activity.

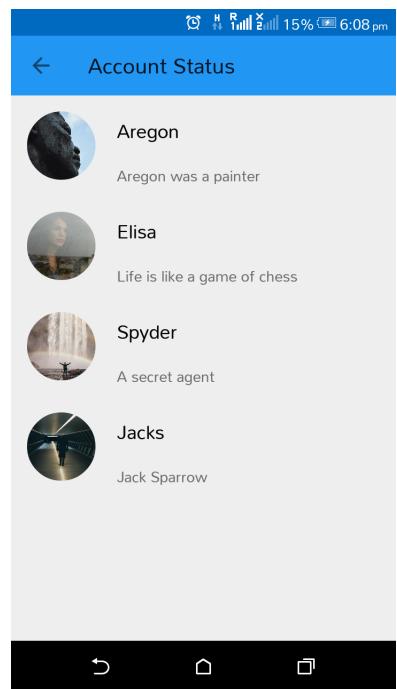


Figure A.8: Find Friends Activity.

# **Appendix B**

## **User manual**

### **B.1 System Specification**

#### **B.1.1 Hardware requirement :**

- Android Version should be greater than 5.1(i.e API 22)
- Ram- minimum :- 512 MB
- Hard disk—minimum :- 5 GB
- Processor :- Qualcomm's Snapdragon 545 or thing else

#### **B.1.2 Software requirement :**

there is no software requirement for this application because Android has in-build all Software requirement.there is nothing to do with this application just take APK file to your Android device and install it.

## **B.2 Installation**

Installation process is very easy, there is nothing to do with this just get APK file(i.e rtchat.APK) and install to your Android device installation will take hardly 1-2 minutes and installation user will be able to user it.

# Bibliography

- [1] Stack Overflow : *Android Notes for Professionals book*. Addison-Wesley, Reading, Massachusetts, 1993.
- [2] John Wiley & Sons Inc : *Android Application Development for Dummies* .
- [3] Android developer : <https://developer.android.com/docs>
- [4] News Doc : <https://newsapi.org/docs>
- [5] Firebase Doc : <https://firebase.google.com/docs/android/setup>