**CHAPTER 1**

**AIMS AND OBJECTIVE OF THE PROJECT**

1. PROJECT OVERVIEW
2. WHAT IS SECURED CHAT APP?
3. HOW DOES IT WORK?
4. BACKGROUND
5. LITERATURE SURVEY
6. PROBLEMS IN USING CHAT
7. CONCLUSION

# 1 CHAPTER 1: INTRODUCTION TO PROJECT

## PROJECT OVERVIEW

Nowadays, everyone all over the world uses chatting application over the standard calls because it’s convenient and hassle free. But at the same time this arose a problem too, many chatting applications uses very high data storage but doesn’t even provide any privacy and security for peace of mind of the user

## WHAT IS SECURED CHAT APP?

Secured Chat App is a Chatting Application that will provide user with better connectivity, Privacy and Security as well. Secured Chat App is a Camouflage Application that will disguise itself like any In-Built Application already installed on your device. Chatting may refer to a kind of communication over the Internet that offers a real-time transmission of text messages from sender to receiver.

## HOW DOES IT WORK?

Chat connects people together in a single window for text-based conversations. You can also send voice messages, connect your webcam and video chat and more from some chatrooms. Chat messages are generally short to enable other participants to respond quickly. Thereby, a feeling like a spoken conversation is created, which distinguishes chatting from other text-based online communication forms such as Internet forums and email.

Secured Chat App is a real-time messaging Application, which makes use of Firebase server. This Server facilitates instant messages transfer between two or many user at any given second.

Android is the programming language used by Secured Chat App. Firebase is free online database provider which uses Android language. It was preferred by techies and developers as it was suitable for accommodating huge communication system with high sustainability.

Android also helps by quickly adapting to instant updates and hot fixes. Whatever may be the improvisations or coding improvements, Android helps in quickly pushing the changes to live implementation without needing restarts. Thus, any changes can be rolled out in quick time and succession.

XMPP (Extensible Messaging and Presence Protocol) is being used by Application. To ensure network security, DSL is being used. This ensures private data transfer.

As and when message is sent, it is queued on the server. The message waits in the queue until the recipient reconnects to receive the message. As and when a message is delivered the sender gets notified by a double check mark near the message.

## BACKGROUND

Above mentioned Application will become live and available for all the people in the App market very soon. This Application covers every Standards and Essence of a Secured Chatting Application that a normal person would need and want for their Application. It is a very innovative , reliable, and very much user friendly as it comes. It provides the user with options for their custom profiles at the time of registration process so that they can design their profile as they want.

Most enhancing and advantageous feature of this Application is that even a novice or new to technology person can also make use of our Application without any issues. Also, we made sure that our users won’t have to worry about his or her personal details leak or losing any of their data, as it will be always being stored on our secured and protected online database and can be only accessed by admins or developers of this projects.

We made sure that we are there to help you with your queries and question at any stage while using our Application. As this Project is totally new and innovative we made attempt to make & design some unique features that will only be available to the users while they use our Application.

The project itself allows users to get new ideas of its uses for this Application and how well they can use this is totally upon them. As this project will be still In-Developing stage when we will be launching this Application in market we will be accepting each and all suggestions and queries from the users for better enhancement and improvement of not just the Application but the root project itself.

## **LITERATUERE SURVEY**

Internet communication is getting more and more popular among the public. Apart from using telephones or automobiles and sending mails, people can now communicate with each other through the chat technology. The chat, in fact, is a kind of Internet technology that supports human-to-human communication. WhatsApp, for instance, is one of the most used chat. Over the past eight years, with the advanced level of technology, there is an increasing trend of using WhatsApp for communication. With WhatsApp, users can chat, send messages, music, videos, Gifs, files and URL’s. Because of the proliferation of using the chat like WhatsApp, studies have been focused mainly on its impact on our society.

Much of the work stresses the good impact of the chat. Hauben’s (1997) writing suggested that as the impact or influence of first impressions is removed, users are free to communicate without fears, limits or apprehension through the chat. This statement actually points out the main reason for the increasingly use of the chat. Only one advantage, however, seems inadequate to attract such a huge number of users to use the chat, so it seems that there may be other benefit. Accordingly, Licklider (1997) claimed that people can communicate online with others who have similar goals and interests, thus their life will be enriched and communication will be more productive and more enjoyable then. Although Licklider is actually the prophet of the Net, it seems that the chat really has this benefit.

Some studies, however, have taken a different approach by looking not so much on the advantages of the chat, but focusing more on its related problems. Randall (1997), for instance, mentions that problems have been existed. First of all, there is no doubt that the chat users will not use their real identities for communication. They will rather create a new cyberspace identities which are very different from their real ones. Because of this, Randall argued that such behaviour makes people difficult to switch back and forth between these two identities. To him, those who have developed multiple cyberspace identities for Internet communication are the most sophisticated rhetoricians on the Internet.

While educators and students are expecting online education to be existed, Randall has already shown concern on the consequence of using such kind of education. In his point of view, the traditional teacher dominance of the classroom will be reduced, no matter this education is workable or not, because of the poor financial situation of the government. It seems that unemployment will be resulted in the near future.

While educators and students are expecting online education to be existed, Randall has already shown concern on the consequence of using such kind of education. In his point of view, the traditional teacher dominance of the classroom will be reduced, no matter this education is workable or not, because of the poor financial situation of the government. It seems that unemployment will be resulted soon.

## PROBLEMS IN USING CHAT

The key problems referred to time and again in the literature relate to technology as such, especially the need for reliable technology. A chat session will easily be spoiled if the software does not work or if any of the participant has connectivity problems. This can discourage participants from further experiments. However, growing familiarity with the medium outside the classroom is making this kind of problem less likely.

Some people also have problems with the abbreviated, highly colloquial kind of language used in chat sessions but this is not a universal concern. As a language teacher Freiermuth expressed concern about the quality of students’ language and warns teachers to try to curb excessive use of abbreviations.

## CONCLUSION

To sum up, the chat has good impact on the society but problems exist at the same time. However, these problems are not serious in fact. Therefore, even if these problems exist continuously, the chat technology will still become central to our lives and it has already begun.

**CHAPTER 2**

**AIMS AND OBJECTIVE OF THE PROJECT**

1. AIMS OF THE PROJECT
2. OBJECTIVE OF THE PROJECT
3. NEED OF THE PROJECT
4. SOME IMPORTAT BENEFITS OF PROJECT ARE LISTED BELOW

# CHAPTER 2: AIMS AND OBJECTIVES OF THE PROJECT

## AIMS OF THE PROJECT

Secured Chat App is developed as a chatting application that will connect user all over the world to get connected to each other on internet and while users do that Secured Chat App will provide them with Privacy and Security, so that they don’t have to worry about anything.

## OBJECTIVE OF THE PROJECT

* Web Support (Internet Connectivity).
* Increasing the privacy of the application by making it camouflaged.
* Extending this application by providing Authorisation service.
* Creating Database and maintaining users.
* To gather information on approaches currently taken on the use of CHAT in online learning in higher education as well.

## NEED OF THE PROJECT

1. Chatting Application are used in our day to day life but modern apps don’t provide with much privacy and security.
2. We wanted to provide people with an app that will not only secure their local chats but also let them access them without any hassle.
3. While our app secures our users chats we wanted them to be at peace knowing that their data is totally protected in our database and no one can access them on their mobile devices.

## SOME IMPORTANT BENEFITS OF PROJECT ARE LISTED BELOW

### Affordable:

As you might have already guessed our Secured Chat App is free for use to anyone. It is freely available on Google Play Store so just go there and download it on Android supported devices.

### Easy to use:

We have already made our Application user friendly, that means it will be very simple and easy to use. But we also know that there might be some unexpected events where user might not understand something so for those cases we have made a FAQ page and also inserted a Slider in our App that will be launched on your first Install/launch of the App.

### Enhanced Performance:

For more efficient and easy use of our application we have developed it to be very low and simple but at the same time the technology that have been implemented at Front-End and Back-End of our Application is as Powerful & Efficient as it comes “Android” for Front-End and “Firebase” for our Back-End connectivity with our database.

### Improvement in the Security:

A standard chatting application does not have any security within its own it needs a third-party Application that uses even more space. Solution to this hindrance is our project, Secured Chat App provides user with much more than just pin-code protection or pattern-lock protection. It disguises the whole app in your App drawer with another protection mechanism already installed within.

### Safe:

Secured Chat App is the most Secured chatting application out there in stores and its big database & unique security mechanism make it even more safe for everyone to use.

**CHAPTER 3**

**PROJECT PLANNING AND MANAGEMENT**

1. PROJECT DEVELOPMENT LIFECYCLE
2. TASKS AND MILESTONES

# CHAPTER 3: PROJECT PLANNNING AND MANAGENMENT

## PROJECT DEVELOPMENT LIFECYCLE

### Initiation

We got the idea of this project in our 5th semester and we started our studies and information gathering on this project from then till 2nd week of 6th semester.

### System Concept Development

In Secured Chat Application, we provide users kind of communication over the Internet that offers a real-time transmission of text messages from sender to receiver.

### Planning

By group discussion and initial planning, we understood most of the requirement of our projects and then we starting to plan accordingly.

### Requirement Analysis

Here we gather all the requirement of every user which is required to develop in system. Also, we gathered software and hardware requirement we needed to develop intended system. Then we create final requirement document list.

### Design

We transform the detail requirement into complete system, design documents and focuses on how to deliver required functionality. We design final UML diagrams of each for our system.

### Development

We prepare test cases, procedure, testing, coding, compiling, refining a program then performing test readiness review.

### Integration and Test

We demonstrate that developed system confirms to provide requirement as specified in the functional requirement document conducted by quality assurance, end user, automated testing and analysis.

### Implementation

We implemented the system in production environment, resolution problem identity in integration and test phases.

### Operation and Maintenance

We describe the task to operate and maintain information system in product.

### Disposition

Here we describe how system works and create user manual for end user. Her importance is given to proper implementation of data.

## TASKS AND MILESTONES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr No | Test Case Step | Expected Cases | Status | Build Test |
| 1 | Database connectivity | Connection should be successful | Connection Successful | Yes |
| 2 | Shared Preferences | Connection should be successful | Connection Successful | Yes |
| 3 | Users Registration | User must get his personal detail stored on firebase | Personal details stored on firebase | Yes |
| 4 | Sign in with Instagram | Signed in with Instagram account | Successful | Yes |
| 5 | Sign in with LinkedIn | Signed in with LinkedIn account | Successful | Yes |
| 6 | Sign in with Twitter | Signed in with Twitter account | Successful | Yes |
| 7 | Users Profile | Users profile getting created and stored on firebase | Profile getting created | Yes |
| 8 | Set Frequency | Users Set frequency getting stored on Database | Frequency stored on Shared Preference and firebase | Yes |
| 9 | User Login | User login with registered credentials | User Logged in | Yes |
| 10 | Chatroom | User selecting friend and going in chatroom with him or her | User and his or her friend getting in a chatroom | Yes |
| 11 | Search user | User searching a friend and selecting his or her | Friend found and going in chatroom with him or her | Yes |
| 12 | Send message | User sent message getting stored on firebase | Message was stored on firebase with user’s unique user id and his or her message in one node | Yes |
| 13 | Retrieve Message | User message was sent to his or her friend | Message was retrieve by intended recipient and was shown in chatroom | Yes |
| 14 | Logout | User getting logged out with his credentials | User was logged out | Yes |

Table 3.2 Tasks and Milestones

**CHAPTER 4**

**TECHNOLOGIES USED**

1. HARDWARE AND SOFTWARE REQUIREMENT
2. ANDROID
3. ANDROID FEATURES
4. ANDROID STUDIO FEATURES
5. FIREBASE
6. FIREBASE + GOOGLE CLOUD STORAGE
7. FIREBASE SERVICES

# CHAPTER 4: TECHNOLOGIES USED

## HARDWARE AND SOFTWARE REQUIREMENT:

### Hardware requirements:

|  |  |  |
| --- | --- | --- |
| Sr No | Name of Components | Requirements |
| 1 | RAM | 512 MB |
| 2 | Processor | Qualcomm |
| 3 | Processor Cores | Dual Core |
| 4 | Processing Speed | 1.0 GHz |

Table: 4.1.1 a) Minimum Hardware requirements

|  |  |  |
| --- | --- | --- |
| Sr No | Name of Components | Requirements |
| 1 | RAM | 1.5 GB |
| 2 | Processor | Qualcomm |
| 3 | Processor Cores | Quad Core |
| 4 | Processing Speed | 1.8 GHz |

Table: 4.1.1 b) Recommended Hardware requirements

### Software requirements:

|  |  |  |
| --- | --- | --- |
| Sr No | Android Versions | Supported |
| 1 | Marshmallow |  |
| 2 | Lollipop |  |
| 3 | KitKat |  |
| 4 | Jelly Bean |  |
| 5 | Ice-Cream Sandwich |  |

Table: 4.1.2 Software requirements

## ANDROID

Android is an open source and Linux-based operating system for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies. This tutorial will teach you basic Android programming and will also take you through some advance concepts related to Android application development.

Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android.

The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 where as the first commercial version, Android 1.0, was released in September 2008.



Fig: 4.2 Android

## ANDROID FEATURES

|  |  |
| --- | --- |
| **Sr. No.** | **Feature & Description** |
| 1 | **Beautiful UI**  Android OS basic screen provides a beautiful and intuitive user interface. |
| 2 | **Connectivity**  GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX. |
| 3 | **Storage**  SQLite, a lightweight relational database, is used for data storage purposes. |
| 4 | **Media support**  H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg, Vorbis, WAV, JPEG, PNG, GIF, and BMP. |
| 5 | **Messaging**  SMS and MMS |
| 6 | **Web browser**  Based on the open-source Web Kit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3. |
| 7 | **Multi-touch**  Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero. |
| 8 | **Multi-tasking**  User can jump from one task to another and same time various application can run simultaneously. |
| 9 | **Resizable widgets**  Widgets are resizable, so users can expand them to show more content or shrink them to save space. |
| 10 | **Multi-Language**  Supports single direction and bi-directional text. |
| 11 | **GCM**  Google Cloud Messaging (GCM) is a service that lets developers send short message data to their users on Android devices, without needing a proprietary sync solution. |
| 12 | **Wi-Fi Direct**  A technology that lets apps discover and pair directly, over a high-bandwidth peer-to-peer connection. |
| 13 | **Android Beam**  A popular NFC-based technology that lets users instantly share, just by touching two NFC-enabled phones together. |

Table: 4.3 Android Feat

## ANDROID STUDIO FEATURES

### The Official IDE for Android

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

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

### Instant Run

Push code and resource changes to your app running on a device or emulator and see the changes instantly come to life. Instant Run dramatically speeds up your edit, build, and run cycles, keeping you "in the flow."

### Intelligent code editor

Write better code, work faster, and be more productive with an intelligent code editor that helps you each step of the way. Android Studio is built on IntelliJ and is capable of advanced code completion, refactoring, and code analysis.

### Fast and feature-rich emulator

Install and run your apps faster than with a physical device and test your app on virtually any Android device configuration: Android phones, Android tablets, Android Wear, and Android TV devices.

The new Android Emulator 2.0 is faster than ever and allows you to dynamically resize the emulator and access a suite of sensor controls.

### Robust and flexible build system

Easily configure your project to include code libraries and generate multiple build variants from a single project.

With Gradle, Android Studio offers high-performance build automation, robust dependency management, and customizable build configurations.

### Develop for all Android devices

Target multiple form factors with a single project to easily share code among your different versions of your app. Android Studio provides a unified environment to develop apps for Android phones, tablets, Android Wear, Android TV, and Android Auto.

### Code templates and GitHub integration

Start projects with code templates for patterns such as navigation drawer and view pagers, or import Google code samples from GitHub. Android Studio's project wizards make it easier than ever to add code in a new project.

## FIREBASE

Fig: 4.5 Firebase

Firebase is a mobile and web application platform with tools and infrastructure designed to help developers build high-quality apps. Firebase is made up of complementary features that developers can mix-and-match to fit their needs. The team is based in San Francisco and Mountain View, California. The company was founded in 2011 by Andrew Lee and James Tamplin. Firebase's initial product was a real-time database, which provides an API that allows developers to store and sync data across multiple clients. Over time, it has expanded its product line to become a full suite for app development. The company was acquired by Google in October 2014 and a significant number of new features were featured in May 2016 at Google I/O.

Firebase evolved from Envolve, a prior start up founded by Tamplin and Lee in 2011. Envolve provided developers an API that let them integrate online chat into their websites. After releasing the chat service, Tamplin and Lee found that the service was being used to pass application data that wasn't chat messages. Developers were using Envolve to sync application data such as game state in real-time across their users. Tamplin and Lee decided to separate the chat system and the real-time architecture that powered it, founding Firebase as a separate company in April 2012.

Firebase raised $1.4 million in seed funding in May 2012 from Flybridge Capital Partners, Greylock Partners, NEA and others. The company raised $5.6 million in Series A funding from Union Square Ventures and Flybridge Capital Partners in June 2013. On 21 October 2014, Firebase announced it had been acquired by Google for an undisclosed amount. On October 13, 2015 Google acquired Divshot to merge it with Firebase team, for an undisclosed amount. Before the acquisition, Divshot had raised $1.18 million in two rounds of funding, according to TechCrunch. Since the acquisition, Firebase has grown inside Google and expanded their services to become a unified platform for mobile developers. Firebase now integrates with various other Google services to offer broader products and scale for developers. The vision of Firebase stays the same and aim to help developers build better apps and grow successful businesses.

## FIREBASE + GOOGLE CLOUD STORAGE

### Google Cloud Platform

Since expanding Firebase to become Google's mobile application development platform at Google I/O last year, their amazing community of developers has created over 1 million Firebase projects. While Firebase is a full suite of products for building and growing apps, they know that some apps need more than they offer out-of-the-box. That's why they’re bringing Firebase much closer to Google Cloud Platform (GCP) to serve even the most demanding applications, whether you're a new start up or a large enterprise.

### Product Integrations

Firebase already shares the same account and billing system as GCP, so you can attach Firebase services to your GCP project and vice-versa. This makes for powerful combinations, such as exporting raw event data from Firebase Analytics into Big Query for ad-hoc analysis. Starting today, they're beginning to share products too.

First, Firebase developers have been asking for ways to extend their app's functionality without spinning up a server, and Cloud Functions for Firebase lets you do just that. Cloud Functions is our new event-driven serverless compute offering that enters public beta today. The infrastructure is shared between Cloud and Firebase, allowing you to invoke a function or access resources from throughout the Cloud/Firebase ecosystem.

Next, they're bringing Firebase Storage closer to Cloud Storage. Firebase Storage launched 10 months ago, and lets you easily upload and download files from your device directly to Cloud Storage. Previously they gave you a single bucket for your files. Now they're fully aligning the two products and letting you use any Cloud Storage bucket, from any global region and from any storage class, all straight from the Firebase SDK. To reflect this alignment, renaming the product Cloud Storage for Firebase.

### Streamlined Terms of Service

They love lawyers almost as much as developers, so they’re extending GCP's Terms of Service to cover several Firebase products. This makes Firebase and Cloud simpler to evaluate and use together. Products to be covered include: Authentication, Hosting, Storage, Functions, and Test Lab. Our streamlined Terms of Service will take effect soon.

### The Big Picture

Firebase brings together the best of Google on mobile, whether that's Google's flagship advertising solutions like AdMob and AdWords, or Google's analytics expertise in the form of Firebase Analytics.

Google Cloud Platform lets you to benefit from the institutional knowledge Google has developed from almost two decades of running global-scale computing infrastructure. By bringing together the ease-of-use of Firebase with the full-range of GCP infrastructure offerings, they're better able to serve you up and down the stack. If you're a start-up using Firebase to quickly get to market, you can now easily scale into a full public cloud. If you're an existing business running on GCP who wants to ship a mobile app, they've got you covered too.

## FIREBASE SERVICES

### Analytics

At the heart of Firebase is Firebase Analytics, a free and unlimited analytics solution. See user behaviour and measure attribution from a single dashboard.

* Unlimited reporting of 500 event types, each with up to 25 attributes
* One dashboard to view user behaviour and cross-network campaign performance
* Demographic segmentation, including age, gender, and location, available out-of-the-box
* Export raw data to BigQuery for custom querying
* The Firebase Analytics Partner database includes leading mobile advertising technology platforms, which have been validated to measure and optimize app campaign performance.

### Develop

Build better apps and leave the operations to them. Save crucial development time and ship a high-quality, bug-free app.

#### Cloud Messaging

Deliver and receive messages across platforms reliably

#### Authentication

Reduce friction with robust authentication

#### Real-time Database

Store and sync app data in real-time

#### Cloud Storage

Store and serve content with ease

#### Cloud Functions

Run your mobile backend code without managing servers

#### Hosting

Deliver web content faster

#### Test Lab

Test in the lab, not on your users

#### Crash Reporting

Keep your app stable

### Grow

Acquire and engage the right users at the right time. Take the guesswork out of growth.

#### Notifications

Engage with users at the right moment

#### Remote Config

Customize your app on the fly

#### App Indexing

Drive organic search traffic to your app

#### Dynamic Links

Send users to the right place inside your app

#### Invites

Empower your users to share your app

#### AdWords

Acquire users with the reach of Google

**CHAPTER 5**

**PROJECT ANALYSIS**

1. USE CASE WITH SCENARIO
2. ACTIVITY DIAGRAM
3. CLASS DIAGRAM
4. COLLABORATION DIAGRAM
5. FLOWCHART DIAGRAM
6. SEQUENCE DIAGRAM’

# CHAPTER 5: PROJECT ANALYSIS

## USE CASE WITH SCENARIO

A use case is a static description of some way in which a system or a business is used, by it customer,

Its user or by other systems. A use case diagram shows how use cases are related to each other and how the users can get at them. Each bubble on a use case diagram represents a use case and each stick person represents a user. Each use case is more than just title such as it must include the actual steps involved in using the system or business.

### Registration

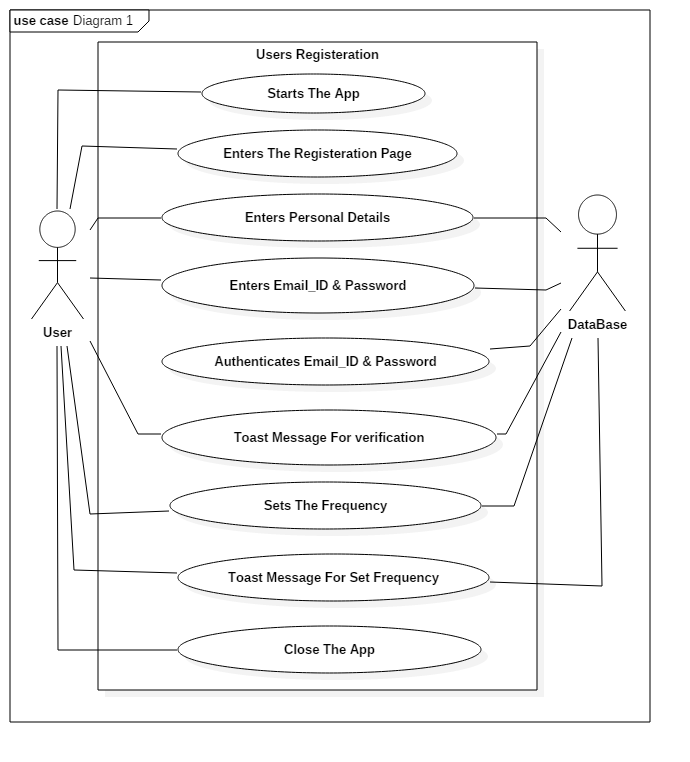


Fig: 5.1.1 Registration System

### Login System

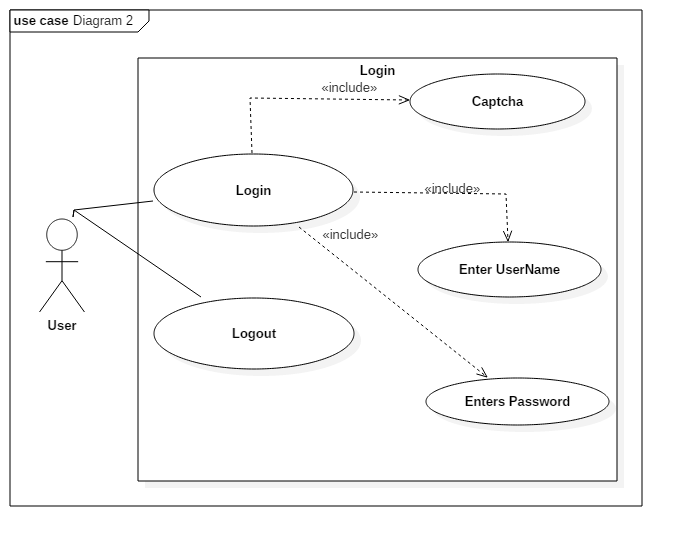


Fig: 5.1.2 Login System

### Authentication

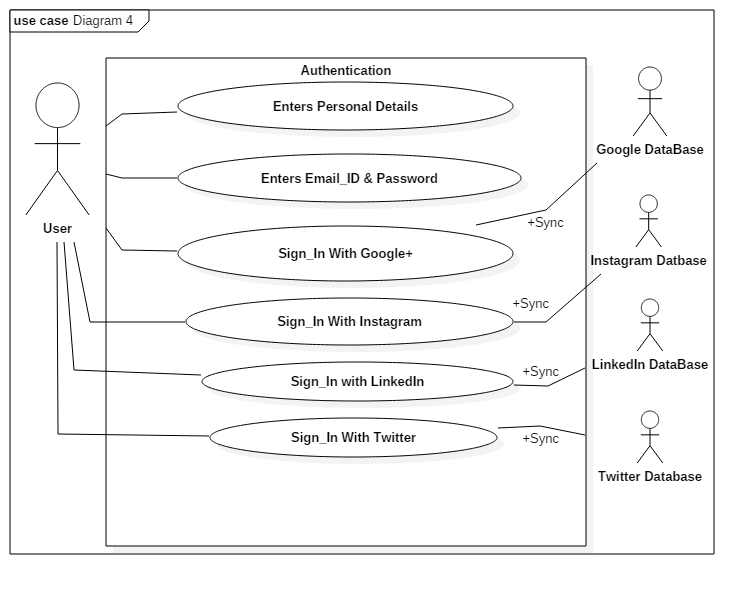


Fig: 5.1.3 Authentication

### Chatting System

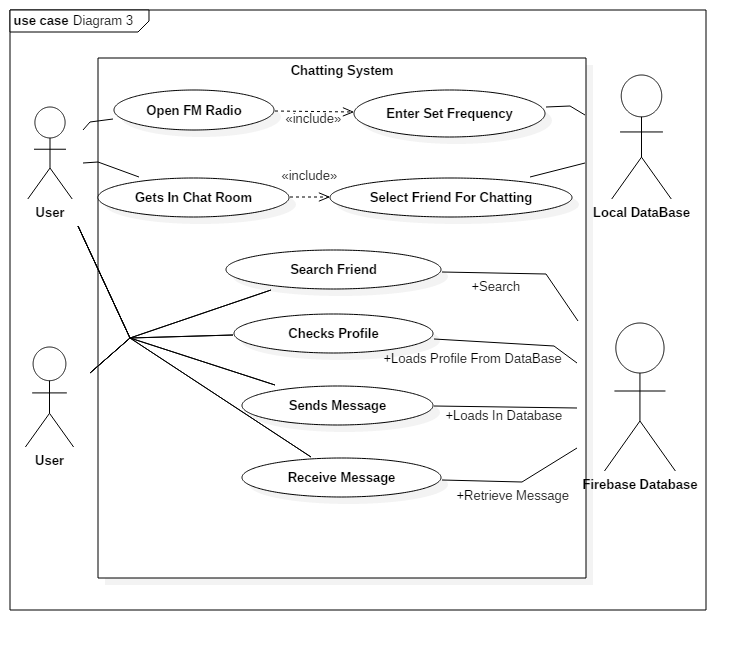


Fig: 5.1.4 Chatting System

## ACTIVITY DIAGRAM

UML includes another kind of diagram that can be useful during business modelling. An Activity diagram shows dependencies between (parallel) activities as we move from an initial starting point to desired goal.

They are similar to flowchart or petri nets, traditionally used to model program flow or human activity.

The figure shows the Activity diagram which necessary action in action state in rounded rectangle and transaction between activity states. It also includes fork for concurrent activity and also fork join.

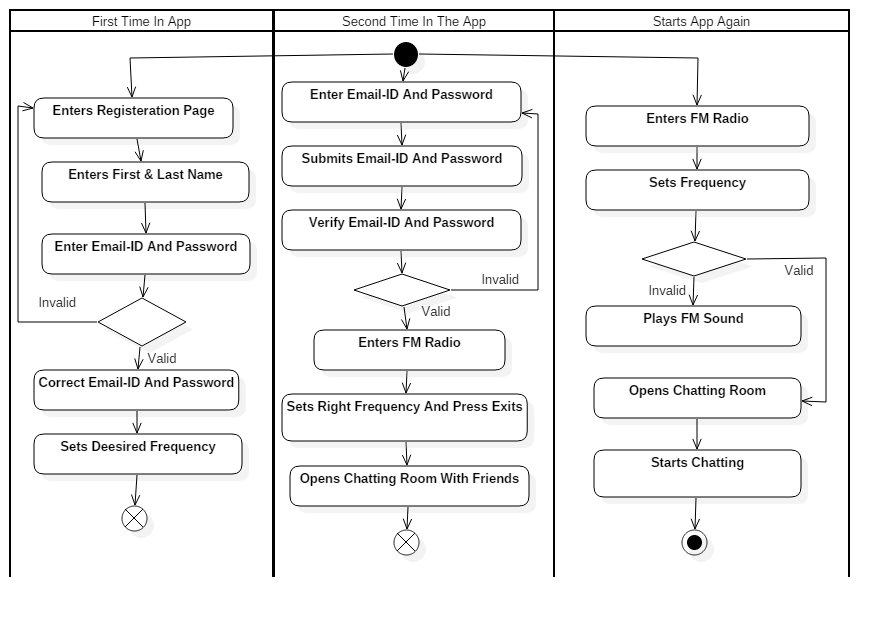


Fig: 5.2 Activity Diagram

## CLASS DIAGRAM

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application.

The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages.

The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram.

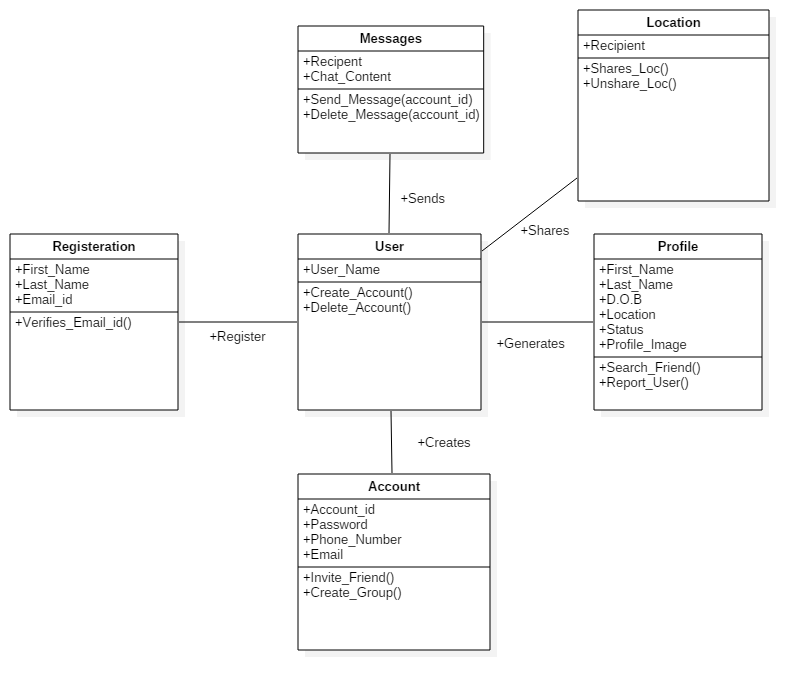


Fig: 5.3 Class diagram

## COLLABORATION DIAGRAM

Communication diagram also known as collaboration diagram is a kind of UML interaction diagram which shows interactions between objects and parts that represented as lifelines using sequenced messages in a free-form arrangement.

Communication diagram corresponds (i.e. could be converted to/from or replaced by) to a simple sequence diagram without structuring mechanisms such as interaction uses and combined fragments. It is also assumed that message overtaking (i.e., the order of the receptions is different from the order of sending of a given set of messages) will not take place or is irrelevant.

The following nodes and edges are drawn in a UML communication diagrams: frame, lifeline, and message. These major elements of the communication diagram are shown on the picture below.

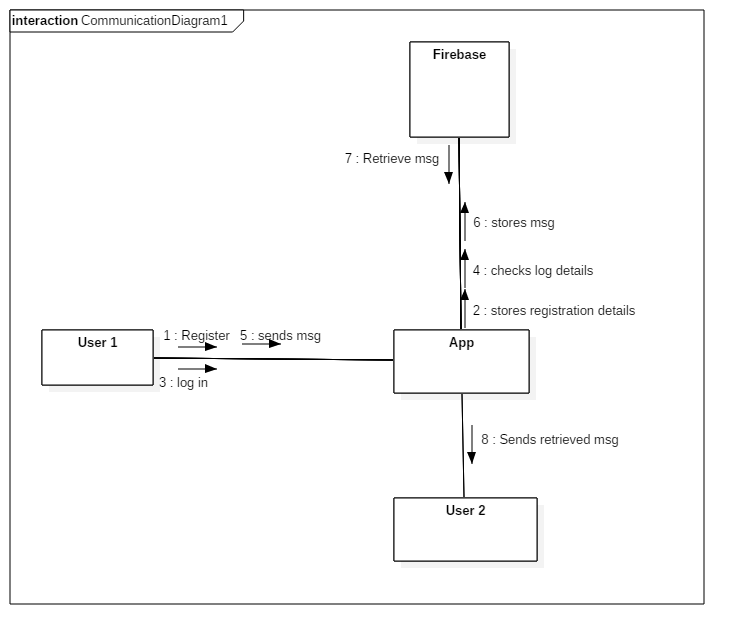


Fig: 5.4 Collaboration Diagram

## FLOWCHART DIAGRAM

A flowchart is a visual representation of the sequence of steps and decisions needed to perform a process. Each step in the sequence is noted within a diagram shape. Steps are linked by connecting lines and directional arrows. This allows anyone to view the flowchart and logically follow the process from beginning to end.

A flowchart is a powerful business tool. With proper design and construction, it communicates the steps in a process very effectively and efficiently.

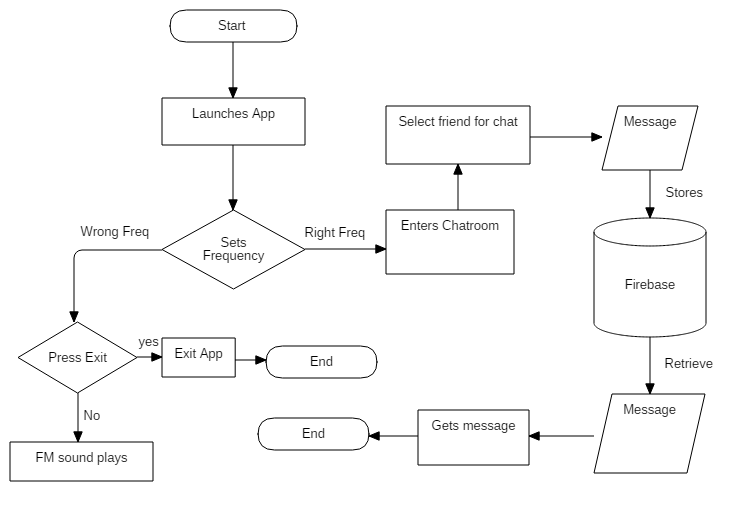


Fig: 5.5 Flowchart Diagram

## SEQUENCE DIAGRAM

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart.

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

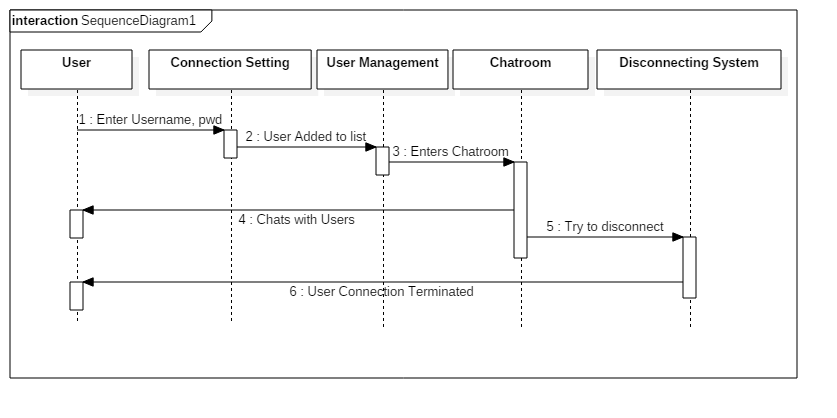


Fig: 5.6 Sequence Diagram

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

**CHAPTER 6**

**PROJECT IMPLEMENTATION**

1. USER INTERFACE

# CHAPTER 6: PROJECT IMPLEMENTATION

## USER INTERFACES

### User Login Interface

### C:\Users\Shreyash\Desktop\device-2017-03-16-184535.png

Fig: 6.1.1 User Login Interface

### Registration Interface

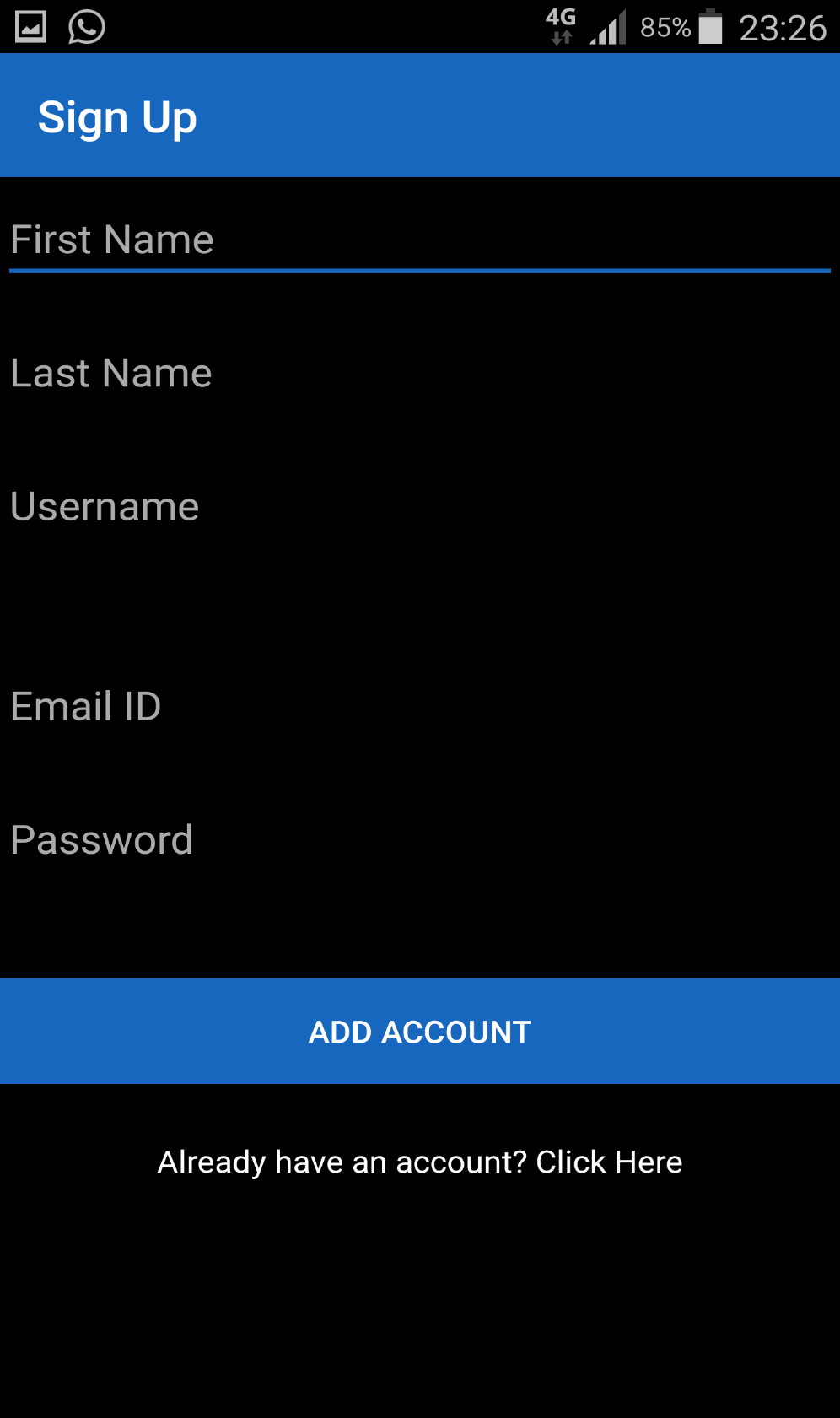


Fig: 6.1.2 User Registration Interface

### FM Radio Interface

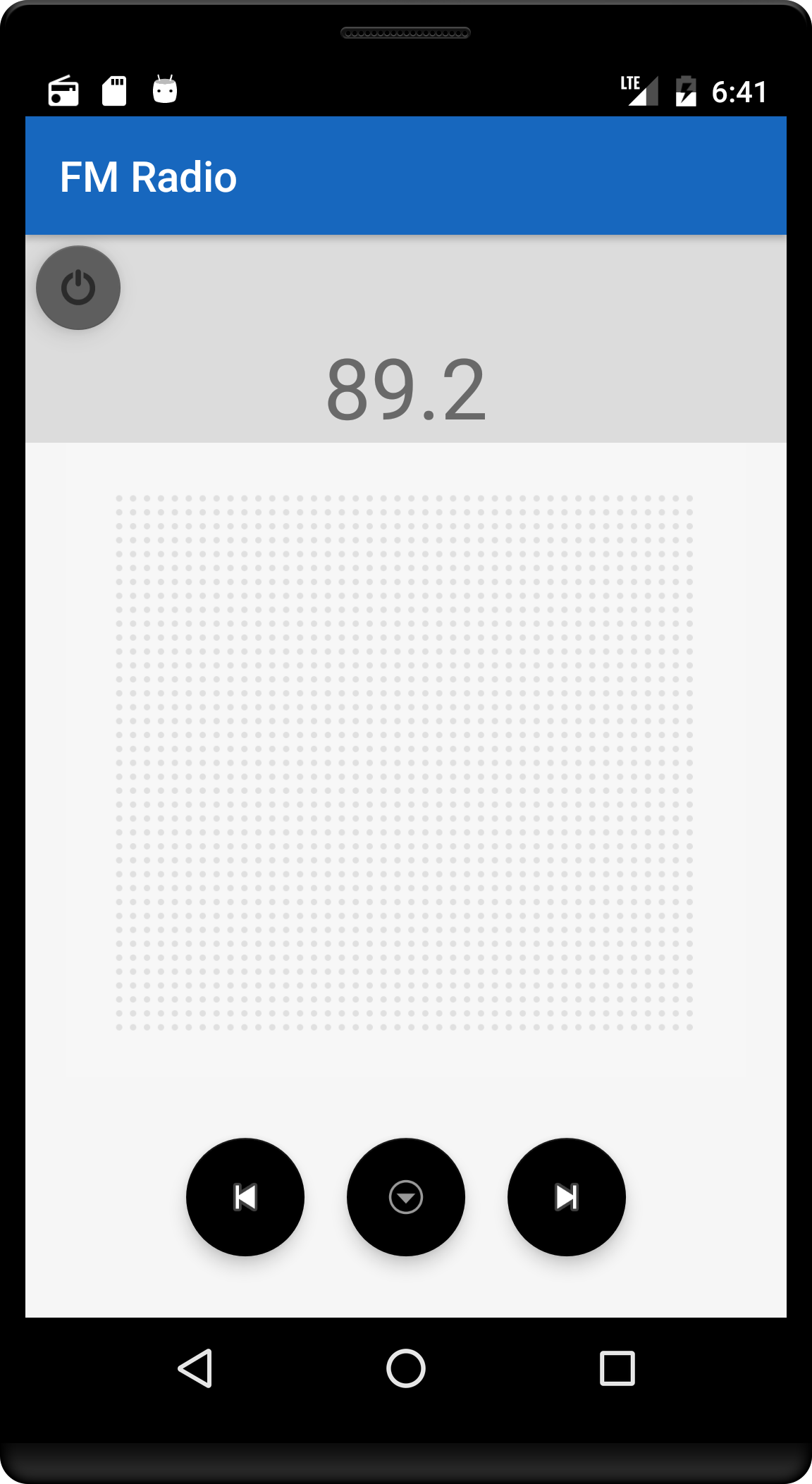


Fig: 6.1.3 FM Radio Interface

### Set Frequency Interface

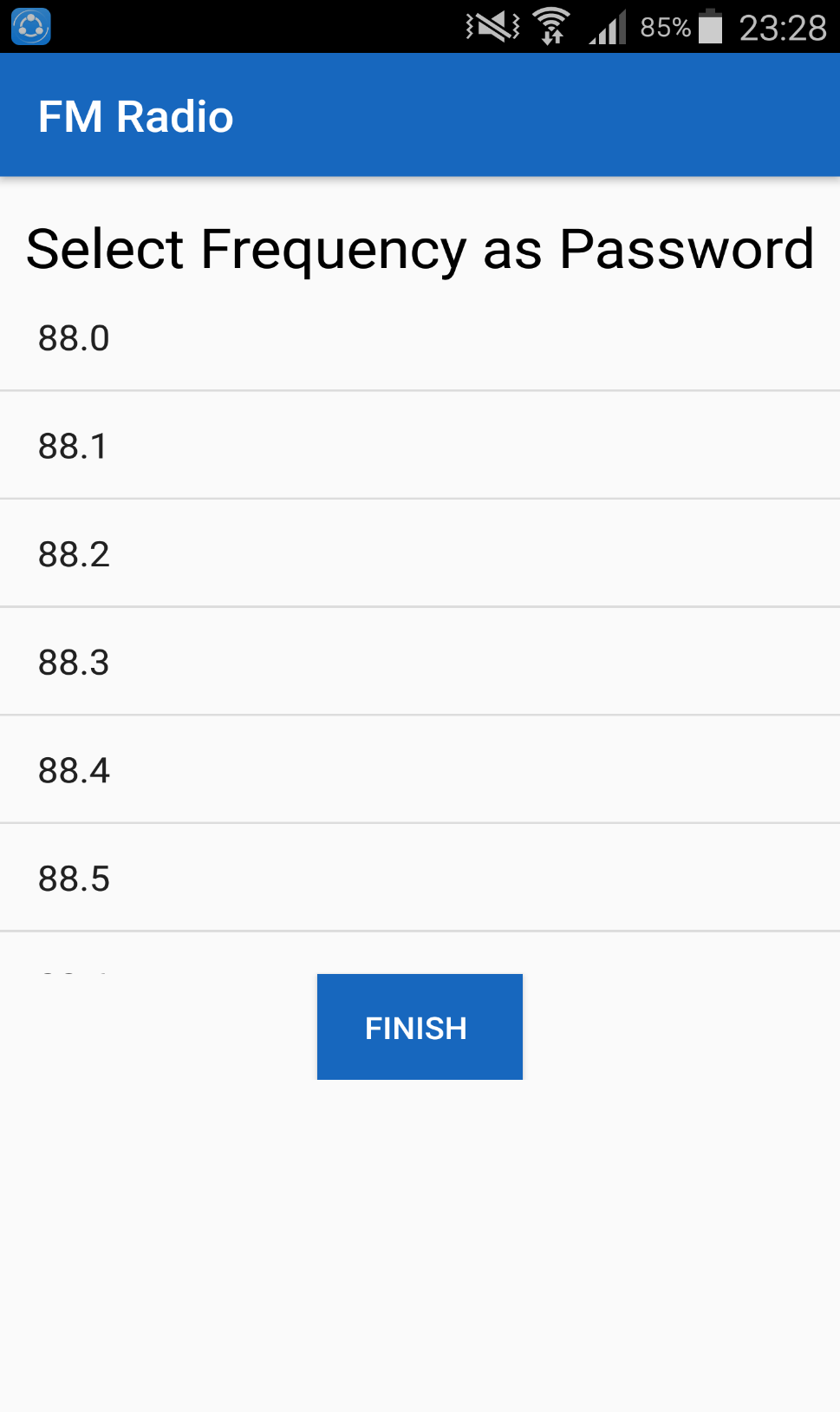


Fig: 6.1.4 Sets Frequency Interface

### Chat Screen Interface

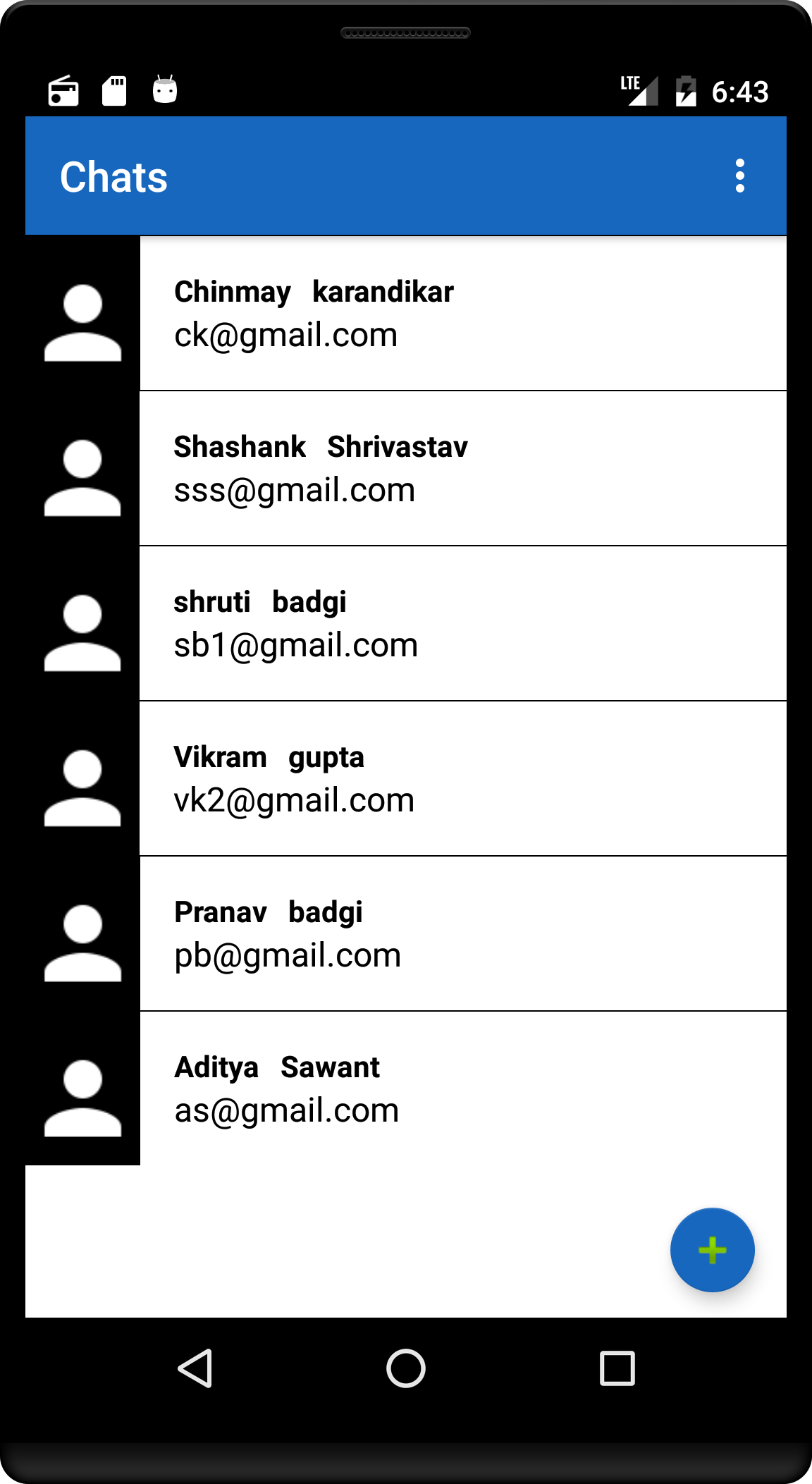


Fig: 6.1.5 Chat Screen Interface

### Chat room Interface



Fig: 6.1.6 Cha room Interface

**CHAPTER 7**

**PROJECT TESTING**

1. MANUAL TESTING

# CHAPTER 7: PROJECT TESTING

## MANUAL TESTING

These tests were performed by developers and tester in controlled situation and we got expected result from these tests. Also, this application was handed out to our friends and even they got the expected result to be satisfying.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No | Test Case Title | Description | Expected Output | Result |
| 1 | Username and password | User entering wrong username and password | Toast message showing error of wrong username and password | Successful |
| 2 | Username and password | User entering correct username and password | Toast message showing successful registration | Successful |
| 3 | Create profile | User creating profile with genuine credentials | User profile was created | Successful |
| 4 | Users details on firebase | Users credentials getting stored on firebase | User details was stored on firebase | Successful |
| 5 | User login | User entering wrong registered email-id and password | Toast message showing wrong email-id and password | Successful |
| 6 | User login | User entering registered email-id and password | Toast message showing logged in with username | successful |
| 7 | Frequency | User entering wrong frequency and pressing exit button | User exits the application | Successful |
| 8 | Frequency | User entering right frequency and pressing exit button | User getting to contact or friend list screen | Successful |
| 9 | Frequency | User going through different frequencies | User hears FM sounds | Successful |
| 10 | Minimize and maximize | User must get back to FM radio interface | User entered FM interface | Successful |
| 11 | User linking SNS | User signing in with Instagram | User registered with Instagram | Successful |
| 12 | User linking SNS | User signing in with LinkedIn | User registered with LinkedIn | Successful |
| 13 | User linking SNS | User signing in with Twitter | User registered with Twitter | Successful |
| 14 | Logout | User logging out with his credentials removed | Users logged out | Successful |
| 15 | Authentication | User registered email-id must be stored on firebase database and checked by GCP | User email-id submitted on firebase and checked by GCP | Successful |
| 16 | Real-time database | Users message and data must get stored on firebase in real time | Users data stored on firebase in real time | Successful |
| 17 | Contacts | User getting to see all available friends for chatting | User gets all contacts available for chatting | Successful |
| 18 | Confirmation Email | Users get a confirmation email | User gets a confirmation email | Successful |
| 19 | Forgot password | User retrieving forgotten password via email | User gets email with forgotten password | Successful |
| 20 | FAQ | Accessing FAQ page for user and sending queries via email | FAQ page accessed and user’s queries were stored | Successful |

Fig: 7.1 Manual Testing

**CHAPTER 8**

**CONCLUSION**

# CONCLUSION

This application was developed after detailed study and research on existing systems and applications. We feel very glad to conclude this report; this our first experience to perform such professional and large project work. Objective of this project was to satisfy user’s requirement, successful implementation of system, design a user friendly and easy to operate system while providing privacy and security at user end.

We have got the opportunity to learn conceptual and practical knowledge of software engineering, software analysis and real time experience of project implementation in Android.

At last we are very thankful to university to include this project as a part of our engineering in diploma. This project work was really gives us chance to learn something out of typical reference book.

It is our pleasure to say thanks once again to all of you on completion of our project work. By developing this system, we found ourselves on pathway to industry requirement.

**CHAPTER 9**

**REFERENCES**

# REFERENCE

1. INSTAGRAM

[http://www.instagram.com/secured\_chat/](mailto:http://www.instagram.com/secured_chat/)

1. FACEBOOK

[http://www.facebook.com/Secured-Chat-App-/936758519759967/](mailto:http://www.facebook.com/Secured-Chat-App-/936758519759967/)

1. TWITTER

[https://www.twitter.com/secured\_chat](mailto:https://www.twitter.com/secured_chat)