ALL IN ONE

AIO3

(Android Based Application)

A

PROJECT REPORT

ON

ALL IN ONE APP

SUBMITTED BY

GOVIND RAMA PARAB

UNDER THE GUIDENCE OF

PROFESSOR: SHRADDHA KOKATE

EXAMINATION BSC.IT, SEMESTER

University of Mumbai

UNIVERSITY OF MUMBAI

DEPARTMENT OF INFORMATION TECHNOLOGY

GHANSHYAMDAS SARAF COLLEGE OF ARTS AND COMMERCE

(Affiliated to university of Mumbai)

MUMBAI 400064

MAHARASHTRA

2021-2022

All In One App

A Project Report

Submitted in partial fulfillment of the

Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

Govind Rama Parab

Roll No. 30

Under the esteemed guidance of

Prof. Shraddha Kokate

DEPARTMENT OF INFORMATION TECHNOLOGY

GHANSHYAMDAS SARAF COLLEGE OF ARTS AND COMMERCE

(Affiliated to university of Mumbai)

MUMBAI 400064

MAHARASHTRA

2021-2022

DEPARTMENT OF INFORMATION TECHNOLOGY

GHANSHYAMDAS SARAF COLLEGE OF ARTS AND COMMERCE

(Affiliated to university of Mumbai)

CERTIFICATE

This is to certify that the project entitled, **"All In One App "**, is bonafied work of **Govind Rama Parab** bearing Seat No: (**30**) submitted in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY** from University of Mumbai.

Internal Guide **Coordinator**

External Examiner

Date: **College Seal**

Abstract

Over the past few years, things have changed dramatically and rapidly. The major reason behind these major changes is due to technology. Well, if we are talking about the technology then mobile apps play a major role in this. Whether it is online payment service, creating online meetings or streaming a new track, it is all shifted to an app. Smart phones are quickly replacing the computers, as they have wide range of functions, and can be carried into pockets easily and at anywhere. The rapid progress in mobile technology is proving to be the next level of learning and teaching. On the other side COVID-19 crisis has forced education systems worldwide to find the alternatives to face-to-face instructions. As a result, online teaching, meetings, learning, etc modes are been used by teachers, students, supervisors, researchers, and others on a larger scale. All these modes have worked finely as per the situations, but when it comes to grab information from these modes it becomes automatically a tedious task. Weather it is digital learning or meetings everyone faced with multiple, consecutive meetings, day after day, week after week. This journey everyone must have started with typically creating physical notes of points discussed in meetings, with the never failing pen and paper, but digital learning is bit fast and to flow with it, making physical notes won't help you that much. Its time were taking notes digitally takes place.

AIO3 is designed with flexibility and ease of use in mind, taking notes is now at it simplest with this application. Using this android application the students, teachers, researchers, supervisors and others can communicate with each other and would be able to access a variety of some technical and reference books notes at no cost. The primary aim of this project is to help students, researchers, supervisors and others by allowing them to take, manage and share notes, amongst other pieces of functionality that aim to make the note-taking process automated and easier for users to manage all aspects of note-taking.

Keywords*:*

Android, Notes Management, Study material, Academic, Education Technology supporting learning, Digital note-taking, Application in note-taking, Evaluation of CAL (Computer Assisted Learning) systems, etc.

ACKNOWLEDGEMENT

We take this opportunity to express our profound gratitude and indebtedness to our project guides **Prof. Shraddha Kokate** for giving us the opportunity to accomplish this project.

We are very much thankful to Prof. Shraddha Kokate for being very much resourceful, kind and helpful. Her positive attitude, unassailable optimism and unawaring faith in us assured that we came out of problems whenever we encountered difficulties.

Finally we wish to thank all my friends and IT department to directly or indirectly helped us in the completion of this project. Last but not the least we would thank our family without whose support, motivation and encouragement this would not have been possible.

DECLARATION

We hereby declare that the project entitled, "**ALL IN ONE**" done at Malad, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of our knowledge other than us, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

GOVIND RAMA PARAB

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

The motive of this chapter is to provide the background details about the place note taking holds in the life of students, teachers, researchers, supervisors and others, the old time-consuming traditional methodologies of taking notes and the other problems associated with these approaches. After covering problems, the chapter gives a brief overview of the project, describing its motivations, key features purpose, scope and applicability.

1.1 Background

Today in our digital world, when almost every students, teachers, researchers, supervisors and others has a phone, tablet or laptop, but found that it's still hard to beat pen and paper. Whether it's online or offline learning, note-taking and sharing has been always important process in studying or remembering the important points. Almost everyone does it, and it is a practical requirement to remember the important points, things, etc. For students with the large amounts of information presented in each course, note-taking helps in encoding the information and thus makes it easier to remember. It also produces study materials to refer to later for exams and projects. Looking at Educational System in India or in other countries it's clear that it has become much more advanced now, than it was even three years ago. In this generation writing by hand might seem outdated to some students, but taking notes "the old-fashioned way" is a two-part action that can be beneficial. The first part of this important action is taking the notes and the second is reviewing and storing them later. Both of these processes help students to understand and retain information. But unfortunately, maintaining hand-written notes is tedious task because they aren't always accessible, sharable, editable, illustrative or easy to refer during collaboration or at last moments. To overcome this drawback that's where the digital note taking comes in.

Taking notes digitally has many benefits, as typing is a fast and easy way to take the information presented in lectures, meetings or in textbooks, and consolidate them for reference later. With this application students can help each other by exchanging or sharing the notes in a simple and

efficient manner. For example, students who do well in chemistry can pass on their chemistry notes, and in return, get notes from a student who excels in a subject that they're struggling with. This allows students to easily refer to their peers for explanations and questions, rather than going right to the teacher. In this busy schedule even today many of us have to search a lot of books, websites and visit different libraries for research purpose, or for a project. This takes a lot of time, money and sometimes handling the large volatile data becomes a tedious task, also there is risk of losing those notes. Even after gathering data, it's difficult to manage it in proper way, sometime we end up by just creating redundant data of same. When someone is out-door and need to memorize something with text or drawing immediately, he or she must find a piece of paper and a pen to capture it, but what if he or she didn't have things available at that moment to capture it by writing or drawing. So such problem can be solved from now on, as you can easily do that with this app. You just launch this application; this application has attractive, stylish and an easy-to-use editor screen with automatic functionality of saving notes that allows you to capture and organize your ideas. AIO3 - Text Editor is the simple application to open, edit, delete, rename, and save. Making note of anything and carrying it anywhere you want. AIO3 notes taking is not only a digital notebook for your to-do lists, lecture and meeting notes, or anything you want to organize or remember, but also its a good reminder app, social app and helpful app. It gives you a quick and simple notepad editing experience when you write notes, memo, email, message, shopping list, and to-do list. It makes to take a note easier than any other notepad and memo application.

1.1.1 Existing Solutions

The below section describes few android application also there are more other such applications available on play store but, particularly I have chosen three applications which are similar in functioning to my application, and found that every app has some drawbacks in their features which are discussed ahead:

1.1.1.1 Evernote:

This app allows users to create notes, which can be text, drawings, photographs, audio, or saved web content. Notes are stored in notebooks and can be tagged, annotated, edited, searched, given

attachments, and exported. Evernote is cross-platform, for Android, iOS, macOS, and Microsoft Windows.

The below points are few drawbacks of Evernote, this drawbacks are referenced from a website whose link is also given below. To read all drawbacks in detail visit this website:-

a) Evernote Free Is Severely Limited.
b) Evernote Premium Is Expensive.
c) Evernote Still Lacks Some Premium Features.
d) Evernote Apps Are Frustrating to Use.
e) Free Evernote Alternatives Are Plentiful.
f) It's also not possible to lock individual notes.

Website/Reference link:-

https://www.makeuseof.com/tag/ditch-evernote/

1.1.1.2 Google Docs:

Google Docs is an online word processor that lets you create and format documents and work with other people. Google Docs has the underlying infrastructure to automatically save all work in real time, providing an extensive history through which rollback to earlier versions of a document is made possible.

The below points are few drawbacks of Evernote, this drawbacks are referenced from a website whose link is also given below. To read all drawbacks in detail visit this website:-

a) Weak Presentation Program.
b) Internet Access Required.

Website/Reference link:-

https://trackvia.com/blog/excelling-at-work/google-docs-a-good-fit-for-your-business/

1.1.1.3 Google Keep:

Google Keep can access and mine your data. No text formatting is available at Keep Notes. The app is limited to just multiple tags for notes. The users are not able to undo their changes. No notifications are sent when tasks are added to shared lists. Google Keep doesn't integrate with Google Calendar or Tasks.

a. Google Keep can access and mine your data.
b. No text formatting is available at keep notes.
c. The app is limited to just multiple tags for note.
d. The user are not able to undo their changes.
e. No notifications are sent when tasks are added to shared lists.
f. Google Keep doesn't integrate with Google Calendar or Tasks.

Website/Reference link:-

https://www.mobileappdaily.com/2018/07/2/google-keep-vs-evernote

1.2 Objectives

The following are some of the major objectives of AIO3 android application:

- To make the whole note-taking procedure digital. As taking digital notes is bit faster than taking notes physically on paper, so valuable time is saved of users.
- To reduce money required and manual work. Since the system is managed digitally the cost of buying books and time required for doing manual work is reduced.
- It saves the paper thus saves environment, as all the data is digitally stored hence lot of paper is saved.
- It saves lot of physical storage as digital data will not consume any kind of physical space.
- To reduce risk of losing data, money problems and time consumed while sharing the notes physically, as digital data can be shared from anywhere and at anytime to anyone within seconds hence no need to make physical copies of data.
- It allows user to login and saves their notes dynamically.

- It allows you to use all the features available that for free of cost and with no advertisement in application.
- It is lightweight, and handy which allows user to take control of their notes and gives them the best note-taking experience with multiple features.
- It allows user to create notes in different formats like plain text, text with images, checklist, etc.
- Users of this application are also allowed to prioritize their tasks with priority levels.
- This application will also work without getting the registrations details or personal data of the user.
- Users will be allowed to capture the essence and share it with the other members.
- It allows users to create their own Text PDF.
- It allows easy search between all your notes anytime and at anywhere you want, as searching information in physical files or books is a hectic job.

1.3 Purpose and Scope

1.3.1 Purpose

The purpose of this project is to create an application that is suitable for every students, teachers, researchers, supervisors and others from both ends; those leading the meetings, lectures, etc to provide information and also those who attend lectures, meetings, visits library, etc for gathering the information. This project aims to create an android based application that provides seamless entry and organisation of notes, a rapid action point entry system, and allows user to take control of their notes by providing them different helpful features which will give the user best note-taking experience.

1.3.2 Scope

Methodology:

The scope of this project particularly aims towards the development of android application which aims to develop the potential among the faculty, students, teachers, supervisors, researchers, etc

to make them actively involved in developing and creating their notes digitally to reduce the manual work and time required to them when doing it on paper. The main advantage of taking notes digitally is that notes can be shared between any persons easily. These notes can serve as reminders to complete assignments or as a pedagogical tool to elaborate on in-class lectures, create discussion questions, and more. The applications with the greatest potential are those that allow for real-time collaboration. In this way, a group of students ranging from as small as two to as large as an entire class can participate in viewing, updating, and/or creating a document, simultaneously. Since note-taking is the process of capturing information from a source, event or writing what we think as when. This usually takes the form of recording, writing, jotting, paraphrasing, sketching, labelling, outlining, and annotating. This project enable users to type, write, draw, edit, set reminders, and also allows user to utilize all the features of this project with free of cost and free of adds on their devices just as they would do on paper while taking notes. Our reminder agenda planner tool allows you to keep track of your duties, create daily sticky ideas and shopping list for items or ideas with unprecedented simplicity, notability and unrivaled time-saving value. It allows you to manage your schedule with proper and good notes like a daily planner. No complicated setup steps are needed, just tap the screen and type in what you came for and create and keep notes, quick lists, checklist or backup for any idea or keep notes as daily planner. With your simple personal notebook you can remember anything fast, shopping list for groceries, to-do list for your daily agenda and easier note-taking to make setting up meetings. Project will also provide some of its own unique features, storage, organization, and sharing capabilities, and is made by keeping all things in mind by making it cost efficient and also allowing user to do less physical work.

Limitations:

- The user can access this application through their android devices only.
- App will not run on android version below 5.0 i.e. Lollipop.
- Some of app features won't function without an active internet connection such as creating account, logging into existing account, saving data online for making it accessible from any android device, etc.

Assumption:

- While sharing the notes, both senders and receivers should have an internet available so that notes can be transferred or collaborate easily.
- To use this application it is strictly recommended that the person must have android device with android version 5.0 or greater.
- This application is designed by keeping the flexibility to use its different features i.e. the users of these application can tailor the features for their own personal uses and employ it in various life roles (as a student, employee, researchers, faculty, supervisors, or at home).
- This application should be used by anyone in their day-to-day life, no matter in which field he belongs to.

Future Scope:

Using this app the students, teachers, researchers, supervisors and others would be able to access a variety of technical and reference books at no cost. User would be able to add different departments also, so that they can easily create a group of department workers and can share their documents with them by providing various accessibility features to the group persons. The feature of coaching videos and tutorials can also be added by this a basic classroom structure with attendance taking, assignments submissions space, events notifications, exam notifications, holiday notifications, etc features can be added here. Through this app the professor, researchers, supervisors and student or anyone would be able to communicate and have group discussion which may lead to innovative ideas and projects. Exams papers, research papers, project documentations, etc from various journals can also be included in the groups. Social groups can be created so that daily updates regarding jobs, interviews, exams, news, etc can be included. Making this application more secure, adding word dictionary, spell corrector or adding search engine can be one of the best features that can be implemented in future. Addition of basic functionalities like chatting system, mail system, video calls, audio calls, E-library system, etc and many more features can be added in future to make this app more feasible and easy to use. As per present scope this is only for android users, but can be made available on different platform with more features.

1.4 Applicability

Managing notes is a very important process since it requires lot of manual work and well maintenance. In this fast paced life to note down the every important things and to remember them on time we humans tends to forget them easily, and to remember these things we need to note down things in a piece of paper, but this doesn't help to us the way we want. To avoid happening this some people hire personal assistant who acts as a reminder to remind them about the important work that needs to be done. This seems good but you have to pay money for hiring someone just to remind you, your own task. For such peoples AIO3 android application can be helpful to remind them, to do such important things. Even though there are many solutions that exist in the current android market that try to address the aforementioned problems, but found that all of them possessed some drawbacks and act as more general purpose applications that target a broader market, trying to encapsulate as many customers as possible, rather than solving the problems of customers of each field. AIO3 application is built to reduce these problems faced by users while creating, managing, editing, sharing, collaborating, etc the notes. AIO3 has wide variety functions that will help every other person somehow. Digitizing notes also makes it easier to index information based on subject, lesson, quarter, etc. In many cases students, teachers, researchers, supervisors and others will also be able to add illustrations and diagrams, and find examples online to put ideas into context. AIO3 application can be used for various functions as you can add your to-do list in this application, some important notes for future reference, etc. The application is very useful in some cases like when you want quick access to the notes. Basically the main purpose of this application is to save time, space and to reduce manual work up to 90%.

1.5 Organization of Report

The subsequent chapters of the project report will focus on the survey of different technologies which includes comparison of different platforms, languages, front- end, back-end, hardware etc. the survey is followed by the analysis of requirements resulting in generation of requirement specification and schedule of activities. It also includes the conceptual design that visualizes the features and operations that can be performed on the system. The final chapter of the project

report includes the system design that describes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudo code and other documentation.

Chapter 2

Survey of Technologies

The use of technology in the day to day life has come up with the drastic change, nowadays there are various technologies and software's available which we can use to develop a system right from designing to development and building components, web services and API technology has made it very easy. Android is a mobile operating system which was released in September 2008. The operating system gets updated every month with new version coming into existence. The first version was Android Alpha while the latest version is Android 11. Android apps are mostly written in Java, C++, Kolin.

In this chapter of Survey of Technologies, I will be elaborating my awareness and understanding of few available technologies related to the topic of my project. Here I will be discussing the details of all technologies that are necessary to complete my project and also I will be discussing some technologies that can be used to do the same. I will be also showing the comparative study of the technologies available in the chosen area of my project with some of other available technologies.

2.1 Software

Eclipse

Eclipse is a powerful, open source, integrated development environment (IDE) that facilitates the creation of desktop, mobile, and web applications. Eclipse is a highly versatile and adaptable tool. Many types of applications and programming languages can be used by adding different "plug-ins". It is mostly written in Java; thus, it's primarily used for Java. It provides tools like project manager, code window, console window, compiler notification, activity log and other important tools to develop any application. That is one of the reasons that Eclipse is well-known for Java IDE. It is simple, portable and has a short build time. It is preferable for the development of small applications. The IDE launched Eclipse is faster. It primarily supports java but can also works for different languages like C, C++, C#, PHP, Perl, and Ruby.

Android Studio

Android Studio is an IDE for Android Operating Systems. An IDE stands for an Integrated Development Environment, it is a tool to make app development easy. It is the official IDE of Google for Android development. Android Studio has various features like auto code completion and refactoring to make the project development hassle-free. Its release date is 16 May 2013. Android Studio uses IntelliJ IDEA. It is specifically for the development of Android applications. It is freely available and open source.

Comparative study between Android Studio and Eclispe:

Android Studio	Eclipse
1. Android Studio is the Official IDE for Google's Android operating system.	1. Eclipse is an IDE widely for java based Android app development.
2. Android Studio supports Android.	2. It supports Android through Google ADT extension.
3. In Android Studio, there is an inbuilt Gradle build tool.	3. Eclipse has a build tool by default that is ANT. Though it is outdated when we see Gradle of Android Studio.
4. It gets regular updates for android development.	4. It does not have regular updates for android development.
5. The developer of Android Studio is Google.	5. The developer of Eclipse was Eclipse Foundation.
6. The use of Android Studio is particularly for the development of android applications.	6. The reason for its development was a variety of applications.
7. When it comes to user interface Android Studio is much easier and quicker.	7. Eclipse is not a native Android IDE and thus, is more complicated than Android Eclipse.
8. Though both support equally high-end auto code completion, Android has an Upper hand on it.	8. Though both support equally high-end auto code completion, Android has an Upper hand on it.

9. In Android, the easy drag and drop feature for easy layout design is available for the developers.	9. Alas, Eclipse has no such thing as the feature of drag and drop.
10. Android is quicker than that Eclipse and developers love the tools and the resources of Android Studio.	10. On the other side, Eclipse loses to Android in the case of Speed as well.
11. Android is lighter than Eclipse. It also often crashes, even though it is more stable than Eclipse.	11. When compared with Android, eclipse is way larger than IDE and requires high RAM space and clearly more CPU speed.

Table 2.1

So, going through the above comparison can easily make it clear before you that Android Studio stands stronger than Eclipse. The reasons are simple: it is more flexible than Eclipse. It has build variants and multiple apk file generation. Android Studio also has a rich layout editor with supports like drag and drops layout editing. It also has proGuard and application signing capabilities and many other capabilities.

Android Studio also supports Git as a version control system to maintain the app changes and push them into github. All java files, layout files (for design) are integrated into a single project easily. After the completion of project, the whole application could be put as an .APK (Android Package) file, in which we can run that APK file in any device and use the application.

Other main tools of Android Studio include Android SDK, ADB and Gradle Build.

Android Software Development Kit (SDK):

One of the main tools used in developing android applications, as it packages many core features into one SDK and it can be used in the application easily. This helps us to avoid writing lot of code, and building applications faster.

Android Debug Bridge (ADB):

Android SDK uses ADB tool as a connection device which allows us to connect the Android Devices or Emulator with the machine via USB. After developing or while developing

applications, we can connect with the device to check how the application runs. Later, we can debug and run the applications.

Gradle Build:

Gradle Scripts are the recent feature that is added to Android Studio. It is basically an automated build system which is used to automate the various phases involved in designing an application that includes design, development, test, debug, and publish. We need to configure the project and modules by mentioning all the supported jar files, SDK's, version name, level, compiled SDK version, build tools version. to ensure that the developed app is compatible with the testing device/emulator. Gradle is also similar to Ant and Maven which helps in maintaining java projects (repositories).

Android Device Monitor:

If we want to access all the hidden files that are generated when we run the application, we can use the monitor. We can select any project and explore the files that are related to that project. But, as they are hidden files, we need root permissions to access them. Suppose, if we run the app in device, we need to root the device and run commands in adb shell to get permissions.

SDK Manager:

It is one of the main tools to maintain the updates of all the installed components required to run the project. It also notifies us when the project is not compatible with device or any other compatibility issues and to download any component that is required.

AVD Manager:

It is used to create virtual devices of any desired API level to support higher level SDK's incase our device does not support. Using emulators to test the application is difficult as it might be little slower when compared to real device.

2.2 Front End Technologies

XML

XML stands for Extensible Markup Language. XML is a markup language much like HTML used to describe data. It is derived from Standard Generalized Markup Language(SMGL). Basically, the XML tags are not predefined in XML. We need to implement and define the tags in XML. XML tags define the data and used to store and organize data. It's easily scalable and simple to develop. In Android, the XML is used to implement UI-related data, and it's a lightweight markup language that doesn't make layout heavy. XML only contains tags, while implementing they need to be just invoked.

Jetpack Compose

Jetpack compose is a modern Android UI toolkit introduced by Google. It simplifies the app development process and speeds it up. With Jetpack Compose, you can write less code compared to the current view building approach – which also means less potential bugs. There is one more great thing about it – it uses Kotlin. If you have been using React or Flutter, you will probably be familiar with the concept and find many similarities. Jetpack Compose is built upon a different programming paradigm. It uses a declarative paradigm.

Gradle

Gradle is an open source build automation tool that is based on the concept of Apache Maven and Apache Ant. It is capable of building almost any type of software. It is designed for the multi-project build, which can be quite large. It introduces a Java and Groovy-based DSL(Domain Specific Language) instead of XML (Extensible Markup Language) for declaring the project configuration. Gradle tools support a wide variety of IDE's, which provide a better user experience, as different people prefer working on a different IDE. It provides the users that like to work on the terminal with the command-line interface, which offers features like Gradle tasks, Command line completion, etc. It has been developed for building automation on many languages and platforms, including Java, Scala, Android, C / C ++, and Groovy. Gradle provides integration with several development tools and servers, including Eclipse, IntelliJ, Jenkins, and

Android Studio. Gradle has taken the advantages of both Ant and Maven and remove the drawbacks of both.

Comparative study between XML and Jetpack Compose:

XML	Jetpack Compose
eXtensible Markup Language, or XML: A markup language created as a standard way to encode data in internet-based applications. Android applications use XML to create layout files. Unlike HTML, XML is case-sensitive, requires each tag be closed, and preserves whitespace.	Jetpack compose is a modern Android UI toolkit introduced by Google. It simplifies the app development process and speeds it up. With Jetpack Compose, you can write less code compared to the current view building approach – which also means less potential bugs.
In Android we use XML for designing our layouts because XML is lightweight language so it doesn't make our layout heavy.	Jetpack Compose is Android's modern toolkit for building native UI.
It simplifies and accelerates UI development on Android by only using Java and Kotlin.	It simplifies and accelerates UI development on Android by only using Kotlin.
XML is completely compatible with Java and 100% portable. Any application that can process XML can use your information, regardless of platform.	Jetpack Compose is completely compatible with Kotlin and it quickly bring your app to life with less code, powerful tools, and intuitive Kotlin APIs.

Table 2.2

Here in my project I am using XML because of following points:

1. The stable version of Jetpack Compose i.e 1.0 is recently released in July 2021, so its completely new for us as compared XML.

2. Jetpack Compose don't has huge community and also resources are limited on developers site.

3. Jetpack Compose for UI development uses only Kotlin language whereas, XML uses both Java and Kotlin for the same.

4. Jetpack Compose holds the promise for increased productivity, but new technologies can have unknown limitations and internal bugs that may need temporary workarounds.

2.3 Back End Technologies

SQLite

SQLite is a Structure query base database, open source, light weight, no network access and standalone database. It support embedded relational database features. SQLite Database Operations Whenever an application needs to store large amount of data then using sqlite is more preferable than other repository system like Shared Preferences or saving data in files. Android has built in SQLite database implementation. It is available locally over the device(mobile & tablet) and contain data in text format. It carry light weight data and suitable with many languages. So, it doesn't requires any administration or setup procedure of the database. SQLite can perform so many operations on the data such as adding new data, updating, reading, and deleting this data. Android SQLite essentially focuses on the core concepts behind building database-driven applications.

FIREBASE

Firebase is a mobile app development platform which offers a huge variety of tools and services to its users to make app development easier. Firebase manages the backend aspect of the applications, giving users more time to work on interface and other features of the application. Firebase can be easily used with a single Software Development Kit (SDK), which is embedded right into Android Studio. Hence, users do not need to create their own server-side program using PHP and MySQL or similar technologies. Firebase is a 'Backend as a Service' (BaaS). This indicates that a user can freely make their own dedicated application. Firebase is compatible with Android, iOS as well as web applications, free of cost. Currently, Google manages all of Firebase's operation.

MySQL

For developing the web applications as well as business applications, there is need to store the data and for this purpose most of the experts prefer the open source database applications which

are also applicable while developing the mobile app. Therefore, this facility of the open source database has been provided by MySQL. MySQL is a fast, easy-to-use RDBMS which is used for storing data generated through the small applications and big business applications. MySQL is developed, and supported by MySQL AB, and it is popular because of following reasons: MySQL is an open-source license so user has nothing to pay to use it. It handles large databases, so it will possible to organize 50 million rows or more in a table with default file size limit for a table is 4GB. MySQL handles a large subset of the functionality of the most expensive and powerful database packages. MySQL uses a standard form SQL data language. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.

Comparative study between SQLite and ROOM:

SQLite	MySQL
SQLite is an open source project available in the public domain.	MySQL is an open source project which is owned by Oracle.
SQLite is a server-less database and is self-contained. This is also referred to as an embedded database which means the DB engine runs as a part of the app.	On the other hand, MySQL requires a server to run. MySQL will require a client and server architecture to interact over a network.
The SQLite library is about 250 KB in size.	MySQL server is about 600 MB.
The SQLite directly stores info in a single file, making it easy to copy. No configurations are required, and the process ca be done using minimal support.	Before copying or exporting MySQL you need to condense it into a single file. For larger databases, this will be a time-consuming activity.
SQLite is file-based and easy to set up and use.	MySQL requires some technical expertise to setup.
SQLite uses standard SQL syntax with minor alterations and is easy to use.	MySQL uses slightly different syntax as compared to conventional SQL.

Table 2.3.1

Comparative study between Firebase and MySQL:

Firebase	MySQL
Firebase is Non- Relational Database.	MySQL is Relational Database.
Firebase is based on the concept of collections and documents.	MySQL is based on the concept of tables.
Firebase is owned by Google.	MySQL is an open source.
Firebase is very flexible in terms of design as any key/field values can be added easily without affecting the existing design.	MySQL is not so flexible design wise, new column insertion may affect the design.
Firebase uses NoSQL.	MySQL uses SQL.
Firebase is horizontally scalable.	MySQL is vertically scalable.
Firebase has dynamic schemas to facilitate unstructured data.	MySQL has predefined schemas.
Firebase can best option when you need to support dynamic queries.	MySQL can be best option when you need to scale based on changing requirements.
Firebase is more suitable for the hierarchical data store as it supports key-value pair method.	MySQL is not suitable for hierarchical data storage.

Table 2.3.2

Here in my project I am using Firebase database and SQLite database for implementing different features in my application. Since Firebase is well integrated into android studio and it provides easy way implement its different methods for the user which allows easy ways to handle the user data. Also, SQLite database is embedded in android by default. So, there is no need to perform any database setup or administration task.

2.4 Languages

Java

Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems Java platform. Java is a high-level, class-based, object-oriented programming language that is

designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let application developers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. It is a computing platform for application development. Java is fast, secure, and reliable, therefore. It is widely used for developing Java applications in laptops, data centers, game consoles, scientific supercomputers, cell phone. Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.). It is one of the most popular programming language in the world. It is easy to learn and simple to use. It is open-source and free. It is secure, fast and powerful. It has a huge community support (tens of millions of developers). Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs. As Java is close to C++ and C#, it makes it easy for beginner programmers to switch to Java or vice versa. Android code is written once and to execute need to compile and optimise native code for better performance on various devices. Java has platform independent feature so it is used for android development. Also large java developer base enables to develop a lot of android apps fast so it is based on java. As Java is the general purpose computer programming language that is classed based and Object Oriented it is one of the most popular language for mobile application development.

Some of the important features of Java are:

- Simple and Familiar
- Compiled and Interpreted
- Platform Independent
- Portable
- Architectural Neutral
- Object-Oriented
- Robust
- Secure
- Distributed
- Multi-threaded and Interactive
- High Performance
- Dynamic and Extensible

Kotlin

Kotlin is a programming language introduced by JetBrains in 2011, the official designer of the most intelligent Java IDE, named Intellij IDEA. Kotlin is an open-source statically typed programming language that runs on Java Virtual Machine (JVM). It combines Object Oriented Programming (OOPs) and functional programming in unrestricted, self-sufficient, and distinctive platforms. It also allows the twinning of functionalities by miniature codes. In 2017, Google announced Kotlin is an official language for android development. Kotlin is an open source programming language that combines object-oriented programming and functional features into a unique platform.

Some features of Kotlin are as follow:

- Efficiency to write code.
- Low Cost of Adoption
- Safe and Reliable.
- Smart Cast Function.
- Massive Interoperability.
- Extension Functions.
- Reducing Crashes at Runtime.

Comparative study between Java and Kotlin programming language:

- Java has the power of using third-party code to make writing codes easier. Java is fairly simple to handle and removing bugs from it becomes easier when compared to Kotlin.
- The standards of safety in Java are of higher quality than Kotlin.
- In terms of compilation speed Kotlin is much slower than that of Java.
- In many cases such as performing incremental builds, Kotlin is faster than Java, there is no doubt about it. However, Java remains a clear winner when it comes to creating clean builds for Android apps.
- Although, Kotlin's increased popularity, especially after Google announced it as the first-class Android app development programming language, has certainly increased the number of Kotlin app developers in the market. There are, however, still less number of

Kotlin developers available in the market compared with Java developers. So the community of Kotlin developers is still small as compared to Java, so it lacks learning materials and professional assistance.

Here in my project I am using Java Language because of the following points:

- Java is multiplatform and works practically on any device, server, or operating system.
- Java has strong security measures which make the chance of memory to get corrupt very low. Also, Java has a good coding base which makes it strong.
- Creating Modular Applications in Java is very easy and since it has features like reusability, it makes the code even stronger. Also, I am familiar with Java Language for almost 2 years.

Chapter 3

Requirements and Analysis

3.1 Problem Definition

3.1.1 Traditional / Handwritten Note-Taking System:-

The main problem of the traditional note taking system .i.e. creating handwritten notes, is that it requires a lot of manual work and all the notes needs to be saved properly and hence it takes a lot of storage space too. This traditional process of writing notes also requires some speed when you are actively taking notes in your work field or else it will risk you of falling behind and missing some things which were important to note down. Sometimes when you need to add more notes to an older section and don't have any room left on the page, then you may have to start another page further in your book which could cause confusion later down the track. So, traditional process of taking notes is not that much helpful to us as they are static in nature i.e. we can capture the notes either by writing or drawing only, and this takes lot of time. Also, these notes are difficult to refer if your handwriting is bad, difficult to edit when space is not available, sharing is not flexible and it itself become's risk as books and papers can be easily lost by someone while carrying them, they can't remind you your task at particular time when you want, searching in books or paper becomes tedious, if once lost they are unrecoverable, and many more problems are there.

Some of the key problems faced while taking notes by traditional methods are discussed below:

- **Can't Search Keywords Quickly -** Searching through your handwritten notes is a time consuming process of visually scanning and flipping through print outs, notebooks, sheets and pads of paper. There's no keyword searching capability.
- **Storage Space Problem -** The traditional note-taking system requires a lot of physical storage space to store the data in notebooks, paper, sheets, or in pads of paper. This data needs to be well maintained and to do so we need to maintain them either week wise,

month wise, year wise, or notebook wise which ultimately takes up lot of physical storage space.

- **Paper Wastage Problem -** Traditional note taking system is all about writing or drawing the notes on paper which has become a serious problem, as even for a single person to create his notes it requires bunch of papers and we all know how the paper production and consumption has affected the world environment.
- **Difficult To Share -** Since all the note in traditional note taking system are written over the papers, notebooks, or sheets it becomes difficult to share them. As while sharing these notes you have to share the whole physical copies of your notes and however, this kind of sharing is not flexible as at a time only one person can get the notes and also there is a risk of losing the notes while physically sharing them. Also, once after sharing the physical copies of your notes to someone it becomes problem for you to study those notes at that particular time, as you left with no copies of your own notes to study.
- **Easiest to lose and destroy -** Notes written over papers, books, sheets are easy to destroy by tearing a page or erasing too much, water damage, your dog ate it, etc. They can also be easy to lose and papers are often misplaced.
- **Difficult to correct mistakes and customize** - Erasing can be somewhat challenging if you have a lot to erase as some of the writing can remain on the page (just faded) or you might rip the whole page with the force of the eraser. Also, to use many colors or highlight, you must buy additional pens and highlighters. These items are almost never erasable when put on a page as well. These reasons make customization and corrections very difficult.
- **No backups** - Unless you scan it or copy the notes, there are no backups, and you have one copy that if destroyed is gone forever. Also, scanning and copying notes takes more of your valuable time that you could use to study the material. This is the riskiest option overall in terms of note safety.
- **Archiving can be difficult** - Instead of all your notes stored in one device, it is likely if they are paper that your notes are in several notebooks. Notebooks can accumulate greatly over time resulting in a bunch of old notes that you are not sure if will use again and takes up a lot of space. Also, since they are not digital, specific notes about a specific

concept can be very challenging to find since notes are not named, tagged, or able to be searched.

- **Difficult To Edit -** Notes need to be a living document i.e. it should constantly adapt the new improved changes with the additions to depth and quality. But in traditional note taking system it is difficult to reformat or add to handwritten notes without completely rewriting them. Also, due to non availability of space it is difficult to edit them.
- **Physical / Manual Work –** The traditional note taking system is time consuming and is totally based on the manual work right from creating, editing, sharing or storing the notes. This project aims to create an application which will reduce the manual work to about 90%. Since it is completely digital and hence physical involvement will be minimized.
- **Writing can be time consuming -** Writing takes longer than typing. Also, it requires some speed when actively taking notes in your work field or else you risk falling behind and missing some things which were important to note down. Writing fast can make your handwriting untidy which makes your notes presentation bad and it becomes difficult to refer in future.

As, the world is getting digital rapidly and the success of the digital age comes with a 'faster is better' mentality. So, there is no doubt that handwritten notes can be slow and cumbersome, especially when you have to take a lot of notes quickly.

3.1.2 Existing Solutions:-

Even though there are many solutions that exist in the current android market that try to address the aforementioned problems, but found that all of them possessed some drawbacks and act as more general purpose applications that target a broader market, which are trying to encapsulate as many customers as possible, rather than solving the problems of customers of different field. Also, it is clear from the research of some existing solutions which were discussed in chapter 1 i.e. Introduction, that there is a need for a productive note taking application that is tailored to the needs of every individual working in the different areas.

Some of the key problems faced while taking notes by using existing technology are discussed below:

- **Lack Of Text Editing Features –** Some of the existing applications doesn't provide any text editing features, only thing you can do over there is just write and this doesn't help you when you want to highlight, strike, underline, change the color of particular text, make the particular text bold, italic or any other text editing which you want to do on text. On the other side, the applications which provide text editing tools also lacks due to limitations in their text editing features. Also, some application doesn't specifically format the selected text rather they directly format the whole text which is not what user actually want.
- **Pricing And Advertisement –** Some of the existing applications don't allow you to use their all available features for free of cost and with no advertisement in application. Such application pricing is very high and after certain amount of storage you have to buy their plans in order do your work. Those applications which display advertisements may irritate the user as the advertisements waste user's valuable time and it may also distract them from their work. Advertisements in the applications also affected the working speed of application which is not what the user wants. And those applications which don't charge for using all their features or don't show any advertisement doesn't provides good features to user i.e. they provide only basic features to user.
- **Complex User Interface –** The user interface in existing application is not user intuitive. Also, some features are not user friendly and are tricky for user to understand them as they appear to be different in different devices. The updates in the applications affect the user interface of application as, it keep making changes to the layout and start up screen which is annoying for the users to work on that application.
- **No Toolbox Available –** Toolbox is a basic feature that most of the existing applications doesn't implement. And this is one of the major drawback that existing applications have. Users always demands for the extra features and these demands are unlimited but the existing applications possess limited features. Most of the existing applications are feature limited. Toolbox is a collection of tools where tools are nothing but extra app features which are stored at one place and can be helpful for anyone.
- **Continuous Internet Access Required –** One of the main problem that some of the existing applications possess is that they require continuous internet connection to create, store, collaborate, access, edit or transfer the data to the database. These applications

don’t save the user data offline to the user device so, without internet connection these application are not that much helpful for users.

- **Problems In Sharing Of Large Notes –** Many of the existing application has trouble in sending larger note files as their basic plans have limited functionality. Certain application uses third-party applications to share the user notes, so in such applications exporting data to other applications is not that easy. Not all the application but some of them posses size limitations while creating or sending them to other application.
- **Problems In Accessing Data –** Notes created by users need to be stored somewhere and most of the existing applications does this in either of two ways which are: storing the data in user device i.e. offline storing of data and another is storing the user data in server where the data is stored online. When the data is stored is offline it is mostly secured but is available only on the user device and these can be easily lost if user clear the data of that application from settings, also if user cell is lost or damaged. When the data is stored online it is less secure but can be accessible from any device which has that application. Some of the existing applications mostly doesn’t use both of these methods they use only one method either offline or online. And only limited application uses method of data synchronization.
- **No Universal / Personal Chat System –** Many of the existing application doesn’t support any universal or personal chat system. This can be one of the major limitation that existing system possess. These features need to be implemented as they can be helpful to users to collaborate with others.

Some of these drawbacks in the existing applications formed the starting point for my project which aimed to create an android based application that will fulfill the need of basic and necessary features which every notes application should possess.

3.2 Requirement Specification

The application should be developed in such a way that most of the problems in the current system will be solved. It should be compatible with minimum Android version 5.0.0 which is Lollipop. The requirements of the application should be minimum so that every user can easily use this application in their phones.

3.2.1 Non-Functional Requirement

Non-Functional Requirements are the constraints or the requirements imposed on the system. They represent the behavior and characteristics of the system and the constraints on the services provided by the system such as timing constraints etc. It describes how the system must work. It is defined as the quality attribute for the system. Some of these are scalability, capacity, volume, usability, security, reliability, portability, maintainability, reusability, flexibility, robustness, performance, etc. They specify the criteria that judge the operation of a system, rather than specific behavior of the system.

1. **Reliability:**

   The system shall be completely operational at least 99% of the time. Down time after a failure shall not exceed more than 30 minutes. The system should be available 24*7 to the user.

2. **Security:**

   The application shall provide password protected access to the notes that are to be viewed by specific users. The details of users having account must be saved in encrypted form in Firebase database. In one to one chatting feature the messages shared between the users should be encrypted to maintain the privacy.

3. **Robustness:**

   In case, if user's device crashes, then all the synchronized notes must be accessible to the user when he/she login to the application from another device. All the synchronized notes must be recoverable to the users when login from different android device of suitable version.

4. **Performance:**

   The application shall provide better component design to get better performance at even peak time. The system shall be able to support at least thousands of simultaneous users. The max time delay for downloading PDF, downloading photos, creating new PDF, editing images, creating new images, etc must not take more than 10 seconds. Application must be lightweight and it should not delay in performing important task such as storing, editing, updating, synchronizing, etc the notes of users. The chatting feature must send messages instantly.

3.2.2 Functional Requirement

Functional Requirements are the properties that must exist in the final system. They are the statements of the services that the system should provide on the basis of how the system should react to particular inputs and responds in a particular situations. It documents the operations and activities that a system must be able to perform.

- For user to communicate with all the known and unknowns who are using this application a universal chat system is made available. This will help the user to ask the question to all the application users and for users to communicate with a particular person a one to one chat app is made available. To use universal chat feature registration via valid Email-ID is required and to use one to one chat feature registration via valid phone number is required. When chatting one to one the user should be able to send the instant message to any contact on his/her application contact list. User should be notified when the message is successfully delivered to the recipient by displaying a tick sign next to the message send. All the communication requires internet connection and all the data of communication should be stored on Firebase database. Universal chat system can be always helpful for anyone who wants someone to help them in making their notes. Personal chat system will help you in your work to collaborate with those to whom you already know. These features can be helpful for users to work in group which will also generate best results for the one who has language problems while making notes.
- The user interface should be user-friendly and all the features must be easy to use. The extra features of application must be stored at one particular category called Toolbox. When creating a new account indirectly the user should be directed on registration page and verification of user Email-ID or phone number must be done instantly. Without verification no user account should be created, same applies to direct registration. Every page in the application should display description about the page itself for users to understand the application quickly. User interface should work smoothly there should not be any bugs in UI designing. The system shall also allow users to save the application state so for the next time when user returns to the application his time will be saved. All

the shortcuts to navigate to all the application features must be included in sliding menu of all application screens so that user can easily navigate to all app features.

- All the notes which are synchronized must be stored online in Firebase database. For this the user must have account and an internet connection, if user doesn't have an account then he/she must be thrown on registration page for creating a new account or for logging into existing account and if user don't have an internet connection then an error message must be displayed to him/her for the times he/she tries to sync the notes. All the notes which are not synchronized must be stored offline in SQLite database. At some cases when the user data state is need to be saved Shared Preference should be used.
- The users must be able to register for the application through their valid Email-ID to use the features like synchronization of notes, universal chatting, etc. and for using one to one chat feature the user must be able to register the application through their valid phone number along with their valid Email-ID too. On installing the application the user should be allowed to use basic features of application without any registration. The registration should be mandatory only when the user wants to use certain features like synchronization of notes, universal chatting, PDF uploading, etc. If user skips the registration part and still tries to use these features then a he should be thrown on the registration page for the times the user tries to use these features. For one to one chat feature the user valid phone number must be the unique identifier of his/her account.
- All the notifications and alarm based reminder notes should function properly with minimum delay of two or three seconds. All the images capture from the application should be stored in user photos gallery and all the PDF's created from the application should be stored in files separately.

3.3 Planning and Scheduling

Number of Persons: 1

Estimated Start Date: 19th July, 2021

Estimated End Date: 20th February, 2022

The project is planned by keeping in mind that there are resources available online which contributes to the application development process. Building a Android Application is a task which involves abundance of user requirements. So, Planning must be done to ensure on time delivery of the project.

Gantt Chart:- Gantt chart is one of the popular way to illustrate project schedule. A Gantt chart is a graphical representation of project that shows each activity ask as a horizontal bar whose length is proportional to its time for completion This project Gantt chart is shown below:

Task Name	Start	Finish	Duration
Selection of Project	19-07-2021	22-07-2021	3.0 d.
Requirement Gathering	23-07-2021	30-07-2021	7.0 d.
Requirement Analysis	01-08-2021	09-08-2021	8.0 d.
System Analysis	11-08-2021	25-08-2021	14.0 d.
Documentation	26-08-2021	25-09-2021	30.0 d.
Designing	01-10-2021	29-10-2021	28.0 d.
Coding/Implementation	29-10-2021	31-12-2021	63.0 d.
Testing	01-01-2022	26-01-2022	25.0 d.
Deployment	27-01-2022	20-02-2022	24.0 d.

Table 3.3.1

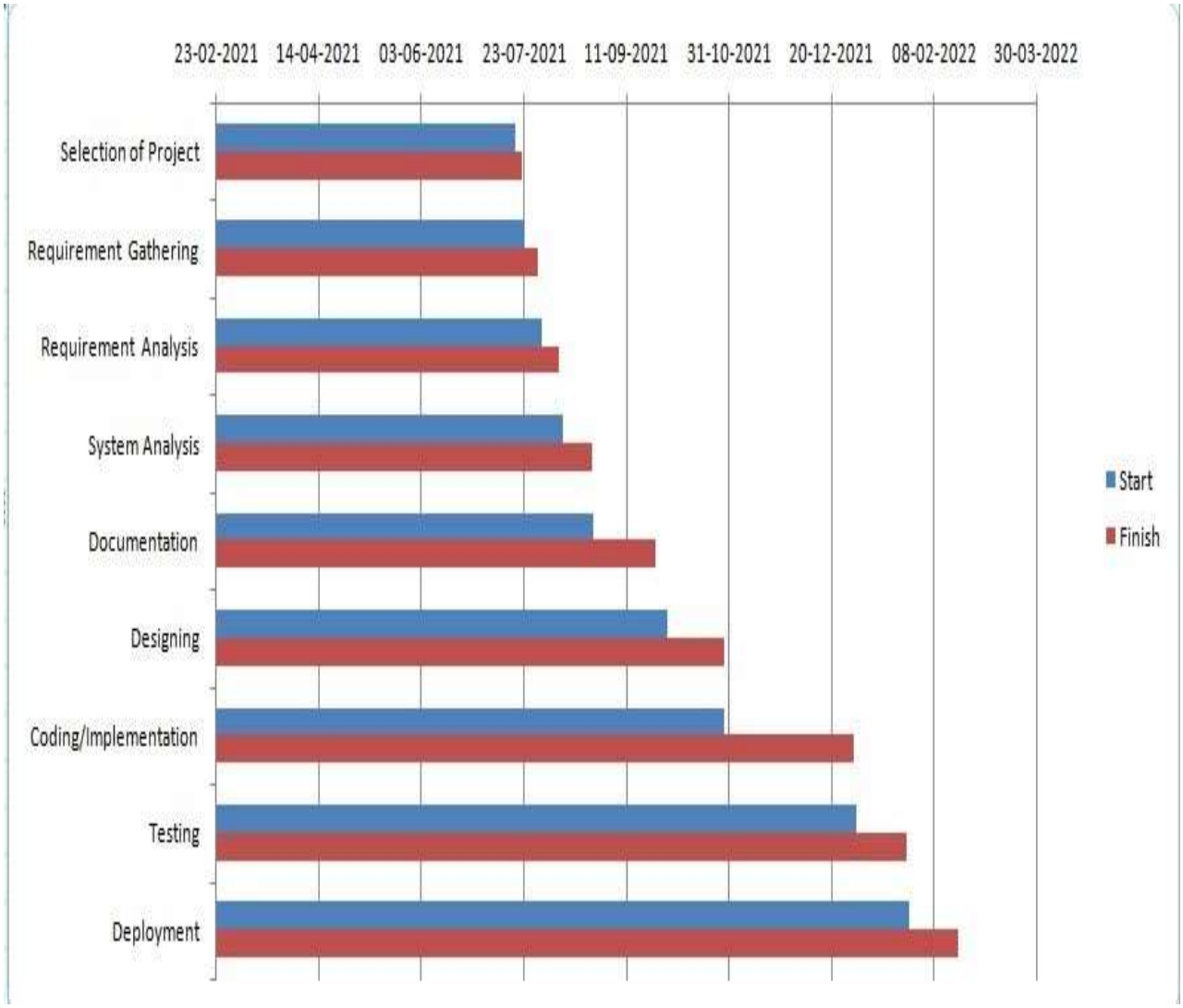

23-02-2021
14-04-2021
03-06-2021
23-07-2021
11-09-2021
31-10-2021
20-12-2021
08-02-2022
30-03-2022
Selection of Project
Requirement Gathering
Requirement Analysis
System Analysis
Documentation
Designing
Coding/Implementation
Testing
Deployment
Start
Finish

Figure 3.3.1

Pert Chart:- A PERT chart is a project management tool that provides a graphical representation of a project's timeline. The Program Evaluation Review Technique (PERT) breaks down the individual tasks of a project for analysis.

Planning the time dimensions and precedents for PERT chart:

Activity	Duration(days)	Precedents
A Selection of Project	3	
B Requirement Gathering	7	A
C Requirement Analysis	8	B
D System Analysis	14	C
E Documentation	30	A
F Designing	28	C,D,E
G Coding/Implementation	63	F
H Testing	25	F,G
I Deployment	24	H

Table 3.3.2

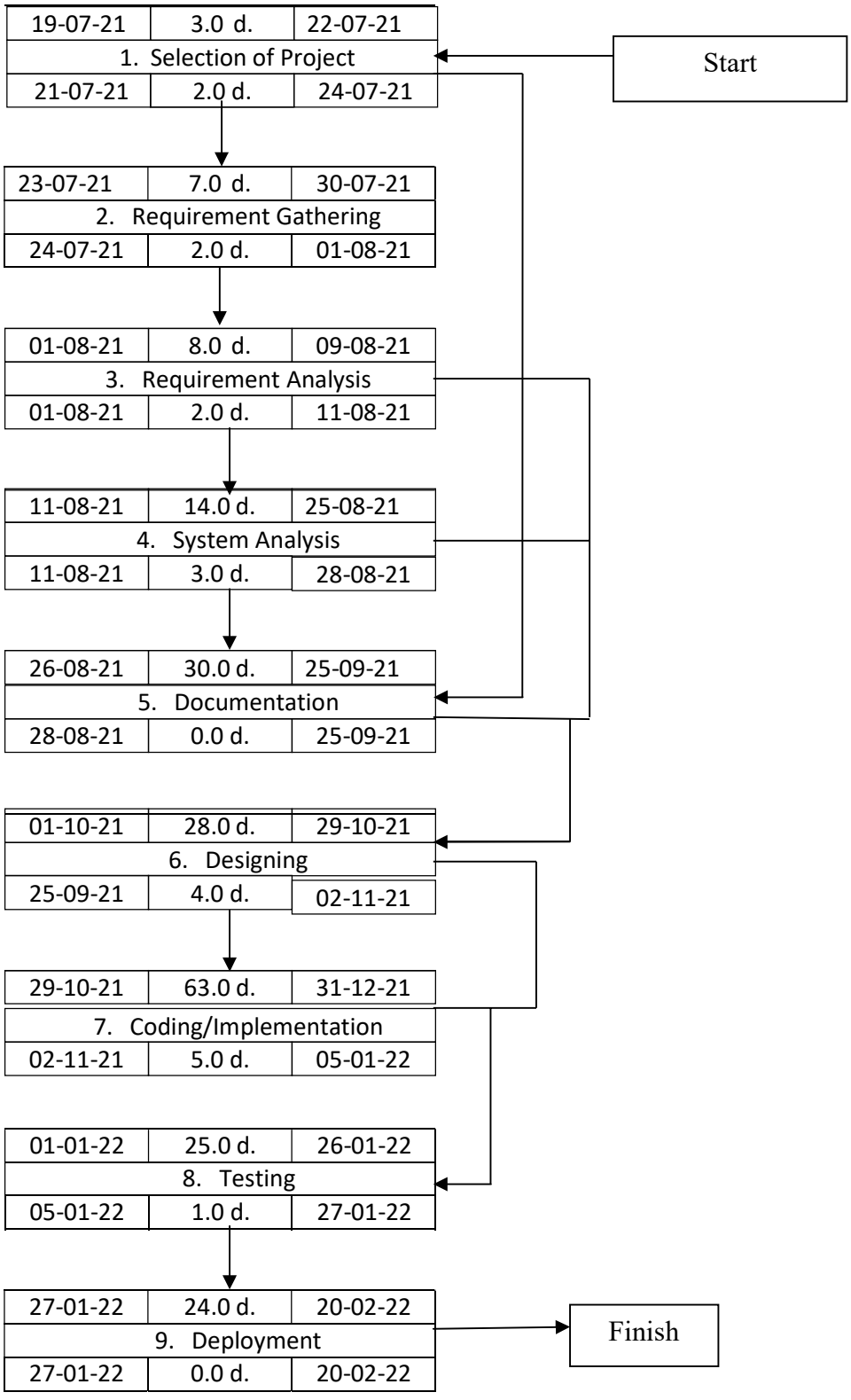

19-07-21
3.0 d.
22-07-21
1. Selection of Project
21-07-21
2.0 d.
24-07-21
Start
23-07-21
7.0 d.
30-07-21
2. Requirement Gathering
24-07-21
2.0 d.
01-08-21
01-08-21
8.0 d.
09-08-21
3. Requirement Analysis
01-08-21
2.0 d.
11-08-21
11-08-21
14.0 d.
25-08-21
4. System Analysis
11-08-21
3.0 d.
28-08-21
26-08-21
30.0 d.
25-09-21
5. Documentation
28-08-21
0.0 d.
25-09-21
01-10-21
28.0 d.
29-10-21
6. Designing
25-09-21
4.0 d.
02-11-21
29-10-21
63.0 d.
31-12-21
7. Coding/Implementation
02-11-21
5.0 d.
05-01-22
01-01-22
25.0 d.
26-01-22
8. Testing
05-01-22
1.0 d.
27-01-22
27-01-22
24.0 d.
20-02-22
9. Deployment
27-01-22
0.0 d.
20-02-22
Finish

Table 3.3.3

3.4 Software and Hardware Requirements

The minimum hardware and software requirements for the project will be as follow:

- **Hardware Requirements:**
  1) **Computer**

     Processor - Core i5 (Minimum)

     RAM -4GB (Minimum)

     Operating System -Windows 10
  2) **Network Connection**

     In some cases like creating, deleting, updating, logging in, logging out of account or for data synchronization the internet access is required to use the system.
  3) **Android Phone**

     An android device is needed with minimum version of 5.0.0 – Lollipop. However, there are multiple emulators you can use for development, but it's always better to test on real device too.
- **Software Requirements:**
  1) **Android Studio**

     This is the official integrated development environment (IDE) for developing android apps.
  2) **Chrome Browser**

     Google Chrome is a cross-platform web browser developed by Google.
  3) **SQLite**

     An open-source relational database i.e. used to perform database operations on android devices such as storing, manipulating or retrieving persistent data from the database. It is embedded in android by default. So, there is no need to perform any database setup or administration task.
  4) **Firebase**

     Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps.

5) **EdrawMax**

EdrawMax is a 2D business technical diagramming software which help create flowcharts, organizational charts, network diagrams, workflow diagrams, business charts, and engineering diagram etc.

3.5 Conceptual Models

The Conceptual Models includes Diagrams such as:

1. Data Flow Diagram
2. Use Case Diagram
3. Activity Diagram
4. Sequence Diagram
5. Class Diagram
6. Event Table

1. Data Flow Diagram:-

Data Flow Diagram is a graphical way of representing a flow of data within a system. The Data Flow Diagram also provides the information about the output and the input of each entity and also about the process itself. It tells that how the data flows within the system, how the input data that originates from the external entities are processed to give the desired output. A Data Flow Diagram has no control flow, there are no decision rules and no loops. DFD is also called as a data flow graph or bubble chart.

DFD Levels:-

DFD levels are numbered 0, 1 or 2 and occasionally go to even level 3 or beyond.

DFD Level 0:

It is also called as context diagram. It's a basic overview of the whole system.

DFD Level 1:

It provides a more detailed breakout of pieces of the context level diagram. It discuss the sub process of context diagram.

DFD Level 2:

DFD Level 2 then goes one step deeper into parts of Level 1. It may require more text to reach the necessary level of detail about the system's functioning.

Progression to Levels 3, 4 and beyond is possible, but going beyond Level 3 is uncommon. Doing so can create complexity that makes it difficult to communicate, compare or model effectively.

Level 0 Data Flow Diagram / Context Diagram:-

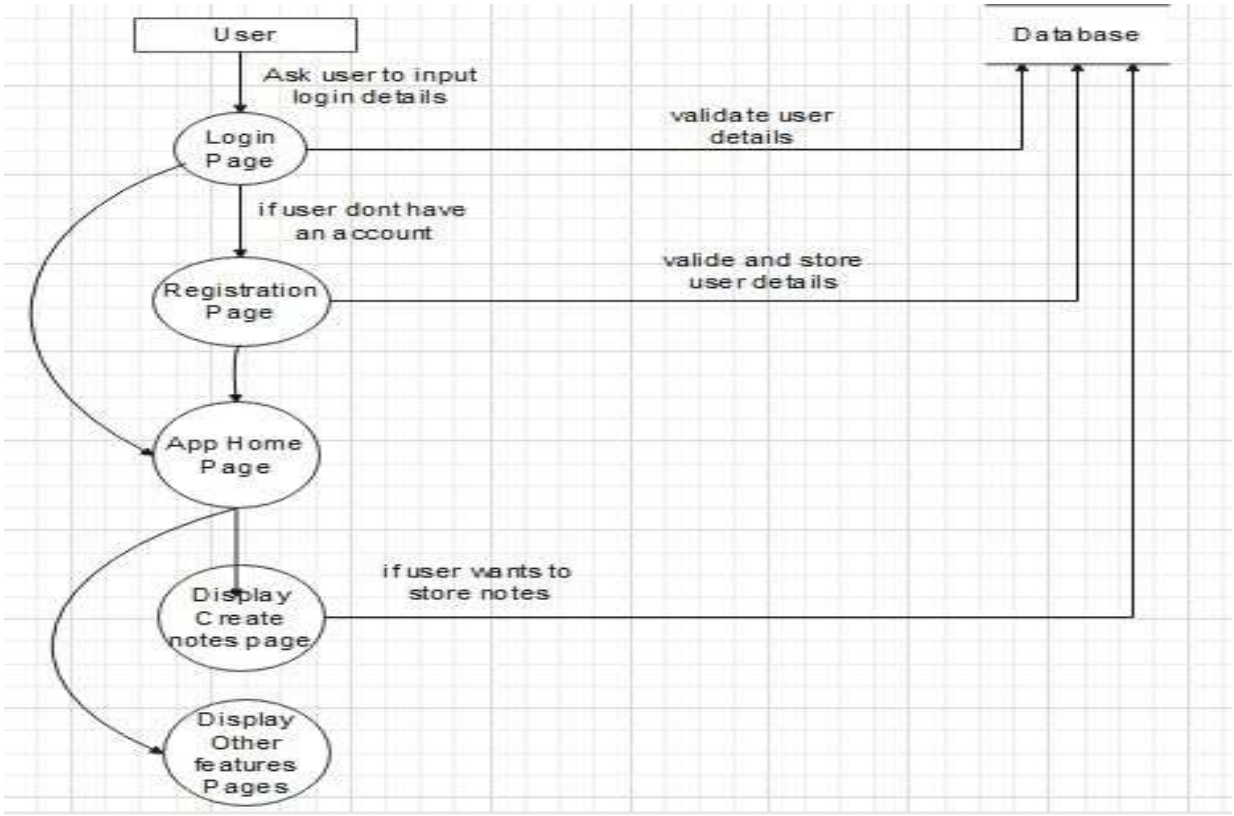


Figure 3.5.1.1

Level 1 Data Flow Diagram:-

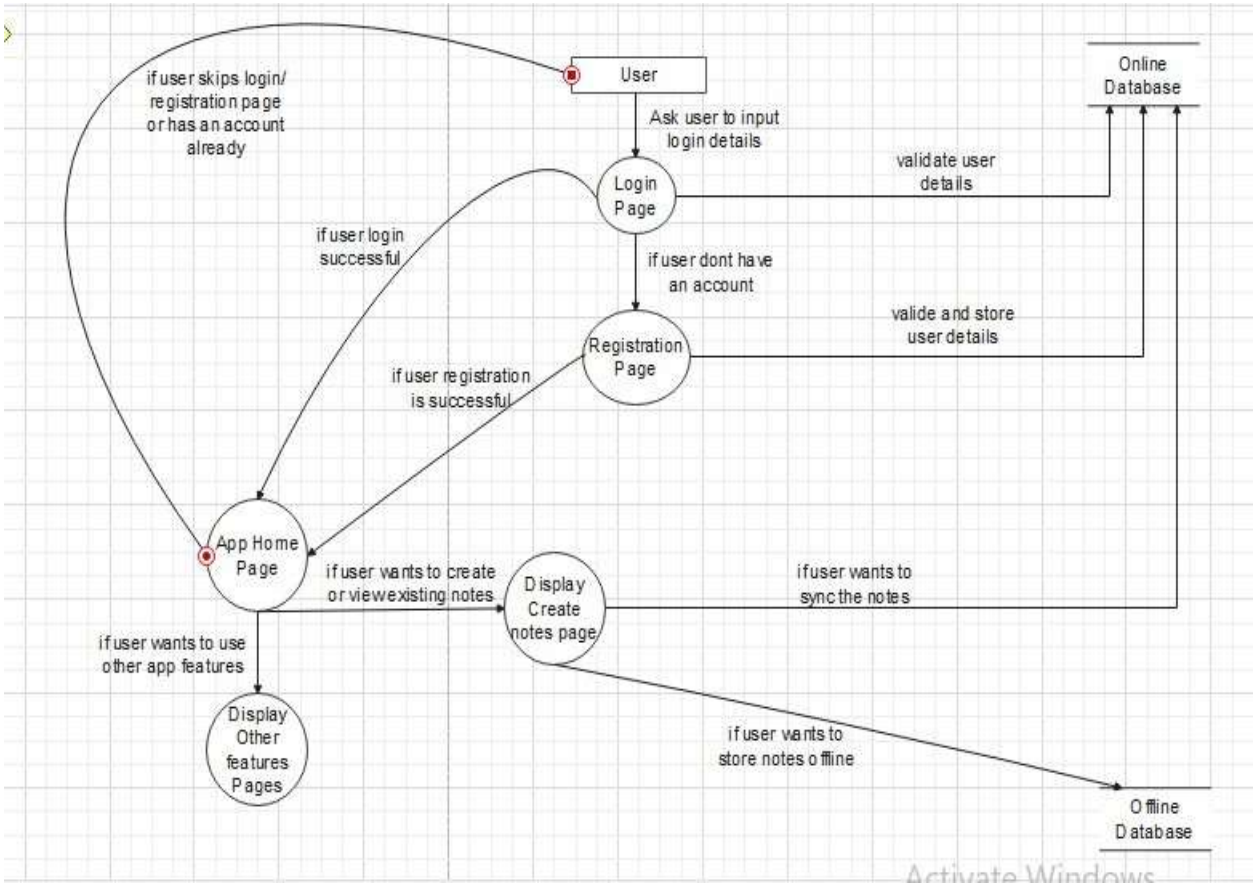


Figure 3.5.1.2

Level 2 Data Flow Diagram:-

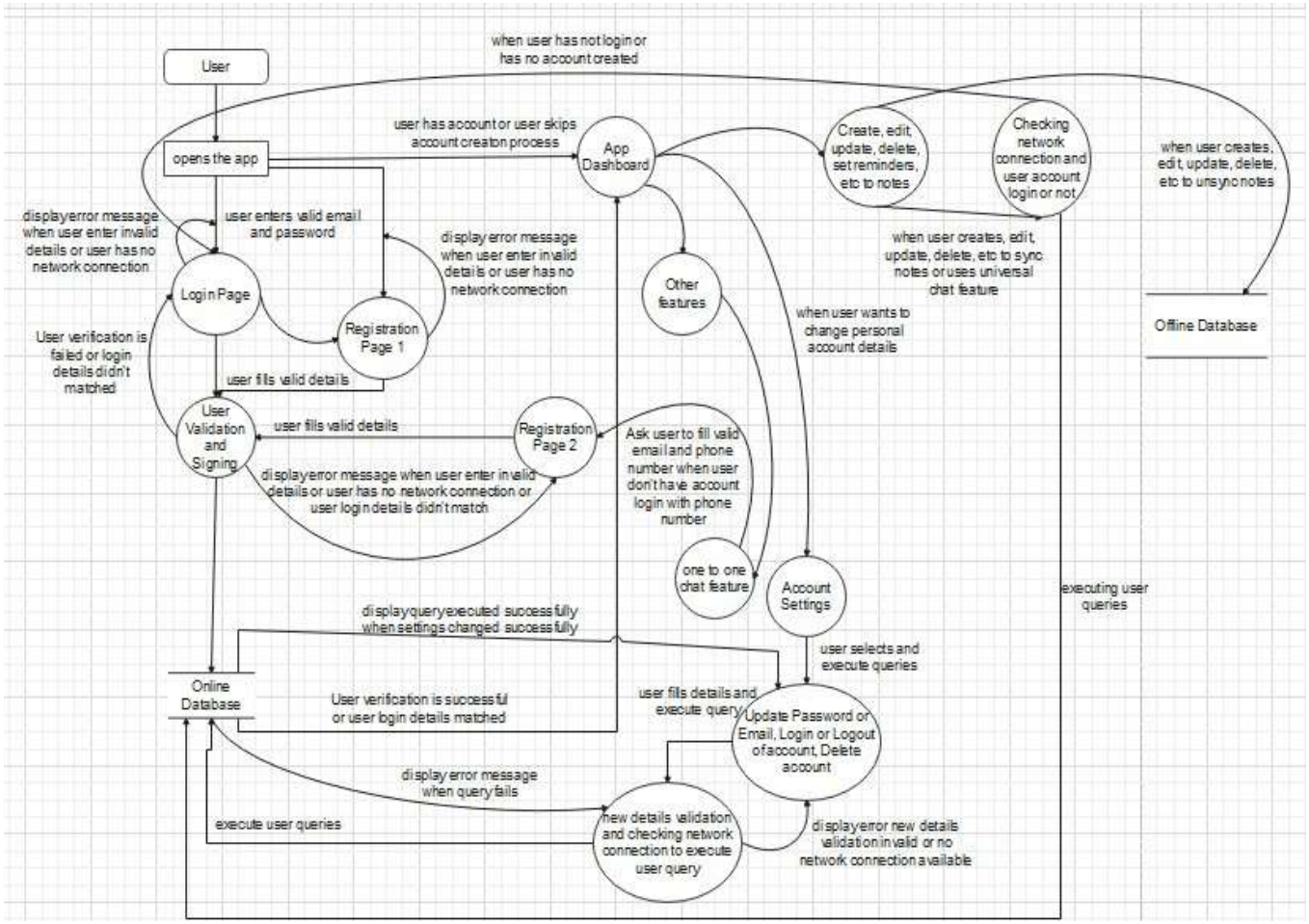


Figure 3.5.1.3

2. Use Case Diagram:-

A Use Case Diagram at it's simplest is a representation of a user interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. It is used to represent the dynamic behavior of a system.

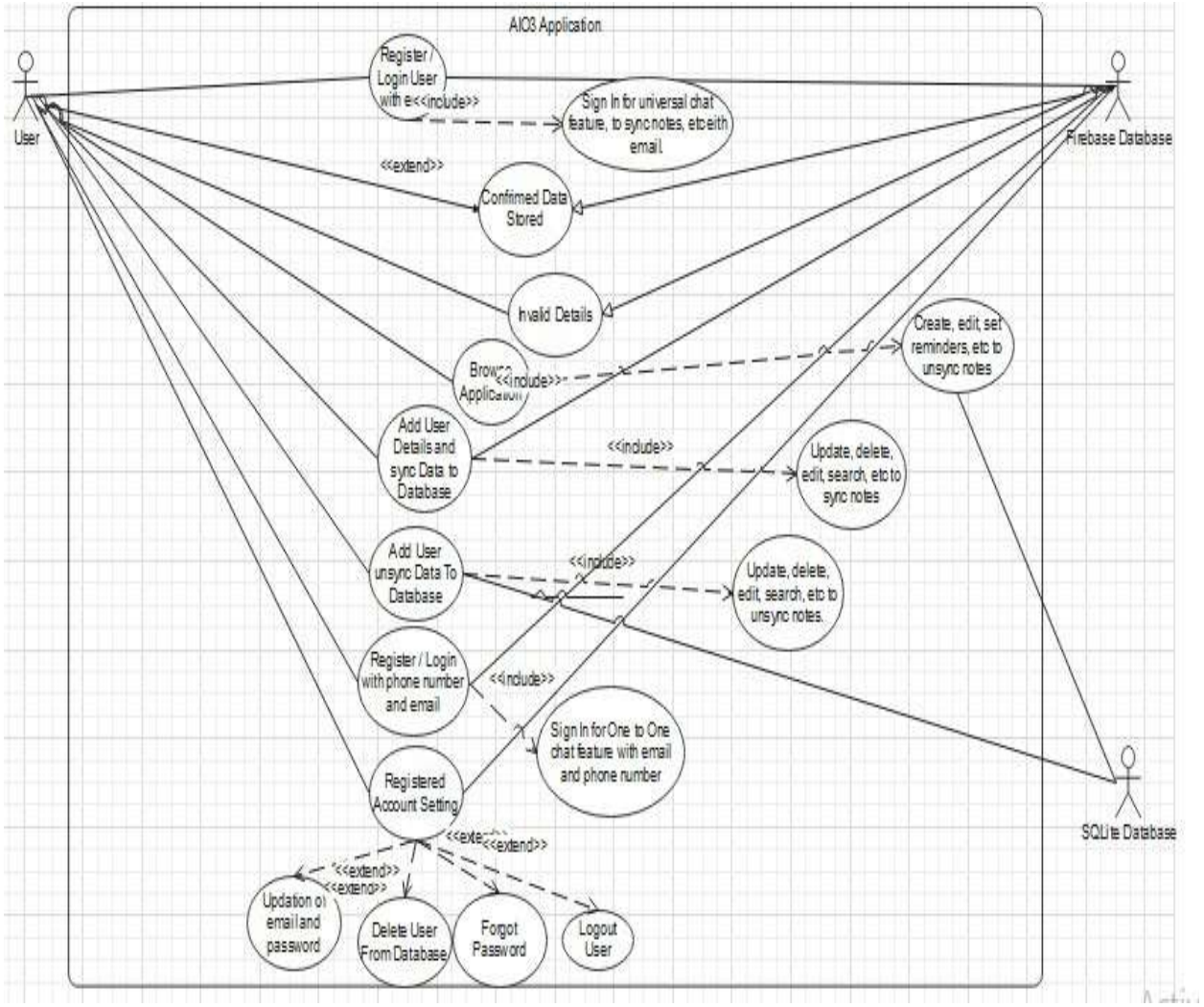


Figure 3.5.2

3. Activity Diagram:-

The activity diagram helps in envisioning the workflow from one activity to another. It put emphasis on the condition of flow and the order in which it occurs. The flow can be sequential, branched, or concurrent, and to deal with such kinds of flows, the activity diagram has come up with a fork, join, etc. It is also termed as an object-oriented flowchart.. Activity diagrams, along with use case and state machine diagrams, are considered behavior diagrams because they describe what must happen in the system being modeled.

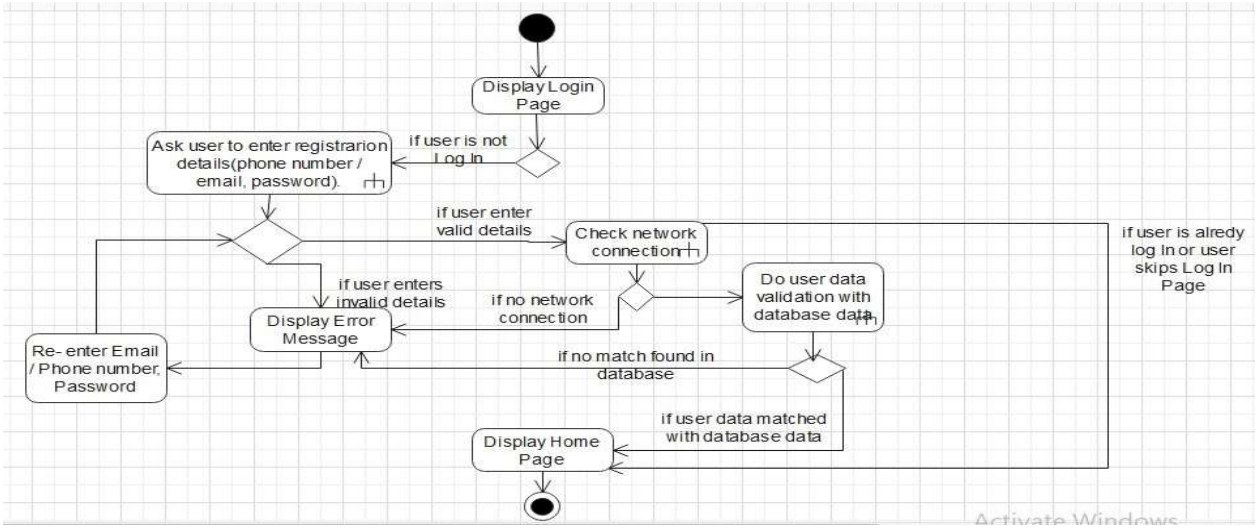


Figure 3.5.3.1

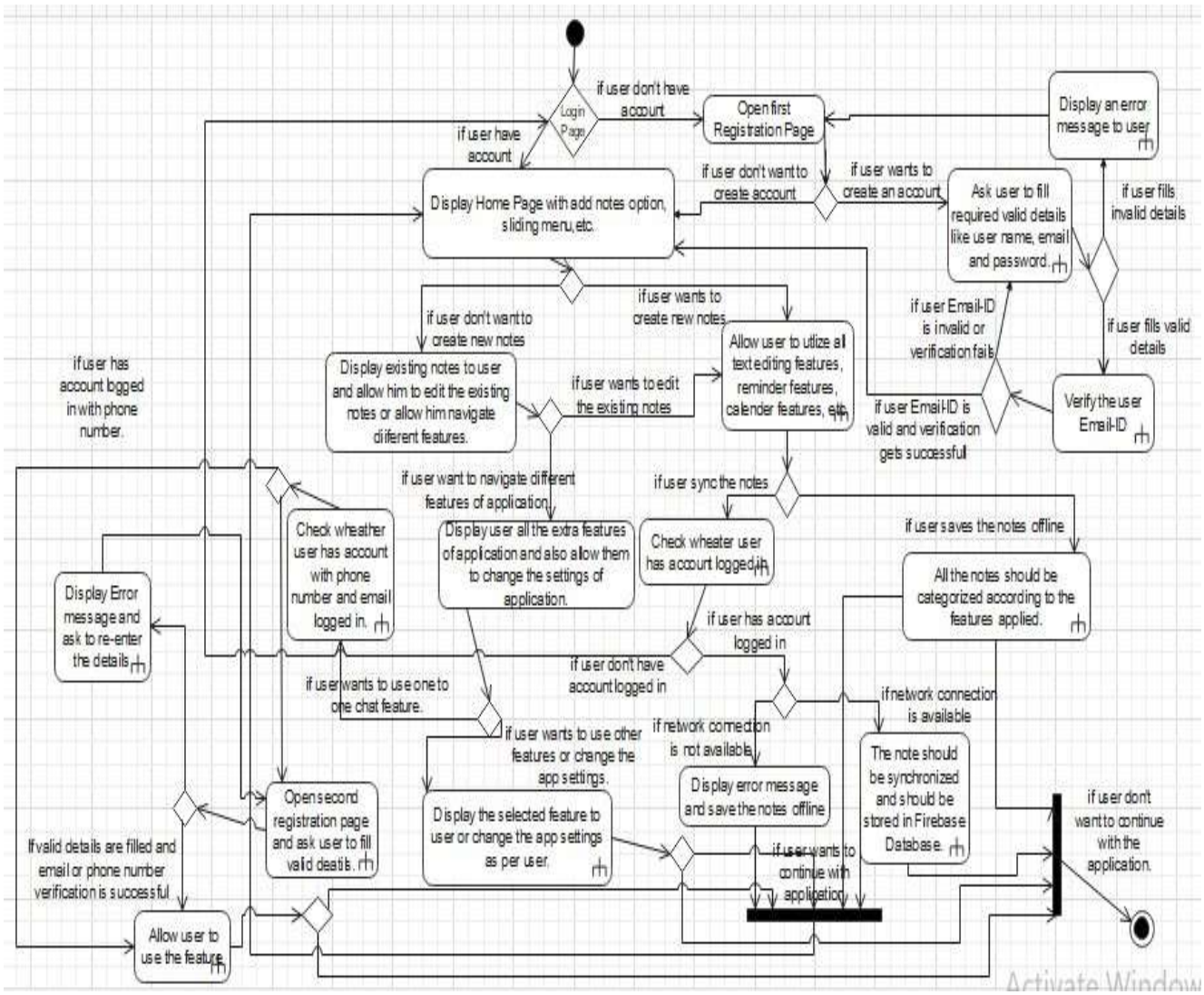
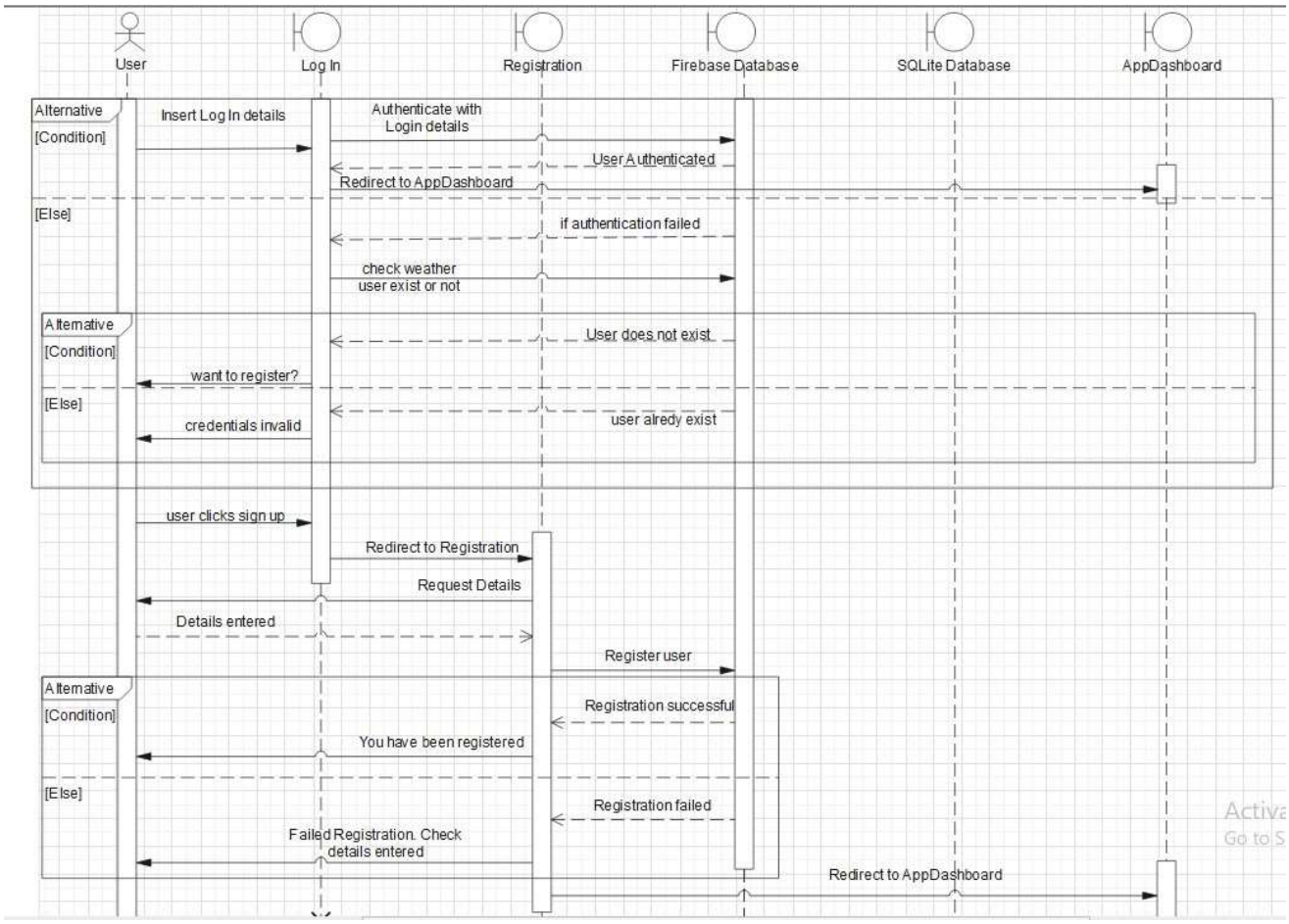

Login Page
if user don't have account
Open first Registration Page
Display an error message to user
if user have account
Display Home Page with add notes option, sliding menu, etc.
if user don't want to create account
if user wants to create an account
Ask user to fill required valid details like user name, email and password.
if user fills invalid details
if user wants to create new notes
if user Email-ID is invalid or verification fails
if user fills valid details
if user has account logged in with phone number.
if user don't want to create new notes
Display existing notes to user and allow him to edit the existing notes or allow him navigate different features.
if user wants to edit the existing notes
Allow user to utlize all text editing features, reminder features, calender features, etc.
if user Email-ID is valid and verification gets successful
Verify the user Email-ID
if user want to navigate different features of application
if user sync the notes
if user saves the notes offline
Check wheather user has account with phone number and email logged in.
Display user all the extra features of application and also allow them to change the settings of application.
Check wheater user has account logged in
All the notes should be categorized according to the features applied.
Display Error message and ask to re-enter the details
if user has account logged in
if user don't have account logged in
if user wants to use one to one chat feature.
if network connection is available
if user wants to use other features or change the app settings.
if network connection is not available
The note should be synchronized and should be stored in Firebase Database.
Open second registration page and ask user to fill valid deatils.
Display the selected feature to user or change the app settings as per user.
Display error message and save the notes offline
if user don't want to continue with the application.
if user wants to continue with application
If valid details are filled and email or phone number verification is successful
Allow user to use the feature

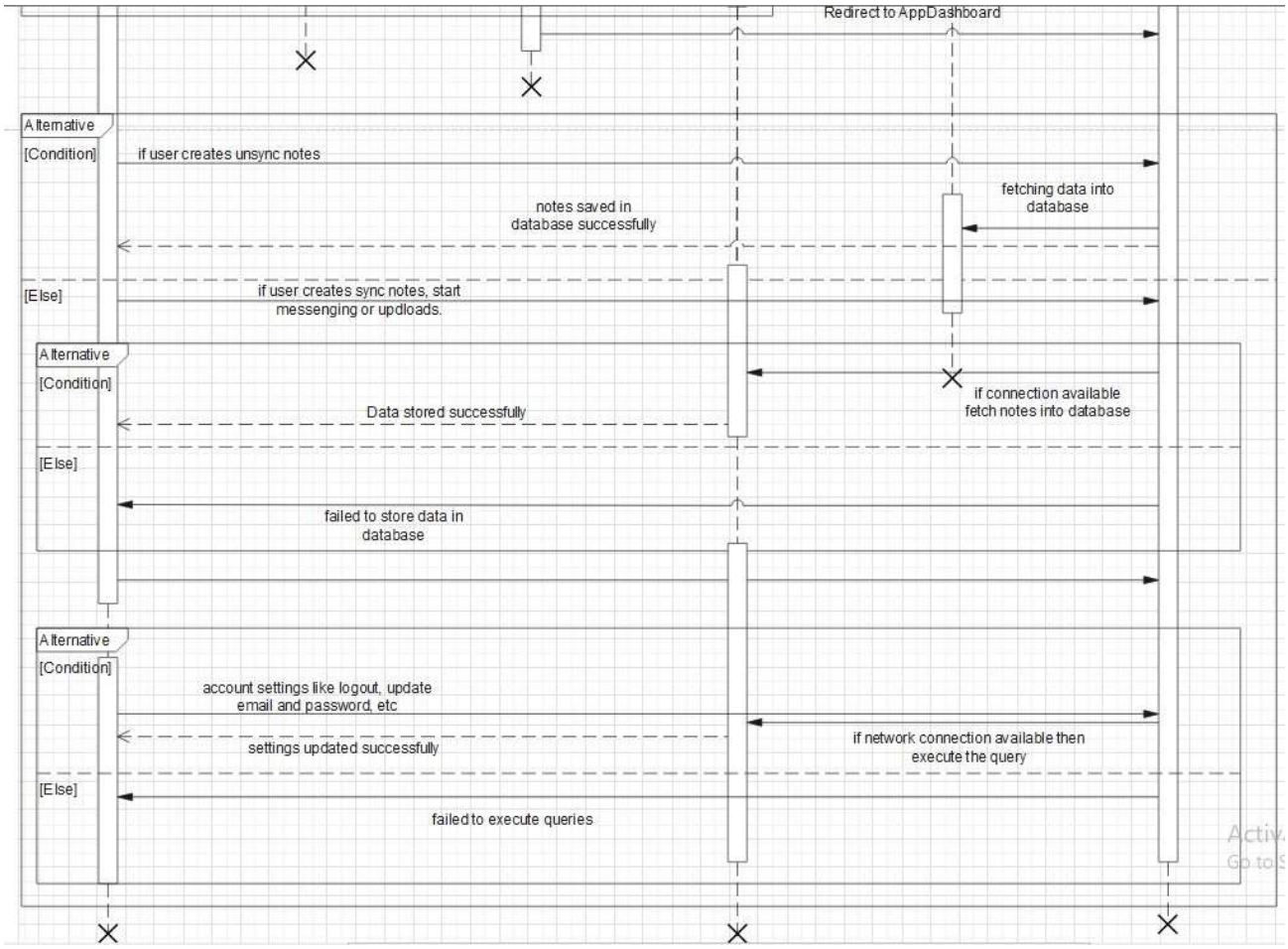
Figure 3.5.3.2

4. **Sequence Diagram:-**

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. It simply shows the object interactions arranged in a time sequence and also the message(data) exchanged. Sequence diagrams are sometimes known as event diagrams or event scenarios.

Figy




Redirect to AppDashboard
Alternative
[Condition]
if user creates unsync notes
fetching data into database
notes saved in database successfully
[Else]
if user creates sync notes, start messenging or updloads.
Alternative
[Condition]
if connection available fetch notes into database
Data stored successfully
[Else]
failed to store data in database
Alternative
[Condition]
account settings like logout, update email and password, etc
if network connection available then execute the query
settings updated successfully
[Else]
failed to execute queries

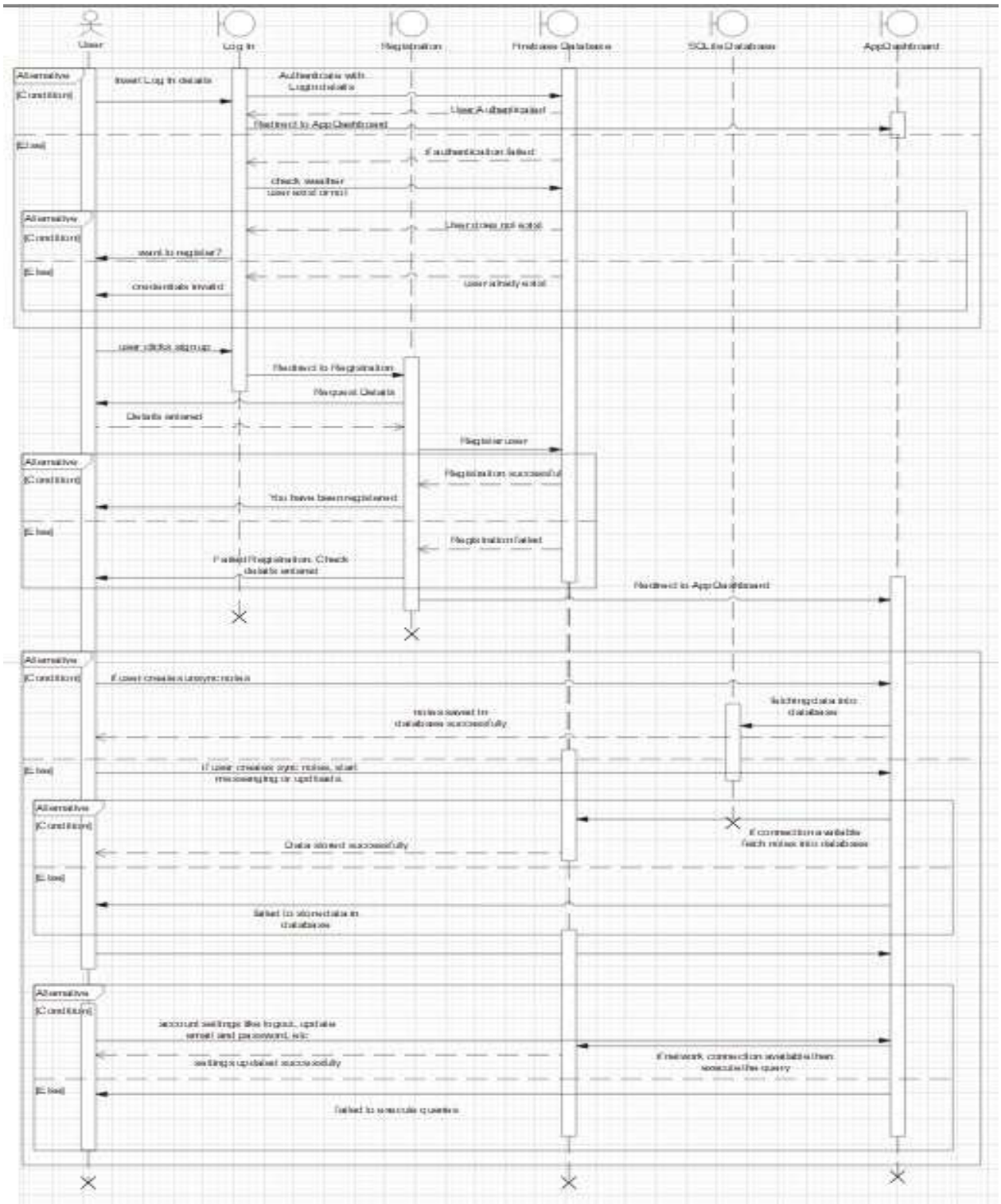

User
Log In
Registration
Firebase Database
SQLite Database
AppDashboard
Alternative
[Condition]
Insert Log In details
Authenticate with Login details
User Authenticated
Redirect to AppDashboard
[Else]
If authentication failed
check weather user exist or not
User does not exist
want to register?
[Else]
credentials invalid
user already exist
user clicks sign up
Redirect to Registration
Request Details
Details entered
Register user
Registration successful
You have been registered
Registration failed
Failed Registration. Check details entered
Redirect to AppDashboard
If user creates sync notes
notes saved to database successfully
fetching data into database
[Else]
If user creates sync notes, start messenging or upload a
If connection available fetch notes into database
Data stored successfully
failed to store data in database
account settings like logout, update email and password, etc
If network connection available then execute the query
setting is updated successfully
failed to execute queries

Figure 3.5.4

5. Class Diagram:-

A Class Diagram in UML(Unified Modeling Language) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations(or method) and the relationship among the objects. The main purpose of the class diagram is to build a static view of an application.

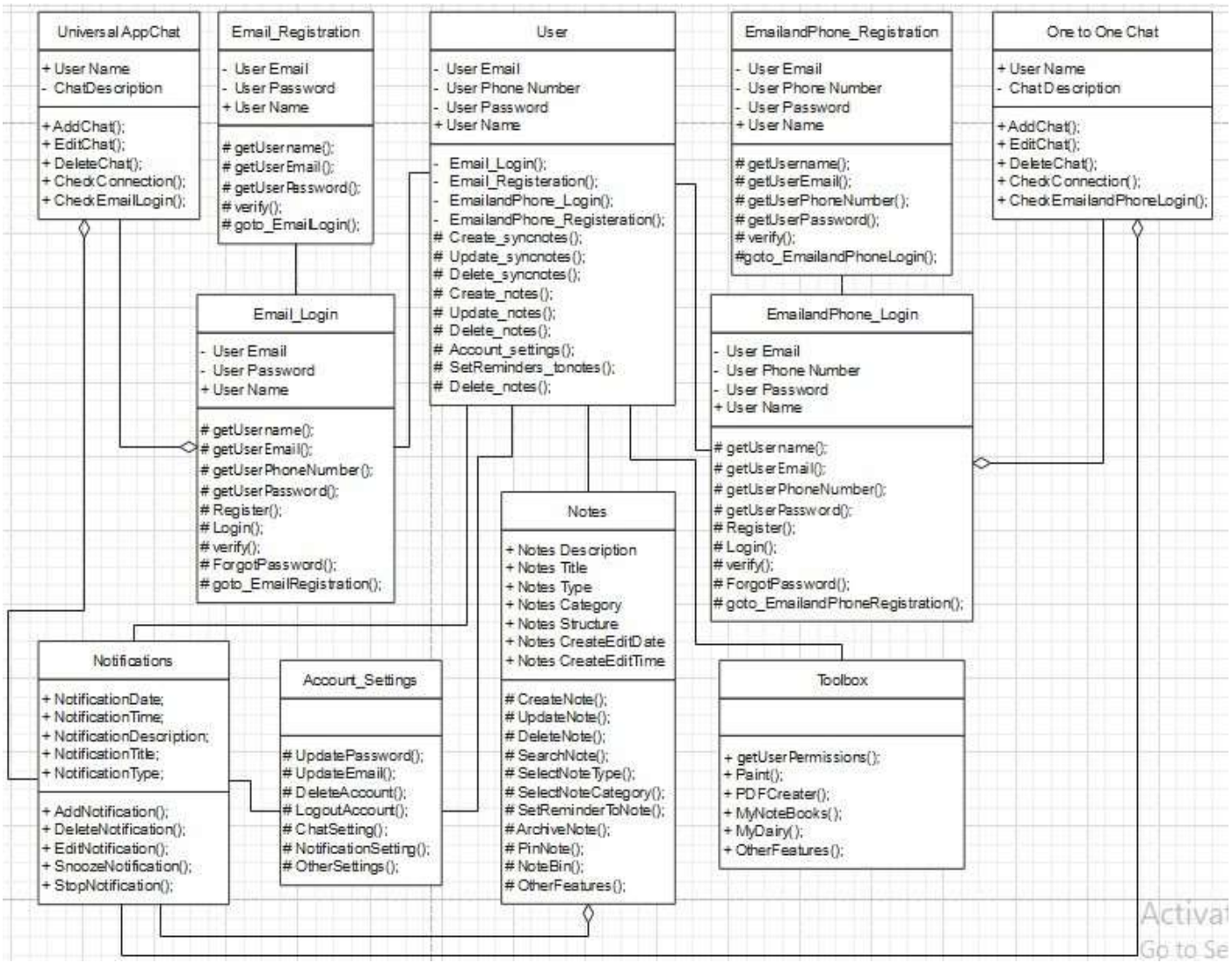


Figure 3.5.5

6. Event Table:-

The event table is a database table created by the user, generally within the same schema as the application table for which it stores events. The event table describes the type of change made to an application table, and also contains an identifier for the changed row.

Event Name	Trigger	Source	Activity	Response	Destination
Email Login	To store, delete, update, retrieve, etc in/from the firebase database or to use universal app chat feature.	User	Enter Email and Password	Home Page	User
Email Registration	To register in the system.	User	Enter registration Details	Login Page	User
Email and Phone Login	To use one to one app chat feature.	User	Enter Login Details	Home Page	User
Email and Phone Registration	To unlock one to one app chat feature.	User	Enter registration Details	Login Page	User
Forgot Password	To enter the system.	User	Enter update/new password	Login Page	User
Create Notes and save Online	To store, delete, update, retrieve, search, etc from/in the database	User	Add, update, delete, search, etc the notes.	Online Database	Online Database
Create Notes and save Offline	To store, delete, update, retrieve, search, etc from/in the database	User	Add, update, delete, search, etc the notes.	Offline Database	Offline Database
Use Toolbox Features	Ask user to give access permissions	User	To create or upload PDF, to use Paint and save pictures in device, to edit user images, etc	Toolbox Page	User
Edit User Details	To update changes in the database.	User	Edit Registered User Details	Online Database	Online Database

Table 3.5.6

Chapter 4

System Design

4.1 Basic Modules

1. Registration

In this module the user can register themselves and the data they have entered while registering will be saved in the database.

2. Login

In this module the user can Login into the system by entering Mobile number or email and password that should be same that was entered while registration.

3. Settings

In this module the user can change the account settings like update profile, update email, update password, logout, delete account, etc. and the changes will be saved in database.

4. Toolbox

In this module the user can use the extra app features by clicking on any of the feature. Some feature will ask for access permissions and user should allow the permission for feature to start.

5. Universal App Chat:

In this module the user can chat with anyone inside a group of all members of application who are using this feature. To use this feature user must have an account created.

6. One to One Chat:

In this module the user can chat with any one person from his contact list who is using this application and has an account registered with phone number.

7. Notes Module:

In this module the user can create, update, delete, search, set reminders, etc to notes. The notes can be saved either online or offline or both. To create, update, delete, search, set reminders, etc to online notes the user must have an account and network connection available. The user can create different types of notes and can use all text editing features for free of cost and free of advertisement.

4.2 Data Design

4.2.1 Data Integrity

User Table

Field	Type	Null
Name	String	No
Phone Number	Int	No
Email	String	No
Password	String	No

Online Database

Field	Type	Null
Title	String	No
Description	String	No
Time	String	No
Date	String	No
Phone Number	Int	No
Email	String	No
Password	String	No
Task	String	No
Status	String	No

Message text	String	No
Message user	String	No
Message time	String	No

Offline Database

Field	Type	Null
Title	String	No
Description	String	No
Time	String	No
Date	String	No
Task	String	No
Status	String	No

Table 4.2.1

4.3 Procedure Design

4.3.1 Flow Chart

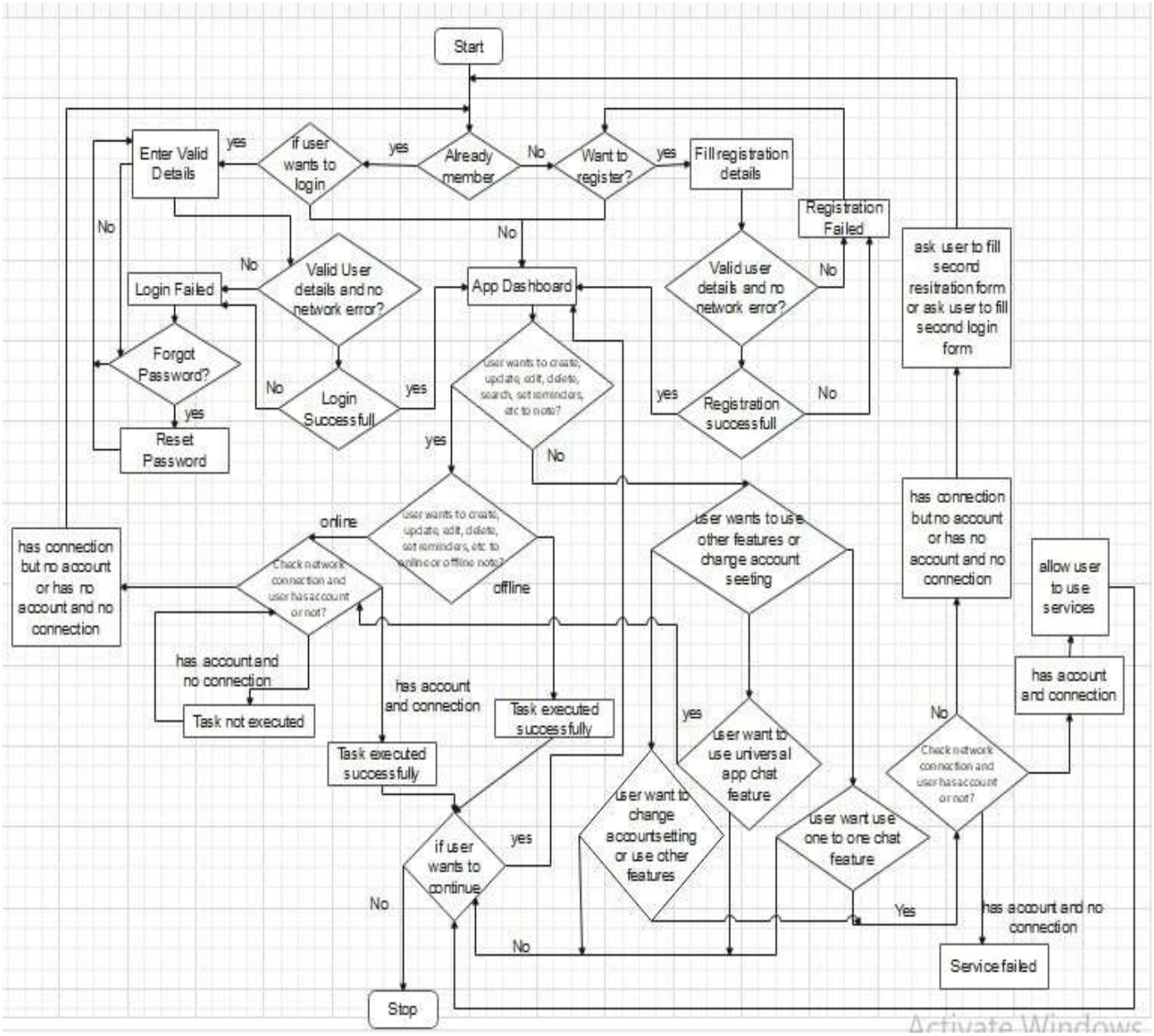


Figure 4.2.1

4.4 Algorithm Design

Fundamental Algorithm of Application:

Step 1: Start

Step 2: If user already member then

If user wants to enter login details then

Display Login Page

Else

Display App Dashboard

Else If user wants to register then

Display registration page

Else

Display App Dashboard

Step 3: (After user enters App Dashboard)

Check If user wants to create, update, delete, search, etc to online notes or offline notes then

If user wants to create, update, delete, search, set reminders, etc to offline notes then

Execute user query and display task executed successfully

If user wants to continue then

Display App Dashboard

Else

Exit

Else Check If user has network connection and account login then

Execute user query and display task executed successfully

If user wants to continue then

Display App Dashboard

Else

Exit

Else If user has account but no network connection then

Display no network connection error and task execution failed

Else

Display Login Page

Else If user wants to use universal app chat feature then

Check If user has network connection and account login then

Allow user to send messages and receive new messages

Else If user has account but no network connection then

Display no network connection error and allow user to view old messages

Else

Display Login Page

Else If user want to change the account setting then

Check If user has network connection and account login then

If user does valid changes then

Change account setting of user

Else

Display error and changes failed to save

Else If user has account but no network connection then

Display no network connection error and changes failed to save

Else

Display Login Page2

Else If user want to use one to one chat feature then

Check If user has network connection and account login then

Allow user to send messages and receive new messages

Else If user has account but no network connection then

Display no network connection error and allow user to view old messages

Else

Display Login Page2

Else

Allow user to view or use other app features

Step 4: Stop

Algorithm for Login Page / Login Page2:

Step 1: Start

Step 2: Open Login Page with forgot password, continue and go to registration page options

Step 3: Check If user wants to login then

Ask user to enter valid details and check user network connection
If user enters valid details and no network connection error then
Toast message login successful and display user App Dashboard
Else
Toast error occurred and display again login page
Else If user wants to register then
Display user registration page
Else If user clicks on forgot password option then
Check If user has network connection and account login then
Ask user to enter valid details and check user network connection
If user enters valid details and changes password by clicking on reset link sent on entered email-id then
Toast message password changed successfully and display again login page to user
Else
Toast error occurred and display again login page
Else If user has account but no network connection then
Display no network connection error and display again login page

Else

Display Registration Page

Else

Display App Dashboard

Step 4: Stop

Algorithm for Registration Page / Registration Page2:

Step 1: Start

Step 2: Open Registration Page with Login page and continue option

Step 3: Check If user wants to register then

Ask user to enter valid details and check user network connection

If user enters valid details and no network connection error then

Do verification process

If verification is successful then

Toast message registration successful and display user App Dashboard

Else

Toast verification unsuccessful and display registration page again

Else

Toast error occurred and display again registration page

Else If user wants to login then

Display user login page

Else

Display App Dashboard

Step 4: Stop

4.5 UI Design

Login Pages Design:

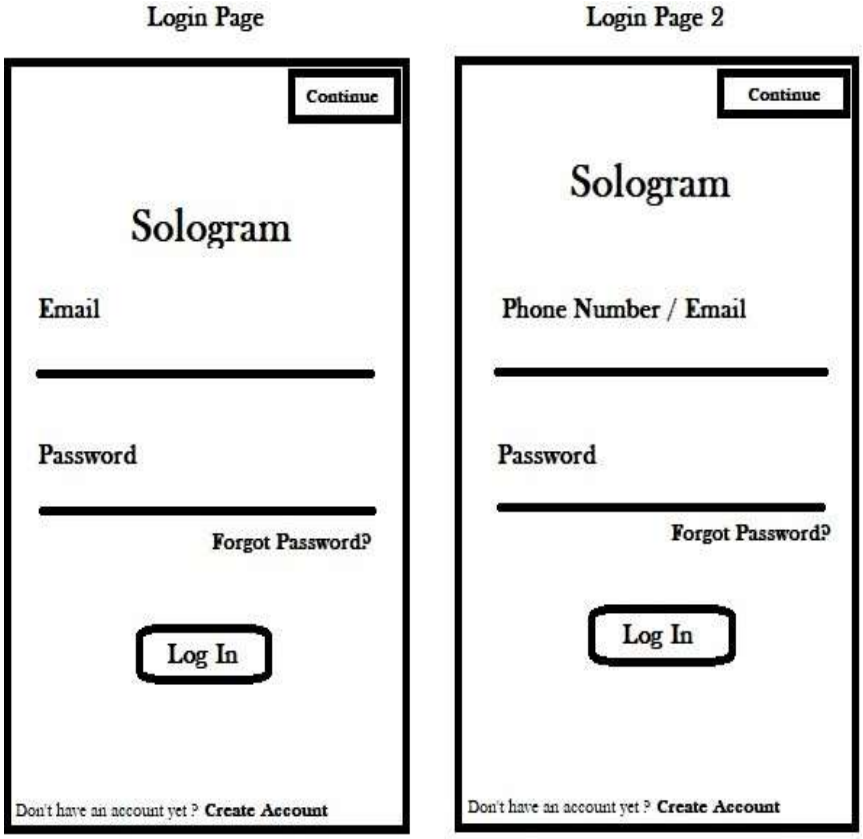


Figure 4.5.1.1

Registration Pages:

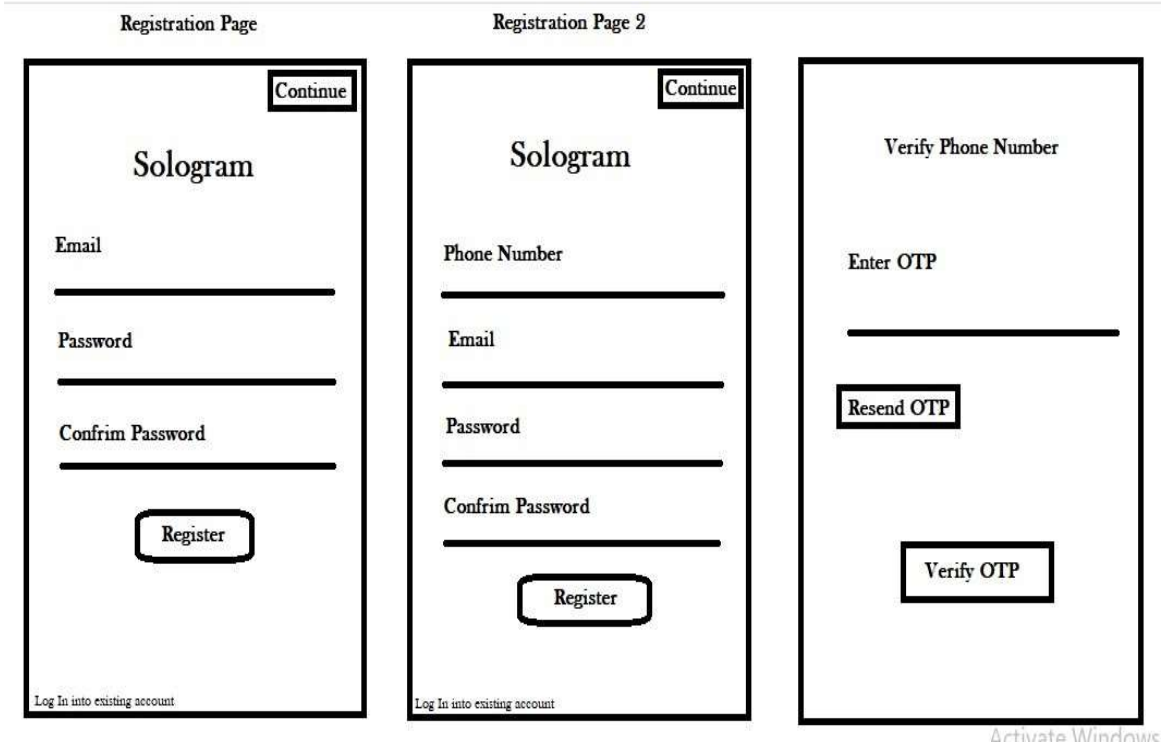


Figure 4.5.1.2

Toolbox Interface Page:

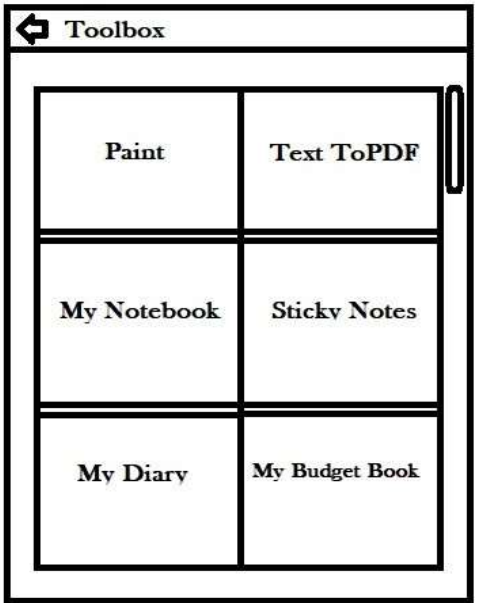


Figure 4.5.1.3

Universal and One to One chat feature Pages:

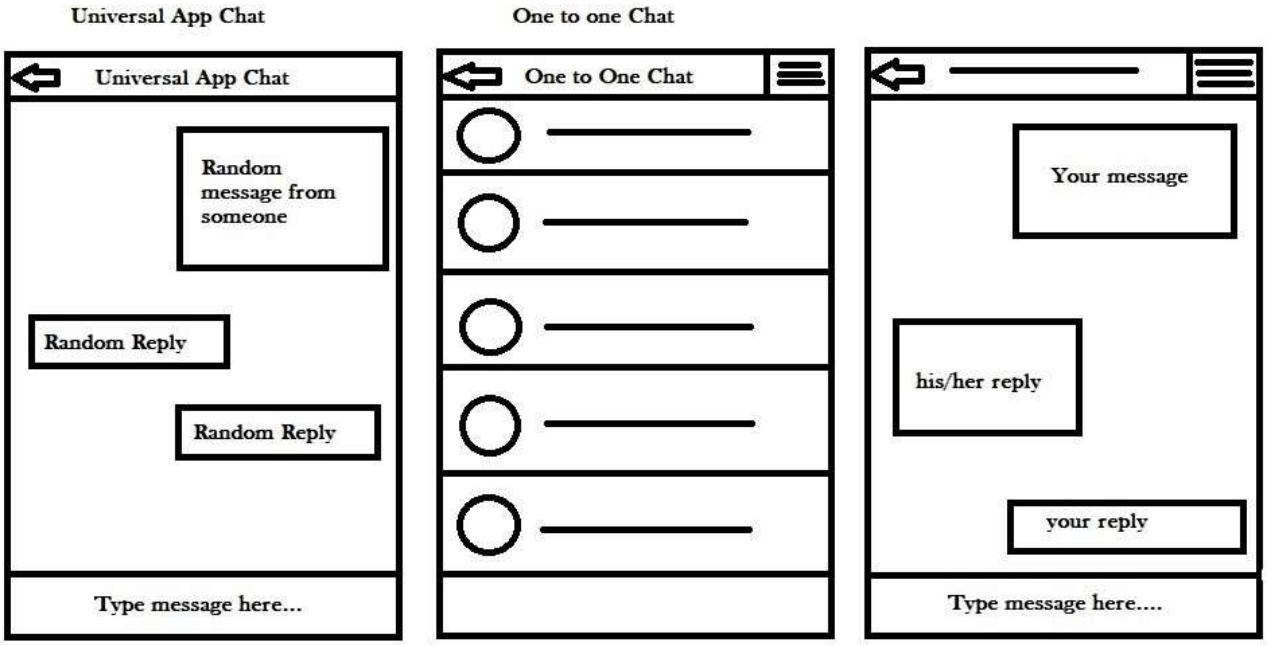


Figure 4.5.1.4

App Dashboard:

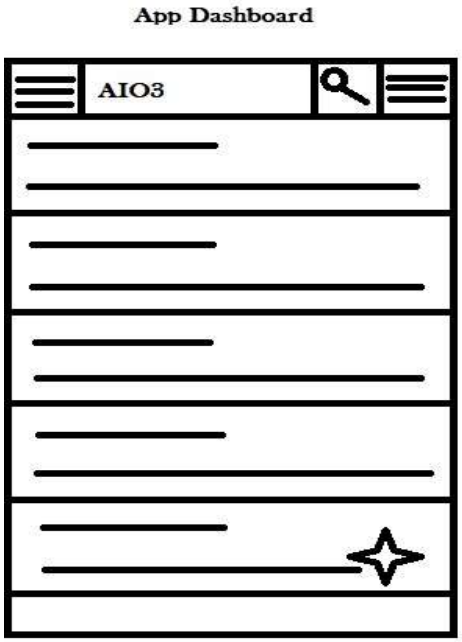


Figure 4.5.1.3

Basic Note Creating Pages:

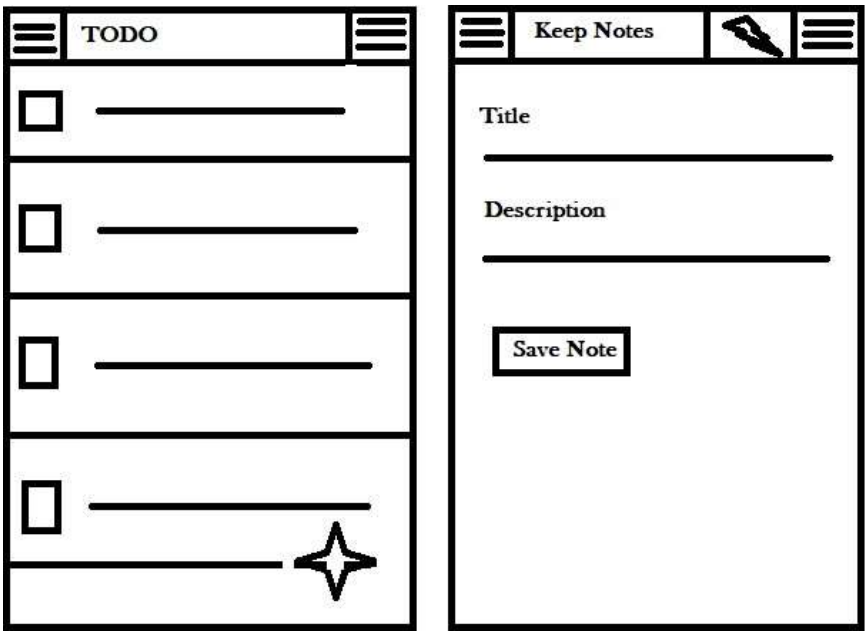


Figure 4.5.1.6

Settings Page:

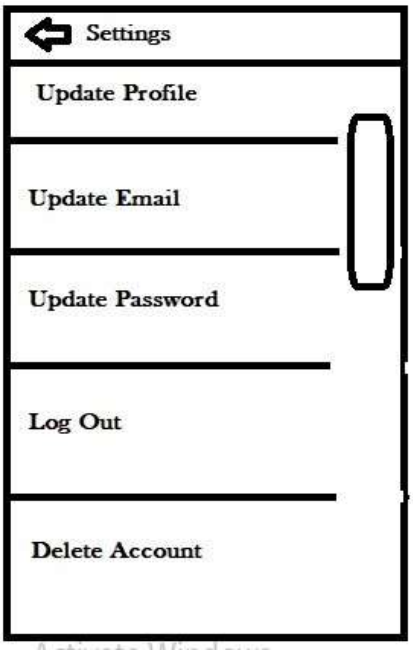


Figure 4.5.1.3

4.6 Test Case Design

Test Case ID	Description	Input Data	Expected Result	Actual Result	Result
TC_1	**Login Page** Login into system by entering correct details.	Enter Email Address and Password.	Open Home Page		
TC_2	**Login Page2** Login into system by entering correct details.	Enter Phone Number and Password.	Open Home Page		
TC_3	**Registration Page** To store the details in the database.	Enter the required registration details.	Open Home Page		
TC_4	**Registration Page2** To store the details in the database.	Enter the required registration details.	Open Home Page		
TC_5	**Forgot Password** To enter the new password and login again	Enter the new password	Open Login Page		
TC_6	**Universal Application Chat** To allow all the application users	Select the feature and start messaging	Open universal application chat Interface.		

	to ask their questions openly to anyone who has account login in application.	your queries to all users.			
TC_7	**One to One Chat** To allow user to chat with particular persons who has account login in the application.	Select the feature and start messaging your queries to particular users.	Open one to one chat Interface.		
TC_8	**Edit User Details** To update the user details in the database.	Enter the details to be updated.	Open Setting Page		
TC_8.1	**Update Email** To update the user Email if he has account.	Select the Update Email option and update the details and click on the update button.	Show Update Email Interface.		
TC_8.2	**Update Password** To update the user Password if he has account.	Select the Update Password option and update the details and click on the update button.	Show Update Password Interface.		

TC_8.3	**Logout of Account** To Logout of account if user has account and wants to logout.	Select the Logout option and click on the confirm button.	Open Login Page.		
TC_8.4	**Delete Account Permanently** To delete account permanently of user if he has account.	Select the Delete Account option and click on the confirm button.	Open Registration Page.		
TC_9	**Notes** To allow user to create, update, delete, search, set reminders, etc to notes which can be either online or offline as per user how user wants to save them.	Select the notes and edit them or create new notes and do use different features provided for free.	Open Notes editing / creating Interface		
TC_10	**Toolbox** To allow the user to use the application extra features.	Select the any of the feature and click on that feature.	Open a new interface as per the feature selected by the user.		
TC_10.1	**Paint** To allow user to draw anything he	Select any paint feature and start doing	Open Paint Interface.		

	wants.	editing.			
TC_10.2	**Text to PDF** To allow user to write anything he wants and convert it into PDF.	Select any Text to PDF feature and start creating PDF.	Open Text to PDF interface.		
TC_10.3	**My Notebook** To allow user to write anything he wants and save it in the book structure.	Select any Notebook feature and start creating book.	Open My Notebook interface.		
TC_10.4	**My Diary** To allow user to write anything he wants as per date wise.	Select any Diary feature and allow user to store his memories and task.	Open Diary interface.		

Table 4.6

CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Implementation Approaches

Scrum

Scrum is an agile development methodology used in the development of Software based on an iterative and incremental processes. Scrum is adaptable, fast, flexible and effective agile framework that is designed to deliver value to the customer throughout the development of the project.

Scrum methodology was chosen to manage the project cycle. Scrum Methodology follow the agile development principle which is an iterative development methodology. However, scrum methodology was not followed to its full potential. A meeting to report what was done and what will be done for the duration of time was held face to face. The meetings were held on regular basis. The overall project was not managed as it should be, the prototype iterations were completed with success. The whole cycle of choosing the priority features and implementing a working version allowed quick progress in implementing the features. Also, the nature of the iterative approach allowed easy adaption of changing requirements during the project.

5.2 Coding Details and Code Efficiency

5.2.1 Coding

SplashFragment.java

```
package com.example.allinone;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import java.util.Objects;
import java.util.Timer;
import java.util.TimerTask;
public class SplashFragment extends Fragment {
   View view;
```

```
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Objects.requireNonNull(((AppCompatActivity)
requireActivity()).getSupportActionBar()).hide();
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        view = inflater.inflate(R.layout.fragment_splash, container, false);
        Timer timer=new Timer();
        timer.schedule(new TimerTask() {
            @Override
            public void run() {
                ((MainActivity) requireActivity()).replaceFragments(StartFragment.class);
            }
        },2000);
        return view;
    }
}
```

AnonymousChatFragment.java

```
package com.example.allinone;
import android.graphics.Color;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.app.ProgressDialog;
import android.content.Intent;
```

```
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.Iterator;
import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ScrollView;
import android.widget.TextView;
import com.firebase.client.ChildEventListener;
import com.firebase.client.DataSnapshot;
import com.firebase.client.Firebase;
import com.firebase.client.FirebaseError;
import java.util.HashMap;
import java.util.Map;
public class AnonymousChatFragment extends Fragment {
```

```
View view;
LinearLayout layout;
ImageView sendButton;
EditText messageArea;
ScrollView scrollView;
Firebase reference1, reference2;
@Override
public void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
}
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                  Bundle savedInstanceState) {
   // Inflate the layout for this fragment
   view = inflater.inflate(R.layout.fragment_anonymous_chat, container, false);
   layout = (LinearLayout)view.findViewById(R.id.layout1);
   sendButton = (ImageView)view.findViewById(R.id.sendButton);
   messageArea = (EditText)view.findViewById(R.id.messageArea);
   scrollView = (ScrollView)view.findViewById(R.id.scrollView);
   Firebase.setAndroidContext(requireActivity());
   // change this firebase link accounding your firebase account
   reference1 = new Firebase("https://all-in-one-d0b51-default-
rtdb.firebaseio.com/AnonymousChat/" + UserDetails.username + "_" + UserDetails.chatWith);
   reference2 = new Firebase("https://all-in-one-d0b51-default-
rtdb.firebaseio.com/AnonymousChat/" + UserDetails.chatWith + "_" + UserDetails.username);
   sendButton.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
         String messageText = messageArea.getText().toString();
         if(!messageText.equals("")){
            Map<String, String> map = new HashMap<String, String>();
```

```
            map.put("message", messageText);
            map.put("user", UserDetails.username);
            reference1.push().setValue(map);
            //reference2.push().setValue(map);
        }
    }
});
reference1.addChildEventListener(new ChildEventListener() {
    @Override
    public void onChildAdded(DataSnapshot dataSnapshot, String s) {
        Map map = dataSnapshot.getValue(Map.class);
        String message = map.get("message").toString();
        String userName = map.get("user").toString();
        if(userName.equals(UserDetails.username)){
            addMessageBox("Anonymous:-\n" + message, 1);
        }
        else{
            addMessageBox(userName + ":-\n" + message, 2);
        }
    }
    @Override
    public void onChildChanged(DataSnapshot dataSnapshot, String s) {
    }
    @Override
    public void onChildRemoved(DataSnapshot dataSnapshot) {
    }
    @Override
    public void onChildMoved(DataSnapshot dataSnapshot, String s) {
    }
    @Override
    public void onCancelled(FirebaseError firebaseError) {
```

```
            }
        });
        return view;
    }
    public void addMessageBox(String message, int type){
        TextView textView = new TextView(getActivity());
        textView.setText(message);
        LinearLayout.LayoutParams lp = new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
        lp.setMargins(0, 0, 0, 10);
        textView.setLayoutParams(lp);
        if(type == 1) {
            textView.setBackgroundColor(Color.GREEN);
        }
        else{
            textView.setBackgroundColor(Color.BLUE);
        }
        layout.addView(textView);
        scrollView.fullScroll(View.FOCUS_DOWN);
    }
}
```

ChatListFragment.java

```
package com.example.allinone;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
```

```
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Iterator;
import java.util.Map;
import java.util.Set;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;
public class ChatroomListFragment extends Fragment {
    View view;
    DatabaseReference reference;
    ArrayList<String> arrayList;
    EditText e1;
    ListView l1;
    ArrayAdapter<String> adapter;
```

```
    String name;
    EditText ee;
    Button add;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        view = inflater.inflate(R.layout.fragment_chatroom_list, container, false);
        e1 = (EditText)view.findViewById(R.id.editText);
        l1 = (ListView)view.findViewById(R.id.listView);
        add = (Button)view.findViewById(R.id.add_room);
        arrayList = new ArrayList<>();
        adapter = new
ArrayAdapter<String>(getActivity(),android.R.layout.simple_list_item_1,arrayList);
        l1.setAdapter(adapter);
        reference = FirebaseDatabase.getInstance().getReference().child("Chatroom");
        request_username();
        reference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                Set<String> set = new HashSet<String>();
                Iterator i = dataSnapshot.getChildren().iterator();
                while (i.hasNext()) {
                    set.add(((DataSnapshot) i.next()).getKey());
                }
                arrayList.clear();
                arrayList.addAll(set);
```

```
            adapter.notifyDataSetChanged();
        }
        @Override
        public void onCancelled(DatabaseError databaseError) {
            Toast.makeText(getActivity(), "No network connectivity",
Toast.LENGTH_SHORT).show();
        }
    });
    add.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            insert_data();
        }
    });
    l1.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
            Bundle bundle = new Bundle();
            //intent.putExtra("noteId", noteId);
            bundle.putString("room_name",((TextView) view).getText().toString());
            bundle.putString("user_name",name);
            FragmentManager fragmentManager = getActivity().getSupportFragmentManager();
            FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
            ChatroomChatFragment chatroomChatFragment = new ChatroomChatFragment();
            chatroomChatFragment.setArguments(bundle);
            fragmentTransaction.replace(R.id.fragment_container,chatroomChatFragment);
            fragmentTransaction.commit();
        }
    });
    return view;
}
```

```
    public void request_username()
    {
        final AlertDialog.Builder builder = new AlertDialog.Builder(requireActivity());
        builder.setTitle("Enter your name?");
        ee = new EditText(getActivity());
        builder.setView(ee);
        builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                name = ee.getText().toString();
            }
        });
        builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                dialogInterface.cancel();
                request_username();
            }
        });
        builder.show();
    }
    public void insert_data()
    {
        Map<String,Object> map = new HashMap<>();
        map.put(e1.getText().toString(), "");
        reference.updateChildren(map);
    }
}
```

LoginFrgment.java

```
package com.example.allinone;
```

```
import android.app.ProgressDialog;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import android.text.TextUtils;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.textfield.TextInputLayout;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import java.util.Objects;
public class LoginFragment extends Fragment {
    View view;
    TextInputLayout inputEmail, inputPass;
    private Button btnLogIn;
    private FirebaseAuth fAuth;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setHasOptionsMenu(true);
        Objects.requireNonNull(((AppCompatActivity)
requireActivity()).getSupportActionBar()).show();
        Objects.requireNonNull(((AppCompatActivity)
requireActivity()).getSupportActionBar()).setDisplayHomeAsUpEnabled(true);
```

```
        Objects.requireNonNull(((AppCompatActivity)
requireActivity()).getSupportActionBar()).setDisplayShowHomeEnabled(true);
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        view = inflater.inflate(R.layout.fragment_login, container, false);
        inputEmail = (TextInputLayout) view.findViewById(R.id.input_log_email);
        inputPass = (TextInputLayout) view.findViewById(R.id.input_log_pass);
        btnLogIn = (Button) view.findViewById(R.id.btn_log);
        fAuth = FirebaseAuth.getInstance();
        btnLogIn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String                                    lEmail                                    =
Objects.requireNonNull(inputEmail.getEditText()).getText().toString().trim();
                String                                     lPass                                    =
Objects.requireNonNull(inputPass.getEditText()).getText().toString().trim();
                if (!TextUtils.isEmpty(lEmail) && !TextUtils.isEmpty(lPass)) {
                    logIn(lEmail, lPass);
                }
            }
        });
        return view;
    }
    private void logIn(String email, String password){
        final ProgressDialog progressDialog = new ProgressDialog(getActivity());
        progressDialog.setMessage("Logging in, please wait...");
        progressDialog.show();
        fAuth.signInWithEmailAndPassword(email, password)
```

```
            .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    progressDialog.dismiss();
                    if (task.isSuccessful()) {
                        ((MainActivity)
requireActivity()).replaceFragments(AnonymousChatFragment.class);
                        Toast.makeText(getActivity(), "Sign in successful",
Toast.LENGTH_SHORT).show();
                    } else {
                        Toast.makeText(getActivity(), "ERROR: " +
Objects.requireNonNull(task.getException()).getMessage(), Toast.LENGTH_SHORT).show();
                    }
                }
            });
    }
    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        super.onOptionsItemSelected(item);
        switch (item.getItemId()){
            case android.R.id.home:
                ((MainActivity) requireActivity()).replaceFragments(StartFragment.class);
                break;
        }
        return true;
    }
}
```

RegisterFragment.java

```
package com.example.allinone;
import android.app.ProgressDialog;
```

```
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.textfield.TextInputLayout;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import java.util.Objects;
public class RegisterFragment extends Fragment {
    View view;
    private Button btnReg;
    private TextInputLayout inName, inEmail, inPass;
    private FirebaseAuth fAuth;
    private DatabaseReference fUsersDatabase;
    private ProgressDialog progressDialog;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setHasOptionsMenu(true);
        Objects.requireNonNull(((AppCompatActivity)
requireActivity()).getSupportActionBar()).show();
```

```
        Objects.requireNonNull(((AppCompatActivity)
requireActivity()).getSupportActionBar()).setDisplayHomeAsUpEnabled(true);
        Objects.requireNonNull(((AppCompatActivity)
requireActivity()).getSupportActionBar()).setDisplayShowHomeEnabled(true);
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        view = inflater.inflate(R.layout.fragment_register, container, false);
        btnReg = (Button) view.findViewById(R.id.btn_reg);
        inName = (TextInputLayout) view.findViewById(R.id.input_reg_name);
        inEmail = (TextInputLayout) view.findViewById(R.id.input_reg_email);
        inPass = (TextInputLayout) view.findViewById(R.id.input_reg_pass);
        fAuth = FirebaseAuth.getInstance();
        fUsersDatabase = FirebaseDatabase.getInstance().getReference().child("Users");
        btnReg.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String                                          uname                                          =
Objects.requireNonNull(inName.getEditText()).getText().toString().trim();
                String                                          uemail                                         =
Objects.requireNonNull(inEmail.getEditText()).getText().toString().trim();
                String                                          upass                                          =
Objects.requireNonNull(inPass.getEditText()).getText().toString().trim();
                registerUser(uname, uemail, upass);
            }
        });
        return view;
    }
    private void registerUser(final String name, String email, String password){
```

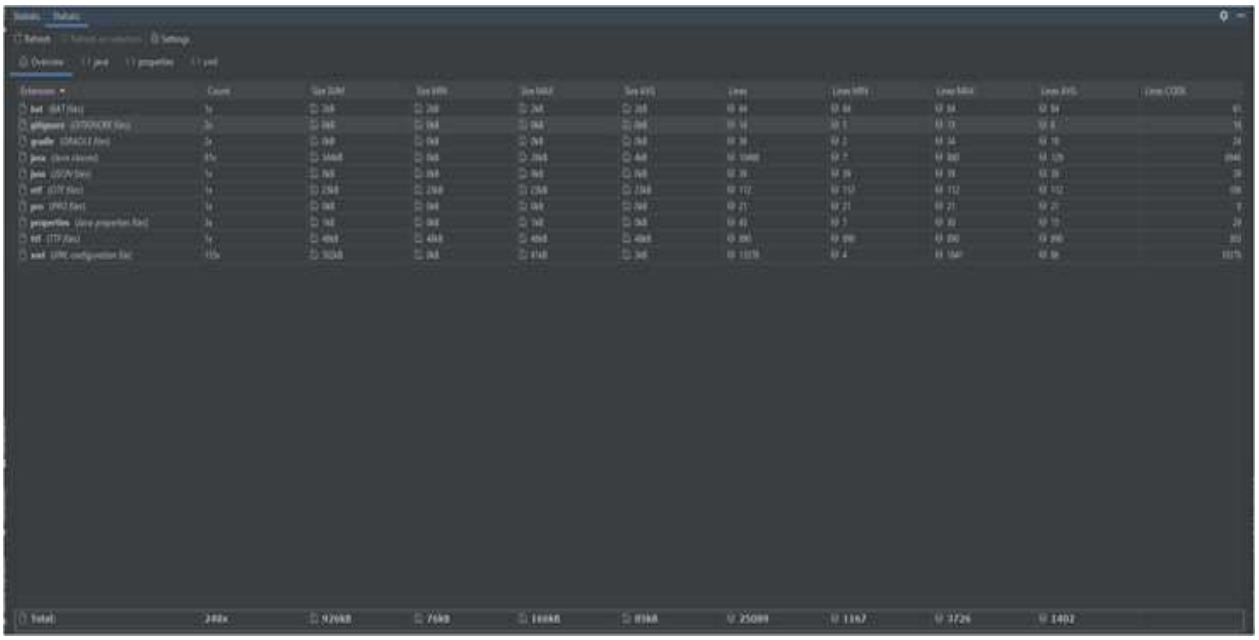
```
        progressDialog = new ProgressDialog(getActivity());
        progressDialog.setMessage("Processing your request, please wait...");
        progressDialog.show();
        fAuth.createUserWithEmailAndPassword(email, password)
                .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if (task.isSuccessful()){

fUsersDatabase.child(Objects.requireNonNull(fAuth.getCurrentUser()).getUid())
                                    .child("basic").child("name").setValue(name)
                                    .addOnCompleteListener(new OnCompleteListener<Void>() {
                                        @Override
                                        public void onComplete(@NonNull Task<Void> task) {
                                            if (task.isSuccessful()){
                                                progressDialog.dismiss();
                                                ((MainActivity)
requireActivity()).replaceFragments(LoginFragment.class);
                                                Toast.makeText(getActivity(),          "User          created!",
Toast.LENGTH_SHORT).show();
                                            } else {
                                                progressDialog.dismiss();
                                                Toast.makeText(getActivity(),       "ERROR       :       "       +
Objects.requireNonNull(task.getException()).getMessage(), Toast.LENGTH_SHORT).show();
                                            }
                                        }
                                    });
                        } else {
                            progressDialog.dismiss();
                            Toast.makeText(getActivity(),          "ERROR:          "          +
Objects.requireNonNull(task.getException()).getMessage(), Toast.LENGTH_SHORT).show();
```

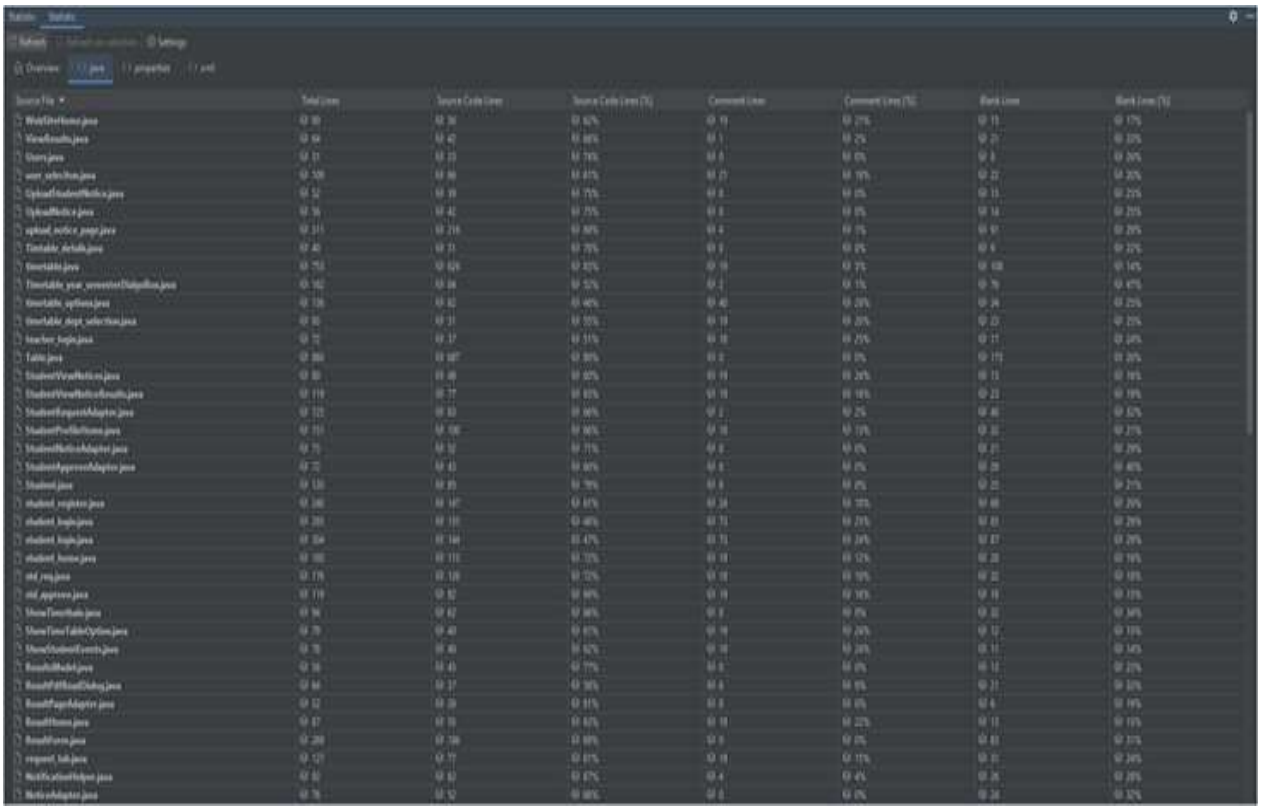
```
                    }
                }
            });
    }
    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        super.onOptionsItemSelected(item);
        switch (item.getItemId()){
            case android.R.id.home:
                ((MainActivity) requireActivity()).replaceFragments(StartFragment.class);
                break;
        }
        return true;
    }
}
```

5.2.2 Code Efficiency

Overview of the Project



Java Files



5.3 Testing Approaches

5.3.1 Unit Testing

Unit testing deals with testing a unit or module as a whole. This would test the interaction of many functions but, do confine the test within one module.

LoginFragment.java **(LC_1)**

Test Case ID	**Input Data**	**Expected Result**	**Actual Result**	**Result**
LC_1.1	Enter Email Address (e.g. parab@gmail.com) Enter password (e.g. Test123@)	Process to Login	Process to Login	Pass

LC_1.2	Enter Email Address (e.g. null or @.a.@bc) Enter password (e.g. Test123@)	Error at email address Edit Text (Please enter Email Address or Please enter correct email address)	Error at Email Address Edit Text (Please enter Email Address or Please enter correct email address)	Pass
LC_1.3	Enter Email Address (e.g. parab@gmail.com) Enter password (e.g. null or test123)	Error at password Edit Text (Please enter the password or Please enter correct password)	Error at password Edit Text (Please enter the password or Please enter correct password)	Pass
LC_1.4	Enter Email Address (e.g. null) Enter password (e.g. null)	Error at Email Address Edit Text (Please enter Email Address)	Error at Email Address Edit Text (Please enter Email Address)	Pass

RegisterFragment.java **(LC_2)**

Test Case ID	**Input Data**	**Expected Result**	**Actual Result**	**Result**
LC_2.1	Enter Name (e.g. ABC or abc)	No Errors	No Error	Pass
LC_2.2	Enter Name (e.g. null)	Error on Name Edit Text (Please Enter Name)	Error on Name Edit Text (Please Enter Name)	Pass
LC_2.3	Enter Email address (e.g.	No Error	No Error	Pass

	abc@domain_name.com)			
LC_2.4	Enter Email address (e.g. null or abc@gmail Or abcgmail.com, etc)	Error at Email address Edit Text (Please enter Email Address or Please Enter valid Email)	Error at Email address Edit Text (Please enter Email Address or Please Enter valid Email	Pass
LC_2.5	Enter Password (e.g. Test123@ or 123@Test)	No Error	No Error	Pass
LC_2.6	Enter Password (e.g. test or test123 or Test123 or Test@)	Error at password Edit Text (The Password should contain 1 uppercase, 1 lowercase, 1 numeric value, 1 special character)	Error at password Edit Text (The Password should contain 1 uppercase, 1 lowercase, 1 numeric value, 1 special character)	Pass
LC_2.7	Enter Confirm password (e.g. should be same as password)	No Error	No error	Pass
LC_2.8	Enter Confirm password (e.g. if not same as confirm password)	Error at confirm password Edit Text (Password should be same as above)	Error at confirm password Edit Text (Password should be same as above)	Pass

UpdateEmailFragment.java **(LC_3)**

Test Case ID	**Input Data**	**Expected Result**	**Actual Result**	**Result**
LC_3.1	Enter Old Email Address (parab@gmail.com)	Process to Verify	Process to Verify	Pass
LC_3.2	Enter old Email Address (e.g. null or kale@gmail.com)	Error at Email Address Edit Text(please enter your old email address or please enter correct email address)	Error at Email Address Edit Text(please enter your old email address or please enter correct email address)	Pass
LC_3.3	Enter New Email (e.g. parabgovind@gmail.com)	Proceed to Verify	Proceed to Verify	Pass
LC_3.4	Enter New Email (e.g. null or parab@m@.com)	Error at Email Address Edit Text (Please enter email address or Please enter valid email)	Error at Email Address Edit Text (Please enter email address or Please enter valid email)	Pass

ForgotPasswordFragment.java **(LC_4)**

Test Case ID	**Input Data**	**Expected Result**	**Actual Result**	**Result**
LC_4.1	Enter Email Address (e.g. +917208453787)	Process to Verify	Process to Verify	Pass

LC_4.2	Enter Email Address (e.g. null or para@b@)	Error at Email Address Edit text (please Enter Email Address or please enter correct Email Address)	Error at Email Address Edit text (please Enter Email Address or please enter correct Email Address)	Pass
LC_4.3	Enter Password on the link sent to this registered email address (e.g. 123456)	Proceed to Verify and password updated	Proceed to Verify and password updated	Pass
LC_4.4	Enter Password on the link sent to this registered email address (e.g. null)	Error (Please enter password or Please Enter valid password)	Error (Please enter password or Please Enter valid password)	Pass

PaintFragment.java **(LC_5)**

Test Case ID	**Input Data**	**Expected Result**	**Actual Result**	**Result**
LC_5.1	Draw anything	No Error	No Error	Pass
LC_5.2	Draw anything by using the provided features	No Error	No Error	Pass
LC_5.3	Allow the Permissions for All In One application (eg. allowed)	No Error	No Error	Pass
LC_5.4	Allow the Permissions for All	No Error	No Error	Pass

	In One application (eg. deny)			
LC_5.5	Save as Image to internal storage (eg permission accepted by user)	No Error	No Error	Pass
LC_5.6	Save as Image to internal storage (eg permission denied by user)	Dispalying the required permissions	Dispalying the required permissions	Pass

TextToPdfFragment.java **(LC_6)**

Test Case ID	**Input Data**	**Expected Result**	**Actual Result**	**Result**
LC_6.1	Type anything and create the pdf of it.	No Error	No Error	Pass
LC_6.2	Type anything by using all features provided.	No Error	No Error	Pass
LC_6.3	Allow the Permissions for All In One application (eg. allowed)	No Error	No Error	Pass
LC_6.4	Allow the Permissions for All In One application (eg. deny)	No Error	No Error	Pass

LC_6.5	Save as PDF to internal storage (eg permission accepted by user)	No Error	No Error	Pass
LC_6.6	Save as PDF to internal storage (eg permission denied by user)	Dispalying the required permissions	Dispalying the required permissions	Pass

ChatroomListFragment.java **(LC_7)**

Test Case ID	**Input Data**	**Expected Result**	**Actual Result**	**Result**
LC_7.1	Display Available Rooms	No Error	No Error	Pass
LC_7.2	Create Any number of Rooms	No Error	No Error	Pass
LC_7.3	Create Room with already existing room name	Error in Toast Message room alredy exists	Error in Toast Message room alredy exists	Pass

ChattroomChatFragment.java **(LC_8)**

Test Case ID	**Input Data**	**Expected Result**	**Actual Result**	**Result**
LC_8.1	Display chats if network connection is available.	No Error	No Error	Pass
LC_8.2	Display User Names	No Error	No Error	Pass

LC_7.3	Display chats if network connection is not available.	No Error and nothing to display	No Error and nothing to display	Pass

AnonymousChatroomFragment.java **(LC_9)**

Test Case ID	**Input Data**	**Expected Result**	**Actual Result**	**Result**
LC_9.1	Display chats if network connection is available.	No Error	No Error	Pass
LC_9.2	Display User As Anonymous	No Error	No Error	Pass
LC_9.3	Display chats if network connection is not available.	No Error and nothing to display	No Error and nothing to display	Pass

5.3.2 Integration testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.

Integration testing bring all the modules together into a special testing environment, the checks for errors, bugs and interoperability. It deals with tests for the entire application. Application limits and feature are tested here.

Test Case ID	**Description**	**Input Data**	**Expected Result**	**Actual Result**	**Result**
TC_1	Login Login into the System by entering correct	Enter Email address and Password	Open Home Page	Open Home Page	Pass

	details				
TC_2	Register Details To Register	Click on Register button from StartFragment page	Open Register Page	Open Register Page	Pass
TC_2.1	Registering User To Store the details in the Database	Enter the details of the user like name, email, password and confirm password	Open Home Page	Open Home Page	Pass
TC_3	Forgot Password To enter the new password and login Again	Click on Forgot password button from Login page	Open ForgotpasswordFragment Page	Open ForgotpasswordFragment Page	Pass
TC_3.1	Changing Password To update the user password	Enter the new password	Show successful changed password and open Login page	Show successful changed password and open Login page	Pass
TC_4	TextToPdfFragment to allow user to make pdf of any input text	Click on any provided features and create the pdf	Saved it as PDF to internal storage	Saved it as PDF to internal storage	Pass
TC_5	PaintFragment	Click on any	Saved it as	Saved it as	Pass

	to allow user to draw anything	provided paint features and saved it as image	image in internal storage	image in internal storage	
TC_6	Edit User Details To update the details of the User	Enter the updated Details	Successfully store in database and Show in Name in Home	Successfully store in database and Show in Name in Home	Pass
TC_7	ChatroomListFragment to show all available rooms	Click on the any room and start chatting.	Display user chats along with their names	Display user name along with their names	Pass
TC_8	AnonymousChatFragment to allow all users to chat in room as one to one by keeping data confidential	Start chatting with all users in room	Display all user chats by keeping their data confidential	Display all user chats by keeping their data confidential	Pass

5.3.3 System Testing

System testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. It includes delay, usability, security, robustness.

Delay

The Application takes approximately 2 seconds to respond from the Database if there is a proper Network Connection. All the features are within the range of 2 to 5 seconds to respond.

Usability

The usability includes how the application is user friendly for the user i.e. is it easy to

understand. All In One Application is very user friendly it is not complex to use and is not very complex to understand for the user to perform any action. There are Icons used for better understanding.

Security

In Security how secure the data is of the user.

The user email address is verified either by their google account or through verification link which is sent to their register email address. The user is also allowed to update their email or password.

The user gets an OTP for registration and for changing the password if the password has forgotten by the user to validate that the Mobile number that the user is using belongs to valid user or not.

The user cannot login into others account without a valid password.

Robustness

The Robustness includes the performance and the crashing of the application.

All In One Application is not crashing frequently. The system operates correctly in the presence of exceptional inputs or stressful environmental conditions. Only when the user has started the service for background and the application is removed from the recent applications list then after some time the application crashes internally which does not affect the user experience.

Chapter 6

Results and Discussions

6.1.1 Informal Testing

Informal Testing is used to see the initial error that was faced was resolved or not

There were many problems faced initially some of that are:

1. The notification is delay with 1 minute sometimes, so to solve this issue I have made the priority of every notification to high. This consumes battery life but works perfectly.
2. While creating the chat room it was difficult to handle the data in firebase realtime database. So to solve the issue I have used the java collections map interface and hashmap class.
3. While doing the testing on real device for the rich edit text features the device sometime get hanged. So to solve this problem I have created an array of objects of SpamableString.

6.1.2 Formal Testing

In Formal testing the summary of the test cases is shown in chapter 5.

File Name	Number of test cases	Success	Failure
LoginFragment.java	4	4	0
RegisterFragment.java	8	8	0
UpdateEmailFragment.java	4	4	0
TextToPdfFragment.java	6	6	0
PaintFragment.java	6	6	0
ChatFragment.java	3	3	0
AnonymousChatFragment.java	3	3	0
ChatroomListFragment.java	3	3	0

Summary of Integration Testing

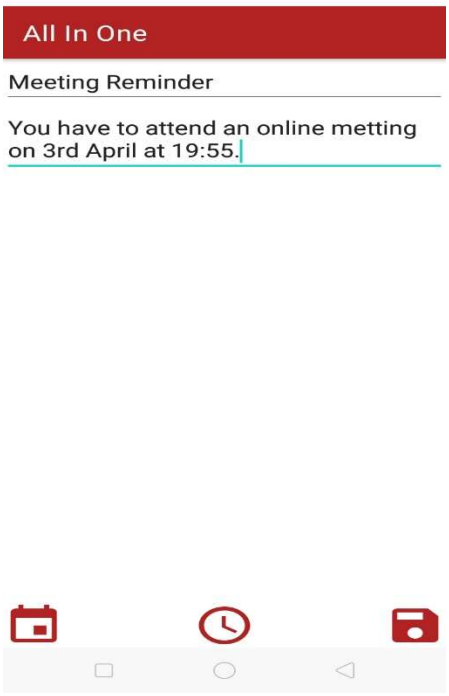
Total Number of Files	Number of test cases	Success	Failure
8	**37**	**37**	**0**

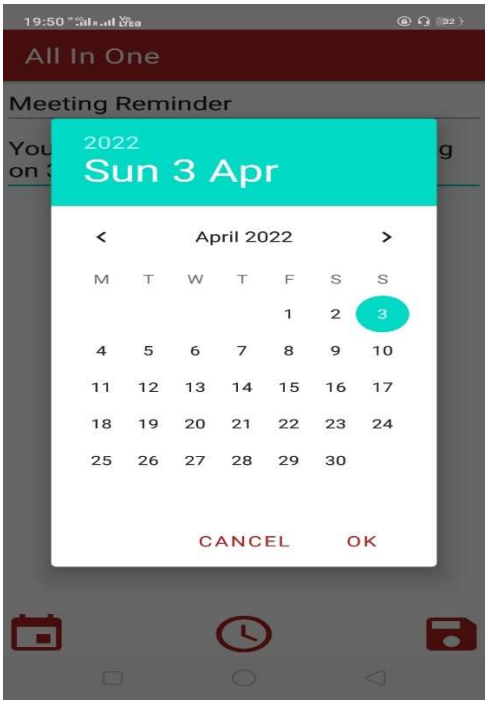
Conclusion of Testing

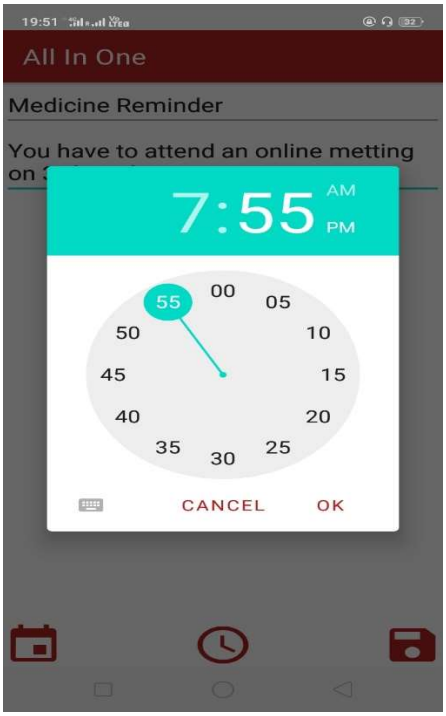
Since the Testing was done manually the testing took more time. If Automatic testing was performed then the time took for testing would have been less. Testcase will be more refined and effective, accurate. The tool will generate the test case automatically and the user does not have

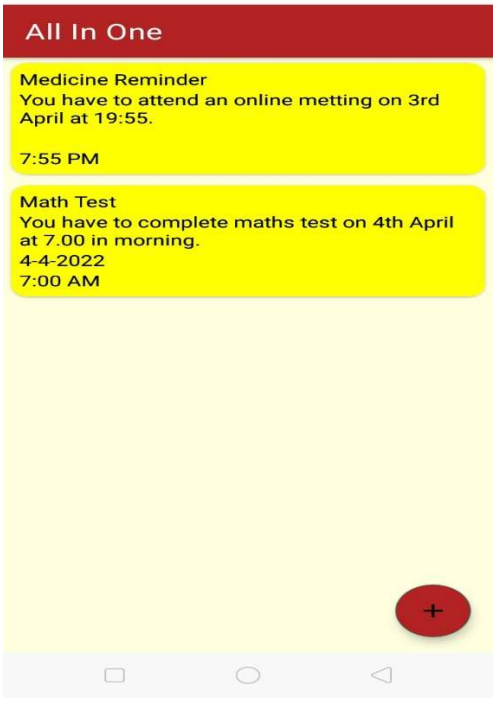
to check again and again if there is a change in the code the testing would have been performed by the tool.

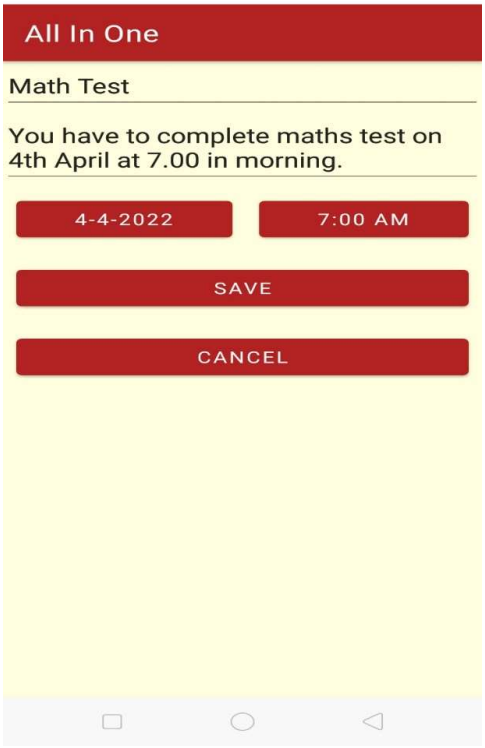
6.2 User Documentation

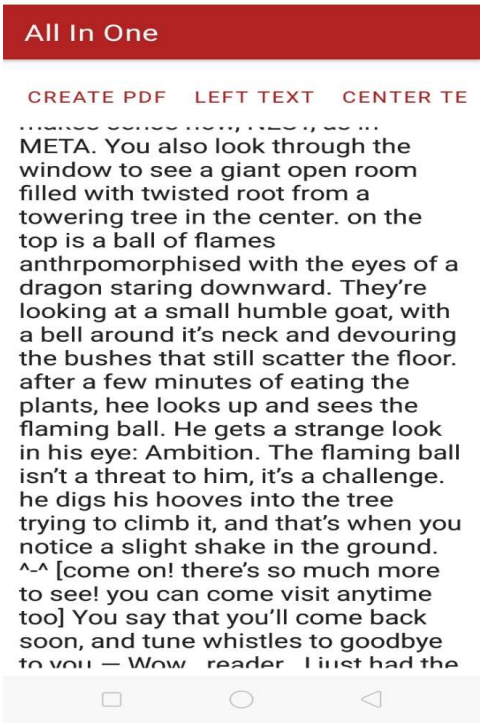


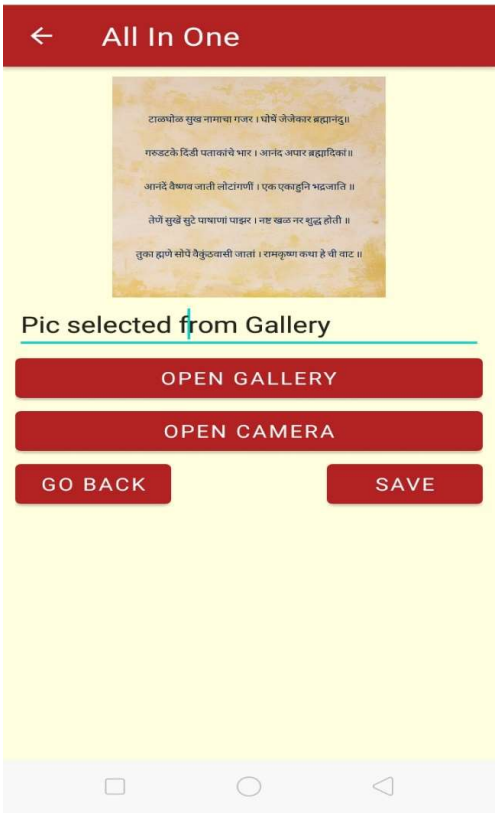


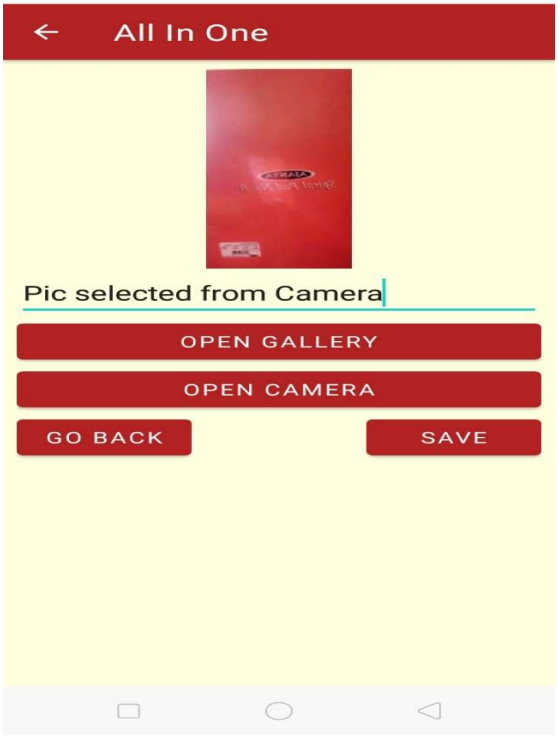


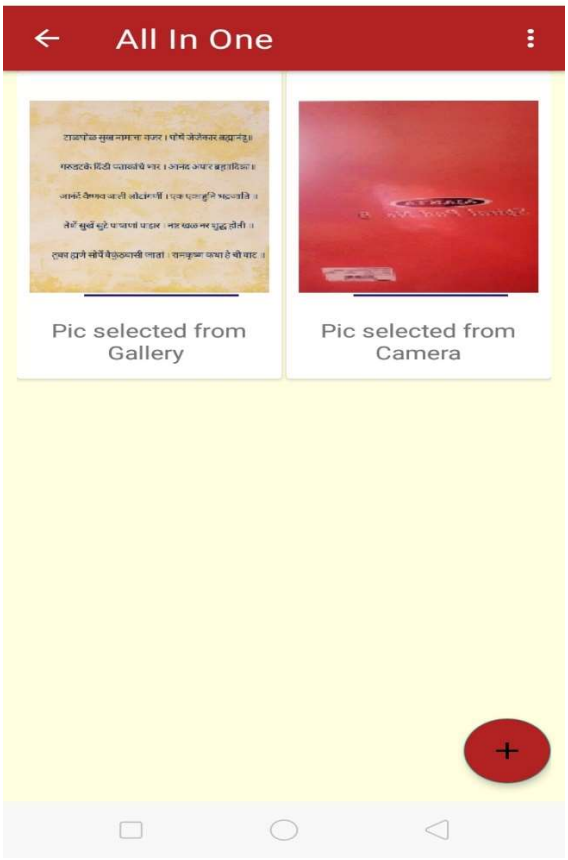


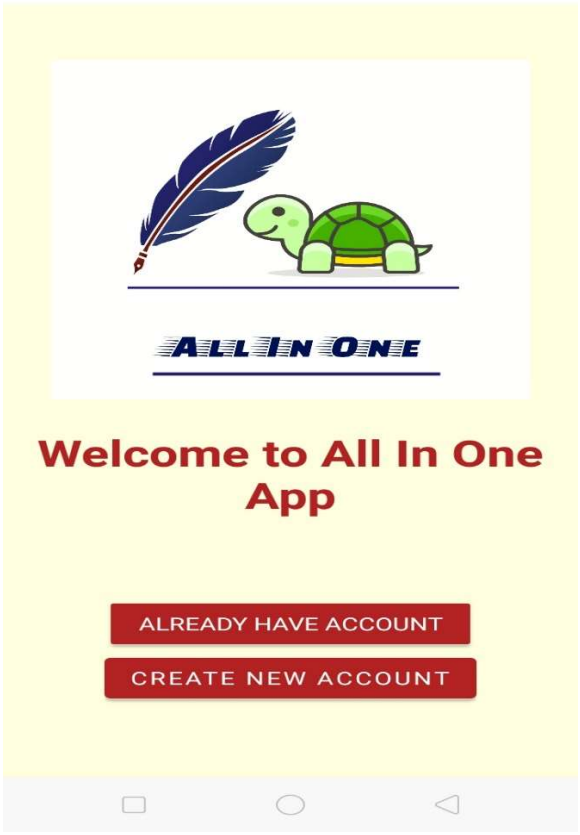

All In One
Math Test
You have to complete maths test on 4th April at 7.00 in morning.
4-4-2022
7:00 AM
SAVE
CANCEL


All In One
CREATE PDF
LEFT TEXT
CENTER TE
META. You also look through the window to see a giant open room filled with twisted root from a towering tree in the center. on the top is a ball of flames anthrpomorphised with the eyes of a dragon staring downward. They're looking at a small humble goat, with a bell around it's neck and devouring the bushes that still scatter the floor. after a few minutes of eating the plants, hee looks up and sees the flaming ball. He gets a strange look in his eye: Ambition. The flaming ball isn't a threat to him, it's a challenge. he digs his hooves into the tree trying to climb it, and that's when you notice a slight shake in the ground. ^-^ [come on! there's so much more to see! you can come visit anytime too] You say that you'll come back soon, and tune whistles to goodbye

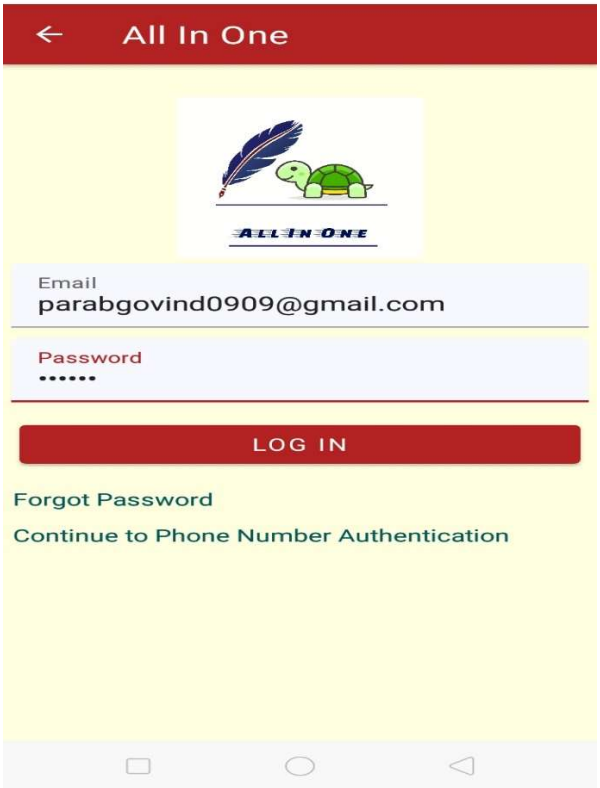

All In One
टाळघोळ सुख नामाचा गजर । घोषें जेजेकार ब्रह्मानंदु ॥
गरुडटके दिडी पताकांचे भार । आनंद अपार ब्रह्मादिकां ॥
आनंदें वैष्णव जाती लोटांगणीं । एक एकातुनि भद्रजाति ॥
तेणें सुखें सुटे पाषाणां पाझर । नष्ट खळ नर शुद्ध होती ॥
तुका ह्मणे सोपें वैकुंठवासी जातां । रामकृष्ण कथा हे ची वाट ॥
Pic selected from Gallery
OPEN GALLERY
OPEN CAMERA
GO BACK
SAVE

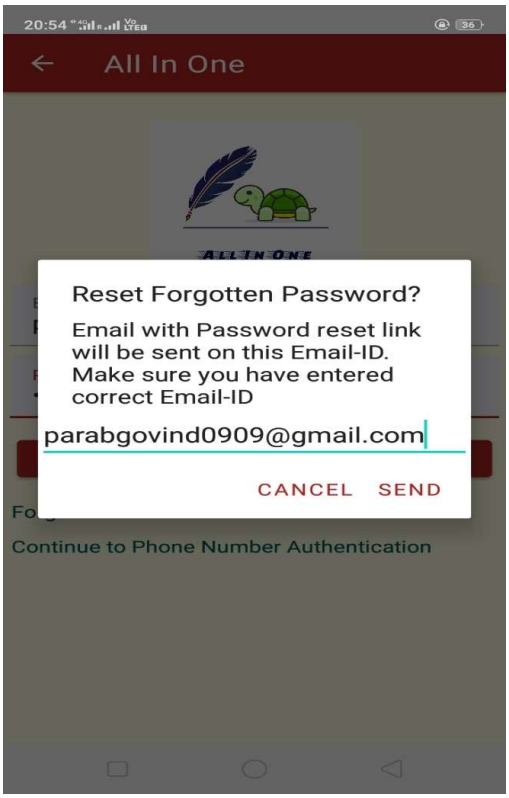

All In One
Pic selected from Camera
OPEN GALLERY
OPEN CAMERA
GO BACK
SAVE

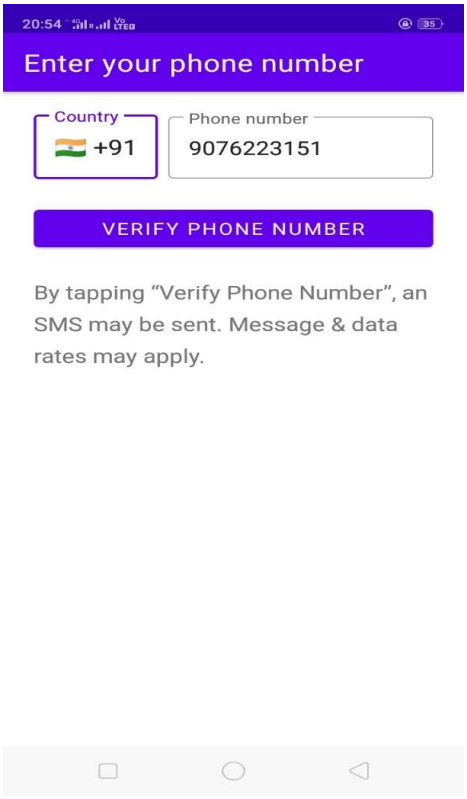

All In One
Pic selected from Gallery
Pic selected from Camera

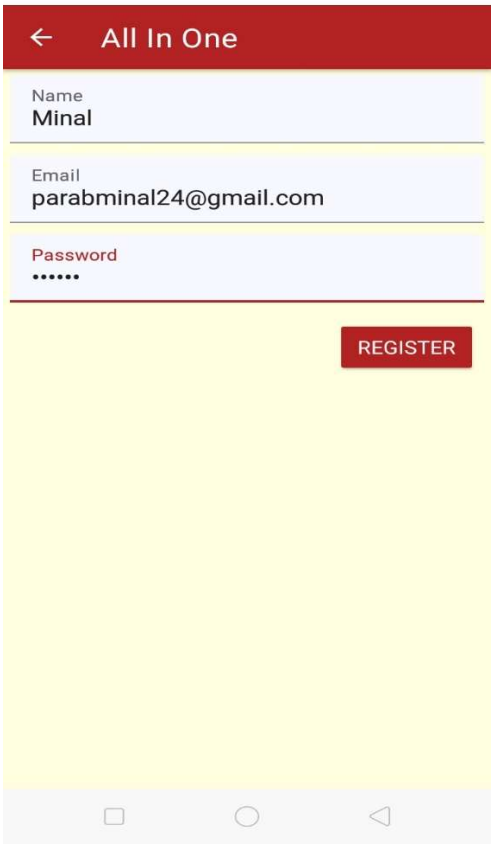

ALL IN ONE
Welcome to All In One App
ALREADY HAVE ACCOUNT
CREATE NEW ACCOUNT


ALL IN ONE

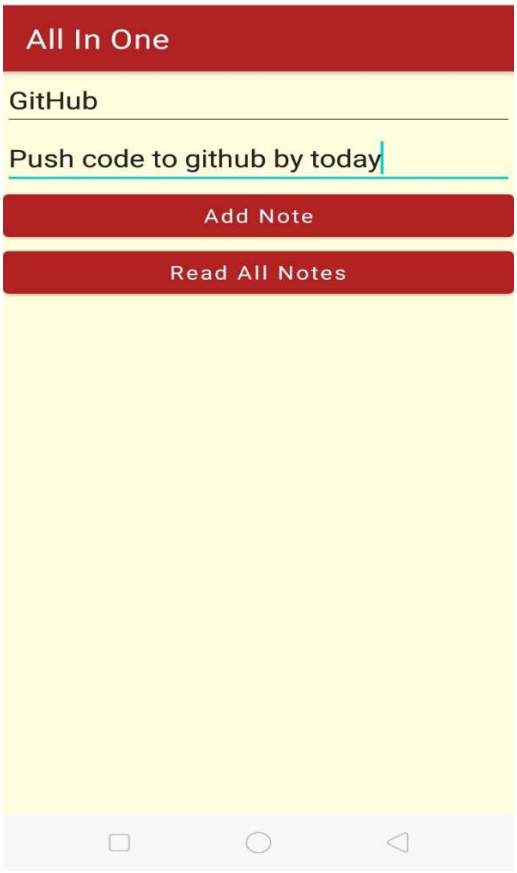

All In One
ALL IN ONE
Email
parabgovind0909@gmail.com
Password
LOG IN
Forgot Password
Continue to Phone Number Authentication

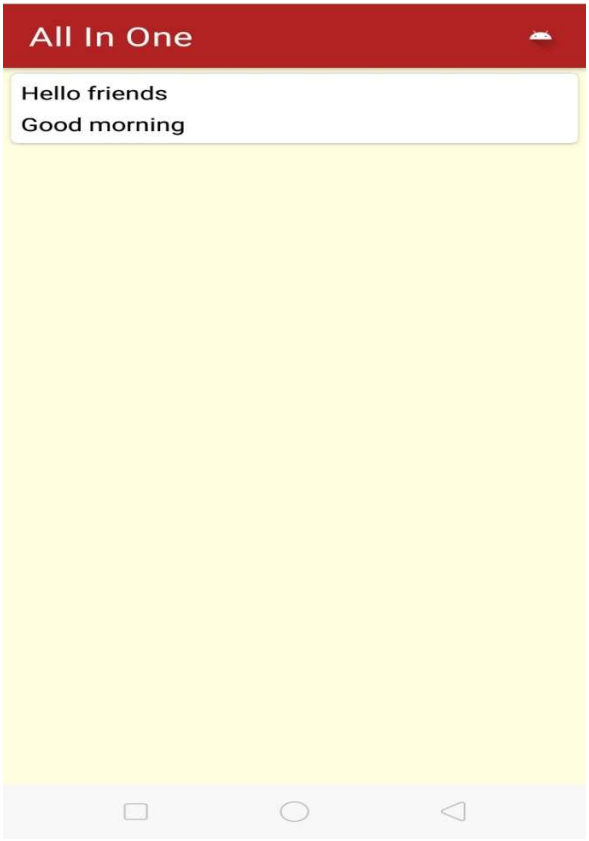

20:54
All In One
Reset Forgotten Password?
Email with Password reset link will be sent on this Email-ID. Make sure you have entered correct Email-ID
parabgovind0909@gmail.com
CANCEL
SEND
Continue to Phone Number Authentication


20:54
Enter your phone number
Country
+91
Phone number
9076223151
VERIFY PHONE NUMBER
By tapping "Verify Phone Number", an SMS may be sent. Message & data rates may apply.


All In One
Name
Minal
Email
parabminal24@gmail.com
Password
REGISTER


All In One
This is my sticky note


All In One
GitHub
Push code to github by today
Add Note
Read All Notes


All In One
Hello friends
Good morning

CHAPTER 7
CONCLUSIONS

7.1 Conclusion

The project turned out to be a valuable experience in the software development. The theory of how the development should be carried out and the things that can go wrong in the development process were experienced first-hand. Many errors had occurred in the process of the development from which many new things were introduced to us. The challenges that were encountered and the fact that we learned that the development of the software is not an easy task grew as the time passes.

The core functionality of creating, editing, updating and fetching of notes in both online and offline mode were successfully implemented in the final prototype of the Notes Maker App. The other functionalities like converting text to pdf, creating sticky notes, providing print features to the user where successfully implemented. PDF created by user or drawing using paint functionality user were able to successfully store the files in user device. The other core features like creating universal rooms so that users of this application can share their knowledge according to the purpose of rooms created was implemented successfully. Also the replies from different users were successfully uploaded on all users' devices along with the replier's name. The one to one chatting feature was successfully implemented allowing all the users to reply with only one name to keep their messages anonymous. All other basic features like login or registering the users were successfully implemented. The user is also able to upload their email or passwords. The feature immutable diary is also successfully implemented in the final stage of the application.

It was unfortunate that the certain editor features could not be implemented as it was planned. The main reason for this was there were many challenges faced in implementing different features of editor by reading the documentation of android development. Much time was consumed in learning the Android studios and firebase service which still has much to be discovered by us. Another cause was the underestimation of the complexity and the time for the submission of the project. The whole process of designing layouts, creating icons, formulating class and the actual implementation of the code took longer than expected.

The successful factor in the project was the iterative nature of the scrum methodology. Having

sprints planned out and iteratively implementing the features allowed fast adaption t change scope and requirements. Following the scrum methodologies ensured that a working feature was produced and any changes or add-ons to the existing feature could be handled easily.

Software Development is a complicated process and developing a full software product for a client gave valuable lessons in managing time, understanding and capturing user requirements, defining scope, estimating task completion, designing for user satisfaction, evaluating and thorough testing and mitigating the changing requirements, which will contribute to a better management of the software development.

7.2 Limitations of the System

- The Application needs continuous access of the internet to work correctly and efficiently
- The Background process when running makes the application slower.
- The user can access this application through their android devices only.
- App will not run on android version below 5.0 i.e. Lollipop.
- Some of app features won't function without an active internet connection such as creating account, logging into existing account, saving data online for making it accessible from any android device, etc.

7.3 Future Scope

Using this app the students, teachers, researchers, supervisors and others would be able to access a variety of technical and reference books at no cost. User would be able to add different departments also, so that they can easily create a group of department workers and can share their documents with them by providing various accessibility features to the group persons. The feature of coaching videos and tutorials can also be added by this a basic classroom structure with attendance taking, assignments submissions space, events notifications, exam notifications, holiday notifications, etc features can be added here. Through this app the professor, researchers, supervisors and student or anyone would be able to communicate and have group discussion which may lead to innovative ideas and projects. Exams papers, research papers, project documentations, etc from various journals can also be included in the groups. Social groups can be created so that daily updates regarding jobs, interviews, exams, news, etc. can be included. Making this application more secure, adding word dictionary, spell corrector or adding search engine can be one of the best features that can be implemented in future. Addition of basic functionalities like chatting system, mail system, video calls, audio calls, E-library system, etc.

and many more features can be added in future to make this app more feasible and easy to use. As per present scope this is only for android users, but can be made available on different platform with more features.

7.4 References

1. **Youtube Channel referred**

   Channel name “Codding in flow”

   Channel name “Fox Android”

   Channel name “Small Academy”

2. **Websites referred**

   https://developer.android.com/guide

   https://stackoverflow.com/

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