University Institute of Technology

PROJECT REPORT

ON

CHATTING APPLICATION

Submitted to

University of Kerala

In partial fulfillment of the

Award of Degree in the Bachelor of Computer Science

By

Pavin Das Y - 32021954017

UNIVERSITY INSTITUTE OF TECHNOLOGY

UNIVERSITY OF KERALA

ADOOR CENTRE

2021 - 2024

Mrs. Maya Philip	**External Examiner**	**Dr. Sajudeen P A**
Project Guide		Principal

Mrs. Lekha I

Head of the Department

MINI PROJECT

Submitted in partial fulfillment for the award of the degree of BSc. Computer Science

University of Kerala

Submitted by :

DINRAJ R	32021954012
GANESH V	32021954013
PAVIN DAS Y	32021954017

Under the Supervisor of

Mrs. Maya Philip

(Lecturer in Computer Science)

DEPARTMENT OF COMPUTER SCIENCE

UIT ADOOR

University of Kerala

ACKNOWLEDGEMENT

The elation and gratification of this project will be incomplete without mentioning all the people who helped me to make it possible, whose gratitude and encouragement were invaluable to me.
First of all, I would like to thank God, almighty, our supreme guide, for bestowing his blessings upon me in my entire endeavor.

I express my deep-felt gratitude to our project guide Mrs.Maya Philip for her sincere guidance and her help during the project work. And I am sincerely thankful to Mrs.Lekha I, Head of the department, BSc Computer Science, for her valuable suggestions.

I also thank all the teachers and all other staff members in our department who helped me directly or indirectly.

Furthermore, I would like to thank all others, especially my parents and friends. This project report would not have been a success without their valuable suggestions and moral support from them throughout the project.

Pavin Das Y
Reg No: 32021954017

CONTENTS

Chapter 1

ABOUT THE PROJECT

1.1 Abstract

Chat application is an easy platform to connect people by enabling them to stay linked to each other. In today's time messaging apps have more worldwide users than conventional social networks—which means they will play a progressively more significant role in the distribution of digital information. Flutter – a high performance framework based on Dart language is used for this purpose. It provides a high UI directly in the operating system's workspace rather than through native framework. Firebase, a next-generation app-development platform on Google Cloud, provides Backend-as-Service. It is a real-time database that permits storing a list of objects in the form of a tree form data structure. This research paper discusses the implementation of a Chat Room for android mobile phone using flutter framework. To provide security to the client the database is created using Google.

1.2 Modules

- Login/Sign up
- Send/Receive text message
- Send/Receive images
- Edit profile
- Log out

Chapter 2

SYSTEM ANALYSIS

2.1 Introduction

Chat Services have a long history with a lot of research being done in the late nineties. In 1996 at North Carolina State University, Zanin-Yost cited an exploratory service using synchronous video chat through CU-SeeMe software. On the other side, the University of Michigan Shapiro Undergraduate Library conducted a similar experiment with CUSeeMe. Many other renowned researchers in several Universities have worked on the Library system & services related to Chat software. Libraries exploring more scaled down software options for real time chat through instant messaging (IM) software such as America Online's AIM, Microsoft's MSN Messenger, and Yahoo Chat. This technology was less expensive, easy to use and many people in the target audience were using it.

Flutter framework introduced by Google, an open source UI software development kit is easy to use and also provides more security to the user.It gives lots of UI options to show creativity. Flutter by default uses dart as a language for interaction. Though a number of platforms for app development is available, the main reason for selecting Flutter is the pre-created libraries which can be used by simply importing them. Further, it gives an option of hot reload for which it only compiles that part of code which is being worked upon so it saves lots of time and CPU utilization and sometimes developer may get stuck while using other platforms for example just for changing color, it had to run the entire code again sometimes bigger in size. So with flutter even new users can have a flow of work and be very much enjoyable for them. As a database Google is used since it is one of the safest among the world. The chats of users in documents form, are stored as individual documents. And the login info is saved in authentication services of Firebase. The architecture of the Chat app is simple as it has a wall chat like structure it can be given the use of app by following example, as now we are hearing of privacy leaks even in WhatsApp and many bigger apps, so for example there is a team in a company of let's say 50 people, so if boss wants to convey message to rest of his team so what they all can do is simply create accounts, one account per user and then the boss will post the message on wall of the App, and thus other members of team can see it and reply accordingly. The purpose of the App is to aim at small companies or a

classroom to make their work easy in different domains by giving them a much simpler UI and a handy app for management.

2.2 Features

- Google Authentication
- Add Users to Chat List
- Search Users By Name or Email
- Show Users Profile Card
- Send/Receive Text Message
- Send/Receive Emojis
- Send/Receive Images
- Edit Text Message
- Copy Text Message
- Delete Message
- Email Profile Picture Used as App profile as Default
- Edit Profile Picture
- Email Name Used as App profile Name as Default
- Edit Profile Name
- Edit User About
- Log Out
- Capable for Both Android & iOS
- Minimal UI
- Easy Customization Via Flutter
- Unlimited Chat Users List
- Save Offline Caches for recent Chats

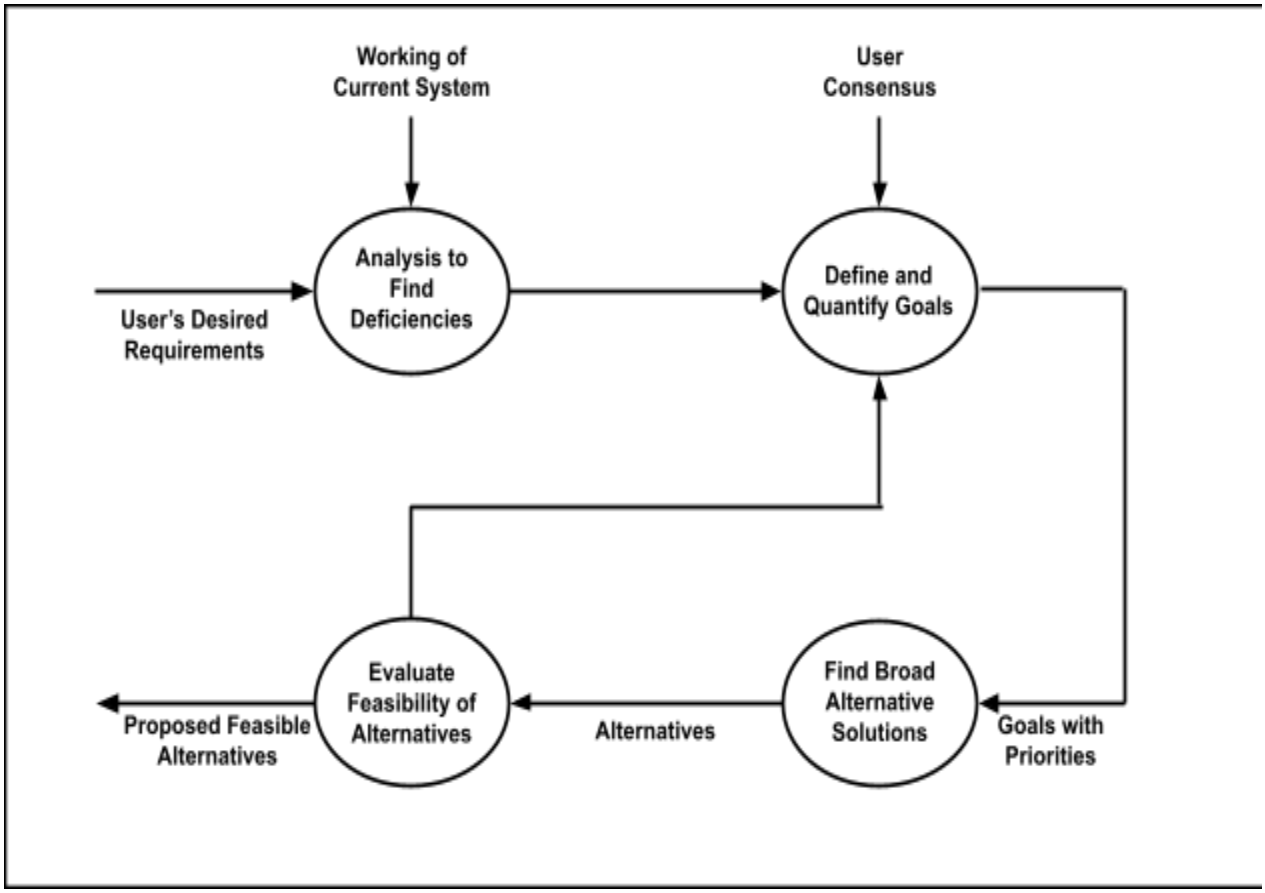
2.3 proposed System

It is an online based mostly instant electronic messaging application that provides the user to speak with different users in a very quick and convenient means each the devices should have a lively net association for the communication. There are several other chat applications like WeChat, hike, WhatsApp, Telegram, Facebook, traveler, Snap Chat, Line etc. however during this system the pdf creation and pdf reading feature are enclosed. As a region of day to day life, educational or vocation users ought to send and receive files. By victimization this technique, the user needn't have the other pdf reader-writer application on the device. Victimization this application user will communicate with any user everywhere the globe. During this application we have a tendency to ar victimization Google base of operations because the backend to store the information of the appliance like messages, pictures, files and additional. User needs to register or sign-in through their various mail id and may use the services. once the user sign-in to the appliance, user will seek for another user wherever the communication is ought to be done. The user is able to delete the chat once the communication. User will produce their profile in keeping with that different users are able to establish one another. This application is intended of humanoid itinerant users. User will reply to the messages received by simply typewriting the reply message and press the send button. This application conjointly provides the user to delete the account. User may sign-out from the current device and sigin through another.

2.4 Feasibility Study

In information system projects, feasibility is the measure ability to achieve the goal within the given constraints. A solution is feasible if it is reasonable within the circumstances. A given probelm can have multiple feasible solutions. Feasibility study means analyzing a problem to determine if it can be effectively solved within the given budgetary, operational, technical and schedule constraints in place. The output of the feasibility study determines the number of feasible solutions for the given problem. It is a continuing process of assessment of the problem. Preliminary investigations examine project feasibility, the likelihood the system will be useful to organization.

The following figure depicts the general steps in feasibility study:



The main steps in brief are :

1. Identify deficiency by pinpointing
    a. Missing functions
    b. Unsatisfactory performance
    c. Excessive cost of operations

1. Set goals to remove these deficiencies

2. Set goals to effectively meet competition

In the present case study, the main deficiencies are :

- Slow accessibility
- Does not provide full features

- Full security is not provided

The goals to remove the above mentioned deficiencies are :

- Security mechanism is provided

Economic Feasibility :

It is the measure of the total cost of the system. In the economical feasibility study we determine whether we can afford the system or not. The following questions are most commonly asked in this feasibility test

- How much will the system cost?
- Do we have the required technology?
- Do we have the required technical expertise?

The cost of a system is divided into the following three categories :

- Development coast
- Operational coast

Behavioral Feasibility :

In behavioral feasibility, the management considers that the proposed systems will fulfill the requirements that are whether the proposed system covers all the problems of the existing system. Understanding the advantages and efficiency of the proposed system the management has decided to develop a new system. The application is behaviorally feasible since it requires no technical guidance, all the 3 modules have its own functions and execute in a manner they were designed to.

Technical Feasibility :

The system must be evaluated from the technical viewpoint first. The assessment of this feasibility must be based on an outline design of the system requirement in terms of input,output,programs,procedure and staff. Having identified the outline of the system, the investigation must go on to suggest the type of equipment, required method of developing the system, and the method of running the system.

Operational Feasibility :

It is the measure of the effectiveness of the project on the routine operations of the organization. Management of the organization is heavily involved in this area of feasibility study, since management knows the complete functioning of the organization. The main questions that are answered in this study are :

- Is this problem worth solving?
- Will our existing system need modification?
- How will the project affect the existing staff?
- Will our operations be interrupted due to the project?
- Will our customers be affected due to the project?

Legal Feasibility :

A determination of any infringement, violation or liability that could result from development of the system. Legal feasibility encompasses a broad range of concerns that include contracts, liability, infringement, and myriad others traps frequently unknown to technical staff.

2.5 Project Plan

Planning is very important in every aspect of development work. Good managers carefully monitor developments at various phases. Improper planning leads to failure of the project. Software project plan can be viewed as the following:

1. Within the organization : How is the project to be implemented? What are various constraints? What is market strategy?
2. With respect to the customer : Weekly or timely meetings with the customer with presentations on status reports. Customer feedback is also taken and further modifications and developments are done. Project milestones and deliverables are also presented to the customer.

For a successful project the following steps can be followed :

Selection of project : includes identifying project's aims and objectives, understanding requirements and specification, methods of analysis, design and implementation, testing techniques and documentation.

1. Project milestones and deliverables.
2. Project estimates: including cost, time, size of code and duration.
3. Resource allocation : including hardware, software,previous relevant project information and digital library.
4. Risk management : including risk avoidance, risk detection, risk control and risk recovery.
5. Scheduling techniques : including work breakdown structure, activity graph, critical path method Gantt chart and Program Evaluation Review Technique.
6. People : including staff requirement, team management and customer interaction.
7. Quality control and standard.

Chapter 3

SYSTEM REQUIREMENT SPECIFICATION

3.1 Introduction

A software system that provides support for the development, repair and enhancement of software and for the management and control of these activities. Different environments vary in the general nature of their databases and in the coverage provided by the set of tools. In particular, some encourage (or even enforce) one specific software engineering methodology, while others provide only general support and therefore allow any of a cycle (rather than just the program development phase) and offer support for project management (rather than just technical activities). These two features normally differentiate a software engineering environment from a program development system.

3.2 Flutter

Flutter is an open source UI software development kit created by Google. It is used to develop applications for Android, iOS, Windows, Mac, Linux, Google Fuchsia and the web. The first version of Flutter was known as codename "Sky" and ran on the Android operating system. It was unveiled at the 2015 Dart developer summit, with the stated intent of being able to render consistently at 120 frames per second. During the keynote of Google Developer Days in Shanghai, Google announced Flutter Release Preview 2 which is the last big release before Flutter 1.0. On December 4, 2018, Flutter 1.0 was released at the Flutter Live event, denoting the first "stable" version of the Framework. On December 11, 2019, Flutter 1.12 was released at the Flutter Interactive event. The major components of Flutter include:

- Dart Platform
- Flutter Engine
- Foundation Library
- Design Specific widgets

Dart platform Flutter apps are written in the Dart language and make use of many of the language's more advanced features. On Windows, macOS and Linux via the semi official Flutter Desktop Embedding project, Flutter runs in the Dart virtual machine which features a just in time execution engine. While writing and debugging an app, Flutter uses Just In Time compilation, allowing for "hot reload", with which modifications to source files can be injected into a running application. Flutter extends this with support for stateful hot reload, where in most cases changes to source code can be reflected immediately in the running app without requiring a restart or any loss of state.This feature as implemented in Flutter has received widespread praise. Release versions of Flutter apps are compiled with ahead of time (AOT) compilation on both Android and iOS, making Flutter's high performance on mobile devices possible.

Flutter engine Flutter's engine, written primarily in C++, provides low level rendering support using Google's Skia graphics library. Additionally, it interfaces with platform specific SDKs such as those provided by Android and iOS.The Flutter Engine is a portable runtime for hosting Flutter applications. It impleme nts Flutter's core libraries, including animation and graphics, file and network I/O, accessibility support, plugin architecture, and a Dart runtime and compile toolchain. Most developers will interact with Flutter via the Flutter Framework, which provides a modern, reactive framework, and a rich set of platform, layout and foundation widgets. Foundation library The Foundation library, written in Dart, provides basic classes and functions which are used to construct applications using Flutter, such as APIs to communicate with the engine.Widgets UI design in Flutter involves using composition to assemble / create "Widgets" from other Widgets. The trick to Widgets UI design in Flutter involves using composition to assemble / create "Widgets" from other Widgets. The trick to understanding this is to realize that any tree of components (Widgets) that is assembled under a single understanding this is to realize that any tree of components (Widgets) that is assembled under a single build() method is build() method is also referred to as a single Widget. This is because those smaller Widgets are also made up of even smaller Widgets, and also referred to as a single Widget. This is because those smaller Widgets are also made up of even smaller Widgets, and each has a build() method of its own. This is how Flutter makes use of Composition. The docs say: " A widget is an each has a build() method of its own. This is how Flutter makes use of Composition. The docs say: " A widget is an immutable description of part of a user interface." A human being will tell you it's a Blueprint, which is a much easier way to describe an immutable description of part of a user interface." A human being will tell you it's a Blueprint, which is a much easier way to think about it. However, one also needs to keep in mind there are many types of Widgets in Flutter, and you cannot seem to think about it. However,

one also needs to keep in mind there are many types of Widgets in Flutter, and you cannot see or touch all of them. Text is a Widget, but so is its TextStyle, which defines things like size, color, font family and weight. all of them. Text is a Widget, but so is its TextStyle, which defines things like size, color, font family and weight. There are Widgets that represent things, ones that represent characteristics (like TextStyle) and even others that do things,There are Widgets that represent things, ones that represent characteristics (like TextStyle) and even others that do things, like Futurlike FutureBuilder and StreamBuilder. eBuilder and StreamBuilder. Complex widgets can be created by combining many simpler ones, and an app is actually just the largest Widget of them all Complex widgets can be created by combining many simpler ones, and an app is actually just the largest Widget of them all (often called "MyApp"). The MyApp Widget contains all the other Widgets, which can contain even smaller W(often called "MyApp"). The MyApp Widget contains all the other Widgets, which can contain even smaller Widgets, and midgets, and together they make up your app. However, the use of widgets is not strictly required to build Flutter apps. An alternative together they make up your app. However, the use of widgets is not strictly required to build Flutter apps. An alternative option, usually only used by people who like to control every pixel drawn to the canvas, is to use the Foundation library option, usually only used by people who like to control every pixel drawn to the canvas, is to use the Foundation library's methods directly. These methods can be used to draw shapes, text, and imagery directly to the canvas. This ability of methods directly. These methods can be used to draw shapes, text, and imagery directly to the canvas. This ability of Flutter has been utilized in a few frameworks, such as the openFlutter has been utilized in a few frameworks, such as the open--source Flame game engine. source Flame game engine.

3.3 Dart

Dart is an object-oriented, client-optimized programming language that is designed to be used for a variety of applications, including web, mobile, and desktop development. It was developed by Google in 2011 and has quickly become a popular choice for developers due to its simplicity, ease of use, and fast development cycle. Dart's syntax is similar to that of other popular programming languages, such as Java and C++, but it also includes features that make it well-suited for modern application development. Some of these features include optional typing, garbage collection, and an asynchronous programming model that allows developers to write code that runs efficiently on both single and multi-core processors. With the rise of Flutter, a mobile development framework also developed by Google, Dart has gained even more popularity as a powerful language for creating cross-platform mobile applications.

3.4 Firebase

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. As of March 2020, the Firebase platform has 19 products, which are used by more than 1.5 million apps. Firebase evolved from Envolve, a prior startup founded by James Tamplin and Andrew Lee in 2011. Envolve provided developers an API that enables the integration of online chat functionality into their websites. After releasing the chat service, Tamplin and Lee found that it was being used to pass application data that were not chat messages. Developers were using Evolve to sync application data such as game state in real time across their users. Tamplin and Lee decided to separate the chat system and the real time architecture that powered it. They founded Firebase as a separate company in September 2011 and it launched to the public in April 2012. Firebase's first product was the Firebase Real time Database, an API that synchronizes application data across iOS, Android, and Web devices, and stores it on Firebase's cloud. The product assists software developers in building real time, collaborative applications. In May 2012, a month after the beta launch, Firebase raised $1.1 million in seed funding from venture capitalists Flybridge Capital Partners, Greylock Partners, Founder Collective, and New Enterprise Associates. In June 2013, the company further raised $5.6 million in Series A funding from Union Square Ventures and Flybridge Capital Partners. In 2014, Firebase launched two products. Firebase Hosting and Firebase Authentication. This positions the company as a mobile backend as a service. In October 2014, Firebase was acquired by Google. A year later, in October 2015, Google acquired Divshot, an HTML5 web hosting platform, to merge it with the Firebase team. In May 2016, at Google I/O, the company's annual developer conference, Firebase introduced Firebase Analytics and announced that it was expanding its services to become a unified backend as a service (BaaS) platform for mobile developers. Firebase now integrates with various other Google services, including Google Cloud Platform, AdMob, and Google Ads to offer broader products and scale for developers. Google Cloud Messaging, the Google service to send push notifications to Android devices, was superseded by a Firebase product, Firebase Cloud Messaging, which added the functionality to deliver push notifications to both iOS and web devices. In January 2017, Google acquired Fabric and Crashlytics from Twitter to add those services to Firebase. In October 2017, Firebase launched Cloud Firestore, a real time document database as the successor product to the original Firebase Realtime Database.

Services - Google Analytics

Google Analytics is a cost free app measurement solution that provides insights on app usage and user engagement.

Firebase Authentication

Firebase Authentication is a service that can authenticate users using only client side code. It supports social login providers Facebook, GitHub, Twitter and Google as well as other service providers like Google Play Games, Apple, Yahoo, and Microsoft. Additionally, it includes a user management system whereby developers can enable user authentication with email and password login stored with Firebase.

Firebase Realtime Database

Firebase provides a real time database and back end as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored in Firebase's cloud. The company provides client libraries that enable integration with Android,iOS,JavaScript,Java,ObjectiveC,Swift and Node.js applications. The database is also accessible through a REST API and bindings for several Java Script frameworks such as AngularJS, React, Ember.js and Backbone.js. The REST API uses the Server Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Developers using the realtime database can secure their data by using the company's server side enforced security rules.

Cloud Firestore

On January 31, 2019, Cloud Firestore was officially brought out of beta, making it an official product of the Firebase lineup. It is the successor to Firebase's original database system, Real time Database, and allows for nested documents and fields rather than the tree view provided in the Real time Database.

Firebase Storage

Firebase Storage provides secure file uploads and downloads for Firebase apps, regardless of network quality, to be used for storing images, audio, video, or other user generated content. It is backed by Google Cloud Storage.

3.5 Android

Android is a multitasking mobile operating system that runs on smartphones, tablets, readers, televisions, and even domestic robots. Google purchased and marketed this operating system, which was developed by 'Android Inc' and based on Linux. Android's introduction as an operating system (OS) in 2008 was a spark, and it quickly became popular among smart gadgets. Modern smartphones and tablets might be considered stash minicomputers thanks to this operating system.

By 2020, the Android platform will have about 35 billion users worldwide, making it the most popular mobile operating system on the planet. A large number of smart gadgets run on this operating system (like phones, tablets, and smartwatches). Every company organization or brand in the world nowadays needs an e-commerce Android app in order to grow their business, money, and popularity. (Android, no date)

Android is a very adaptive and engaging system, and a basic acquaintance takes less than an hour. Because there are so many essential programs accessible, any customer may easily configure OS settings. You can modify everything past recognition: if you don't like the look, symbols, or ringtone, simply go to the Google Play Store, download a significant program, and quickly customize everything to your desire

3.6 Visual Studio Code

Visual Studio Code, often abbreviated as VS Code, is a popular and versatile code editor that has become an essential tool for developers and programmers. Developed by Microsoft, VS Code is celebrated for its open-source nature, robust feature set, and an extensive library of extensions that cater to a wide range of programming languages and workflows. With its intuitive interface, powerful debugging capabilities, and seamless integration with version control systems, Visual Studio Code has established itself as a go-to choice for software development across multiple platforms, making it a preferred code editor for both beginners and seasoned professionals in the world of coding.

3.7 FAQ

- **How to Run an Android Application in Android Studio?**
    - Install Android Studio from https://developer.android.com/studio
    - Extract the scource_code.zip. You will find this inside the main zip.

- Open the folder in your android studio.
- Even if you are building an app for ios, use android studio for the build.
- Then in your android studio terminal run:
  flutter pub get ** You need this to get all 3rd party packages from pub.dev

- **How to Configure the Launcher icon?**
  - This helps you change your app's launcher icon. Change the app_logo.png in assets folder with your own logo.Your file name should also be app_logo.png and it should be a 512x512 png image and the image format should be the same.

  - After replacing the file , uninstall your app from your emulator.Otherwise the logo will not be changed.

  - Then in your android studio terminal run:
    flutter pub get

  - Then run:
    Flutter pub run flutter_launcher_icons:main

  - Then run your app. The app will be installed again with your given launcher icon

- **How to change the package name?**
  - This is very important. Your app cannot have the same package name as other app. If it does, the playstore will not accept it as a unique application. So rename your app according to your business/brand name.Try to write a unique package name.

    Naming Convention:
    https://docs.oracle.com/javase/tutorial/java/package/namingpkgs.html

  - For example
    Let's say your package is : com.onatcipli.networkUpp
    And your app name is "Network Upp"

  - Then ,
    Run this command inside your flutter project root.
    Run the command in android studio terminal :
    futter pub run rename --bundleId com.onatcipli.networkUpp

futter pub run rename --appname "Network Upp"

    - Try uninstalling the app from the emulator , then run the commands and then restart the app.
- **How to Build the App Testing (Build an APK)?**
    - https://flutter.dev/docs/deployment/android see the doc for reference
      In terminal run : flutter build apk
    - It will build an apk and show the folder. You can then install it in your phone to test, or share to multiple users for testing.

3.7 Hardware Specifications

Processor	:	Intel i3 10th gen
Memory	:	8GB RAM
Hard Disk	:	256GB
Keyboard	:	Standard 102/105 keys
Mouse	:	Optical Mouse

3.8 Software Required

Operating System	: Windows 10 or above
Front end	: Dart, Flutter
Back end	: Dart,Google Firebase
IDE	: Visual Studio Code
Emulator	: Android Studio
Android Version	: Android 10 or above
iOS Version	: iOS 15 or above

Chapter 4

SYSTEM DESIGN

4.1 Introduction

The most creative and challenging phase of the system life cycle is the system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in implementing the candidate system. It also includes the construction of programs and program testing.

The first step in the system design is to determine how the output is to be produced and in what format. Samples of the inputs and the output are also presented. In the second step, input data and master files are to be designed to meet requirements of the proposed output. The processing phases are handled through program construction and testing, including a list of the programs needed to meet the system's objectives and complete documentation.

Finally, details related to justification of the system and an estimate of the impact of the candidate system on the user and organization are documented and implemented and evaluated by management as a step towards implementation. The final report prior to the implementation phase includes procedure flow chart, record layouts, and a workable plan for implementing the candidate system.

System design has two phases:

- Logical Design
- Physical Design

In the logical design, the designer produces a specification of the major features of the system which meets the objectives. The delivered product of logical design includes current requirements of the following system components:

- Input Design
- Output Design
- Database Design

Physical design takes this logical design blueprint and produces the program software, files and a working system.

4.2 Navigation Design

When the Chatting App arcitecture has been established and the component of the architecture has been identified, you must define navigation path way that enable user to access Chatting App content and functions.
To accomplish this you should

1. Identify the semantics of navigation for different users of the app
2. Define the mechanism of achieving the navigation

The main pages used in the system are:

Login Screen

- Login Button

Home Screen

- Search Button (using user name or email)
- Menu Button
- Add user Button (using email)
- User Profile View Icon
- Chat Users List

User Profile View Icon

- About Icon

Chat Screen

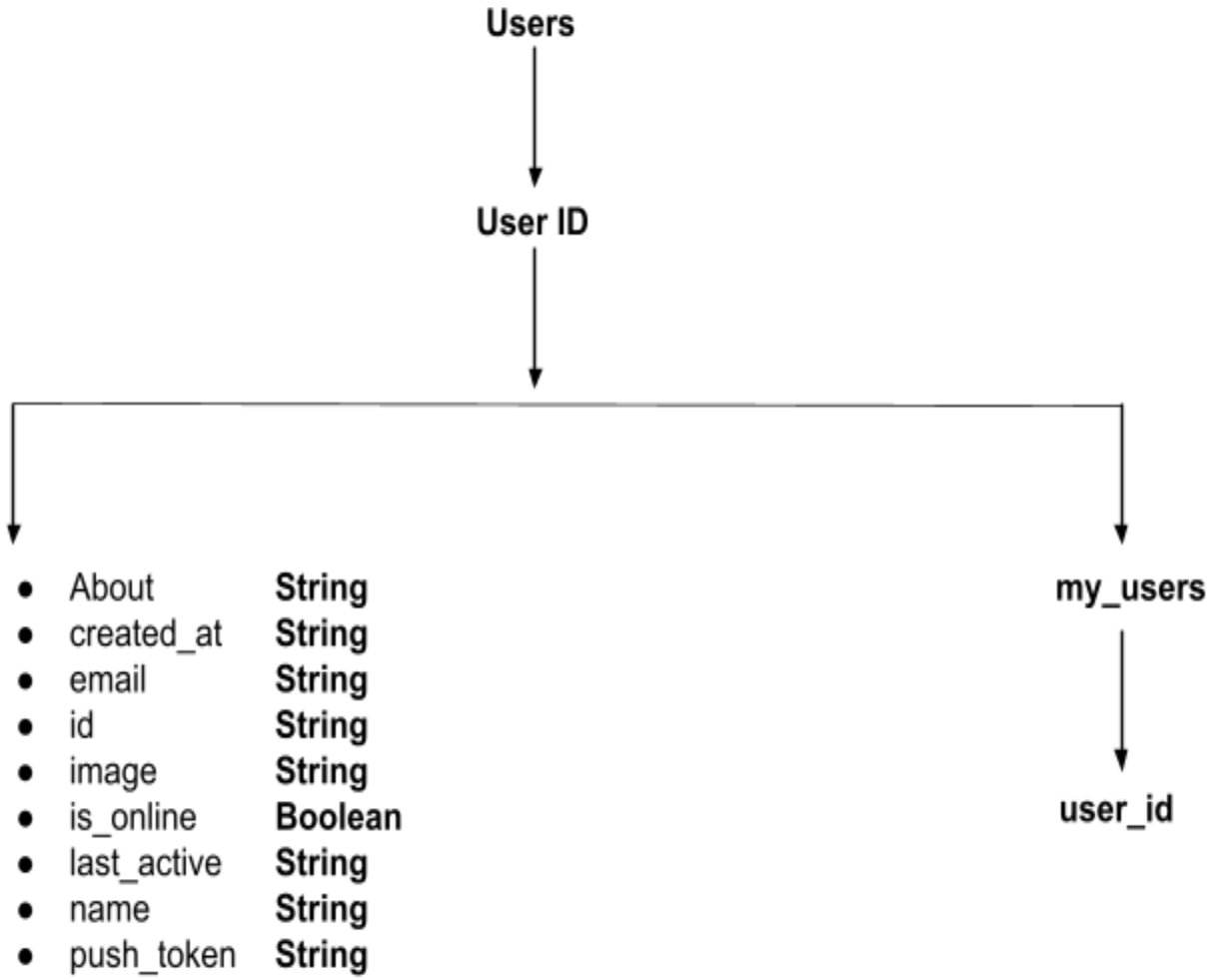
- User Profile
- Messages
- Text Input Field
- Imoji Icon
- Gallery Icon
- Camera Icon
- Send Button

Chat Screen

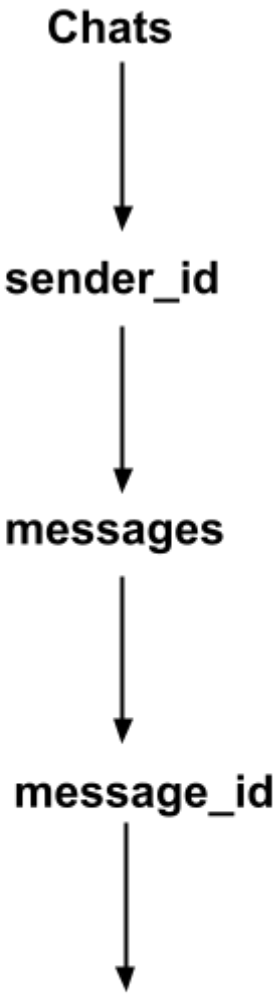
- View and Edit Profile Picture
- Email Id
- Vew and Edit User Name
- Vew and Edit User’s About
- Update profile Button
- Log Out Button

4.3 Database Design

User level



Chat level



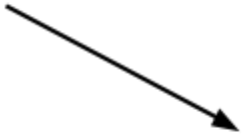
- from_id **String**
- message **String**
- read **String**
- sent **String**
- to_id **String**
- type **String**

4.4 Data Flow Diagram (DFD)

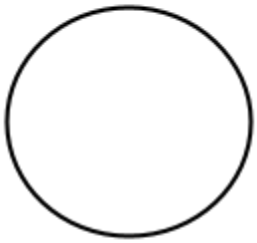
Analysis models help us to understand the relationship between different components in the system design. Analysis model shows the user clearly how a system will function. This is the first technical representation of a system. The analysis modeling must achieve three primary objectives.

Components:

Square, This defines source or destination of data



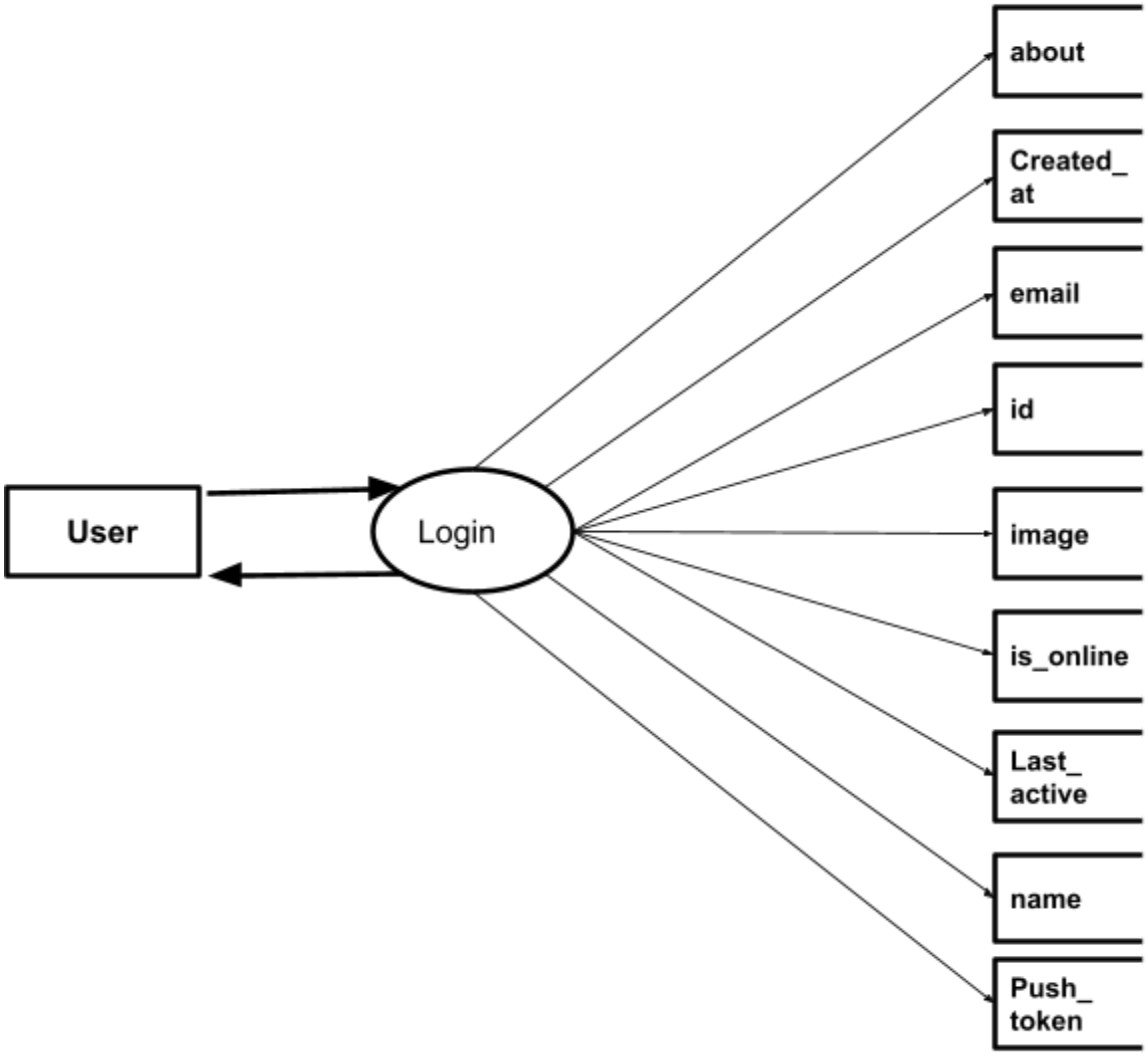
Arrow, Which shows data flow



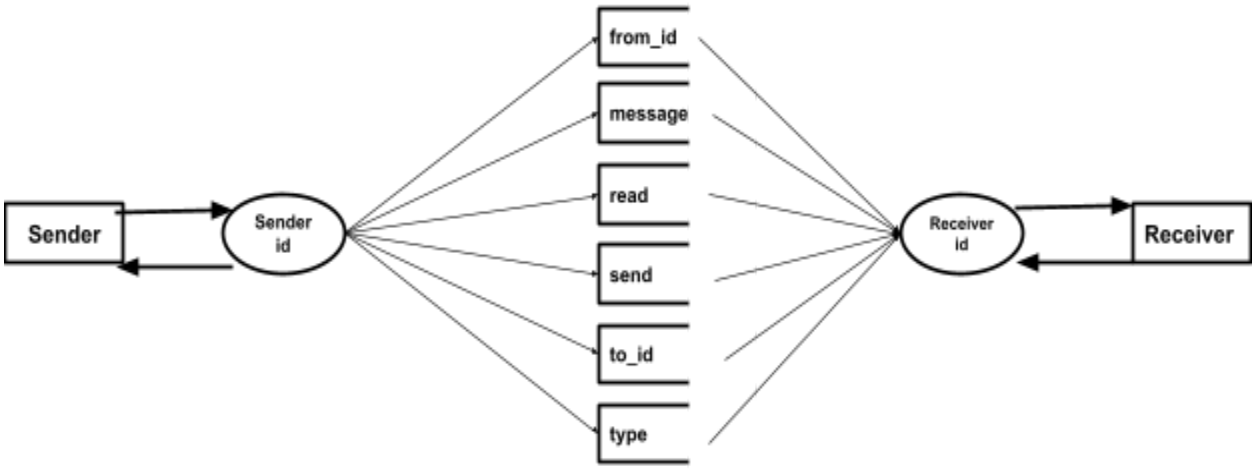
Circle, Which represents a process that transforms incoming data into outgoing flow

Open Rectangle, Which shows a data store

Level 1 DFD


User
Login
about
Created_
at
email
id
image
is_online
Last_
active
name
Push_
token

Level 2 DFD


Sender
Sender id
from_id
message
read
send
to_id
type
Receiver id
Receiver

Chapter 5

SYSTEM TESTING

Testing is the last stage of the software development before we release the product to the customer. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Software testing can be looked upon as one among the many processes. Testing cannot show the absence of defects, it can only show that software defects are present.

Software Testing Techniques

The importance of testing and its impact on software cannot be underestimated. The greater visibility of software systems and the cost associated with the software failure are motivating factors for planning through testing. It is not uncommon for a software organization to spend 40% of its effort on testing.

A number of rules that act as Testing Objectives are

Testing is a process of executing a program with the aim of finding errors.

A good test case will have a good chance to find an undiscovered error. System testing can be broadly classified into:

i. Black box testing

ii. White box Testing

iii. Unit Testing

iv. Integration Testing

v. Validation Testing

- **Black Box Testing**

Black Box Testing is not a type of testing it instead is a testing strategy, which does not need any knowledge of internal design or code etc. As the name “black box” suggests, no knowledge of internal logic or code structure is required. The types of testing under this strategy are totally based / focused on the testing for requirements and functionality of the work product/software application.

Black box testing is sometimes also called “Opaque testing “, “Functional/Behavioral tests” and “Closed Box testing”. The base of the Black box testing strategy lies in the selection of appropriate data as per functionality and testing it against the functional specifications in order to check for normal and abnormal behavior of the system.

- **White Box Testing**

White box testing strategy deals with the internal logic and structure of the code. White box testing is also called glass, structural, open box or clears box testing. The tests written based on the white box testing strategy incorporate coverage of the code written, branches, paths, statements and internal logic of the code etc.

In order to implement white box testing, the tester has to deal with the code and hence is needed to possess knowledge of coding and logic i.e., internal working of the code. White box test also needs the tester to look into

the code and find out which unit/ statement/ chunk of the code is malfunctioning

- **Unit Testing**

The first level testing is unit testing. Unit testing concentrates on each unit if the software is implemented in source code. Initially tests focus on each module individually, ensuring that it functions properly as a unit. The modules must then be assembled or integrated to form the complete software package.

There are tests that occur as part of unit testing. The module interfaces are tested to ensure that information properly flows into and out of a program under test. The data structures are also tested for integrity. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing.All independent paths through the control structures are exercised to ensure that all statements in a module have been executed at least once. Finally all error handling is tested.

- **Integration Testing**

The next level of testing is often called as Integration testing in which many tested modules are combined into sub-system, which are then tested. Data can be lost across an interface; one module can have an adverse effort on the other sub functions, when combined may not produces the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors

within the interface. This testing was done with sample data. The developed system has run successful for this sample data. The need for integrated test is to find the overall system performance

· **Validation Testing**

Data Validation is the process of testing the accuracy of data; a set of rules you can apply to a control to specify the type and range of data that users can enter. It can be used to display error alerts when users enter incorrect values into a form. Rather than checking for errors after a form is completed, data validation verifies values as the form is being filled out.

A strategy for software testing integrates software test case design method in to a well- planned series of steps that result in the successful construction of the software. The strategy provides a road map that describes the step to be conducted as part of testing, when these steps are planned and then undertaken, and how much effort, time and resources will be required. Therefore any testing strategy must incorporate test planning, test case, design, test execution and resultant data collection and evaluation. A software testing strategy should be flexible enough to promote a customized testing approach. At the same time, it must be rigid enough to promote reasonable planning and management tracking as the project progress. The project manager, software engineers and testing specialists develop a strategy for software testing.

Chapter 6

SYSTEM IMPLEMENTATION AND MAINTENANCE

Implementation is the process of converting a new or revised system design into operation. It is the key stage in achieving a successful new system because, usually it reveals a lot of up heal. It must therefore be carefully planned and controlled. The system will help the various government departments to co-ordinate and control the migrant in an efficient and effective manner .Once webmaster is aware of it, the system can be tested. Implementation is the stage of the project where the theoretical design is turned into working system or it is the key stage in achieving a successful new system. Therefore it must be carefully planned and controlled. It can also be considered to be the most crucial stage in achieving a successful new system and in giving the user confidence that the new system will work and be effective.

Implementation is the final and important phase. It is the phase where theoretical design is turned into working system, which works for the user in the most effective manner. It involves careful planning, investigation of the present system and the constraints involved, user training, system testing and successful running of developed proposed system. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system. The user tests the developed system and changes are made according to their needs. The testing phase involves the testing of a system using various kinds of data. This method also offers the greatest security since the old system can take

over if the errors are found or inability to handle certain type of transactions while using the new system.

Implementation involves following tasks:

- Careful planning
- Investigation of system and constraints
- Design of methods to achieve the changeover
- Evolution of changeover method

Chapter 7

PROJECT SOURCE CODE

Main Class

```
import 'package:chatzone/screens/screen_splash.dart';
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import
'package:flutter_notification_channel/flutter_notification_channel.dart';
import
'package:flutter_notification_channel/notification_importance.dart';
import 'firebase_options.dart';
import 'dart:developer';

//* Global object to get device screen size
late Size mq;

void main() {

  WidgetsFlutterBinding.ensureInitialized();
  _initFirebase();
  runApp(
    const MyApp(),
  );
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
```

```
      title: 'ChatZone',
      theme: ThemeData(
        primarySwatch:  Colors.deepPurple,
        appBarTheme: const AppBarTheme(
          centerTitle: true,
          elevation: 1,
          titleTextStyle: TextStyle(
            fontWeight: FontWeight.w700,
            fontSize: 23,
          ),
        ),
      ),
      home: const ScreenSplash(),
    );
  }
}

//* Firebase initialize in flutter
_initFirebase() async {
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );

  var result = await
FlutterNotificationChannel.registerNotificationChannel(
    description: 'For Showing Message Notification',
    id: 'chats',
    importance: NotificationImportance.IMPORTANCE_HIGH,
    name: 'Chats',

);
log('\n Notification Channel result : $result');
}
```

Google Authentication

```dart
import 'dart:developer';
import 'dart:io';
import 'package:chatzone/api/apis.dart';
import 'package:chatzone/helper/dialogs.dart';
import 'package:chatzone/main.dart';
import 'package:chatzone/screens/screen_home.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_sign_in/google_sign_in.dart';

class ScreenLogin extends StatefulWidget {
  const ScreenLogin({super.key});

  @override
  State<ScreenLogin> createState() => _ScreenLoginState();
}

class _ScreenLoginState extends State<ScreenLogin> {
  //* Actions for LogIn button
  _handleGoogleButtonClick() {
    Dialogs.showProgressBar(context);
    _signInWithGoogle().then(
      (user) async {
        Navigator.pop(context);
        if (user != null) {
          log('\nuser: ${user.user}');
          log('\nuserAdditionalInfo: ${user.additionalUserInfo}');

          if ((await APIs.userExist())) {
            Navigator.pushReplacement(
              context,
              MaterialPageRoute(
                builder: (context) {
                  return const ScreenHome();
                },
              ),
            );
```

```
          } else {
            await APIs.createUser().then(
              (value) {
                Navigator.pushReplacement(
                  context,
                  MaterialPageRoute(
                    builder: (context) {
                      return const ScreenHome();
                    },
                  ),
                );
              },
            );
          }
        }
      },
    );
  }

  //* Firebase authenticator auto generated method
  Future<UserCredential?> _signInWithGoogle() async {
    try {
      await InternetAddress.lookup('google.com');
      // Trigger the authentication flow
      final GoogleSignInAccount? googleUser = await
GoogleSignIn().signIn();

      // Obtain the auth details from the request
      final GoogleSignInAuthentication? googleAuth =
          await googleUser?.authentication;

      // Create a new credential
      final credential = GoogleAuthProvider.credential(
        accessToken: googleAuth?.accessToken,
        idToken: googleAuth?.idToken,
      );

      // Once signed in, return the UserCredential
      return await APIs.auth.signInWithCredential(credential);
    } catch (e) {
```

```
      log('\n_signInWithGoogle: $e');
      Dialogs.showSnackbar(
        context,
        'Something went wrong. Check internet connection',
      );
      return null;
    }
  }

  @override
  Widget build(BuildContext context) {
    mq = MediaQuery.of(context).size;

    return Scaffold(
      backgroundColor: Colors.indigo[50],
      //* AppBar
      appBar: AppBar(
        backgroundColor: Colors.deepPurple,
        //* App Title
        title: const Text('Welcome to ChatZon'),
      ),
      body: Stack(
        children: [
          //* App Icon
          Positioned(
            top: mq.height * .15,
            left: mq.width * .26,
            width: mq.width * .5,
            child: Image.asset('assets/images/icon.png'),
          ),

          //* Google LogIn icon
          Positioned(
            bottom: mq.height * .15,
            left: mq.width * .1,
            width: mq.width * .8,
            height: mq.height * .07,
            child: ElevatedButton.icon(
              onPressed: () {
                _handleGoogleButtonClick();
```

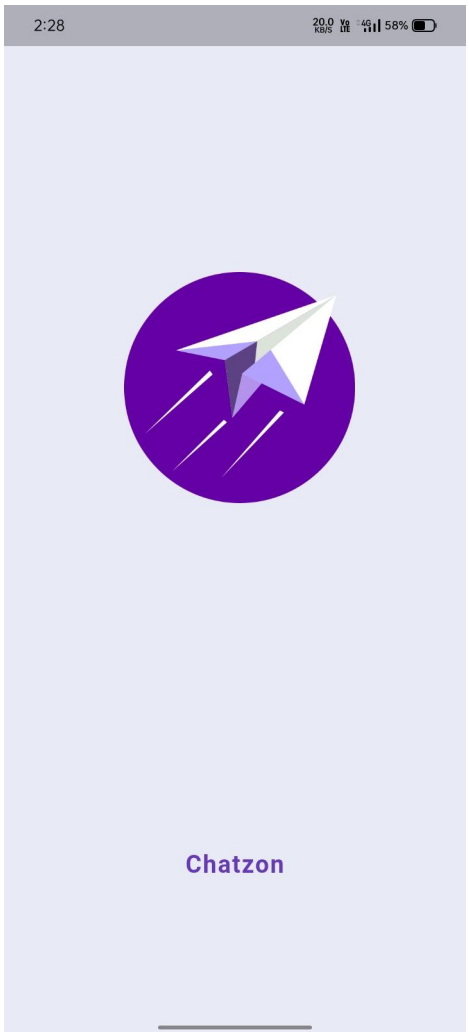
```
              },

              //* Google png logo
              icon: Padding(
                padding: const EdgeInsets.all(15),
                child: Image.asset('assets/images/google.png'),
              ),
              label: RichText(
                text: const TextSpan(
                  style: TextStyle(fontSize: 16),
                  children: [
                    TextSpan(text: 'LogIn with '),
                    TextSpan(
                      text: 'Google',
                      style: TextStyle(
                        fontWeight: FontWeight.w700,
                      ),
                    ),
                  ],
                ),
              ),
              style: ElevatedButton.styleFrom(
                shape: const StadiumBorder(),
                elevation: 1,
                backgroundColor: Colors.deepPurple,
              ),
            ),
          ),
        ],
      ),
    );
  }
}
```

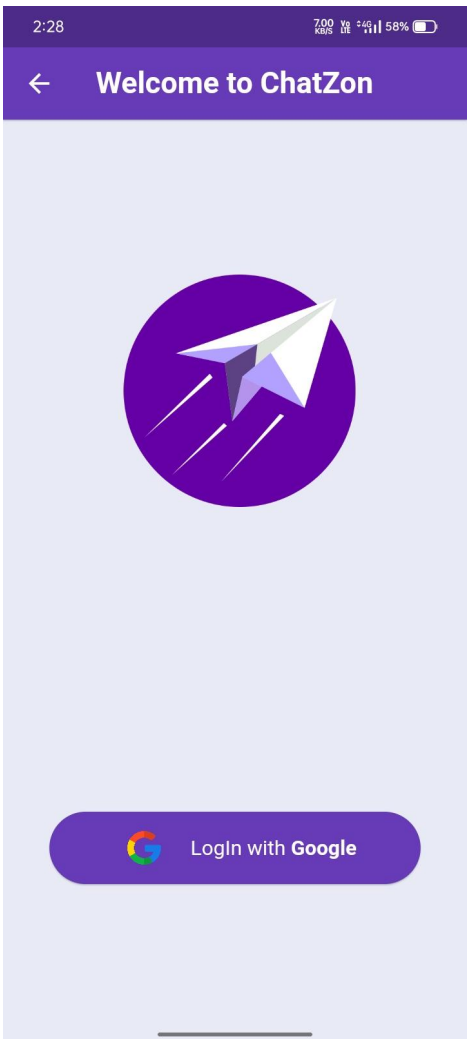
Send/Receive Message

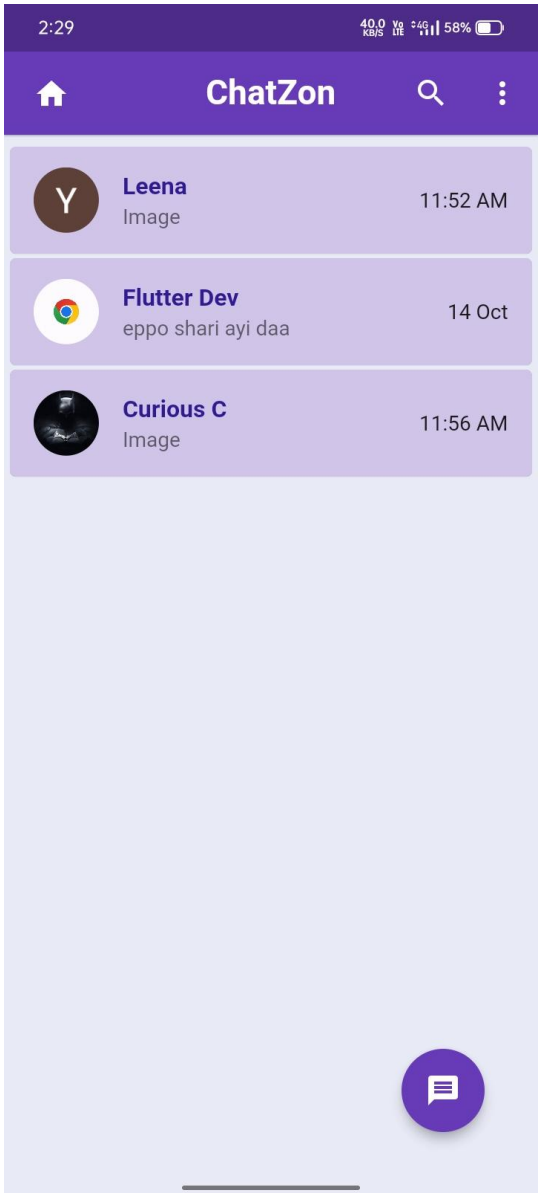
```
class Message {
  Message({
    required this.toId,
    required this.msg,
    required this.read,
    required this.type,
    required this.fromId,
    required this.sent,
  });
  late final String toId;
  late final String msg;
  late final String read;
  late final String fromId;
  late final String sent;
  late final Type type;

  Message.fromJson(Map<String, dynamic> json){
    toId = json['toId'].toString();
    msg = json['msg'].toString();
    read = json['read'].toString();
    type = json['type'].toString()  == Type.image.name ? Type.image :
Type.text;
    fromId = json['fromId'].toString();
    sent = json['sent'].toString();
  }
  Map<String, dynamic> toJson() {
    final data = <String, dynamic>{};
    data['toId'] = toId;
    data['msg'] = msg;
    data['read'] = read;
    data['type'] = type.name;
    data['fromId'] = fromId;
    data['sent'] = sent;
    return data;
  }
}
  enum Type{ text,image }
```

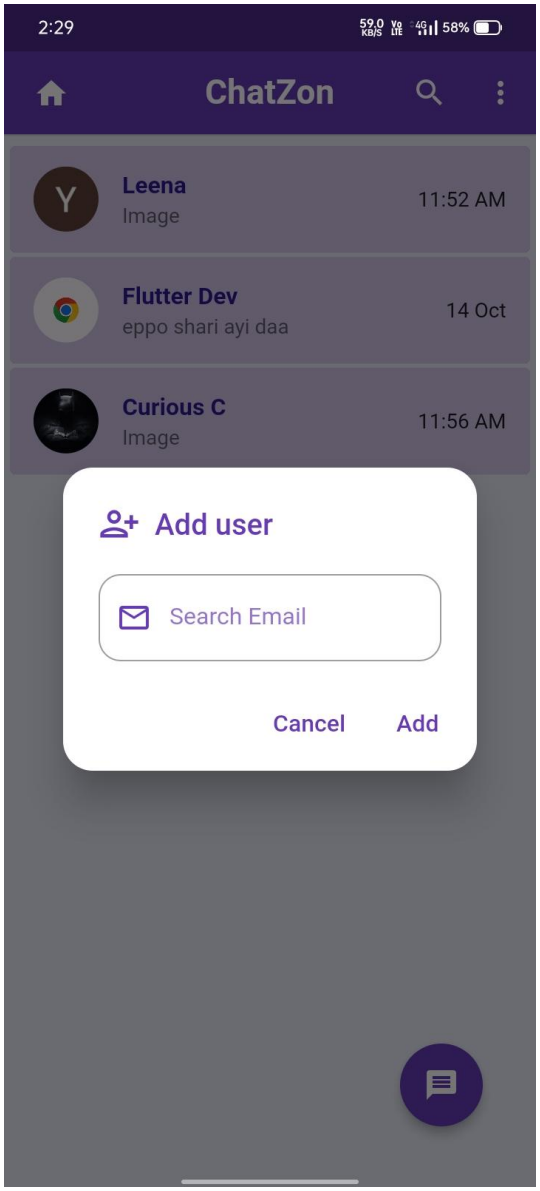
Chapter 8

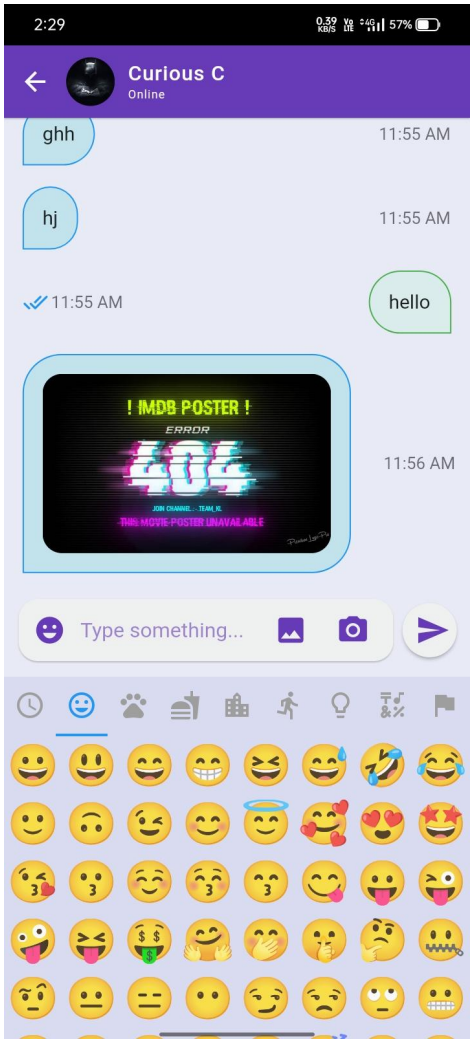
APPENDIX

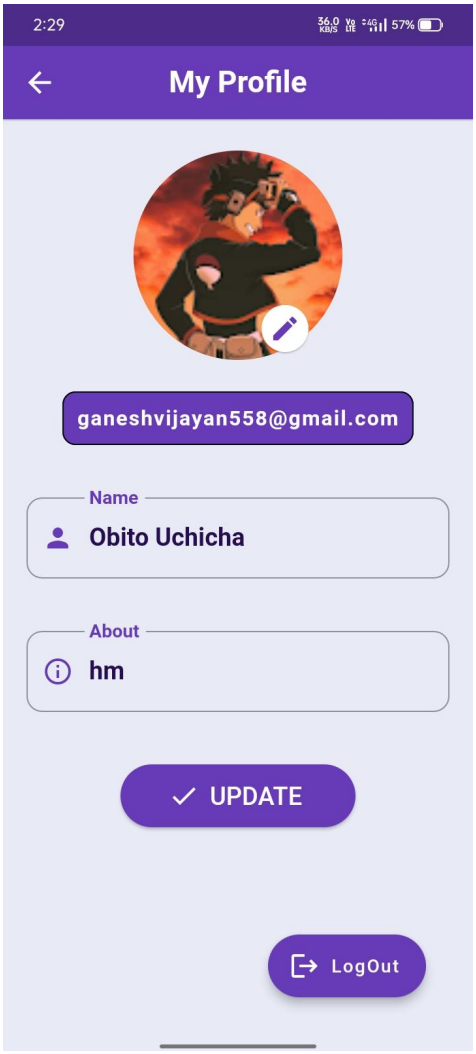





2:29
ChatZon
Leena
Image
11:52 AM
Flutter Dev
eppo shari ayi daa
14 Oct
Curious C
Image
11:56 AM


2:29
ChatZon
Leena
Image
11:52 AM
Flutter Dev
eppo shari ayi daa
14 Oct
Curious C
Image
11:56 AM
Add user
Search Email
Cancel
Add


2:29
Curious C
Online
ghh
11:55 AM
hj
11:55 AM
11:55 AM
hello
! IMDB POSTER !
ERROR
404
11:56 AM
Type something...


2:29
My Profile
ganeshvijayan558@gmail.com
Name
Obito Uchicha
About
hm
UPDATE
LogOut

Chapter 9

CONCLUSION

The new system has over come most of the limitations of the existing system and work according to the design specification given. The developed systems dispense the problem and meet the needs of by providing reliable and comprehensive information. All the requirements projected by the user have been meet by the system .

The newly developed system consumes less processing time and all the details are update and processed immediately. Since the screen provides online help messages and is very user-friendly , any user will get familiarized with its usage . Modules are designed to be highly flexible so that any failure requirements can be easily added to the modules without facing man problems.

Chapter 10

REFERENCES

YouTube Channels

Bototype Malayalam :
https://youtube.com/playlist?list=PLY-ecO2csVHcUlBVvIMAa3dbja12TFJiN&si=54ir9E9n8S-0ouiL

Flutter Teacher : https://youtube.com/@FlutterTeacher?si=_JIElvgmMRxroMQp

The Flutter Way : https://youtube.com/@TheFlutterWay?si=wPF8rzPqAfd5imMW