VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

An Internship Project Report
on

Socialize App

Submitted in partial fulfillment of the requirements for the VIII Semester of degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi

by

Vidyashree E
1RN18IS123

Under the Guidance of

Mrs. Kusuma R
Assistant Professor
Department of ISE

ESTD:2001
An Institute with a Difference

Department of Information Science and Engineering

RNS Institute of Technology

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post,
Channasandra, Bengaluru-560098

2021-2022

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post,
Channasandra, Bengaluru - 560098

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Internship work entitled ***Socialize App*** has been successfully completed by **Vidyashree E (1RN18IS123)** a bonafide student of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements of 8th semester for the award of degree in **Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi** during academic year **2021-2022.** The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

Mrs. Kusuma R	**Dr. Suresh L**	**Dr. M K Venkatesha**
Internship Guide	Professor and HoD	Principal
Assistant Professor	Department of ISE	RNSIT
Department of ISE	RNSIT	

External Viva

Name of the Examiners	**Signature with Date**
1. _____	**1.** _____
2. _____	**2.** _____

DECLARATION

I, **VIDYASHREE E [USN: 1RN18IS123]** student of VIII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Internship work entitled ***Socialize App*** has been carried out by us and submitted in partial fulfillment of the requirements for the *VIII Semester degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi* during academic year 2021-2022.

Place : Bengaluru

Date :

VIDYASHREE E

(1RN18IS123)

ABSTRACT

The advent of social media has been instrumental in providing consumers with quick, relevant and convenient information for social causes. Social media is vast which connects an individual to the entire world. Over the last few years, online communication has moved toward user-driven technologies, such as online social networks (OSNs), blogs, online virtual communities, and online sharing platforms. These social technologies have ushered in a revolution in user-generated data, online global communities, and rich human behavior-related content. Human-generated data and human mobility patterns have become important steps toward developing smart applications in many areas.

The power of these digital networks can be used by individuals and groups for good causes, to have a positive impact on the society at large. Social media platforms are starting to be used by citizens for promoting social causes, creating community engagement to answer societal needs. Yet, precisely because social media platforms have a viral effect, they pose completely new challenges: (1) emerging from a crowded environment, (2) monitoring/managing the truthfulness of information and (3) taking into account cultural differences and preferences.

Socialize app is a social media app where the user can create his or her own account and creates a platform for engagement for people with common interests to associate together and build relationships. It is a cross platform application which is developed using Flutter . This application focuses only on the basic front-end of the application and the flutter application is developed on Visual Studio by using Dart as the programming language .

Socialize app has a total of three screens on focus: the loading page, login and the home screen which displays the user profile. The entire application is set to be made out of widgets , which describes the view of the application in a given state. Depending on the interaction with the user the screens are designed to be either stateless widget or stateful widget . Navigation to the following screen is performed on Button Click using the built-in method.

ACKNOWLEDGMENT

At the very onset I would like to place my gratefulness to all those people who helped me in making the Internship a successful one.

Coming up, this internship to be a success was not easy. Apart from the sheer effort, the enlightenment of the very experienced teachers also plays a paramount role because it is they who guided me in the right direction.

First of all, I would like to thank the **Management of RNS Institute of Technology** for providing such a healthy environment for the successful completion of internship work.

In this regard, I express sincere gratitude to our beloved Principal **Dr. M K Venkatesha,** for providing me all the facilities.

I am extremely grateful to our own and beloved Professor and Head of Department of Information science and Engineering, **Dr. Suresh L**, for having accepted to patronize me in the right direction with all his wisdom.

I place my heartfelt thanks to **Mrs. Kusuma R,** Assistant Professor, Department of Information Science and Engineering for having guided internship and all the staff members of the department of Information Science and Engineering for helping at all times.

I thank **Mr. Akshay D R**, **Co-Founder and CEO**, **ENMAZ**, for providing the opportunity to be a part of the Internship program and having guided me to complete the same successfully.

I also thank my internship coordinator **Dr. R Rajkumar,** Associate Professor, Department of Information Science and Engineering. I would thank my friends for having supported me with all their strength and might. Last but not the least, I thank my parents for supporting and encouraging me throughout. I have made an honest effort in this assignment.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

UI - User Interface

HTML - Hyper Text MarkUp Language

SDK - Software Development Kit

OS - Operating system

XML - Extensible MarkUp Language

DB - Database

IDE - Integrated Development Environment

SRS - Software Requirement Specification

API - Application Programming Interface

MVC - Model View Controller

Chapter 1

Introduction

In the digital world, social media emerge as a powerful weapon to discuss any public issue and sometimes people are interested to involve and give their opinion. Social media play a big role to develop an online community because with the social platform you can discuss all these issues on one platform. On community forum, you can add thousands of peoples in one group because it was difficult to stand half a dozen people on one place to get their opinion, with the help of social media, you can know the vision of many people's on one place.

Social media and community are two different things. Social media help to get connected with each other. We all have groups of friends, relatives, co-workers and many other people and we talk with them on a daily basis and social networking sites help to connect them. With the help of the social networking platform, you are connected with some peoples who are already known but the social community gives a facility to touch with different peoples at one place. The biggest difference between social media and communities is that the communities are the group of different people with different background and history on those people who probably have never met yet, they are on the same page because of the same interest.

Social Community development is a process where different types of community member come to discuss and find a solution for a common problem. A community range can be a small group of audience with a little issue or large initiatives that involve the large group.

Social networking sites provide an environment where pre-established relationships are maintained. We added those people in a group which are already known by us or our friend-circle. We communicate with them, share our thoughts, photos and congratulate them on every special occasion. Social networking sites involve people from all over the world and provide a platform to share their personal thought. This project is a social media app where the user can create his or her own account and creates a platform for engagement for people with common interests to associate together and build relationships.

The project is implemented using flutter which is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia, Web platform, and the web from a single codebase. This app allows users to post their photos of their interests and others users can follow them if they incline towards the same interest. For example, a user who has interest in traveling to places, can post photos of the places they traveled and allows other users to follow him or her and get updates of about the places the user visited over time or recently so they get information about the places prior the visit. Thus, help users connect with each other on a particular interest. The app is handy as it can be used as an app as well as a web application.

Chapter 2

Literature Review

2.1 Android Studio

Android Studio is an integrated development environment (IDE) for Google Android Operating System. It is built based on JetBrains' IntelliJ IDEA Community Edition, and it is specifically designed for creating applications on Android devices. Some of the key features of Android Studio are as follows:

1. Instant Run – a feature that pushes code and resource changes to the running app. It allows changes to be made to the app without the need to restart the app, or rebuilding the APK, so that the effects can be seen instantly.
2. An Emulator – a virtual android device that can simulate a variety of hardware features such as GPS location, network latency, motion sensors, and multi-touch input that can be used to run and install the app. It can then be used for testing purposes.
3. Testing Tools and Frameworks – extensive testing tools such as, JUnit 4 and functional UI test frameworks are included with Android Studio. Espresso Test Recorder can generate UI test code by recording the developer's interactions with the app on a device or emulator. The tests can be run on a device, an emulator, in Firebase Test Lab, or on a continuous integration environment.

2.2 Java Programming Language

Java is an object-oriented programming language created by James Gosling, Mike Sheridan, and Patrick Naughton in 1991. In the paper The Java Language Specification Java SE 8 Edition James Gosling states, "Java programming language is a general-purpose, concurrent, class based, object-oriented language. It is designed to be simple enough that many programmers can achieve fluency in the language. The Java programming language is related to C and C++ but is organized rather differently, with a number of aspects of C and C++ omitted and a few ideas from other languages included. It is intended to be a production language, not a research

language." Java is a very flexible programming language which is used to create many different types of applications for many different operating systems. This is possible because Java can be run on any operating system, as long as the Java Runtime Environment is available. The applications created for Android devices must be coded using Java programming language. This allows these apps to work on variety of different devices, no matter the company that has manufactured the device.

2.3 XML

XML or Extensible Markup Language is a text language that can be used to describe the behavior of programming languages that process them. XML was developed XML working group in 1996. According to World Wide Web Consortium there are ten design goals for XML. These design goals are:

1. XML shall be straightforwardly usable over the Internet.
2. XML shall support a wide variety of applications.
3. XML shall be compatible with SGML.
4. It shall be easy to write programs which process XML documents.
5. The number of optional features in XML is to be kept to the absolute minimum, ideally zero.
6. XML documents should be human-legible and reasonably clear.
7. The XML design should be prepared quickly.
8. The design of XML shall be formal and concise.
9. XML documents shall be easy to create.
10. Terseness in XML markup is of minimal importance.

XML is used when transferring data from the database to the client, and in designing the visual aspect of Android applications. When data is sent from the database, it is sent using XML. This allows the data to be processed by any programming language the same way, since the data

is always sent using XML. As mentioned, XML is also used to design the user interface of Android applications. This means that all the visual aspects such as, the layout of the page, the position of all button and text fields, as well as the color of anything on the page is specified using XML. Since XML is human-legible, it makes the process of designing a page in the app relatively easy and intuitive.

Chapter 3

Analysis

3.1 Introduction

The project deals with the basic User Interface for a social media app using a mobile application.The user must login in order to proceed further with the application. The splash screen is followed by the login screen where the user can login using the existing account details like username,email ID and password or if they are a new user,one can sign up into the application by clicking on sign up and creating a new account. On login, they are directed to the home screen that consists of a user profile.

3.2 Software requirement specification

3.2.1 Introduction

A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform.

3.2.1.1 Flutter

Flutter is an open source framework to create high quality, high performance mobile applications across mobile operating systems - Android and iOS. It provides a simple, powerful, efficient and easy to understand SDK to write mobile applications in Google's own language, *Dart*. Flutter also offers many ready to use widgets (UI) to create a modern application. These widgets are optimized for mobile environments and designing the application using widgets is as simple as designing HTML.Flutter widgets also supports animations and gestures.

3.2.1.2 Visual Studio

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++.It is based on the Electron framework which is used to develop Node.js Web applications that run on the Blink layout engine. It allows users to open one or more directories, which can then be saved in

workspaces for future reuse.Visual Studio Code can be extended via extensions, available through a central repository. A notable feature is the ability to create extensions that add support for new languages, themes, and debuggers, perform static code analysis, and add code linters using the Language Server Protocol. Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software.

3.2.1.3 SQFlite

SQFlite is a plugin for flutter. It allows us store, retrieve and manipulate our SQLite databases via flutter code. SQFlite supports both Android and iOS platforms.

Features of SQFlite:

- SQFlite provides for both database transactions as well as batches.
- SQlite has inbuilt automatic version management.
- SQFlite provides easy to use methods for inserting, querying, updating as well as deleting data from database.
- CRUD operations are performed in the background thread on both iOS and Android. This frees the UI to remain responsive.

3.2.1.4 Dart

Dart is a client-optimized language for fast applications on any platform.Dart is a programming language designed for client development, such as for the web and mobile apps. It is developed by Google and can also be used to build server and desktop applications. Dart is an object-oriented, class-based, garbage-collected language with C -style syntax. Dart can compile to either native code or JavaScript.

3.2.2 Requirements

3.2.2.1 Functional Requirements

1. Splash Screen : The loading screen needs to be navigated to the login screen with a time gap of few seconds
2. Sign Up : Account Creation if the user is new to the application

3. Login : If the user is already having the account
4. Email id and password validations upon insertion
5. Display of the images and user profile in the home screen

3.2.2.2 Non-Functional Requirements

1. The concept of splash screen or Launch Screen is implemented using the built in method Delayed().
2. Model is designed to manage multiple users and their respective data.
3. Navigation has to be implemented to traverse across the screens . The class Navigator has a lot of methods where each perform different actions . The route widget has all the pages or screens in the Flutter application

3.2.2.3 Software and Hardware Requirements

Operating System : Windows10

Programming Language : Dart

Processor : Intel Core i5 7th Gen

Speed : 2.33 Ghz

RAM : 8GB

Software Development Kit : Flutter

IDE : Visual Studio or Android Studio(Official)

Tools : Windows Powershell , Git

3.2.3 General Description

3.2.3.1 Assumptions and Dependencies

Flutter supports using shared packages contributed by other developers to the Flutter and Dart ecosystems. This allows quickly building an app without having to develop everything from scratch. A Flutter app can depend on a plugin via a file system path: dependency. The path can be either relative or absolute. Relative paths are evaluated relative to the directory containing pubspec.yaml.

Scoped Model:

A set of utilities that allow you to easily pass a data Model from a parent Widget down to its descendants. In addition, it also rebuilds all of the children that use the model when the model is updated. This library was originally extracted from the Fuchsia codebase.

SQflite:

SQflite is the SQLite plugin for Flutter. Supports iOS, Android and MacOS. Support transactions and batches. Automatic version management during open. It helps in insert/query/update/delete queries. DB operation executed in a background thread on iOS and Android

Path_provider:

A Flutter plugin for finding commonly used locations on the filesystem. Supports Android, iOS, Linux, macOS and Windows. Not all methods are supported on all platforms.

Chapter 4

System Design

4.1 Introduction

The major aim of the application is to implement a social media app. The focus of the application is on the UI design rather than on the Working of the application as an social media app. It has a login screen which enables you to go to the next screen. The sign in screen allows you to create or register with the application. The home screen consists of user profile details.

The system design can be represented as a widget tree. Flutter widgets are built using a modern framework that takes inspiration from React. The central idea is that you can build one UI out of widgets. Widgets describe what their view should look like given their current configuration and state. When a widget's state changes, the widget rebuilds its description, which the framework differs against the previous description in order to determine the minimal changes needed in the underlying render tree to transition from one state to the next.

There are different kinds of widget as follows,

1.**Container Widget:** Container Widget in Flutter is a parent widget that can contain multiple child widgets and manage them efficiently through various attributes.It is a widget that combines common painting, positioning, and sizing of the child widgets. It is also a class to store one or more widgets and position them on the screen according to our needs.

2.**Column Widget :** Column Widget in Flutter does not scroll . This widgets places the children widget one below the other .

3.**Row Widget :** This widget arranges its children in a horizontal direction on the screen.

4.**Text Widget :** Text Widget allows us to display a string of text with a single line in our application.

5.**Icon widget** : This is used to create icons in our applications. We can create icons in Flutter, either using inbuilt icons or with the custom icons. Flutter provides the list of all icons in the Icons class.

6. **Scafflold Widget :**The Scaffold is a widget in Flutter used to implement the basic material design visual layout structure. This widget is able to occupy the whole device screen. It provides many widgets or APIs for showing Drawer, Snack Bar, Bottom Navigation Bar, App Bar, Floating Action Button.

7.**Box Decoration** : This widget in flutter is an in-built API. It describes how a box should be painted on the screen and its shape. It comes with a ton of properties to further enhance it.

8.**Align Widget:** Align Widget is the widget that is used to align its child within itself and optionally sizes itself based on the child's size. Align Widget is quite flexible and can change its size according to the size of its child.

9. **Stateless widget :** Stateless widget is a widget that describes part of the user interface by building a constellation of other widgets that describe the user interface more concretely. The widgets whose state can not be altered once they are built are called stateless widgets. These widgets are immutable once they are built . It is used when the UI depends on the information within the object itself.

10.**Stateful widget:** Stateful Widgets are the ones that change its properties during run-time. They are dynamic i.e., they are mutable and can be drawn multiple times within its lifetime. It can change its appearance in response to events triggered by user interactions or when it receives data.

4.2 System Architecture

4.2.1 Overview of Application

Fig. 4.2 explains about the overview of the application.The application starts with the splash screen. It contains the company logo which is displayed for about 2 seconds.

This is followed by the login screen. It contains the basics details for login like username, email ID and password. If a user does not have an account can create an account by going to the sign up page from the login page. The user will have to enter the details in order to create an account and these details will be uploaded or saved in the database.

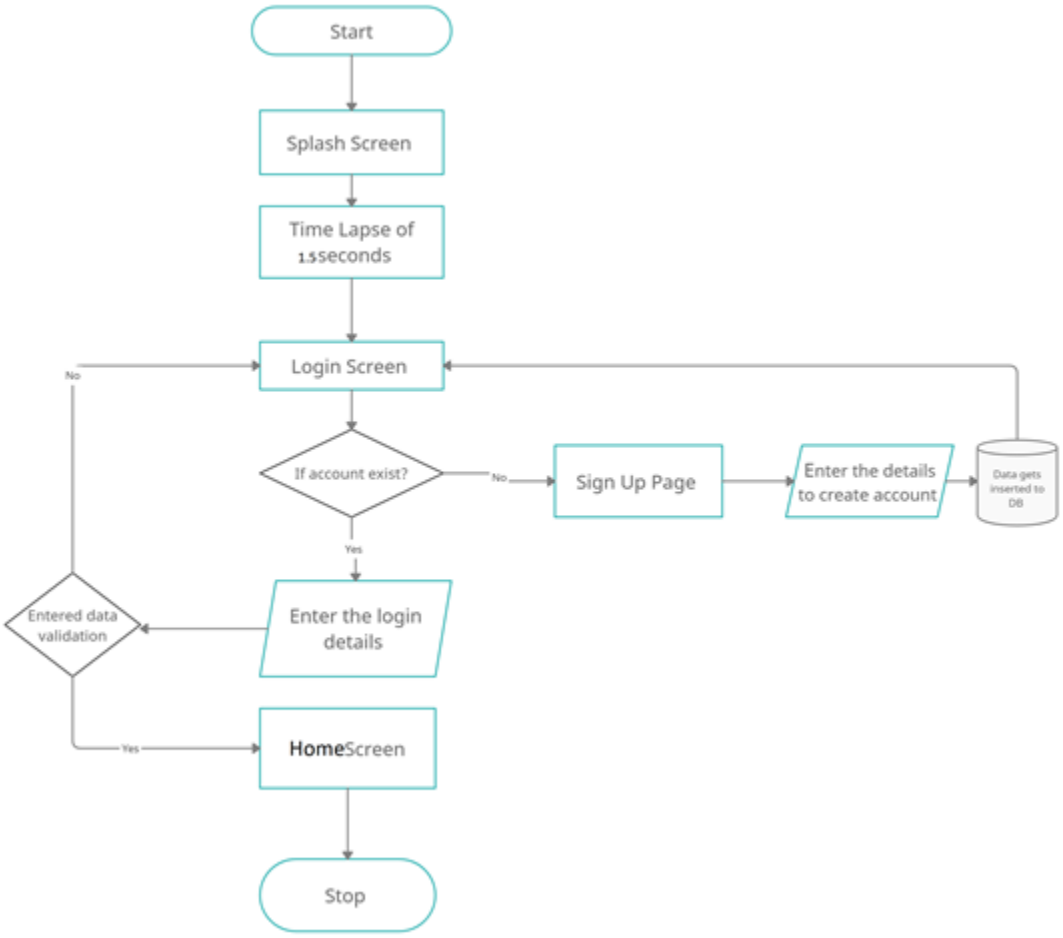

Fig. 4.2 : General Overview of the Application

Once a person registers or creates an account, it directs back to the login page. In the login page the user enters the details like the registered email id and password. It will check if the entered emailId is registered in our database and if it is then checks if the password entered is correct. If the entered password matches, then goes the next page that is the menu page. Else it will remain in the login page itself.

Chapter 5

Detailed Design

5.1 High Level Design

5.1.1 Widget Tree for Splash Screen

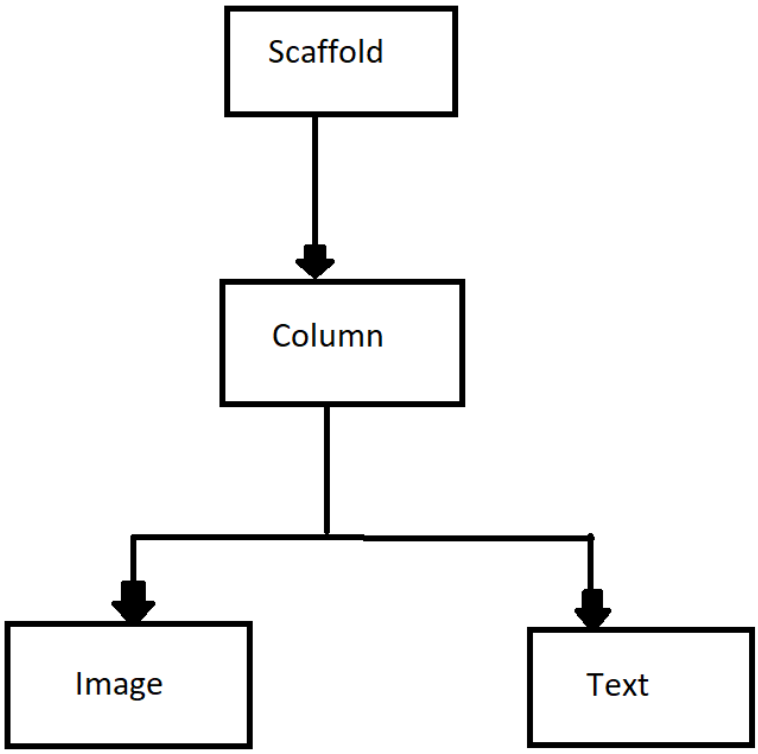

Fig. 5.1 : Widget tree of splash screen

Fig. 5.1 represents Splash screen or Launch screen of the app consists of scaffold widget which consist of column widget.The column widget helps to arrange the image and text widgets one below the other.

5.1.2 Widget Tree for Login Screen

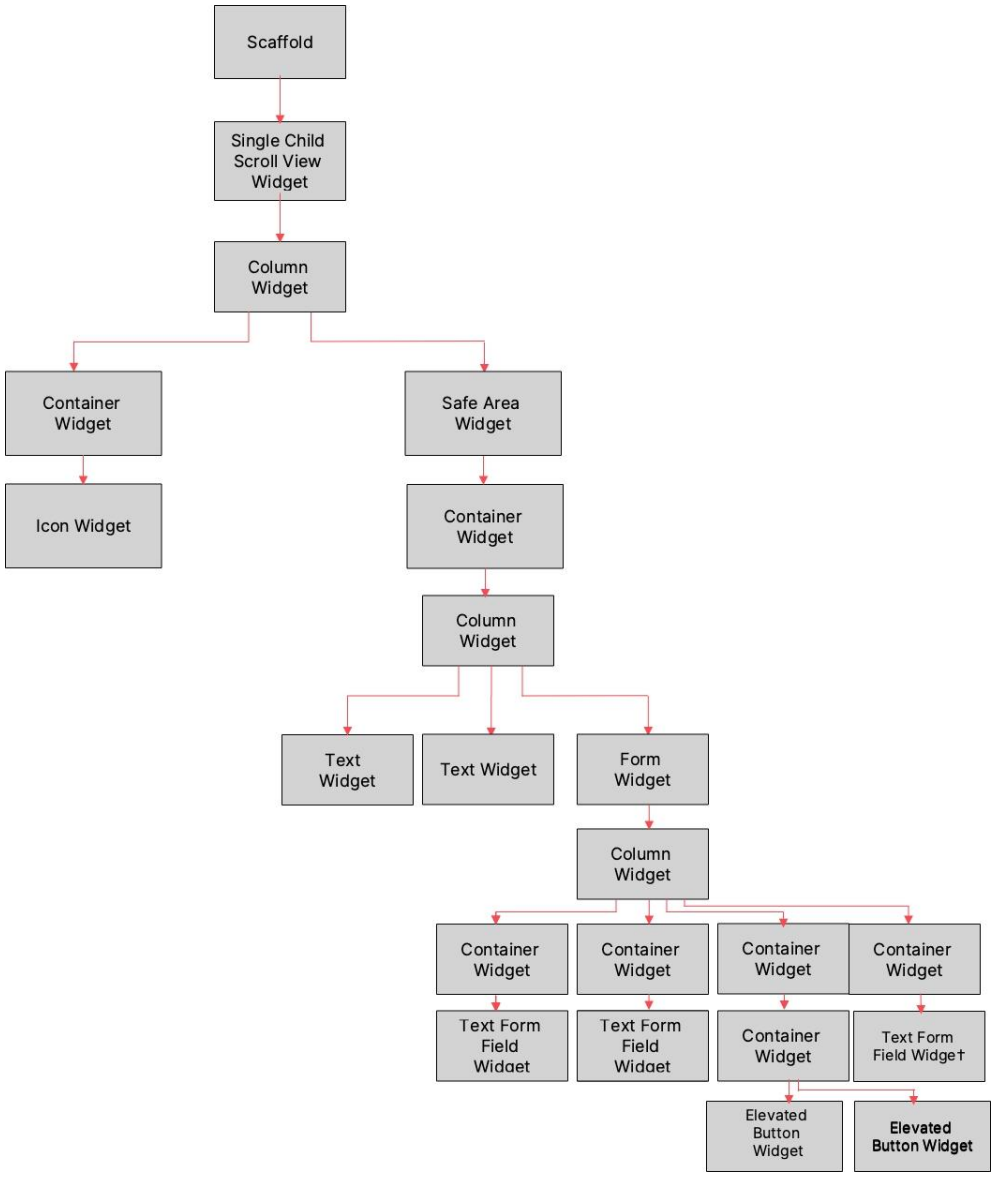

Fig. 5.2 : Widget tree of Login screen

The Fig. 5.2 represents the widget tree of the login screen .It consists of widgets such as form widgets in order to handle user data and many other widgets such as text,icon,elevated buttons and container widgets.

5.1.3 Widget Tree for Sign Up Screen

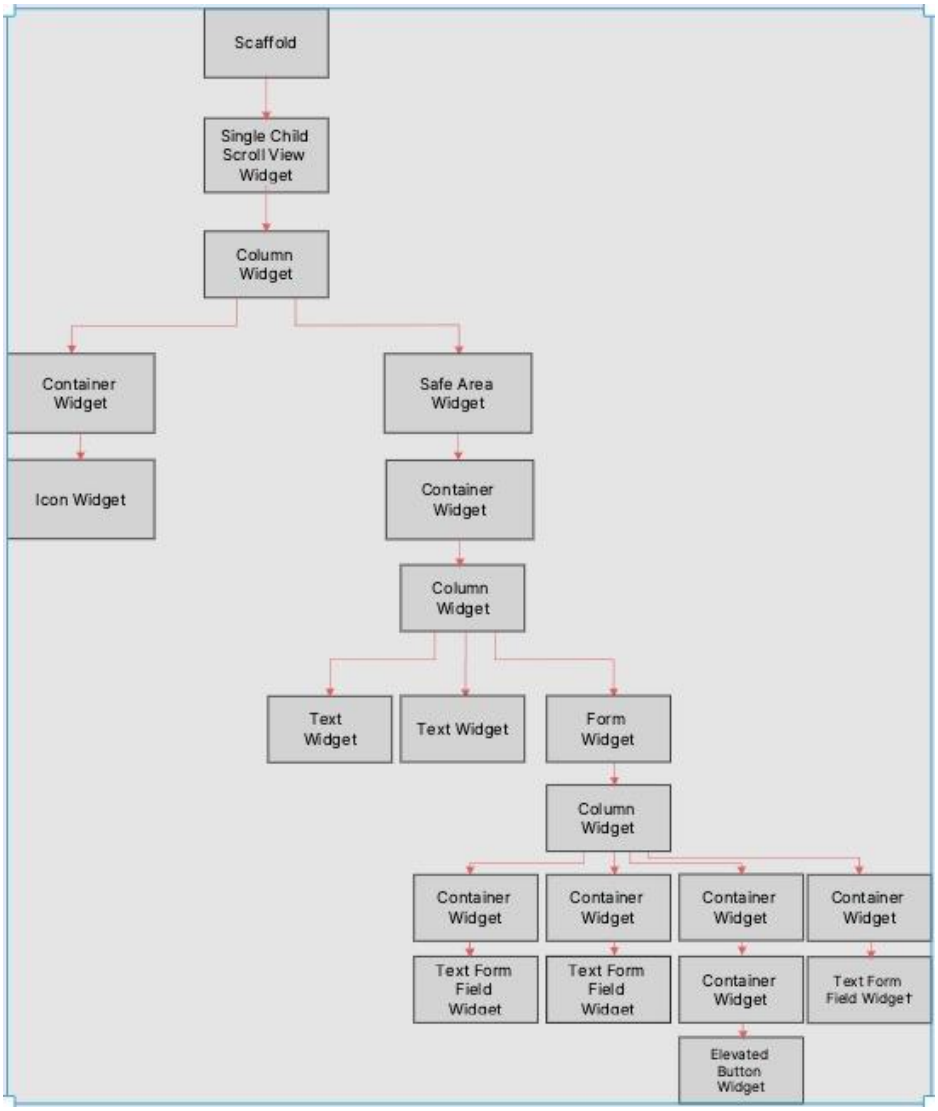

Fig. 5.3 : Widget tree of Sign Up screen

The Fig. 5.3 represents the widget tree of the sign up screen.It consists of widgets such as form widgets in order to handle user data and many other widgets such as text,icon,elevated buttons and container widgets.

5.1.4 Widget Tree for Home screen

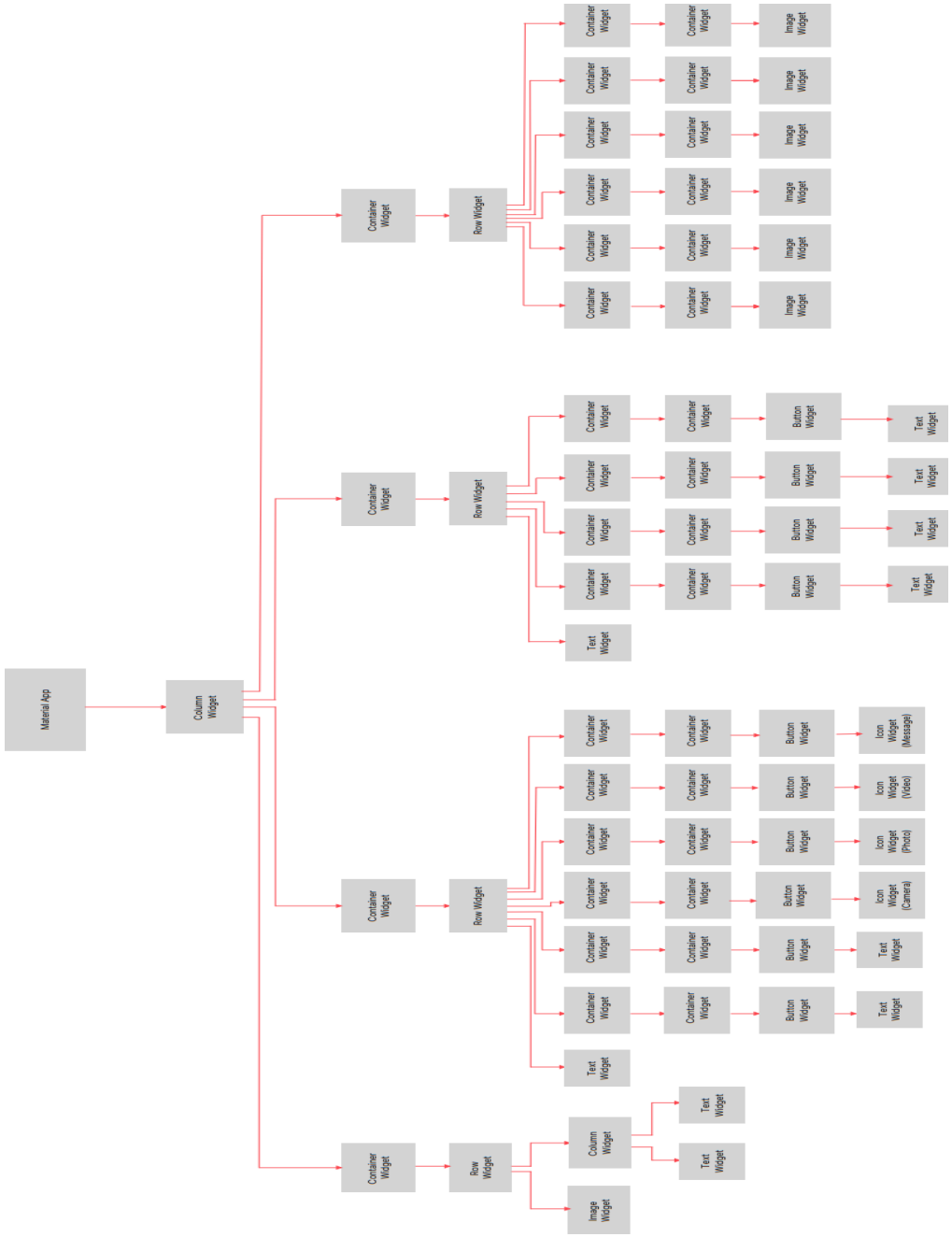

Fig. 5.4 : Widget tree of Home Screen

Fig. 5.4 represents the home screen. It consists of column widget followed by other containers along image widget,icon widgets,elevated button widgets,row and column widgets to represent the UI of the screen.

5.2 Low Level Design

5.2.1 Use Case Diagram for Login

The table 5.1 shows the use case diagram for Login.

Table 5.1 Use Case Diagram for Login

User	**System**
1. The user has to choose to sign up if they don't have an account. If they have an account they can enter the username, email Id and password.	
	2. Once the user types the details it appears on the input box.
3. The user have to click the login button to proceed to the next page.	
	4. The system is going to check if the entered email Id exists in the database and if it exists it checks if the password entered matches with password entered during signup. If it matches it is going to display the next page ie., the home page.

5.2.2 Use Case Diagram for Sign Up

The table 5.2 shows the use case diagram for Signup.

Table 5.2 Use Case Table for Sign Up

	User	**System**
	1. The user has to enter all the necessary details like username , emailId, password ,confirm password.	
		2. The system enters all the details into the Database and navigates to the login screen.

5.2.3 Use Case Diagram for Home Screen

The table 5.3 shows the use case diagram for Home Screen.

Table 5.3 Use Case Table for Home Screen

User	System
1. The user needs to select on the log out button if he wishes to log out of the application	
	2. On button press,the application exits and shows the login screen

Chapter 6

Implementation

6.1 Implementation Support

6.1.1 Installation of Visual Studio Code

- Download VS code from **https://code.visualstudio.com/download.**
- Download the Visual Studio Code installer for suitable OS. Once it is downloaded, run the installer (VSCodeUserSetup-{version}.exe). Then, run the file – it will only take a minute.
- Accept the agreement and click "next."
- After accepting all the requests press finish button. By default, VS Code installs under: the desires path the user wishes to download

6.1.2 Installation of Flutter

- Download the following installation bundle to get the latest stable release of the Flutter SDK **https://docs.flutter.dev/get-started/install/windows**
- Extract the zip file and place the contained flutter in the desired installation location for the Flutter SDK
- Update the Path
    - From the Start search bar, enter 'env' and select Edit environment variables for your account.
    - Under User variables check if there is an entry called Path:
        - If the entry exists, append the full path to flutter\bin using ; as a separator from existing values.
        - If the entry doesn't exist, create a new user variable named Path with the full path to flutter\bin as its value.

6.2 PseudoCodes

6.2.1 PseudoCode for Database Connectivity

```
Future<Database?> get database async{
            if(_database!=null)
                    return _database;
          _database=await _initiateDatabase();
                    return _database;
 }

 _initiateDatabase () async{
   Directory directory=await getApplicationDocumentsDirectory();
   String path=join(directory.path,_dbName);
   return await openDatabase(
           path,
          version:_dbVersion,
           onCreate:_onCreate
   );
 }
```

6.2.2 Creating Table in the Database

```
Future? _onCreate(Database db, int version){
   db.execute(
     '''
     CREATE TABLE $_tableName(
       $_columnUserName text not null,
       $_columnEmail text not null,
       $_columnPAssword text not null
```

```
    )
    '''
  );
 }
```

6.2.3 Inserting Data into Table

```
Future insert(Map<String,dynamic> row) async{
   Database? db=await instance.database;
   await db?.insert(_tableName,row);
  }
```

6.2.4 Route implementation

```
Route <dynamic> generateRoute(RouteSettings settings)
{
 User userL=new User(email: '', userName: '', password: '');
 switch(settings.name)
 {
  case HomeViewRoute:return MaterialPageRoute(builder: (context)=>Splash());
  case LoginViewRoute:
   return MaterialPageRoute(builder: (context)=>LoginPage());
  case SecondViewRoute:
  return MaterialPageRoute(builder:    (context)=>SecondScreen(user:userL, userName:
'',));
 case SignupRoute:return MaterialPageRoute(builder:(context)=>SignUpPage());
    default:
 return MaterialPageRoute(builder: (context)=>SecondScreen(user:userL, userName:'',));
 } }
```

6.2.5 Validation for login

```
if (_formKey.currentState!.validate())
 {
    _formKey.currentState!.save();
          bool response=await auth.authUser(userL);
          if (response == true) {
        //Navigator.pushNamed(context,SecondViewRoute);
          Navigator.push(
          context,
          MaterialPageRoute(
          builder: (context) =>SecondScreen(
          userName:userL.userName, user:userL,
     ),
 }
else
          ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text('Login Unsuccessful')));
```

6.2.6 Splash Screen implementation

```
 class _SplashState extends State<Splash> {
 @override
void initState(){
 super.initState();
 _navigatetohome();
```

```
}
_navigatetohome() async
{
  await Future.delayed(Duration(milliseconds:1500),);
    Navigator.pushNamed(context,LoginViewRoute);
}
 @override
 Widget build(BuildContext context){
   return Scaffold(
  body:Center(

            child:Column(
            mainAxisAlignment : MainAxisAlignment.center,
            children: [

        Image.asset('assets/images/icon.png',fit: BoxFit.cover,),
            Text("Ilcome to Socialize",style: TextStyle(
            fontSize: 24,
  fontIight: FontIight.bold),),
            ],)  )
  );
```

Chapter 7

Testing

7.1 Introduction

Testing is a process of executing a program with the interest of finding an error. A good test is one that has a high probability of finding the yet undiscovered error. Testing should systematically uncover different classes of errors in a minimum amount of time with a minimum number of efforts. Two classes of inputs are provided to test the process.

1. A software configuration that includes a software requirement specification, a design specification and source code.
2. A software configuration that includes a test plan and procedure, any testing tool and test cases and their expected results.

7.2 Levels Of Testing

7.2.1 Unit Testing

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

Table 7.2 Unit Testing for the Application

Case_Id	Description	Input data	Expected Output	Actual Output	Output Status
1.	Validating username for login. User name of any combination and any length is alloId	internship	Accept the username, password and Email-id, display Home page	Home Page	Pass
2	Validating password of length 6 for login	internship	Accept the username, email-id password and display Home page	Home Page	Pass
3	Validating email for login	internship@gmail.com	Accept the username, email-id password and display Home page	Home Page	Pass

4	Validating username for signup. User name of any combination and any length is alloId	socialize	Accept the username, email-id password and display Login Page	Login Page	Pass
5.	Validating password for signup. Length of password should be 6	socialize	Accept the username, email-id password and display Login Page	Login Page	Pass
6	Validating confirm password for signup.	socialize	Accept the username, email-id password and display Login Page	Login Page	Pass
6.	Validating username for sign up and login		Display Please Enter Username	Please Enter Username	Pass
7.	Validating password for sign up and login		Display Please Enter Password	Please Enter Password	Pass

8.	Validating email-id for login and signup		Display Please Enter Email-id	Please Enter Email-id	Pass
9.	Validation password of length 3 in signup	123	Display Please Enter Password of Length 6	Please Enter Password of Length 6	Pass
10.	Validation of confirm password in signup Input Password: Confirm Password:	 Socialize 12345	Display Password Doesn't match	Password Doesn't match	Pass
11	Validating Password. When password not authenticated	12345	Display Login Unsuccessful	Display Login Unsuccessful	Pass

Chapter 8

Results

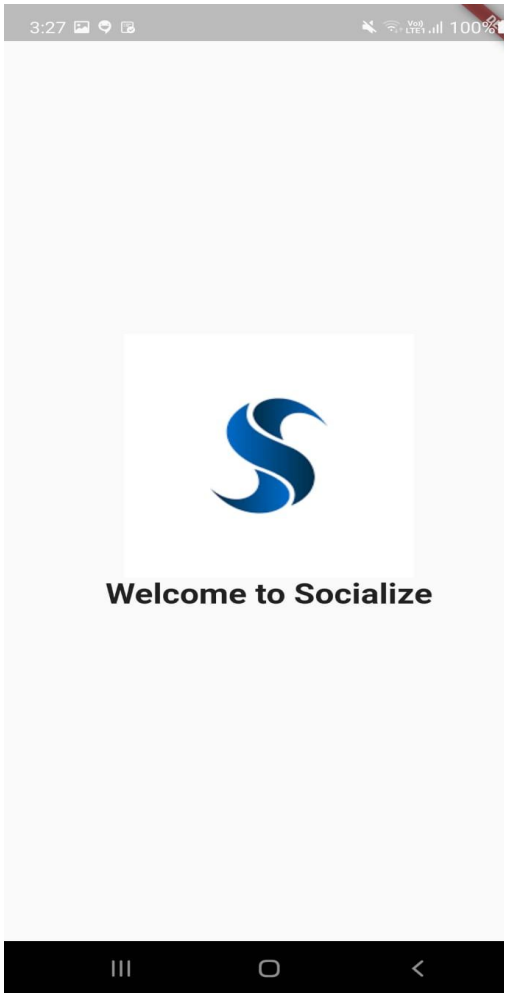

Fig. 8.1 : Splash/Launch Screen

The above Fig. 8.1 gets displayed when the app is opened and the screen is called the Launch or the Splash screen which automatically navigates to the login screen after a time lapse of 1.5 seconds.

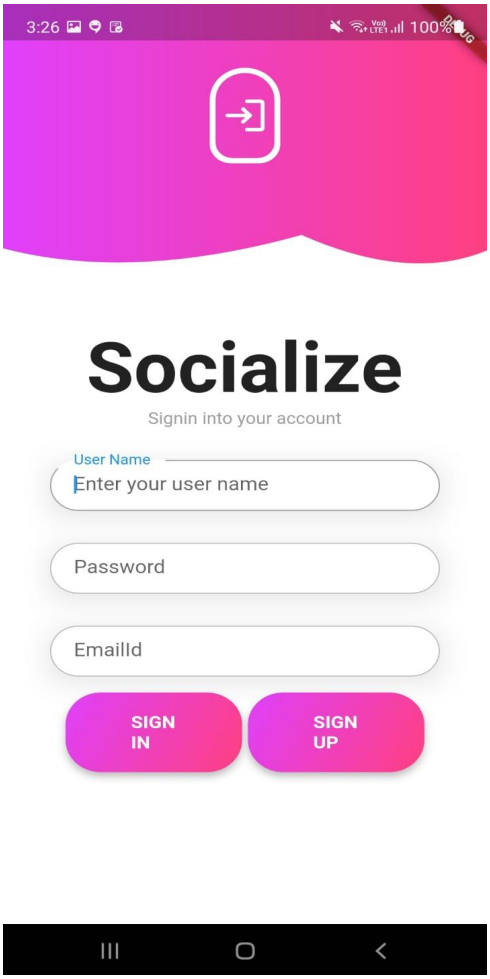

Fig. 8.2: Login Screen

The Fig. 8.2 which is the login Screen accepts input from the user and validates the entered data with the data stored in the database.If the inserted data is correct than navigation to home screen takes place. If there is a new user who doesn't have an account this page gives the option of sign up which navigates to another page .

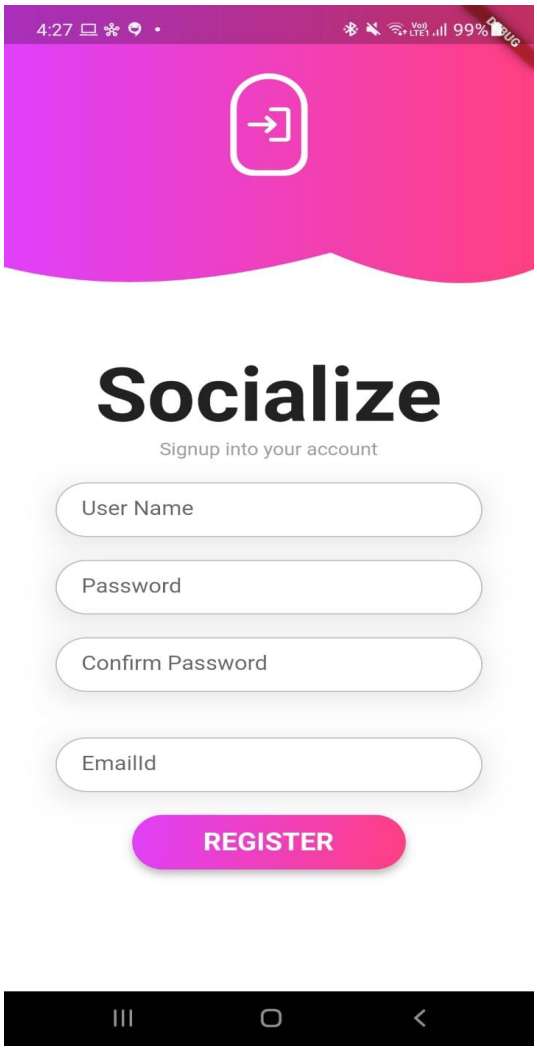

Fig. 8.3 : SignUp Page

The Fig. 8.3 shows our sign up page where the new user will have to enter details like username , email address password and confirm password . The same gets stored into the database.Once the data gets stored in the database the user is navigated back to the login page.

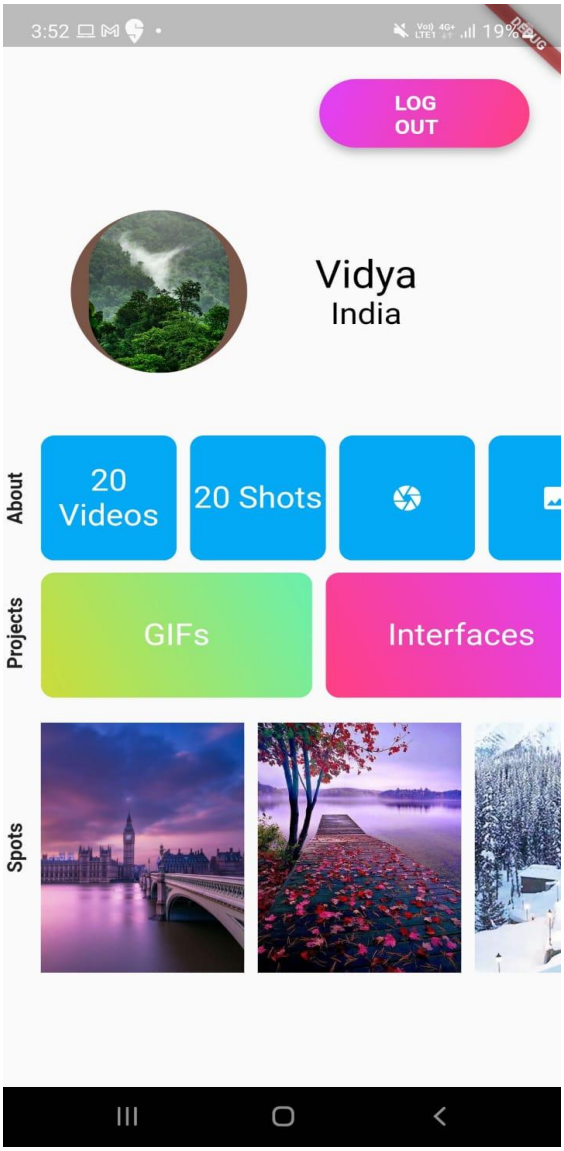

Fig. 8.4 : Home Screen

The Fig. 8.4 displays the home page where the user profile is displayed.

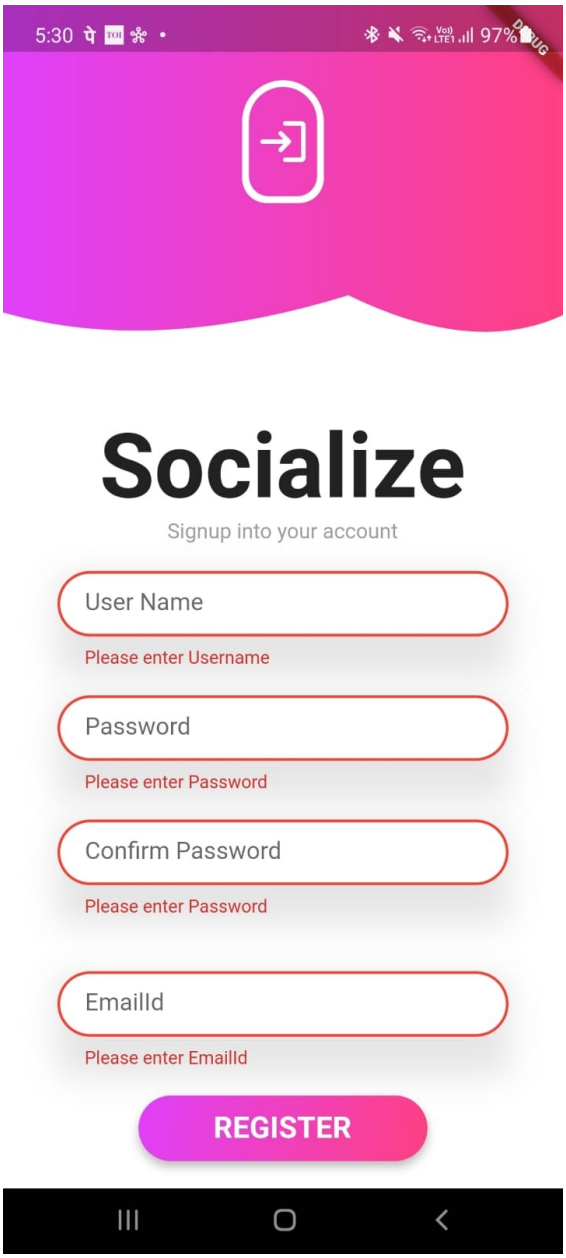

Fig. 8.5 : Validation of empty input

Fig. 8.5 displays the case of validation where the user has not entered any details. So error message pops up asking the user to enter the same .

Chapter 9

Conclusion and Future Work

Conclusion

Socialize app, the online social media application is a 3 screen application . The developed application focuses purely on the UI. The application performs basic validation of the user using the database. It displays the user profile which consists of the user details such as images,number of shots and videos.

Future Work

1. Forgot Password Screen can be implemented.
2. Home screen can be designed to have better interaction.
3. User profile consisting of ones details can be displayed.
4. User can add any number of photos.
5. Backend for this application can be developed.

References

[1] https://flutter.dev/

[2] https://www.javatpoint.com/flutter

[3] https://www.geeksforgeeks.org/flutter-tutorial/

[4] https://medium.com/flutter

[5] https://stackoverflow.com/questions/tagged/flutter