

MINI PROJECT

(2020-2021)

Secure E-Learning Chat Application

MID-TERM REPORT



Institute Of Engineering and Technology

Team Members

Shalini Kumari

(181500649)

Ritik Srivastava

(181500578)

Adarsh Kumar

(181500037)

Shashank Agrawal

(181500652)

Vanshika Agrawal

(181500780)

-Supervised By

Name- Mr. Amit Kumar Chaudhary

Training and Placement Dept.

Department of Computer Engineering & Applications

CONTENT

Abstract

1. Introduction

- 1.1 General Introduction to the topic
- 1.2 How does online chat works?
- 1.3 Hardware and Software Requirements

2. Problem definition

3. Objectives

4. Implementation Details

5. Progress till Date & The Remaining work

6. Some Screenshots

7. ER Diagram

8. Data Flow Diagram

9. References

Abstract of the project e-chat Application:

The purpose of e-chat Application is to automate the existing manual system by the help of computerized equipment's and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The software and hardware are easily available and easy to work with.

E-chat Application, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the recording keeping. Thus it will help organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

The aim is to automate its existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirement, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically the project describe how to manage for good performance and better services for clients.

Introduction to e-chat Application:



Online e-chat application is a form of communication that utilizes computer programs that allow for two-way conversations between users in real time (events that occur in cyberspace at the same speed that they would occur in real life).

Typically the users will connect to a chat server using a chat client and meet in a chat room. Once the users are in the same chat room, they can converse with one another by typing messages into a window where all of the other users in the chat room can see the message. The user can also see all of the messages entered by the other users. Conversations are then carried on by reading the messages entered by the other users in the chat room and responding to them.

How does online chat work?

The technologies that allows users to converse in real time via typed messages are the chat servers and the chat clients. These special servers allow the messages to be quickly received and posted so that the chat clients can receive the messages and display them in the window on the users' computers. Fortunately, most chat clients are relatively inexpensive shareware programs, and there are a number of chat servers that let the user connect to them for free. Online chat could be an alternative to meeting in a traditional classroom. While all of the participants would still be able to freely converse with one another, they would

carrying on the conversation from the comfort of their own home, thus avoiding the expense and inconvenience of traveling to another place.

Problem Statement:

As we all know about the current pandemic situation due to COVID-19, because of which we all are locked at our homes for so many months and don't know what future situation will be. Especially for students, this situation seems very difficult to handle.

So many students are slowly developing big confusions and distractions in their brains. But many are doing e-learning courses and online classes to develop their skills, but then also there are many questions that are always inside them and they don't have anyone to guide them. So, to clear all these questions and put their cars back on track our app can really help. This app will connect students directly with the teachers of their fields and will help them to solve their queries.

Reason:

In the past few years, chat rooms have become a popular way to support a forum for n-way conversation or discussion among a set of people with interest in a common topic. Chat applications range from simple, text-based ones to entire virtual worlds with exotic graphics. But also, many of us still have many questions in our brains.

The reasons behind this maybe are:

1. There is no authority to check the mentors' guidance.
2. They have to rely only on their faculty and their guidance.
3. Delay in the response due to a bunch of emails to be filtered by them to give any reply to anyone.
4. The questions can be missed out by the faculty due to a lot of messages.
5. Fail to recall the consultation after a respective time.

So by taking all these points under consideration we proposed to build a project which can solve these issues, like:

1. If a reply is not given within an allotted time the ranking of the teacher will decrease.
2. Huge and the wide network is created by this app having all university teachers on a single platform.

Objective:

The main objective of the project is to make an E-learning chat mobile application at the university level where queries of students could be solved by teachers which can be achieved by the following:

- Login by University E-mail Id.
- Verifying teachers.
- Creating a chat-box.

Future Scope:

The application can be extended to cover a wide range of intra-organizational as well as extra-organizational issues. Apart from doubt sessions related to studies, other areas like management and mental wellness could also be touched upon.

It could evolve as a single platform for knowledge sharing within an organization. Content moderation can be done using the sentimental analysis to ensure the code of conduct. Extending it to Web Support.

Requirements:

1. Hardware:
Ram required- 8GB or above
Processor- core i5 or above
2. Software:
Linux or Windows Operating System
Android Studio or Vs-code
Android or IOS Emulator or Physical Smartphone.
3. Technology Used:
Flutter 1.20
Dart Programming Language
Firebase

Methodology:

The idea of this methodology is to develop an e-Learning system that favours the competitive position of the organization that uses it and is adaptable to suit the student's needs. It refers to the mode and manner in which they produce the content that will be used in training.

In this application students and the faculty can sign up using their university email id and help each other to solve the queries they have. Students can broadcast or can directly message to a particular faculty for their queries. Students can also tag their messages to a particular faculty or experts in that field. To differentiate between students and faculty we will provide verified tags to the faculty using this application and for that, the faculty just have to make a request on the verification request tab.

Modules:

Description:

Login/Sign Up Screen	Users can sign up with their university email id and specify their role as student or faculty.
Contact Screen	Users can see the list of contacts they messaged previously or got a message from.
Chat Screen	Users can see the chats/messages they send or receive.
User Profile Screen	Users can see their information here.
Verification Screen	Faculty can send a verification request from this page.
Feedback Screen	The students can give their feedback or any suggestion regarding any topic.

Testing:

The Flutter framework provides comprehensive support for Flutter automated testing of mobile apps. Manually testing mobile apps can be hard, especially if your app has many features. Automated tests help to ensure that your app performs correctly before you publish it while maintaining your feature and bug-fix velocity.

Automated testing falls into a few categories:

- A unit testing tests a single function, method, or class.
- A widget test (in other UI frameworks referred to as component test) tests a single widget.
- An Integration Test tests a complete app or a large part of an app.

Contributions:

Today people know how to use smart devices and these devices are a huge part of their life. Hence, no time is better than now to introduce this technology (educational mobile app) via their mobile devices. These mobile learning platforms are very appealing and will have the total attention of the learners.

E-learning is a new education concept by using internet technology, it delivers the digital content, provides a learner-orient environment for the teachers and students. E-learning promotes the construction of life-long learning opinions and learning society

E-Learning has completely transformed the way in which learning is imparted to students. Unlike traditional chalk and board methods of teaching, e-learning makes learning simpler, easier, and more effective.

E-Learning mobile app allows users to learn by using podcasts, videos and other mechanisms on their smartphone. Hence, they make the process of learning quite flexible for learners.

E-Learning mobile app allows the student to directly be in touch with the subject professors. Hence they can clear their doubts quickly. Often students suffer from lack of collaboration, however, these mobile learning apps offer them collaborative learning via online learning community app.

E-learning app ensure higher engagement, as learning is easily accessible by the students at any time and any place. One of the best benefits of e-learning is multi-device support.

Progress till date and some screenshot:

Welcome Screen

```
1 import 'package:comperio/constants.dart';
2 import 'package:comperio/screen/login_screen.dart';
3 import 'package:comperio/screen/registration_screen.dart';
4 import 'package:comperio/welcome_screen_button.dart';
5 import 'package:flutter/material.dart';
6
7 class WelcomeScreen extends StatefulWidget {
8   final String id = 'WelcomeScreen';
9
10  @override
11  _WelcomeScreenState createState() => _WelcomeScreenState();
12 }
```

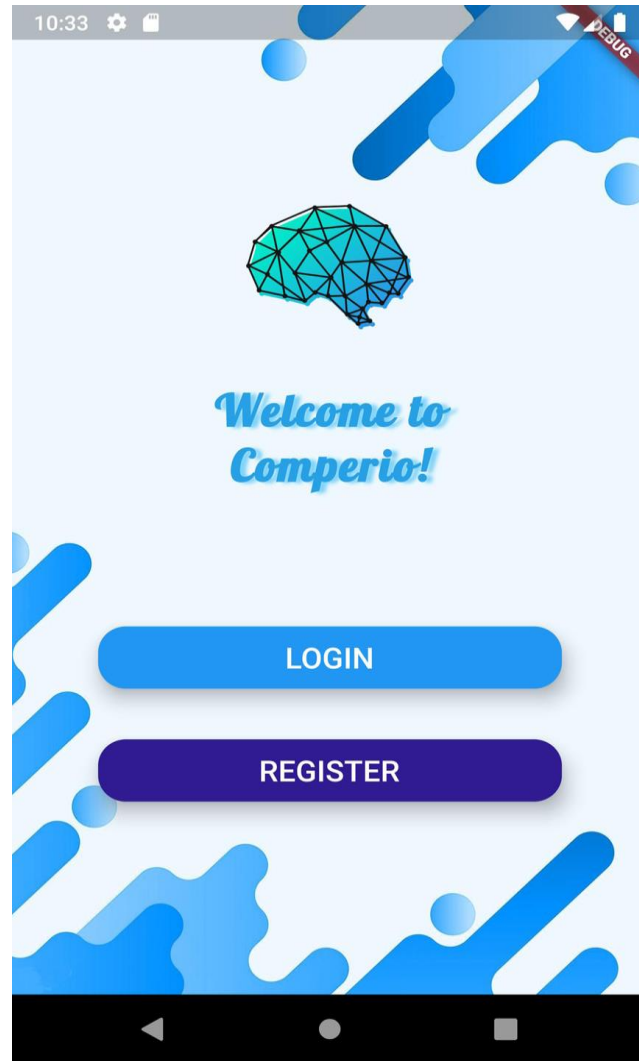


```
13
14 class _WelcomeScreenState extends State<WelcomeScreen>
15   with SingleTickerProviderStateMixin {
16   AnimationController controller;
17   @override
18   void initState() {
19     super.initState();
20     controller = AnimationController(
21       duration: Duration(seconds: 3),
22       vsync: this.
23     );
24     controller.forward();
25     controller.addListener() {
26       setState() {});
27     });
28   }
29
30   @override
31   void dispose() {
32     controller.dispose();
33     super.dispose();
34   }
35
36   @override
37   Widget build(BuildContext context) {
38     return Container(
39       decoration: BoxDecoration(
40         color: Colors.white,
41         image: DecorationImage(
42           image: AssetImage('images/WelcomeScreen3.jpg'),
43           fit: BoxFit.cover,
44         ),
45       ),
46       constraints: BoxConstraints.expand(),
47       child: Scaffold(
48         backgroundColor: Colors.transparent,
49         body: Container(
50           child: Column(
51             mainAxisAlignment: MainAxisAlignment.spaceEvenly,
52             children: <Widget>[
53               Container(
54                 padding: EdgeInsets.only(left: 30.0, right: 30.0),
55                 child: Column(
```

```
56      children: <Widget>[
57        Hero(
58          tag: 'logo',
59          child: Container(
60            height: controller.value * 100,
61            child: Image(
62              image: AssetImage(
63                'images/comperio-logo.png',
64              ),
65            height: 110.0,
66            width: 110.0,
67          ),
68        ),
69      ),
70      SizedBox(
71        height: 30.0,
72      ),
73      Center(
74        child: Text(
75          'Welcome to',
76          style: KWelcomeTextStyle,
77        ),
78      ),
79      Center(
80        child: Text(
81          'Comperio!',
82          style: KWelcomeTextStyle,
83        ),
84      ),
85      SizedBox(
86        height: 90.0,
87      ),
88      WelcomeScreenButtons(
89        buttonText: 'Login',
90        redirectionScreen: LoginScreen().id,
91        colour: Colors.blue,
92      ),
93      SizedBox(
94        height: 30.0,
95      ),
96      WelcomeScreenButtons(
97        buttonText: 'Register',
```

```
98         redirectionScreen: RegistrationScreen().id,  
99         colour: Color(0xff311b92)  
100     ),  
101     ],  
102     ),  
103     ),  
104     ],
```

```
105     ),  
106     ),  
107     ),  
108     );  
109 }  
110 }
```



Registration Screen

```
1 import 'dart:io';
2 import 'package:cloud_firestore/cloud_firestore.dart' show FirebaseFirestore;
3 import 'package:comperio/constants.dart';
4 import 'package:comperio/screen/contacted_person_screen.dart';
5 import 'package:comperio/screen_app_logo.dart';
6 import 'package:firebase_auth/firebase_auth.dart';
7 import 'package:firebase_storage/firebase_storage.dart';
8 import 'package:flutter/cupertino.dart';
9 import 'package:flutter/material.dart';
```

```

10 import 'package:flutter/widgets.dart';
11 import 'package:image_picker/image_picker.dart';
12 import 'package:modal_progress_hud/modal_progress_hud.dart';
13 import 'package:path/path.dart';
14 import '../helper_functions.dart';
15
16 class RegistrationScreen extends StatefulWidget {
17   final String id = 'RegistrationScreen';
18
19   @override
20   _RegistrationScreenState createState() => _RegistrationScreenState();
21 }
22
23 class _RegistrationScreenState extends State<RegistrationScreen> {
24   final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
25   bool _autoValidate = false;
26   bool showSpinner = false;
27   final _auth = FirebaseAuth.instance;
28   String email;
29   String username;
30   String password;
31   File _image;
32   String url = 'https://firebasestorage.googleapis.com/v0/b/comperio-1071d.appspot.com/o/default-
   profile.webp?alt=media&token=52b10457-a10a-417e-b5af-3d84e5833fae';
33   Future getImages() async {
34     PickedFile pickedFile = await ImagePicker().getImage(source: ImageSource.gallery);
35     var image = File(pickedFile.path);
36     setState(() {
37       _image = image;
38     });
39   }
40   void _addToDatabase(String userName, String dpUrl) {
41     print('dpurl is: $dpUrl');
42     List<String> splitList = username.split(" ");
43     List<String> indexList = [];
44     for (int i = 0; i < splitList.length; i++) {
45       for (int y = 1; y < splitList[i].length + 1; y++) {
46         indexList.add(splitList[i].substring(0, y).toLowerCase());
47       }
48     }
49     FirebaseFirestore.instance.collection('users').doc().set({
50       'username': userName,
51       'searchKeywords': indexList,

```

```

52  'profileURL': dpUrl,
53  'email': email,
54  });
55  }
56  Future uploadPic(BuildContext context) async {
57    String fileName = basename(_image.path);
58    StorageReference firebaseStorageRef =
59      FirebaseStorage.instance.ref().child(fileName);
60    StorageUploadTask uploadTask = firebaseStorageRef.putFile(_image);
61    var dowUrl = await (await uploadTask.onComplete).ref.getDownloadURL();
62    url = dowUrl.toString();
63    print(url);
64    StorageTaskSnapshot taskSnapshot = await uploadTask.onComplete;
65    setState(() {
66      print("Profile Picture uploaded")
67    });
68  }
69  String validateEmail(String value) {
70    Pattern pattern = r'^([^\<()[]\|.,;:\s@"']+(\.[^\<()[]\|.,;:\s@""]+)*)(\.".\")@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\])|([a-zA-Z-0-9]+\.)+[a-zA-Z]{2,}))$';
71    RegExp regex = new RegExp(pattern);
72    if (!regex.hasMatch(value))
73      return 'Enter Valid Email';
74    else
75      return null;
76  }
77  String validateUserName(String value) {
78    Pattern pattern = r'^[A-Za-z0-9]+(?:[ _][A-Za-z0-9]+)*$';
79    RegExp regex = new RegExp(pattern);
80    !regex.hasMatch(value) ? return 'Invalid username' : return null;
81  }
82  String validatePassword(String value) {
83    Pattern pattern = r'^(?=.*[0-9]+.)(?=.*[a-zA-Z]+.)(?=.*[0-9a-zA-Z]{6,})$';
84    RegExp regex = new RegExp(pattern);
85    !regex.hasMatch(value) ? return 'Invalid Password, include letters and numbers' : return null;
86  }
87  bool _validateInputs() {
88    if (_formKey.currentState.validate())
89      _formKey.currentState.save();
90    return true;
91  } else {
92    setState(() {
93      _autoValidate = true;

```

```

94     });
95     return false;
96   }
97 }
98
99 @override
100 Widget build(BuildContext context) {
101   return Container(
102     decoration: BoxDecoration(
103       image: DecorationImage(
104         image: AssetImage('images/app-background.jpg'),
105         fit: BoxFit.cover,
106       ),
107     ),
108     constraints: BoxConstraints.expand(),
109     child: Scaffold(
110       resizeToAvoidBottomInset: true,
111       resizeToAvoidBottomPadding: true,
112       backgroundColor: Colors.transparent,
113       body: ModalProgressHUD(
114         inAsyncCall: showSpinner,
115         child: Container(
116           child: Column(
117             crossAxisAlignment: CrossAxisAlignment.start,
118             children: <Widget>[
119               Container(
120                 padding: EdgeInsets.only(
121                   top: 60.0, left: 30.0, right: 30.0, bottom: 30.0),
122                 child: Column(
123                   crossAxisAlignment: CrossAxisAlignment.start,
124                   children: <Widget>[
125                     ScreenAppLogo(),
126                     Text(
127                       'Comperio',
128                       style: KAppNameTextStyle,
129                     ),
130                   ],
131                 ),
132               ),
133             Flexible(
134               fit: FlexFit.tight,
135               child: Container(
136                 width: double.infinity,

```

```

137         padding: EdgeInsets.only(left: 20.0, right: 20.0),
138         decoration: BoxDecoration(
139           color: Colors.white,
140           borderRadius: BorderRadius.only(
141             topLeft: Radius.circular(20.0),
142             topRight: Radius.circular(20.0),
143           ),
144         ),
145         child: SingleChildScrollView(
146           child: Padding(
147             padding: const EdgeInsets.only(
148               left: 16.0, right: 16.0, top: 25, bottom: 16.0),
149             child: Column(
150               mainAxisAlignment: MainAxisAlignment.min,
151               crossAxisAlignment: CrossAxisAlignment.stretch,
152               children: <Widget>[
153                 Row(
154                   mainAxisAlignment: MainAxisAlignment.spaceBetween,
155                   children: [
156                     Column(
157                       crossAxisAlignment: CrossAxisAlignment.start,
158                       children: [
159                         Text(
160                           'New',
161                           style: KCardTextStyle,
162                         ),
163                         SizedBox(
164                           height: 5.0,
165                         ),
166                         Text(
167                           'Account',
168                           style: KCardTextStyle,
169                         ),
170                       ],
171                     ),
172                     Container(
173                       child: Stack(
174                         alignment: Alignment.center,
175                         overflow: Overflow.visible,
176                         children: [
177                           Container(
178                             width: 80.0,
179                             height: 80.0,

```



```

180      child: CircleAvatar(
181        radius: 40.0,
182        backgroundColor: Colors.black12,
183        child: ClipOval(
184          child: SizedBox(
185            width: 80.0,
186            height: 80.0,
187            child: (_image != null)
188              ? Image.file(
189                _image,
190                fit: BoxFit.fill,
191              )
192              : Image.asset(
193                'images/default-profile.jpg',
194                fit: BoxFit.fill,
195              ),
196          ),
197        ),
198      ),
199    ),
200    Positioned(
201      top: -1,
202      right: -30,
203      bottom: -40,
204      left: 40,
205      child: IconButton(
206        icon: Icon(
207          Icons.camera,
208          color: Colors.blueAccent,
209          size: 20,
210        ),
211        onPressed: () {
212          getImages();
213        },
214      ),
215    ),
216  ],
217 ),
218 ),
219 ],
220 ),
221 SizedBox(
222   height: 30.0,

```

```

223         ),
224         Flexible(
225             fit: FlexFit.loose,
226             child: Form(
227                 key: _formKey,
228                 autovalidate: _autoValidate,
229                 child: Column(
230                     mainAxisAlignment:
231                         MainAxisAlignment.spaceEvenly,
232                     children: <Widget>[
233                         TextFormField(
234                             keyboardType: TextInputType.emailAddress,
235                             decoration: const InputDecoration(
236                                 icon: Icon(Icons.email),
237                                 hintText: 'Enter your Email',
238                                 hintStyle: TextStyle(
239                                     color: Colors.grey,
240                                 ),
241                                 labelText: 'Email *',
242                             ),
243                             validator: validateEmail,
244                             onSave: (String value) {
245                                 email = value;
246                             },
247                         ),
248                         SizedBox(
249                             height: 20.0,
250                         ),
251                         TextFormField(
252                             decoration: const InputDecoration(
253                                 icon: Icon(Icons.person),
254                                 hintText: 'Enter Username',
255                                 hintStyle: TextStyle(
256                                     color: Colors.grey,
257                                 ),
258                                 labelText: 'Username *',
259                             ),
260                             validator: validateUserName,
261                             onSave: (String value) {
262                                 username = value;
263                             },
264                         ),
265                         SizedBox(

```

```

266         height: 20.0,
267     ),
268     TextFormField(
269         decoration: const InputDecoration(
270             icon: Icon(Icons.lock),
271             hintText: 'Enter the password',
272             hintStyle: TextStyle(
273                 color: Colors.grey,
274             ),
275             labelText: 'Password *',
276         ),
277         obscureText: true,
278         validator: validatePassword,
279         onSave: (String value) {
280             password = value;
281         },
282     ),
283     SizedBox(
284         height: 50.0,
285     ),
286     Padding(
287         padding: const EdgeInsets.symmetric(
288             horizontal: 30.0,
289         ),
290     child: RaisedButton(
291         elevation: 10.0,
292         shape: StadiumBorder(),
293         color: Colors.lightBlueAccent,
294         onPressed: () async {
295             bool isShowSpinner =
296                 _validateInputs();
297             setState(() {
298                 showSpinner = isShowSpinner;
299             });
300             try {
301                 await _auth
302                     .createUserWithEmailAndPassword(
303                         email: email,
304                         password: password)
305                     .then((newUser) {
306                         if (newUser != null) {
307                             HelperFunctions
308                                 .saveUserLoggedInInSharedPreference(

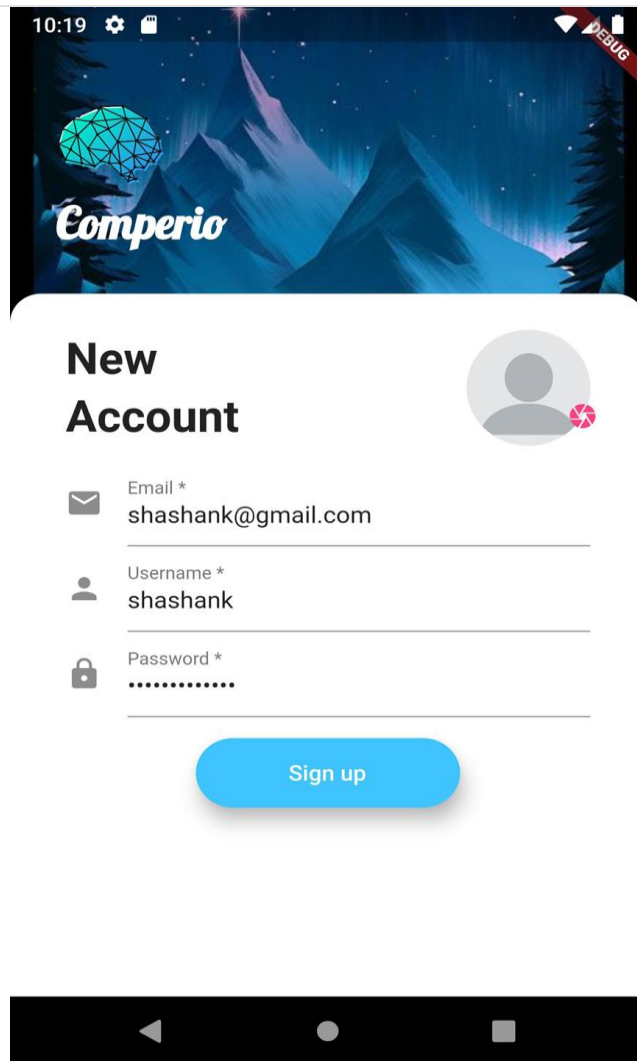
```

```

309             true);
310         HelperFunctions
311             .saveUserNameSharedPreference(
312                 username);
313         HelperFunctions
314             .saveUserEmailSharedPreference(
315                 email);
316         HelperFunctions
317             .saveUserPhotoUrlSharedPreference(url);
318         Navigator.pushNamed(context,
319             ContactedPersonScreen().id);
320     }
321   });
322   if (_image != null) {
323     await uploadPic(context);
324   }
325   _addToDatabase(username, url);
326   setState(() {
327     showSpinner = false;
328   });
329   } catch (e) {
330     print(e);
331   }
332   },
333   child: Text(
334     'Sign up',
335     style: KLoginRegistrationButtonStyle,
336   ),
337   padding: EdgeInsets.symmetric(
338     vertical: 15.0,
339     horizontal: 60.0,
340   ),
341   ),
342   ),
343   ],
344   ),
345   ),
346   ),
347   ],
348   ),
349   ),
350   ),
351   ),

```

```
352     ),  
353     ],  
354     ),  
355     ),  
356     ),  
357     ),  
358     );  
359 }  
360 }
```



Login Screen

```

1 import 'package:cloud_firestore/cloud_firestore.dart' show FirebaseFirestore;
2 import 'package:comperio/constants.dart';
3 import 'package:comperio/screen/contacted_person_screen.dart';
4 import 'package:comperio/screen_app_logo.dart';
5 import 'package:firebase_auth/firebase_auth.dart';
6 import 'package:flutter/cupertino.dart';
7 import 'package:flutter/material.dart';
8 import 'package:modal_progress_hud/modal_progress_hud.dart';
9 import '../helper_functions.dart';
10 class LoginScreen extends StatefulWidget {
11   final String id = 'LoginScreen';
12   @override
13   _LoginScreenState createState() => _LoginScreenState();
14 }
15 class _LoginScreenState extends State<LoginScreen> {
16   final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
17   bool _autoValidate = false;
18   bool showSpinner = false;
19   final _auth = FirebaseAuth.instance;
20   String email;
21   String password;
22   String validateEmail(String value) {
23     Pattern pattern =
24   r'^(([^<>()[\]\\\.,;:~\s@"]+(\.[^<>()[\]\\\.,;:~\s@"]+)*)|(\\".+\"))@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\])|([a-zA-Z\0-9]+\.)+[a-zA-Z]{2,}))$';
25     RegExp regex = new RegExp(pattern);
26     return !regex.hasMatch(value) ? 'Enter Valid Email' : null;
27   }
28   String validatePassword(String value) {
29     Pattern pattern = r'^(?=.*[0-9]+.*)(?=.*[a-zA-Z]+.*)[0-9a-zA-Z]{6,}$';
30     RegExp regex = new RegExp(pattern);
31     if (!regex.hasMatch(value))
32       return 'Invalid password';
33     else
34       return null;
35   }
36   bool _validateInputs() {
37     if (_formKey.currentState.validate()) {
38       _formKey.currentState.save();
39       return true;
40     } else {
41       setState() {
42         _autoValidate = true;

```

```

43   });
44   return false;
45 }
46 }
47 @override
48 Widget build(BuildContext context) {
49   return Container(
50     decoration: BoxDecoration(
51       color: Colors.lightBlueAccent,
52       image: DecorationImage(
53         image: AssetImage('images/app-background-2.jpg'),
54         fit: BoxFit.cover,
55       ),
56     ),
57     constraints: BoxConstraints.expand(),
58     child: Scaffold(
59       resizeToAvoidBottomInset: true,
60       resizeToAvoidBottomPadding: true,
61       backgroundColor: Colors.transparent,
62       body: ModalProgressHUD(
63         inAsyncCall: showSpinner,
64         child: Container(
65           child: Column(
66             crossAxisAlignment: CrossAxisAlignment.start,
67             children: <Widget>[
68               Container(
69                 padding: EdgeInsets.only(
70                   top: 60.0, left: 30.0, right: 30.0, bottom: 30.0),
71                 child: Column(
72                   crossAxisAlignment: CrossAxisAlignment.start,
73                   children: <Widget>[
74                     ScreenAppLogo(),
75                     Text(
76                       'Comperio',
77                       style: KAppNameTextStyle,
78                     ),
79                   ],
80                 ),
81               ),
82               Flexible(
83                 fit: FlexFit.tight,
84                 child: Container(
85                   width: double.infinity,

```

```

86         padding: EdgeInsets.symmetric(horizontal: 20.0),
87         decoration: BoxDecoration(
88           color: Colors.white,
89           borderRadius: BorderRadius.only(
90             topLeft: Radius.circular(20.0),
91             topRight: Radius.circular(20.0),
92           ),
93         ),
94         child: Padding(
95           padding: const EdgeInsets.only(
96             left: 16.0, right: 16.0, top: 35, bottom: 16.0),
97           child: Form(
98             key: _formKey,
99             autovalidate: _autoValidate,
100            child: SingleChildScrollView(
101              child: Column(
102                mainAxisAlignment: MainAxisAlignment.min,
103                crossAxisAlignment: CrossAxisAlignment.stretch,
104                children: <Widget>[
105                  Text(
106                    'Welcome',
107                    style: KCardTextStyle,
108                  ),
109                  SizedBox(
110                    height: 5.0,
111                  ),
112                  Text(
113                    'Back',
114                    style: KCardTextStyle,
115                  ),
116                  SizedBox(
117                    height: 30.0,
118                  ),
119                  Flexible(
120                    fit: FlexFit.loose,
121                    child: Column(
122                      mainAxisAlignment:
123                        MainAxisAlignment.spaceEvenly,
124                      children: <Widget>[
125                        TextFormField(
126                          keyboardType: TextInputType.emailAddress,
127                          decoration: const InputDecoration(
128                            icon: Icon(Icons.person),

```



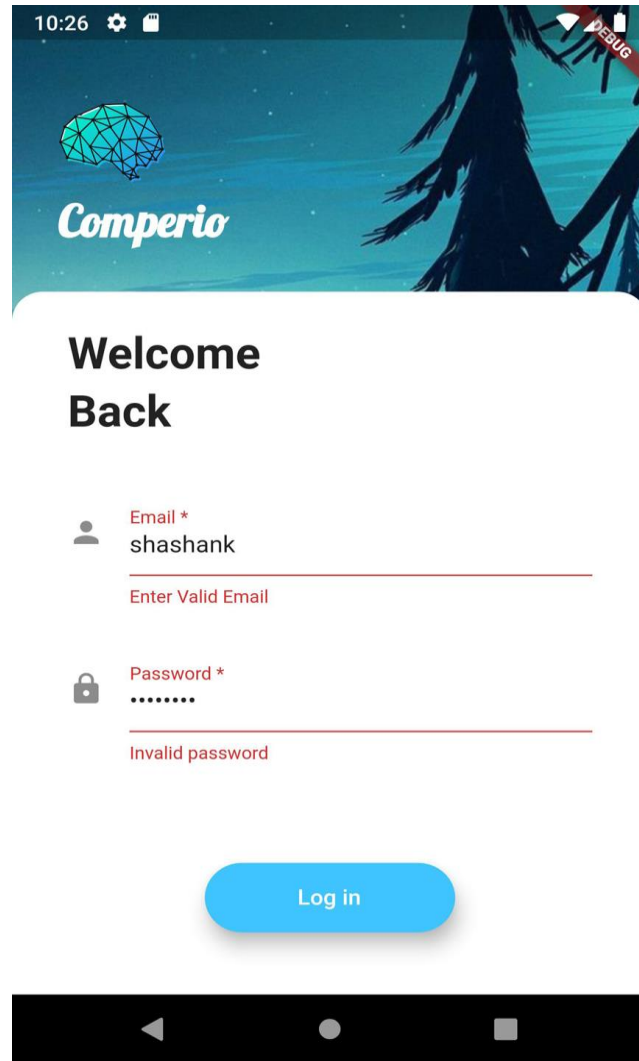
```
129         hintText: 'Enter the Email',
130         hintStyle: TextStyle(
131           color: Colors.grey,
132         ),
133         labelText: 'Email *',
134       ),
135       validator: validateEmail,
136       onSave: (String value) {
137         email = value;
138       },
139     ),
140     SizedBox(
141       height: 30.0,
142     ),
143     TextFormField(
144       decoration: const InputDecoration(
145         icon: Icon(Icons.lock),
146         hintText: 'Enter the password',
147         hintStyle: TextStyle(
148           color: Colors.grey,
149         ),
150         labelText: 'Password *',
151       ),
152       obscureText: true,
153       validator: validatePassword,
154       onSave: (String value) {
155         password = value;
156       },
157     ),
158     SizedBox(
159       height: 50.0,
160     ),
161     Padding(
162       padding: const EdgeInsets.symmetric(
163         horizontal: 30.0,
164       ),
165     ),
166     child: RaisedButton(
167       elevation: 10.0,
168       shape: StadiumBorder(),
169       color: Colors.lightBlueAccent,
170       onPressed: () async {
171         bool isShowSpinner =
172           _validateInputs();
```

```

172         setState() {
173             showSpinner = isShowSpinner;
174         });
175         try {
176             await _auth
177                 .signInWithEmailAndPassword(
178                 email: email,
179                 password: password)
180                 .then((newUser) async {
181                 if (newUser != null) {
182                     var userInfoSnapshot =
183                         await FirebaseFirestore
184                             .instance
185                             .collection("users")
186                             .where('email',
187                                 isEqualTo: email)
188                             .getDocuments()
189                             .catchError((e) {
190                                 print(e.toString());
191                             });
192                     HelperFunctions
193                         .saveUserLoggedInSharedPreference(
194                         true);
195                     HelperFunctions
196                         .saveUserNameSharedPreference(
197                         userInfoSnapshot
198                             .documents[0]
199                             .get('username'));
200                     HelperFunctions
201                         .saveUserEmailSharedPreference(userInfoSnapshot.documents[0].get('email'));
202                     HelperFunctions
203                         .saveUserPhotoUrlSharedPreference(userInfoSnapshot.documents[0].get('profileURL'));
204                     Navigator.pushNamed(context,
205                         ContactedPersonScreen().id);
206                 }
207             });
208             setState() {
209                 showSpinner = false;
210             });
211         } catch (e) {
212             print(e);
213         }
214     },

```

```
215         child: Text(  
216           'Log in',  
217           style: KLoginRegistrationButtonStyle,  
218         ),  
219         padding: EdgeInsets.symmetric(  
220           vertical: 15.0,  
221           horizontal: 60.0,  
222         ),  
223       ),  
224     ),  
225   ],  
226 ),  
227 ),  
228 ],  
229 ),  
230 ),  
231 ),  
232 ),  
233 ),  
234 ),  
235 ],  
236 ),  
237 ),  
238 ),  
239 ),  
240 );  
241 }  
242 }
```



Chat Screen

```
1 import 'package:cloud_firestore/cloud_firestore.dart';
2 import 'package:comperio/app_icons.dart';
3 import 'package:comperio/attach_file_components.dart';
4 import 'package:comperio/constants.dart';
5 import 'package:comperio/screen/contacted_person_screen.dart';
6 import 'package:firebase_auth/firebase_auth.dart';
7 import 'package:flutter/cupertino.dart';
8 import 'package:flutter/material.dart';
9 import 'package:font_awesome_flutter/font_awesome_flutter.dart';
10 import 'package:intl/intl.dart';
```

```
11
12 final _firestore = FirebaseFirestore.instance;
13 User loggedInUser;
14
15 class ChatScreen extends StatefulWidget {
16   final String id = 'ChatScreen';
17
18   @override
19   _ChatScreenState createState() => _ChatScreenState();
20 }
21
22 class _ChatScreenState extends State<ChatScreen> {
23   final messageTextController = TextEditingController();
24   final _auth = FirebaseAuth.instance;
25   String messageText;
26
27   @override
28   void initState() {
29     super.initState();
30     getCurrentUser();
31   }
32
33   void getCurrentUser() async {
34     try {
35       final user = await _auth.currentUser;
36       if (user != null) {
37         loggedInUser = user;
38         print(loggedInUser.email);
39       }
40     } catch (e) {
41       print(e);
42     }
43   }
44
45   void messageStream() async {
46     await for (var snapshot in _firestore.collection('Messages').snapshots()) {
47       for (var message in snapshot.docs) {
48         print(message.data());
49       }
50     }
51   }
52
53   @override
```

```

54 Widget build(BuildContext context) {
55   return Container(
56     decoration: BoxDecoration(
57       color: Color(0xFF1d2d50),
58     ),
59     constraints: BoxConstraints.expand(),
60     child: Scaffold(
61       resizeToAvoidBottomInset: false,
62       resizeToAvoidBottomPadding: false,
63       backgroundColor: Colors.transparent,
64       body: Column(
65         crossAxisAlignment: CrossAxisAlignment.start,
66         children: <Widget>[
67           Container(
68             padding: EdgeInsets.only(
69               top: 40.0, left: 10.0, right: 20.0, bottom: 20.0),
70             child: Row(
71               mainAxisAlignment: MainAxisAlignment.start,
72               children: [
73                 IconButton(
74                   icon: AppIcons(
75                     iconName: Icons.arrow_back,
76                     iconSize: 30.0,
77                     colour: Colors.white,
78                   ),
79                   onPressed: () {
80                     Navigator.pushNamed(context, ContactedPersonScreen().id);
81                   },
82                 ),
83                 CircleAvatar(
84                   radius: 17.0,
85                   backgroundColor: Colors.white,
86                 ),
87                 SizedBox(
88                   width: 10.0,
89                 ),
90                 Text(
91                   'username',
92                   style: KUserTextStyle,
93                 ),
94               ],
95             ),
96           ),

```

```

97     Expanded(
98       child: Container(
99         decoration: BoxDecoration(
100           color: Colors.white,
101           image: DecorationImage(
102             image: AssetImage('images/chat_bg8.jpeg'),
103             fit: BoxFit.cover,
104           ),
105           borderRadius: BorderRadius.only(
106             topLeft: Radius.circular(20.0),
107             topRight: Radius.circular(20.0),
108           ),
109         ),
110         child: Column(
111           mainAxisAlignment: MainAxisAlignment.spaceBetween,
112           crossAxisAlignment: CrossAxisAlignment.stretch,
113           children: <Widget>[
114             MessagesStream(),
115           ],
116         ),
117       ),
118     ),
119     Container(
120       padding: EdgeInsets.all(10.0),
121       color: Colors.white,
122       child: SingleChildScrollView(
123         child: Container(
124           margin: EdgeInsets.only(bottom: 5.0, left: 3.0),
125           padding: EdgeInsets.only(
126             bottom: MediaQuery.of(context).viewInsets.bottom),
127           child: Row(
128             crossAxisAlignment: CrossAxisAlignment.center,
129             children: <Widget>[
130               IconButton(
131                 icon: Icon(FontAwesomeIcons.paperclip),
132                 color: Colors.blueGrey,
133                 onPressed: () {
134                   showModalBottomSheet(
135                     context: context,
136                     builder: (context) {
137                       return AttachFileBottomSheet();
138                     },
139                   );

```

```

140     },
141   ),
142   Expanded(
143     child: TextField(
144       keyboardType: TextInputType.multiline,
145       maxLines: null,
146       controller: messageTextController,
147       onChanged: (value) {
148         //do something when pressed
149         messageText = value;
150       },
151       decoration: InputDecoration(
152         contentPadding: EdgeInsets.symmetric(
153           vertical: 10.0, horizontal: 20.0),
154         hintText: 'Type your message here...',
155         enabledBorder: OutlineInputBorder(
156           borderRadius:
157             BorderRadius.all(Radius.circular(12.0)),
158           borderSide: BorderSide(
159             color: Colors.grey,
160           ),
161         ),
162         focusedBorder: OutlineInputBorder(
163           borderRadius:
164             BorderRadius.all(Radius.circular(10.0)),
165           borderSide:
166             BorderSide(color: Colors.lightBlueAccent),
167         ),
168       ),
169     ),
170   ),
171   RaisedButton(
172     padding: EdgeInsets.all(15.0),
173     color: Color(0xFF1d2d50),
174     onPressed: () {
175       messageTextController.clear();
176       _firestore.collection('Messages').add({
177         'message': messageText,
178         'sender': loggedInUser.email,
179         'Date': DateFormat('dd-MMM-yy hh:mm')
180           .format(DateTime.now())
181           .toString(),
182         'fileUrl': " ",

```



```

183         'orderDateFormat': DateFormat('dd-MMM-yy hh:mm:ss')
184         .format(DateTime.now())
185         .toString(),
186     });
187 },
188 shape: CircleBorder(),
189 child: AppIcons(
190     iconName: Icons.send,
191     iconSize: 20.0,
192     colour: Colors.white,
193 ),
194 ),
195 ],
196 ),
197 ),
198 ),
199 )
200 ],
201 ),
202 ),
203 );
204 }
205 }
206
207 class MessagesStream extends StatelessWidget {
208   @override
209   Widget build(BuildContext context) {
210     return StreamBuilder<QuerySnapshot>(
211       stream: _firestore
212         .collection('Messages')
213         .orderBy('orderDateFormat')
214         .snapshots(),
215       builder: (context, snapshot) {
216         if (!snapshot.hasData) {
217           return Center(
218             child: CircularProgressIndicator(
219               backgroundColor: Colors.lightBlueAccent,
220             ),
221           );
222         }
223         final messages = snapshot.data.docs.reversed;
224         List<MessageBubble> messageBubbles = [];
225         for (var message in messages) {

```

```

226     final messageText = message.data()['message'];
227     final messageSender = message.data()['sender'];
228     final messageDateTime = message.data()['Date'];
229     final messageFile = message.data()['fileUrl'];
230
231     final currentUser = loggedInUser.email;
232
233     final messageBubble = MessageBubble(
234         sender: messageSender,
235         text: messageText,
236         dateTime: messageDateTime,
237         isMe: currentUser == messageSender,
238         messageFileUrl: messageFile,
239     );
240     messageBubbles.add(messageBubble);
241 }
242 return Expanded(
243     child: ListView(
244         reverse: true,
245         padding: EdgeInsets.symmetric(
246             horizontal: 10.0,
247             vertical: 20.0,
248         ),
249         children: messageBubbles,
250     ),
251 );
252 },
253 );
254 }
255 }
256
257 class MessageBubble extends StatelessWidget {
258     final String sender;
259     final String text;
260     final bool isMe;
261     final String dateTime;
262     final String messageFileUrl;
263
264     MessageBubble(
265         {this.sender, this.text, this.isMe, this.dateTime, this.messageFileUrl});
266
267     @override
268     Widget build(BuildContext context) {

```

```

269   return Padding(
270     padding: EdgeInsets.all(10.0),
271     child: Column(
272       crossAxisAlignment:
273         isMe ? CrossAxisAlignment.end : CrossAxisAlignment.start,
274       children: <Widget>[
275         Text(
276           sender,
277           style: TextStyle(
278             fontSize: 12.0,
279             color: Colors.black54,
280           ),
281         ),
282         (text != " ")
283           ? Container(
284             constraints: BoxConstraints(
285               maxWidth: 2 * (MediaQuery.of(context).size.width) / 3,
286             ),
287             child: Material(
288               borderRadius: isMe
289                 ? BorderRadius.only(
290                   topLeft: Radius.circular(30.0),
291                   bottomLeft: Radius.circular(30.0),
292                   bottomRight: Radius.circular(30.0),
293                 )
294                 : BorderRadius.only(
295                   topRight: Radius.circular(30.0),
296                   bottomLeft: Radius.circular(30.0),
297                   bottomRight: Radius.circular(30.0),
298                 ),
299             elevation: 10.0,
300             color: isMe ? Color(0xff81d4fa) : Colors.white,
301             child: Padding(
302               padding: EdgeInsets.symmetric(
303                 vertical: 12.0, horizontal: 20.0),
304               child: Text(
305                 text,
306                 style: TextStyle(
307                   fontSize: 15.0,
308                   color: Colors.black,
309                 ),
310               ),
311             ),

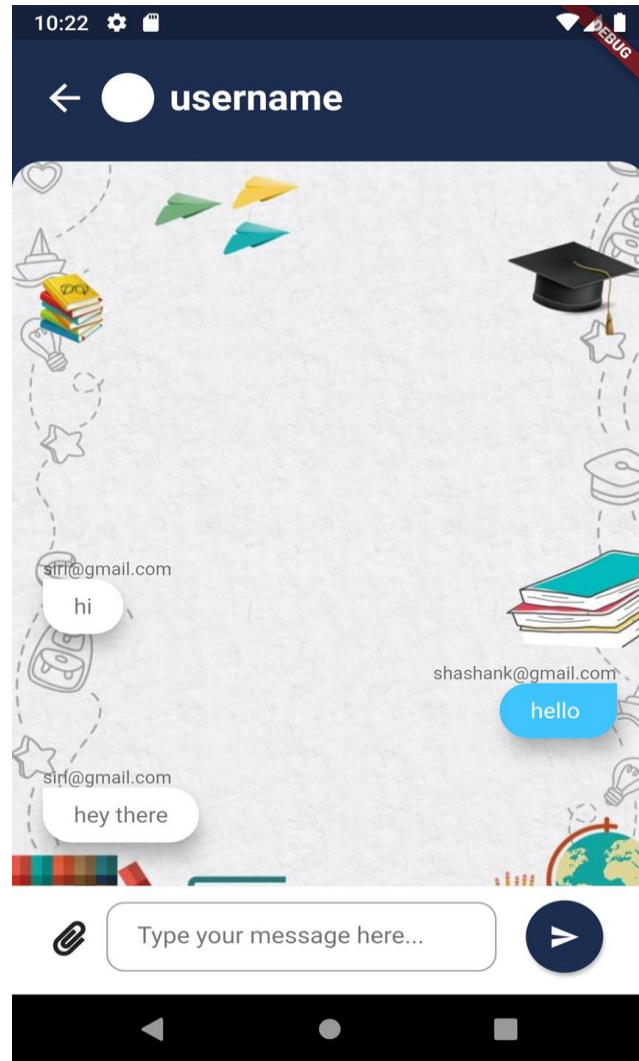
```

```

312     ),
313   )
314   : ClipRRect(
315     borderRadius: BorderRadius.circular(8.0),
316     child: Column(
317       mainAxisAlignment: MainAxisAlignment.spaceEvenly,
318       children: <Widget>[
319         Stack(
320           alignment: AlignmentDirectional.center,
321           children: <Widget>[
322             Container(
323               width: 130,
324               color: Colors.black87,
325               height: 80,
326               child: Column(
327                 mainAxisAlignment: MainAxisAlignment.spaceEvenly,
328                 children: <Widget>[
329                   Icon(
330                     Icons.insert_drive_file,
331                     color: Colors.lightBlueAccent,
332                   ),
333                   Text(
334                     'File',
335                     style: TextStyle(
336                       fontSize: 20,
337                       color: isMe
338                         ? Colors.lightBlueAccent
339                         : Colors.lightBlueAccent),
340                   ),
341                 ],
342             ),
343           ),
344         ],
345       ),
346     Container(
347       height: 40,
348       width: 130.0,
349       color: Colors.lightBlueAccent,
350       child: IconButton(
351         icon: Icon(
352           Icons.file_download,
353           color: Colors.white,
354         ),

```

```
355         onPressed: () {},
356       ),
357     )
358   ],
359 ),
360 ),
361   Text(
362     dateTime,
363     style: TextStyle(
364       fontSize: 10.0,
365       color: Colors.black54,
366     ),
367   ),
368 ],
369 ),
370 );
371 }
372 }
```



Profile Screen

```
1 import 'package:cloud_firestore/cloud_firestore.dart';
2 import 'package:comperio/helper_functions.dart';
3 import 'package:flutter/material.dart';
4 class ProfileScreen extends StatefulWidget {
5   final String id = 'ProfileScreen';
6   @override
7   _ProfileScreenState createState() => _ProfileScreenState();
8 }
9 class _ProfileScreenState extends State<ProfileScreen> {
10   QuerySnapshot snapshotUserInfo;
```

```

11 String userName;
12 String emailId;
13 String picURL;
14 getUserInfo() async {
15   userName = await HelperFunctions.getUserNameSharedPreference();
16   emailId = await HelperFunctions.getUserEmailSharedPreference();
17   picURL = await HelperFunctions.getUserPhotoUrlSharedPreference();
18 }
19 @override
20 void initState() {
21   super.initState();
22   getUserInfo();
23 }
24 @override
25 Widget build(BuildContext context) {
26   return MaterialApp(
27     home: Scaffold(
28       backgroundColor: Colors.teal,
29       body: SafeArea(
30         child: Container(
31           constraints: BoxConstraints.expand(),
32           decoration: BoxDecoration(
33             image: DecorationImage(
34               image: AssetImage('images/app-background-3.jpg'),
35               fit: BoxFit.cover),
36         child: Column(
37           crossAxisAlignment: CrossAxisAlignment.center,
38           mainAxisAlignment: MainAxisAlignment.center,
39           children: <Widget>[
40             Container(
41               decoration: BoxDecoration(
42                 color: Colors.white,
43                 shape: BoxShape.circle,
44                 boxShadow: [
45                   BoxShadow(
46                     blurRadius: 7.0,
47                     color: Colors.black,
48                   )
49                 ],
50             ),
51             child: CircleAvatar(
52               radius: 50.0, backgroundImage: NetworkImage(picURL)),
53           ),

```

```
54     SizedBox(  
55       height: 15.0,  
56     ),  
57     Text(  
58       userName != null ? userName : 'abc',  
59       style: TextStyle(  
60         fontSize: 40.0,  
61         color: Colors.white,  
62         fontWeight: FontWeight.bold,  
63       ),  
64     ),  
65     Text(  
66       '@Professor',  
67       style: TextStyle(  
68         color: Colors.teal[100],  
69         fontSize: 20.0,  
70         letterSpacing: 2.5,  
71         fontWeight: FontWeight.bold,  
72       ),  
73     ),  
74     SizedBox(  
75       height: 10.0,  
76     ),  
77     Card(  
78       color: Colors.white,  
79       margin: EdgeInsets.symmetric(vertical: 20.0, horizontal: 30.0),  
80       child: ListTile(  
81         visualDensity: VisualDensity(horizontal: 0, vertical: -4),  
82         leading: Icon(Icons.email, color: Colors.teal),  
83         title: Align(  
84           child: Text(  
85             emailId != null ? emailId : 'abc@gmail.com',  
86             style: TextStyle(  
87               color: Colors.teal[900],  
88               fontSize: 20.0,  
89             ),  
90           ),  
91           alignment: Alignment(-2.5, 0),  
92         ),  
93       ),  
94     ),  
95     Container(  
96       height: 50.0,
```



```
97      width: 170.0,
98      child: Material(
99        borderRadius: BorderRadius.circular(20.0),
100        shadowColor: Colors.blueAccent,
101        color: Colors.blueAccent,
102        elevation: 7.0,
103        child: GestureDetector(
104          onTap: () {},
105          child: Center(
106            child: Text(
107              'Send Message',
108              style: TextStyle(
109                color: Colors.white,
110                fontSize: 18.0,
111                fontWeight: FontWeight.bold,
112              ),
113            ),
114          ),
115        ),
116      ))
117    ],
118  ),
119 ),
120 ),
121 ),
122 );
123 }
124 }
```



Search Screen

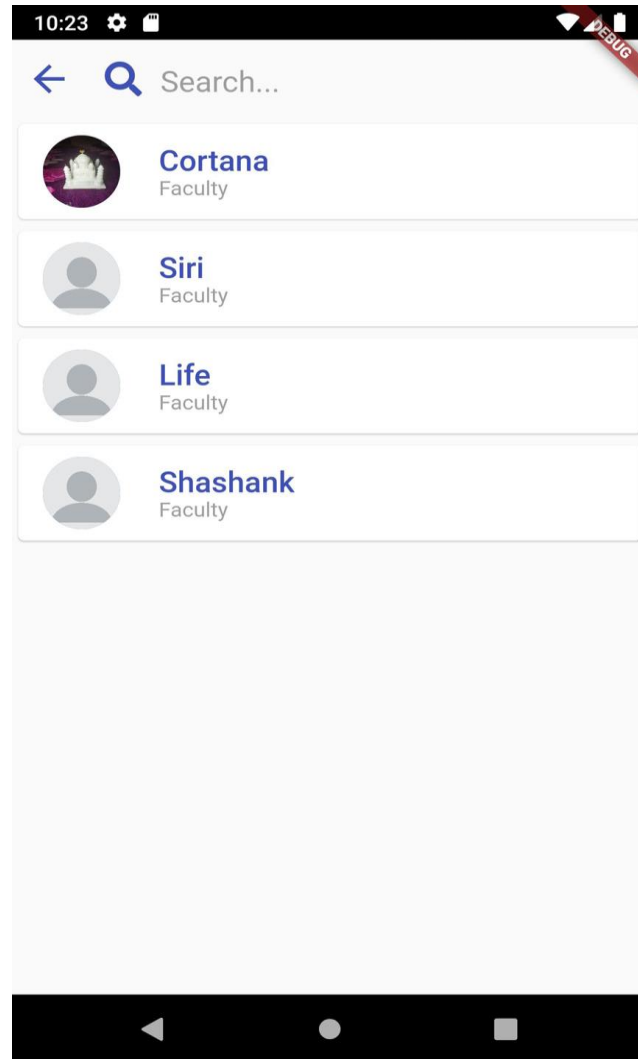
```
1 import 'package:cloud_firestore/cloud_firestore.dart';
2 import 'package:comperio/app_icons.dart';
3 import 'package:comperio/search_stream_builder.dart';
4 import 'package:flutter/material.dart';
5 import 'package:font_awesome_flutter/font_awesome_flutter.dart';
6 import '../constants.dart';
7 class SearchScreen extends StatefulWidget {
8   final String id = 'SearchScreen';
9   @override
10  _SearchScreenState createState() => _SearchScreenState();
```

```

11 }
12 class _SearchScreenState extends State<SearchScreen> {
13   final searchText = TextEditingController();
14   String name = "";
15   @override
16   Widget build(BuildContext context) {
17     return SafeArea(
18       child: Scaffold(
19         body: Container(
20           child: Column(
21             children: [
22               Row(
23                 children: [
24                   IconButton(
25                     icon: AppIcons(
26                       iconName: Icons.arrow_back,
27                       iconSize: 30.0,
28                       colour: Colors.indigo,
29                     ),
30                     onPressed: () {
31                       Navigator.pop(context);
32                     },
33                   ),
34                   Expanded(
35                     child: TextField(
36                       controller: searchText,
37                       cursorColor: Colors.indigo,
38                       autofocus: true,
39                       textInputAction: TextInputAction.search,
40                       style: TextStyle(
41                         height: 1.5,
42                         fontSize: 20.0,
43                       ),
44                       decoration: InputDecoration(
45                         prefixIcon: Icon(FontAwesomeIcons.search, color: Colors.indigo),
46                         suffixIcon: name.length > 0
47                           ? IconButton(
48                             icon: Icon(
49                               FontAwesomeIcons.times,
50                               color: Colors.blueGrey,
51                             ),
52                             onPressed: () {
53                               setState(() {

```

```
54         searchText.clear();
55         name = "";
56     });
57 },
58     ) : null,
59     hintText: 'Search...',
60     hintStyle: TextStyle(color: Colors.grey, fontSize: 20.0,),
61     border: InputBorder.none,
62 ),
63 onChanged: (value) {
64     setState() {
65         name = value;
66     });
67 },
68 ),
69 ),
70 ],
71 ),
72 Expanded(
73     child: SearchStreamBuilder( username: name, ),
74 ),
75 ],
76 ),
77 ),
78 ),
79 );
80 }
81 }
```



Contacted Person Screen

```
1 import 'package:comperio/app_icons.dart';
2 import 'package:comperio/choice.dart';
3 import 'package:comperio/constants.dart';
4 import 'package:comperio/contact_popup_menu.dart';
5 import 'package:comperio/helper_functions.dart';
6 import 'package:comperio/screen/searchScreen.dart';
7 import 'package:firebase_auth/firebase_auth.dart';
8 import 'package:flutter/material.dart';
9 import 'package:font_awesome_flutter/font_awesome_flutter.dart';
10
```

```

11 class ContactedPersonScreen extends StatefulWidget {
12   final String id = 'ContactedPersonScreen';
13
14   @override
15   _ContactedPersonScreenState createState() => _ContactedPersonScreenState();
16 }
17 List<Choice> choices = const <Choice>[
18   const Choice(title: 'Profile', icon: FontAwesomeIcons.user),
19   const Choice(title: 'Change Password', icon: FontAwesomeIcons.key),
20   const Choice(title: 'Log out', icon: FontAwesomeIcons.signOutAlt),
21 ];
22
23 class _ContactedPersonScreenState extends State<ContactedPersonScreen> {
24   final user = FirebaseAuth.instance.currentUser;
25
26
27   @override
28   void initState() {
29     getUserInfo();
30     super.initState();
31   }
32
33   getUserInfo() async => Constants.myName = await HelperFunctions.getUserNameSharedPreference();
34
35   @override
36   Widget build(BuildContext context) {
37     return Container(
38       decoration: BoxDecoration(
39         color: Color(0xff311b92),
40         image: DecorationImage(
41           image: AssetImage('images/app-background.jpg'),
42           fit: BoxFit.cover,
43         ),
44       ),
45       constraints: BoxConstraints.expand(),
46       child: Scaffold(
47         floatingActionButton: FloatingActionButton(
48           onPressed: () {
49             },
50           child: AppIcons(
51             iconName: Icons.group_add,
52             iconSize: 30.0,
53             colour: Colors.white,

```

```

54     ),
55     backgroundColor: Color(0xff0d47a1),
56     ),
57     resizeToAvoidBottomInset: false,
58     resizeToAvoidBottomPadding: false,
59     backgroundColor: Colors.transparent,
60     body: Column(
61       crossAxisAlignment: CrossAxisAlignment.start,
62       children: <Widget>[
63         Container(
64           padding: EdgeInsets.only(
65             top: 40.0, left: 20.0, right: 20.0, bottom: 20.0),
66           child: Row(
67             mainAxisAlignment: MainAxisAlignment.spaceBetween,
68             children: [
69               Text(
70                 'Comperio',
71                 style: KAppNameTextStyle,
72               ),
73               Row(
74                 mainAxisAlignment: MainAxisAlignment.end,
75                 children: [
76                   IconButton(
77                     icon: AppIcons(
78                       iconName: Icons.search,
79                       colour: Colors.white,
80                       iconSize: 35.0),
81                     onPressed: () {
82                       Navigator.pushNamed(context, SearchScreen().id);
83                     },
84                   ),
85                   ContactPopupMenu(choices: choices),
86                 ],
87               ),
88             ],
89           ),
90         ),
91         Expanded(
92           child: Container(
93             padding: EdgeInsets.symmetric(horizontal: 20.0),
94             decoration: BoxDecoration(
95               color: Colors.white,
96               borderRadius: BorderRadius.only(

```

```
97         topLeft: Radius.circular(20.0),
98         topRight: Radius.circular(20.0),
99       ),
100     ),
101     child: ListView(
102       ),
103     ),
104   ),
105 ],
106 ),
107 ),
108 );
109 }
110 }
```

Feedback screen

```
1  import 'package:comperio/app_icons.dart';
2  import 'package:comperio/constants.dart';
```



```

3  import 'package:flutter/cupertino.dart';
4  import 'package:flutter/material.dart';
5  import 'package:font_awesome_flutter/font_awesome_flutter.dart';
6  class FeedbackScreen extends StatefulWidget {
7    final String id = 'FeedbackScreen';
8    @override
9    _FeedbackScreenState createState() => _FeedbackScreenState();
10 }
11 class _FeedbackScreenState extends State<FeedbackScreen> {
12   String myFeedbackText = "COULD BE BETTER";
13   double sliderValue = 0.0;
14   IconData myFeedback = FontAwesomeIcons.sadTear;
15   Color myFeedbackColor = Colors.white;
16   @override
17   Widget build(BuildContext context) {
18     return Scaffold(
19       appBar: AppBar(
20         backgroundColor: Colors.blueAccent,
21         leading: IconButton(
22           icon: AppIcons(
23             iconName: Icons.arrow_back,
24             iconSize: 25.0,
25             colour: Colors.white,
26           ),
27           title: Center(child: Text("Feedback")),
28           actions: <Widget>[
29             IconButton(
30               icon: Icon(FontAwesomeIcons.solidStar),
31             ],
32           ),
33       body: Container(
34         decoration: BoxDecoration(
35           image: DecorationImage(
36             image: AssetImage("assets/container_background2.jpg"),
37             fit: BoxFit.cover),
38         ),
39         child: Column(
40           children: <Widget>[
41             Container(
42               child: Padding(
43                 padding: EdgeInsets.all(15.0),
44                 child: Container(
45                   child: Center(

```

```

46      child: Text(
47        "On a scale of 1 to 5, rate our app",
48        style: KFeedbackHintTextStyle,
49        textAlign: TextAlign.center,
50      ),
51    )),
52  ),
53 ),
54 SizedBox(height: 30.0),
55 Container(
56   child: Material(
57     color: Color(0xFFf2fcfe),
58     elevation: 14.0,
59     borderRadius: BorderRadius.circular(24.0),
60     shadowColor: Color(0xFF70e1f5),
61     child: Container(
62       width: 350.0,
63       height: 450.0,
64       child: Column(
65         mainAxisAlignment: MainAxisAlignment.center,
66         children: <Widget>[
67           Padding(
68             padding: EdgeInsets.all(16.0),
69             child: Container(
70               child: Text(
71                 myFeedbackText,
72                 style: KFeedbackTextStyle,
73               ),
74             ),
75           ),
76           Padding(
77             padding: EdgeInsets.all(10.0),
78             child: Container(
79               child: AppIcons(
80                 iconName: myFeedback,
81                 iconSize: 120.0,
82                 colour: myFeedbackColor,
83               ),
84             ),
85           ),
86           Padding(
87             padding: EdgeInsets.all(10.0),
88             child: Container(

```

```

89         child: Slider(
90             min: 0.0,
91             max: 5.0,
92             divisions: 5,
93             value: sliderValue,
94             activeColor: Color(0xFF005C97),
95             inactiveColor: Color(0xFF70e1f5),
96             onChanged: (newValue) {
97                 setState() {
98                     sliderValue = newValue;
99                     if (sliderValue <= 0.4) {
100                         myFeedback = FontAwesomeIcons.neos;
101                         myFeedbackColor = Color(0xeeb6fbff);
102                         myFeedbackText = "REMARK";
103                     }
104                     if(sliderValue >= 0.5 && sliderValue <= 1.0){
105                         myFeedback = FontAwesomeIcons.sadTear;
106                         myFeedbackColor = Color(0xFFED213A);
107                         myFeedbackText = "COULD BE BETTER";
108                     }
109                     if (sliderValue >= 1.1 &&
110                         sliderValue <= 2.0) {
111                         myFeedback = FontAwesomeIcons.frown;
112                         myFeedbackColor = Color(0xFFFF27121);
113                         myFeedbackText = "BELOW AVERAGE";
114                     }
115                     if(sliderValue >= 2.1 && sliderValue <= 3.0){
116                         myFeedback = FontAwesomeIcons.meh;
117                         myFeedbackColor = Colors.amber;
118                         myFeedbackText = "NORMAL";
119                     }
120                     if(sliderValue >= 3.1 && sliderValue <= 4.0){
121                         myFeedback = FontAwesomeIcons.smile;
122                         myFeedbackColor = Colors.green;
123                         myFeedbackText = "GOOD";
124                     }
125                     if(sliderValue >= 4.1 && sliderValue <= 5.0){
126                         myFeedback = FontAwesomeIcons.laugh;
127                         myFeedbackColor = Color(0xFF00B4DB);
128                         myFeedbackText = "EXCELLENT";
129                     }
130                 });
131             },

```

```
132         ),
133     ),
134 ),
135     Padding(
136         padding: EdgeInsets.all(15.0),
137         child: Container(
138             child: Text(
139                 "Your Rating: $sliderValue",
140                 style: KFeedbackTextStyle,
141             ),
142         ),
143     ),
144     Padding(
145         padding: EdgeInsets.all(10.0),
146         child: Container(
147             height: 45.0,
148             width: 150.0,
149             child: RaisedButton(
150                 child: Text( 'Submit', style: TextStyle(color: Colors.white), ),
151                 shape: RoundedRectangleBorder(
152                     borderRadius: BorderRadius.circular(10.0),
153                 ),
154                 color: Colors.blueAccent,
155             ),
156         ],
157     )),
158 ),
159 ),
160 ],
161 ),
162 ),
163 );
164 }
165 }
```



The screenshot shows a mobile application interface for 'Comperio'. At the top, there is a header with a blue background featuring a stylized mountain and a glowing sphere, with the word 'Comperio' in white. Below the header, a white card contains the text: 'On a scale of 1 to 5, rate the overall performance of the teacher?'. Underneath this text is the word 'NORMAL' in bold black letters. A yellow circular icon with a neutral face is centered below the text. Below the icon is a horizontal slider bar with a blue dot indicating the rating. Below the slider, the text 'Your Rating: 3.0' is displayed. At the bottom of the card is a blue button with the word 'Submit' in white. The entire interface is framed by a black border, and the bottom of the screen shows standard Android navigation icons.

On a scale of 1 to 5, rate the overall performance of the teacher?

NORMAL



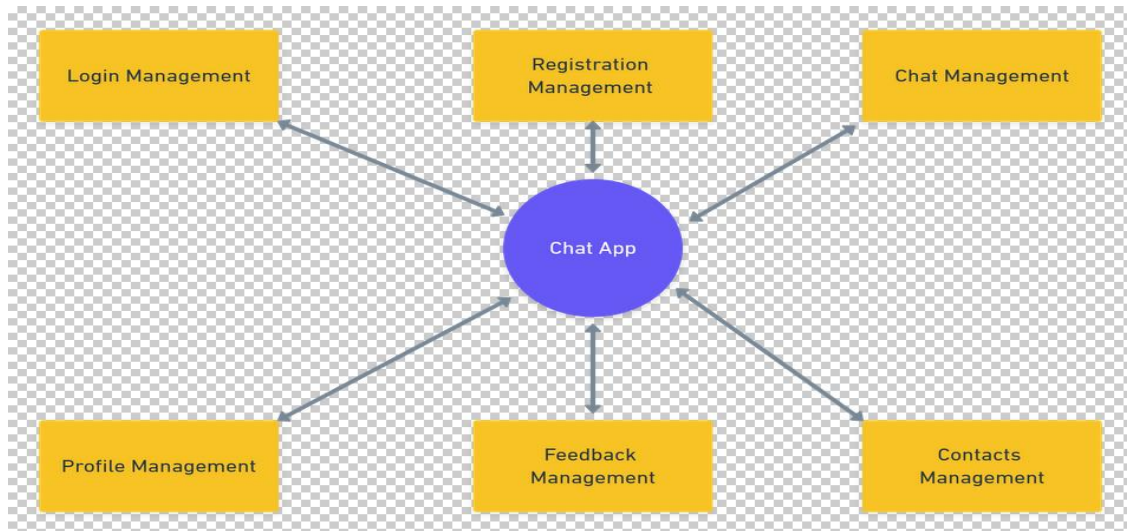


Your Rating: 3.0

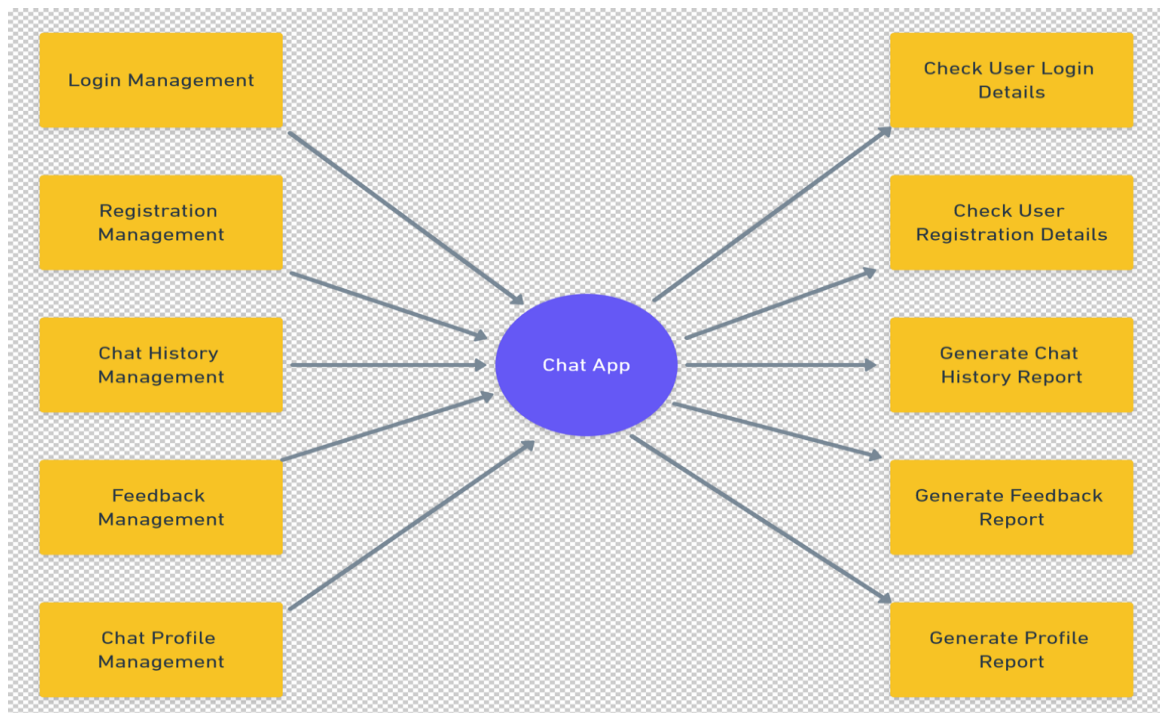
Submit

Data Flow Diagram:

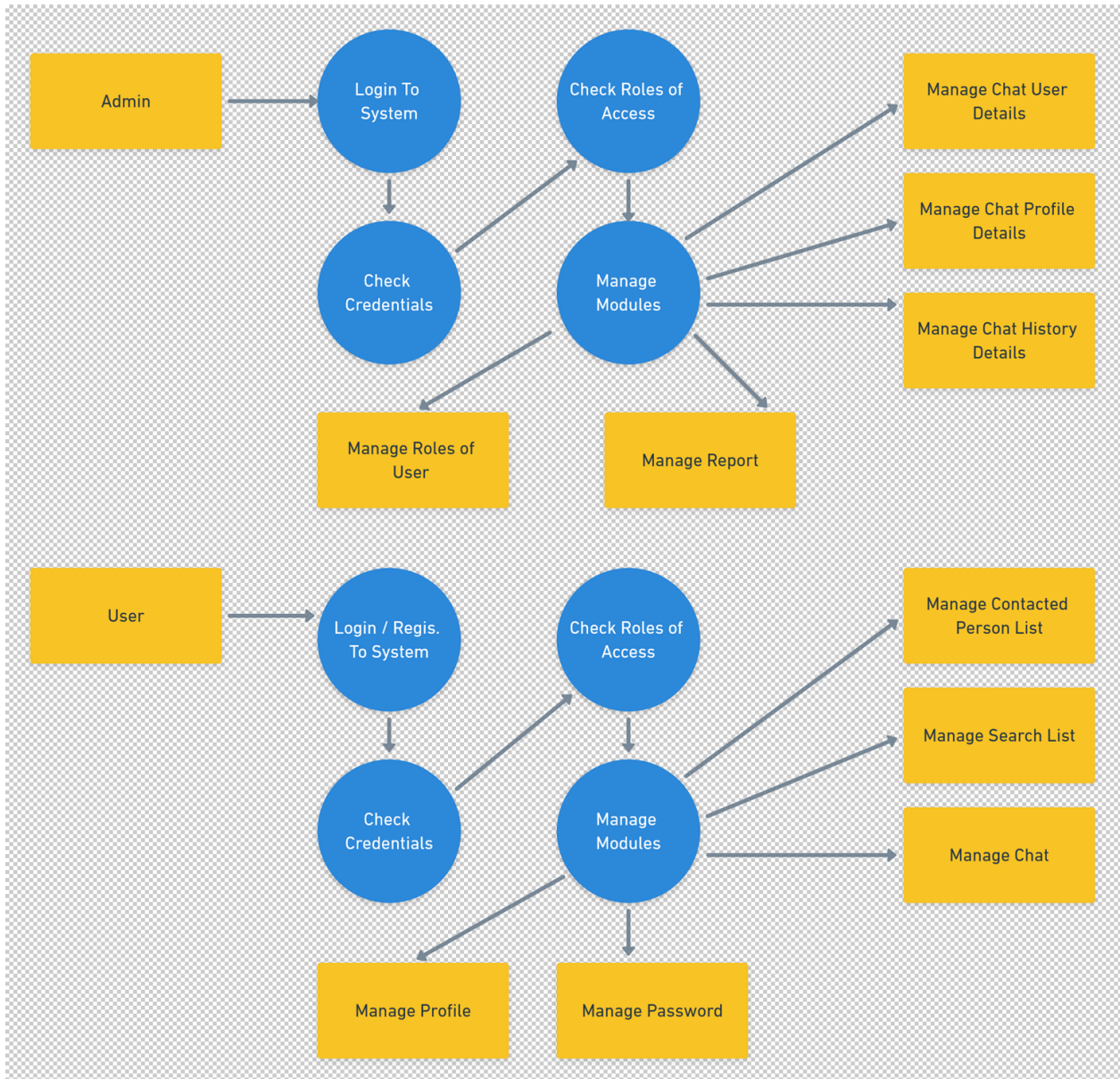
Level 0 DFD



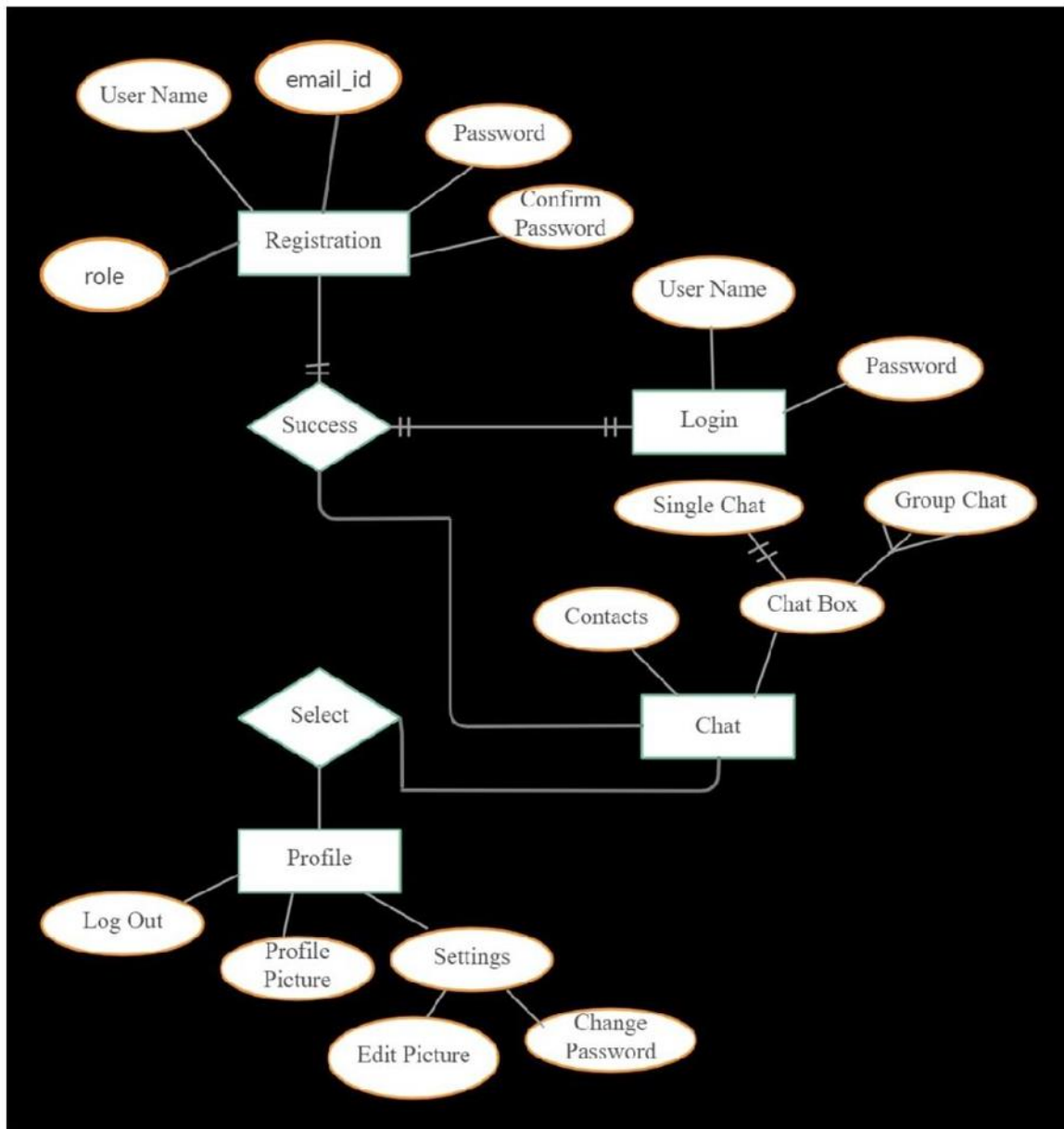
Level 1 DFD



Level 2 DFD



ER Diagram:



Conclusion:

From above discussion, it is evident how chat applications are purposed to play various roles in addition to acting as a chat communication medium.

The main objective of the project is to develop a Secure Chat Application. We had taken a wide range of literature review in order to achieve all the tasks, where we came to know about some of the products that are existing in the market. We made detailed research on that path to cover the loopholes that existing systems are facing and to eradicate them in our application. In the process of research, we came to know about the latest technologies and different algorithms.

References:

1. Reference for flutter documentation <https://flutter.dev/>.
2. About Android Studio <https://developer.android.com/studio>.
3. About Dart Language <https://dart.dev/>.
4. Some of the contents are from Internet.