



Sri Lankan Institute of Information Technology

Crruent Trends of Software Engineering (SE4010)

Flutter Project - Learn A Disease

IT17134668 Gamage V.S

Table of Contents

1	Introduction	1
2	Technology and resources	2
2.1	Flutter	2
2.2	Dart.....	4
2.3	Cloud Firebase	4
3	Requirements	5
3.1	User Story 01	5
3.2	User Story 02	5
3.3	Use Case Scenario.....	5
3.4	Requirements Gathered	6
4	System workflow execution	7
4.1	App Icon.....	7
4.2	Functionality Description	8
4.3	Validating forms.	15
5	Project File Structure	21
6	References	22
7	Appendix	23
7.1	Models.....	23
7.2	Views.....	24

TABLE OF FIGUERS

Figure 2.1 Widget Hierarchy of the entire application.....	2
Figure 2.2 Widget Hierarchy of the individual functionality.....	3
Figure 2.3 users Collection.....	4
Figure 4.1 Application Icon.....	7
Figure 4.2 Slider.....	8
Figure 4.3 Welcome Page.....	9
Figure 4.4 Register Page.....	10
Figure 4.5 Login Page.....	11
Figure 4.6 Disease List and More Details on searched disease.....	12
Figure 4.7 User Profile and Update Profile form.....	13
Figure 4.8 User Profile and update profile page of a male user.....	14
Figure 4.9 Register Page Field Validation.....	15
Figure 4.10 Validation for username existence.....	16
Figure 4.11 Success message for user registration.....	17
Figure 4.12 Login Page Field Validation.....	18
Figure 4.13 Page for invalid credentials and Success message on user login.....	19
Figure 4.14 Field validation for disease name and Error message for invalid disease name.....	20
Figure 5.1 Project file Structure.....	21

LIST OF TABLES

Table 3.1 Use Case Scenario 1.....5

Table 3.1 Use Case Scenario 2.....6

1 INTRODUCTION

At present learning about diseases has become very vital in our day-to-day lives. Most of the infectious diseases are disorders caused by organisms such as bacteria, viruses, fungi or parasites. Being aware of diseases has resulted in higher survival rates from diseases and injuries that ultimately lead to a healthy world community.

The following report reflects the “Learn Disease” Flutter application that is developed for academic purposes of the module Current Trends in Software Engineering. The application is the combination of both frontend and backend with a database connection.

Frontend user interfaces and the backend of the application are developed using Flutter. Cloud firebase is used as the application's database.

The application consists of two portals as the admin portal and user portal. This report will be describing about User portal where it will enable the end-user to register to the application by creating an user account and users will be able to login to the application through login page. Logged users will be able to view all the diseases available, the user can tap on the disease for more details. Users can view, update their user accounts and if needed the user account can be deleted.

2 TECHNOLOGY AND RESOURCES

2.1 Flutter

Flutter is the technology that is used to develop cross-platform applications. Having the potential to build impressive and expensive user interfaces fast, high performance and hot reload [1] are major advantages of using flutter. The applications both frontend and backend are developed using flutter.

The core concept of the Flutter framework is the widget. Therefore, the “Learn disease” application is a collection of widgets. The high-level widget hierarchy of the application is specified in the following diagram.

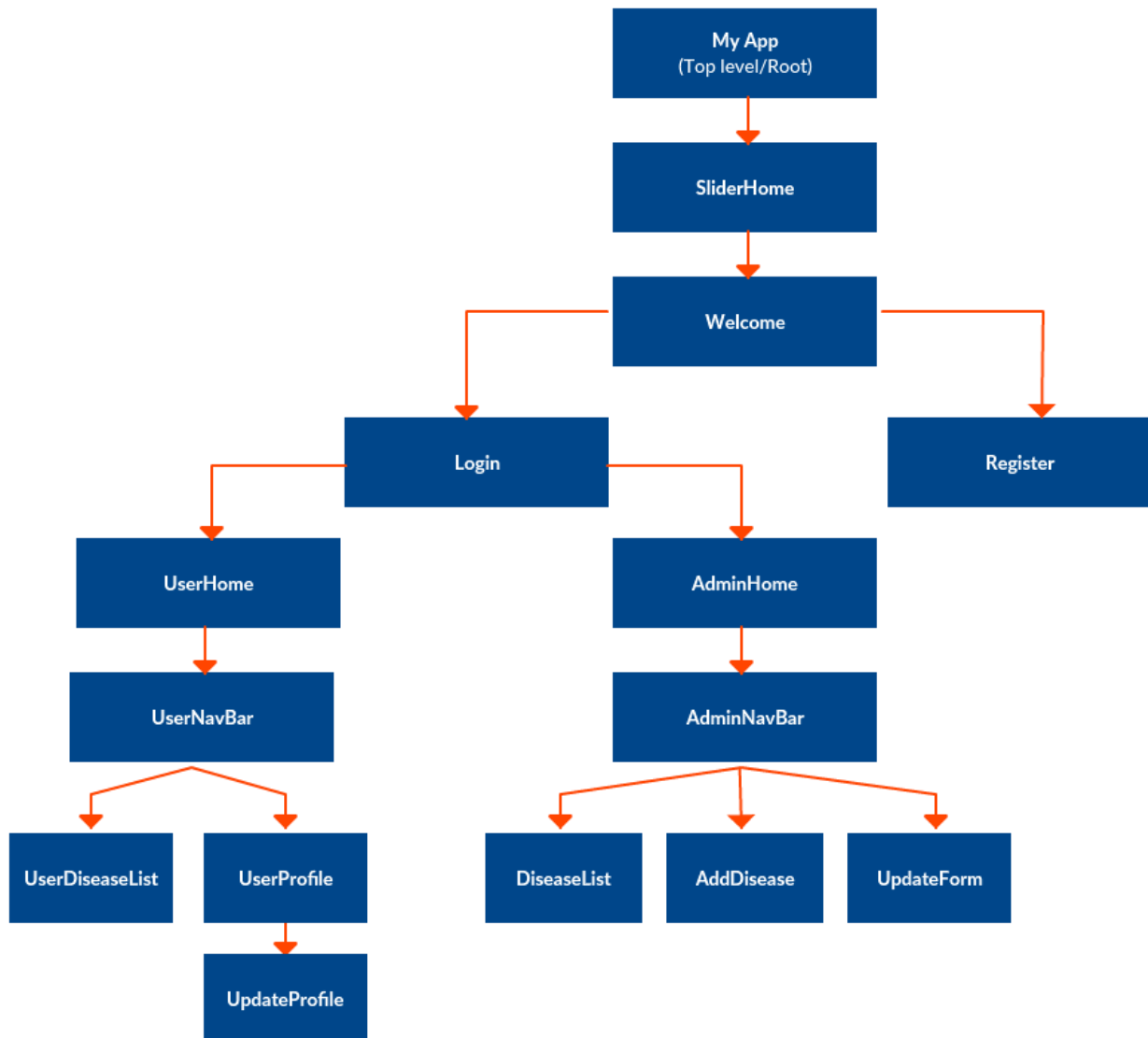


Figure 2.1 Widget Hierarchy of the entire application

Widget hierarchy of the individual functionality

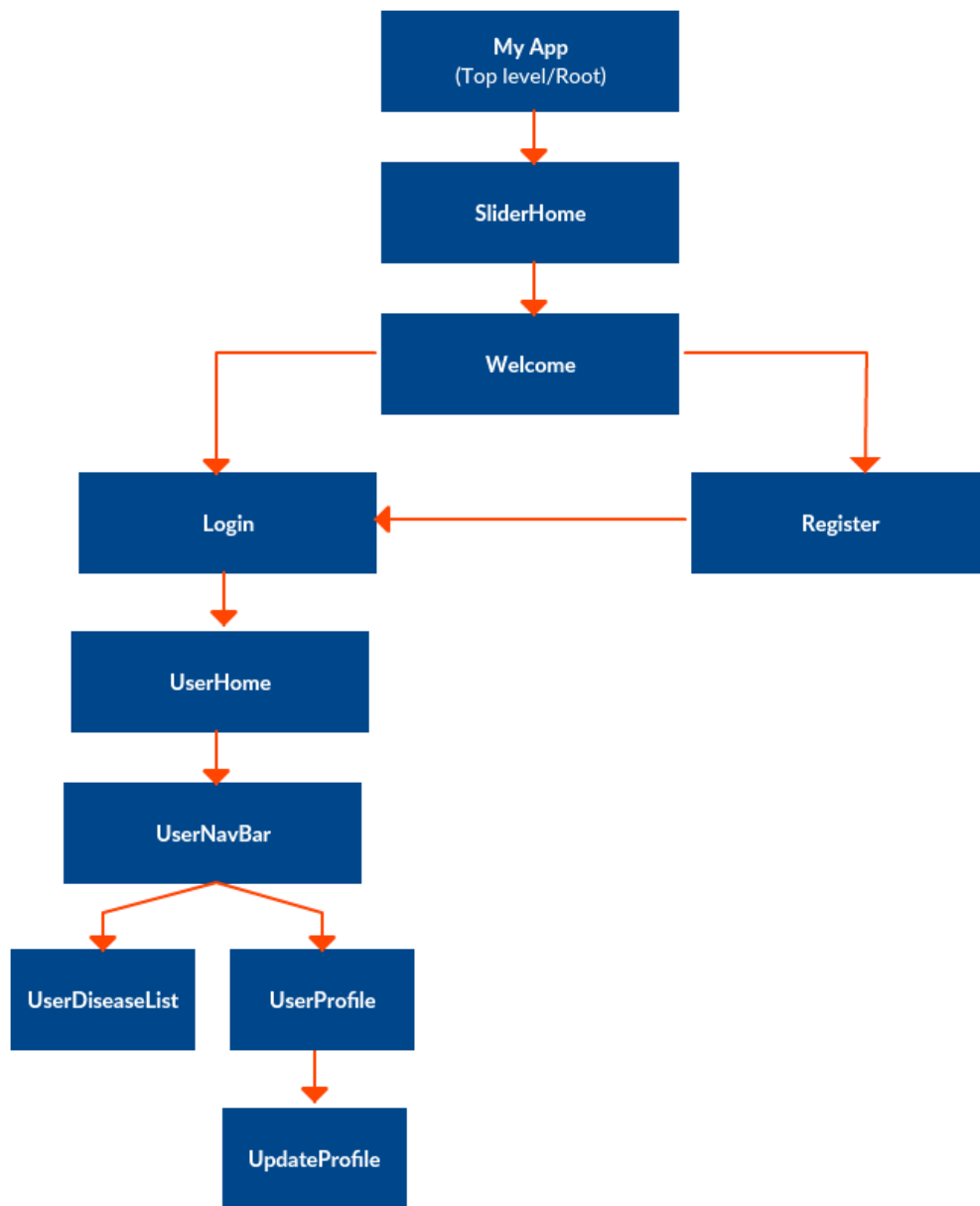


Figure 2.2 Widget Hierarchy of the individual functionality

2.2 Dart

Dart is an object-oriented, strongly typed and garbage-collected programming language that is used by Flutter. Using Just-In-Time compilation is a special aspect of Dart.



2.3 Cloud Firestore

Cloud Firestore is the database that is used for this application along with Flutter. Cloud Firestore is used to store data about users in “users” collection.

A document in “users” collection consists of documents with the fields' firstname(user's first name), lastname(user's last name), sex(gender of the user – Male, Female), username, password.

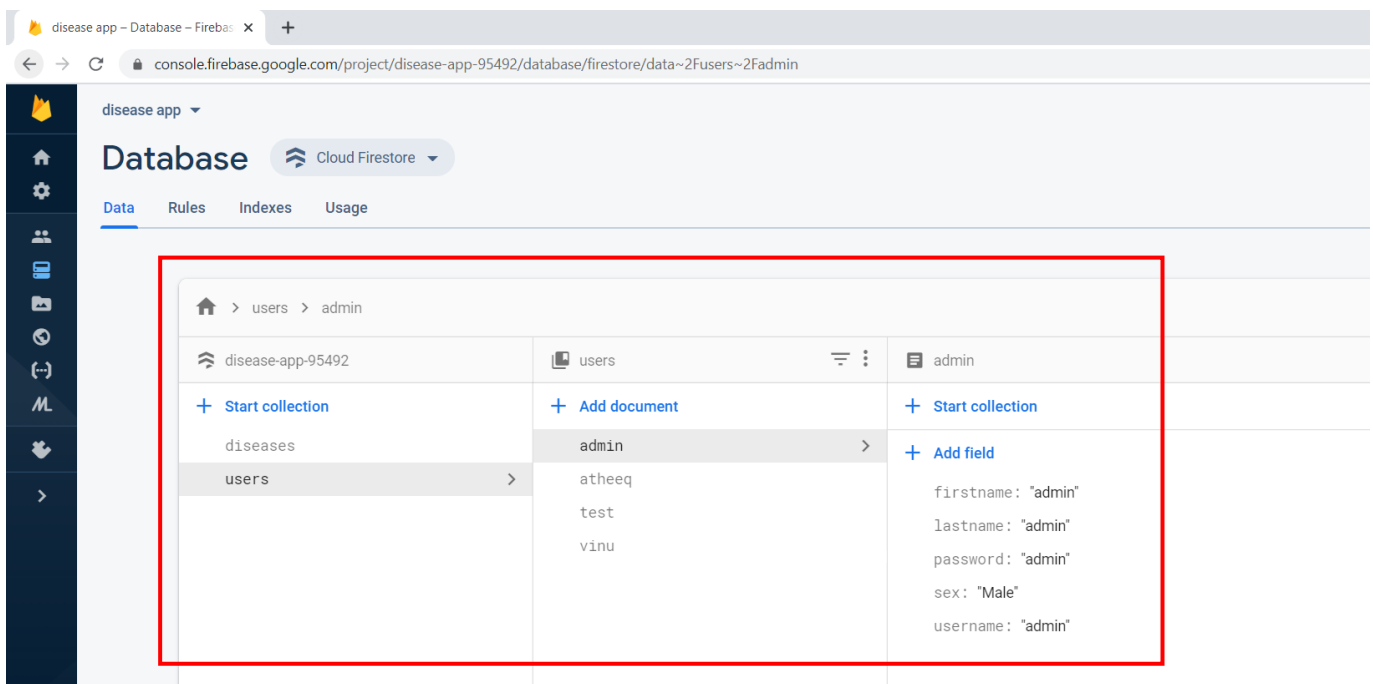


Figure 2.2 users Collection

3 REQUIREMENTS

3.1 User Story 01

As a learn a disease app user

I need to register and log in to the application

So that I can manage my accounts

3.2 User Story 02

As a learn a disease app user

I need to search for a disease from a disease list

So that I can know more details about the searched disease

3.3 Use Case Scenario

3.1 Use Case Scenario 1

Use case Id	01
Use case name	Log in to the application that is registered
Actor	End-user
Pre-Conditions	The application must be running and available The user must navigate to login page The user must have a valid account.
Post-Conditions	The user being able to create an account
Main Scenario	1. The user enters the username. 2. The user enters the password. 3. The user clicks “login” button.
Extensions	1.a. If the username is empty an error message will be prompt. b. If the username is invalid a toast message will be prompt. 3.a. If the password is empty an error message will be prompt. b. If the password is invalid a toast message will be prompt.

3.2 Use Case Scenario 2

Use case Id	02
Use case name	Searching for a disease using disease name
Actor	End-user
Pre-Conditions	The application must be running and available. The user must be logged into the system. The user must navigate to the disease list page in user home.
Post-Conditions	The user being search for a disease and being able to get more details about the disease.
Main Scenario	1. The user enters the disease. 2. The user clicks the “search” button.
Extensions	1.a. If the disease name is empty an error message will be prompt b. If the disease name is not available a toast message will be prompt.

3.4 Requirements Gathered

According to the above-stated user story 01 user can create an account to register to the application and using valid credentials user will be able to login to the application. User story 02, the user should be able to view a list of diseases that the user can read and learn about diseases using the list.

Requirement 01 – Registering and log in to the application and account management.

Requirement 02 – Searching for a disease from the disease list to get more details about the searched disease.

4 SYSTEM WORKFLOW EXECUTION

4.1 App Icon

The mobile application that is created to learn about diseases is named as “Learn Disease” with a significant app icon. With help of flutter_launcher_icons dependency the app icon was cutermized and the application name was custerized by changing the android:label field in AndroidManifest.xml file.

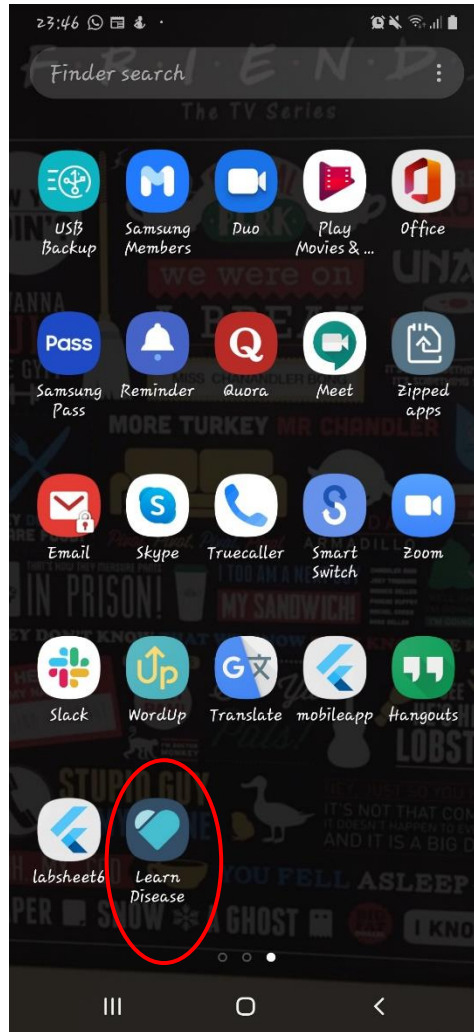


Figure 4.1 Application Icon

4.2 Functionality Description

Initially, the application will illustrate a slider providing information on the application “learn disease”. The user either can skip the slider process, or can go through each slide to get to know more about the application [2].

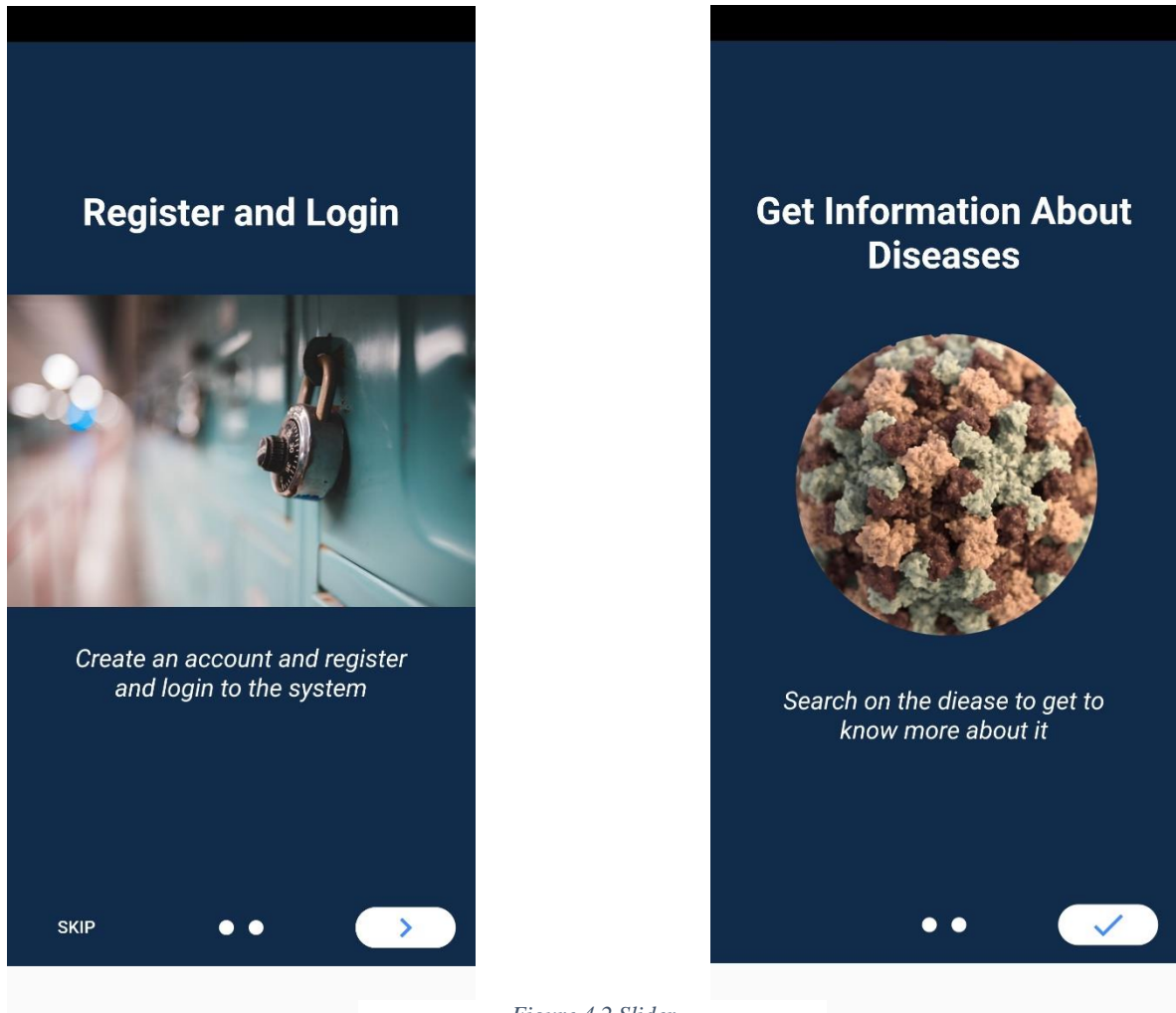


Figure 4.2 Slider

Once the user clicks “done icon” in the final slide the application will direct the user to the welcome page [3] where the user can create an account(sign up) or login to the application.

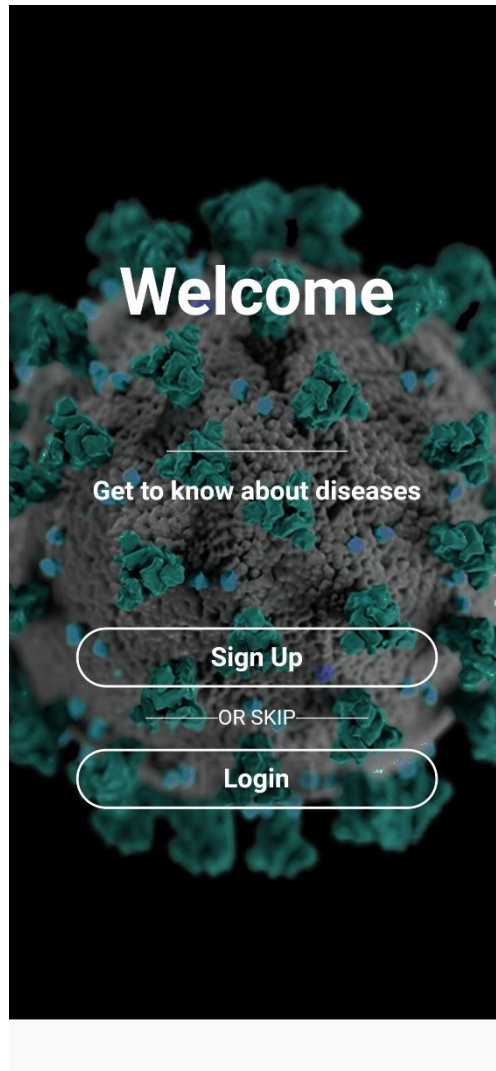
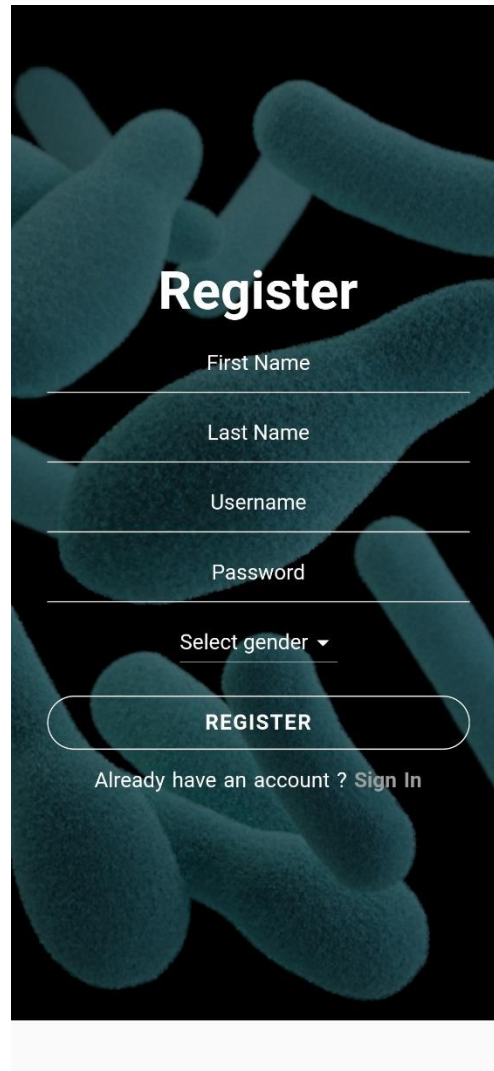


Figure 4.3 Welcome Page

If the user clicks “sign up” it will direct to register page where the user can create an account. User can provide their first name, last name, username, password, and gender.

The image shows a mobile application's registration screen. The background is dark with a pattern of glowing teal, pill-shaped elements. The word "Register" is centered at the top in a large, white, sans-serif font. Below it are five input fields, each with a white label and a horizontal line: "First Name", "Last Name", "Username", "Password", and "Select gender" (which includes a small downward arrow). Below these fields is a rounded rectangular button with a white border and the word "REGISTER" in bold, uppercase, white letters. At the bottom, the text "Already have an account ? Sign In" is displayed in a smaller white font, with "Sign In" acting as a link.

Register

First Name

Last Name

Username

Password

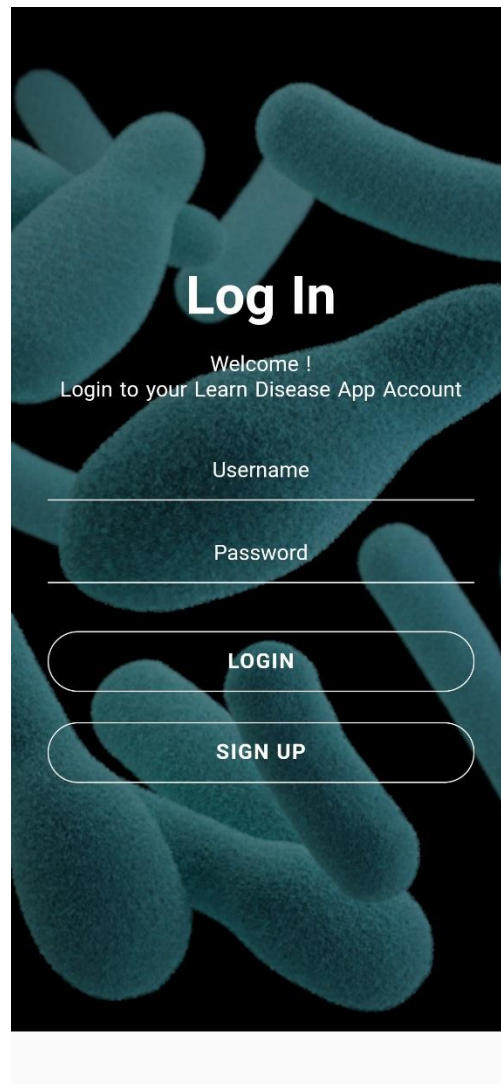
Select gender ▼

REGISTER

Already have an account ? [Sign In](#)

Figure 4.4 Register Page

Next, the user can provide user credentials and successfully log in to the application.

The image shows a mobile application login screen. The background is black with a pattern of glowing teal, pill-shaped bacteria. The text 'Log In' is centered in a large, white, sans-serif font. Below it, the text 'Welcome !' and 'Login to your Learn Disease App Account' are centered in a smaller white font. There are two input fields: 'Username' and 'Password', each with a white underline. Below the input fields are two rounded rectangular buttons with white outlines and black text. The top button is labeled 'LOGIN' and the bottom button is labeled 'SIGN UP'.

Log In

Welcome !
Login to your Learn Disease App Account

Username

Password

LOGIN

SIGN UP

Figure 4.5 Login Page

The User home page consists of the user navigation bar(curved navigation bar) [4] which is used to navigate to the disease list page and user profile page [5].

By default, the user will be navigated to the page with the diseases. The user can search for a disease to learn more about that specific disease by giving the disease name. Users can navigate back to the user's home.

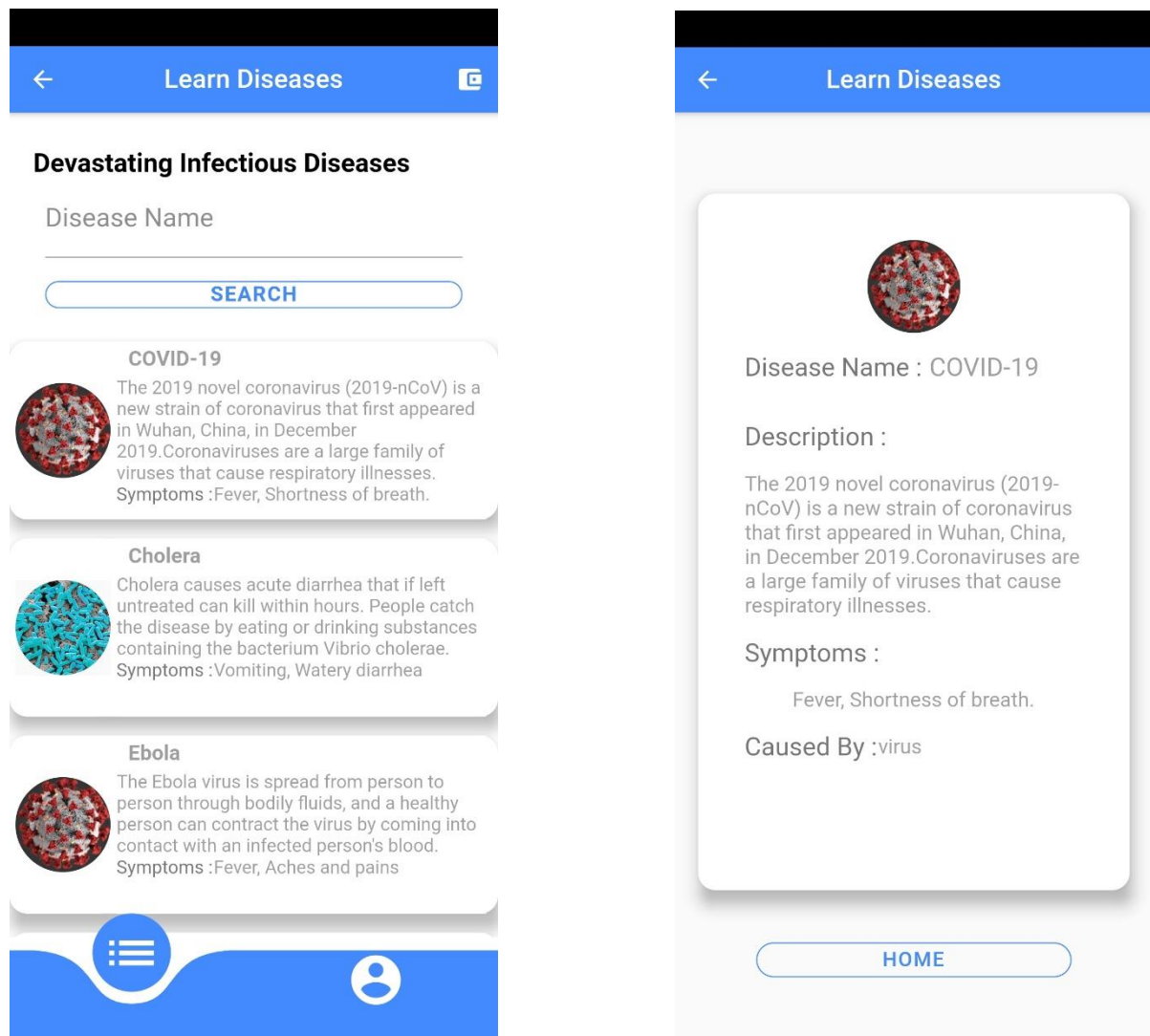


Figure 4.6 Disease List and More Details on searched disease

Users also can navigate to the user's profile where the user can view all user details. There if a user requires to update user details, by clicking the update profile it will navigate to the page to update user details.

The image displays two mobile application screens side-by-side. The left screen, titled 'Learn Diseases' in a blue header, shows a user profile for 'vinu'. It includes a circular profile picture, a 'Welcome vinu' greeting, and a list of user details: First Name :Vinu, Last Name :Gamage, Username :vinu, Password :vinu1, and Gender :Female. Below the details are two buttons: 'UPDATE PROFILE' and 'DELETE PROFILE'. The right screen, titled 'Update Profile', features the same profile picture and a form to update the user's information. The form fields are: Username (vinu), First Name (Vinu), Last Name (Gamage), Password (masked with dots), and a gender dropdown menu currently set to 'Female'. An 'UPDATE PROFILE' button is located at the bottom of the form.

Figure 4.7 User Profile and Update Profile form

If a user requires to delete user profile user can click delete profile soo that the profile will be deleted and will be navigated to register page to re-create an account.

According to the gender the profile picture of the end user will be changed.

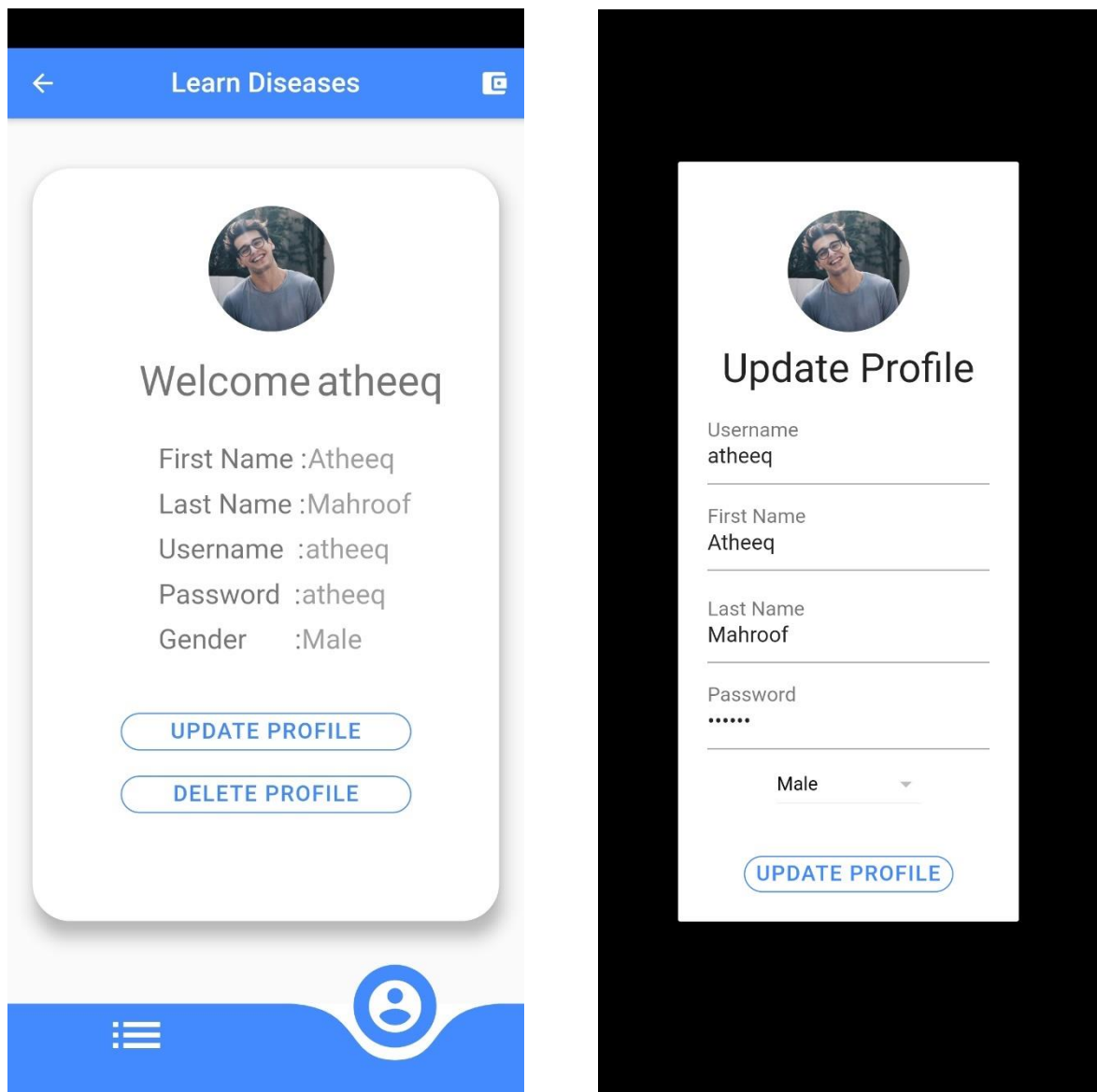


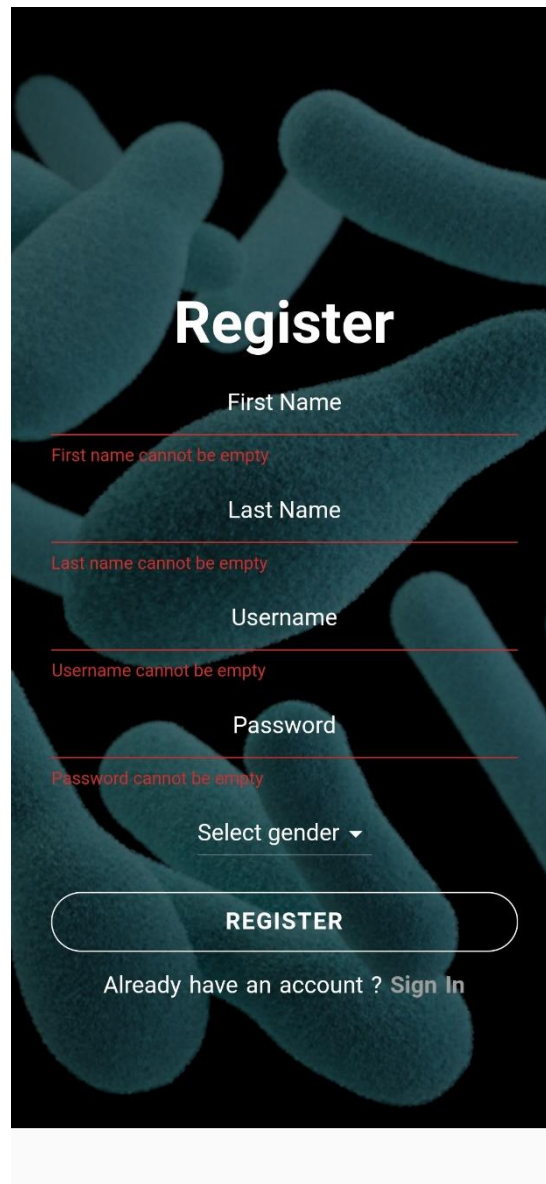
Figure 4.8 User Profile and update profile page of a male user

If a user requires to sign out from the application user can tap the logout icon on the app bar.

4.3 Validating forms.

Registration form validation

If the data is not provided application will validate the form fields.



Register

First Name
First name cannot be empty

Last Name
Last name cannot be empty

Username
Username cannot be empty

Password
Password cannot be empty

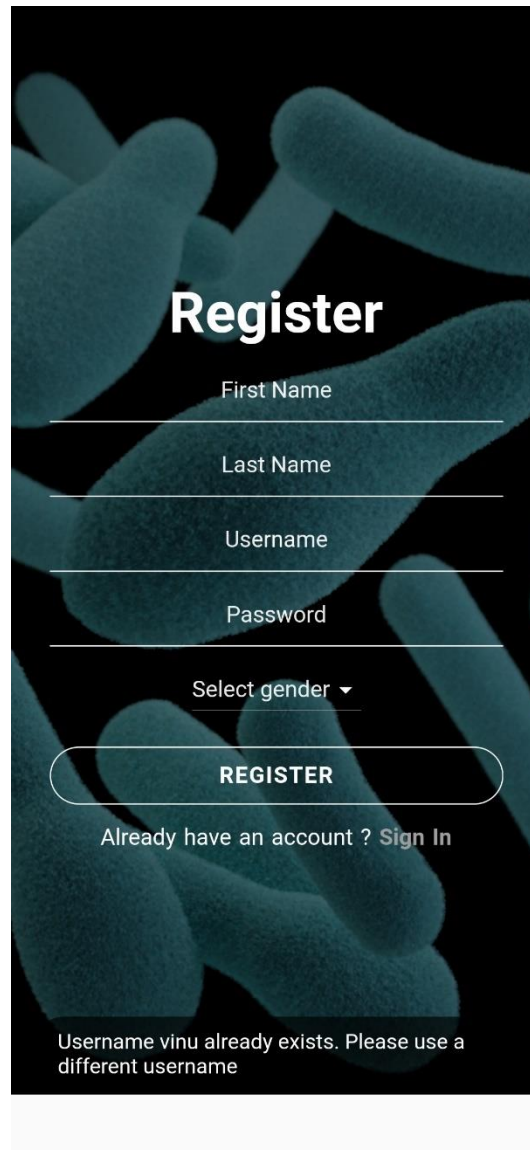
Select gender ▼

REGISTER

Already have an account ? [Sign In](#)

Figure 4.9 Register Page Field Validation

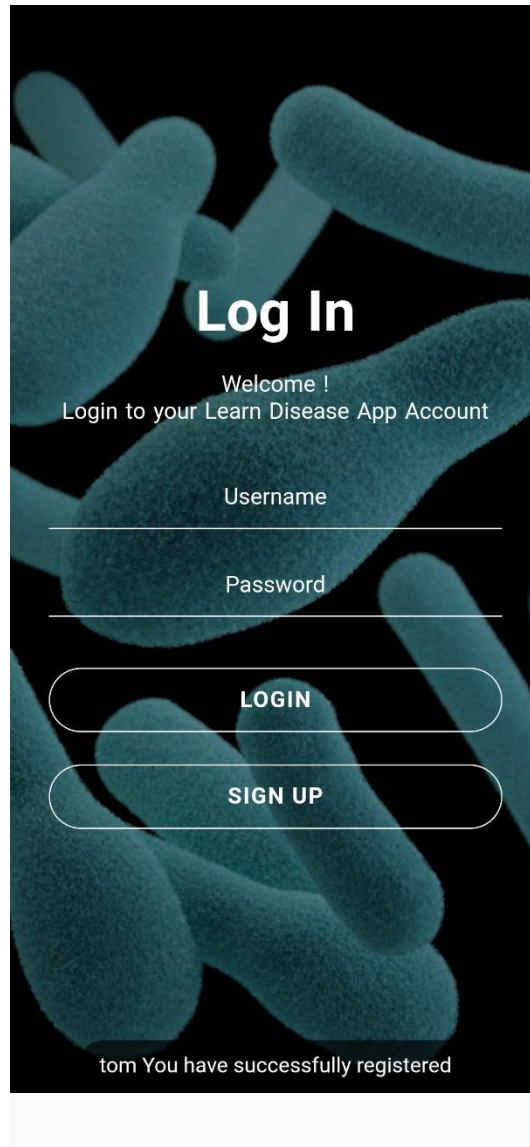
The user must enter a unique username(a username that does not contain in the database) creating the account. If the user provides the same username a toast message [6] will be generated to alert the user.



The image shows a mobile application's registration screen. The background is dark with a teal, abstract pattern. The word "Register" is prominently displayed in white. Below it are input fields for "First Name", "Last Name", "Username", and "Password". There is also a "Select gender" dropdown menu. A white, rounded rectangular button labeled "REGISTER" is centered below the form fields. Below the button, there is a link that says "Already have an account ? Sign In". At the bottom of the screen, a white toast message box displays the text: "Username vinu already exists. Please use a different username".

Figure 4.10 Validation for username existence

If none of the fields in the registration form is empty the user will be successfully registered to the application so that the user will be directed to the login page with a toast message saying that the user was successfully registered.



The image shows a login screen for the 'Learn Disease App'. The background is dark with a teal, abstract pattern. The text 'Log In' is prominently displayed in white. Below it, a welcome message and a prompt to login are shown. There are input fields for 'Username' and 'Password'. Two buttons, 'LOGIN' and 'SIGN UP', are located below the input fields. At the bottom, a toast message states 'You have successfully registered'.

Log In

Welcome !
Login to your Learn Disease App Account

Username

Password

LOGIN

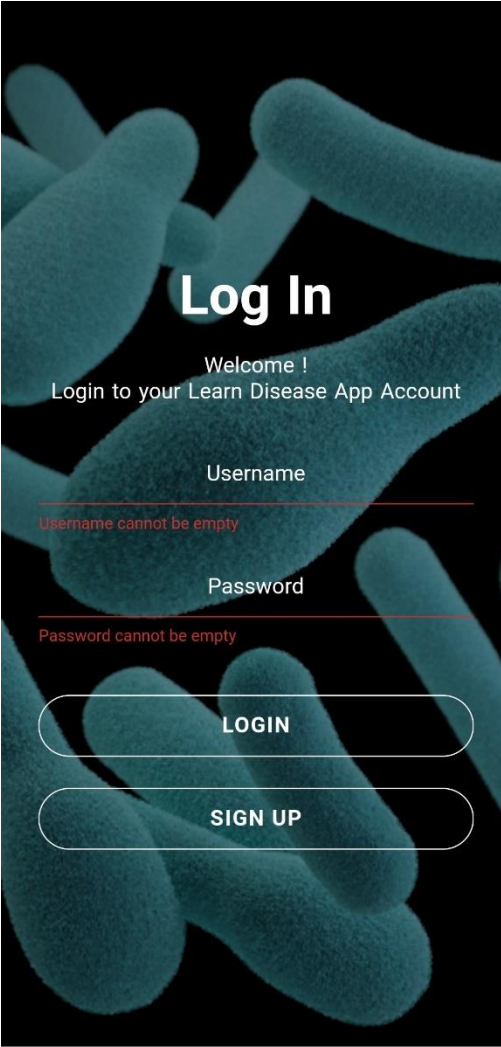
SIGN UP

You have successfully registered

Figure 4.11 Success message on user registration

Login page form validation

If the form fields are empty the application will validate form fields.



The image shows a mobile application login screen with a dark background and a teal, abstract, organic pattern. The text is white. At the top, it says "Log In" in a large font. Below that, it says "Welcome !" and "Login to your Learn Disease App Account". There are two input fields: "Username" and "Password". Both fields have red error messages below them: "Username cannot be empty" and "Password cannot be empty". At the bottom, there are two buttons: "LOGIN" and "SIGN UP".

Log In

Welcome !
Login to your Learn Disease App Account

Username

Username cannot be empty

Password

Password cannot be empty

LOGIN

SIGN UP

Figure 4.12 Login Page Field Validation

If user credentials are invalid a toast message will be generated and the user will have to provide the credentials again else the application will prompt a success toast message and will direct to the user home page.

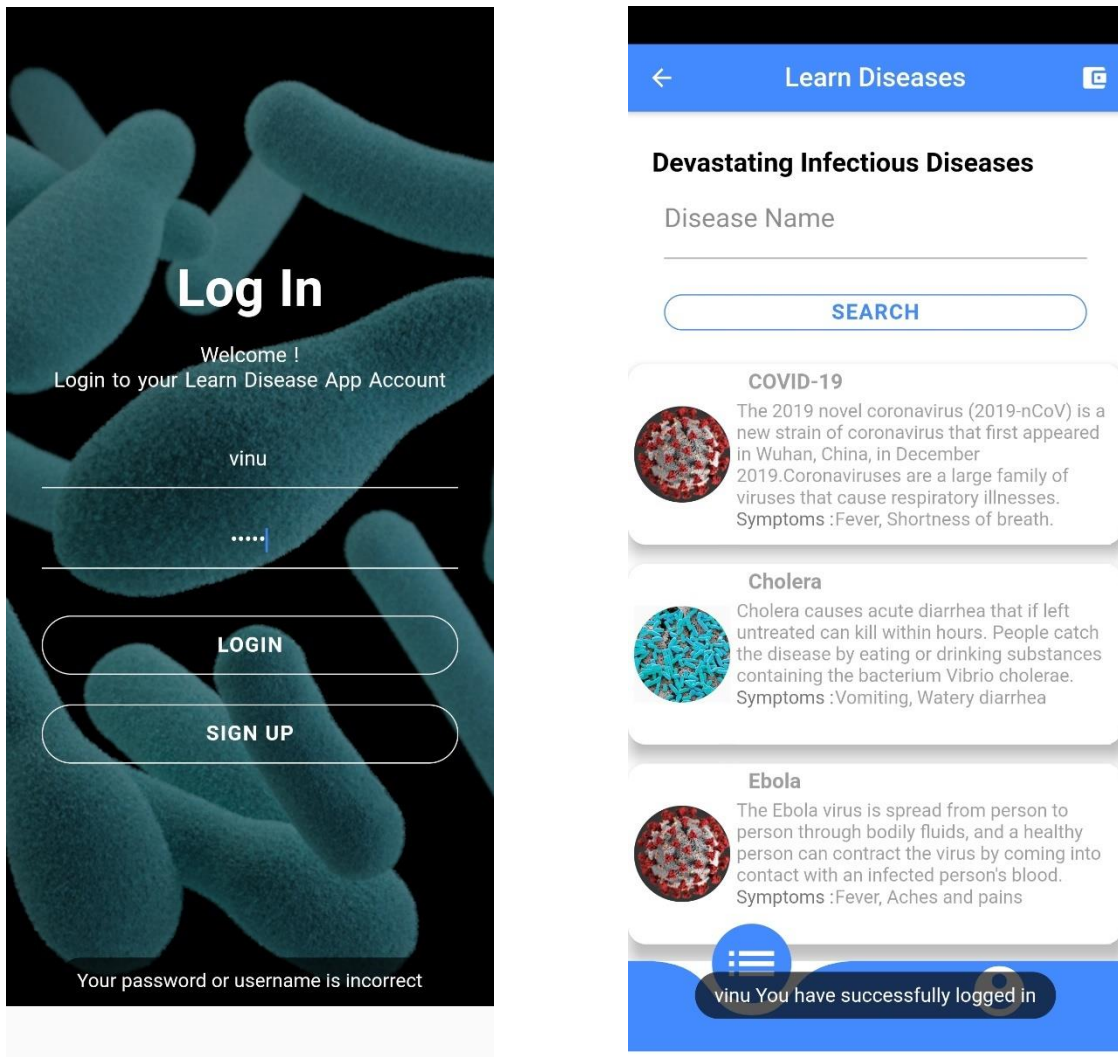


Figure 4.13 Page for invalid credentials and Success message on user login

UserDiseaseList page form validation

Inorder to search a disease using the disease list the user must provide the name as well as the name that is provided must be a valid disease name. If the name that is entered by the user is invalid a Toast message will be generated with a message saying to include a disease name from the list.

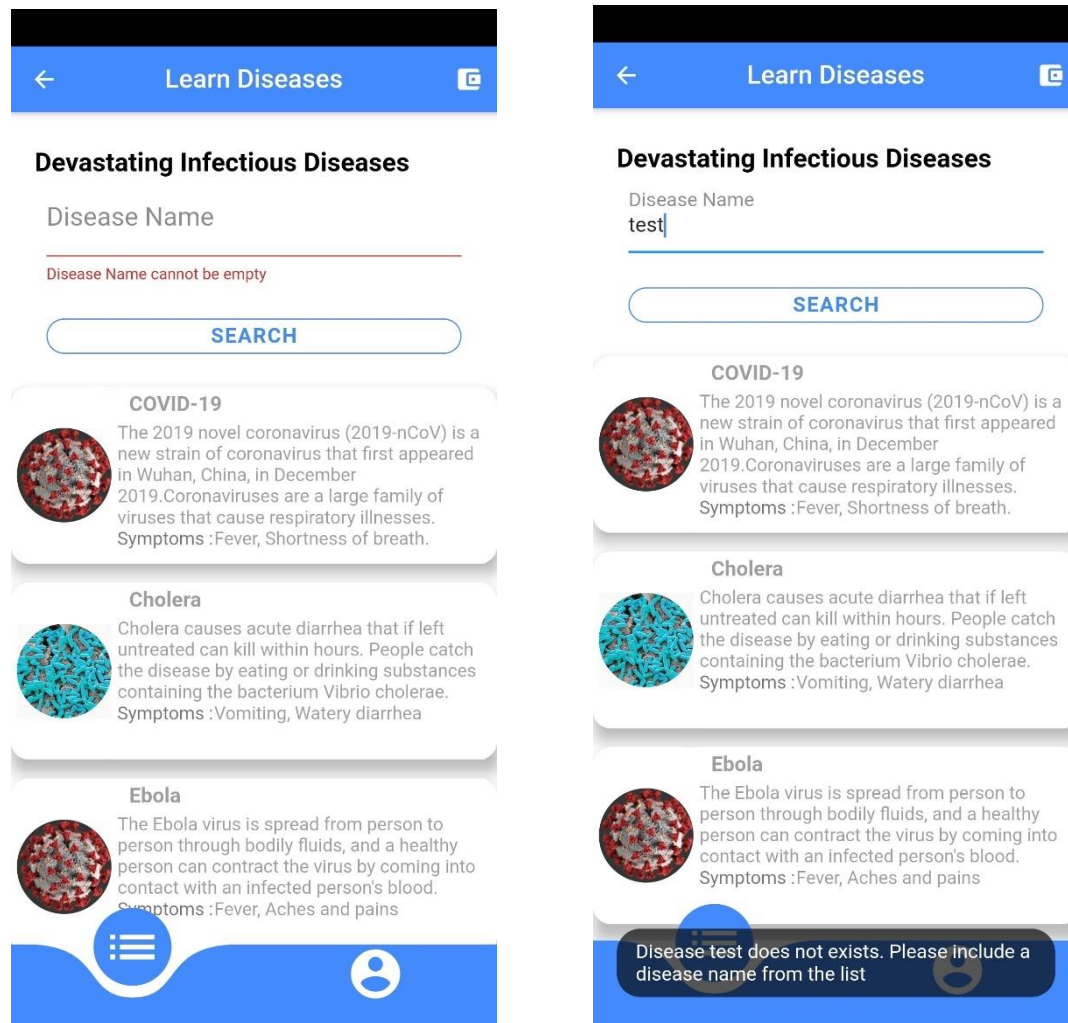


Figure 4.14 Field validation for disease name and Error message for invalid disease name

5 PROJECT FILE STRUCTURE

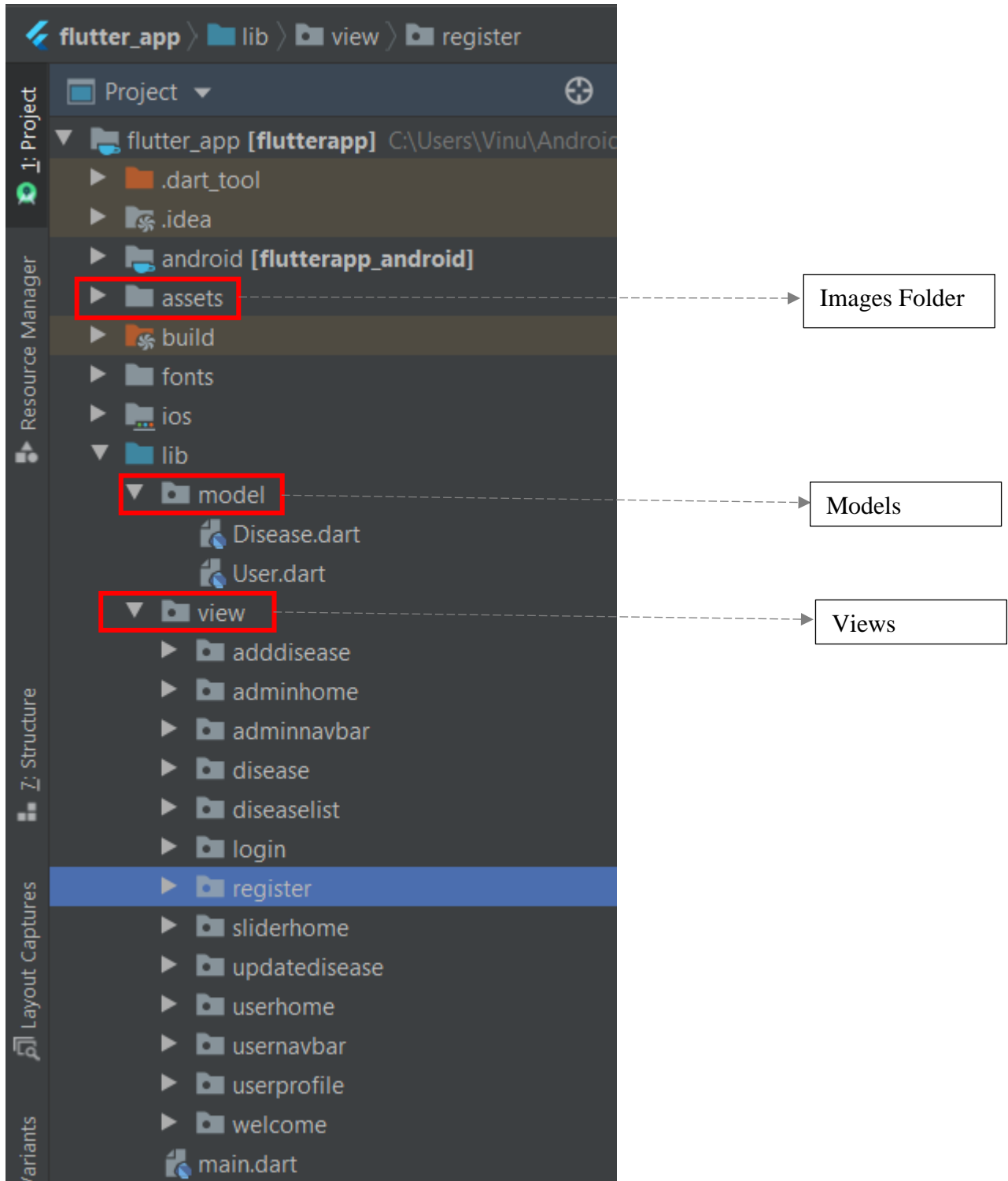


Figure 5.1 Project file Structure

6 REFERENCES

- [1] Flutter, "Hot reload," 2020. [Online]. Available: <https://flutter.dev/docs/development/tools/hot-reload>. [Accessed 5 April 2020].
- [2] IFlutter, "Flutter Intro Slider," 2020. [Online]. Available: <https://www.iflutter.in/flutter-intro-slider/>. [Accessed 5 April 2020].
- [3] Flutter, "Navigate to a new screen and back," 2020. [Online]. Available: <https://flutter.dev/docs/cookbook/navigation/navigation-basics>. [Accessed 01 April 2020].
- [4] Pub Developers, "curved_navigation_bar 0.3.2," 20 February 2020. [Online]. Available: https://pub.dev/packages/curved_navigation_bar. [Accessed 4 April 2020].
- [5] Stack Overflow, "Flutter: BottomNavigationBar rebuilds Page on change of tab," 2020. [Online]. Available: <https://stackoverflow.com/questions/52598900/flutter-bottomnavigationbar-rebuilds-page-on-change-of-tab>. [Accessed 01 April 2020].
- [6] Pub Developers, "Toast," 2020. [Online]. Available: <https://pub.dev/packages/toast>. [Accessed 4 April 2020].

7 APPENDIX

7.1 Models

User Model

```
/*
 *   Class Name : User class
 *   Author : IT17134668
 * */

import 'package:cloud_firestore/cloud_firestore.dart';

class User{

  final String firstname;
  final String lastname;
  final String username;
  final String password;
  final String sex;
  final DocumentReference reference;
  User.fromMap(Map<String, dynamic> map, {this.reference})
    : assert(map['firstname'] != null),
      assert(map['lastname'] != null),
      assert(map['username'] != null),
      assert(map['password'] != null),
      assert(map['sex'] != null),
      firstname = map['firstname'],
      lastname = map['lastname'],
      password = map['password'],
      sex = map['sex'],
      username = map['username'];

  User.fromSnapshot(DocumentSnapshot snapshot)
    : this.fromMap(snapshot.data, reference: snapshot.reference);
  @override
  String toString() => "Record<$password:$username>";
}
```

7.2 Views

SliderHome

```
/*
 * Widget Name : SliderHome stateful widget
 * Author : IT17134668
 * */

import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutterapp/view/welcome/Welcome.dart';
import 'package:intro_slider/slide_object.dart';
import 'package:intro_slider/intro_slider.dart';

class SliderHome extends StatefulWidget {
  @override
  _SliderHomeState createState() => _SliderHomeState();
}

class _SliderHomeState extends State<SliderHome> {
  //Create List of Slides
  List<Slide> slides = new List();

  //Create goToTab function
  Function goToTab;

  //Initialize Images
  @override
  void initState() {
    super.initState();

    slides.add(
      new Slide(
        title: "Register and Login",
        colorBegin: Color(0xff112c4a),
        colorEnd: Color(0xff112c4a),
        styleTitle: TextStyle(
          color: Colors.white,
          fontSize: 30.0,
          fontWeight: FontWeight.bold,
          fontFamily: 'RobotoMono'),
        pathImage: "assets/register.jpeg",
        foregroundImageFit: BoxFit.cover,
        description:
          "Create an account and register and login to the system",
        styleDescription: TextStyle(
          color: Colors.white,
          fontSize: 20.0,
          fontStyle: FontStyle.italic,
          fontFamily: 'Raleway'),
      ),
    );
    slides.add(
```

```

    new Slide(
      title: "Get Information About Diseases",
      colorBegin: Color(0xff112c4a),
      colorEnd: Color(0xff112c4a),
      styleTitle: TextStyle(
        color: Colors.white,
        fontSize: 30.0,
        fontWeight: FontWeight.bold,
        fontFamily: 'RobotoMono'),
      description:
        "Search on the disease to get to know more about it",
      styleDescription: TextStyle(
        color: Colors.white,
        fontSize: 20.0,
        fontStyle: FontStyle.italic,
        fontFamily: 'Raleway'),
      pathImage: "assets/diseases.jpeg",

    ),
  );
}

//Create Other functions
void onDonePress() {
  // Back to the first tab
  Navigator.push(
    context,
    MaterialPageRoute(builder: (context) => Welcome()),
  );
}

//renderNextBtn()
Widget renderNextBtn() {
  return Icon(
    Icons.navigate_next,
    color: Colors.blueAccent,
    size: 35.0,
  );
}

//renderDoneBtn()
Widget renderDoneBtn() {
  return Icon(
    Icons.done,
    size: 35,
    color: Colors.blueAccent,
  );
}

/*Widget renderSkipBtn() {
  return Icon(
    Icons.skip_next,
    color: Color(0xfffffcc5c),
  );
}

```

```

}*/

//Custom Tabs
List<Widget> renderListCustomTabs() {
  List<Widget> tabs = new List();
  for (int i = 0; i < slides.length; i++) {
    Slide currentSlide = slides[i];
    tabs.add(Container(
      width: double.infinity,
      height: double.infinity,
      child: Container(
        margin: EdgeInsets.only(bottom: 30.0, top: 30.0),
        child: ListView(
          children: <Widget>[
            Container(
              padding: EdgeInsets.only(top:80.0),
              child: Text(
                currentSlide.title,
                style: currentSlide.styleTitle,
                textAlign: TextAlign.center,
              ),
              margin: EdgeInsets.only(top: 20.0),
            ),
            Container(
              padding: EdgeInsets.only(top:40.0),
              child: GestureDetector(
                child: Image.asset(
                  currentSlide.pathImage,
                  width: 100.0,
                  height: 300.0,
                  fit: BoxFit.contain,
                ),
              ),
            ),
            Container(
              child: Text(
                currentSlide.description,
                style: currentSlide.styleDescription,
                textAlign: TextAlign.center,
                maxLines: 5,
                overflow: TextOverflow.ellipsis,
              ),
              margin: EdgeInsets.only(top: 20.0, left:50.0, right:50.0),
            ),
          ],
        ),
      ),
    );
  }
  return tabs;
}

@override
Widget build(BuildContext context) {
  return MaterialApp(
    debugShowCheckedModeBanner: false,

```

```

home: Scaffold(
  body: IntroSlider(
    // List slides
    slides: this.slides,

    /*// Skip button
    //renderSkipBtn: this.renderSkipBtn(),
    colorSkipBtn: Color(0x33ffcc5c),
    highlightColorSkipBtn: Color(0xffffcc5c),*/

    // Next button
    renderNextBtn: this.renderNextBtn(),

    // Done button
    renderDoneBtn: this.renderDoneBtn(),
    onDonePress: this.onDonePress,
    colorDoneBtn: Colors.white,
    highlightColorDoneBtn: Colors.white30,

    // Dot indicator
    colorDot: Colors.white,
    sizeDot: 13.0,

    // Tabs
    listCustomTabs: this.renderListCustomTabs(),
    backgroundColorAllSlides: Color(0xff112c4a),
    refFuncGoToTab: (refFunc) {
      this.goToTab = refFunc;
    },

    // Show or hide status bar
    shouldHideStatusBar: true,

  ),
);
}

```

Welcome

```
/*
 * Widget Name : Welcome stateful widget
 * Author : IT17134668
 * */

import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutterapp/view/login/Login.dart';
import 'package:flutterapp/view/register/Register.dart';

class Welcome extends StatefulWidget {
  @override
  _WelcomeState createState() => _WelcomeState();
}

class _WelcomeState extends State<Welcome> {
  @override
  Widget build(BuildContext context) {

    return
      Scaffold(

        body: Container(
          height: double.infinity,
          width: double.infinity,
          decoration: BoxDecoration(
            image: DecorationImage(
              //colorFilter: new
              ColorFilter.mode(Colors.lightBlue[200].withOpacity(0.1), BlendMode.colorBurn),
              colorFilter: new ColorFilter.mode(Colors.transparent.withOpacity(0.5),
              BlendMode.colorBurn),
              image: AssetImage('assets/background1.jpg'),
              fit: BoxFit.cover
            ),
          ),
        ),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Text(
              'Welcome',
              style: TextStyle(
                color: Colors.white,
                fontSize: 50.0,
                fontWeight: FontWeight.bold,
                fontFamily: 'Prompiere',
              ),
            ),
            SizedBox(height: 100,),
            Container(
              height: 1,
              width: 150,
              color: Colors.white,
            ),
          ],
        ),
      ),
  );
}
```



```

    ),
    SizedBox(height: 20,),
    Text(
      "Get to know about diseases",
      style: TextStyle(
        fontSize: 20,
        color: Colors.white,
        fontWeight: FontWeight.bold,
        fontFamily: 'Prompiere',
      ),
    ),
    ),
    SizedBox(height: 100,),
    Container(
      decoration: BoxDecoration(
        border: Border.all(
          width: 2,
          color: Colors.white,
        ),
        borderRadius: BorderRadius.all(Radius.circular(60))
      ),
      height: 50,
      width: 300,
      child: Center(
        child: RaisedButton(
          child: Text("Sign Up", style: TextStyle(fontSize: 20, color:
Colors.white, fontWeight: FontWeight.bold,fontFamily: 'Prompiere',)),
          color: Colors.transparent,
          onPressed: (){
            Navigator.push(
              context,
              MaterialPageRoute(builder: (context) => Register()),
            );
          },
        ),
      ),
    ),
    ),
    SizedBox(height: 15,),
    Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: <Widget>[
        Container(
          height: 1,
          width: 60,
          color: Colors.white,
        ),
        Text("OR SKIP", style: TextStyle(fontSize: 16, color:
Colors.white)),
        Container(
          height: 1,
          width: 60,
          color: Colors.white,
        ),
      ],
    ),
    ),
    SizedBox(height: 15,),

```

```

        Container(
          decoration: BoxDecoration(
            border: Border.all(
              width: 2,
              color: Colors.white,
            ),
            borderRadius: BorderRadius.all(Radius.circular(60))
          ),
          height: 50,
          width: 300,
          child: Center(
            child: RaisedButton(
              child: Text("Login", style: TextStyle(fontSize: 20, color:
Colors.white, fontWeight: FontWeight.bold, fontFamily: 'Prompiere',)),
              color: Colors.transparent,
              onPressed: (){
                Navigator.push(
                  context,
                  MaterialPageRoute(builder: (context) => Login()),
                );
              },
            ),
          ),
        ),
      ],
    ),
  );
}
}

```

Register

```
/*
 * Widget Name : Register stateful widget
 * Author : IT17134668
 * */

import 'package:flutter/material.dart';
import 'package:flutterapp/view/login/Login.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:toast/toast.dart';

class Register extends StatefulWidget {
  @override
  _RegisterState createState() => _RegisterState();
}

class _RegisterState extends State<Register> {
  String id;
  final db = Firestore.instance;
  final _formKey = GlobalKey<FormState>();
  String firstName;
  String lastName;
  String username;
  String password;
  String sex;
  @override
  Widget build(BuildContext context) {

    return Scaffold(
      body: Container(
        height: double.infinity,
        width: double.infinity,
        padding: const EdgeInsets.all(30.0),
        decoration: BoxDecoration(
          image: DecorationImage(
            colorFilter: new ColorFilter.mode(Colors.transparent.withOpacity(0.6),
BlendMode.hardLight),
            image: AssetImage('assets/background2.jpg'),
            fit: BoxFit.cover
          ),
        ),
      child: Form(
        key: _formKey,
        child: SingleChildScrollView(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            crossAxisAlignment: CrossAxisAlignment.center,
            textDirection: TextDirection.ltr,

            children: <Widget>[
              Container(
                padding: EdgeInsets.only(top: 150.0),
```

```

    ),
    Text(
      'Register',
      style: TextStyle(
        color: Colors.white,
        fontSize: 40.0,
        fontWeight: FontWeight.bold,
        wordSpacing: 1.5,
        fontFamily: 'gothic'
      ),
    ),
  ),
  SizedBox(height: 10,),
  TextFormField(
    textAlign: TextAlign.center,
    style: TextStyle(color: Colors.white),
    decoration: new InputDecoration(
      hintText: 'First Name',
      hintStyle: TextStyle(
        color: Colors.white,
      ),
      enabledBorder: UnderlineInputBorder(
        borderSide: BorderSide(color: Colors.white),
      ),
      focusedBorder: UnderlineInputBorder(
        borderSide: BorderSide(color: Colors.white),
      ),
    ),
  ),
  onSaved: (val) => firstName = val,
  validator: (val) {
    if (val.length == 0) {
      return "First name cannot be empty";
    } else {
      return null;
    }
  },
  keyboardType: TextInputType.text,
),
  SizedBox(height: 10,),
  TextFormField(
    textAlign: TextAlign.center,
    style: TextStyle(color: Colors.white),
    decoration: new InputDecoration(
      hintText: 'Last Name',
      hintStyle: TextStyle(
        color: Colors.white,
      ),
      enabledBorder: UnderlineInputBorder(
        borderSide: BorderSide(color: Colors.white),
      ),
      focusedBorder: UnderlineInputBorder(
        borderSide: BorderSide(color: Colors.white),
      ),
    ),
  ),
  onSaved: (val) => lastName = val,
  validator: (val) {

```

```

        if (val.length == 0) {
            return "Last name cannot be empty";
        } else {
            return null;
        }
    },
    keyboardType: TextInputType.text,
),
 SizedBox(height: 10,),
 TextFormField(
    textAlign: TextAlign.center,
    style: TextStyle(color: Colors.white),
    decoration: new InputDecoration(
        hintText: 'Username',
        hintStyle: TextStyle(
            color: Colors.white,
        ),
    ),
    // fillColor: Colors.white,
    enabledBorder: UnderlineInputBorder(
        borderSide: BorderSide(color: Colors.white),
    ),
    focusedBorder: UnderlineInputBorder(
        borderSide: BorderSide(color: Colors.white),
    ),
),
    onSave: (val) => username = val,
    validator: (val) {
        if (val.length == 0) {
            return "Username cannot be empty";
        } else {
            return null;
        }
    },
),
 SizedBox(height: 10,),
 TextFormField(
    textAlign: TextAlign.center,
    style: TextStyle(color: Colors.white),
    decoration: new InputDecoration(
        hintText: 'Password',
        hintStyle: TextStyle(
            color: Colors.white,
        ),
    ),
    // fillColor: Colors.white,
    enabledBorder: UnderlineInputBorder(
        borderSide: BorderSide(color: Colors.white),
    ),
    focusedBorder: UnderlineInputBorder(
        borderSide: BorderSide(color: Colors.white),
    ),
),
    obscureText: true,
    onSave: (val) => password = val,
    validator: (val) {
        if (val.length == 0) {

```

```

        return "Password cannot be empty";
    } else {
        return null;
    }
  },
),
 SizedBox(height: 10,),
 new DropdownButton<String>(
   focusColor: Colors.white,
   iconDisabledColor: Colors.white,
   iconEnabledColor: Colors.white,
   hint: Text(
     'Select gender',
     style: TextStyle(
       color: Colors.white,
       fontSize: 16
     ),
   ),
 ),
 items: <String>['Male', 'Female'].map((String value) {
   return new DropdownMenuItem<String>(
     value: value,
     child: new Text(value),
   );
 }).toList(),
 onChanged: (value) {
   setState(() {
     this.sex = value;
   });
 },
 value: this.sex,
 style: new TextStyle(
   color: Colors.grey[500],
 ),
 ),
 ),
 SizedBox(height: 20,),
 Container(
   height: 45.0,
   child: GestureDetector(
     onTap: () {},
     child: Container(
       decoration: BoxDecoration(
         border: Border.all(
           color: Colors.white,
           style: BorderStyle.solid,
           width: 1.0,
         ),
       ),
       color: Colors.transparent,
       borderRadius: BorderRadius.circular(30.0),
     ),
     child: Row(
       mainAxisAlignment: MainAxisAlignment.center,
       children: <Widget>[
         Center(
           child: InkWell(
             child: Text(

```

[illegible]

```

    );
}

//Registering user to the application
void createData() async {
  if (_formKey.currentState.validate()) {
    print('form key ::::::::::' + _formKey.currentContext.toString());
    _formKey.currentState.save();
    print('firstname :::::::::::' + firstName);
    var document = Firestore.instance.collection('users').document('$username');
    document.get().then((doc) {
      //no documents exists
      if(!doc.exists){
        db.collection("users").document("$username").setData({
          'firstname': '$firstName',
          'lastname': '$lastName',
          'username': '$username',
          'password': '$password',
          'sex': '$sex'
        }); //adding data
        Toast.show("$username You have successfully registered", context, duration:
Toast.LENGTH_LONG, gravity: Toast.BOTTOM);
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) => Login()),
        );
      }else{
        Toast.show("Username $username already exists. Please use a different
username", context, duration: Toast.LENGTH_LONG, gravity: Toast.BOTTOM);
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) => Register()),
        );
      }
    });
  }
}
}

```


Login

```
/*
 * Widget Name : Login stateful widget
 * Author : IT17134668
 * */

import 'package:flutter/material.dart';
import 'package:flutterapp/view/adminhome/AdminHome.dart';
import 'package:flutterapp/view/register/Register.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:toast/toast.dart';
import 'package:flutterapp/view/userhome/UserHome.dart';

class Login extends StatefulWidget {

  @override
  _LoginState createState() => _LoginState();
}

class _LoginState extends State<Login> {

  String username,password;
  final db = Firestore.instance;
  final GlobalKey<FormState> formKey = new GlobalKey();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        height: double.infinity,
        width: double.infinity,
        padding: const EdgeInsets.all( 30.0),
        decoration: BoxDecoration(
          image: DecorationImage(
            colorFilter: new ColorFilter.mode(Colors.transparent.withOpacity(0.5),
BlendMode.hardLight),
            image: AssetImage('assets/background2.jpg'),
            fit: BoxFit.cover
          ),
        ),
      ),
      child: Form(
        key: formKey,
        child: SingleChildScrollView(
          child: Column(
            //mainAxisAlignment: MainAxisAlignment.center,
            //crossAxisAlignment: CrossAxisAlignment.center,
            //textDirection: TextDirection.ltr,

            children: <Widget>[
```



```

    },
    //keyboardType: TextInputType.emailAddress,
  ),
  SizedBox(height: 20,),
  new TextFormField(
    textAlign: TextAlign.center,
    style: TextStyle( color: Colors.white),
    decoration: new InputDecoration(
      hintText: 'Password',
      hintStyle: TextStyle(
        color: Colors.white,
      ),
      enabledBorder: UnderlineInputBorder(
        borderSide: BorderSide( color: Colors.white),
      ),
      focusedBorder: UnderlineInputBorder(
        borderSide: BorderSide(color: Colors.white),
      ),
    ),
    onSave: (val) => password = val,
    validator: (val) {
      if(val.length==0) {
        return "Password cannot be empty";
      }else{
        return null;
      }
    },
  ),

  obscureText: true,
),
SizedBox(height: 40,),
Container(
  height: 50.0,
  child: GestureDetector(
    onTap: () {},
    child: Container(
      decoration: BoxDecoration(
        border: Border.all(
          color: Colors.white,
          style: BorderStyle.solid,
          width: 1.0,
        ),
        color: Colors.transparent,
        borderRadius: BorderRadius.circular(30.0),
      ),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          Center(
            child: InkWell(
              child: Text(
                "LOGIN",
                style: TextStyle(

```



```

    },
    ),
    ),
    ],
    ),
    ),
    ),
    ),
    );
}

// validating user login
void validateLogin() {
    String pass = "";

    if (formKey.currentState.validate()) {
        print('form key ::::::' + formKey.currentContext.toString());
        formKey.currentState.save();
        print('USERNAME ***** ' + username);
        print('PASSWORD ***** ' + password);
        var document = Firestore.instance.collection('users').document('$username');
        document.get().then((doc) {
            //if there are documents
            if (doc.exists){
                //username is not admin
                if(!(username=='admin')){
                    String pass = doc.data.values.elementAt(1);
                    print('PASSWORD FROM DATABASE ***** ' + pass);
                    if(pass==password){
                        //where passwords are matching
                        Toast.show("$username You have successfully logged in", context,
duration: Toast.LENGTH_LONG, gravity: Toast.BOTTOM);
                        Navigator.push(
                            context,
                            MaterialPageRoute(builder: (context) => UserHome(username)),
                        );
                        //create toast and login
                    }else {
                        //error message username password incorrect
                        Toast.show("Your password or username is incorrect", context,
duration: Toast.LENGTH_LONG, gravity: Toast.BOTTOM);
                    }
                }
            }else{
                //where username is admin
                String pass = doc.data.values.elementAt(3);
                //success message
                if(pass==password){
                    Toast.show("$username You have successfully logged in", context,
duration: Toast.LENGTH_LONG, gravity: Toast.BOTTOM);
                    Navigator.push(
                        context,
                        MaterialPageRoute(builder: (context) => AdminHome()),
                    );
                }
            }
        });
    }
}

```



```

        Icons.account_balance_wallet,
        color: Colors.white,
      ),
      onPressed: (){
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) => Login()),
        );
      },
    ],
  ),
),
body: UserNavBar(this.username),
);
}
}

```

UserNavBar

```

/*
 * Widget Name : UserNavBar stateful widget
 * Author : IT17134668
 * */

import 'package:curved_navigation_bar/curved_navigation_bar.dart';
import 'package:flutter/material.dart';
import 'package:flutterapp/view/diseaselist/UserDiseaseList.dart';
import 'package:flutterapp/view/userprofile/UserProfile.dart';

class UserNavBar extends StatefulWidget {
  String username;
  UserNavBar(this.username);
  @override
  _UserNavBarState createState() => _UserNavBarState(this.username);
}

class _UserNavBarState extends State<UserNavBar> {
  PageController _pageController;
  String username;
  _UserNavBarState(this.username);
  @override
  void initState() {
    super.initState();
    _pageController = PageController();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: PageView(
        controller: _pageController,
        children: <Widget>[

```

```

        UserDiseaseList(this.username),
        UserProfile(this.username),
      ],
      onPageChanged: (int index) {
        setState(() {
          _pageController.jumpToPage(index);
        });
      }
    ),
    bottomNavigationBar: CurvedNavigationBar(
      animationCurve: Curves.easeInOutBack,
      index: 0,
      items: <Widget>[
        Icon(Icons.list, size: 50, color: Colors.white),
        Icon(Icons.account_circle, size: 50, color: Colors.white),
      ],
      color: Colors.blueAccent,
      backgroundColor: Colors.white,
      //buttonBackgroundColor: Colors.white,
      height: 50.0,
      onTap: (index) {
        print("INDEX##### " + index.toString());
        //Handle button tap
        setState(() {
          _pageController.jumpToPage(index);
        });
      },
    ),
  );
}
}

```

UserDiseaseList

```

/*
 * Widget Name : UserDiseaseList stateful widget
 * Author : IT17134668
 * */

import 'package:flutter/material.dart';
import 'package:flutterapp/model/Disease.dart';
import 'package:flutterapp/view/disease/UserDiseaseInfo.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:toast/toast.dart';

class UserDiseaseList extends StatefulWidget {
  String username;
  UserDiseaseList(this.username);
  @override

```



```

    _UserDiseaseListState createState() => _UserDiseaseListState(this.username);
}

class _UserDiseaseListState extends State<UserDiseaseList> {
  String name;
  String username;
  _UserDiseaseListState(this.username);
  final GlobalKey<FormState> formKey = new GlobalKey();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: SingleChildScrollView(
        child: Column(
          //mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Container(
              padding: EdgeInsets.only(left: 20, top: 30),
              width: double.infinity,
              child: Text("Devastating Infectious Diseases", style: TextStyle(color:
Colors.black, fontSize: 20, fontWeight: FontWeight.bold)),
            ),
            Container(
              padding: EdgeInsets.only(left: 30, right: 30),
              child: Column(
                children: <Widget>[
                  Form(
                    key: formKey,
                    child: TextFormField(
                      decoration: InputDecoration(
                        labelText: "Disease Name ",
                        labelStyle: TextStyle(
                          fontSize: 20.0,
                          color: Colors.black45
                        ),
                      ),
                    ),
                    onSave: (val) => name = val,
                    validator: (val) {
                      if (val.length == 0) {
                        return "Disease Name cannot be empty";
                      } else {
                        return null;
                      }
                    }
                  ),
                ],
              ),
            ),
            SizedBox(height: 30.0,),
            Container(
              /*child: IconButton(
                icon: Icon(
                  Icons.search,
                  color: Colors.black,
                ),
                onPressed: directToDiseaseInfo,

```

```

    )*/
    // padding: EdgeInsets.only(left: 40.0,right: 40.0),
    height: 30.0,
    decoration: BoxDecoration(
      border: Border.all(
        color: Colors.blueAccent,
        style: BorderStyle.solid,
        width: 1.0,
      ),
      color: Colors.transparent,
      borderRadius: BorderRadius.circular(30.0),
    ),
    child: Center(
      child: InkWell(
        child: Text(
          "SEARCH",
          style: TextStyle(
            color: Colors.blueAccent,
            fontFamily: 'Montserrat',
            fontSize: 16,
            fontWeight: FontWeight.bold,
            letterSpacing: 1,
          ),
        ),
      ),
      onTap: directToDiseaseInfo,
    ),
  ),
),
],
),
),
);

Container(
  height: MediaQuery.of(context).size.height-330,
  child: _buildBody(context),
),

],
),
),
);
}

//navigating to disease information page
void directToDiseaseInfo(){
  if (formKey.currentState.validate()) {
    formKey.currentState.save();
    var document = Firestore.instance.collection('diseases').document('$name');
    document.get().then((doc) {
      //If document exists
      if (doc.exists) {
        print('Documents exists');
        Navigator.push(
          context,

```

```

        MaterialPageRoute(builder: (context) => UserDiseaseInfo(name,username)),
    );
  } else {
    //where no data exists
    Toast.show("Disease $name does not exists. Please include a disease name
from the list", context, duration: Toast.LENGTH_LONG, gravity: Toast.BOTTOM);
    print('ERORRRRRRRRRRRRRRRRR');
  }
});
}
}
}

Widget _buildBody(BuildContext context) {
  //retrieving data from diseases collection
  return StreamBuilder<QuerySnapshot>(
    stream: Firestore.instance.collection('diseases').snapshots(),
    builder: (context, snapshot){
      if(!snapshot.hasData) return LinearProgressIndicator();
      return _buildList(context, snapshot.data.documents);
    },
  );
}

Widget _buildList(BuildContext context, List<DocumentSnapshot> snapshot) {
  return ListView(
    padding: const EdgeInsets.only(top: 20.0),
    children: snapshot.map((data) => _buildListItem(context, data)).toList(),
  );
}

Widget _buildListItem(BuildContext context, DocumentSnapshot data) {
  final record = Disease.fromSnapshot(data);

  return Container(
    padding: const EdgeInsets.symmetric(horizontal: 0.0, vertical: 8.0),
    child: Material(
      key: ValueKey(record.name),
      color: Colors.white,
      elevation: 10,
      borderRadius: BorderRadius.circular(15),
      child: Container(
        height: 150,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(15),
          //color: Colors.red
        ),
      ),
      child: Row(
        children: <Widget>[
          Container(
            width: 90,
            decoration: BoxDecoration(
              borderRadius: BorderRadius.circular(15),
              color: Colors.white,

```



```

        style: TextStyle(color: Colors.black54, fontSize: 14,
        ),),
        Text(record.symptoms,
        style: TextStyle(color: Colors.black38, fontSize: 14,
        ),),
      ],
    ),
  ],
),
),
],
),
),
),
);
}

```

UserDiseaseInfo

```

/*
 * Widget Name : UserDiseaseInfo StatefulWidget
 * Author : IT17134668
 * */

import 'package:flutter/material.dart';
import 'package:flutterapp/model/Disease.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutterapp/view/userhome/UserHome.dart';

class UserDiseaseInfo extends StatefulWidget {
  String name;
  String username;
  UserDiseaseInfo(this.name,this.username);
  @override
  _UserDiseaseInfoState createState() =>
  _UserDiseaseInfoState(this.name,this.username);
}

class _UserDiseaseInfoState extends State<UserDiseaseInfo> {
  String name;
  String username;
  _UserDiseaseInfoState(this.name,this.username);
  @override
  Widget build(BuildContext context) {
    return Scaffold(

```

```

appBar: AppBar(
  backgroundColor: Colors.blueAccent,
  centerTitle: true,
  title: Text("Learn Diseases", style: TextStyle(color: Colors.white)),
),
body: Container(
  child: Column(
    children: <Widget>[
      _buildBody(context, this.name),
      SizedBox(height: 25,),
      Container(
        padding: EdgeInsets.only(left: 70.0, top: 20.0, right: 70.0),
        height: 50.0,
        child: GestureDetector(
          onTap: () {},
          child: Container(
            decoration: BoxDecoration(
              border: Border.all(
                color: Colors.blueAccent,
                style: BorderStyle.solid,
                width: 1.0,
              ),
              color: Colors.transparent,
              borderRadius: BorderRadius.circular(30.0),
            ),
            child: Row(
              mainAxisAlignment: MainAxisAlignment.center,
              children: <Widget>[
                Center(
                  child: InkWell(
                    child: Text(
                      "HOME",
                      style: TextStyle(
                        color: Colors.blueAccent,
                        fontFamily: 'Montserrat',
                        fontSize: 16,
                        fontWeight: FontWeight.w600,
                        letterSpacing: 1,
                      ),
                    ),
                    onTap: (){
                      Navigator.push(
                        context,
                        MaterialPageRoute(builder: (context) =>
UserHome(this.username)),
                      );
                    },
                  ),
                ),
              ],
            ),
          ),
        ),
      ],
    ),
  ),
)
]

```

```

    )
  )
);
}

}

Widget _buildBody(BuildContext context,name) {
  //retrieving data from diseases collection(as snapshots)
  return StreamBuilder<DocumentSnapshot>(
    stream: Firestore.instance.collection('diseases').document(name).snapshots(),
    builder: (context,snapshot){
      if(!snapshot.hasData) return LinearProgressIndicator();
      return _buildListItem(context, snapshot.data);
    },
  );
}

Widget _buildListItem(BuildContext context, DocumentSnapshot data) {
  final record = Disease.fromSnapshot(data);

  return Container(
    // padding: const EdgeInsets.symmetric(horizontal: 0.0, vertical: 8.0),
    padding: EdgeInsets.only(top: 70.0,left: 20.0,right: 20.0),
    child: Material(
      key: ValueKey(record.name),
      color: Colors.white,
      elevation: 10,
      borderRadius: BorderRadius.circular(15),
      child: Container(
        height: 600,
        width: 500,
        padding: EdgeInsets.only(top: 40.0),
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(15),
        ),
        child: Column(
          children: <Widget>[
            Container(
              width: 90,
              decoration: BoxDecoration(
                borderRadius: BorderRadius.circular(15),
                color: Colors.white,
              ),
            ),
            child: Center(
              child: record.causedby=='virus' ? Container(
                padding: EdgeInsets.only(left: 30.0),
                height:80,
                width: 80,
                decoration: BoxDecoration(
                  color: Colors.white,
                  borderRadius: BorderRadius.circular(100),
                  image: DecorationImage(
                    image: AssetImage('assets/virus.jpg'),
                    fit: BoxFit.cover,

```



```

        ),),
    ],
),
Text(record.symptoms,
  style: TextStyle(color: Colors.black38, fontSize: 16,
  ),),
Row(
  children: <Widget>[
    SizedBox(width: 10,height: 60,),
    Text('Caused By :',
      style: TextStyle(color: Colors.black54, fontSize: 20,
      ),),
    Text(record.causedby,
      style: TextStyle(color: Colors.black38, fontSize: 16,
      ),),
  ],
),
],
),
),
],
),
),
);
}

```

UserProfile

```

/*
 * Widget Name : UserProfile stateful widget
 * Author : IT17134668
 * */

import 'package:flutter/material.dart';
import 'package:flutterapp/model/User.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutterapp/view/register/Register.dart';
import 'package:flutterapp/view/userprofile/UpdateProfile.dart';

```

```

class UserProfile extends StatefulWidget {
  String username;
  UserProfile(this.username);
  @override
  _UserProfileState createState() => _UserProfileState(this.username);
}

class _UserProfileState extends State<UserProfile> {
  String username;
  _UserProfileState(this.username);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: _buildBody(context, this.username),
    );
  }
}

Widget _buildBody(BuildContext context, username) {
  //retrieving data from users collection(snapshot)
  return StreamBuilder<DocumentSnapshot>(
    stream: Firestore.instance.collection('users').document(username).snapshots(),
    builder: (context, snapshot){
      if(!snapshot.hasData) return LinearProgressIndicator();
      return _buildListItem(context, snapshot.data);
    },
  );
}

Widget _buildListItem(BuildContext context, DocumentSnapshot data) {
  final record = User.fromSnapshot(data);

  return Container(
    padding: const EdgeInsets.only(left: 20.0, right: 20.0, top: 40.0),

    child: Material(
      key: ValueKey(record.username),
      color: Colors.white,
      elevation: 10,
      borderRadius: BorderRadius.circular(30),
      child: Container(
        height: 600,
        width: 500,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(30),
          //color: Colors.red
        ),
        child: SingleChildScrollView(
          child: Column(
            children: <Widget>[
              Row(
                children: <Widget>[
                  Container(
                    padding: EdgeInsets.only(left: 140.0, top: 30.0),

```

```

        child: Center(
          child: record.sex=='Female' ? Container(
            padding: EdgeInsets.only(left: 30.0),
            height:100,
            width: 100,
            decoration: BoxDecoration(
              color: Colors.white,
              borderRadius: BorderRadius.circular(100),
              image: DecorationImage(
                image: AssetImage('assets/girl.jpg'),
                fit: BoxFit.cover,
              ),
            ),
          ) : Container(
            padding: EdgeInsets.only(left: 30.0),
            height:100,
            width: 100,
            decoration: BoxDecoration(
              color: Colors.white,
              borderRadius: BorderRadius.circular(100),
              image: DecorationImage(
                image: AssetImage('assets/boy.jpg'),
                fit: BoxFit.cover,
              ),
            ),
          ),
        ),
      ),
    ],
  ),
  Container(
    padding: EdgeInsets.only(left: 80.0,top: 20.0),
    child: Row(
      children: <Widget>[
        Text(
          'Welcome ',
          style: TextStyle(color: Colors.black54, fontSize: 30),
          textAlign: TextAlign.center,
        ),
        Text(record.username, style: TextStyle(color: Colors.black54,
fontSize: 30, )),
      ],
    ),
  ),
  Container(
    color: Colors.white,
    width: MediaQuery.of(context).size.width - 100,
    child: Column(
      children: <Widget>[
        Container(
          padding: EdgeInsets.only(left: 70.0,top: 30.0),

```

```

        child: Row(
          children: <Widget>[
            Text("First Name :", style: TextStyle(color:
Colors.black54, fontSize: 20,)),
            Text(record.firstname, style: TextStyle(color:
Colors.black38, fontSize: 20,)),
            SizedBox(width: 30,),
          ],
        ),
      ),
    Container(
      padding: EdgeInsets.only(left: 70.0,top: 10.0),
      child: Row(
        children: <Widget>[
          Text("Last Name :", style: TextStyle(color: Colors.black54,
fontSize: 20,)),
          Text(record.lastname, style: TextStyle(color:
Colors.black38, fontSize: 20, )),
          SizedBox(width: 30,),
        ],
      ),
    ),
    Container(
      padding: EdgeInsets.only(left: 70.0,top: 10.0),
      child: Row(
        children: <Widget>[
          Text("Username :", style: TextStyle(color: Colors.black54,
fontSize: 20,)),
          Text(record.username, style: TextStyle(color:
Colors.black38, fontSize: 20, )),
          SizedBox(width: 30,),
        ],
      ),
    ),
    Container(
      padding: EdgeInsets.only(left: 70.0,top: 10.0),
      child: Row(
        children: <Widget>[
          Text("Password :", style: TextStyle(color: Colors.black54,
fontSize: 20,)),
          Text(record.password, style: TextStyle(color:
Colors.black38, fontSize: 20, )),
          SizedBox(width: 30,),
        ],
      ),
    ),
    Container(
      padding: EdgeInsets.only(left: 70.0,top: 10.0),
      child: Row(
        children: <Widget>[
          Text("Gender      :", style: TextStyle(color:
Colors.black54, fontSize: 20,)),
          Text(record.sex, style: TextStyle(color: Colors.black38,
fontSize: 20,)),
          SizedBox(width: 30,),

```



```

    ),
    Container(
      padding: EdgeInsets.only(left: 70.0, top: 20.0, right: 70.0),
      height: 50.0,
      child: GestureDetector(
        onTap: () {},
        child: Container(
          decoration: BoxDecoration(
            border: Border.all(
              color: Colors.blueAccent,
              style: BorderStyle.solid,
              width: 1.0,
            ),
            color: Colors.transparent,
            borderRadius: BorderRadius.circular(30.0),
          ),
          child: Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
              Center(
                child: InkWell(
                  child: Text(
                    "DELETE PROFILE",
                    style: TextStyle(
                      color: Colors.blueAccent,
                      fontFamily: 'Montserrat',
                      fontSize: 16,
                      fontWeight: FontWeight.w600,
                      letterSpacing: 1,
                    ),
                  ),
                ),
              ),
            ],
            onTap: (){
Firestore.instance.collection('users').document(record.username).delete();
              Navigator.push(
                context,
                MaterialPageRoute(builder: (context) => Register()),
              );
            },
          ),
        ],
      ),
    ),
  ],
),
);

```

```
}
```

UpdateProfile

```
/*
 * Widget Name : UpdateProfile stateful widget
 * Author : IT17134668
 * */

import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutterapp/view/userhome/UserHome.dart';
import 'package:toast/toast.dart';

class UpdateProfile extends StatefulWidget {

  String username;
  String firstname;
  String password;
  String lastname;
  String sex;
  UpdateProfile(this.firstname,this.lastname,this.password,this.username,this.sex);
  @override
  _UpdateProfileState createState() =>
  _UpdateProfileState(this.firstname,this.lastname,this.password,this.username,this.sex
);
}

class _UpdateProfileState extends State<UpdateProfile> {
  //final record = User.fromSnapshot(data);
  String username;
  String firstname;
  String password;
  String lastname;
  String sex;

  _UpdateProfileState(this.firstname,this.lastname,this.password,this.username,this.sex
);
  final _formKey = GlobalKey<FormState>();
  TextEditingController
controller,lastNameController,usernameController,passwordController,sexController;

  @override
  void initState() {
    super.initState();
    controller = TextEditingController(text: this.firstname);
    lastNameController = TextEditingController(text: this.lastname);
    usernameController = TextEditingController(text: this.username);
    passwordController = TextEditingController(text: this.password);
    sexController = TextEditingController(text: this.sex);
  }
}
```

```

@override
Widget build(BuildContext context) {
  //Navigator.pop(context,true);
  return Container(
    //padding: EdgeInsets.only(top: 1.0,bottom: 50.0),
    child: AlertDialog(
      content: Form(
        key: _formKey,
        child: SingleChildScrollView(
          child: Column(
            children: <Widget>[
              Container(
                padding: EdgeInsets.only(top: 20.0),
                child: this.sex=='Female' ? Container(
                  padding: EdgeInsets.only(left: 30.0),
                  height:100,
                  width: 100,
                  decoration: BoxDecoration(
                    color: Colors.white,
                    borderRadius: BorderRadius.circular(100),
                    image: DecorationImage(
                      image: AssetImage('assets/girl.jpg'),
                      fit: BoxFit.cover,
                    ),
                  ),
                ),
              ) : Container(
                padding: EdgeInsets.only(left: 30.0),
                height:100,
                width: 100,
                decoration: BoxDecoration(
                  color: Colors.white,
                  borderRadius: BorderRadius.circular(100),
                  image: DecorationImage(
                    image: AssetImage('assets/boy.jpg'),
                    fit: BoxFit.cover,
                  ),
                ),
              ),
            ),
          ),
          SizedBox(height:10.0),
          Text(
            'Update Profile',
            style: TextStyle(
              fontSize: 30.0,
            ),
          ),
          SizedBox(height:10.0),
          TextFormField(
            decoration: InputDecoration(
              labelText: "Username ",
              labelStyle: TextStyle(

```



```

        fontSize: 20.0
    ),
),
controller: usernameController,
onSaved: (val){
    username = val;
},
readOnly: true,
),
 SizedBox(height: 5.0,),
 TextFormField(
    decoration: InputDecoration(
        labelText: "First Name ",
        labelStyle: TextStyle(
            fontSize: 20.0
        ),
    ),
),
    controller: controller,
    onSaved: (val){
        firstname = val;
    },
),
 SizedBox(height: 10.0,),
 TextFormField(
    decoration: InputDecoration(
        labelText: "Last Name ",
        labelStyle: TextStyle(
            fontSize: 20.0
        ),
    ),
),
    controller: lastNameController,
    onSaved: (val){
        lastname = val;
    },
),
 SizedBox(height: 5.0,),
 TextFormField(
    decoration: InputDecoration(
        labelText: "Password ",
        labelStyle: TextStyle(
            fontSize: 20.0
        ),
    ),
),
    controller: passwordController,
    obscureText: true,
    onSaved: (val){
        password = val;
    },
),
 SizedBox(height: 5.0,),
 new DropdownButton<String>(
    focusColor: Colors.black26,
    iconDisabledColor: Colors.black26,
    iconEnabledColor: Colors.black26,
    hint: Text(

```

```

        'Select gender',
        style: TextStyle(
          color: Colors.white
        ),
      ),
    ),
    items: <String>['Male', 'Female'].map((String value) {
      return new DropdownMenuItem<String>(
        value: value,
        child: new Text(value),
      );
    }).toList(),
    onChanged: (value) {
      setState(() {
        this.sex = value;
      });
    },
    value: this.sex,
    style: new TextStyle(
      color: Colors.black,
    ),
  ),
  SizedBox(height: 15.0,),
  Container(
    padding: EdgeInsets.only(left: 30.0,top: 20.0,right: 30.0),
    height: 50.0,
    child: GestureDetector(
      onTap: () {},
      child: Container(
        decoration: BoxDecoration(
          border: Border.all(
            color: Colors.blueAccent,
            style: BorderStyle.solid,
            width: 1.0,
          ),
          color: Colors.transparent,
          borderRadius: BorderRadius.circular(30.0),
        ),
        child: Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Center(
              child: InkWell(
                child: Text(
                  "UPDATE PROFILE",
                  style: TextStyle(
                    color: Colors.blueAccent,
                    fontFamily: 'Montserrat',
                    fontSize: 16,
                    fontWeight: FontWeight.w600,
                    letterSpacing: 1,
                  ),
                ),
              ),
            ),
          ],
          onTap: updateProfile,
        ),
      ),
    ),
  ),
),

```

```

    )
    ],
    ),
    ),
    ),
    ],
    ),
    ),
    ),
    );
}

//updating user profile
void updateProfile() async{
  if (_formKey.currentState.validate()) {
    _formKey.currentState.save();

    Firestore.instance.collection("users").document("$username").updateData({
      'firstname': '$firstname',
      'lastname': '$lastname',
      'username': '$username',
      'password': '$password',
      'sex': '$sex'
    }); //updating data
    Toast.show("$username You have successfully updated your profile", context,
duration: Toast.LENGTH_LONG, gravity: Toast.BOTTOM);
    Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => UserHome(username)),
    );
  }
}
}

```

Main

```
/*
 * main dart file
 * */

import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutterapp/view/sliderhome/SliderHome.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: SliderHome(),
    );
  }
}
```

Pubspec.yml file

```
name: flutterapp
description: A new Flutter application.

# The following defines the version and build number for your application.
# A version number is three numbers separated by dots, like 1.2.43
# followed by an optional build number separated by a +.
# Both the version and the build number may be overridden in flutter
# build by specifying --build-name and --build-number, respectively.
# In Android, build-name is used as versionName while build-number used as
versionCode.
# Read more about Android versioning at
https://developer.android.com/studio/publish/versioning
# In iOS, build-name is used as CFBundleShortVersionString while build-number used as
CFBundleVersion.
# Read more about iOS versioning at
#
https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html
version: 1.0.0+1

environment:
  sdk: ">=2.1.0 <3.0.0"
```

```

dependencies:
  flutter:
    sdk: flutter
  cloud_firestore: ^0.13.0+1
  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^0.1.2
#  firebase_auth: 0.15.5+2
  toast: ^0.1.5
  curved_navigation_bar: ^0.3.2
  intro_slider: ^2.2.9
  shared_preferences: ^0.5.6+2

dev_dependencies:
  flutter_test:
    sdk: flutter

  flutter_launcher_icons: ^0.7.0

flutter_icons:
  image_path: "assets/icon.JPG"
  ios: true
  android: true

# For information on the generic Dart part of this file, see the
# following page: https://dart.dev/tools/pub/pubspec

# The following section is specific to Flutter.
flutter:

  # The following line ensures that the Material Icons font is
  # included with your application, so that you can use the icons in
  # the material Icons class.
  uses-material-design: true

  # To add assets to your application, add an assets section, like this:
  assets:
    - assets/

  # An image asset can refer to one or more resolution-specific "variants", see
  # https://flutter.dev/assets-and-images/#resolution-aware.

  # For details regarding adding assets from package dependencies, see
  # https://flutter.dev/assets-and-images/#from-packages

  # To add custom fonts to your application, add a fonts section here,
  # in this "flutter" section. Each entry in this list should have a
  # "family" key with the font family name, and a "fonts" key with a
  # list giving the asset and other descriptors for the font. For
  # example:
  fonts:
    - family: gothic

```

```
fonts:
  - asset: fonts/GOTHIC.TTF

# For details regarding fonts from package dependencies,
# see https://flutter.dev/custom-fonts/#from-packages
```

AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.example.flutterapp">
  <!-- io.flutter.app.FlutterApplication is an android.app.Application that
       calls FlutterMain.startInitialization(this); in its onCreate method.
       In most cases you can leave this as-is, but you if you want to provide
       additional functionality it is fine to subclass or reimplement
       FlutterApplication and put your custom class here. -->
  <application
    android:name="io.flutter.app.FlutterApplication"
    android:label="Learn Disease"
    android:icon="@mipmap/ic_launcher">
    <activity
      android:name=".MainActivity"
      android:launchMode="singleTop"
      android:theme="@style/LaunchTheme"

      android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreenS
      ize|locale|layoutDirection|fontScale|screenLayout|density|uiMode"
      android:hardwareAccelerated="true"
      android:windowSoftInputMode="adjustResize">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
    <!-- Don't delete the meta-data below.
         This is used by the Flutter tool to generate
         GeneratedPluginRegistrant.java -->
    <meta-data
      android:name="flutterEmbedding"
      android:value="2" />
  </application>
</manifest>
```

Other resources –

- Make fullscreen image in flutter - <https://medium.com/@petehouston/make-fullscreen-image-in-flutter-cde0461c61ee>
- Introduction to textfield in flutter - <https://medium.com/flutter-community/a-deep-dive-into-flutter-textfields-f0e676aaab7a>
- Buttons in flutter - <https://medium.com/@16shuklahul/create-a-rounded-button-button-with-border-radius-in-flutter-b0cfb131e43f>
- Local storage - <https://medium.com/@lawrey/flutter-local-storage-ac2eaf1194be>
- Passing data between screens - <https://stackoverflow.com/questions/53861302/passing-data-between-screens-in-flutter>
- Add data to cloud store - <https://firebase.google.com/docs/firestore/manage-data/add-data>
- Get data from cloud store - <https://firebase.google.com/docs/firestore/query-data/get-data>
- Pop up dialog box on pressing button - <https://stackoverflow.com/questions/56806136/pop-up-dialog-box-on-pressing-button-in-flutter>
- How to store value in textfield when editing - <https://dev.to/mightytechno/flutter-text-field-important-things-you-need-to-know-3980>
- Deeping Firestore queries with Flutter - <https://medium.com/@kfarsoft/deeping-firestore-queries-with-flutter-2210fd3b49e1>
- Read writes in flutter - <https://www.javatpoint.com/firebase-read-and-write-in-firestore>

