EXCESS FOOD MANAGEMENT SYSTEM

## A MINI-PROJECT REPORT

***Submitted by***

|  |  |
| --- | --- |
| **BHARATHEESHWAR S** | **210701041** |
| **BHUVANESH C S** | **210701042** |
| **DEVADHARSHINI D** | **210701049** |
| **DHANUSH S** | **210701052** |
| **DHARSHINI S** | **210701055** |
| **DHARUNPRASANTH S** | **210701056** |
| **GURU PRASATH T** | **210701064** |
| **KISHORE S** | **210701501** |

## in partial fulfilment of the award of the degree

***of***

# BACHELOR OF ENGINEERING

### IN

**COMPUTER SCIENCE AND ENGINEERING**

**RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI**



**MAY 2023**

## BONAFIDE CERTIFICATE

Certified that this project **“EXCESS** FOOD MANAGEMENT SYSTEM**”** is the bonafide work of **“BHARATHEESHWAR S**,

## BHUVANESH C S, DEVADHARSHINI D, DHANUSH S DHARSHINI S, DHARUN PRASATH S, GURU PRASATH T, KISHORE S” who carried out the project under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| **Dr. N. Duraimurugan** | **Mrs. M. BHAVANI** |
| **ACADEMIC HEAD,**  **ASSOCIATE PROFESSOR** | **ASSISTANT PROFESSOR** |
| Dept. of Computer Science and Engg, | Dept. of Computer Science and Engg, |
| Rajalakshmi Engineering College, Chennai | Rajalakshmi Engineering College, Chennai |

This mini project report is submitted for the viva voce examination to be held on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

## ABSTRACT

The main idea of our app is to use wasted food in a useful way and to help the needy hands. Our app will be very efficient during 10:30(post breakfast), 2:30(post lunch). Will be there to collect the food when there is an excessive amount of meal if we receive the call through this app. We are planning to develop the app using the java application with the help of android studios. We are also planning to develop a database containing the status and the name of the hotels that are going to sign in with our application. So, our app will have both front end and back-end development features. When the hotels are ready to give food, they will press one button which will be provided by us in the app, and the information will reach us and we will do our best to serve the needy.

## ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M. THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance.

No Words of Gratitude Will Suffice for The Unquestioning Support Extended to Us by Our Head of The Department **Dr. P. Kumar, For** Being Ever Supporting Force During Our Project Work.

We also extend our sincere and hearty thanks to our internal guide **Mrs. M. Bhavani,** for his valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and

other staff members of information technology.

BHARATHEESHWAR S BHUVANESH C S DEVADHARSHINI D DHANUSH S DHARSHINI S DHARUNPRASATH S GURU PRASATH T KISHORE S

**APPENDIX 3**

## TABLE OF CONTENTS CHAPTER NO. TITLE PAGE

|  |  |  |
| --- | --- | --- |
|  | **ABSTRACT** |  |
| **1** | **INTRODUCTION** | **06** |
|  | 1.1 INTRODUCTION | 06 |
|  | 1.2 SCOPE OF THE WORK | 06 |
|  | 1.3 PROBLEM STATEMENT |  |
|  | 1.4 AIM AND OBJECTIVES OF THE PROJECT | 07 |
|  |  |  |
| **2** | **SYSTEM SPECIFICATIONS** | **08** |
|  | 2.1 HARDWARE SPECIFICATIONS | 08 |
|  | 2.2 SOFTWARE SPECIFICATIONS | 08 |
|  |  |  |
| **3** | **MODULE DESCRIPTION** | **09** |
| **4** | **SYSTEM DESIGN** | **10** |
|  | 4.1 ARCHITECTURE DIAGRAM | 10 |
|  | 4.2 USECASE DIAGRAM | 11 |
|  | 4.3. CLASS DIAGRAM | 12 |
|  | 4.4 SEQUENCE DIAGRAM | 13 |
|  | 4.5 ACTIVITY DIAGRAM | 14 |
|  | 4.6 STATE DIAGRAM | 15 |
|  |  |  |
| **5** | **CODING** | **16** |
| **6** | **SCREEN SHOTS** | **32** |
|  |  |  |
| **7** | **CONCLUSION AND FUTURE ENHANCEMENT** | **40** |

**REFERENCES 41**

# CHAPTER 1

## INTRODUCTION

**INTRODUCTION**

An excess food management system is a process or framework designed to manage surplus food effectively, prevent or minimize food waste, and promote sustainability. It involves the collection, transportation, storage, and distribution of excess food from businesses, institutions, and households to those in need. By implementing an excess food management system, surplus food can be redirected to those in need, reducing the environmental impact of food waste and providing food security to vulnerable populations. Overall, an effective excess food management system is critical to promoting sustainability, reducing food waste, and ensuring equitable access to food for all.

## SCOPE OF THE WORK

The scope of work for an excess food management system can include the collection of surplus food from various sources, such as businesses, restaurants, and households. The system can ensure that surplus food is safe and of high quality by implementing appropriate storage, handling, and transportation procedures. This android-based Excess Food Management system can assist in collecting the leftover food from hotels & restaurants to distribute among those in need. NGOs helping poor communities to battle against starvation &malnutrition can raise a request for food supply from restaurants through this application. Once the request is accepted, the NGOs can collect the food from the restaurants for its distribution. In this way this android-based Excess food management system will help restaurants to reduce food waste and will help in feeding the poor and needy people. Overall, the scope of work for an excess food management system involves the coordination of activities related to the collection, redistribution, and reduction of surplus food in a sustainable and equitable manner

## PROBLEM STATEMENT

A drastic increase can be seen in food waste. As per data given by Food and Agriculture Organization, 1/3rd of food produced for human consumption is wasted globally, which accounts for almost 1.3 billion tons per year. On the other hand, also as per WHO 20% of the population face extreme food shortages. Hence there is a need to come up with a solution that can avoid food waste & can help feed the needy

## AIM AND OBJECTIVES OF THE PROJECT

The aim of an excess food management system is to prevent or minimize food waste and ensure that surplus food is effectively managed in a sustainable manner. The primary objective is to maximize the utilization of excess food, thereby reducing its environmental impact and benefiting society. Some specific objectives of an excess food management system may include:

1. Reducing food waste.
2. Ensuring food safety and quality.
3. Providing food security.
4. Promoting sustainability.

Overall, the aim and objective of an excess food management system is to create a more sustainable and equitable food system that benefits all stakeholders, including producers, consumers, and the environment.

**CHAPTER 2**

**SYSTEM SPECIFICATIONS**

* 1. HARDWARE SPECIFICATIONS Processor : Intel i5

Memory size : 8GB(Minimum)

HDD : 1 TB(Minimum)

* 1. SOFTWARE SPECIFICATIONS

Operating System : WINDOWS 11

Front – End : Flutter, Dart

Back -End : Firebase

Language : Dart

# CHAPTER 3

## MODULE DESCRIPTION

### Module Name:

main.dart - It contain login page, serve and receive page. In case the user is not yet registered, he can enter the details and register to create his account. Once his account is created, he can ‘Login’ which asks the user to type his username and password. If the user entered either his username or password incorrectly then an error message appears.

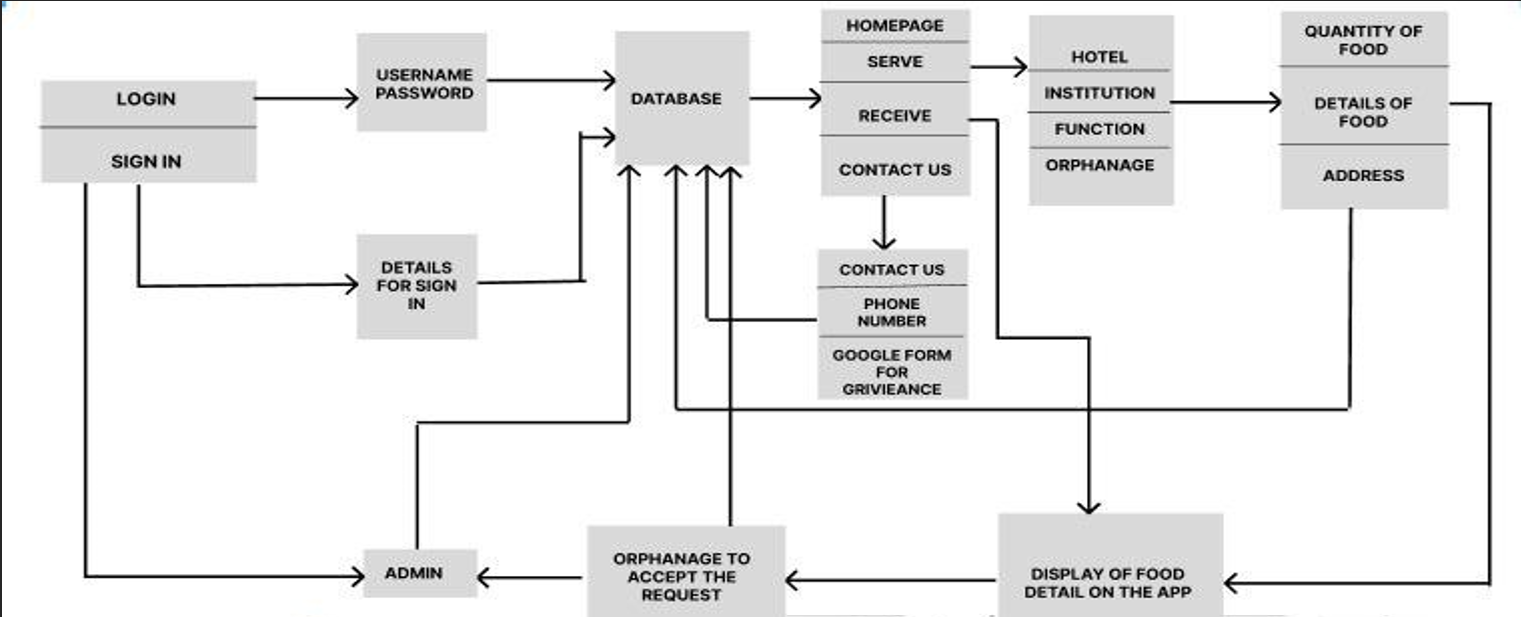
backend.dart - It contain page of food item list and backend connectivity. Category’s view shows the categories of food available and provides the ability to the hotel user to add/edit or delete categories from the list.

fooddetails. dart - It contains the page where the food items are added by the hotel users. This control panel will allow hotel users to add/remove Food; add, edit, or remove a resource. They can edit the frequency of food availability and the amount of food they can serve.

# CHAPTER 4

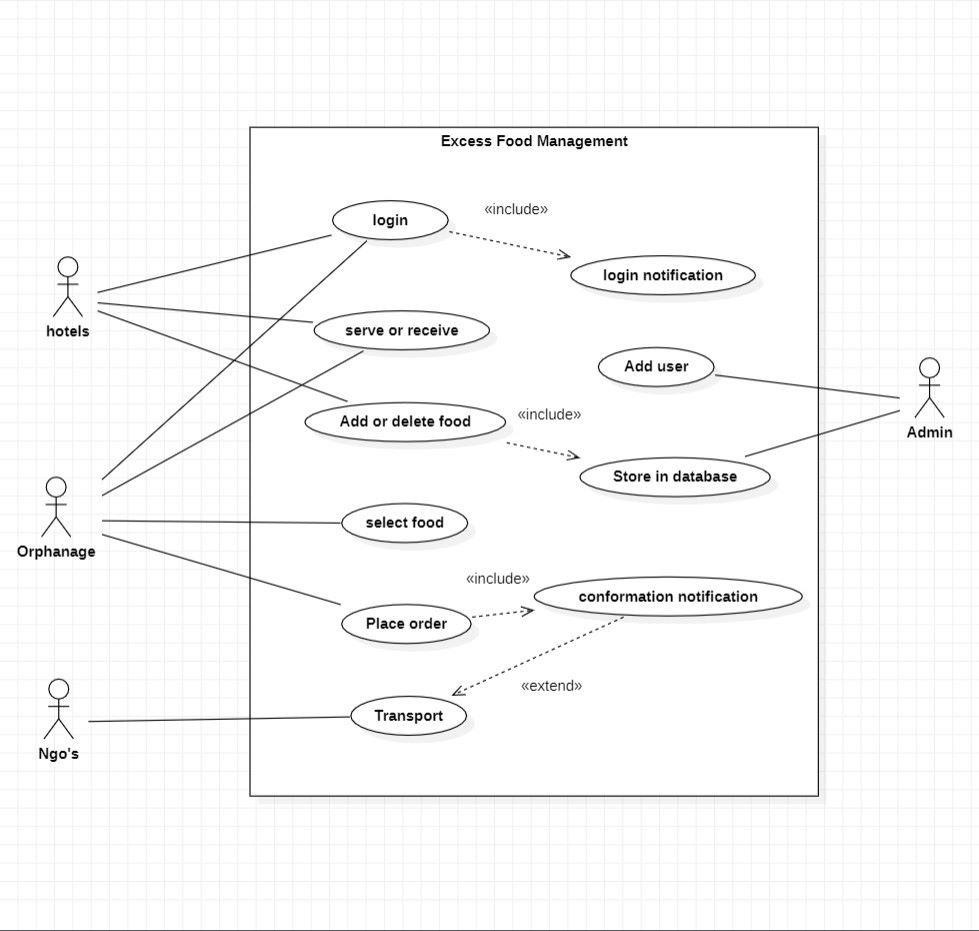
## SYSTEM DESIGN

* 1. **ARCHITECTURE DIAGRAM**

****

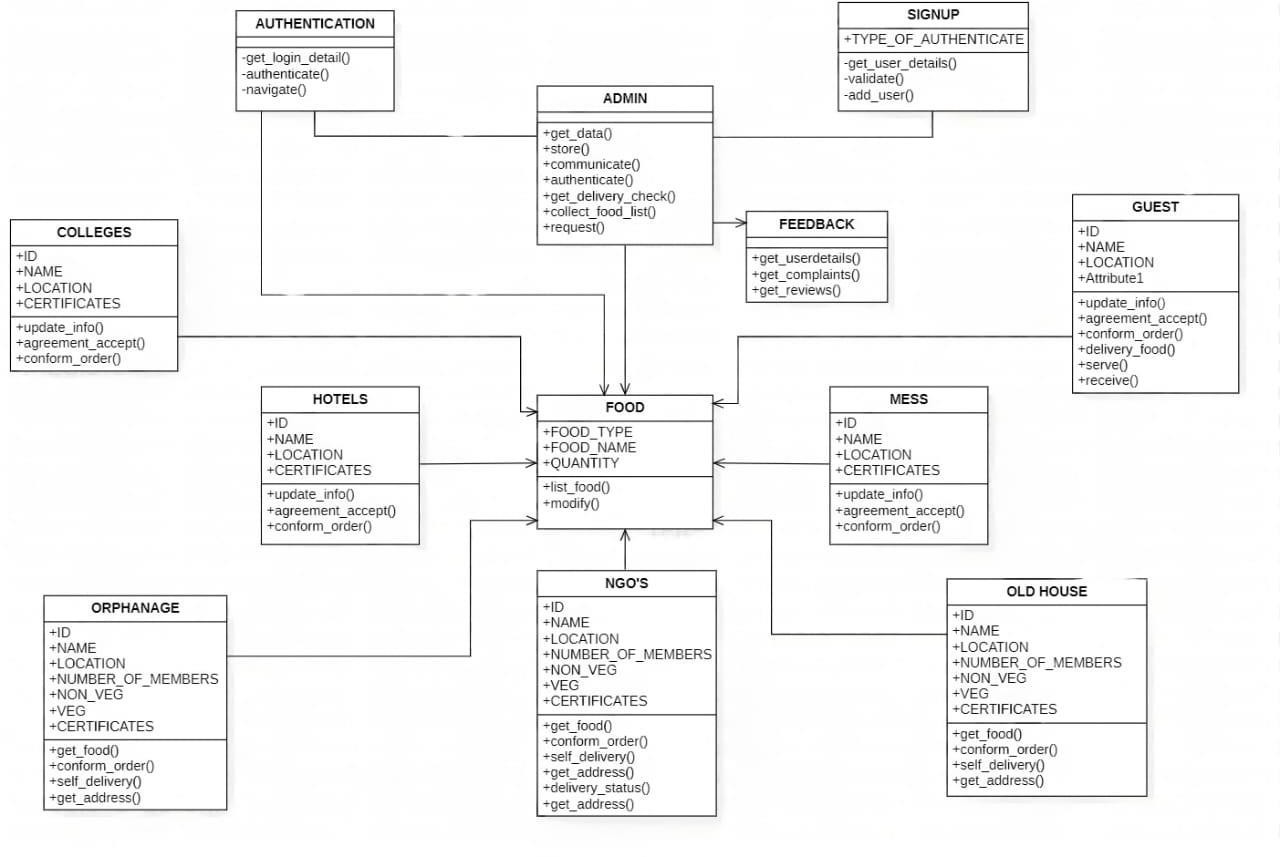
## USECASE DIAGRAM

A use case is a description of how a user or an actor interacts with a system to achieve a specific goal or objective. It describes the steps that the user takes to interact with the system and the responses that the system provides in return. Use cases are an essential tool in software development as they help to understand and document system requirements, identify potential issues, and ensure that the system meets the needs of its users.



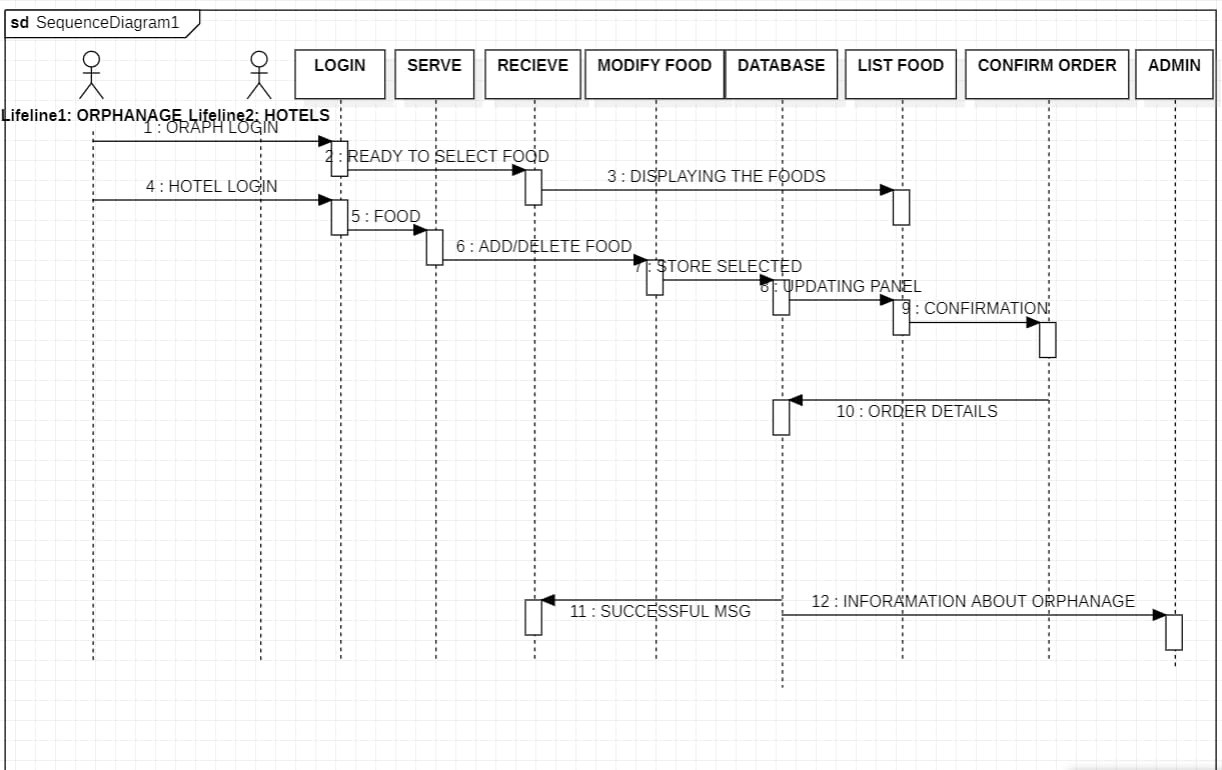
## CLASS DIAGRAM

A class diagram is a type of UML (Unified Modeling Language) diagram used to represent the structure of a system in terms of its classes and their relationships. It is a visual representation of the classes, interfaces, and their associations, which form the building blocks of an object-oriented software system.



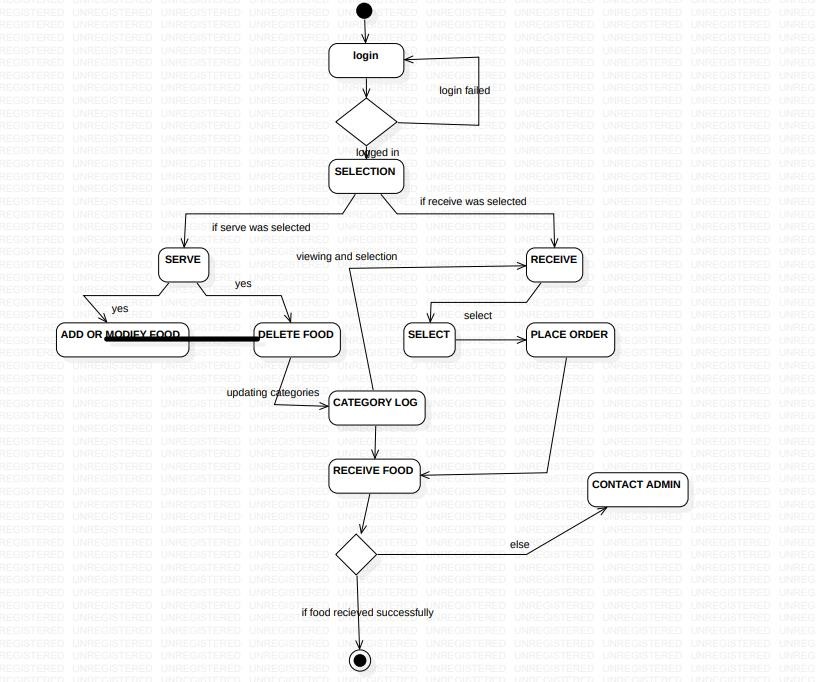
## SEQUENCE DIAGRAM

A sequence diagram is a type of UML (Unified Modeling Language) diagram that shows how objects in a system interact with each other in a particular sequence or order. It represents the behavior of the system over time, by showing the flow of messages or events between objects.



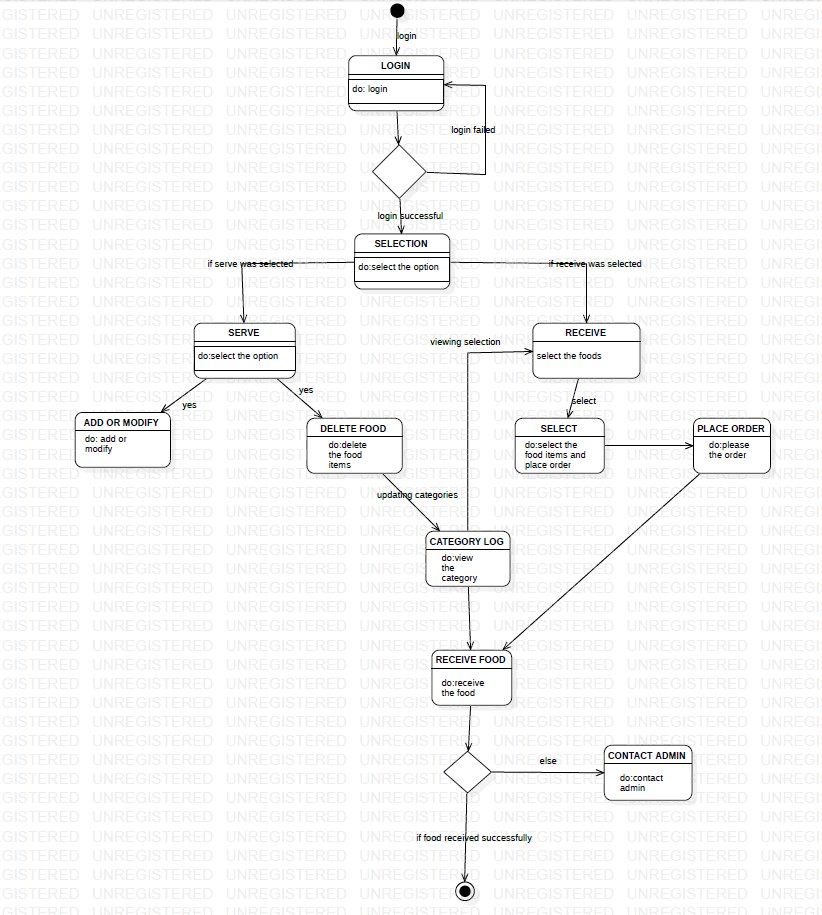
## ACTIVITY DIAGRAM

An activity diagram is a type of UML (Unified Modeling Language) diagram that shows the flow of activities or actions in a system. It is a graphical representation of a workflow or a business process, and it illustrates how the activities are connected and how they interact with each other.



## STATE DIAGRAM

A state diagram is a type of UML (Unified Modeling Language) diagram that represents the different states that an object or a system can be in, and how it transitions from one state to another based on internal or external events. It shows the behavior of the system or object over time.



# CHAPTER 5

## SAMPLE CODING

**main.dart**

import 'package:cloud\_firestore/cloud\_firestore.dart'; import 'package:flutter/material.dart';

import 'package:second/db/backend.dart'; import 'package:second/db/Receivehistory.dart'; import 'package:second/fooddetail.dart';

import 'package:url\_launcher/url\_launcher.dart'; import 'package:firebase\_auth/firebase\_auth.dart'; import 'package:firebase\_core/firebase\_core.dart'; String? \_user, \_pass;

void main() async { WidgetsFlutterBinding.ensureInitialized(); await Firebase.initializeApp(); runApp(const MaterialApp( debugShowCheckedModeBanner: false, home: FirstRoute(),

));}

class FirstRoute extends StatelessWidget { const FirstRoute({super.key}); @override

Widget build(BuildContext context) { return Container(

width: double.infinity,

padding: const EdgeInsets.fromLTRB(20, 50, 20, 50),

decoration: const BoxDecoration(gradient: LinearGradient( transform: GradientRotation(10), begin: Alignment.topLeft,

end: Alignment.bottomRight, colors: [

Colors.teal, Colors.white, Colors.white, ]), ),

child: Scaffold(backgroundColor: Colors.transparent,

body: ListView(children: [ Center(child: Text("Eat-N-Share",style: TextStyle(fontSize: 30))), SizedBox( height: 50,),

Image.asset('Assets/fooddemo2.png'), const SizedBox(height: 120,), Container(color: Colors.transparent,

child: Center(child: const Text('Login As', style: TextStyle(fontSize: 30))),), const SizedBox(

height: 20,),

Center(child: ElevatedButton( style: ElevatedButton.styleFrom(

backgroundColor: Colors.transparent,),

child: const Text(' Hotel ',style: TextStyle(fontSize: 27.5)),

onPressed: () {

Navigator.push(context, MaterialPageRoute(builder: (context) => const SecondRoute()),);},),), const SizedBox(height: 30,),

Center(child: ElevatedButton(

style: ElevatedButton.styleFrom( backgroundColor: Colors.transparent,), child: Container( padding: EdgeInsets.fromLTRB(21, 8.5, 21, 8.5),

child: const Text('Orphanage', style: TextStyle(fontSize: 20))),

onPressed: () { Navigator.push(context, MaterialPageRoute(builder: (context) => const

ThirdRoute()) );},) ), const SizedBox(height: 20,),

const Center(child: Text('Or',style: TextStyle(fontSize: 25))), const SizedBox( height: 15),

Center(child: TextButton(onPressed: () {

Navigator.push(context, MaterialPageRoute(builder: (context) => const Signup(),)); }, child: const Text('Create Account',

style: TextStyle(fontSize: 21.5, color: Colors.white)))])),); }} GlobalKey<FormState> Fkey = GlobalKey<FormState>();

void delay() {Future.delayed(const Duration(seconds: 4), () {},);} void signin(BuildContext context) async {

await FirebaseAuth.instance .signInWithEmailAndPassword(email: \_user!, password: \_pass!)

.catchError((onError) { return "Username and password not match";

}).then((authUser) {

if (authUser.user != null) {

Navigator.push(context, MaterialPageRoute(builder: (context) => const Secondsub()));}});} void signiin(BuildContext context) async {

await FirebaseAuth.instance

.signInWithEmailAndPassword(email: \_user!, password: \_pass!)

.catchError((onError) { return "Username and password not match";

}).then((authUser) {

if (authUser.user != null) { Navigator.push(

context, MaterialPageRoute(builder: (context) => const backend()));

}});}

class SecondRoute extends StatelessWidget { const SecondRoute({super.key}); @override

Widget build(BuildContext context) { return Container(

width: double.infinity,

padding: const EdgeInsets.fromLTRB(20, 70, 20, 100),

decoration: const BoxDecoration(

gradient: LinearGradient( transform: GradientRotation(10), begin: Alignment.topLeft,

end: Alignment.bottomRight,

colors: [Colors.teal, Colors.white, Colors.white,]),), child: Scaffold(backgroundColor: Colors.transparent, body:Form(

key: Fkey,

child: ListView(children: [

const Text("Hotel's Login:",style: TextStyle(fontSize: 30,color: Colors.black)), const SizedBox(height: 30,),

Image.asset('Assets/login.png'), const SizedBox(height: 50,), TextFormField(

validator: (value) {

if (value!.isEmpty) {

return "Enter the username";

} else if (!(RegExp(r""^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'\*+-/=?^\_`{|}~]+@[a-zA-Z0-9]+\.[a-zA- Z]+""").hasMatch(value))) {

return "Enter a valid email id ";

} else {

\_user = value; } return null; },

onSaved: (String? value) {

\_user = value!; },

decoration: const InputDecoration( iconColor: Colors.teal, focusedBorder: OutlineInputBorder(

borderSide: BorderSide(color: Colors.teal, width: 2), borderRadius: BorderRadius.all(Radius.circular(5.0))),

icon: Icon(Icons.account\_circle), hintText: "Username"),), const SizedBox(height: 20),

TextFormField( obscureText: true, validator: (String? pass) { if (pass!.isEmpty) {

return "Enter the Password";

} else {

if (!(pass.length > 8 && pass.length < 15)) {

return "Password length must greater than 8 and less than 15";}

return null; } },

onSaved: (String? pass) => \_pass = pass!, decoration: const InputDecoration( iconColor: Colors.teal,

focusedBorder: OutlineInputBorder(

borderSide: BorderSide(color: Colors.teal, width: 2), borderRadius: BorderRadius.all(Radius.circular(5.0))),

icon: Icon(Icons.lock), hintText: "Password"),), const SizedBox(height: 70),

ElevatedButton(style: ElevatedButton.styleFrom( backgroundColor: Colors.transparent,), onPressed: () {

if (Fkey.currentState!.validate()) { Fkey.currentState!.save(); signiin(context); }},

child: const Text('Login'),)]))));}}

class ThirdRoute extends StatelessWidget { const ThirdRoute({super.key}); @override

Widget build(BuildContext context) { return Container(

padding: const EdgeInsets.fromLTRB(20, 70, 20, 10), decoration: const BoxDecoration(

gradient: LinearGradient( transform: GradientRotation(10), begin: Alignment.topLeft,

end: Alignment.bottomRight,

colors: [Colors.teal, Colors.white, Colors.white,]),), child: Scaffold(backgroundColor: Colors.transparent, body:Form(key: Fkey,

child: ListView(children: [

const Text("Orphanage's Login:",style: TextStyle(fontSize: 25)),

Image.asset('Assets/login.png'), TextFormField(

validator: (value) {

if (value!.isEmpty) {

return "Enter the username";…}

### Fooddetail.dart

import 'package:cloud\_firestore/cloud\_firestore.dart'; import 'package:flutter/material.dart';

final FirebaseFirestore \_firestore = FirebaseFirestore.instance;

final CollectionReference \_Collection = \_firestore.collection('My student');

class fooddetail extends StatefulWidget { final String? user;

fooddetail({this.user}); @override

State<fooddetail> createState() => \_fooddetailState(user: user);

}

class \_fooddetailState extends State<fooddetail> { String? user;

\_fooddetailState({this.user}); bool valuefirst = false;

late String ph, fname, pname, add; late String quantity; getfood(String a) => fname = a;

getprovider(String b) => pname = b; getquantity(String c) => quantity = c; getadd(String d) => add = d; getph(String e) => ph = e;

Future createdata() async {

DocumentReference documentReferencer = \_Collection.doc(); Map<String, dynamic> food = <String, dynamic>{

"Food name": fname, "Provider": pname, "Quantity": quantity, "Address": add,

"Phno": ph, "Transport": valuefirst

};

await documentReferencer

.set(food)

.whenComplete(() => print("Successfull"))

.catchError((e) => print(e));

}

@override

Widget build(BuildContext context) {

return Container(

decoration: const BoxDecoration( gradient: LinearGradient(

transform: GradientRotation(10), begin: Alignment.topLeft,

end: Alignment.bottomRight, colors: [

Colors.teal, Colors.white, Colors.white,

]

)

),

child: Scaffold(backgroundColor: Colors.transparent, appBar: AppBar(

title: const Text("Food Details"), centerTitle: true,

flexibleSpace: Container( decoration: const BoxDecoration(

gradient: LinearGradient(

begin: Alignment.topCenter, end: Alignment.bottomCenter, colors: [

//Colors.purpleAccent, Colors.black38, Colors.teal,

])),

),

backgroundColor: Colors.transparent),

body: ListView( children: [

const SizedBox( height: 40,

),

Padding(

padding: const EdgeInsets.all(8.0), child: TextFormField(

decoration: const InputDecoration( labelText: "Food name", fillColor: Colors.white70, focusColor: Colors.teal, iconColor: Colors.teal,

labelStyle: TextStyle(color: Colors.black87), focusedBorder: OutlineInputBorder(

borderSide: BorderSide(color: Colors.teal, width: 2), borderRadius: BorderRadius.all(Radius.circular(5.0)),

),

icon: Icon(Icons.food\_bank)),

onChanged: (String value) => getfood(value),

),

),

const SizedBox( height: 40,

),

Padding(

padding: const EdgeInsets.all(8.0), child: TextFormField(

decoration: const InputDecoration( labelText: "Provider name", iconColor: Colors.teal, fillColor: Colors.white70,

labelStyle: TextStyle(color: Colors.black87), focusedBorder: OutlineInputBorder(

borderSide: BorderSide(color: Colors.teal, width: 2), borderRadius: BorderRadius.all(Radius.circular(5.0))),

icon: Icon(Icons.man)),

onChanged: (String value) => getprovider(value)),

),

const SizedBox(

height: 40,

),

Padding(

padding: const EdgeInsets.all(8.0), child: TextFormField(

obscureText: false,

decoration: const InputDecoration( labelText: "Quantity",

fillColor: Colors.white70, iconColor: Colors.teal,

labelStyle: TextStyle(color: Colors.black87),

focusedBorder: OutlineInputBorder(

borderSide: BorderSide(color: Colors.teal, width: 2), borderRadius: BorderRadius.all(Radius.circular(5.0))),

icon: Icon(Icons.production\_quantity\_limits)), onChanged: (String value) => getquantity(value)),

),

const SizedBox( height: 40,

),

Padding(

padding: const EdgeInsets.all(8.0),

child: TextFormField(

decoration: const InputDecoration( labelText: "Enter the Address", fillColor: Colors.white70, iconColor: Colors.teal,

labelStyle: TextStyle(color: Colors.black87), focusedBorder: OutlineInputBorder(

borderSide: BorderSide(color: Colors.teal, width: 2), borderRadius: BorderRadius.all(Radius.circular(5.0))),

icon: Icon(Icons.add\_location)),

onChanged: (String value) => getadd(value)),

),

const SizedBox( height: 40,

),

Padding(

padding: const EdgeInsets.all(8.0), child: TextFormField(

decoration: const InputDecoration( labelText: "Enter the Phone number", fillColor: Colors.white70,

iconColor: Colors.teal,

labelStyle: TextStyle(color: Colors.black87), focusedBorder: OutlineInputBorder(

borderSide: BorderSide(color: Colors.teal, width: 2), borderRadius: BorderRadius.all(Radius.circular(5.0))),

icon: Icon(Icons.add\_call)), onChanged: (value) => getph(value)),

),

const SizedBox(

height: 64,

),

SizedBox( width: 375, child: Row( children: [

const Padding(

padding: EdgeInsets.fromLTRB(8.0, 0, 5, 0),

child: Icon(Icons.delivery\_dining, color: Colors.teal),

),

const Text(" Transport facility:",

style: TextStyle(fontSize: 17.5, color: Colors.black87)), Checkbox(

checkColor: Colors.greenAccent, activeColor: Colors.teal,

value: valuefirst, onChanged: (bool? value) { setState(() {

valuefirst = value!;

});

},

),

],

),

),…}

### Backend.dart

import 'package:cloud\_firestore/cloud\_firestore.dart'; import 'package:flutter/material.dart';

import 'package:second/main.dart'; import '../fooddetail.dart';

class backend extends StatefulWidget {

const backend({Key? key}) : super(key: key);

@override

State<backend> createState() => \_backendState();

}

class \_backendState extends State<backend> { @override

Widget build(BuildContext context) { return Container(

width: double.infinity,

padding: const EdgeInsets.fromLTRB(0, 0, 0, 10), decoration: const BoxDecoration(

gradient: LinearGradient( transform: GradientRotation(10), begin: Alignment.topLeft,

end: Alignment.bottomRight, colors: [

//fColors.purple, Colors.teal, Colors.white, Colors.white,

]),

),

child: Scaffold(backgroundColor: Colors.transparent, appBar: AppBar(

flexibleSpace: Container( decoration: const BoxDecoration(

gradient: LinearGradient( begin: Alignment.topCenter,

end: Alignment.bottomCenter,

colors: [

//Colors.purpleAccent, Colors.black38, Colors.teal,

])),

),

backgroundColor: Colors.transparent,

title: const Text("Dashboard",style: TextStyle(fontSize: 30)), centerTitle: true,

),

drawer: Drawer(

backgroundColor: Colors.transparent,

child: Column( children: <Widget>[

const UserAccountsDrawerHeader(

accountName: Text('abc', style: TextStyle(fontSize: 20)), accountEmail:

Text('ABC@gmail.com', style: TextStyle(fontSize: 20)), decoration: BoxDecoration(color: Colors.teal), currentAccountPicture: CircleAvatar(

backgroundColor: Colors.black45,

child: Text('A',

style:

TextStyle(fontSize: 30, fontWeight: FontWeight.bold)),

),

),

const SizedBox( height: 400,

),

ElevatedButton( onPressed: () { Navigator.push(context,

MaterialPageRoute(builder: (context) => const Third()));

},

style: ElevatedButton.styleFrom( backgroundColor: Colors.transparent,

),

child: const Text( 'Contact us',

style: TextStyle(fontSize: 20),

),

)

],

),

),

body: Center(

child: ListView(children: [ const SizedBox(

height: 50,

),

ElevatedButton( onPressed: () { Navigator.push(context,

MaterialPageRoute(builder: (context) => fooddetail(user: "")));

},

style: ElevatedButton.styleFrom(

tapTargetSize:

MaterialTapTargetSize.shrinkWrap, backgroundColor: Colors.transparent, shadowColor: Colors.transparent

// tapTargetSize:

),

child:Padding(padding:const EdgeInsets.all(0), child: Card(

color: Colors.transparent, child: Column(

children: [ Padding(

padding: const EdgeInsets.all(30.0), child: Image.asset("Assets/Serve.png"),

),

const Text( 'serve',

style: TextStyle(color: Colors.white,fontSize: 50, fontWeight: FontWeight.w400)

)]

),

)

)

),

const SizedBox(height: 50), ElevatedButton(

onPressed: () { Navigator.push(context,

MaterialPageRoute(builder: (context) => const Request()));

},

style: ElevatedButton.styleFrom( tapTargetSize:

MaterialTapTargetSize.shrinkWrap, backgroundColor: Colors.transparent, shadowColor: Colors.transparent

// tapTargetSize:

),

child:Padding(padding:const EdgeInsets.fromLTRB(0,0,0,0), child: Card(

color: Colors.transparent, child: Column(

children: [ Padding(

padding: const EdgeInsets.fromLTRB(72,30,72,30), child: Image.asset("Assets/request final.png"),

),

const Text( 'Receive',

style: TextStyle(color: Colors.white,fontSize: 50, fontWeight: FontWeight.w400)

),

const SizedBox( height: 10,

)

]

),

)

)

),

const SizedBox(height: 50),

]),

),

),

);

}

}

//final FirebaseFirestore \_firestore = FirebaseFirestore.instance;

// ignore: non\_constant\_identifier\_names

//final CollectionReference \_Collection = \_firestore.collection('My student'); class Request extends StatefulWidget {

const Request({Key? key}) : super(key: key);

@override

State<Request> createState() => \_RequestState();

}

class \_RequestState extends State<Request> { @override

Widget build(BuildContext context) { return Container(

width: double.infinity,

padding: const EdgeInsets.fromLTRB(2, 50, 20,0), decoration: const BoxDecoration(

gradient: LinearGradient( begin: Alignment.topCenter, end: Alignment.bottomCenter, colors: [

//Colors.purple, Colors.white, Colors.white, Colors.teal,

],

),

),

child: Scaffold(

backgroundColor: Colors.transparent, appBar: AppBar(

centerTitle: true, backgroundColor: Colors.white, shadowColor: Colors.transparent,

title: const Text("Requested:",style: TextStyle(color: Colors.black,fontSize: 30)),

),

body: StreamBuilder(

stream: FirebaseFirestore.instance.collection("Request").snapshots(), builder: (context, snapshot) {

if (snapshot.hasData) { return ListView.builder( shrinkWrap: true,

itemCount: snapshot.data!.docs.length, itemBuilder: (BuildContext context, int index) { DocumentSnapshot documentSnapshot = snapshot.data!.docs[index];

return Card(

margin: const EdgeInsets.fromLTRB(25,25,10,25),

color: Colors.white, child: Container(

decoration: const BoxDecoration( gradient: LinearGradient(

colors: [ Colors.blueGrey, Colors.teal

]

)

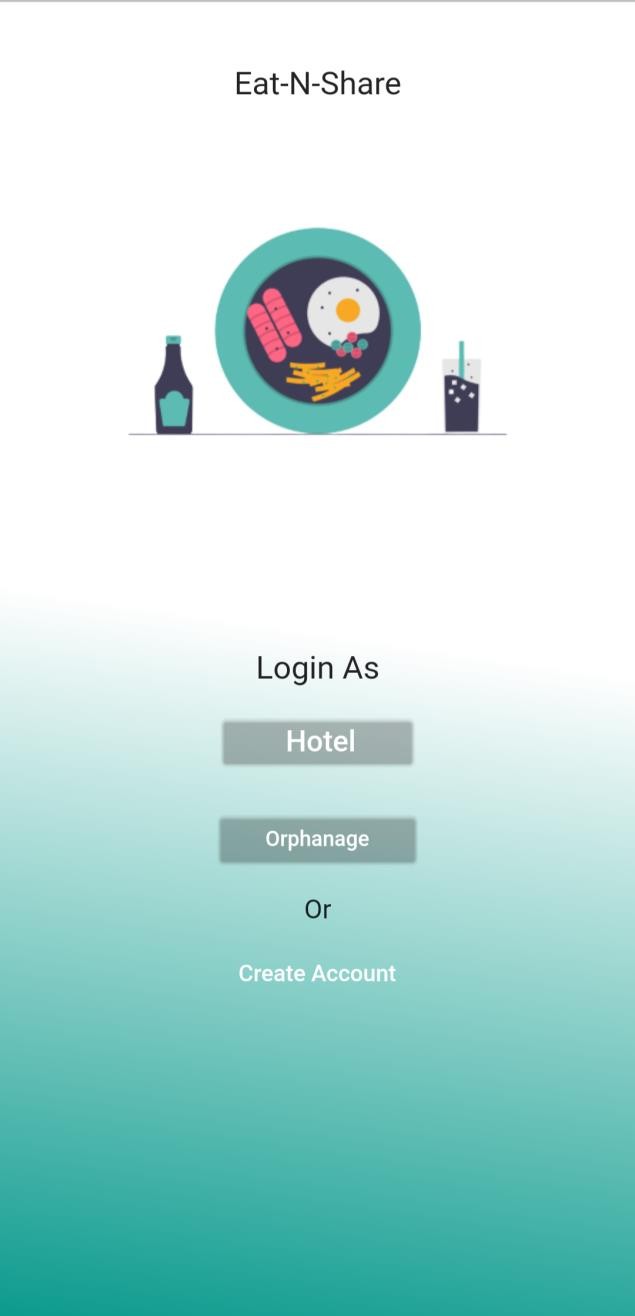
),

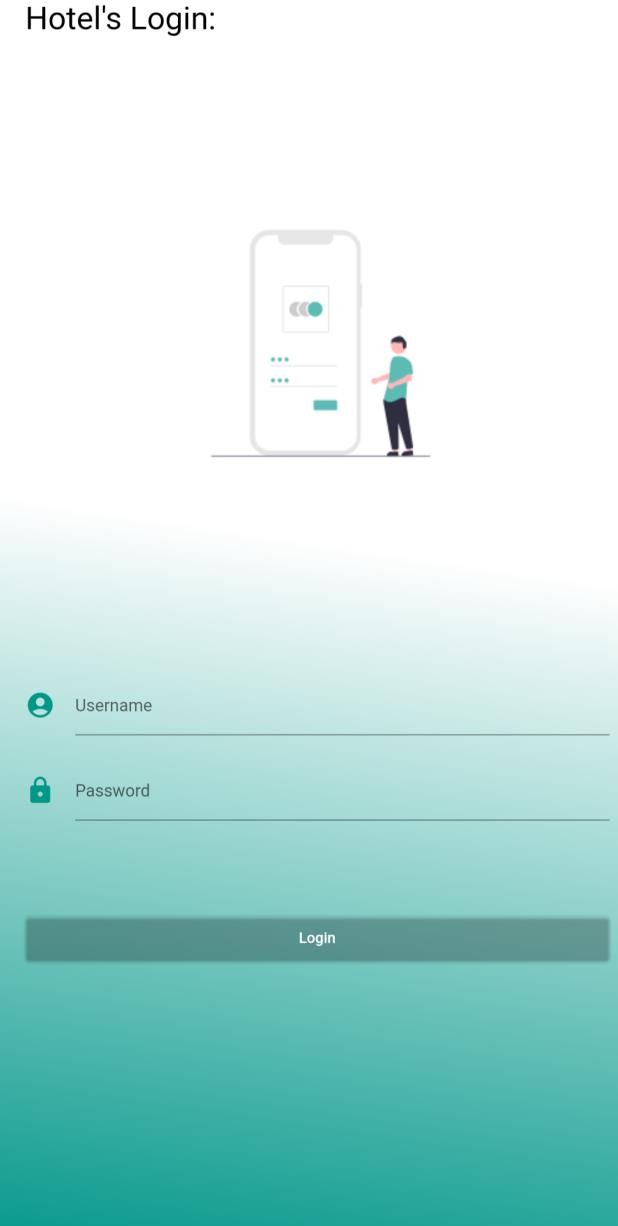
child: Column( children: <Widget>[ Padding(

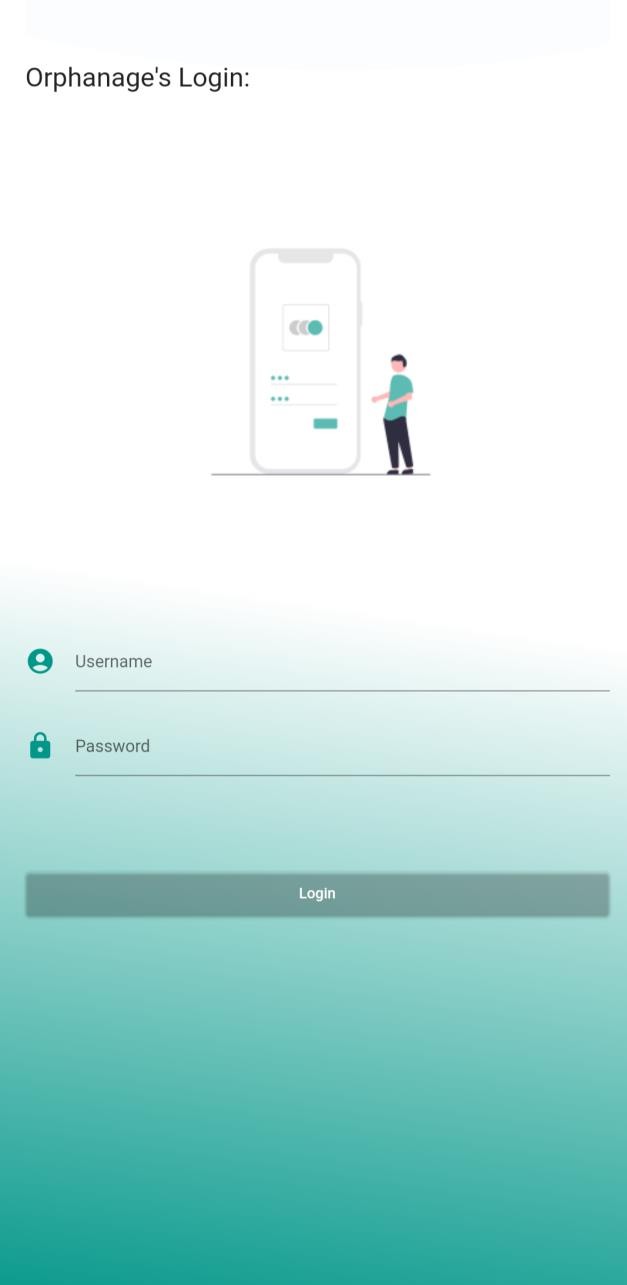
padding: const EdgeInsets.fromLTRB(5.0, 8, 5, 2), child: Row(…..}

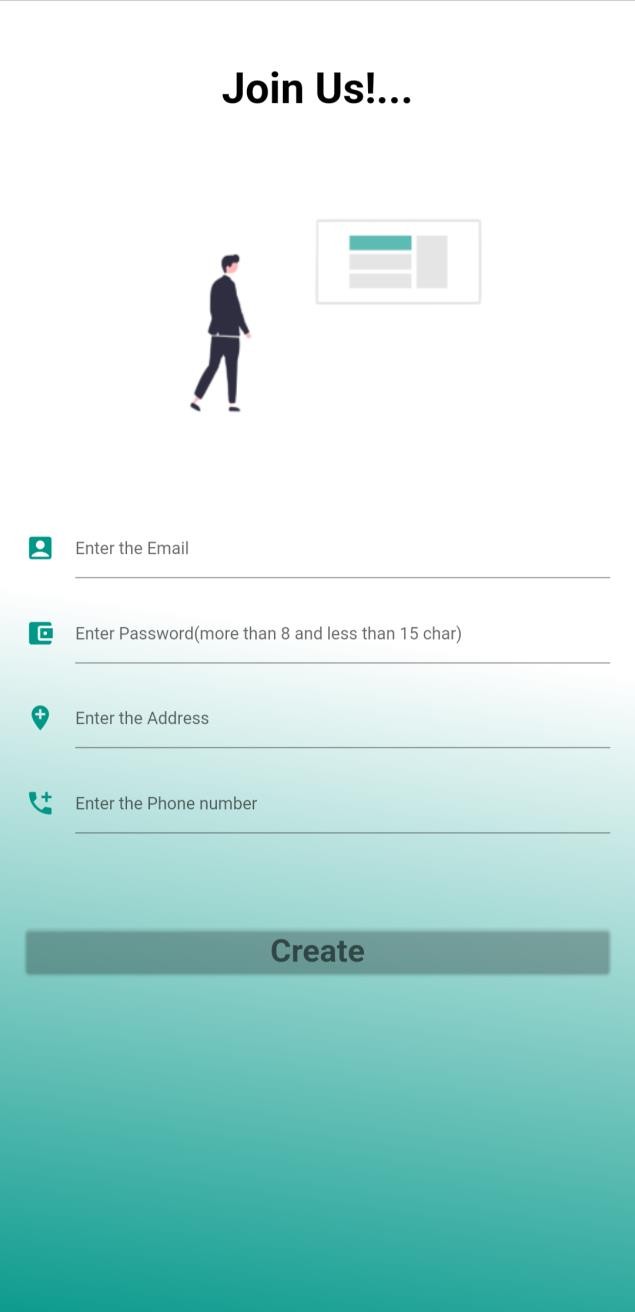
# CHAPTER 6

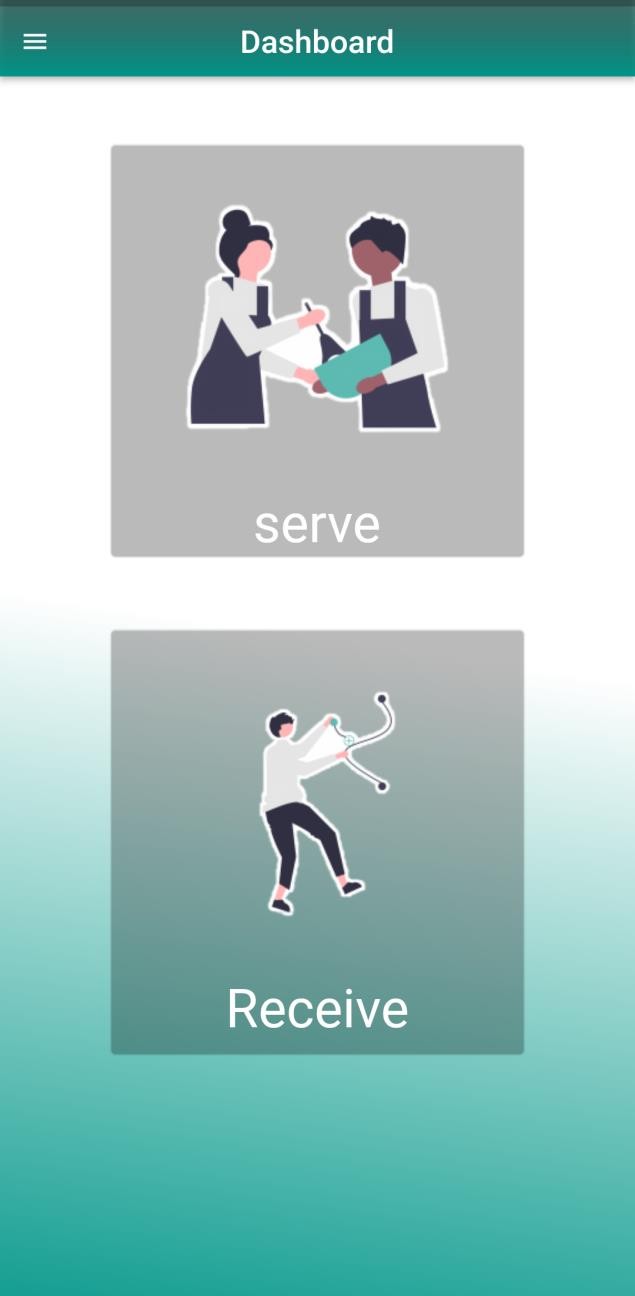
**SCREEN SHOTS**

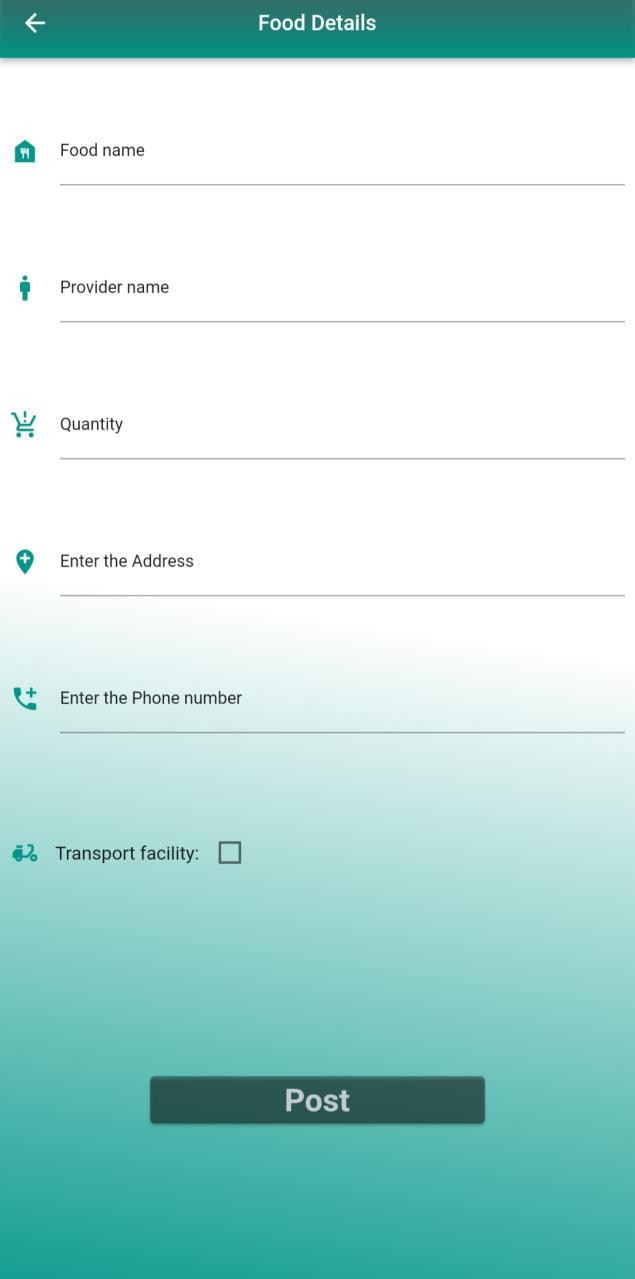


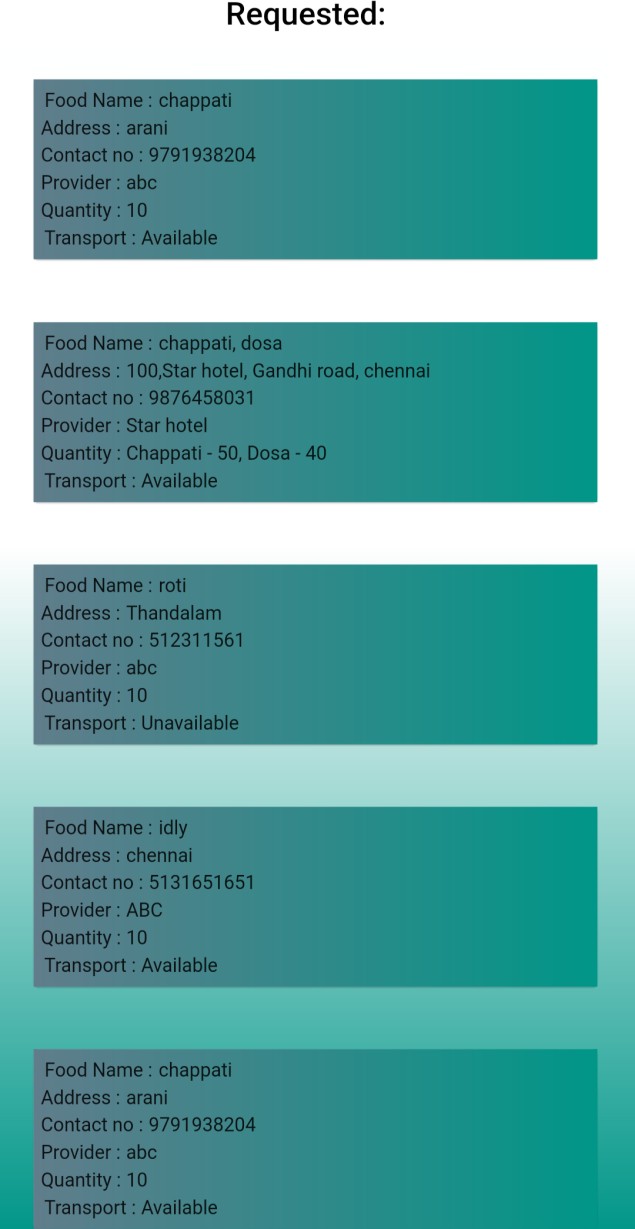


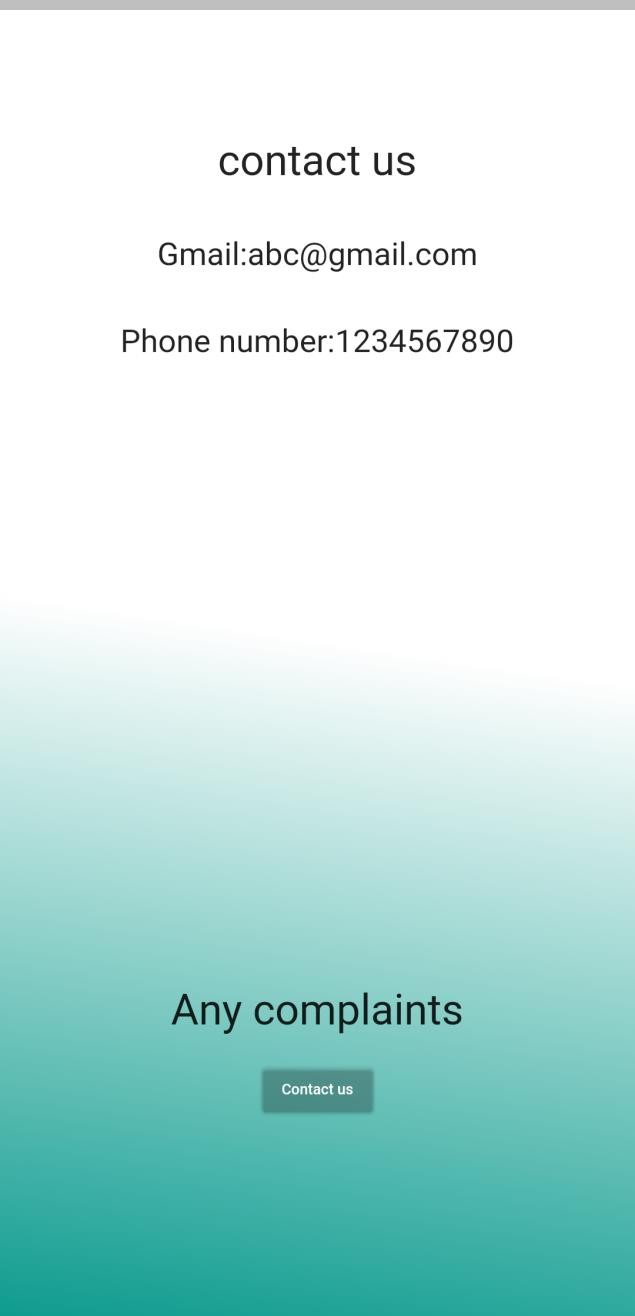












# CHAPTER 7

## 

## CONCLUSION AND FUTURE ENHANCEMENT

As of now we conclude that, the admin group will work similar to that of the call centers, who will clarify the customer’s queries and need. In future we are planning to make it as ai bot which will communicate with the user in an efficient manner to an extent and half of the admin’s job will be reduced. We are also studying about the training principles of the “Raza” bot. And in future we need to improve the user interface in an interactive and catchy manner.

# REFERENCES

* W3SCHOOLS.COM
* Flutter Cookbook
* Beginning App Development with Flutter
* Angela Yu Lectures
* Mobile Deep Learning with TensorFlow Lite, ML Kit

and Flutter

* Indian Economy by Ramesh Singh
* Indian Economy by Sanjiv Verma
* Dart Book
* Times of India