Summer Project On

Synergy

By

Anish Redkar (2021510055)
Omkar Ingle (2021510022)

Under the guidance of

Internal Supervisor

Prof. Nikhita Mangaonkar.

Department of Master Of Computer Application
Sardar Patel Institute of Technology
Autonomous Institute Affiliated to Mumbai University

2022-23

CERTIFICATE OF APPROVAL

This is to certify that the following students

Anish Redkar (2021510055)
Omkar Ingle (2021510022)

Have satisfactorily carried out work on the project
entitled

"Synergy"

Towards the fulfilment of project, as laid down
by
Sardar Patel Institute of Technology
during year
2022-23.

Project Guide:
Prof. Nikhita Mangaonkar

PROJECT APPROVAL CERTIFICATE

This is to certify that the following students

Anish Redkar (2021510055)
Omkar Ingle (2021510022)

Have successfully completed the Project report on

"Synergy",

which is found to be satisfactory and is approved

at

SARDAR PATEL INSTITUTE OF TECHNOLOGY,
ANDHERI (W), MUMBAI

INTERNAL EXAMINER EXTERNAL EXAMINER

HEAD OF DEPARTMENT PRINCIPAL

Contents

Abstract

Synergy is an online food delivery platform. Synergy includes 3 applications which are cross platform, The 3 applications are Customer app, Seller app, and the Delivery partner app.

Customer app – This app is for the customers who wants to order food online. Synergy customer app providers different restaurants and their menu items for order food with easy-to-use functionality simple user-friendly UI.

Seller App – This app provides various restaurants the opportunity to sell food online on the synergy platform with easy-to-use UI and functionality for providing fast and efficient delivery without any hassle, this provides more opportunity to grow business and reach to the customers.

Delivery Partner App – Anyone can sign up for becoming a delivery partner on this app. This app provides all the functionality to the partner for making successful deliveries on time and making great income.

Objectives

The Flutter based Application "Synergy" is used

- Providing customers easy to use application for ordering food online.
- Providing a platform for restaurants to sell their food items online with hassle free synergy platform.
- To provide all the functionality to the delivery partner for making successful deliveries on time and making great income.

List of Figures

List of Tables

1 Introduction

1.1 Problem Definition

In this world where everything is online now Ordering things online has become common ordering food is not that easy to accomplish as several players are involved in this, for example, restaurants that act as the sellers need a system where they can easily track and manage orders, and deliver them with the help of delivery partners who also need an effective system, which can help delivery partner chose the parcels they want to deliver and then successfully deliver them on time on to the correct customer on the mentioned address, all this is possible with a system and chain of applications which provides all the necessary functionalities and user experience for all the customers, sellers, and the delivery partner. This problem is solved with synergy and the three apps within this platform.

1.2 Objectives and Scope

1.2.1 Objectives

The Flutter based application "Synergy" is

- Providing a platform for restaurants to sell their food items online with hassle free synergy platform.
- To provide all the functionality to the delivery partner for making successful deliveries on time and making great income.
- To provide all the functionality to the delivery partner for making successful deliveries on time and making great income.

1.2.2 Scope

The main aim of Synergy is to bring all the food business online at one place where the customer can order the food items, sellers can take these orders and in turn the delivery agencies can coordinate with the sellers and can deliver the available parcels to the customers that ordered them. This will be beneficial to both the customers, sellers and the delivery partners as the process will be smooth and structured.

1.3 Proposed System

Synergy is a chain of three different flutter applications which acts as a platform for a bunch of activities such as food ordering, food delivery and more. For Database we have made use of googles firebase.

The three applications are:

- Users App
  User app is where the customers can order the food. In this app, on the home page customer can view different restaurants and their menus. The customer can go to any menu and select the desired food item he/she wants to order.
- Seller app
  Seller app is where the restaurants can create their menus and add different food items in them. Here the sellers can see all the orders placed by the customers form the users app. Once, the order is ready for the pickup by the delivery-partners the restaurants can dispatch the order further to the delivery partners app.
- Delivery-Partners App
  The orders that are dispatched form the seller's app are displayed in this app. The delivery men using this app can accept the available orders and pick them up form the restaurant and delivery it to the customer.

1.4 System Requirements

- Hardware Requirements on Server Side

Table 1.5.1: Hardware Requirements on Server Side

Processor	Dual Core Processor or Above
RAM	Minimum 4 GB RAM
Storage	Minimum 10 GB Hard Disk Space for smooth run

- Hardware Requirements on Client Side

Table 1.5.2: Hardware Requirements on Client Side

Device	Smartphone Device with Touch Screen
Processor	Dual Core Processor or Above
RAM	Minimum 2 GB RAM
Storage	Minimum 250 MB Storage Space

- Software Requirements on Server Side

Table 1.5.3: Software Requirements on Server Side

Operating System	OS Independent
Database	Firestore

- Software Requirements on Client Side

Table 1.5.3: Software Requirements on Client Side

Operating System	Android/IOS Smartphone
Server	Not Required

2 Software Requirement Specification (SRS) and Design

2.1 Purpose

Synergy is a combination of three different flutter application. These three applications work in integration with one another to provide a platform for customers, sellers and delivery partners. Customers are able to order food items online, Sellers can create menus and add different food item to the menu and once the parcels are ready, they are dispatched to the delivery partners so that they can pickup the parcel and deliver it to the customer who ordered.

2.2 Overall Description

2.2.1 Product Functions

The product function includes: The basic functionality of this chain of applications is to provide a platform where the restaurants can upload their menus, customers can order their desired food items etc. All the process is done online at one place without creating any mess unlike the traditional one which was not so convenient. Most importantly it saves the time of the customer by breaking the physical barrier and also provides more dynamic structure to the seller's business and delivery partners business.

2.2.2 User Characteristics

There are three types of users:

- Customer App: The users make use of this application to make orders of their desired food items.
- Seller App: The restaurants are the users of this application where they can add menus and keep track of orders from all the customers.
- Delivery agency app: Users of this application are the delivery partners that work in close coordination with the sellers to pick and delivery the available parcels to the customers that ordered it.

2.2.3 Dependencies and Assumptions

The system is dependent on the internet. Since the backend is done using Firebase so there is a need for the internet here as well. Flutter dependencies are subject to minimum hardware requirements to run these applications. There are no other known dependencies or assumptions of this system.

3 Project Analysis and Design

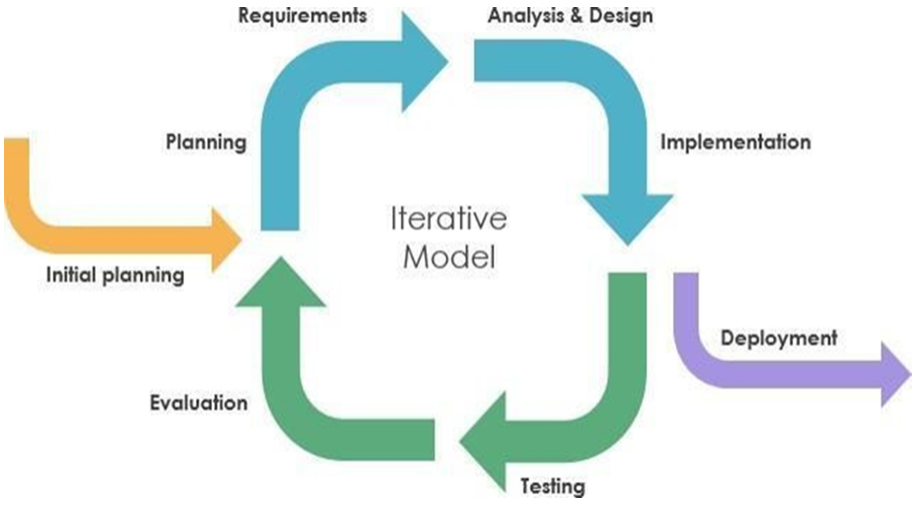
3.1 Methodologies Adapted

The iterative model is a particular implementation of a software development life cycle (SDLC) that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete.

In this Model, you can start with some of the software specifications and develop the first version of the software. After the first version if there is a need to change the software, then a new version of the software is created with a new iteration. Every release of the Iterative Model finishes in an exact and fixed period that is called iteration.

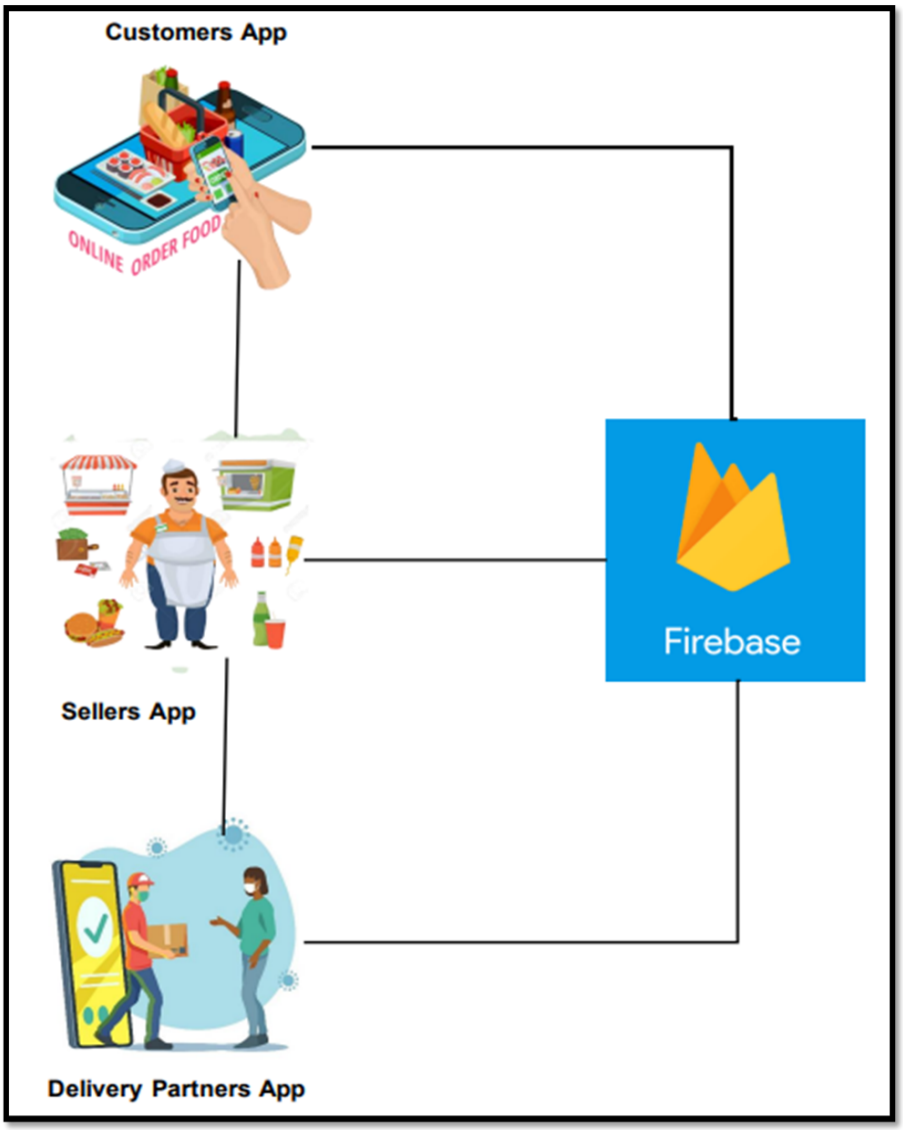
The Iterative Model allows the accessing earlier phases, in which the variations made respectively. The final output of the project renewed at the end of the Software Development Life Cycle (SDLC) process.

While building these applications in the first iteration we focused on creating the simple modules like static pages i.e., splash screen. We focused on building the complex modules such as database connections, fetching data from the data base and displaying that data in the application in second iteration and similarly in each of the further iterations some modules were implemented. And then as modules were completed, they were evaluated and again if any glitches were there, planning was done to solve those and the process went on.



3.1.1: Diagrammatic Representation of Waterfall Model

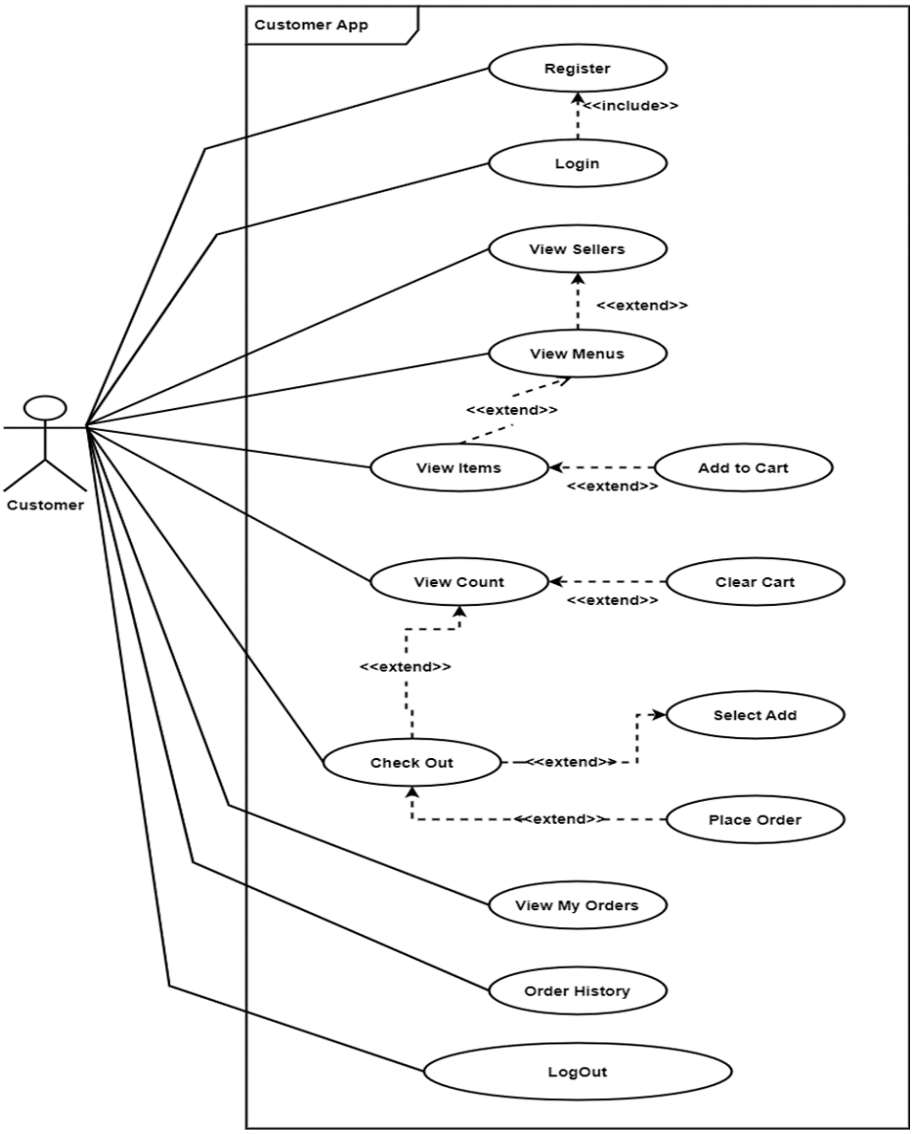
3.2 Architectural Design



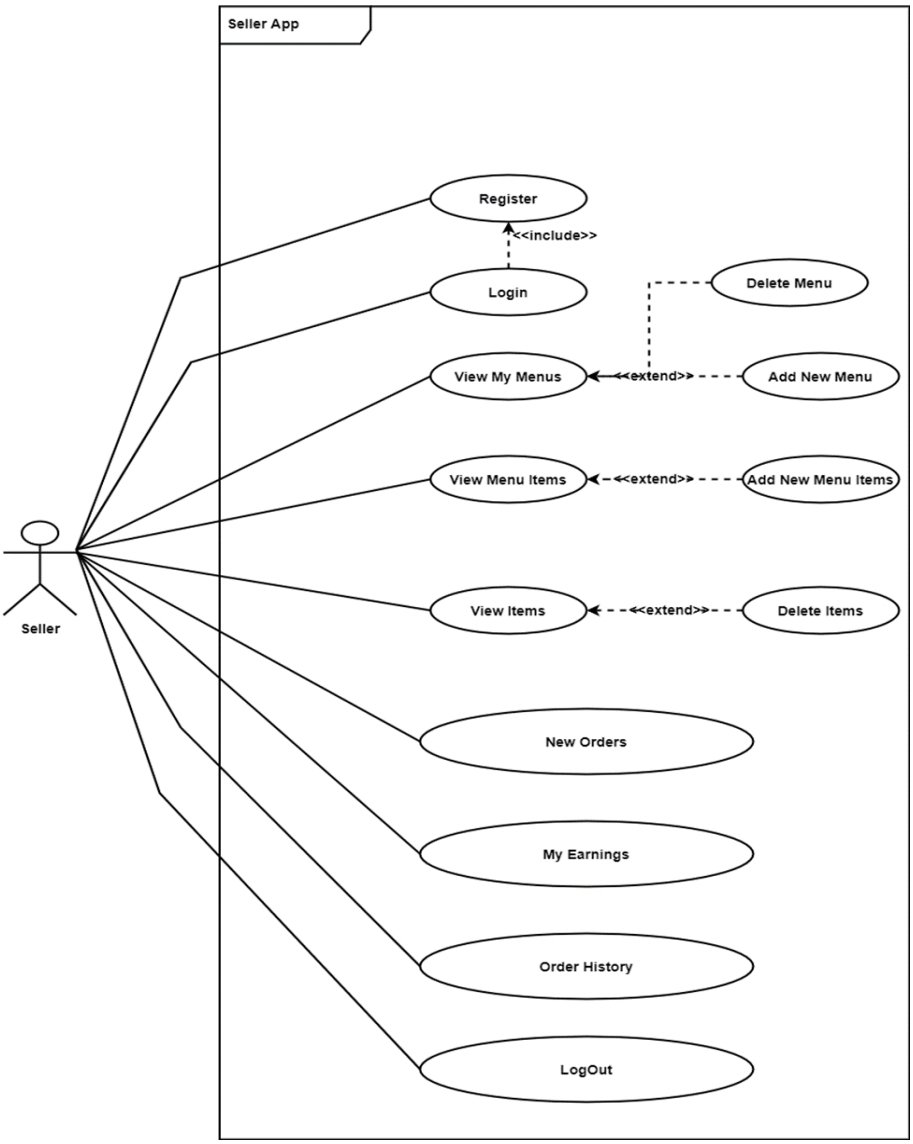
3.2.1: Activity Diagram

3.3 Diagrams

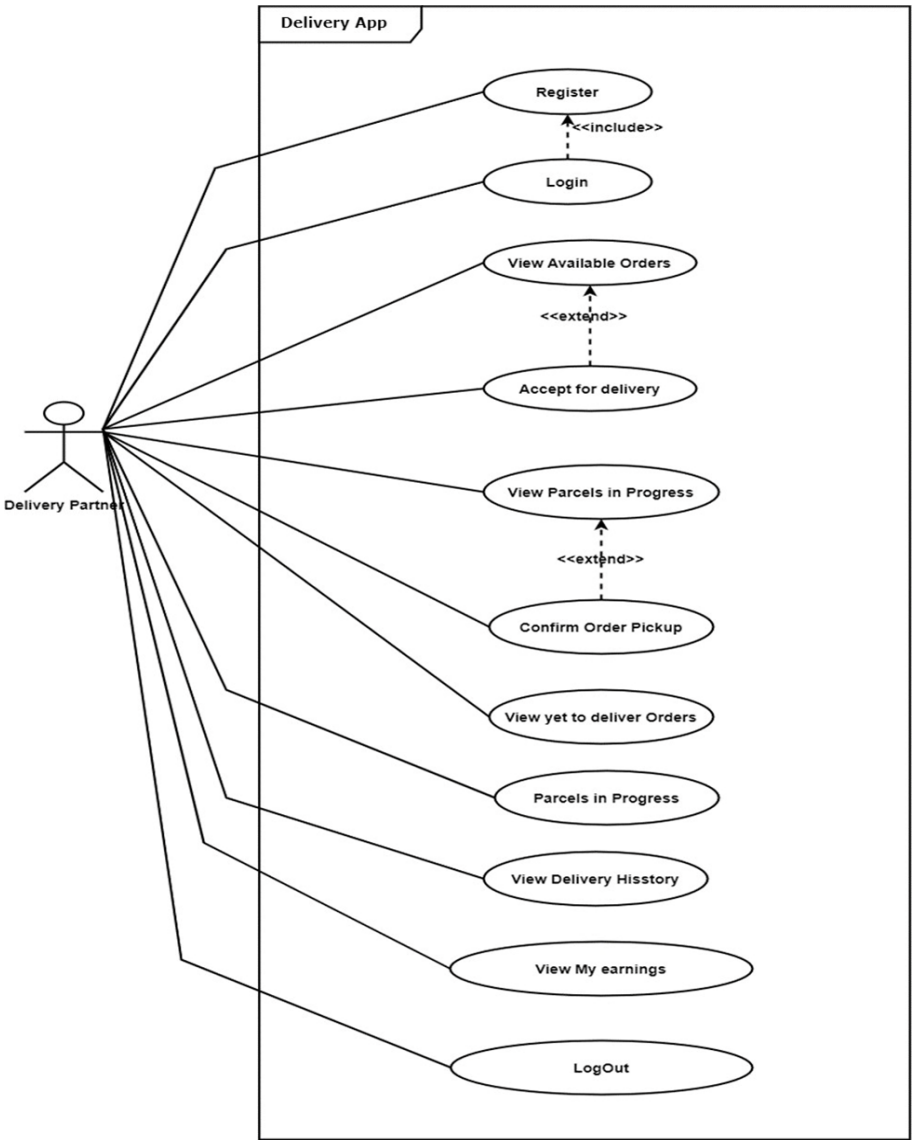
3.3.1 Use case diagram



3.3.1.1: Use case Diagram

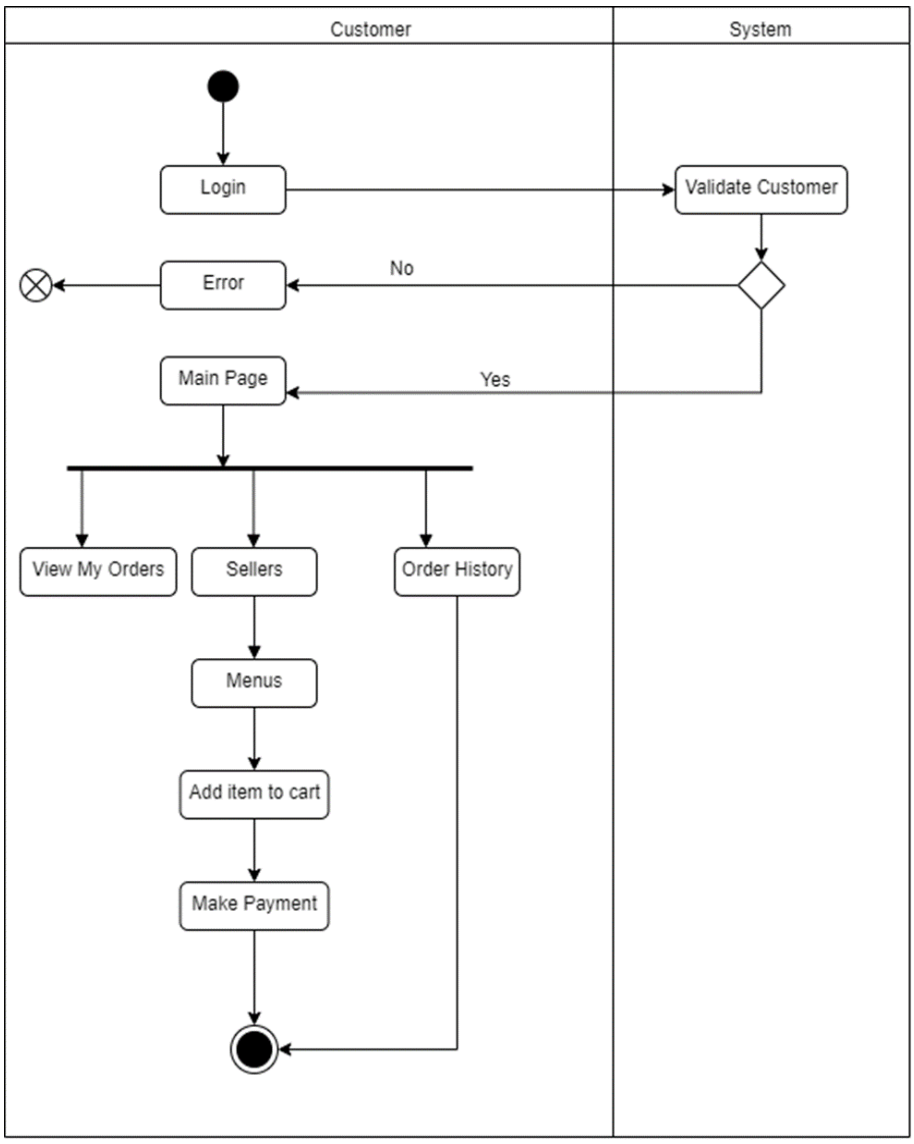


3.3.1.2: Use case Diagram

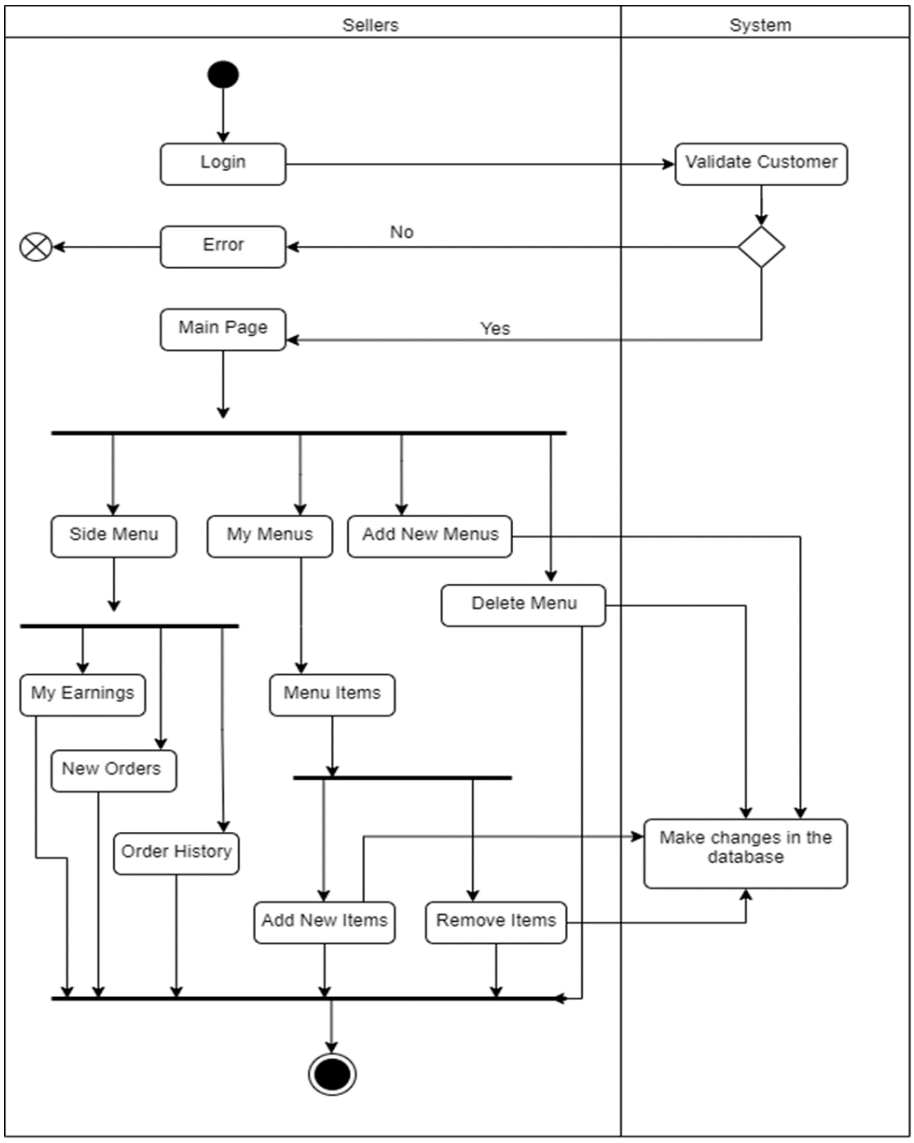


3.3.1.3: Use case Diagram

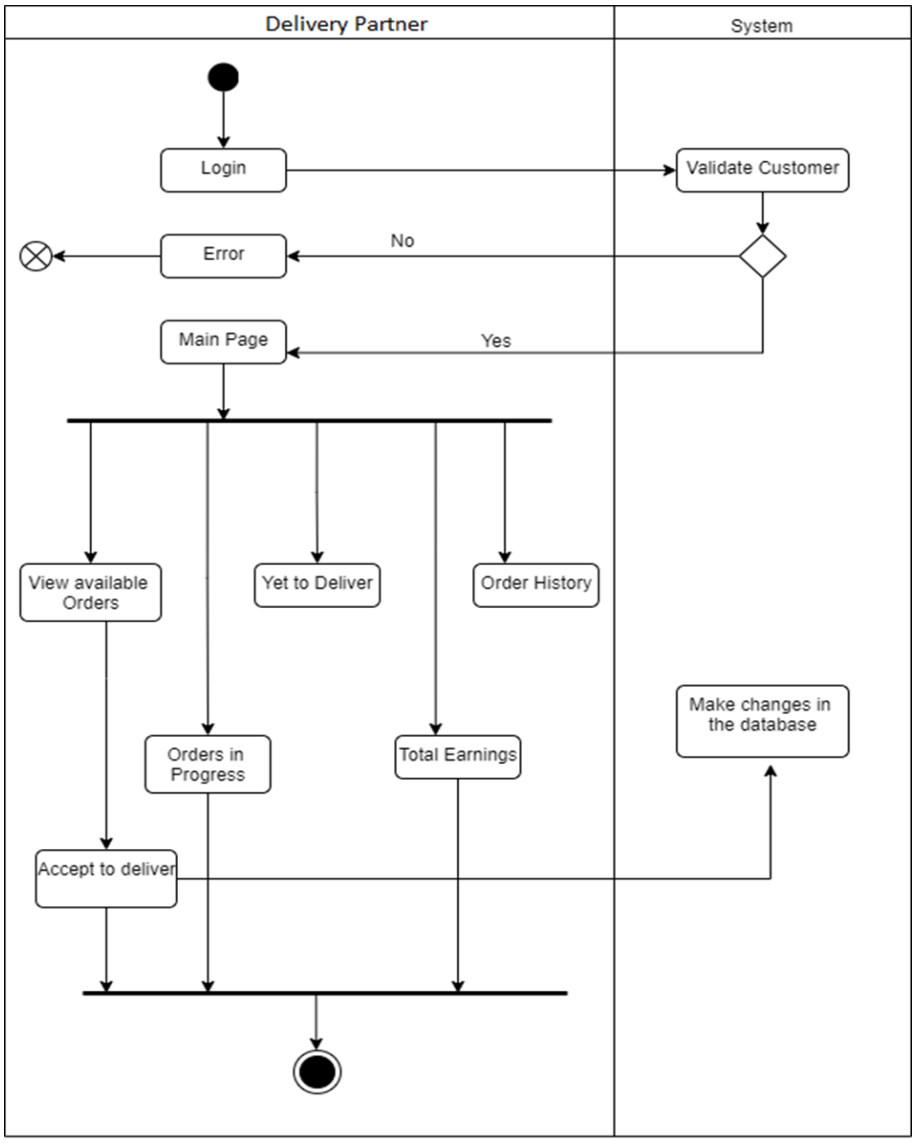
3.3.2 Activity diagram



3.3.2.4: Customer activity

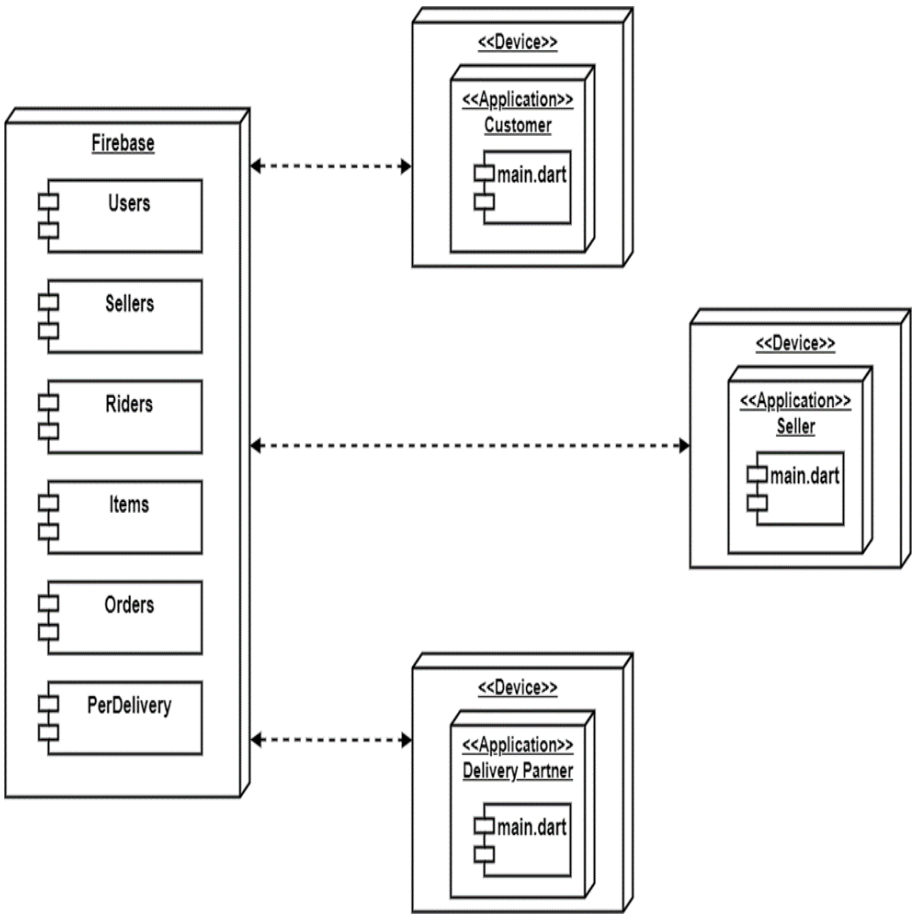


3.3.2.5: Seller activity



3.3.2.6: Delivery Partner activity

3.3.3 Deployment Diagram



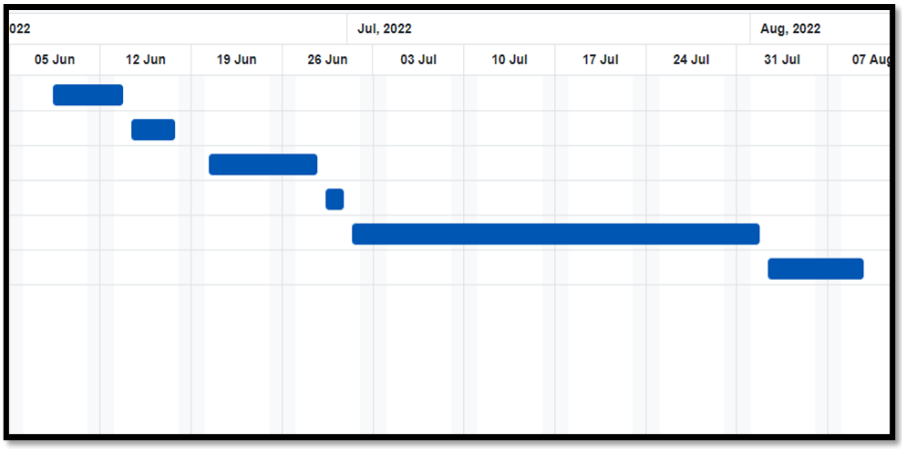
3.3.3.7: Deployment diagram

3.3.4 Work Breakdown Structure

	ID	Name	Start Date	End Date	Duration	Progress %
⠿	1	Requirement gathering	Jun 08, 2022	Jun 13, 2022	4 days	100
⠿	2	Flutter environment setup	Jun 14, 2022	Jun 17, 2022	4 days	100
⠿	3	Work on UI	Jun 20, 2022	Jun 28, 2022	7 days	100
⠿	4	Signup, Signin, Data fetching fro...	Jun 29, 2022	Jun 30, 2022	2 days	100
⠿	5	Core functionalities	Jul 01, 2022	Aug 01, 2022	22 days	100
⠿	6	Testing	Aug 02, 2022	Aug 09, 2022	6 days	100

3.3.4.8: Work breakdown Structure

3.3.5 Gantt Chart

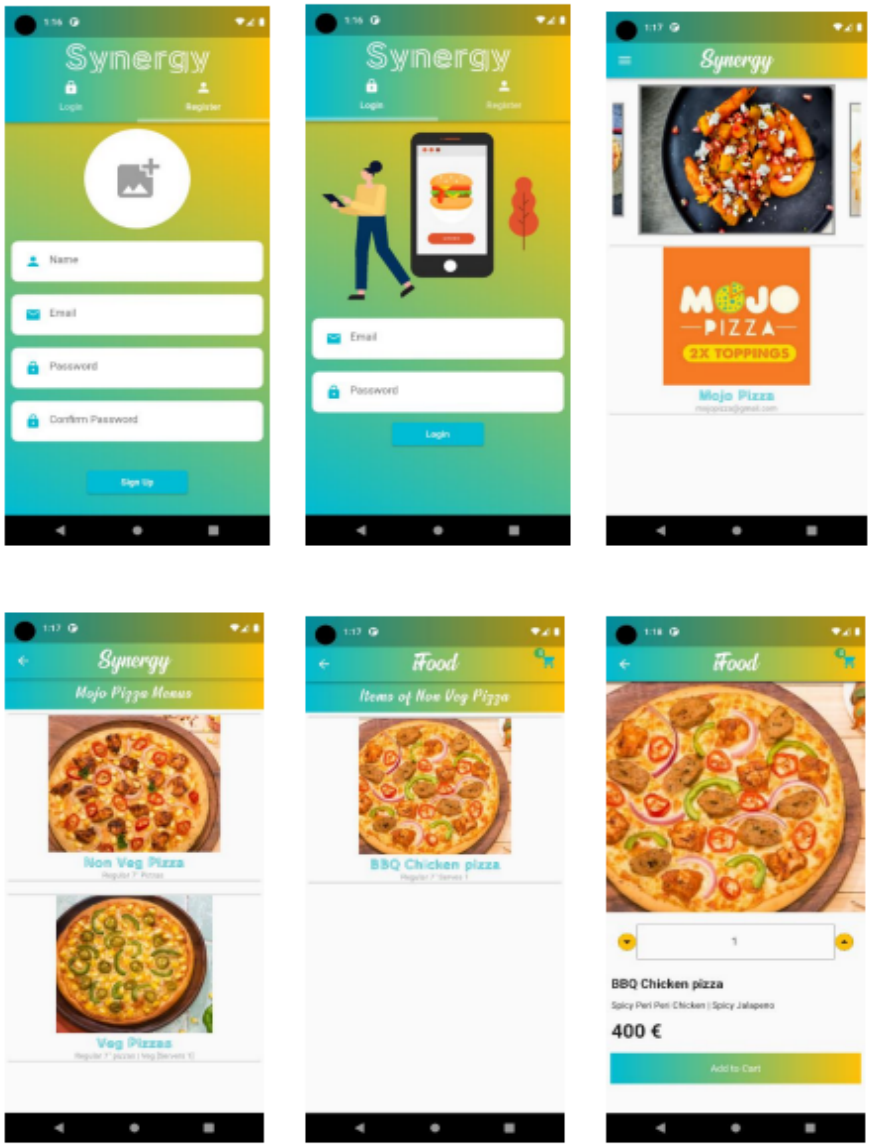


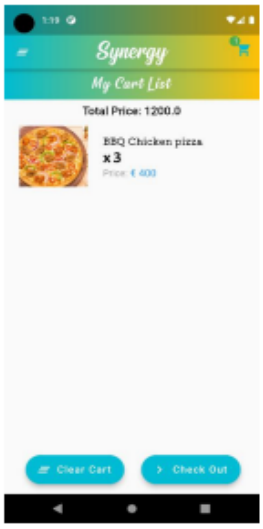
3.3.5.9: Gantt Chart

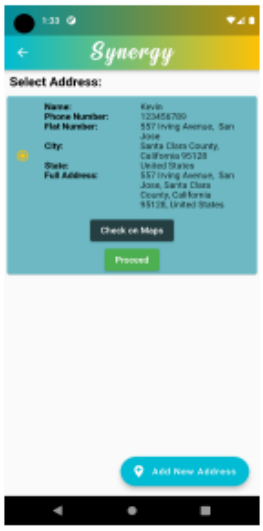
4 Project Implementation and Testing

4.1 UI screenshots

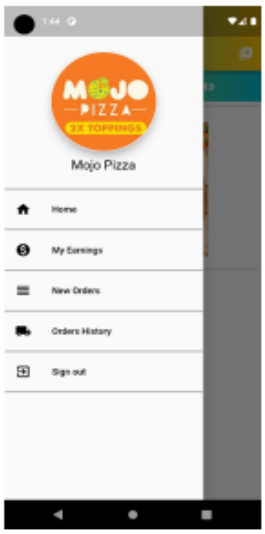
Synergy Customer App –

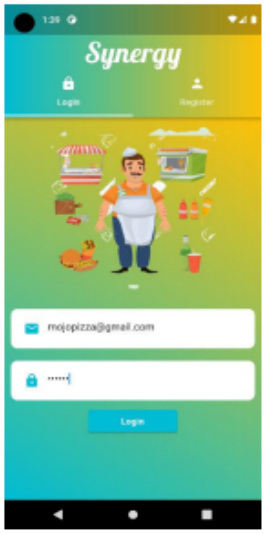


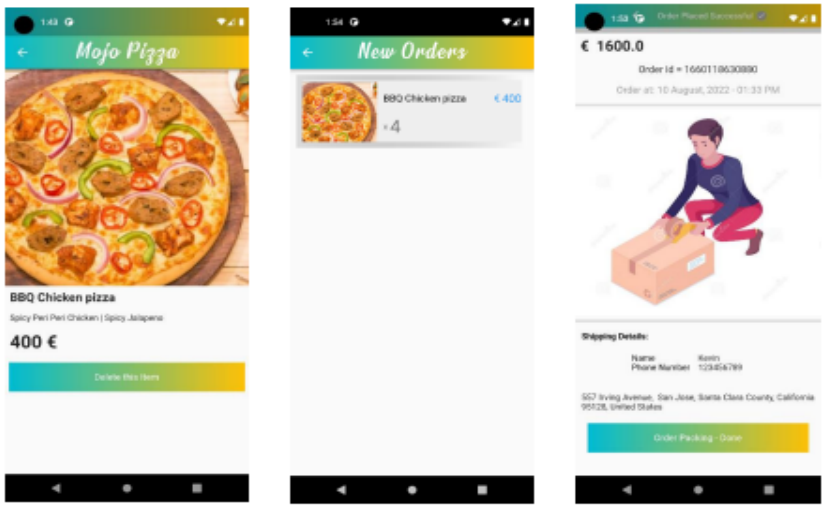




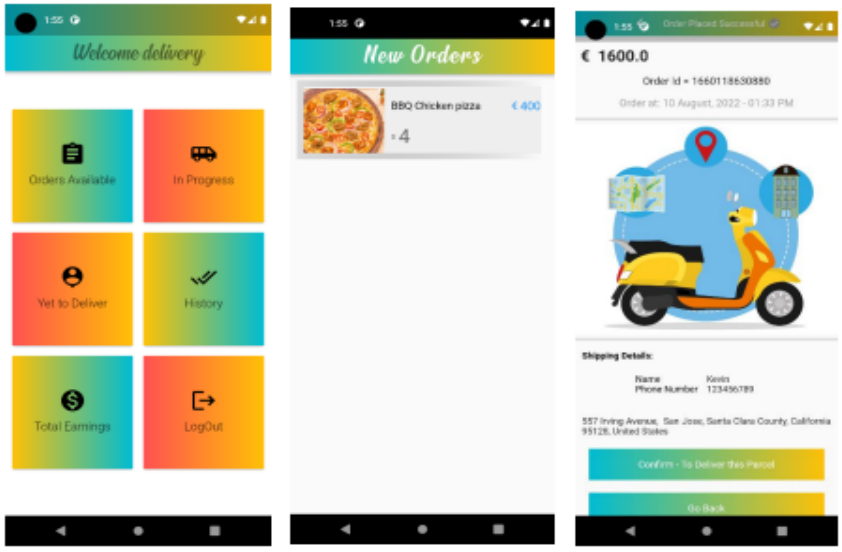
Synergy Seller App –



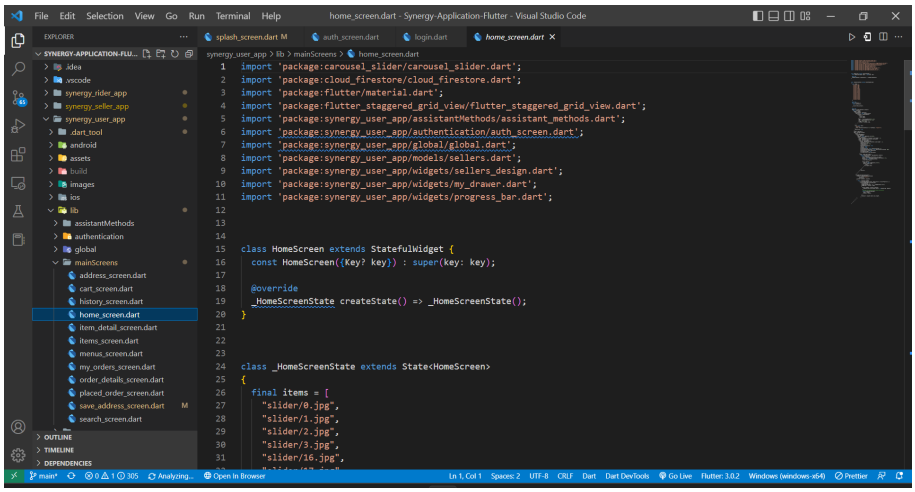




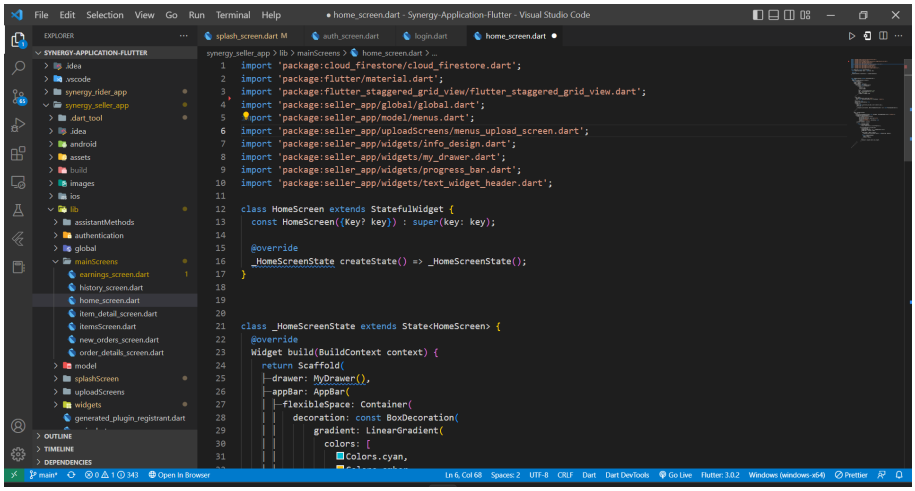
Synergy Delivery Partner app –



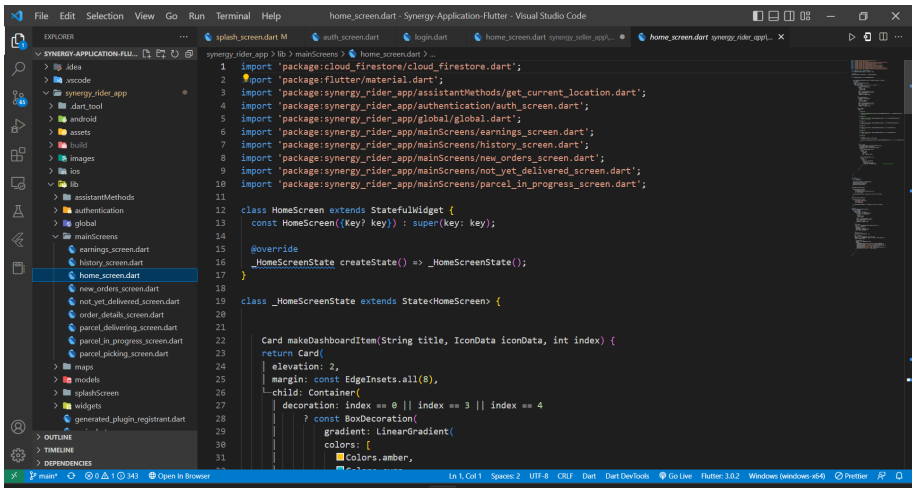
4.2 Code 1



4.3 Code 2



4.4 Code 3


File Edit Selection View Go Run Terminal Help
home_screen.dart - Synergy-Application-Flutter - Visual Studio Code
EXPLORER
SYNERGY-APPLICATION-FLU...
.idea
vscode
synergy_rider_app
.dart_tool
android
assets
build
images
ios
lib
assistantMethods
authentication
global
mainScreens
earnings_screen.dart
history_screen.dart
home_screen.dart
new_orders_screen.dart
not_yet_delivered_screen.dart
order_details_screen.dart
parcel_delivering_screen.dart
parcel_in_progress_screen.dart
parcel_picking_screen.dart
maps
models
splashScreen
widgets
generated_plugin_registrant.dart
OUTLINE
TIMELINE
DEPENDENCIES
splash_screen.dart M
auth_screen.dart
login.dart
home_screen.dart synergy_seller_app\...
home_screen.dart synergy_rider_app\...
synergy_rider_app > lib > mainScreens > home_screen.dart > ...
1 import 'package:cloud_firestore/cloud_firestore.dart';
2 import 'package:flutter/material.dart';
3 import 'package:synergy_rider_app/assistantMethods/get_current_location.dart';
4 import 'package:synergy_rider_app/authentication/auth_screen.dart';
5 import 'package:synergy_rider_app/global/global.dart';
6 import 'package:synergy_rider_app/mainScreens/earnings_screen.dart';
7 import 'package:synergy_rider_app/mainScreens/history_screen.dart';
8 import 'package:synergy_rider_app/mainScreens/new_orders_screen.dart';
9 import 'package:synergy_rider_app/mainScreens/not_yet_delivered_screen.dart';
10 import 'package:synergy_rider_app/mainScreens/parcel_in_progress_screen.dart';
11
12 class HomeScreen extends StatefulWidget {
13 const HomeScreen({Key? key}) : super(key: key);
14
15 @override
16 _HomeScreenState createState() => _HomeScreenState();
17 }
18
19 class _HomeScreenState extends State<HomeScreen> {
20
21
22 Card makeDashboardItem(String title, IconData iconData, int index) {
23 return Card(
24 elevation: 2,
25 margin: const EdgeInsets.all(8),
26 child: Container(
27 decoration: index == 0 || index == 3 || index == 4
28 ? const BoxDecoration(
29 gradient: LinearGradient(
30 colors: [
31 Colors.amber,
main*
Open In Browser
Ln 1, Col 1
Spaces: 2
UTF-8
CRLF
Dart
Dart DevTools
Go Live
Flutter: 3.0.2
Windows (windows-x64)
Prettier

5 Test Cases

Table 6.1: User App

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	Customer Register	Customer enters email id and password	Login successfully	Logged in successfully	Pass
2	Customer Register	Customer enters wrong email id and password	Check your credentials	Login fails. Check credentials	Pass
3	Customer Login	Customer enters right email id and password	Log in successfully	Logged in Successfully	Pass
4	Customer Login	Customer enters wrong email id and password	Check your credentials	Login fails. Check credentials	Pass
5	Add to cart	Customer adds an item to the cart	Item gets added to the cart	Item gets added to the cart	Pass
6	Remove from cart	Customer removes an item from the cart	Item gets removed from the cart	Item gets removed from the cart	Pass
7	View my orders	Customer can view placed orders	Orders are visible	Orders are visible	Pass
8	Checkout	Able to checkout items added to cart	Checked out items are added to cart	Checked out items are added to cart	Pass

6 Test Cases

Table 6.2: Seller App

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	Seller Register	Seller enters email id and password	Login successfully	Logged in successfully	Pass
2	Seller Register	Seller enters wrong email id and password	Check your credentials Login fails.	Check credentials	Pass
3	Seller Login	Seller enters right email id and password	Log in successfully	Logged in Successfully	Pass
4	Seller Login	Seller enters wrong email id and password	Check your credentials Login fails.	Check credentials	Pass
5	Add menu	Seller can add new menu	Menu added successfully	Menu added successfully	Pass
4	Add item	Seller can add new item to the menu	New item added successfully to the menu	New item added successfully to the menu	Pass
7	View earnings	Seller can view his earnings	Total earnings are displayed	Total earnings are displayed	Pass
8	New orders	New orders are displayed	Seller can view new orders	Seller can view new orders	Pass
9	Logout	User Session closes	Session is closed	Session closed successfully	Pass

7 Test Cases

Table 6.3: Delivery App

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	Delivery Agent Register	Delivery Agent enters email id and password	Login successfully	Logged in successfully	Pass
2	Delivery Agent Register	Delivery Agent enters wrong email id and password	Check your credentials	Login fails. Check credentials	Pass
3	Delivery Agent Login	Delivery Agent enters right email id and password	Log in successfully	Logged in Successfully	Pass
4	Delivery Agent Login	Delivery Agent enters wrong email id and password	Check your credentials	Login fails. Check credentials	Pass
5	Available orders	View available orders	All the available orders are displayed	All the available orders are displayed	Pass
6	View and confirm available orders	Delivery agent confirms available order	Delivery agent is able to confirm available orders	Delivery agent is able to confirm available orders	Pass
7	history of completed orders	Delivery agent can view all the completed orders	All the completed orders are displayed	All the completed orders are displayed	Pass
8	View earnings	Total earnings are displayed	Total earnings are displayed	Total earnings are displayed	Pass
9	Logout	User Session closes	Session is closed	Session closed successfully	Pass

8 Limitations

- Internet connection is mandatory.
- Higher memory usage compared to native applications.

9 Future Enhancements

- Admin Portal can be added to have a centralized control over all the three applications.
- Payment Gateway can be added to make secure payments.
- Live location tracking can be added.
- Recommendations based on user activity.
- Push notifications.

10 Conclusion

The system is developed considering all the issues related to the users which are included in this system. Wide range of people can use this if they know how to operate android smart phone. Various issues related to restaurants Service will be solved by providing them a full-fledged system. Thus, implementation of Online Food Ordering system is done to help and solve one of the important problems of people.

It can be concluded: It helps customer in making order easily; It gives information needed in making order to customer.

11 Bibliography

11.1 Web References

[1.] https://docs.flutter.dev/
[2.] https://firebase.google.com/docs
[3.] https://www.youtube.com/user/Firebase
[4.] https://docs.flutter.dev/development/packages-and-plugins/using-packages
[5.] https://www.draw.io/
[6.] https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/
[7.] https://www.geeksforgeeks.org/
[8.] https://stackoverflow.com/