

**BILLY**

A project submitted to

**UKA TARSADIA UNIVERSITY**

in partial fulfillment of the requirements for the degree of

**Master of Science**

in

**Information Technology**

for

**5 Years Integrated M.Sc. (IT)**

By

**CHOKSI ABHISHEK PREMALKUMAR**

(201906100110032)

And

**PATEL MEET DHIRAJBHAI**

(201906100110015)

Guided by

**Ms. Jigna Solanky**  
Assistant Professor



**Babu Madhav Institute of Information Technology  
Uka Tarsadia University  
Bardoli – 394350**

**April 2023**

(201906100110032, 201906100110015)

# CERTIFICATE

This is to certify that **CHOKSI ABHISHEK PREMALKUMAR (201906100110032)** and **PATEL MEET DHIRAJBHAI (201906100110015)** have submitted a project entitled “**BILLY**” as the partial fulfillment for the award of the degree of Master of Science in Information Technology for 5 Years Integrated M.Sc. (IT) in 2023– 2024 .

Date: 29-04-2023

Place: Uka Tarsadia University  
Bardoli – 394350

**Ms. Jigna Solanky**

Guide

**Dr. Jitendra Nasriwala**

Programme Coordinator

External Examiner



**Babu Madhav Institute of Information Technology,**

**Uka Tarsadia University,**

**Bardoli – 394350**

(201906100110032, 201906100110015)

## Acknowledgment

The satisfaction that accompanies the successful completion of this project would be incomplete without mentioning the people who made it possible, without whose constant guidance and encouragement would have made efforts go in vain. I consider myself privileged to express gratitude and respect towards all those who have guided me through the completion of projects.

I convey thanks to my project guide, **Ms. Jigna Solanky**, for his encouragement, constant support, and guidance which was of great help to complete this project work successfully.

I am grateful to **Dr. Jitendra Nasriwala**, Head of the Department, Babu Madhav Institute of Information Technology, **BMIIT** for giving us the support and encouragement that was necessary for the completion of this project.

I would like to thank all the faculty members for their patience, understanding, and guidance that gave me the strength and willpower to work through the long tedious hours of developing a project and preparing the report.

Last but not the least, I would also like to thank my colleagues, who give their ideas and helped me a lot to improve my project.

# Table of Content

<b>Chapters</b>	<b>Particulars</b>	<b>Page No.</b>
<b>1</b>	<b>Introduction</b>	<b>1-3</b>
	1.1 Problem Definition	1
	1.2 Initial Requirement Document	1
	1.3 Project Objective	3
	1.4 Product Scope	3
<b>2</b>	<b>Overall Description</b>	<b>4</b>
	2.1 Product Perspective/ Environment Description	4
	2.1.1 Hardware Interface/ Hardware Specification	4
	2.1.2 Software Interface/ Software Specification	4
<b>3</b>	<b>System Planning</b>	<b>4-5</b>
	3.1 Software Engineering Model	4
<b>4</b>	<b>System Specific Requirements</b>	<b>6-17</b>
	4.1 Functional Requirement	6
	4.2 Non-functional Requirement	17
<b>5</b>	<b>System Analysis</b>	<b>18-25</b>
	5.1 Use case Diagram	18
	5.2 Activity Diagrams	22
	5.3 Sequence Diagrams	25
<b>6</b>	<b>System Design</b>	<b>26-46</b>
	6.1 Database Design	26
	6.1.1 Database Schema	26
	6.1.2 Data Dictionary	30
<b>7</b>	<b>System Implementation</b>	<b>47-88</b>
	7.1 Screenshots	47
<b>8</b>	<b>Testing</b>	<b>89-95</b>
	8.1 Test Cases	89
<b>9</b>	<b>Future Enhancement</b>	<b>96</b>
	<b>Conclusion</b>	<b>96</b>
	<b>Bibliography</b>	<b>96</b>

## List of Figures

<b>Figure no.</b>	<b>Figure Description</b>	<b>Page no.</b>
5.1.1	Use Case Diagram for Billy Admin	18
5.1.2	Use Case Diagram for Billy Partner	19
5.1.3	Use Case Diagram for Billy Customer	20
5.1.4	Use Case Diagram for Billy Delivery Boy	21
5.2.1	Activity Diagram for User Registration	22
5.2.2	Activity Diagram for Menu	22
5.2.3	Activity Diagram for Checkout	23
5.2.4	Activity Diagram for Change Password	24
5.3.1	Sequence Diagram for Login Admin	25
5.3.2	Sequence Diagram for Add Menu	25
5.3.3	Sequence Diagram for Cancel Order by Partner	25

## List of Tables

<b>Table no.</b>	<b>Table Description</b>	<b>Page no.</b>
6.1.2	Data Dictionary	30-46
8.1.1	Test Cases for Login	89-92
8.1.1.1	Authorized Admin	89
8.1.1.2	Authorized Delivery Boy	90
8.1.2	Test Cases for Manage Functionality	92-96
8.1.2.1	Manage Dish	92
8.1.2.2	Manage Delivery Boy	94

# **1. Introduction**

## **1.1 Problem Definition**

In earlier days, all restaurants can use manual ways to store their data and records so maintaining a large amount of data was difficult. They need full function software to maintain their day-to-day life transaction, order, cash transaction, and customer feedback. Because of large data, they need some features or systems which can maintain and store the data or records accurately. The Billy reduces manual work and improves the efficiency of the store. To remove all the disadvantages of conventional methods, a system is proposed which is a **Billy**.

## **1.2 Initial Requirement Document**

### **Initial Requirement Document Billy**

Title of the project	Billy
Stakeholders involved in capturing requirement	Owner, Project Guide, Customer, Partners
Techniques used for requirement capturing	Brainstorming
Name of the person along with a designation	-
Date	July 2022
Users of the system	Admin, Delivery Boy, Customer, Partners, Visitor
Version	3.0

### **Consolidated list of initial requirements:**

Req #1	The system shall be allowed to authenticate to the admin, Partners, Customer, and Delivery Boy to register into the system.
Req #2	The admin shall be able to manage the Cuisines.
Req #3	The admin shall be able to approve or reject the restaurant after checking the restaurant documents.
Req #4	The admin shall be able to approve or reject the menu (category, item, added on, add extra).
Req #5	The admin shall be able to manage the delivery boy.
Req #6	The admin shall be able to manage the Customer.
Req #7	The admin shall be able to manage the Help.
Req #8	The admin shall be able to view all the customer orders.
Req #9	The admin shall be able to manage the questions.
Req #10	The admin shall be able to view the rating of a particular item or a restaurant.
Req #11	The admin shall be able to manage complaints.

Req #12	The admin shall be able to manage the referral amount.
Req #13	The partner shall be able to view orders details like (new orders, preparing orders, ready orders, and past orders)
Req #14	The partner shall be able to the manage category.
Req #15	The partner shall be able to the manage item.
Req #16	The partner shall be able to the manage added-on.
Req #17	The partner shall be able to the manage added extra.
Req #18	The partner shall be able to see their profile and also edit them.
Req #19	The partner shall be able to see menu history like (in-review, approved, canceled)
Req #20	The partner shall be able to manage the banner.
Req #21	The partner shall be able to manage the setting.
Req #22	The partner shall be able to communicate with Billy's admin using Help.
Req #23	The partner shall be able to see all the customer reviews.
Req #24	The partner shall be able to set the order status and also allocate the delivery boy.
Req #25	The partner shall be able to View the matric(report).
Req #26	The customer shall be able to view the cuisines and select the cuisines on the home page.
Req #27	The customer shall be able to view the latest offers banner and also view the top 10 products on the home page.
Req #28	The customer shall be able to view the cuisines name, Description, and cuisines banner on the Restaurant page.
Req #29	The customer shall be able to view all the restaurants based on the cuisines selected by the customer.
Req #30	The customer shall be able to filter the restaurant rating-wise.
Req #31	The customer shall be able to view items based on the category on the menu page.
Req #32	The customer shall be able to view and search (filter) the item price-wise, rating wise and item-type-wise (Veg/Non-Veg/Egg).
Req #33	The customer shall be able to customize the menu, like (adding or removing added-on, add extra, and variants).
Req #34	The customer shall be able to manage the cart by adding the item in the cart, updating the quantity, or removing the item and customizing (added-on, add extra, and variant).
Req #35	At checkout time customer change their shipping address and also add a new shipping address.
Req #36	The customer shall be able to make payment using one of the online payment gateways, through cash on delivery, or using the wallet.
Req #37	The system shall be able to generate the bill and the customer also downloads the bill in pdf format.
Req #38	The customer shall be able to cancel the order if the order status is pending.
Req #39	The customer shall be able to view the order status. (Like pending, cooking, on the way, delivered)

Req #40	The customer shall be able to get the mail for their order.
Req #41	The customer shall be able to view previous(past) order history and also download the pdf of that.
Req #42	The customer shall be able to give feedback.
Req #43	The customer also adds an item to the wish list and later they can view their like item and order it.
Req #44	The customer shall be able to edit their account.
Req #45	The customer shall be able to change their password and set a new password.
Req #46	The customer also has forgotten the password facility.
Req #47	The delivery boy shall be able to view the order detail and which order allocate to that delivery boy.
Req #48	The delivery boy shall be able to update the payment status if the payment is pending.
Req #49	The delivery boy shall be able to update the order status and set the order status delivered.
Req #50	<p>The system should be able to generate reports like:</p> <ol style="list-style-type: none"> <li>1. Categories of complaints             <ol style="list-style-type: none"> <li>a. Wrong Item</li> <li>b. Missing Item</li> <li>c. Quality &amp; Quantity Issues</li> <li>d. Package Spillage</li> <li>e. Late Delivery</li> </ol> </li> <li>2. Accepted Orders</li> <li>3. Canceled Orders</li> </ol> <p>Status of all customer's orders (Day/Date, Weekly, Monthly, Yearly)</p>

### 1.3 Project Objective

- To reduce the cost of the business operation.
- To improve the customer service.
- To simplify the workflow.
- To save the environment.
- This will be adding on benefit for the customers as it will save their time, plus it adds on for the owner also.

### 1.4 Product Scope

The user first registration into the system and then the customer can buy food from the billy. The system will deliver food only to Bardoli city. The scope of this system is confined to one city and involves multiple customers and partners. This system is a multi-restaurants system, allowing customers to buy and browse items from various restaurants.

## **2. Overall Description**

### **2.1 Product Perspective/Environment Description**

#### **2.1.1 Hardware Interface/Hardware Specification**

CPU:	Intel Core Pentium
HDD Space:	120 GB HDD
RAM:	4 GB RAM

#### **2.1.2 Software Interface/Software Specification**

Operating System:	Multi-Platform (Window, IOS, Linux, Web Server)
Front-end Technology:	React JS, Flutter, HTML, CSS, Bootstrap, JavaScript, jQuery
Back-end Technology:	Node JS, Express JS, MongoDB
Other Tool:	Visual Studio Code, MongoDB Compass, Postman, Android Studio, Android Virtual Device, XCode, IOS Simulator

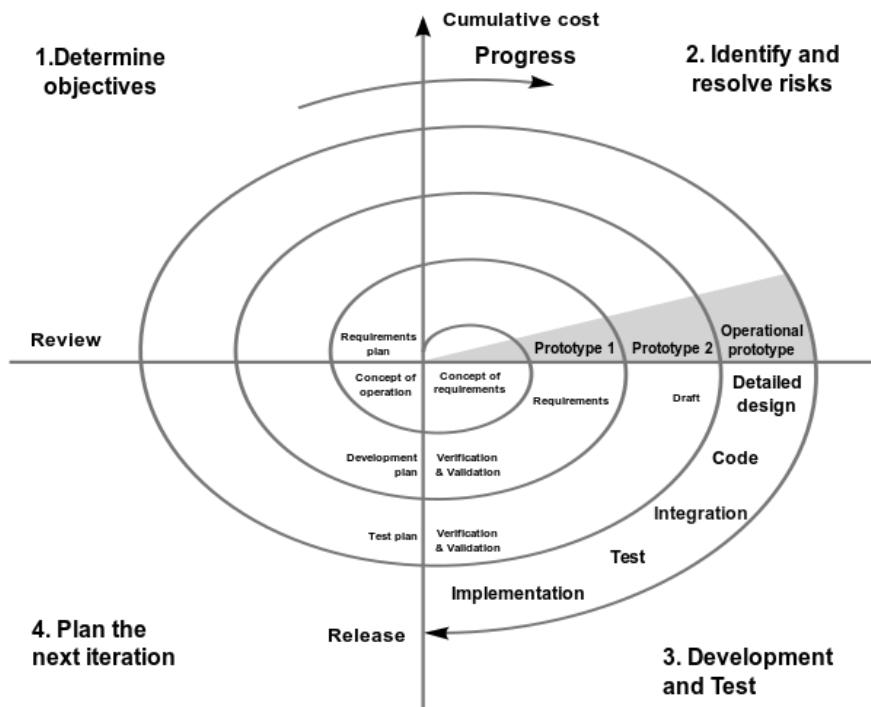
## **3. System Planning**

### **3.1 Software Engineering Model**

This system follows **Spiral Model**.

- Why do we use Spiral Model?**

- An evolutionary software process model that couples the iterative nature of prototyping with the controlled and systematic aspects of the Waterfall Model.
- Provides the potential for rapid development of an increasingly more complex version of the Software.
- Using this, Software is developed in a series of evolutionary releases.
- During early iteration, the release may be a paper model or prototype.
- During later iterations, an increasingly more complete version of the engineered system is produced.
- Each of the framework activities represents one segment of the Spiral Model.
- The spiral work in a clockwise direction beginning at the center.



[Figure: Spiral Model]

### Advantage of Spiral Model:

The advantage of using the spiral model is varied. Its design flexibility allows changes to be implemented at several stages of the project.

Provides a working model to the user early in the process, enabling early assessment and increasing the user's confidence.

The developer gains experience and insight by developing prototyping thereby resulting in better implementation of requirements.

## **4. System Specific Requirements**

### **4.1 Functional Requirement**

#### 4.1.1. Functional requirement for Manage User login

RN	Description	Comments
<b>FR1</b>	This Module will be managed by Admin, Partner, Delivery Boy, and Customer.	User Login Page
<b>FR2</b>	This Module accepts the password which will be created by the user.	User Login Page
<b>FR3</b>	This Module also provides the functionality of forgot passwords for changing user passwords.	User Login Page

#### 4.1.2. Functional requirement for Manage Cuisines

RN	Description	Comments
<b>FR1</b>	This Module will be managed by Admin.	Cuisines Page
<b>FR2</b>	The system will assign a unique id (Cuisines ID) to each Cuisines.	Add Cuisines Page
<b>FR3</b>	Whenever any new Cuisines arrives, the admin is allowed to record the following details of the Cuisines Like ... Cuisines name, Cuisines image, Cuisines Description and Cuisines Banner.	Add Cuisines Page
<b>FR4</b>	To manage this module, the admin will be able to perform the following operations: ➤ Add new Cuisines ➤ Update an existing Cuisines	Add Cuisines Page Update Cuisines Page
<b>FR5</b>	To manage this module, the admin will be able to perform the following operations: ➤ Active/De-active Cuisines ➤ View/Search Cuisines	View/Search and Active/Deactivate Cuisines Page

#### 4.1.3. Functional Requirement for Manage Restaurant

RN	Description	Comments
<b>FR1</b>	This Module will be managed by Partner.	Restaurant Page
<b>FR2</b>	The system will assign a unique id (Restaurant ID) to each Restaurant.	Add Restaurant Page

<b>FR3</b>	Whenever any new Item arrives, the Partner is allowed to record the following details of the Restaurant Like ... Restaurant name, Restaurant image, Restaurant Address, Restaurant City, Restaurant Contact, Owner name, Owner Contact, owner Email Id, Owner Password.	Add Restaurant Page
<b>FR4</b>	After restaurant verified by admin, the Partner is allowed to record the following details of the Restaurant Like ... Documents (GST Certificate, FSSAI Certificate, Sample Bill, Sample Menu, Owner Pan card), Restaurant timing, cuisines id.	Restaurant Registration Page
<b>FR4</b>	To manage this module, the Partner will be able to perform the following operations: ➤ Add new Item ➤ Update an existing Item	Add Restaurant Page  Update Restaurant Page
<b>FR5</b>	To manage this module, the Partner will be able to perform the following operations: ➤ Active/De-active Item ➤ View/Search Item	View Search Active/Deactivate Restaurant Page

#### 4.1.4. Function requirement for Manage Category

RN	Description	Comments
<b>FR1</b>	This Module will be managed by Partner.	Category Page
<b>FR2</b>	The system will assign a unique id (category ID) to each category.	Add category Page
<b>FR3</b>	Whenever any new Category arrives, the Partner is allowed to record the following details of the Category Like ... Category name.	Add category Page
<b>FR4</b>	To manage this module, the Partner will be able to perform the following operations: ➤ Add new Category ➤ Update an existing Category ➤ Delete an existing Category	Add category Page  Update category Page  Delete category Page
<b>FR5</b>	To manage this module, the Partner will be able to perform the following operations: ➤ Active/De-active Category ➤ View/Search Category	View/Search and Active/Deactivate category Page

#### 4.1.5. Functional Requirement for Manage Added On

RN	Description	Comments
<b>FR1</b>	This Module will be managed by Partner.	Added-on Page
<b>FR2</b>	The system will assign a unique id (added-on id) to each added-on.	Add      Added-on Page
<b>FR3</b>	Whenever any new added-on arrives, the partner is allowed to record the following details of the added-on Like ...  Added-on id with its restaurant id, Name, Type, Price, Additional Price, Final Price	Add      Added-on Page
<b>FR4</b>	To manage this module, the Partner will be able to perform the following operations: ➤ Add new Added-on ➤ Update an existing Added-on	Add      Added-on Page  Update    Added-on Page
<b>FR4</b>	To manage this module, the Partner will be able to perform the following operations: ➤ Active/De-active Added-on ➤ View/Search Added-on ➤ Delete Added-on	View/Search   and Active/Deactivate Added-on Page  Delete    Added-on Page
<b>FR5</b>	The customer will be able to: ➤ View Added On	Customize   Item Page

#### 4.1.6. Functional Requirement for Manage Add Extra

RN	Description	Comments
<b>FR1</b>	This Module will be managed by Partner.	Add-Extra Page
<b>FR2</b>	The system will assign a unique id (add-extra id) to each add-extra.	Add      Add-Extra Page
<b>FR3</b>	Whenever any new add-extra arrives, the partner is allowed to record the following details of the add-extra Like ...	Add      Add-Extra Page

	Add extra id with its restaurant id, Name, Type, Price, Additional Price, Final Price	
<b>FR4</b>	To manage this module, the Partner will be able to perform the following operations: ➤ Add new Add-Extra ➤ Update an existing Add-Extra	Add Page  Update Add-Extra Page
<b>FR4</b>	To manage this module, the Partner will be able to perform the following operations: ➤ Active/De-active Add-Extra ➤ View/Search Add-Extra ➤ Delete Add-Extra	View/Search and Active/Deactivate Add-Extra Page  Delete Add-Extra Page
<b>FR5</b>	The customer will be able to: ➤ View Add-Extra	Customize Item Page

#### 4.1.7. Functional Requirement for Manage Item

RN	Description	Comments
<b>FR1</b>	This Module will be managed by Partner.	Item Page
<b>FR2</b>	The system will assign a unique id (Item ID) to each item.	Add Item Page
<b>FR3</b>	Whenever any new Item arrives, the Partner is allowed to record the following details of the Item Like ...  Item id with its Category id, Item name, Item type, Item description, Item image, Add extra, Added on.	Add Item Page
<b>FR4</b>	To manage this module, the Partner will be able to perform the following operations: ➤ Add new Item ➤ Update an existing Item	Add Item Page  Update Item Page
<b>FR5</b>	To manage this module, the Partner will be able to perform the following operations: ➤ Active/De-active Item ➤ View/Search Item	View/Search and Active/Deactivate Item Page
<b>FR6</b>	The customer will be able to: ➤ View/Search Item (Filter wise)	Customer View Item Page

#### 4.1.8. Functional Requirement for Manage Variant

RN	Description	Comments
<b>FR1</b>	This Module will be managed by Partner.	Variant Page
<b>FR2</b>	The system will assign a unique id (variant id) to each variant.	Add Variant Page
<b>FR3</b>	Whenever any new variant arrives, the partner is allowed to record the following details of the variant Like ...  Variant id with its Item id, Name, UOM, Price, Sales Price	Add Variant Page
<b>FR4</b>	To manage this module, the Partner will be able to perform the following operations: ➤ Add new Variant ➤ Update an existing Variant	Add Variant Page  Update Variant Page
<b>FR4</b>	To manage this module, the Partner will be able to perform the following operations: ➤ Active/De-active Variant ➤ View/Search Variant ➤ Delete Variant	View/Search and Active/Deactivate Variant Page  Delete Variant Page
<b>FR5</b>	The customer will be able to: ➤ View Variant	Customize Item Page

#### 4.1.9. Functional requirement for Manage Banner

RN	Description	Comments
<b>FR1</b>	This Module will be managed by the Partner.	Banner Page
<b>FR2</b>	The system will assign a unique id (Banner ID) to each Banner.	Add Banner Page
<b>FR3</b>	Whenever any new banner arrives, the Partner is allowed to record the following details of the Banner Like ...  Banner image, Banner Description	Add Banner Page
<b>FR4</b>	To manage this module, the partner will be able to perform the following operations Like: ➤ Add new banner ➤ Update an existing Banner	Add Banner Page  Update Banner Page
<b>FR4</b>	To manage this module, the partner will be able to perform the following operations Like: ➤ Active/De-active Banner ➤ Delete Banner	Active/Deactivate Banner Page

		Delete Banner Page
<b>FR5</b>	The customer and Partner will be able to: ➤ View Banner	Home Page Banner Page

#### 4.1.10. Functional requirement for Manage Customer

RN	Description	Comments
<b>FR1</b>	This Module will be managed by Admin and the customer.	Customer Page
<b>FR2</b>	The system will assign a unique id (Customer ID) to each Customer.	Customer Registration Page
<b>FR3</b>	Whenever a new Customer Register then the system records the following details of the customer Like ... Customer Name, Email id, password, mobile No, Billing Address	Customer Registration Page
<b>FR4</b>	To manage this module, the admin will be able to perform the following operations: ➤ Active/De-active Customer ➤ View/Search Customer	View Search Active/Deactivate Customer Page
<b>FR5</b>	The customer will be able to: ➤ Registration ➤ Update an existing Account ➤ Change password	Customer Registration Page Customer account Page

#### 4.1.11. Functional Requirement for Manage Wishlist

RN	Description	Comments
<b>FR1</b>	This Module will be managed by customer.	Wishlist Page
<b>FR2</b>	The system will assign a unique id (wish list ID) to each wish list customer id and item id.	Add Wishlist Page
<b>FR3</b>	The customer can add there like item in wish list.	Add Wishlist Page
<b>FR4</b>	To manage this module, the customer will be able to perform the following operations: ➤ Add new wish list ➤ View/Search wish list ➤ Delete wish list	Add Wishlist Page View/Search Wishlist Page Delete Wishlist Page

#### 4.1.12. Functional requirement for Manage Cart

RN	Description	Comments
<b>FR1</b>	This Module will be managed by the Customer.	Cart Page
<b>FR2</b>	The system will assign a unique id (Cart ID) to each Cart.	Cart Page
<b>FR3</b>	Whenever the Customer adds Item to the cart then the system records the following details of the Cart Like ...  Cart Id with its customer Id, Item Id, Variant Id and Qty.	Cart Page
<b>FR4</b>	To manage this module, the Customer will be able to perform the following operations Like:  ➤ Add new menu ➤ Update an existing Cart ➤ Empty Cart	Cart Page

#### 4.1.13. Functional Requirement for Billing Address management

RN	Description	Comments
<b>FR1</b>	This Module will be managed by the customer.	Billing Address Page
<b>FR2</b>	The system will assign a unique id (Billing Address ID) to each Billing Address.	Add Billing Address Page
<b>FR3</b>	Whenever any new Billing address arrives, the customer is allowed to record the following details of the Billing address Like ...  Title, Address, Country, State, City, Pincode.	Add Billing Address Page
<b>FR4</b>	To manage this module, the customer will be able to perform the following operations Like:  ➤ Add new Billing Address ➤ Update an existing Billing Address	Add Billing Address Page  Update Billing Address Page

#### 4.1.14. Functional requirement for Manage Customer Order Management

RN	Description	Comments
<b>FR1</b>	This Module will be managed by the Customer, Admin, Partner.	Order Page
<b>FR2</b>	The system will assign a unique id (Order ID) to each Order.	Order Page
<b>FR3</b>	The System will generate a bill for each Customer order.	Order Page

<b>FR4</b>	To manage this module, the Partner will be able to perform the following operations: ➤ Update Order Status ➤ View/Search Order ➤ Allocate Delivery Boy ➤ Download Invoice	Order Invoice Page View Order Page
<b>FR5</b>	To manage this module, the admin will be able to perform the following operations: ➤ View/Search Order ➤ Download Invoice	Order Page
<b>FR6</b>	The customer will be able to: ➤ View Order ➤ Download Invoice	Customer View Order Page

#### 4.1.15. Function Requirement for Manage Delivery Boy

RN	Description	Comments
<b>FR1</b>	This Module will be managed by Admin.	Delivery Boy Page
<b>FR2</b>	The system will assign a unique id (Delivery Boy ID) to each Delivery Boy.	Add Delivery Boy Page
<b>FR3</b>	Whenever any new Delivery Boy arrives, the admin is allowed to record the following details of the Delivery Boy Like...  Delivery Boy name, Mobile No, Password.	Add Delivery Boy Page
<b>FR4</b>	To manage this module, the admin will be able to perform the following operations:  ➤ Add new Delivery Boy ➤ Update an existing Delivery Boy	Add Delivery Boy Page  Update Delivery Boy Page
<b>FR5</b>	To manage this module, the admin will be able to perform the following operations:  ➤ Active/De-active Delivery Boy ➤ View/Search Delivery Boy	View/Search and Active/Deactivate Delivery Boy Page

#### 4.1.16. Functional Requirement for Report management

RN	Description	Comments
<b>FR1</b>	This Module will be managed by Admin and Partner.	Report Page
<b>FR2</b>	The System will be generating the report.	Report Page

<b>FR3</b>	The Partner will be able to: 1. Accepted Orders 2. Cancelled Orders  Status of all customer's orders (Day/Date, Weekly, Monthly, Yearly)	Report Page
<b>FR4</b>	The admin will be able to: 1. Categories of complaints a. Wrong Item b. Missing Item c. Quality & Quantity Issues d. Package Spillage e. Late Delivery 2. Accepted Orders 3. Cancelled Orders  Status of all customer's orders (Day/Date, Weekly, Monthly, Yearly)	Report Page

#### 4.1.17. Functional Requirement for Manage Get Touch (Help)

RN	Description	Comments
<b>FR1</b>	This Module will be managed by Partner.	Get Touch Page
<b>FR2</b>	The system will assign a unique id (Get Touch ID) to each Get Touch.	Add Get Touch Page
<b>FR3</b>	Whenever the partner Add get touch, the partner is allowed to record the following details of the Setting Like... Partner id with Subject, Message.	Add Get Touch Page
<b>FR4</b>	To manage this module, the Partner will be able to perform the following operations: ➤ Add new Get Touch	Add Get Touch Page
<b>FR5</b>	To manage this module, the admin will be able to perform the following operations: ➤ View Get Touch	View/Search Get Touch Page

#### 4.1.18. Function Requirement for Manage Setting (Partner Setting)

RN	Description	Comments
<b>FR1</b>	This Module will be managed by Partner.	Setting Page
<b>FR2</b>	The system will assign a unique id (Setting ID) to each Setting.	Add Setting Page

<b>FR3</b>	Whenever the partner updates the setting, the partner is allowed to record the following details of the Setting Like...  Cart Min Price, Cart Min Price Message, Delivery charge and GST.	Add Setting Page
<b>FR4</b>	To manage this module, the partner will be able to perform the following operations:  ➤ Add new Setting ➤ Update an existing Setting	Add Setting Page  Update Setting Page

#### 4.1.19. Functional Requirement for Manage Question

RN	Description	Comments
<b>FR1</b>	This Module will be managed by admin.	Question Page
<b>FR2</b>	The system will assign a unique id (question id) to each question.	Add Question Page
<b>FR3</b>	Whenever any new question arrives, the admin is allowed to record the following details of the question Like ...  Questions	Add Question Page
<b>FR4</b>	To manage this module, the admin will be able to perform the following operations:  ➤ Add new Question ➤ Update an existing Question	Add Question Page  Update Question Page
<b>FR4</b>	To manage this module, the admin will be able to perform the following operations:  ➤ Active/De-active question ➤ View/Search question	View/Search and Active/Deactivate Question Page

#### 4.1.20. Functional Requirement for Manage Complain

RN	Description	Comments
<b>FR1</b>	This Module will be managed by Customer.	Complain page
<b>FR2</b>	The system will assign a unique id (complain id) to each complain.	Add Complain Page

<b>FR3</b>	Whenever any new complain arrives, the Customer is allowed to record the following details of the complain Like ...  Complain id with its User id, Question Id and Message,	Add Complain Page
<b>FR4</b>	To manage this module, the customer will be able to perform the following operations: ➤ Add new Complain	Add Complain Page
<b>FR4</b>	To manage this module, the admin will be able to perform the following operations: ➤ View/Search Complain	View/Search Complain Page

#### 4.1.21. Functional Requirement for Manage Referral Amount

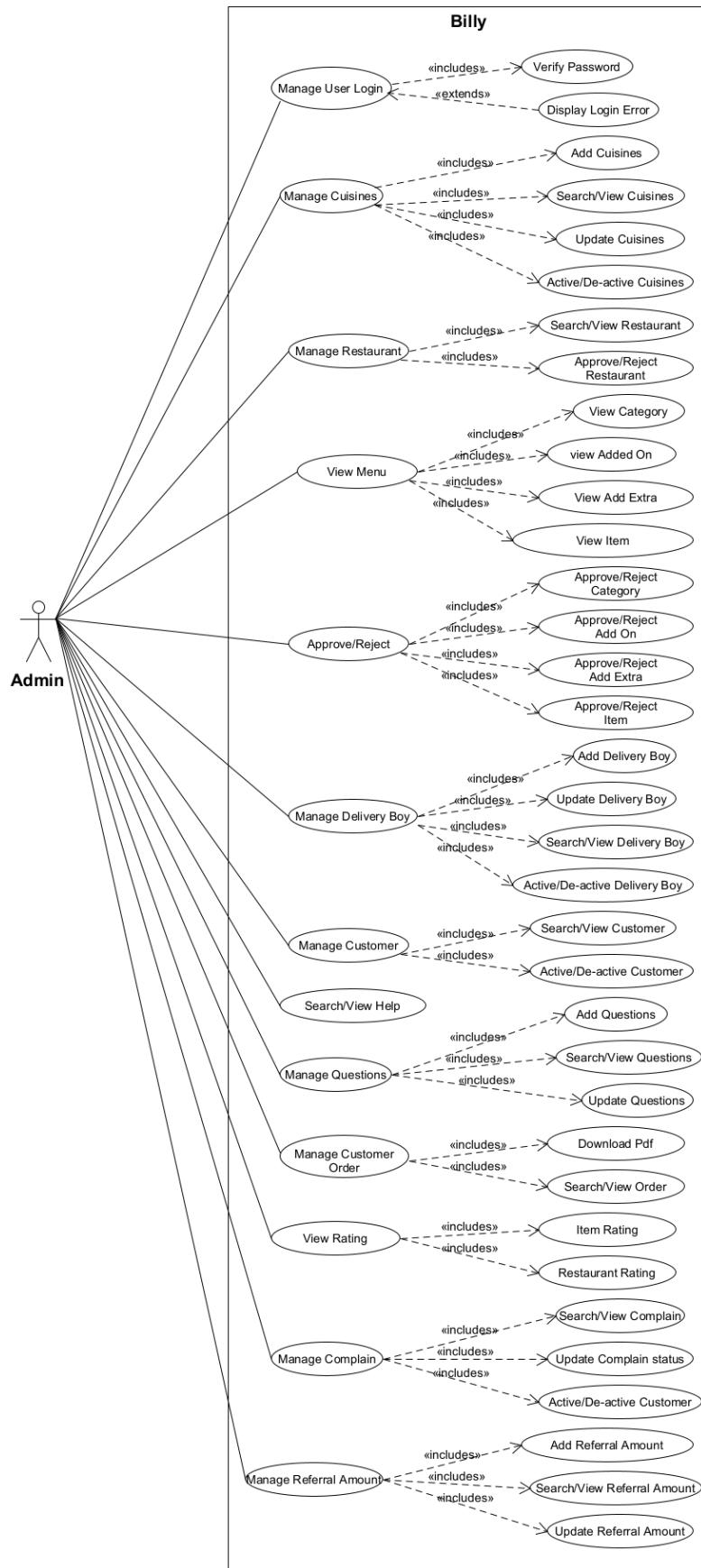
RN	Description	Comments
<b>FR1</b>	This Module will be managed by Admin.	Referral Page
<b>FR2</b>	The system will assign a unique id (Referral ID) to each Setting.	Add Referral Page
<b>FR3</b>	Whenever the admin updates the Referral, the admin is allowed to record the following details of the Referral Like...  Referral Amount	Add Referral Page
<b>FR4</b>	To manage this module, the partner will be able to perform the following operations: ➤ Add new Referral ➤ Update an existing Referral	Add Referral Page Update Referral Page

## **4.2 Non-Functional Requirement**

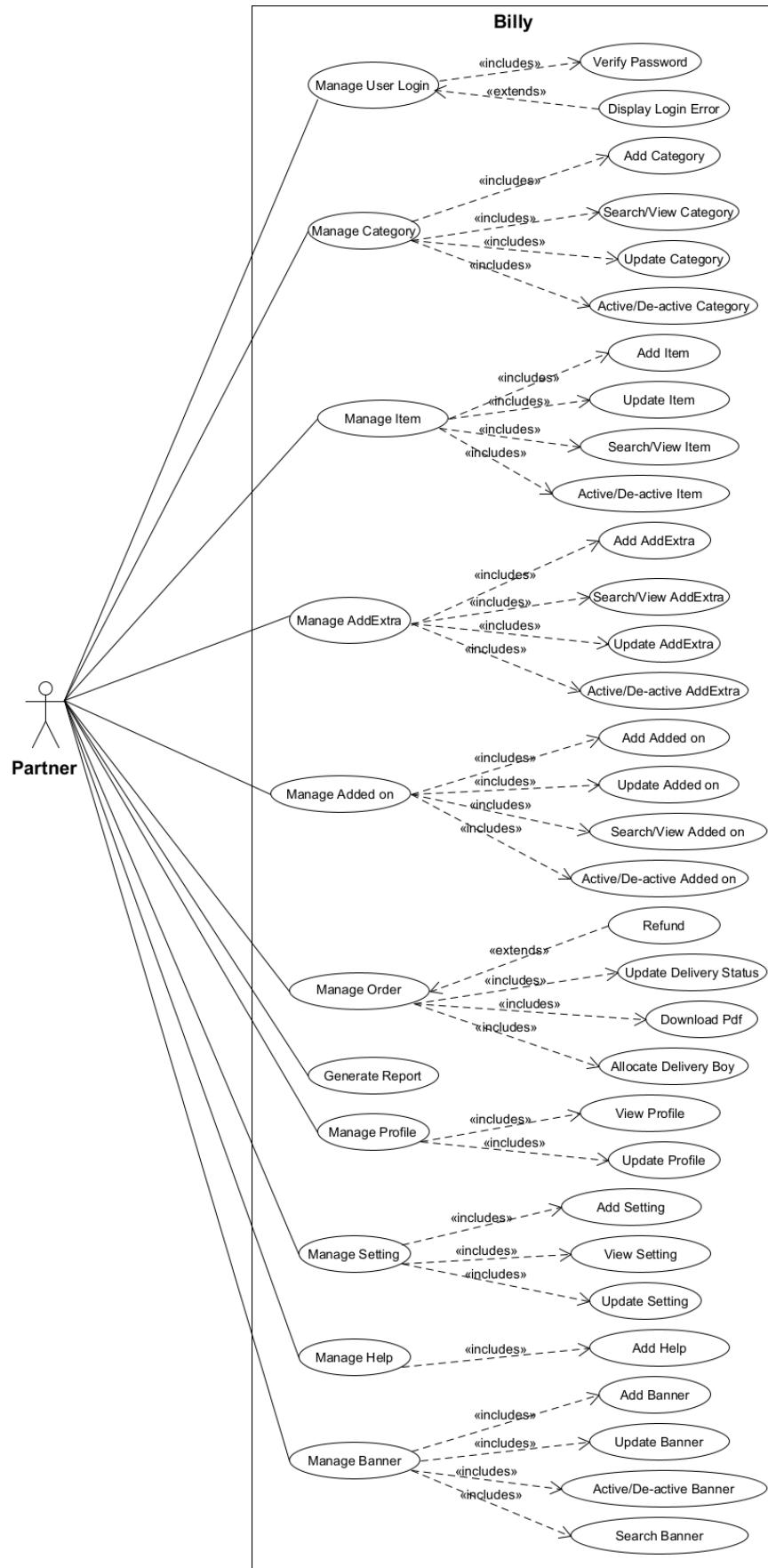
RN	Description	Comments
<b>NFR1</b>	This system has one feature called email verification. The user's entered email id is also verified by the system.	Security
<b>NFR2</b>	This system prevents spam user attacks and automated software (bots) by using ReCAPTCHA.	Security
<b>NFR3</b>	The architecture of our system is based on a typical MVC model.	Security
<b>NFR4</b>	This system is SEO-friendly means that we apply the on-page SEO in this system.	Efficiency
<b>NFR5</b>	To send coupon codes to users the system provides the facility of sending bulk emails.	Efficiency
<b>NFR6</b>	The system has feature translator users can change the system language according to their regional.	Usability
<b>NFR7</b>	The system is provided with a mechanism to verify the owner of certain JSON data using JWT token authentication.	Security

## **5. System Analysis**

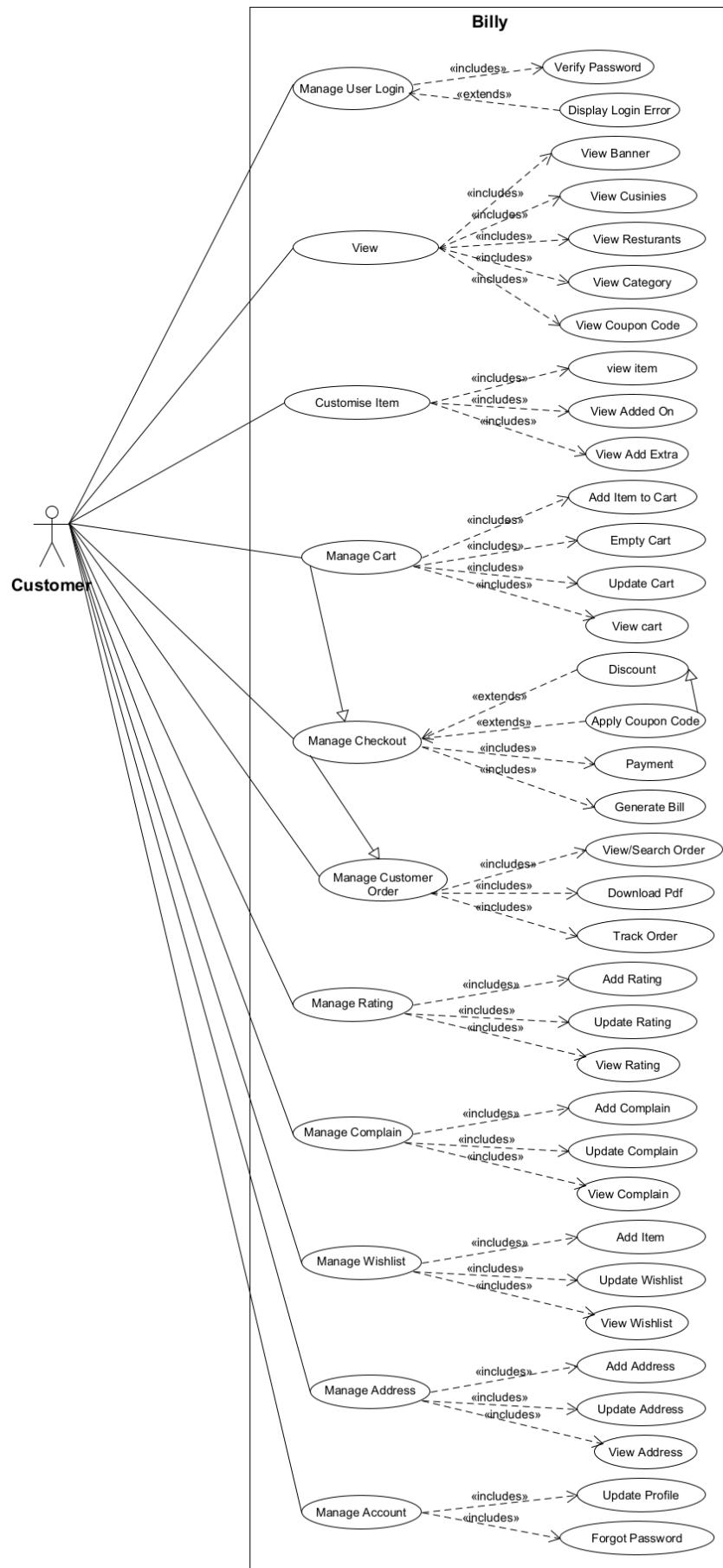
### **5.1.1 Use Case Diagram for Billy Admin**



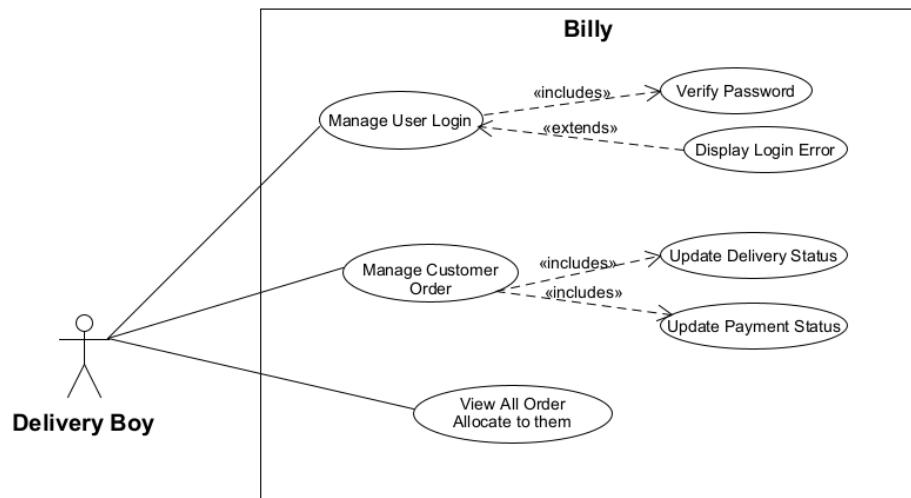
### 5.1.2 Use Case Diagram for Billy Partner



### 5.1.3 Use Case Diagram for Billy Customer

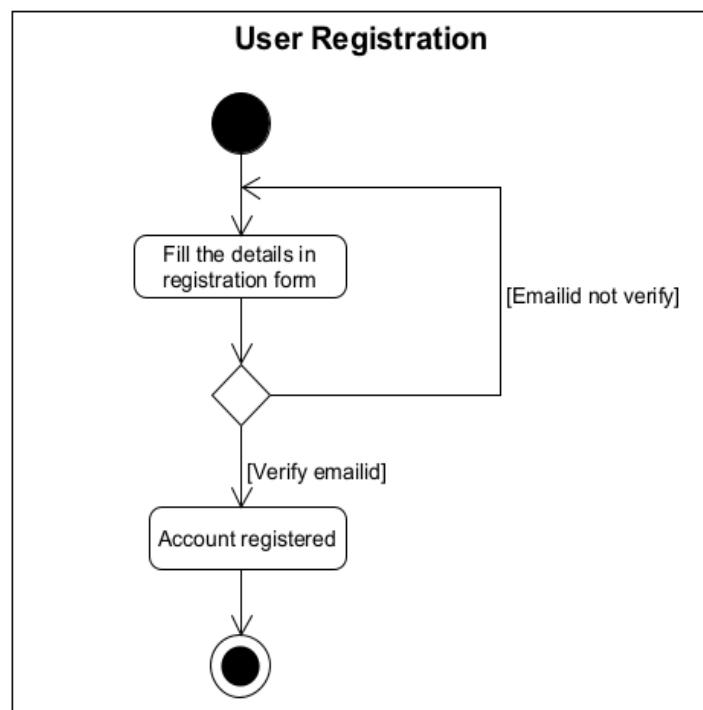


#### 5.1.4 Use Case Diagram for Billy Delivery Boy

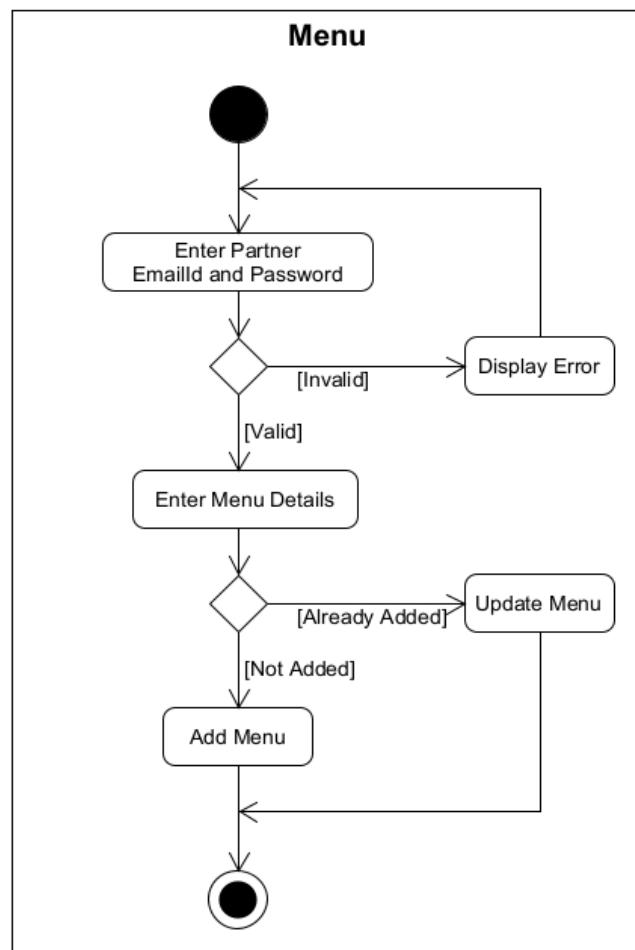


## **5.2 Activity Diagram**

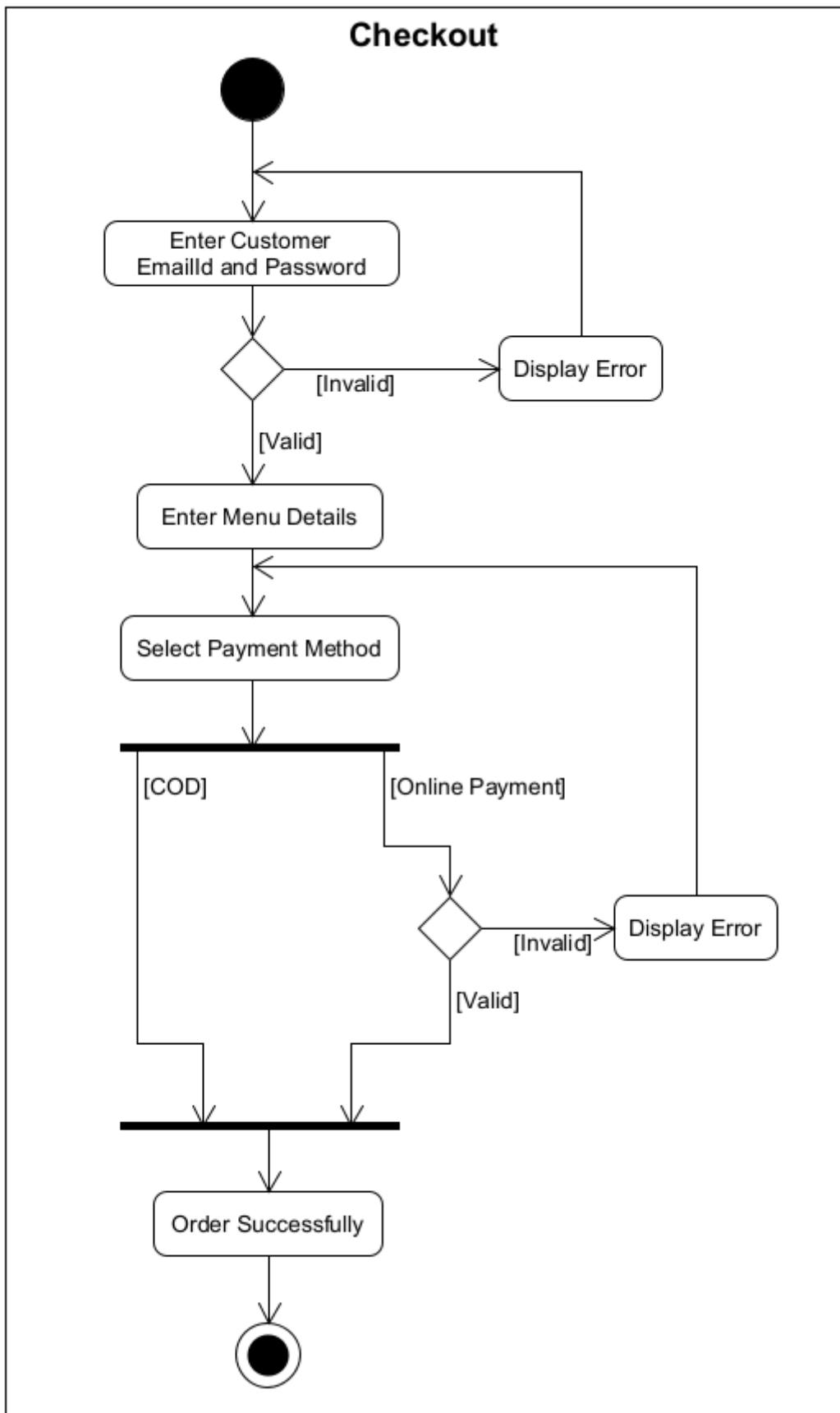
### **5.2.1 Activity Diagram for User Registration**



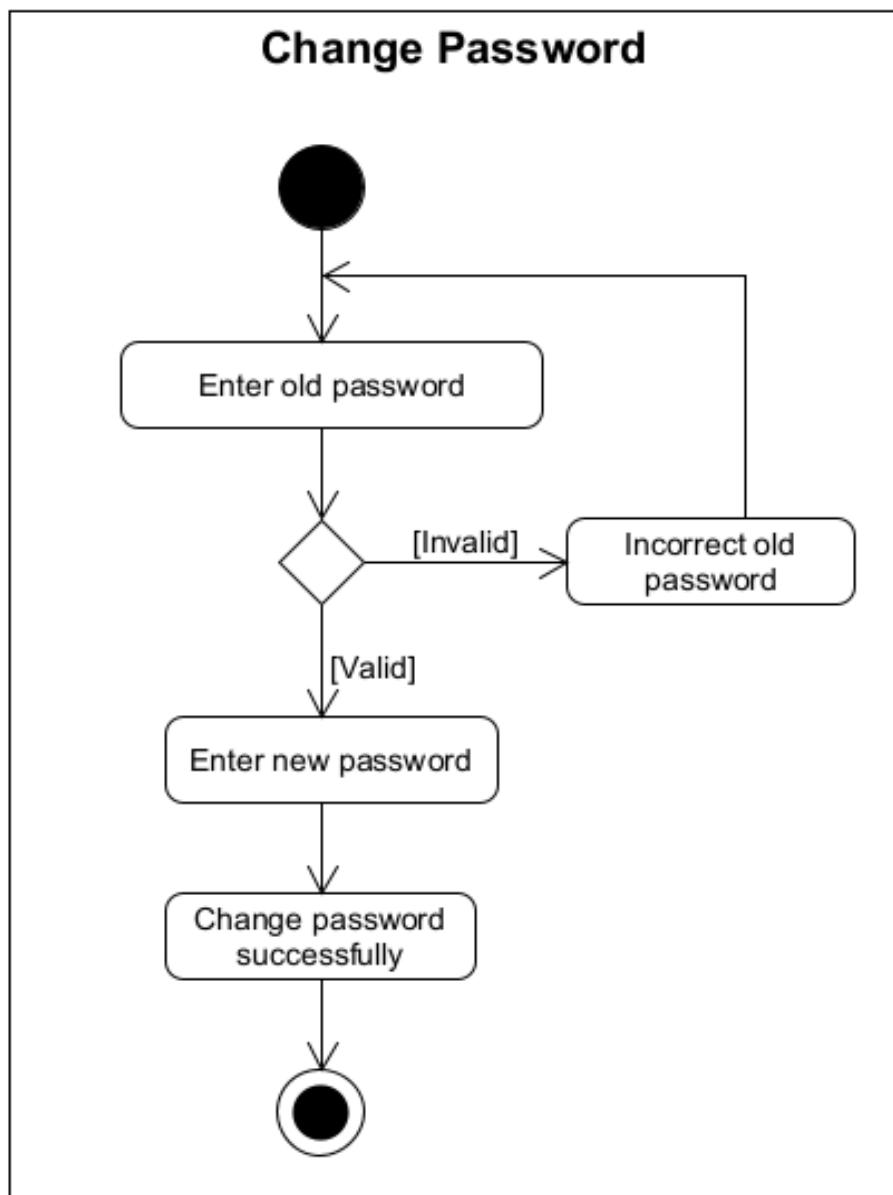
### **5.2.2 Activity Diagram for Menu**



### 5.2.3 Activity Diagram for Checkout

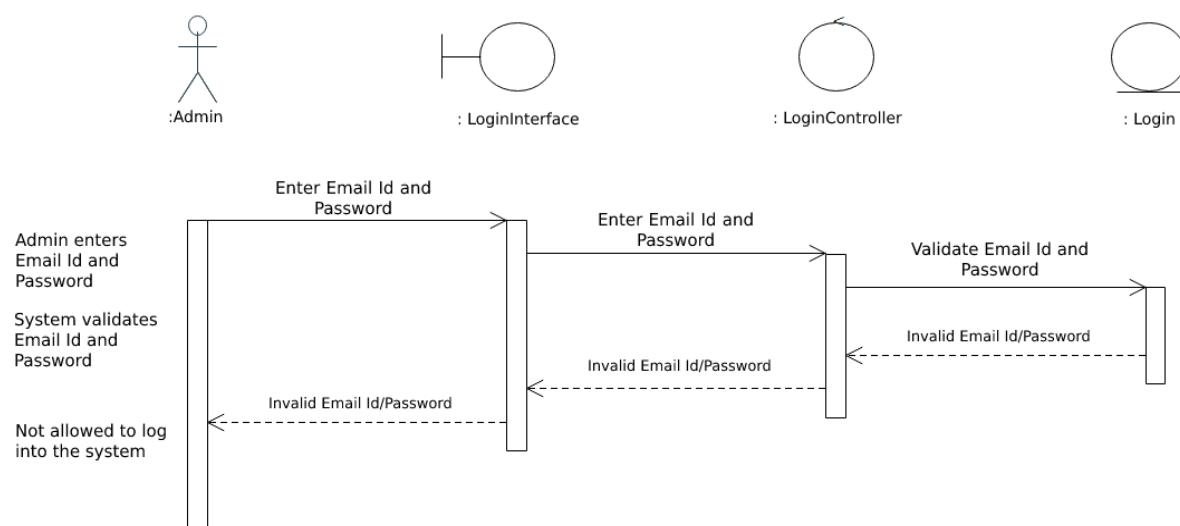


#### 5.2.4 Activity Diagram for Change Password

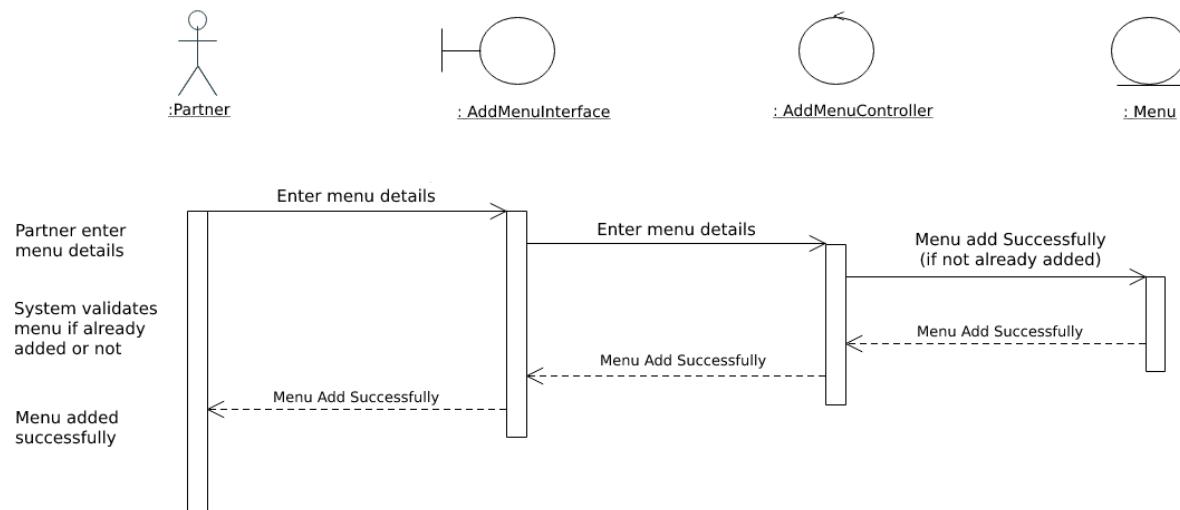


## 5.3 Sequence Diagram

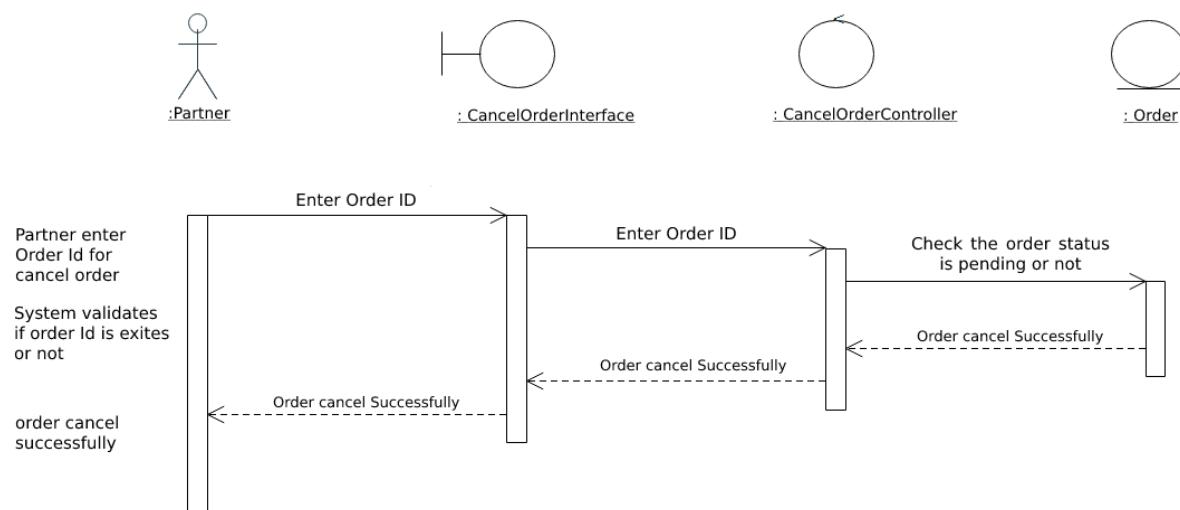
### 5.3.1 Login Admin



### 5.3.2 Add Menu



### 5.3.3 Cancel Order by Partner



## **6. System Design**

### **6.1 Database Design**

#### 6.1.1 Database Schema

##### **1. Admin Table**

admin (ID[PK], Name, Username, Password, Email, createdAt, updatedAt)

FD: ID-> {Name, Username, Password, Email, createdAt, updatedAt}

##### **2. Cuisines Table**

cuisines (ID[PK], Name, Image, Description, Banner, Status, createdAt, updatedAt)

FD: ID-> {Name, Image, Description, Banner, Status, createdAt, updatedAt}

##### **3. Restaurant Table**

restaurant (ID[PK], Restaurant Name, Restaurant Image, Restaurant Address, Restaurant City, Restaurant Contact No., Owner Name, Owner Contact No., Owner Email ID, Owner Password, Reset Password Token, Restaurant Documents, Restaurant Timing, cuisines [FK], Status, createdAt, updatedAt)

FD: ID-> {Restaurant Name, Restaurant Image, Restaurant Address, Restaurant City, Restaurant Contact No., Owner Name, Owner Contact No., Owner Email ID, Owner Password, Reset Password Token, Restaurant Documents, Restaurant Timing, cuisines [FK], Status, createdAt, updatedAt}

FD: Documents-> {GST Certificate, FSSAI Certificate, Sample Bill, Sample Menu, Owner PAN Card}

FD: Restaurant Timing-> {Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday}

##### **4. Added-On Table**

addedon (ID[PK], restaurant [FK], Name, Type, Price, Additional Price, Final Price, Approval Status, Status, createdAt, updatedAt)

FD: ID-> {restaurant [FK], Name, Type, Price, Additional Price, Final Price, Approval Status, Status, createdAt, updatedAt}

##### **5. Add Extra Table**

addextra (ID[PK], restaurant [FK], Name, Type, Price, Additional Price, Final Price, Approval Status, Status, createdAt, updatedAt)

FD: ID-> {restaurant [FK], Name, Type, Price, Additional Price, Final Price, Approval Status, Status, createdAt, updatedAt}

## **6. Category Table**

category (ID[PK], restaurant [FK], Name, Approval Status, Status, createdAt, updatedAt)

FD: ID-> {restaurant [FK], Name, Approval Status, Status, createdAt, updatedAt}

## **7. Item Added-On Table**

itemaddedon (ID[PK], title, Customer Selection, addedon [FK], createdAt, updatedAt)

FD: ID-> {title, Customer Selection, addedon [FK], createdAt, updatedAt}

## **8. Item Add Extra Table**

itemadddextra (ID[PK], title, Customer Selection, addextra [FK], createdAt, updatedAt)

FD: ID-> {title, Customer Selection, addextra [FK], createdAt, updatedAt}

## **9. Variant Table**

variant (ID[PK], Name, UOM, Price, Sales Price, Approval Status, Status, createdAt, updatedAt)

FD: ID -> {Name, UOM, Price, Sales Price, Approval Status, Status, createdAt, updatedAt}

## **10. Item Table**

item (ID[PK], restaurant [FK], category [FK], variant [FK], Name, Type, Description, itemaddedon [FK], itemadddextra [FK], Image, Approval Status, Status, createdAt, updatedAt)

FD: ID-> {restaurant [FK], category [FK], variant [FK], Name, Type, Description, itemaddedon [FK], itemadddextra [FK], Image, Approval Status, Status, createdAt, updatedAt}

## **11. Banner Table**

banner (ID[PK], restaurant [FK], Name, Image, Approval Status, Status, createdAt, updatedAt)

FD: ID-> {restaurant [FK], Name, Image, Approval Status, Status, createdAt, updatedAt}

## **12. Delivery Boy Table**

deliveryboy (ID[PK], Name, Mobile, Password, Status, createdAt, updatedAt)

FD: ID-> {Name, Mobile, Password, Status, createdAt, updatedAt}

## **13. Coupon Code Table**

couponcode (ID[PK], Coupon code, Coupon type, Coupon value, Cart min value, Expired on, Status, createdAt, updatedAt)

FD: ID-> {Coupon code, Coupon type, Coupon value, Cart min value, Expired on, Status, createdAt, updatedAt}

#### **14. Billing Address Table**

billingaddress (ID[PK], Title, Address City, State, Country, Pin Code, createdAt, updatedAt)

FD: ID-> {Title, Address, City, State, Country, Pin Code, createdAt, updatedAt}

#### **15. Customer Table**

customer (ID[PK], Image Name, Email ID, Password, Contact No., Billing Address [FK], OTP, Hash, Referral Code, From Referral Code, Status, createdAt, updatedAt)

FD: ID-> {Image Name, Email ID, Password, Contact No., Billing Address [FK], OTP, Hash, Referral Code, From Referral Code, Status, createdAt, updatedAt}

#### **16. Wishlist Table**

wishlist (ID[PK], customer [FK], item [FK], addedon [FK], addextra [FK], createdAt, updatedAt)

FD: ID-> {customer [FK], item [FK], createdAt, updatedAt}

#### **17. Cart Table**

cart (ID[PK], customer [FK], item [FK], Qty, variant [FK], addedon [FK], addextra [FK], Price, createdAt, updatedAt)

FD: ID-> {customer [FK], item [FK], Qty, variant [FK], addedon [FK], addextra [FK], Price, createdAt, updatedAt}

#### **18. Order Master Table**

Order (ID[PK], customer [FK], Billing Address [FK], Price, Coupon Code [FK], Total Price, Delivery Boy [FK], Payment Status, Payment Type, Payment ID, Order Status, Cancel By, Cancel At, Delivered On, Refund Status, createdAt, updatedAt)

FD: ID-> {customer [FK], Billing Address [FK], Price, Coupon Code [FK], Total Price, Delivery Boy [FK], Payment Status, Payment Type, Payment ID, Order Status, Cancel By, Cancel At, Delivered On, Refund Status, createdAt, updatedAt}

#### **19. Order Detail Table**

Order (ID[PK], Order Master [FK], item [FK], Qty, variant [FK], addedon [FK], addextra [FK], createdAt, updatedAt)

FD: ID-> {Order Master [FK], item [FK], Qty, variant [FK], addedon [FK], addextra [FK], createdAt, updatedAt}

#### **20. Get Touch Table**

gettouch (ID [PK], restaurant [FK], Subject, Message, Status, createdAt, updatedAt)

FD: ID-> {restaurant [FK], Subject, Message, Status, createdAt, updatedAt}

## **21. Setting Table (For Partner)**

setting (ID [PK], restaurant [FK], Cart Min Price, Cart Min Price Message, Delivery charge, GST, createdAt, updatedAt)

FD: ID-> {restaurant [FK], Cart Min Price, Cart Min Price Message, Delivery charge, GST, createdAt, updatedAt}

## **22. Question Table**

Question (ID[PK], Question, createdAt, updatedAt)

FD: ID-> {question, createdAt, updatedAt}

## **23. Complain Table**

complain (ID[PK], ordermaster [FK], question [FK], Message, Approval Status, Status, createdAt, updatedAt,)

FD: ID-> {ordermaster [FK], question [FK], Message, Approval Status, Status, createdAt, updatedAt}

## **24. Referral Amount Table**

referral (ID[PK], Referral Amount, createdAt, updatedAt)

FD: ID-> {Referral Amount, createdAt, updatedAt}

## **25. Review Table**

review (ID[PK], customer [FK], orderdetail [FK], item [FK], ratting, comment, createdAt, updatedAt)

FD: ID-> {customer [FK], orderdetail [FK], item [FK], ratting, comment, createdAt, updatedAt}

## **26. Wallet Table**

wallet (ID[PK], customer [FK], amount, message, type, payment id, createdAt, updatedAt)

FD: ID-> {customer [FK], amount, message, type, payment id, createdAt, updatedAt}

## 6.1.2 Data Dictionary

### 1. Admin

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of admin.
2	Name	String	-	Unique, Not Null	Specifies the name of admin.
3	Username	String	-	Not Null	Specifies the username of admin.
4	Password	String	-	Not Null	Specifies the user define password of admin.
5	Email	String	-	Not Null	Specifies the email id of admin.
6	createdAt	timestamps	-	Not Null	Specifies the createdAt of admin.
7	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of admin.

### 2. Cuisines Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Cuisines.
2	Name	String	-	Not Null	Specifies the name of Cuisines.
3	Image	String	-	Not Null	Specifies the image URL of Cuisines.
4	Description	String	-	Not Null	Specifies the description of Cuisines.
5	Banner	String	-	Not Null	Specifies the banner URL of Cuisines.
6	Status	Boolean	-	Default: True	Specifies the status of Cuisines.

7	createdAt	timestamps	-	Not Null	Specifies the createdAt of Cuisines.
8	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Cuisines.

### 3. Restaurant Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Restaurant.
2	Restaurant Name	String	-	Unique, Not Null	Specifies the name of Restaurant.
3	Restaurant Image	String	-	Not Null	Specifies the image URL of Restaurant.
4	Restaurant Address	String	-	Not Null	Specifies the address of Restaurant.
5	Restaurant City	String	-	Default: Bardoli	Specifies the city of Restaurant.
6	Restaurant Contact No.	Number	-	Not Null	Specifies the contact no of Restaurant.
7	Owner Name	String	-	Not Null	Specifies the name of Owner.
8	Owner Contact No.	Number	-	Unique, Not Null	Specifies the contact no of Owner.
9	Owner Email ID	String	-	Unique, Not Null	Specifies the email id of Owner.
10	Owner Password	String	-	Not Null	Specifies the password of Owner.
11	Reset Password Token	String	-	Not Null	Specifies the Reset password Token of Owner.

12	Restaurant Documents	Object of String	-	Null	Specifies the documents of Restaurant.
13	Restaurant Timing	Object of Array (Date)	-	Null	Specifies the timing of Restaurant
14	cuisines	Array of ObjectId	-	Foreign Key	Reference from cuisines (Id).
15	Status	Boolean	-	Default: True	Specifies the status of Restaurant.
16	createdAt	timestamps	-	Not Null	Specifies the createdAt of Restaurant.
17	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Restaurant.

#### 4. Added-On Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Added-On.
2	restaurant	ObjectId	-	Foreign Key	Reference from restaurant (Id).
3	Name	String	-	Unique, Not Null	Specifies the name of Added-On.
4	Type	String	-	enum ['veg', 'non-veg', 'egg']	Specifies the type of Added-On.
5	Price	Number	-	Not Null	Specifies the price of Added-On.
6	Additional Price	Number	-	Not Null	Specifies the additional price of Added-On.
7	Final Price	Number	-	Not Null	Specifies the final price of Added-On.
8	Approval Status	Number	-	Default: 1	Specifies the approval status of Added-On.
9	Status	Boolean	-	Default: True	Specifies the status of Added-On.

10	createdAt	timestamps	-	Not Null	Specifies the createdAt of Added-On.
11	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Added-On.

## 5. Add Extra Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Add Extra.
2	restaurant	ObjectId	-	Foreign Key	Reference from restaurant (Id).
3	Name	String	-	Unique, Not Null	Specifies the name of Add Extra.
4	Type	String	-	enum ['veg', 'non-veg', 'egg']	Specifies the type of Add Extra.
5	Price	Number	-	Not Null	Specifies the price of Add Extra.
6	Additional Price	Number	-	Not Null	Specifies the additional price of Add Extra.
7	Final Price	Number	-	Not Null	Specifies the final price of Add Extra.
8	Approval Status	Number	-	Default: 1	Specifies the approval status of Add Extra.
9	Status	Boolean	-	Default: True	Specifies the status of Add Extra.
10	createdAt	timestamps	-	Not Null	Specifies the createdAt of Add Extra.
11	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Add Extra.

## 6. Category Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Category.
2	restaurant	ObjectId	-	Foreign Key	Reference from restaurant (Id).
3	Name	String	-	Unique, Not Null	Specifies the name of Category.
4	Approval Status	Number	-	Default: 1	Specifies the approval status of Category.
5	Status	Boolean	-	Default: True	Specifies the status of Category.
6	createdAt	timestamps	-	Not Null	Specifies the createdAt of Category.
7	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Category.

## 7. Item Added-On Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Item Added-On.
2	Title	String	-	Unique, Not Null	Specifies the title of Item Added-On.
3	Customer Selection	String	-	enum ['optional', 'compulsory']	Specifies the customer selection of Item Added-On.
4	addedon	Array of ObjectId	-	Foreign Key	Reference from added-on (Id).
5	createdAt	timestamps	-	Not Null	Specifies the createdAt of Item Added-On.

6	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Item Added-On.
---	-----------	------------	---	----------	---

## 8. Item Add Extra Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Item Add Extra.
2	Title	String	-	Unique, Not Null	Specifies the title of Item Add Extra.
3	Customer Selection	String	-	enum ['optional', 'compulsory']	Specifies the customer selection of Item Add Extra.
4	adddextra	Array of ObjectId	-	Foreign Key	Reference from Add Extra (Id).
5	createdAt	timestamps	-	Not Null	Specifies the createdAt of Item Add Extra.
6	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Item Added-On Item Add Extra.

## 9. Variant Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Variant.
2	Name	String	-	Not Null	Specifies the name of Variant.
3	UOM	String	-	Not Null	Specifies the uom of Variant.
4	Price	Number	-	Not Null	Specifies the price of Variant.
5	Sales price	Number	-	Not Null	Specifies the sales price of Variant.

6	Approval Status	Number	-	Default: 1	Specifies the approval status of Variant.
7	Status	Boolean	-	Default: True	Specifies the status of Variant.
8	createdAt	timestamps	-	Not Null	Specifies the createdAt of Variant.
9	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Variant.

## 10. Item Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Item.
2	restaurant	ObjectId	-	Foreign Key	Reference from restaurant (Id).
3	category	ObjectId	-	Foreign Key	Reference from category (Id).
4	variant	ObjectId	-	Foreign Key	Reference from variant (Id).
5	Name	String	-	Not Null	Specifies the name of Item.
6	Type	String	-	enum ['veg', 'non-veg', 'egg']	Specifies the type of Item.
7	Description	String	-	Not Null	Specifies the description of Item.
8	itemaddedon	ObjectId	-	Foreign Key	Reference from itemaddedon (Id).
9	itemaddextra	ObjectId	-	Foreign Key	Reference from itemaddextra (Id).
10	Image	String	-	Not Null	Specifies the image URL of Item.
11	Approval Status	Number	-	Default: 1	Specifies the approval status of Item.
12	Status	Boolean	-	Default: False	Specifies the status of Item.

13	createdAt	timestamps	-	Not Null	Specifies the createdAt of Item.
14	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Item.

## 11. Banner Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Banner.
2	restaurant	ObjectId	-	Foreign Key	Reference from restaurant (Id).
3	Name	String	-	Not Null	Specifies the name of Banner.
4	Image	String	-	Not Null	Specifies the image URL of Banner.
5	Approval Status	Number	-	Default: 1	Specifies the approval status of Banner.
6	Status	Boolean	-	Default: True	Specifies the status of Banner.
7	createdAt	timestamps	-	Not Null	Specifies the createdAt of Banner.
8	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Banner.

## 12. Delivery Boy Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Delivery Boy.
2	Name	String	-	Not Null	Specifies the name of Delivery Boy.
3	Mobile No	Number	-	Unique, Not Null	Specifies the mobile no. of Delivery Boy.

4	Password	String	-	Not Null	Specifies the password of Delivery Boy.
5	Status	Boolean	-	Default: True	Specifies the status of Delivery Boy.
6	createdAt	timestamps	-	Not Null	Specifies the createdAt of Delivery Boy.
7	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Delivery Boy.

### 13. Coupon Code Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Coupon code.
2	Coupon Code	String	-	Unique, Not Null	Specifies the name of Coupon code.
3	Coupon Type	String	-	enum ['p', 'f']	Specifies the type of Coupon code.
4	Coupon Value	Number	-	Not Null	Specifies the value of Coupon code.
5	Cart Min Value	Number	-	Not Null	Specifies the Cart Minimum Value.
6	Expired On	Date	-	Not Null	Specifies date on which the coupon code is expired.
7	Status	Boolean	-	Default: True	Specifies the status of Coupon code.
8	createdAt	timestamps	-	Not Null	Specifies the createdAt of Coupon code.
9	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Coupon code.

#### 14. Billing Address Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Billing Address.
2	Title	String	-	Not Null	Specifies the title of Billing Address.
3	Address	String	-	Not Null	Specifies the address of Billing Address.
4	City	String	-	Not Null	Specifies the city of Billing Address.
5	State	String	-	Not Null	Specifies the state of Billing Address.
6	Country	String	-	Not Null	Specifies the country of Billing Address.
7	Pin Code	Number	-	Not Null	Specifies the pin code of Billing Address.
8	createdAt	timestamps	-	Not Null	Specifies the createdAt of Billing Address.
9	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Billing Address.

#### 15. Customer Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Customer.
2	Image	String	-	Not Null	Specifies the image URL of Customer.
3	Name	String	-	Not Null	Specifies the name of Customer.

4	Email ID	String	-	Unique, Not Null	Specifies the email id of Customer.
5	Password	String	-	Not Null	Specifies the password of Customer.
6	Contact No.	Number	-	Unique, Not Null	Specifies the contact no. of Customer.
7	Billing Address	Array of ObjectId	-	Foreign Key	Reference from Billing Address (Id).
8	OTP	String	-	Null	Specifies the OTP of Customer.
9	Hash	String	-	Null	Specifies the hash of Customer.
10	Referral Code	String	-	Null	Specifies the referral code of Customer.
11	From Referral Code	String	-	Null	Specifies the from referral code of Customer.
12	Status	Boolean	-	Default: True	Specifies the status of Customer.
13	createdAt	timestamps	-	Not Null	Specifies the createdAt of Customer.
14	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Customer.

## 16. Wishlist Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Wishlist.
2	customer	ObjectId	-	Foreign Key	Reference from customer (Id).
3	item	ObjectId	-	Foreign Key	Reference from item (Id).
4	addedon	ObjectId	-	Foreign Key	Reference from addedon (Id).
5	adddextra	ObjectId	-	Foreign Key	Reference from adddextra(Id).

6	createdAt	timestamps	-	Not Null	Specifies the createdAt of Wishlist.
7	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Wishlist.

## 17. Cart Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Cart.
2	customer	ObjectId	-	Foreign Key	Reference from customer (Id).
3	item	ObjectId	-	Foreign Key	Reference from item (Id).
4	Qty	Number	-	Default: 1	Specifies the qty of Cart.
5	variant	ObjectId	-	Foreign Key	Reference from variant (Id).
6	addedon	ObjectId	-	Foreign Key	Reference from addedon (Id).
7	adextra	ObjectId	-	Foreign Key	Reference from adextra(Id).
8	Price	Number	-	Not Null	Reference from price.
9	createdAt	timestamps	-	Not Null	Specifies the createdAt of Cart.
10	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Cart.

## 18. Order Master Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Order Master.
2	customer	ObjectId	-	Foreign Key	Reference from customer (Id).
3	billingaddress	ObjectId	-	Foreign Key	Reference from billing address (Id).

4	Price	Number	-	Not Null	Specifies the price of Order Master.
5	couponcode	ObjectId	-	Foreign Key	Reference from coupon code (Id).
6	Total Price	Number	-	Not Null	Specifies the total price of Order Master.
7	deliveryboy	ObjectId	-	Foreign Key	Reference from delivery boy (Id).
8	Payment Status	String	-	enum ['pending', 'paid']	Specifies the payment status of Order Master.
9	Payment Type	String	-	default: cod	Specifies the payment type of Order Master.
10	Payment ID	String	-	Null	Specifies the payment id of Order Master.
11	Order Status	String	-	Enum ['pending', 'cooking', 'ontheway', 'delivered']	Specifies the order status of Order Master.
12	Cancel By	String	-	Null	Specifies the cancel by of Order Master.
13	Cancel At	Date	-	Null	Specifies the cancel at of Order Master.
14	Delivered On	Date	-	Null	Specifies the delivered on of Order Master.
15	Refund Status	Boolean	-	default: False	Specifies the order status of Order Master.
16	createdAt	timestamps	-	Not Null	Specifies the createdAt of Order Master.
17	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Order Master.

## 19. Order Detail Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Order Detail.
2	ordermaster	ObjectId	-	Foreign Key	Reference from order master (Id).
3	item	ObjectId	-	Foreign Key	Reference from item (Id).
4	Qty	Number	-	Default: 1	Specifies the qty of Order Detail.
5	variant	ObjectId	-	Foreign Key	Reference from variant (Id).
6	addedon	array of ObjectId	-	Foreign Key	Reference from addedon (Id).
7	adddextra	array of ObjectId	-	Foreign Key	Reference from adddextra(Id).
8	createdAt	timestamps	-	Not Null	Specifies the createdAt of Order Detail.
9	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Order Detail.

## 20. Get Touch Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Get Touch.
2	restaurant	ObjectId	-	Foreign Key	Reference from restaurant (Id).
3	Subject	String	-	Not Null	Specifies the subject of Get Touch.
4	Message	String	-	Not Null	Specifies the message of Get Touch.
5	Status	Boolean	-	Default: True	Specifies the status of Get Touch.

6	createdAt	timestamps	-	Not Null	Specifies the createdAt of Get Touch.
7	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Get Touch.

## 21. Setting Table (For Partner)

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Setting.
2	restaurant	ObjectId	-	Foreign Key	Reference from restaurant (Id).
3	Cart Min Price	Number	-	Not Null	Specifies the cart minimum price of Setting.
4	Cart Min Price Message	String	-	Not Null	Specifies the cart minimum price message of Setting.
5	Delivery charge	Number	-	Not Null	Specifies the delivery charge of Setting.
6	GST	Number	-	Not Null	Specifies the GST of Setting.
7	createdAt	timestamps	-	Not Null	Specifies the createdAt of Setting.
8	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Setting.

## 22. Question Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Question.
2	Name	String	-	Not Null	Specifies the name of Question.
3	createdAt	timestamps	-	Not Null	Specifies the createdAt of Question.

4	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Question.
---	-----------	------------	---	----------	--------------------------------------

### 23. Complain Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Complain.
2	order master	ObjectId	-	Foreign Key	Reference from order master (Id).
3	question	ObjectId	-	Foreign Key	Reference from question (Id).
4	Message	String	-	Not Null	Specifies the message of Complain.
5	Approval Status	Number	-	Default: 1	Specifies the approval status of Complain.
6	Status	Boolean	-	Default: False	Specifies the status of Complain.
7	createdAt	timestamps	-	Not Null	Specifies the createdAt of Complain.
8	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Complain.

### 24. Referral Amount Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Referral Amount.
2	Referral Amount	Number	-	Not Null	Specifies the amount of Referral Amount.
3	createdAt	timestamps	-	Not Null	Specifies the createdAt of Referral Amount.

4	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Referral Amount.
---	-----------	------------	---	----------	---

## 25. Review Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Review.
2	Customer	ObjectId	-	Foreign Key	Reference from customer(Id).
3	orderdetail	ObjectId	-	Foreign Key	Reference from order detail (Id).
4	Item	ObjectId	-	Foreign Key	Reference from item(Id).
5	rating	Number	-	Not Null	Specifies the rating of Review.
6	comment	String	-	Null	Specifies the comment of Review.
7	createdAt	timestamps	-	Not Null	Specifies the createdAt of Review.
8	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Review.

## 26. Wallet Table

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1	Id	ObjectId	-	Primary key, Auto increment	Identifies Unique id of Wallet.
2	customer	ObjectId	-	Foreign Key	Reference from customer (Id).
3	amount	Number	-	Not Null	Specifies the amount of Wallet.
4	message	String	-	Not Null	Specifies the message of Wallet.
5	type	String	-	enum ['in', 'out']	Specifies the type of Wallet.
6	payment id	String	-	Null	Specifies the payment id of Wallet.
7	createdAt	timestamps	-	Not Null	Specifies the createdAt of Wallet.
8	updatedAt	timestamps	-	Not Null	Specifies the updatedAt of Wallet.

## **7. System Implementation**

### **7.1 Screenshots**

#### **7.1.1 Home Page (Web Portal)**



Welcome to Billy

[Admin Portal](#)

[Partner Portal](#)

POPULAR CITIES IN INDIA

Bardoli



#### **7.1.2 Admin Portal**

##### **1. Login**

The image shows a mobile-style login interface for the 'BILLY' application. At the top is a small orange house icon. Below it is the 'BILLY' logo. The text 'Admin Login' is centered above two input fields. The first field is a light blue rectangle containing the placeholder 'Admin@gmail.com'. The second field is a light blue rectangle containing a series of dots representing a password, with an eye icon to its right. At the bottom is a large orange rectangular button with the white text 'SIGN IN'.

## 2. Dashboard

The dashboard features a sidebar with navigation links: Dashboard, Orders, Cuisines, Restaurant, Items, Customer, Couponcode, DeliveryBoy, GetTouch, Question, Complain, and referralamount. The main area displays four cards: Total User (1), Total Restaurant (1), Total Items (5), and Total Delivery Boy (1). Below these are two more cards: Most Liked Dish (represented by a drink icon) and Most Active User (represented by a person icon). A section for Latest 5 Order shows a table with one row:

Order Id	Name/Email/Mobile	Address/Zipcode	Price	Payment Type	Payment Status	Order Status	Added On
1	meet 19bmit0015@gmail.com 9665457841	Surat 395854	455	COD	Pending	Pending	25/10/2022

Copyright © 2023 . All rights reserved.

## 3. Cuisines

### Add Cuisines

Cuisines							
S.No #	CuisinesImage	CuisinesBanner	CuisinesName	Description	Added On	Cuisines Status	Actions
1			Pizza	pizza, dish of Italian origin ...	2022-08-07	<input checked="" type="checkbox"/>	<a href="#">Edit</a> <a href="#">Delete</a>
2			Indian Snacks	A snack is a small portion of ...	2022-11-06	<input checked="" type="checkbox"/>	<a href="#">Edit</a> <a href="#">Delete</a>

Rows per page: 5 rows ▾ 1-2 of 2 | < 1-2 of 2 > |

## 4. Manage Cuisines

Manage Cuisine

Name  
Pizza

Description  
pizza, dish of Italian origin consisting of a flattened disk of bread dough topped with some combination of olive oil, oregano, tomato, olives, mozzarella or other cheese, and many other ingredients, baked quickly—usually, in a cor

Cuisine Image  
 Upload  
 /uploads/cuisines/images/1659867112004-pizza.png

Cuisine Banner  
 Upload  
 /uploads/cuisines/images/1659867199445-slider\_1.jpg

**Submit**

## 5. Restaurant

Restaurants						<input type="text"/> Search	X	
S.No #	restaurantName	restaurantImage	restaurantAddress	restaurantContact	Added On	Actions		
1	Dominos		LAKE VIEW POINT,BARDOLI, GUJARAT, Baben	18002081234	2022-07-25			
2	Shree Ganesh Aloopuri		Khaudhra gali, Bardoli, Surat, Gujarat	9664949012	2022-12-09			
3	Tea Shop		dham dod lumbha road, Bardoli, Surat, Gujarat	7874376578	2023-03-20			

Rows per page: 5 rows ▾ 1-3 of 3 |< < 1-3 of 3 > >|

## 6. Restaurant Details

### Restaurant Details

#### Restaurant Details



Name  
**Dominos**

Mobile Number  
**18002081234**

Address  
**LAKE VIEW POINT,BARDOLI, GUJARAT, Baben**

City  
**Bardoli**

#### Owner Details

Owner Name  
**Name**

Email ID  
**dominos@gmail.com**

Owner ContactNo.  
**9212320920**

City  
**Bardoli**

#### Document Details

Gst Certificate

Sample Menu

Fssai Certificate

Owner

Sample Bill

#### Time Details

Monday

9:30 AM

11:00 PM

Tuesday

10:00 AM

11:00 PM

Wednesday

9:30 AM

11:00 PM

Thursday

10:00 AM

11:00 PM

Friday

9:30 AM

11:00 PM

Saturday

10:00 AM

11:00 PM

#### Show Restaurant (Addon,AddExtra,Variants) Details

Show Addon

Show AddExtra

Show Variants

#### Show Restaurant (Banner) Details

Show Banners

Show Category

## 7. Display Selected Restaurant Add On

Addon							
S.No #	Addon Name	Addon Type	Addon Price	Addon Additional Price	Addon Final Price	Added On	Approval Status
1	Paneer	veg	50	0	50	2022-08-09	Submission Pending
2	Onion	veg	30	0	30	2022-08-31	Submission Pending In Review Approved Rejected
3	Cheese Burst	veg	30	0	30	2022-10-15	Approved
4	100% Wheat Thin Crust	veg	20	0	20	2022-10-16	Approved

Rows per page: 5 rows ▾ 1-4 of 4 |< < 1-4 of 4 > >|

## 8. Item

Item							
S.No #	Item Image	Item Name	Item Description	Restaurant	Category	Added On	Approve Status
1		Farmhouse	Delightful combinati ...	Dominos Name 18002081234	Veg Pizza	2022-09-23	Approved
2		Kadhai Paneer	Paneer chunks in you ...	Dominos Name 18002081234	Veg Pizza	2022-10-16	Approved
3		Black Tea	Black tea is a type ...	Tea Shop Ram Modi 7874376578	Recommended	2023-03-21	Approved
4		Pizza	This is an veg pizza ...	Dominos Name 18002081234	Veg Pizza	2023-04-25	Approved
5		Aloopuri	Images used are for ...	Shree Ganesh Aloopuri Ganesh 9664949012	Recommdded	2023-04-28	Approved

Rows per page: 5 rows ▾ 1-5 of 5 |< < 1-5 of 5 > >|

## 9. Customer

Customer						
S.No #	Name	Email	Mobile	Wallet	Added On	Actions
1	Abhishek Choksi	19bmiit032@gmail.com	7874376579	0	2023-03-01	

## 10. Coupon Code

Coupon Code								
S.No #	Coupon Code Name	Coupon Code Type	Coupon Code Value	Coupon Code Cart Min Value	Coupon Code Expiredon	Added On	Addon Status	Actions
1	sumer2023	p	5	29	2023-04-30	2023-04-28		

## 11. Manage Coupon Code

### Manage Coupon Code

Name  
sumer2023

Coupon Code Type  
Percentage

couponCodeValue  
5

Coupon Code Cart Min Value  
29

Coupon Code Expiredon  
30-04-2023

## 12. Send Coupon Code

Select User

Abhishek Choksi (19bmiit032@gmail.com) Select

Submit

## 13. Receive Coupon Code in Email

Coupon Code Inbox ×



Billy <abhishekmeet032015@gmail.com>  
to me ▾

15:15 (0 minutes ago)



#srno	Coupon Code	Coupon Code value
1	sumer2023	5

Reply

Forward

## 14. Delivery Boy

Add Delivery Boy

Delivery Boy					
S.No #	Name	Mobile	Added On	Delivery Boy Status	Actions
1	Meet Patel	9678458547	2023-04-28	<input checked="" type="checkbox"/>	<span>Edit</span> <span>Delete</span>

## 15. Manage Delivery Boy

Manage Delivery Boy

Name

Meet Patel

Mobile Number

9678458547

Submit

## 16. Get Touch

Get Touch				
S.No #	RestaurantName	Subject	Message	Actions
1	Dominos	Approval Item	Please approve my pizza item.	
Rows per page: 5 rows ▾ 1-1 of 1  < < 1-1 of 1 > >				

## 17. Complain Question

[Add Question](#)

Question			
S.No #	Question	Added On	Actions
1	Container Leak	2023-04-25	
2	Cold Food Provide	2023-04-28	
3	Melt Food	2023-04-28	
Rows per page: 5 rows ▾ 1-3 of 3  < < 1-3 of 3 > >			

## 18. Manage Complain Question

[Manage Question](#)

Name	<input type="text" value="Container Leak"/>

## 19. Complain

Complain						
S.No #	Customer	Order	Question	Message	Added On	Actions
No records to display						
Rows per page:	5 rows	▼	0-0 of 0	<	< 0-0 of 0 >	>

## 20. Referral Amount

Manage Referral Amount

Amount	<input type="text" value="50"/>
<input type="button" value="Submit"/>	

### 7.1.2 Partner Portal

#### 1. Registration

The screenshot shows a registration form for the Bibly Partner program. The form includes fields for basic information and a file upload section.

**Fields:**

- First Name: Lapinos
- Last Name: Meet Patel
- Address: Canel Road, near silver plaza
- Phone Number: 7854512458
- Email: meetpatel@gmail.com
- City: Bardoli
- Pincode: 9658451244
- File Upload: A placeholder box labeled "Screenshot\_2021-06-10-09-58-45-777\_com.android.chrome.jpg" with an "Upload" button.

**Buttons:**

- Upload button
- Sign Up button

## 2. Upload Documents

1 Documents      2 Timing      3 Cuisines

**GST Certificate**

+ Upload

file 1659434204766-gstCertificate.pdf

**Fssai Certificate**

+ Upload

file 1659434204767-fssaiCertificate.pdf

**Sample Bill**

+ Upload

file 1659434204769-sampleBill.pdf

**Sample Menu**

+ Upload

file 1659434204771-sampleMenu.pdf

**Owner Pancard**

+ Upload

file 1659434204774-ownerPan.pdf

Next

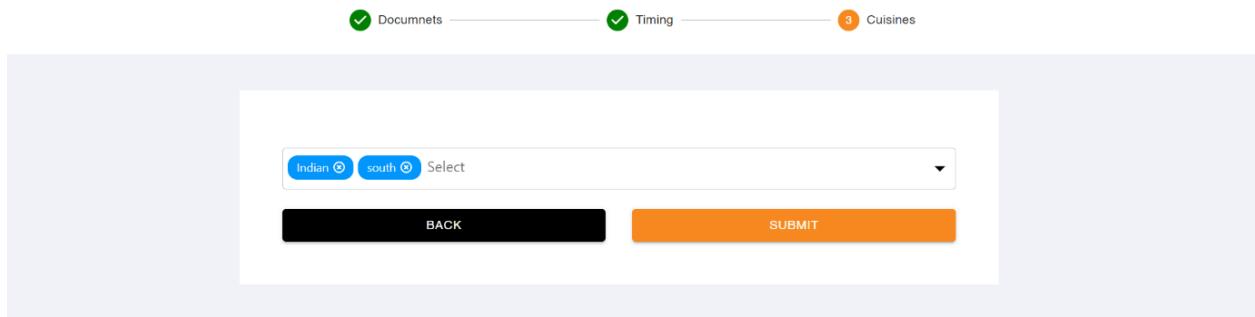
## 3. Add Restaurant Time

1 Documents      2 Timing      3 Cuisines

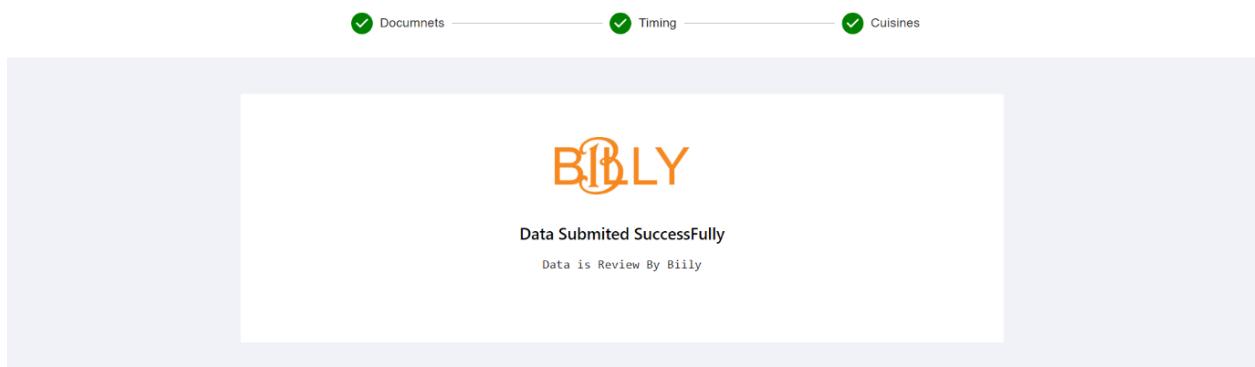
Back

Next

#### 4. Select Cuisines



#### 5. Registration Successfully



#### 6. Category

Add Category						
Category				Category Status	Approval Status	Actions
S.No #	Restaurant	Category	Added On	Category Status	Approval Status	Actions
1	Dominos	Veg Pizza	2022-08-05			
Rows per page: 5 rows < 1-1 of 1 >						

## 7. Manage Category

Manage Category

Name  
Veg Pizza

Submit

## 8. Add On

S.No #	Restaurant	Addon Name	Addon Type	Addon Final Price	Added On	Addon Status	Approval Status	Actions
1	Dominos	Paneer	veg	50	2022-08-09	<input checked="" type="checkbox"/>	Approved	<button>Edit</button>
2	Dominos	Onion	veg	30	2022-08-31	<input checked="" type="checkbox"/>	Approved	<button>Edit</button>
3	Dominos	Cheese Burst	veg	30	2022-10-15	<input checked="" type="checkbox"/>	Approved	<button>Edit</button>
4	Dominos	100% Wheat Thin Crust	veg	20	2022-10-16	<input checked="" type="checkbox"/>	Approved	<button>Edit</button>

Rows per page: 5 rows ▾ 1-4 of 4 | < 1-4 of 4 > >|

## 9. Manage Add On

Manage Addons

Name  
Paneer

Addons Type  
veg

Addon Price  
50

Addon Additional Price  
0

Addon Final Price  
50

Submit

## 10. Add Extra

Add Extra

Add Extra								
S.No #	Restaurant	Add Extra Name	Add Extra Type	Add Extra Final Price	Add Extra	Add Extra Status	Approval Status	Actions
1	Dominos	Red Pepper	veg	35	2022-08-19	<input checked="" type="checkbox"/>	Approved	<button>Edit</button>
2	Dominos	Extra Cheese	veg	50	2022-09-03	<input checked="" type="checkbox"/>	Approved	<button>Edit</button>

Rows per page: 5 rows ▾ 1-2 of 2 | < < 1-2 of 2 > >|

## 11. Manage Add Extra

Manage Add Extra

Name

Add Extra Type

Add Extra Price

Add Extra Additional Price

Addon Final Price

Submit

## 12. Variant

Add Variant

Variant							
S.No #	Variant Name	Variant UOM	Variant Price	Variant Sales Price	Added On	Variant Status	Actions
Variant							
S.No #	Variant Name	Variant UOM	Variant Price	Variant Sales Price	Added On	Variant Status	Actions
1	Regular	Size	269	269	2022-09-11	<input checked="" type="checkbox"/>	<button>Edit</button> <button>Delete</button>
2	Medium	Size	529	529	2022-09-12	<input checked="" type="checkbox"/>	<button>Edit</button> <button>Delete</button>
3	Medium	Size	579	579	2022-10-16	<input checked="" type="checkbox"/>	<button>Edit</button> <button>Delete</button>
4	Regular	Size	319	319	2022-10-16	<input checked="" type="checkbox"/>	<button>Edit</button> <button>Delete</button>
5	Large	Size	839	839	2022-10-16	<input checked="" type="checkbox"/>	<button>Edit</button> <button>Delete</button>

Rows per page: 5 rows ▾ 1-5 of 5 | < < 1-5 of 5 > >|

## 13. Manage Variant

Manage Variant

Name	Regular
Variant UOM	Size
Variant Price	269
Variant Sales Price	269
<input type="button" value="Submit"/>	

## 14. Item

Add Item

Item								
S.No #	Item Image	Item Name	Item Description	Category	Added On	Actions	Approval Status	Actions
1		Farmhouse	Delightful combination of onion, capsicum, tomato & grilled mushroom	Veg Pizza	2022-09-23	<input checked="" type="checkbox"/>	Approved	<input type="button" value="Edit"/>
2		Kadhai Paneer	Paneer chunks in your favourite spicy kadhai masala flavors, with toppings of onion & capsicum	Veg Pizza	2022-10-16	<input checked="" type="checkbox"/>	Approved	<input type="button" value="Edit"/>
3		Pizza	This is an veg pizza.	Veg Pizza	2023-04-25	<input checked="" type="checkbox"/>	Approved	<input type="button" value="Edit"/>

Rows per page: 5 rows ▾ 1-3 of 3 |< < 1-3 of 3 > >|

## 15. Manage Item

### Manage Item

Item Name

Farm House

Item Type

veg

Item Description

Delightful combination of onion, capsicum, tomato & grilled mushroom

+ Upload

 1662197687846-Farmhouse.jpg

#### Add Extra

Title

Topping

Customer Selection

optional

Add Extras

Red Pepper  Select

Red Pepper

Extra Cheese

#### Addons

Title

Crust

Customer Selection

compulsory

Addons

Cheese Burst   100% Wheat Thin Crust  Select

#### Variant

Varinat

Regular(269)   Medium(579)   Large(839)  Select

#### Category

Category

Veg Pizza  Select

Veg Pizza

## 16. Banner

Add Banner

Banner							
S.No #	Restaurant	BannerName	BannerImage	Added On	Banner Status	Approval Status	Actions
1	Dominos	Banner 1		2023-01-26		Approved	<button>Edit</button>
2	Dominos	Banner 2		2023-01-26		Approved	<button>Edit</button>
3	Dominos	Banner 3		2023-01-26		Approved	<button>Edit</button>

## 17. Manage Banner

Manage Banner

Name

Banner Image  
  


 /uploads/banner/images/1666969651964-slider\_1.jpg

Submit

## 18. Get Touch

Get Touch

Subject

Message

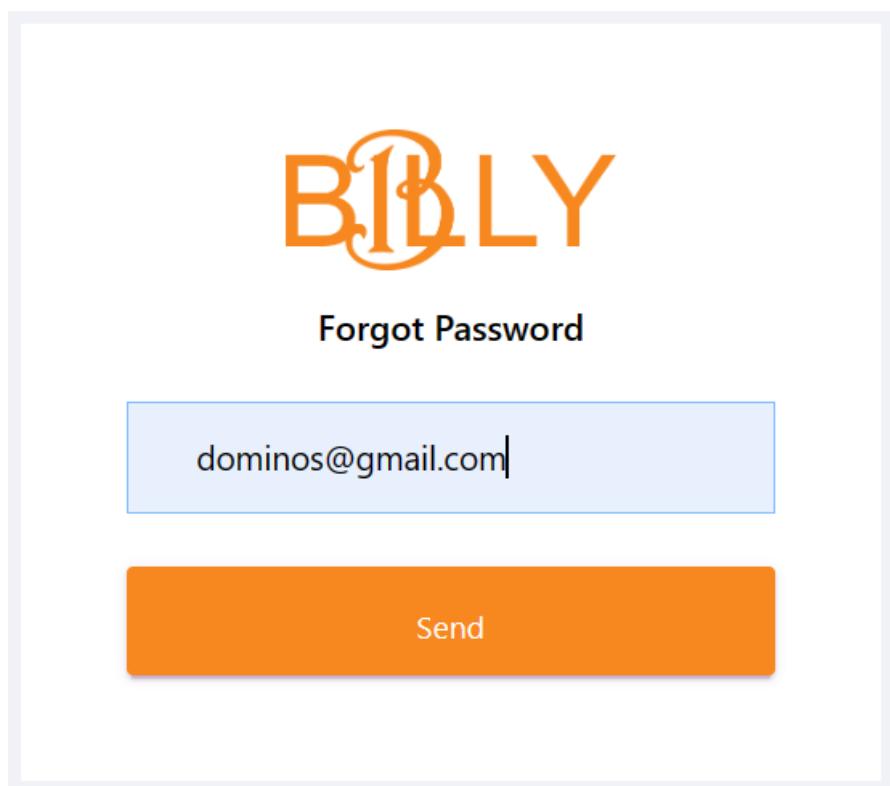
Submit

## 19. Setting

Setting

Cart Min Price	60
Cart Min Price Message	Cart Price is Fixed 60 Rs
Delivery Charge	20
Gst	3

## 20. Forgot Password



#### 7.1.4 Simulator (iPhone 14 Pro and iPhone SE)



## 7.1.5 Customer Portal

### 1. Splash Screen

**IOS**

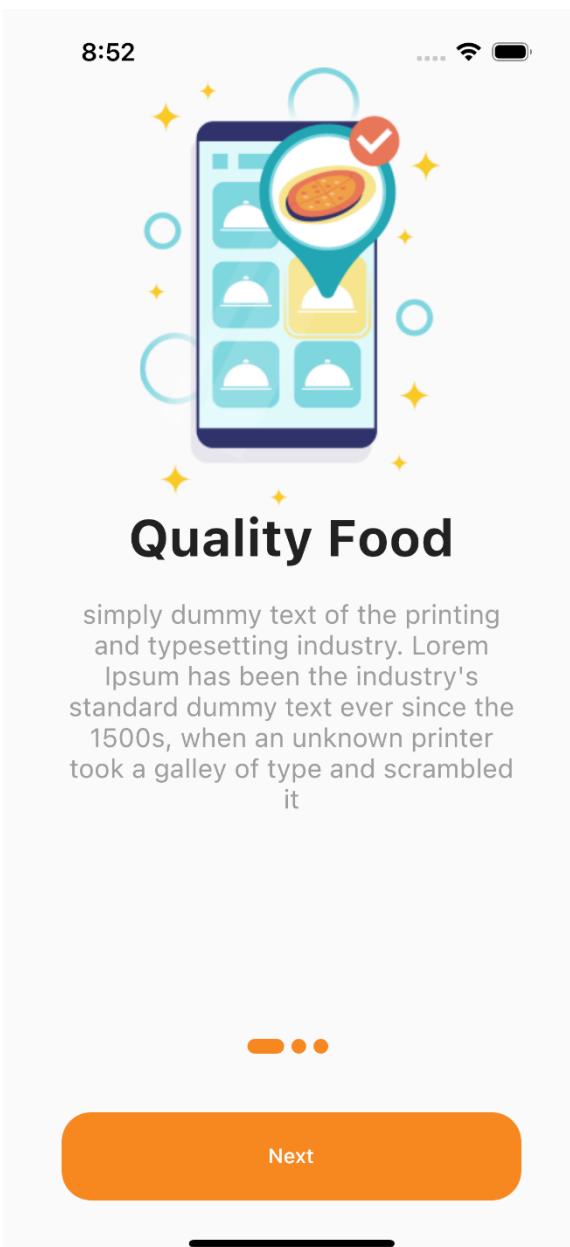


**Android**

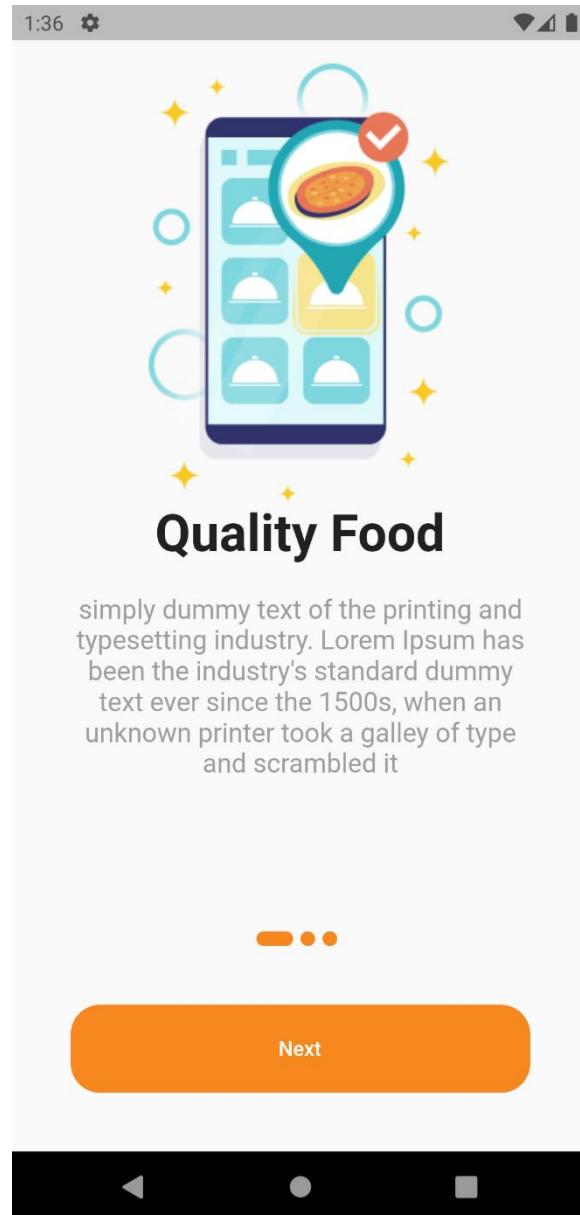


## 2. Onboard Screen (First Screen)

**IOS**

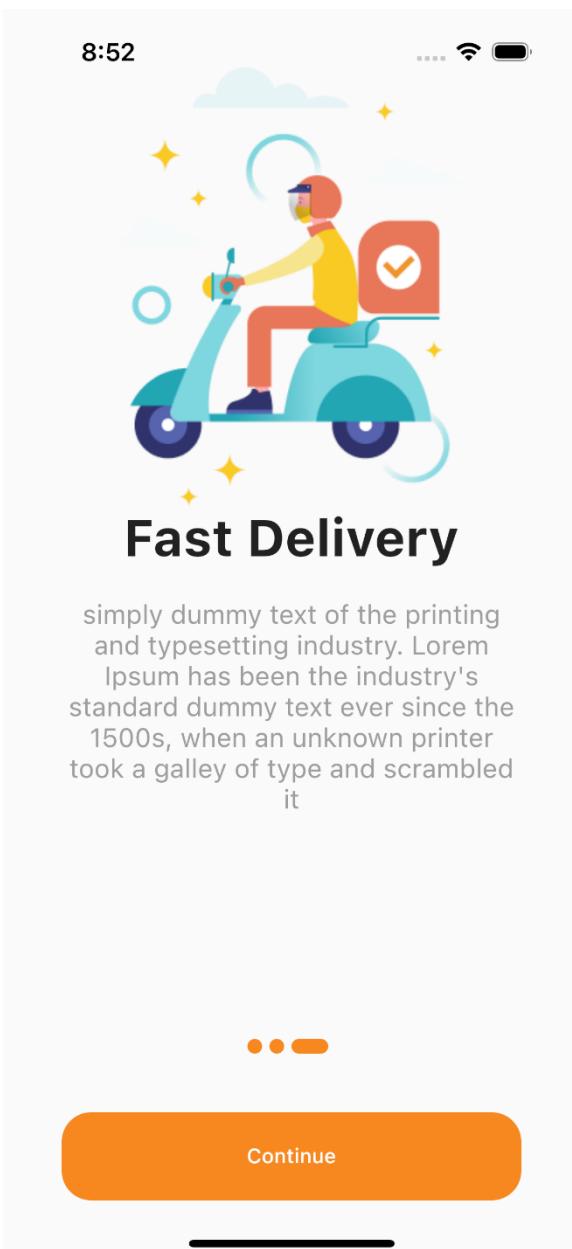


**Android**

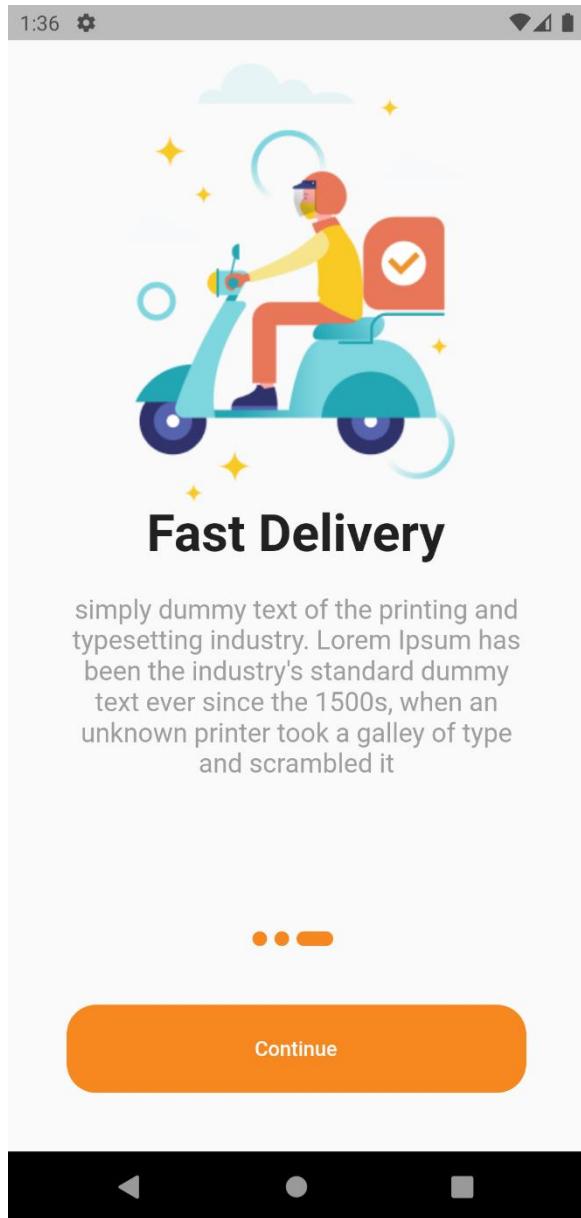


### 3. Onboard Screen (Last Screen)

**IOS**



**Android**



#### 4. Sign Up

IOS

9:02



Abhishek Choksi

19bmiit032@gmail.com

\*\*\* Abhishek#7874



.....



7874376579

Sign Up

Have an account already?

Sign up using one of the following methods



Android

1:40



Abhishek Choksi

19bmiit032@gmail.com

.....



Abhishek#7874



7874376579

Sign Up

Have an account already?

Sign up using one of the following methods



5. Sign In

IOS

7:54



# Hello

Sign into your account

7874376579

\*\*\* ..... |



Sign into your account

Sign In

or

Generate OTP

Don't have an account? [Create](#)

Android

1:41



# Hello

Sign into your account

7874376579

\*\*\* ..... |



Sign into your account

Sign In

or

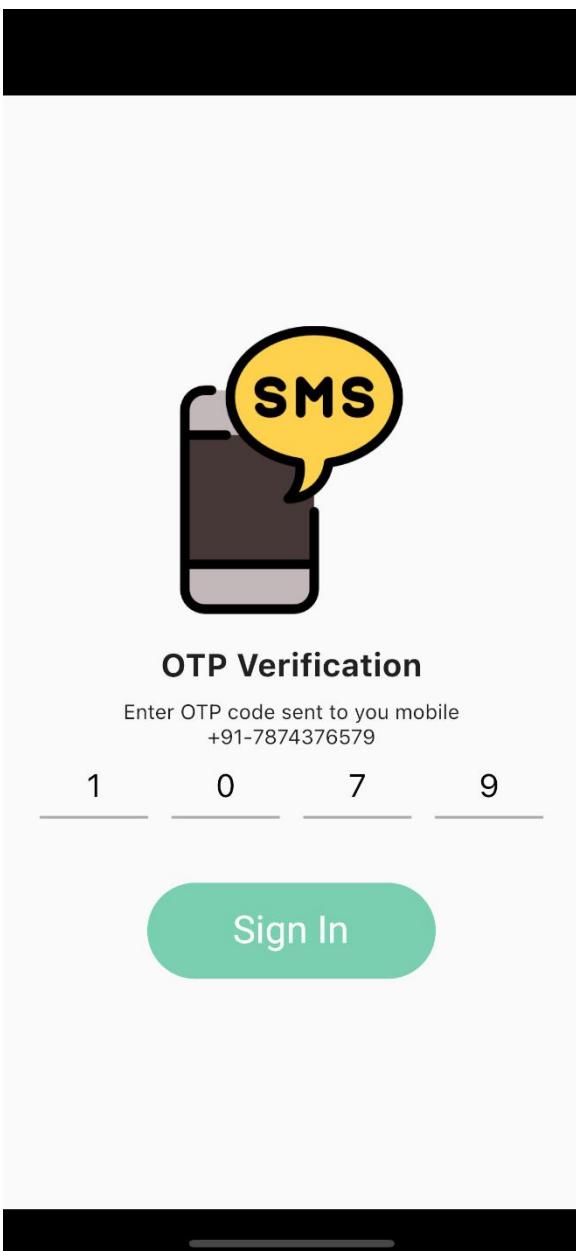
Generate OTP

Don't have an account? [Create](#)

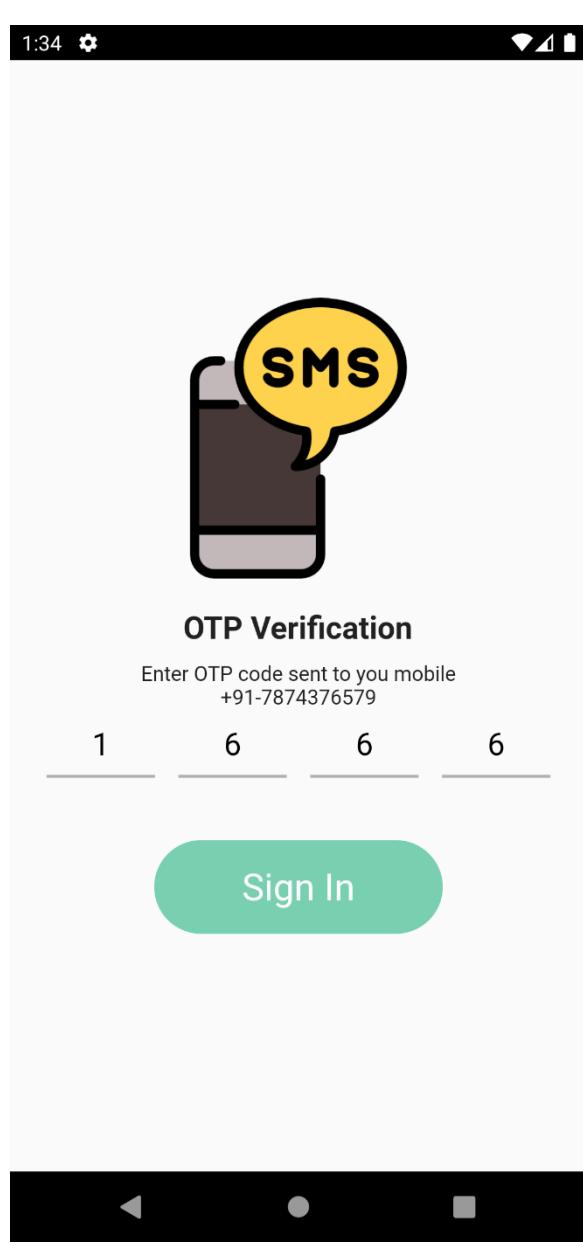


6. Sign In with OTP

IOS

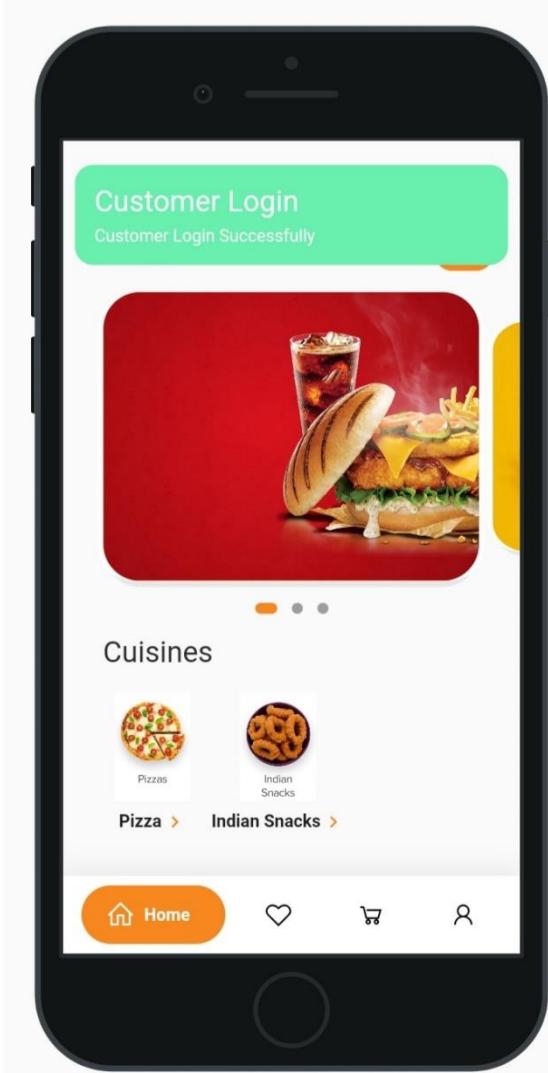


Android

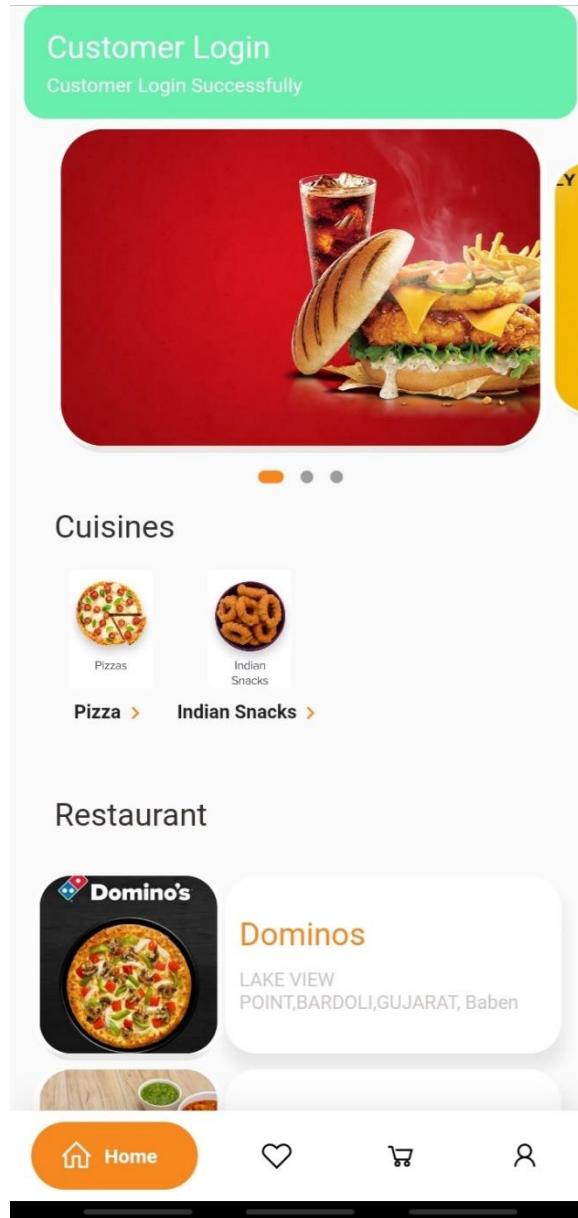


## 7. Sign In Message

**IOS**

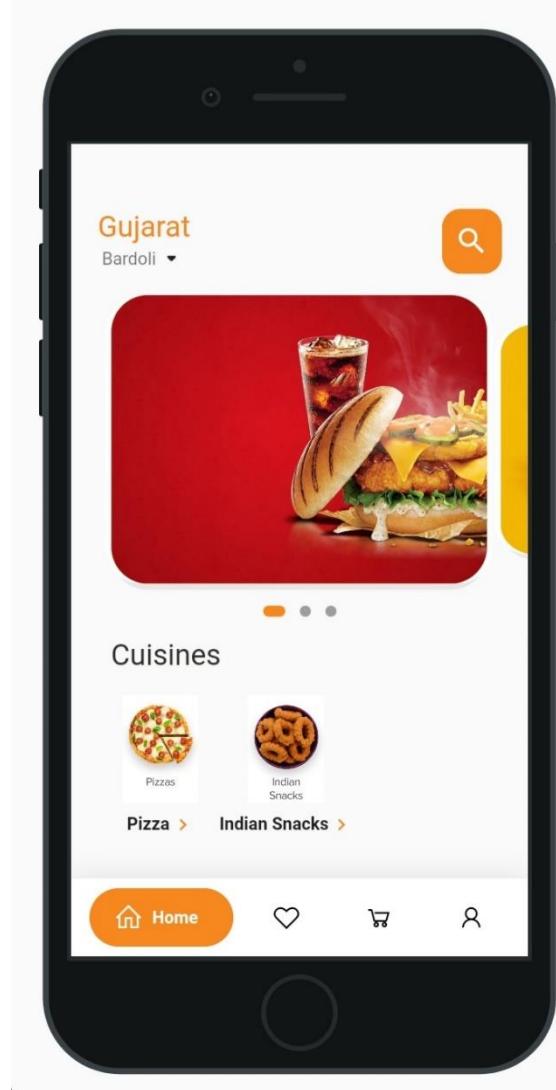


**Android**

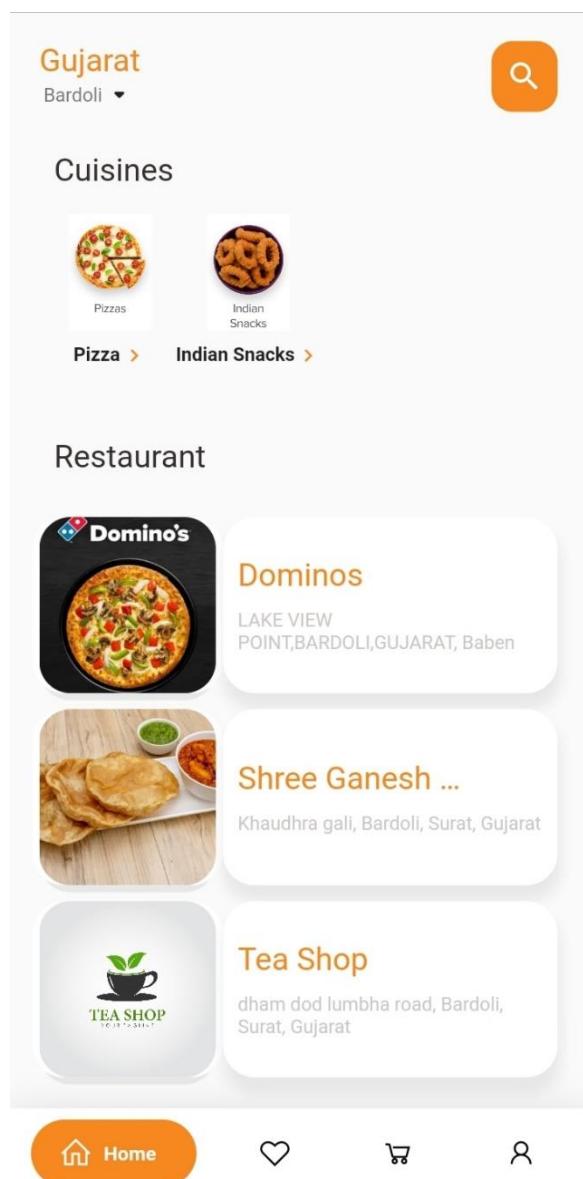


## 8. Home

**IOS**

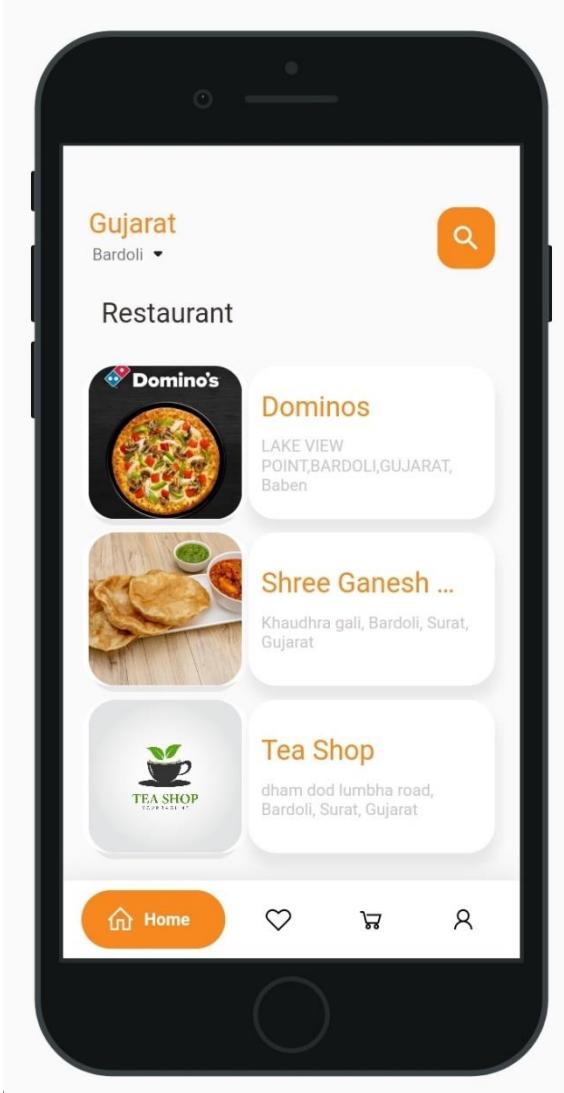


**Android**

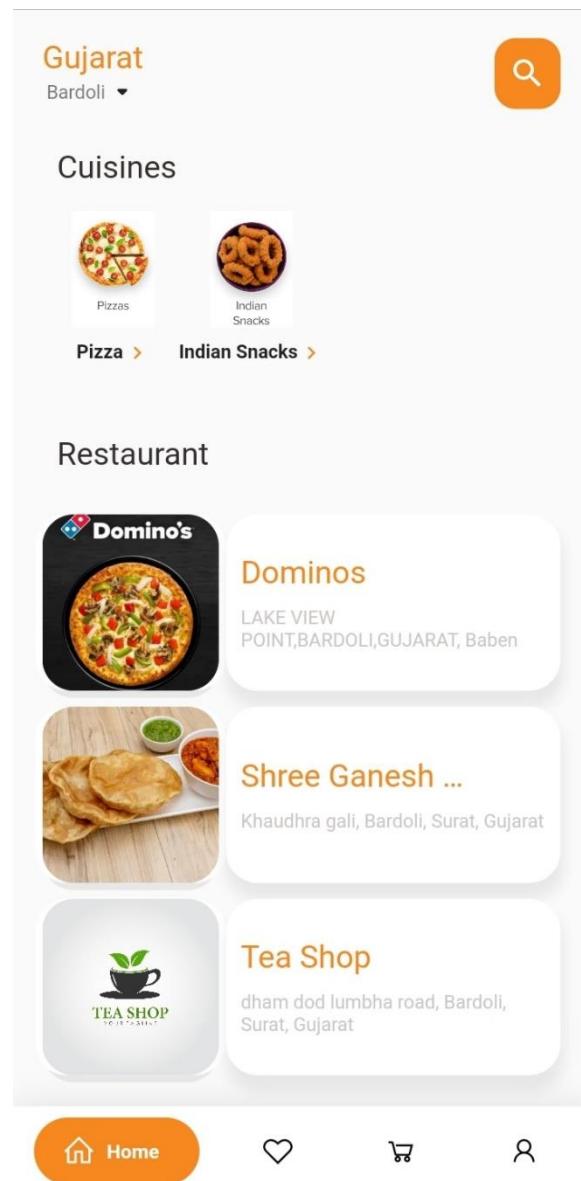


## 9. Restaurant

**IOS**



**Android**



## 10. Cuisines

### IOS



### Android



### Pizza

pizza, dish of Italian origin consisting of a flattened disk of bread dough topped with some combination of olive oil, oregano, tomato, olives, mozzarella or other cheese, and many other ingredients, baked quickly—usually, in a commercial setting, using a wood-fired oven heated to a very high temperature

Show more ▾

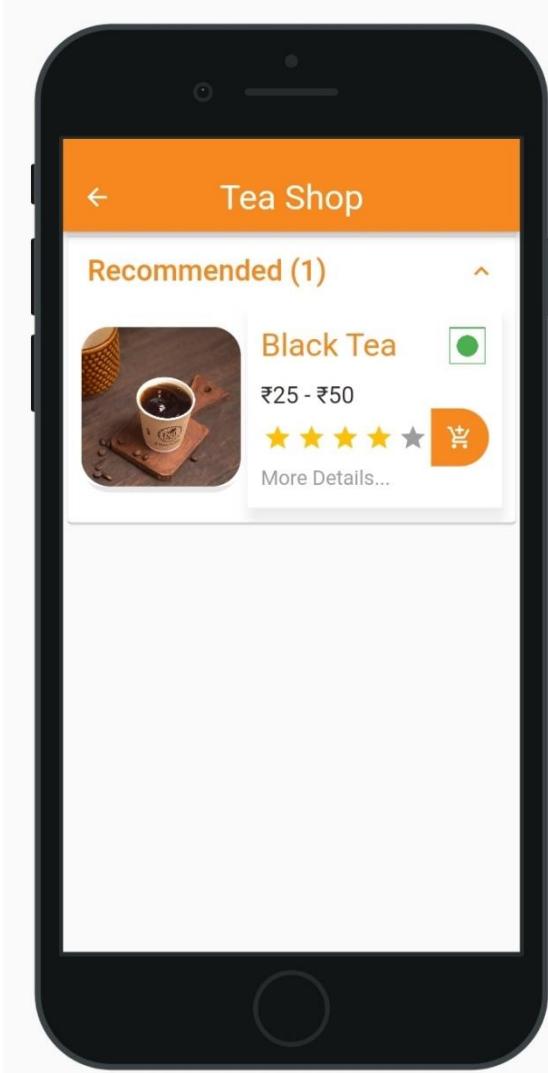


### Domino's

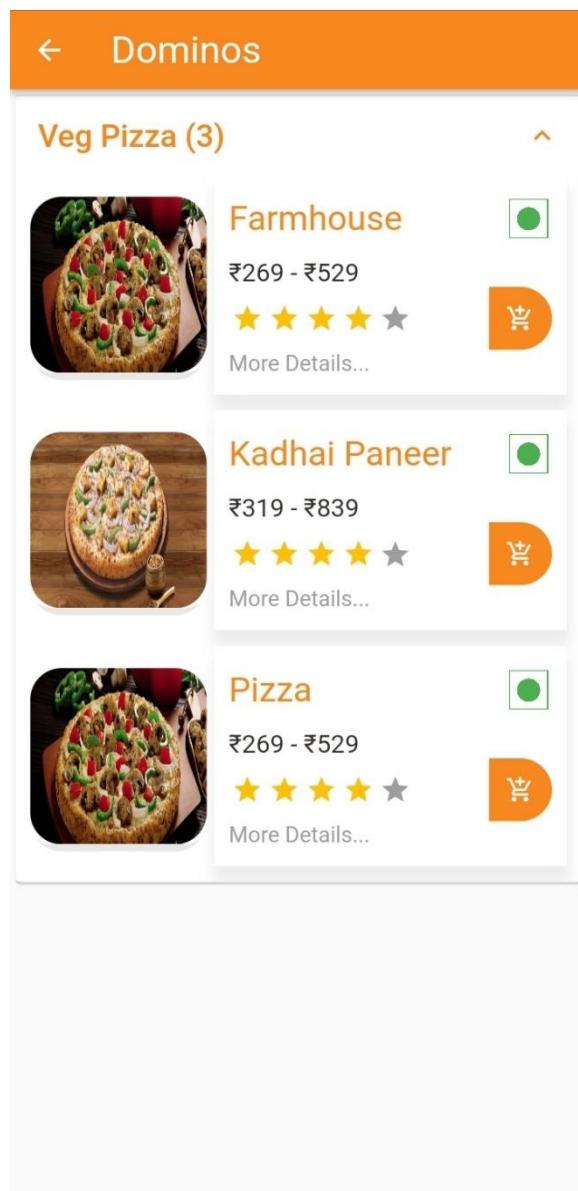
LAKE VIEW POINT,BARDOLI,GUJARAT,  
Baben

## 11. Item

IOS

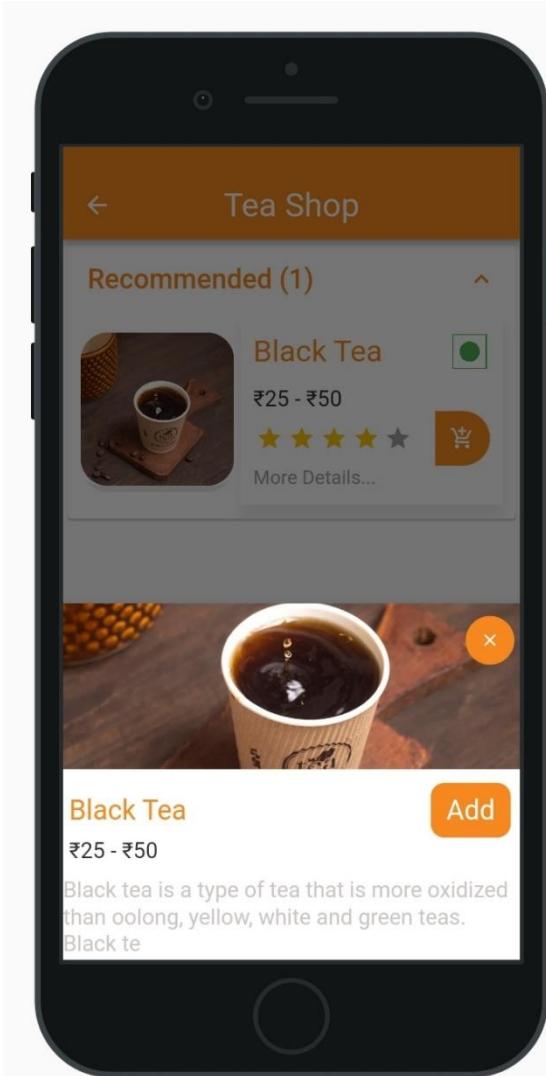


Android

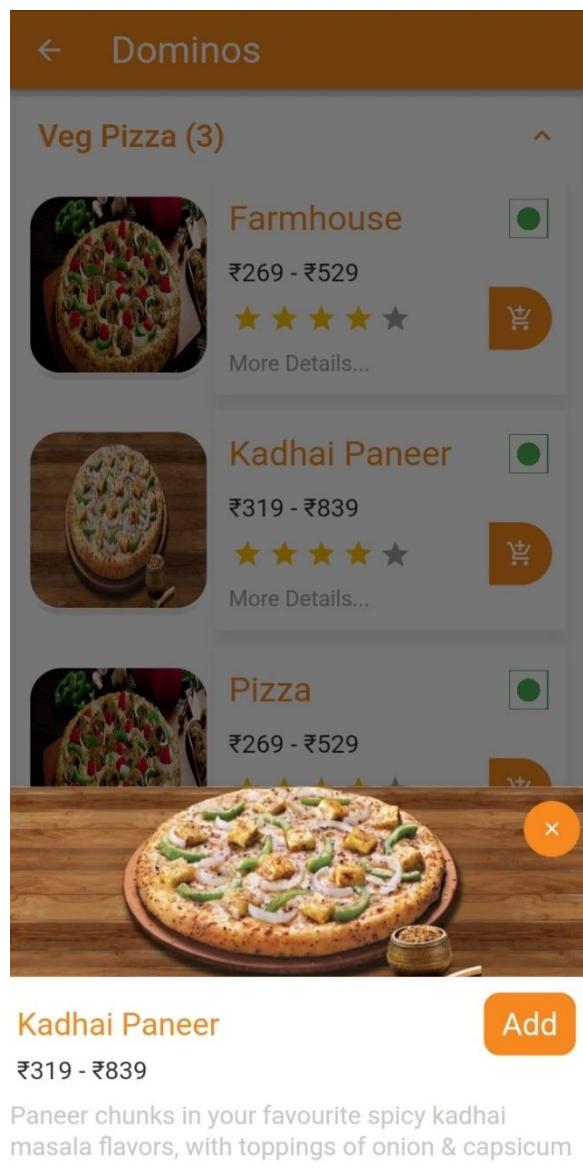


## 12. Item Details

IOS

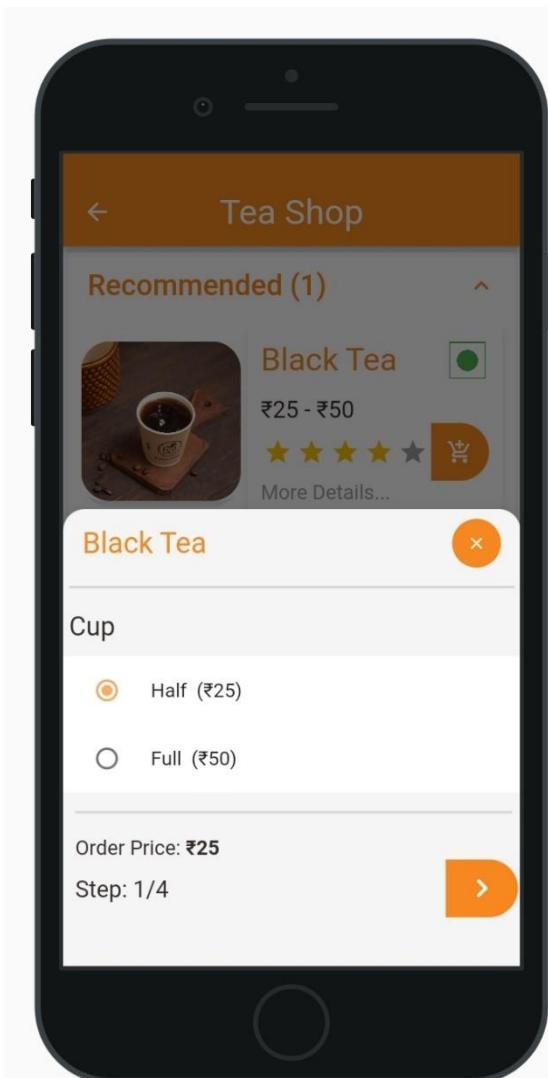


Android

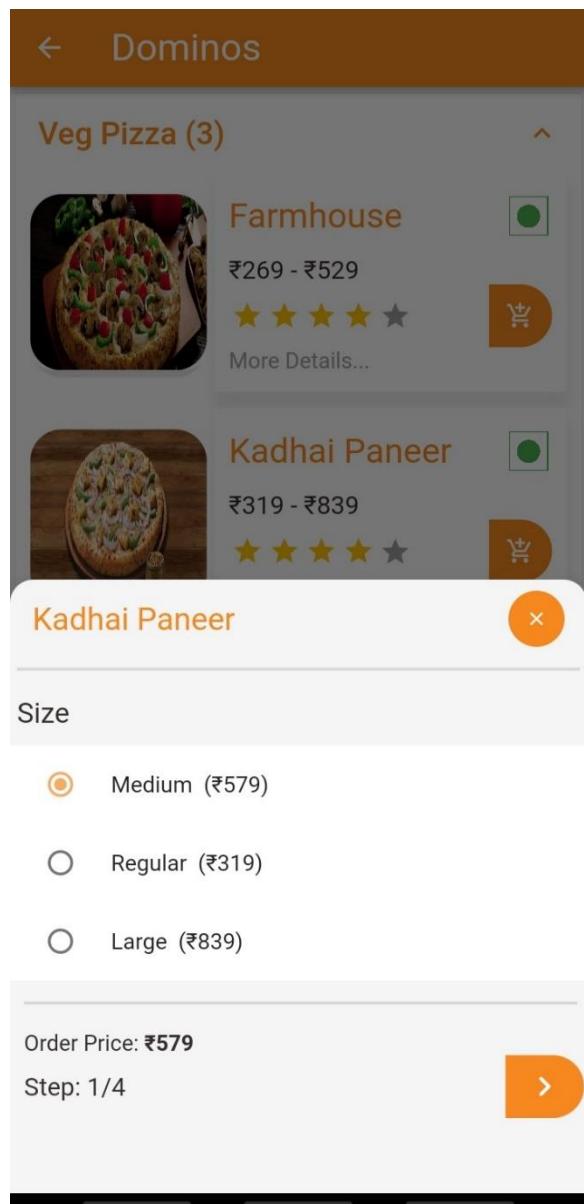


### 13. Item Variant

IOS

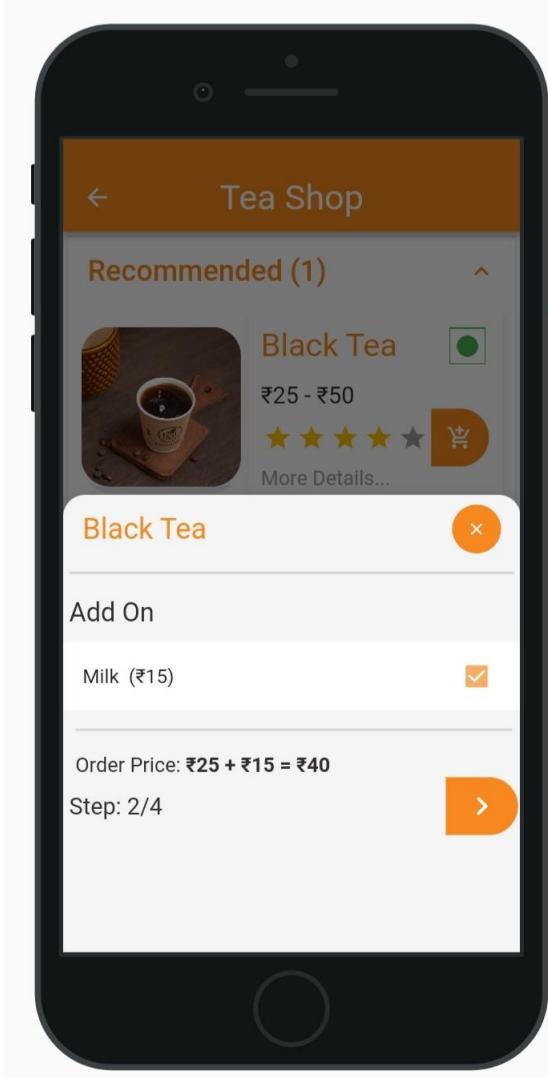


Android

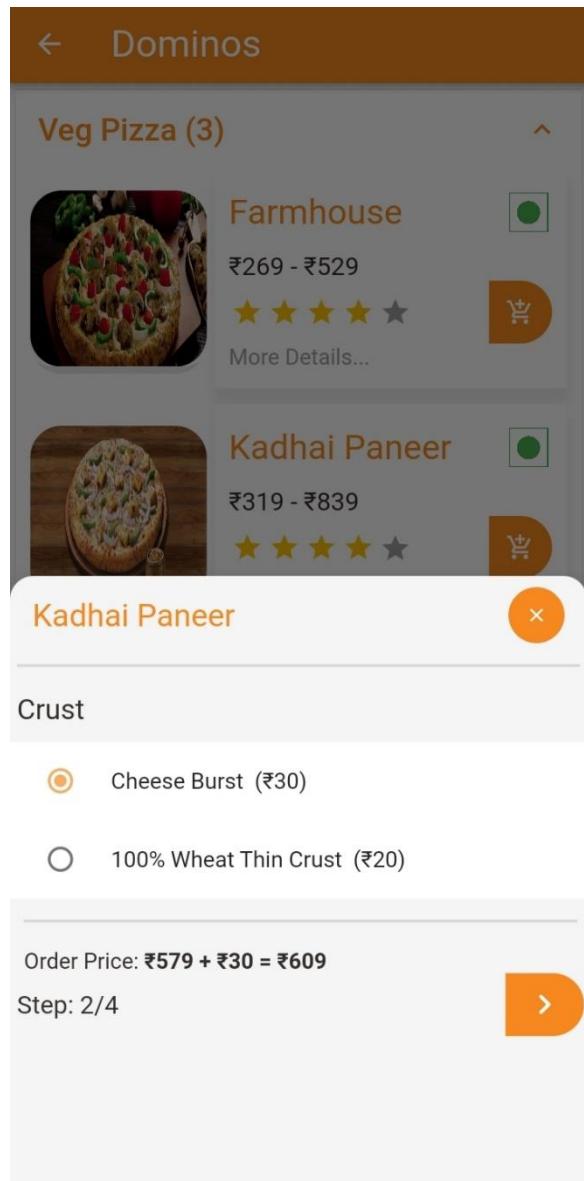


#### 14. Item Add On

IOS

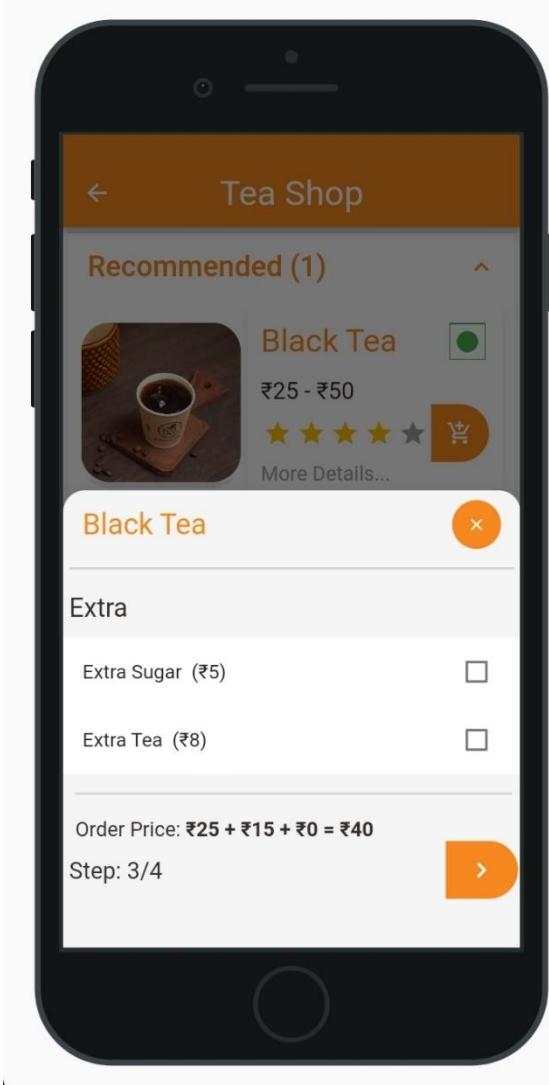


Android

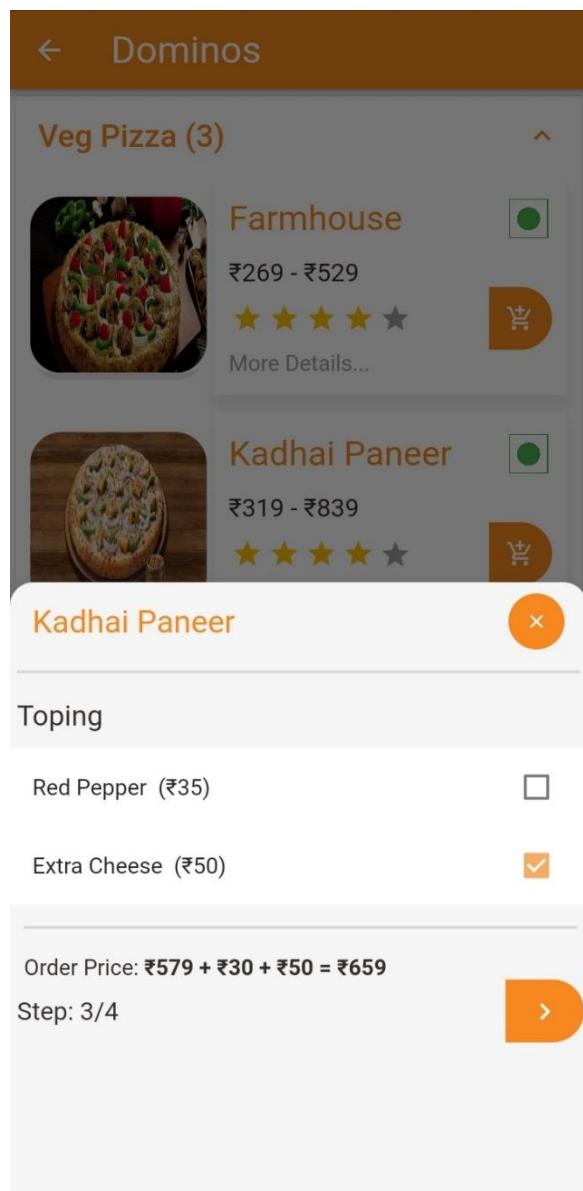


## 15. Item Add Extra

IOS

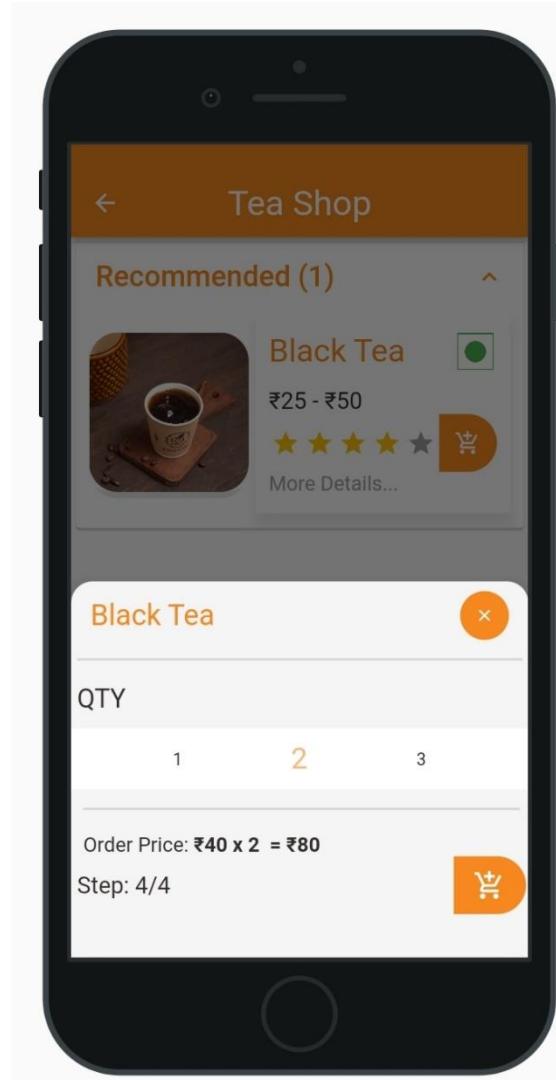


Android

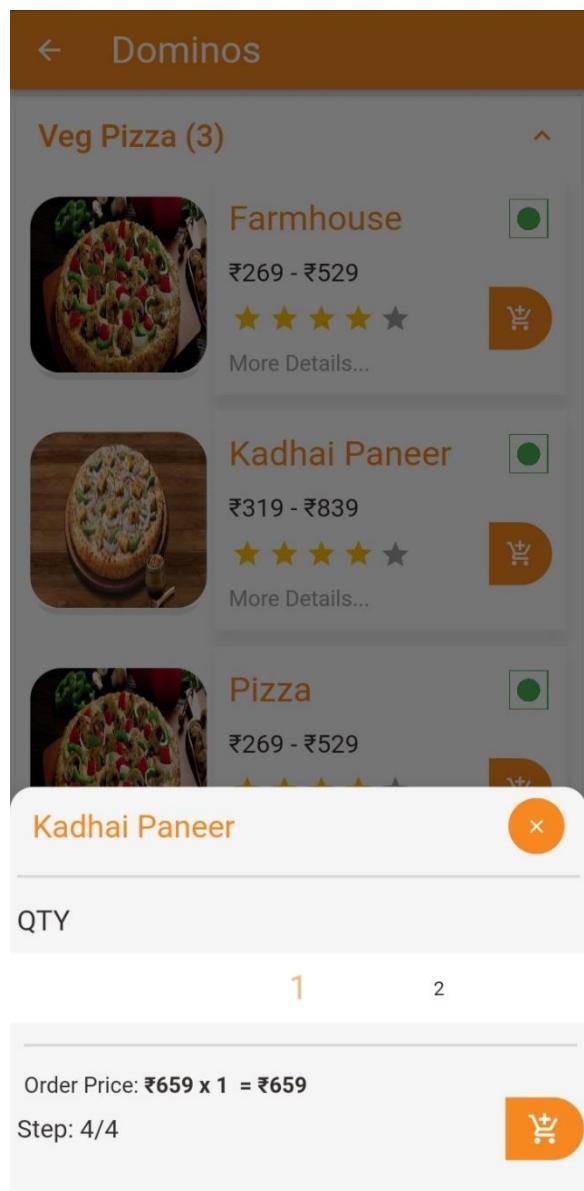


## 16. Item Qty

**IOS**

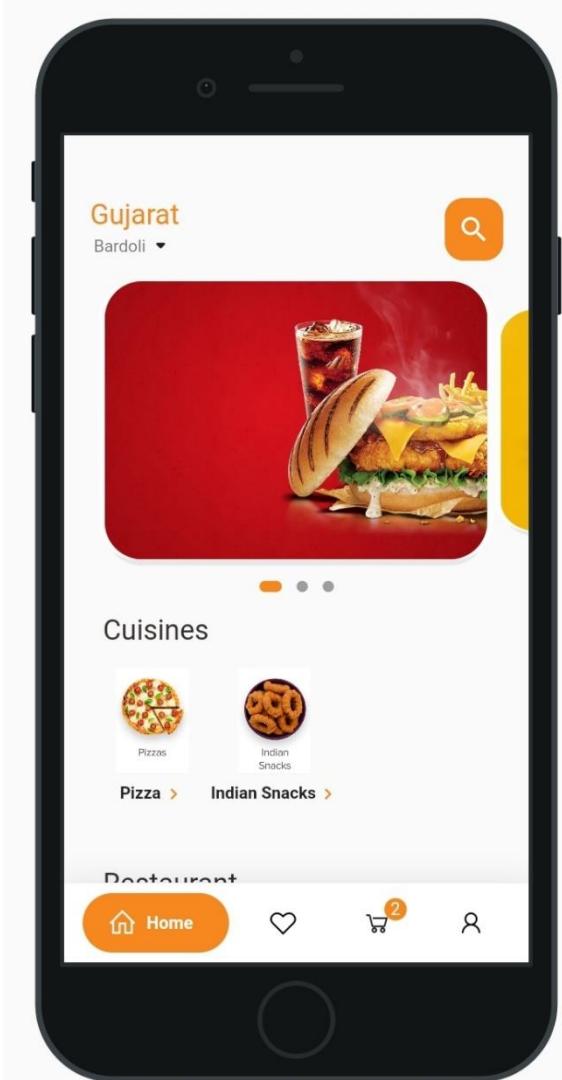


**Android**

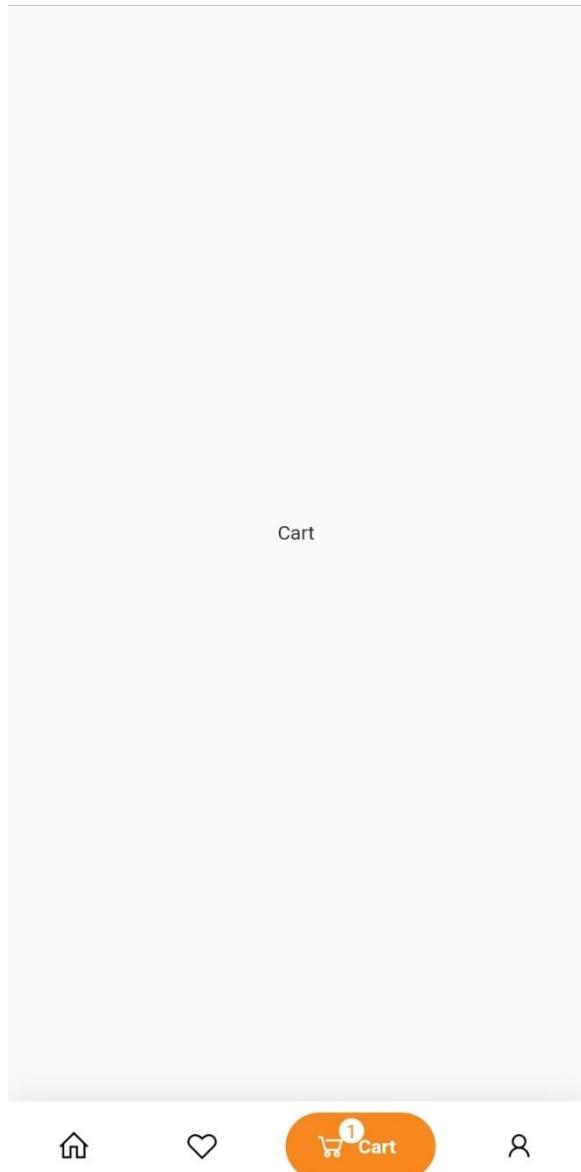


## 17. Cart Badge

**IOS**

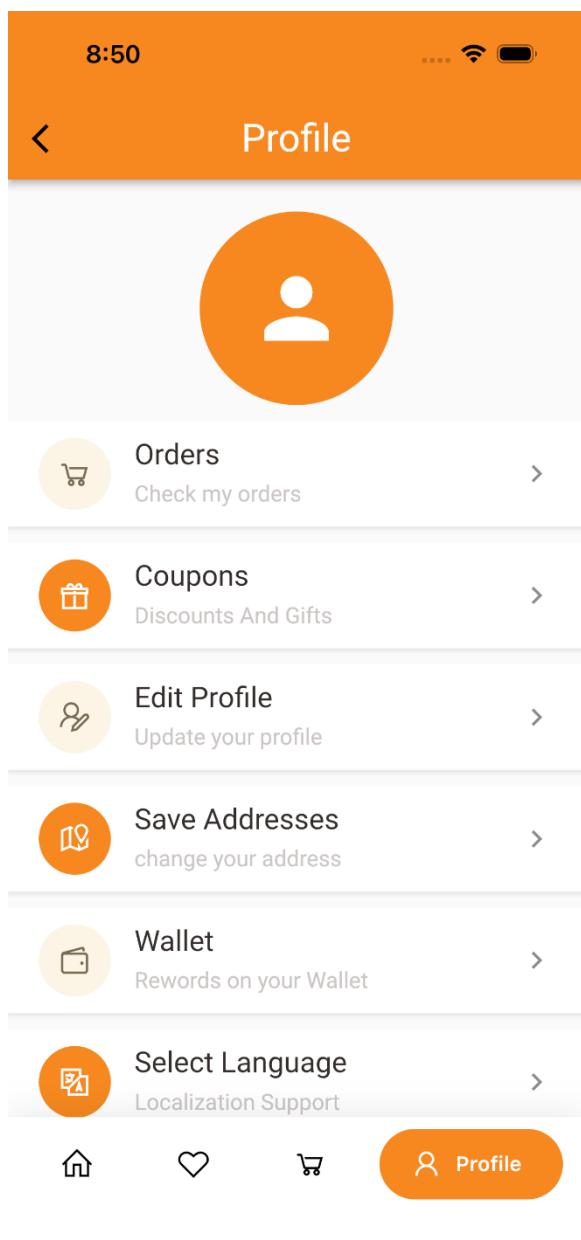


**Android**

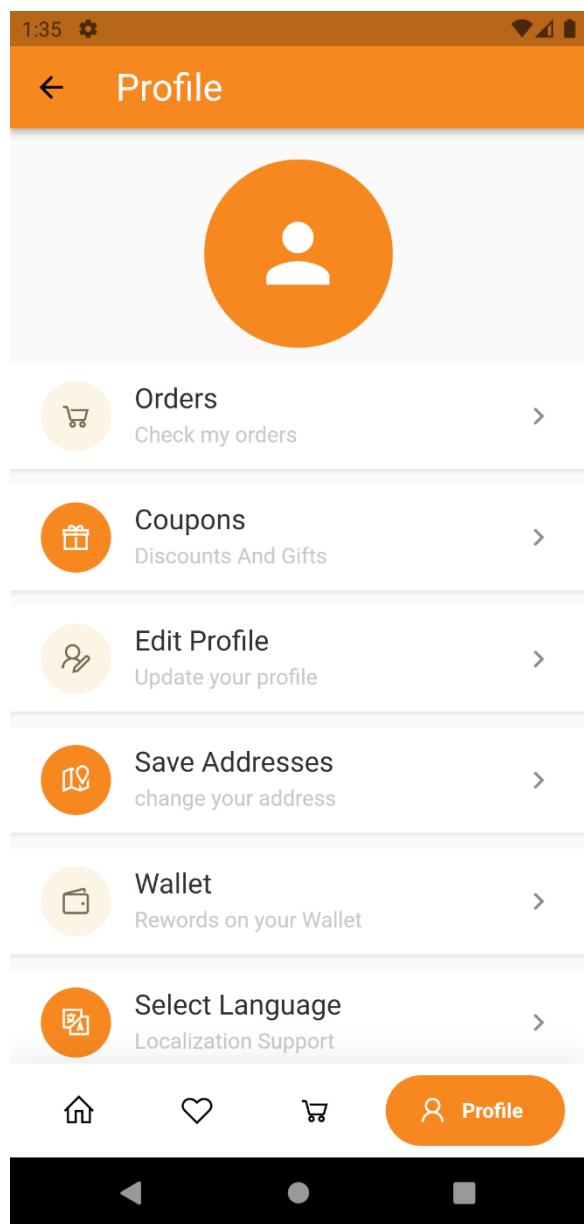


## 18. Profile

IOS

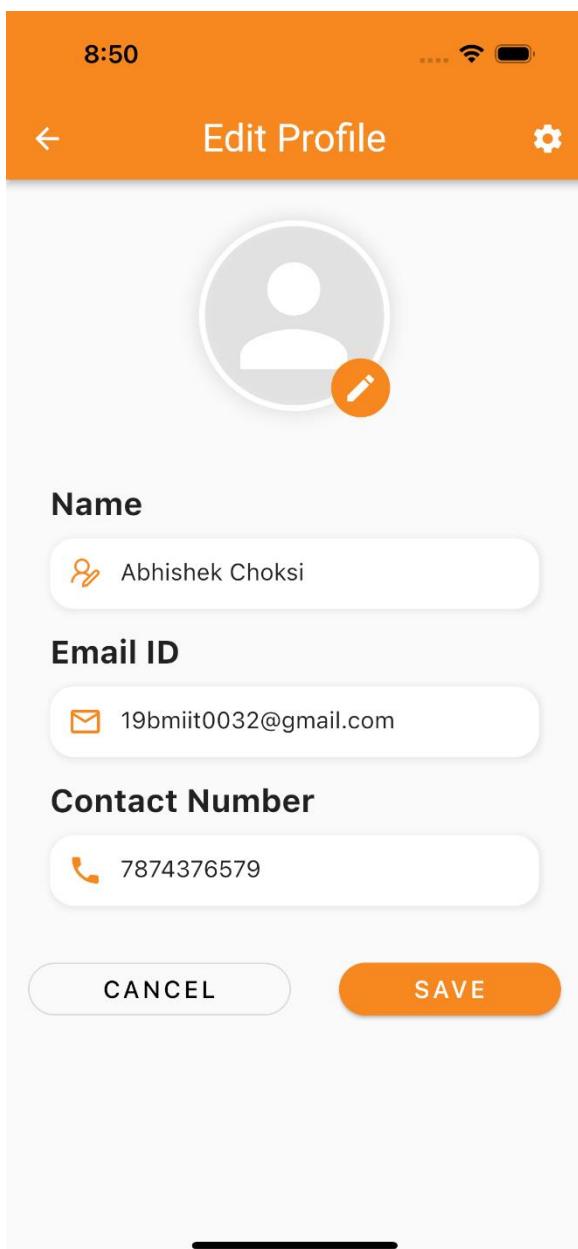


Android

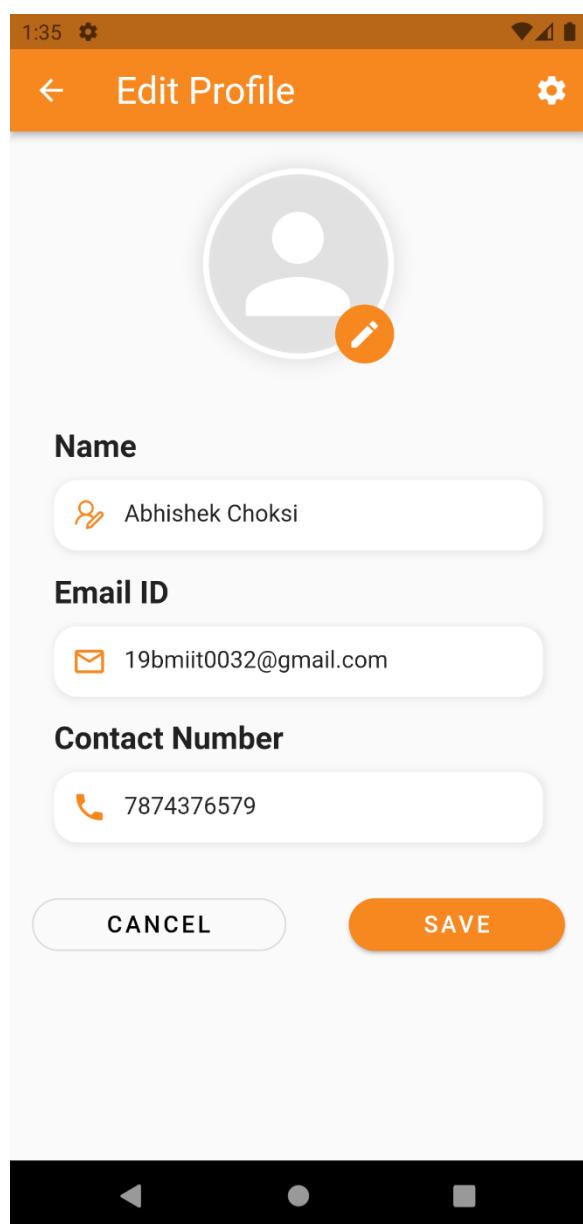


## 19. Edit Profile

IOS

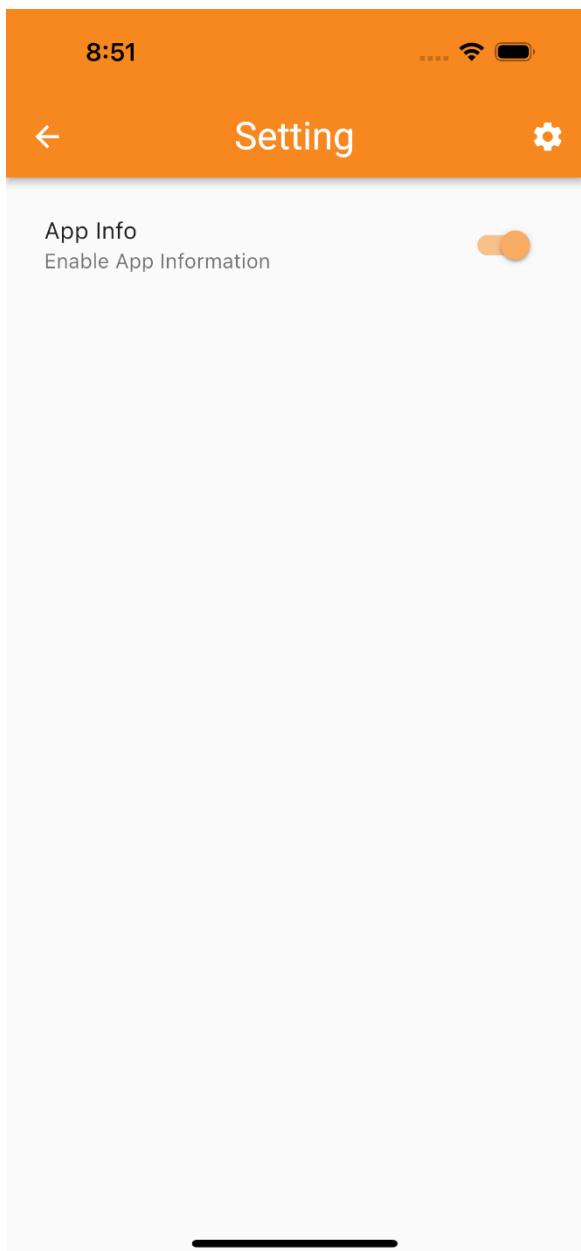


Android

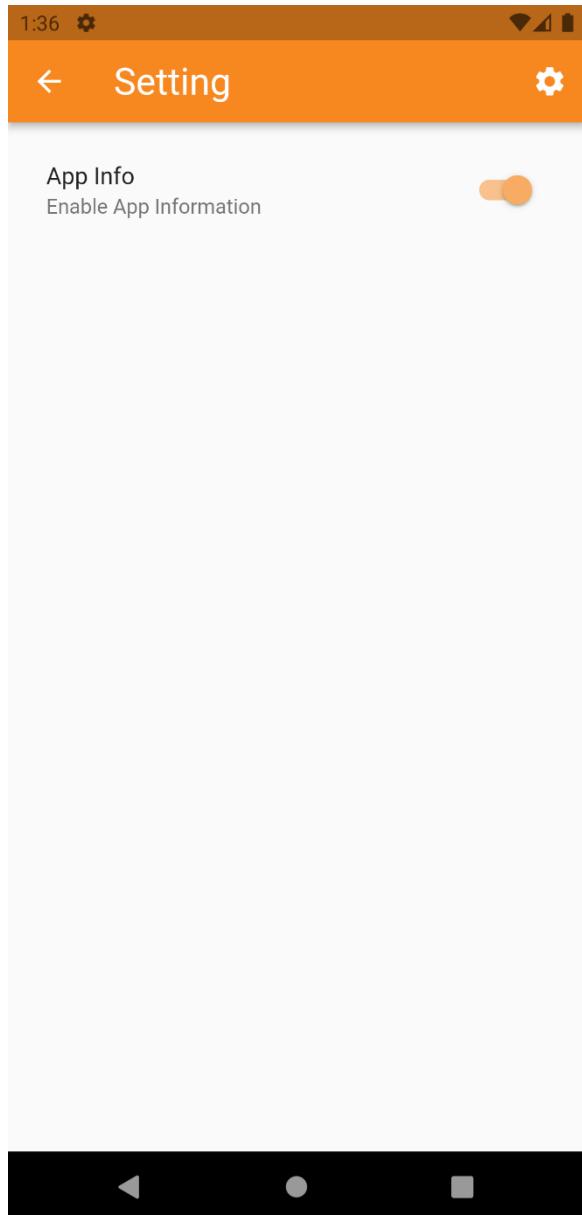


## 20. Setting

**IOS**



**Android**

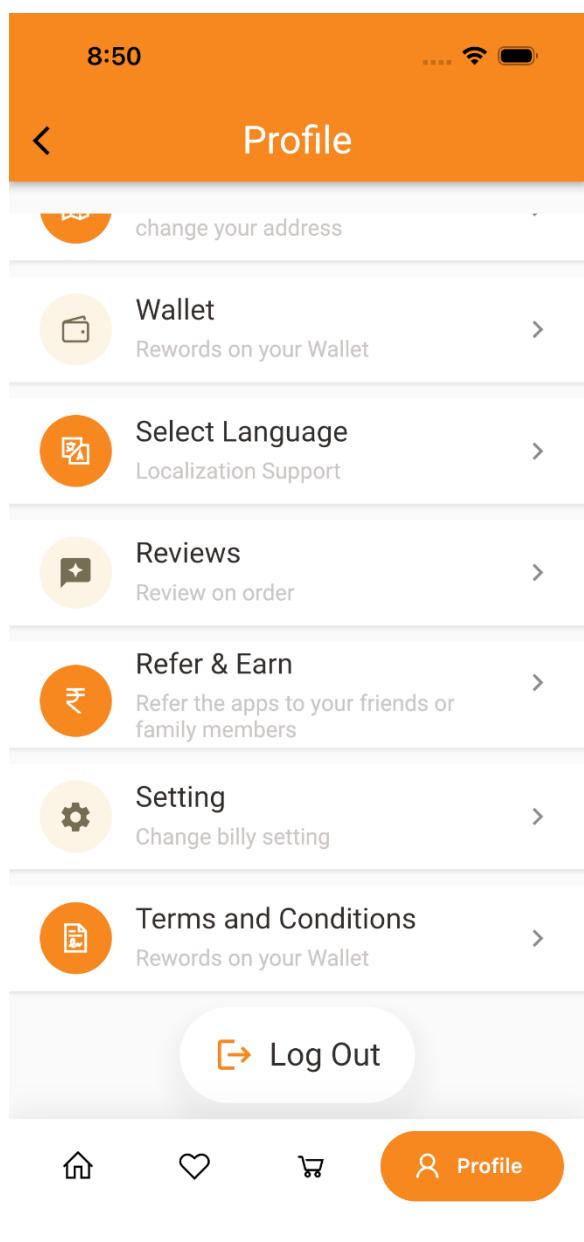


**IOS**

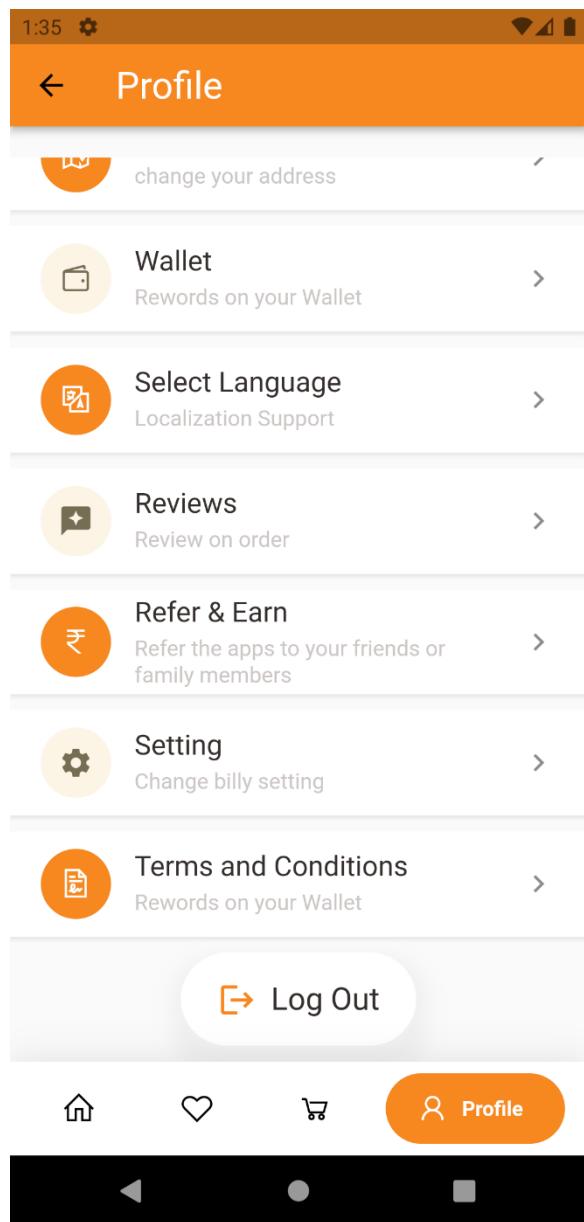
**Android**

## 21. Logout

IOS



Android

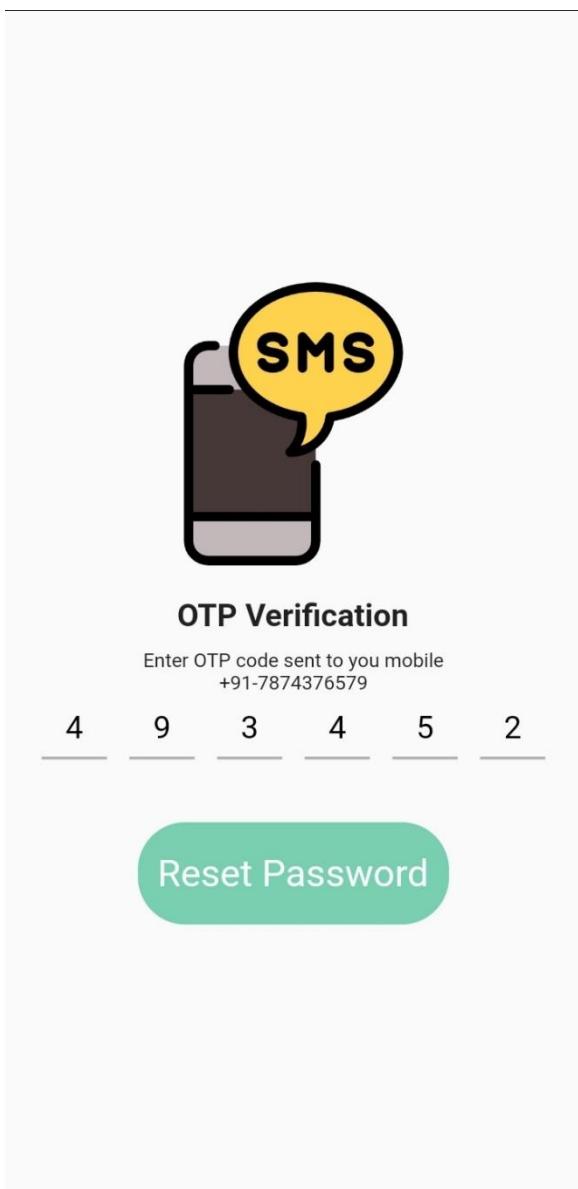


22. Reset Password (Page 1)

**IOS**



**Android**



23. Reset Password (Page 2)

IOS



Android

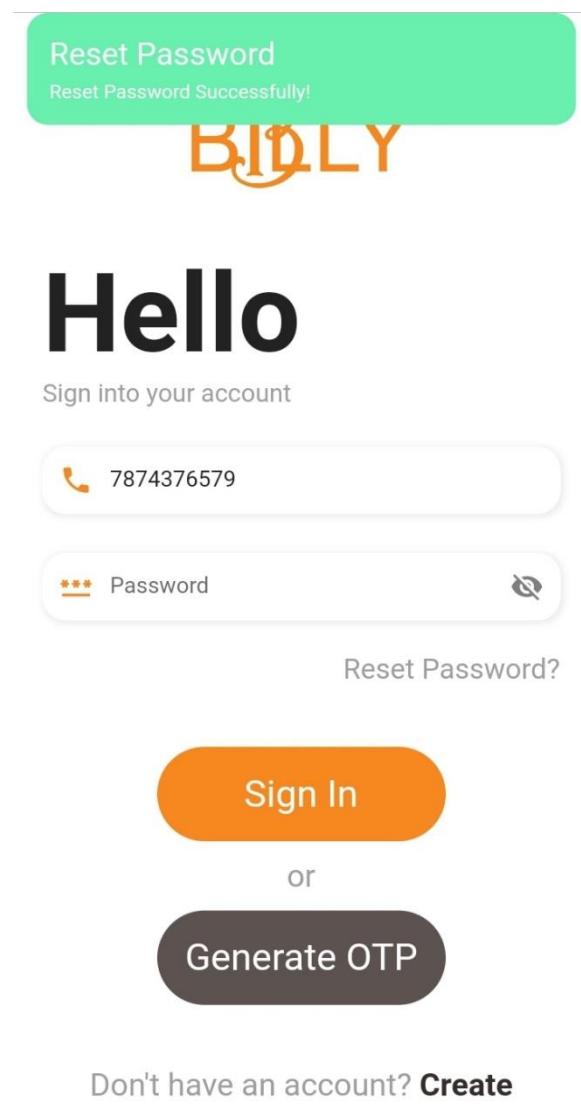


24. Reset Password Notification (Page 3)

IOS



Android



## TEST CASE

### 8.1.1 Test cases for Login

#### 8.1.1.1 Authorized Admin

<u>Test Case Id</u>	<u>Test Case Description</u>	<u>Input-1</u>	<u>Input-2</u>	<u>Expected Result</u>	<u>Remark (if any)</u>
		username	userpassword		
TC <sub>1</sub>	Login Basic Flow	Valid Input	Valid Input	User is allowed to login	-
TC <sub>2</sub>	Login alternative flow: Invalid entry	Invalid Input	Valid Input	username invalid	Please enter valid login details
TC <sub>3</sub>	Login alternative flow: Invalid entry	Valid Input	Valid Input	username invalid	Please enter valid login details
TC <sub>4</sub>	Login alternative flow: Invalid entry	Valid Input	Invalid Input	userpassword invalid	Please enter valid login details
TC <sub>5</sub>	Login alternative flow: Invalid entry	Valid Input	Valid Input	userpassword invalid	Please enter valid login details
TC <sub>6</sub>	Login alternative flow: Invalid entry	Invalid Input	Invalid Input	username and userpassword invalid	Please enter valid login details.
TC <sub>7</sub>	Login alternative flow: User exits	Valid/Invalid Input	Valid/Invalid Input	User comes out of the system	-

### 8.1.1.2 Authorized Delivery Boy

<u>Test Case Id</u>	<u>Test Case Description</u>	<u>Input-1</u>	<u>Input-2</u>	<u>Input-3</u>	<u>Input-4</u>	<u>Input-5</u>	<u>Input-6</u>	<u>Expected Result</u>	<u>Remark (if any)</u>
		mobile no	userpassword	Active/deactiv e	old passwo rd	new passwo rd	Verify email id		
TC <sub>1</sub>	Login Basic Flow	Valid Input	Valid Input	n/a	n/a	n/a	n/a	User is allowed to login	-
TC <sub>2</sub>	Login alternative flow: Invalid entry	Invalid Input	Valid Input	n/a	n/a	n/a	n/a	mobileno invalid	Please enter valid mobileno.
TC <sub>3</sub>	Login alternative flow: Invalid entry	Valid Input	Valid Input	n/a	n/a	n/a	n/a	mobileno invalid	mobileno does not exist in database.
TC <sub>4</sub>	Login alternative flow: Invalid entry	Valid Input	Invalid Input	n/a	n/a	n/a	n/a	userpassword invalid	Please enter correct password.
TC <sub>5</sub>	Login alternative flow: Invalid entry	Valid Input	Valid Input	n/a	n/a	n/a	n/a	userpassword invalid	userpassword does not exist in database.
TC <sub>6</sub>	Login alternative flow: Invalid entry	Valid Input	Valid Input	n/a	n/a	n/a	Valid Input	User is allowed to login	-
TC <sub>7</sub>	Login alternative flow: Invalid entry	Valid Input	valid Input	n/a	n/a	n/a	Invalid Input	mobileno not verified	Please verify your mobileno.

TC <sub>8</sub>	Login alternative flow: User activated	Valid Input	Valid Input	Valid Input	n/a	n/a	Invalid Input	User is allowed to login	-
TC <sub>9</sub>	Login alternative flow: User deactivated	Valid Input	Valid Input	Invalid Input	n/a	n/a	Invalid Input	User is deactivated allowed to login	Your account has been deactivated.
TC <sub>10</sub>	Login alternative flow: Invalid entry	Invalid Input	Invalid Input	n/a	n/a	n/a	n/a	mobilenumber and userpassword invalid	mobilenumber and userpassword is not in specified format.
TC <sub>11</sub>	Login alternative flow: User exits	Valid/Invalid Input	Valid/Invalid Input	n/a	n/a	n/a	n/a	User comes out of the system	-
TC <sub>12</sub>	Change password	Valid Input	n/a	Valid Input	Valid Input	Valid Input	Valid Input	User is allowed to change password	Password is changed in the database.
TC <sub>13</sub>	Change password alternative flow: Invalid entry	Invalid Input	n/a	n/a	Valid/Invalid Input	Valid/Invalid Input	Valid Input	mobilenumber invalid	Please enter valid mobilenumber.
TC <sub>14</sub>	Change password alternative flow: Invalid entry	Valid Input	n/a	n/a	Invalid Input	Valid/Invalid Input	Invalid Input	mobilenumber not verified	Please verify your mobilenumber.
TC <sub>15</sub>	Change password alternative flow: Invalid entry	Valid Input	n/a	Invalid Input	Valid Input	Valid Input	n/a	User is not activated	Your account has been deactivated.

TC <sub>16</sub>	Change password	Valid Input	n/a	Valid Input	Valid Input	Valid Input	Valid Input	Userpassword reset	Password has been reset and send it to your mobileno
TC <sub>17</sub>	Change password alternative flow: User exits	Valid/Invalid Input	n/a	n/a	Valid/Invalid Input	Valid/Invalid Input	Valid/Invalid Input	User is allowed to exit and returns to login screen	-

### 8.1.2 Test cases for Manage

#### 8.1.2.1 Manage Delivery Boy

<u>Test Case Id</u>	<u>Test Case Description</u>	<u>Input-1</u>	<u>Input-2</u>	<u>Input-3</u>	<u>Input-4</u>	<u>Input-5</u>	<u>Expected Result</u>	<u>Remark (if any)</u>
		Id	name	mobile	password	status		
TC <sub>1</sub>	Add a Delivery Boy	Valid Input	Valid Input	Valid Input	Valid Input	Valid Input	Delivery Boy is added successfully	-
TC <sub>2</sub>	Name is invalid	Valid Input	Invalid Input	Valid / Invalid Input	Valid / Invalid Input	Valid Input	Invalid name	Delivery Boy name is not in specified format.
TC <sub>3</sub>	Mobile No is invalid	Valid Input	Valid Input	Invalid Input	Valid / Invalid Input	Valid Input	Invalid mobile no.	Mobile no is not in specified format.

TC <sub>4</sub>	Mobile No is invalid	Valid Input	Valid Input	Valid Input	Valid / Invalid Input	Valid Input	Invalid mobile no.	Mobile No is already exist in database.
TC <sub>5</sub>	Password is invalid	Valid Input	Valid Input	Valid Input	Invalid Input	Valid Input	Invalid password	Password is not in specified format.
TC <sub>6</sub>	Update a Delivery Boy	Valid Input	Valid Input	Valid Input	Valid Input	Valid Input	Delivery boy is update successfully	-
TC <sub>7</sub>	Update mobile No is invalid	Valid Input	n/a	Invalid Input	n/a	Valid Input	Invalid updated mobile no.	Mobile no is already added.
TC <sub>8</sub>	Search a Delivery Boy	Valid Input	n/a	n/a	n/a	Valid Input	The Delivery Boy details are displayed.	-
TC <sub>9</sub>	Delivery boy is not found	Valid Input	n/a	n/a	n/a	Valid Input	Delivery Boy is not found	Delivery Boy with the specified name not found in the database.

### 8.1.2.2 Manage Menu

<u>Test Case Id</u>	<u>Test Case Description</u>	<u>Input-1</u>	<u>Input-2</u>	<u>Input-3</u>	<u>Input-4</u>	<u>Input-5</u>	<u>Input-6</u>	<u>Input-7</u>	<u>Input-8</u>	<u>Expected Result</u>	<u>Remark (if any)</u>
		<u>Id</u>	<u>category</u>	<u>item</u>	<u>item_detail</u>	<u>image</u>	<u>type</u>	<u>Price</u>	<u>Status</u>		
TC1	Add an Item	Valid Input	Valid Input	Valid Input	Valid Input	Valid Input	Valid Input	Valid Input	Valid Input	Item is added successfully	-
TC2	Category Name is invalid	Valid Input	Invalid Input	Valid / Invalid Input	Valid / Invalid Input	Valid / Invalid Input	Valid / Invalid Input	Valid / Invalid Input	Valid Input	Invalid category	Category not selected.
TC3	Item Name is invalid	Valid Input	Valid Input	Invalid Input	Valid / Invalid Input	Valid / Invalid Input	Valid / Invalid Input	Valid / Invalid Input	Valid Input	Invalid Item Name	Item name is not in specified format.
TC4	Item Name is invalid	Valid Input	Valid Input	Valid Input	Valid / Invalid Input	Valid / Invalid Input	Valid / Invalid Input	Valid / Invalid Input	Valid Input	Invalid Item Name	Item name is already exist in database.
TC5	Item detail is invalid	Valid Input	Valid Input	Valid Input	Invalid Input	Valid/ Invalid Input	Valid / Invalid Input	Valid / Invalid Input	Valid Input	Invalid item detail	Item detail is not in specified format.

TC <sub>6</sub>	Item image is invalid	Valid Input	Valid Input	Valid Input	Valid Input	Invalid Input	Valid / Invalid Input	Valid / Invalid Input	Valid Input	Invalid item image	Item image is not in specified format. (jpeg, png)
TC <sub>7</sub>	Item price is invalid	Valid Input	Valid / Invalid Input	Invalid Input	Valid Input	Invalid item image	Item price is not in specified format.				
TC <sub>8</sub>	Update an Item	Valid Input	Valid Input	Valid Input	Valid Input	Item is update successfully	-				
TC <sub>9</sub>	Delete an Item	Valid Input	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Item is deleted successfully	-
TC <sub>10</sub>	Search an Item	Valid Input	n/a	n/a	n/a	n/a	n/a	n/a	n/a	The Item details are displayed.	-
TC <sub>11</sub>	Item is not found	Valid Input	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Item is not found	Item with the specified name not found in the database.

## **Future Enhancement**

- This module can be modified if the future according to the need of the organization.
- For future enhancement, we try to track the customer order and the customer's live location is automated fetch by using an AI system.
- And also, we will try to use the QR Code for the coupon code. So, the customer easily scanned the coupon code.

## **Conclusion**

A billy is proposed, useful in small family-run restaurants. This also helps restaurant owners to develop healthy customer relationships by providing reasonably good service.

## **Bibliography**

- Website Referred:
  - [Order food online from India's best food delivery service. Order from restaurants near you \(swiggy.com\)](#)
  - [Stack Overflow - Where Developers Learn, Share, & Build Careers](#)
  - [La Pino'z Pizza](#)
  - [Flutter Dev Document](#)
  - [Pub Dev \(Flutter Packages\)](#)
- Video Referred:
  - [Register on Swiggy || Add restaurant || रेस्टोरेंट SWIGGY पे कैसे लाये || Hindi || March 2021](#)
  - [How to Edit Menu on Swiggy|Add items,Change Price,Add items Photo,Delete items|Step-By-Step|Swiggy](#)
  - [How To Order Food From Swiggy App | Swiggy App Se Khana Order Kaise Karen | GagTech](#)
  - [React and node js project in Hindi \(MERN stack\)](#)
  - [Getting Started with MERN Stack Application](#)