GIKI EATS

……………………..,

…………………………….

Date: 02/07/2020

Supervisor: Ahsan shah Group Members:

Annas Ali 2017074

Omer Munam 2017219

Muhammad Kashif 2017287

Saif Ur Rehman 2017403

Co-Supervisor:

Revision History:

|  |  |  |
| --- | --- | --- |
| ***Revision History*** | ***Date*** | ***Comments*** |
| 1.00 |  |  |
| 2.00 |  |  |
|  |  |  |

Document Approval:

The following document has been accepted and approved by the following:

|  |  |  |
| --- | --- | --- |
| ***Signature*** | ***Date*** | ***Name*** |
|  |  |  |
|  |  |  |
|  |  |  |

List of Contents

[1. INTRODUCTION 6](#_Toc44626968)

[1.1. PURPOSE 6](#_Toc44626969)

[1.2. PRODUCT SCOPE 6](#_Toc44626970)

[1.3. OVERVIEW 7](#_Toc44626971)

[2. THE OVERALL DESCRIPTION 8](#_Toc44626972)

[2.1. PRODUCT PERSPECTIVE 8](#_Toc44626973)

[3. WORK BREAKDOWN STRUCTURE 9](#_Toc44626974)

[4. Design 10](#_Toc44626975)

[4.1 ARCHITECTURAL DESIGN 10](#_Toc44626976)

[4.2 Why we choose MVC Architecture Design? 10](#_Toc44626977)

[4.2.1 Components 11](#_Toc44626978)

[4.3 MODULE IDENTIFICATION 12](#_Toc44626979)

[5. 4+1 ARCHITECTURE VIEW MODEL 13](#_Toc44626980)

[5.1 Use Case View 14](#_Toc44626982)

[5.1.1 System Use case 14](#_Toc44626983)

[5.1.2 Login and Registration Use Case 15](#_Toc44626984)

[5.1.3 Ordering Process use case 16](#_Toc44626985)

[5.2 Logical View: 17](#_Toc44626986)

[5.2.1 Class Diagram 17](#_Toc44626987)

[5.3 Development View 18](#_Toc44626988)

[5.3.1 Component Diagram 18](#_Toc44626989)

[5.4 Process View 19](#_Toc44626990)

[5.4.1 Restaurant Side Sequence Diagram 19](#_Toc44626991)

[5.4.2 Customer Side Sequence Diagram 20](#_Toc44626992)

[5.5 Physical View 21](#_Toc44626993)

[5.5.1 Deployment Diagram 21](#_Toc44626994)

[5.6 User Interface Design 22](#_Toc44626995)

List of Figures

[Figure 3‑1: Work Breakdown Structure 9](#_Toc44627028)

[Figure 4‑1: Architectural Design 10](#_Toc44627029)

[Figure 4‑2: MVC Architectural Design 11](#_Toc44627030)

[Figure 5‑1: 4+1 Architecture View 13](#_Toc44627031)

[Figure 5‑2: System Use case 14](#_Toc44627032)

[Figure 5‑3: Login and Registration Use Case 15](#_Toc44627033)

[Figure 5‑4: Ordering Process Use Case 16](#_Toc44627034)

[Figure 5‑5: Class Diagram 17](#_Toc44627035)

[Figure 5‑6: Component Diagram 18](#_Toc44627036)

[Figure 5‑7: Restaurant Side Sequence Diagram 19](#_Toc44627037)

[Figure 5‑8: Customer Side Sequence Diagram 20](#_Toc44627038)

[Figure 5‑9: Deployment Diagram 21](#_Toc44627039)

[Figure 5‑10: UI - Welcome 22](#_Toc44627040)

[Figure 5‑11: UI - Login 22](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627041)

[Figure 5‑12: UI - Signup 22](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627042)

[Figure 5‑13: UI - Customer Home 23](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627043)

[Figure 5‑14: UI - Restaurant Menu 23](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627044)

[Figure 5‑15: UI - Restaurant Menu 23](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627045)

[Figure 5‑16: UI - Restaurant Info 23](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627046)

[Figure 5‑20: UI - Previous Orders 24](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627047)

[Figure 5‑19: UI - Cart 24](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627048)

[Figure 5‑17: UI - Menu Item Details 24](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627049)

[Figure 5‑18: UI - Empty Cart 24](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627050)

[Figure 5‑23: UI - Restaurant’s Orders 25](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627051)

[Figure 5‑22: UI - Restaurant Home 25](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627052)

[Figure 5‑21: UI - Previous Order Details 25](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627053)

[Figure 5‑24: UI - Restaurant's Orders 25](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627054)

[Figure 5‑28: UI - Create Menu Item 26](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627055)

[Figure 5‑27: UI - Restaurant's Inactive Menu 26](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627056)

[Figure 5‑26: UI - Restaurant's Active Menu 26](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627057)

[Figure 5‑25: UI - Order Details 26](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627058)

[Figure 5‑29: UI - Navigation Drawer 27](file:///F:\Giki\Semester%206\SE\Project\SDS.docx#_Toc44627059)

**List of Tables**

[Table 1: Terms used in this document and their description 7](#_Toc44626611)

# INTRODUCTION

GIKIEATS is a multi-platform flutter-based application to provide online food ordering facilities to its customer within GIK. This app maintains records of all Restaurants of GIK and their food items. In this app, Restaurants can add their food items and their details on the menu with ease. Customers can view the information menu provided by the restaurant.

This is an Android/IOS application that allows you to access all the whole information about all Restaurants of GIK. Within Institute it is difficult to keep track of food items in different restaurants. This app allows you access to all the information about food items in a single place.

The purpose of this project is to collect, display, and store all the deals and details of food of all Restaurants of GIK in one Application. To provide information at a single place firebase will be used to keep details about Users, Restaurants, Orders, and Menu. Manual searching for the Restaurants and reasonable prices is the most time consuming and difficult as compared to the Manual Suggestion Based System. This system shows the comparison of various existing food items and their prices. In this project, we will introduce some new ideas for improving the existing systems.

## PURPOSE

This project aims at creating an E-menu for online food ordering throughout GIK. This allows registered users of the system to easily log in and can easily visit all restaurants’ menu online that are available on the site and choose the menu available for the order. Saving time, money & easily order by sitting at home are major goals of this project. Customers can register themselves and view all the allowed details of all restaurants in GIK. The main objective of GIKIEATS is to gather all the Restaurants details to one same place and suggest. In this way, every customer will be able to order any kinds of food from any Restaurant from this App. Below are some objectives we will cover in this project:

* We can save detail of all Restaurants in GIK.
* All Deals and Food Staff Record.
* All Customer Details.
* Ordering System for any person, any place, and from any Restaurant.
* Price comparing Module is Useful for all the customers.

## PRODUCT SCOPE

The GIKIEATS app that is to be developed provides the restaurants as well as customers of GIK deals and Food Stuff and notes information and order details. Firstly, all restaurants of GIK will be covered. GIKIEATS is supposed to have the following features:

The project provides the customers and restaurants with Menu viewing and allows restaurants to edit the menu.

The project provides a login facility to the customers.

The system provides the restaurants to add/remove items from their menu daily and contact information in case of any queries

The customers can find out descriptions about food items on the menu. The Order details are stored in the database which can be reviewed by both restaurants and customers.

Documents are used in the future phases of the project development cycle. The features described here meet the needs of all the customers. The success criteria for the system are based on the level up to which the features described in this document are implemented in the system.

Table 1: Terms used in this document and their description

|  |  |
| --- | --- |
| Name | Description |
| UI – User interface | The visible front-end which the user can interact with. |
| Customer | Customer who orders Food |
| Restaurant | The restaurant which approves the order and adds menu items |
| Menu/ Menu Items | Menu added by the restaurant. Displayed to customer |
| Order | Customer Orders food. It is accessed by the Restaurant to be approved. |

## OVERVIEW

The following document contains the architectural design of the product and represents multiple views of the product by using the 4+1 architectural design. This document also displays the user interface of the, so far, developed product.

The following designs and patterns are meant for different types of viewers, so this document is available for inspection by all stakeholders of the developing product.

# THE OVERALL DESCRIPTION

The description of the proposed system is given below. This description defines the product in different directions.

## 2.1. PRODUCT PERSPECTIVE

The Application is developed in context has been influenced by the Food Panda App. Keeping in mind the necessities of the Ghulam Ishaq Khan Institute’s Campus and under the influence of well-known Food Panda App, GIKEATS is being developed as a self-contained product for the ease and effectiveness of the order process that every child in Ghulam Ishaq Khan Institute has to go through.

The product consists of a mobile application to allow easy accessibility to the product; mobile app will have a user-friendly interface that allows the user to access all the product functionality. All the user data will be stored on a real-time database that is Google firebase fire store as it provides all the data integrity constraints and authentication.

To use the product, users will be asked to register through the mobile interface. As soon as the new user is registered into the product all his/her data will be saved in the database. Users will then be given access to the user platform where he/she can use all the product functionality. Later, the user will be able to log in and logout from the developed app anytime he wants. Since every operation that the user performs reflected in our real-time database, the user will find the platform updated from the last time he/she had login.

From the customer’s point of view, users can add items into the cart which upon checkout is added into the order list in the real-time database this order can then be viewed and approved by the restaurant role. Both users can communicate with each other using a real-time food ordering application that will be integrated into mobile applications.

The restaurant can add/edit the menu which will change the menu entries in the database. It can then be viewed by the customer when logged in to his/her account.

# WORK BREAKDOWN STRUCTURE

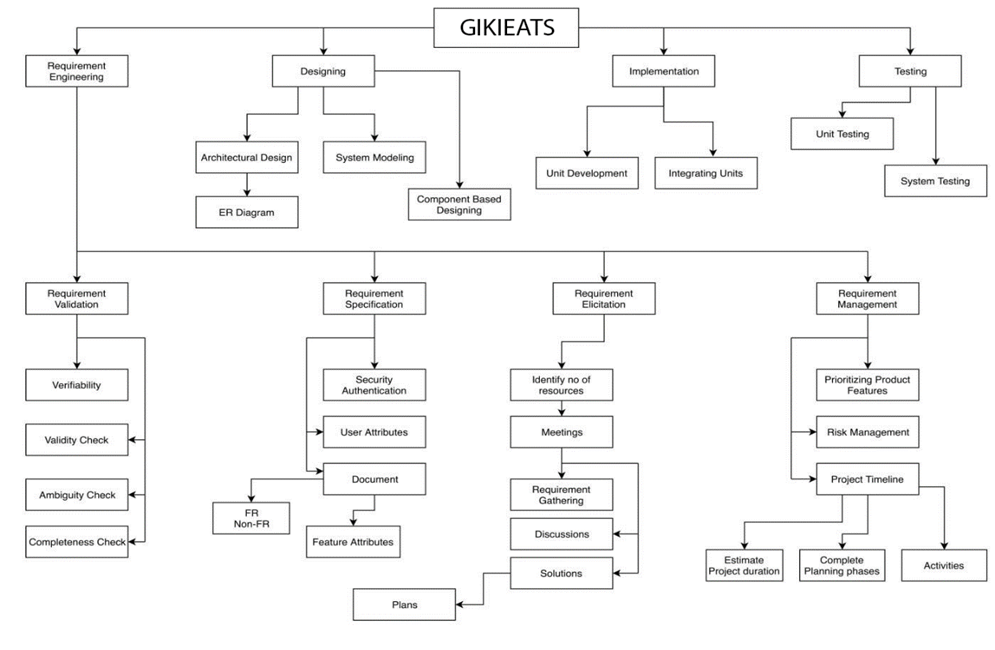


Figure 3‑1: Work Breakdown Structure

# Design

## ARCHITECTURAL DESIGN

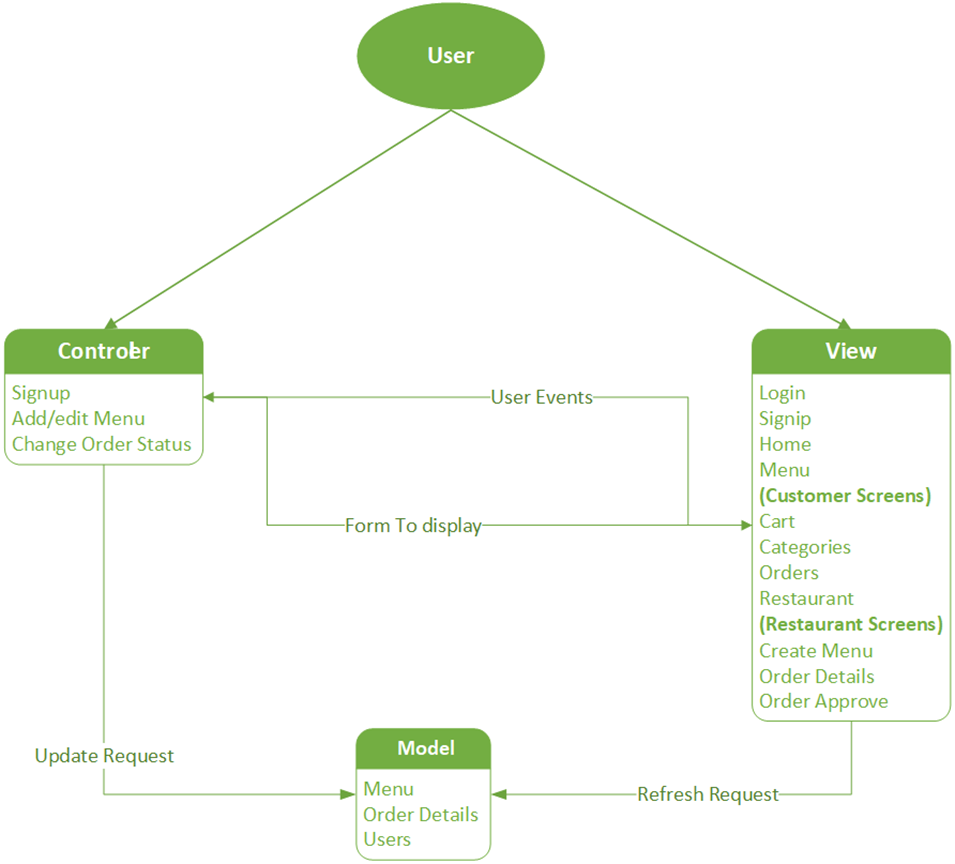


Figure 4‑1: Architectural Design

## Why we choose MVC Architecture Design?

The architectural design used for the GIKIEATS application is the Model View Controller.

**Model–view–controller** (usually known as **MVC**) is a software design pattern commonly used for developing user interfaces that divide the related program logic into three interconnected elements. This is done to separate internal representations of information from the ways information is presented to and accepted by the user. This kind of pattern is used for designing the layout of the page.

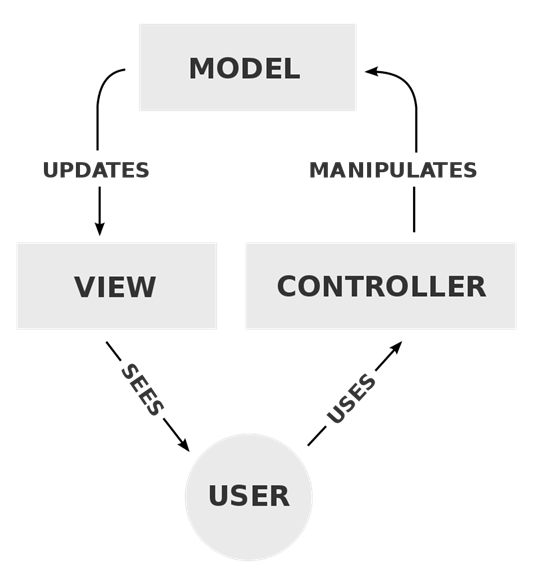


Figure 4‑2: MVC Architectural Design

### Components

**Model**

Firebase is used as our model. Restaurants and customer both are connected to firebase. It directly manages the data, logic, and rules of the application.

**View**

Our main view is divided into two main roles, Customer and Restaurant. The restaurant has a specific logon which takes them to their interface.

**Controller**

This part controls our Model. Following are the control elements which affects our model:

* As a user signs up, an entry is created in the database for that user.
* The restaurants can Add/Edit Menu which changes the menu items in the database.
* When Order is delivered, order status is changed to Delivered in the Database

## MODULE IDENTIFICATION

The architectural design is divided into 4 different parts. The application has 2 user roles named customer and restaurant.

Firebase authentication is used both on the server-side and client-side is the module that provides client-side and server-side authentication for secure access of data by authorized users. This module ensures that no harmful or malicious user can access or change any other user’s private data.

The Firebase Real-time Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in real-time. Restaurant updates menu to be viewed by customers in real-time. Similarly, order details are also updated to be approved by the restaurant.

App’s UI is the view and Classes like signup, order, and Menu edit which manipulates Databases in any way are controllers.

# 4+1 ARCHITECTURE VIEW MODEL

In this section, you draw the architecture using the views defined in the “4+1” model.

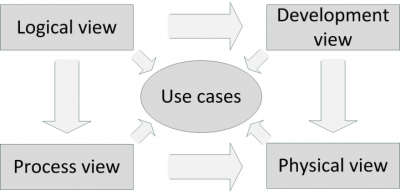


Figure 5‑1: 4+1 Architecture View



## Use Case View

This is a list of use-cases that represent major functionality of the final system:

### System Use case

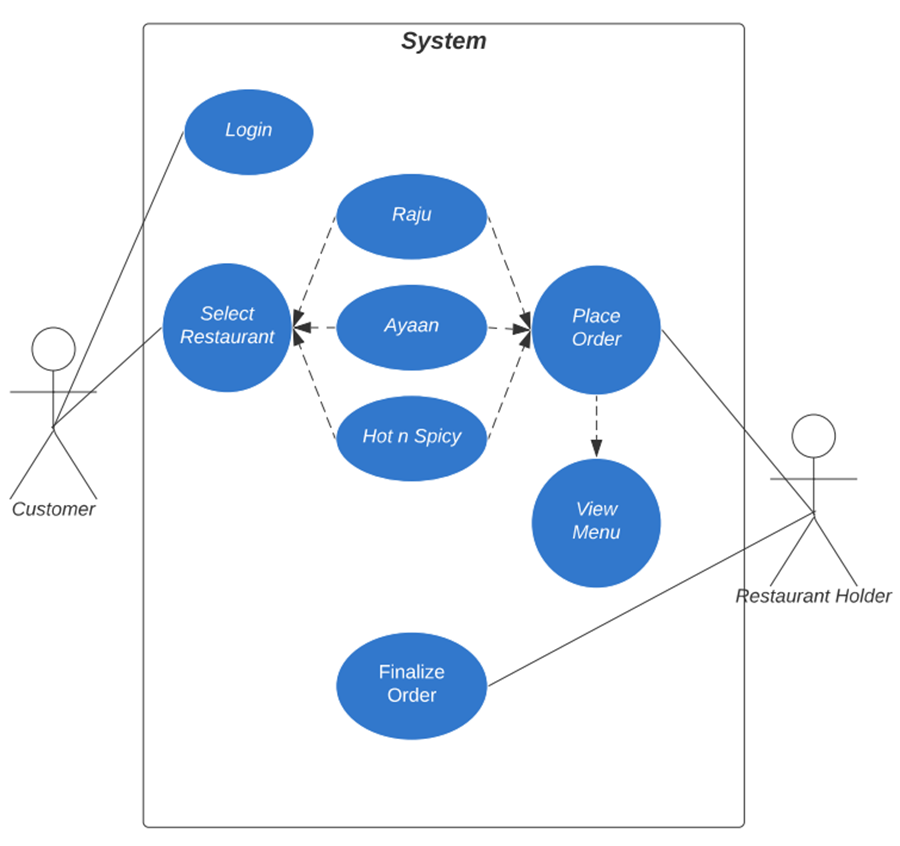


Figure 5‑2: System Use case

### Login and Registration Use Case

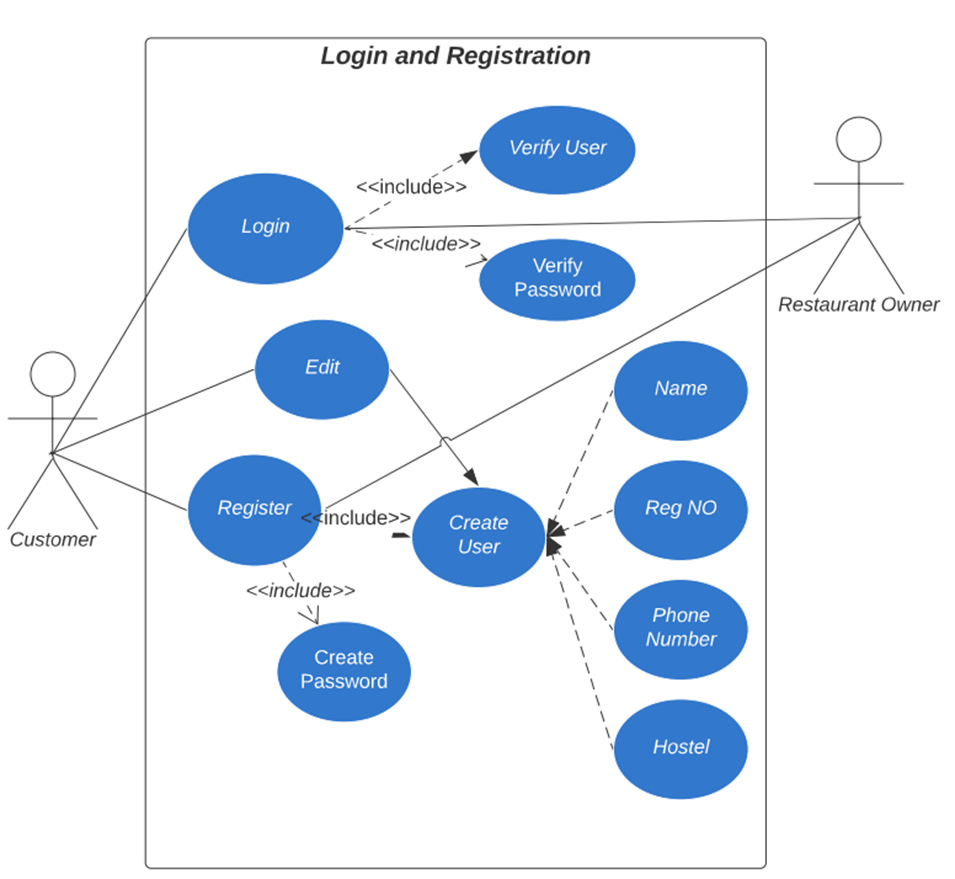


Figure 5‑3: Login and Registration Use Case

### Ordering Process use case

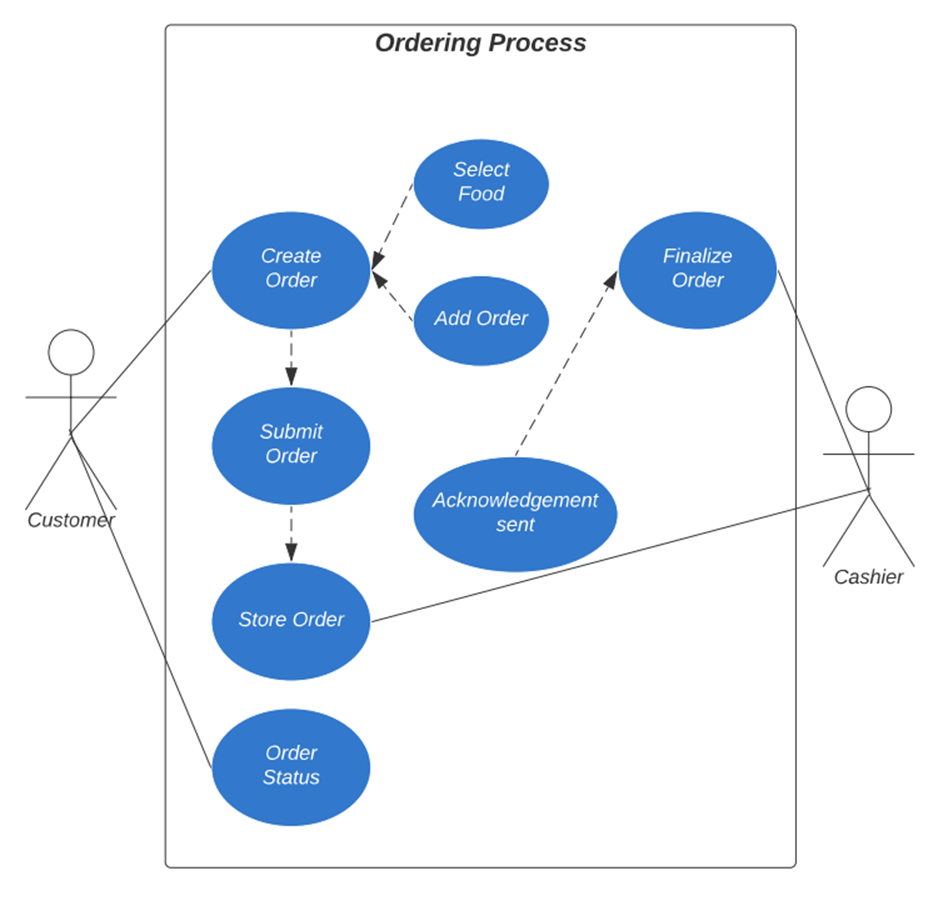


Figure 5‑4: Ordering Process Use Case

## Logical View:

### Class Diagram

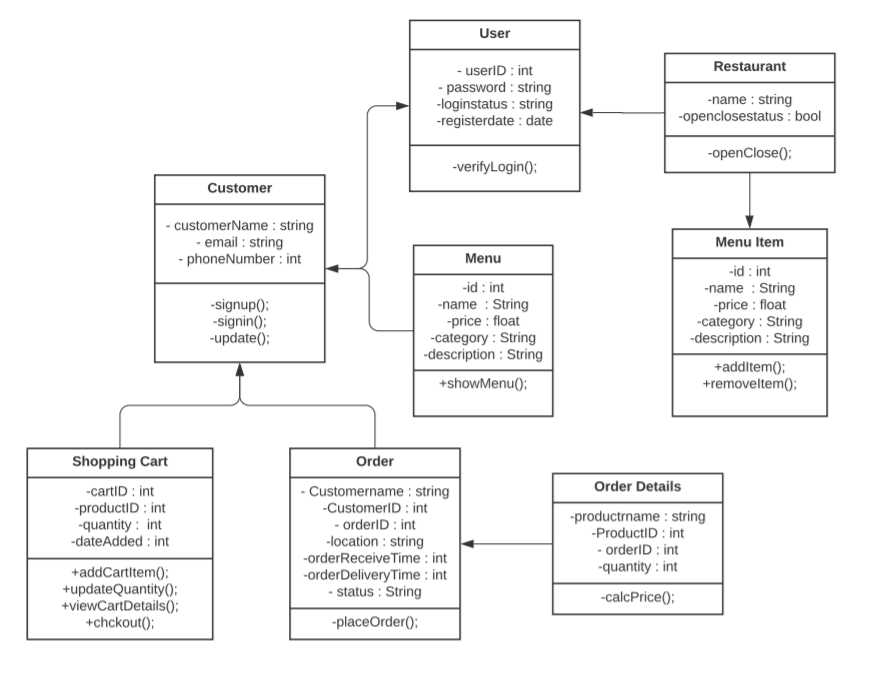


Figure 5‑5: Class Diagram

## Development View

### Component Diagram

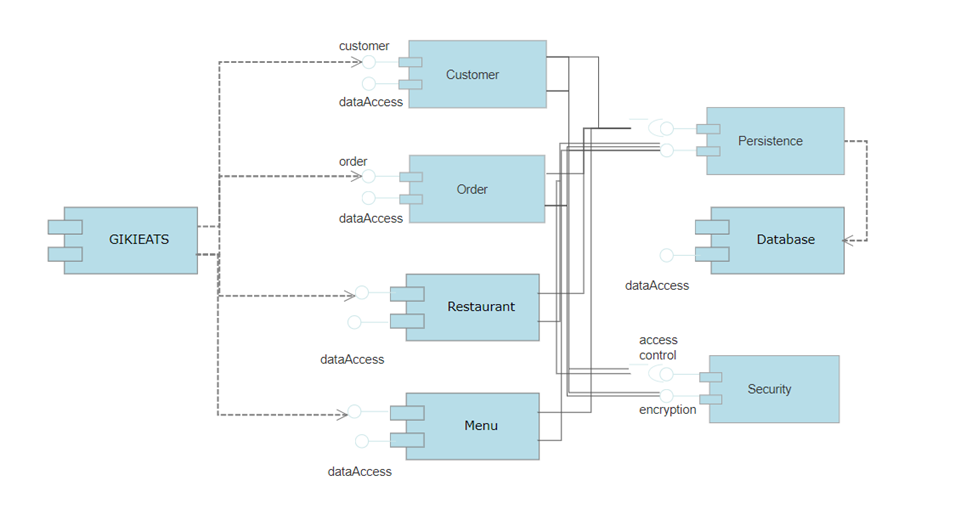


Figure 5‑6: Component Diagram

## Process View

### Restaurant Side Sequence Diagram

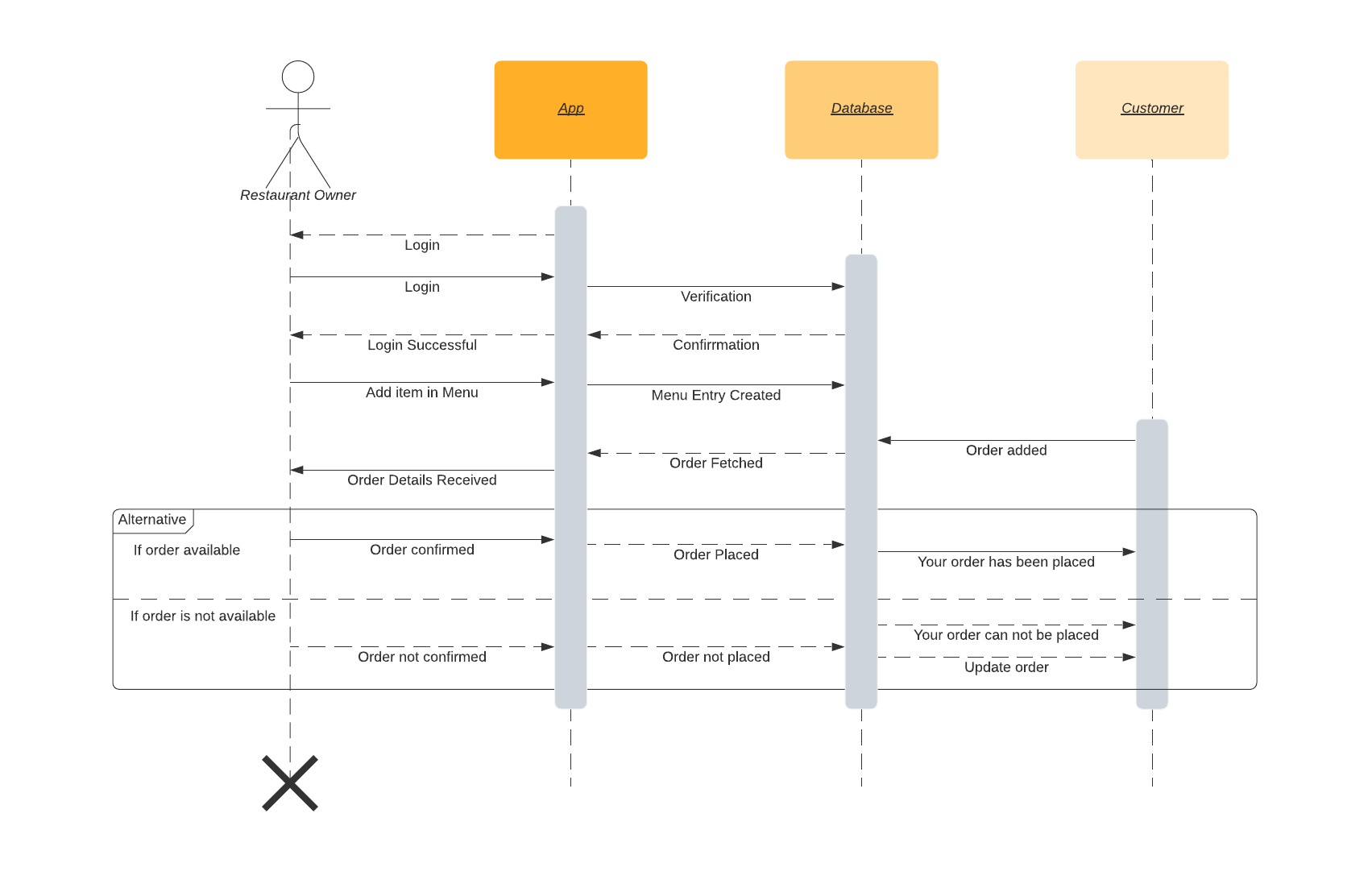


Figure 5‑7: Restaurant Side Sequence Diagram

### Customer Side Sequence Diagram

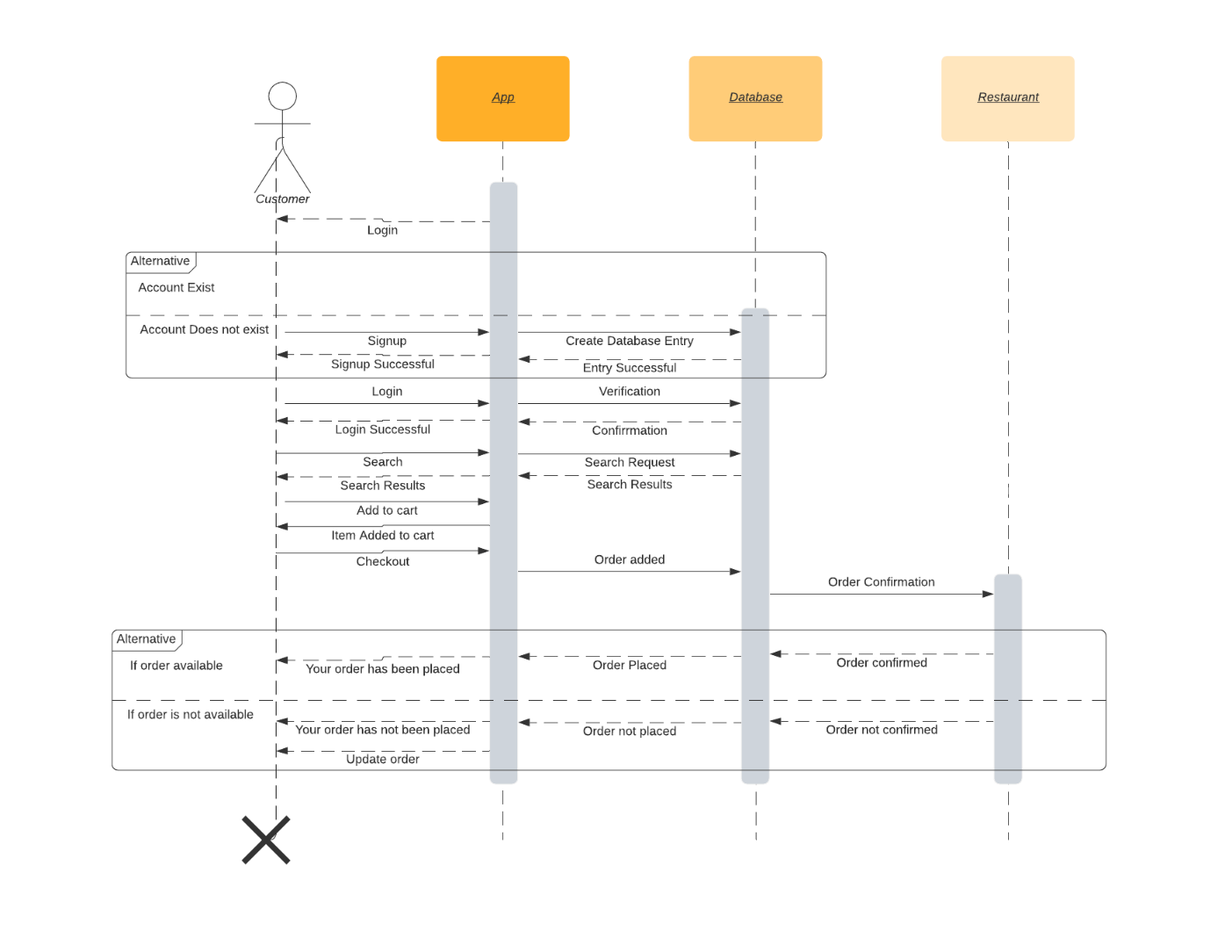


Figure 5‑8: Customer Side Sequence Diagram

## Physical View

### Deployment Diagram

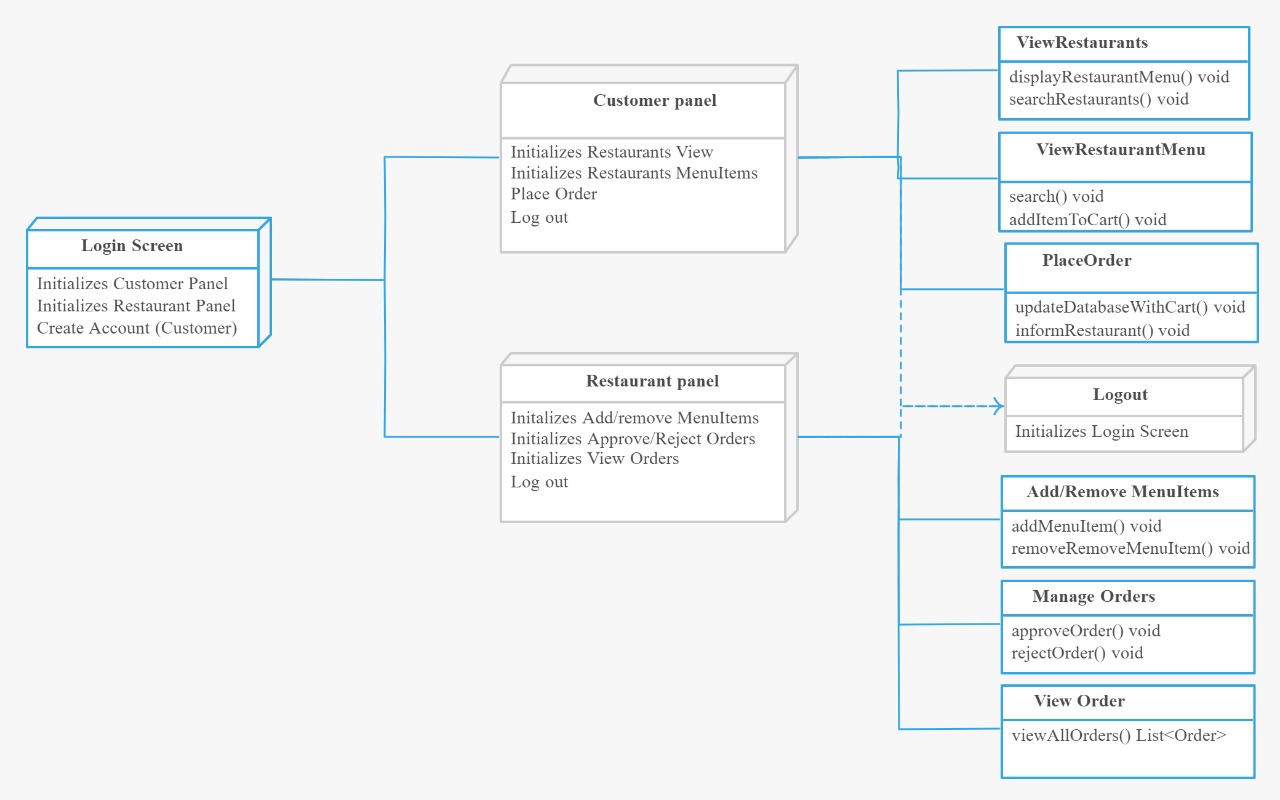


Figure 5‑9: Deployment Diagram

## User Interface Design

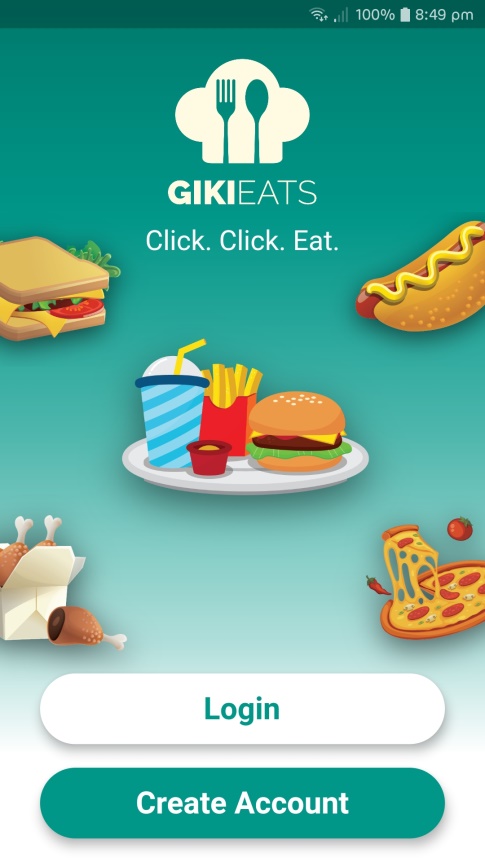


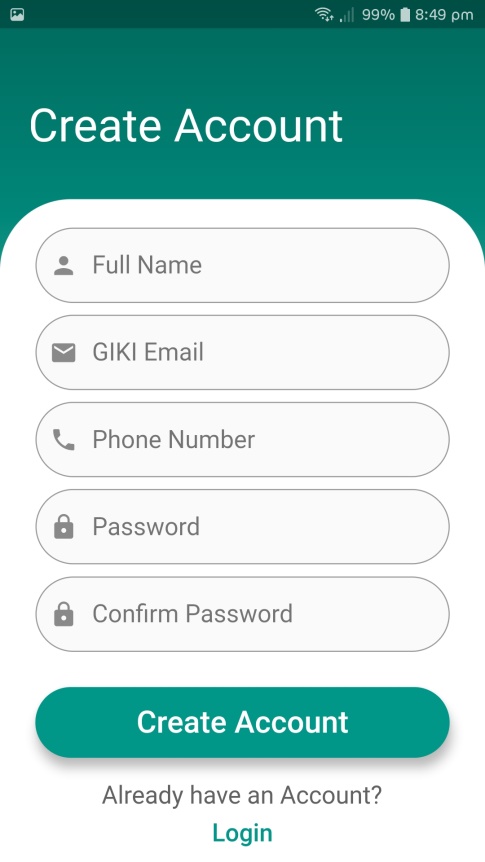
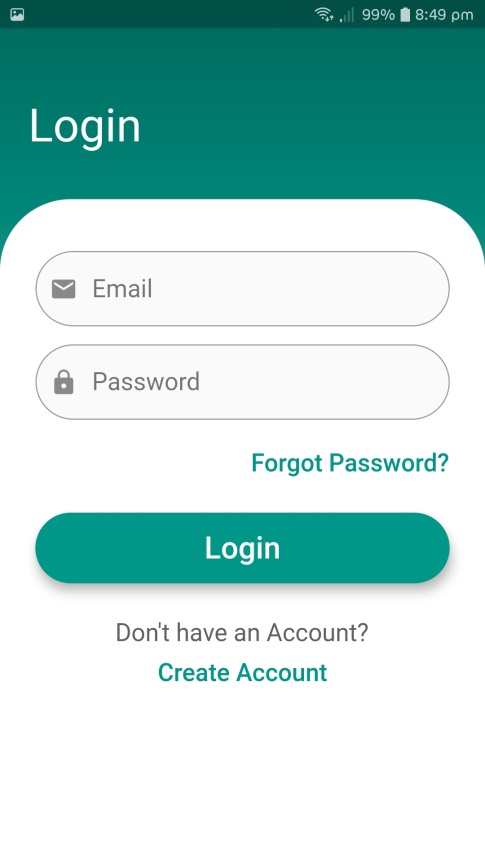
Figure 5‑10: UI - Welcome

Figure 5‑11: UI - Login

Figure 5‑12: UI - Signup

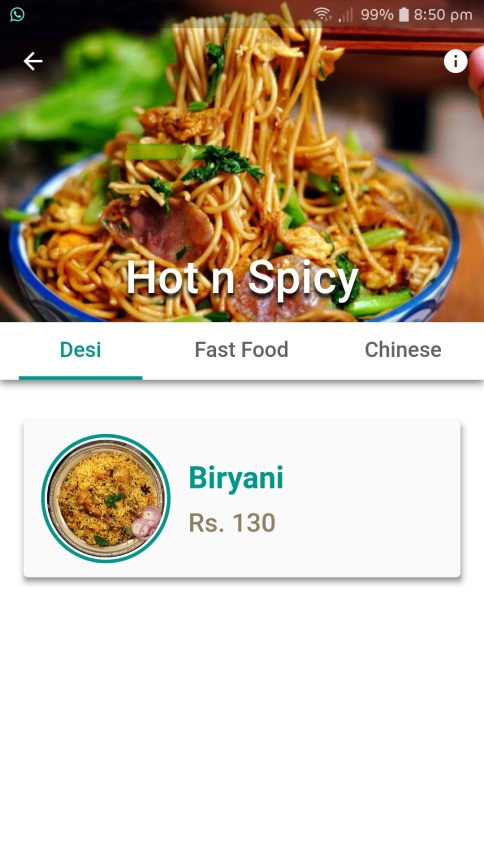
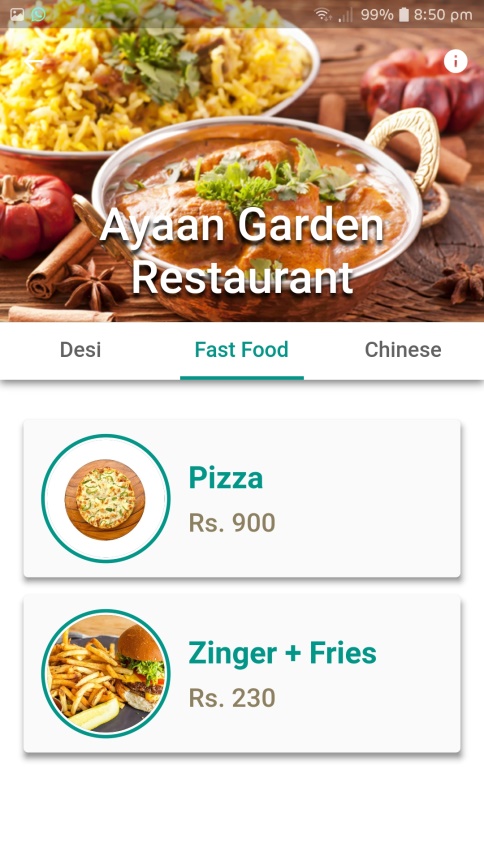
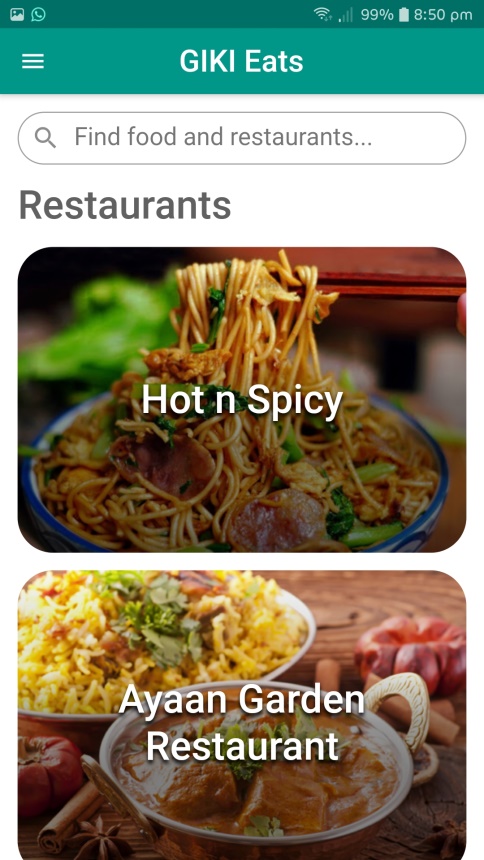


Figure 5‑13: UI - Customer Home

Figure 5‑14: UI - Restaurant Menu

Figure 5‑15: UI - Restaurant Menu

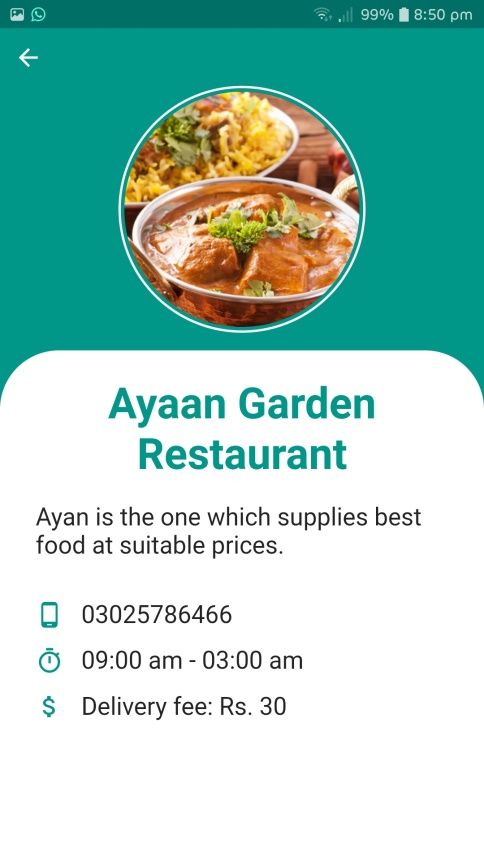
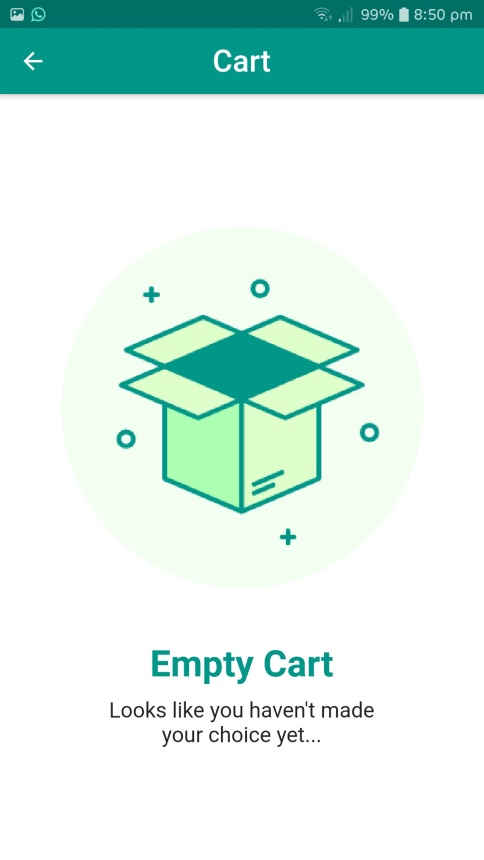


Figure 5‑16: UI - Restaurant Info



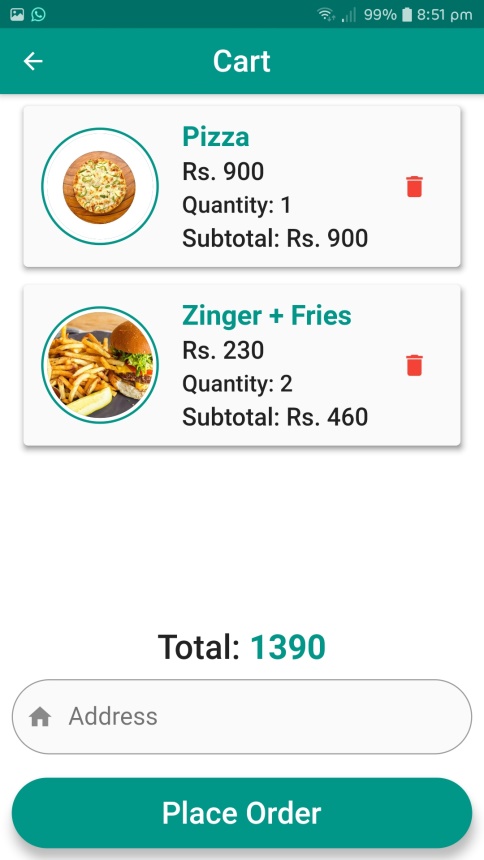
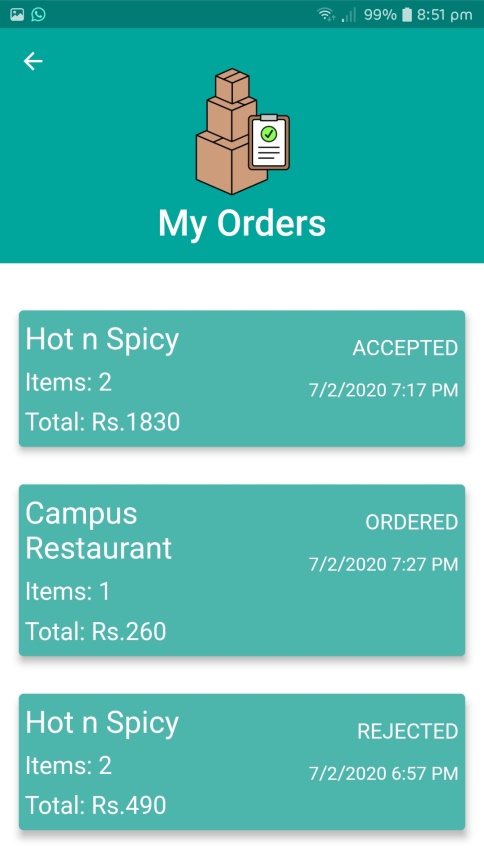


Figure 5‑17: UI - Previous Orders

Figure 5‑18: UI - Cart

Figure 5‑19: UI - Menu Item Details

Figure 5‑20: UI - Empty Cart

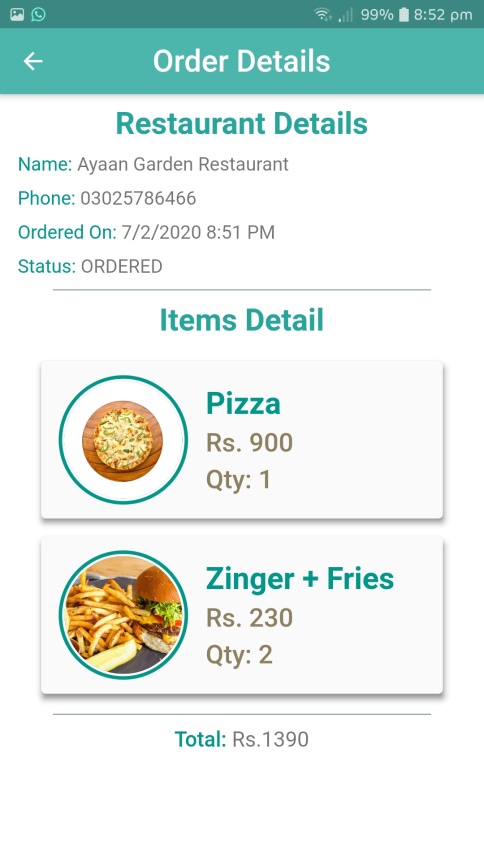
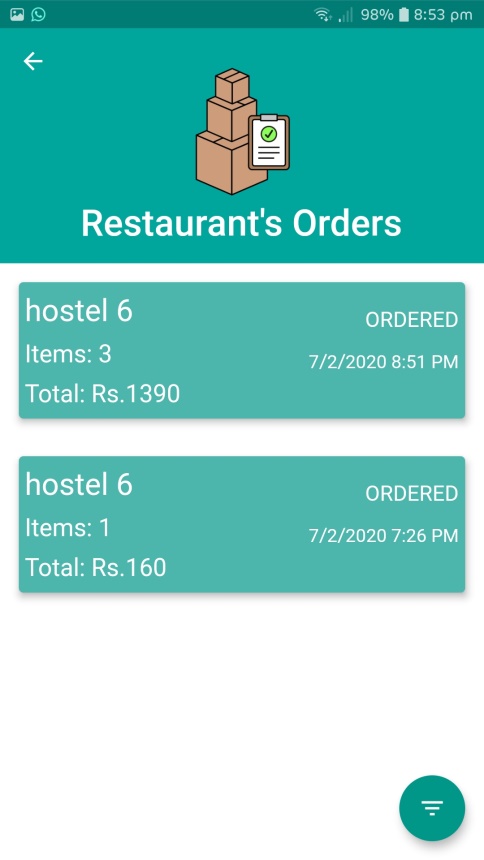
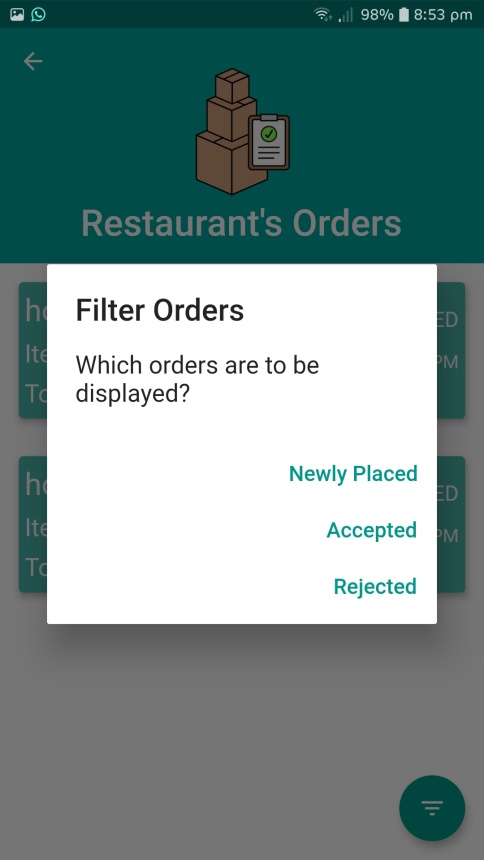
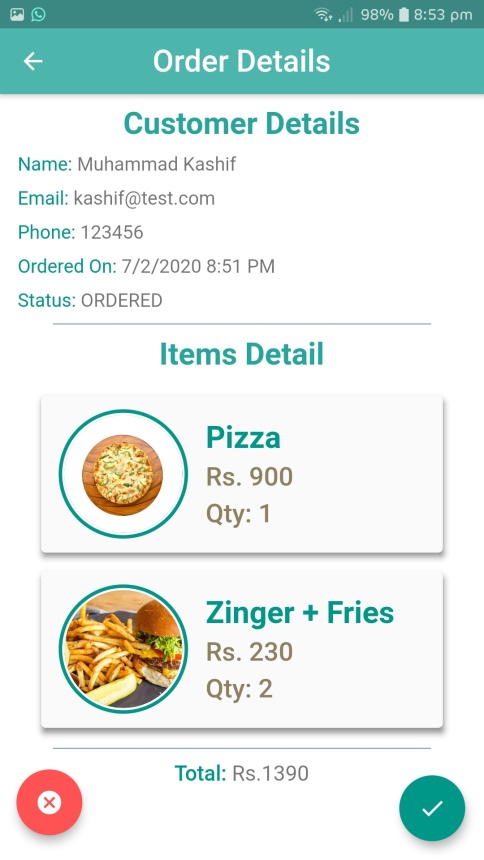
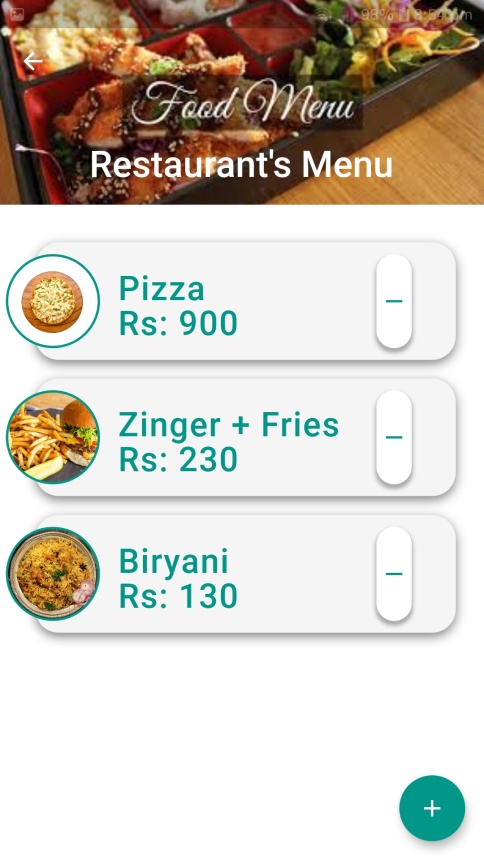


Figure 5‑21: UI - Restaurant’s Orders

Figure 5‑22: UI - Restaurant Home

Figure 5‑23: UI - Previous Order Details

Figure 5‑24: UI - Restaurant's Orders



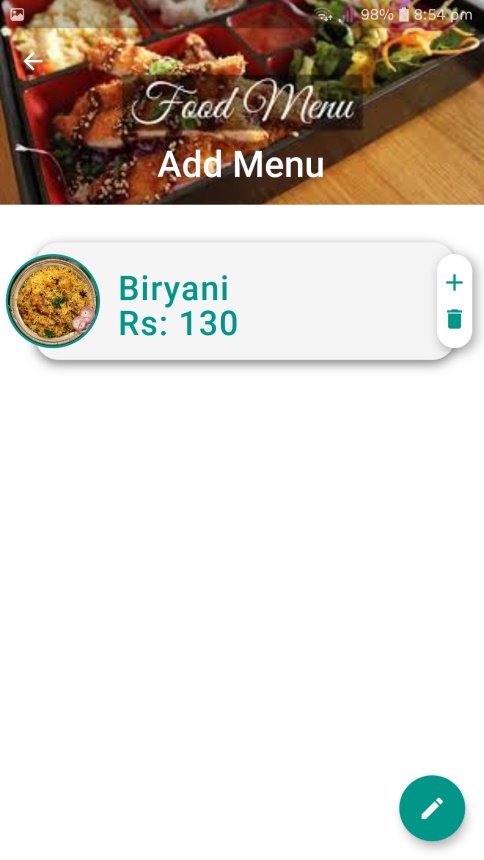
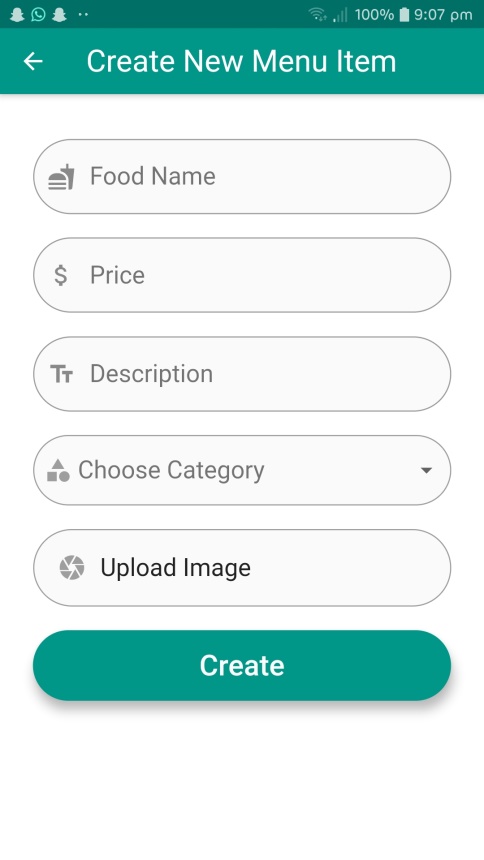


Figure 5‑25: UI - Create Menu Item

Figure 5‑26: UI - Restaurant's Inactive Menu

Figure 5‑27: UI - Restaurant's Active Menu

Figure 5‑28: UI - Order Details

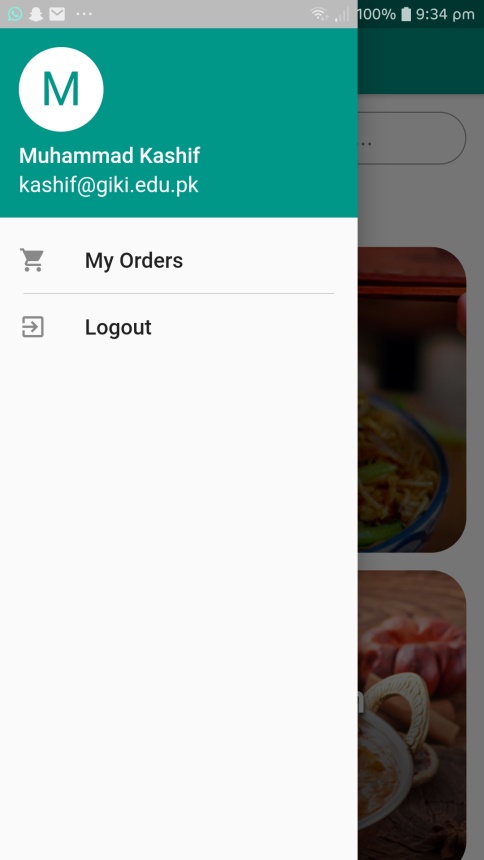


Figure 5‑29: UI - Navigation Drawer