RESTAURANT MANAGEMENT SYSTEM

Course of

EXL – Certified Software Test Engineer

Submitted By

Nikita P. Ovhal

Batch: 2022-7669

Enrollment Number: EBEON0722619400

Under the Guidance of

Ms. A. Mohana Priya

Centre Trainer

JANUARY-2023

INDEX

Table of Contents

CHAPTER 1

INTRODUCTION

"Kitchen Café & Resto" is a restaurant located in pune. They are handling food ordering process and daily routine of their restaurant manually. Customers cannot order food online in this hotel. In case of take away food or dining there, customers have to visit or call and order. Table reservation is also the same. This project is based on automating the above mentioned processes. The problem that many businesses face today is to make sure that they attract new customers and also they keep their existing customers. The cost to attract a new customer is costlier than retaining the old customer. Therefore, there is an argument that for a business, existing customer is worthier than a new customer. In this industry, a customer is likely to return to the restaurant in the future if they received an excellent customer service as well as appetizing food . However, if they had to wait for an unreasonable amount of time or there was a mistake in the order, it's very unlikely the customer would return.

1.1 Objective of the Project

The Main Objective: To build a web based restaurant management system for "Kitchen Café & Resto" Restaurant. In order to fulfill the main objective following goals have to be achieved.

- Improve customer relationship management
- Avoid long queue
- Bulk Processing
- Customer feedback
- Stock Control
- Menu item management

1.2 Existing Solutions

There are many computerized restaurant management systems available but for each system there exist disadvantages or missing features. The most common type of restaurant management system contains a static order entry computer system usually in the shape of a desktop computer with a touch screen. Typically this common approach is adequate to the restaurants requirements but still requires handwritten orders to be relayed to the order entry computer system. A slightly different approach was implemented in a restaurant in Nuremberg, Germany, named s Baggers. The restaurant utilizes a roller coaster approach to serving the food and an order entry system fully operated by the customer. As reviewed by the BBC there is no need for any waiters as the customers use touch-screen monitors to browse the menu. This new invention can save on operating costs, but the initial injection of cash required is substantial as every table requires the necessary hardware. The next section will introduce the project proposal listing the proposed features of the system

1.3 Project Proposal

The aim of this project is to create a restaurant management system that can incorporate the benefits of all the existing solutions but without any of the drawbacks as well as including many new features. Many of the existing solutions to POS (Point-of-Sale) systems are sold with the required expensive hardware so for any business looking to work to a budget, the more enriched software solutions are just out of their range.

1.4 Scope of the project

Proposed system is valuable for both customer and the restaurant because it is simplifying the order processing process. Customers have to create an account with valid phone number or email and can log-in to the system. The web page has up-to-date and interactive menu with all the available food items. When customer made a selection, items added to their order. Customer can review order at any time and change the selection before the payment. Online payment and cash on the collecting counter is possible. Confirmation is prompt to the customer. If it is required, customers can check the table availability and make a table reservation if necessary. Proposed system is not handling delivering food, due to restaurant is not having delivery team. After a customer placed an order, order details are visible to the kitchen. They can see what the quantity that is required from each item is and they can fulfill multiple orders with same food item in same time. This happens because proposed system can combine orders during allowed time period. Stock of the ingredients should be always up to date. Otherwise, restaurant cannot fulfill the customer orders properly and may accept orders that cannot be fulfilled due to lack of ingredients. Real-time view of ingredient stock levels is very much important to so much necessary to any restaurant. Then only the meals with enough ingredient stock can be sold. The stock levels will be updated by the kitchen staff at the end of the day through the proposed

system. Restaurant should be able to control the menu items. This ensures that the customers can only order available food items. Only authorized employee can handle the menu. They can create and remove food categories, food items. When creating these categories and food items, employee can add photos and description also. Adding new food items also possible. Remove food items and change visibility of food items. Create and remove options of food items. They can edit und update prices of the food items. They can select “Food Item of the day” and allocate special discount to them. Giving reasonable discounts to the customers, can keep customers without moving out from the restaurants customer base. First restaurant must identify if the customer is new customer or regular customer. Giving proper discounts to regular customers will increase their satisfaction and they will not move to another restaurant.

Restaurant can keep track of all the orders, retrieve and display order information. This information invisible to the kitchen for order processing. These order details can be useful to report generation. Customers can give general feedback and suggestions about the restaurant and meals. Those are only visible to the management of the restaurant. Also, customers can give star rating to each and every meal. Other customer can view those when they are selecting their meals. Order logging and report generation are also implemented in proposed system.

CHAPTER 2

METHODOLOGY

2.1 Analysis

Existing method in this restaurant is all manual and study of the current system is done by using the following fact finding techniques.

Observations

Interviews

According to the information that gathered from interviews, employees have following problems.

Kitchen Staff and serving staff

It is difficult to remember large number of orders at once. And sometimes they have to prepare same food item several times. They prepare bulks. But it's difficult to pack the orders during busy times. If there is a method to view orders by food items (Bulks) it's easy to prepare. There should be a way to view orders separately for easiness of packing purpose for take-away customers. It's easy for serving staff to arrange the orders in frequent manner and that will help to serve customers as first come first out

Cashiers

Currently they have to remember all the item codes, if there is a new employee it is very difficult to bill the order in cash register and its time consuming. If there is a method to see the items in category wise that would be easy and efficient. By observing the working environment, it is very clear about the drawbacks of the current system. Current method causes customer dissatisfaction and due to those competitors also getting advantages.

Nature of the current method

The current method is based on papers. Only way to track order is cash register and the bill that given to the customer. Menu cards are also paper based or verbal. Those menu cards are printed and if any change is required restaurant must redesign the menu cards and reprint them. This is a big waste and it is costly and it is impossible to reprint every time even for a miner change. Current system is time consuming. Customers have to waste their time in queue and place the order. Several case of serving a wrong order are also happened due to miscommunication. Therefore, briefly, For placing any orders customers have to visit restaurant to know about food items and then place order and pay. In this method time and manual work is required. It is

difficult to ensure whether the order has placed correctly or not because cash register codes are unknown to the customer. Sometime there are miscommunication between kitchen and front desk. No database is present; therefore, analysis is impossible

2.2 Requirements of the proposed system

Web Ordering Function

This is the front end of the system. Customers can place their order and supply necessary details throughout this module. Following functionalities must be provided to the restaurant stakeholders.

Feature	Customer	Cashier	Admin User	Chef
Account creation	✓	✓	✓	✓
Managing own account	✓		✓	
Log-In to the system	✓	✓		
Check Menu	✓	✓		
Select food Items	✓	✓		
Select food Items				
Select food Items	✓	✓		
View order	✓	✓		✓
Cancel order	✓		✓	
Make payment	✓	✓		
Table Reservation	✓	✓		

Menu Management Function

Provide functionality for the Admin User only. It will not be available to any other users of the system like Restaurant Employees or Customers. Using a graphical interface, it will allow an Admin to manage the menu that is displayed to users of the web ordering system:

- Add/update/delete food category to/from menu.
- Add/update/delete food item to/from menu.
- Update price for a given food item.
- Update additional information (description , photo, etc.) for a given food item.

Before customers can actually use this system, functionality provided by this component will have to be configured first. Once the initial configuration is done, this will be the least likely used component as menu updates are mostly seasonal and do not occur frequently.

Order Retrieval Function

Designed to be used by Admin and cashiers This function provides the following functions:

- New order retrieval from the database.
- Display the orders in an easily readable, graphical manner.
- Cashiers can view/Edit/Remove and make orders from visiting customers.

Kitchen Function

This is the Simplest Module. Kitchen Staff and serving staff can use this. It provides flowing functions.

- View the orders as bulks for the easiness of preparation.
- Mark orders as ready.
- Mark some items as not available.
- View orders in the sequence they come

Inventory Control Function

- Only useable for the Admin User
- Update Stock Levels
- Inform when stocks are in re-order level

User Management function

Only useable for the Admin User

- Admin users can add/edit/remove cashier accounts
- Remove or edit customers

Discount and offers function

Only useable for the Admin User

- Add Discounts
- Edit Discounts

2.3 Non-functional requirements

Non-functional requirements do not directly affect the system. But rather than specific behaviors, these requirements are specific criteria that can be used to evaluate the working behavior of the system. If the system is failed to meet non-functional requirements sometimes whole system may unusable. Some constraints and restrictions can be considered as nonfunctional requirements when developing this kind of software. Quality behaviors, quality attributes and quality of service may fall in to this category.

Following are the non-functional requirements for this restaurant management system.

- The System should provide user friendly environment including flexible interfaces.
- Person with average computer skills can work with the system with a short period of training
- The system should be accurate and consistent, when manipulating the fed data in proper way and displaying correct information.
- The System should be accurate and consistent, when manipulating the fed data in proper way and displaying correct information,
- The System should keep up security and reliability, because the system handles important data related to business processes of the company.
- Occasionally backups should be taken to maintain reliability the system should be reusable and maintainable.

CHAPTER 3

SYSTEM DESIGN

3.1 Design Techniques

To model the system, design techniques are used. Object Oriented design, Rapid application development, prototyping and Modern structured design are examples to design techniques that currently used.

For this development Object oriented design techniques has been used. In this method there are advantages such as, code reusable facility, design benefits and maintainable facilities with Objects and classes.

Unified Modeling Language (UML) plays a significant role in Object Orient designing. UML allows programmers easy understand models of objects so that programmers can easily write software. Some of structural and behavioral UML diagrams use for design proposes systems are mentioned below.

- Use case diagrams – This makes clear what the capabilities of the system are.
- Use case narratives – Make use cases more clear
- Activity diagrams – shows how activities are coordinated to provide a service which can be at different levels of abstraction.
- DFD's – Data flow diagrams representing a flow of data through a process or a system
- Class diagrams – this is the main building block in object oriented modeling

Use case report for Restaurant Management System

Sr. no	Use cases	Actors	Description
1.	Create Account	Customer , Admin	Customer Create their Account. Admin Create Account
2.	Login	Admin ,Customer	Admin can login into Account Customer can login into account
3.	Update Menu	Admin	Admin add new item to menu. Admin delete item from Menu Admin edit prices on food item
4.	View Menu	Admin , Customer , Staff	Admin view in Menu Customer view in menu Staff View in Menu
5.	Order Food	Customer	Customer can order food
6.	Reserve Table	Customer , Admin	Customer can reserve table Admin can give confirmation for reserve table
7.	View Order	Customer , Admin , Staff , Cashier	Customer can View Order Admin can View Order Staff Can View Order Cashier can View Order
8.	Confirm Order	Staff , Admin	Staff Can Confirm Order Admin can Confirm Order
9.	Clear Bill	Cashier , Customer	Cashier Generate Bill Customer Make Payment
10.	View Transaction	Cashier , Admin	Cashier can View all Transactions Admin Can View Transactions

1. Introduction: This use case outlines the steps that need to be followed in order to login into the system
Actors:

- Customer
- Admin
- Staff
- Cashier

2. Precondition: Customer has visited to our website and search the menu they want to order. They are now ready to place their order.

3. Post Condition: Customer Received the Order

4. Basic Flow:

- Login
- The page will request the user/actor to provide valid credentials
- User/actor enters the credentials
- User/actor enters into the system

5. Alternate Flow:

- Invalid Credentials
- If user/actor provides invalid credentials in the basic flow, a validation message or error message should appear. Hence, returning the user to the basic flow..

6. Special Requirements

None

Usecase1: Create Account

Overview: This Scenario describe customer and admin can create their account

Actors: Admin , Customer

Precondition: Customer has visited to our website.

Scenario:

Action	Software Reaction
Customer can Create Account Enter name Enter phone no Enter password Enter confirm password	User can enter into login
Admin Can Create Account Enter name Enter phone no Enter password Enter confirm password	User can enter into login

Post condition: user enter into the login

Use case Dependencies:

Usecase2: login

Usecase3: Update Menu

Usecase4: View Menu

Usecase5: Order food

Usecase6: Reserve table

Usecase7: View Order

Usecase8: Confirm Order

Usecase9: Clear Bill

Usecase10: View Transaction

Usecase2: Login

Overview: This Scenario describe customer and admin can login into their account
Actor: Admin, customer

Precondition: Customer has visited to our website.

Scenario:

Action	Software Reaction
Admin Login Enter name Enter password	If it is valid credential admin can enter into the system If it is invalid credential admin have to move #1
User Login Enter name Enter password	If it is valid credential Customer can enter into the system If it is invalid credential Customer have to move #1

Post condition: if use case successfully executed the customer / admin should be logged into the system Otherwise, it move to #1

Use case Dependencies:

Usecase3: Update Menu

Usecase4: View Menu

Usecase5: Order food

Usecase6: Reserve table

Usecase7: View Order

Usecase8: Confirm Order

Usecase9: Clear Bill

Usecase10: View Transaction

Usecase3: Update Menu

Overview: This Scenario describes admin can update Menu

Actors: Admin

Precondition: Admin can View Menu List

Scenario:

Action	Reaction
Update Menu 1. Add item 2. Delete Item 3. Edit Price on item	 Items added to the menu Item deleted from the menu Item prices are edited

Post condition: Admin can successfully update the items in menu

Exception:

None

Special Requirement:

None

Use case Dependencies:

Usecase4: View Menu

Usecase5: Order food

Usecase6: Reserve table

Usecase7: View Order

Usecase8: Confirm Order

Usecase9: Clear Bill

Usecase10: View Transaction

Usecase4: View Menu

Overview: This Scenario describe admin, customers and staff can view Menu

Actors: Admin, customer, staff

Precondition: Customer has visited to our website and searches the menu they want to order.

Scenario:

Action	Reaction
Customer indicates that they would like to view menu. Admin want to view menu. Staff want to view menu	System shows following data to the customer , admin and staff • Menu list • Price • Availability

Post condition: Menu list successfully shown to customer, admin and staff

Exception:

None

Special Requirement:

None

Use case Dependencies:

Usecase5: Order food

Usecase6: Reserve table

Usecase7: View Order

Usecase8: Confirm Order

Usecase9: Clear Bill

Usecase10: View Transaction

Usecase5: Order Food

Overview: This Scenario describe customer is ready to order food

Actors: Customer

Precondition: Customer has visited to our website and search the menu they want to order. They are now ready to place their order.

Scenario:

Action	Reaction
Customer indicates that they would like to place order	System shows following data to the customer • Menu • Price • Availability • Payment

Post condition: order successfully placed .

Exception:

None

Special Requirement:

None

Use case Dependencies:

Usecase6: Reserve table

Usecase7: View Order

Usecase8: Confirm Order

Usecase9: Clear Bill

Usecase10: View Transaction

Usecase6: Reserve table

Overview: This Scenario describe customer reserve table, admin can send confirmation to the customer

Actors: Customer, Admin

Precondition: Customer place their order and they want to book a table

Scenario:

Action	Reaction
Customer indicates that they would like to Reserve table	System shows following data to the customer • Day • Time

Post condition: successfully reserve table

Exception:

None

Special Requirement:

None

Use case Dependencies:

Usecase7: View Order

Usecase8: Confirm Order

Usecase9: Clear Bill

Usecase10: View Transaction

Usecase7: View Order

Overview: This Scenario describe admin, customers and staff can view Order

Actors: Admin, customer, staff, cashier

Precondition: Customer has visited to our website and they want to View order.

Scenario:

Action	Reaction
Customer indicates that they would like to view View order. Admin want to view Order. Staff want to view Order Cashier want to View Order	System shows following data to the customer , admin and staff • Menu • Price • Availability • Time • Day • Payment

Post condition: Order is successfully shown to the customer , admin and staff

Exception:

None

Special Requirement:

None

Use case Dependencies:

Usecase8: Confirm Order

Usecase9: Clear Bill

Usecase10: View Transaction

Usecase8: Confirm Order

Overview: This Scenario describe admin, customers and staff can confirm Order

Actors: Admin, customer, staff

Precondition: Order is placed by customer and staffs want to confirm order

Scenario:

Action	Reaction
Staff indicates that they would like to confirm order for prepare food. Admin can send notification to the customer for confirm order Customer received notification for confirm order	 Notification send successfully to customer registered number

Post condition: Notification send successfully to customer registered number for confirm order

Exception:

None

Special Requirement:

None

Use case Dependencies:

Usecase9: Clear Bill

Usecase10: View Transaction

Usecase9: Clear Bill

Overview: This Scenario describe cashier generate bill and customer make payment.

Actors: Admin, customer, Cashier

Precondition: Order is confirm by customer and Cashier start generating bill

Scenario:

Action	Reaction
Cashier start generating bill. Customer received bill and make payment	System shows what is order and price System shows overall payment

Post condition: payment done by customer

Exception:

None

Special Requirement:

None

Use case Dependencies:

Usecase10: View Transaction

Usecase10: View Transaction

Overview: This Scenario describe cashier and admin can view transaction.

Actors: Admin , Cashier

Precondition: Cashier generates a bill. And admin want to view transactions

Scenario:

Action	Reaction
Cashier wants to view transaction. Admin want to View transaction.	System shows recent transactions

Post condition: Admin logout from the system

Exception:

None

Special Requirement:

None

Use case Dependencies: None

Fig.3.1.1 Use case Diagram for Restaurant Management System

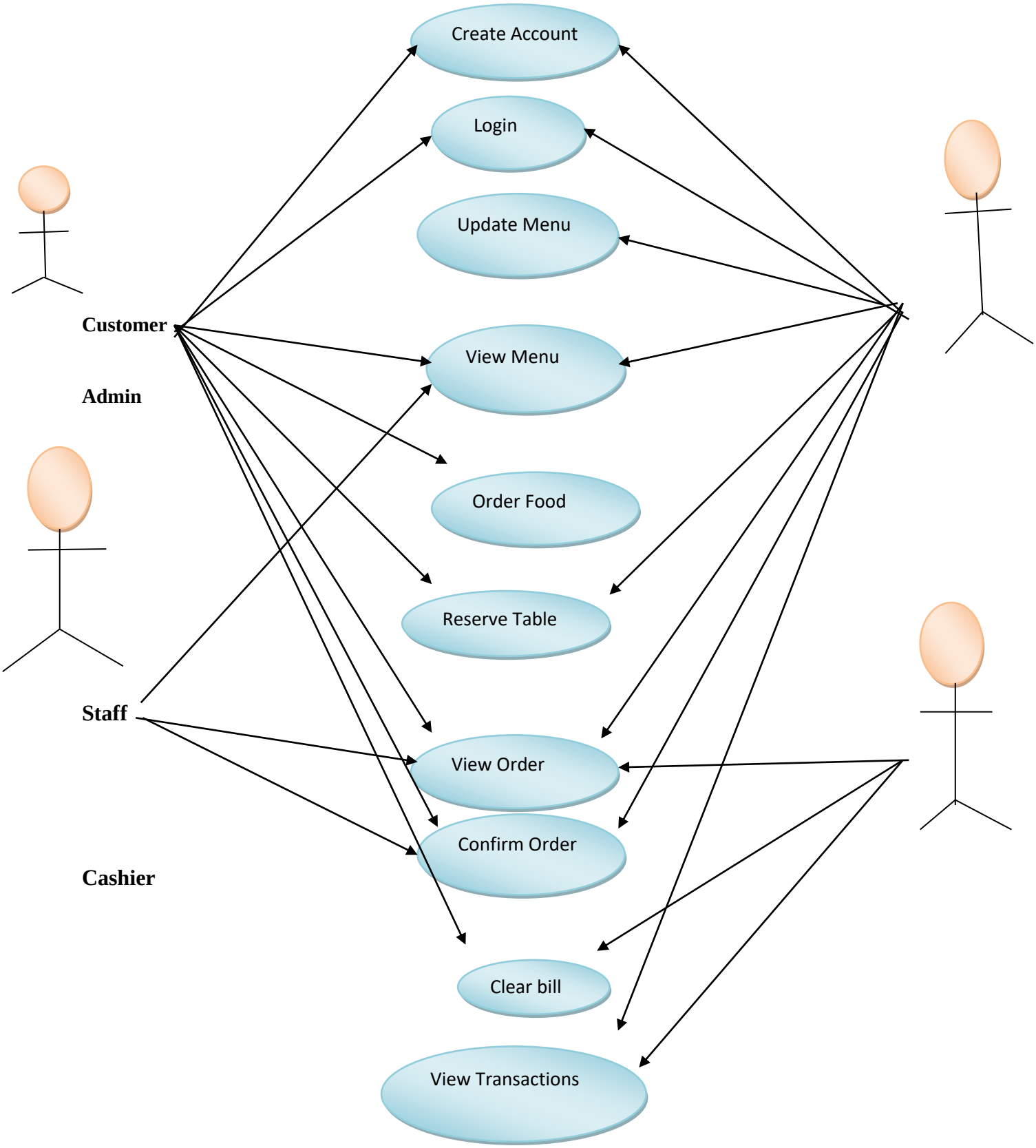


Fig.3.1.2 Activity Diagram

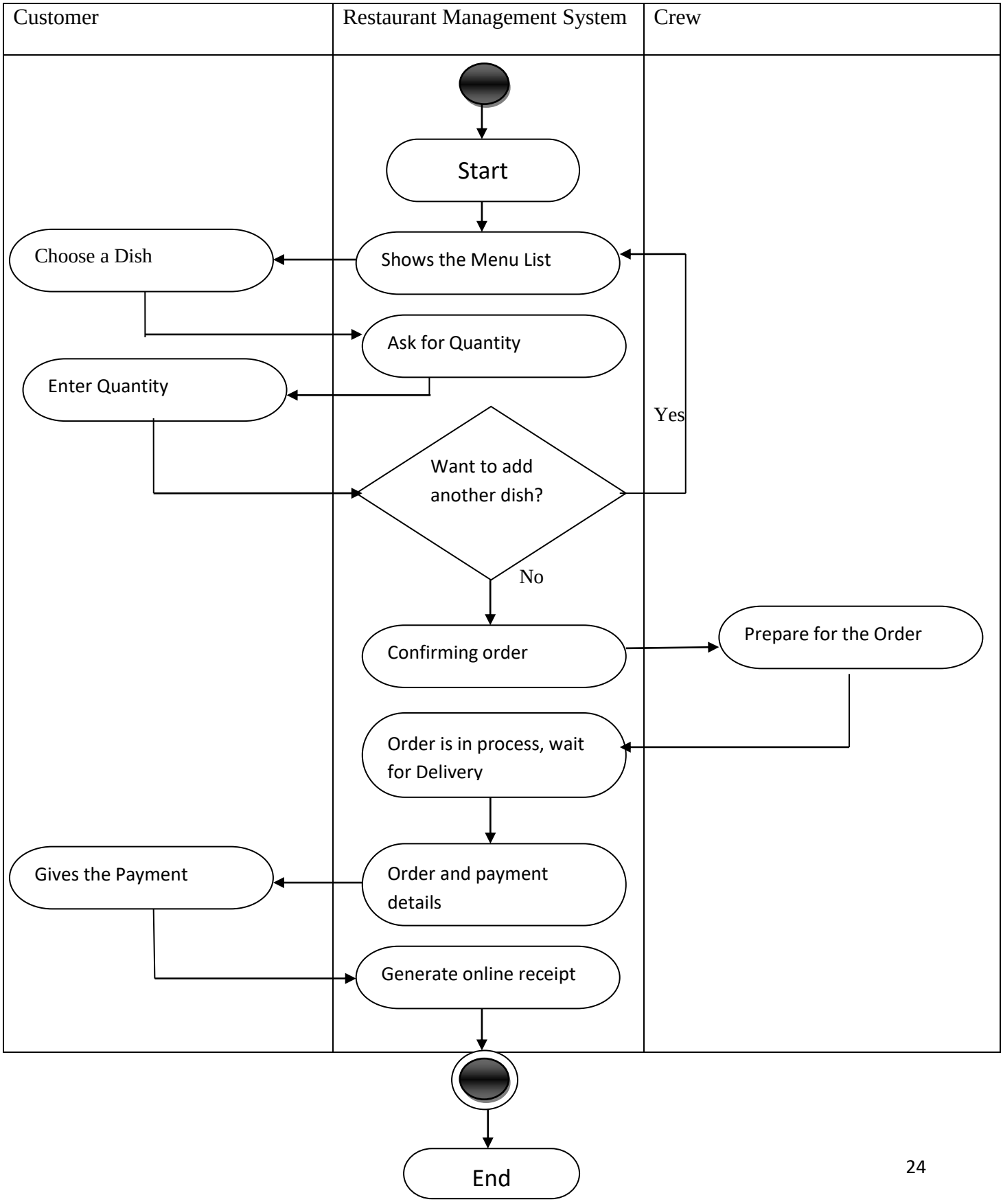


Fig.3.1.3 Class Diagram

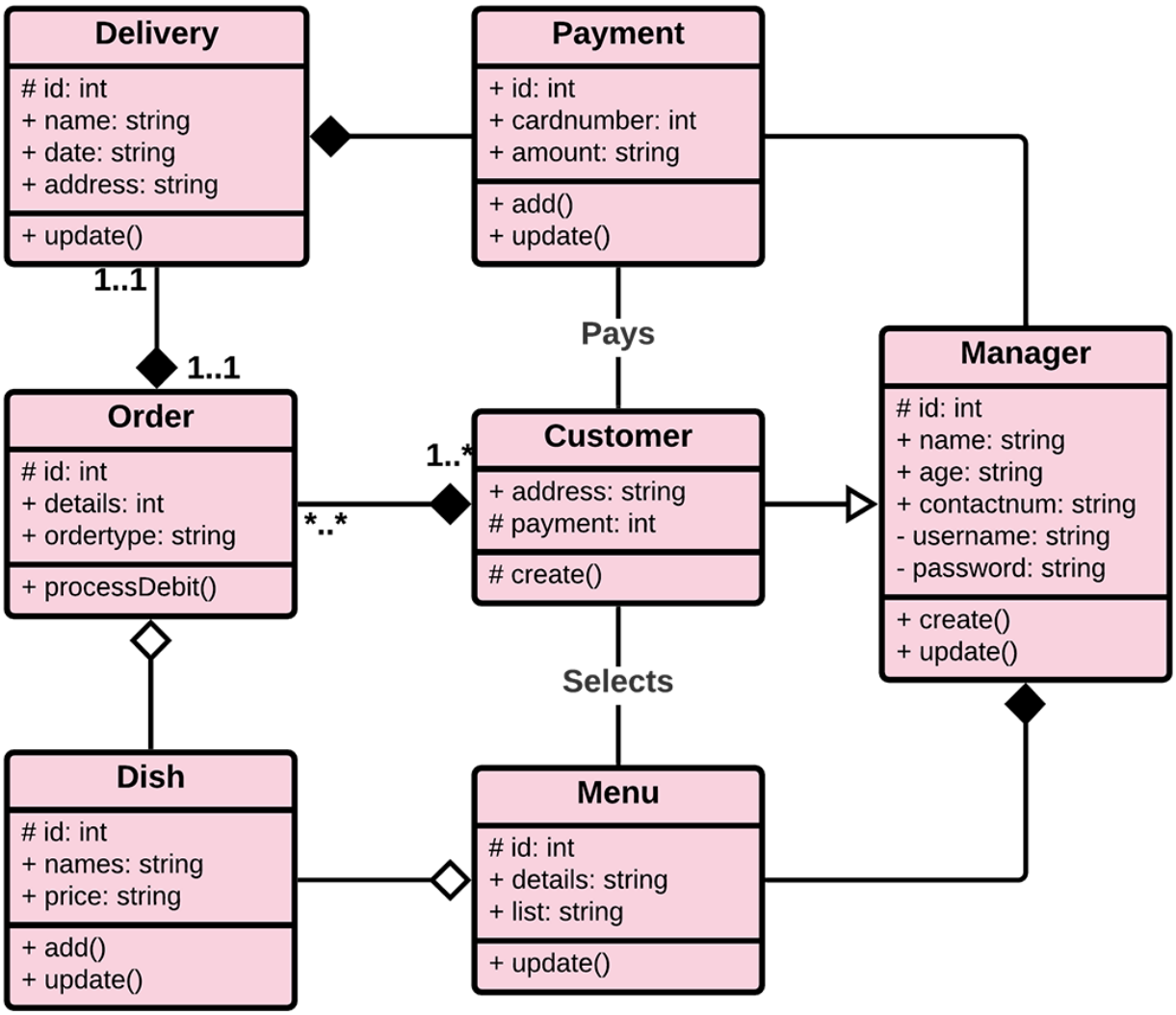
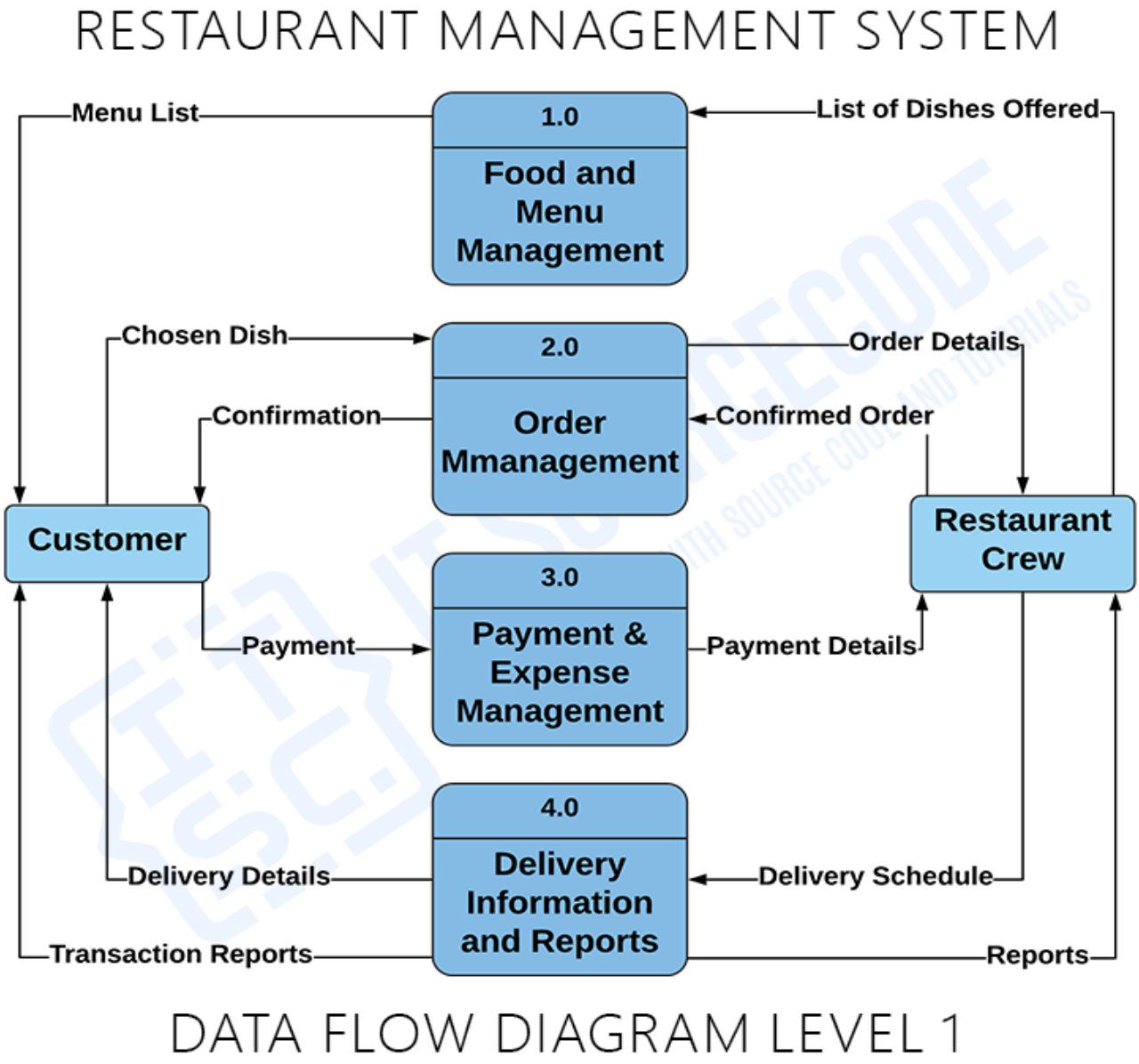


Fig.3.1.4 Data Flow Diagram



CHAPTER 4

IMPLEMENTATION

In Implementation process client environments were considered. There are environment requirements in this process. Those requirements are divided as software and hardware requirements.

4.1 Operating Environment

- Windows 11 Operating system - 32 or 64 Bit. (earlier Windows 7 was enough)
- Active Internet Connection

4.2 Language Used:

- **HTML:** HTML stands for Hyper Text Markup Language. HTML is the standard markup language for creating Web pages .HTML describes the structure of a Web page .HTML consists of a series of elements .HTML elements tell the browser how to display the content.HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.
- **CSS:** CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files.
- **JavaScript:** JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities. JavaScript can update and change both **HTML** and **CSS.** JavaScript can **calculate**, **manipulate** and **validate** data.
- **MySQL:** All database related developments were handled using this because development tools and administration objects are available with this software.

4.3 Network Architecture

Users can interact with the restaurant management system using web browser .Can be access via Internet.

4.4 Frontend code for Restaurant Management System

4.4.1 Html code:

Main.html file

```
<html>
<head>
<title>Kitchen Cafe and Resto</title>
<link rel="stylesheet" type="text/css" href="style1.css" />
<link rel="stylesheet" type="text/css" href="css/font-awesome.min.css" />
<style>
.slideshow-container {
  max-width: 800px;
  position: relative;
  margin: auto;}
 .dot {height: 10px;
  width: 10px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;}
.active {
  background-color: #717171;}
.fade {
  animation-name: fade;
  animation-duration: 1.5s;}
@keyframes fade {
```

```
  from {opacity: .4}
  to {opacity: 1}}
</style>
</head>
<body>
<div class="main">
<!----- Start Header ------>
<div class="header">
<p  style="opacity: 0.8;
   float:left;
  color: cyan;
  width: 5%;
  padding: 10px;
  margin: 5px 0 20px 0;
  font-size:15px;
 text-align: center;
font-family:cursive" id="clock"></p>
<script>
function currentTime() {
let date = new Date();
let hh = date.getHours();
let mm = date.getMinutes();
let ss = date.getSeconds();
let session = "AM";
 if(hh == 0){
 hh = 12;}
 if(hh > 12){
```

```
hh = hh - 12;
session = "PM"; }
hh = (hh < 10) ? "0" + hh : hh;
mm = (mm < 10) ? "0" + mm : mm;
ss = (ss < 10) ? "0" + ss : ss;
let time = hh + ":" + mm + ":" + ss + " " + session;
document.getElementById("clock").innerText = time;
let t = setTimeout(function(){ currentTime() }, 1000);}
currentTime();
</script>
<div class="header-left"><img src="logo.jpg" width="300" height="150" alt="Logo"></div>
<div class="header-right">
 <ul>
 <li><a href="Main.html">Home</a></li>
 <li><a href="About.html">About</a></li>
  <li><a href="Menu.html">Menu Card</a></li>
   <li><a href="bookatable.html">Reservation</a></li>
   <li><a href="#!">Review</a></li>
   <li><a href="contactus.html">Contact</a></li>
  </ul> </div>
 <div class="header-bottom">
  <div class="header-font">Eat healthy food &amp;amp; Enjoy your life.</div>
 <div class="p50_0" align="center">
 <a href="bookatable.html" class="header-btns">BOOK A TABLE</a>     
 
 <a href="Menu.html" class="header-btns">SEE THE MENU </a>
</div></div></div>
<!----- End Header ------>
```

```
<!----- Start Content ------>

<div class="content-part-1">

<div class="content-part-1-left">

 <div class="content-part-1-left-h3"Kichen Cafe & Resto></div>

 <div class="content-part-1-left-p">Kitchen Cafe & Resto strives to source local, sustainable and organic when possible. We work hard to source premium ingredients and we cook everything from scratch with love. We also do our best to pay our employees living wages (tips are shared with all employees, including kitchen staff) and to reduce our environmental footprint wherever we can. Overall, these factors translate to higher menu prices, but we hope that you find value and feel a sense of comfort in knowing that we aim to get better everyday at doing what is important to uu.</div>

 <div class="p20_0"><a href="About.html" class="content-part-1-left-btn">Read More.</a></div> </div>

<div class="content-part-1-right" align="center">

  <img src="r4.jpeg" width="700" height="434" alt="Restaurant"></div></div>

<div class="content-part-2">

<div class="content-part-2-inner">

 <div class="interior-font">Restaurant Interior</div>

  <div class="content-interior">

 <img src="r5.jpeg" width="350" height="200">

 <div class="content-interior-h3">Non AC Hall</div>

<div class="content-interior-p">Kichen Cafe and Resto provide you non ac hall.Enjoy your lunch without air conditioning</div></div>

<div class="content-interior">

<img src="r6.jpeg" width="350" height="200">

 <div class="content-interior-h3">Central AC Hall</div>

<div class="content-interior-p">Kichen Cafe and Resto provide you Ac hall.Enjoy your lunch with air conditioning. taxes are appliacble for air conditioning hall</div></div> </div>

</div>

<div class="content-part-3">

<div class="content-part-3-left"><img src="r8.jpeg" width="230" height="155"></div>
```

```
 <div class="content-part-3-right">

  <div class="content-part-3-right-h4">Chicken Biryani</div>

  <div class="price">Price : <i class="green"><span class="fa fa-inr"></span>350.00/-</i></div>

 <div class="margin_p3"><a href="#!" class="order-now">Order Now</a></div> </div>

 <div class="content-part-3-left"><img src="r9.jpeg" width="230" height="155"></div>

 <div class="content-part-3-right">

  <div class="content-part-3-right-h4">Chicken Handi</div>

  <div class="price">Price : <i class="green"><span class="fa fa-inr"></span>450.00/-</i></div>

  <div class="margin_p3"><a href="#!" class="order-now">Order Now</a></div> </div>

<!----- End Content ------>

<!------ Start Footer ------>

<div class="footer">

<div class="footer-parts">

<div class="footer-parts-h3">About Restaurant</div>

<div class="footer-parts-p">Kitchen Cafe & Resto are located across India, in metro regions including the NCR, Mumbai, Kolkata, Bengaluru, Hyderabad and Chennai, as well as numerous other tier I and II cities such as Pune, Ahmedabad, Chandigarh, Jaipur, Indore, Aurangabad, Udaipur, Vishakhapatnam, Kochi, Ludhiana, Thiruvananthapuram and Vijayawada. The company expanded internationally with hotels opening in Dubai in December 2019 and in Bhutan in February 2020. New hotels are also set to open internationally in Bhutan and Nepal.</div> </div>

<div class="footer-parts">

 <div class="footer-parts-h3">Opening Hours</div>

<div class="footer-parts-p">

  Mon - Thu : 7:00 AM - 10:00 PM <br />Friday : 7:00 AM - 11:00 PM <br /> Sat - Sun : 7:00 AM - 11:45 PM </div></div>

 <div class="footer-parts">

 <div class="footer-parts-h3">Our Location</div>

 <div class="footer-parts-p">
```

```
 MG Road,<br />Opp - Aurora Tower,<br />Pune, Maharashtra,<br /> India - 411017</div>

<div class="footer-icons">

 <ul><li><a href="#!"><i class="fa fa-facebook"></i></a></li>

        <li><a href="#!"><i class="fa fa-twitter"></i></a></li>

        <li><a href="#!"><i class="fa fa-linkedin"></i></a></li>

        <li><a href="#!"><i class="fa fa-google-plus"></i></a></li>

 </ul> </div></div>

</div>

<div class="footer-bottom">

<div class="white">&copy; <script language="javascript" type="text/javascript">document.write(new Date().getFullYear());</script>.Kitchen Cafe & Resto <a href="#" target="_blank">Ltd</a>.</div></div>

<!------ End Footer ------->

<div class="clearfix"></div></div></body></html>
```

4.4.2 CSS code

style1.css file

```
body{ font-family:Arial, Helvetica, sans-serif; margin:0px auto !important;}

.main{

width:1500px;

margin:0px auto;

 -webkit-box-shadow: 0 6px 10px 0 rgba(0, 0, 0, 0.14), 0 1px 18px 0 rgba(0, 0, 0, 0.12), 0 3px 5px -1px rgba(0, 0, 0, 0.3) !important;

box-shadow: 0 6px 10px 0 rgba(0, 0, 0, 0.14), 0 1px 18px 0 rgba(0, 0, 0, 0.12), 0 3px 5px -1px rgba(0, 0, 0, 0.3) !important;

}

.p10_0{ padding:10px 0px;} .p20_0{ padding:20px 0px;}.p30_0{ padding:30px 0px;}

.p40_0{ padding:40px 0px;} .p50_0{ padding:50px 0px;}

.clearfix{ clear:both;}

/********** Start Header **********/

.header{ width:1500px; float:left; padding:10px 0px; background:url("r24.jpeg"); height:400px; background-attachment:fixed;}

.header-left{ width:360px; float:left; padding:15px 20px;}

.header-right{ width:800px; float:left; padding-top:60px;}

.header-right ul{ list-style:none; padding:0px;}

.header-right ul li{ padding:5px 15px; cursor:pointer; margin:0px 10px; float:left; border:1px solid #FFF; border-radius:2px;}

.header-right ul li:hover{border:1px solid #7db549; background:#7db549;}

.header-right ul li a{ font-size:14px; color:#FFF; text-decoration:none;}

.header-right ul li a:hover{ color:#FFF;}

.white{ color:#FFF; text-decoration:none;}

.white a{ color:#FFF; text-decoration:none;}

.white a:hover{ color:#CCC;}
```

```
.header-bottom{ width:1200px; float:left; padding:75px 0px 20px;}

.header-font{ color:#FFF; text-align:center; font-size:48px; text-transform:uppercase; font-weight:bold; padding:0px;}

.header-btns{ color:#FFF; font-size:16px; font-weight:500; border:1px solid #FFF;

padding:10px 30px; text-decoration:none; border-radius:2px;}

.header-btns:hover{ background:#7db549; color:#FFF; border:1px solid #7db549;}

/********** End Header **********/

/********** Start Content Part 1 *********/

.content-part-1{ width:1200px; float:left;}

.content-part-1-left{ width:480px; float:left; padding:100px 0px 20px 20px;}

.content-part-1-left-h3{ font-size:28px; text-align:center; color:#7db549; font-weight:500;}

.content-part-1-left-p{font-size:14px; text-align:justify; color:#000; font-weight:500; line-height:26px; padding:10px 12px;}

.content-part-1-left-btn{ font-size:16px; font-weight:500; border:1px solid #7db549; color:#FFF;

padding:10px 30px; text-decoration:none; border-radius:2px; background:#7db549; margin:0px 10px;}

.content-part-1-left-btn:hover{ background:#689242;}

.content-part-1-right{ width:700px; float:left;}

/********** End Content Part 1 *********/

/********** Start Content Part 2 *********/

.content-part-2{ width:1500px; float:left; padding:10px 0px; background:url(r7.jpeg) no-repeat; height:550px; background-position:center  bottom; background-attachment:fixed;}

.interior-font{ font-size:36px; text-transform:uppercase; padding:20px 0px 30px; font-weight:600;

text-align:center; color:#000; text-shadow: 2px 2px 3px #333;}

.content-part-2-inner{background-color: rgba(255,255,255,.5); padding:20px 0px 20px; margin:50px 200px;

border-radius:3px; width:800px; float:left;}
```

```
.content-interior{ width:350px; padding:10px 25px; float:left;}

.content-interior-h3{ color:#000; font-size:18px; text-shadow: 2px 2px 3px #333; padding:10px 0px; text-align:center;}

.content-interior-p{ color:#000; padding:5px 0px 10px; font-size:14px; line-height:26px; text-align:justify;}

/********** End Content Part 2 *********/

/********** Start Content Part 3 *********/

.content-part-3{ width:1200px; float:left; padding:20px 0px;}

.content-part-3-left{ width:230px; float:left; padding:10px;}

.content-part-3-right{ width:330px; float:left; padding:5px 10px 10px; height:170px; line-height:26px; text-align:justify;}

.content-part-3-right-h4{ font-weight:600; color:#000; padding:0px;}

.margin_5{ margin:5px 0px; float:left;}

.margin_p3{ margin:10px 0px 5px;}

.price{color:#000; padding:5px 0px; font-weight:bold; font-size:15px;}

.green{ color:#7db549; font-weight:bold; font-size:16px;}

.order-now{ color:#FFF; background:#7db549; padding:5px 15px; border-radius:2px; text-decoration:none; margin:5px auto;}

.order-now:hover{background:#689242;}

/********** End Content Part 3 *********/

/********** Start Footer *********/

.footer{ width:1500px; float:left; padding:10px 0px; background:deepskyblue;}

.footer-parts{ width:500px; padding:20px 40px; float:left;}

.footer-parts-h3{ font-size:15px; font-weight:400; color:#FFF; padding:10px 0px;}

.footer-parts-p{ color:#000; line-height:20px; text-align:justify; color:#FFF;}

.footer-icons{list-style:none;}

.footer-icons ul{list-style:none; padding:0px;}

.footer-icons ul li{float:left; padding:0px 25px 0px 5px; text-transform:uppercase;}

.footer-icons ul li a{color:#FFF; font-weight:600; font-size:18px;}
```

```
.footer-icons ul li a:hover{ color:#CCC;}

.footer-bottom{ width:1479px; float:left; padding:10px; text-align:center; color:#FFF; font-size:13px; background:#000;}

.white{ color:#FFF; text-decoration:none;}

.white a{ color:#FFF; text-decoration:none;}

.white a:hover{ color:#CCC;}

/********** End Footer *********/
```

Style2.css file

```
body {font-family: Helvetica, sans serif;}

.number {

padding: .3em;

text-align: center;

display: inline-block;

width: 1em;

height: 1em;

background-color: seashell;

border-radius: 50%;

}

h1 {

font-family: 'Life Savers', cursive;

color:  cyan;

text-align: center;

margin: 1em;

font-size: 2.5em;

text-shadow: 2px 2px 3px grey;}
```

```
form {
display: flex;
flex-direction: column;
background: thistle;
margin: 0 auto 2em;
border-radius: 1em;
padding: 1em 0;}
input,
textarea {
border: none;
padding: .5em;
border-radius: .25em;}
input:focus,
input:hover,
textarea:focus,
textarea:hover {
background: seashell;
border-bottom: 2px solid purple;
outline: none;}
input[type="radio"],
input[type="checkbox"] {
        margin-right: .25em;
        width: 1em;
        height: 1em;}
fieldset {
        border: none;
        margin: .5em;}
```

```
legend {
        font-family: 'Comfortaa', cursive;
        font-size: 1.4em;
        font-weight: bold;
        padding-bottom: .5em;}
label,
.button {
        font-family: 'Baloo Chettan 2', cursive;}
.button {
        box-shadow: 1px 1px 3px black;
        border: none;
        background-color: #DFDBE5;
        background-image:
url("data:image/svg+xml,%3Csvgxmlns='http://www.w3.org/2000/svg' width='4' height='4' viewBox='0 0 4 4'%3E%3Cpath fill='%239C92AC' fill-opacity='0.4' d='M1 3h1v1H1V3zm2-2h1v1H3V1z'%3E%3C/path%3E%3C/svg%3E");
        font-size: 1em;
        align-self: center;
       padding: .7em;
        color: black;}
.button:hover {
        cursor: pointer;
        background: radial-gradient(black, purple);
        color: white;
        box-shadow: inset 2px 2px 5px 0px rgba(0,0,0,0.75);}
#name,
#mail,
```

```
#phone,
textarea {
        margin: .5em 0;
        box-shadow: 1px 1px 2px grey;}
.order {
        display: none;
        text-align: center;}
img {max-width: 100%;}
@media (min-width: 550px) {
        form {
                max-width: 50%;}}
```

4.4.3 JavaScript Code

separate.js file

```
const submitForm = () => {
        document.querySelector(".main").style.display="none";
        document.querySelector(".order").style.display="block";
};
```

CHAPTER 5

MODULES

Fig.5.1 Home Page

Home page contains three section i.e. header part, content part and footer part.

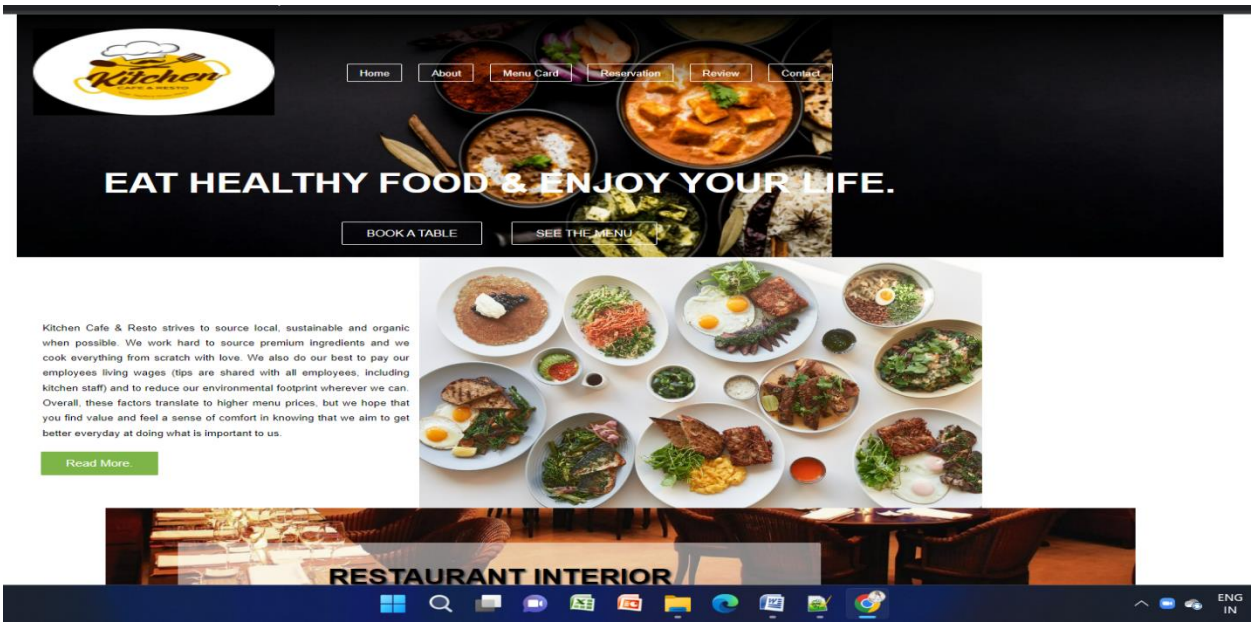


Fig.5.1.1 Header part:

used <header> tag for header part.Header contain s navigations ,logo, text.

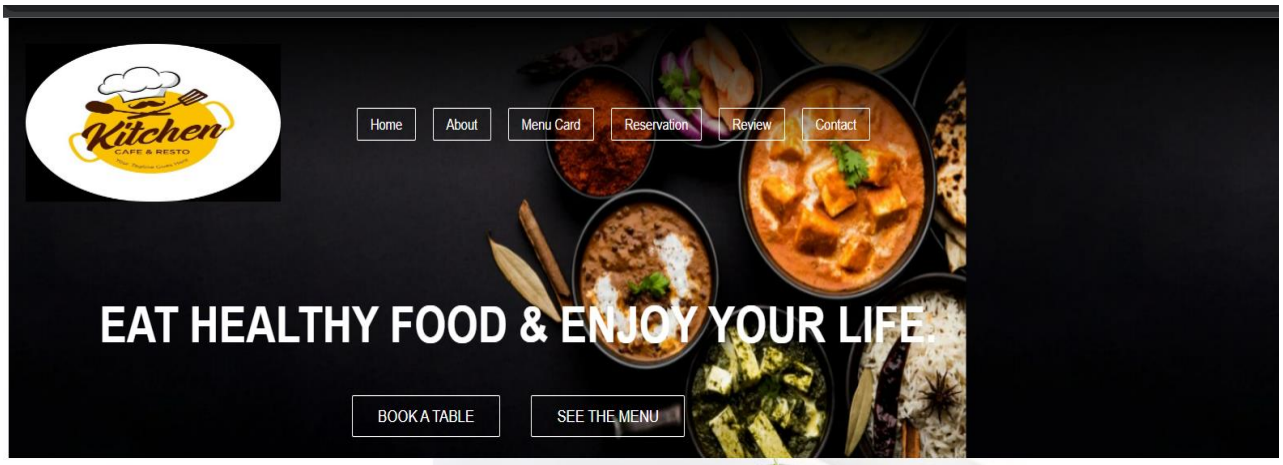


Fig.5.1.2 Content part:

content part contains some information and images about restaurant

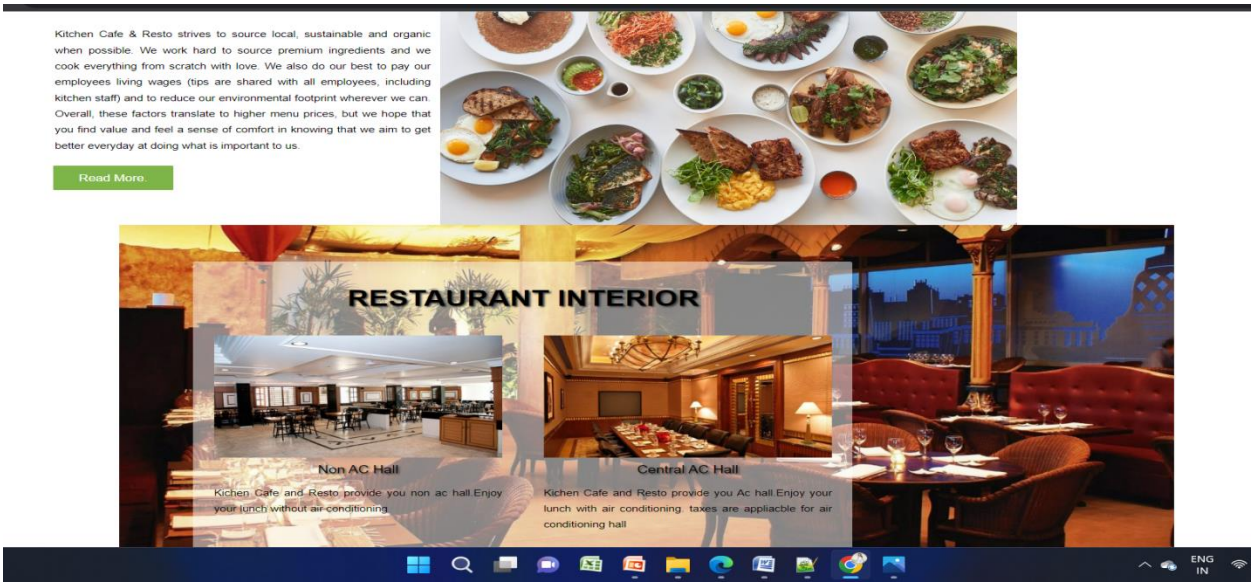


Fig.5.1.3 Footer part:

This is the footer section

Fig.5.2 About Page

About page contains information about restauarant.

Fig.5.3 Menu card:

When customer is required to order food online can visit the website and first they can see menu card. They can select from a menu card if they wish.

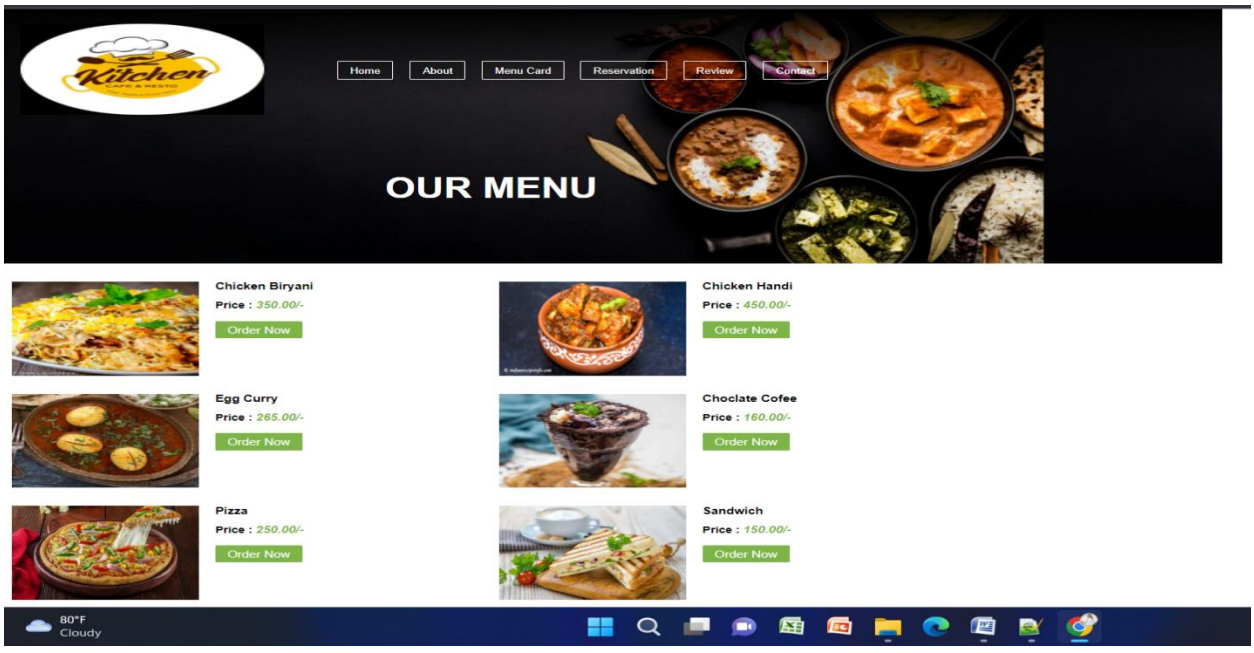


Fig.5.4 Reservation

Customer can confirm a table for their party at a specified time.

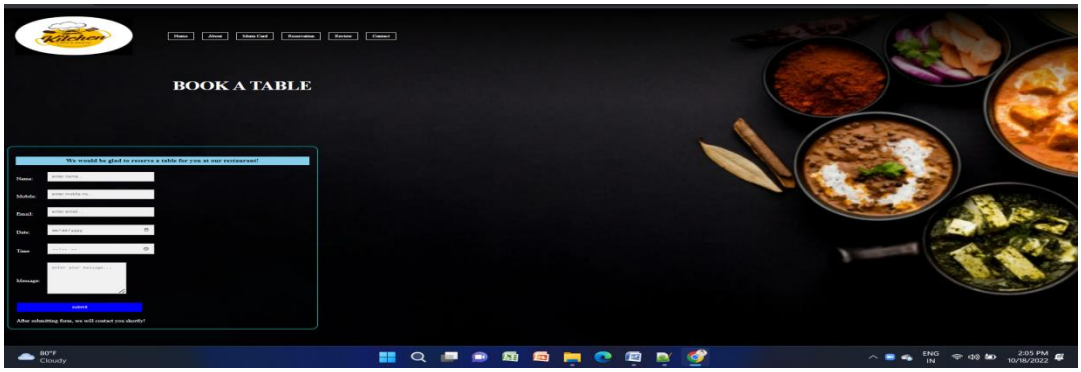


Fig.5.5 Contact Us page

This is the contact form.Customers fill conact form and directly contact to the restaurant manager. Customers gives Feedback about service provide by restaurant.

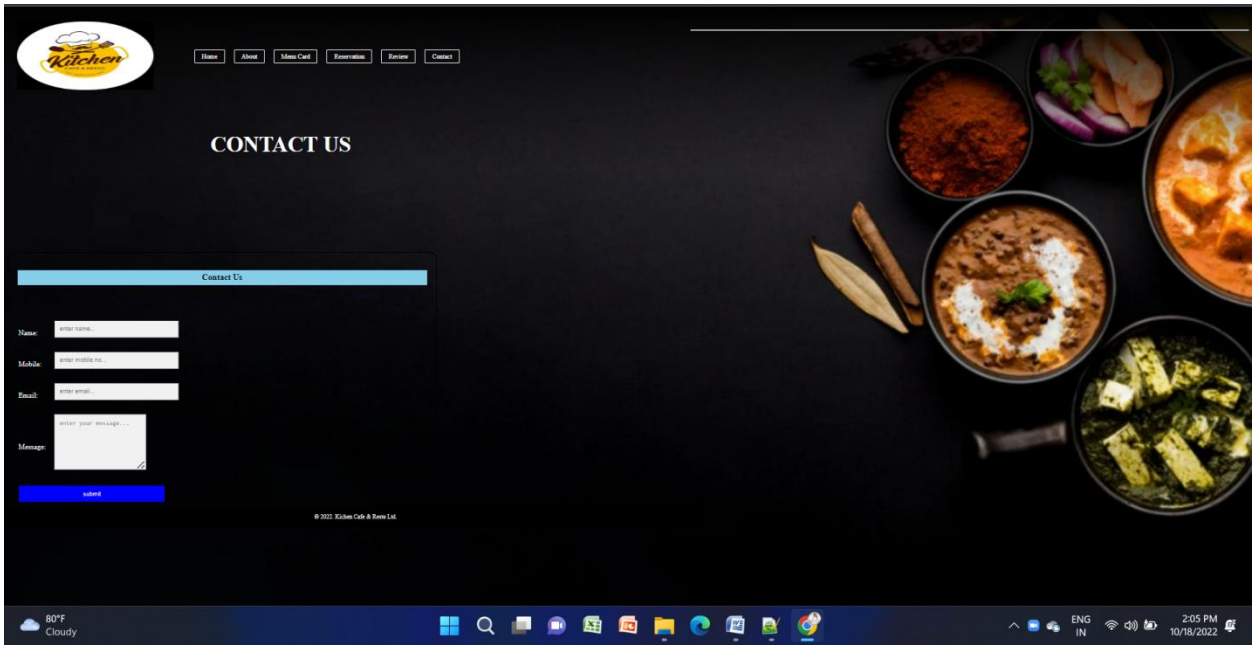
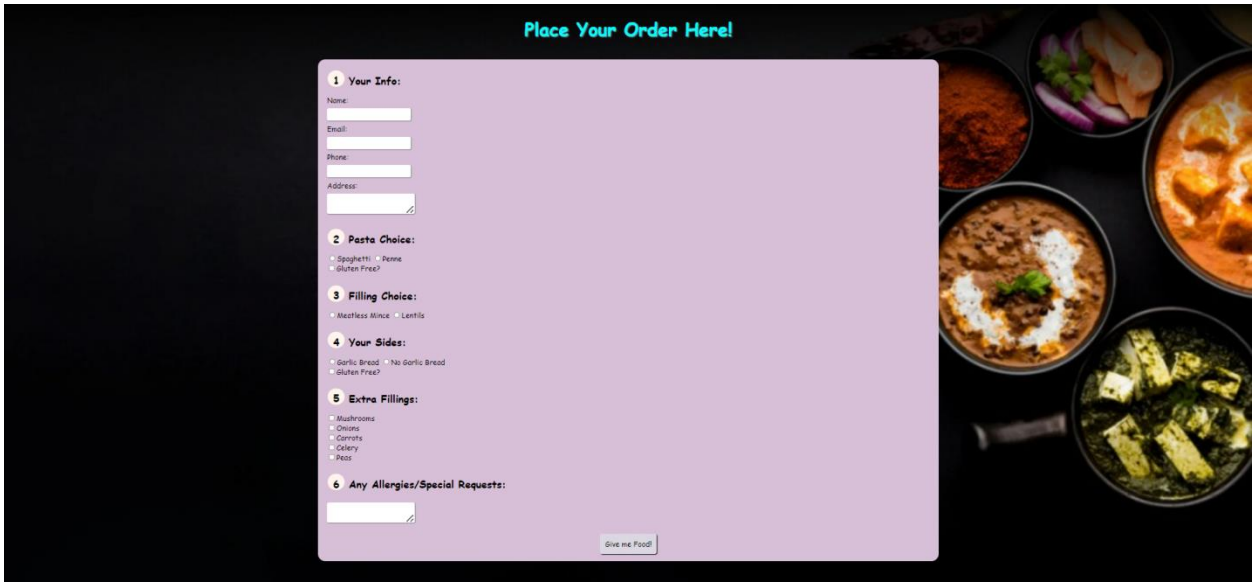


Fig.5.6 Orderfood

This is orderfood page. Cutomer can order food.



CHAPTER 6

SYSTEM TESTING

6.1 Introduction

Software Testing is a critical component of software quality assurance that represents the ultimate analysis of specification, design, and code generation of software product. The Testing method is basically combined with Verification and Validation. Validation refers to testing whether the system satisfies the requirements while Verification refers to whether the system implements the specifies functions properly. A basic goal of test evaluation is determining whether the promises about the invention by the supplier and the requirements of the customer are met on an acceptable level.

6.1.1 Levels of Testing

Level 1: Unit Testing

Unit testing is the first level of software testing, which is used to test if software modules are satisfying the given requirement or not. The first level of testing involves analyzing each unit or an individual component of the software application.

Unit testing is also the first level of functional testing. The primary purpose of executing unit testing is to validate unit components with their performance.

Level 2: Integration Testing

The second level of software testing is the integration testing. The integration testing process comes after unit testing. It is mainly used to test the data flow from one module or component to other modules.

Level 3: System Testing

The third level of software testing is system testing, which is used to test the software's functional and non-functional requirements. It is end-to-end testing where the testing environment is parallel to the production environment. In the third level of software testing, we will test the application as a whole system. To check the end-to-end flow of an application or the software as a user is known as System testing.

Level 4: Acceptance Testing

The last and fourth level of software testing is acceptance testing, which is used to evaluate whether a specification or the requirements are met as per its delivery. The software has passed

through three testing levels (Unit Testing, Integration Testing, System Testing). Some minor errors can still be identified when the end-user uses the system in the actual scenario.

In simple words, we can say that Acceptance testing is the squeezing of all the testing processes that are previously done.

- Alpha Testing: Testing or verifying product at organization.
- Beta Testing: Checking the product in customer's place

6.1.2 Black Box Testing:

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.

Black Box Testing Types

The following are the several categories of black box testing:

- Functional Testing
- Regression Testing
- Nonfunctional Testing (NFT)

Functional Testing: This black box testing type is related to the functional requirements of a system; it is done by software testers.

Regression Testing: It ensures that the newly added code is compatible with the existing code. In other words, a new software update has no impact on the functionality of the software. This is carried out after a system maintenance operation and upgrades.

Nonfunctional Testing: Nonfunctional testing is also known as NFT. This testing is not functional testing of software. It focuses on the software's performance, usability, and scalability.

Black Box Testing Techniques

- **Equivalence Class Testing:** It is used to minimize the number of possible test cases to an optimum level while maintains reasonable test coverage.
- **Boundary Value Testing**: Boundary value testing is focused on the values at boundaries. This technique determines whether a certain range of values are acceptable by the system or not. It is very useful in reducing the number of test cases. It is most suitable for the systems where an input is within certain ranges.
- **Decision Table Testing:** A decision table puts causes and their effects in a matrix. There is a unique combination in each column.

Advantages of Black Box Testing:

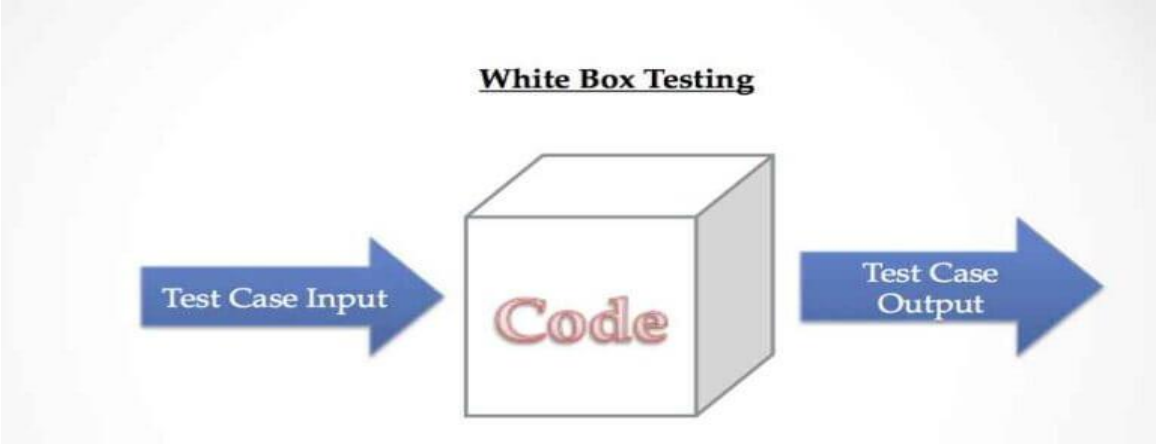
- The tester does not need to have more functional knowledge or programming skills to implement the Black Box Testing.
- It is efficient for implementing the tests in the larger system.
- Tests are executed from the user's or client's point of view.
- Test cases are easily reproducible.
- It is used in finding the ambiguity and contradictions in the functional specifications.

Disadvantages of Black Box Testing:

- There is a possibility of repeating the same tests while implementing the testing process.
- Without clear functional specifications, test cases are difficult to implement.
- It is difficult to execute the test cases because of complex inputs at different stages of testing.
- Sometimes, the reason for the test failure cannot be detected.
- Some programs in the application are not tested.
- It does not reveal the errors in the control structure.
- Working with a large sample space of inputs can be exhaustive and consumes a lot of time.

6.1.3 White Box Testing

White Box Testing is a testing technique in which software's internal structure, design, and coding are tested to verify input-output flow and improve design, usability, and security. In white box testing, code is visible to testers, so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing, and Glass box testing.



Working process of white box testing:

- **Input:** Requirements, Functional specifications, design documents, source code.
- **Processing:** Performing risk analysis for guiding through the entire process.
- **Proper test planning:** Designing test cases so as to cover entire code. Execute rinse-repeat until error-free software is reached. Also, the results are communicated.
- **Output:** Preparing final report of the entire testing process.

Following are important White Box Testing Techniques:

- Statement Coverage
- Decision Coverage
- Branch Coverage
- Condition Coverage
- Multiple Condition Coverage
- Finite State Machine Coverage
- Path Coverage
- Control flow testing
- Data flow testing

Advantages:

- White box testing is very thorough as the entire code and structures are tested.
- It results in the optimization of code removing error and helps in removing extra lines of code.
- It can start at an earlier stage as it doesn't require any interface as in case of black box testing.
- Easy to automate.

Disadvantages:

- Main disadvantage is that it is very expensive.
- Redesign of code and rewriting code needs test cases to be written again.
- Testers are required to have in-depth knowledge of the code and programming language as opposed to black box testing.
- Missing functionalities cannot be detected as the code that exists is tested.
- Very complex and at times not realistic.

6.1.4 Gray Box Testing:

Gray box testing is a software testing technique to test a software product or application with partial knowledge of internal structure of the application. The purpose of grey box testing is to search and identify the defects due to improper code structure or improper use of applications.

Gray Box Testing is a software testing method, which is a combination of both White Box Testing and Black Box Testing method.

It is primarily used in integration testing and penetration testing.

Techniques used for Grey box Testing are-

- **Matrix Testing**
  This testing technique comes under Grey Box testing. It defines all the used variables of a particular program. In any program, variable are the elements through which values can travel inside the program. It should be as per requirement otherwise, it will reduce the readability of the program and speed of the software. Matrix technique is a method to remove unused and uninitialized variables by identifying used variables from the program.
- **Orthogonal Array Testing**
  The purpose of this testing is to cover maximum code with minimum test cases. Test cases are designed in a way that can cover maximum code as well as GUI functions with a smaller number of test cases.
- **Pattern Testing**

Pattern testing is applicable to such type of software that is developed by following the same pattern of previous software. In these type of software possibility to occur the same type of defects. Pattern testing determines reasons of the failure so they can be fixed in the next software.

6.2 Testing Procedure

Software testing and implementation are iterative processes. Most of the time both stages work as simultaneous system components. The propose system was tested while the development was ongoing. Following iterative RUP development life cycle made it possible to test Iterative increments of the software.

Structural testing techniques were followed in this phase including “White Box “testing which tests “how a program/system does something”. Functional testing techniques were used, which includes “Black box” testing which tests the behavior of a system or program. These techniques were exploited in different testing levels like unit testing.

System followed object oriented methodology, so object oriented testing was also carried out in this phase. Individual operations associated with object were tested initially, followed by testing individual classes and clusters of objects, and finally test the system as whole. User acceptance testing was completed in client site, participation of few staff members.

6.3 Test Plan and Test Cases

Implementation stage starts the testing process. In development stage code is reviewed. Test

Plan included all phases of testing and also used as a guide for the overall testing process. The

Test plan was designed before the implementation of the system. Test Objectives, Test Schedule

Test Logistics and Test Strategies are included in Test plan. Test cases are highly considered in here.

Test cases were created according to the designed test plan. That contains data, procedure, and expected result and represents which use to system or part of the system run. To reduce complexity of the testing process test cases were designed for each module independently. The following tables specify some test cases.

Manual Testing method and procedure used for testing rather than automation tools and technologies.

6.3.1 Test Cases for Restaurant Management System

Test scenario TS_01 – Home page						
Test case id	Test case	Steps	Input	Actual result	Expected result	Status
TC_01	Check Url for home page	1.Go to url 2.click enter	file:///C:/Users/ADMIN/OneDrive/Desktop/RestaurantManagementSystem_Nikita_Ovhal/Main.html	Display Home page	Same as Expected result	Pass
Test scenario TS_02 – About page						
Test case id	Test case	Steps	Input	Actual result	Expected result	Status
TC_02	Check url for about page	1.Go to home page 2.click About and 3.click enter	file:///C:/Users/ADMIN/OneDrive/Desktop/RestaurantManagementSystem_Nikita_Ovhal/About.html	Navigate to About us page	Same as Expected result	Pass
Test scenario TS_03– Menu page						
Test case id	Test case	Steps	Input	Actual result	Expected result	Status
TC_03	Check url for menu	1.Go to home page 2.click Menu and 3.click enter	file:///C:/Users/ADMIN/OneDrive/Desktop/RestaurantManagementSystem_Nikita_Ovhal/Menu.html	Navigate to Menu page	Same as Expected result	Pass
Test scenario TS_04 – Reservation page						
Test case id	Test case	Steps	Input	Actual result	Expected result	Status
TC_04	Check for reservation page url	1.Go to home page 2.click Reservation 3.click enter	file:///C:/Users/ADMIN/OneDrive/Desktop/RestaurantManagementSystem_Nikita_Ovhal/bookatable.html	Navigate to Reservation page	Same as Expected result	Pass
TC_05	Check for user entering	1.Go to reservation page 2.Enter name	Nikita Ovhal	It should display message	Same as Expected result	Pass

	details	3.Enter mobile no 4.Enter email 5.Enter date 6.Enter time 7.Enter message 8.And click on Submit	9657851165 nikitaovhal1297@gmail.com 10/01/2022 12.07 PM	Your details submitted successfully		
Test scenario TS_05 – Contact us page						
Test case id	Test case	Steps	Input	Actual result	Expected result	Status
TC_05	Check for contact us page url	1.Go to home page 2.click contact 3.click enter	file:///C:/Users/ADMIN/OneDrive/Desktop/RestaurantManagementSystem_Nikita_Ovhal/contactus.html	Navigate to contact us page	Same as Expected result	Pass
TC_06	Check for user entering contact details	1.Go to contact us page 2.Enter name 3.Enter mobile no 4.Enter email 5.Enter message 6.And click on Submit	Nikita Ovhal 9657851165 nikitaovhal1297@gmail.com	It should display Your details submitted successfully	Same as Expected result	Pass
Test scenario TS_06 – Order food page						
Test case id	Test case	Steps	Input	Actual result	Expected result	Status
TC_08	Check for order food page	1.Go to menu card 2.select menu 3.click order	file:///C:/Users/ADMIN/OneDrive/Desktop/RestaurantManagementSystem_Nikita_Ovhal/Menu.html	Display order page	Same as Expected result	Pass
TC_09	Check user place order	1.Go to Menu and 2.select food item 3.and click order now 3.Enter name 4.Enter email 5.Enter phone 6.Enter address 7.Select choices as per customer requirement 8.And click on Give me food	Nikita Ovhal nikitaovhal1297@gmail.com 9657851165 Pune	It should display thanks for your order	Same as Expected result	Pass

6.4 Automation Testing

Automation testing refers to the automatic testing of the software in which developer or tester write the test script once with the help of testing tools and framework and run it on the software. The test script automatically test the software without human intervention and shows the result (either error, bugs are present or software is free from them).

Automation testing needs manual effort when creating initial scripts, and further process is performed automatically to compare the actual testing result with expected results.

In automation testing, the test automation engineer will write the test script or use the automation testing tools to execute the application. On the other hand, in manual testing, the test engineer will write the test cases and implement the software on the basis of written test cases.

In test automation, the test engineer can execute repetitive tasks and other related tasks. In manual testing, it is a tedious process to implement the repetitive take again and again.

In other words, we can say that the main concentration of test Automation is to change the manual human activity with systems or devices.

The automation testing process is a time-saving process as it spends less time in exploratory testing and more time in keeping the test scripts whereas enhancing the complete test coverage.

6.5 Selenium

Selenium is one of the most widely used open source Web UI (User Interface) automation testing suite. It was originally developed by Jason Huggins in 2004 as an internal tool at Thought Works. Selenium supports automation across different browsers, platforms and programming languages.

Selenium can be easily deployed on platforms such as Windows, Linux, Solaris and Macintosh. Moreover, it supports OS (Operating System) for mobile applications like iOS, windows mobile and android.

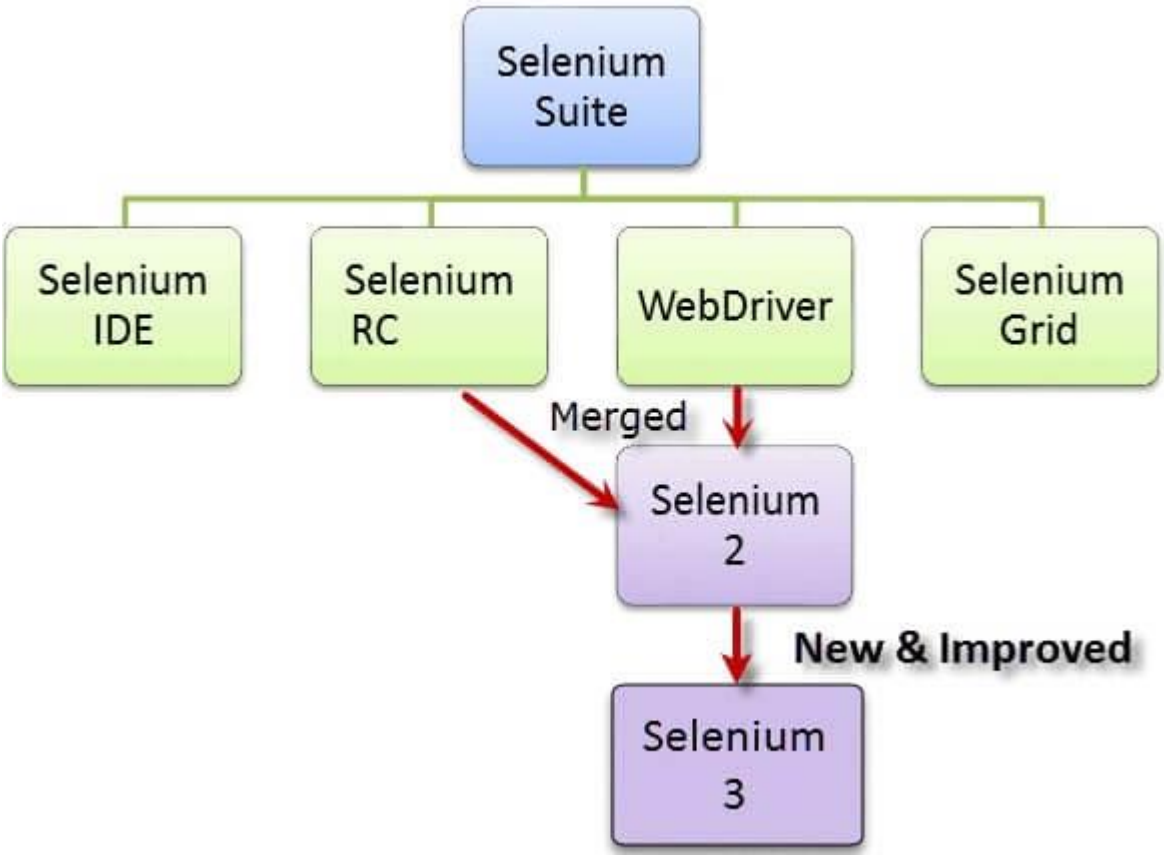
Selenium supports a variety of programming languages through the use of drivers specific to each language. Languages supported by Selenium include C#, Java, Perl, PHP, Python and Ruby. Currently, Selenium Web driver is most popular with Java and C#. Selenium test scripts can be coded in any of the supported programming languages and can be run directly in most modern web browsers. Browsers supported by Selenium include Internet Explorer, Mozilla Firefox, Google Chrome and Safari.

Selenium can be used to automate functional tests and can be integrated with automation test tools such as Maven, Jenkins, & Docker to achieve continuous testing. It can also be integrated with tools such as TestNG, & JUnit for managing test cases and generating reports.

6.5.1 Selenium Tool Suite

Selenium Software is not just a single tool but a suite of software, each piece catering to different Selenium QA testing needs of an organization. Here is the list of tools

- Selenium Integrated Development Environment (IDE)
- Selenium Remote Control (RC)
- Web Driver
- Selenium Grid



What is Selenium IDE?

Selenium Integrated Development Environment (IDE) is the simplest framework in the Selenium suite and is the easiest one to learn. It is a Chrome and Firefox plugin that you can install as easily as you can with other plugins. However, because of its simplicity, Selenium IDE should only be used as a prototyping tool. If you want to create more advanced test cases, you will need to use either Selenium RC or Web Driver.

What is Selenium Remote Control (Selenium RC)?

Selenium RC was the flagship testing framework of the whole Selenium project for a long time. This is the first automated web testing tool that allows users to use a programming language **they** prefer. As of version 2.25.0, RC can support the following programming languages:

- Java
- C#
- PHP
- Python
- Perl
- Ruby

What is WebDriver?

The WebDriver proves to be better than Selenium IDE and Selenium RC in many aspects. It implements a more modern and stable approach in automating the browser's actions. WebDriver, unlike Selenium RC, does not rely on JavaScript for Selenium Automation Testing. It controls the browser by directly communicating with it.

The supported languages are the same as those in Selenium RC.

- Java
- C#
- PHP
- Python
- Perl

What is Selenium Grid?

Selenium Grid is a tool used together with Selenium RC to run parallel tests across different machines and different browsers all at the same time. Parallel execution means running multiple tests at once.

Features:

- Enables simultaneous running of tests in multiple browsers and environments.
- Saves time enormously.
- Utilizes the hub-and-nodes concept. The hub acts as a central source of Selenium commands to each node connected to it.

What is Cucumber?

Cucumber is a testing tool that supports Behavior Driven Development (BDD). It offers a way to write tests that anybody can understand, regardless of their technical knowledge. In BDD, users (business analysts, product owners) first write scenarios or acceptance tests that describe the behavior of the system from the customer's perspective, for review and sign-off by the product owners before developers write their codes. Cucumber framework uses Ruby programming language.

Why Cucumber?

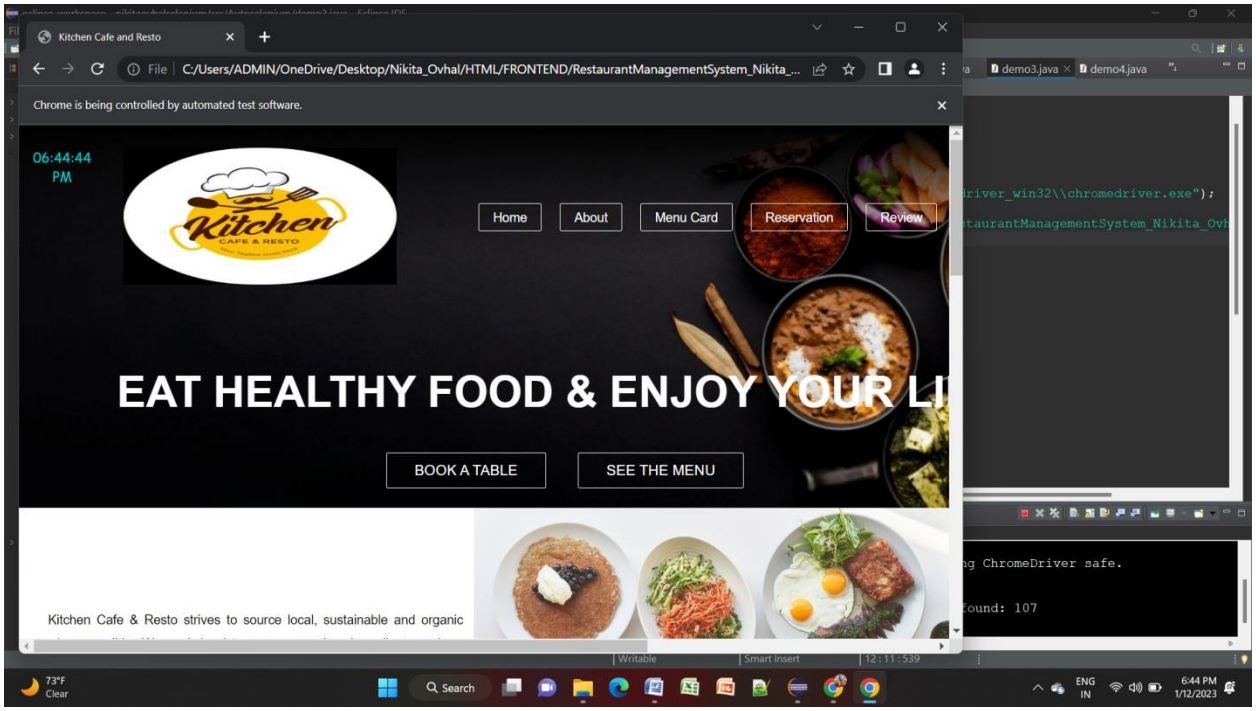
1. **Multiple Language Support:** It supports almost all popular languages like Java, .net, JavaScript, Ruby, PHP, Python, etc.
2. **Code Reusability:** Due to simple test script architecture, Hooks, Background, and Data Table, Cucumber provides code reusability. Also, Exact Gherkin Steps we can use in other test scenarios.
3. **Easy to Understand:** The test case writing is straightforward and understandable. It works as a bridge between business and technical language, and this bridge is sustainable because of test cases written in plain English text. It uses Simple grammar defined by a language called Gherkin. Even your manual testers can write test scenarios. Actually, Not just a Manual tester; even BA and other stakeholders can write stories in Gherkin Language.
4. **Acceptance Testing Tool:** It bridges the gap between business people and developers. It is more than a test automation tool; it's a collaboration tool, or you can say It is an acceptance testing tool. Cucumber enables the direct automation of the specification, which means that anyone can see, at a glance, what functionality has been implemented and what hasn't.
5. **Less Documentation:** You can save a lot of documentation time when your Business Analyst writes your stories in the Gherkin language. Your manual tester also writes in test cases in Cucumber-friendly language and the same scenarios you use for automation.

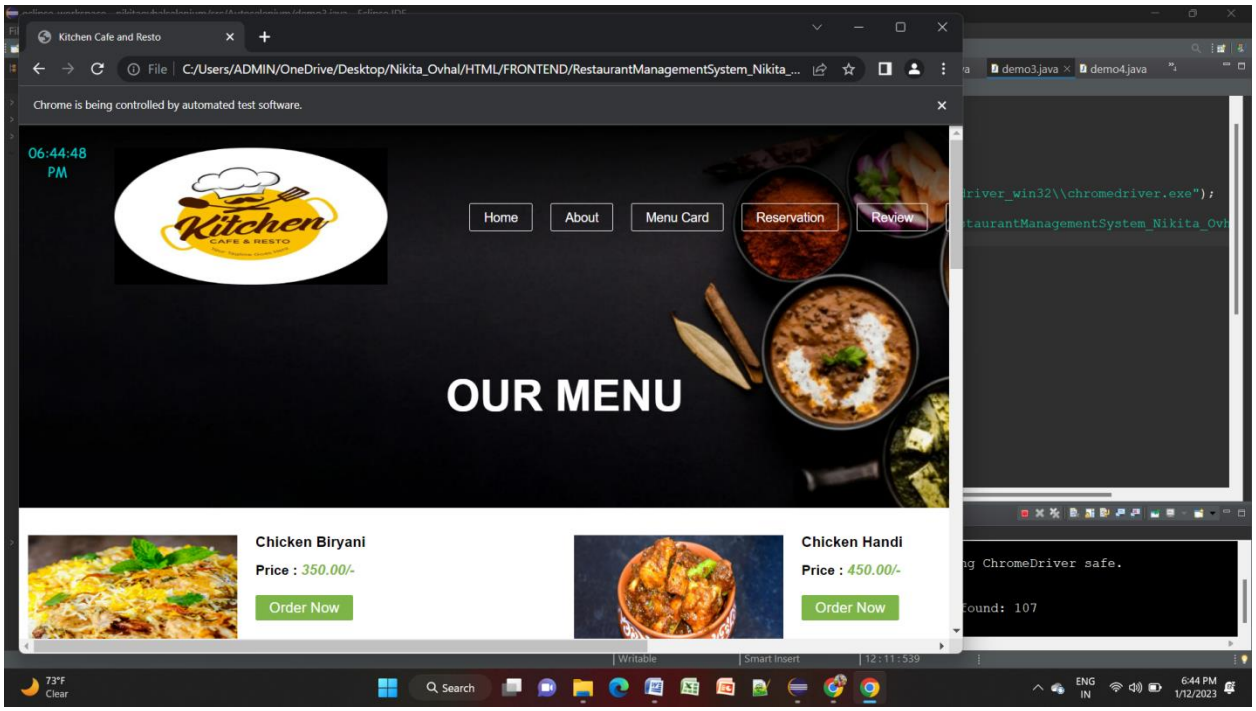
6.5.1 Selenium code for testing Restaurant Management System Webpage

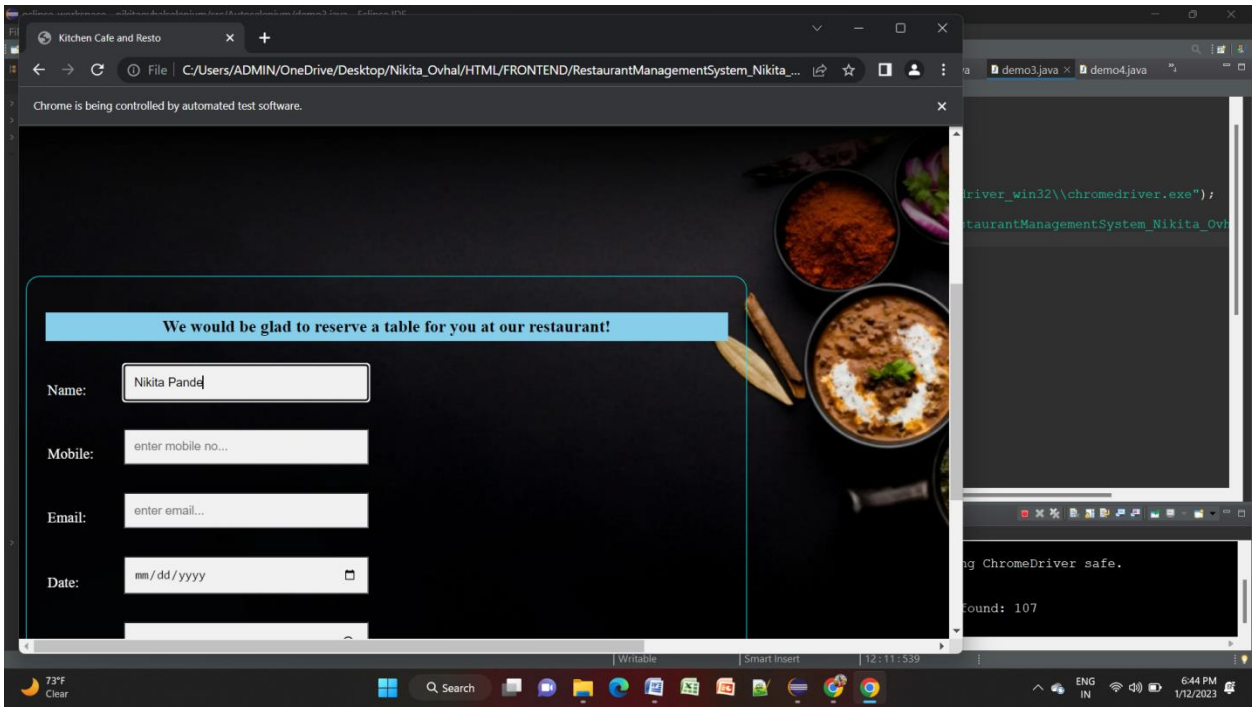
```
package Autoselenium;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class demo3 {

	public static void main(String args[]) throws InterruptedException {

System.setProperty("webdriver.chrome.driver","C:\\Users\\ADMIN\\Downloads\\chromedriver_
win32\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.get("file:///C:/Users/ADMIN/OneDrive/Desktop/Nikita_Ovhal/HTML/FRONTEND/Resta
urantManagementSystem_Nikita_Ovhal/Main.html");

driver.manage().window().maximize();

Thread.sleep(3000);

driver.findElement(By.linkText("Menu Card")).click();

Thread.sleep(3000);

driver.navigate().back();

Thread.sleep(3000);

driver.findElement(By.linkText("BOOK A TABLE")).click();

Thread.sleep(1000);

driver.findElement(By.cssSelector("input[name='name']")).sendKeys("Nikita Pande");

Thread.sleep(1000);

driver.findElement(By.xpath("//*[contains(@name,'mobile')]")).sendKeys("9657851165");
```

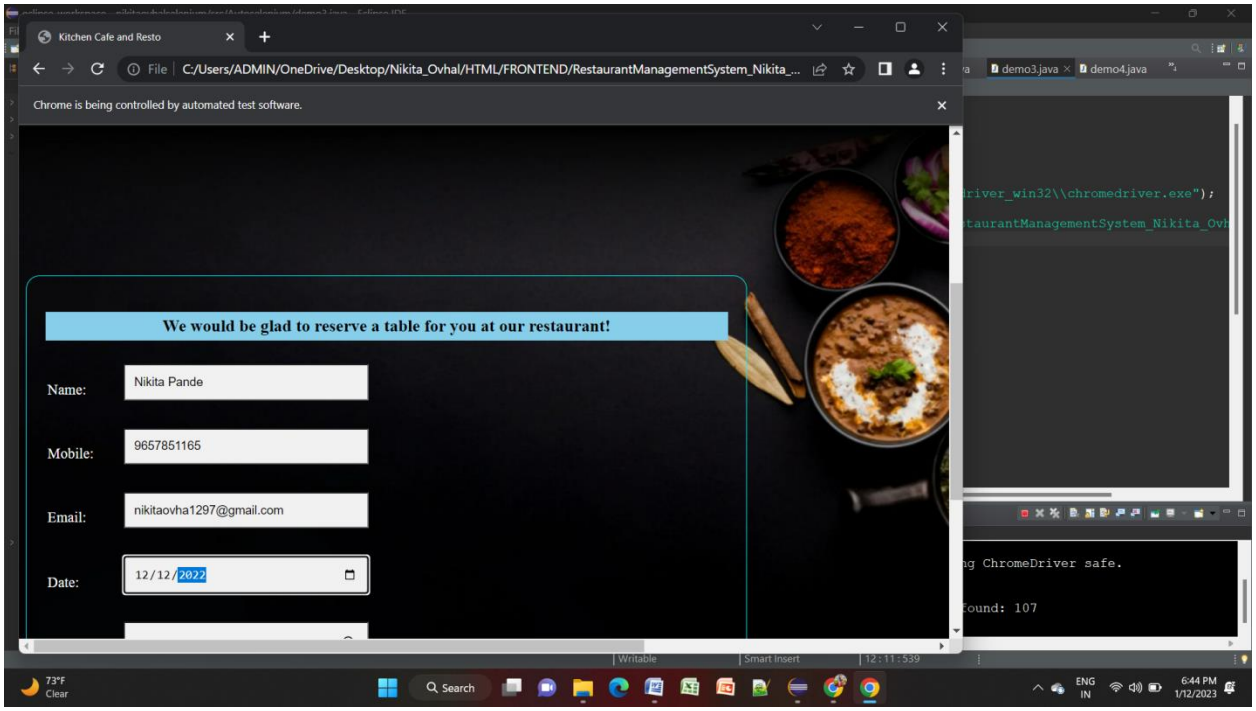
```
Thread.sleep(1000);
driver.findElement(By.name("email")).sendKeys("nikitaovha1297@gmail.com");
Thread.sleep(1000);
driver.findElement(By.name("date")).sendKeys("12/12/2022");
Thread.sleep(1000);
driver.findElement(By.name("time")).sendKeys("12:38PM");
Thread.sleep(2000);
driver.findElement(By.name("message")).sendKeys("Hello I am Nikita");
Thread.sleep(2000);
 driver.findElement(By.xpath("//*[@type ='submit' or @className = 'subbutton']")).click();
Thread.sleep(4000);
driver.navigate().back();
Thread.sleep(2000);
driver.findElement(By.linkText("Home")).click();
driver.quit();
} }
```

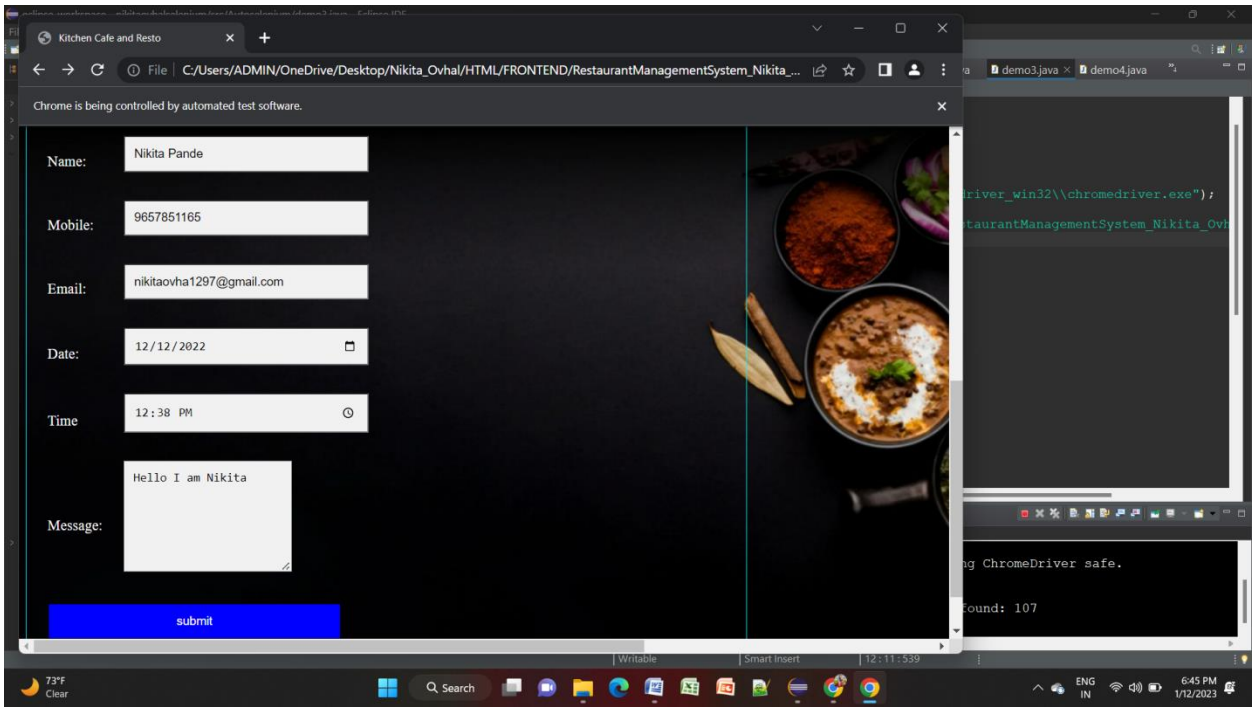
6.5.2 Output for automation of Frontend Design






Kitchen Cafe and Resto
File | C:/Users/ADMIN/OneDrive/Desktop/Nikita_Ovhal/HTML/FRONTEND/RestaurantManagementSystem_Nikita_...
Chrome is being controlled by automated test software.
We would be glad to reserve a table for you at our restaurant!
Name:
Nikita Pande
Mobile:
enter mobile no...
Email:
enter email...
Date:
mm/dd/yyyy
demo3.java
demo4.java
ng ChromeDriver safe.
found: 107
73°F
Clear
Search
ENG
IN
6:44 PM
1/12/2023


Kitchen Cafe and Resto
File | C:/Users/ADMIN/OneDrive/Desktop/Nikita_Ovhal/HTML/FRONTEND/RestaurantManagementSystem_Nikita_...
Chrome is being controlled by automated test software.
We would be glad to reserve a table for you at our restaurant!
Name:
Nikita Pande
Mobile:
9657851165
Email:
nikitaovha1297@gmail.com
Date:
12/12/2022
demo3.java
demo4.java
ng ChromeDriver safe.
found: 107
73°F
Clear
Search
ENG
IN
6:44 PM
1/12/2023


Kitchen Cafe and Resto
File | C:/Users/ADMIN/OneDrive/Desktop/Nikita_Ovhal/HTML/FRONTEND/RestaurantManagementSystem_Nikita_...
Chrome is being controlled by automated test software.
Name:
Nikita Pande
Mobile:
9657851165
Email:
nikitaovha1297@gmail.com
Date:
12/12/2022
Time
12:38 PM
Message:
Hello I am Nikita
submit
demo3.java
demo4.java
driver_win32\\chromedriver.exe");
taurantManagementSystem_Nikita_Ovh
ng ChromeDriver safe.
found: 107
Writable
Smart Insert
12:11:539
73°F
Clear
Search
ENG
IN
6:45 PM
1/12/2023

CHAPTER 7

CONCLUSION

'Kitchen Cafe and Resto' restaurant is become more popular than previous times. They have a plan to start delivery services also. Therefore, it has been implemented as a web based system to meet the requirements of increasing demand .This system helps to manage entire restaurant easily. Restaurant operations also can be well mange using this solution.