8-2023

CONTACTLESS FOOD ORDERING SYSTEM

Rishivar Kumar Goli

CONTACTLESS FOOD ORDERING SYSTEM

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

---

by

Rishivar Kumar Goli

August 2023

CONTACTLESS FOOD ORDERING SYSTEM

A Project

Presented to the

Faculty of

California State University,

San Bernardino

---

by

Rishivar Kumar Goli

August 2023

Approved by:

Dr. Yan Zhang, Advisor, School of Computer Science

Dr. Jennifer Jin, Committee Member

Dr. Bilal Khan, Committee Member

ABSTRACT

Contactless food ordering has revolutionized the way a customer interacts with restaurants by allowing them to place orders and make transactions. Through these web-based platforms, customers can now browse menus, customize orders, and make payments seamlessly. By scanning the restaurant's QR code, customers can reserve a table. If the table is available, then automatically it will be reserved. However, if the table is occupied the customer will be added to the waiting list. Once the customer selects desired food then they can securely make payments based on ordered food items. The food will be delivered straight to the customer's table. Contactless food ordering has truly revolutionized and made the dining experience more convenient.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Dr. Yan Zhang, my project advisor, and committee members Dr. Jennifer Jin and Dr. Bilal Khan for believing in me and accepting to be part of my committee by providing their continuous support. Their encouragement and guidance pushed me to work harder and explore new technologies.

I am very grateful for the trust they have placed in me while working on this project. I appreciate my university for offering courses that allowed me to expand my knowledge and skills in various areas of technology.

Finally, I would like to thank my family and friends for their appreciation and guidance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

Background

The advent of contactless food ordering has revolutionized the way customers to have an interaction with restaurants. With the help of web-based platforms and QR codes, customers can easily browse menus, customize orders, and securely make payments using their smartphones or devices. Digital platforms provide customers user-friendly interface and allow them to explore restaurants.

By using contactless food ordering systems customers can no longer need to wait for physical menus. The process begins by leveraging the power of QR codes which is playing a major role in the restaurant industry. This also provides a very secure process for payments. With the help of a secure streamlined payment process, customer can easily make their transactions. Restaurants can provide customers safe dining experience by setting a new standard for future dining.

Overall contactless food ordering provides a safe and efficient way for restaurants to provide service to their customers. This technology enhances the dining experience and improves the operational efficiency of restaurants.

Purpose

Digital technologies have brought automation to many industries including the food industry. The advancement of wireless technology not only provided the end users convenience of using these systems. The use of mobile technology has become a part of handling the features and opened a way for exchanging information. The main purpose of this project is to order food using a QR code more safely and responsively.

Customers can scan a QR code and can easily access the menu and order food without any physical contact. Generating a QR code is a hassle-free process for restaurants and can keep track of orders and payments from customers. Maintaining physical records can be more tedious, the data can be accessed and modified easily. Food ordering technology has huge competition in the restaurant industry.

With our user-friendly interface, customers can explore restaurants, order and customize food, make payments, and enjoy a safe dining experience. Both customers and restaurants can prioritize safety and efficiency.

Motivation

In recent years contactless food ordering has gained significant popularity and been adopted widely due to several key motivations. Minimizes the physical contact between customers and restaurant staff and surfaces by reducing the risk of transmitting diseases by providing a safe and hygienic dining environment. It makes it more convenient and efficient for both customers and restaurant staff.

Customers can easily browse the menus customize orders and make payments securely. By using this system, the orders can be tracked and maintained records which also helps in reducing the errors. Customer satisfaction plays a major role, and it can be better when customization of orders is provided. It improves the experience and offers a personalized and interactive experience between restaurants and the customer. Restaurants can gather the data and consider the reviews or customer feedback for executing better marketing strategies which improves operational efficiency.

CHAPTER TWO

LITERATURE REVIEW

Related Existing Research

This paper [1] mainly focuses on customer feedback, offers, and customization options to the customers. Lack of preferences, customization options, and real-time feedback, are explained and described in this wireless food ordering system. To overcome these challenges faced by online food ordering systems several new methods are implemented where customers can order food remotely from their devices. A user-friendly interface allows the customer to place orders seamlessly. By using a user-friendly interface users can experience an interactive way with the system to place the order. Once the order is placed the system at the restaurant processes it and the restaurant starts preparing the food. After completing the preparation of food, it is sent to the table. Using such types of systems improves accuracy in orders, reduces errors, minimizes waiting time, and improves service quality. The restaurants can provide customers a very effective and efficient services. With the help of real-time feedback customers can give ratings and write reviews. The feedback and ratings provided by the customers the restaurants can improve their marketing strategies which can help in the growth of their business. In summary, the author's aim is to introduce a customizable wireless food-ordering system using wireless communication technology.

The paper [9] has focused mainly on the Smart Menu system so that customers can customize their orders using digital or personal devices to enhance a better dining experience. In order to develop and build an intelligent restaurant smart system with the latest technologies. Customers get attracted to these digital systems by providing them with interactive and customized menu options. With the help of the customer's previous order history recommendations can be made so that it makes the process easier for customers to select desired food. These digital displays provide information about various food items in the restaurant which helps customers to make quick decisions while selecting their desired food items. Using a smart menu can offer recommendations to customers based on their previous orders and make the food ordering process easier. The author's aim is to provide customers with a better and easier way to customize the menu selection process. The author explains the need for and importance of creating a user interface with customized preferences in these digital displays. Food items are added with images of the dish in these digital displays so that they attract users before placing an order. A preparation process can also be added to these smart digital menus so that customer can read the process of how it is made before he orders. The system also allows the customers to provide ratings and feedback on the restaurant website. These feedbacks which are provided by the customers can be utilized by the backend system to improve the overall menu selection process. In summary, the author explains the significance and importance of using the Smart Menu digital displays

for an interactive and user-friendly interface to attract customers by providing them with all the latest inbuilt functionalities in the system.

The paper [10] clearly explains the development and implementation of an electronic restaurant system that uses a wireless local area network and radio frequency identification technology. The author proposed to address the traditional methods in the restaurant system like manual food order taking, delay in time, and lack of customization in the menu. Using the RFID technology helps in tracking the table location, and availability of the table allocating the tables by providing a unique identification number for customers while reserving a table. Using this process helps in minimizing the waiting time and the manual workload is reduced. The author explains the significant importance of using this technology to improve accuracy, reduce errors, and improve efficiency. This electronic restaurant system retrieves preferences and personalization options from the customer. The feedback and previous order data are recorded to enhance and provide the customer with a better dining experience. The online payment methods and gateways help with transparency and making secure payments. Customers can also make a secure payment by using RFID technology-enabled payments in order to eliminate the need for cash transactions. The author emphasizes that by using this technology restaurants can automate the table reservation process in a very easy way.

The paper [11] mainly focuses on and explains the Multi-Touchable Electronic Restaurant System where the advanced touchscreen technology is

used to improve the efficiency and effectiveness of restaurants. These touch screens are placed at different locations at different places. By using these touch screens which are placed at the tables customers can easily access them and order their desired food. The manual process of taking orders by restaurant staff and the manual process of making payments can be eliminated. The author addressed these issues and explained about an electronic restaurant management system. This system is built around the concept of multi-touch screens by allowing the customers to interact with the digital display user interface. Customers can search for their specific items mentioned by the categories mentioned in the menu. The author tries to explain the use of multi-touch technology and its advantages to customers and restaurants. Customers can directly order food from these multi-touchable screens which reduces the manual workload and waiting time. The author proposed this paper to address the issues that are in use of the traditional restaurant management system. Customers can browse through different menu categories and can customize their orders before placing their orders. Once the order is placed the restaurant kitchen staff starts preparing the food according to the customer's preferences. Restaurant staff can keep track of orders and provide effective services to customers. A restaurant owner or the staff can access these displays and they can track the records and data of customers which helps to improve operational efficiency.

The paper [6] focuses on the web-based application for the food ordering process. The author's main aim is to provide a user-friendly interface platform for customers to interact and order food. It is a web-based application where customers can order their food online based on their preferences such as adding additional ingredients, increasing or decreasing spice levels, adding additional toppings, etc. Based on previous orders a personalized menu is also suggested to the customers which also helps in user satisfaction and provides customers an effective dining experience. The author also explains user satisfaction while experiencing a proper user interface. The significance and importance of adding additional features such as secure payments, status updates, tracking orders and transactions. The preparation and delivery process details can be tracked by the customer using the application on his device. Order notifications can be sent to customers to update them on estimated delivery time or any changes to an order. The system allows customers to browse menus order food and make payments securely and track the progress of their orders. The author emphasizes the importance of user-friendly interface customization options and feedback mechanisms in providing a safe and satisfactory online food ordering experience.

An automated food purchasing aiding system was developed by the efforts of [6] by integrating the famous platform along with wireless technology just to cut down the burden of conventionally operating restaurants.

With the deployment of QR codes [7] a food purchasing aiding system by targeting the cafeteria and all other processes involved in food purchase. A

cinema complex-oriented food purchasing aiding system was developed by the efforts of [2] by empowering the users to order via the internet and by scanning QR codes. Every detail of a food servicing establishment was posed by the system developed by [8] via a QR code. Despite being able to reduce the burden of servers in a food servicing establishment this system was holding good to all establishments.

Existing Food Ordering Systems and its Challenges

Existing food ordering system websites allow customers to browse menus, customize orders, and make payments electronically. QR codes are often used to enable touchless interactions at restaurants. While contactless food ordering offers numerous advantages and disadvantages

1) Technological Barriers: Some people who might have limited access to technology may face challenges in using mobile applications or websites to place orders for food.

2) Limited Sensory Experience: The inability to inspect the food can also be a drawback because restaurants providing images and descriptions of the dishes may not fully capture the actual presentation or portion size. This can also lead to customer dissatisfaction if the food does not meet the customer's expectations upon delivery or pickup.

3) Connectivity Issues: Internet connectivity can lead to potential disruptions or delays in placing orders, particularly in areas with weak Internet signals.

4) Lack of personal Interactions: Contactless food ordering can eliminate face-to-face interactions with restaurant staff and some customers might prefer special requests. This can make customers feel more structured and may limit the flexibility for last-minute changes or additions to the order.

5) Potential errors or glitches: Technical issues or payment mode errors can occur so these can be resolved by customer support.

6) Security and Privacy Concerns: Customers may have concerns about data breaches, by sharing their personal information, or misuse of information.

CHAPTER THREE

SYSTEM REQUIREMENTS

Tools and Technologies

Django Framework

Django is a web framework that is used to build web applications using Python language. Django helps us to develop a web application easily by providing all the built-in tools and libraries. In this project, we have used the stable version of Django. This application is built using a framework with a model view template architectural pattern which is responsible for generating HTML output. By using this framework, we can perform various actions and view the URLs by mapping them to the correct objects. Here, we are using ORM which is an inbuilt database in this framework.

ORM stands for object-relational mapping. This is an inbuilt database that is provided in the Django framework. By using this in our application we can easily communicate with the database directly. ORM mainly helps in mapping the data models to the database tables and the database operations can be managed and performed easily. There is no need to use separate SQL queries the ORM interacts between Python language and the database. This enables developers to interact with the database by using Python instead of SQL queries. ORM helps in reducing the code and web applications can be built in a very efficient way.

MySQL Database

In this project, we used MYSQL as a database. This is a structured query language where we can perform actions while storing the data. The data in the database can be manipulated and managed. The data in this project will be stored and automated in the form of a table format. The example table, which is shown below represents a user with all the characteristics needed for that user. It is represented with entities like ID, Name, Age, and Status.

Table 1. Example Database Table

ID	Name	Age	Status
1	John	10	Reserved
2	Sarah	12	Available
3	Rishi	14	Waiting List

SQLyog

In this application, we have used SQLyog to visualize the database in this system. It is an open-source application where users can download it for free. The data can be visualized with a user interface provided in it. By using this tool, we can perform various actions while visualizing the data in the database. The data can be managed easily by creating, updating, and manipulating the data in the database. It is easier to use and can run by using simple queries. Through different file formats, the data can be visualized and managed. An example data table for SQLyog is represented in the below figure for restaurant admin details. The table is represented with ID, First Name, Last Name, Password, and Email.

Table 2. Representation of data in SQLyog

ID	First Name	Last Name	Password	Email
1	John	Mark	*********	John@gmail.com
2	Mike	Mathew	*********	Mike@gmail.com
3	Tony	Tony	*********	Tony@gmail.com

Python

Python is a popular and high-level programming language for its readability. It is easy to read syntax which focuses more on readability. Python is available for various environments, and it is known for its cross-platform compatibility. It has gained huge progress in fields such as web development, data science, and machine learning.

XAMPP

XAMPP is an open-source web server solution package. Developers can easily set up a local web server environment on their computers for development and testing purposes. It is a software distribution that provides and allows you to run Apache, My SQL, PHP, and Perl together without the need for separate installations and configurations. It simplifies the process of using a local web environment for developers and is available for various operating systems.

Visual Studio Code

Visual Studio Code is an open-source and integrated development environment. It is used for code development like executing the code and all types of extensions are inbuilt and can be used and accessed. It has gained

huge popularity among developers for its performance, extensibility, and user-friendly interface. It also provides a and supports wide range of programming languages, frameworks, and tools.

Software Requirements

Operating System:	Windows 7, 8 or 10
Server-side Script:	HTML, MYSQL, CSS, and Bootstrap
Programming Language:	Python
IDE:	PyCharm, Visual Studio Code.
Framework:	Django
Tools:	XAMPP, ORM, MySQL

Hardware Requirements

RAM:	4 GB
Hard Disk:	160 GB
Processor:	Intel i3 configuration
Keyboard:	Standard Keyboard
Monitor:	SVGA

CHAPTER FOUR

DESIGN

The design of this proposed contactless food ordering system involves websites to order food. Customers can access the menu to order food. They can customize orders and make secure payments. The generated QR code should be scanned to view and access the food menu.

System Architecture

The architecture in this proposed contactless food ordering system supports both customer and restaurant admin. This architecture is built with many features and functionalities that help the restaurant admin and staff to quickly access the orders. This system is developed in a user-friendly way so that customers can have a better experience in using this interface. Customers can make reservations online or on the phone to reserve a table in advance. The admin should sign up and then log in to his account to view customer orders and payments. Customers can sign up and sign into their account, they can view restaurants by scanning the QR code, entering the member count and ordering food, making payments, and logging out. If the restaurant is full, he will be added to the waiting list for a time frame until he gets a reservation.

The system architecture based on which our QR code food ordering system has been built supports both admins as well as the customer. The below figure represents the system architecture.

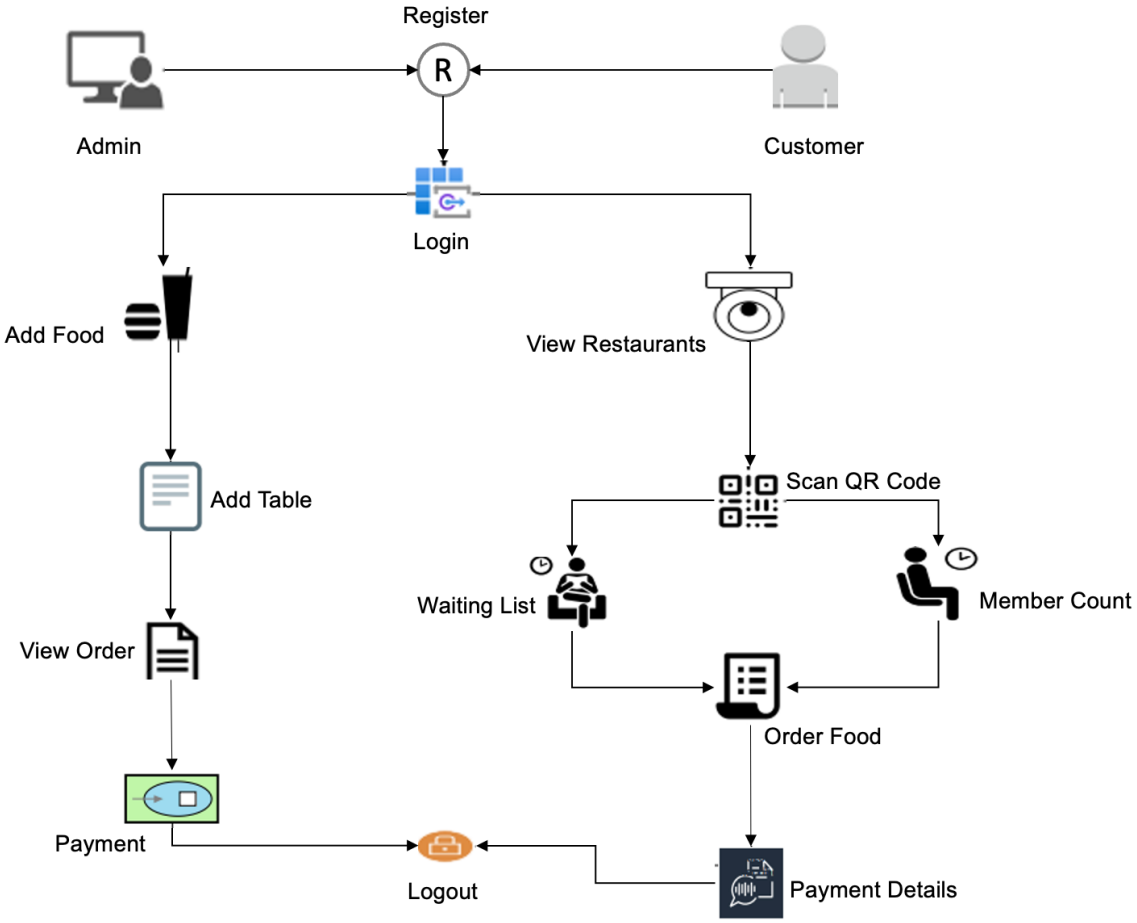

Figure 1. System Architecture

System Design

This is a user-friendly application where customers scan the QR code and access the system. They can browse the menu, customize orders, and make payments securely. Admin and Customer are two entities in our QR code food ordering system. Figure 2 explains the system design of the contactless food ordering system.

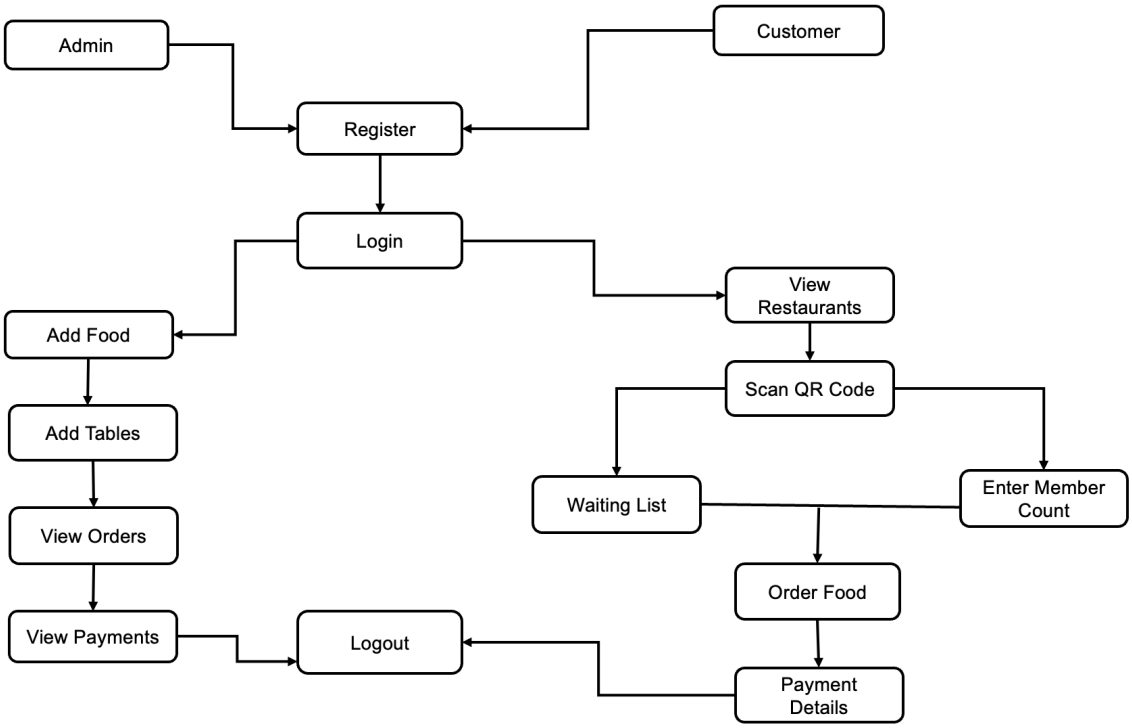

Figure 2. System Design

Admin can access and manage the menu by updating food items, viewing orders, managing prices, configuring other system settings, and managing table reservation availabilities. It also processes the updates on the waiting list in real-time. Secure payment gateways are integrated to handle customer payments seamlessly.

UML Representation

UML stands for Unified Modelling Language. It provides a set of notations and guidelines for creating visual representations of software systems. visualize various aspects of a system aiding in analysis, design, and implementation.

Use Case Representation

It is a form of behavior representation and shows a pictorial schema of capability given by a system. The below figure shows the use case representation between the restaurant and customer.

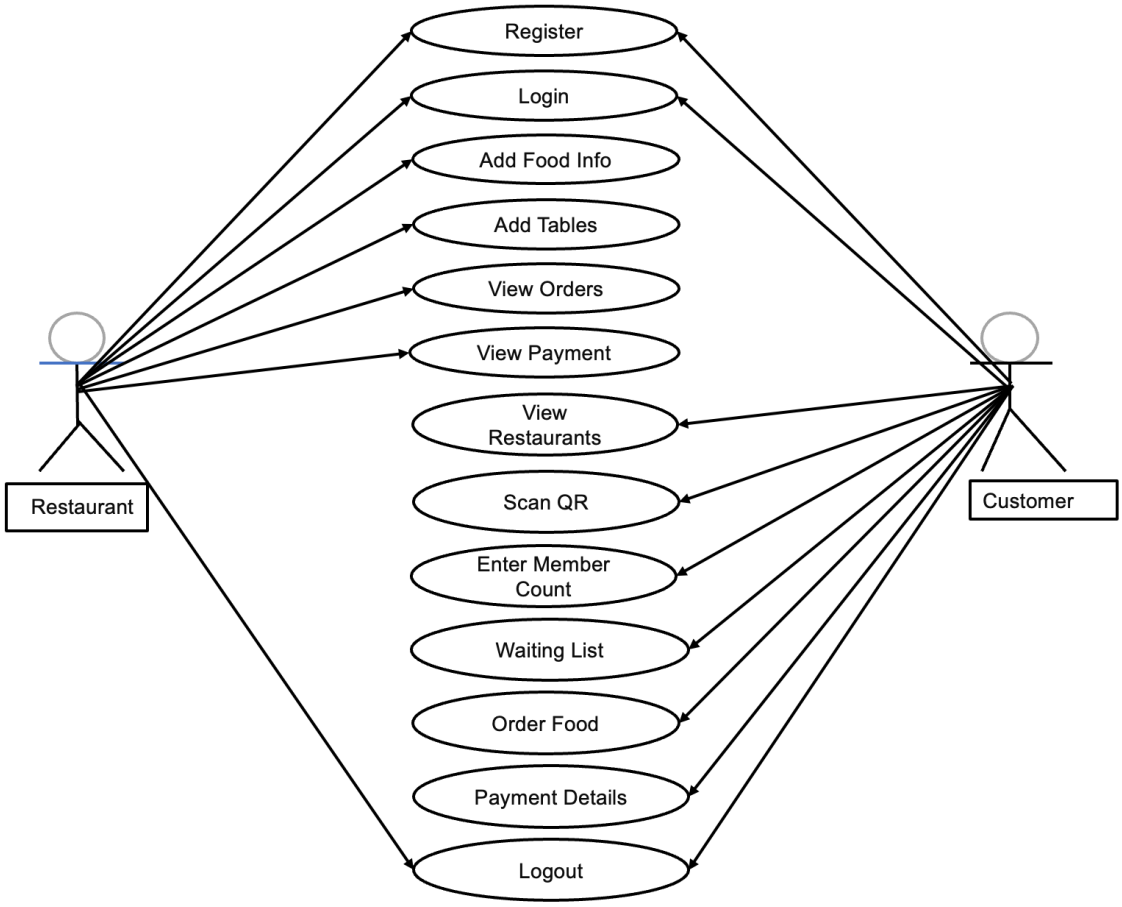

Figure 3. Use Case Representation

Class Representation

A class representation is a kind of static arrangement representation that reports and depicts the structure of a system by presenting classes, attributes,

processes, and associations between the classes in the system.

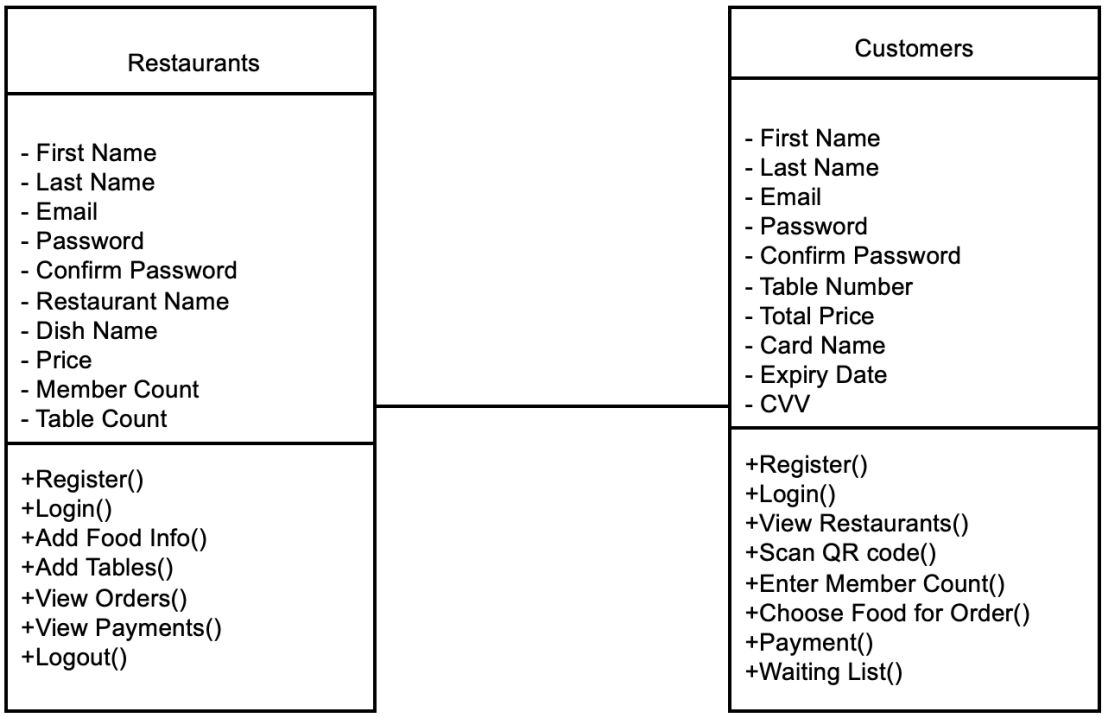

Figure 4. Class Representation

Sequence Representation

Sequence representation is a type of communication representation presenting how operations process and operate each other. It is made up of a message sequence chart. This kind of representation also has different names such as event diagrams, timing diagrams, and event scenarios.

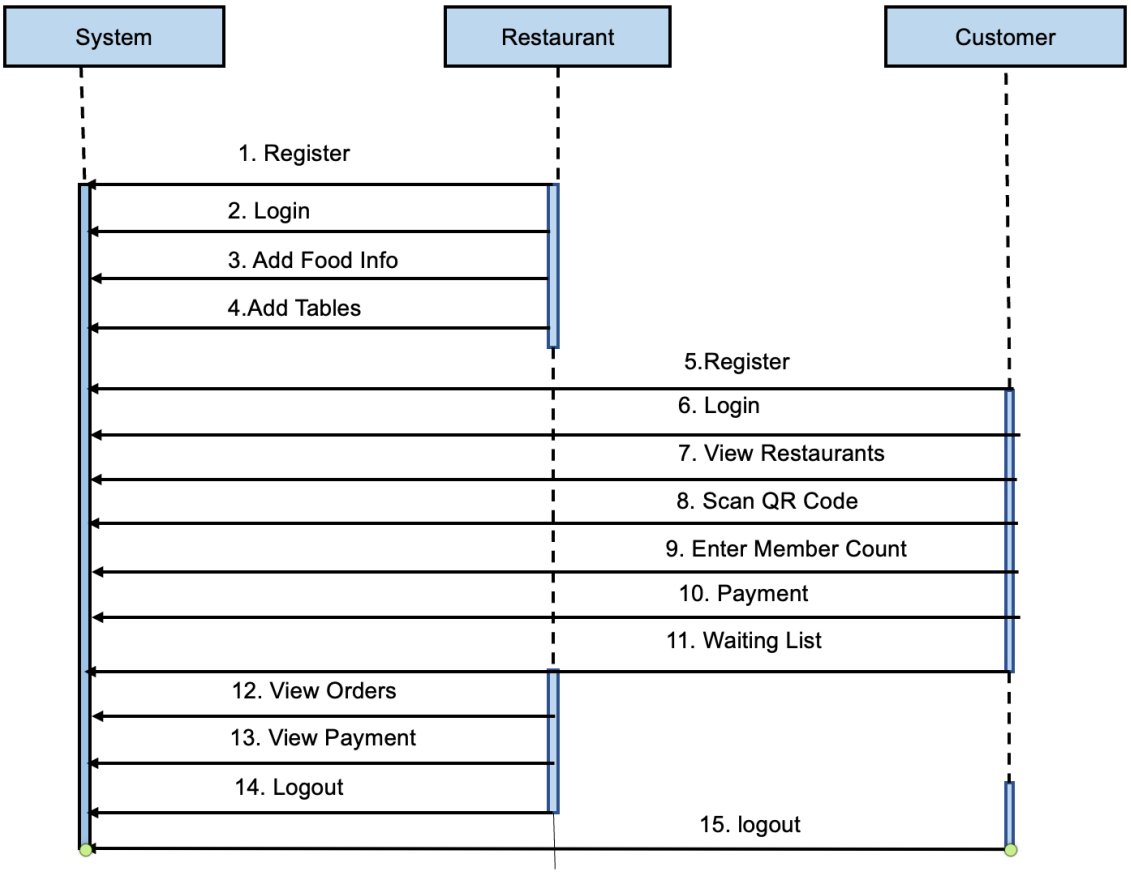

Figure 5. Sequence Representation

Activity Representation

An activity diagram provides a visual representation of functionalities and control flow within a system. This makes it easier to understand and communicate the processing logic. These are used for modeling workflow systems, and business processes, involving multiple steps and decision points. These diagrams effectively represent the commercial and processing functionalities of the elements contained in the system.

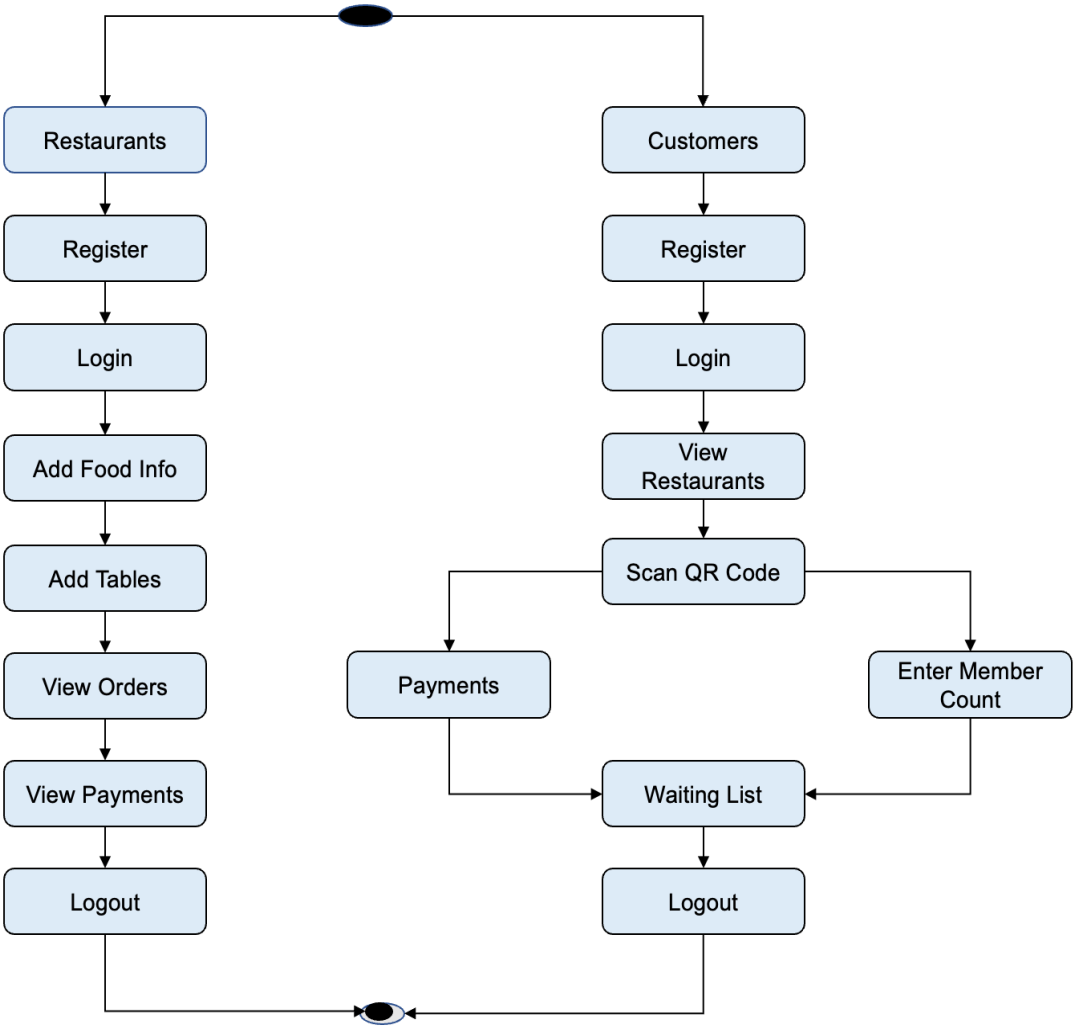

Figure 6. Activity Representation

ER Representation

This is an entity-relationship model which reports the arrangement of a database. The major element of this representation includes the relationship set and entity set. This type of representation represents the entire logical

arrangement of the database through an indication of an association between the attributes.

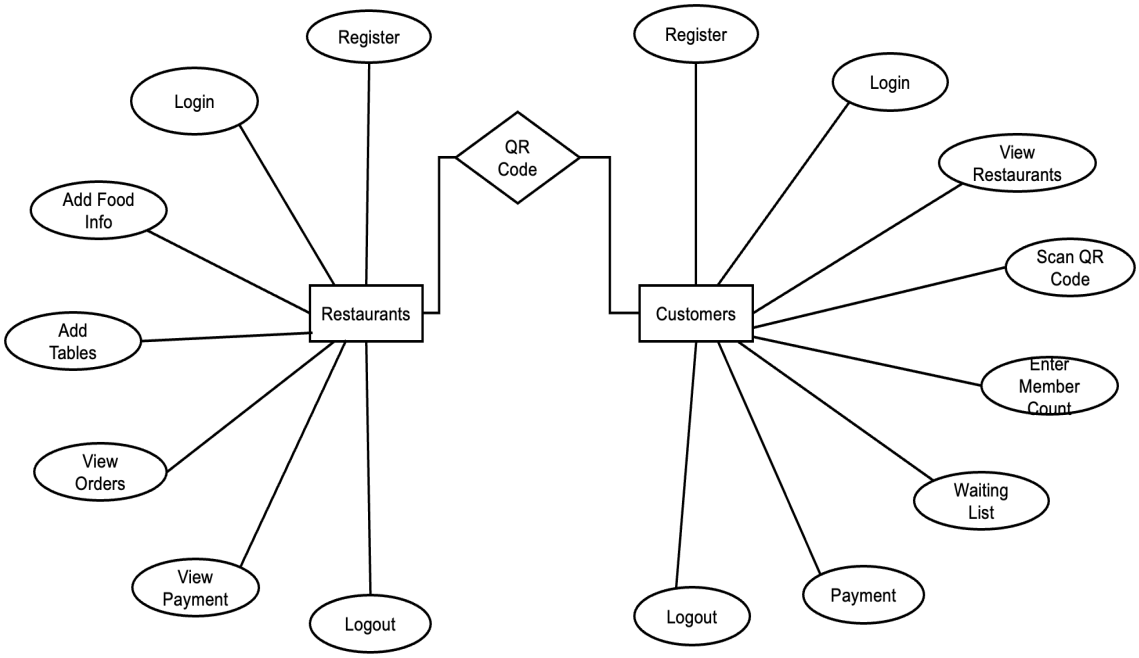

Figure 7. ER Representation

CHAPTER FIVE

IMPLEMENTATION

Setup and Configuration

The implementation and configuration process of this project and a high-level overview are explained in this chapter.

Download and Install the XAMPP control panel which provides Apache webserver and MySQL database. Start the MySQL service in the control panel.

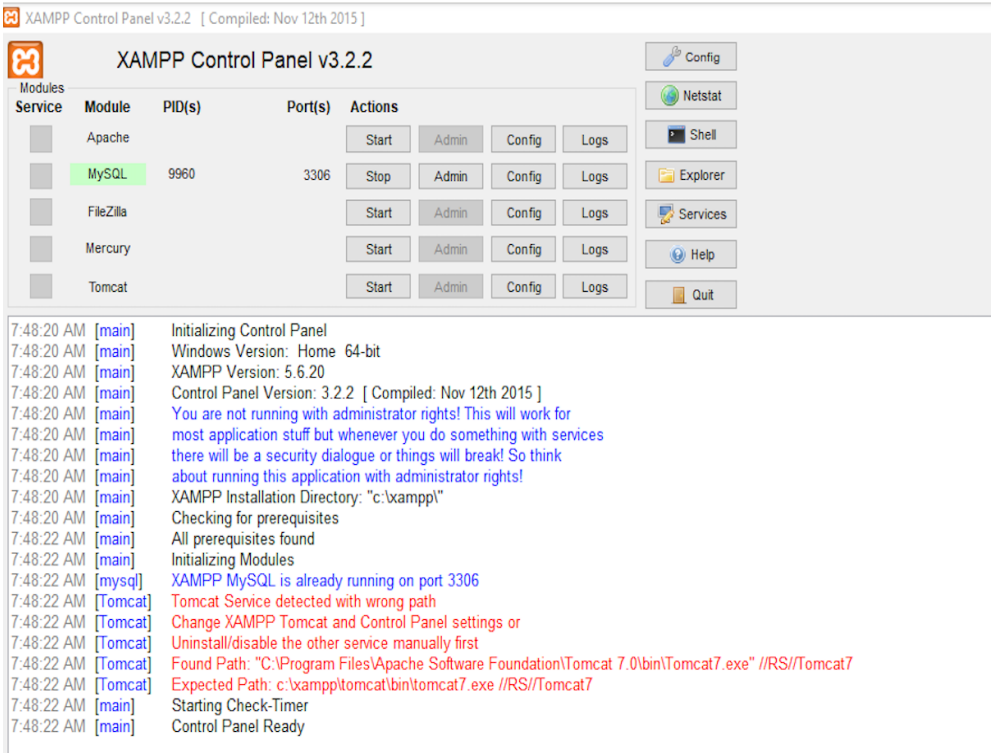

Figure 8. XAMPP Control Panel

Now download and install MySQLyog or any other MySQL client and configure the database connection settings using the same port number in your

Django project. MySQLyog is a powerful graphical user interface where we can create, delete, and modify database tables, indexes, and other database objects.

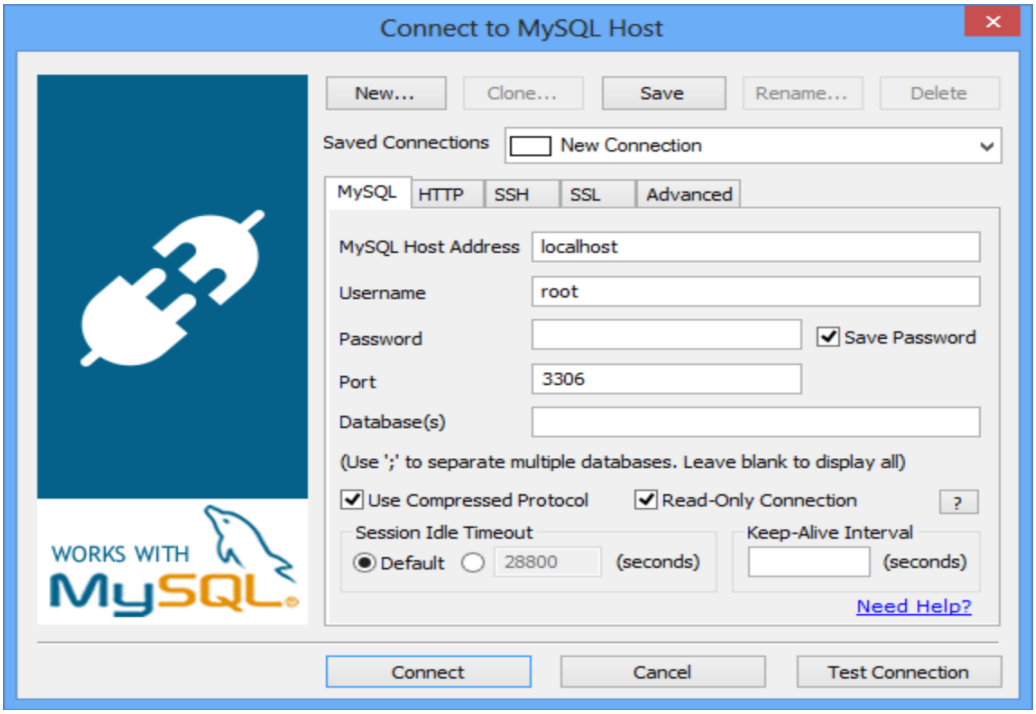

Figure 9. SQLyog Configuration

Install the Django framework using the Python package manager. This project is developed with MVT (Model View Template) architecture using the Django framework.

Commands to install Django use pip install Django. To create a project, use the command django-admin startproject “project name”. To create an application, use the command python manage.py startapp “app name”. In this project, we used the project name as a contactless food ordering system and application

modules as restaurant and customer. A root folder is created with setting.py where the main source files with connections are placed.

Define all the Django models to represent entities like Users, Menus, Orders, Member Count, and Payments, in the app models.py file. In this file define all the models in the models.py file for Customer details, Table booking, User orders, Waiting list, and Payment details.

Perform migrations based on these models to create database tables by using the command python manage.py make migrations. It analyzes the current state of models and generates migration files. This examines the models in our project and will compare them to the database schema. The Python code will be in each migration file that describes the database schema. The Python code includes operations like creating tables, rows, and columns. Run the command python manage.py migrate this will read the migration files and executes the SQL statements to modify the database. These migration files help to keep track of the database by making it easier to maintain and migrate the application.

All the tables are created for our contactless food ordering system for customer details, customer payment details, table booking, user orders, restaurant add info, restaurant qrcode, and restaurant tables.

Below Figure 10 clearly shows us that the tables are created for the project contactless food ordering system.

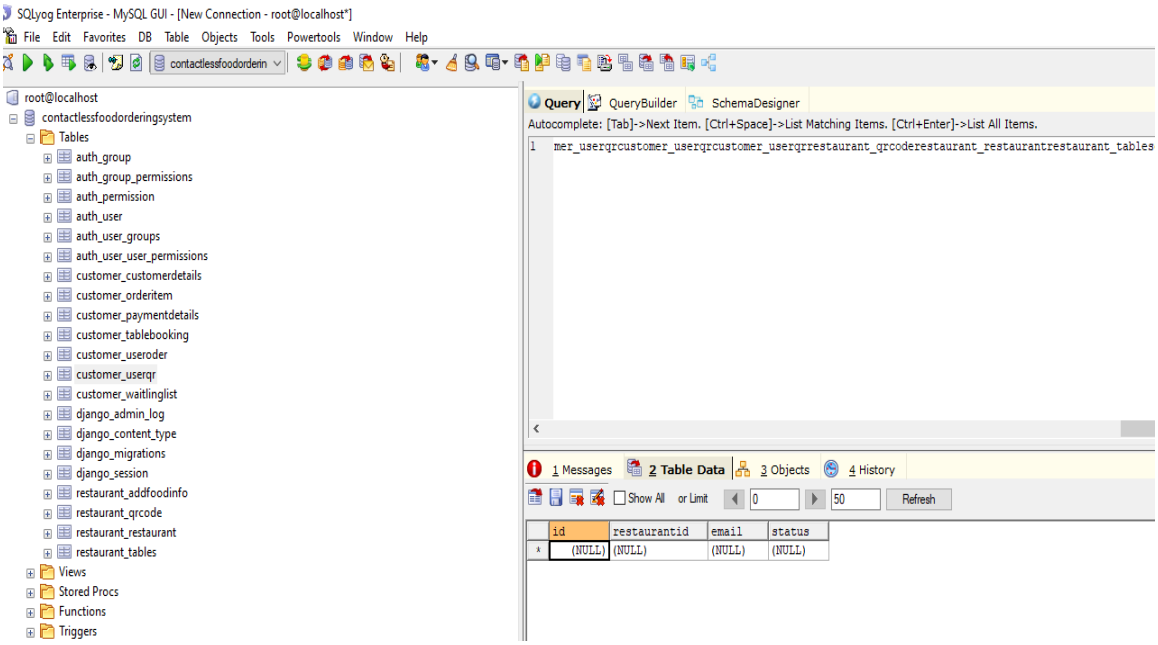

Figure 10. Creation of Database Tables

To handle all the HTTP requests, define all views in apps (Customer and Restaurant) views.py file and implement the logic for user authentication, menu, etc. In these apps create html templates directory to render the user interface. We need to include the path of the customer module and restaurant module in both apps in the url.py file. Integrate a QR code generation library like qrcode to generate QR codes for individual menus for restaurants.

To start use the command python manage.py run server. In our project, it redirects to the local host on the system viewing the homepage of the contactless food ordering system.

Exploring Database

The Django framework has an inbuilt database known as ORM Object Relational Mapping. Select MySQL database which is configured in the project in the settings.py file. The tables can be created based on the models defined in the project. SQLyog is configured with the database it provides an interface to view, modify and visualize the table structure. Using SQlyog we can examine the structure of each table which includes table name, column name, and datatypes.

After performing the migrations by using the command python manage.py make migrations and python manage.py migrate blueprints of the database tables which are defined in models.py are created for admin, customers, restaurants, add tables, food info, order items, waiting list, and payments. The data stored in the database in tables in the form of rows represent specific records or entries and columns represent specific information. Define all the models in the models.py file for Customer Details, Table Booking, User Order, Waiting List, and Payment Details. The below code shows us how to define the models.

```
class Customerdetails(models.Model):
    first = models.CharField(max_length=50)
    last = models.CharField(max_length=50)
    email = models.CharField(max_length=50)
    password = models.CharField(max_length=50)

    def __str__(self):
        return self.username
```

Table 3. Data Table Representation for Customer Details

id	First Name	Last Name	Email	Password
1	mark	mark	mark@gmail.com	12345678
2	mike	mike	mike@gmail.com	34592678
3	tony	tony	tony@gmail.com	67543214

The example data table representation for the waiting list is also shown in the below data table. It represents with id, table number, member count and time for availability.

Table 4. Data Table Representation for Waiting List

id	Table Number	Member Count	Email	Time
1	T2	2	mark@gmail.com	0 min
2	T6	3	mike@gmail.com	20 min
3	T16	2	tony@gmail.com	60 min

In this way, all the database models should be defined in the modules.py file for apps restaurant and customer module to generate database tables. The root folder settings.py contains various settings and configurations for the Django project. database connections and URLs. All the database configurations and installed modules for this project static URL files (CSS, JavaScript files), and template configurations are defined in this root folder.

```
# Database
# https://docs.djangoproject.com/en/4.1/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE' : 'django.db.backends.mysql',
        'NAME' : 'Contactlessfoodorderingsystem',
        'HOST' : 'localhost',
        'USER' : 'root',
        'PASSWORD' : '',
        'PORT': 3306,

    }
}
```

Figure 11. Database Connection

The database connection is set up in the settings.py file in the root folder. In the terminal run the command python manage.py run server. Homepage of contactless food ordering system is viewed on local host http://127.0.0.1:8000.

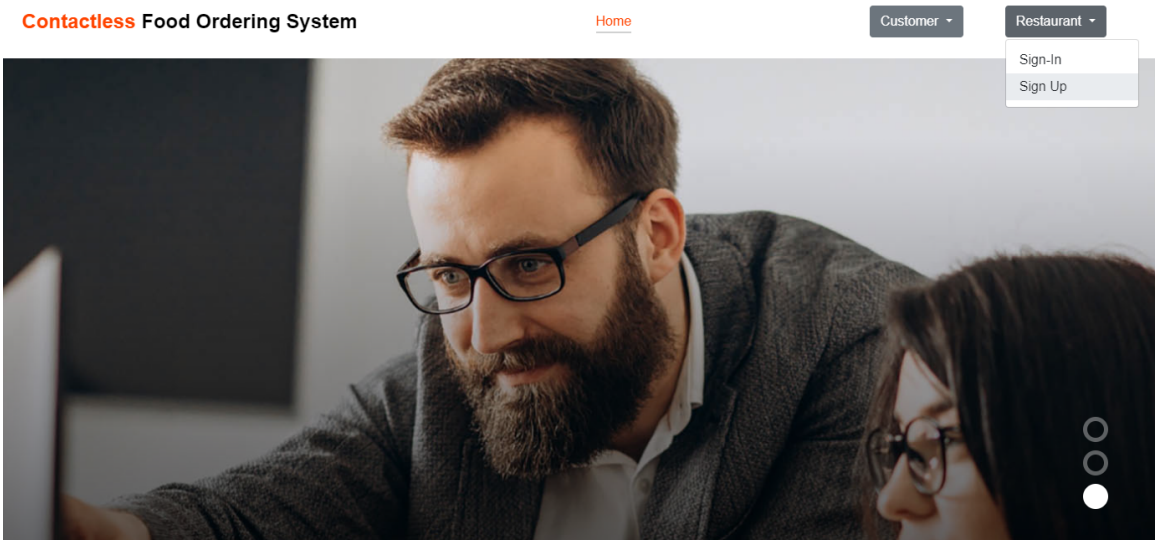

Figure 12. Homepage of Contactless Food Ordering System.

CHAPTER SIX

TESTING AND VALIDATION

The home page of the QR code-based food ordering system is designed with two modules Restaurant (Admin) and Customer module. The restaurant owner or admin must register to sign up by entering his details. This information will be used to create an account.

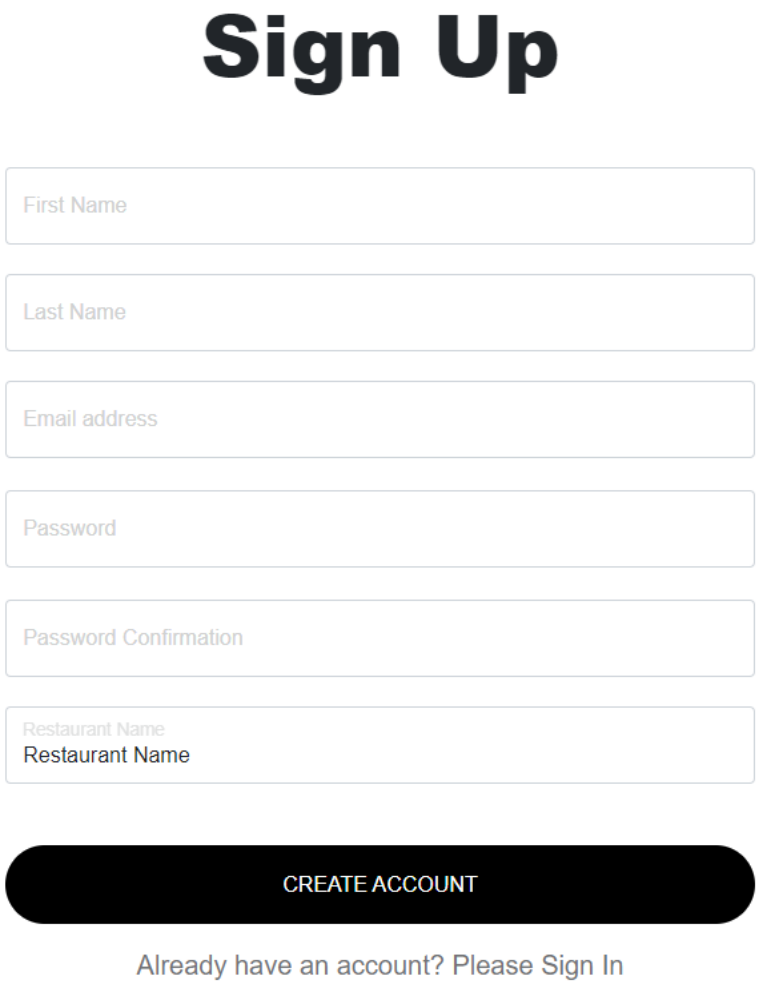

Figure 13. Restaurant Admin Signup

After creating an account, the admin can log in with his credentials.

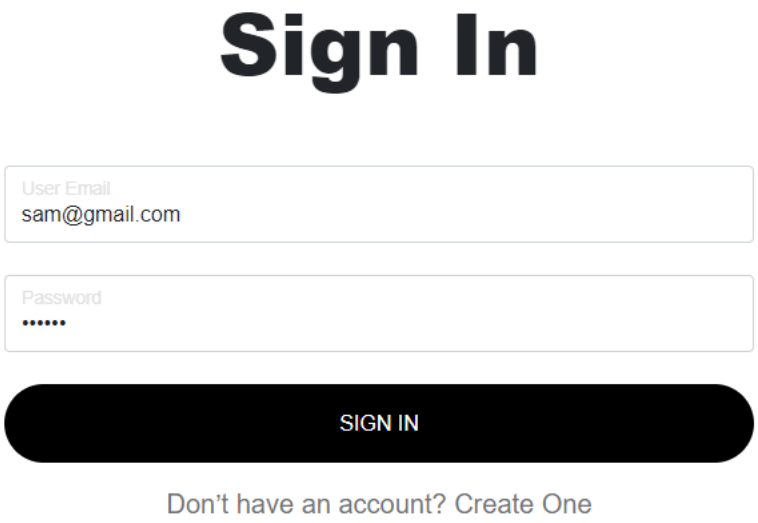

Figure 14. Restaurant Admin Sign-In

After signing into his account admin can access the menu.

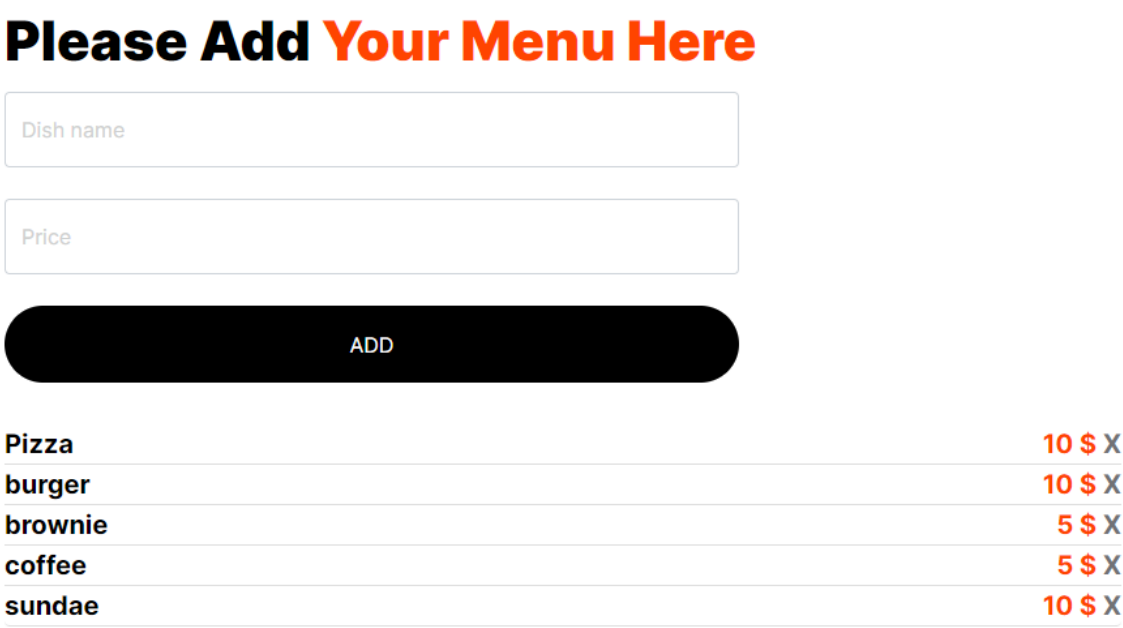

Figure 15. Admin can add menu.

The restaurant admin can add, edit or remove food items from the menu. Admin can add or create tables according to the member count entered by the customer. Admin can enter the food item name and price details. Admin also has access to add and manage tables specifying the table number and the maximum capacity it can accommodate.

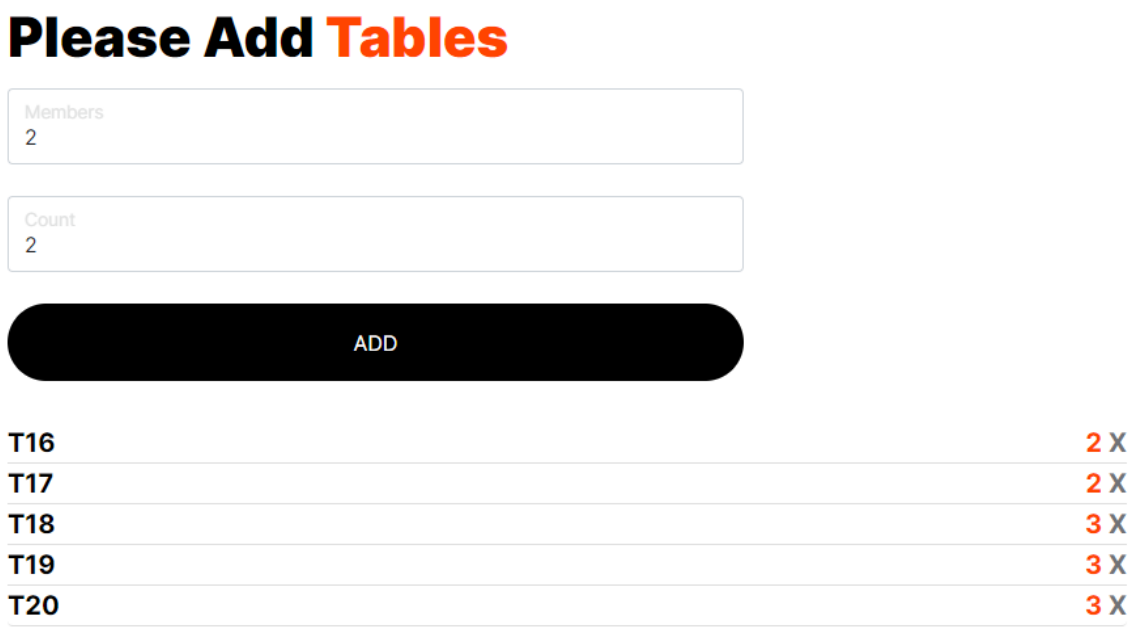

Figure 16. Add Tables

The restaurant admin can log out after entering all the restaurant details in the account.

A customer must register or sign up with his details and create an account. He can sign up by entering his details such as First Name, Last Name, Email, and Password, and create an account.

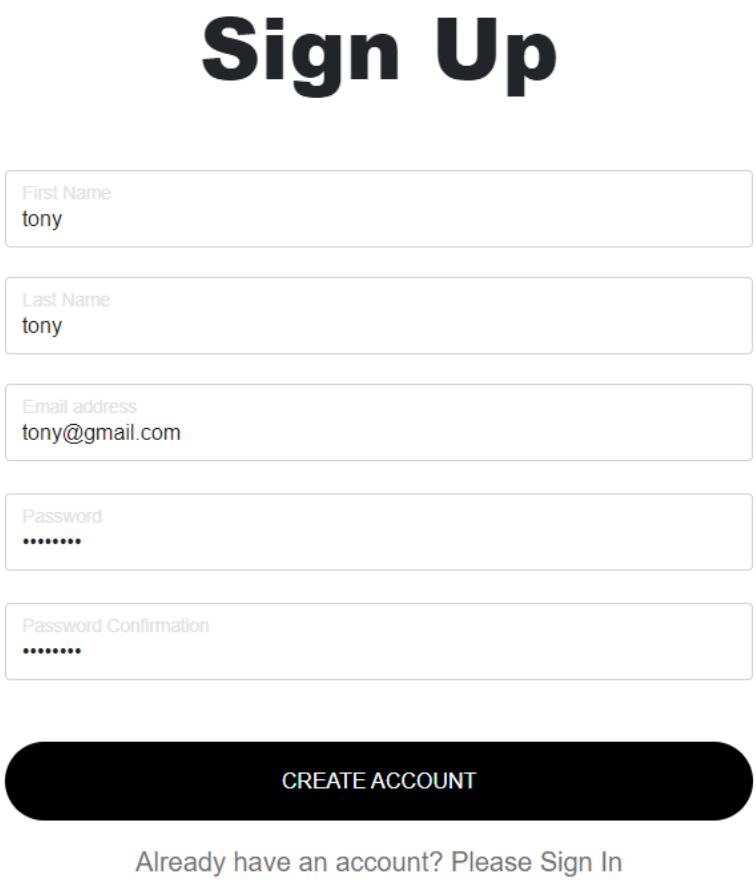

Figure 17. Customer Sign-Up.

If the customer already has an account, he can directly sign in by using his credentials and access his account. Customers can view all the restaurants listed on the home page and can access each restaurant. Customers can select desired food from the menu.

Customers can sign in by entering the details provided during the registration process. After signing in with the credentials it is redirected to the homepage.

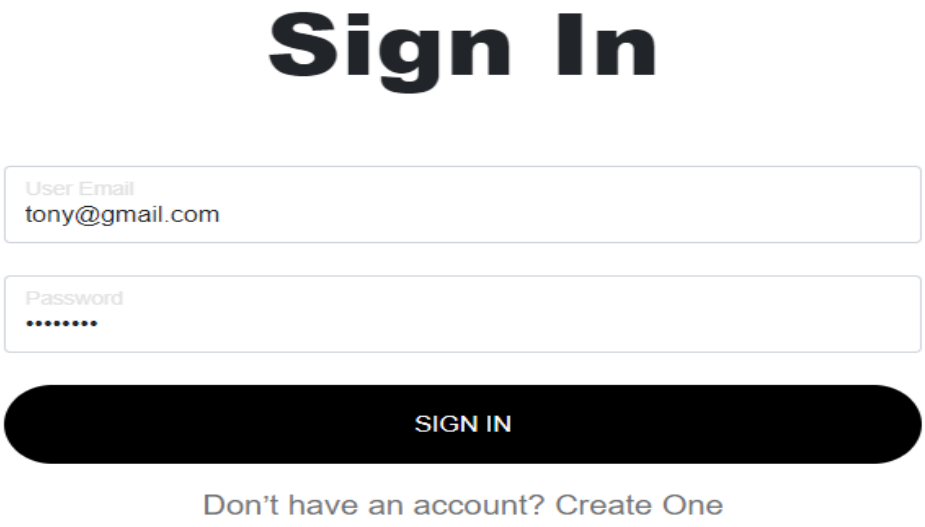

Figure 18. Customer Sign-In

Here, customers can view the restaurants after signing in.

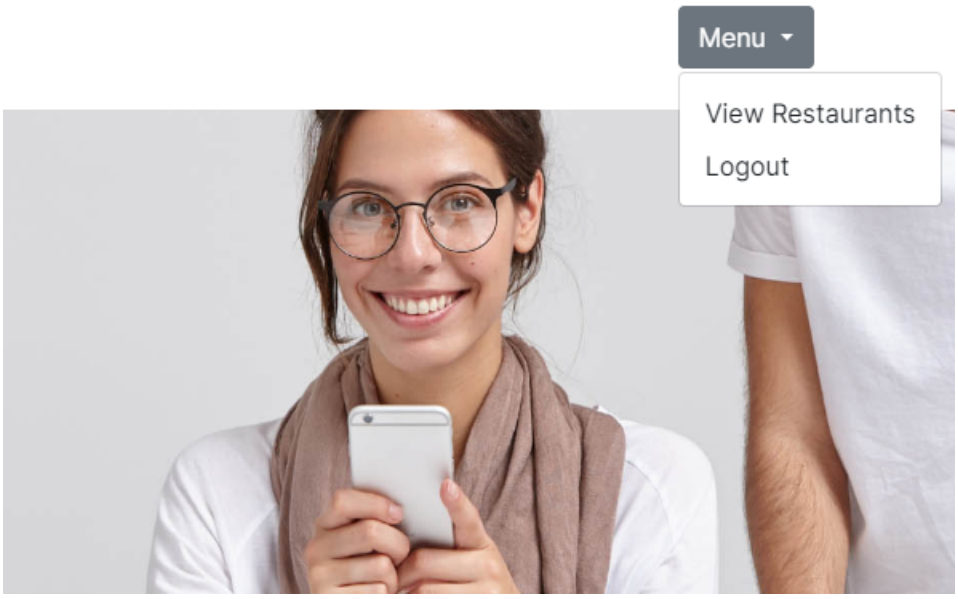

Figure 19. View Restaurants

After clicking on view restaurants all the restaurants which are updated in the system will be viewed. A QR code is generated when a customer selects and clicks on the restaurant.

sams

Figure 20. QR code

Now the customer needs to scan the QR code and enter the number of people so that the system reserves a table for the customer depending upon the count.

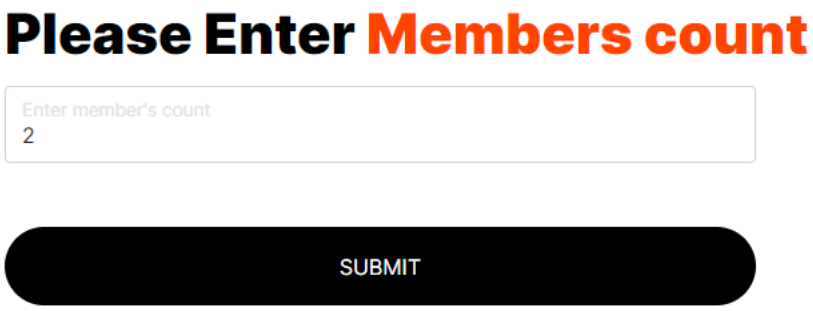

Figure 21. Enter Member Count

After entering the member count, the customer can view and browse the menu. Customers can add or select the desired food items from the menu and further proceed with the order process to place an order.

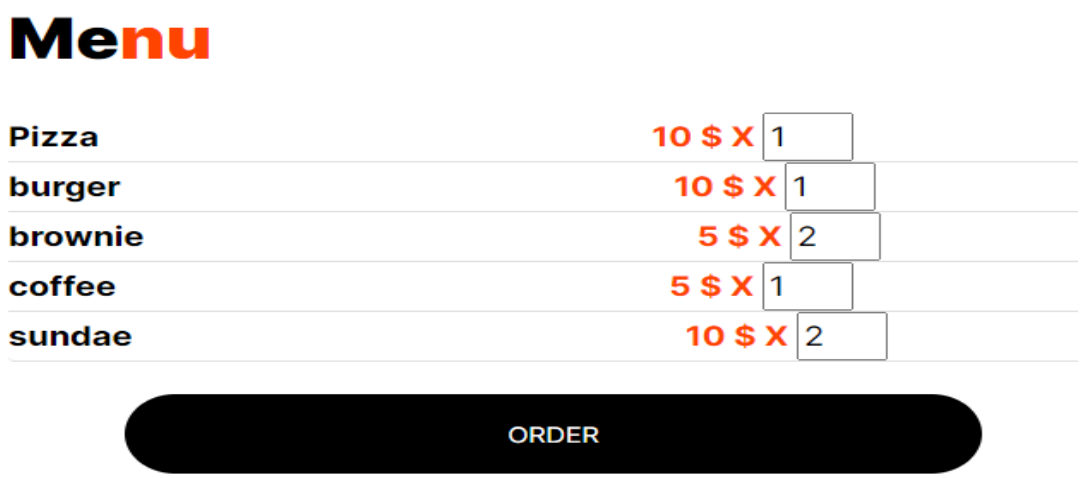

Figure 22. Order Food

Users can view and review their orders after selecting the desired food before finalizing and making payment transactions.

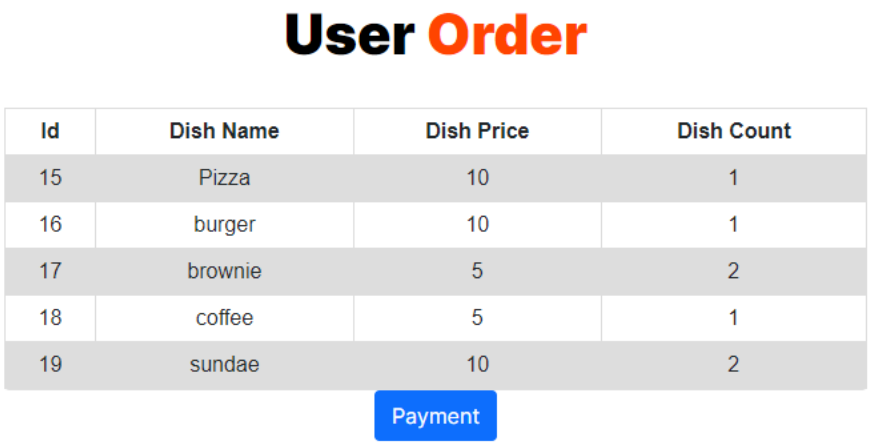

Figure 23. User Order

If the restaurant is busy or there are no available tables for reservation customers will be added to the waiting list. After entering the member count, he will be added to the waiting list with a specified time duration. It will show the wait time estimated to the customer before he gets the availability.

Table Number	Member Count	Time For Availability
T2	2	0 Min
T6	2	0 Min
T16	2	0 Min
T17	2	0 Min
T16	2	60 Min

Click on Menu

Figure 24. Waiting List

The customer is added to a waiting list and now the customer can click on the menu and select desired items to place an order before he gets a table.

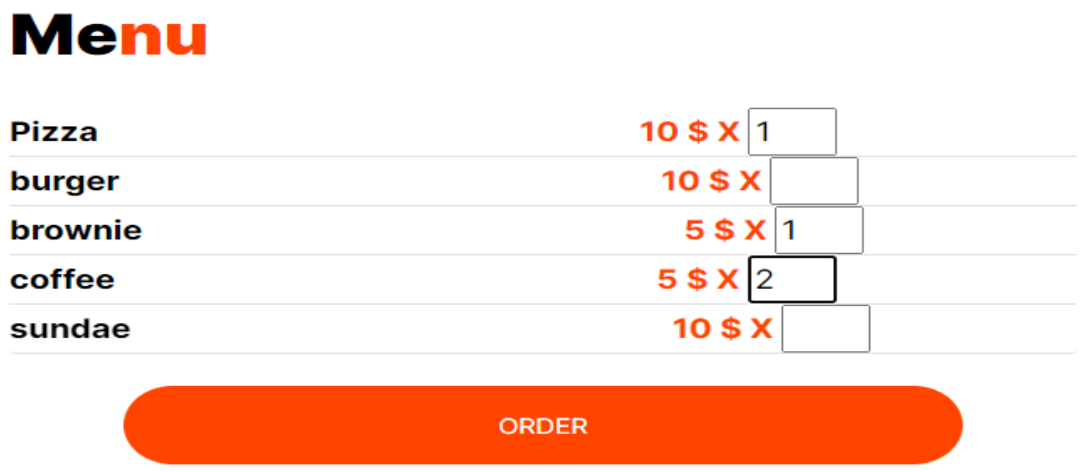

Figure 25. Pre-order food

After placing an order customer can view and review the order and proceed with the payment in user order, Customers can view the name, price, and count the items that he has ordered.

Figure 26. Review Order

By clicking on payment, it will redirect to the payment page where the customer can enter all his card details.

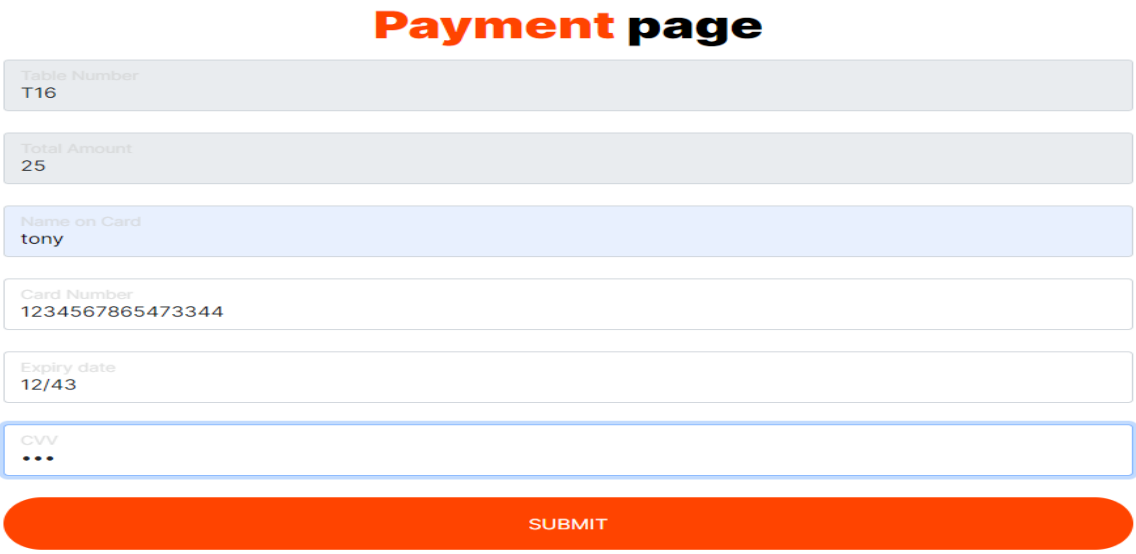

Figure 27. Payment page

The customer should enter his credit card details and make a payment. Once payment is submitted. It will show the status of the payment successful.

The restaurant admin has access to view and keep track of all the orders and customer payment details. Admin can view all the payment details with table number, id, email, and total amount to be paid by the customer.

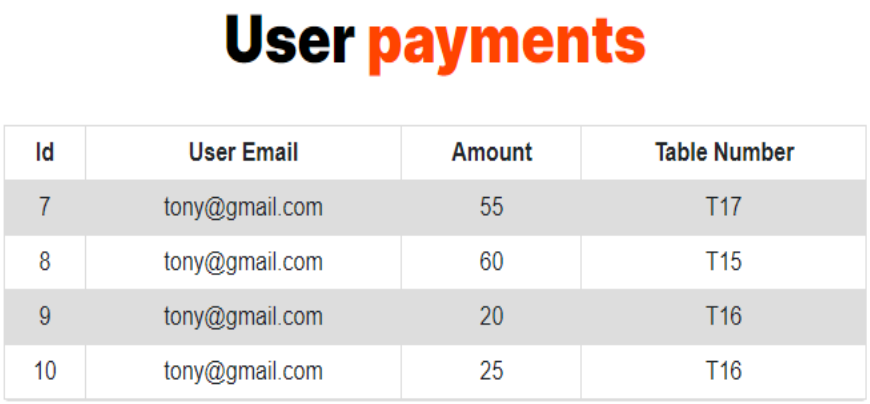

Figure 28. Review User Payments

CHAPTER SEVEN

CONCLUSION

The contactless food ordering system has improved by adding many features and functionalities to provide customers with a better dining experience. It provides numerous benefits to customers and restaurants. The key features of this system are it is easy to use, very secure and convenient, and time-saving are some of the advantages of using this system. Covid-19 has affected our day-to-day lives. Contactless food ordering has allowed many customers to maintain social distancing, reducing the need for physical contact while serving the food, and minimizing exposure to infectious diseases.

By using contactless food ordering system customers can easily access the menu, order food, and customize their orders. It also provides a convenient way to have a secure payment method. This payment process is very secure and transparent, and we can track the orders by eliminating the need to physically visit the restaurant.

In addition to these, many additional features can also be added to this system. Providing customers with hygienic standards in delivering the food and customer experience and efficiency plays a major role in this system.

CHAPTER EIGHT

FUTURE ENHANCEMENTS

A user-friendly interface makes it easier for customers to look for food items, customize their orders, and proceed to the ordering process seamlessly. A well-organized interface contributes to overall customer satisfaction.

QR code scanning in the application simplifies the process of accessing menus and payments. It is a convenient and efficient way to interact with customers with the interface. Real-time order tracking and notifications add convenience to the dining experience. Customers can stay updated on the progress of their orders from delivery to pick-up. These notifications provide valuable information; we can keep updates on orders and interact throughout the process.

These enhancements like improving the app interface, real-time order tracking, and notifications will help the customers in providing a better dining experience. Based on customer feedback and updating with the latest technologies will help the industry to maintain a competitive edge.

APPENDIX A

CRUCIAL CODE

```
"""
Django settings for contactlessfoodorderingsystem project.

Generated by 'django-admin startproject' using Django 4.1.4.

"""
from pathlib import Path
import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

TIME_ZONE = 'America/Los_Angeles'
SECRET_KEY = 'django-insecure-
4*5!b!!a_dr1a6jx(v26m7exe730ev=&76sgw#mssqxly7idu$'

DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'Restaurant',
    'Customer'
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
```

```
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'contactlessfoodorderingsystem.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR,"templates")],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'contactlessfoodorderingsystem.wsgi.application'

# Database
# https://docs.djangoproject.com/en/4.1/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE' : 'django.db.backends.mysql',
        'NAME' : 'Contactlessfoodorderingsystem',
        'HOST' : 'localhost',
        'USER' : 'root',
        'PASSWORD' : '',
        'PORT': 3306,

    }
```

```
}

# Password validation

AUTH_PASSWORD_VALIDATORS = [
   {
      'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
   },
   {
      'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
   },
   {
      'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
   },
   {
      'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
   },
]

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)

STATIC_URL = 'static/'
STATICFILES_DIRS=[os.path.join(BASE_DIR,'static')]

# Default primary key field type
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

APPENDIX B

CODE FOR DEFINING DATABASE MODELS

Customer

```
from django.db import models
from datetime import datetime
from django.conf.global_settings import TIME_ZONE  # importing time zone
from django.utils.timezone import utc

current_time = datetime.now()

# Create your models here.

class Customerdetails(models.Model):
    first = models.CharField(max_length=50)
    last = models.CharField(max_length=50)
    email = models.CharField(max_length=50)
    password = models.CharField(max_length=50)

    def __str__(self):
        return self.username

class Userqr(models.Model):
    restaurantid = models.CharField(max_length=50)
    email = models.CharField(max_length=50)
    status = models.CharField(max_length=50, default="pending")

    def __str__(self):
        return self.restaurantid

class Tablebooking(models.Model):
    restaurantid = models.CharField(max_length=50)
    email = models.CharField(max_length=50)
    tableno = models.CharField(max_length=50)
    memberscount = models.CharField(max_length=50)
    status = models.CharField(max_length=50, default="pending")

    def __str__(self):
        return self.restaurantid
class OrderItem(models.Model):
```

```
"""Order Item Model"""
price = models.IntegerField(null=True)
restaurent_id = models.CharField(max_length=100)
dish_id = models.CharField(max_length=100)
dish_count = models.CharField(max_length=100)
table_number = models.CharField(max_length=100)
dish_name = models.CharField(max_length=100)

order = models.ForeignKey("Customer.Useroder",
on_delete=models.CASCADE)

def __str__(self):
   return str(self.price)

class Useroder(models.Model):
"""Order Model Containing Many OrderItems"""
user_email = models.EmailField(null=True)
total_cost = models.IntegerField(default=0)
created_at = models.DateTimeField(auto_now_add=True, null=True)

def __str__(self):
   return self.user_email

class PaymentDetails(models.Model):
tablenumber = models.CharField(max_length=100,null=True)
resid = models.CharField(max_length=100,null=True)
Email = models.EmailField()
cardname = models.CharField(max_length=100,null=True)
cardno = models.CharField(max_length=100)
amount = models.CharField(max_length=100)
expirydate = models.CharField(max_length=100)
cvv = models.CharField(max_length=100)

class WaitlingList(models.Model):
"""Waiting List Model"""
time = models.DateTimeField(auto_now_add=True)
count = models.IntegerField(default=1)
table_number = models.CharField(max_length=50)
```

```
    user_email = models.EmailField(null=True)

    @property
    def get_time(self):
        now = current_time.utcnow().replace(tzinfo=utc)
        time_diff = now - self.time
        print(time_diff)
        diff = round((60) - (time_diff.total_seconds() / 60))
        return diff if diff > 0 else 0

    def __str__(self):
        return str(self.time)
```

Restaurant

```
from django.db import models
# import uuid

# Create your models here.
class Restaurant(models.Model):
    # id = models.UUIDField(primary_key=True, default=uuid.uuid4,
editable=False)
    firstname = models.CharField(max_length=100)
    lastname = models.CharField(max_length=100)
    email = models.CharField(max_length=100)
    password = models.CharField(max_length=100)
    restaurantname = models.CharField(max_length=100)
    qrcode = models.CharField(max_length=100,null=True)
    qrcodepath = models.CharField(max_length=100,null=True)
    def __str__(self):
        return self.restaurantname

class qrcode(models.Model):
    restaurantid = models.CharField(max_length=100)
    restaurantname = models.CharField(max_length=100,null=True)
    qrcode = models.CharField(max_length=100)
    qrcodepath = models.CharField(max_length=100)
```

```
    customeremail = models.CharField(max_length=100,null=True)
    status = models.CharField(max_length=100,null=True)
    def __str__(self):
        return self.restaurantid

class addfoodinfo(models.Model):
    restaurantid = models.CharField(max_length=100)
    foodname = models.CharField(max_length=100)
    foodprice = models.CharField(max_length=100)
    def __str__(self):
        return self.restaurantid

class tables(models.Model):
    restaurantid = models.CharField(max_length=100)
    memberscount = models.CharField(max_length=100)
    tableno = models.CharField(max_length=100)
    status = models.CharField(max_length=100,default="available")
    user_email = models.EmailField(null=True)

    def __str__(self):
        return self.restaurantid
```

REFERENCES

[1] N. A. Samsudin, S. K. Ahmad Khalid, “A Customizable Wireless Food Ordering System with Real-Time Customer Feedback”, 2011 IEEE Symposium on Wireless Technology and Applications (ISWTA), September 25-28, Langkawi, Malaysia, 2011.

[2] R. R. Chavan, S. K. Ravi Kurale, “Application of QR Code Technology to Automate Ordering System in Movie Theatres”, International Journal of Advance Research in Science and Engineering, Vol. 7, no. 1, 2018.

[3] C. R. S. Ramees, M. Sreerag, L. V. Urmila Pillai, “QR Code Based Smart Dining System” Journal of Engineering Sciences, Vol. 11, no. 5, 2020.

[4] B. Jagadeeswari, J. Soundariya, M. Jeevitha, and Y. Deepika,” The Implementation of the Virtual Reality Technology in the Hotel Marketing Management using IoT,” in 2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), pp. 1067-1072, 2022.

[5] V. Chavan, P. Jadhav, Priyanka Teli of Computer Department, Pune University Indapur, Maharashtra, India; “Implementing Customizable Online Food Ordering System Using Web Based Application”; IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2, 2015.

[6] A. Nikose, A. Hatwar, D. Adikane, “Cafeteria Food Ordering System using QR Code,” International Journal of Scientific Research in Science, Engineering, and Technology, vol. 10 no. 2, pp. 157-163, 2023.

[7] V. B. Ravi Pati, K. J. Mohan, P. S. Reddy, and S. Akula," Smart Restaurant System using QR Code," vol.21, pp. 382-396, 2022.

[8] S. Pieska, M. Liuska, Antti Auno of Centria University of Applied Sciences; "Intelligent Restaurant System Smart Menu That Digital Technology"; 2013 4th IEEE International Conference on Cognitive Info communications December 2–5, 2013, Budapest, Hungary.

[9] S. N. Cheong, et al., "Enriching Dining Experience with the Multi Touchable Entertainment Applications" Faculty of Engineering Multimedia University, Malaysia, 2010.

[10] C. S. Chang, et al., "Development and Implementation of a Restaurant for Customer-Centric Service Using WLAN and RFID Technologies" Department of Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan, 2008.

[11] T.P. Liang et al., "Adoption of mobile technology in business: a fit-viability model", Industrial Management & Data Systems, vol. 107, pp. 1154-1169, 2007.

[12] K. J. Patel et al., "PDA-based Wireless Food Ordering System for Hospitality Industry - A Case Study of Box Hill Institute", Wireless Telecommunications Symposium 2007, pp. 1-8, 2007.

[13] X. U. Hong Zhen, T. Bin, et al., "Wireless Food Ordering System Based on Web Services" Department of Computer Science and Technology, East China Institute of Technology Fuzhou, China, 2009.

[14] Y. C. Tan, et al., “Automated Food Ordering System with Interactive User Interface Approach” Faculty of Engineering and Science, University Tunk Abdul Rahman, Malaysia, 2010.

[15] M. H. A. Wahab, et al., “Implementation of Network-based Smart Order System” Faculty of Electrical and Electronic Engineering, University Tun Hussein Onn, Malaysia, 2008.