A Smart Food Ordering System with OCR-Based Menu Digitization

A PROJECT REPORT

Submitted by

ARUNKUMAR. R	**812621104014**
ARAVINDAN K	**812621104011**
ILAIYANITHI V	**812621104047**
JOSHUVA MANAVALAN S	**812621104051**

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

MAM COLLEGE OF ENGINEERING, TRICHY

ANNA UNIVERSITY :: CHENNAI 600 025

MAY 2025

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**A SMART FOOD ORDERING SYSTEM WITH OCR-BASED MENU DIGITIZATION"** is the Bonafide of **"ARUNKUMAR R (812621104014), ARAVINDAN K (812621104051), ILAIYANITHI V (812621104047), JOSHUVA MANAVALAN S (812621104051)"** who carried out the project work under my supervision.

SIGNATURE
PAVALAKODI.P
HEAD OF THE DEPARTMENT

COMPUTER SCIENCE AND ENGINEERING
M.A.M. COLLEGE OF ENGINEERING
SIRUGANUR, TRICHY

SIGNATURE
MURALIDHARAN
SUPERVISOR
ASSISTANT PROFESSOR
COMPUTER SCIENCE AND ENGINEERING
M.A.M. COLLEGE OF ENGINEERING
SIRUGANUR, TRICHY

Certificated that the candidate is examined by us on   in the project and viva practical examination of the Anna University, Chennai-25.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

“THANKS” is a simple word but its eloquence is magnified when it comes from the depth of the heart. We take this opportunity to thank all the encourages and supporters of this project.

First and foremost we praise and thank “the lord almighty” from the depth of our heart, which has been unfailing source of strength, comfort and inspiration of this word.

We sincerely thank our beloved Secretary DR.M.A.Mohamed Nizam, Our Principal Dr.Prof M.Shanmugapriya for providing all the necessary facilities to do our work successfully.

We take third opportunity to express our deep sense of gratitude to our Head of the department, Computer Science and Engineering P. PAVALAKODI and our guide & project coordinator MURALIDHARAN, for supporting us throughout our venture and for their invaluable interest and support, constructive suggestions, encouragement.

We are thankful to all the staff members of CSE, M.A.M College Engineering for their valuable comments during our project. A hearty thanks to the non-teaching staffs of CSE who provided us the necessary facilities and help in completing our project. We are very thankful to our parents, friends who have been so encouraging and supporting us morally and helping us in time of despair

ABSTRACT

The rapid digital transformation in the food service industry has led to a surge in demand for smart, contactless, and user-friendly ordering solutions. However, small and medium-scale restaurants often lack access to the tools and infrastructure required to embrace such technologies. This project, titled **“A Smart Food Ordering System with OCR-Based Menu Digitization”** aims to bridge that gap by offering a comprehensive, scalable, and easy-to-use web-based platform that modernizes the food ordering experience for both customers and restaurant owners.

At the heart of the system is **Optical Character Recognition (OCR)**, powered by **Tesseract OCR**, which allows restaurant owners to convert physical or handwritten menus into a structured digital format by simply uploading a photo. This digitized menu is then made accessible through an intuitive web interface, enabling customers to easily browse items and place orders online. The platform is built using the **Django web framework**, ensuring reliability, modularity, and robust backend management.

To support real-world usage, the system automatically generates a **custom QR code** for each restaurant, which can be printed and displayed at the venue. When customers scan the QR code using their smartphones, they are instantly redirected to the restaurant’s personalized digital menu, where they can place orders and

complete **direct payments**. These payments are securely deposited into the restaurant owner’s linked bank account, ensuring a seamless and cashless experience.

In addition to core ordering and payment functionality, the platform offers a suite of smart features such as **menu customization**, **discount code generation**, **advertisement placement**, and **customer engagement tools**. Restaurant owners can update items, run limited-time promotions, offer personalized coupon codes, and maintain real-time communication with customers via **WhatsApp Business integration**, which ensures instant delivery of order details directly to the restaurant’s mobile device.

This system eliminates the need for expensive POS terminals and instead empowers local food businesses to digitize their services independently. By combining OCR automation, mobile accessibility, smart analytics, and marketing capabilities, the project presents a practical, innovative, and scalable solution tailored for the modern food service landscape.

Table of Contents

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	**Full Form**
SMB	Small and Medium Business
OCR	Optical Character Recognition
UI	User Interface
UX	User Experience
API	Application Programming Interface
UPI	Unified Payments Interface
PWA	Progressive Web App
QR Code	Quick Response Code
DB	Database
CRUD	Create, Read, Update, Delete
OTP	One-Time Password *(if used in login/verification)*
PDF	Portable Document Format
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
HTTP	HyperText Transfer Protocol
SQL	Structured Query Language
Django	High-Level Python Web Framework *(not an abbreviation, but often treated like one in reports)*

Abbreviation	**Full Form**
VM	Virtual Machine
SMS	Short Message Service
POS	Point of Sale *(referenced in comparative studies)*
UTAUT2	Unified Theory of Acceptance and Use of Technology 2

CHAPTER 1

Introduction

1.1 Overview

In today’s fast-paced and digitally interconnected world, having a strong online presence is vital for businesses across all sectors, including the food service industry. However, Small and Medium-sized Businesses (SMBs), especially local restaurants, often struggle to create and maintain this presence due to high development costs, platform fees, and lack of technical expertise. While traditional methods such as printed menu cards and manual order taking are still in use, they fall short when it comes to reaching modern customers who expect seamless digital services, contactless ordering, and instant payments.

This project introduces "A Smart Food Ordering System with OCR-Based Menu Digitization," which is specifically designed to simplify and digitalize restaurant operations for SMBs. The system allows restaurant owners to upload a photo or PDF of their menu, which is then converted to a digital format using Optical Character Recognition (OCR). This digitized menu can be accessed by customers through a QR code that the restaurant can display on tables or entrances. Customers scan the QR code, browse the menu, place orders, and pay online—ensuring a completely contactless experience. The payment goes directly to the restaurant owner's account, avoiding intermediaries and commission fees.

Additionally, the platform enables restaurant owners to manage their menu, create promotional offers and coupons, run advertisements, and engage with customers via WhatsApp integration. The platform is built using the Django framework and focuses on ease of use, affordability, and accessibility, ensuring that even non-technical users can benefit from its features.

1.2 Background of the Study

The growing popularity of online food delivery platforms has transformed the restaurant industry, but it has also widened the digital gap between large restaurant chains and small local eateries. While major brands can afford to invest in dedicated mobile apps and web platforms, SMBs find it challenging to compete in the same digital space.

Many local restaurants rely solely on physical menus, handwritten orders, and cash-based payments. Although functional, these methods limit their customer reach and do not cater to modern-day expectations. Food aggregators often charge high commissions or require technical integration, discouraging smaller restaurants from adopting them.

Recognizing this digital divide, the proposed project aims to offer a lightweight yet powerful food ordering platform tailored for SMBs. The key innovation lies in using OCR to transform physical menus into digital versions without requiring manual entry. This significantly lowers the barrier to entry for digital transformation. The integration of QR codes simplifies access for customers, while WhatsApp API and online payment features enhance convenience for both restaurant owners and diners.

The system provides restaurant owners a way to not only manage their digital presence but also directly connect with customers, promote their brand, and stay competitive in a tech-savvy marketplace.

1.3 Problem Statement

Most small and medium-sized restaurants lack a robust digital presence due to high costs, lack of technical expertise, and dependence on third-party delivery apps that charge high commissions. Traditional order-taking methods and static menu cards limit their ability to reach new customers, streamline operations, or provide modern conveniences like contactless ordering and digital payments.

This results in:

- Difficulty attracting and retaining customers.
- Operational inefficiencies due to manual ordering and payments.
- Inability to offer promotions, update menu items quickly, or gather customer insights.
- High platform dependency and financial burden when relying on third-party food apps.

To address these challenges, this project proposes an easy-to-use, affordable, and intelligent system that allows restaurant owners to:

- Digitize physical menus using OCR with one-click upload.
- Generate and display QR codes to provide customers with instant access to digital menus.
- Receive and manage online orders without intermediary apps.
- Accept secure digital payments with direct bank deposits.
- Add coupon codes, advertise promotions, and customize their digital presence.
- Connect directly with customers via WhatsApp-based communication.

This platform is a complete digital solution built for non-technical restaurant owners who want to modernize their services and expand their customer base.

1.4 Objectives of the Project

The main objectives of the project are:

1. To provide an OCR-based tool that allows restaurant owners to digitize printed menus easily.
2. To develop a Django-based web application to handle order management, menu customization, and business operations.
3. To enable generation of unique QR codes for each restaurant to facilitate contactless access to menus.
4. To integrate WhatsApp Business API for real-time communication and order confirmation.
5. To implement digital payment gateways that deposit customer payments directly to the restaurant owner’s account.
6. To support promotional features like advertisements, coupon codes, and offers for increased customer engagement.
7. To offer a user-friendly admin interface that allows restaurant owners to customize menus, view insights, and manage orders.
8. To reduce platform dependency by empowering SMBs with an independent and scalable food ordering solution.

1.5 Scope of the Project

The system is primarily designed for small and medium-sized restaurants seeking a low-cost and easy-to-implement digital ordering solution. It includes the following features:

- OCR engine to digitize physical or scanned menu images.
- Django-based backend and responsive frontend for food menu browsing and ordering.
- QR code generation specific to each restaurant for display at tables or entrances.
- Customer interface for selecting items, placing orders, and paying online.
- WhatsApp Business API integration for real-time order notifications and confirmations.
- Dashboard for restaurant owners to update menu items, prices, availability, and promotions.
- Functionality for coupon management, advertisement banners, and customer engagement.

While delivery tracking and customer reviews are not part of this version, the system is modular and can be extended to include these in the future.

1.6 Significance of the Project

This project holds significant value for the following reasons:

- It bridges the digital divide for SMBs, allowing them to offer modern customer experiences without high technical or financial barriers.
- It removes the need for expensive mobile apps or dependence on aggregator platforms.
- It empowers restaurant owners to digitize their business with a simple upload of a menu image.
- It provides customers with a fast, clean, and contactless ordering and payment experience.
- It supports direct owner-customer communication and brand-building through WhatsApp.

- It enhances visibility and marketing for local eateries through ad and promotion tools.

By enabling a plug-and-play style digital transformation, the system serves as a catalyst for local restaurant growth, giving SMBs a competitive edge in today's digital-first market.

1.7 Literature Review

In this research paper, the authors from Zomato explore how restaurant menus can be digitized efficiently using Amazon Textract for OCR and Amazon SageMaker for categorizing food items into structured data. The goal was to reduce manual data entry and speed up restaurant onboarding on their platform. The method involves scanning physical menus and extracting structured digital text using machine learning. The main merits include its scalability, automation of repetitive tasks, and time-saving capabilities. However, it also requires high-quality input images and incurs additional cloud service costs. This study validates the feasibility of OCR technology in real-world restaurant platforms.

In this research paper published in the International Journal of Engineering Research & Technology (IJERT), a QR-code-based ordering system was designed to allow customers to scan a code, view menus, and place orders without interaction. Developed using PHP and basic web technologies, this solution offers a hygienic and user-friendly experience, particularly suited to small and mid-sized restaurants. It eliminates the need for mobile apps and supports easy access. Its merits include ease of use and rapid deployment, while its demerits involve lack of deeper engagement or offline functionality. The paper highlights QR-based systems as a practical contactless ordering solution.

In this research paper from the Journal of Retailing and Consumer Services (Elsevier), the authors analyze the behavioral acceptance of QR code menus in the post-pandemic dining experience. Using the UTAUT2 model, the study identifies factors like ease of use, privacy, and social influence that affect adoption. This research is entirely survey-based and provides insight into how customers perceive QR technologies. While it gives strong behavioral data, it lacks system architecture or implementation aspects. The paper supports QR ordering systems by understanding customer habits and expectations, helping in the design of more intuitive interfaces

In this research paper published in the International Journal of Hospitality Management, a smart restaurant system integrating QR code ordering and real-time analytics was developed. The backend dashboard analyses customer behaviour, sales patterns, and order volume through dynamic charts. This system enhances decision-making and business intelligence in food services. The key strengths are in its data visualization features and performance analytics, although it requires more complex backend infrastructure. The study is significant for future enhancements in your system involving admin-level analytics and reporting.

In this research paper from IRJMETS, a full-stack QR-based ordering platform is proposed using PHP, MySQL, and HTML/CSS. The system covers the complete workflow: from customer ordering to admin control and kitchen display. It provides basic but effective digitalization of restaurant processes. Its strengths include modularity and simple tech stack for implementation. However, the UI is outdated and lacks modern touch support, and it does not integrate with messaging APIs. The project demonstrates a ground-up system ideal for small establishments.

In this research paper published by IJCRT, the authors present an innovative food ordering system that sends order details directly to a restaurant's WhatsApp via the Business API. Developed using PHP and Firebase, it removes the need for a dedicated dashboard by leveraging WhatsApp as the primary interface. The approach is simple and aligns well with SMB workflows. Its advantages include accessibility and ease of integration. However, it depends heavily on third-party API uptime and lacks automated confirmation unless callbacks are supported. The research validates WhatsApp as an effective tool for direct order communication.

CHAPTER 2

System Analysis

2.1 Introduction

The development of any software solution begins with a critical examination of the problem it aims to solve. System analysis involves understanding existing workflows, identifying inefficiencies, and defining the scope of the proposed system in terms of technological feasibility and operational relevance. In this chapter, we analyze the limitations of current food ordering processes in small and medium-sized restaurants, especially in Tier-2 and Tier-3 cities, and present the rationale and detailed framework for the proposed smart food ordering system that leverages OCR and WhatsApp integration for seamless menu digitization and order communication.

2.2 Existing System

Traditional food ordering workflows in many small eateries and local restaurants still rely heavily on manual processes. Menus are typically static—either printed on paper, handwritten, or uploaded as unstructured images on platforms like Zomato or Swiggy. Orders are received via phone calls or direct WhatsApp messages, which, while convenient, introduce scope for errors such as wrong item selection, poor order tracking, and ambiguity in communication.

Major food delivery platforms do provide digital menus, but they also introduce challenges:

- High commission rates (often 20–30%) reduce margins for restaurant owners.
- Menu management and updates require interaction with third-party support teams.
- Restaurants are dependent on aggregator algorithms for visibility.
- Many small businesses lack the technical expertise or budget to build and manage their own ordering platforms.

Moreover, restaurants without aggregator presence often rely solely on WhatsApp for orders. While WhatsApp is user-friendly, it lacks structured workflows for handling item selection, order confirmation, and customer interaction, resulting in inconsistent service delivery and missed orders.

References:

- Zomato plans to use OCR to improve menu digitization
- Zomato deploys OCR for digitisation
- DotPe’s initiative to empower SMBs with digital ordering

2.3 Limitations of the Existing System

The limitations inherent in current systems are both technological and operational:

- **Manual Dependency:** Orders are handwritten or orally communicated, leading to misunderstandings or incorrect order fulfillment.
- **Lack of Automation:** No structured mechanism exists for tracking orders or automating confirmations.
- **Low Digital Presence:** Small eateries are digitally invisible or hard to discover online.
- **Aggregator Reliance:** High dependence on third-party apps, which reduce profit margins and restrict control.
- **No Feedback Loop:** Traditional systems lack integration for capturing customer feedback or improving service based on insights.
- **Non-searchable Menus:** Uploaded images or PDF menus are not machine-readable, making it harder to index or update them dynamically.

2.4 Proposed System

The proposed system is a smart, lightweight, and scalable web application designed specifically for small and medium-sized restaurants to digitize their food menus using OCR (Optical Character Recognition) and automate the order workflow through WhatsApp. It removes the need for a physical POS system or a full-fledged delivery aggregator by introducing a digitized yet familiar approach using tools that restaurant owners and customers already use.

Key Components:

- **OCR-Based Menu Digitization:** Printed or handwritten menus are converted to structured digital text using the Tesseract OCR engine. This allows restaurants to upload images of their menus, which are then parsed and rendered into a searchable, interactive digital format.
- **User-Friendly Ordering Interface:** Customers can access the restaurant's dedicated web link to browse the digitized menu and place orders directly.
- **WhatsApp Order Integration:** Once an order is placed, the system sends detailed order information to the restaurant owner's WhatsApp using the

WhatsApp Business API. This message includes items ordered, total cost, and a one-tap button to confirm or reject the order.

- **Order Confirmation Automation:** By clicking the confirmation button, the system triggers backend logic to notify the customer of the confirmed order and update internal logs, thereby reducing response delays and manual intervention.
- **Custom Dashboard (Admin Panel):** Restaurant owners have access to a backend dashboard for managing incoming orders, updating menu items, and tracking customer interactions.

This system ensures that even the smallest restaurants can operate digitally without complex onboarding, technical maintenance, or high operational costs.

2.5 Feasibility Study

2.5.1 Technical Feasibility

The solution is architected using open-source and stable technologies:

- **Backend:** Django (Python), providing scalability and a secure backend structure.
- **Frontend:** HTML, CSS, and JavaScript for responsive user interaction.
- **OCR Engine:** Tesseract OCR, an industry-standard open-source library for text recognition.
- **Communication:** WhatsApp Business API for real-time messaging and action buttons.

The system architecture is modular and extendable, enabling future integration with payment gateways, inventory tracking, or customer analytics tools.

2.5.2 Operational Feasibility

The primary users—restaurant owners—are already familiar with WhatsApp, reducing the learning curve. Uploading menu images and setting up the platform requires minimal technical skills. The system can be operated via mobile devices, making it accessible and practical in low-tech environments.

2.5.3 Economic Feasibility

- **Development Cost:** Minimal, since the stack uses free or open-source technologies.
- **Maintenance Cost:** Negligible for single-instance deployments.
- **Revenue Model:** The platform can be extended as a SaaS product, where restaurants pay a small monthly fee, significantly cheaper than aggregator commissions.

The ROI (Return on Investment) is high, especially in regions underserved by mainstream tech platforms.

2.6 System Requirements

2.6.1 Hardware Requirements

- **Server:** Minimum 2 GB RAM, 1 vCPU (suitable for deployment on cloud providers like DigitalOcean or AWS EC2).
- **Client Devices:** Any modern smartphone, tablet, or desktop with internet connectivity.

2.6.2 Software Requirements

- **Operating System (Server):** Ubuntu 20.04 LTS or above.
- **Web Browsers (Client):** Chrome, Firefox, Safari (latest versions).
- **Dependencies:**
  - Python 3.9+
  - Django 4.x
  - Tesseract OCR Engine
  - SQLite for database
  - WhatsApp Business API (On-Premises)

CHAPTER 3

System Design

3.1 System Architecture

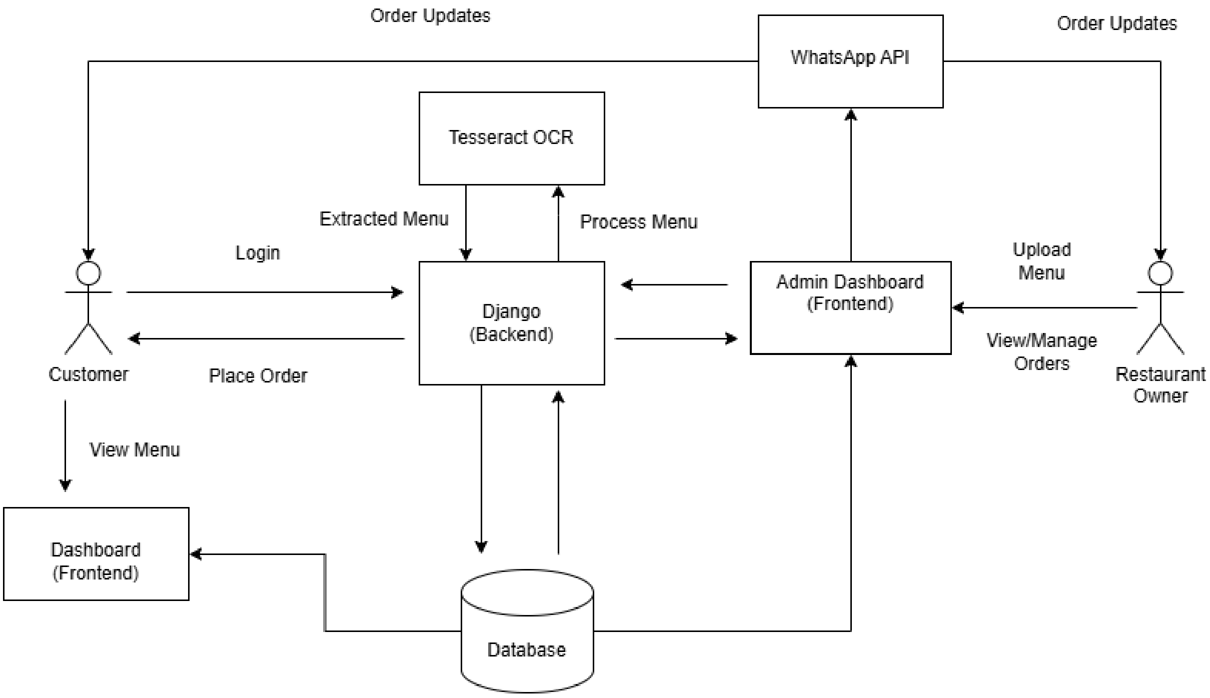
The system architecture is a critical part of any software project as it lays the foundation for how various components of the system interact with each other. For the smart food ordering system with OCR-based menu digitization, the architecture follows a client-server model, where the client (the user) interacts with the frontend (web application), and the server (the backend) handles the logic, data processing, and integrations.

The architecture can be divided into several key components:

1. **Frontend**: The client interface is a web application that allows customers to view digitized menus and place orders. This frontend is built using HTML, CSS, and JavaScript, ensuring an intuitive and responsive design across various devices.
2. **Backend**: The backend is built using Django (Python), which handles requests from the frontend, processes data, interacts with the database, and integrates with external services such as the Tesseract OCR engine and the WhatsApp Business API.
3. **OCR Module**: The Tesseract OCR engine is used to digitize the restaurant menu by extracting text from uploaded images or PDFs. The extracted data is then structured and stored for use by customers.
4. **Database**: The database stores the digitized menu, user orders, restaurant details, and transaction logs. It ensures efficient management and retrieval of information when needed.
5. **WhatsApp Integration**: The system integrates with the WhatsApp Business API, allowing restaurants to receive order details via WhatsApp. The restaurant owner can confirm or reject the order directly through WhatsApp buttons.
6. **Admin Dashboard**: A web-based dashboard is provided to restaurant owners for managing orders, viewing order history, and analyzing customer interactions.

The diagram for this architecture should include these components with arrows to represent the communication flow between them. For instance:

- The frontend communicates with the backend to retrieve digitized menu data and submit orders.
- The backend interacts with the OCR engine to process menu images.
- The backend sends order notifications to the restaurant owner via WhatsApp.
- The database stores all the data and supports retrieval operations.



3.2 UML Diagrams

3.2.1 Use Case Diagram

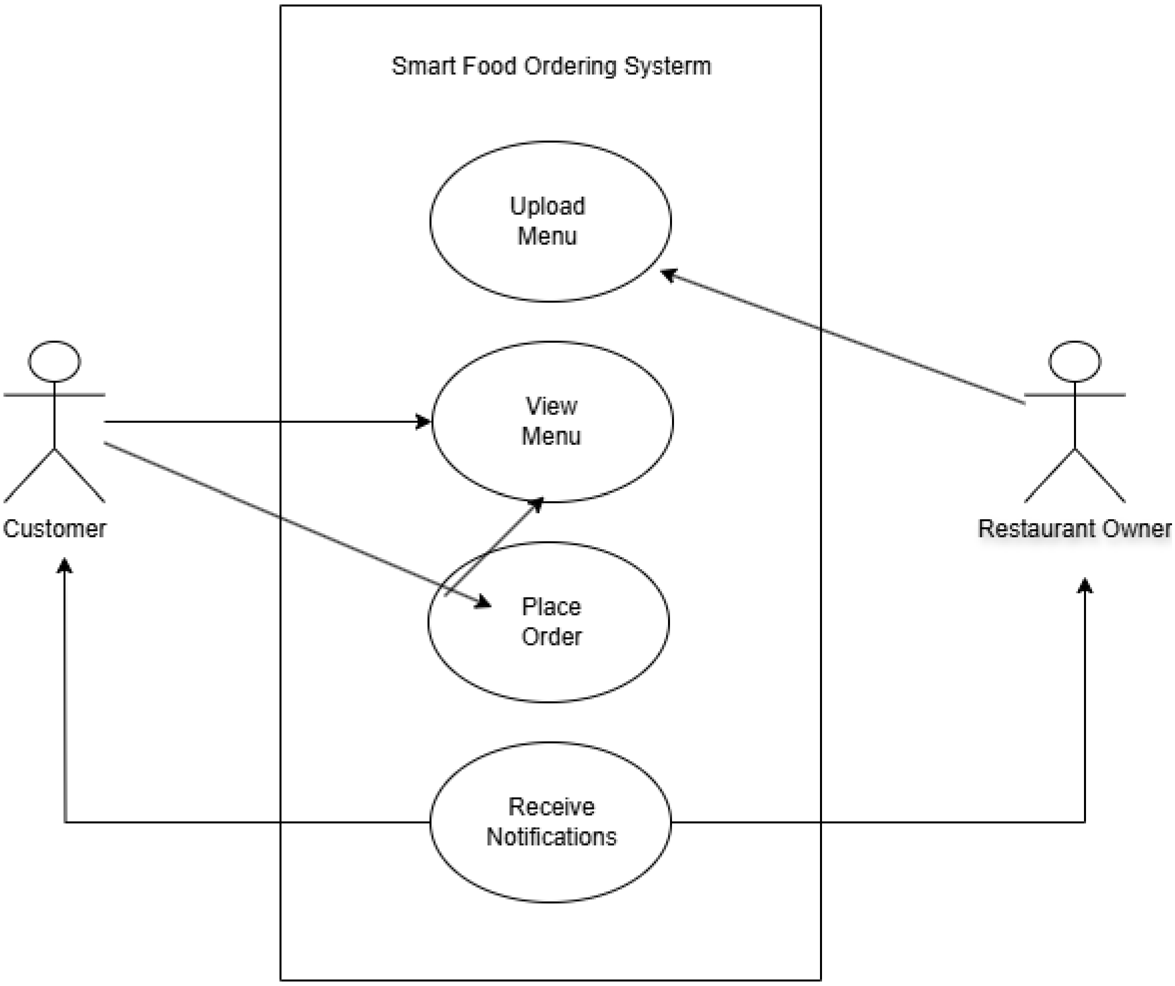
The use case diagram identifies the main users of the system and their interactions with it. The primary actors in this system are:

- **Customer**: A customer views the digitized menu, places an order, and tracks order status.
- **Restaurant Owner**: The restaurant owner receives order notifications via WhatsApp and confirms or rejects the orders.

- **Admin**: The admin (restaurant management) oversees the entire system and manages restaurant-specific settings.

The use case diagram for this system can include interactions such as:

- Customer views menu and places an order.
- Customer receives order confirmation and status updates.
- Restaurant owner views new orders and confirms them.
- Admin manages the restaurant's digital presence and menu.

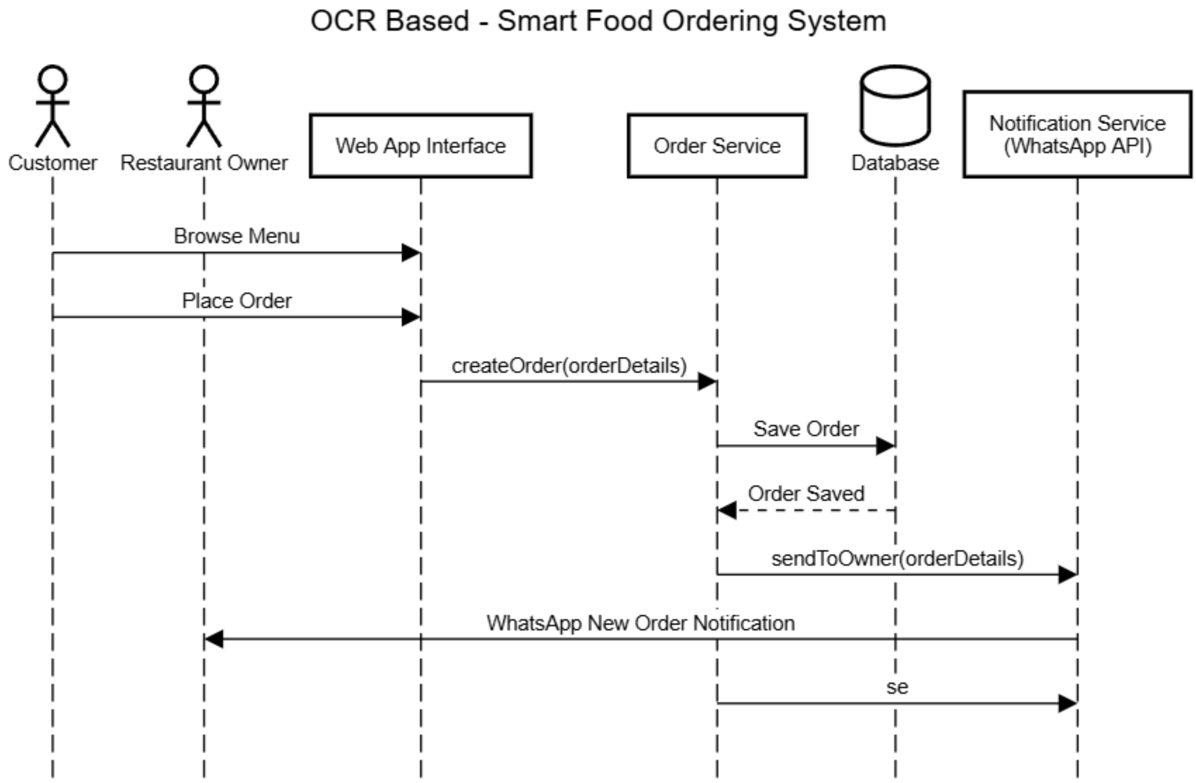


3.2.2 Sequence Diagram

A sequence diagram shows the sequence of operations between different components to complete a specific process. For this system, we can create a sequence diagram for the **order placement** process:

1. The customer logs into the web application.

2. The customer selects a menu item.
3. The system retrieves the menu data (from the database).
4. The customer places the order.
5. The order is sent to the backend, which processes it.
6. The backend triggers the WhatsApp Business API to notify the restaurant owner.
7. The restaurant owner confirms or rejects the order via WhatsApp.
8. The system updates the customer with the order status.

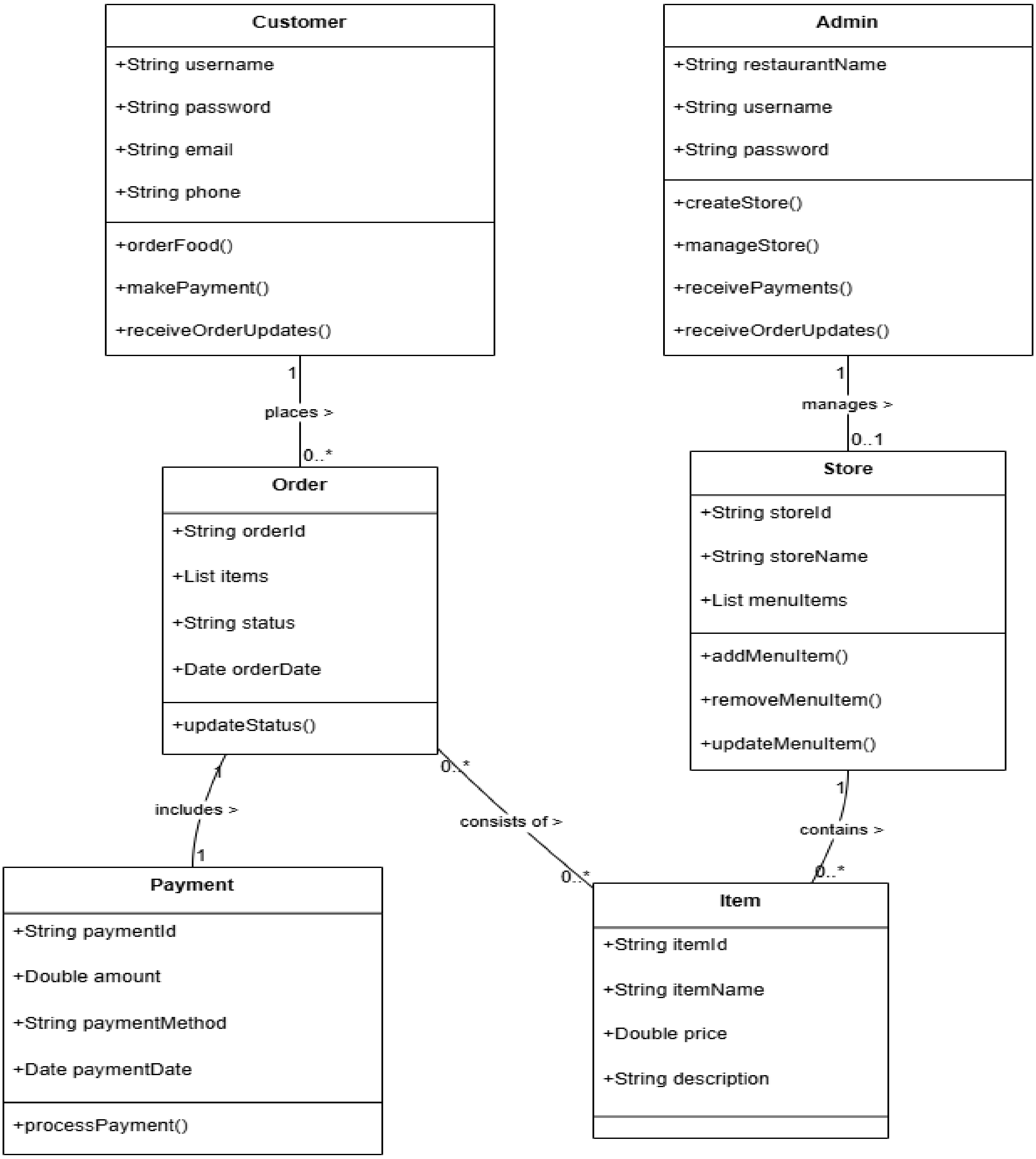

OCR Based - Smart Food Ordering System
Customer
Restaurant Owner
Web App Interface
Order Service
Database
Notification Service (WhatsApp API)
Browse Menu
Place Order
createOrder(orderDetails)
Save Order
Order Saved
sendToOwner(orderDetails)
WhatsApp New Order Notification
se

3.2.3 Class Diagram

The class diagram illustrates the system's structure and defines the main classes, their attributes, and their relationships. Key classes for your system include:

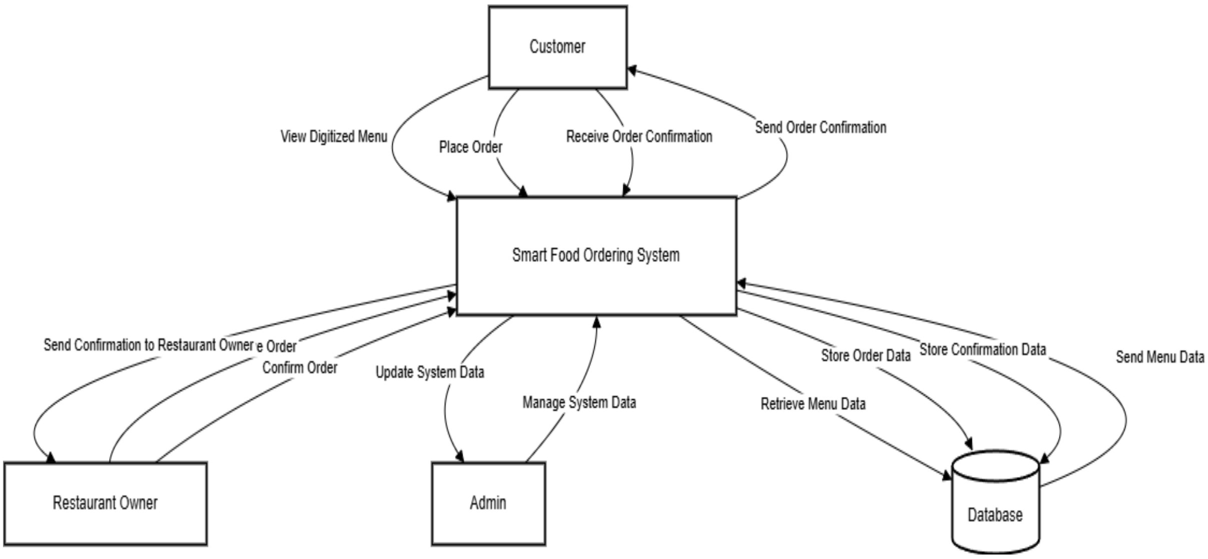
- **User**: Represents the customer with attributes such as name, email, and order history.
- **Menu**: Represents the restaurant menu, including details like item name, description, and price.

- **Order**: Represents a customer's order, containing order details like items, quantity, and order status.
- **Restaurant**: Represents the restaurant, including details like name, contact information, and menu.
- **WhatsAppNotification**: Represents the interaction with the WhatsApp Business API for sending and receiving notifications.

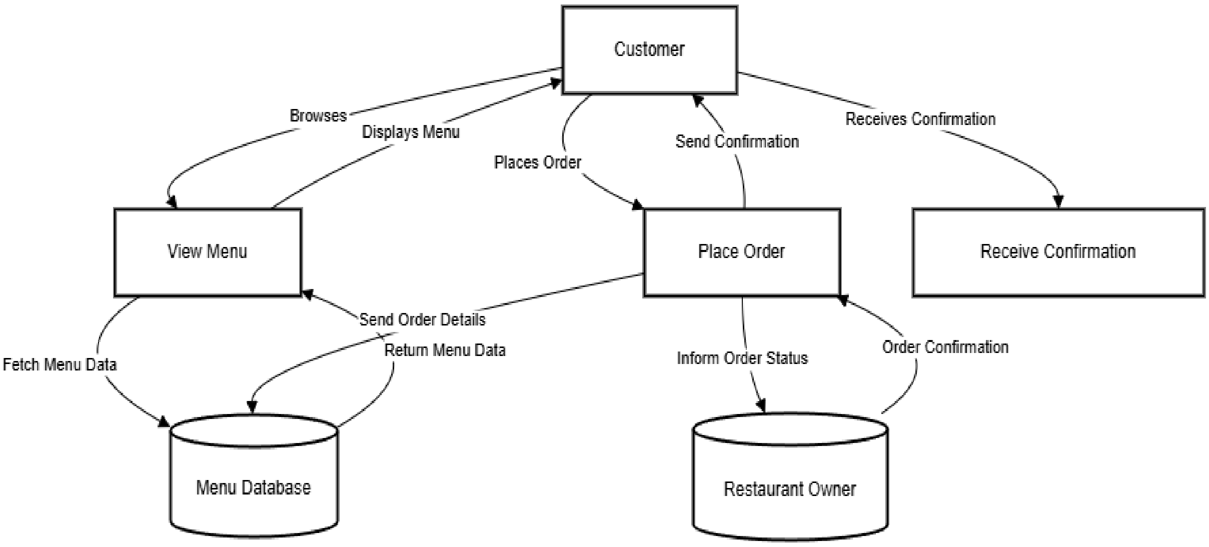


3.3 Data Flow Diagrams (DFD)

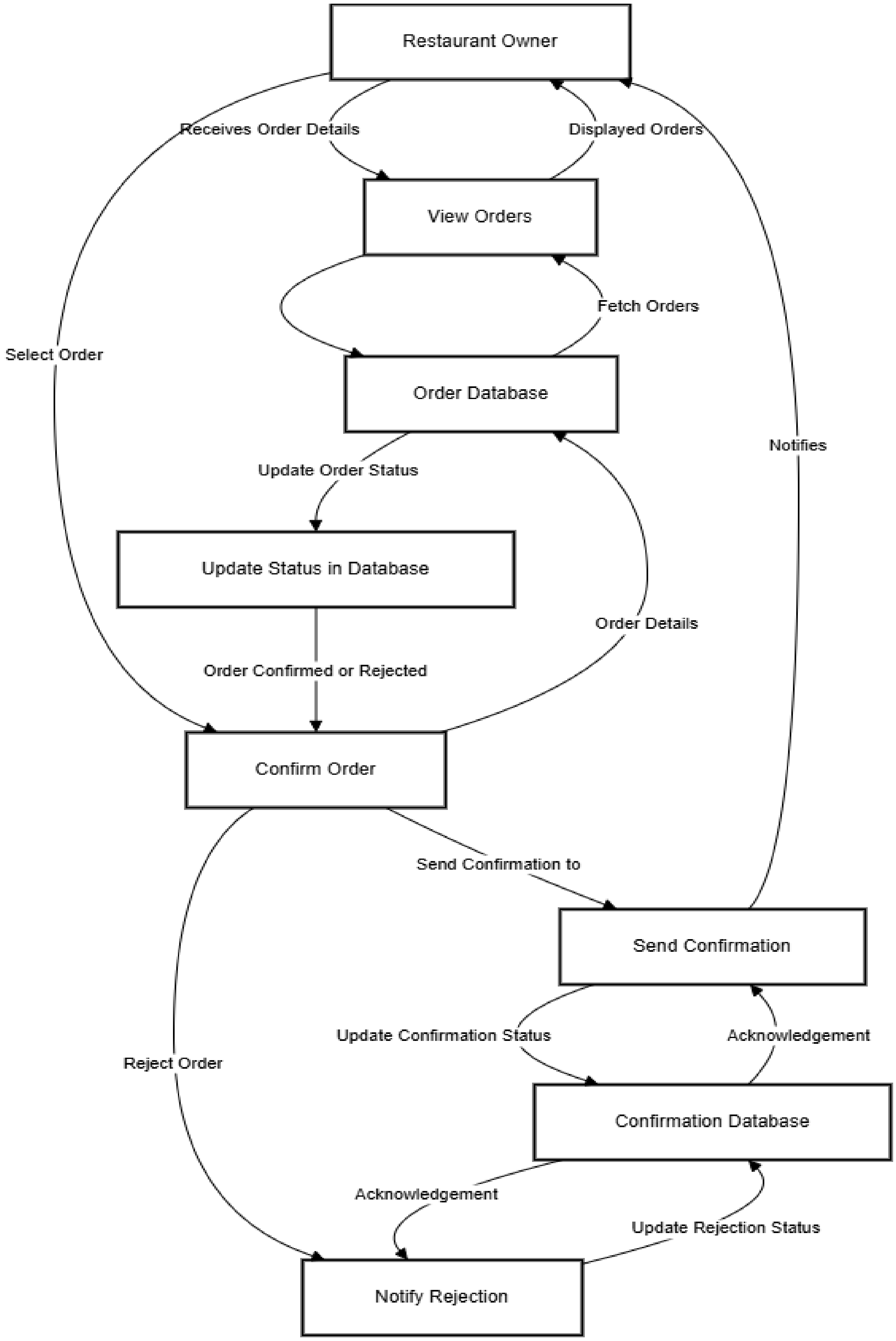
Level 0 DFD: This represents the high-level flow of data between the external entities (customer, restaurant owner) and the system. It includes processes like menu viewing, order placement, and order confirmation.



Level 1 DFD: This breaks down the main processes into sub-processes. For example, the "Order Placement" process would be further broken down into sub-processes like "Validate Order," "Send Order to Restaurant," and "Confirm Order."

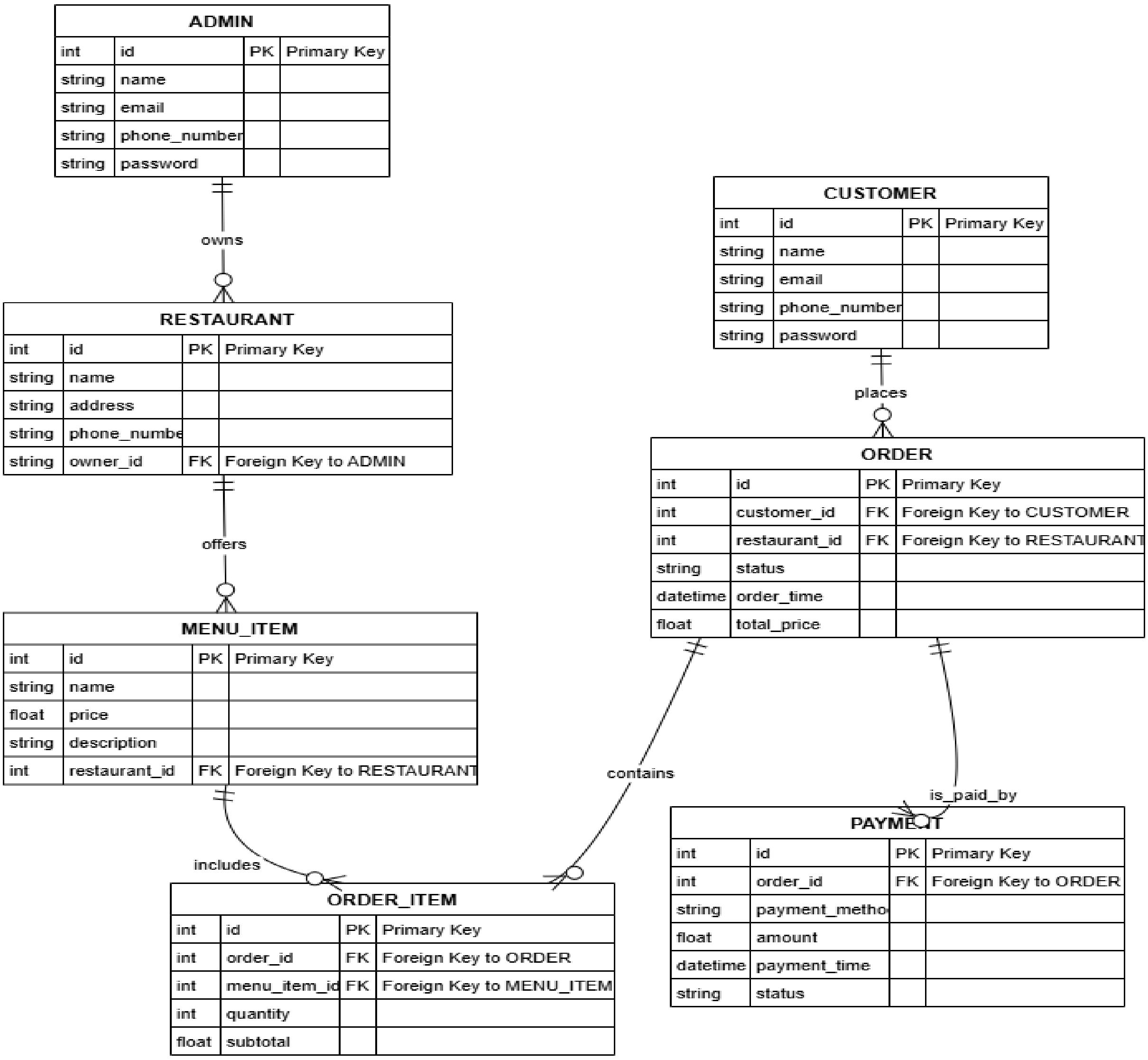


Level 2 DFD: This can be created for more detailed processes, such as "OCR-based Menu Digitization," showing the flow of data between the OCR engine, database, and the backend.



3.4 ER Diagram (Entity-Relationship Diagram)

- Customer: Stores customer details like ID, name, phone, and email.
- Restaurant: Holds restaurant info such as ID, name, address, and contact.
- Menu_Item: Represents food items with item ID, name, price, and restaurant ID (foreign key).
- Order: Contains order ID, customer ID, restaurant ID, status, and timestamp.
- Order_Item: Links orders and menu items with quantity and price.
- Payment: Manages payments with payment ID, order ID, amount, and payment status.
- Admin: Stores admin user details like ID, username, and role for system management.



ADMIN

int	id	PK	Primary Key
string	name		
string	email		
string	phone_number		
string	password		

RESTAURANT

int	id	PK	Primary Key
string	name		
string	address		
string	phone_numbe		
string	owner_id	FK	Foreign Key to ADMIN

MENU_ITEM

int	id	PK	Primary Key
string	name		
float	price		
string	description		
int	restaurant_id	FK	Foreign Key to RESTAURANT

CUSTOMER

int	id	PK	Primary Key
string	name		
string	email		
string	phone_number		
string	password		

ORDER

int	id	PK	Primary Key
int	customer_id	FK	Foreign Key to CUSTOMER
int	restaurant_id	FK	Foreign Key to RESTAURANT
string	status		
datetime	order_time		
float	total_price		

ORDER_ITEM

int	id	PK	Primary Key
int	order_id	FK	Foreign Key to ORDER
int	menu_item_id	FK	Foreign Key to MENU_ITEM
int	quantity		
float	subtotal		

PAYMENT

int	id	PK	Primary Key
int	order_id	FK	Foreign Key to ORDER
string	payment_metho		
float	amount		
datetime	payment_time		
string	status		

3.5 Module Description

1. Customer Module

This module serves as the interface for end-users (customers) who wish to browse restaurant menus and place orders. Customers can either scan physical menus using the OCR functionality or access digital menus directly through the platform. The module provides features such as menu browsing, item selection, cart management, placing orders, viewing order history, and making secure payments. It ensures a smooth and user-friendly experience for customers.

2. Restaurant Owner Module

The Restaurant Owner Module allows restaurant managers to receive and manage customer orders. Orders are sent directly to the owner's WhatsApp using the WhatsApp Business API, where they can accept or reject orders via interactive buttons. The module also includes functionality to update and manage menu items, monitor incoming orders, and maintain the restaurant profile. It simplifies order handling and improves communication between customers and restaurants.

3. Admin Module

The Admin Module oversees the entire system and acts as the central point of control. Admins have the authority to manage user accounts, approve or deactivate restaurant registrations, monitor system performance, and handle complaints or disputes. This module provides tools for managing backend operations and ensures the system runs securely and efficiently.

4. OCR Digitization Module

This module focuses on converting physical or handwritten menus into structured digital formats. It employs Optical Character Recognition (OCR) to extract text (such as item names and prices) from uploaded images of restaurant menus. The processed data is then stored in the database and linked with the respective restaurant's profile, allowing for fast onboarding of restaurants onto the platform.

5. Ordering & Cart Module

The Ordering & Cart Module handles the core e-commerce functionality of the system. Users can add or remove items from their cart, modify quantities, and proceed to checkout. The module calculates the total cost and manages the order placement process. Once an order is placed, the details are sent to the restaurant owner via the integrated WhatsApp module.

6. WhatsApp API Integration Module

This module enables seamless communication between the platform and restaurant owners through the WhatsApp Business API. Upon order placement, a structured message containing the order details is sent to the restaurant's registered WhatsApp number. The module also tracks responses for order confirmation or rejection, and updates the system accordingly.

7. Payment Module

The Payment Module ensures that the transaction process is secure and efficient. It integrates with popular third-party payment gateways to support online payments. After successful payment, the module updates the order status and logs payment information for future reference. It also handles payment failures and ensures that the customer is notified of the transaction status.

CHAPTER 4

System Implementation

4.1 Technologies Used

The development of the "Smart Food Ordering System with OCR-Based Menu Digitization" incorporates a variety of robust, scalable, and open-source technologies that are widely accepted in the software industry. These technologies were carefully selected to ensure that the system is efficient, maintainable, and cost-effective for small and medium-sized businesses.

- **Frontend:**
  - HTML5 and CSS3 for structuring and styling web pages.
  - JavaScript (Vanilla JS) for implementing client-side logic.
- **Backend:**
  - Django (Python-based web framework) for building a secure and scalable backend.
- **Database:**
  - SQLite for development and testing.
  - PostgreSQL or MySQL recommended for production deployment.
- **OCR Engine:**
  - Tesseract OCR to convert uploaded images or PDFs into digital, searchable text.
- **Messaging Platform:**
  - WhatsApp Business API for sending order notifications to restaurant owners.
- **Hosting Services:**
  - PythonAnywhere / Heroku / DigitalOcean / AWS for web hosting.
- **Version Control System:**
  - Git for source code management.
  - GitHub for remote repository and collaboration.

These technologies collectively provide a lightweight, easy-to-use, and modular solution that can be adopted by non-technical restaurant owners.

4.2 Module-Wise Implementation

To ensure modularity and maintainability, the system is divided into several independent yet interrelated modules. Each module is responsible for specific functionality within the system.

4.2.1 Menu Digitization Module

- Provides the interface to upload scanned menu images or PDFs.
- Uses Tesseract OCR to recognize and extract text from the uploaded media.
- Parses recognized text into structured menu items and stores them in the database.
- Allows manual correction and formatting if OCR results are imperfect.

4.2.2 Admin Dashboard Module

- Secure login system for restaurant owners.
- Allows CRUD (Create, Read, Update, Delete) operations for menu items.
- Displays QR code for customer ordering interface.
- Shows analytical data such as daily orders, revenue, and most-ordered items.
- Provides settings for restaurant details, payment configurations, and contact information.

4.2.3 Customer Ordering Module

- Allows customers to scan the QR code placed in the restaurant.
- Redirects to the restaurant-specific ordering webpage.
- Enables browsing through categories and adding items to the cart.
- Collects customer name and contact number for identification.
- Validates the order before submission.

4.2.4 WhatsApp Notification Module

- Once an order is placed, details are formatted and sent to the restaurant's WhatsApp Business number.
- Includes list of ordered items, quantities, customer name, contact, and total amount.
- (Optional) Future scope includes clickable confirmation buttons for restaurant staff.

4.2.5 Payment Module

- Integrates UPI links or Razorpay for secure digital payments.
- Redirects customers to the payment gateway after order confirmation.
- On success/failure, updates the payment status in the backend.
- Sends payment success notification to both customer and restaurant owner (via email or WhatsApp).

4.3 Database Design

The system relies on a relational database that stores and manages data related to restaurants, menus, orders, and payments. The following are key entities and their attributes:

- **Restaurant Table:**
  - Restaurant_ID (Primary Key)
  - Name
  - Contact Number
  - Email
  - WhatsApp Number
  - Address
  - Password (hashed)
- **MenuItem Table:**
  - Item_ID (Primary Key)

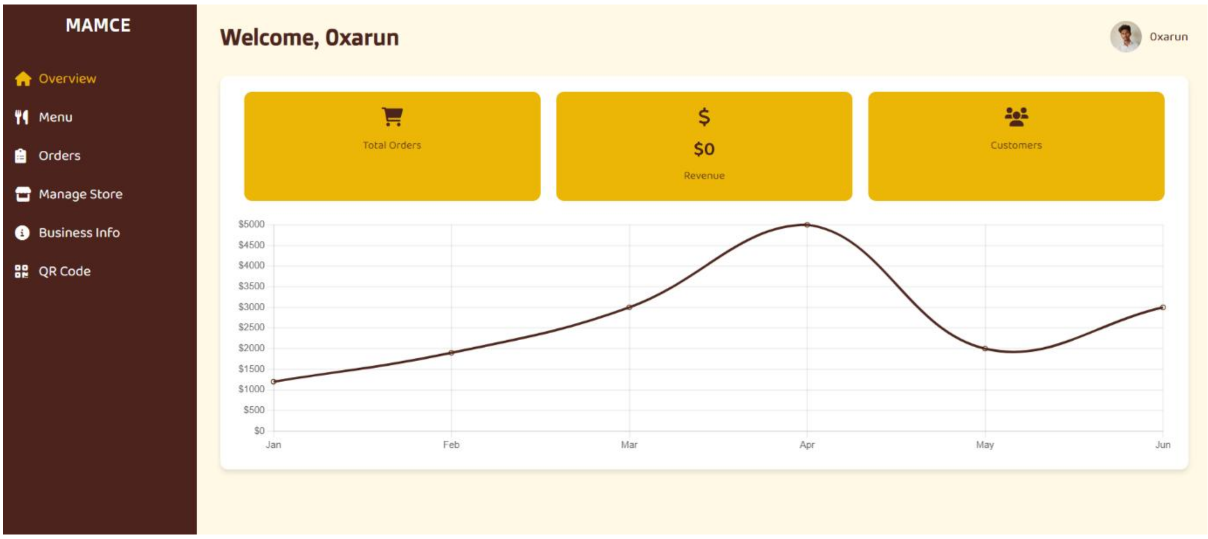
    - Restaurant_ID (Foreign Key)
    - Item Name
    - Category
    - Description
    - Price
    - Availability (Boolean)
- **Order Table:**
    - Order_ID (Primary Key)
    - Restaurant_ID (Foreign Key)
    - Customer Name
    - Customer Phone
    - Order Date/Time
    - Total Amount
- **OrderItem Table:**
    - ID (Primary Key)
    - Order_ID (Foreign Key)
    - Item_ID (Foreign Key)
    - Quantity
    - Subtotal
- **Payment Table:**
    - Payment_ID (Primary Key)
    - Order_ID (Foreign Key)
    - Method (UPI, Razorpay, etc.)
    - Status (Success/Failed)
    - Transaction ID
    - Timestamp

These tables are interrelated using primary-foreign key relationships and ensure data consistency across the platform.

4.4 Screenshots and Explanation

Admin Dashboard

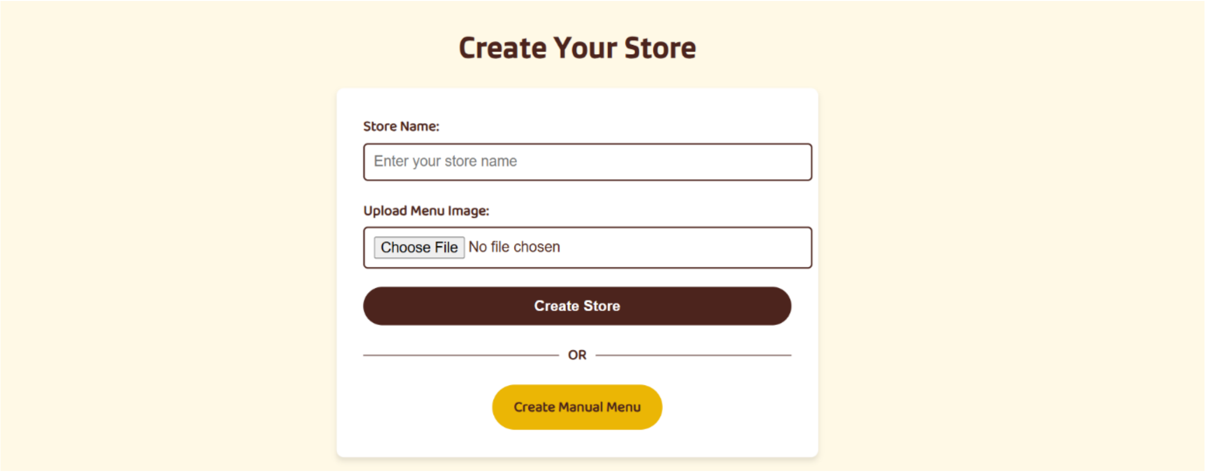
- Displaying overview, menu management, QR code, and analytics.
- Shows restaurant performance insights including daily revenue and order count.
- Easy toggle for availability status of menu items.



Menu Upload Page

- Before and after OCR result with manual correction interface.
- Real-time preview of digitized menu before saving.
- Restaurant owners can upload their menu images, and extracted text can be edited before saving.
- A separate menu management screen allows owners to manually add, update, or delete menu items.

Screenshot 1: Upload Page with OCR input and correction interface.

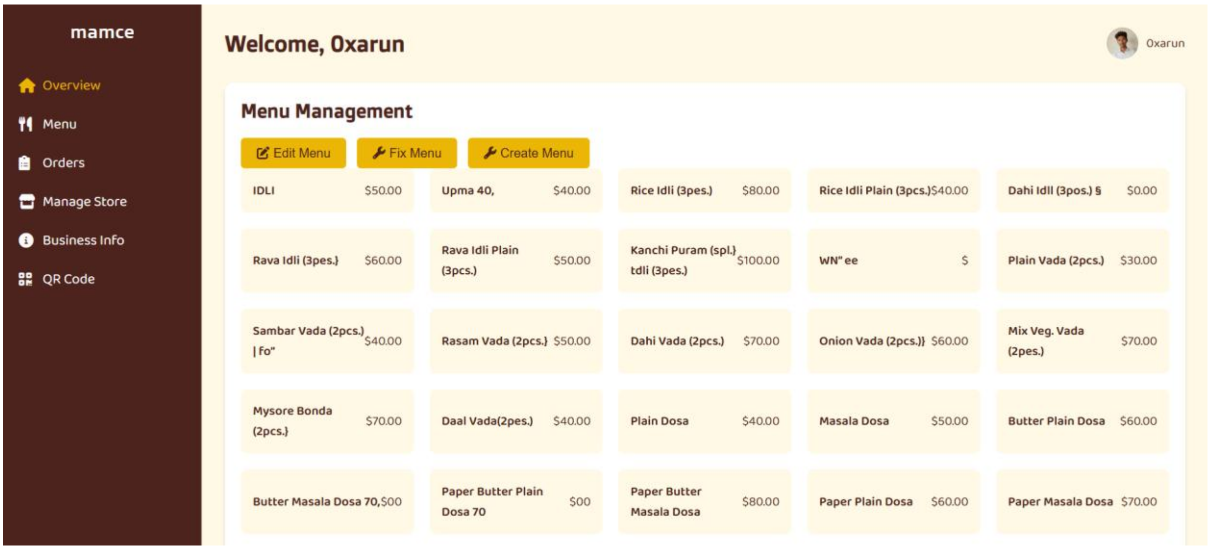


Create Your Store

Store Name: Enter your store name

Upload Menu Image: Choose File No file chosen

Screenshot 2: Menu Management Dashboard showing editable item list.



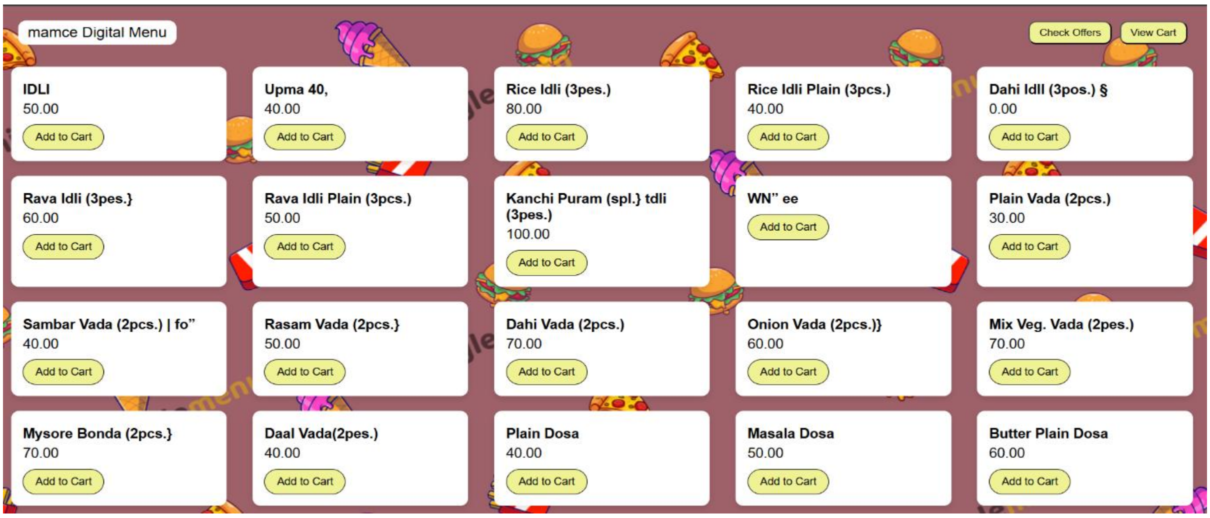
Welcome, Oxarun

Menu Management

IDLI	$50.00
Upma 40,	$40.00
Rice Idli (3pes.)	$80.00
Rice Idli Plain (3pcs.)	$40.00
Dahi Idll (3pos.) §	$0.00
Rava Idli (3pes.}	$60.00
Rava Idli Plain (3pcs.)	$50.00
Kanchi Puram (spl.} tdli (3pes.)	$100.00
WN" ee	$
Plain Vada (2pcs.)	$30.00
Sambar Vada (2pcs.) | Fo"	$40.00
Rasam Vada (2pcs.}	$50.00
Dahi Vada (2pcs.)	$70.00
Onion Vada (2pcs.)}	$60.00
Mix Veg. Vada (2pes.)	$70.00
Mysore Bonda (2pcs.}	$70.00
Daal Vada(2pes.)	$40.00
Plain Dosa	$40.00
Masala Dosa	$50.00
Butter Plain Dosa	$60.00
Butter Masala Dosa 70,	$00
Paper Butter Plain Dosa 70	$00
Paper Butter Masala Dosa	$80.00
Paper Plain Dosa	$60.00
Paper Masala Dosa	$70.00

Customer Ordering Page

- Mobile view of the digital menu after scanning QR code.
- Item selection and cart summary screen.



mamce Digital Menu

IDLI
50.00

Upma 40,
40.00

Rice Idli (3pes.)
80.00

Rice Idli Plain (3pcs.)
40.00

Dahi Idll (3pos.) §
0.00

Rava Idli (3pes.}
60.00

Rava Idli Plain (3pcs.)
50.00

Kanchi Puram (spl.} tdli (3pes.)
100.00

WN” ee

Plain Vada (2pcs.)
30.00

Sambar Vada (2pcs.) | fo”
40.00

Rasam Vada (2pcs.}
50.00

Dahi Vada (2pcs.)
70.00

Onion Vada (2pcs.)}
60.00

Mix Veg. Vada (2pes.)
70.00

Mysore Bonda (2pcs.}
70.00

Daal Vada(2pes.)
40.00

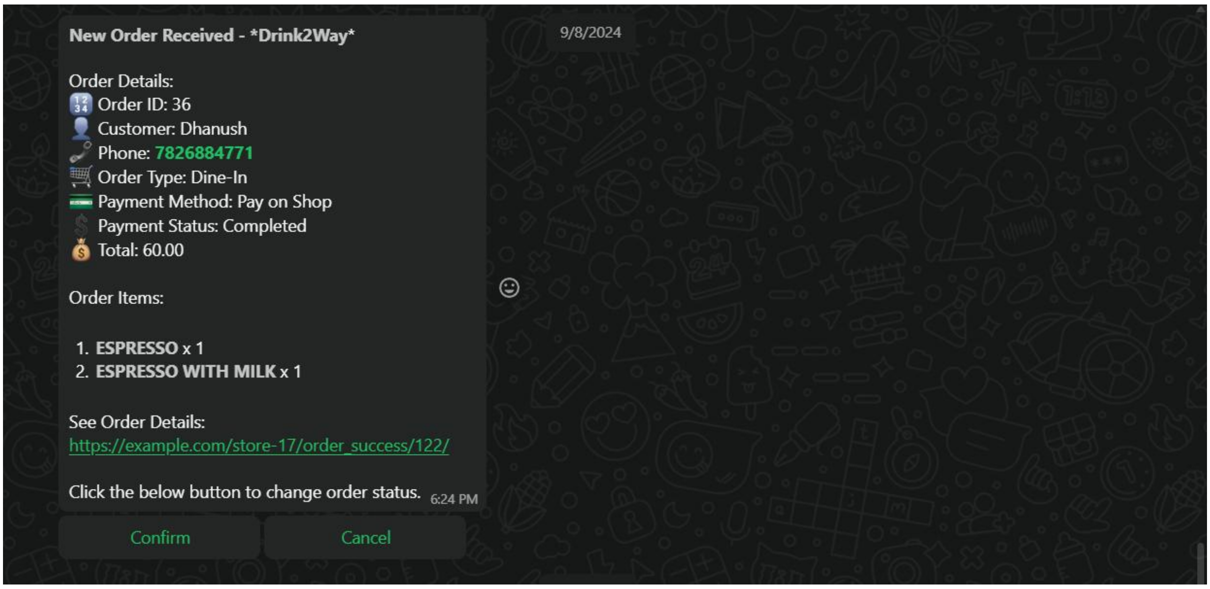
Plain Dosa
40.00

Masala Dosa
50.00

Butter Plain Dosa
60.00

WhatsApp Notification Preview

- Example of how the order is displayed in WhatsApp.



9/8/2024

New Order Received - *Drink2Way*

Order Details:
🔢 Order ID: 36
👤 Customer: Dhanush
📞 Phone: 7826884771
🛒 Order Type: Dine-In
💵 Payment Method: Pay on Shop
💲 Payment Status: Completed
💰 Total: 60.00

Order Items:

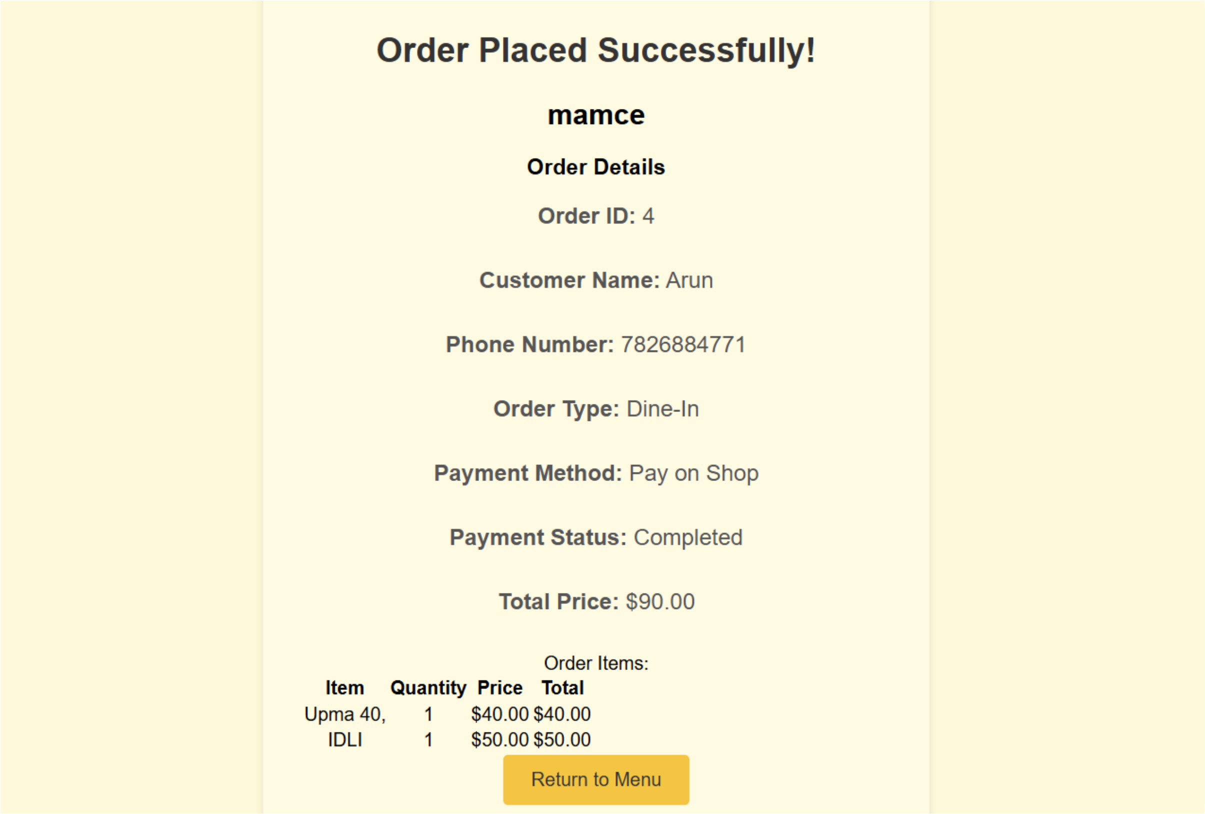
1. **ESPRESSO** x 1
2. **ESPRESSO WITH MILK** x 1

See Order Details:
https://example.com/store-17/order_success/122/

Click the below button to change order status. 6:24 PM

Payment Success Screen

- Confirmation screen shown to customer after payment.



Order Placed Successfully!

mamce

Order Details

Order ID: 4

Customer Name: Arun

Phone Number: 7826884771

Order Type: Dine-In

Payment Method: Pay on Shop

Payment Status: Completed

Total Price: $90.00

Order Items:

Item	Quantity	Price	Total
Upma 40,	1	$40.00	$40.00
IDLI	1	$50.00	$50.00

CHAPTER 5

Testing

5.1 Introduction to Testing

Testing is a critical phase in the software development lifecycle that ensures the final product meets its intended functionality, performs reliably, and is free of critical bugs. The purpose of testing is to identify and resolve issues before deployment so the system delivers a smooth user experience. For the "Smart Food Ordering System with OCR-Based Menu Digitization," testing was conducted at multiple levels to validate OCR accuracy, menu management, order placement, payment functionality, and WhatsApp notification delivery.

5.2 Testing Techniques

5.2.1 Unit Testing

- Individual components such as the OCR parsing logic, menu CRUD operations, order total calculations, and WhatsApp message formatting were tested independently.
- Tools: Python unittest module and Django’s built-in testing framework.

5.2.2 Integration Testing

- Focused on the flow between modules, such as OCR to menu creation, order placement to WhatsApp messaging, and payment redirection to database update.
- Ensured components worked together without data loss or logic errors.

5.2.3 Functional Testing

- Verified core user functions:
    - Restaurant registration
    - Menu upload and editing
    - Order placement through QR code interface
    - WhatsApp order notification

    - Digital payment integration

5.2.4 Usability Testing

- Conducted informally with a group of non-technical users.
- Participants tested QR scanning, browsing the menu, placing orders, and navigating the admin dashboard.
- Feedback was collected to improve the UI flow and error messages.

5.2.5 Regression Testing

- Whenever changes were made to improve OCR performance or integrate new features, previously working modules were retested to ensure no new bugs were introduced.

5.2.6 System Testing

- The entire system was tested end-to-end on both desktop and mobile environments to simulate real-life use in restaurants and customer-facing interfaces.

5.3 Test Cases and Results

Test Case 1: OCR Menu Upload

- **Input:** Uploaded a JPEG menu with readable font.
- **Expected Output:** Extracted menu items displayed in structured form.
- **Result:** PASS (minor formatting edits needed)

Test Case 2: Menu Editing

- **Input:** Manually edited extracted text.
- **Expected Output:** Updated items saved to database correctly.
- **Result:** PASS

Test Case 3: QR Code Menu Access

- **Input:** Scan QR code from printed flyer.
- **Expected Output:** Redirect to digital menu page.
- **Result:** PASS

Test Case 4: Order Placement

- **Input:** Customer selects items and submits order.
- **Expected Output:** Order saved in database and WhatsApp message sent.
- **Result:** PASS

Test Case 5: WhatsApp Notification

- **Input:** Order placed by customer.
- **Expected Output:** Order details appear correctly formatted in WhatsApp.
- **Result:** PASS (formatting adjustments applied)

Test Case 6: Payment Flow

- **Input:** Customer clicks "Pay Now" and completes payment via UPI.
- **Expected Output:** Transaction status saved and confirmation displayed.
- **Result:** PASS

Test Case 7: Admin Dashboard Access

- **Input:** Owner logs in and views dashboard.
- **Expected Output:** Dashboard loads with correct analytics and menu control.
- **Result:** PASS

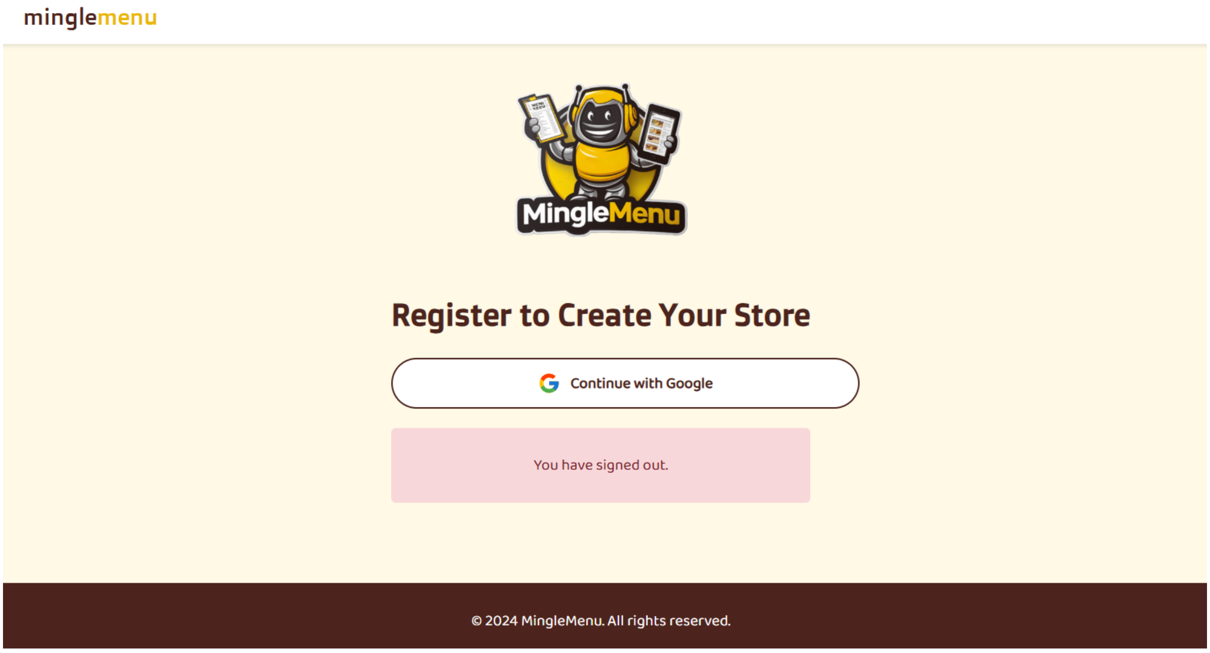
CHAPTER 6
Results and Discussion

6.1 Output Screens

To validate the functionality and usability of the "Smart Food Ordering System with OCR-Based Menu Digitization," several output screens were captured throughout the development and testing stages. These screens serve as visual proof of the system's capabilities and are used to support the documentation and project demonstration.

The following output screens are recommended for inclusion in the final report or presentation:

- **Figure 6.1: Admin Login Page** — Secure login interface for restaurant owners.

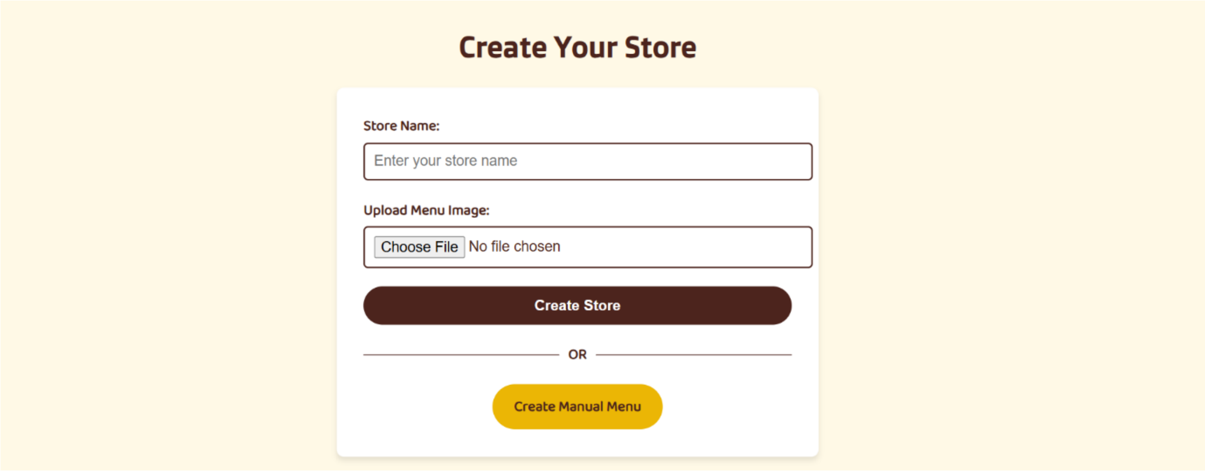


Register to Create Your Store

You have signed out.

© 2024 MingleMenu. All rights reserved.

- **Figure 6.2: Menu Upload Interface** — Upload section with OCR preview and manual correction option.



Create Your Store

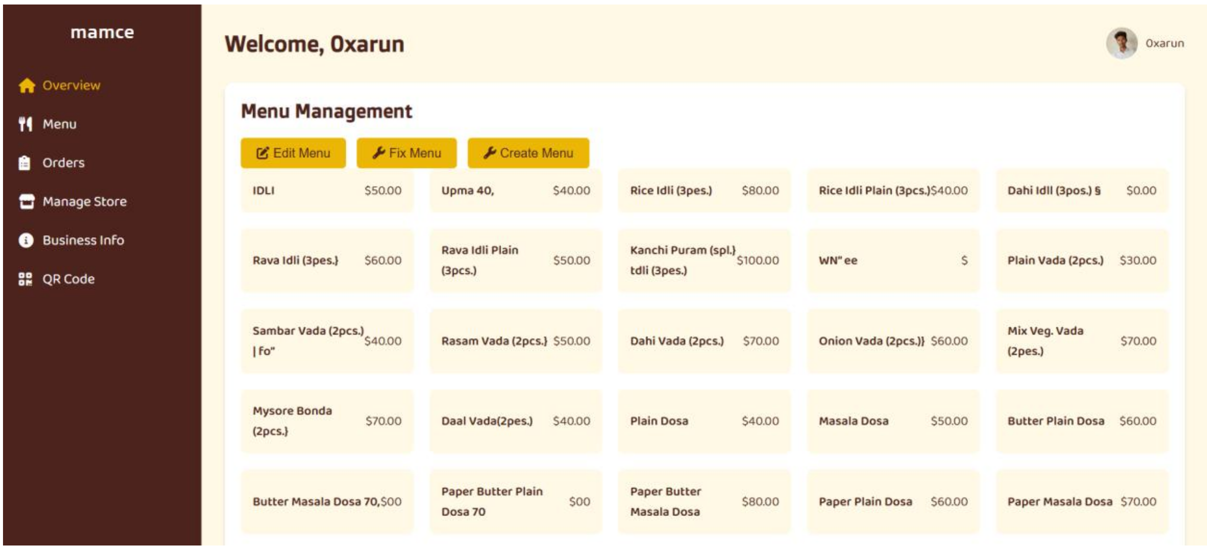
Store Name:

Enter your store name

Upload Menu Image:

Choose File No file chosen

- **Figure 6.3: Menu Management Dashboard** — Interface for adding, editing, or deleting menu items.

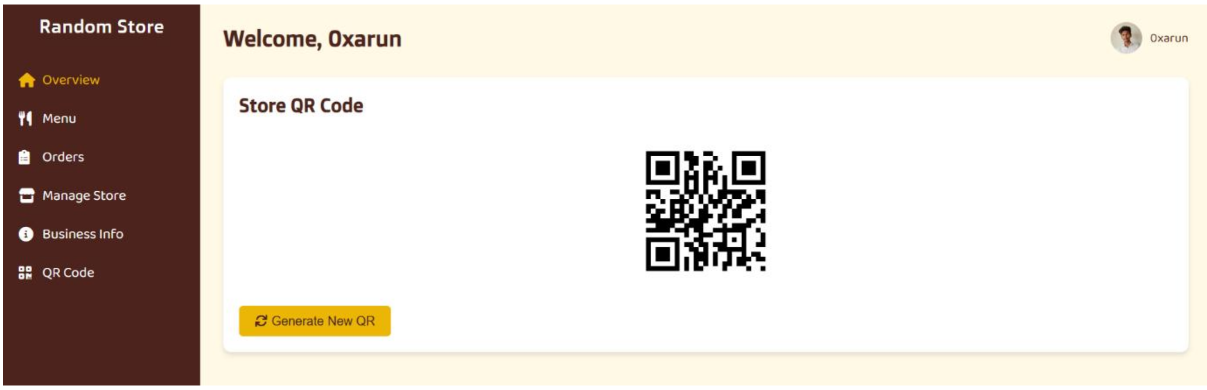


Welcome, Oxarun

Menu Management

IDLI $50.00	Upma 40, $40.00	Rice Idli (3pes.) $80.00	Rice Idli Plain (3pcs.)$40.00	Dahi Idll (3pos.) § $0.00
Rava Idli (3pes.} $60.00	Rava Idli Plain (3pcs.) $50.00	Kanchi Puram (spl.} tdli (3pes.) $100.00	WN" ee $	Plain Vada (2pcs.) $30.00
Sambar Vada (2pcs.) | Fo" $40.00	Rasam Vada (2pcs.} $50.00	Dahi Vada (2pcs.) $70.00	Onion Vada (2pcs.)} $60.00	Mix Veg. Vada (2pes.) $70.00
Mysore Bonda (2pcs.} $70.00	Daal Vada(2pes.) $40.00	Plain Dosa $40.00	Masala Dosa $50.00	Butter Plain Dosa $60.00
Butter Masala Dosa 70,$00	Paper Butter Plain Dosa 70 $00	Paper Butter Masala Dosa $80.00	Paper Plain Dosa $60.00	Paper Masala Dosa $70.00

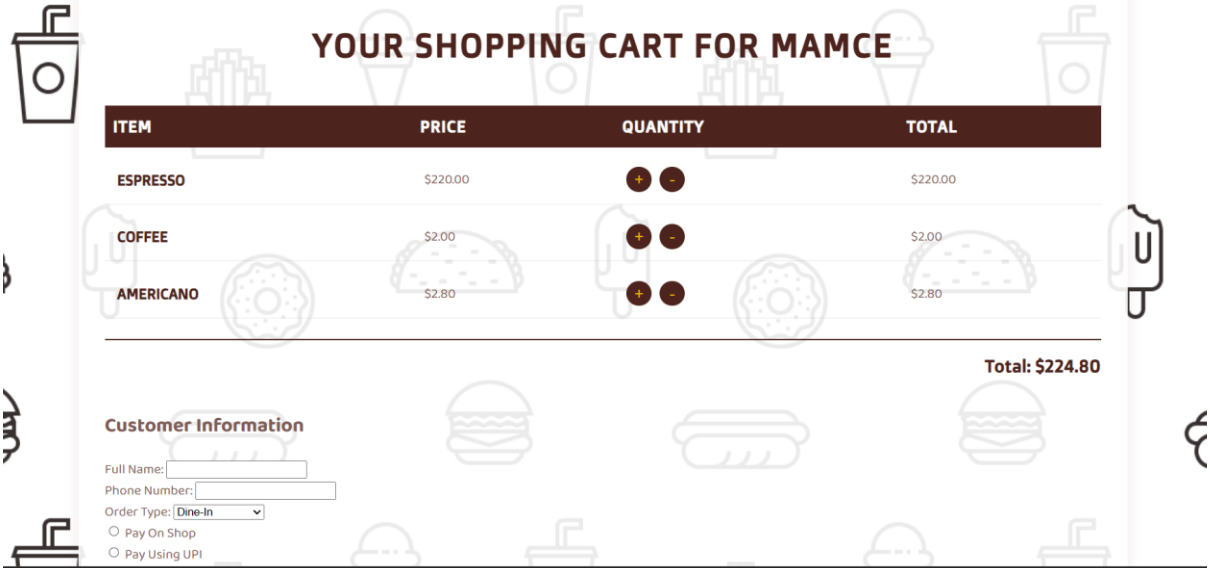
- **Figure 6.4: Customer QR Menu View** — Mobile-friendly digital menu loaded via QR code.



Welcome, Oxarun

Store QR Code

- **Figure 6.5: Cart and Order Placement Screen** — Selection of items and order summary.



YOUR SHOPPING CART FOR MAMCE

ITEM	PRICE	QUANTITY	TOTAL
ESPRESSO	$220.00	+ -	$220.00
COFFEE	$2.00	+ -	$2.00
AMERICANO	$2.80	+ -	$2.80

Total: $224.80

Customer Information

Full Name:

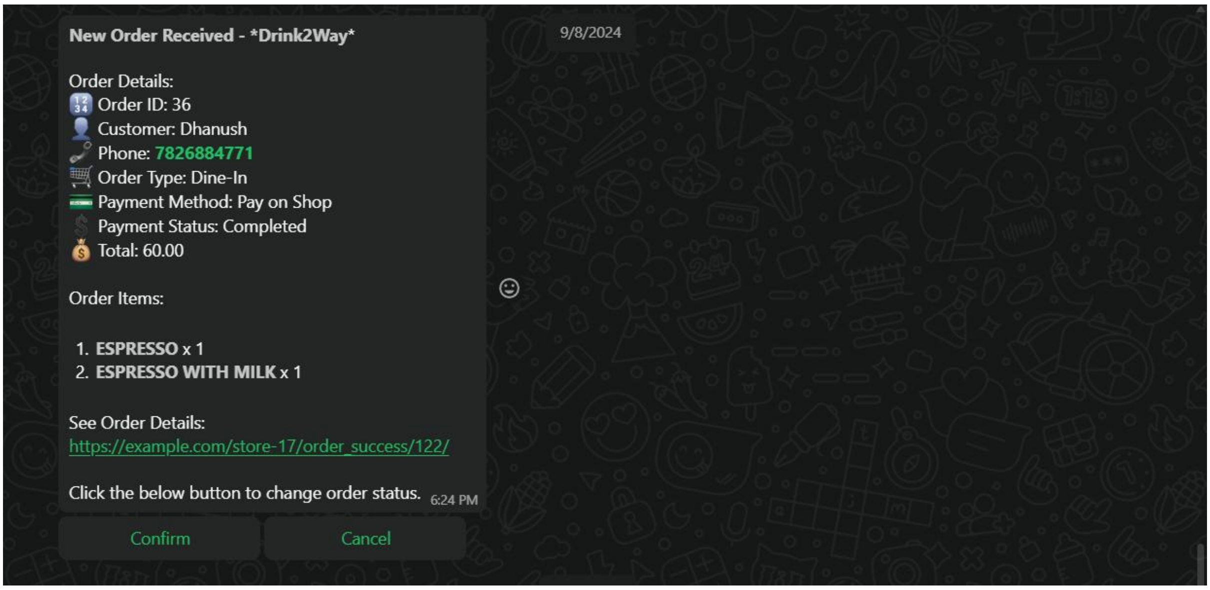
Phone Number:

Order Type: Dine-In

○ Pay On Shop

○ Pay Using UPI

- **Figure 6.6: WhatsApp Order Notification** — Order details received by the restaurant on WhatsApp.



9/8/2024

New Order Received - *Drink2Way*

Order Details:
🔢 Order ID: 36
👤 Customer: Dhanush
📞 Phone: 7826884771
🛒 Order Type: Dine-In
💵 Payment Method: Pay on Shop
💲 Payment Status: Completed
💰 Total: 60.00

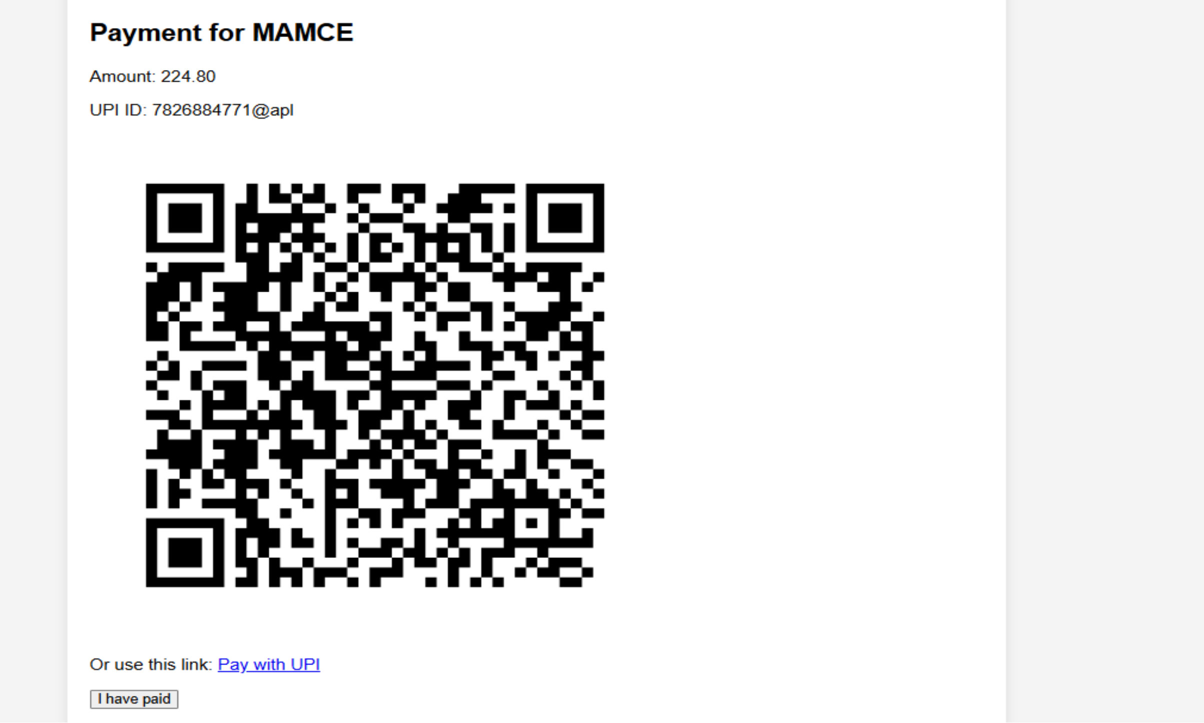
Order Items:

1. **ESPRESSO** x 1
2. **ESPRESSO WITH MILK** x 1

See Order Details:
https://example.com/store-17/order_success/122/

Click the below button to change order status. 6:24 PM

- **Figure 6.7: Payment QR / UPI Interface** — Payment gateway preview with QR code generation.



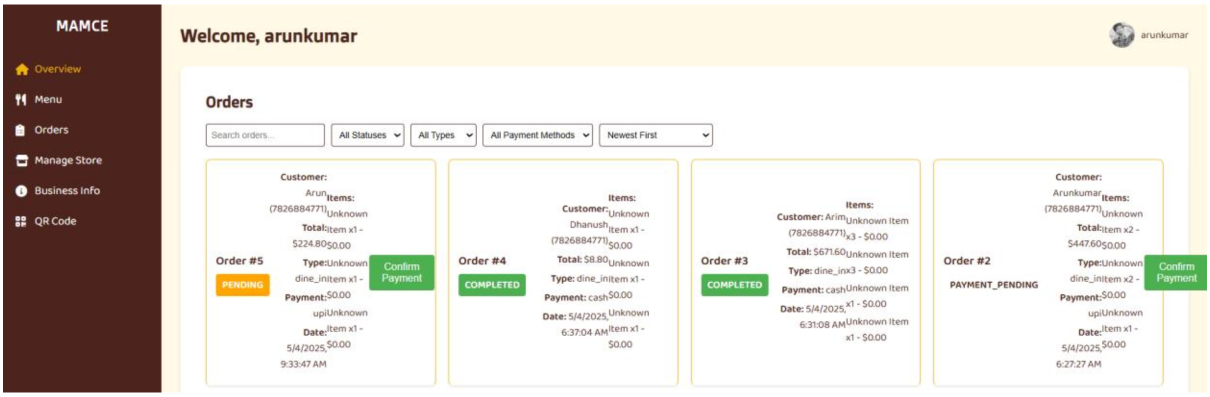
Payment for MAMCE

Amount: 224.80

UPI ID: 7826884771@apl

Or use this link: Pay with UPI

- **Figure 6.8: Admin Order Confirmation Screen** — Manual verification of payment and confirmation.



Welcome, arunkumar

Orders

Order #5 PENDING

Customer: Arun (7826884771)
Total: $224.80
Type: dine_in
Payment: upi
Date: 5/4/2025, 9:33:47 AM

Items:
Unknown Item x1 - $0.00
Unknown Item x1 - $0.00
Unknown Item x1 - $0.00

Order #4 COMPLETED

Customer: Dhanush (7826884771)
Total: $8.80
Type: dine_in
Payment: cash
Date: 5/4/2025, 6:37:04 AM

Items:
Unknown Item x1 - $0.00
Unknown Item x1 - $0.00
Unknown Item x1 - $0.00

Order #3 COMPLETED

Customer: Arim (7826884771)
Total: $671.60
Type: dine_in
Payment: cash
Date: 5/4/2025, 6:31:08 AM

Items:
Unknown Item x3 - $0.00
Unknown Item x3 - $0.00
Unknown Item x1 - $0.00
Unknown Item x1 - $0.00

Order #2 PAYMENT_PENDING

Customer: Arunkumar (7826884771)
Total: $447.60
Type: dine_in
Payment: upi
Date: 5/4/2025, 6:27:27 AM

Items:
Unknown Item x2 - $0.00
Unknown Item x2 - $0.00
Unknown Item x1 - $0.00

6.2 Performance Analysis

The system was tested across various modules to assess processing speed, stability, responsiveness, and accuracy. Overall, the platform performs efficiently under normal use conditions. However, a few performance observations were made:

- **OCR Engine Performance (Tesseract):**
    - Image-to-text conversion is generally accurate, but OCR results vary depending on image quality, alignment, and font type.
    - Larger menu images or poorly scanned documents take a few extra seconds to process.
    - Minor manual corrections are often needed to clean up extracted text.
- **WhatsApp API Integration:**
    - Messages are successfully sent to the restaurant owner's WhatsApp via BhashSMS API.
    - There is a short delay in API message delivery, typically under 5 seconds.
    - Since callback URL support is not available from the provider, two-way automation for payment confirmation is not possible.
- **UI Load Time:**
    - All pages load smoothly with minimal delay.
    - Background cover images (used for branding and aesthetics) slightly increase load time on slower internet connections.
- **Ordering & Database Sync:**
    - Orders are correctly stored in the database in real time.
    - Menu updates reflect instantly across the customer-facing interface.

In summary, except for some latency in OCR and WhatsApp API, the application performs well, offering a fluid experience for both restaurant owners and customers.

6.3 Challenges Faced

During development and deployment, several technical and integration challenges were encountered. These were addressed through debugging, redesigning certain components, and making practical compromises.

- **WhatsApp API Template Formatting:**
    - The structure of the message and parameters for the WhatsApp template required careful customization.
    - Items list formatting had to be optimized to fit WhatsApp's display constraints.
- **Lack of Callback Support from WhatsApp Provider (BhashSMS):**
    - Real-time status updates and button click callbacks (e.g., payment confirmations) were not supported.
    - As a workaround, the system was designed so the owner manually verifies the payment and confirms it.
- **OCR Accuracy and Text Cleanup:**
    - Tesseract OCR performed well for clean images but required user validation for noisy scans.
    - Custom logic was implemented to help clean up line breaks and misread characters.
- **Payment Gateway Simplification:**
    - Instead of integrating complex gateways like Razorpay with backend callbacks, a lightweight solution was implemented.
    - The system generates a UPI payment link or static QR code for customer scanning.

    - Owners confirm payment manually on the dashboard or via WhatsApp follow-up.

- **Mobile Optimization:**
    - Ensuring consistent performance across mobile browsers took additional time.
    - UI adjustments were made to support responsive layouts and touch-friendly interactions.

Despite these challenges, the system successfully delivers on its core promise — a simple, one-click digital food ordering platform for SMB restaurant owners, complete with OCR menu digitization, WhatsApp notifications, and digital payments.

CHAPTER 7

Conclusion and Future Enhancement

7.1 Summary of Work

The project titled **"A Smart Food Ordering System with OCR-Based Menu Digitization"** was developed with the aim of providing small and medium-sized restaurants with a simplified, cost-effective digital ordering platform. The system allows restaurant owners to digitize their physical menus using Optical Character Recognition (OCR), host them online, and share them with customers via QR codes. Orders are placed through a mobile-friendly web interface and received by the restaurant via WhatsApp using the WhatsApp Business API.

The following major components summarize the work completed:

- Developed a fully functional web-based platform using Django, HTML, CSS, and JavaScript.
- Integrated Tesseract OCR to convert scanned menu images into editable digital text.
- Enabled menu upload and management through a secure admin dashboard.
- Implemented QR code generation for instant customer access to the restaurant's menu.
- Built a seamless order placement process with WhatsApp notification delivery.
- Provided a simple UPI-based payment process using QR codes or payment links.
- Tested all modules thoroughly with multiple testing techniques to ensure functionality and performance.

Throughout the development lifecycle, the project was structured into distinct modules: menu digitization, menu management, customer ordering, WhatsApp integration, and digital payments. Each module was carefully designed, implemented, and tested to ensure seamless functionality and user experience. Testing techniques such as unit testing, integration testing, and usability testing were applied to validate the system's performance. Additionally, the application was built with open-source tools such as Django, Tesseract OCR, and standard web technologies to ensure ease of deployment and scalability.

The final product is a standalone, lightweight solution that enables even non-technical restaurant owners to digitize their food ordering process within minutes.

7.2 Conclusion

In conclusion, the Smart Food Ordering System successfully meets its core objective of digitizing and modernizing the food ordering experience for SMBs. It bridges the digital gap that prevents many local restaurants from establishing an online presence, due to high costs or technical barriers. By using OCR, the system removes the manual workload involved in digitizing menu content. QR code access allows customers to browse and order without downloading any application, enhancing convenience and accessibility.

Moreover, the integration of WhatsApp Business API makes the system unique and practical. Restaurant owners receive orders directly through WhatsApp, a platform they are already familiar with, eliminating the learning curve and reducing the need for additional tools. Though limited by the lack of callback support from the API provider, the system compensates with manual confirmation features that are both simple and effective.

The simplicity of the payment module using UPI and QR ensures low-cost transaction handling without relying on complex payment gateways. Overall, the system delivers a strong, user-focused solution and stands as a reliable digital companion for restaurant businesses seeking modern infrastructure with minimal overhead.

7.3 Future Scope and Enhancements

Although the current system fulfills its intended functionality, there are several potential enhancements that can be pursued in the future to expand its capability and usability:

1. **Callback API Integration:** With support from a better WhatsApp provider or an alternative messaging platform, real-time status updates and automated confirmations can be implemented.

2. **Real-Time Analytics:** Adding advanced analytics to the admin dashboard could help restaurant owners track customer behavior, peak ordering times, and most-ordered items.
3. **Inventory Management:** The system could be extended to automatically update stock levels based on orders, alerting the owner when items are running low.
4. **Customer App or PWA:** A lightweight Progressive Web App (PWA) could enhance user experience, enabling customers to receive notifications or save favorites.
5. **Multi-Language Support:** Adding support for multiple languages would increase accessibility for both restaurant owners and customers.

REFERENCES

1. Zomato digitizes menus using Amazon Textract and Amazon SageMaker. https://aws.amazon.com/blogs/machine-learning/zomato-digitizes-menus-using-amazon-textract-and-amazon-sagemaker
2. QR Based Online Food Ordering System, SSGMCE. https://www.ssgmce.ac.in/uploads/UG_Projects/it/Gr%20No-17-Project-Report.pdf
3. IRJMETS - Online Food Ordering System Using QR Code. https://www.irjmets.com/uploadedfiles/paper//issue_4_april_2025/72014/final/fin_irjmets1744221837.pdf
4. Indian Retailer. Restaurants in India Focus on Digital-Friendly Menu. https://www.indianretailer.com/restaurant/article/restaurants-in-india-focus-on-digital-friendly-menu.13823
5. Koay, K. Y., & Ang, M. C. (2024). Consumer intentions to use QR code menus post-COVID-19. Journal of Retailing and Consumer Services. https://doi.org/10.1016/j.jretconser.2023.103528
6. Wong, T. W., Ng, C. H., & Yuen, K. F. (2023). Smart Restaurant Systems using QR-based food ordering. International Journal of Hospitality Management. https://doi.org/10.1016/j.ijhm.2022.103123

7. Krishnan, A., & Likhitha, S. QR Code Based Food Ordering System. IJERT. https://www.ijert.org/research/qr-code-based-food-ordering-system-IJERTV9IS050369.pdf

8. Google Developers. Tesseract OCR Documentation. https://tesseract-ocr.github.io/tessdoc/

9. Django Documentation – The Web framework for perfectionists with deadlines. https://docs.djangoproject.com/ Zomato digitizes menus using Amazon Textract and Amazon SageMaker. https://aws.amazon.com/blogs/machine-learning/zomato-digitizes-menus-using-amazon-textract-and-amazon-sagemaker

10. QR Based Online Food Ordering System, SSGMCE. https://www.ssgmce.ac.in/uploads/UG_Projects/it/Gr%20No-17-Project-Report.pdf

11. IRJMETS - Online Food Ordering System Using QR Code. https://www.irjmets.com/uploadedfiles/paper//issue_4_april_2025/72014/final/fin_irjmets1744221837.pdf

12. Indian Retailer. Restaurants in India Focus on Digital-Friendly Menu. https://www.indianretailer.com/restaurant/article/restaurants-in-india-focus-on-digital-friendly-menu.13823