Restaurant Management System

Submitted By

Shahjada Moon Shimul

Student Id: 191-35-434

Supervised By

Mr. Md Rajib Mia

Designation: Lecturer (Senior Scale)

This project report has been submitted in fulfilment of the requirements for the degree of **Bachelor of Science in Software Engineering**

@ All right Reserved by Daffodil International University

APPROVAL

This project titled on "**Restaurant Management System**", submitted by **Shahjada Moon (ID: 191-35-434)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Chairman

Dr. Imran Mahmud
Associate Professor & Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1

Nuruzzaman Faruqi
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

Md. Rajib Mia
Lecturer (Senior Sacle)
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

External Examiner

Md. Fazle Munim
Associate Director & Vice President
Government & Public Sector
Ernst & young (EY)

DAFFODIL INTERNATIONAL UNIVERSITY

DECLARATION OF THESIS AND COPYRIGHT

I declare that this thesis is classified as:

I acknowledge that Daffodil International University reserves the following rights:

1. The Project is the Property of Daffodil International University.
2. The Library of Daffodil International University has the right to make copies of the Project for the purpose of research only.
3. The Library of Daffodil International University has the right to make copies of the Project for academic exchange.

Certified by:

Shahjada Noon
(Student's Signature)

(Supervisor's Signature)

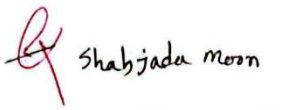
191-35-434
Student ID
Date:

mr. md Rajib mia.
Name of Supervisor
Date:

NOTE: * If the Project is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

STUDENT'S DECLARATION

I hereby declare that the work in this project is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Daffodil International University or any other institution.

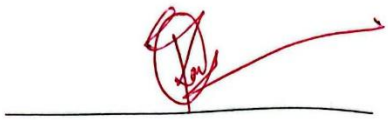


(Student's Signature)

Full Name : **Shahjada Moon Shimul**
ID Number : 191-35-434
Date : 31 December 2024

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this project, and in my opinion, this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Science.



(Supervisor's Signature)

Full Name : **Mr. Md Rajib Mia**
Position : Lecturer (Senior Scale)
Date : 31 December 2024

Restaurant Management System

Shahjada Moon Shimul

Project submitted in fulfillment of the requirements for
the award of the degree of
Bachelor of Science/Master of Science

Department of Software Engineering

DAFFODIL INTERNATIONAL UNIVERSITY

31 DECEMBER 2024

ACKNOWLEDGEMENTS

First of all, we are grateful to The Almighty Allah for making us eligible to complete this thesis.

Then we would like to thank our supervisor **"MD Rajib Mia", "Lecturer (Sr. Scale)",** Department of Software Engineering. We are extremely grateful and indebted to her for hers expert, sincere and valuable guidance and encouragement extended to us.

I would also like to thank them who were involved in the survey for this thesis project. Without their passionate participation and input, the validation survey could not have been successfully conducted.

We take this opportunity to record our sincere thanks to all the faculty members of the Department of Software Engineering for their help and encouragement.

Last but not least, we would like to thank to our parents, for their unconditional support, love and without this we would not have come this far.

DEDICATION

I therefore declare that I have done this project under the oversight of "**Mr. Md Rajib Mia**", **"Lecturer (Senior Scale)"**, Department of Software Engineering, Daffodil International University. Also declare that neither entire record nor any portion of this record has been submitted somewhere else for my degree.

ABSTRACT

A Restaurant Management System is software designed to streamline restaurant operations, such as menu management, order handling, billing, inventory tracking, and customer management. It typically includes features for staff management, table reservations, and reporting.

The system can be built using a frontend framework like React.js or Flutter, a backend like Django or Node.js, and a database like MySQL. It supports multi-platform use (web, mobile, and tablet) and integrates tools for payments, SMS, and email.

Development involves phases like requirement analysis, design, coding, testing, deployment, and maintenance, ensuring the system meets the restaurant's needs efficiently.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

RMS	Restaurant Management System
CRM	Customer Relationship Management
POS	Point of Sale
UML	Unified Modelling Language

CHAPTER 1 INTRODUCTION

1.1 Background

The Restaurant Management System project aims to modernize and streamline restaurant operations by automating tasks like menu management, order processing, billing, and inventory tracking. It addresses the inefficiencies of traditional manual methods, enhancing accuracy and customer satisfaction. The system includes features such as table reservations, staff management, and analytics, with multi-platform support for web and mobile. By integrating payment gateways and real-time notifications, this solution provides a seamless, efficient experience for restaurant owners, staff, and customers, meeting the evolving demands of the hospitality industry.

1.1.1 Context and Relevance

- The Restaurant Management System addresses challenges in the fast-evolving hospitality industry, where efficiency, accuracy, and personalized experiences are key. As trends like online ordering and contactless payments grow, many restaurants still struggle with operational inefficiencies and inventory issues. This project taps into opportunities for innovation by providing a modern, automated solution to operations and streamline, costs and reduce, and customer satisfaction enhance.

1.1.2 Problem Identification

- Many restaurants still rely on manual processes, leading to inefficiencies in order handling, billing, and inventory management. Existing digital solutions are often fragmented or expensive, making them unsuitable for small and medium-sized businesses. This project aims to fill this gap by offering an integrated, user-friendly system that automates key tasks, improving efficiency and reducing errors.

1.1.3 Purpose and Justification

This project is important because it addresses key challenges in the restaurant industry, such as inefficiency, errors, and high operational costs. By automating tasks like order processing, billing, and inventory management, it improves accuracy, saves time, and enhances customer satisfaction. The system provides a comprehensive, integrated solution that is easy to use and affordable, especially for small and medium-sized businesses. It adds value by streamlining

operations, enabling data-driven decision-making, and helping restaurants stay competitive in a rapidly evolving market.

1.1.4 Scope

- **Operational Scope:** Implementing better workflow systems, using technology (like POS systems), and training staff on efficient processes. Automation in inventory management and ordering systems can also reduce operational bottlenecks.
- **Customer Service Scope:** Staff training and development programs focused on customer service skills, creating a service standard protocol, and feedback mechanisms to address customer complaints. Providing personalized service can improve loyalty.
- **Food Quality and Consistency Scope:** Standardizing recipes, improving quality control measures, and conducting regular staff training on food preparation. Monitoring supplier quality and ensuring fresh, high-quality ingredients are used consistently.
- **Technology Scope**: Implementing modern technology solutions, such as mobile ordering apps, reservation systems, and integrated POS systems. Leveraging data analytics can help understand customer preferences, optimize operations, and improve decision-making.
- **Cost Control and Financial Management Scope:** Implementing better inventory management practices, reducing food waste through portion control, and monitoring labour costs by optimizing staff schedules. Conducting regular financial analysis can help identify inefficiencies and opportunities for improvement.
- **Health and Safety Scope:** Ensuring rigorous compliance with food safety protocols and conducting regular staff training on hygiene practices. Introducing routine health checks and audits will help maintain high standards.
- **Customer Feedback and Relationship Management Scope:** Establishing systems for gathering, analysing, and responding to customer feedback, such as surveys, online reviews, or loyalty apps. Personalizing follow-ups with customers and resolving complaints quickly can enhance the dining experience.

1.2 Planning and Starting a Project

Step-by-Step Feasibility Study

A feasibility study for the Restaurant Management System project evaluates its practicality across several dimensions to ensure it is viable, efficient, and beneficial for stakeholders. The steps involved are:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility
- Legal Feasibility
- Schedule Feasibility

Phase 1 Preliminary Analysis & Project Scope Definition: The Restaurant Management System (RMS) is designed to automate and streamline key restaurant operations, such as order management, billing, inventory control, and customer relationships. The system aims to improve efficiency, reduce errors, and enhance the overall customer experience for small and medium-sized restaurants.

Stakeholders:

- **Restaurant Owners/Managers** – Primary users who will manage the system, monitor performance, and make business decisions.
- **Staff (Servers, Chefs, Cashiers)** – Users who will interact with the system for order management, billing, and inventory tracking.
- **Customers** – Indirect users who benefit from faster service and more personalized experiences.
- **Suppliers** – Indirect stakeholders who will be affected by inventory management features.
- **Development Team** – Responsible for designing, coding, and deploying the system.
- **Investors/Business Partners** – May be involved in the funding and strategic

Key Functional Areas:

- **Menu Management** – Add, edit, and categorize menu items.
- **Order Management** – Capture orders, manage table assignments, and track order status.
- **Billing System** – Generate invoices with payment integration.
- **Inventory Management** – Track stock levels and manage supplier orders.
- **Staff Management** – Monitor work schedules, performance, and roles.
- **Customer Relationship Management** – Track customer preferences and loyalty programs.
- **Reporting & Analytics** – Sales reports, inventory reports, and customer insights.

Non-Functional Requirements:

- **Performance** – System must handle multiple simultaneous users without lag.
- **Usability** – User-friendly interfaces for restaurant staff with minimal training.
- **Security** – Secure user authentication and data encryption for payment processing.
- **Scalability** – Ability to scale as the restaurant grows or expands to multiple locations.
- **Availability** – High system availability, with minimal downtime.

Project Phases:

- **Planning & Requirements Analysis** – Define system requirements and project scope.
- **Design** – Create system architecture and UI/UX design.
- **Development** – Build the system features and backend.
- **Testing** – Conduct unit, integration, and user acceptance testing
- **Deployment** – Deploy the system to the live environment.
- **Maintenance** – Provide ongoing support and feature updates.

Potential Risks:

- **Technical Challenges** – Issues with system integration or scalability.
- **Budget Overruns** – Costs exceeding initial estimates due to unforeseen complications.
- **User Adoption** – Resistance to using new technology from restaurant staff.
- **Data Security** – Risks related to handling sensitive customer and payment data.

Phase 2 Market Feasibility Analysis (or Market Research):

The Restaurant Management System project aims to address the growing demand for technology solutions in the food service industry, which is evolving rapidly due to increasing customer expectations for faster service, personalization, and convenience. As the restaurant industry becomes more competitive, automation and integration of operations have become essential to stay ahead. The RMS is designed to cater to this need, offering an efficient, cost-effective solution to small and medium-sized restaurants that face challenges in managing their daily operations.

Market Overview: The globally operating software for restaurant management market is experiencing substantial expansion due to the growing popularity of technological advances. With restaurants looking to streamline operations, enhance customer experiences, and reduce operational costs, the demand for comprehensive restaurant management systems has risen sharply. This market includes not only large-scale enterprises but also a substantial number of small and medium-sized restaurants that require affordable and scalable solutions.

Target Market: The primary target market for the RMS includes small and medium-sized restaurants, cafes, and food outlets. These establishments often face resource constraints and are looking for affordable, easy-to-use solutions that can automate manual processes and improve operational efficiency. Secondary markets include fast-food chains, delivery-focused restaurants, and franchises that require centralized control over multiple locations.

Phase 3 Technical Feasibility Analysis:

The Restaurant Management System will integrate seamlessly with existing Point of Sale systems to ensure smooth transactions, order processing, and real-time inventory updates. POS integration is crucial for automating the billing process and linking orders to the inventory. Compatibility with popular POS platforms will be a key consideration. This integration will streamline operations, ensuring quick billing and real-time updates to inventory levels as items are sold.

Inventory and Supply Chain Management: The RMS will feature an inventory management module that tracks stock levels, orders, and supplier relationships. It will provide automated alerts when stock is low and suggest reordering based on usage patterns. The system will integrate with suppliers' systems (if available) to enable seamless inventory replenishment. Real-time monitoring of stock, waste management, and vendor management will optimize the supply chain and help restaurants reduce waste and control costs.

Customer Relationship Management (CRM): The CRM module of the RMS will track customer preferences, order history, and feedback. It will allow restaurants to offer personalized services and loyalty programs, improving customer retention. The CRM will also support targeted marketing campaigns via email or SMS, enabling promotions and special offers to be sent to specific customer segments. This will enhance customer engagement and increase repeat business.

Cloud and Mobile Access: The system will be designed with cloud-based architecture to provide flexibility, scalability, and real-time data access. Cloud hosting ensures that data is securely stored and easily accessible from multiple devices. Staff can access the system from mobile devices or tablets, whether they are on the floor taking orders or in the back office managing inventory. This mobile-first approach increases operational efficiency and ensures that key restaurant functions can be managed on the go.

Development Tools and Environment: The RMS will be developed using the following tools and technologies are Frontend, Backend, Database, Mobile Development, Cloud Hosting.

Phase 4 Financial Feasibility Analysis:

A Financial Feasibility Analysis evaluates the costs, expected benefits, and financial risks associated with implementing the Restaurant Management System (RMS). This analysis ensures the project is financially viable and will provide a return on investment

- **Initial Costs:**
- **Software Development**: The cost to develop or purchase the RMS, including custom coding, third-party tools, and integration with POS systems.

- **Hardware**: Expenses for POS terminals, tablets, servers, and networking infrastructure.
- **Licensing Fees**: Fees for any third-party software, such as POS platforms, payment processors, or cloud hosting services.
- **Training**: Costs for training staff on how to use the system effectively.
- **Cost Savings and Benefits:**
- **Increased Efficiency**: Automation of tasks like billing, inventory tracking, and order management leads to reduced labour costs and operational errors.
- **Reduced Inventory Loss**: Real-time inventory tracking minimizes stock wastage and ensures timely restocking, reducing unnecessary purchases.
- **Faster Transactions**: Integration with POS systems speeds up transactions, improving customer satisfaction and potentially increasing sales volume.
- **Better Data Insights**: The system provides valuable insights into sales trends, allowing for more informed decisions on menu pricing, inventory management, and promotions.
- **Risk Assessment:**
- **Implementation Delays**: Unexpected technical issues or delays in system integration can extend the time before the system starts generating benefits.
- **Training Costs**: Unanticipated costs for training employees or addressing resistance to change.
- **Unexpected Operational Costs**: Unforeseen expenses in hardware, software, or maintenance.

1.3 Target User Profile and Tentative Elicitation Process

1.3.1 Target User

- **Restaurant Owners/Managers**: Oversee overall operations, manage staff, monitor sales, and ensure that the restaurant is running smoothly.
  Track performance, inventory, and financial data to make informed decisions.
- **Chefs/Cooks**: Use the system to view orders in real-time and manage the kitchen's workflow. Ensure timely preparation of dishes and monitor ingredient stock levels.
- **Waitstaff/Servers**: Input orders directly into the system to reduce errors. Manage customer requests, check on food status, and communicate with the kitchen. View real-time updates on customer orders and payment status.
- **Cashiers/Payment Processors**: Use the system to process bills, handle payments, and manage invoices. Ensure accurate and smooth payment transactions for customers.
- **Customers**: Some RMS platforms allow customers to place orders via kiosks, mobile apps, or websites. Can view the menu, order food, and pay bills from their own devices, enhancing convenience.

- **Accounting and Financial Staff**: Use RMS for generating financial reports, tracking sales, and analysing revenue trends. Handle payroll, expenses, and overall financial management.
- **Inventory and Supply Managers**: Monitor stock levels, track ingredient usage, and reorder supplies automatically when low stock is detected. Manage food cost and optimize inventory based on sales data.

1.3.2 User profile

Table 0: User Profile

User Class	**Note on Characteristics**
Category of client	Restaurant staff (waiters, chefs, managers, cashiers)
Age range	18–50 years
Frequency of use	Daily (multiple times during shifts)
Mandatory	Yes (required for managing orders, inventory, and billing)
Computer experience	Basic to intermediate (familiar with POS systems and smartphones)
Education	High school or college level (some may have hospitality-related qualifications)
goal	Efficiently manage orders, billing, and inventory, improving overall restaurant operations
Language skills	Typically fluent in the local language (may also need basic English for system interaction)
Number of users	5–30 users per restaurant
Training	Required initial training for all staff members (may need refresher training periodically)

Others system use	POS systems, inventory management software, mobile apps for communication
Way of working	Fast-paced, multitasking, collaborative with real-time updates and communication among staff

1.3.3 Elicitation Process

- **Interviews:** One-on-one or group discussions with key stakeholders, such as restaurant managers, waitstaff, chefs, and cashiers. Interviews help understand the specific tasks, pain points, and expectations of users regarding the system. Prepare a set of open-ended questions about existing processes, challenges, and desired system features. Conduct face-to-face, phone, or virtual interviews depending on the users' availability.
- **Surveys/Questionnaires:** Structured forms that users fill out, often distributed electronically or on paper. Surveys are useful for gathering feedback from a larger sample of restaurant staff to identify common requirements and preferences.
- **Focus Groups:** Group discussions with a small set of users (waiters, kitchen staff, and managers) facilitated by a moderator. This method helps identify group consensus and prioritize system features by involving different perspectives from various roles.
- **Observations:** Directly observing restaurant staff as they perform their day-to-day tasks to understand how they interact with existing systems and identify inefficiencies or unmet needs.
- **Use Case Analysis:** Analysing the typical tasks performed by different users in the restaurant to understand the system's functionality requirements. Involves creating scenarios that describe how users will interact with the RMS during daily operations.
- **Prototyping:** Creating a basic version (prototype) of the RMS and getting user feedback on its design and functionality. This method allows users to visualize and interact with the system early in the development process.
- **Document Analysis:** Reviewing existing documents, such as restaurant operations manuals, POS system documentation, and reports, to identify current workflows and areas for improvement. Provides insights into the current system's limitations.

1.4 Project Block Diagram

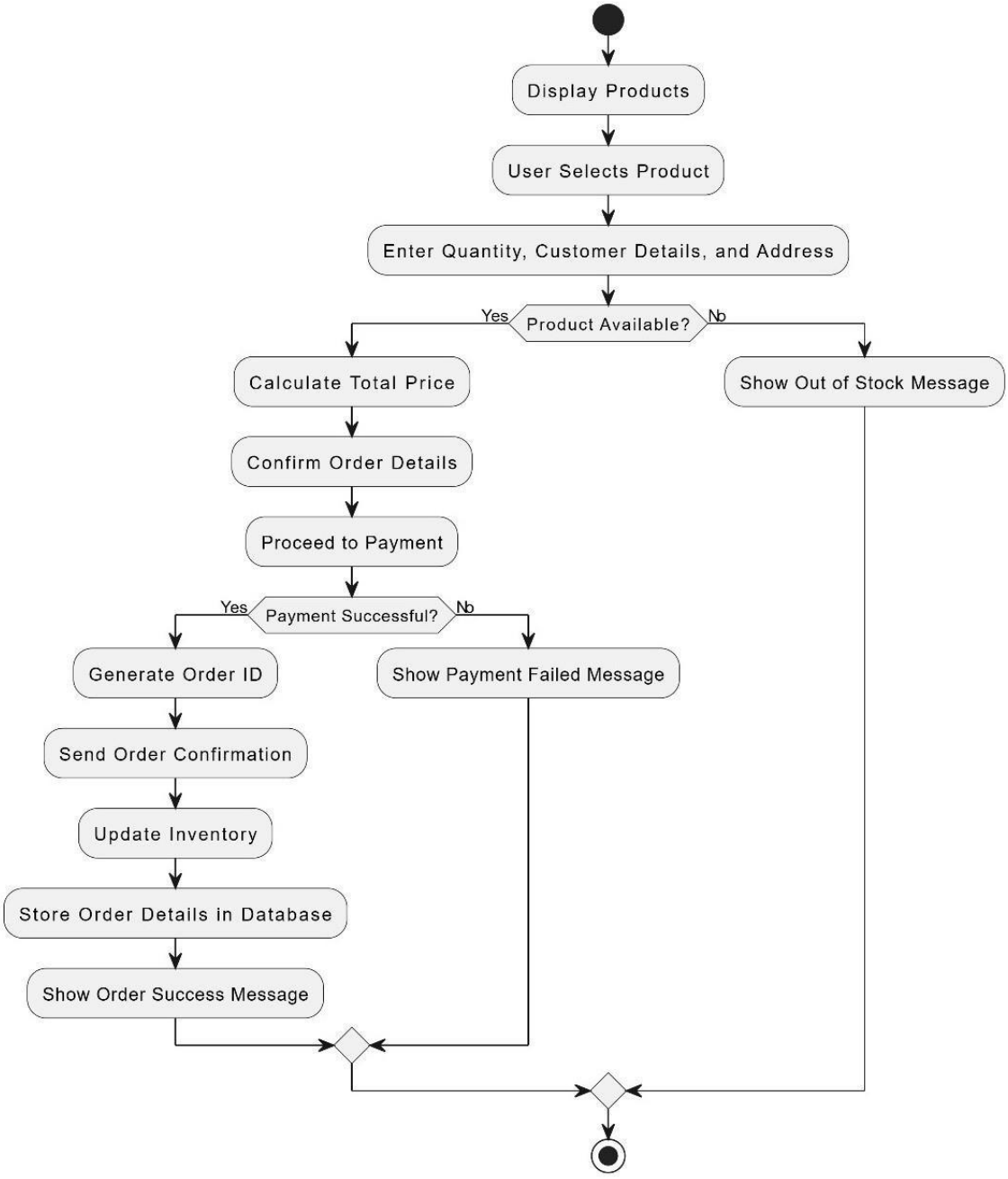


Figure 1: System Block Diagram

1.5 System prerequisite

1.5.1 Hardware prerequisite

Hardware Component	Requirements
Servers	-Processor: Intel Xeon / AMD EPYC -RAM: 8 GB or more -Storage: 250 GB SSD or higher - Network: 10 Mbps or higher internet
POS Terminals	-Processor: Intel i3 or higher -RAM: 4 GB or more -Storage: 128 GB SSD or higher - Touchscreen: 15" or larger
Kitchen Display System	-Processor: Intel i3 or higher -RAM: 4 GB or more - Display: 10-15inch, high resolution
Receipt Printers	- Thermal receipt printers for speed and low maintenance
Kitchen Printers	- Thermal or impact printers depending on kitchen conditions
Networking Equipment	-Routers: High-speed, dual-band -Wi-Fi Access Points for POS and customer devices
Mobile Devices/Tablets	-iOS/Android devices -Processor: 2.0 GHz or higher -RAM: 2-4 GB -Storage: 32 GB or higher

Security Hardware	-CCTV Cameras -Access Control Systems (for restricted areas)
Operating System	-Windows or Linux-based systems for servers and POS devices

1.5.2 Software Requirements

Software Component	Key Requirements
POS System	-Order processing, payment handling -Receipt printing, order routing to kitchen
Inventory Management	-Real-time stock tracking -Supplier and purchase order management -Integration with POS
Employee Scheduling	- Shift scheduling - Time tracking and payroll integration - Employee attendance monitoring
Mobile App (optional)	- For staff and customer use (order taking, payments, loyalty program)
Online Ordering	- Web and app integration - Menu display, payment processing, order tracking
Backup & Recovery	- Automated backups - Cloud-based or local storage
Cloud Hosting	- Scalable resources, secure data, high uptime guarantee

1.5.3 Constraints and Dependencies

Constraints:

Category	Description
Budget	Limited financial resources for software, hardware, and implementation costs.
Scalability	Need for the system to scale with growing operations, including multiple locations or increased traffic.
Integration	Compatibility with existing systems (POS, accounting, HR software) or third-party services.
Compliance	Adherence to regulations like PCI-DSS for payment security, GDPR for data protection, etc.
User Training	Staff must be trained to use the system effectively, impacting adoption and productivity.
Data Storage	Sufficient data storage capacity for transaction, customer, and inventory data.
Network Infrastructure	Requires stable internet connection, especially for cloud-based solutions or remote access.

Security	Ensuring data protection and cybersecurity, especially for customer payment and personal information.
System Downtime	Avoiding system downtime during peak hours or critical operation times.

Dependencies:

Category	Description
POS System Integration	RMS depends on seamless integration with the POS system for real-time order processing and payments.
Hardware	Functionality of the system relies on appropriate hardware (e.g., terminals, printers, servers).
Payment Gateway	Online ordering, POS, and mobile payments depend on third-party payment gateways for processing.
Internet Connectivity	A stable internet connection is essential for cloud-based systems, online ordering, and data backups.
Staff and Management	Dependence on employee involvement for system usage, training, and proper data entry.

Inventory Systems	Accurate inventory management depends on integration with stock tracking and supplier systems.
Third-Party Integrations	The system may depend on integrations with external services like delivery platforms, marketing tools, or accounting software.
Security Protocols	Payment processing and data storage depend on secure encryption methods and compliance with standards.

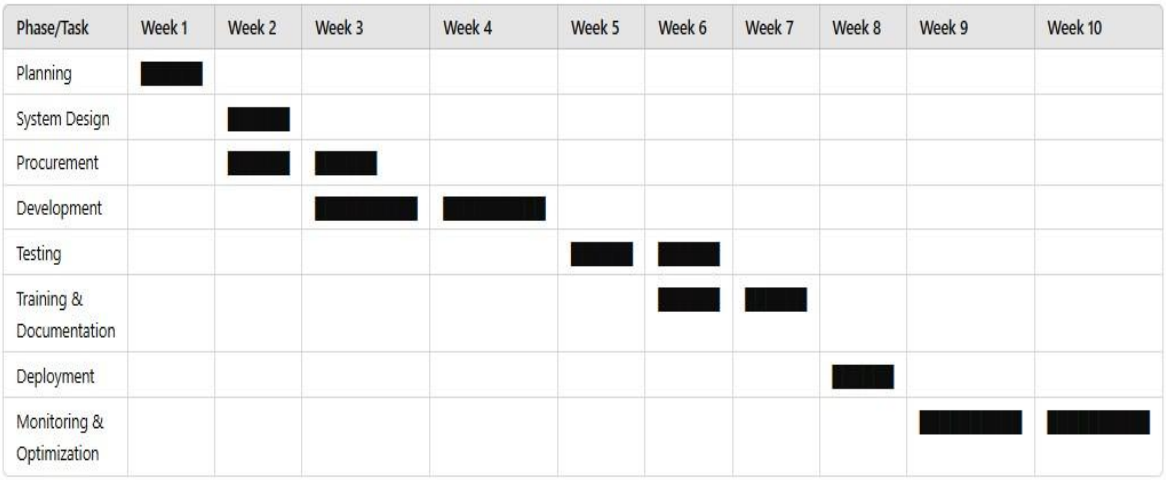
1.6 Project Scheduling Time Frame:

Phase	Tasks	Time Frame
Planning	- Requirement gathering - Budget and resource allocation	1-2 Weeks
System Design	- Designing system architecture - Selecting hardware and software platforms	2 Weeks
Development	- Configuring the software (customizations,integrations) - Database setup	3-4 Weeks

Testing	- Hardware testing - Software testing	2 Weeks
Training and Documentation	- Staff training - System documentation for future reference	1-2 Weeks
Deployment	- Full system rollout - Final troubleshooting	1 Week
Monitoring & Optimization	- Monitoring system performance - Addressing issues and optimizing workflows	Ongoing

Total estimated time: 9-11 weeks

Gantt Chart:



Risk Management:

Risk	Likelihood	Impact	Mitigation Strategy
Budget Overruns	Medium	High	- Define clear scope early - Monitor costs regularly and address overruns promptly
Hardware Delivery Delays	Medium	Medium	- Order hardware early - Have backup suppliers
Integration Issues	Medium	High	- Test all integrations thoroughly before deployment
System Downtime	Low	High	- Schedule deployments during off-peak hours - Have a rollback plan
Staff Resistance	Medium	Medium	- Provide training - Highlight system benefits
Data Security Breaches	Low	High	- Implement robust encryption - Use secure networks and firewalls
Internet Connectivity Issues	Medium	High	- Use reliable ISPs - Ensure backup internet connection for cloud-based systems
Software Bugs	Medium	Medium	- Test thoroughly - Plan for bug fixes and updates during monitoring phase

1.7 Summary

This chapter outlines the foundational aspects of the Restaurant Management System project, focusing on its feasibility, scope, and technical design. The system is designed to automate and streamline core restaurant operations, such as menu management, order processing, billing,

inventory control, and customer relationship management. The feasibility analysis confirms the project's viability across market, technical, and operational dimensions, highlighting its scalability and cost-effectiveness for small and medium-sized businesses. Key technical elements include integration with POS systems, cloud-based accessibility, secure payment processing, and user-friendly interfaces. The chapter also addresses potential risks, resource requirements, and a phased development timeline, ensuring a structured and practical approach to project implementation.

CHAPTER 2 DESIGN AND IMPLEMENTATION

2.1 Introduction

The Design and Implementation chapter focuses on translating the requirements and feasibility studies of the Restaurant Management System into a practical and functional solution. This phase involves creating a detailed system architecture, designing user-friendly interfaces, and ensuring seamless integration of all components. Key elements such as menu management, order processing, billing, inventory control, and customer relationship management are designed with scalability, security, and efficiency in mind. The implementation process is guided by modular development, ensuring that each feature is tested and integrated systematically. This chapter serves as the blueprint for transforming the conceptual framework into a fully operational system, addressing both technical and user-centric aspects.

2.2 Functional Requirements

The Functional Requirements define the specific features and capabilities of the Restaurant Management System to ensure it meets the needs of its users. Key requirements include:

- **Menu Management**
- **Order Processing**
- **Billing System**
- **Inventory Management**
- **Customer Relationship Management**
- **Reporting and Analytics**

FR01	**Registration**
Description	Before using the restaurant management system manager and customers must be registered first
Stakeholder	Manager, Customer

FR02	**Login**
Description	Manager and Customers must login before using the hotel management system
Stakeholder	Manager, Customer, Admin

FR03	**Add Profile Info**
Description	Users can add personal and professional details to complete their profiles.
Stakeholder	Manager, Customer

FR04	**Update Profile Info**
Description	Users can update their profile information as needed to keep it current.
Stakeholder	Manager, Customer

FR05	**Ordering Food**
Description	Customers can browse the menu and place orders for food and beverages.
Stakeholder	Customer, Manager

FR06	**Payment Method**
Description	Customers can choose and process payments through multiple payment options.
Stakeholder	Customer, Manager

FR07	**Customer Relationship Management**
Description	Tracks customer preferences, manages loyalty programs, and enables targeted promotions.
Stakeholder	Customer, Manager

FR08	**Reporting and Analysis**
Description	Provides insights into sales, inventory, and operational performance.
Stakeholder	Manager, Admin

FR09	**Menu Management**

Description	Enables the manager to create, update, and organize the menu with item details.
Stakeholder	Manager

2.3 Non-Functional Requirements

Non-Functional Requirements refer to the qualities or attributes that define how a system should perform, rather than what it should do. These requirements ensure the system meets certain standards in terms of performance, usability, security, and reliability. While functional requirements describe the system's specific functions, non-functional requirements address the overall user experience and operational efficiency.

2.3.1 Performance

The system should handle a high number of concurrent users (waitstaff, kitchen staff, managers) without lag, ensuring fast order processing and transaction handling.

2.3.2 Reliability

The system should be reliable, with minimal downtime or errors. Any failures should be quickly detected, reported, and rectified.

2.3.3 Portability

This is a key non-functional requirement for a Restaurant Management System and it refers to the ability of the system to operate across different environments, devices, or platforms with minimal modification. Ensuring portability allows the system to be flexible and adaptable to future changes or expansions, making it easier for the restaurant to implement updates or switch between different hardware or software platforms.

2.4 Object-oriented development the system development via UML

2.4.1 Diagram of the Use Case

Figure 2: Diagram of the Use Case

2.4.2 Description of the Case

Description of the Case-01: Signing up

<table>
<tr><td>Use Case</td><td>Signing up</td></tr>
<tr><td>Goal</td><td>Users can register to sign in to the system.</td></tr>
<tr><td>Precondition</td><td>Users must install the Hotel Management app for registration.</td></tr>
<tr><td>Success End Condition</td><td>Notification: !!!Successfully Registered!!!</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Submission Not Submitted”</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Customer</td></tr>
<tr><td>Trigger</td><td>User will request a registration form to fill up</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td>
<table>
<tr><td>1.</td><td>Click the "Registration" button.</td></tr>
<tr><td>2.</td><td>Give the registration form.</td></tr>
<tr><td>3.</td><td>Enter the data.</td></tr>
<tr><td>4.</td><td>Click the "Submit" button.</td></tr>
<tr><td>5.</td><td>Data stored</td></tr>
<tr><td>6.</td><td>The details are saved and shown by the system! Your registration has been successful! Alert</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>Error in the System</td></tr>
<tr><td></td><td>1.1.a. Give it another go!</td></tr>
<tr><td>1.2</td><td>The user did not complete the information!</td></tr>
<tr><td></td><td>1.2.a. Verified by the system and alerted by "Please! Complete the box.</td></tr>
<tr><td>1.3</td><td>The system failed to display the error message.</td></tr>
<tr><td></td><td>1.3.a. Show Error Message.</td></tr>
<tr><td>1.4</td><td>The information is not saved by the system.</td></tr>
<tr><td></td><td>1.4.a. "Details did not save" is the notification.</td></tr>
</table>
</td></tr>
<tr><td>Quality Standards</td><td>In half an hour, the user will complete all the information.</td></tr>
</table>

Description of the case-02: Login

<table>
<tr><td>Use Case</td><td>Sign in</td></tr>
<tr><td>Goal</td><td>To authenticate users (managers, customers, admins) and provide access to system features.</td></tr>
<tr><td>Precondition</td><td>The user has to have legitimate login credentials.</td></tr>
<tr><td>Success End Condition</td><td>The user is successfully logged in and redirected to their respective dashboard.</td></tr>
<tr><td>Failed End Condition</td><td>The user is denied access with an appropriate error message for invalid credentials.</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Manager, Customer, Admin

Authentication Server</td></tr>
<tr><td>Trigger</td><td>The user initiates the login process by entering their credentials.</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td>
<table>
<tr><td>1.</td><td>The user navigates to the login page.</td></tr>
<tr><td>2.</td><td>The system prompts the user to enter their username and password.</td></tr>
<tr><td>3.</td><td>The user submits their credentials.</td></tr>
<tr><td>4.</td><td>The system verifies the credentials against the database.</td></tr>
<tr><td>5.</td><td>If the credentials are valid, the user is authenticated.</td></tr>
<tr><td>6.</td><td>The system redirects the user to their respective dashboard (Manager, Customer, or Admin).</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>The system displays an error message: "Invalid username or password."</td></tr>
<tr><td></td><td>The user is prompted to re-enter credentials.</td></tr>
<tr><td>1.2</td><td>The system displays an error: "Login service temporarily unavailable."</td></tr>
<tr><td></td><td>The user is advised to try again later.</td></tr>
<tr><td>1.3</td><td>If multiple failed attempts occur, the account is locked temporarily.</td></tr>
<tr><td></td><td>The user is notified: "Your account is locked. Please reset your password or contact support."</td></tr>
<tr><td>1.4</td><td>If the session expires during login, the system prompts the user to re-enter credentials.</td></tr>
<tr><td></td><td>Passwords must be hashed and stored securely in the database.</td></tr>
</table></td></tr>
<tr><td>Quality Standards</td><td>The login module should handle simultaneous login attempts without performance degradation.</td></tr>
</table>

Case Description-03: Admin Manage Item

Use Case	Admin Manage Item
Goal	To enable the admin to add, update, delete, or view items in the restaurant menu.
Precondition	The admin must be logged into the system with the required permissions to manage menu items.
Success End Condition	The menu is successfully updated with the changes made by the admin, and the updates are saved.
Failed End Condition	Changes are not saved due to validation errors, system errors, or insufficient permissions.
Primary Actors: Secondary Actors:	Admin Database, System
Trigger	The admin selects the "Manage Item" option from the admin dashboard.

<table>
<tr><td>Description / Main

Success

Scenario</td><td><table>
<tr><td>1.</td><td>The admin navigates to the "Manage Item" section on the dashboard.</td></tr>
<tr><td>2.</td><td>The system displays the current list of menu items.</td></tr>
<tr><td>3.</td><td>The admin selects an action: Add, Update, Delete, or View an item</td></tr>
<tr><td>4.</td><td>For Add/Update, the admin fills in or edits item details (name, price, category, description)</td></tr>
<tr><td>5.</td><td>The system validates the inputs and saves the changes to the database.</td></tr>
<tr><td>6.</td><td>The updated menu is displayed to the admin, reflecting the changes made.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>If the admin is not logged in, the system redirects to the login page.</td></tr>
<tr><td>1.2</td><td>If invalid details are entered (missing fields or incorrect formats), the system prompts for corrections.</td></tr>
<tr><td>1.3</td><td>If the database fails to save changes, the system shows an error message and logs the issue.</td></tr>
<tr><td>1.4</td><td>If the updated menu fails to load, the system notifies the admin to retry later.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>This use case ensures that the admin has full control over menu management while maintaining system integrity and user satisfaction</td></tr>
</table>

Case Description-04: Admin Manage Booking

Use Case	Admin Manage Booking
Goal	The goal of this use case is to allow the admin to manage bookings made by customers, including viewing, updating, or canceling reservations.
Precondition	The admin is logged into the restaurant management system. The restaurant has an active booking system in place. The system has valid customer reservation data available for management.

<table>
<tr><td>Success End Condition</td><td>The admin successfully manages a booking (viewing, editing, or canceling), and the changes are reflected in the system.</td></tr>
<tr><td>Failed End Condition</td><td>The admin is unable to manage the booking due to system errors, missing data, or a booking conflict.</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Admin: The restaurant staff responsible for managing bookings.

Customer: The individual who made the booking (though the customer does not directly interact with the admin in this case, their booking details are managed).</td></tr>
<tr><td>Trigger</td><td>The admin selects a booking from the booking management interface to view or manage its details.</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td>
<table>
<tr><td>1.</td><td>Admin logs into the RMS and navigates to the booking management section.</td></tr>
<tr><td>2.</td><td>Admin views a list of current bookings to check the status and details of reservations.</td></tr>
<tr><td>3.</td><td>Admin selects a booking they want to manage (view, update, or cancel).</td></tr>
<tr><td>4.</td><td>Admin updates booking details (e.g., changing the reservation time or party size).</td></tr>
<tr><td>5.</td><td>Admin confirms the changes to ensure the updated information is saved in the system.</td></tr>
<tr><td>6.</td><td>Admin receives confirmation that the booking has been successfully updated or canceled.</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>If the admin is not logged in, they are prompted to log in before accessing the booking management section.</td></tr>
<tr><td>1.2</td><td>If there is a conflict (e.g., double booking for the same time), the admin is notified and asked to choose another time or confirm with the customer.</td></tr>
<tr><td>1.3</td><td>If the admin makes an error while updating (e.g., incorrect data format), the system prompts them to correct the mistake before proceeding.</td></tr>
<tr><td>1.4</td><td>If the admin cancels a booking, the system asks for confirmation to avoid accidental cancellations.</td></tr>
</table>
</td></tr>
</table>

Quality Requirements	This use case provides the admin with an efficient way to manage bookings, ensuring that the reservation process is smooth and
	reliable, while also addressing potential issues such as conflicts or errors during the process.

Case Description-05: View All User

Use Case	View All User
Goal	The goal of this use case is to allow the admin to view the list of all users (customers, staff, and other stakeholders) within the Restaurant Management System
Precondition	The admin is logged into the RMS with appropriate access privileges to view user information. The system has a valid user database with all relevant user data stored.
Success End Condition	The admin successfully views a list of all registered users, with their information (name, email, role) displayed correctly in the system.
Failed End Condition	The admin is unable to view the user list due to issues such as system errors, access restrictions, or missing data.
Primary Actors: Secondary Actors:	Admin: The staff member responsible for managing and viewing user data in the system. User: Customers, staff, or other individuals registered in the system, whose information is being viewed.
Trigger	The admin selects the "View All Users" option from the system's user management section.
Description / Main Success Scenario	1. Admin logs into the RMS and navigates to the user management section. 2. Admin selects the "View All Users" option, prompting the system to retrieve the list of users. 3. The system processes the request, retrieving all user data from the database. 4. The system displays the list of all users, showing their relevant details (name, email, role, and registration status). 5. Admin reviews the user list, and can optionally filter or sort by user role, name, or registration status. 6. Admin exits or continues with further user management tasks**,** such as updating or removing user details.

<table>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>If the admin is not logged in, the system prompts the admin to log in before proceeding.</td></tr>
<tr><td>1.2</td><td>If there are too many users for a single page, the system implements pagination to display the list in manageable sections.</td></tr>
<tr><td>1.3</td><td>If there are any errors retrieving the user list (for instance, an unsuccessful database connection), the system shows an error message and prompts the administrator to try again or get in touch with assistance.</td></tr>
<tr><td>1.4</td><td>If the admin wants to filter the users (e.g., by role), they can apply filters to the user list, and the system updates the list accordingly.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>This use case outlines how an admin can effectively view all users in the system while ensuring that the process is fast, secure, and easy to use. By addressing scalability and error handling, the system will remain functional even as the number of users grows.</td></tr>
</table>

Case Description-06: Admin Add Item

Use Case	Admin Add Item
Goal	The goal of this use case is to allow the admin to add a new item (such as a dish, drink, or special) to the restaurant's menu.
Precondition	The admin is logged into the restaurant management system. The restaurant has an active menu system in place where items can be added. The admin has the required permissions to add items to the menu.
Success End Condition	A new item is successfully added to the restaurant's menu, and the item details are saved in the system.
Failed End Condition	The admin fails to add the item due to missing or invalid data, system errors, or permission issues.
Primary Actors: Secondary Actors:	Admin: The staff responsible for managing and updating the restaurant's menu. System: The restaurant management system, which processes the item addition and stores it in the database.
Trigger	The admin selects the "Add Item" option from the menu management section in the system.

<table>
<tr><td>Description / Main

Success

Scenario</td><td><table>
<tr><td>1.</td><td>Admin logs into the RMS and navigates to the menu management section</td></tr>
<tr><td>2.</td><td>Admin selects the "Add Item" option to begin adding a new item to the menu.</td></tr>
<tr><td>3.</td><td>Admin enters the item details, including name, description, price, category (e.g., appetizer, main course), and any relevant attributes (e.g., vegetarian, gluten-free).</td></tr>
<tr><td>4.</td><td>Admin uploads an image of the item (optional), if required by the system</td></tr>
<tr><td>5.</td><td>Admin reviews the item details and confirms that the information is correct</td></tr>
<tr><td>6.</td><td>Admin clicks "Save" to add the item to the menu. A confirmation message confirming the successful addition of the item is shown by the system.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>If the admin is not logged in, they are prompted to log in before accessing the menu management section.</td></tr>
<tr><td>1.2</td><td>If the image uploaded is not in the correct format or exceeds the file size limit, the system will prompt the admin to upload a valid image.</td></tr>
<tr><td>1.3</td><td>If any required information (e.g., item name, price) is missing or invalid, the system will highlight the fields and prompt the admin to correct them before proceeding.</td></tr>
<tr><td>1.4</td><td>If the admin cancels the operation at any point, the system will discard the entered data and return to the previous menu without saving.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>This use case ensures that admins can easily and efficiently add new menu items, contributing to a flexible and up-to-date menu system. The focus on usability and error handling helps prevent issues and supports smooth operation.</td></tr>
</table>

Case Description-07: Manage Booking

Use Case	Manage Booking

Goal	The goal of this use case is to enable users (e.g., customers or staff) to manage bookings, including making, viewing, modifying, or canceling reservations.
Precondition	The system has a working booking management module in place. Availability of tables or reservations for the selected date and time.
Success End Condition	The booking has been successfully managed (made, updated, or canceled) and reflected in the system.
Failed End Condition	The booking management action fails due to issues like system errors, invalid input, double booking, or non-availability of tables.
Primary Actors: Secondary Actors:	Customer: The person who is making, modifying, or canceling a reservation. Admin: The person managing the bookings on behalf of the restaurant. Restaurant Staff: They may be involved in confirming bookings or assisting customers with booking inquiries.
Trigger	The user (customer or admin) initiates an action to manage a booking, such as making a reservation or updating an existing booking.
Description / Main Success Scenario	

1.	User accesses the booking interface via the website or mobile app (or admin system).
2.	User selects a date and time for the booking or locates an existing reservation to manage.
3.	User provides necessary details such as name, contact information, and party size.
4.	User submits the booking request (or modification) to the system.
5.	System confirms the availability of the selected time or booking modification.
6.	System processes the booking**,** updating the restaurant's availability and notifying the user of the successful booking or changes.

<table>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>If the user is not logged in, they are prompted to log in to complete the booking process.</td></tr>
<tr><td>1.2</td><td>If the selected date/time is unavailable due to double booking or overcapacity, the system suggests alternative times or dates for the user</td></tr>
<tr><td>1.3</td><td>If there is an error in the booking details (e.g., invalid email or phone number format), the system prompts the user to correct the information.</td></tr>
<tr><td>1.4</td><td>If the booking process encounters an issue (e.g., system downtime or server error), the user is notified and provided with the option to retry or contact customer support.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>This use case enables the management of bookings, ensuring that users can easily make, view, and modify reservations while maintaining the efficiency and reliability of the system.</td></tr>
</table>

Case Description-08: View Menu

Use Case	View Menu
Goal	The goal of this use case is to allow customers (or admin) to view the menu items offered by the restaurant, including descriptions, prices, and availability.
Precondition	The menu is uploaded and updated in the system.
Success End Condition	The customer (or admin) successfully views the complete menu with all items, descriptions, prices, and availability.
Failed End Condition	The customer (or admin) is unable to view the menu due to technical issues, lack of internet connectivity, or an empty menu.
Primary Actors: Secondary Actors:	Customer: The individual browsing the menu to choose items. System: The backend that delivers the menu information to the user interface.
Trigger	The customer or admin clicks on the "View Menu" option in the restaurant's system (website, mobile app, or in-restaurant tablet interface).

<table>
<tr><td>Description / Main

Success

Scenario</td><td><table>
<tr><td>1.</td><td>Customer navigates to the menu section by clicking on the "View Menu" button on the app or website.</td></tr>
<tr><td>2.</td><td>The system retrieves the menu from the database and displays all items (e.g., appetizers, main courses, desserts, beverages).</td></tr>
<tr><td>3.</td><td>The customer browses through the menu, filtering by categories (e.g., vegetarian, vegan, non-vegetarian) or viewing descriptions and prices.</td></tr>
<tr><td>4.</td><td>The customer views individual item details, such as ingredients, portion size, allergens, or special notes.</td></tr>
<tr><td>5.</td><td>The customer sees availability status (e.g., out of stock or special offers).</td></tr>
<tr><td>6.</td><td>The customer can make a selection to add items to their order, or simply finish viewing the menu.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>If the customer is logged in (for personalized menus or previous preferences), the system retrieves the personalized menu.</td></tr>
<tr><td>1.2</td><td>If an item is unavailable, the system shows a notification and suggests alternatives from the same category (e.g., "This item is out of stock. Would you like to try [alternative item]?").</td></tr>
<tr><td>1.3</td><td>If the customer’s internet connection is lost or the system encounters a technical issue, the customer is shown an error message like "Unable to retrieve the menu. Please try again later."</td></tr>
<tr><td>1.4</td><td>If the menu is displayed on an in-restaurant tablet or kiosk, the system allows the customer to navigate through categories or search using a search bar.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>This use case ensures that the restaurant's menu is easily accessible and functional for customers, promoting a smooth and efficient browsing experience. It also addresses the possibility of alternative actions or failures in case of unavailability or technical issues.</td></tr>
</table>

2.4.3 Activity Diagram

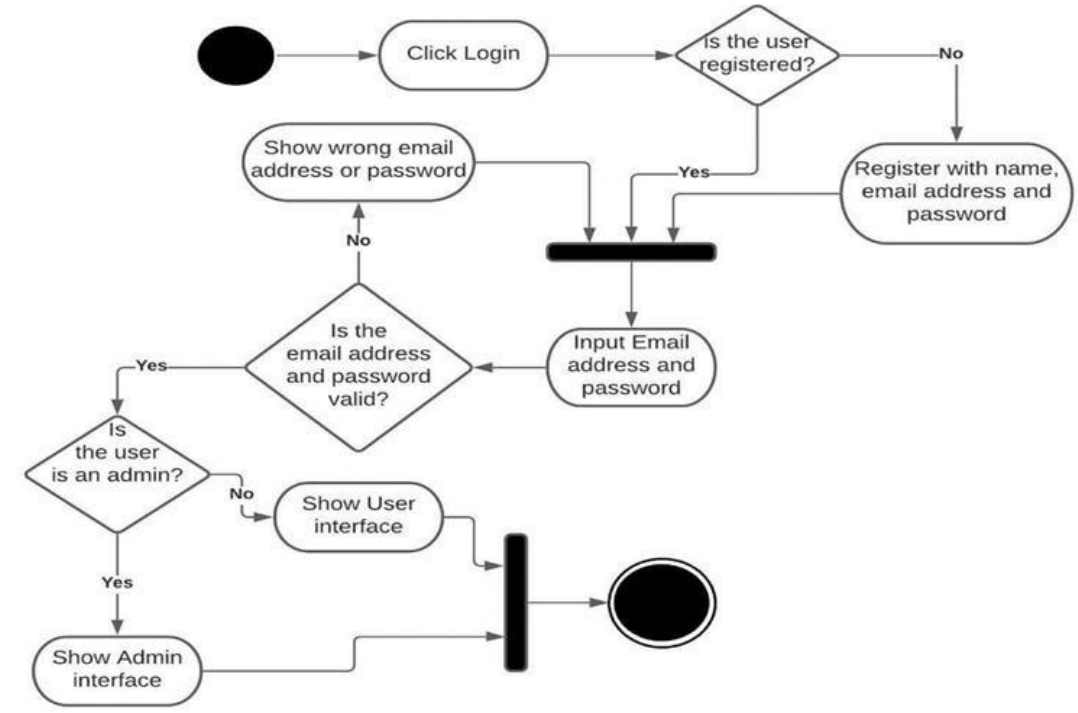


Figure 3: Activity Diagram for Login and register

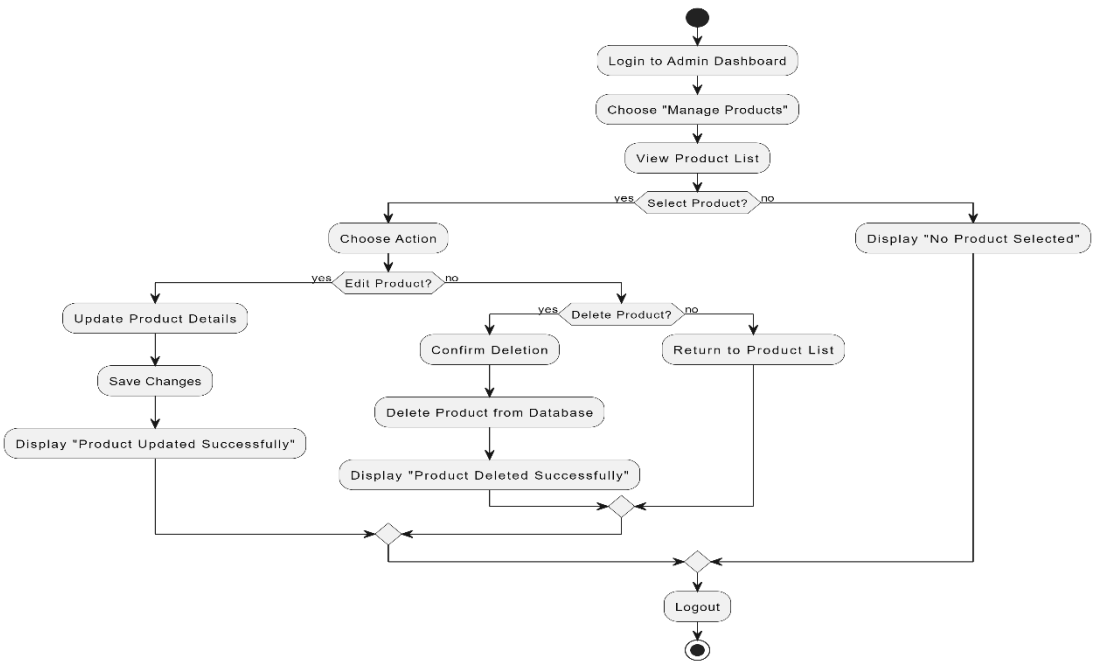


Figure 3.1: Activity Diagram for Manage item

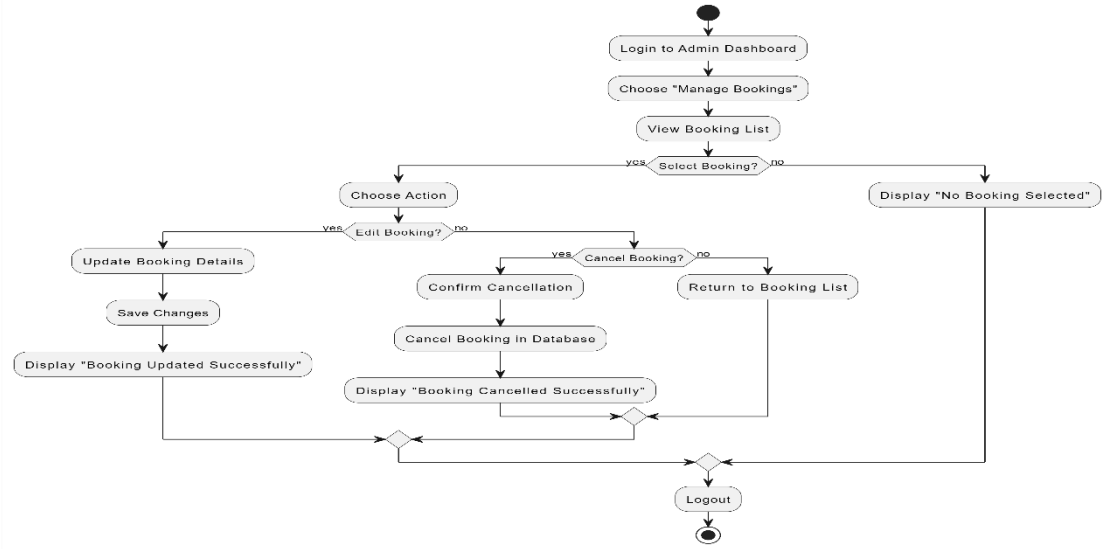


Figure 3.2: Activity Diagram for Management Booking

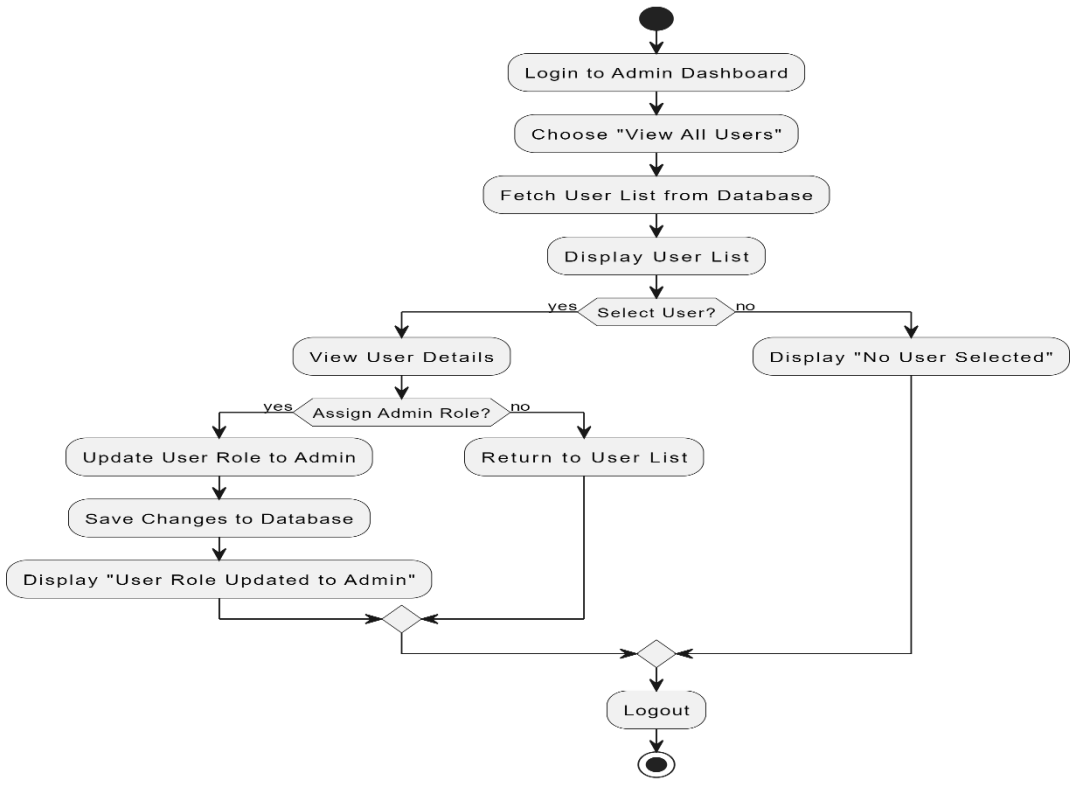


Figure 3.3: Activity Diagram for all user

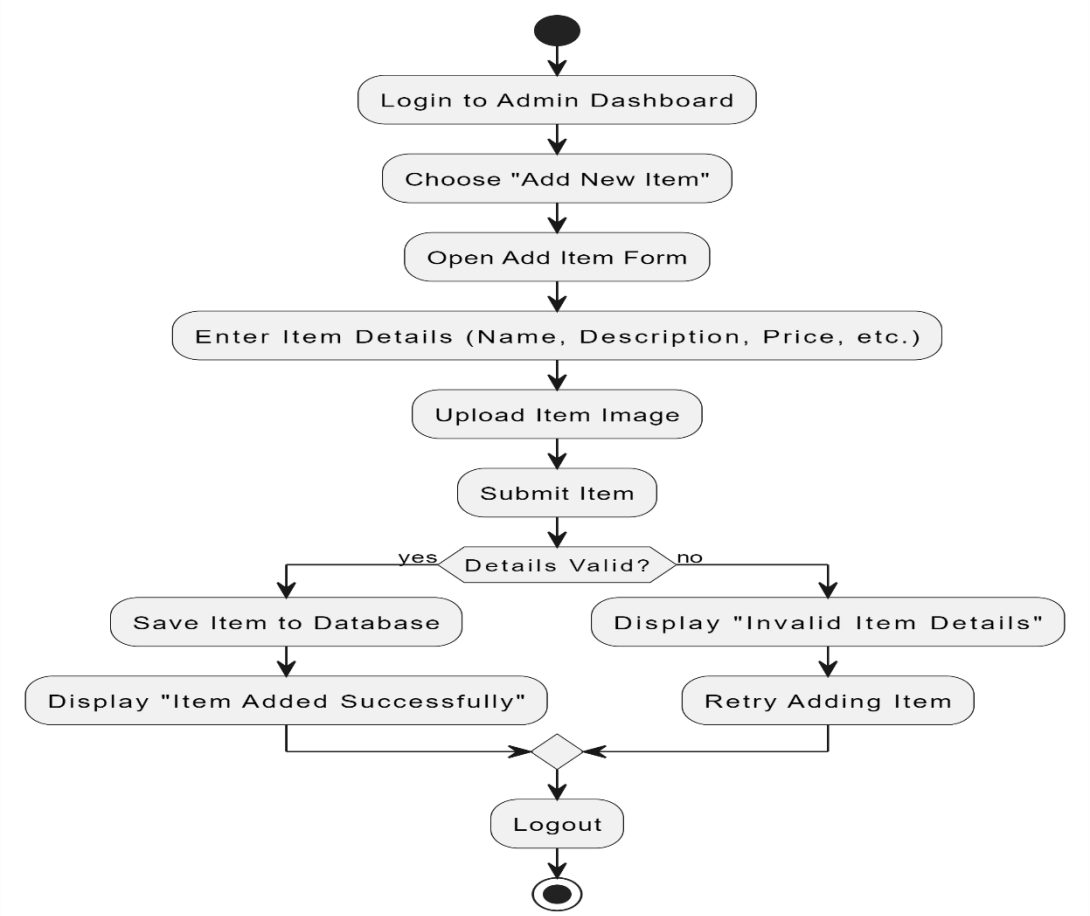


Figure 3.4: Activity Diagram for add item

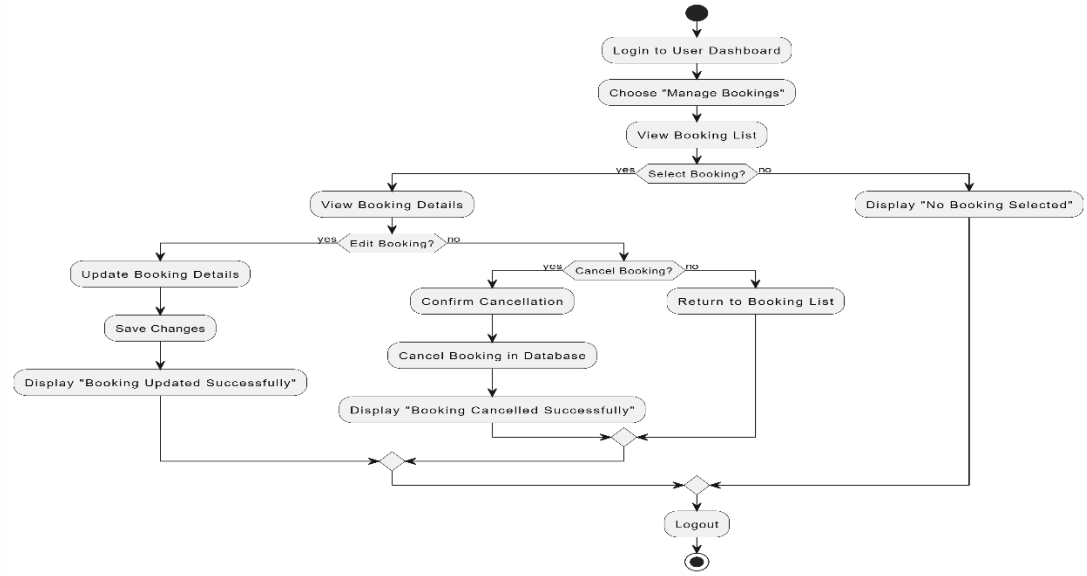


Figure 3.5: Activity Diagram for Management Booking

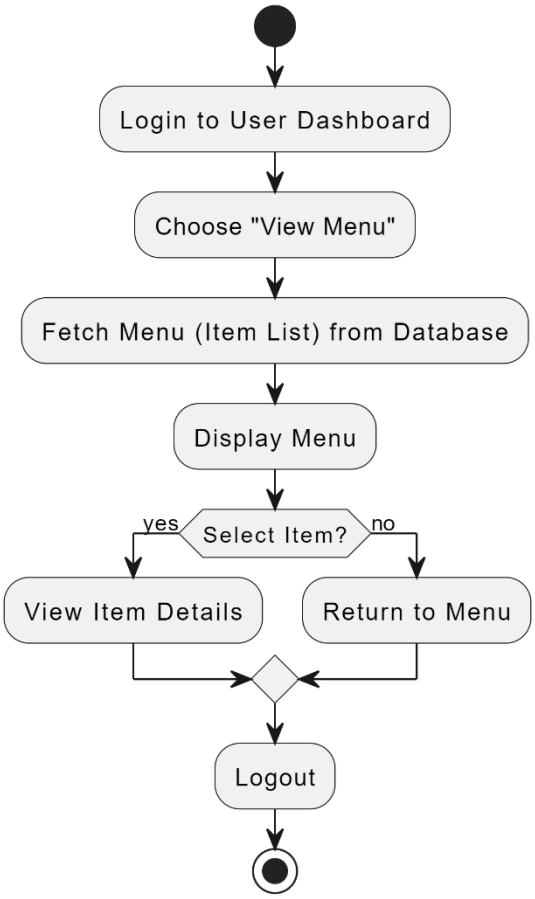


Figure 3.6: Activity Diagram for view menu

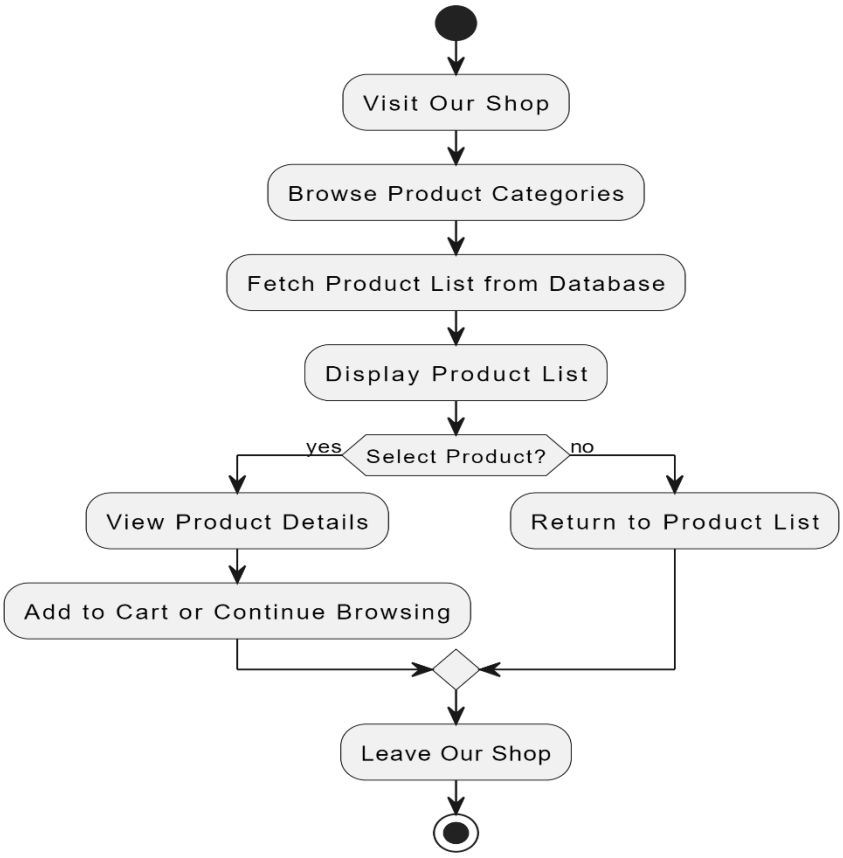


Figure 3.7: Activity Diagram for our shop

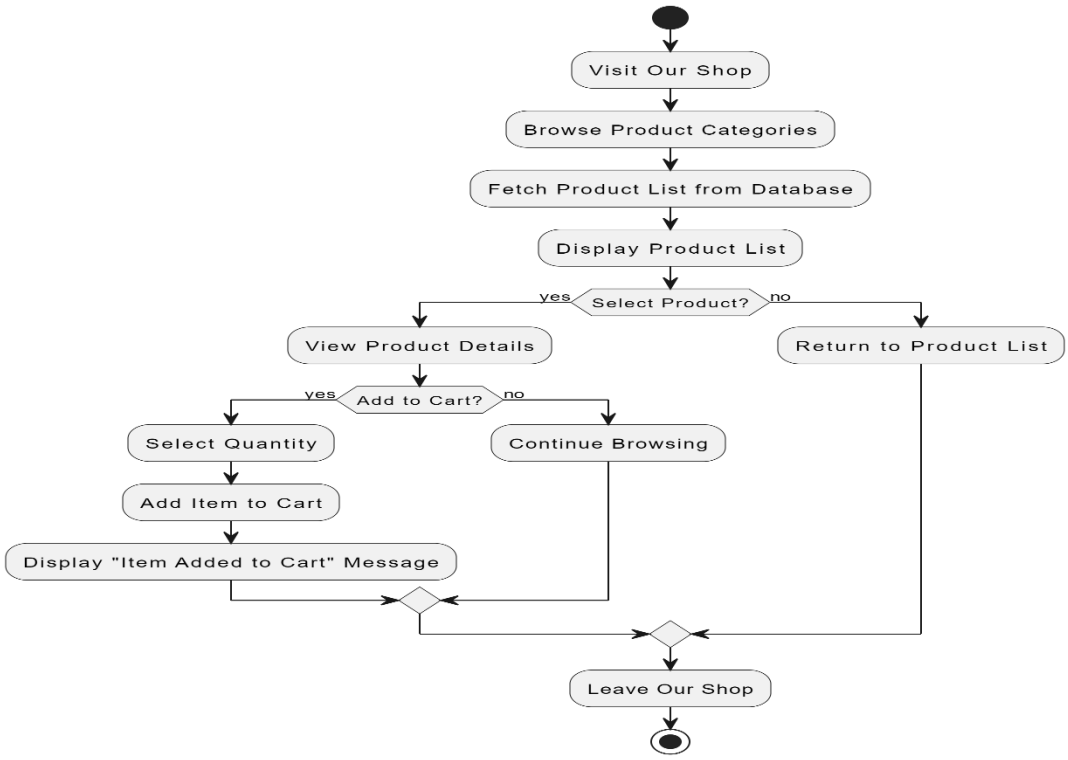


Figure 3.8: Activity Diagram for add item to cart

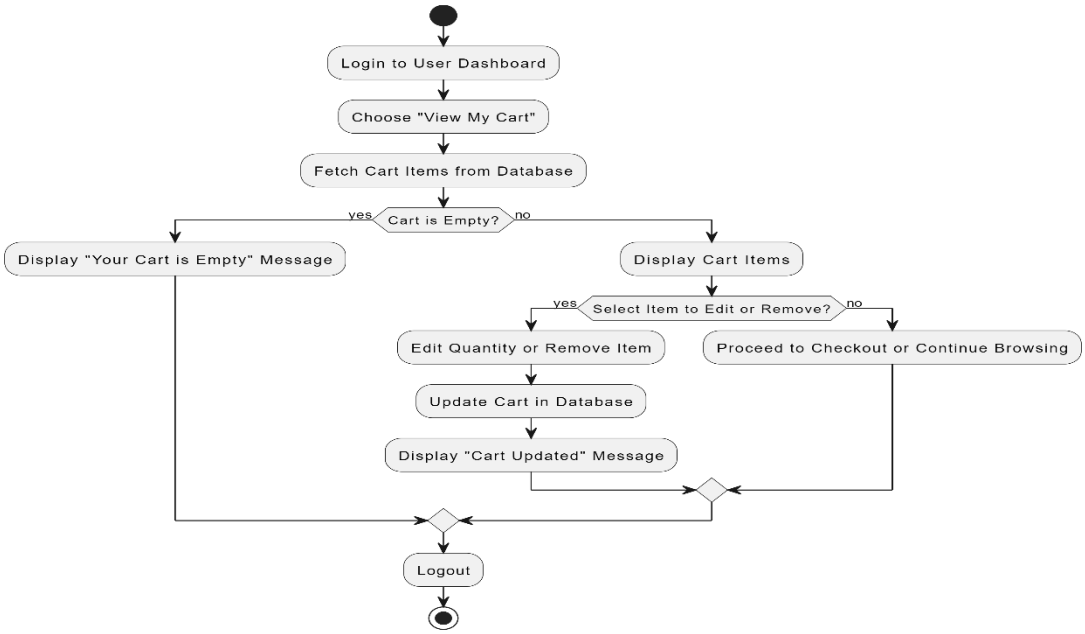


Figure 3.9: Activity Diagram for view my cart

2.4.4 Sequence Diagram

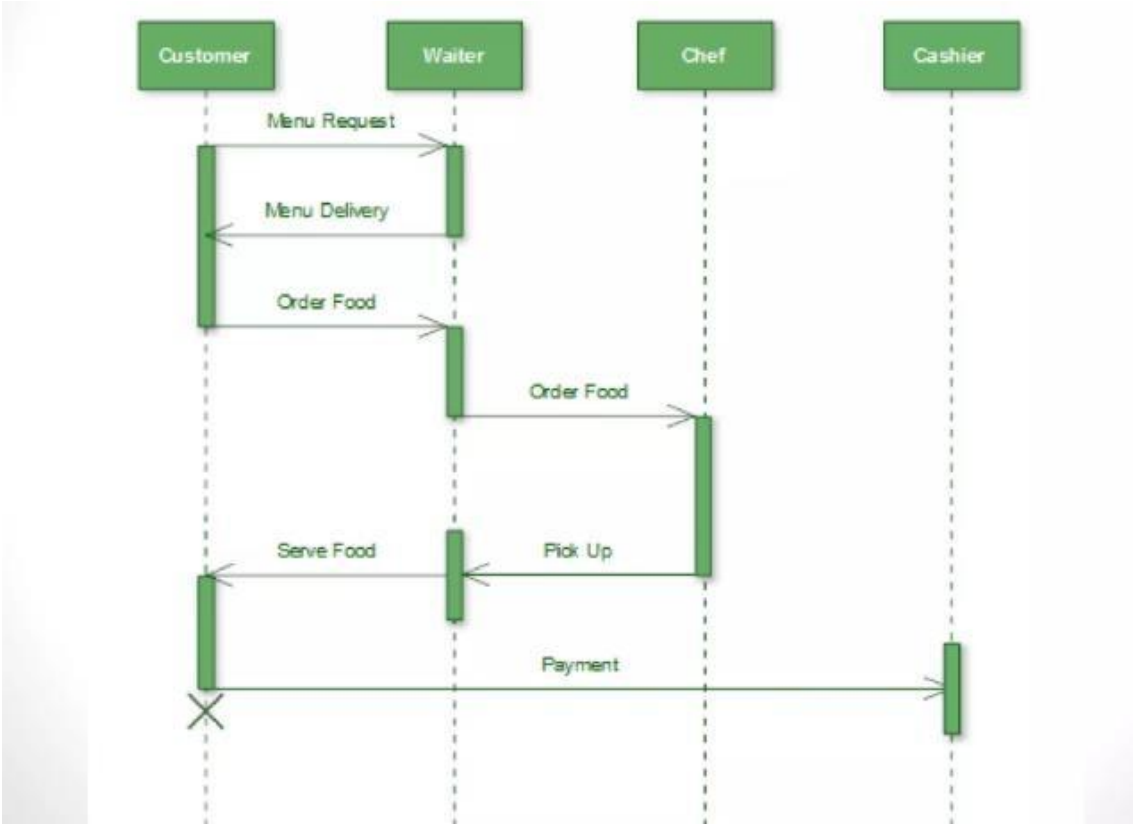


Figure 3: Sequence Diagram

2.4.5 Class Diagram

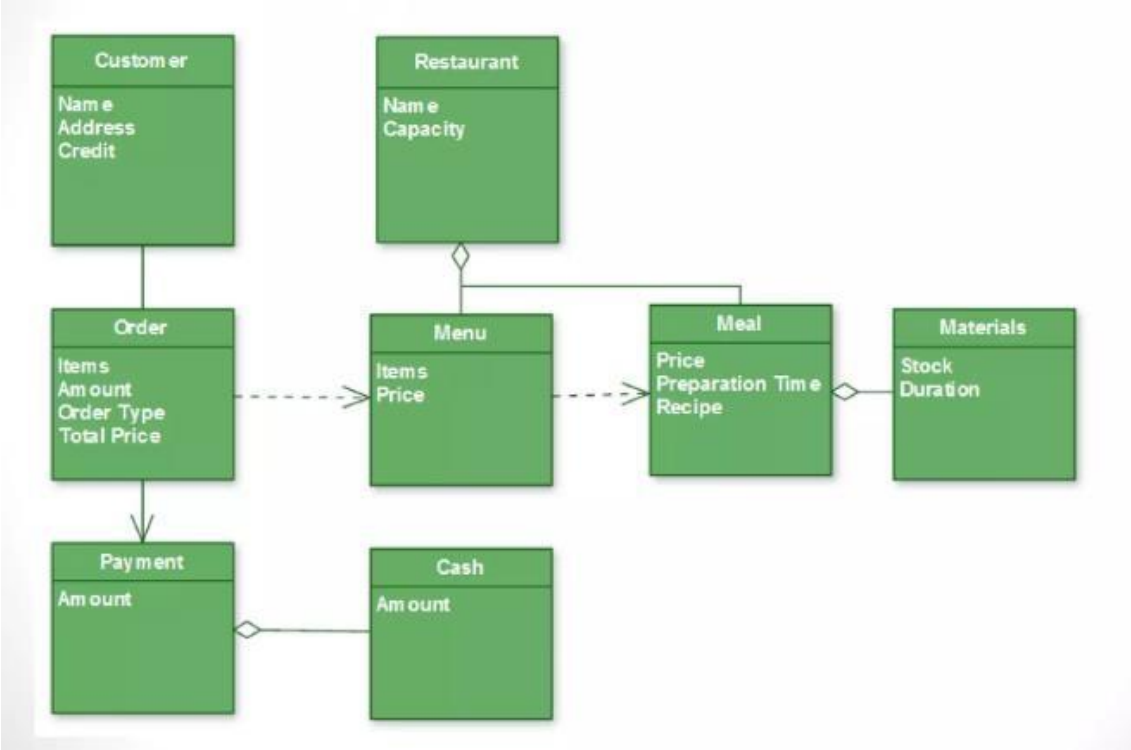


Figure 4: Class Diagram

2.4.6 ER Diagram

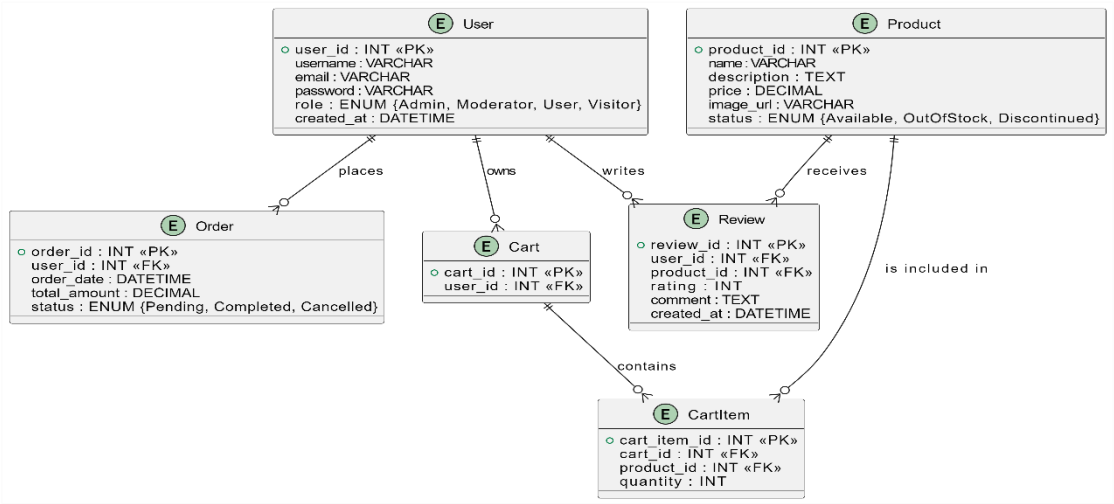
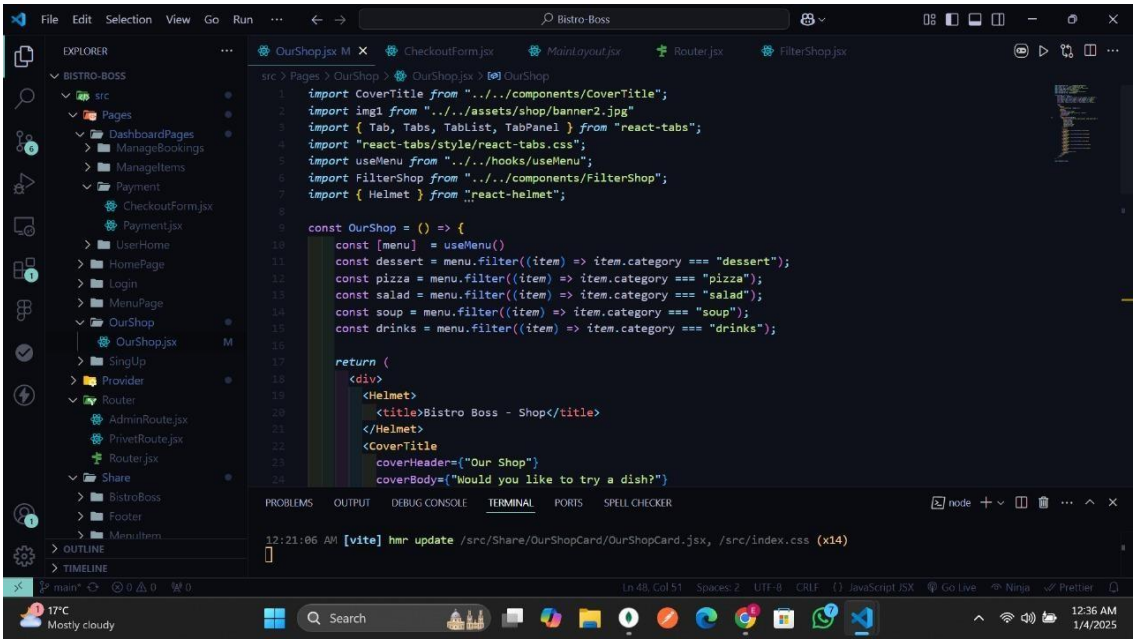


Figure 5: ER Diagram

2.5 Coding: Appendix A



2.6 Summary

This chapter focuses on the Design and Implementation of the Restaurant Management System by defining its functional and non-functional requirements, as well as its object-oriented design using UML. Functional requirements outline core features such as user registration, login, menu management, ordering, payment processing, and reporting. Non-functional requirements emphasize system performance, security, usability, and scalability. The object-oriented design, represented through UML diagrams like use case, class, and sequence diagrams, provides a structured blueprint of the system's architecture and interactions. Together, these components form the foundation for building a robust, efficient, and user-centric system.

Chapter 3 Software Testing

3.1 Introduction

Software testing is a critical phase in the development lifecycle of the Restaurant Management System to ensure its reliability, functionality, and usability. This process involves systematically identifying and fixing defects, verifying that the system meets its functional and non-functional requirements, and ensuring it performs well under real-world conditions. The testing phase covers all modules, including registration, login, menu management, order processing, payment integration, and customer relationship management. By employing a combination of manual and automated testing methods, the project aims to validate the system's security, scalability, and compatibility across devices. Software testing ensures that the RMS delivers a seamless user experience, minimizes errors, and maintains high operational standards.

3.2 Testing Features

3.2.1 The feature that will be tested

a. User Registration
b. User Login
c. Restaurant Recommendation
d. Admin Login
e. Admin Dashboard
f. Add Some Restaurant (Admin)
g. User Logout
h. Admin Logout

3.3 Methods of Testing

3.3.1 Test Methodology

The test approach for the Restaurant Management System involves a combination of functional, integration, and user acceptance testing to ensure that the system functions as expected. Below is the detailed approach for each feature:

a. User Registration

Objective: Ensure users can register with valid credentials. Testing
Steps**:**

- Test registration with valid and invalid data.

- Validate error messages for incorrect inputs.
- Verify successful account creation in the database.
  Test Types: Functional testing, input validation testing.

b. User Login

Objective: Ensure users can log in with correct credentials and are denied access with invalid credentials.
Testing Steps:

- Test login with valid credentials.
- Attempt login with invalid credentials and verify error messages.
- Test account lockout after multiple failed attempts.
  Test Types: Functional testing, security testing.

c. Restaurant Recommendation

Objective: Verify that recommendations are personalized based on user preferences or location.
Testing Steps:

- Test recommendation logic with various user profiles.
- Validate the display of recommendations based on location or cuisine.
- Ensure recommendations update dynamically with user interactions.
  Test Types: Functional testing, algorithm validation testing.

d. Admin Login

Objective: Ensure admins can securely log in to manage the system.

Testing Steps:

- Test login with valid admin credentials.
- Verify error messages for invalid inputs.
- Test multi-factor authentication if applicable.
  Test Types: Functional testing, security testing.

e. Admin Dashboard

Objective: Validate that the admin dashboard displays relevant data and allows access to all administrative functions.

Testing Steps:

- Test the loading of dashboard data (user metrics, restaurant stats).
- Verify the functionality of each feature (view users, manage orders).

- Test the responsiveness of the dashboard on different devices.
  Test Types: Functional testing, UI/UX testing, performance testing.

f. Add Restaurant (Admin)

Objective: Ensure admins can add restaurants with valid information. Testing
Steps:

- Test adding restaurants with complete and incomplete data.
- Verify data validation for mandatory fields (name, location).
- Check the addition of the restaurant in the database.
  Test Types: Functional testing, input validation testing.

g. User Logout

Objective: Ensure users can securely log out of the system. Testing
Steps:

- Test the logout functionality for different user states (e.g., active sessions).
- Verify that a logged-out user cannot access secured pages.
- Check that session tokens are invalidated upon logout.
  Test Types: Functional testing, security testing.

h. Admin Logout

Objective: Ensure admins can securely log out of the system. Testing
Steps:

- Test logout functionality after administrative actions.
- Verify redirection to the login page after logout.
- Ensure admin session is invalidated post-logout.
  Test Types: Functional testing, security testing.

3.3.2 Pass/Fail Criteria

The Pass/Fail Criteria define the standards used to evaluate whether the tests for the Restaurant Management System features have succeeded or failed. These criteria ensure that the system meets functional, performance, and security requirements. Below are the detailed pass/fail criteria:

General Criteria

Pass:

- The test case executes successfully without errors.
- The feature behaves as expected according to the requirements.
- Data is correctly processed, stored, and retrieved.
- The user interface displays the intended outputs with no visual or functional glitches.

Fail:

- The feature produces incorrect results or errors during execution.
- The system crashes, hangs, or shows unexpected behaviour.
- Validation rules fail, or incorrect inputs are accepted.
- Data integrity issues arise

Feature-Specific Pass/Fail Criteria

User Registration:

- **Pass**: The user successfully registers with valid data, and the account is created in the database.
- **Fail**: Registration fails with valid inputs, or the system allows invalid data to pass validation.

User Login:

- **Pass**: The user can log in with valid credentials and access authorized features.
- **Fail**: Login attempts fail with correct credentials, or unauthorized access is granted.

Restaurant Recommendation:

- **Pass**: Recommendations match user preferences or location and display correctly.
- **Fail**: Recommendations are inaccurate or fail to load.

Admin Login:

- **Pass**: The admin successfully logs in and accesses the dashboard with appropriate permissions.
- **Fail**: Login fails with correct credentials, or unauthorized access is granted.

Admin Dashboard:

- **Pass**: All data loads correctly, and features function as intended.
- **Fail**: Data is missing or incorrect, or dashboard features do not respond.

Add Restaurant (Admin):

- **Pass**: A new restaurant is successfully added to the database with complete and valid data.
- **Fail**: The restaurant addition process fails or incomplete data is saved.

User/ Admin Logout:

- **Pass**: The system securely logs out the user/admin and invalidates the session.
- **Fail**: The session remains active, or secured pages are accessible after logout.

Performance Criteria:

- **Pass**: The system responds within the acceptable time limit for all tested features.
- **Fail**: The system exhibits slow performance, timeouts, or crashes under load.

Security Criteria:

- **Pass**: The system prevents unauthorized access and protects sensitive user data.
- **Fail**: Vulnerabilities like SQL injection, session hijacking, or unauthorized data access are identified.

3.4 System Testing (Test Cases with Report)

Test Case: 3.4.1			**Test Case Name: Admin Login**		
System: Restaurant Management System			**Subsystem:** Admin Authentication		
Designed by: Shahjada Moon			**Design Date:** 14.08.24		
Executed by: Shahjada Moon			**Execution date:** 25.12.24		
Description:	The admin login for the Restaurant Management system by providing valid login information.				
Pre-condition:	The admin accesses the login page.				
Step	**Email**	**Password**	**Response**	**Pass/ Fail**	**Comment**

1	admin@gmail.com	123456	Login successful	pass	Admin Login is completed with valid information
2		123456	Email field empty	fail	Admin must put an email id
3	admin@gmail.com		Password field empty	fail	Admin must put a password

Test Case: 3.4.2			**Test Case Name: Login**		
System: Restaurant management system			**Subsystem:** User Authentication		
Designed by: Shahjada Moon			**Design Date:** 14.08.24		
Executed by: Shahjada Moon			**Execution date:** 25.12.24		
Description:	The user login for the Restaurant Management System by providing valid login information.				
Pre-condition:	The user accesses the login page.				
Step	**Email**	**Password**	**Response**	**Pass/ Fail**	**Comment**
1	Moon@gmail.com	123456	Login successful	pass	User Login is completed with valid information
2		123456	Email field empty	fail	User must put an email id
3	Moon@gmail.com		Password field empty	fail	User must put a password

Test Case: 3.4.3	**Test Case Name: Registration**
System: Restaurant Management System	**Subsystem:** User Authentication
Designed by: Shahjada Moon	**Design Date:** 14.08.24
Executed by: Shahjada Moon	**Execution date:** 25.12.24
Description:	The user register for the restaurant Management System by providing valid registration information.
Pre-condition:	The user accesses the registration page.

Step	**Username**	**Email**	**Password**	**Response**	**Pass/ Fail**	**Comment**
1	Moon02	Moon@gmail.com	123456	Registration successful	pass	User registration is completed with valid information
2		Moon @gmail.com	123456	Username field empty	fail	User must put a username
3	Moon02		123456	Email field empty	fail	User must put an email id
4	Moon02	Moon @gmail.com		Password field empty	fail	User must put a password

Test Case: 3.4.4	**Test Case Name: Admin Dashboard**
System: Restaurant Management System	**Subsystem:** Admin Pannel
Designed by: Shahjada Moon	**Design Date:** 14.08.24
Executed by: Shahjada Moon	**Execution date:** 25.12.24
Description:	The admin check for the dashboard.
Pre-condition:	Admin must login to the system.

Step	**Response**	**Comment**
1. Click Dashboard	Successful	Successfully viewing the admin dashboard

Test Case: 3.4.5		**Test Case Name: Create Order**
System: Restaurant Management System		**Subsystem:** User Account
Designed by: Shahjada Moon		**Design Date:** 14.08.24
Executed by: Shahjada Moon		**Execution date:** 25.12.24
Description:	User create an order.	
Pre-condition:	The User must login to the account.	
Step	**Response**	**Comment**
1. Click Order	Successful	Successfully place an order

Test Case: 3.4.6		**Test Case Name: Track Order**
System: Restaurant Management System		**Subsystem:** User Account
Designed by: Shahjada Moon		**Design Date:** 14.08.24
Executed by: Shahjada Moon		**Execution date:** 25.12.24
Description:	User tracking Order.	
Pre-condition:	The User must login to the account.	
Step	**Response**	**Comment**
1. Input Tracking ID	Successful	Successfully track the Order.

Test Case: 3.4.7		**Test Case Name: User Logout**
System: Restaurant Management System		**Subsystem:** User Account
Designed by: Shahjada Moon		**Design Date:** 14.08.24
Executed by: Shahjada Moon		**Execution date:** 25.12.24
Description:	User Logout from the System.	
Pre-condition:	The User must login to the account.	

Step	Response	Comment
1. Click on Logout icon	Successful	Successfully logout from the System

Test Case: 3.4.8		Test Case Name: Admin Logout
System: Restaurant Management System		**Subsystem:** Admin Account
Designed by: Shahjada Moon		**Design Date:** 14.08.24
Executed by: Shahjada Moon		**Execution date:** 25.12.24
Description:	Admin Logout from the System.	
Pre-condition:	Admin must login to the account.	
Step	**Response**	**Comment**
1. Click on Logout icon	Successful	Successfully logout from the System

3.5 Summary

This chapter outlines the testing process for the Restaurant Management System emphasizing the importance of validating its functionality, performance, and security. Key features such as user registration, login, restaurant recommendations, admin dashboard, and logout are tested against defined pass/fail criteria. The testing ensures seamless user interactions, data integrity, and system reliability. The criteria establish clear benchmarks for success, ensuring the system meets user expectations and operates without errors. By addressing functional, performance, and security requirements, this chapter ensures the RMS delivers a robust and user-friendly experience.

Chapter 4 Deployment and Maintenance

4.1 Introduction

The deployment and maintenance phase ensures the smooth transition of the Restaurant Management System from development to a live environment. Deployment involves setting up the system on production servers, configuring databases, and ensuring all features are operational. Post-deployment, the maintenance phase focuses on monitoring system performance, resolving issues, applying updates, and incorporating user feedback. Regular maintenance ensures system reliability, security, and scalability, adapting to evolving business needs while delivering a seamless experience to users and administrators.

4.2 Try to follow the SRLC (software release life cycle)

1. Planning

- Create a deployment plan, set hardware/software requirements, and get stakeholder approval.

2. Design

- Define deployment architecture (cloud/on-premise) and plan maintenance processes for bug fixes and updates.

3. Development

- Build the system with essential features and automate deployment using scripts.

4. Testing

- Conduct unit, integration, and user acceptance testing (UAT), ensuring stress testing for peak loads.

5. Deployment

- Deploy the system to production, migrate data, and provide go-live support.

6. Maintenance

- Fix bugs, release updates, gather user feedback, and monitor system performance.

By following SDLC, the SMS ensures smooth deployment, effective maintenance, and continuous improvement.

Chapter 5 User Manual

5.1 Introduction

The User Manual serves as a comprehensive guide to help users effectively navigate and utilize the Restaurant Management System. It includes step-by-step instructions for key features such as registration, login, ordering food, and managing profiles, as well as administrative tasks like adding restaurants and generating reports. The manual provides troubleshooting tips, FAQs, and visuals to enhance understanding and ensure a seamless user experience. Designed for both customers and administrators, it simplifies interactions with the system, promoting efficiency and ease of use.

5.2 Project Functionalities

Admin Login Page

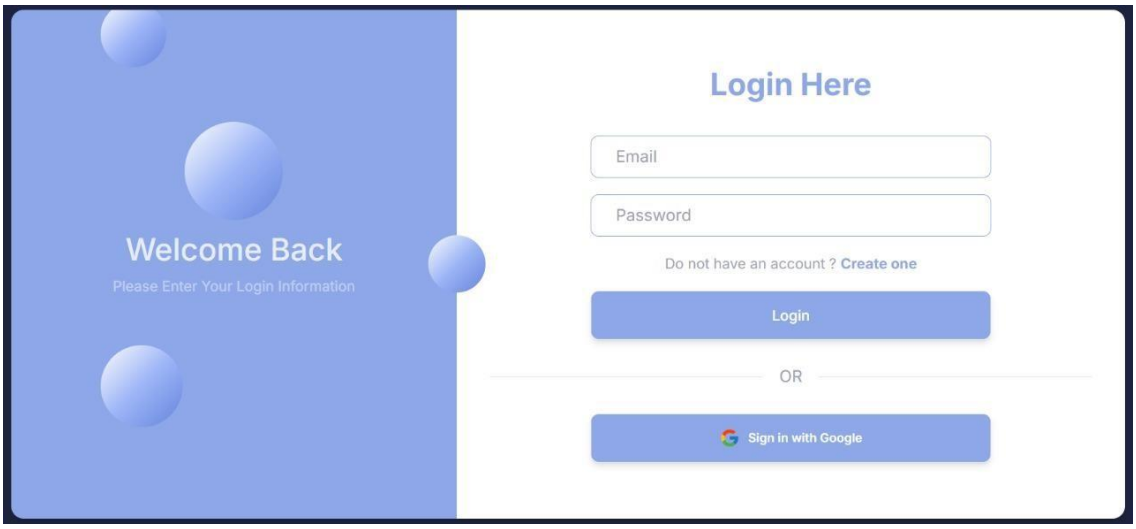


Figure: Admin Login Page

Register Page

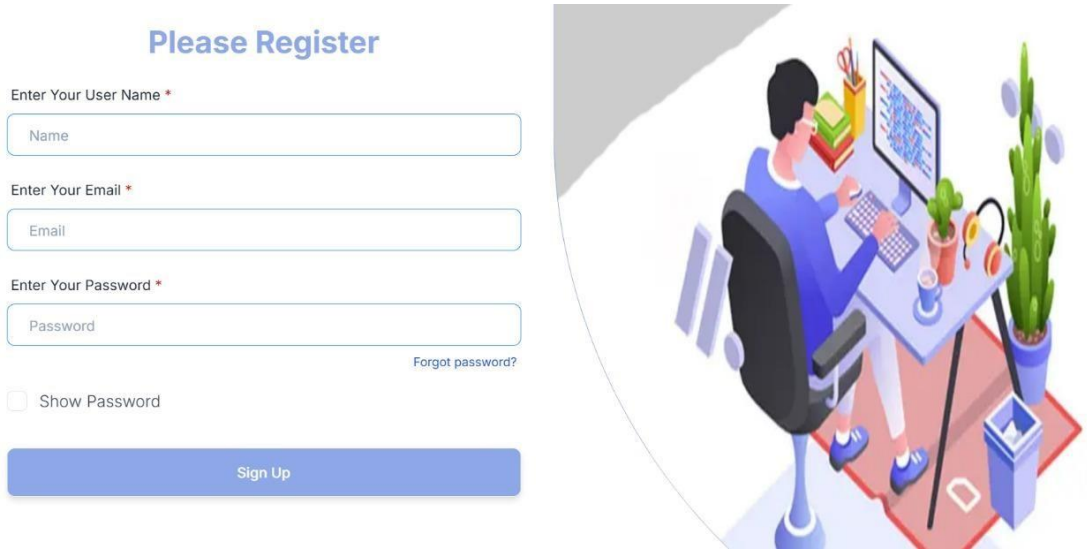


Figure: Register Page

Home page

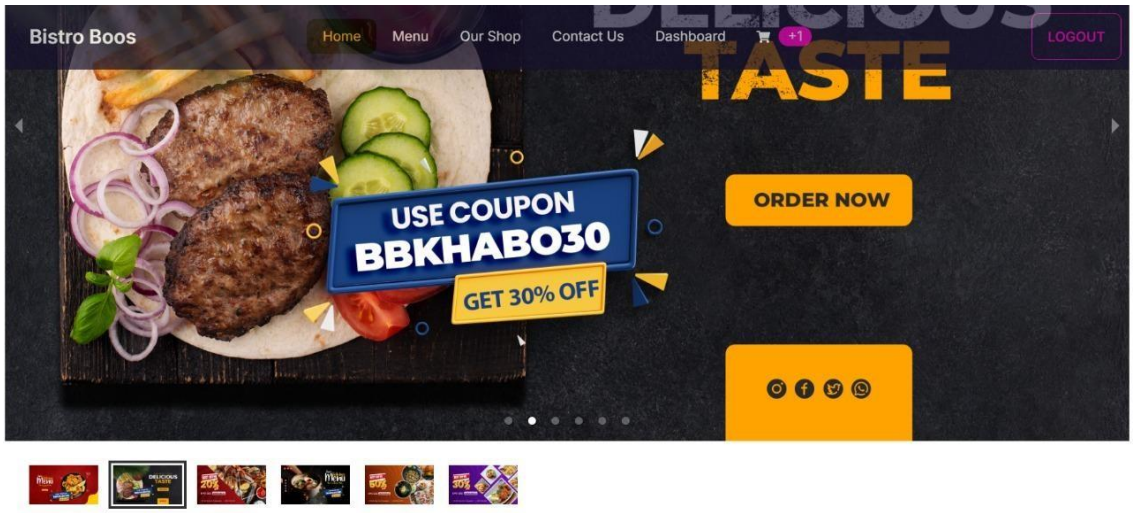


Figure: Home Page

User Dashboard

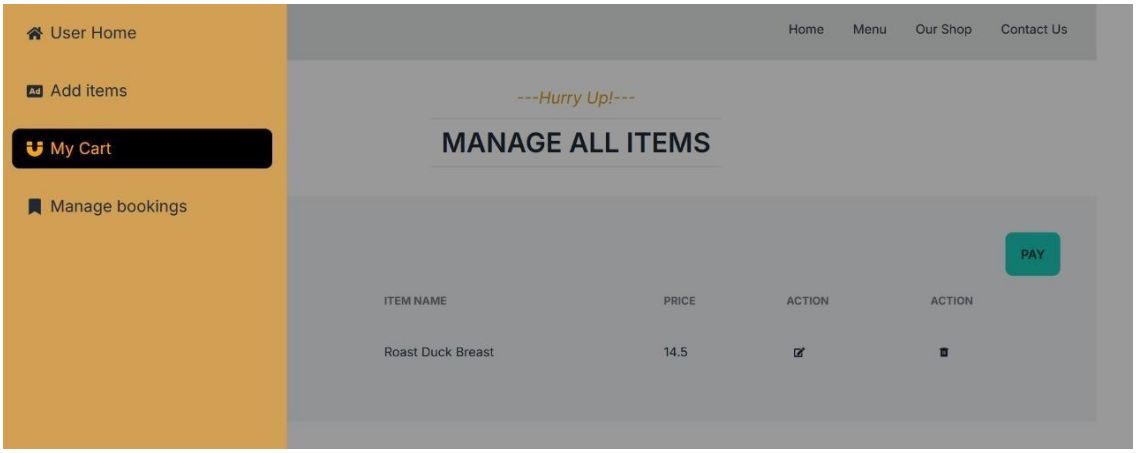


Figure: User Dashboard

Admin Dashboard

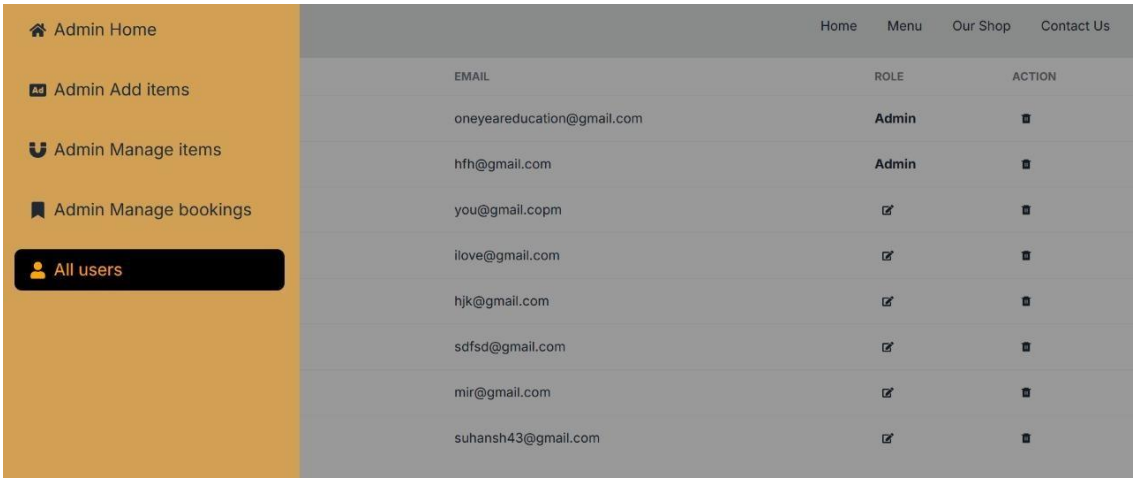


Figure: Admin Dashboard

5.3 Summary

A centralized hub for admins to manage reservations, menu, inventory, sales reports, employee schedules, and customer feedback. A personalized interface for customers to view past orders, manage reservations, access loyalty rewards, and update profiles. Welcoming page showcasing menu highlights, special offers, customer reviews, contact details, and quick access to key features. Admin and login page form for new users to create an account with personal details, password setup, and optional subscription to newsletters. A secure portal for admins to log in with credentials, featuring two-factor authentication and password recovery options.

Chapter 6 Project Summary

6.1 Introduction

The Restaurant Management System is a comprehensive solution designed to streamline restaurant operations and enhance customer experiences. This project integrates functionalities such as user registration, menu management, food ordering, payment processing, and reporting, all while ensuring security, usability, and scalability. By leveraging object-oriented design principles and UML diagrams, the system offers a structured, efficient, and user-friendly platform tailored to the needs of restaurant managers, customers, and administrators. This project aims to

modernize restaurant management through automation and seamless integration of essential processes.

6.2 Project Limitation

Time Constraints:

- Limited time for project execution, especially during peak restaurant operations, may lead to rushed testing or skipped features.
- Deadlines for deployment may compromise the thoroughness of system testing or staff training.

Technological Constraints:

- Insufficient budget can restrict access to premium hardware or software features, forcing reliance on cost-effective but less robust alternatives.
- Customizations or third-party integrations may be deprioritized due to financial limitations.

Technological Constraints:

- Compatibility issues between new systems and legacy hardware/software may lead to inefficiencies or increased costs for replacement.
- Internet dependency for cloud-based systems may hinder performance in areas with unreliable connectivity.

Unaddressed Requirements:

- Some non-critical requirements, like support for niche payment methods or rare hardware devices, may not be addressed during initial implementation.
- Scalability for multi-location restaurants might be delayed, impacting larger businesses planning to expand soon.

By acknowledging these limitations, the project team can focus on future upgrades or extensions to address unresolved issues, improving the system's performance and utility over time.

6.3 Scope

Included Features and Modules:

- Operational Scope: Implementing better workflow systems, using technology, and training staff on efficient processes. Automation in inventory management and ordering systems can also reduce operational bottlenecks.
- Customer Service Scope: Staff training and development programs focused on customer service skills, creating a service standard protocol, and feedback mechanisms to address customer complaints. Providing personalized service can improve loyalty.
- Food Quality and Consistency Scope: Standardizing recipes, improving quality control measures, and conducting regular staff training on food preparation. Monitoring supplier quality and ensuring fresh, high-quality ingredients are used consistently.
- Technology Scope: Implementing modern technology solutions, such as mobile ordering apps, reservation systems, and integrated POS systems. Leveraging data analytics can help understand customer preferences, optimize operations, and improve decision-making.
- Cost Control and Financial Management Scope: Implementing better inventory management practices, reducing food waste through portion control, and monitoring labour costs by optimizing staff schedules. Conducting regular financial analysis can help identify inefficiencies and opportunities for improvement.
- Health and Safety Scope: Ensuring rigorous compliance with food safety protocols and conducting regular staff training on hygiene practices. Introducing routine health checks and audits will help maintain high standards.
- Customer Feedback and Relationship Management Scope: Establishing systems for gathering, analysing, and responding to customer feedback, such as surveys, online reviews, or loyalty apps. Personalizing follow-ups with customers and resolving complaints quickly can enhance the dining experience.

6.4 Future Work

Suggested Improvements

- **Enhanced User Interface**: Improve the design for a more intuitive and visually appealing interface. Add customizable dashboards for different user roles (manager, cashier, kitchen staff).

Faster Reporting:

- Implement real-time analytics for immediate insights. Introduce drill-down capabilities in reports for granular analysis.

Offline Mode:

- Enable offline functionality to ensure operations during internet outages, with data sync upon reconnection.

Additional Features

- **Loyalty and Rewards Program**: Create a dedicated module for tracking and managing customer loyalty points, discounts, and personalized offers.
- **Mobile Accessibility**: Develop mobile apps for staff to take orders, manage tables, or view reports remotely.
- **Customer Feedback Collection**: Include a feature to gather and analyse customer feedback directly through receipts or mobile apps.

Areas for Further Research

- **Emerging Technologies:** Explore blockchain for secure payment processing and transparent supply chain management. Investigate AI models for better predictive analytics and personalized customer experiences.
- **Sustainability Integration:** Research features to track food waste and energy usage, providing insights for sustainability initiatives.
- **Global Compatibility**: Study international standards and regional preferences to make the system adaptable for a global market.

6.5 Conclusion

Upgrading the Restaurant with new features, such as predictive analytics, mobile apps, and enhanced integration, can improve operational efficiency and customer satisfaction. Exploring emerging technologies like IoT, AI, and blockchain can position the system as a market leader, while addressing sustainability and global adaptability can ensure long-term relevance.

REFERENCES

Use a reference manager such as *Mendeley, EndNote* or any reference manager software to generate all your list of references here. Once all the references are included then apply *Caption for Reference* style.

Alexandrov, N., & Alexandrov, V. (2015). Computational science research methods for science education at PG level. Procedia Computer Science, 51(1), 1685–1693. https://doi.org/10.1016/j.procs.2015.05.305

Crooks, N. M., & Alibali, M. W. (2014). Defining and measuring conceptual knowledge in mathematics. Developmental Review, 34(4), 344–377. https://doi.org/10.1016/j.dr.2014.10.001

Gertler, M. S. (2003). Tacit knowledge and the economic geography of context, or the undefinable tacitness of being (there). Journal of Economic Geography, 3(1), 75–99. https://doi.org/10.1093/jeg/3.1.75

APPENDICES

Appendix A: Acronyms and Abbreviations

- **RMS**: Restaurant Management System
- **CRM**: Customer Relationship Management
- **POS**: Point of Sale
- **UML**: Unified Modelling Language

Appendix B: UML Diagrams

- **Use Case Diagram**: Illustrates interactions between actors (Admin, Customer, Manager) and system functionalities.
- **Class Diagram**: Defines the structure of the system, including classes like User, Menu, Order, and Payment.
- **Sequence Diagram**: Describes the flow of events during processes like ordering food or managing items.

Appendix C: Functional Requirements

- User registration and login.
- Admin dashboard for managing items, users, and reports.
- Food ordering and payment integration.

Appendix D: Non-Functional Requirements

- Performance: Response time under 2 seconds.
- Security: Secure login and data encryption.
- Usability: Intuitive user interface.

Appendix E: Test Cases

Example Test Case for User Login: •

Test ID: TC001

- **Description**: Verify user login with valid credentials.
- **Expected Result**: Successful login and redirection to the user dashboard.

Appendix F: Deployment Details • **Server**:

Hosted on cloud infrastructure

- **Database**: MySQL for data storage.
- **Languages**: Frontend (HTML, CSS, React), Backend (Nodejs).

Appendix G: Tools and Technologies

- **Development Tools**: Visual Studio Code, GitHub for version control.
- **Testing Tools**: Selenium for automation testing.

Appendix H: References

- Online tutorials and documentation for software development.

- Research papers on restaurant management trends.
- Manuals for UML tools like Lucidchart or Visual Paradigm.

Restaurant Management System

ORIGINALITY REPORT

17%	9%	2%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	3%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	2%
3	Submitted to Arab Open University Student Paper	2%
4	Submitted to Universiti Malaysia Pahang Student Paper	1%
5	123dok.com Internet Source	1%
6	Submitted to De Montfort University Student Paper	1%
7	Submitted to NCC Education Student Paper	1%
8	Submitted to Blue Mountain Hotel School Student Paper	<1%
9	Submitted to UC, Irvine Student Paper	<1%

10	Submitted to University of Greenwich Student Paper	<1%
11	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
12	Submitted to The Robert Gordon University Student Paper	<1%
13	erepository.uonbi.ac.ke:8080 Internet Source	<1%
14	Submitted to Southern New Hampshire University - Continuing Education Student Paper	<1%
15	greenict.com.kh Internet Source	<1%
16	www.mdpi.com Internet Source	<1%
17	www.slideshare.net Internet Source	<1%
18	Submitted to Middle East College Student Paper	<1%
19	Submitted to University Of Tasmania Student Paper	<1%
20	www.coursehero.com Internet Source	<1%

21	Submitted to INTI Universal Holdings SDM BHD Student Paper	<1 %
22	Submitted to King's Own Institute Student Paper	<1 %
23	Submitted to Victorian Institute of Technology Student Paper	<1 %
24	trap.ncirl.ie Internet Source	<1 %
25	Submitted to University of Warwick Student Paper	<1 %
26	gcu-dl.bsmlib.com Internet Source	<1 %
27	Submitted to American University of Iraq Student Paper	<1 %
28	Submitted to Brunel University Student Paper	<1 %
29	Submitted to Florida Institute of Technology Student Paper	<1 %
30	Submitted to LUT-korkeakoulut Student Paper	<1 %
31	Submitted to Liverpool John Moores University Student Paper	<1 %

32	www.bartleby.com Internet Source	<1%
33	Submitted to Queen Mary and Westfield College Student Paper	<1%
34	Submitted to Trine University Student Paper	<1%
35	Submitted to UOW Malaysia KDU University College Sdn. Bhd Student Paper	<1%
36	Submitted to University Tun Hussein Onn Malaysia Student Paper	<1%
37	guides.visual-paradigm.com Internet Source	<1%
38	may1524.sd.ece.iastate.edu Internet Source	<1%
39	Submitted to UNITEC Institute of Technology Student Paper	<1%
40	Submitted to University of Sydney Student Paper	<1%
41	Submitted to University of Zululand Student Paper	<1%
42	access.atis.org Internet Source	<1%

43	www.vskills.in Internet Source	<1%
44	Submitted to Singapore Institute of Technology Student Paper	<1%
45	Submitted to Wawasan Open University Student Paper	<1%
46	Sanchez, Alyssia. "Home-Based Reminder System for Enhanced Dementia Support: Leveraging Data-Driven Insights for Daily Living.", University of Toronto (Canada), 2024 Publication	<1%
47	Submitted to Universiti Teknologi Malaysia Student Paper	<1%
48	Submitted to University of Houston Clear Lake Student Paper	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		