

RESTAURANT CHAIN INFORMATION SYSTEM

Group 7 – Lab CL02



Zihua Zeng	7164695
Phuong Anh Phan	7862118
Khodedah Shadman	8820867
Yiheng Huang	8846145

Recipients: Chain Restaurant Management (Client)

Date: 31/May/2025

Executive Summary

Modern restaurant chains face significant challenges in maintaining operational efficiency and customer satisfaction due to the complexity of managing multiple outlets, diverse menus, and an extensive workforce. As the hospitality sector continues to evolve, integrating information systems has become crucial to overcoming these challenges and gaining a competitive edge (Alrawadieh 2019; Ivanov & Webster 2017). This report presents a comprehensive systems analysis for a proposed IT solution to manage a chain of restaurants. Furthermore, by consolidating all core business data into a single, integrated platform, the system is expected to deliver significant operational benefits, including streamlined order processing, accurate employee time tracking, and enhanced data consistency across all restaurant locations.

Moreover, Unified Modelling Language (UML) diagrams will be used to depict the interactions of users, including customers, staff and managers. The system, along with the linkages between essential data entities (restaurants, employees, menu items, and orders) will also be demonstrated using the UML models. These visualisations offer a comprehensive representation of business processes and data flows, supporting the design and future advancement of the solution.

Table of Contents

Executive Summary	1
1. Introduction	3
2. System Vision Document	4
• 2.1 Problem Description	4
• 2.2 System Capabilities	4
• 2.3 Business Benefits	5
3. Project Planning and Management	8
• 3.1 Project Iteration Schedule	8
• 3.2 Work Breakdown Structure (WBS)	9
• 3.3 Gantt Chart	10
4. Stakeholder Analysis	11
5. Feasibility Analysis	11
• 5.1 Technical Feasibility	11
• 5.2 Economic Feasibility	12
• 5.3 Quantify Project Approval Factors	14
7. Cost/Benefit Analysis	18
8. Requirements Analysis and Modelling	19
• 8.1 UML Activity Diagrams	19
• 8.2 Use Case and UML Use Case Diagrams	24
○ 8.2.1 Use Case Diagram	24
○ 8.2.2 User Goals	25
○ 8.2.3 Use Case Descriptions	27
○ 8.2.4 Fully Developed Use Case Descriptions	31
• 8.3 Domain Model Class Diagram	67

9. Conclusion	71
References	73

1. Introduction

In modern hospitality management, restaurant chains face significant operational challenges related to complexity, including managing multiple locations, diverse menu selections, and varied employee responsibilities. The absence of centralized data management presently results in inefficiencies, discrepancies, and protracted decision-making processes (Deloitte 2016; NCR Corporation 2020). This report presents a systems analysis for a centralized information system intended for handling the occurred issues. The system aims to optimize operations by enhancing data precision and elevating customer experience through effective service delivery.

The analysis undertaken in this report follows established system analysis methodologies as taught in CSIT114, ensuring rigorous requirement gathering, stakeholder alignment, and feasibility assessment. The Unified Modelling Language (UML) will also be adopted as the primary method of representing and validating system interactions and data flows, which is consistent with industry best practices (Object Management Group 2017). Besides that, use cases, activity diagrams, and domain model class diagrams are employed to articulate clearly and comprehensively how users, customers, managers, and other restaurant staff will interact with the proposed system.

Overall, this systems analysis report provides the foundation for the next steps in design and execution. It outlines benefits, risks, and resource concerns in an organized and evidence-based manner, ensuring that the new system would greatly increase operational efficiency and customer service across the restaurant chain.

2. System Vision Document

2.1. Problem Description

The current operational environment for the Restaurant chain lacks an integrated system, resulting in duplicate records, inefficient shift tracking, and reliance on manual processes. This leads to errors in inconsistent customer experience, inefficient operations and fragmented data management. Additionally, the disjointed environment would cause delays in order processing, delays in payroll approvals and limit scalability. Therefore, a unified, comprehensive information system is essential to enhance operational efficiency, ensure data integrity and improve customer experience

2.2. System Capabilities

The suggested information system addresses restaurant chain operating issues by providing clearly specified functionality. The project brief states that the new system will record restaurant information, track personnel hours, maintain and amend menus, and quickly process customer orders. These features aim to streamline important business processes, improve operational efficiency, and provide managers and workers with reliable, real-time data.

Restaurant information system

- The centralised database will automatically record restaurant information, such as restaurant name, city, phone number and type.
- Easy to update restaurant details

Employee Management

- The system will record the information of the employee, such as employee's name, hire date, salary and role
- The system allows restaurant managers to manage employee details, modify work shifts and record work periods.

- Differentiate employee roles, such as excluding a waiter from the fast food industry.

Menu Management

- Allows the restaurant manager to modify menu items, update existing ones and remove items.
- Store detailed item descriptions, such as “percentage of alcohol contents” for drinks and the “name of ingredients” and the “amount of each ingredient used”.

Customer order processing

- User-friendly interface for customers to search menu items by keywords or categories.
- Support self-service order placement and generate a unique order number for each order.
- Track the order submission.

Payment handling

- Payment processing supports both cash and cards.
- Securely record the order number, total cost and card number for the payment.

Order management

- Real-time tracking of order status for managers and staff.
- Recording of order completion details and service time.

2.3. Business Benefits

The implementation of the restaurant chain information system will deliver measurable improvements and significant business benefits:

- Improved operational efficiency:
 - ❖ Automating manual workflows, such as menu updates and order tracking, to reduce manual effort and processing time.
 - ❖ Faster and more accurate order taking and processing can lead to quicker food serving.
- Better cost control

- ❖ Recording Employee work periods could provide accurate payroll and optimised labour costs
- ❖ The payment process with accurate recording the card number, order number and total cost would be beneficial to future auditing
- Enhance customer satisfaction
 - ❖ More efficient order processing can significantly reduce the waiting time of customers
- Data security and accuracy
 - ❖ The centralized management of a chain restaurant can minimise redundancies and inconsistency.
- Scalability for growth
 - ❖ The standardised system will make it easy to add in new restaurant and new staff.
- Strategic Decision-Making
 - ❖ With the consistent and accurate data input, the manager and owner could analyze the business trend.

3. Project Planning and Management

3.1. Project Iteration Schedule

Phase 1: Project Initiation (20/03/25 - 8 days)

Task 1.1: Form project teams and assign roles (Project Manager, Planner, Analysts).- 1 day

Task 1.2: Conduct initial meetings to discuss project scope and objectives.- 2 days

Task 1.3: Research and gather relevant information about restaurant information systems.- 5 days

Phase 2: Requirement Analysis & Planning(28/03/25 - 10 days)

Task 2.1: Identify key stakeholders and conduct stakeholder analysis.-2 days

Task 2.2: Draft the System Vision Document.- 2 days

Task 2.3: Develop the Project Iteration Schedule.- 2 days

Task 2.4: Create a Work Breakdown Structure (WBS).- 2 days

Task 2.5: Prepare a Gantt Chart for project tracking.- 2 days

Phase 3: Initial Checkpoint & Progress Review(25/03/25 - 5 days)

Task 3.1: Present project progress in the laboratory.- 1 day

Task 3.2: Address feedback and refine documents.- 4 days

Phase 4: System Analysis & Modeling(1/04/25 - 10 days)

Task 4.1: Conduct feasibility analysis.

Task 4.2: Develop use case diagrams and descriptions.

Task 4.3: Create UML Activity Diagrams.

Task 4.4: Design the UML Domain Model Class Diagram.

Phase 5: System Design & Refinement(14/04/2025 - 10 days)

Task 5.1: Refine system requirements based on feedback. - 6 days

Task 5.2: Validate UML models and update diagrams as needed. - 4 days

Phase 6: Final Report Compilation & Submission(28/04/2025 - 5 days)

Task 6.1: Compile all documents into a single final report. - 2 days

Task 6.2: Conduct peer reviews and final proofreading. - 2 days

Task 6.3: Submit the final report by the deadline - 1 day

Phase 7: Post-Submission Reflection(05/05/2025 - 1 days)

Task 7.1: Conduct team retrospective to evaluate project execution.

Task 7.2: Identify lessons learned for future projects.

3.2. Work Breakdown Structure

Requirements Analysis

- 1.1 Gather functional and non-functional requirements
- 1.2 Identify entities and relationships
- 1.3 Confirm scope with stakeholders

System Design

- 2.1 Create Entity-Relationship (ER) diagram
- 2.2 Create UML class diagram
- 2.3 Define database schema
- 2.4 Design user interfaces

Project Planning

- 3.1 Prepare Work Breakdown Structure
- 3.2 Define project iterations
- 3.3 Allocate resources and tasks

System Implementation

- 4.1 Set up development environment
- 4.2 Develop data models
- 4.3 Implement core modules (restaurant, employee, menu, orders)
- 4.4 Implement UI and data entry forms

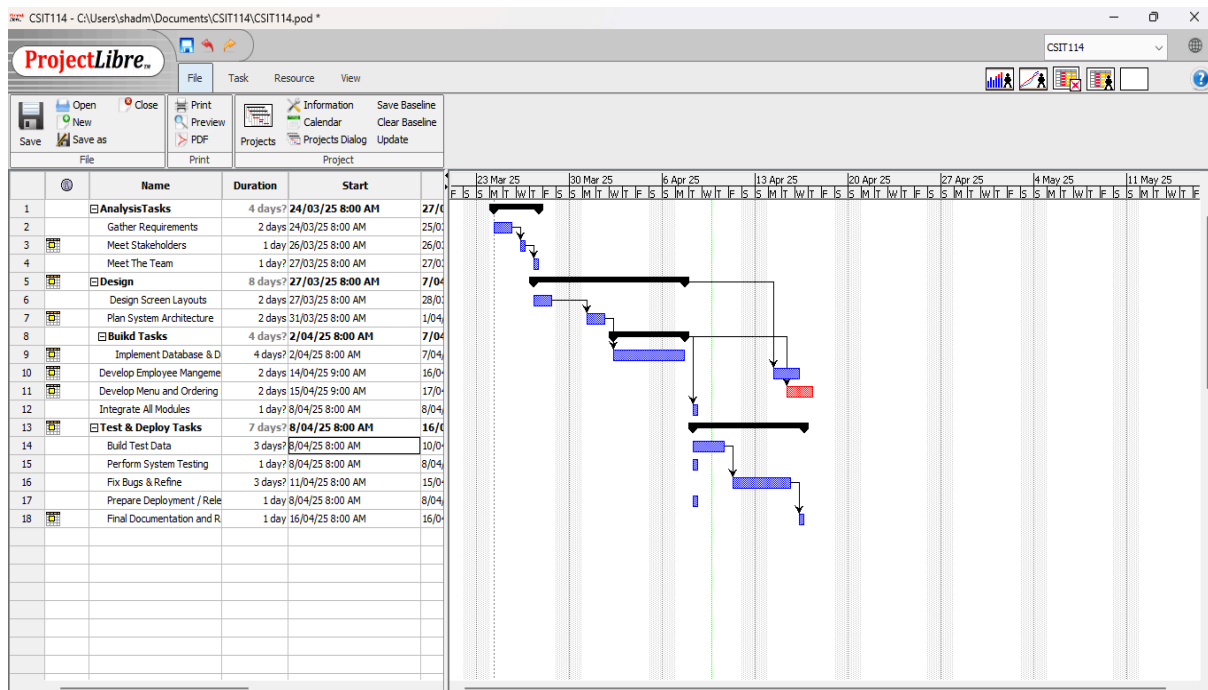
Testing

- 5.1 Unit testing
- 5.2 Integration testing
- 5.3 System testing

Documentation & Finalization

- 6.1 User manual
- 6.2 Final report
- 6.3 Presentation/demo

3.3. Gantt Chart



4. Stakeholder Analysis

Stakeholder	Type	Role / Description	Interest	Influence	Engagement Strategy
Restaurant Manager	Internal, Operational	Oversees operations and schedules	High	High	Manage Closely
Cooks	Internal, Operational	Prepares meals based on menus	Medium	Medium	Keep Satisfied
Waiters (Fine Dining)	Internal, Operational	Serve customers during dine-in	Medium	Medium	Keep Satisfied
Customers	External, Operational	Place orders and make payments	High	Medium	Keep Informed
IT System Administrator	Internal, Executive	Maintains system integrity and performance	High	High	Manage Closely
Restaurant Chain Owner	Internal, Executive	Oversees overall business performance	High	High	Manage Closely

5. Feasibility Analysis

5.1. Introduction

This section of the Project Report will detail the Feasibility Analysis for the proposed information system for the chain of restaurants. The purpose of this analysis is to evaluate the viability of the project across several key dimensions before significant resources are committed. The core process of "Identify the problem or need and obtain approval" includes the activity to "Perform risk and feasibility analysis". This analysis helps in making an informed decision about whether to proceed with the project, identifying potential challenges, and outlining mitigation strategies

5.2. Technical Feasibility

The successful implementation of the proposed information system necessitates a robust database-driven application capable of effectively managing data related to restaurants, employees, menus, and customer orders. A relational database management system (RDBMS), such as MySQL or PostgreSQL, is recommended to ensure reliable data storage, efficient querying, and high scalability. Such RDBMS solutions are mature, widely used, and have well-established community support, which enhances both the development process and long-term maintainability.

An essential consideration for the new system is the user interface, which could be delivered as either a web-based or desktop application, depending upon the accessibility requirements and the operational environment of the restaurant chain. Web-based applications offer the advantage of broader accessibility, ease of updates, and greater platform independence, whereas desktop applications might offer richer interactions and potentially better offline support.

Furthermore, the proposed solution incorporates the use of Computer-Aided Software Engineering (CASE) tools, facilitating efficient system analysis, design, and documentation. The availability and proficiency with these tools further enhance technical feasibility by streamlining the development lifecycle and reducing potential technical risks.

5.3. Economic feasibility

Economic feasibility determines if the anticipated benefits of the system outweigh its estimated costs. While a detailed cost/benefit analysis with NPV, payback period calculations is extensive, this section will outline the types of costs and benefits.

- **Estimated Costs:**

- **Development Costs:**

- Personnel (analysts, designers, developers, testers – for a real project).
 - Software/Hardware (servers, databases, development tools, payment gateways).
 - Training costs for staff.
 - Consulting fees (if any).

- **Operational Costs (Recurring):**

- Maintenance (software updates, bug fixes).
 - Support staff.
 - Hosting fees (if cloud-based).
 - Licensing fees for software.

- **Anticipated Benefits:**

- **Tangible Benefits (Quantifiable):**

- Increased efficiency in order processing (faster turnover, fewer errors).
 - Reduced manual data entry for managers (e.g., if staff self-service for time recording is implemented).
 - Potential for increased sales through better menu management or online ordering capabilities.
 - Reduced food wastage through better tracking (if inventory features were added, though not explicit in core requirements).

- **Intangible Benefits:**

- Improved customer satisfaction (faster service, accurate orders, easier access to information).
 - Enhanced employee morale (easier access to schedules, efficient tools).
 - Better management decision-making through improved data access and reporting.
 - Improved data accuracy and consistency across the chain.
 - Professional image for the restaurant chain.

- **Potential Economic Risks:**

- **Cost Overruns:** Projects often exceed budget if scope is not well managed or estimates are inaccurate.
 - *Mitigation:* Thorough planning, clear scope definition, and iterative development to manage costs in phases.
- **Benefits Not Realized:** The anticipated benefits might not be achieved if the system is poorly designed or not adopted well by users.
 - *Mitigation:* User-centered design, effective training, and ensuring the system genuinely solves business problems.

6. Quantify Project Approval Factors

Time Estimate for the Project Completion

Subsystem	Functional Requirements	Iterations Required	Estimated Time
Restaurant & Employee Management	10	3	12 weeks
Menu & Inventory Management	12	3	12 weeks
Order Management (incl. POS features)	15	4	16 weeks
Reporting & Analytics	8	2	8 weeks
Total Development Time			20 weeks
System Integration		2	8 weeks
Total Project Time			28 weeks

Estimated Cost for Developing the Project and System

Expense category	Amount	Explanations
Salaries/Wages	\$196,000	Project manager: \$100/hr Developer: \$80/hr Coder: \$40/hr
Hardware	\$15,000	POS terminals, Server

Software Licenses	\$5,000	<ul style="list-style-type: none"> • Projectlibre • UMLet • Microsoft Word • POS system
Training	\$3,000	Training staff to know how to use the system
Facilities	\$7,000	The facility for operation
Contingency	\$22,600	For any unforeseen expenses
Total Estimated Development Cost	\$248,600	

Estimated Annual benefits for deploying the System

Benefit or Cost Saving	Amount	Explanation
Increased Order Processing Efficiency	\$75,000	Reduced staff time per order, faster table turnover in Fine Dining.
Reduction in Order Errors & Spoilage	\$40,000	Accurate order taking, better ingredient tracking for menu items
Optimized Employee Scheduling & Reduced Overtime	\$30,000	System records employee start/end times, manages roles
Improved Inventory Management (Reduced Waste)	\$25,000	Based on tracking ingredients for meals and desserts

Increased Sales through Data-Driven Menu Adjustments/Promos	\$50,000	Manager can modify menu items, reporting enables insights.
Total Estimated Annual Tangible Benefits	\$220,000	

7. Cost/Benefit Analysis

RMO Cost / Benefit Analysis for CSMS							
Category	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5	
1 Value of benefits	0	220,000	220,000	220,000	220,000	220,000	220,000
2 Development costs	-248,600	-2,000	-1,000	0	0	0	0
3 Annual expenses	0	20,000	20,000	20,000	20,000	20,000	20,000
4 Net benefit/costs	-248600	198,000	199,000	200,000	200,000	200,000	200,000
5 Discount factor (6%)	10,000	1	1	1	1	1	1
6 Net present value	-248600	186,793	177,510	167,920	158,420	149,460	
7 Cumulative NPV	-248600	-61,807	115,703	283,623	442,043	591,503	
8 Payback period		Year 2					

8. Requirements Analysis and Modelling

8.1. UML Activity Diagram

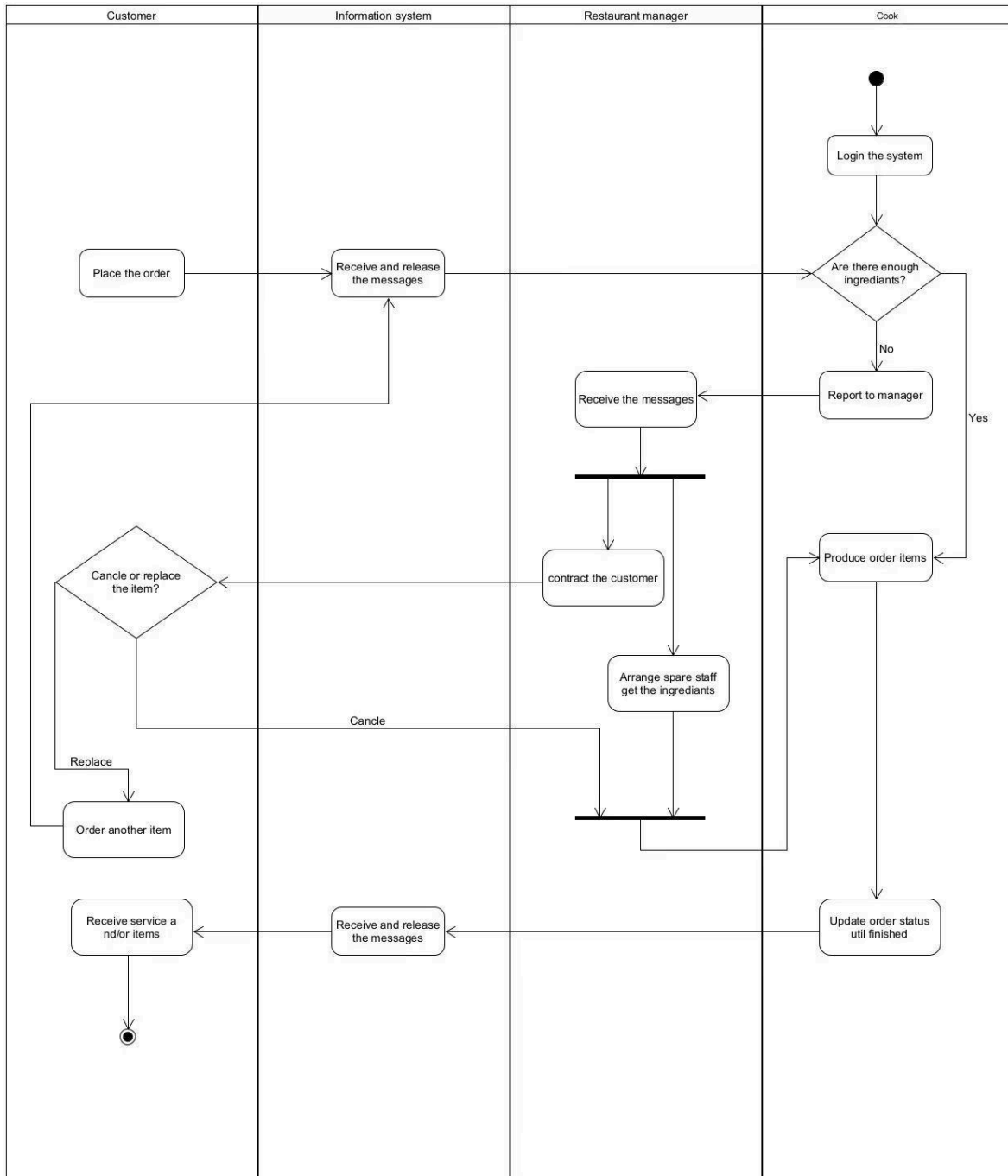


Figure 1: UML Activity Diagram for Cook

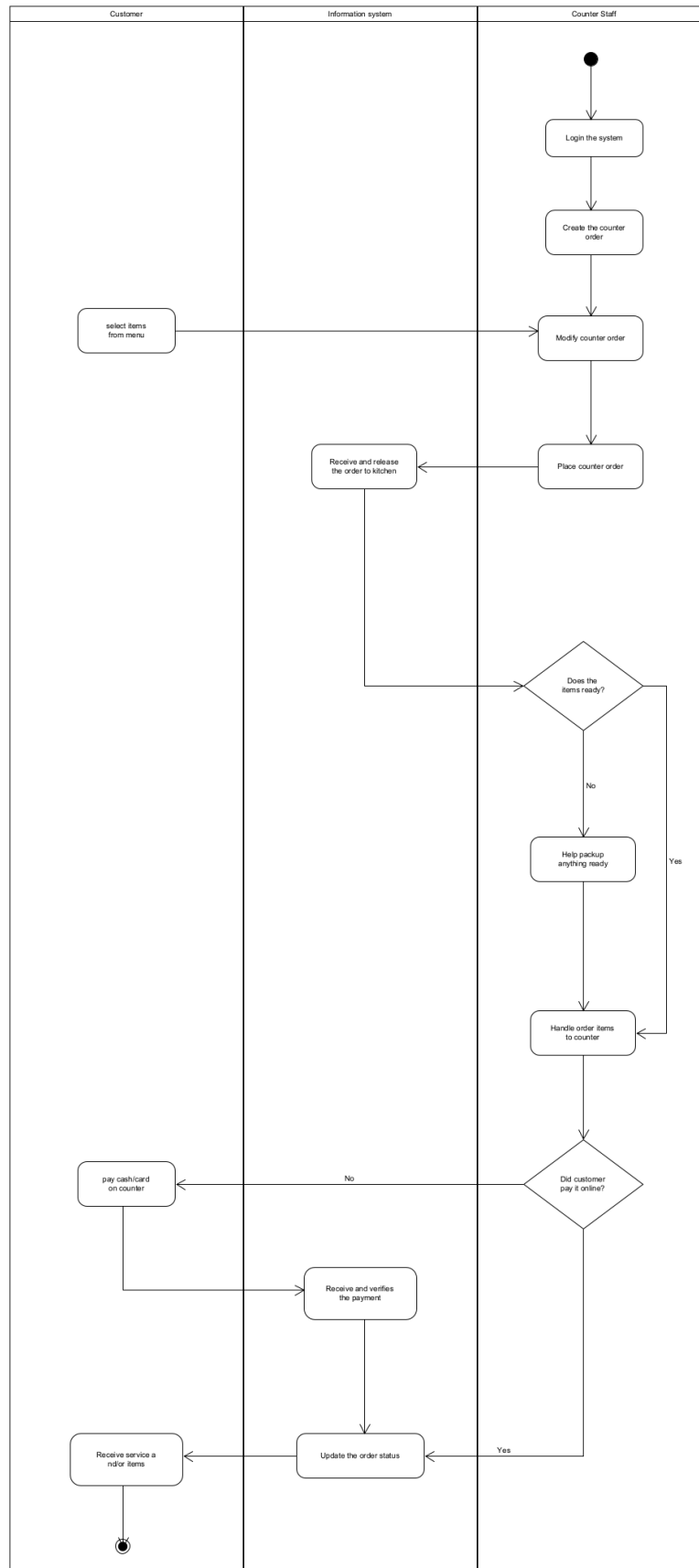


Figure 2: UML Activity Diagram of Counter Staff

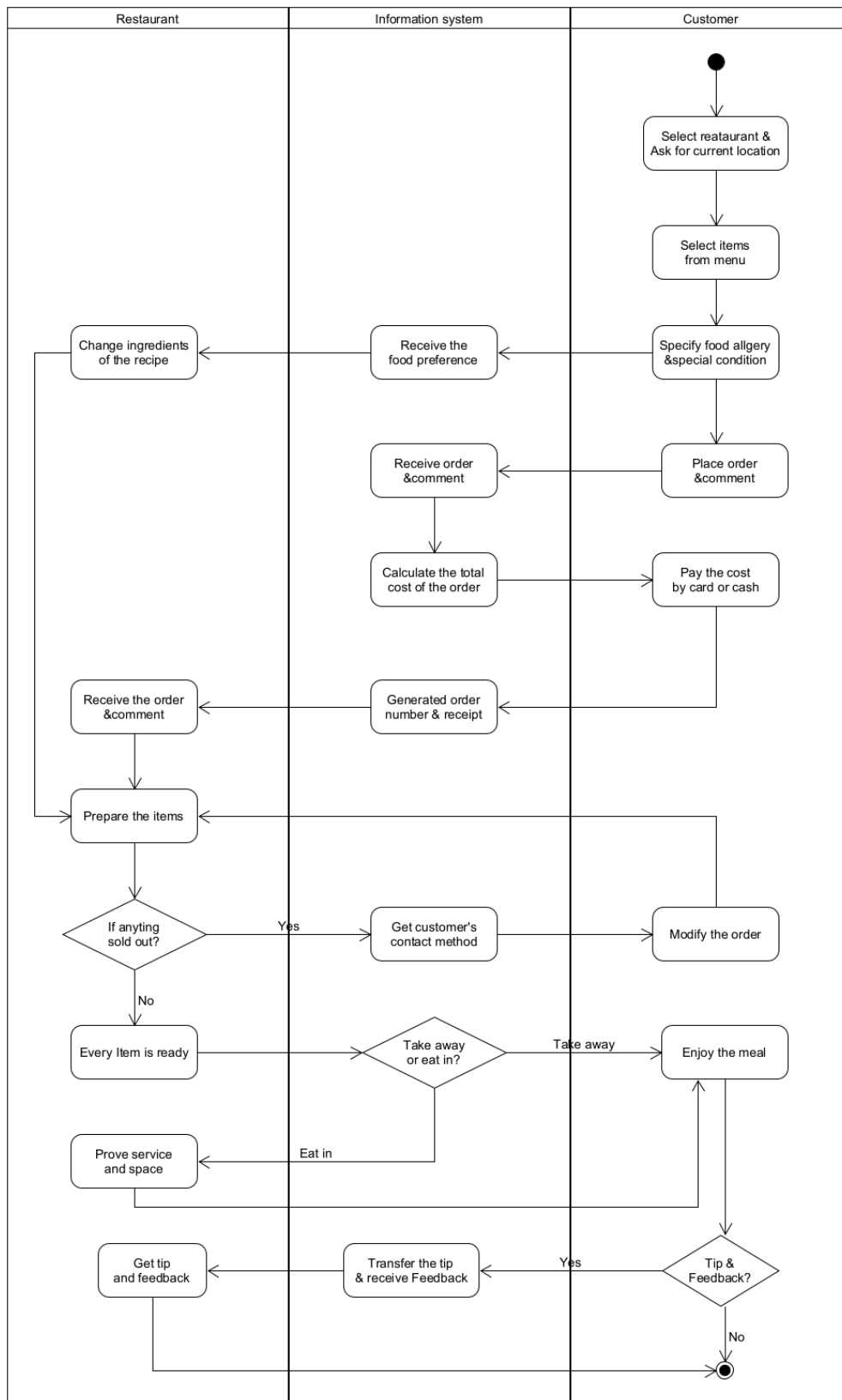


Figure 3: UML Activity Diagram of Customer

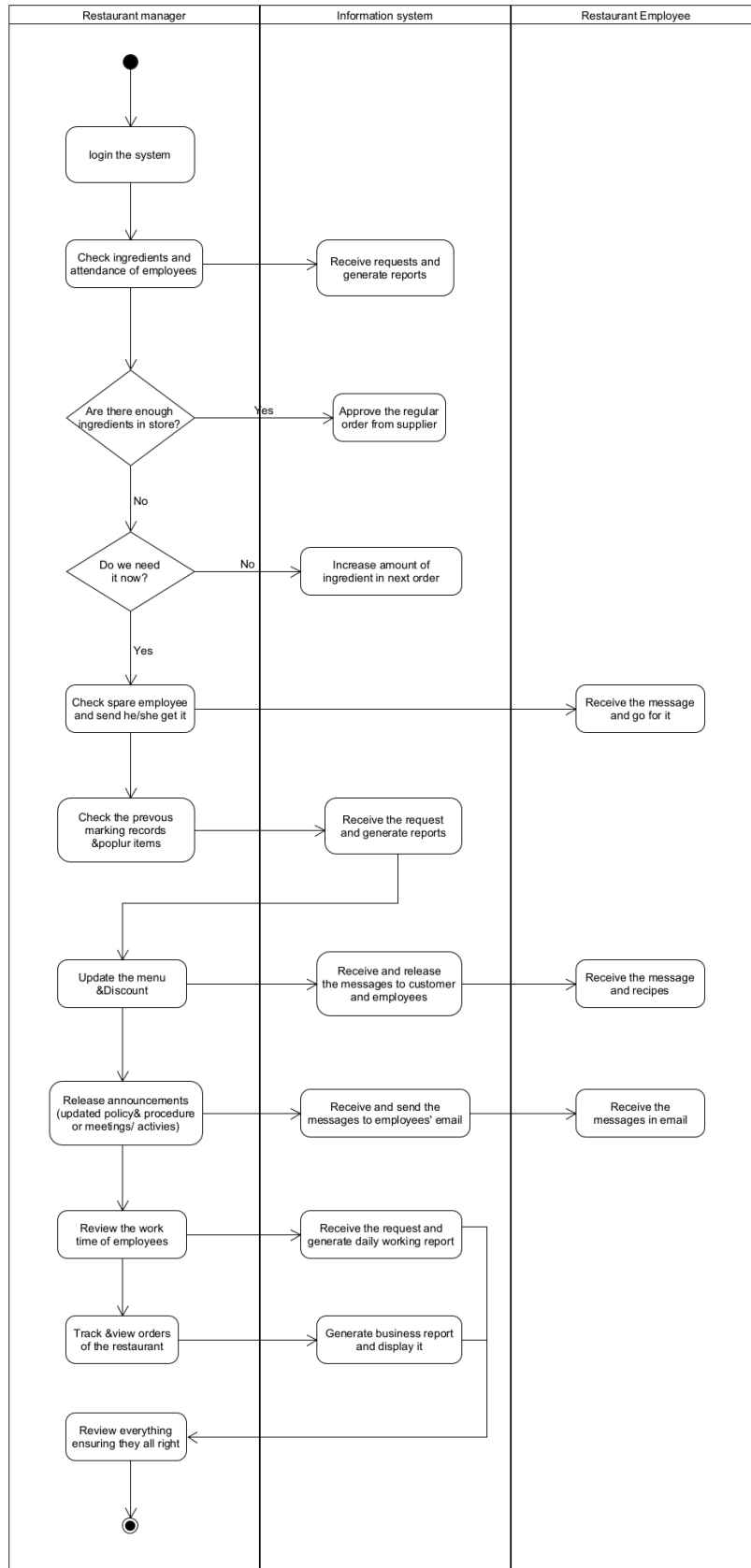


Figure 4: UML Activity Diagram of Restaurant Manager

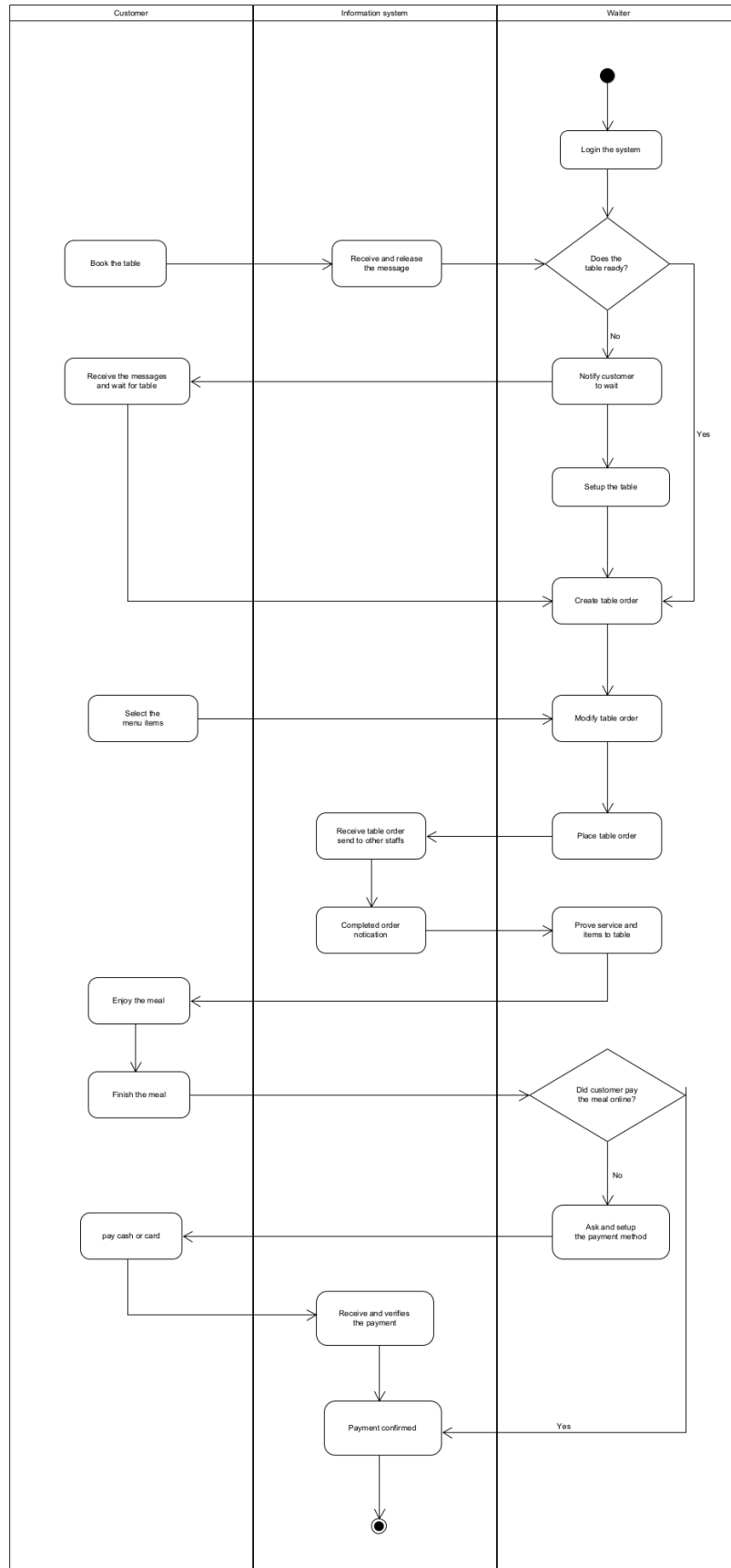


Figure 5: UML Activity diagram of Waiter

8.2. Use Case and UML Use Case Diagrams

8.2.1. The use case diagram :



Actors :

- Customer
- Restaurant manager
- Cook
- Waiter
- Counter Staff

8.2.2. User Goal

User (Actor)	User goal and resulting use case
Customer	Search for menu item Place self-service order Make self-service payment View order history Provide feedback
Restaurant Manager	Manage menu items Manage employee info Manage restaurant info Record employee work times View restaurant order history Track order status Manage ingredients View work schedule Generate sales reports Login to system
Waiter	View menu items Create table order Modify table order Confirm dine-in order Serve order Process table payment
Counter Staff	Create counter order Modify counter order Place counter order Process counter payment
Cook	Update order status

8.2.3. Use Case:

Use case	Actor	Use case description
Search Menu Item	Customer	Customers can search for menu items by entering keywords or selecting categories.
Place self-service Order	Customer	Customers select desired menu items, specify quantities, and submit the order. The system records the order, assigns an order number, and notifies the customer.
Make self-service Payment	Customer	After placing an order, customers choose a payment method (e.g., credit card, digital wallet) and complete the payment. The system processes the transaction and updates the order status.
View Order History	Customer	The customers who login can view a list of their past orders, including order dates, items ordered, total costs, and order statuses.

Provide feedback	Customer	Customers can submit feedback about their experience. The system stores it for restaurant management to review
Manage Menu Items	Restaurant Manager	Manager can add , updates, delete menus items, such as item name, price.
Manage Restaurant Info	Restaurant Manager	Manager can update restaurant details, such as location and name
Record Employee Work Time	Restaurant Manager	Manager can record the start time and end time of working of Employee, provide clear working period
View Restaurant Order History	Restaurant Manager	Enables the Restaurant Manager to access and review a comprehensive list of past orders for their restaurant, with capabilities to filter and search for specific transactions.
Manage Ingredients	Restaurant Manager	Enables the Restaurant Manager to define and manage the list of ingredients used in meals and desserts, including their names and potentially stock levels or suppliers.

Track Order Status	Restaurant Manager	Allows the Restaurant Manager to monitor the current status of active orders within their restaurant
Manage Employee Info	Restaurant Manager	Enable the Restaurant Manager to create, update or delete employee information records
View work schedule	Restaurant Manager	Allows the Restaurant Manager to view the work schedules for all employees in their restaurant, and potentially for themselves.
Generate Sales Reports	Restaurant Manager	Enables the Restaurant Manager to produce sales for further business analysis
Login to System	Restaurant Manager	Restaurant Manager can securely access their management functionalities by providing their credentials.
Manage Work Schedule	Restaurant Manager	Restaurant Manager can manage and modify employee's work schedule
View Menu Items	Waiter/ Counter Staff/ Restaurant Manager	Enables the Waiter, Restaurant Manager or Counter Staff to browse, search, and view details of menu items, which could assist customers with their selections or answer questions.

Create table order	Waiter	Waiter create an order for a specific table by selecting menu items and quantities as requested by the customers seated at that table.
Modify table order	Waiter	Before confirming, waiter can modify the table order by adding or removing items or adjusting quantities or adding in allergy notes
Confirm Table order	Waiter	Waiter confirm the table order, sending it to kitchen staff
Mark Order as Served	Waiter	After the kitchen has prepared the order, waiter retrieve the items and serve them to the corresponding table.
Ask Payment Type (included by “process payment”)		Waiter or counter staff ask customer the type of payment preferred, cash or card
Process Payment	Waiter / Counter Staff	Waiter or counter staff process customer’s payment

Create Counter order	Counter Staff	Enables the Counter Staff to initiate and build a new order for a customer who is present at the counter. This involves selecting menu items based on the customer's requests, specifying quantities, and noting any available customizations. The order is assembled in the system but not yet finalized or sent for preparation.
Modify Counter Order	Counter Staff	Allows the Counter Staff to make changes to an existing order that has been created but not yet finalized or paid for. Modifications can include adding new items, removing items, changing quantities, or altering customizations.
Confirm Counter Order	Counter Staff	Enables the Counter Staff to finalize an order that has been created (and possibly modified) at the counter. This action confirms the customer's selections and typically submits the order to the kitchen or preparation area.

Ask Payment Type (included by “Process payment”)	Counter Staff	Counter staff or waiter ask customer the type of payment, by cash or card
Process Payment	Counter Staff	Waiter or counter staff process customer’s payment
View Menu Items	Counter Staff	Counter staff or waiter can browse menu to assist customers
Update Order Status	Cook	Cook could update the order status, confirm the dish is ready

8.2.4. Fully developed Use case description

· Search Menu Item

Customers can search for menu items by entering keywords or selecting categories. The system displays matching items with details like name, description, price, and availability.

Use case name:	Search menu item
Scenario:	Customer finds products from menu
Triggering event:	Customer selected restaurant and searches items from menu
Brief description:	Online customer wants to find product from menu.
Actors:	Customer
Related use cases:	None
Stakeholders:	Marketing team, restaurant manager

Preconditions:	<p>Restaurant and menu information must be available.</p> <p>User must be classified as customer or visitor by system.</p>	
Postconditions:	<p>One or more items must be selected.</p> <p>One or more restaurants must be selected.</p>	
Flow of activities:	Actor	System
	<ol style="list-style-type: none"> 1. Customer indicated desire to order the items from practical restaurant. 2. Customer chooses the restaurant 3. Customer searches items 4. Customer selects the item and quantity of item 	<ol style="list-style-type: none"> 1.1 System starts display as default mode for visitor or receives customer login 2.1 System provides list of restaurants based on customer's current location, previous record or user input 2.2 System displays list of restaurants based on distance/past order record or user inputs 3.1 System receives customer's option or input 3.2 System displays the menu items of selected restaurant. 4.1 System creates chart to save the items

		<p>4.2 System calculates the total price of items in chart</p> <p>4.3 System displays the results and selected items information</p>
Exception conditions:	<p>1.1 Error of user inputs</p> <p>2.1 Restaurant is not trading</p> <p>2.2 Menu update/changes</p> <p>3.1 Item does not exist</p> <p>3.1 input error</p> <p>3.2 The number of items is not valid (less than zero)</p>	

· Place self-service Order

Customers select desired menu items, specify quantities, and submit the order. The system records the order, assigns an order number, and notifies the customer.

Use case name:	Place self-service order
Scenario:	Customer decides to order selected items from a menu.
Triggering event:	Customer selects items from the menu and decides to place an order.
Brief description:	After browsing the menu, the customer selects items to order from a specific restaurant.

Actors:	Customer	
Related use cases:	Search menu item Pay self-service order	
Stakeholders:	Customer Cook Waiter (if eat in fine dining)	
Preconditions:	Menu and restaurant info must be available. User must be identified (visitor or logged-in customer).	
Postconditions:	Selected items and restaurant details are recorded as an order. Order details are forwarded for payment and processing.	
Flow of activities:	Actor	System
	1. Customer selects menu items and adds them to the cart 2. Customer makes a reservation (dine-in), selects pick-up, or enters delivery address	1.1 System displays the list of selected items and their total price 2.1 System receives the selected order method 2.2 System returns order details and prepares for payment
Exception conditions:	1.1 Selected items are unavailable or sold out 1.1 Invalid quantity selected 1.1,2.1&2.2 Restaurant is closed	

Make self-service Payment

After placing an order, customers choose a payment method (e.g., credit card, digital wallet) and complete the payment. The system processes the transaction and updates the order status.

Use case name:	Make self-service payment	
Scenario:	Customer proceeds to payment after placing an order.	
Triggering event:	Customer clicks to pay and submits payment information.	
Brief description:	Customer provides payment details to finalize the order.	
Actors:	Customer Financial system	
Related use cases:	Place Self-Service Order	
Stakeholders:	Customer Restaurant financial team Waiter team	
Preconditions:	Order has been placed and contains valid items. Credit/debit card services are available.	
Postconditions:	Payment is authorized and order is finalized. Payment details are saved in the system.	
Flow of activities:	Actor	System
	1. Customer selects "Pay Now" and chooses credit/debit card	1.1 System creates checkout page 1.2 System verifies payment authorization 1.3 System confirms and stores payment details

Exception conditions:	1.2 Invalid card details 1.2 Payment declined 1.2 Authorization service unavailable	

View Order History

Logged-in customers can view a list of their past orders, including order dates, items ordered, total costs, and order statuses.

Use case name:	View order history	
Scenario:	Customer wants to review their past orders	
Triggering event:	Customer accesses their profile/order section	
Brief description:	Customer views a list of previously placed orders along with details such as items, prices, and order dates.	
Actors:	Customer	
Related use cases:	Place order Pay order	
Stakeholders:	Customer, Restaurant manager and Marketing team	
Preconditions:	Customer must be logged in	
Postconditions:	Order history is retrieved and displayed	
Flow of activities:	Actor	System

	1. Customer logs into the system 2. Customer navigates to order history	1.1 System verifies login credentials 2.1 System displays the lists of previous order of the customer account
Exception conditions:	1.1 Customer not logged in 2.1 No order history available	

- Provide feedback

Customers can submit feedback about their experience or specific orders. The system stores the feedback for restaurant management to review.

Use case name:	Provide Feedback
Scenario:	Customer provides feedback on a completed order
Triggering event:	Customer views past order and chooses to leave feedback
Brief description:	Customer submits comments or ratings about their dining experience or delivery.
Actors:	Customer
Related use cases:	Place order Pay order View order history
Stakeholders:	Customer, Restaurant manager and Marketing team

Preconditions:	Customer must be logged in and have completed at least one order	
Postconditions:	Feedback is saved and sent to relevant stakeholders	
Flow of activities:	Actor	System
	1. Customer accesses order history 2. Customer selects an order to review 3. Customer submits feedback	1.1 System verifies login credentials 1.2 System displays order history 2.1 System shows the feedback form 3.1 System stores feedback 3.2 System notifies manager
Exception conditions:	1.1 Customer not logged in 2.1 Feedback form not loading 3.1 Invalid input	

Restaurant Manager Actor Use Case:

- Manage Menu Item

Authorized staff can add new menu items, update existing ones (e.g., change price or description), or remove items from the menu.

Use case name:	Modify menu items	
Scenario:	Restaurant manager makes changes and /or promotion for items in menu.	
Triggering event:	Restaurant manager adds/deletes items from current menu	
Brief description:	Restaurant manager tries to make discount and changes to menu items.	
Actors:	Restaurant manager	
Related use cases:	System record menu item	
Stakeholders:	Restaurant manager, marketing team, cook, customer	
Preconditions:	Marketing team decides promotion for certain items Season affects certain material of menu items	
Postconditions:	System updates the menu items and/or prices Cook starts makes new items if there are new produces Customers getting discount/promotion through the system online	
Flow of activities:	Actor	System

	<p>1. Restaurant manager indicates idea to make the changes of items on menu.</p> <p>2. Restaurant manager selects the menu/item.</p> <p>3. Restaurant manager applies the changes on items</p> <p>4. Restaurant manager saves the changes.</p>	<p>1.1 System identifies the restaurant manager</p> <p>1.2 System displays the lists of items information of current menu.</p> <p>2.1 System displays the details of selected item/menu</p> <p>2.2 System creates an editor page for manager</p> <p>3.1 System options for manager modify the menu</p> <p>3.2 System receives the changes from manager</p> <p>4.1 System save the changes</p> <p>4.2 System displays the updated information on menu</p>
Exception conditions:	<p>1.1 User does not have permission for making changes</p> <p>2.1 Invalid category/menu page</p> <p>3.2 Invalid input from restaurant manager</p> <p>4.2 Update information delay</p>	

Manage Employee Info

Managers or HR staff can add new employees, update employee details (e.g., contact information, role), or deactivate employee accounts.

Use case name:	Manage employee information	
Scenario:	Restaurant manager maintains employee information	
Triggering event:	Manager adds, edits, or deletes employee records	
Brief description:	Restaurant manager accesses the system to manage employee details such as employee's unique number at a restaurant, full name, hire date, end date (optional) and salary.	
Actors:	Restaurant manager	
Related use cases:	Record employee work time	
Stakeholders:	HR team, accounting team, restaurant manager, restaurant employees	
Preconditions:	<p>New employee is hired by restaurant</p> <p>Restaurant manager set the employee as staff of the restaurant</p> <p>Manager must be logged in and authorized</p>	
Postconditions:	<p>Updated employee info is stored in the system</p> <p>HR team knows the employee's detail information</p> <p>Accounting team can calculate employee's salary</p> <p>HR team knows how many years of experience manager and employees have</p>	
Flow of activities:	Actor	System

	1. Manager logs in to the system 2. Manager selects an employee record or chooses to add new 3. Employee updates information if there are any changes	1.1 System verifies the user 2.1 System displays the employee's information page 2.2 System receives the manager's inputs 2.3 System displays the list of employees or new entry form 2.4 System gets confirmation 3.1 System displays the information 3.2 System gives option to change details 3.3 System obtains request and saves the information
Exception conditions:	1.1 User permission is not valid 3.2 User inputs error 2.1 Missing confirmation	

· Manage Restaurant Info

Authorized personnel can update restaurant information such as address, phone number, operating hours, or seating capacity.

Use case name:	Manage restaurant information
----------------	-------------------------------

Scenario:	Modifying restaurants' information	
Triggering event:	Manager updates restaurant's information	
Brief description:	Every restaurant manager can update his/her restaurant's information if there are any changes.	
Actors:	Restaurant manager	
Related use cases:	None	
Stakeholders:	Restaurant manager, marketing	
Preconditions:	<p>Restaurant does exist in real world</p> <p>Restaurant has one restaurant manager and a few cooks, waiters, and dishwashers.</p> <p>Restaurant can be described by name, location (city, street, building number), phone number and restaurant type</p>	
Postconditions:	<p>Restaurant manager updates its menu items</p> <p>Online customer can place order from added restaurant</p>	
Flow of activities:	Actor	System

	1. Restaurant manager opens restaurant's information and updates information 2. Restaurant manager review and save the changes	1.1 System displays the information 1.2 System verifies user's permission 1.3 System displays editor page 1.4 System receives the inputs 2.1 System displays the changes 2.2 System receives the confirmation 2.3 System saves the changes into databased
Exception conditions:	1.2 User permission error 1.4 Input error	

· Record Employee Work Times

Allows the Restaurant Manager to manually enter, review, or approve the start and end times of work periods for employees, ensuring accurate recording of work hours.

Use case name:	Record employee work time
Scenario:	Record employees' working hours
Triggering event:	Staff needs to log in and log off the system to record their working hours

Brief description:	Salaries of staff working in restaurants are based on their experience and/or hours of work. Restaurant manager and financial team needs a clear record of employees' working hours.	
Actors:	Restaurant manager	
Related use cases:	Arrange Employees' Work Schedule	
Stakeholders:	Restaurant manager Restaurant employees HR team Accounting team	
Preconditions:	Restaurant staff are located in a restaurant and have started work. The restaurant manager hired at least one staff in restaurant Accounting team needs to calculate employees' salaries with evidence.	
Postconditions:	Restaurant manager can arrange shift timetable HR team will have better understanding of each employee Accounting team can calculate employees' salary based on work hours Restaurant employees will be paid fair and properly.	
Flow of activities:	Actor	System

	1. Employee arrives the restaurant he/she worked login the information system with their unique employee number 2. Restaurant manager logs in and clicks the “attendance” on dashboard 3. Employees sign off from device	1.1 System receives the login requests from employees 1.2 System records starting time of each employee 2.1 System verifies the manager 2.2 System displays the existing staffs on screen 3.1 System records the ending time.
Exception conditions:	1.1 Invalid employee number 2.1 Invalid user permission 2.2 Employee forget log into system 3.1 Employee forget to sign out	

· View Restaurant Order History

Enables the Restaurant Manager to access and review a comprehensive list of past orders for their restaurant, with capabilities to filter and search for specific transactions.

Use case name:	View order history
Scenario:	Restaurant manager checks past orders
Triggering event:	Restaurant manager selects “View History”
Brief description:	Restaurant manager reviews all orders for performance analysis

Actors:	Restaurant manager	
Related use cases:	Record Completed Order	
Stakeholders:	Marketing and Financial team	
Preconditions:	Order data exists	
Postconditions:	Previous orders data were displayed on screen	
Flow of activities:	Actor	System
	1. Restaurant manager logs in and navigates to history. 2. Restaurant manager selects a period for practical record	1.1 System identifies the restaurant manager 1.2 System retrieves past orders 2.1 System filters and shows results
Exception conditions:	1.1 User does not have permission for making changes 1.2 No records found	

Tracking Order Status

Allows the Restaurant Manager to monitor the current status of active orders within their restaurant

Use case name:	Track Order Status
Scenario:	Monitor real-time progress of orders
Triggering event:	Order is placed
Brief description:	Restaurant manager can follow status of all orders

Actors:	Restaurant manager	
Related use cases:	None	
Stakeholders:	Waiter, counter staff, cook, customer	
Preconditions:	Active orders must exist	
Postconditions:	Status changes shown in dashboard	
Flow of activities:	Actor	System
	1. Restaurant manager logs in and opens order dashboard 2. Manager reviews order status	1.1 System identifies the restaurant manager 1.2 System lists current orders 2.1 System updates status in real time
Exception conditions:	1.1 User does not have permission for making changes 1.2&2.1 System sync delay 1.2&2.1 Incorrect status reported	

· Manage ingredients

Enables the Restaurant Manager to define and manage the list of ingredients used in meals and desserts, including their names and potentially stock levels or suppliers.

Use case name:	Manage Ingredients
Scenario:	Monitor and update kitchen inventory

Triggering event:	Manager selects inventory tab	
Brief description:	Restaurant manager tries to keep ingredients up-to-date for operational needs	
Actors:	Restaurant manager	
Related use cases:	Modify Menu Items	
Stakeholders:	Cook, Kitchen Staffs	
Preconditions:	Ingredient database is initialized Cook updates the usage of materials	
Postconditions:	Ingredients updated Manager order for ingredients out of stock	
Flow of activities:	Actor	System
	1. Restaurant manager logs in to inventory. 2. Manager adds/edits/removes entries	1.1 System identifies the restaurant manager 1.2 System loads ingredient list 2.1 System saves and updates stock
Exception conditions:	1.1 User does not have permission for making changes 2.1 Invalid quantity 2.1 Missing fields	

· Arrange employees' work schedule

Allows the Restaurant Manager to view the work schedules for all employees in their restaurant, and potentially for themselves.

Use case name:	Arrange employees' work schedule	
Scenario:	Manager assigns shifts to employees	
Triggering event:	Manager accesses schedule panel	
Brief description:	Enables managers to define everyone's work and shifts of day	
Actors:	Restaurant manager	
Related use cases:	Record Employee Work Time	
Stakeholders:	HR team, Restaurant employees	
Preconditions:	Employee list exists Employee provides email address	
Postconditions:	Schedule saved and sent to employees' email	
Flow of activities:	Actor	System
	1. Manager opens scheduling tool 2. Manager assigns shifts	1.1 System identifies the restaurant manager 1.2 System loads employee availability 2.1 System saves and distributes schedule
Exception conditions:	1.1 User does not have permission for making changes 1.2 Overlapping shifts 2.1 Incomplete assignments	

--	--

- Generate Sales Reports

Enables the Restaurant Manager to produce reports summarizing sales data over various periods, by item, or other criteria to analyze business performance.

- Login to System

Allows the Restaurant Manager to securely access their management functionalities by providing their credentials.

Use case name:	Login to System	
Scenario:	Restaurant staff log in to access system features	
Triggering event:	Employee opens login interface	
Brief description:	All staff (manager, cook, waiter and counter staff) log into the system to access their tools and responsibilities.	
Actors:	Restaurant manager, cook, waiter and counter staff	
Related use cases:	All staff-related use cases	
Stakeholders:	All restaurant staffs	
Preconditions:	User must have a valid employee number	
Postconditions:	User is authenticated and granted access with their user permission	
Flow of activities:	Actor	System

	1. Employee enters unique employee number into devices 2. Login successful	1.1 System verifies login credentials 2.1 System displays dashboard based on account's permission
Exception conditions:	1.1 Invalid employee number 1.1 Account inactive or locked	

Waiter Actor Use case:

· View Menu Items

Enables the Waiter or Counter Staff to browse, search, and view details of menu items, often to assist customers with their selections or answer questions.

Use case name:	View menu items
Scenario:	Waiter wants to check available items
Triggering event:	Waiter accesses menu view page
Brief description:	Displays up-to-date list of items available for ordering
Actors:	Waiter
Related use cases:	Create table order
Stakeholders:	Waiter, Customer, Cook

Preconditions:	Menu must be loaded in the system	
Postconditions:	Waiter is informed of item availability	
Flow of activities:	Actor	System
	1. Waiter logs in 2. Waiter selects “View Menu”	1.1 System identifies the waiter 1.2 System shows waiter’s dashboard 2.1 System displays categorized menu list
Exception conditions:	1.1 Invalid user 2.1 Menu unavailable 2.1 Item no longer exists	

· Create table order

Waitstaff create an order for a specific table by selecting menu items and quantities as requested by the customers seated at that table.

Use case name:	Create table order
Scenario:	Customer at a table is ready to order
Triggering event:	Waiter inputs new order
Brief description:	Waiter starts entering a new table order
Actors:	Waiter

Related use cases:	View Menu Items Modify Table Order Confirm Table Order	
Stakeholders:	Customer, Cook	
Preconditions:	Table assigned and menu available	
Postconditions:	Draft order is created	
Flow of activities:	Actor	System
	1. Waiter selects table 2. Waiter selects items and inputs notes	1.1 System identifies the waiter 1.2 System displays table-specific order screen 2.1 System saves draft order
Exception conditions:	1.1 Invalid user 2.1 Item not available 2.1 Invalid quantity	

· Modify table order

Before confirming, waitstaff can modify the table order by adding or removing items or adjusting quantities based on customer requests.

Use case name:	Modify table order
Scenario:	Customer requests a change before finalizing

Triggering event:	Waiter selects existing draft order	
Brief description:	Waiter can edit item list and notes	
Actors:	Waiter	
Related use cases:	View Menu Items Create Table Order Confirm Table Order	
Stakeholders:	Customer, Cook	
Preconditions:	Draft order must exist	
Postconditions:	Updated draft saved	
Flow of activities:	Actor	System
	1. Waiter retrieves existing order 2. Waiter updates details	1.1 System identifies the waiter 1.2 System displays table-specific draft order 2.1 System saves draft updated
Exception conditions:	1.1 Invalid user 2.1 Order already finalized	

· Confirm dine-in order

Once the table order is complete, waitstaff confirm it, which sends the order to the kitchen for preparation.

Use case name:	Confirm table order	
Scenario:	Customer confirms final order	
Triggering event:	Waiter clicks "Confirm"	
Brief description:	Order is finalized and sent to kitchen	
Actors:	Waiter	
Related use cases:	View Menu Items Create Table Order Modify Table Order	
Stakeholders:	Customer, Cook	
Preconditions:	Draft order must exist	
Postconditions:	Updated draft saved	
Flow of activities:	Actor	System
	1. Waiter confirms order 2. Waiter submits	1.1 System identifies the waiter 1.2 System assigns order number 2.1 System sends to kitchen
Exception conditions:	1.1 Invalid user 1.2 Invalid data 2.1 Kitchen not accepting orders	

· Serve order

After the kitchen has prepared the order, waitstaff retrieve the items and serve them to the corresponding table.

Process Table Payment

Waitstaff process the payment for a table's order, handling payment methods, splitting bills if necessary, and issuing receipts.

Use case name:	Process payment	
Scenario:	Staff processes payment using the chosen method	
Triggering event:	Payment method is selected	
Brief description:	System processes the payment, validates it, and issues confirmation.	
Actors:	Counter Staff Waiter	
Related use cases:	Select payment type	
Stakeholders:	Restaurant manager, Customer, Financial Team	
Preconditions:	Payment method selected	
Postconditions:	Payment is confirmed and recorded	
Flow of activities:	Actor	System
	1. Staff enters payment details 2. Payment successful	1.1 System identifies the staff 1.2 System validates and processes payment 2.1 System confirms and stores transaction

Exception conditions:	1.1,1.2&2.1 System error 1.2 Payment declined 1.2 Incorrect amount	

Counter Staff Actor use cases

· Create Counter Order

Enables the Counter Staff to initiate and build a new order for a customer who is present at the counter. This involves selecting menu items based on the customer's requests, specifying quantities, and noting any available customizations. The order is assembled in the system but not yet finalized or sent for preparation.

Use case name:	Create counter order
Scenario:	Customer approaches the counter to place a new order.
Triggering event:	Counter staff selects “New Order” on system.
Brief description:	Counter staff begins a new order entry by selecting the restaurant, menu items, and customer preferences.
Actors:	Modify Counter Order, Place Counter Order
Related use cases:	Modify Counter Order Place Counter Order
Stakeholders:	Customer, Cook, Restaurant Manager

Preconditions:	Menu and restaurant must be set up in the system.	
Postconditions:	Draft order is created and stored temporarily.	
Flow of activities:	Actor	System
	1. Counter staff logs in and initiates a new order 2. Staff selects restaurant and items 3. Staff inputs quantity and notes	1.1 System identifies the counter staff 1.2 System loads ordering interface 2.1 System displays menu items and accepts item selection 3.1 System records draft order with timestamp
Exception conditions:	1.1 Invalid user 2.1 Invalid item selected 3.1 Invalid user input	

· Modify Counter Order

Allows the Counter Staff to make changes to an existing order that has been created but not yet finalized or paid for. Modifications can include adding new items, removing items, changing quantities, or altering customizations.

Use case name:	Modify counter order
Scenario:	Counter staff updates an order before it's finalized.

Triggering event:	Staff opens an existing draft order.	
Brief description:	Changes can be made to the current counter order before submission.	
Actors:	Counter Staff	
Related use cases:	Create Counter Order Place Counter Order	
Stakeholders:	Cook, customer	
Preconditions:	Draft counter order exists	
Postconditions:	Order is updated with changes	
Flow of activities:	Actor	System
	1. Staff accesses an existing draft order 2. Staff updates item list and/or quantities 3. Staff saves changes	1.1 System identifies the counter staff 1.2 System displays the order details 2.1 System validates and applies updates 3.1 System stores updated draft
Exception conditions:	1.1 Invalid user 1.2 Draft order not existed 2.1 Invalid user inputs	

Place Counter Order

Enables the Counter Staff to finalize an order that has been created (and possibly modified) at the counter. This action confirms the customer's selections and typically submits the order to the kitchen or preparation area.

Use case name:	Place counter order	
Scenario:	Counter staff confirms and submits order to kitchen	
Triggering event:	Staff clicks “Place Order”	
Brief description:	The final order is submitted and marked as ready for preparation.	
Actors:	Counter staff	
Related use cases:	Create counter order Modify counter order	
Stakeholders:	Restaurant manager, cook, customer	
Preconditions:	Draft order exists	
Postconditions:	Order is locked and send to kitchen	
Flow of activities:	Actor	System
	1. Staff reviews and confirms order 2. Staff submits the order	1.1 System identifies the staff 1.2 System displays confirmation prompt 2.1 System assigns order number and routes to kitchen
Exception conditions:	1.1 Invalid user 1.2 & 2.1 Missing or invalid details 2.1 Kitchen system offline	

--	--

Select Payment Type

This is typically a step within a payment processing use case rather than a standalone use case initiated by the Counter Staff as a distinct goal. It refers to the interaction where the Counter Staff inquires about the customer's preferred method of payment (e.g., cash, card) before proceeding with the transaction.

Use case name:	Select payment type	
Scenario:	Customer is ready to pay	
Triggering event:	Customer decides how to pay	
Brief description:	Waiter/ counter staff helps the customer select a payment method (e.g., cash, card, mobile)	
Actors:	Counter Staff Waiter	
Related use cases:	Process payment	
Stakeholders:	Restaurant manager, customer, Financial Team	
Preconditions:	Order must be finalized and ready for payment	
Postconditions:	Payment type is selected and ready for processing Valid payment confirmation	
Flow of activities:	Actor	System

	1. Staff initiates payment 2. Staff selects payment type	1.1 System identifies the staff 1.2 System displays payment method options 2.1 System records payment method
Exception conditions:	1.1 Invalid user 1.1 Input error 1.2 No valid payment methods available	

Process payment

If it refers to the overall handling of an order by the Counter Staff from creation to payment, it might be an overarching business process supported by several use cases

Use case name:	Process payment
Scenario:	Staff processes payment using the chosen method
Triggering event:	Payment method is selected
Brief description:	System processes the payment, validates it, and issues confirmation.
Actors:	Counter Staff Waiter
Related use cases:	Select payment type

Stakeholders:	Restaurant manager, Customer, Financial Team	
Preconditions:	Payment method selected	
Postconditions:	Payment is confirmed and recorded	
Flow of activities:	Actor	System
	1. Staff enters payment details 2. Payment successful	1.1 System identifies the staff 1.2 System validates and processes payment 2.1 System confirms and stores transaction
Exception conditions:	1.1,1.2&2.1 System error 1.2 Payment declined 1.2 Incorrect amount	

Chef Actor use case

· Process Order

Enables the Cook to manage an active food order by interacting with the system to update its status. This includes indicating when an order (or parts of it) is "Ready" for serving or pickup, or flagging

specific items within an order as "Out of Stock." These system updates would then trigger appropriate notifications or subsequent actions

Use case name:	Process order	
Scenario:	Cook prepares the food based on orders	
Triggering event:	Order is received by kitchen	
Brief description:	Cook views the order details and begins preparing the items	
Actors:	Cook	
Related use cases:	Place order	
Stakeholders:	Customer, Waiter, Counter Staff and Restaurant Manager	
Preconditions:	Order is confirmed and routed to kitchen	
Postconditions:	Order is marked as ready or completed	
Flow of activities:	Actor	System
	1. Cook logs into kitchen terminal. 2. Cook selects an order to prepare. 3. Cook finishes preparing order	1.1 System identifies the chef 1.2 System displays the lists of active orders 2.1 System shows detailed items 3.1 System updates order status to "Ready"

Exception conditions:	<ul style="list-style-type: none">1.1 Invalid user2.1 Order not found2.1 Items missing or unclear3.1 Kitchen device error
-----------------------	--

8.3. Domain Model class diagram

Class “Restaurant”

Attribute:

- Name{key1}: The restaurant is identified by name
- City{key2}: the city where restaurant locate in (location)
- phoneNumber{key}
- Street:the street name (location)
- buildingNumber
- restaurantType

Class “Customer”

Attribute:

- customerName{key}
- phoneNumber
- email

Class “ Employee”

Attribute:

- employeeNumber{key}
- fullName
- hireDate
- endDate
- Salary
- roleType

Class “Manager” (subclass of Employee):

Attribute:

- yearsOfExperience

Class “Cook” (subclass of Employee)

Attribute:

- yearsOfExperience

Class “Waiter” (subclass of Employee)

Class “ Dishwasher” (subclass of Employee)

Class “ EmployeeWorkTimeLog”

Attribute:

- workLogID{key}
- workDate
- startTime
- endTime

Class “Menu”

Attribute:

- menuID{key}
- menuName

Class “MenuItem”

Attribute:

- itemNumber{key}
- itemName
- Price
- category

Class “ingredients”(association class between “Menu” and “MenuItem”)

Attributes:

- ingredientName
- amount

Class “Meal”(subclass of MenuItem)

Class “Dessert" (Subclass of MenuItem)

Class “ Drink” (Subclass of MenuItem)

Attribute:

- alcoholPercentage: the percentage of alcohol in the wine

Class “Order”

Attribute:

- orderNumber
- dateSubmitted
- timeSubmitted
- dateServed
- timeServed
- totalCost
- paymentMethod
- cardNumber

Class “ OrderItem”(association Class between “Order” and “menuItem”)

Attribute:

- quantity

Association between the class :

- Restaurant1 – 0..*Employee
- Restaurant 1 - 1..* Menu
- Restaurant 1 – 0..* order
- Menu 1 - 1..* MenuItem
- Customer 1- 0..* Order
- Order 1 – 1..* OrderItem
- OrderItem 1 – 0..* menuItem
- Employee 1 – 0..*EmployeeWorkTimeLog

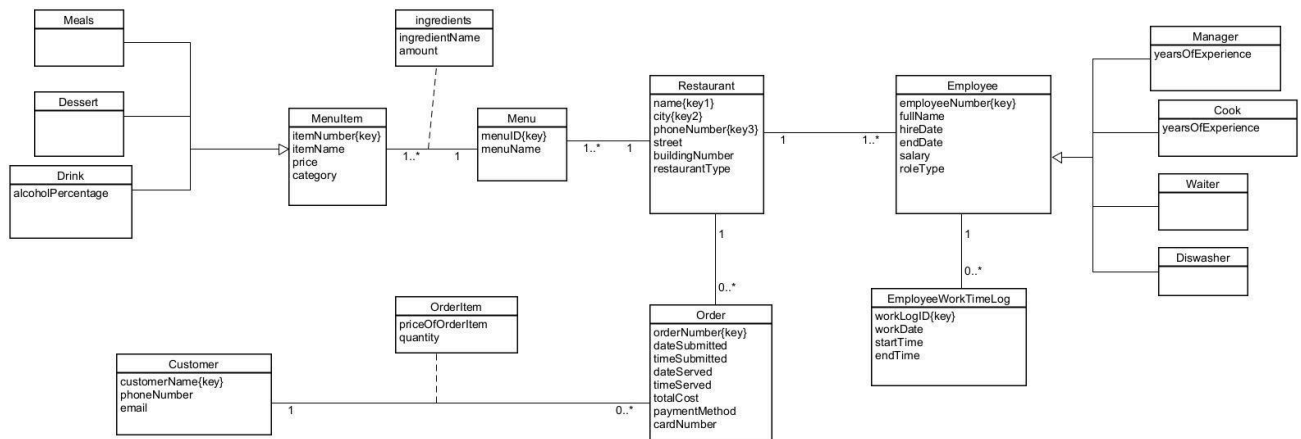


Figure 6: Domain Model Class diagram

9. Conclusion

This systems analysis report demonstrates a comprehensive understanding of the restaurant chain's requirements and confirms the feasibility of the proposed centralized information system. The core vision established in this analysis addresses the critical issue of fragmented and inefficient information management, which currently hampers operational efficiency (Deloitte 2016). By integrating capabilities to manage restaurant details, employee schedules, menu administration, and customer orders into a single system, the proposed solution directly aligns with organizational needs and stakeholder expectations.

The stakeholder analysis conducted ensures that all user groups—from managers and staff to customers—have their requirements effectively captured and represented, significantly reducing the risk of poor user adoption. Additionally, the feasibility study indicates strong prospects for successful implementation, given the maturity and availability of technologies involved and the competencies of the project team (Kendall & Kendall 2019).

All identified functional requirements have been meticulously captured through fully developed use cases and visualized using UML diagrams, providing clarity on interactions between system actors and data entities (Object Management Group 2017). The dynamic processes represented through activity diagrams and the system's static structure in domain model class diagrams remain consistently aligned, reflecting best practices in system analysis and design methodologies (Booch, Rumbaugh & Jacobson 2005; Fowler 2004).

Furthermore, the structured project management approach—including the iteration schedule, Work Breakdown Structure (WBS), and Gantt chart—demonstrates a clearly articulated path for timely project delivery. Key milestones, such as iterative checkpoints, ensure project tracking and risk management are proactively handled (Project Management Institute 2021; Kerzner 2017).

As the project moves into the design and implementation phases, maintaining the analysis documentation as living documents is crucial. Any adjustments or enhancements to the requirements should be updated accordingly, and stakeholder involvement, particularly in user interface design and early testing, is recommended to validate and refine system usability (Norman & Draper 1986; ISO 9241-210:2019).

Overall, this analysis report confirms that the proposed IT solution is well-conceived and strategically aligned with the restaurant chain's operational needs. Implementing the system will lead to business improvements, including enhanced operational efficiency, reduced costs, and improved customer satisfaction (Trivedi & Gaur 2015). Therefore, it is recommended that the project proceed with confidence, leveraging this analysis as a robust foundation for successful system design and implementation.

References

- Alrawadieh, Z 2019, 'Information technology in the hospitality industry: An overview', *Tourism and Hospitality Research*, vol. 19, no. 3, pp. 263-276.
- Booch, G, Rumbaugh, J & Jacobson, I 2005, *The Unified Modeling Language User Guide*, 2nd edn, Addison-Wesley, Boston.
- Deloitte 2016, *The restaurant of the future: Creating the next-generation customer experience*, Deloitte University Press, New York.
- Fowler, M 2004, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd edn, Addison-Wesley, Boston.
- ISO 9241-210:2019, *Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems*, International Organization for Standardization, Geneva.
- Ivanov, S & Webster, C 2017, 'Adoption of information and communication technologies in tourism', *Tourism Management Perspectives*, vol. 23, pp. 1-5.
- Kendall, KE & Kendall, JE 2019, *Systems Analysis and Design*, 10th edn, Pearson, Boston.
- Kerzner, H 2017, *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, 12th edn, Wiley, Hoboken.
- Laudon, KC & Laudon, JP 2020, *Management Information Systems: Managing the Digital Firm*, 16th edn, Pearson, Boston.
- NCR Corporation 2020, *The Restaurant Technology Report*, NCR Corporation, Atlanta.
- Norman, DA & Draper, SW 1986, *User Centered System Design: New Perspectives on Human-Computer Interaction*, Lawrence Erlbaum, Hillsdale.
- Object Management Group 2017, *Unified Modeling Language (UML) Specification*, Version 2.5.1, OMG, Boston.
- Oracle 2021, *Oracle MICROS Point of Sale: Restaurant Management Solutions*, Oracle, Redwood Shores.

Project Management Institute 2021, *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 7th edn, PMI, Newtown Square.

Trivedi, RH & Gaur, SS 2015, 'Barriers to the adoption of restaurant management systems: A study of small and medium enterprises', *International Journal of Hospitality Management*, vol. 51, pp. 28-40.

Xia. T 2025. 'Stakeholder Analysis & Project Planning' Lecture, CSIT114, University of Wollongong, viewed 12 March 2025.

Xia. T 2025. 'Investigating System Requirements' Lecture, CSIT114, University of Wollongong, viewed 19 March 2025.

Xia. T 2025. 'User Stories and Use Cases' Lecture, CSIT114, University of Wollongong, viewed 26 March 2025.

Xia. T 2025. 'Domain Modelling' Lecture, CSIT114, University of Wollongong, viewed 2 April 2025.

Xia. T 2025. 'Use Case Modelling' Lecture, CSIT114, University of Wollongong, viewed 9 April 2025.

Xia. T 2025. 'Foundation for System Design' Lecture, CSIT114, University of Wollongong, viewed 16 April 2025.

Xia. T 2025. 'Defining the System Architecture' Lecture, CSIT114, University of Wollongong, viewed 30 April 2025.

Xia. T 2025. 'Designing the User Interface' Lecture, CSIT114, University of Wollongong, viewed 7 May 2025.

Xia. T 2025. 'Object-Oriented Design: Fundamentals' Lecture, CSIT114, University of Wollongong, viewed 14 May 2025.

Xia. T 2025. 'System Development Approaches' Lecture, CSIT114, University of Wollongong, viewed 21 May 2025.

Xia. T 2025. 'Deploying the New System' Lecture, CSIT114, University of Wollongong, viewed 28 May 2025.

IBM 2023, *UML diagrams* on *IBM DevOps Model Architect*, viewed on 22 May, available at:
<<https://www.ibm.com/docs/en/dma?topic=diagrams-uml>>.