Software Requirements and Design Document

RestoRover

Course Instructor
Ma'am Nighar Azhar

Submitted By

22i-2692	Musfirah Zunnoon
22i-2610	Noor Fatima
22i-2564	Muhammad Ghaus

Date
06-06-2024

Department of Computer Science

FAST – National University of Computer & Emerging Sciences

Document Version

Version	Date	Description	Authors
V1.0	22-02-24	Initial Vision Document	Musfirah Zunnoon Noor Fatima Muhammad Ghaus

Table Document Version

1. Introduction

1.1 Purpose

The purpose of this document is to outline the key requirements and functionalities of the Restaurant Management System (RMS). It serves to gather, analyze, and define the essential needs of stakeholders and target users within the restaurant industry. By focusing on these requirements, this document aims to provide a clear understanding of why specific features are necessary and how they contribute to the overall effectiveness and efficiency of restaurant operations. Detailed information on how the Restaurant Management System addresses these needs will be elaborated upon in the subsequent sections, including use-case scenarios and supplementary specifications.

1.2 Scope

The Restaurant Management System (RMS) is a dynamic platform tailored specifically for the restaurant industry, offering a comprehensive suite of tools and features to streamline operations and enhance customer experiences. Accessible via both desktop computers and web browsers, the RMS ensures ease of use across multiple devices, with future support to Android devices through APK installations. Additionally, the RMS provides valuable resources for restaurant owners and managers, enabling them to seamlessly integrate the system into their existing workflows and monitor key performance metrics. This versatility makes the RMS an indispensable tool for optimizing restaurant operations, whether for single-location establishments or multi-chain franchises.

1.3 Definitions, Acronyms, and Abbreviations

RMS – Restaurant Management System: A restaurant management system (RMS) is a specialized software tailored for businesses in the food service industry, including restaurants, bars, and food trucks. Unlike a traditional point-of-sale (POS) system, an RMS addresses comprehensive back-end requirements such as inventory management and staff operations.

CRM - Customer Relationship Management: refers to the technological system utilized to effectively oversee and maintain all interactions and relationships between a company and its current or prospective customers.

APK (Android Package Kit) Installations - serves as the file type for applications within the Android operating system. It is widely utilized across various platforms, including mobile phones, video games, and middleware for distribution and installation purposes.

IT - Information Technology the management and delivery of information via voice, data, and video through the use of hardware, software, services, and supporting infrastructure

IVR - Interactive Voice Response: The automated phone system technology enables incoming callers to access pre-recorded messages through a voice response system without the need to speak to an agent.

POS - Point of Sales: is the place where a client completes the payment for goods or services and where any applicable sales taxes are paid.

E-Wallet - Electronic Wallet: a particular kind of pre-paid account where users can keep money for use in future internet transactions

BOT - Internet bot: self-contained software that can communicate with other networks or systems over the internet.

Data Silos: refers to a set of data exclusively held by one team or department within an organization, making it difficult for other groups within the same organization to access or utilize the information effectively.

Franchise: an approval given to a person or organization by a government agency or business to engage in specific commercial activities, such as serving as an agent for a company's goods.

Third-Party Integration: A computer program written or developed by a different company than the one that made the computer's operating system

TBD - To be Determined

MOH - Minister of Health

1.4 References

TBD

2. Positioning

2.1 Business Opportunity

The Restaurant Management System (RMS) is a software solution aimed at optimizing restaurant operations. It offers tools to streamline tasks, enhance customer experiences, and boost efficiency across various restaurant sizes. Users include restaurant owners, managers, staff, and customers. Their goals revolve around improving operations, increasing profitability, and ensuring seamless dining experiences. The RMS enhances job satisfaction, reduces wait times, and boosts productivity. It features order management, inventory tracking, reservations, CRM, and analytics, enabling efficient operations and exceptional customer service. Restaurants can easily install the RMS into their existing systems. It provides monitoring tools

for real-time performance tracking, inventory management, and customer feedback analysis, facilitating proactive decision-making. The target market for this system includes restaurants of all sizes, from small family-owned establishments to large franchise chains.

2.2 Problem Statement

The problem of	inefficiencies and challenges in restaurant management
affects	restaurant owners, managers, customers, and staff
the impact of which is	decreased operational efficiency, suboptimal customer experiences, and reduced profitability for restaurants.
a successful solution would be	the development and implementation of a comprehensive Restaurant Management System (RMS) that addresses key pain points in restaurant operations. This RMS should offer modules for order management, menu customization, table seating optimization, inventory control, employee scheduling, customer relationship management, and analytics reporting. By providing a centralized platform with integrated tools and functionalities, the RMS aims to streamline operations, enhance customer satisfaction, and drive profitability for restaurants of all sizes and types.

Table 2.2 Problem Statement

2.3 Position Statement

For	restaurant owners, managers, customers, and staff
Who	intend to refine restaurant operations, improve customer satisfaction and retention, and increase profitability
RestoRover	is a software product
That	integrates advanced modules such as intuitive menu configuration, order management, table reservation system, inventory management, tracking restaurant finances, analytics and reporting etc.
Unlike	other traditional restaurant management systems have outdated technology and complicated interfaces that are difficult to operate. They demand significant maintenance and are not flexible enough to adjust to changing needs. Effective decision-making and responsiveness are further hampered by data silos and a lack of real-time information.
Our solution	is a comprehensive Restaurant management system that provides modules for inventory control, employee scheduling, order management, menu customization, table seating optimization, and analytics reporting. The Restaurant Management System (RMS) provides a centralized platform with integrated tools and functionalities to improve customer satisfaction, streamline operations, and increase restaurant profitability.

Table 2.3 Position Statement

3. Stakeholder

3.1 Non-User Stakeholders

Name	Description	Responsibilities
System Analyst	This stakeholder collaborates with the other stakeholders to identify their needs.	Leads and organizes requirements elicitation and use-case modeling by defining the system's boundaries and specifying its functionality, such as by identifying the players involved and the use cases they will need to interact with the system.
Requirements Specifier	This stakeholder collaborates with the analysts to accurately convert requests and needs into design requirements.	Describes one or more aspects of the requirements, both functional and non-functional, in-depth to describe one or more components of the functioning of the system.
Technical Reviewer	For the development cycle to be maintained, this stakeholder must be engaged frequently.	Accountable for offering review process comments. The technical review of project artifacts is under the purview of this role's area of review. Giving prompt, pertinent input on the project artifacts under review is the responsibility of this function.

Software Architect	This stakeholder plays a key role in directing the system development.	Accountable for the project's software architecture, which includes the major technical choices that limit the project's overall design and implementation. Assures that the system will be maintainable and that the architectural solution fulfills both functional needs and non-requirements.
Project Manager	This stakeholder is primarily responsible for overseeing the system's development.	Plans, organizes, and distributes resources; establishes priorities; organizes contacts with clients and users; and maintains the project team's concentration. provides a system of procedures that also guarantees the accuracy and excellence of project artifacts.
Market Analyst	This is a stakeholder who will help our efforts to successfully position our goods.	Make certain that there will be a market for the new service and the product's features.

Table 3.1 Non-User Stakeholders

3.2 User Stakeholders

Name	Description	Responsibilities	Stakeholders
Sponsor / Owner / Shareholder	Stakeholders who have both control and responsibility for cost and income. (Primary Stakeholder)	To establish an atmosphere that promotes successful projects where performance quality, final costs, and completion time are all within predetermined bounds.	Self
Staff/ Managers	The worker whose direct actions determined whether the project was successful or not. (End User)	To ensure that the work is completed by the established plan, policies, and standards of quality to fulfill the mission statement that the managers and management board established.	Self

Customers	The party that pays a seller, vendor, or supplier money or other valuable consideration in exchange for receiving a good, service, product, or idea (End User)	To evaluate the new food and service to see if it truly satisfies their needs and meets their expectations in terms of quality and cost.	Self
Supplier	A vendor is a company that contributes to the development process by providing goods and services to other companies. (Primary Stakeholder)	To guarantee the food supplier's quality, delivery timetables, and reasonable pricing.	Self
Creditors/ Bank	A person or institution to whom money is owned. (Secondary Stakeholder)	Ensuring the restaurant's stability and the stability of the money owed involves ensuring that the money is collected in the precise amount and at the exact time.	Self

Local Government / MOH	The local government or Ministry of Health who sets the health criteria and standards. (Secondary Stakeholder)	Determining tax rates and whether to grant future extensions, like building warehouses and hiring labor from the country, will also make these decisions. If the surroundings comply with health regulations	Health department Environmental department Licensing department Economic development office

Table 3.2 User Stakeholders

4. Target Audience

A restaurant management system's (RMS) main target audience consists of managers, employees, and restaurant owners. These parties are in charge of managing the marketing, finance, inventory, and customer service departments, among other facets of restaurant operations. Through the provision of features and capabilities customized to user requirements, an RMS seeks to maximize productivity, optimize efficiency, and enhance overall business performance. Furthermore, franchise owners, catering managers, and IT administrators all of whom have important responsibilities in overseeing particular facets of restaurant operations may also be the target of RMS solutions. The people and groups that operate and oversee restaurants of all shapes and sizes from independent establishments to franchise chains are ultimately the target market for an RMS. Customers' dining experiences are further improved by the RMS, which offers effective table management, simple online reservations, and individualized service based on their order history and preferences. All things considered, an RMS is a complete solution that enables seamless interaction between customers and restaurants, guaranteeing efficient operations and a pleasurable dining experience.

5. Product Features

5.1 Point-of-Sale System (POS)

5.1.1 Payment Processing

Accept variouspayment methods (credit cards, debit cards, mobile payments, contactless).

5.1.2 E-wallet System

Offer refunds through a dedicated e-wallet.

5.2 Intuitive Menu Configuration

5.2.1 Easy Updates

Update menu items and prices without the hassle.

5.2.2 Customized Menu

Customers have the option to personalize their menu selections by adding or removing things based on their preferences, dietary restrictions, or allergies.

5.3 Order Management

5.3.1 Online Ordering

Offer a seamless online ordering platform with customization options.

5.3.2 Third-Party Integrations

Simplify order flow with seamless integration with popular delivery platforms.

5.3.3 Offline Orders - IVR

Integrate call recordings for efficient phone order management with IVR suggestions and handling long calls.

5.4 Inventory Management

5.4.1 Ingredient Tracking

The inventory management system in a restaurant helps accurately measure ingredient consumption, define menus, and specify quantities for each dish.

5 .4.2 Food Costing

Calculate recipe costs and analyze profitability per dish.

5.4.3 Theft Prevention

Track stock levels to minimize theft and loss

5.4.4 Recipe Management

involves organizing, categorizing, and easily accessing recipes for seamless cooking experiences.

5.5 Table Reservation System

Allow customers to book tables online through your website or app.

5.6 Feedback and Analysis

5.6.1 Customer Feedback Collection

Customers can seamlessly provide their valuable feedback and insights.

5.6.2 Rating Scale

Customers can give ratings and suggestions.

5.6.3 Feedback Trends and Insights

Identify key themes and trends in customer feedback.

5.7 Customer Relationship Management (CRM)

5.7.1 Member Registration

Offer clients online and in-store registration for loyalty programs.

5.7.2 Points/Awards Accumulation

Reward customers for purchases with points or awards.

5.7.3 Redemption

Customers can redeempoints and awards for discounts, freebies, or special promotions.

5.7.4 Personalized Recommendations

Use customer data to suggest relevant dishes and offers.

5.8 Tracking Restaurant Finances

5.8.1 Taxes

Simplify tax calculations and manage complex tax categories.

5.8.2 Sales Analysis

Identify top-selling items, outlets, and servers to optimize performance through sales tracking.

5.8.3 Cost & Profit Calculations

Track food costs, waste, and theft for accurate profit calculations.

5.9 Analytics and Reporting

5.9.1 Customer Data Analysis

Understand customer preferences, payment methods, and communication behaviors.

5.9.2 Sales & Menu Insights

Identify best-selling items, adjust menus, and encourage add-ons and tipping.

5.9.3 Automatic Reporting

Generate automatic reports after each shift or day with on-demand access to insights.

5.9.4 Inventory Analysis

Minimize waste and loss by analyzing historical data and identifying frequently wasted items.

5.10 Enhanced Responsiveness via Bot Support

Bots will be designed to respond to customer calls exclusively during periods of high online ordering traffic, allowing for efficient management of multiple orders.

5.11 Payroll and Accounting Service

5.11.1 Automated Processing

Simplify payroll calculations, and deductions.

5.11.2 Cash Flow Monitoring

Monitor cash flow in real-time and generate financial reports.

5.11.3 Integration with POS

Integrate with your POS system for automatic sales data feed.

5.13 Event Management

5.13.1 Private Event Booking

Manage private event bookings with availability calendars and online booking.

5.13.2 Catering Orders and Management

Handle catering orders, including menu customization, pricing, and logistics.

5.14 Task and Employee Management

5.14.1 Staff Task Assignment and Tracking

Assign tasks to individual staff members and track progress.

5.14.2 Shift Scheduling and Availability

Used to check employee attendance, shifts, and time off.

5.15 Supplier Management

5.15.1 Supplier Database

Maintain a centralized database of supplier information and contact details.

5.15.2 Automated Ordering

Automate ordering processes based on inventory levels and reorder points.

6.Product Overview

6.1 Summary of Capabilities

Customer Benefit	Supporting features
Optimized Dining Experience	Seamless Ordering with Order History: Pre-Order History enables fast and simple reordering of favorite dishes, reducing the time spent on decision-making and offering customers a personalized and effective dining experience.
Convenience and Streamlined Reservation	Table Reservation System: provides a user-friendly online reservation that allows the customers to make reservations for the preferred date, time, table and settings.
Customized Services	Personalized Recommendations: Offers and recommendations that are specifically tailored to a customer's preferences, order history, and behavior are made possible by using customer data. Customer-Driven Menu Suggestions: Customers are allowed to suggest menu items which fosters a sense of interest and engagement while ensuring that the restaurant caters to the customers' preferences effectively.
Minimized Waiting Time	Pre Orders and Table Reservations: minimizes customer wait times, ensuring a streamline process and enjoyable dining experience.
Persistent quality	Analytics and Food Reports: Quality assurance of food and service standards , fosters customer confidence and loyalty to restaurants.
Flexibility in Payment Options	Payment Processing: Accept various payment options like credit cards or debit cards and mobile payments.

	E-Wallet System: is a secure, efficient and convenient method for customers to receive refunds.
Customer Satisfaction	Feedback/Review System: Gather customer feedback and reviews to identify areas for improvement and enhance customer satisfaction.
Increased Transparency and Assurance	Through the RMS, customers can monitor the progress of their orders in real-time, eliminating doubt and enhancing transparency in the dining process.
Access to Latest Culinary Trends	Real Time Updates: Restaurants can easily update their menus in real-time, reflecting changes in ingredients, seasonal offerings, or promotional items, providing customers with accurate and relevant information.
Versatile Ordering Experience	Online Order Management: enables clients to easily make orders online, which shortens wait times and improves convenience. Third-Party Order Integration: Allows clients to easily place orders from the platforms of their choice. Offline Order Management (IVR): Use an IVR system to make reservations and orders over the phone. This ensures that calls are handled and processed quickly, improving customer service.

Table 6.1 Summary of Capabilities

6.2 Cost and Pricing

TBD

6.3 Licensing and Installation

The product needs to be installed by a professional. Only authorized individuals are permitted to install it.

7. Dependencies and Constraints

7.1 Transaction Speed

The POS system guarantees transaction processing within 3 seconds on average, ensuring swift checkout experiences for customers.

7.2 Capacity

The POS system can handle up to 1000 transactions per hour without any performance degradation.

It is capable of processing up to 500 transactions simultaneously during peak hours.

7.3 Performance

Transaction Speed: Ensure fast processing of transactions to minimize waiting times for customers and improve overall efficiency.

Response Time: The system should respond promptly to user inputs and commands to provide a seamless user experience within 1 second.

7.4 Usability

Intuitive User Interface: The system should have a user-friendly interface to facilitate smooth and efficient operation by staff members.

Training Requirements: Minimize training requirements by designing the system with simplicity and ease of use in mind.

7.5 Reliability

The system must perform without failure in 95 percent of use cases during a month

7.6 Scalability

Scalability of Payment Methods: The system should be able to accommodate an increasing number of payment methods as the business grows.

Scalability of Transaction Volume: Ability to handle a growing volume of transactions during peak periods without performance degradation.

7.7 Security

Encryption: All payment transactions and sensitive data must be encrypted to ensure data security.

1. Non Finctional Requirements

Performance Requirements

The RMS must process transactions within 3 seconds, handle 1000 transactions per hour, and support 500 simultaneous transactions, ensuring a response time of less than 1 second.

Safety Requirements

The RMS must achieve 95% uptime, prevent data loss and corruption through regular backups, and include safety protocols for power outages, hardware failures, and cyber threats.

Security Requirements

The RMS must encrypt payment transactions and personal data, implement multi-factor authentication, ensure compliance with data protection regulations, and conduct regular security audits and updates.

User Interface

The RMS must have an intuitive user interface, demonstrate high reliability, support scalability for increased transactions, and be easy to maintain with clear documentation and support.

Business Rules

The RMS must comply with health and safety regulations, support accurate financial reporting, enforce inventory control measures, and provide flexibility in pricing, promotions, and loyalty programs.

2. Extended Use cases

1.1: Give Feedback

<table>
<tr><td>Use Case:</td><td>UC01</td></tr>
<tr><td>Use Case Name:</td><td>Give Feedback</td></tr>
<tr><td>Scope:</td><td>Restaurant Management System</td></tr>
<tr><td>Level:</td><td>User goal</td></tr>
<tr><td>Primary Actor:</td><td>Customer</td></tr>
<tr><td>Stakeholders and Interests:</td><td>● Customer: Wants a convenient and efficient way to provide feedback, and feel heard.
● Restaurant Manager: Wants valuable insights into customer experience, identify areas for improvement, and improve customer satisfaction.
● Employees: Wants to hear a review of their service.</td></tr>
<tr><td>Preconditions:</td><td>Customers have finished their meal and paid their bills.</td></tr>
<tr><td>Post-Conditions</td><td>Feedback is successfully submitted and recorded in the system</td></tr>
<tr><td>Main Success Scenario</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. Customer access the feedback section</td><td></td></tr>
<tr><td></td><td>2. System presents a form for feedback submission, including fields for ratings and comments.</td></tr>
<tr><td>3. Customers enter ratings and comments.</td><td></td></tr>
<tr><td>4. Customer summits the form</td><td></td></tr>
<tr><td></td><td>5. System validates the feedback form.</td></tr>
</table>
</td></tr>
</table>

Extensions:	**5a.** Feedback form validation fails (Empty Field ..) **1.** The system prompts the customer to correct any errors. **2.** Customer corrects the errors and resubmits the form.
Special Requirements:	● The system should be user-friendly and easy to navigate. ● The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations. ● Language internationalization on the text displayed
Technology and Data Variations List:	**4.** Customer adds the feedback through the keyboard.
Frequency of Occurrence:	Could be nearly continuous.
Open Issues:	● Are there any specific feedback questions? ● What kind of rating scales are used? ● What is the process for handling negative feedback and addressing customer concerns? ● Is there a timeframe for manager's response to feedback?

1.2: Reserve Table

Use Case:	UC02
Use Case Name:	Reserve Table
Scope:	Restaurant Management System
Level:	User goal
Primary Actor:	Restaurant Staff (Receptionist), Customer
Stakeholders and Interests:	● **Customer:** Wants a smooth and convenient way to reserve a table at their desired time and date. ● **Staff:** Wants a user-friendly system to manage reservations effectively and communicate efficiently with customers.

<table>
<tr><td>Preconditions:</td><td>

Restaurant staff members are logged in to the table reservation system.
The system has access to the restaurant's seating layout and availability.

</td></tr>
<tr><td>Post Conditions:</td><td>

Table reservation is successfully recorded in the system.
Customer receives confirmation of the reservation.

</td></tr>
<tr><td>Main Success Scenario</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. Staff members check the reservation system for available tables in the requested room.</td><td></td></tr>
<tr><td>2. If available, Staff enters reservation details into the system, including customer cnic, table id, date, and time.</td><td></td></tr>
<tr><td></td><td>3. The system automatically confirms the reservation by generating a receipt and sends a confirmation email or text message to the customer.</td></tr>
</table>
</td></tr>
<tr><td>Extensions:</td><td>a. Cancel reservation:
1. Customers cancel their reservation.
2. Staff marks the reservation as canceled and updates availability in the system.</td></tr>
<tr><td>Special Requirements:</td><td>

The system should be user-friendly and easy to navigate.
The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations.
The system should update all relevant records within 30 seconds after any changes are made.

</td></tr>
<tr><td>Technology and Data Variations List:</td><td>2.Staff enters reservation details through the keyboard.</td></tr>
<tr><td>Frequency of Occurrence:</td><td>Could be nearly continuous.</td></tr>
</table>

Open Issues:	• What is the cancellation policy and associated fees?

1.3: Manage Staff

Use Case:	UC03
Use Case Name:	Manage Staff

<table>
<tr><td>Scope:</td><td>Restaurant Management System</td></tr>
<tr><td>Level:</td><td>User goal</td></tr>
<tr><td>Primary Actor:</td><td>Restaurant Manager</td></tr>
<tr><td>Stakeholders and Interests:</td><td>● Staff: Want clear communication regarding schedules, and tasks.
● Manager: Wants a user-friendly system to manage staff information, schedules, and their respective tasks.</td></tr>
<tr><td>Preconditions:</td><td>● The Manager is logged in to the staff management system with appropriate permissions.
● The system has access to employee data and job roles defined within the restaurant.</td></tr>
<tr><td>Post Conditions:</td><td>● Staff information is accurately recorded and updated in the system.
● Staff roles and permissions are assigned and managed effectively.
● Managers have a clear overview of the staff team and their assigned roles.</td></tr>
<tr><td>Main Success Scenario:</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. The Restaurant Manager adds new staff</td><td></td></tr>
<tr><td>2. Manager enters the Staff name , schedule, salary.</td><td></td></tr>
<tr><td></td><td>4. System confirms the details.</td></tr>
</table>
</td></tr>
</table>

Extensions:	**4a. Remove staff** 1. The manager selects the option Remove staff . 2. Then the manager selects staff id to remove. 3. System removes the staff from the system. **4b. Update staff** 1. The manager selects the option update staff . 2. Then the manager selects staffid to update and enters a new schedule. 3. System updates the staff details.
Special Requirements:	● The system should be user-friendly and easy to navigate. ● The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations. ● The system should update all relevant records within 30 seconds after any changes are made.
Technology and Data Variations List:	Manager enters detail through the keyboard.
Frequency of Occurrence:	Staff management tasks occur regularly, typically on a weekly or bi-weekly basis for scheduling and daily for task assignments.
Open Issues:	● Handling staff turnover and onboarding new employees. ● Managing overtime and ensuring compliance with labor regulations.

1.4: Manage Supplier

Use Case:	UC04
Use Case Name:	Manage Supplier
Scope:	Restaurant Management System
Level:	User goal
Primary Actor:	Restaurant Manager

Stakeholders and Interests:	● **Manager:** Aims to efficiently manage supplier relationships, orders, and inventory levels. ● **Suppliers:** Want clear communication regarding orders, delivery schedules, and payment terms. ● **Accounting Department:** Requires accurate tracking of supplier invoices and payments.
Preconditions:	● Manager is logged in to the supplier management system with appropriate permissions. ● The system has access to supplier data and product categories relevant to the restaurant.
Post Conditions:	● Supplier information is accurately recorded and updated in the system. ● Orders are placed efficiently and tracked through the system.

Main Success Scenario:

Actor Action	System Responsibility
1. Manager accesses the supplier management system.	
2. Manager reviews inventory levels and identifies supplies that need replenishing.	
3. Manager creates purchase orders for the required supplies, specifying quantities, delivery dates, and any special instructions.	
	4. System generates purchase order confirmations and sends them to the selected suppliers.
5. Suppliers acknowledge receipt of the purchase orders and confirm delivery schedules.	
6. Manager monitors the status of pending orders and tracks delivery schedules.	
7. Manager updates inventory levels in the system based on the received supplies.	

Extensions:

3a. **Update Supplier**

1. The manager selects the option update Supplier .
2. Then the manager selects the supplier to update and enters the details.
3. System updates the supplier details.

3b. **Remove Supplier**

1. The manager selects the option Remove staff .
2. Then the manager selects the supplier to remove.
3. System removes the supplier from the system.

5a. Supplier fails to confirm or fulfill the order

	1. The manager follows up with the supplier to resolve the issue or finds an alternative supplier. **6a**. Supplier delivery is delayed or incomplete 1. The manager communicates with the supplier to expedite delivery or arrange for partial shipments.
Special Requirements:	● The system should be user-friendly and easy to navigate. ● The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations. ● The system should update all relevant records within 30 seconds after any changes are made.
Technology and Data Variations List:	Manager enters detail through the keyboard.
Frequency of Occurrence:	Supplier management tasks occur regularly, typically on a weekly basis for order placement.
Open Issues:	How to manage relationships with multiple suppliers and negotiate favorable terms ?

1.5: Manage Menu

<table>
<tr><td>Use Case:</td><td>UC05</td></tr>
<tr><td>Use Case Name:</td><td>Manage Menu</td></tr>
<tr><td>Scope:</td><td>RestoRover (Restaurant Management System)</td></tr>
<tr><td>Level:</td><td>User level goal</td></tr>
<tr><td>Primary Actor:</td><td>Manager</td></tr>
<tr><td>Stakeholders and Interests:</td><td>● Manager: desires an easy-to-use and effective method for updating menu items and prices.
● Customer : expects up-to-date menu information.</td></tr>
<tr><td>Preconditions:</td><td>● The manager has internet access and a compatible device..
● The manager is authorized to make changes to the menu and prices.</td></tr>
<tr><td>Post Conditions:</td><td>● The menu with the items and prices is successfully updated.
● The changes are reflected in the system and reachable by customers.</td></tr>
<tr><td>Main Success Scenario (two-column format):</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. The manager logs into the system</td><td></td></tr>
<tr><td>2. The manager selects the menu update option from the menu dashboard.</td><td></td></tr>
<tr><td></td><td>3. The system displays the current menu with options to edit and enter new items and prices.</td></tr>
<tr><td>4. The manager selects the items to be updated.</td><td></td></tr>
<tr><td></td><td>5. The system shows the form for selected items with fields like name, description and price.</td></tr>
<tr><td>6. The manager modifies the necessary option.</td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td>7. The manager saves the changes.</td><td></td></tr>
<tr><td></td><td>8. The system makes changes.</td></tr>
<tr><td></td><td>9. The system shows the saved dialogue box.</td></tr>
<tr><td>10. The manager exits the menu update.</td><td></td></tr>
</table>
</td></tr>
<tr><td>Extensions:</td><td>*a. At any time, the System fails during the menu update:
To support recovery events can be recovered from any step of the scenario.
1. When the system restarts, the manager logs in and asks for the previous state to be restored.
2. From the last known stable point before the menu update, the system tries to reconstruct the state.
4a. The manager selects the option to add an item.
1. The system prompts to enter all necessary details of the new item to be added.
2. The manager enters all the details.
3. The system verifies the accuracy of all the details entered.
4. The system seeks confirmation from the manager to add the item to the menu.
5. The manager grants the confirmation.
6. The system updates the menu.
4b. The manager selects the option to delete an item.
1. The system prompts to enter item id.
2. The manager entersthe Item ID.
3. System deletes the Item.</td></tr>
<tr><td>Special Requirements:</td><td>● The system should offer an easy-to-use interface that simplifies editing and navigation.
● The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations.
● The system should update the menu within 30 seconds after making any changes.</td></tr>
<tr><td>Technology and Data Variations List:</td><td>4a. The manager enters the details using the keyboard.</td></tr>
<tr><td>Frequency of Occurrence:</td><td>Nearly Continuous</td></tr>
<tr><td>Open Issues:</td><td>● Is a Notification system implemented to inform customers about menu updates?
● Can the menu update in offline mode?</td></tr>
</table>

	• Are prices including taxes?

1.6: Customize Menu

<table>
<tr><td>Use Case:</td><td>UC06</td></tr>
<tr><td>Use Case Name:</td><td>Customize Menu</td></tr>
<tr><td>Scope:</td><td>RestoRover (Restaurant Management System)</td></tr>
<tr><td>Level:</td><td>User level goal</td></tr>
<tr><td>Primary Actor:</td><td>Customer</td></tr>
<tr><td>Stakeholders and Interests:</td><td>● Customer: Desires a user-friendly method to customize menu items based on personal preferences.
● Manager: interested in seeing the customer's choice for better results.</td></tr>
<tr><td>Preconditions:</td><td>● The customer logged into the system.
● The customer has the authorization to customize the menu.</td></tr>
<tr><td>Post Conditions:</td><td>● The customer's customized menu has been successfully updated.
● Modifications are saved for later orders and reflected in the system.</td></tr>
<tr><td>Main Success Scenario (two-column format):</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. The Customer logs into the system</td><td></td></tr>
<tr><td>2. The Customer selects the menu customize option from the menu dashboard.</td><td></td></tr>
<tr><td></td><td>3. The system displays the current menu with customization options.</td></tr>
<tr><td>4. The customer browses through the menu and selects items to be customized.</td><td></td></tr>
<tr><td></td><td>5. The system shows the form for selected items with customizable options.</td></tr>
<tr><td>6. The customer modifies the options</td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td>based on preferences.</td><td></td></tr>
<tr><td>7. The customer saves the changes.</td><td></td></tr>
<tr><td></td><td>8. The system makes changes.</td></tr>
<tr><td></td><td>9. The system shows the saved dialogue box.</td></tr>
<tr><td>10. The customer exits the menu customization.</td><td></td></tr>
</table>
</td></tr>
<tr><td>Extensions:</td><td>*a. At any time, the System fails during the menu customization:
To support recovery events can be recovered from any step of the scenario.

1. When the system restarts, the customer logs in and asks for the previous state to be restored.
2. From the last known stable point during the menu customization, the system tries to reconstruct the state.</td></tr>
<tr><td>Special Requirements:</td><td>● The system should offer an easy-to-use interface that simplifies editing and navigation.
● The system should be able to handle concurrent requests with minimum degradation in performance striving for 95% success rate under load.
● The system should update all relevant records within 30 seconds after any changes are made.</td></tr>
<tr><td>Technology and Data Variations List:</td><td>6a. The customer enters the details using the keyboard.</td></tr>
<tr><td>Frequency of Occurrence:</td><td>Infrequent</td></tr>
<tr><td></td><td>● Is a Notification system implemented to inform managers about menu customizations?
● Can the menu be customized in offline mode?
● Are taxes included in the cost of customized items?</td></tr>
</table>

1.7: Manage Recipe

<table>
<tr><td>Use Case:</td><td>UC07</td></tr>
<tr><td>Use Case Name:</td><td>Manage Recipe</td></tr>
<tr><td>Scope:</td><td>RestoRover (Restaurant Management System)</td></tr>
<tr><td>Level:</td><td>User level goal</td></tr>
<tr><td>Primary Actor:</td><td>Chef</td></tr>
<tr><td>Stakeholders and Interests:</td><td>● Chef: Desires an efficient method to organize, categorize, and easily access recipes for seamless cooking experiences.
● Manager: Benefits from a well-organized and easily accessible recipe database, contributing to the overall efficiency of the kitchen.</td></tr>
<tr><td>Preconditions:</td><td>● The chef logged into the system.
● The chef has the authorization to organize, categorize, and access recipes.
● Relevant recipes are added to the recipe database.</td></tr>
<tr><td>Post Conditions:</td><td>● Recipes are categorized and successfully organized for easy access.
● The system saves and applies modifications, guaranteeing smooth cooking experiences.
● Staff members in the kitchen can locate and follow recipes with ease, increasing productivity.</td></tr>
<tr><td>Main Success Scenario (two-column format):</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. The Chef logs into the system</td><td></td></tr>
<tr><td>2. The chef navigates to the recipe management section.</td><td></td></tr>
<tr><td></td><td>3. The system displays the current list of recipes with organizing and categorizing options.</td></tr>
<tr><td>4. Chef looks through categories or searches for a particular recipe.</td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>5. The system shows a form for selected recipes with fields like name, category, ingredients, and cooking instructions.</td></tr>
<tr><td>6. The chef modifies the options based on preferences.</td><td></td></tr>
<tr><td>7. The chef saves the changes.</td><td></td></tr>
<tr><td></td><td>8. The system makes changes.</td></tr>
<tr><td></td><td>9. The system shows the saved dialogue box.</td></tr>
<tr><td>10. The chef exits the recipe management menu.</td><td></td></tr>
</table>
</td></tr>
<tr><td>Extensions:</td><td>*a. At any time, the System fails during the recipe update:
To support recovery events can be recovered from any step of the scenario.

1. When the system restarts, the chef logs in and asks for the previous state to be restored.
2. From the last known stable point during the recipe modification, the system tries to reconstruct the state.

4a. The chef selects the option to add a new recipe.
1. The system prompts to enter all necessary details of the new recipe to be added.
2. The chef enters all the details.
3. The system verifies the accuracy of all the details entered.
4. The system seeks confirmation from the chef to add a new recipe.
5. The chef grants the confirmation.
6. The system updates the record.</td></tr>
<tr><td>Special Requirements:</td><td>● The system should offer an easy-to-use interface that simplifies editing and navigation.
● The system should update all relevant records within 30 seconds after making any changes.
● The system should maintain a 99.9% uptime ensuring minimal downtime of operations.
● The system should be able to handle concurrent requests with minimum degradation in performance striving for a 95% success rate under load.</td></tr>
<tr><td>Technology and Data Variations List:</td><td>6a. The chef enters the details using the keyboard.</td></tr>
<tr><td>Frequency of Occurrence:</td><td>Infrequent</td></tr>
</table>

Open Issues:	● Is a Notification system implemented to inform kitchen staff about recipe changes? ● Can the recipe be modified in offline mode? ● Can staff access the recipe in offline mode?

1.8: Place Order Ingredients

<table>
<tr><td>Use Case:</td><td>UC08</td></tr>
<tr><td>Use Case Name:</td><td>Place Order Ingredients</td></tr>
<tr><td>Scope:</td><td>RestoRover (Restaurant Management System)</td></tr>
<tr><td>Level:</td><td>User level goal</td></tr>
<tr><td>Primary Actor:</td><td>Chef</td></tr>
<tr><td>Stakeholders and Interests:</td><td>● Chef: Interested in keeping a current inventory, streamlining menu options, and cutting expenses.
● Restaurant Manager: Concerned with the availability of ingredients and overall cost control.
● Suppliers: Interested in receiving accurate and timely orders.</td></tr>
<tr><td>Preconditions:</td><td>● The RestoRover management system is operational.
● Ingredient inventory is accessible.
● Ingredient data is stored in the database</td></tr>
<tr><td>Post Conditions:</td><td>● Ingredient inventory is updated and reflects the current stock of ingredients.
● Costs associated with ingredient usage are tracked appropriately.
● Ingredients expiry dates stored.</td></tr>
<tr><td>Main Success Scenario (two-column format):</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. The chef logs into the system</td><td></td></tr>
<tr><td>2. The chef selects the ingredient management option from the menu dashboard.</td><td></td></tr>
<tr><td></td><td>3. The system displays the current stock of ingredients.</td></tr>
<tr><td>4. The chef checks for the stock and expiry dates of ingredients.</td><td></td></tr>
<tr><td>5. The chef generates purchases for needed ingredients.</td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>6. The system shows the total bill</td></tr>
<tr><td>7. Chef verifies and proceed with payment.</td><td></td></tr>
<tr><td></td><td>8. The system sends the purchase order to the supplier.</td></tr>
<tr><td>9. The chef exits the ingredient management menu.</td><td></td></tr>
</table>
</td></tr>
<tr><td>Extensions:</td><td>*a. At any time, the System fails during the ingredient management:
To support recovery events can be recovered from any step of the scenario.

1. When the system restarts, the chef logs in and asks for the previous state to be restored.
2. From the last known stable point before the ingredient management, the system tries to reconstruct the state.

*b. Chef cancels the placed order Ingredients
1. The chef navigates to the cancellation option.
2. The chef prompts the user to enter the order identification number.
3. The chef enters the order identification number.
4. The chef verifies the accuracy of the order identification number.
5. If the order cancellation request is received within the fixed time limit, the system then cancels the order.
6. The system sends a confirmation message and email to the user.</td></tr>
<tr><td>Special Requirements:</td><td>● The system should offer an easy-to-use interface that simplifies editing and navigation.
● The system should update all relevant records within 30 seconds after any changes are made.
● The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations.
● The system should be able to handle concurrent requests with minimum degradation in performance striving for 95% success rate under load.
● The system will guarantee 99.9% accuracy in data security and confidentiality.</td></tr>
<tr><td>Technology and Data Variations List:</td><td>4. Chef uses a barcode reader for data entry of ingredients.
5a. The chef enters the details using the keyboard.
5b. The catering manager will enter the bank name, card number or IBN number and csv.</td></tr>
<tr><td>Frequency of Occurrence:</td><td>Nearly continuous</td></tr>
<tr><td>Open Issues:</td><td>● Is a Notification system implemented to inform kitchen staff about ingredient inventory updates?</td></tr>
</table>

	● Can the chef check the stock in offline mode? ● Is the system handling unit variations in ingredient measurements like (mg, g, kilo)? ● Can the system automatically generate alerts for the ingredient expiration dates?

1.10: Place Order Catering

<table>
<tr><td>Use Case:</td><td>UC10</td></tr>
<tr><td>Use Case Name:</td><td>Place Order Catering</td></tr>
<tr><td>Scope:</td><td>RestoRover (Restaurant Management System)</td></tr>
<tr><td>Level:</td><td>User level goal</td></tr>
<tr><td>Primary Actor:</td><td>Catering Manager</td></tr>
<tr><td>Stakeholders and Interests:</td><td>● Catering Manager: Interested in managing the ordering process for catering, allocating resources as efficiently as possible, and preserving client satisfaction.
● Suppliers: Interested in receiving accurate and timely orders.</td></tr>
<tr><td>Preconditions:</td><td>● The catering manager has access to the catering management system.
● Catering management system is operational.
● The catering manager has an internet and compatible device.</td></tr>
<tr><td>Post Conditions:</td><td>● Catering orders placed and processed successfully.
● Catering order received timely.</td></tr>
<tr><td>Main Success Scenario (two-column format):</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. The catering manager logs into the system.</td><td></td></tr>
<tr><td>2. The catering manager navigates to the catering order.</td><td></td></tr>
<tr><td></td><td>3. The system displays the form with fields for order.</td></tr>
<tr><td>4. . The catering manager places an order for new catering.</td><td></td></tr>
<tr><td></td><td>5. The system sends the purchase order to the supplier.</td></tr>
<tr><td></td><td>6. The system automatically updates the order list.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>7. System generates the notification for order placement</td></tr>
<tr><td>8. The catering manager supervises and resolves any delivery issues related to the timely delivery of catering orders.</td><td></td></tr>
</table>
</td></tr>
<tr><td>Extensions:</td><td>*a. At any time, the System fails during the catering order:
To support recovery events can be recovered from any step of the scenario.

1. When the system restarts, the catering manager logs in and asks for the previous state to be restored.
2. From the last known stable point during the catering order, the system reconstructs the state.

*b. Manager cancels the placed order catering
7. The manager navigates to the cancellation option.
8. The manager prompts the user to enter the order identification number.
9. The manager enters the order identification number.
10. The manager verifies the accuracy of the order identification number.
11. If the order cancellation request is received within the fixed time limit, the system then cancels the order.
12. The system sends a confirmation message and email to the user.</td></tr>
<tr><td>Special Requirements:</td><td>● The system should offer an easy-to-use interface that simplifies editing and navigation.
● The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations.
● The system should be able to handle concurrent requests with minimum degradation in performance striving for 95% success rate under load.
● The system should update all relevant records within 30 seconds after any changes are made.
● The system will guarantee 99.9% accuracy in data security and confidentiality.</td></tr>
<tr><td>Technology and Data Variations List:</td><td>4a. The catering manager adds data through the keyboard.
4b. The catering manager will enter the bank name, card number or IBN number and csv.</td></tr>
<tr><td>Frequency of Occurrence:</td><td>Nearly continuous</td></tr>
<tr><td>Open Issues:</td><td>● Is the system optimizing delivery logistics and integrating with third-party delivery services?
● Is tax applied to order prices?</td></tr>
</table>

1.11: Manage Ingredients

<table>
<tr><td>Use Case:</td><td>UC11</td></tr>
<tr><td>Use Case Name:</td><td>Manage Ingredients</td></tr>
<tr><td>Scope:</td><td>RestoRover (Restaurant Management System)</td></tr>
<tr><td>Level:</td><td>User level goal</td></tr>
<tr><td>Primary Actor:</td><td>Manager</td></tr>
<tr><td>Stakeholders and Interests:</td><td>● Chef: Wants to easily access, update, and add ingredients to recipes.
● Administrator: Ensures the integrity and accuracy of ingredient data within the system.</td></tr>
<tr><td>Preconditions:</td><td>● Ingredient Inventory system is operational.
● The catering manager has the internet and compatible devices.</td></tr>
<tr><td>Post Conditions:</td><td>● The ingredient is successfully added, edited, or deleted from the system.</td></tr>
<tr><td>Main Success Scenario (two-column format):</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. The manager logs into the system.</td><td></td></tr>
<tr><td>2. The manager navigates to the Ingredient management.</td><td></td></tr>
<tr><td></td><td>3. The system displays the list of existing ingredients.</td></tr>
<tr><td>4. The manager selects the required ingredient to modify.</td><td></td></tr>
<tr><td></td><td>5. The system shows a form for selected Ingredient with fields like name, description and price.</td></tr>
<tr><td>6. The manager modifies the options based on preferences.</td><td></td></tr>
<tr><td>7. The manager saves the changes.</td><td></td></tr>
</table>
</td></tr>
</table>

	<table><tr><td></td><td>**8**. The system makes changes.</td></tr><tr><td></td><td>**9.** The system shows the saved dialogue box.</td></tr><tr><td>**10.** The manager exits the ingredient management menu.</td><td></td></tr></table>
Extensions:	***a.** At any time, the System fails during the catering order: To support recovery events can be recovered from any step of the scenario. **1.** When the system restarts, the manager logs in and asks for the previous state to be restored. **2.** From the last known stable point during the catering order, the system reconstructs the state. **4a.** The manager selects the option to add a new ingredient. 7. The system prompts to enter all necessary details of the new ingredient to be added. 8. The manager enters all the details. 9. The system verifies the accuracy of all the details entered. 10. The system seeks confirmation from the manager to add a new ingredient. 11. The manager grants the confirmation. 12. The system updates the record.
Special Requirements:	● The system should offer an easy-to-use interface that simplifies editing and navigation. ● The system should update all relevant records within 30 seconds after any changes are made. ● The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations. ● System should recover in 20 seconds maximum after system failure.
Technology and Data Variations List:	**4a.** Manager adds data through the keyboard.
Frequency of Occurrence:	Usaually.
Open Issues:	● Is a Notification system implemented to inform chefs about ignredient updates? ● Can the ingredient inventory update in offline mode?

1.12: Make payment

Use Case:	UC12
Use Case Name:	Make Payment
Scope:	RestoRover (Restaurant Management System)
Level:	User level goal
Primary Actor:	Customer, Waiter
Stakeholders and Interests:	● **Customer :** desires safe transactions and immediate assistance. ● **Waiter :** wants the transaction to be quick, safe, and accurate.
Preconditions:	● The customer intends to make payment. ● The system is active and in running state. ● The system has a point of sale system capable of processing payments. ● The point of sale system is integrated with multiple payments through credit and debit card.
Post Conditions:	● The customer receives a confirmation message and email. ● The sale is recorded. ● The system's accounting and inventory are updated. ● The payment authorization approvals are recorded. ● The receipt is generated.

<table>
<tr><td>Main Success Scenario (two-column format):</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. The waiter presents the customer with a bill that indicates the total amount to be paid.</td><td></td></tr>
<tr><td>2. The customer reviews the bill and communicates the preferred payment method to the waiter which is credit card payment.</td><td></td></tr>
<tr><td>3. The waiter enters the selected payment method into the system.</td><td></td></tr>
<tr><td></td><td>4. The system prompts the customer to enter the necessary payment details.</td></tr>
<tr><td>5. The customer enters the required payment details.</td><td></td></tr>
<tr><td></td><td>6. The system sends requests to the appropriate payment gateway, passing the payment details.</td></tr>
<tr><td></td><td>7. The payment gateway processes the payment transaction.</td></tr>
<tr><td></td><td>8. Upon successful payment authorization the system receives a confirmation message from the payment gateway.</td></tr>
<tr><td></td><td>9. The system generates a receipt consisting of the transaction details.</td></tr>
<tr><td>10. The customer receives the receipt.</td><td></td></tr>
</table>
</td></tr>
<tr><td>Extensions:</td><td>*a. At any time, the system fails during the make payment:
To support recovery events can be recovered from any step of the scenario.

1. When the system restarts, the waiter logs in and asks for the previous state to be restored.
2. From the last known stable point during the payment, the system reconstructs the state.

4a. Customer’s selected payment fails
The system prompts the customer to select an alternate payment method.</td></tr>
</table>

	8a. Payment authorization fails The system prompts the customer to select an alternate payment method.
Special Requirements:	● The system's user interface should be intuitive and easy to navigate. ● Language internationalization on the text displayed. ● The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations. ● The system should be able to handle concurrent requests with minimum degradation in performance striving for 95% success rate under load. ● The system should update all relevant records within 30 seconds after any changes are made. ● The system will guarantee 99.9% accuracy in data security and confidentiality, especially with regard to customer information and reservation details.
Technology and Data Variations List:	**5a.** The customer enters credit card information through a keypad or keyboard.
Frequency of Occurrence:	Could be nearly continuous.
Open Issues:	What are the tax law variations?

1.13: Place Order

<table>
<tr><td>Use Case:</td><td>UC13</td></tr>
<tr><td>Use Case Name:</td><td>Place Order</td></tr>
<tr><td>Scope:</td><td>RestoRover (Restaurant Management System)</td></tr>
<tr><td>Level:</td><td>User level goal</td></tr>
<tr><td>Primary Actor:</td><td>Customer</td></tr>
<tr><td>Stakeholders and Interests:</td><td>● Customer : desires safe transactions and immediate assistance.
● Manager : wants the transaction to be quick, safe, and accurate.</td></tr>
<tr><td>Preconditions:</td><td>● The customer has internet access and a compatible device.
● The customer is logged into the system.</td></tr>
<tr><td>Post Conditions:</td><td>● The customer receives a confirmation message and email.
● The sale is recorded.
● The system’s accounting and inventory are updated.
● The payment authorization approvals are recorded.</td></tr>
<tr><td>Main Success Scenario (two-column format):</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1.The customer visits the restaurant website.</td><td></td></tr>
<tr><td>2. The customer browses through the menu and adds the items to the virtual cart.</td><td></td></tr>
<tr><td>3. The customer proceeds to check out.</td><td></td></tr>
<tr><td></td><td>4. The system calculates and prompts the total bill.</td></tr>
<tr><td></td><td>5. The system prompts the customer to enter the delivery details.</td></tr>
<tr><td>6. The customer enters all the necessary</td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td>details.</td><td></td></tr>
<tr><td></td><td>7. The system verifies the accuracy of delivery details entered.</td></tr>
<tr><td></td><td>9. Perform Use Case Make Payment.</td></tr>
</table>
</td></tr>
<tr><td>Extensions:</td><td>*a. At any time, the System fails during the placing order:
To support recovery events can be recovered from any step of the scenario.

1. When the system restarts, the customer logs in and asks for the previous state to be restored.
2. From the last known stable point during the catering order, the system reconstructs the state.

*b. Customer cancels the placed order
1. The customer navigates to the cancellation option.
2. The system prompts the user to enter the order identification number.
3. The customer enters the order identification number.
4. The system verifies the accuracy of the order identification number.
5. The system verifies that the request is received within the fixed time limit.
6. The system cancels the order and prompts that it is successfully canceled.
8a. Customer selected e-wallet payment
The system deducts the total bill from the customer’s e-wallet.
9a. Customer’s selected payment fails
The system prompts the customer to select an alternate payment method.
13a. Payment authorization fails
The system prompts the customer to select an alternate payment method.</td></tr>
<tr><td>Special Requirements:</td><td>● The system’s user interface should be intuitive and easy to navigate.
● Language internationalization on the text displayed.
● The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations.
● The system should be able to handle concurrent requests with minimum degradation in performance striving for 95% success rate under load.
● The system should update all relevant records within 30 seconds after any changes are made.
● The system will guarantee 99.9% accuracy in data security and confidentiality, especially with regard to customer information and reservation details.</td></tr>
<tr><td>Technology and Data Variations List:</td><td>6a. The customer enters delivery information through a keypad or keyboard.

14a. The customer enters credit card information through a keypad or keyboard.</td></tr>
<tr><td>Frequency of Occurrence:</td><td>Could be nearly continuous.</td></tr>
</table>

Open Issues:	● What are the tax law variations? ● What is the fixed time limit for the order to be canceled?

1.14: Book Event

<table>
<tr><td>Use Case:</td><td>UC14</td></tr>
<tr><td>Use Case Name:</td><td>Book Event</td></tr>
<tr><td>Scope:</td><td>RestoRover (Restaurant Management System)</td></tr>
<tr><td>Level:</td><td>User level goal</td></tr>
<tr><td>Primary Actor:</td><td>Receptionist, Customer</td></tr>
<tr><td>Stakeholders and Interests:</td><td>● Customers: want the event booking process to be accurate and efficient.
● Receptionist: wants to ensure a seamless and effective booking procedure..</td></tr>
<tr><td>Preconditions:</td><td>● The system is active and in running state.
● The receptionist is logged into the system.</td></tr>
<tr><td>Post Conditions:</td><td>● The customer receives a confirmation message and email.
● The record of booked events and the availability calendar is updated.</td></tr>
<tr><td>Main Success Scenario (two-column format):</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1.The customer visits the restaurant to book an event.</td><td></td></tr>
<tr><td>2. The receptionist initiates booking of the event.</td><td></td></tr>
<tr><td>3. The receptionist asks the customer for customer information and event information.</td><td></td></tr>
<tr><td>4. The receptionist enters the details into the system.</td><td></td></tr>
<tr><td></td><td>5. The system validates the accuracy of the details entered.</td></tr>
<tr><td></td><td>6. The system checks the availability calendar for the requested date and time.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>7. . The system books the event and prompts that booking of the event is successfully done.</td></tr>
<tr><td>8. . The receptionist communicates successful booking of the event to the customer.</td><td></td></tr>
<tr><td></td><td>9. Perform Use Case Make Payment.</td></tr>
</table>
</td></tr>
<tr><td>Extensions:</td><td>*a. At any time, the System fails during the booking of an event:
To support recovery events can be recovered from any step of the scenario.

1. When the system restarts, the receptionist logs in and asks for the previous state to be restored.
2. From the last known stable point during the booking of an event, the system reconstructs the state.

*b. Customer cancels the placed order
1. The customer asks the receptionist to cancel the event booking he/she placed.
2. The receptionist navigates to the event cancellation option.
3. The system prompts the user to enter the event booking identification number.
4. The receptionist enters the event booking identification number.
5. The system verifies the accuracy of the event booking identification number.
6. The system verifies that the event booking cancellation request is received within the fixed time limit.
7. The system then cancels it and prompts about the successful cancellation.
8. The receptionist informs the customer about the successful cancellation.
5a. The requested date and time are not available
The receptionist suggests alternate date or time to the customer or offers to add them on the waiting list incase of cancellations.</td></tr>
<tr><td>Special Requirements:</td><td>● The system’s user interface should be intuitive and easy to navigate.
● Language internationalization on the text displayed.
● The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations.
● The system should be able to handle concurrent requests with minimum degradation in performance striving for 95% success rate under load.
● The system should update all relevant records within 30 seconds after any changes are made.</td></tr>
<tr><td>Technology and Data Variations List:</td><td>3a. The receptionist enters contact details and event details through a keypad or keyboard.</td></tr>
<tr><td>Frequency of Occurrence:</td><td>Could be nearly continuous.</td></tr>
</table>

Open Issues:	● What are the tax law variations? ● What is the fixed time limit for the event booking to be canceled?

1.15: Manage Member Registration

<table>
<tr><td>Use Case:</td><td>UC15</td></tr>
<tr><td>Use Case Name:</td><td>Manage Member Registration</td></tr>
<tr><td>Scope:</td><td>RestoRover (Restaurant Management System)</td></tr>
<tr><td>Level:</td><td>User level goal</td></tr>
<tr><td>Primary Actor:</td><td>Receptionist</td></tr>
<tr><td>Stakeholders and Interests:</td><td>● Customer: wants the member registration process to be efficient and accurate.
● Receptionist: wants the registration process to be seamless.</td></tr>
<tr><td>Preconditions:</td><td>● The system is active and running.
● The customer wasn't registered as a member before.
● The receptionist is logged into the system.</td></tr>
<tr><td>Post Conditions:</td><td>● The customer receives a confirmation message and email.
● The system’s record of registered customers is updated.</td></tr>
<tr><td>Main Success Scenario (two-column format):</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. The customer intends to join as a member.</td><td></td></tr>
<tr><td>2. The receptionist provides details related to the loyalty program to the customer.</td><td></td></tr>
<tr><td>3.The receptionist initiates the member registration process.</td><td></td></tr>
<tr><td></td><td>4. The system prompts to enter the details of the customer.</td></tr>
<tr><td>5. The receptionists ask the customer for the details.</td><td></td></tr>
<tr><td>6. The receptionist enters the details of the customer.</td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>7. The system verifies the accuracy of the details of the customer entered.</td></tr>
<tr><td></td><td>8. The system verifies that the customer isn’t already registered.</td></tr>
<tr><td></td><td>9. Upon confirmation, the system successfully registers the customer.</td></tr>
<tr><td>10. The receptionist issues a membership card to the customer.</td><td></td></tr>
</table>
</td></tr>
<tr><td>Extensions:</td><td>*a. At any time, the System fails during registering of member:
To support recovery events can be recovered from any step of the scenario.

1. When the system restarts, the receptionist logs in and asks for the previous state to be restored.
2. From the last known stable point during the catering order, the system reconstructs the state.

*b. Customer intends to cancel the membership
1. Customer asks the receptionist to cancel the membership.
2. The receptionist navigates to the cancel membership option.
3. The system asks the receptionist to enter the membership identification number.
4. The receptionist asks the customer for the membership identification number.
5. The receptionist enters the membership identification number.
6. The system verifies the accuracy of the membership identification number.
7. The system initiates the cancellation of membership.
8. The system prompts that the membership has been successfully canceled.
9. The receptionist informs the customer about the successful cancellation.
9a. The customer is already registered.
The system prompts that the customer is already registered.</td></tr>
<tr><td>Special Requirements:</td><td>● The system’s user interface should be intuitive and easy to navigate.
● Language internationalization on the text displayed.
● The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations.
● The system should be able to handle concurrent requests with minimum degradation in performance striving for 95% success rate under load.</td></tr>
<tr><td>Technology and Data Variations List:</td><td>6a. The receptionist enters customer details through a keypad or keyboard.</td></tr>
<tr><td>Frequency of Occurrence:</td><td>Could be nearly continuous.</td></tr>
</table>

Open Issues:	● What actions or factors contribute to the accumulation of rewards for the customers? ● What options are available for the customers to redeem their rewards? ● How are the accumulated rewards of the customer handled if the member decides to cancel their membership?

1.16: Blacklist Customer

<table>
<tr><td>Use Case:</td><td>UC16</td></tr>
<tr><td>Use Case Name:</td><td>Blacklist Customer</td></tr>
<tr><td>Scope:</td><td>RestoRover (Restaurant Management System)</td></tr>
<tr><td>Level:</td><td>User level goal</td></tr>
<tr><td>Primary Actor:</td><td>Manager</td></tr>
<tr><td>Stakeholders and Interests:</td><td>Manager: wants the blacklisting of a customer to be accurate, efficient and seamless.</td></tr>
<tr><td>Preconditions:</td><td>● The system is active and running.
● The manager is logged into the system.
● The customer isn’t already blacklisted.</td></tr>
<tr><td>Post Conditions:</td><td>The system’s record of blacklisted customers is updated.</td></tr>
<tr><td>Main Success Scenario (two-column format):</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. The manager accesses the restaurant management system.</td><td></td></tr>
<tr><td>2. The manager navigates to the blacklist option.</td><td></td></tr>
<tr><td></td><td>3. The system prompts to enter the details of the customer.</td></tr>
<tr><td>4. The manager input all the required details of the customer.</td><td></td></tr>
<tr><td></td><td>5. The system verifies the accuracy of the details entered.</td></tr>
<tr><td></td><td>6. The system checks that the customer isn’t already blacklisted.</td></tr>
<tr><td></td><td>7. The system checks that the customer</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td><table>
<tr><td></td><td>isn’t a member.</td></tr>
<tr><td></td><td>6. The system adds the customer to the blacklist.</td></tr>
<tr><td></td><td>7. The system confirms the addition of the customer to the blacklist.</td></tr>
</table></td></tr>
<tr><td>Extensions:</td><td>*a. At any time, the System fails during the blacklisting of a customer:
To support recovery events can be recovered from any step of the scenario.

1. When the system restarts, the manager logs in and asks for the previous state to be restored.
2. From the last known stable point during the blacklisting of a customer, the system reconstructs the state.
6a. The customer is already blacklisted.
The system prompts that the customer is already blacklisted.</td></tr>
<tr><td>Special Requirements:</td><td>● The system’s user interface should be intuitive and easy to navigate.
● Language internationalization on the text displayed.
● The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations.
● The system should update all relevant records within 30 seconds after any changes are made.</td></tr>
<tr><td>Technology and Data Variations List:</td><td>4a. The receptionist enters customer details through a keypad or keyboard.</td></tr>
<tr><td>Frequency of Occurrence:</td><td>Infrequent</td></tr>
<tr><td>Open Issues:</td><td>Is there a review process that reassess the behavior of the blacklisted customer to determine whether they remain blacklisted or should be removed?</td></tr>
</table>

1.17: Manage Table

Use Case:	UC17
Use Case Name:	Manage Table
Scope:	RestoRover (Restaurant Management System)
Level:	User level goal
Primary Actor:	Manager
Stakeholders and Interests:	**Staff:** wants a user-friendly system or tools to manage the table layout effectively and adapt to reservation needs.
Preconditions:	● The system is active and running. ● The staff is logged into the system. ● Availability of tables for seating and modification.
Post Conditions:	● Table additions, removals, and changes in location are reflected accurately. ● Table seating arrangements are successfully updated in the system.
Main Success Scenario (two-column format):	

Actor Action	**System Responsibility**
1. The Man	
2. The manager navigates to the manage table option	
	3. The system displays a visual representation of the current layout, including the location.
4.The staff selects "Add Table" and chooses the desired size and location	

			5. The system allows dragging and dropping the new table icon onto the desired location	

			6. The system adds the table and shows a confirmation Message.

Extensions:	**4a. Remove Table** 1. Staff select option “Remove Table”. 2. Staff selects the table on the layout. **4b. Update Table** 1. Staff select option “Update Table”. 2. Staff selects the table on the layout and changes the required information.
Special Requirements:	● The system should be user-friendly and easy to navigate. ● The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations. ● The system should update all relevant records within 30 seconds after any changes are made.
Technology and Data Variations List:	**4a.**Tables can be drag/drop through mouse , or enter the table details through keyboard
Frequency of Occurrence:	Infrequent
Open Issues:	How to manage table configurations for different types of events (e.g., large parties, private dining)?

1.18: Make Supplier Order

Use Case:	UC18
Use Case Name:	Make Supplier Order
Scope:	Restaurant Management System
Level:	User goal
Primary Actor:	Restaurant Manager
Stakeholders and Interests:	● **Manager:** Aims to efficiently manage supplier relationships, orders, and inventory levels. ● **Suppliers:** Want clear communication regarding orders, delivery schedules, and payment terms. ● **Accounting Department:** Requires accurate tracking of supplier invoices and payments.
Preconditions:	● Manager is logged in to the supplier management system with appropriate permissions. ● The system has access to supplier data and product categories relevant to the restaurant.
Post Conditions:	● Supplier information is accurately recorded and updated in the system. ● Orders are placed efficiently and tracked through the system.

<table>
<tr><td>Main Success Scenario:</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. Manager creates a new Supplier Purchase order by entering supplier id.</td><td></td></tr>
<tr><td>2. Manager enters item id and description.</td><td></td></tr>
<tr><td></td><td>3. System returns item description and prize</td></tr>
<tr><td>3. Manager ends the sale</td><td></td></tr>
<tr><td>5. Manager makes new payment and enters preferred payment method</td><td></td></tr>
<tr><td>6. Manager enters card details like card id and pin</td><td></td></tr>
<tr><td></td><td>7. System authorizes from payments gateway and generate receipts</td></tr>
</table>
</td></tr>
<tr><td>Special Requirements:</td><td>

The system should be user-friendly and easy to navigate.
The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations.
The system should update all relevant records within 30 seconds after any changes are made.

</td></tr>
<tr><td>Technology and Data Variations List:</td><td>Manager enters detail through the keyboard.</td></tr>
</table>

Frequency of Occurrence:	Supplier management tasks occur regularly, typically on a weekly basis for order placement.
Open Issues:	How to manage relationships with multiple suppliers and negotiate favorable terms ?

1.19: Manage BnaquetHall

<table>
<tr><td>Use Case:</td><td>UC19</td></tr>
<tr><td>Use Case Name:</td><td>Manage Menu</td></tr>
<tr><td>Scope:</td><td>RestoRover (Restaurant Management System)</td></tr>
<tr><td>Level:</td><td>User level goal</td></tr>
<tr><td>Primary Actor:</td><td>Manager</td></tr>
<tr><td>Stakeholders and Interests:</td><td>● Manager: desires an easy-to-use and effective method for updating menu items and prices.
● Customer : expects up-to-date menu information.</td></tr>
<tr><td>Preconditions:</td><td>● The manager has internet access and a compatible device..
● The manager is authorized to make changes to the menu and prices.</td></tr>
<tr><td>Post Conditions:</td><td>● The banquet hall details and prices are successfully updated.
● The changes are reflected in the system and accessible to customers.</td></tr>
<tr><td>Main Success Scenario (two-column format):</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. The manager logs into the system</td><td></td></tr>
<tr><td>2. The manager selects the banquet hall update option from the menu dashboard.</td><td></td></tr>
<tr><td></td><td>3. The system displays the current banquet hall details with options to edit and enter new details and prices.</td></tr>
<tr><td>4. The manager selects the hall to be updated.</td><td></td></tr>
<tr><td></td><td>5. The system shows the form for the selected hall with fields like name,</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>description, capacity, and price.</td></tr>
<tr><td>6. The manager modifies the necessary details</td><td></td></tr>
<tr><td>7. The manager saves the changes.</td><td></td></tr>
<tr><td></td><td>8. The system makes the changes.</td></tr>
<tr><td></td><td>9. The system shows the saved dialogue box.</td></tr>
<tr><td>10. The manager exits the banquet hall update.</td><td></td></tr>
</table>
</td></tr>
<tr><td>Extensions:</td><td>a. At any time, the System fails during the banquet hall update:

1. When the system restarts, the manager logs in and asks for the previous state to be restored.
2. From the last known stable point before the banquet hall update, the system tries to reconstruct the state.

4a. The manager selects the option to add a banquet hall:

1. The system prompts to enter all necessary details of the new hall to be added.
2. The manager enters all the details.
3. The system verifies the accuracy of all the details entered.
4. The system seeks confirmation from the manager to add the hall to the list.
5. The manager grants the confirmation.
6. The system updates the list of banquet halls.

4b. The manager selects the option to delete a banquet hall:

1. The system prompts to enter the hall ID.
2. The manager enters the hall ID.
3. The system deletes the hall.</td></tr>
<tr><td>Special Requirements:</td><td>● The system should offer an easy-to-use interface that simplifies editing and navigation.
● The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations.
● The system should update the menu within 30 seconds after making any changes.</td></tr>
<tr><td>Technology and Data Variations List:</td><td>4a. The manager enters the details using the keyboard.</td></tr>
</table>

Frequency of Occurrence:	Nearly Continuous
Open Issues:	● Is a Notification system implemented to inform customers about menu updates? ● Can the menu update in offline mode? ● Are prices including taxes?

1.20: Manage Customer

<table>
<tr><td>Use Case:</td><td>UC19</td></tr>
<tr><td>Use Case Name:</td><td>Manage Menu</td></tr>
<tr><td>Scope:</td><td>RestoRover (Restaurant Management System)</td></tr>
<tr><td>Level:</td><td>User level goal</td></tr>
<tr><td>Primary Actor:</td><td>Manager</td></tr>
<tr><td>Stakeholders and Interests:</td><td>● Manager: Desires an easy-to-use and effective method for updating customer details.
● Customer: Expects accurate and up-to-date information about their account and reservations.</td></tr>
<tr><td>Preconditions:</td><td>● The manager has internet access and a compatible device.
● The manager is authorized to make changes to customer details.</td></tr>
<tr><td>Post Conditions:</td><td>● The customer details are successfully updated.
● The changes are reflected in the system and accessible to customers.</td></tr>
<tr><td>Main Success Scenario (two-column format):</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. The manager logs into the system</td><td></td></tr>
<tr><td>2. The manager selects the customer management option from the menu dashboard.</td><td></td></tr>
<tr><td></td><td>3. The system displays the current customer list with options to edit and enter new customer details.</td></tr>
<tr><td>4. The manager selects the customer to be updated.</td><td></td></tr>
<tr><td></td><td>5. The system shows the form for the selected customer with fields like name, contact information, and preferences.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td>6. The manager modifies the necessary details</td><td></td></tr>
<tr><td>7. The manager saves the changes.</td><td></td></tr>
<tr><td></td><td>8. The system makes the changes.</td></tr>
<tr><td></td><td>9. The system shows the saved dialogue box.</td></tr>
<tr><td>10. The manager exits the banquet hall update.</td><td></td></tr>
</table>
</td></tr>
<tr><td>Extensions:</td><td>a. At any time, the System fails during the customer update:

1. When the system restarts, the manager logs in and asks for the previous state to be restored.
2. From the last known stable point before the customer update, the system tries to reconstruct the state.

4a. The manager selects the option to add a customer:

1. The system prompts to enter all necessary details of the new customer to be added.
2. The manager enters all the details.
3. The system verifies the accuracy of all the details entered.
4. The system seeks confirmation from the manager to add the customer to the list.
5. The manager grants the confirmation.
6. The system updates the customer list.

4b. The manager selects the option to delete a customer:

1. The system prompts to enter the customer ID.
2. The manager enters the customer ID.
3. The system deletes the customer.</td></tr>
<tr><td>Special Requirements:</td><td>● The system should offer an easy-to-use interface that simplifies editing and navigation.
● The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations.
● The system should update the menu within 30 seconds after making any changes.</td></tr>
<tr><td>Technology and Data Variations List:</td><td>4a. The manager enters the details using the keyboard.</td></tr>
</table>

Frequency of Occurrence:	Nearly Continuous
Open Issues:	● Is a Notification system implemented to inform customers about menu updates? ● Can the menu update in offline mode? ● Are prices including taxes?

1.21: Manage Item

<table>
<tr><td>Use Case:</td><td>UC19</td></tr>
<tr><td>Use Case Name:</td><td>Manage Item</td></tr>
<tr><td>Scope:</td><td>RestoRover (Restaurant Management System)</td></tr>
<tr><td>Level:</td><td>User level goal</td></tr>
<tr><td>Primary Actor:</td><td>Manager</td></tr>
<tr><td>Stakeholders and Interests:</td><td>● Manager: Desires an easy-to-use and effective method for updating customer details.
● Customer: Expects accurate and up-to-date information about their account and reservations.</td></tr>
<tr><td>Preconditions:</td><td>● The manager has internet access and a compatible device.
● The manager is authorized to make changes to customer details.</td></tr>
<tr><td>Post Conditions:</td><td>● The customer details are successfully updated.
● The changes are reflected in the system and accessible to customers.</td></tr>
<tr><td>Main Success Scenario (two-column format):</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>4. The manager logs into the system</td><td></td></tr>
<tr><td>5. The manager selects the Item management option from the menu dashboard.</td><td></td></tr>
<tr><td></td><td>6. The system displays the current item list with options to edit and enter new item details.</td></tr>
<tr><td>4. The manager selects the item to be updated.</td><td></td></tr>
<tr><td></td><td>5. The system shows the form for the selected item with fields like name, category, price</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td>6. The manager modifies the necessary details</td><td></td></tr>
<tr><td>7. The manager saves the changes.</td><td></td></tr>
<tr><td></td><td>8. The system makes the changes.</td></tr>
<tr><td></td><td>9. The system shows the saved dialogue box.</td></tr>
<tr><td>10. The manager exits the banquet hall update.</td><td></td></tr>
</table>
</td></tr>
<tr><td>Extensions:</td><td>a. At any time, the System fails during the item update:

3. When the system restarts, the manager logs in and asks for the previous state to be restored.
4. From the last known stable point before the customer update, the system tries to reconstruct the state.

4a. The manager selects the option to add a customer:

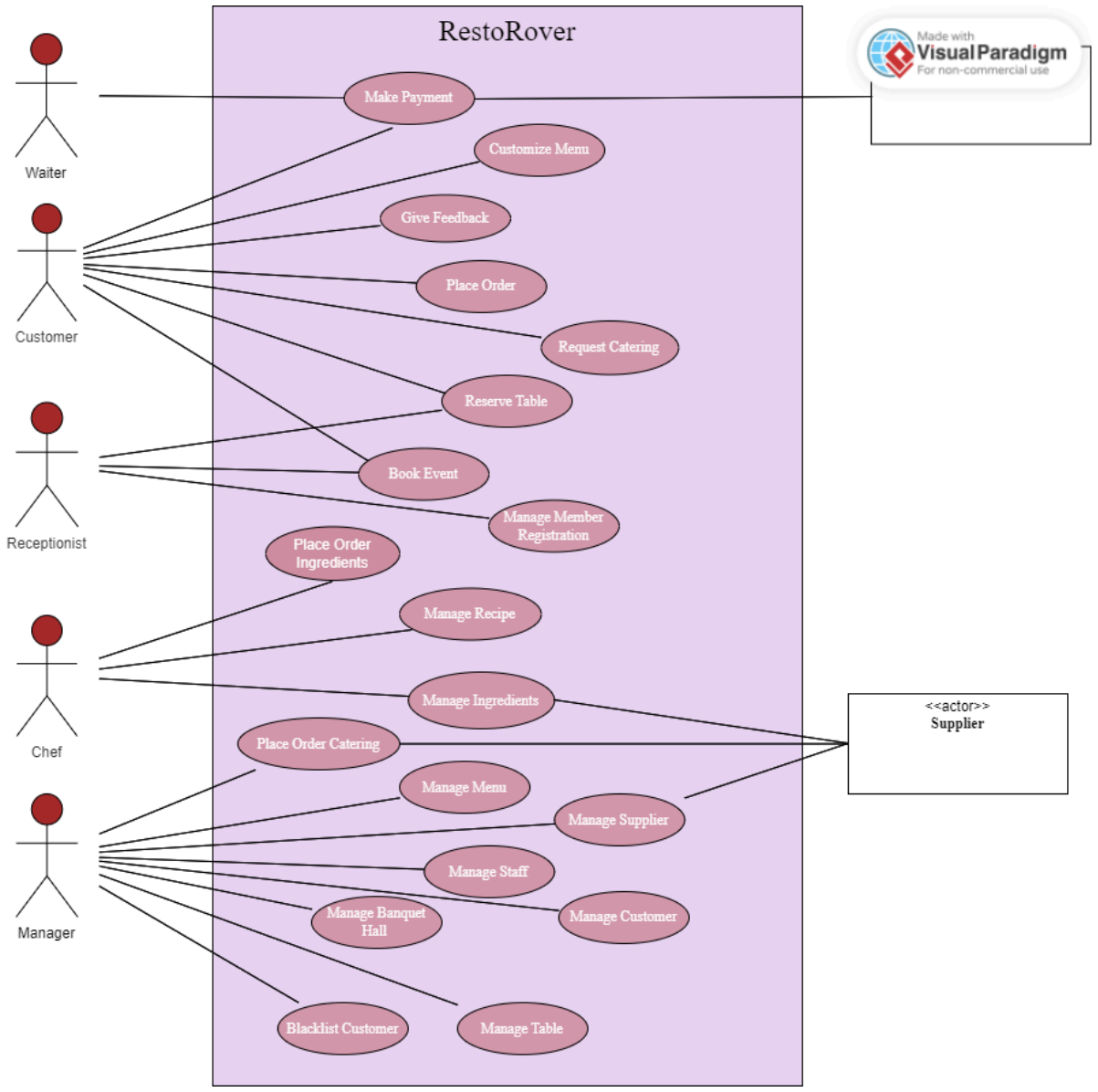
7. The system prompts to enter all necessary details of the new customer to be added.
8. The manager enters all the details.
9. The system verifies the accuracy of all the details entered.
10. The system seeks confirmation from the manager to add the item to the list.
11. The manager grants the confirmation.
12. The system updates the customer list.

4b. The manager selects the option to delete a customer:

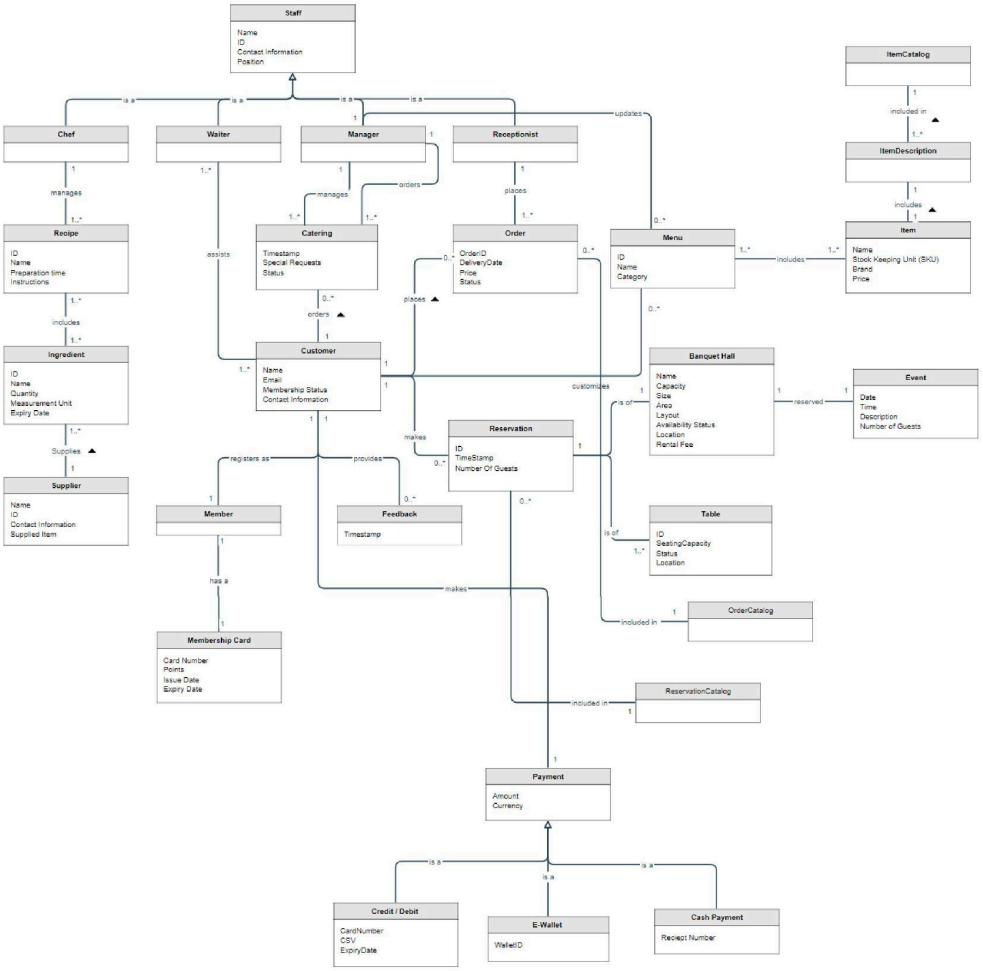
4. The system prompts to enter the customer ID.
5. The manager enters the customer ID.
6. The system deletes the customer.</td></tr>
<tr><td>Special Requirements:</td><td>● The system should offer an easy-to-use interface that simplifies editing and navigation.
● The system should maintain a 99.9% uptime ensuring minimal downtime of critical operations.
● The system should update the menu within 30 seconds after making any changes.</td></tr>
<tr><td>Technology and Data Variations List:</td><td>4a. The manager enters the details using the keyboard.</td></tr>
</table>

Frequency of Occurrence:	Nearly Continuous
Open Issues:	● Is a Notification system implemented to inform customers about menu updates? ● Can the menu update in offline mode? ● Are prices including taxes?

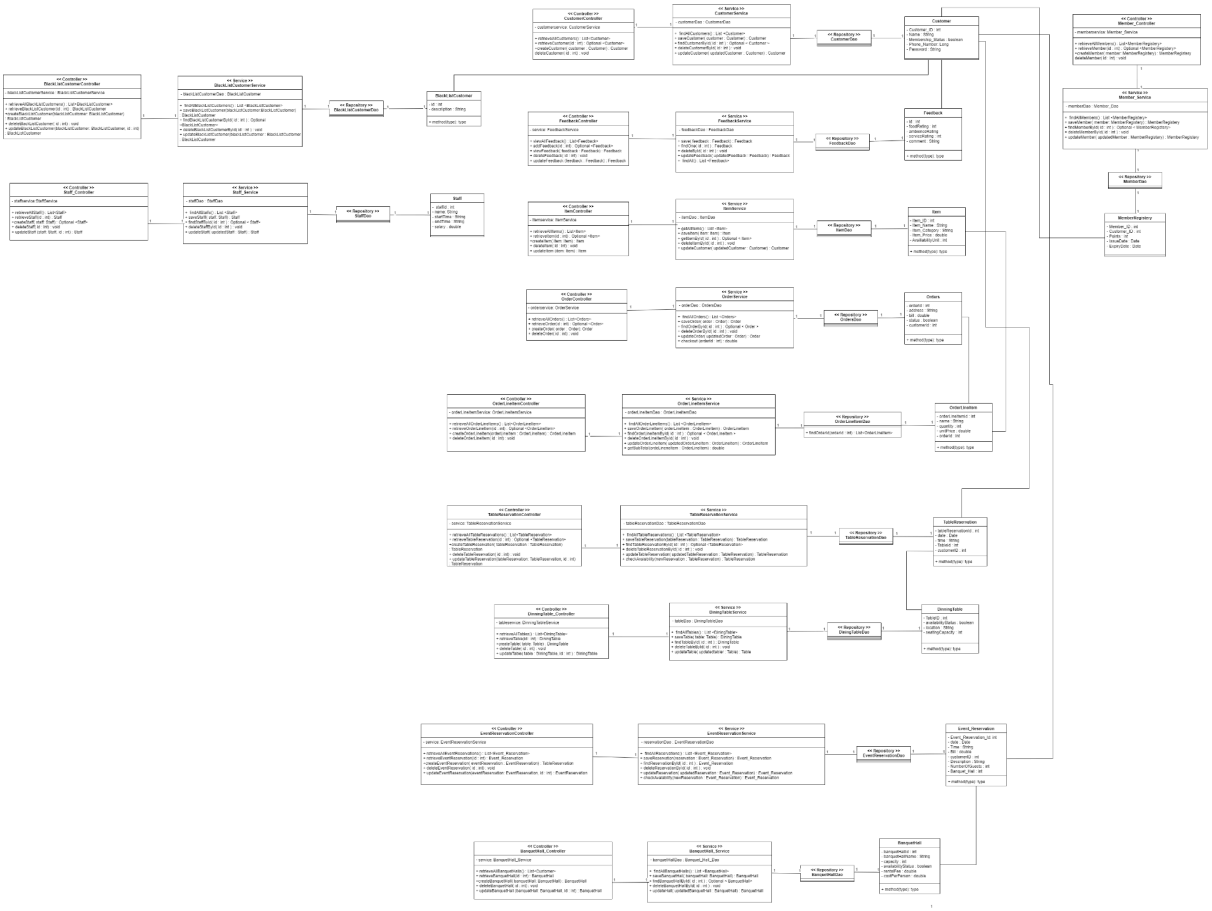
3. UseCase Diagram



4. Domain Model



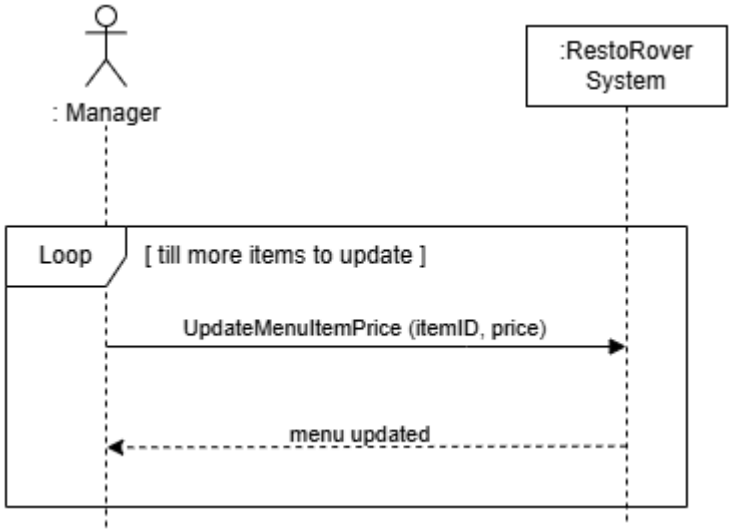
5. Class Diagram



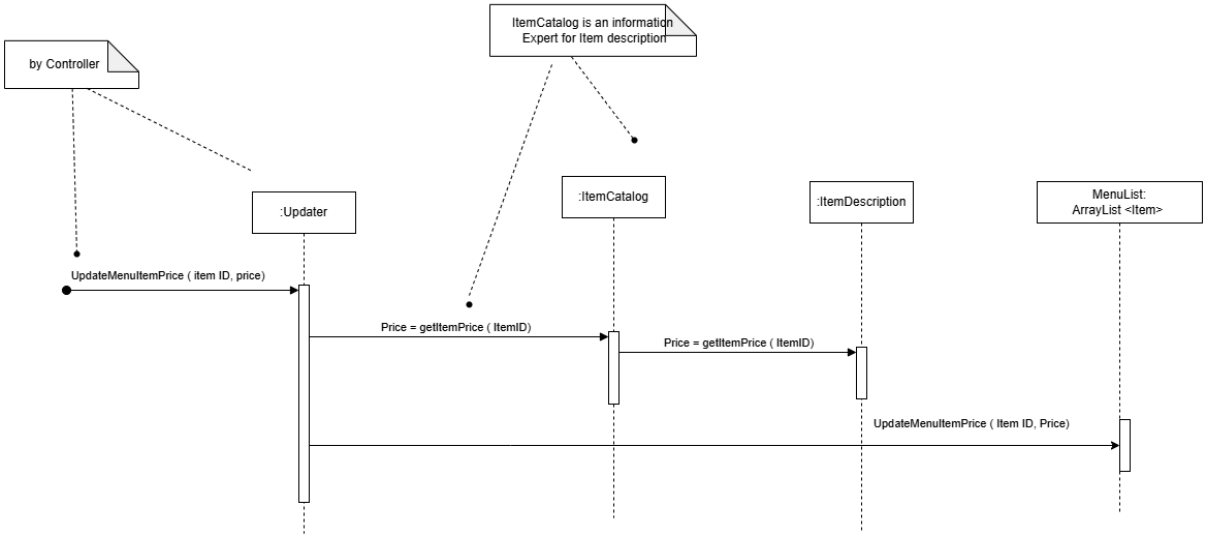
Link
https://app.diagrams.net/#G19cD6Lr2ZvPxsnuJ_YQCZ5P8ETf_LbiqH#%7B%22pageId%22%3A%22BlnSBqig3esf3ockY-1-%22%7D

Update Menu - Manage Menu

System Sequence Diagram (SSD)



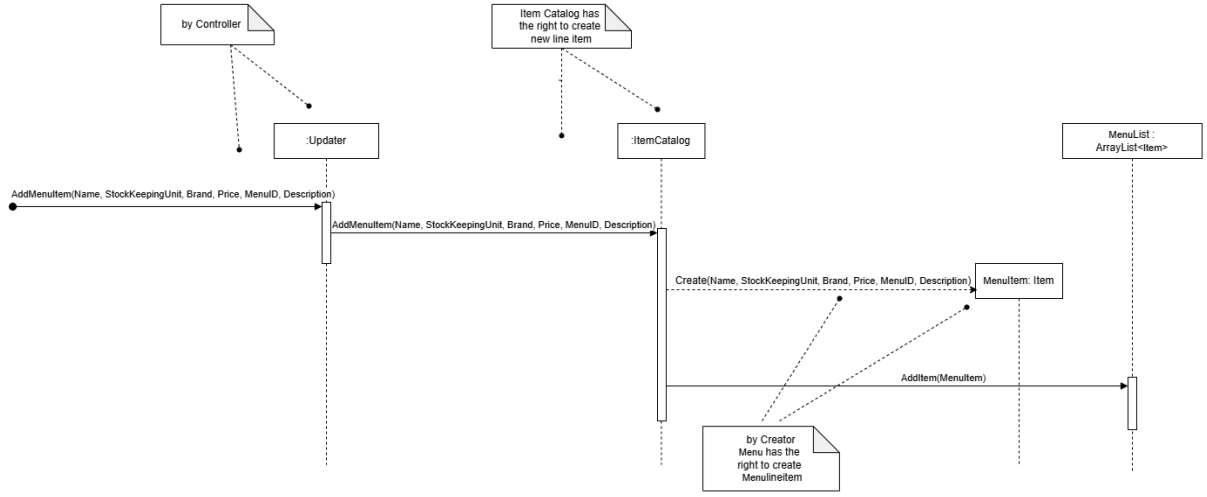
Sequence Diagram (SD)



AddItemInMenu - Manage Menu

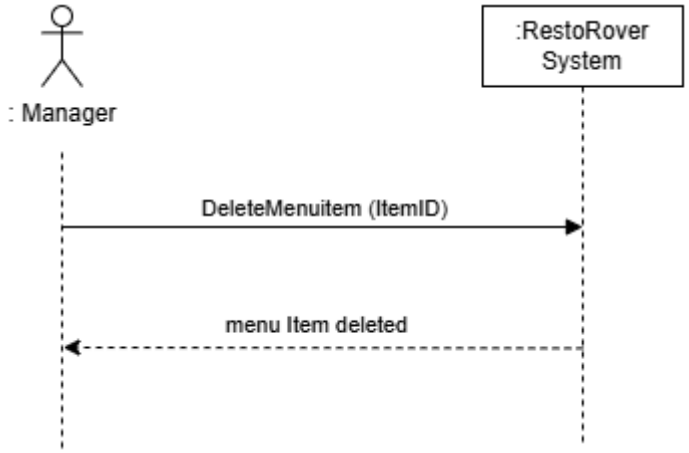
System Sequence Diagram (SSD)

Sequence Diagram (SD)

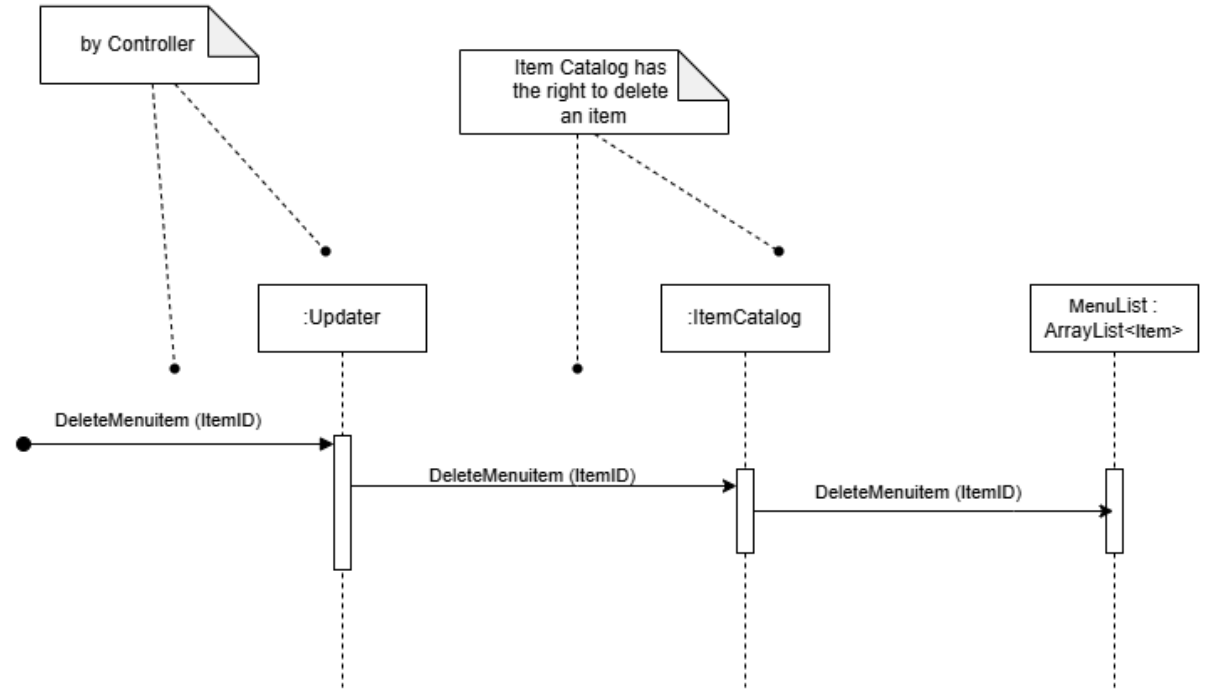


DeleteMenuItem- Manage Menu

System Sequence Diagram (SSD)

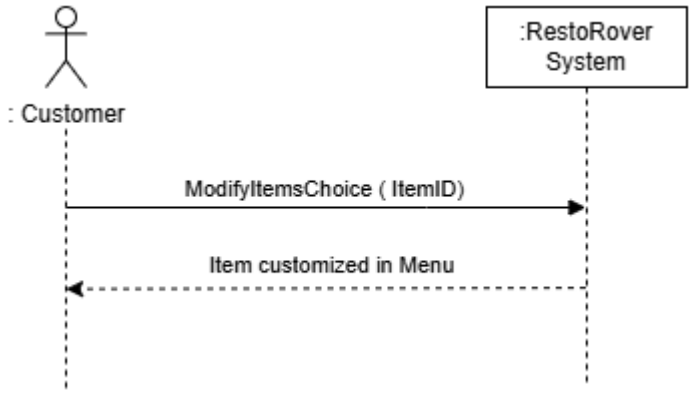


Sequence Diagram (SD)

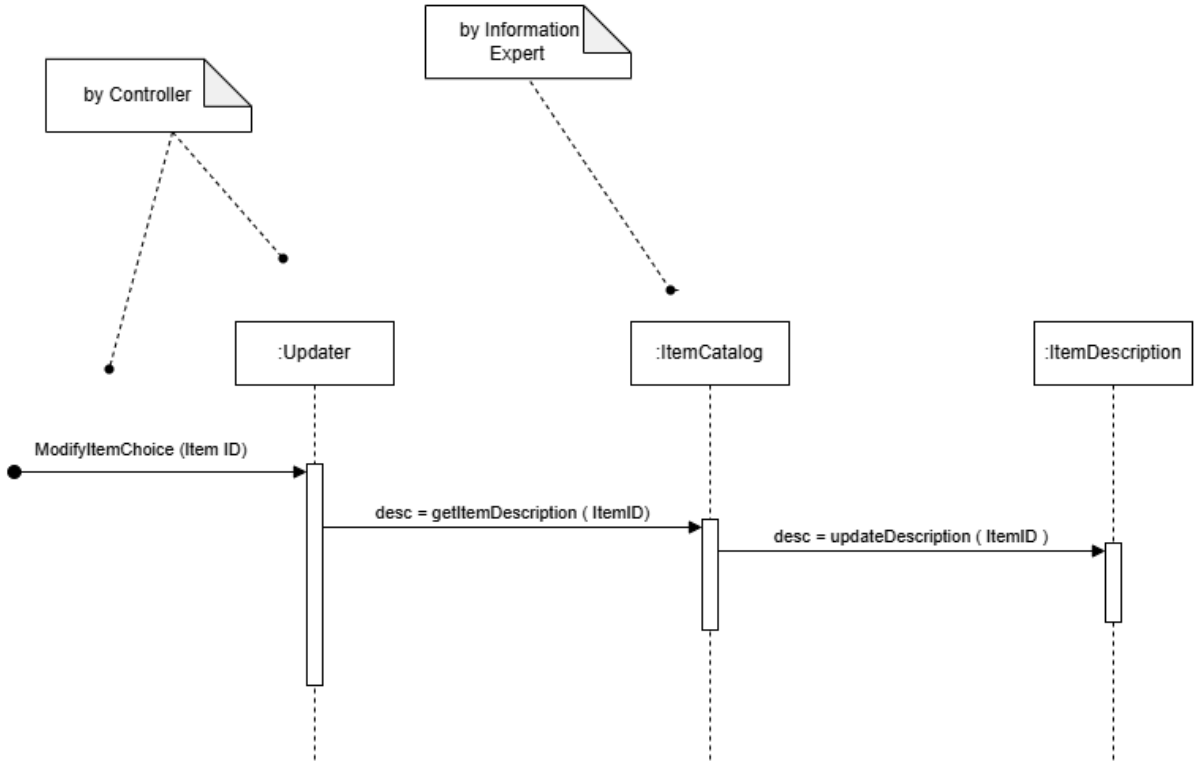


Customize Menu

System Sequence Diagram (SSD)

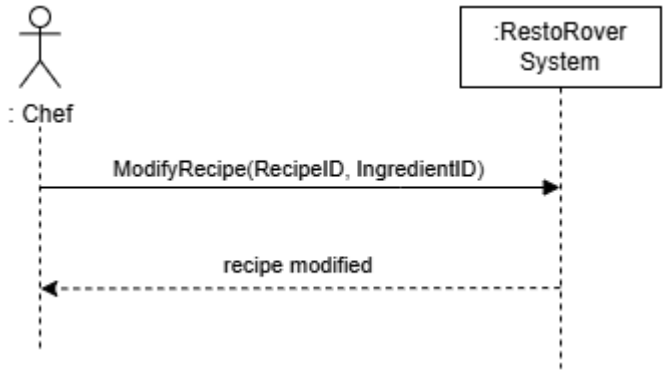


Sequence Diagram (SD)

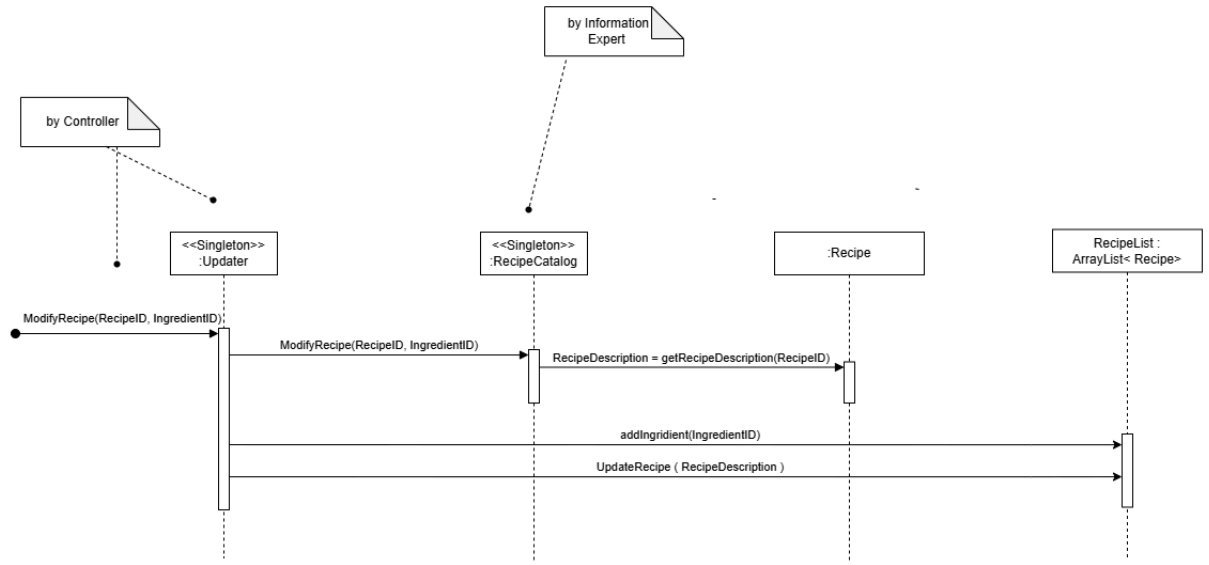


Modify Recipe - Manage Recipe

System Sequence Diagram (SSD)

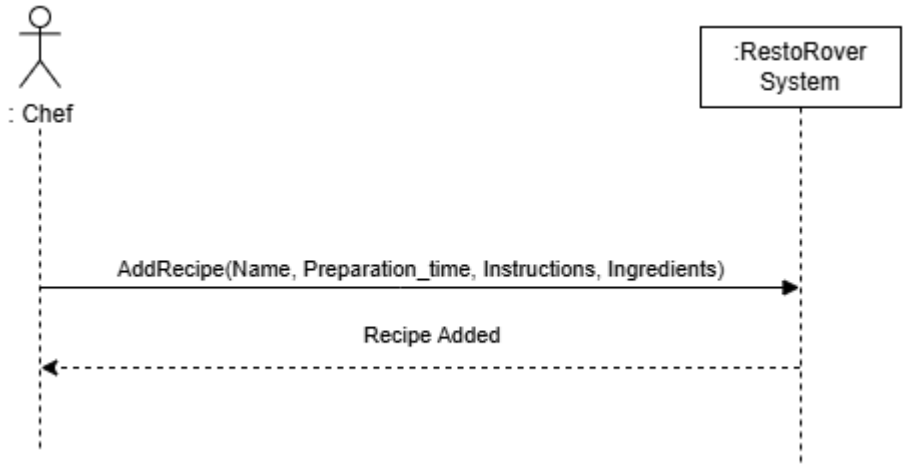


Sequence Diagram (SD)

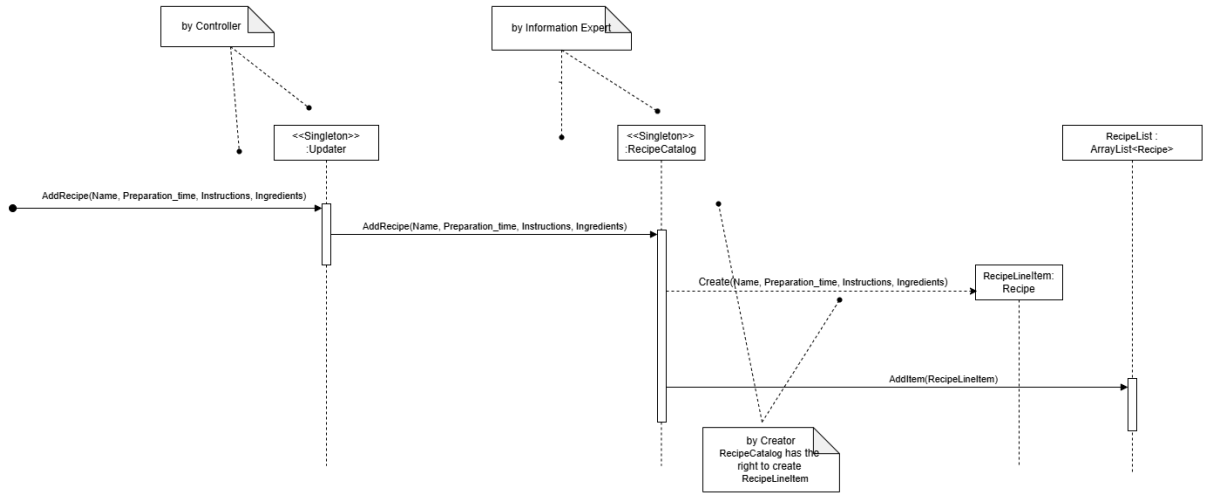


Add Recipe - Manage Recipe

System Sequence Diagram (SSD)

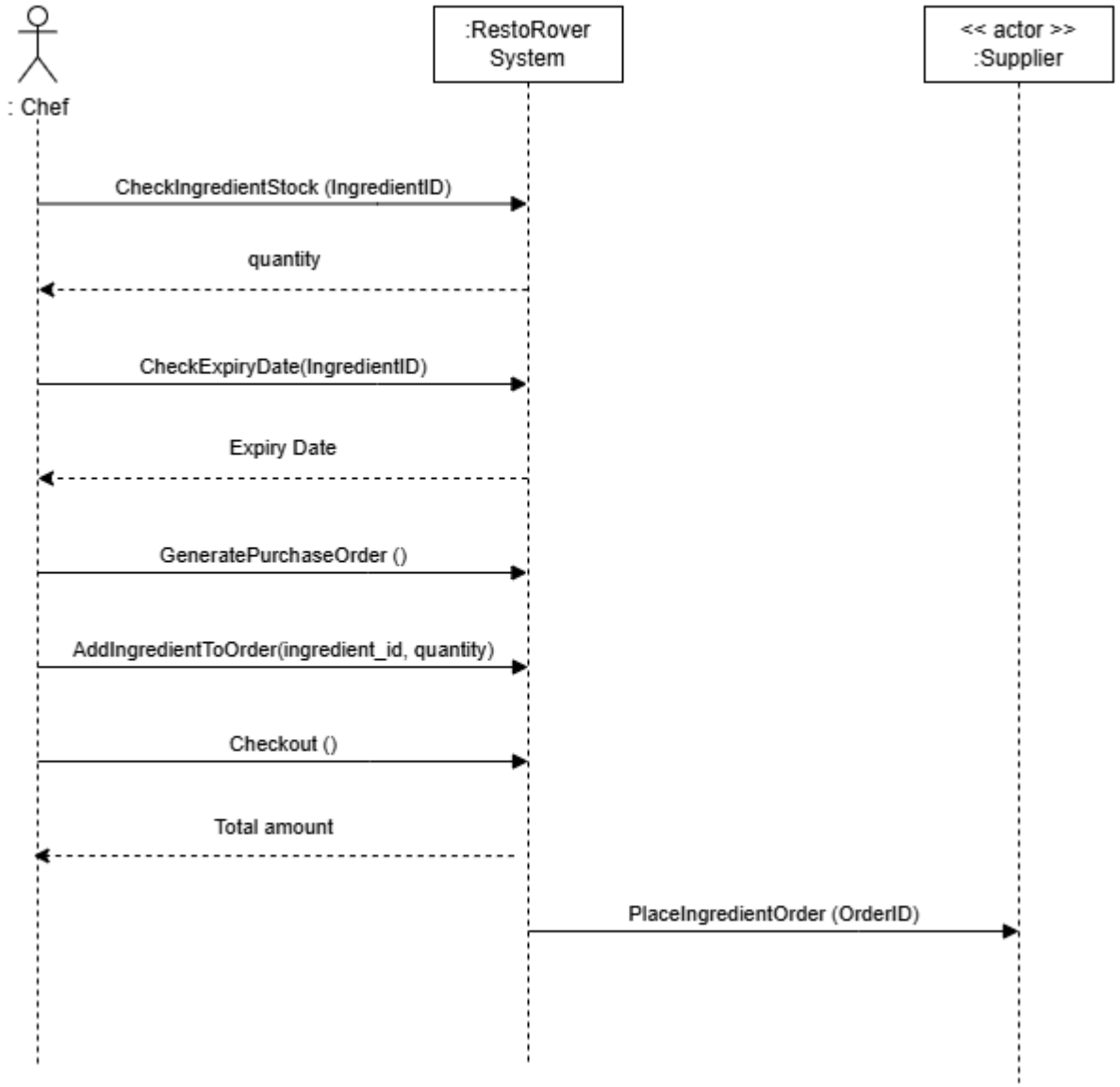


Sequence Diagram (SD)

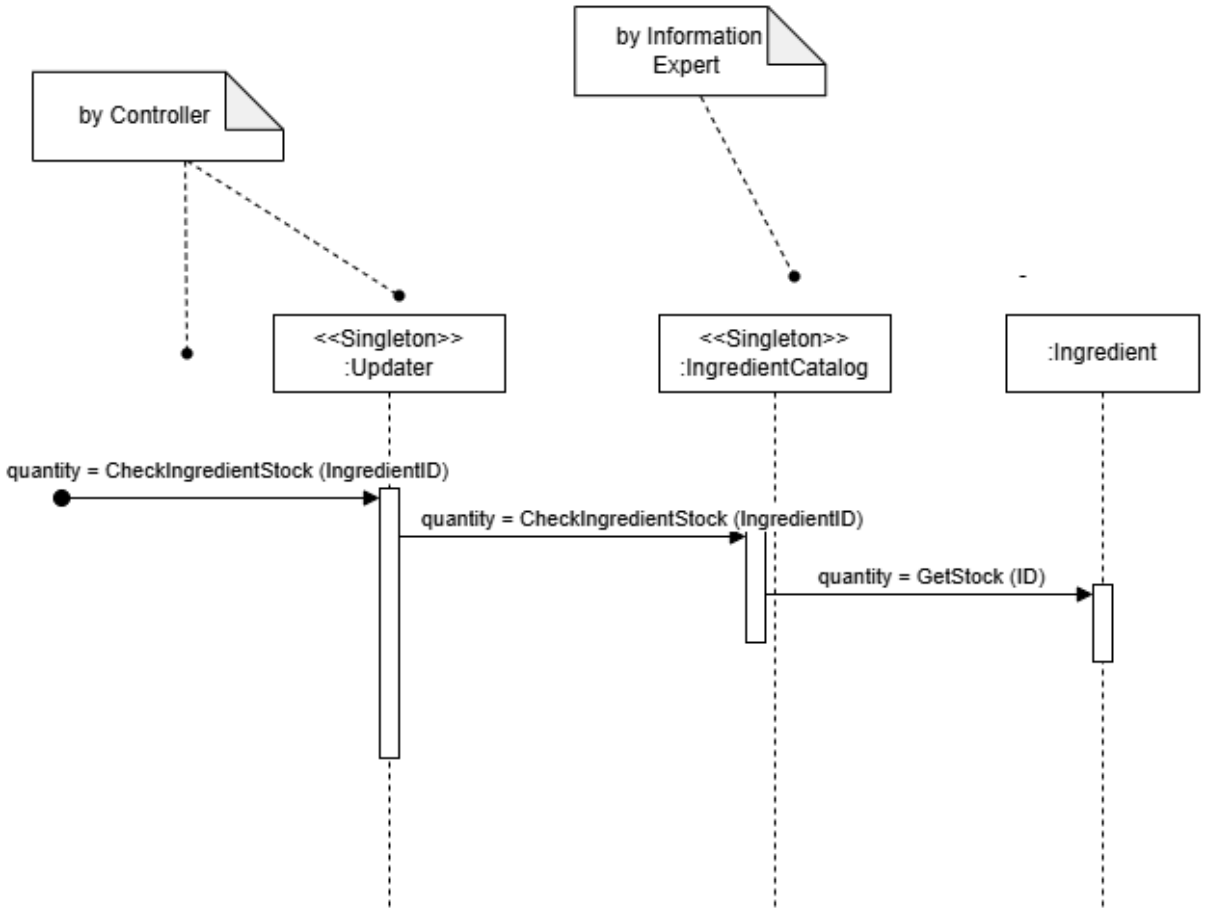


Place Order Ingredients

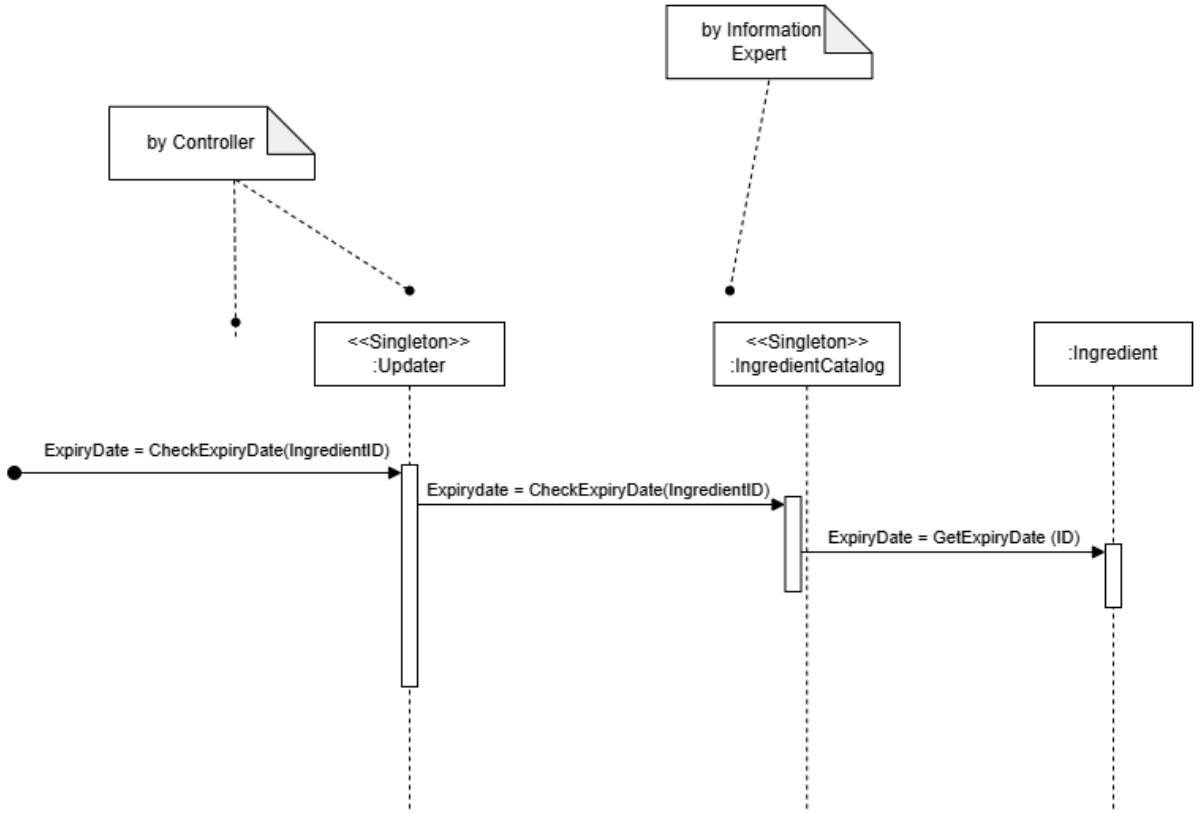
System Sequence Diagram (SSD)



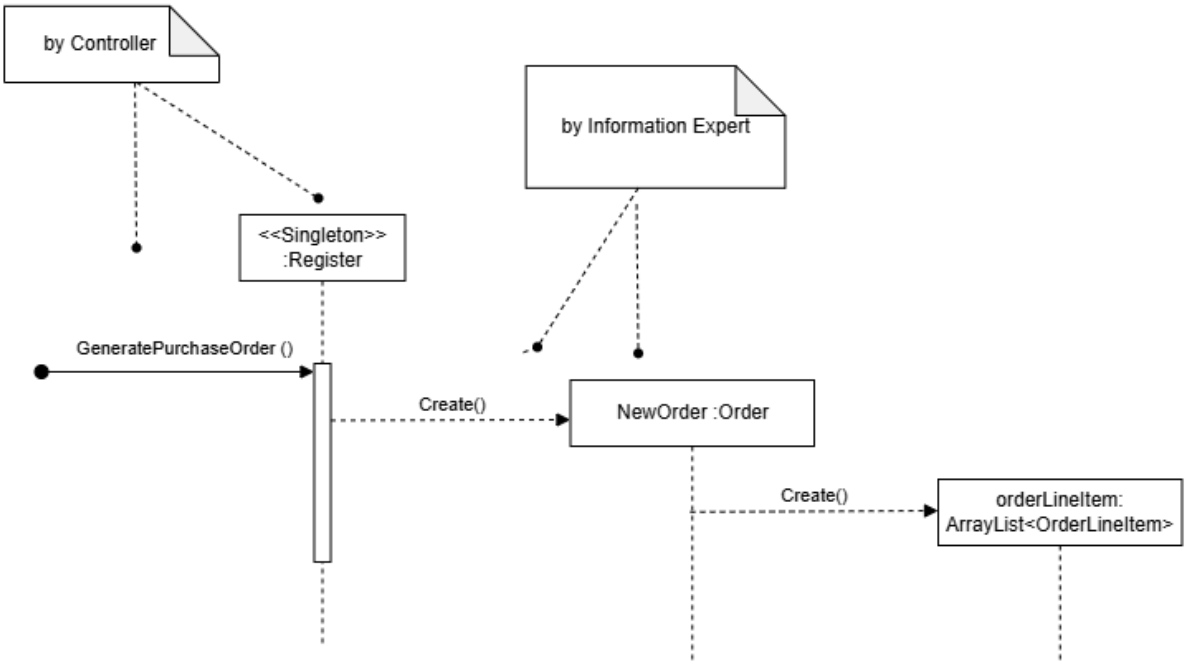
Sequence Diagram (SD) - 1



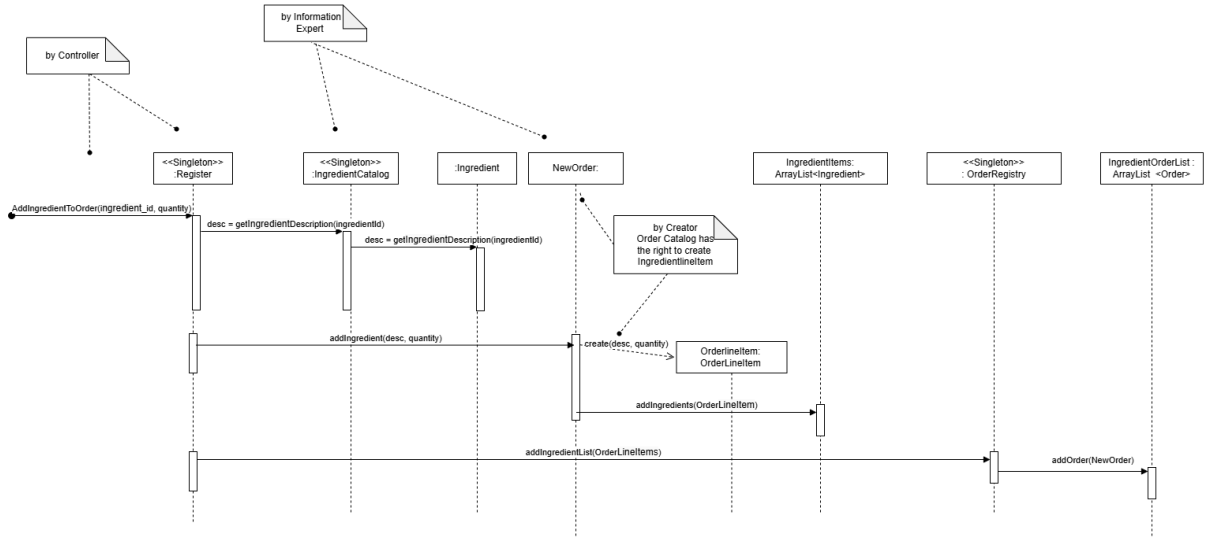
Sequence Diagram (SD) - 2



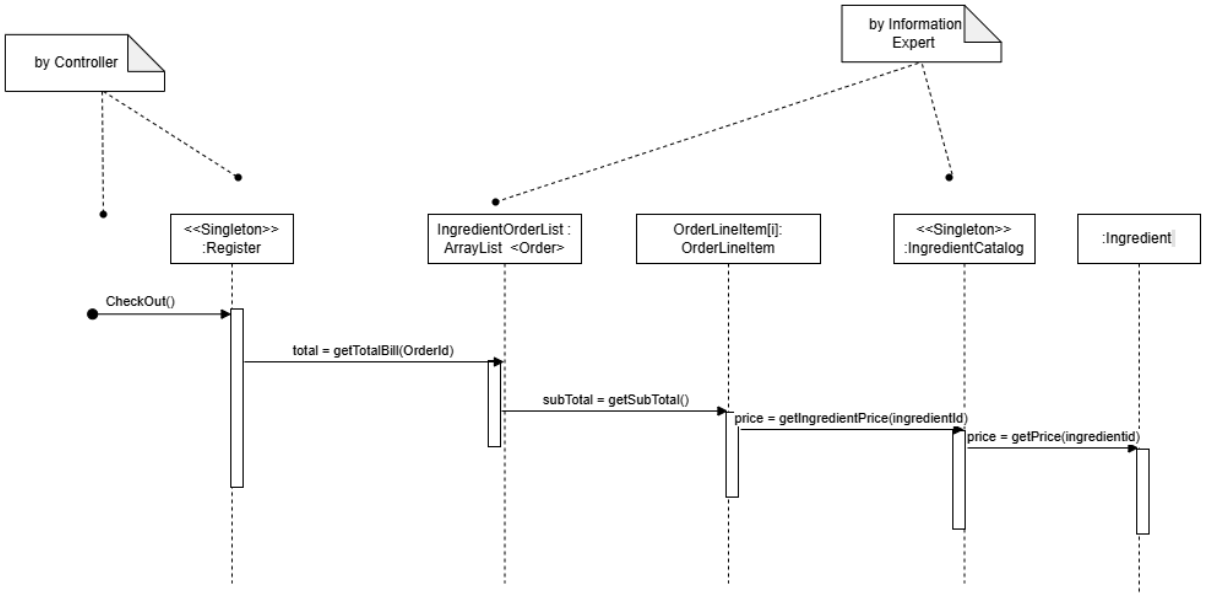
Sequence Diagram (SD) - 3



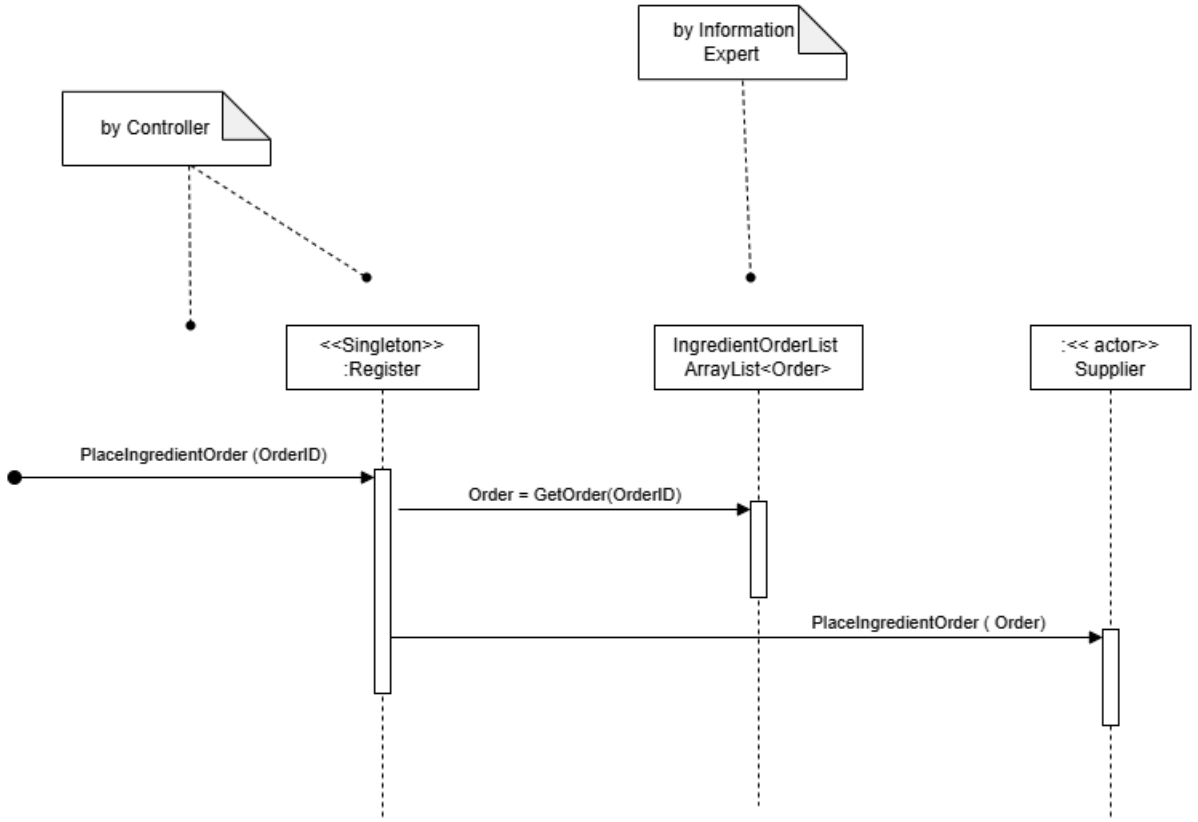
Sequence Diagram (SD) - 4



Sequence Diagram (SD) - 5

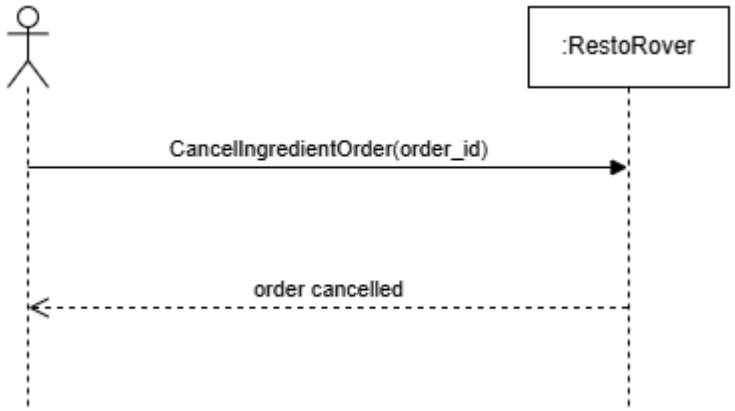


Sequence Diagram (SD) - 6

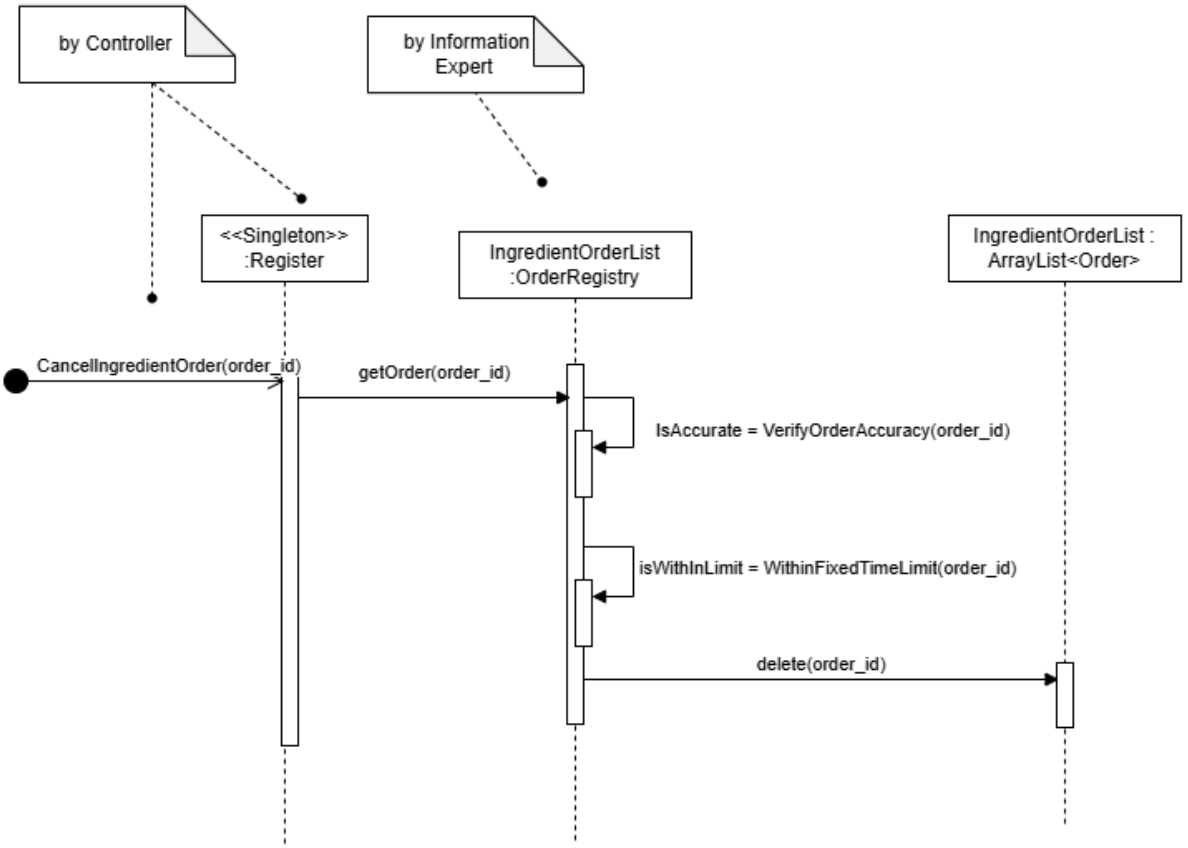


Cancel Ingredient Order - Place Order Ingredients

System Sequence Diagram (SSD)

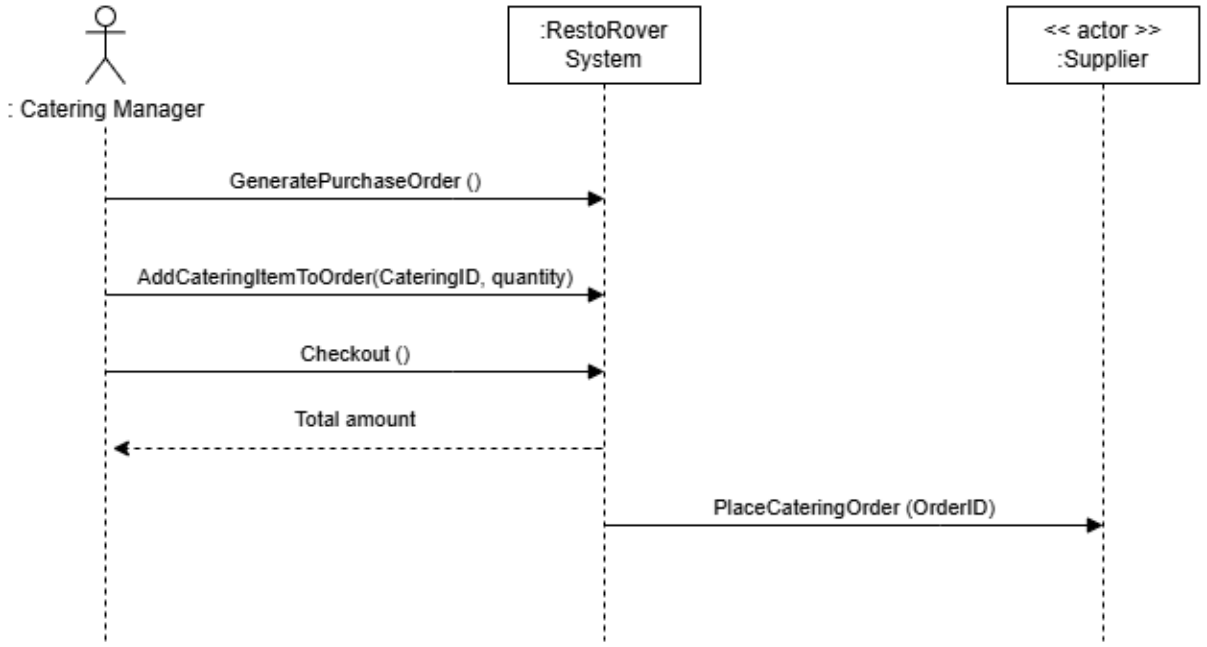


Sequence Diagram (SD)

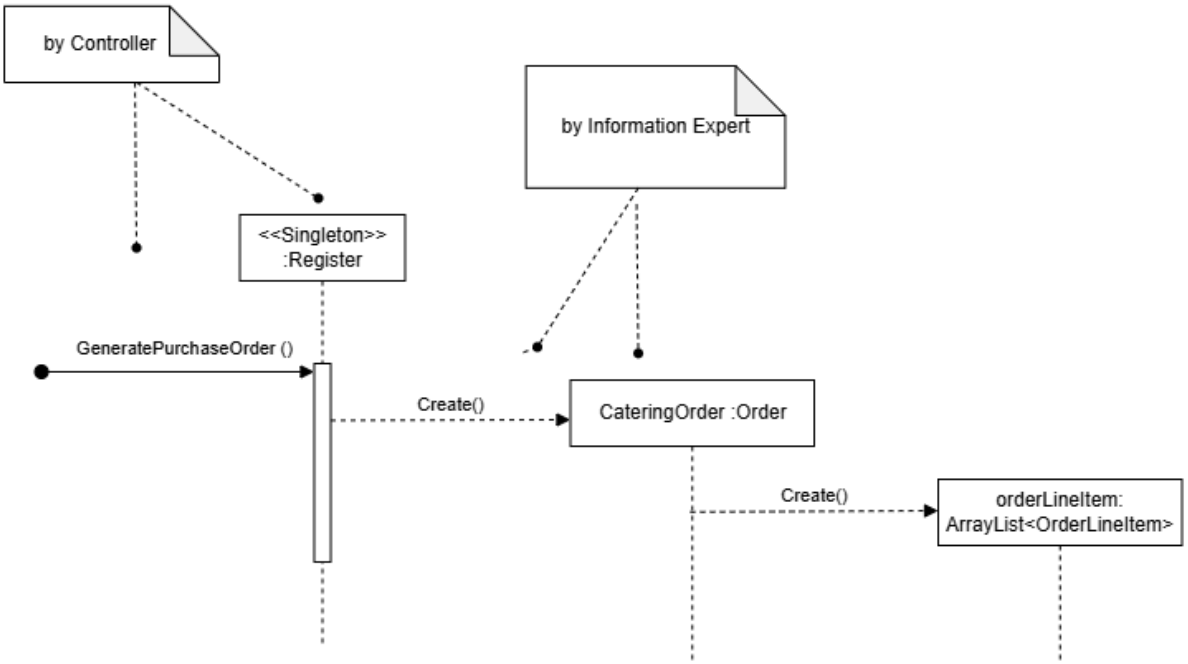


Place Catering Order

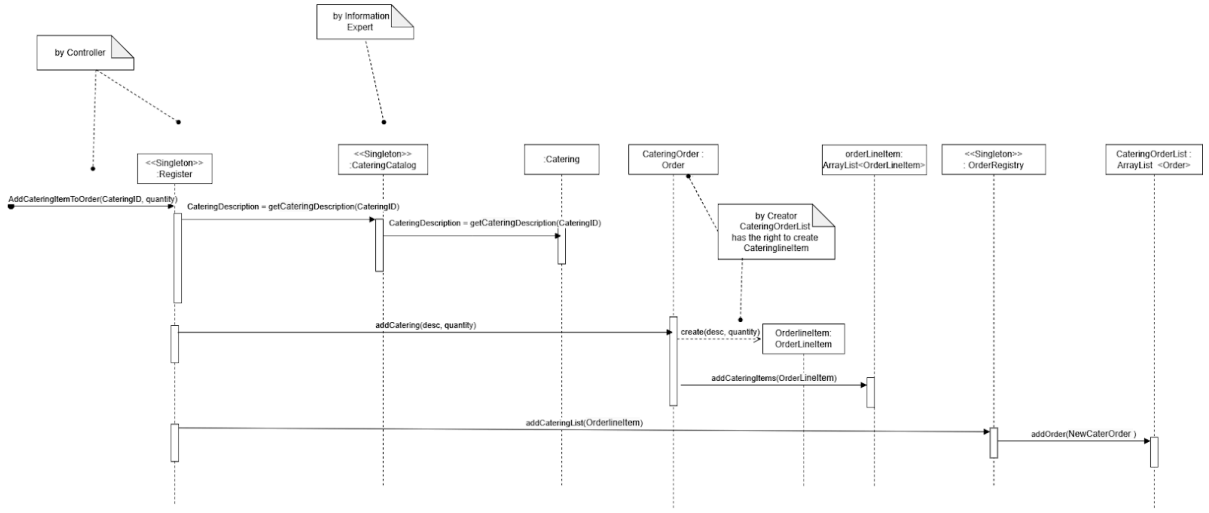
System Sequence Diagram (SSD)



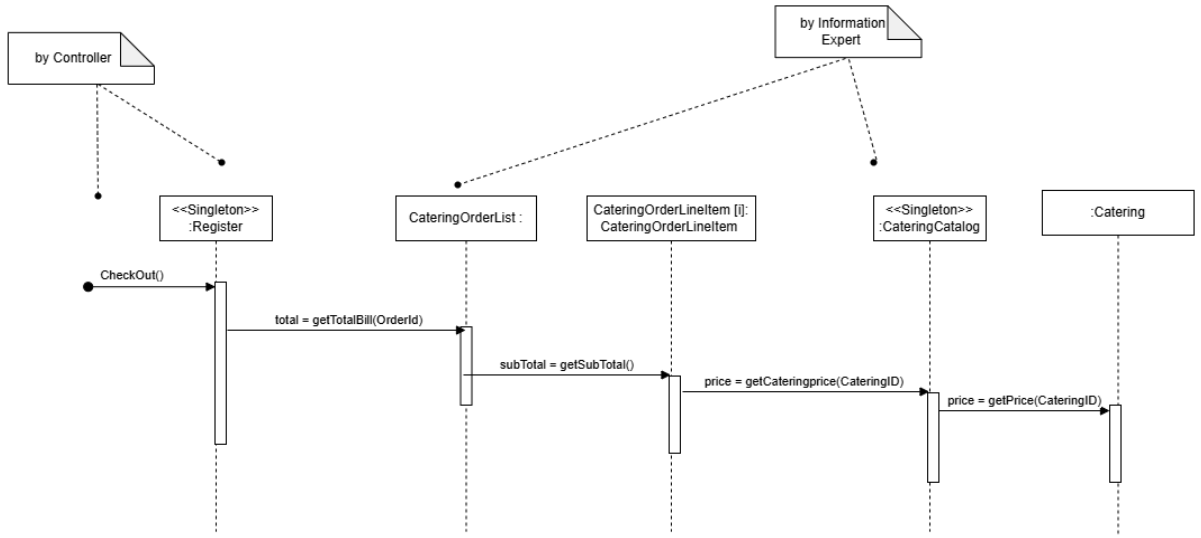
Sequence Diagram (SD) - 1



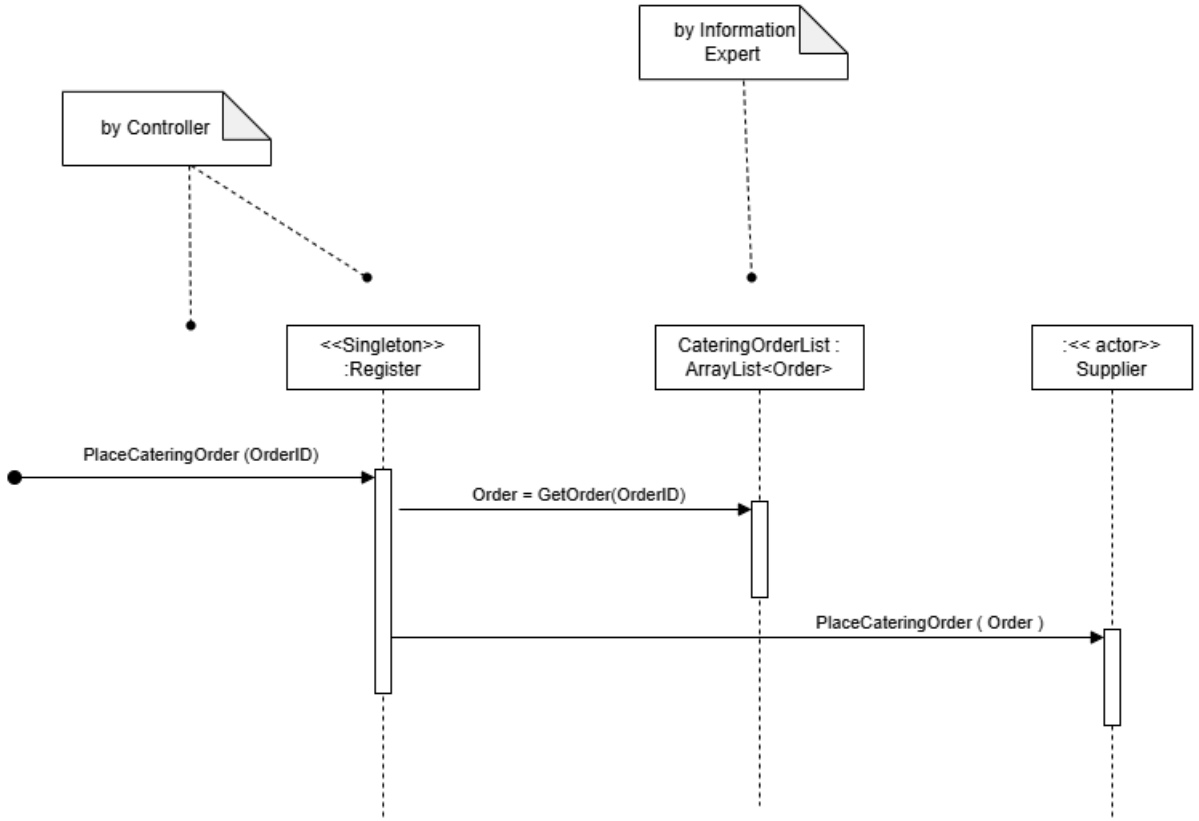
Sequence Diagram (SD) - 2



Sequence Diagram (SD) - 3

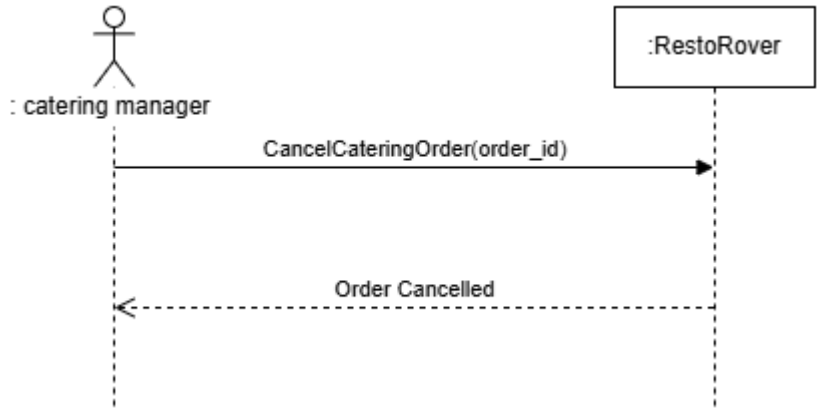


Sequence Diagram (SD) - 4

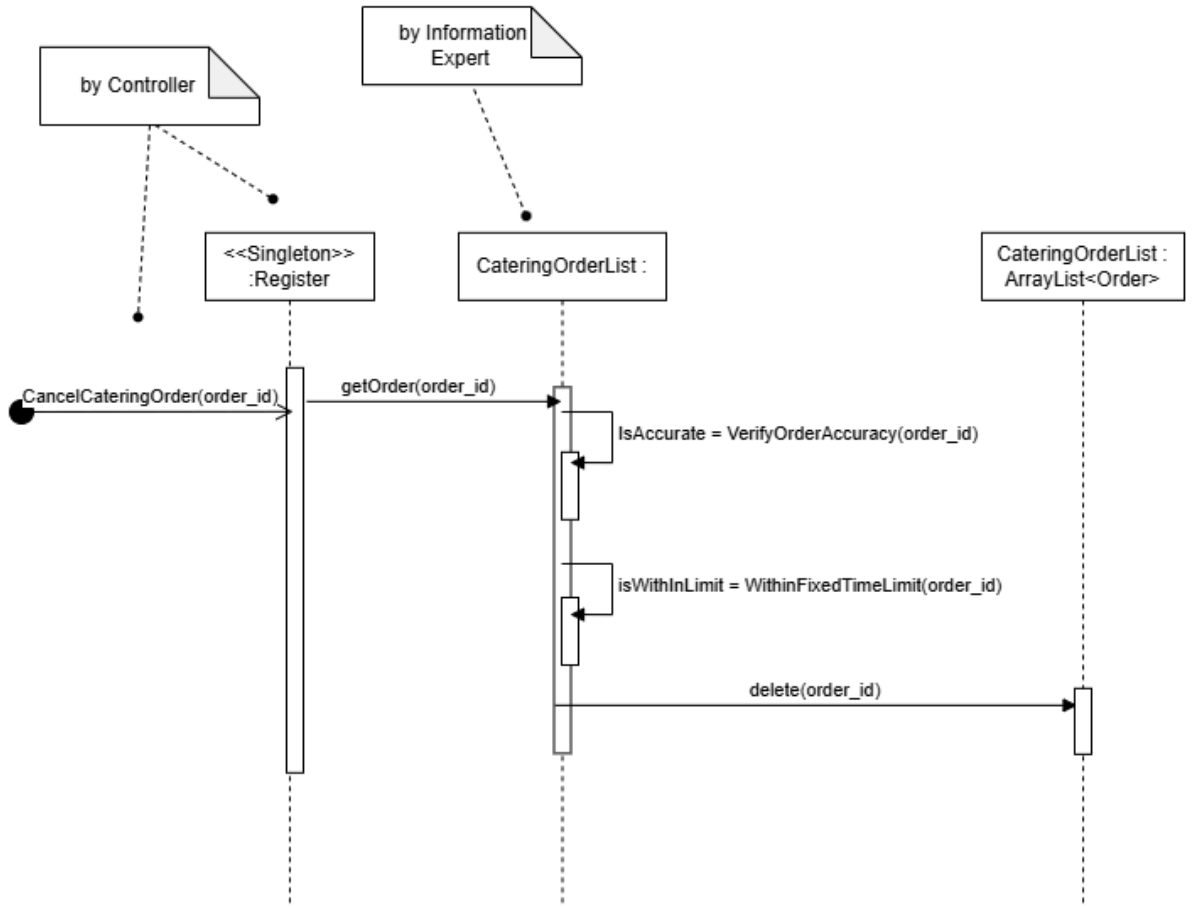


Cancel Catering Ordr - Place Catering Order

System Sequence Diagram (SSD)



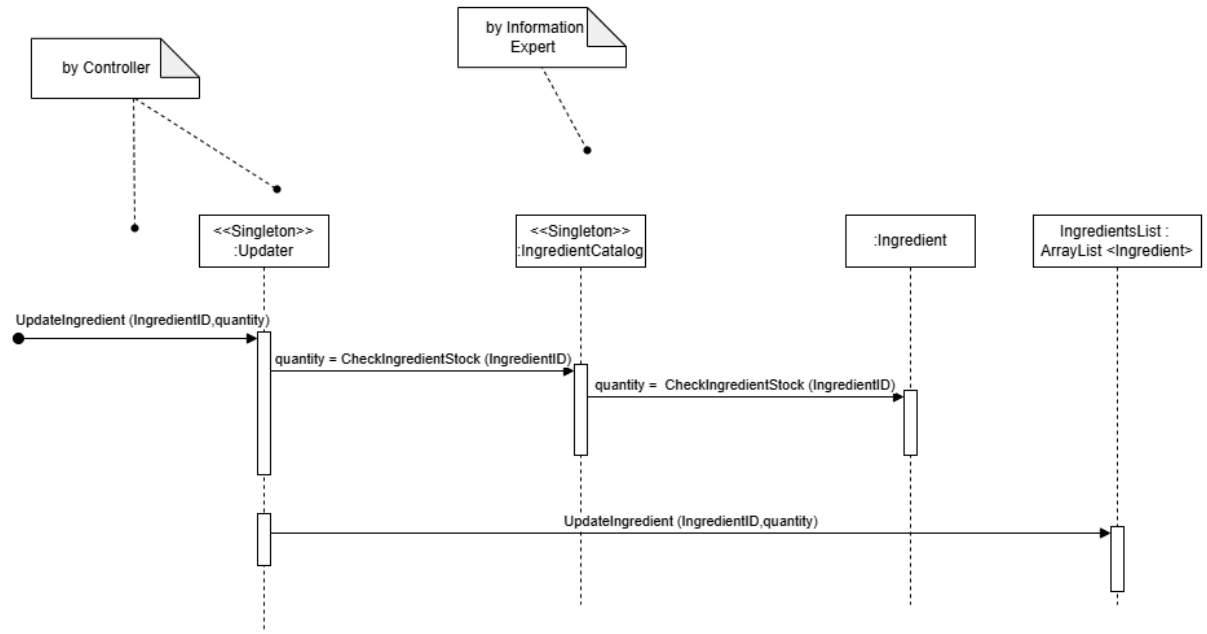
Sequence Diagram (SD)



Update Ingredients - Manage Ingredients

System Sequence Diagram (SSD)

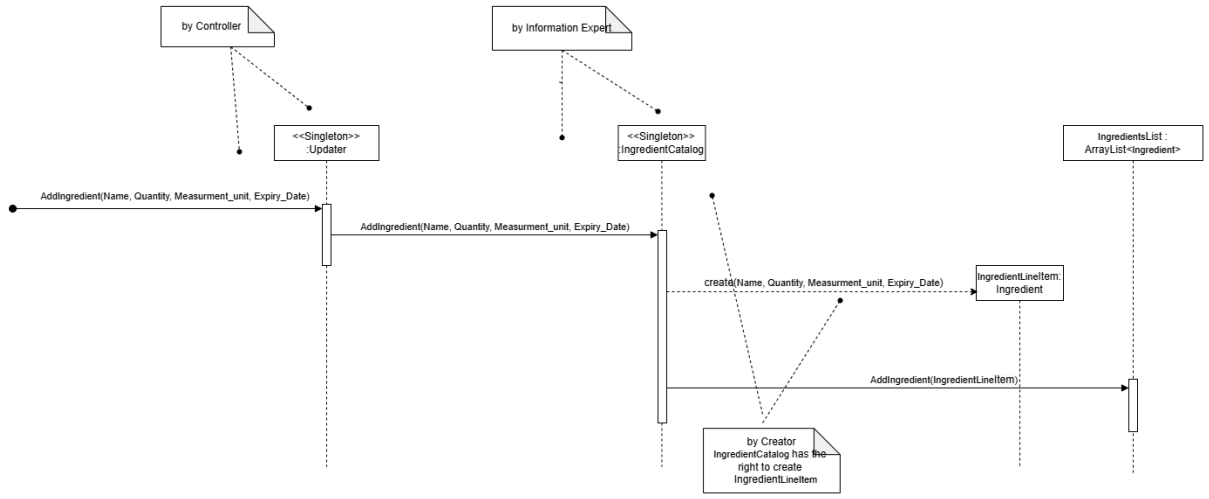
Sequence Diagram (SD)



Add Ingredients - Manage Ingredients

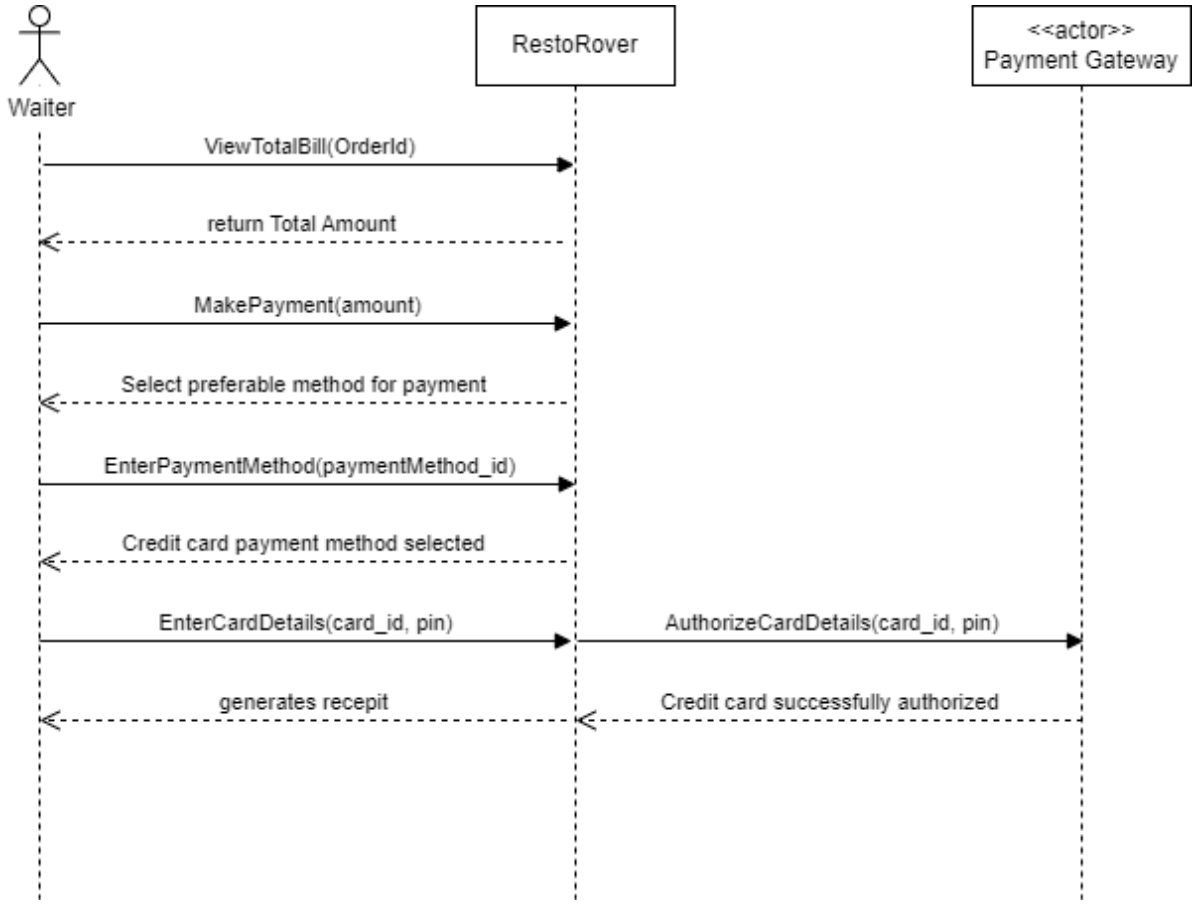
System Sequence Diagram (SSD)

Sequence Diagram (SD)

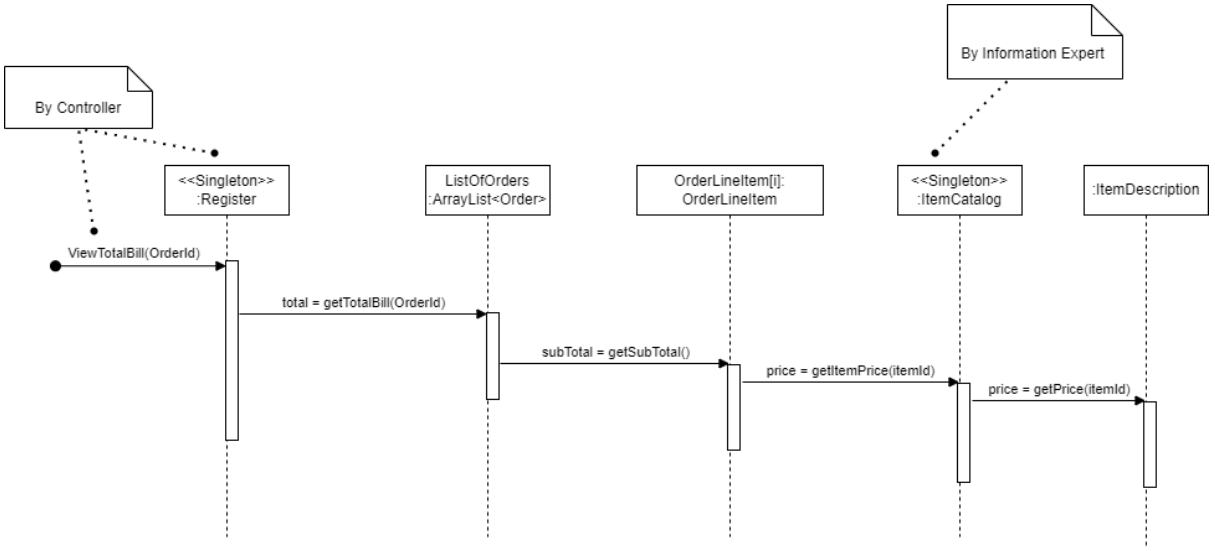


Make Payment

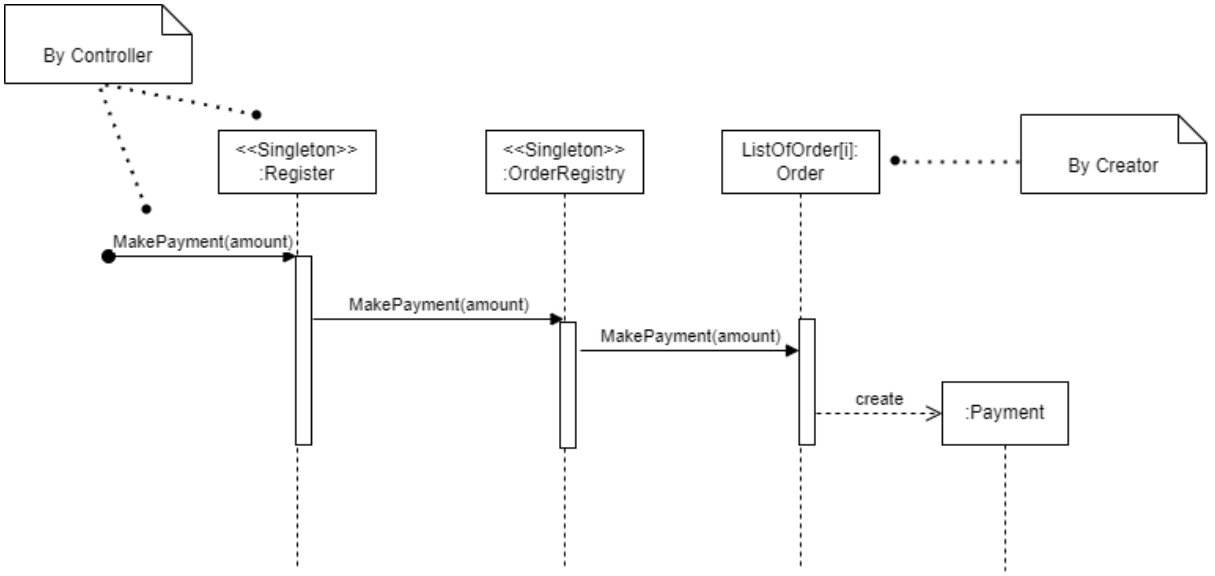
System Sequence Diagram (SSD)



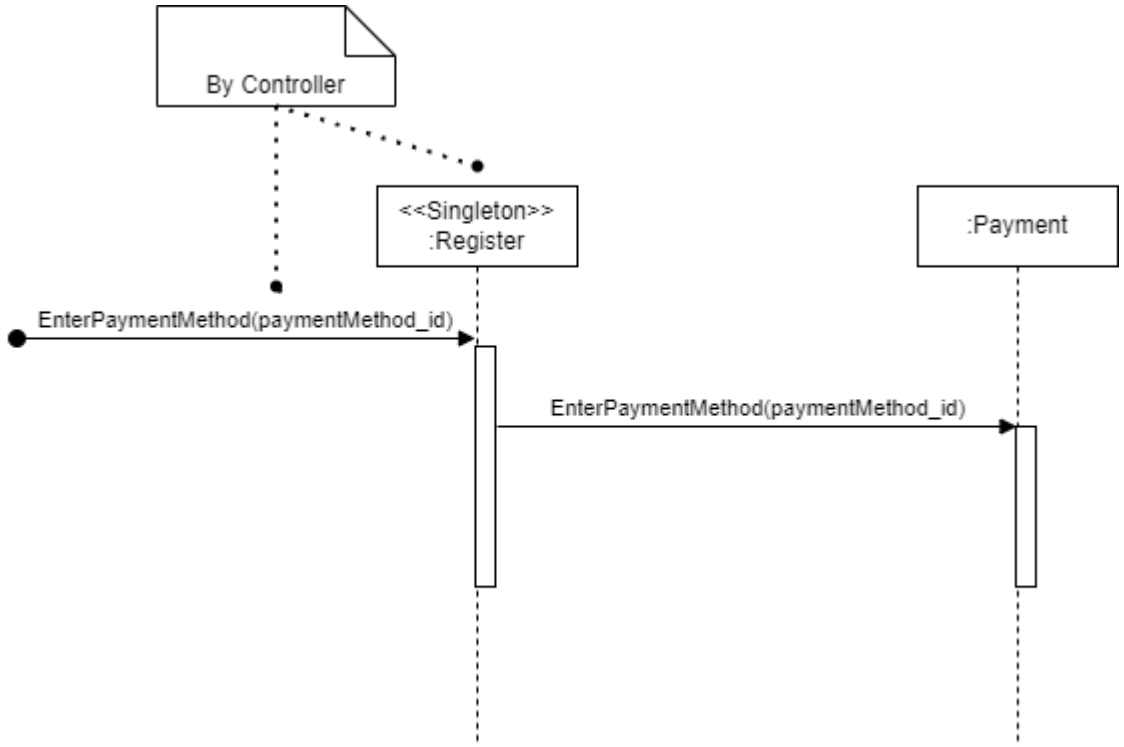
Sequence Diagram (SD) - 01



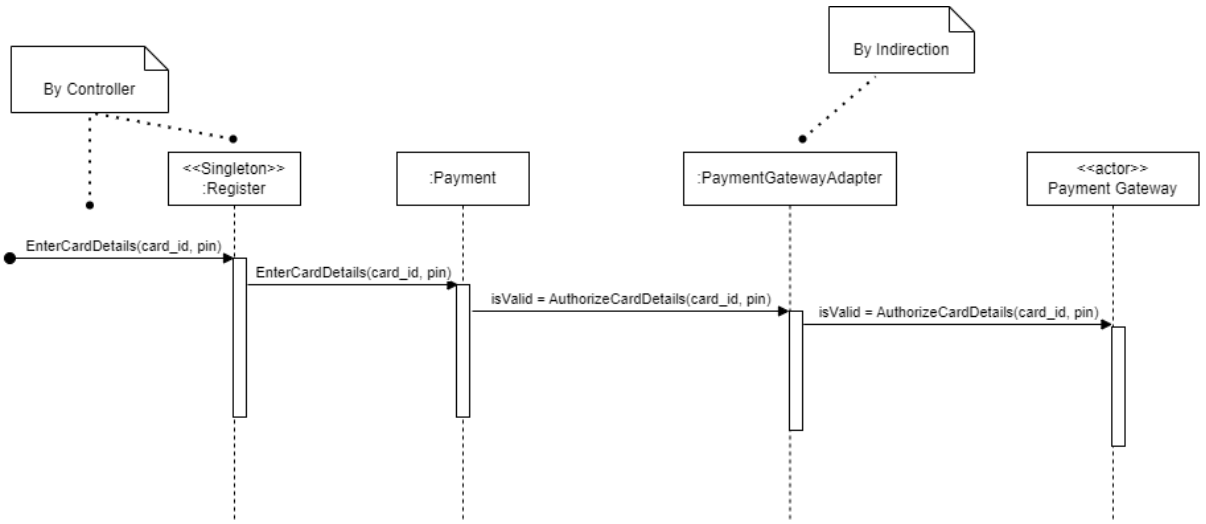
Sequence Diagram (SD) - 02



Sequence Diagram (SD) - 03

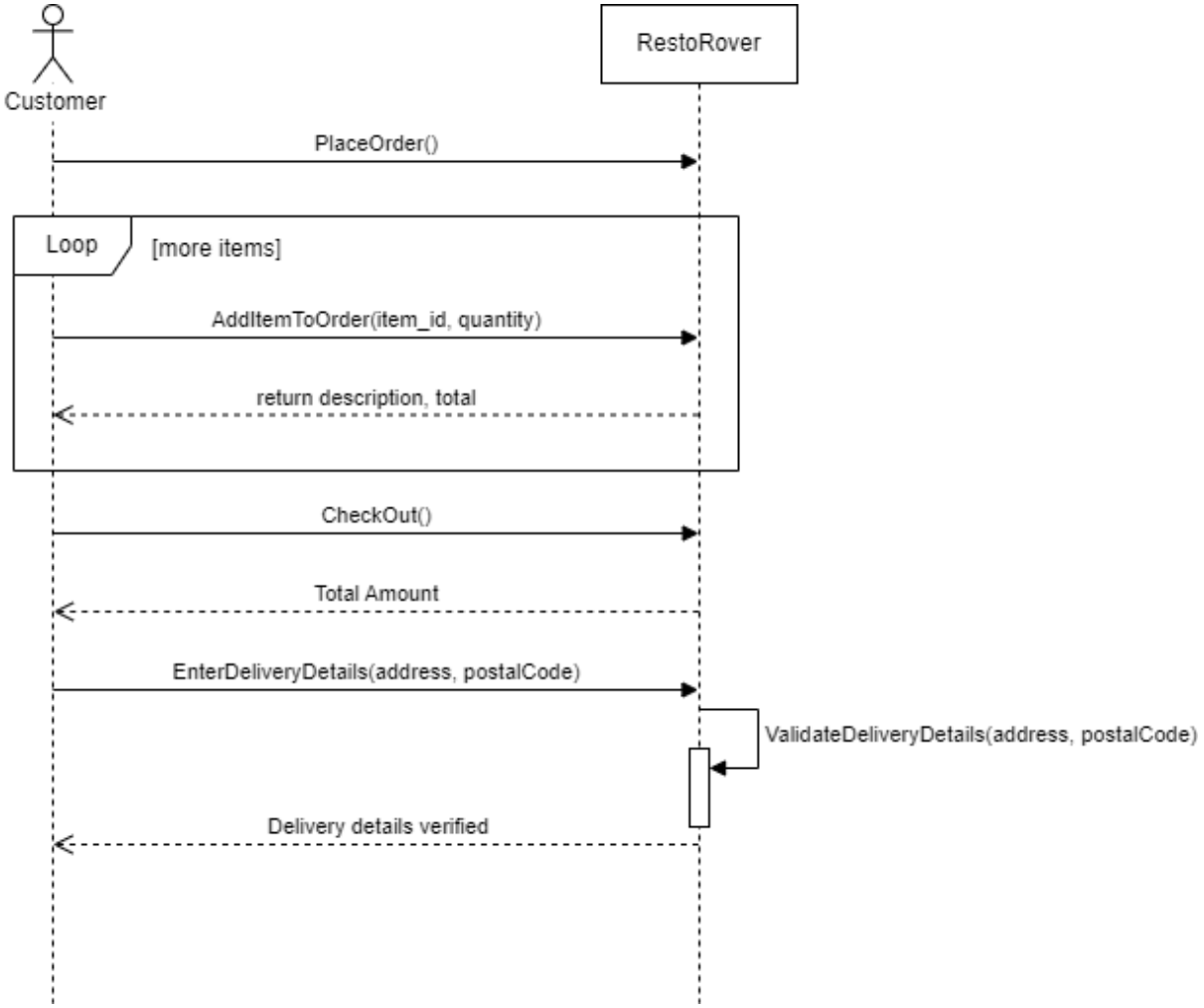


Sequence Diagram (SD) - 04

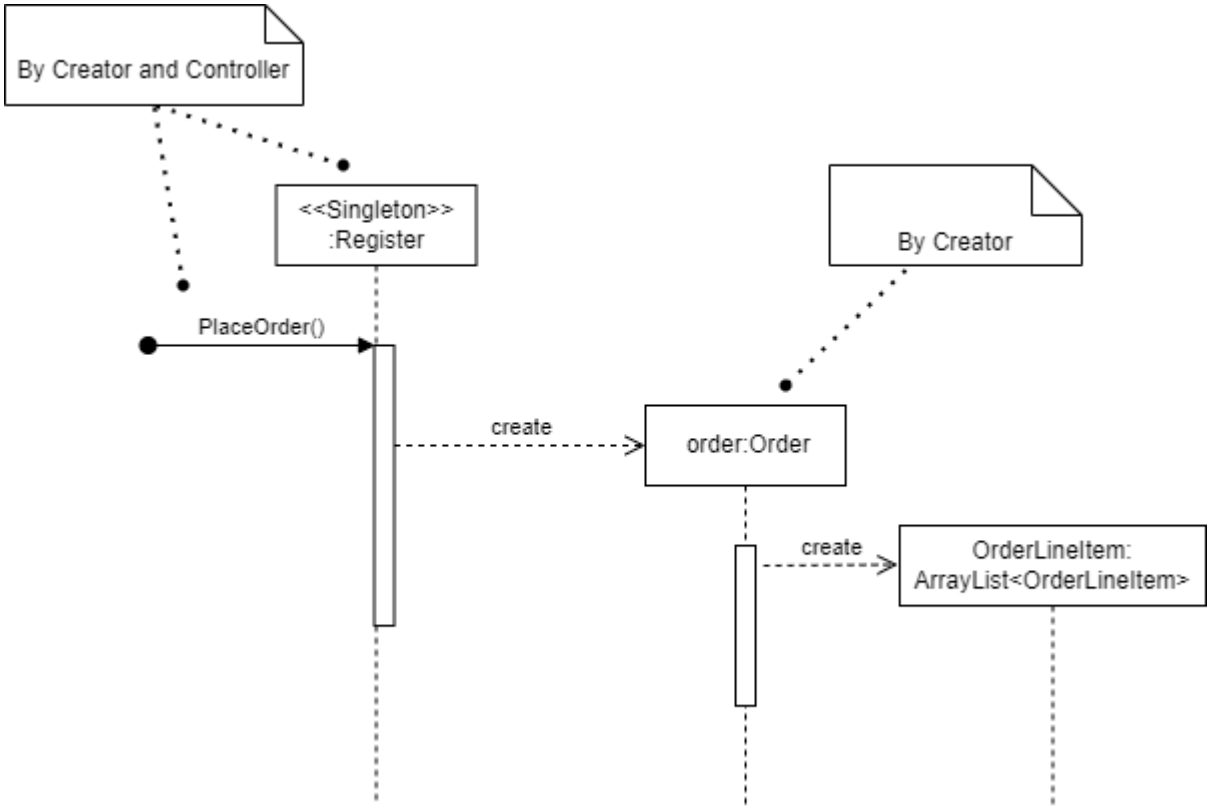


Place Order

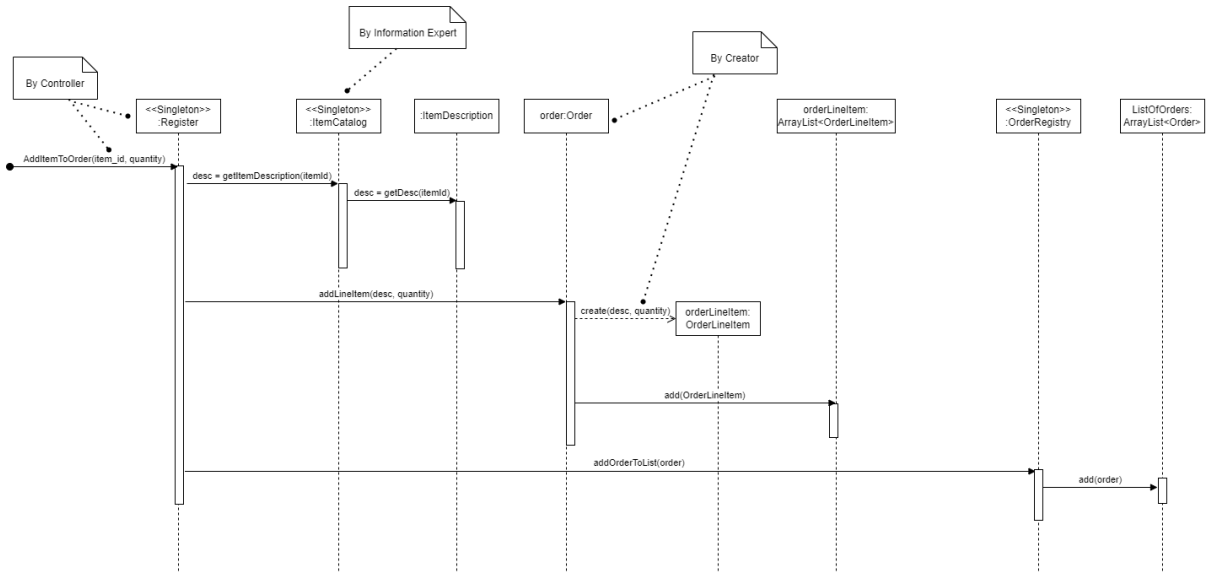
System Sequence Diagram (SSD)



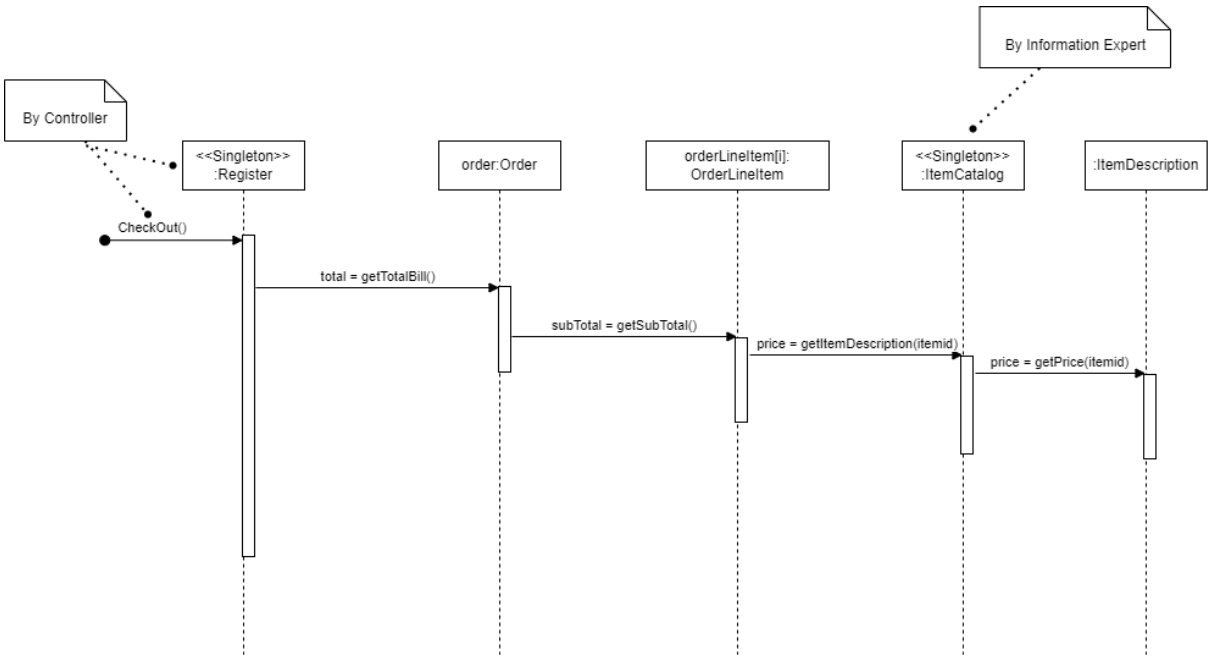
Sequence Diagram (SD) - 01



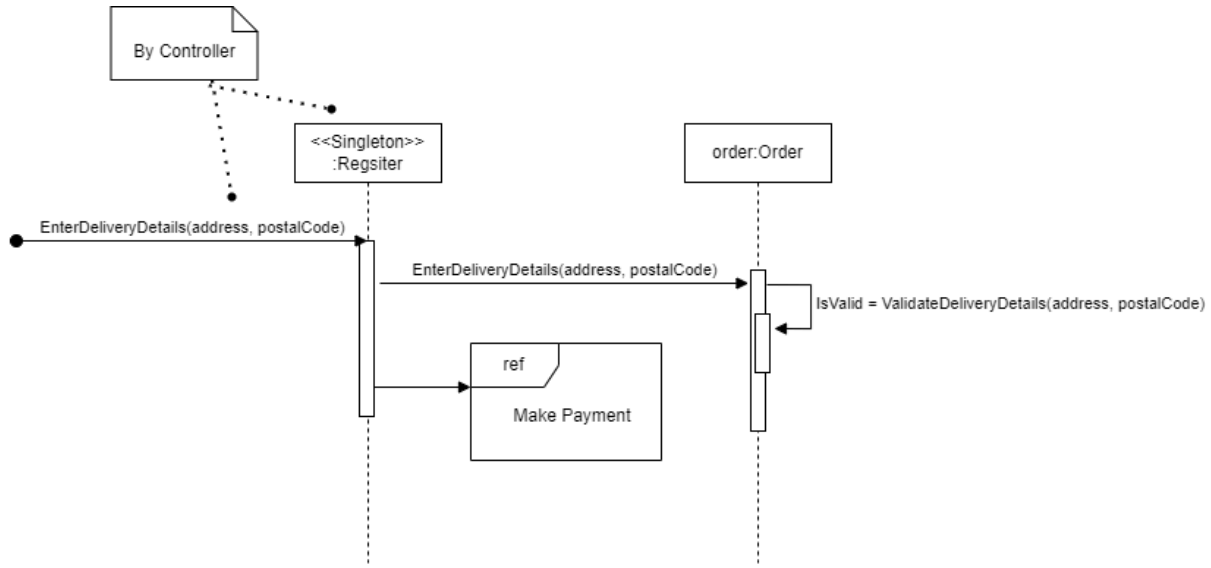
Sequence Diagram (SD) - 02



Sequence Diagram (SD) - 03

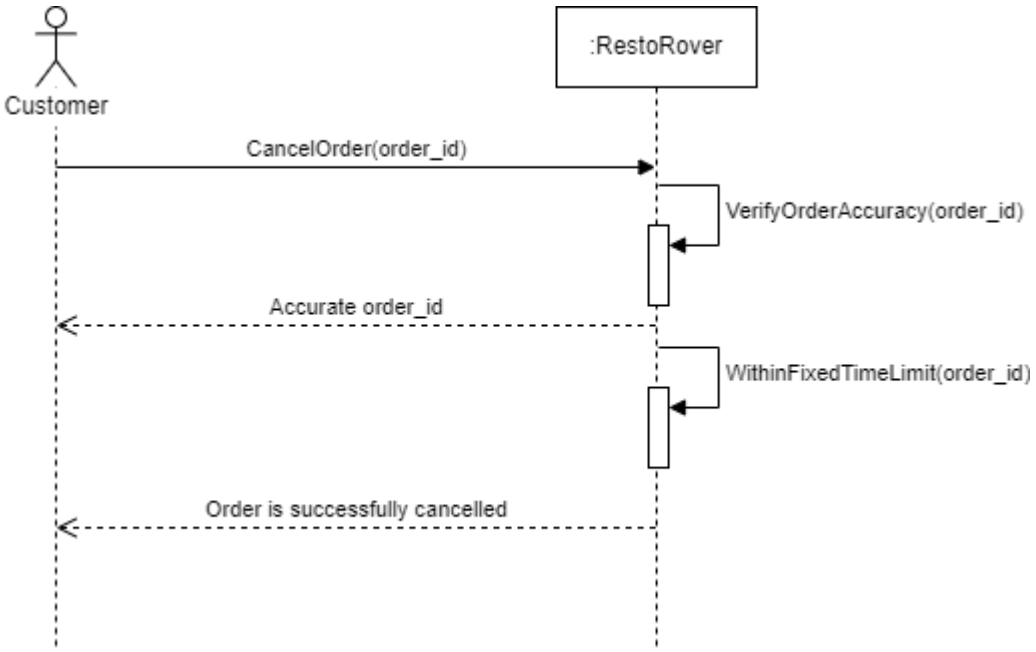


Sequence Diagram (SD) - 04

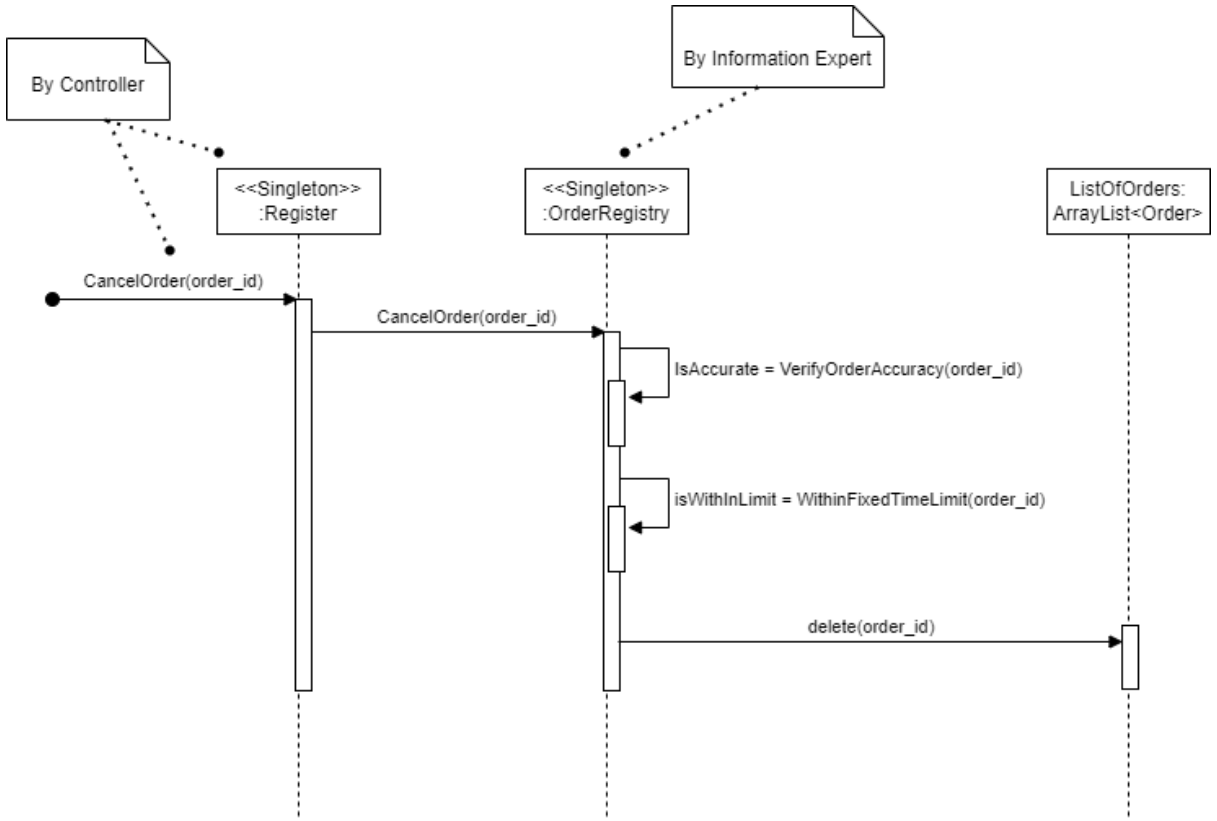


Cancel Order - Place Order

System Sequence Diagram (SSD)

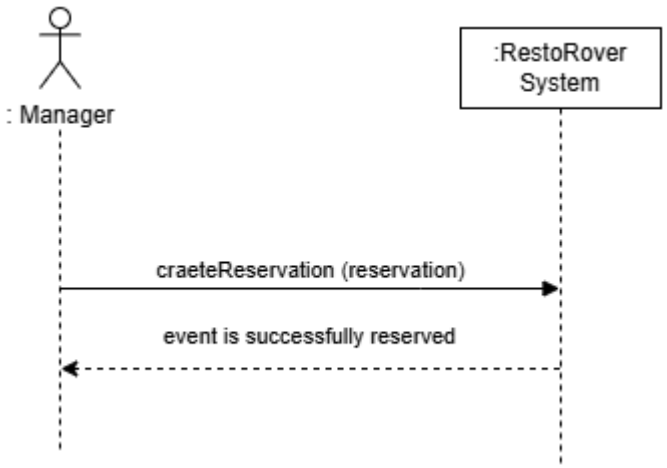


Sequence Diagram (SD) - 01

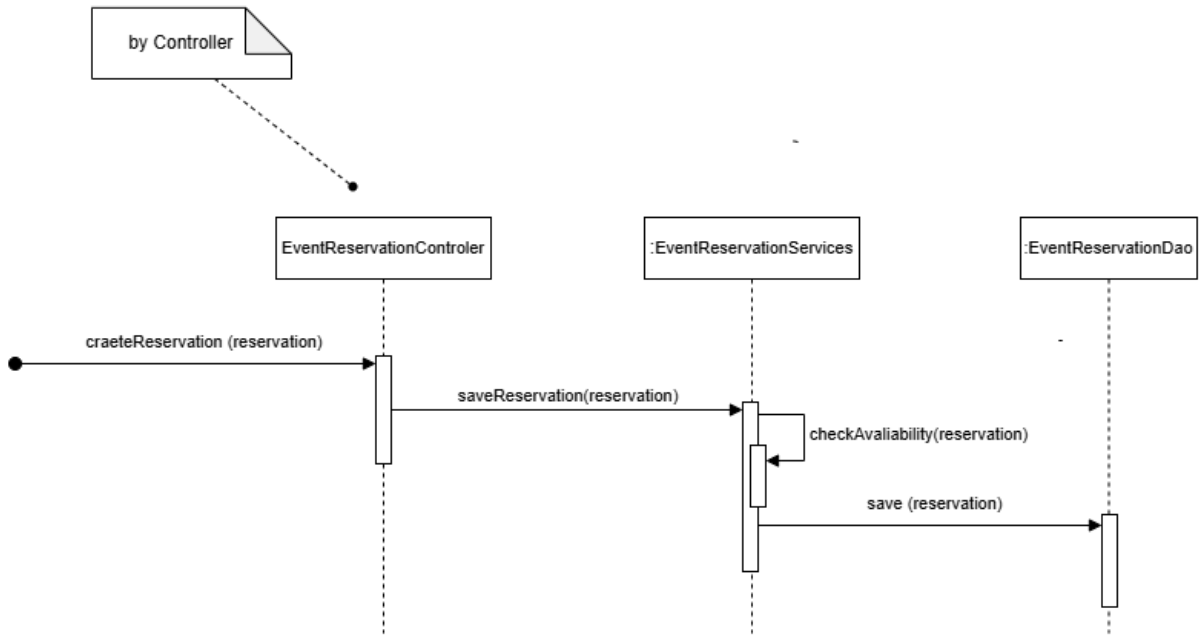


Book Event

System Sequence Diagram (SSD)

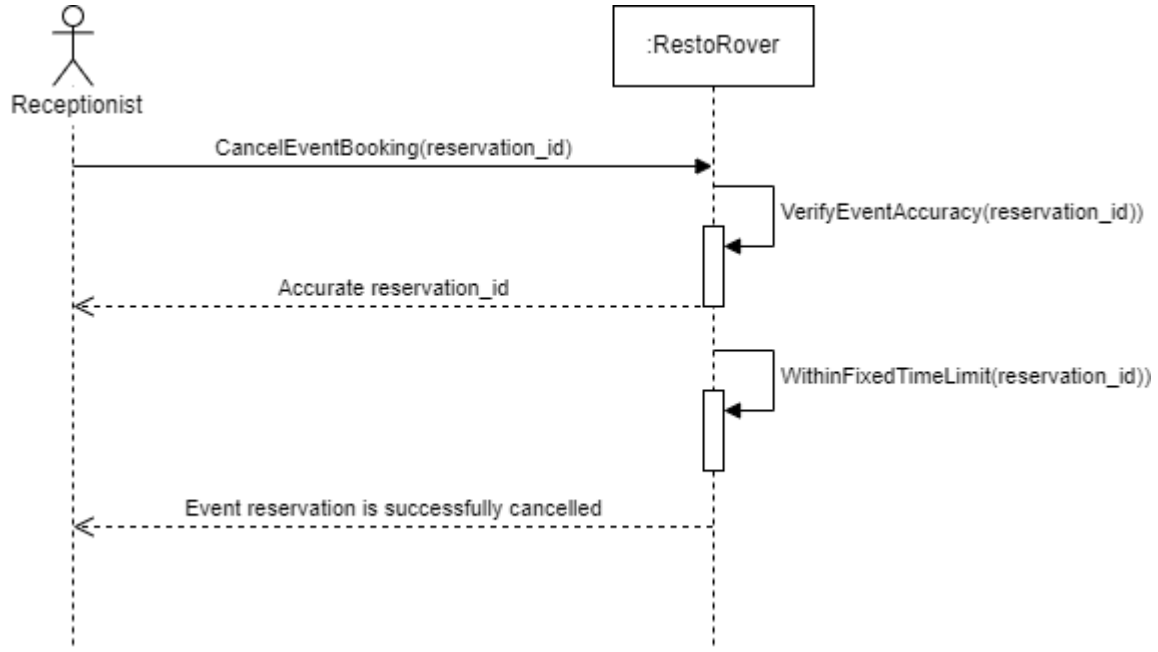


Sequence Diagram (SD) - 01

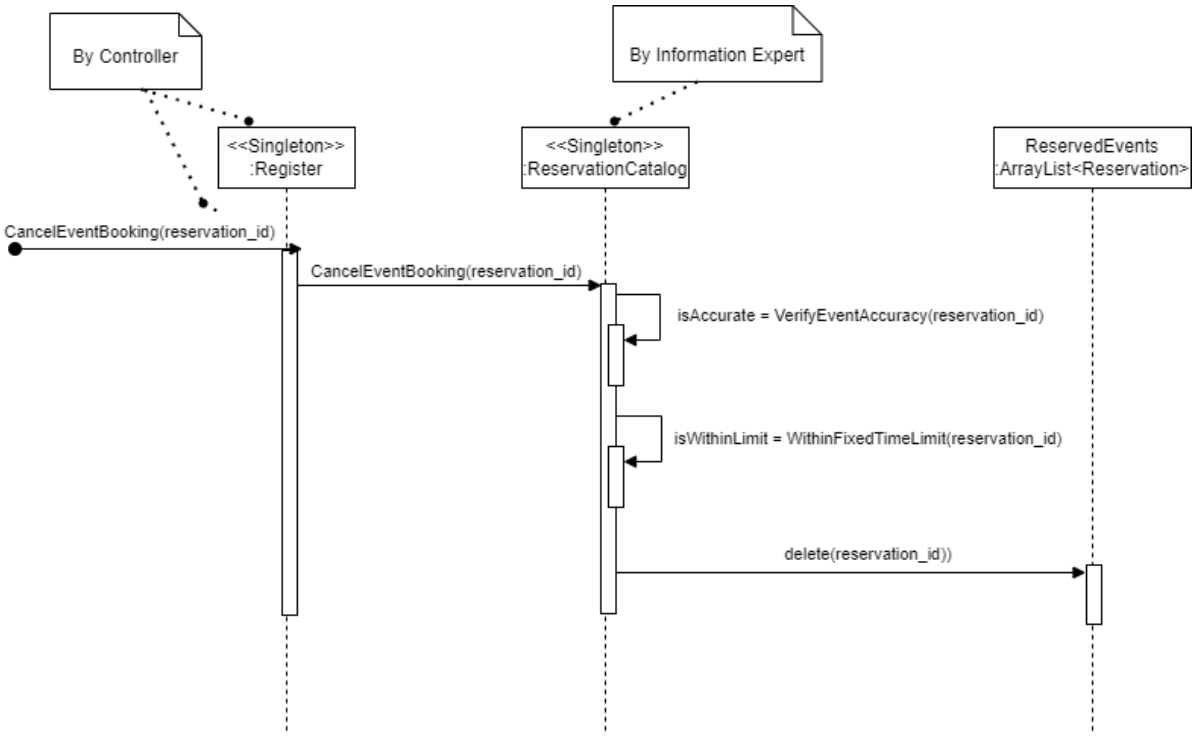


Cancel Book Event - Book Event

System Sequence Diagram (SSD)

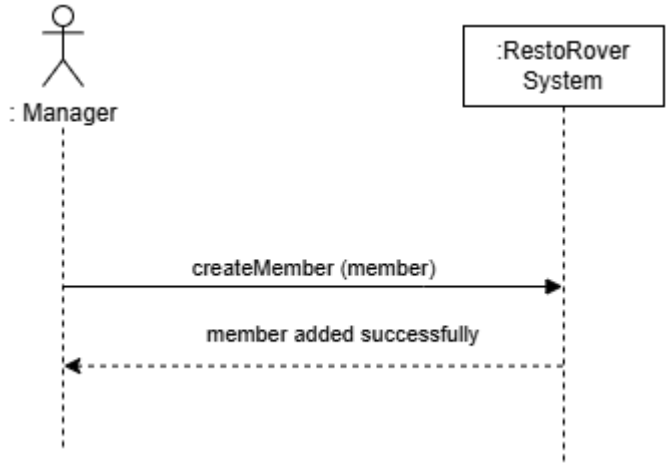


Sequence Diagram (SD) - 01

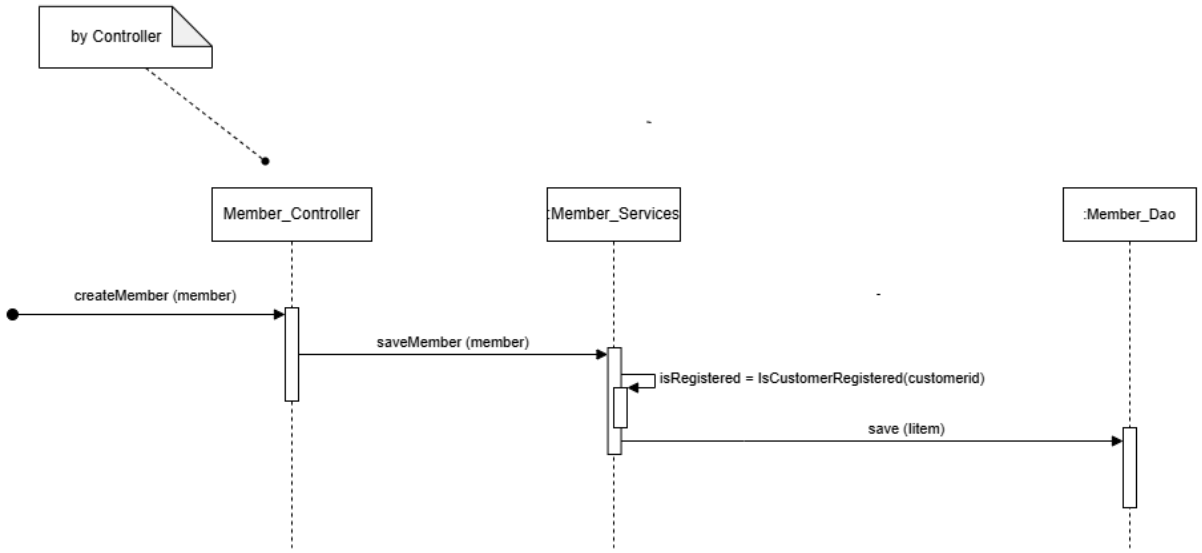


Manage Member Registration

System Sequence Diagram (SSD)

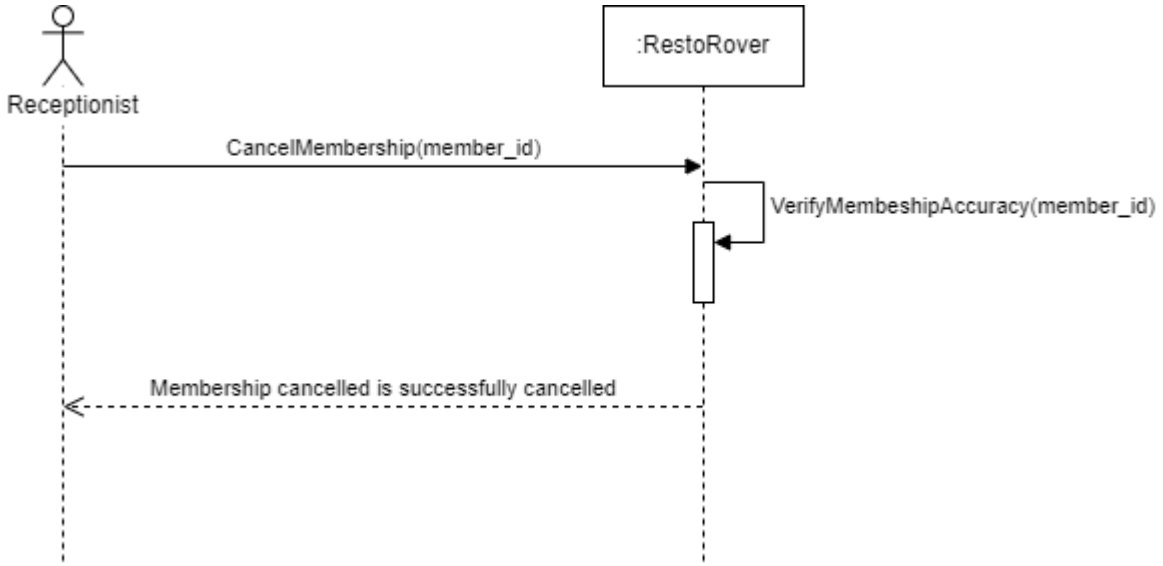


Sequence Diagram (SD) -

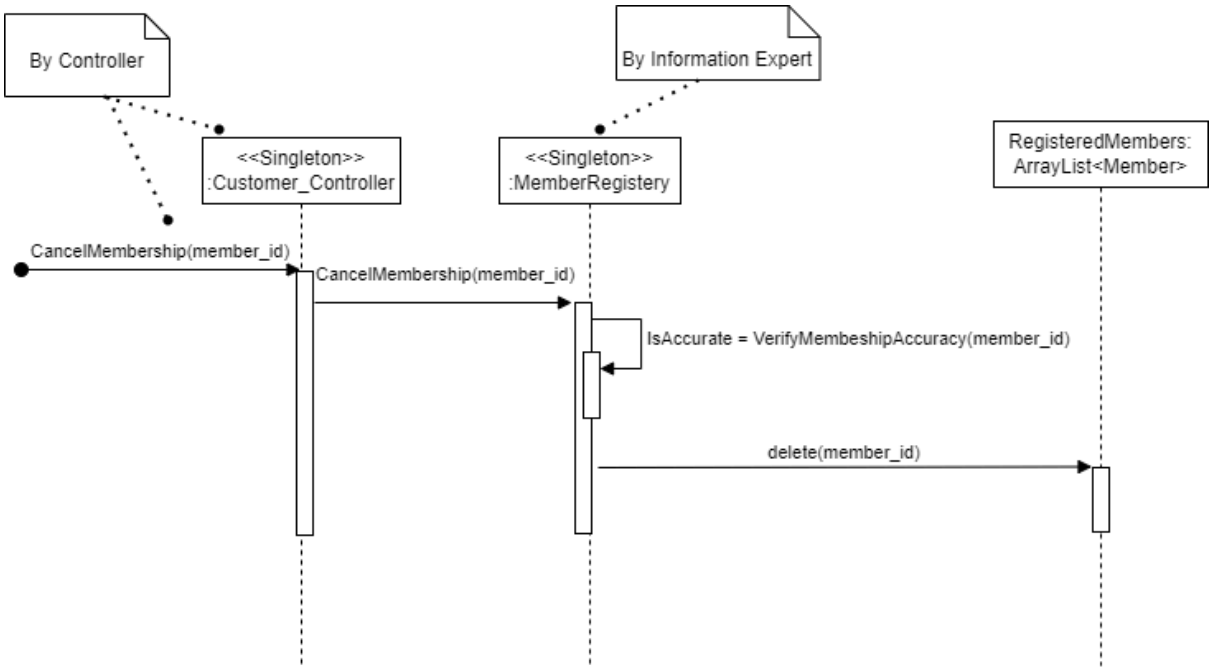


Cancel Member Registration - Manage Member Registration

System Sequence Diagram (SSD)

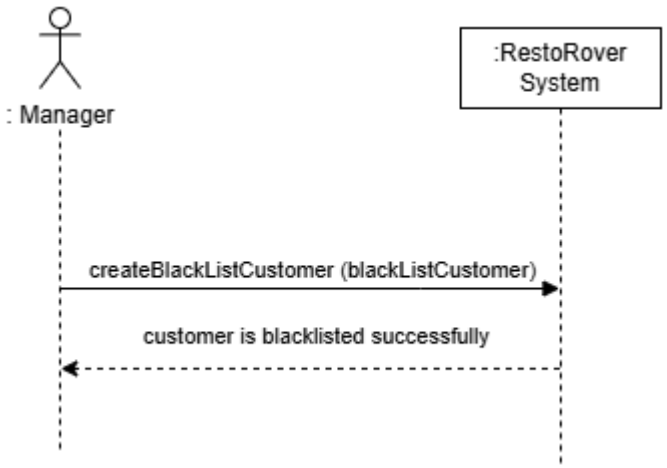


Sequence Diagram (SD) - 01

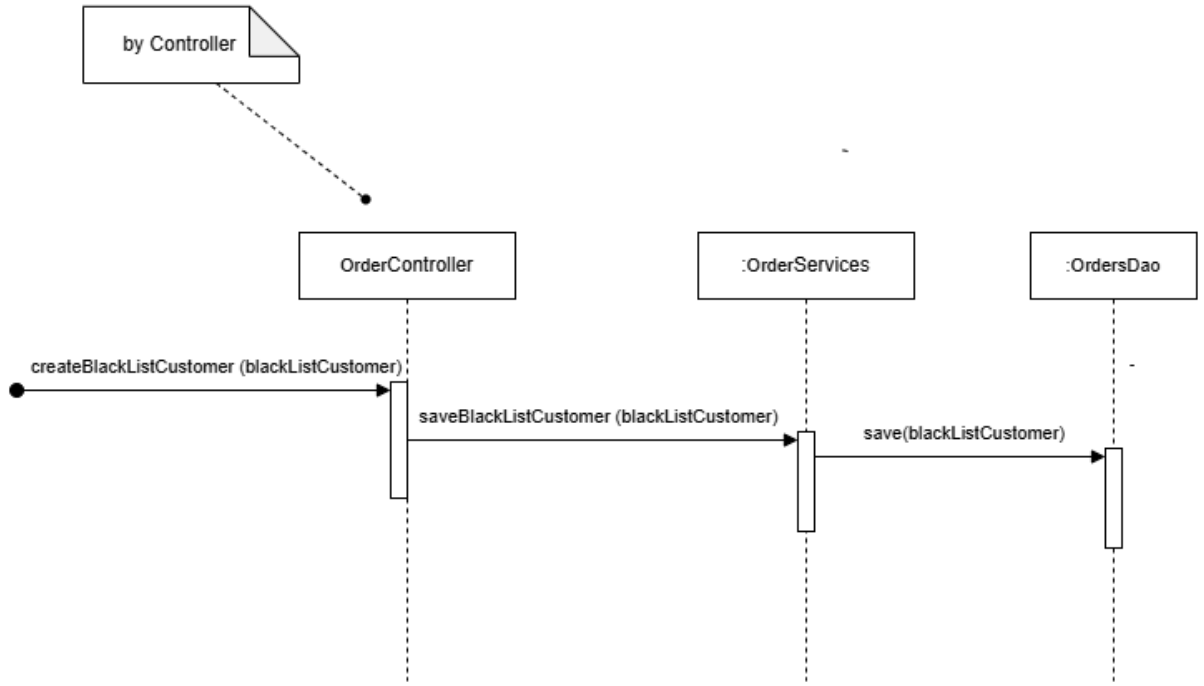


Blacklist Customer

System Sequence Diagram (SSD)

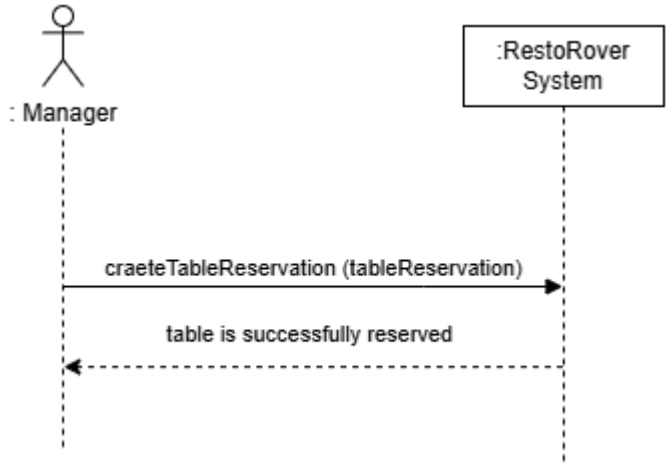


Sequence Diagram (SD) - 01

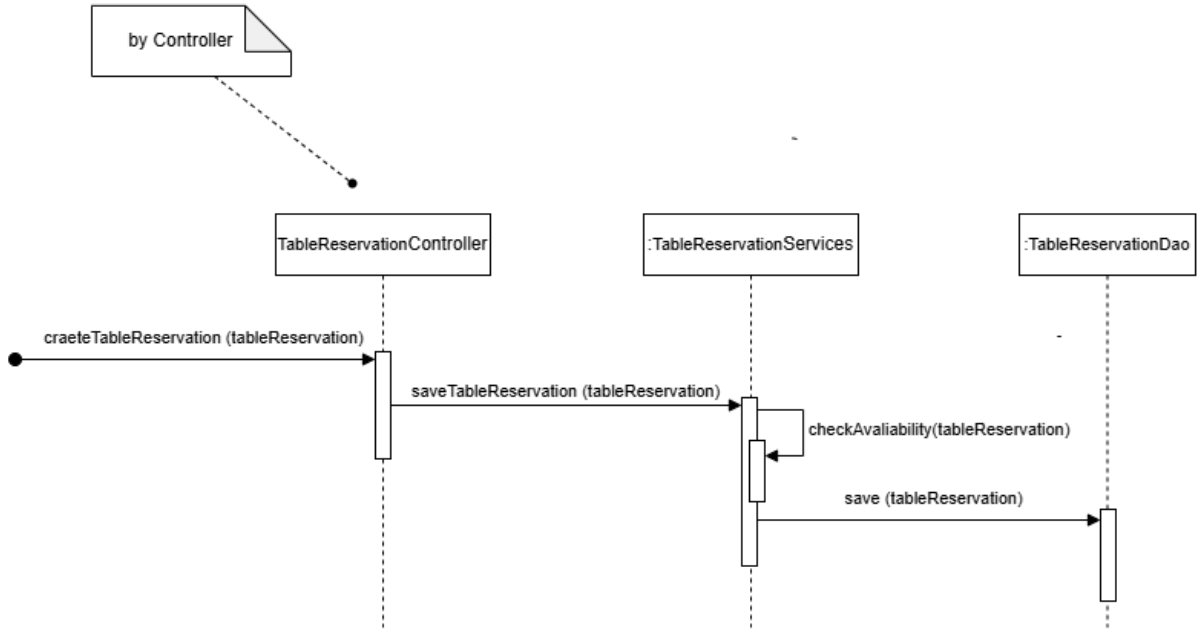


Manage Reservation

System Sequence Diagram (SSD)

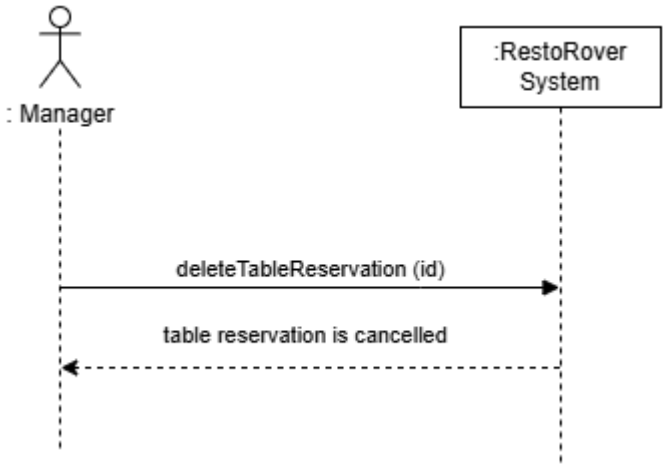


Sequence Diagram (SD) 01

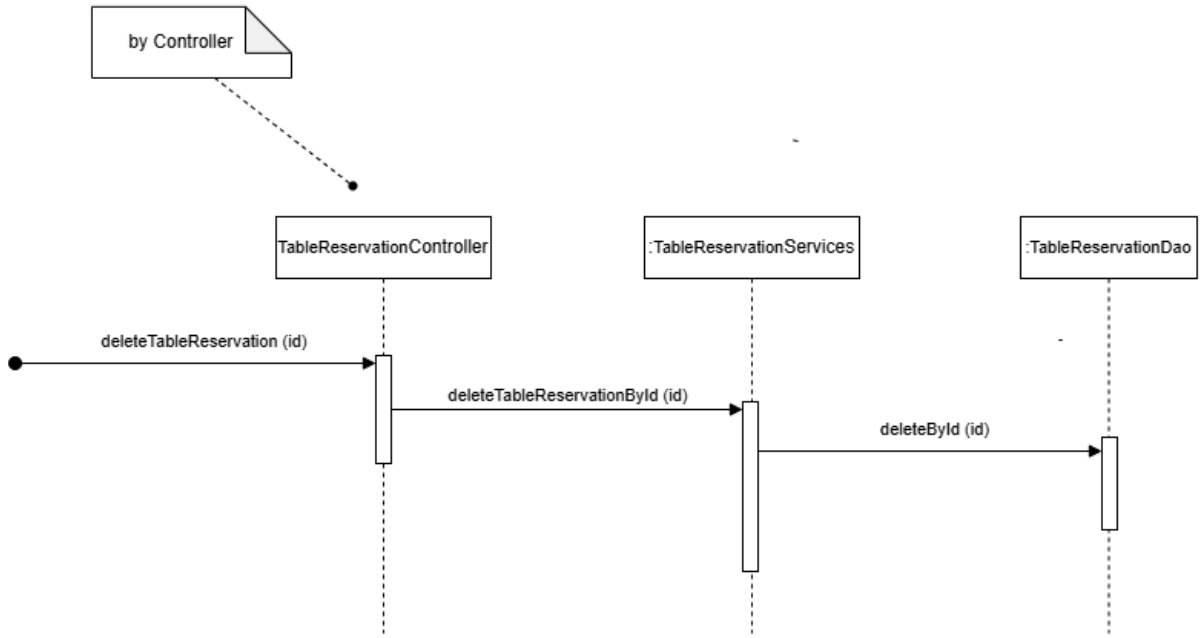


Cancel reservation - Manage Reservation

System Sequence Diagram (SSD) 01

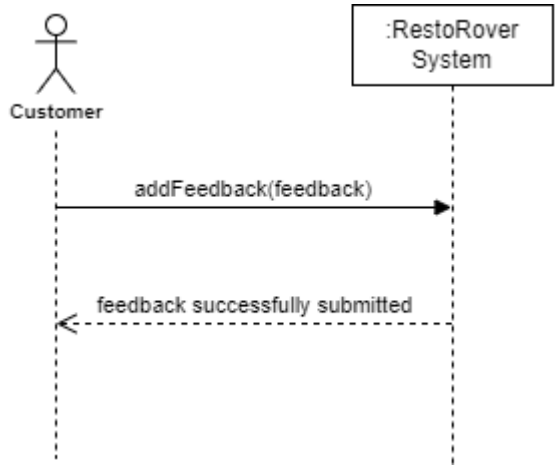


Sequence Diagram (SD) 01

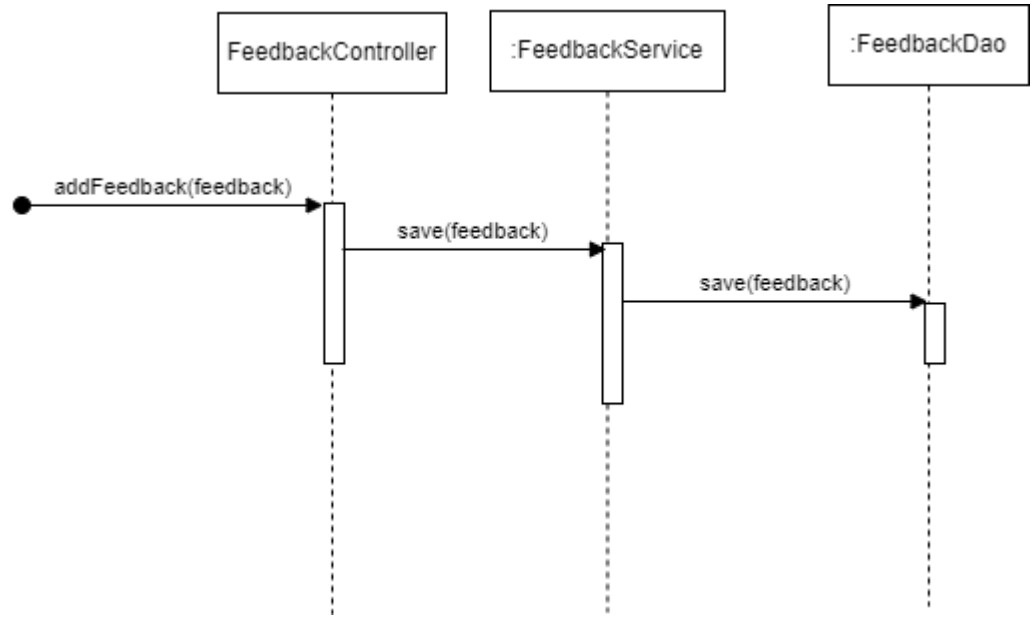


Give Feedback

System Sequence Diagram (SSD)

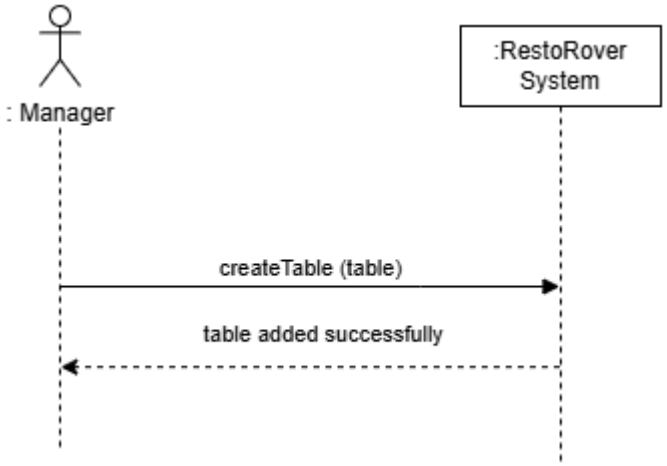


Sequence Diagram (SD) 01

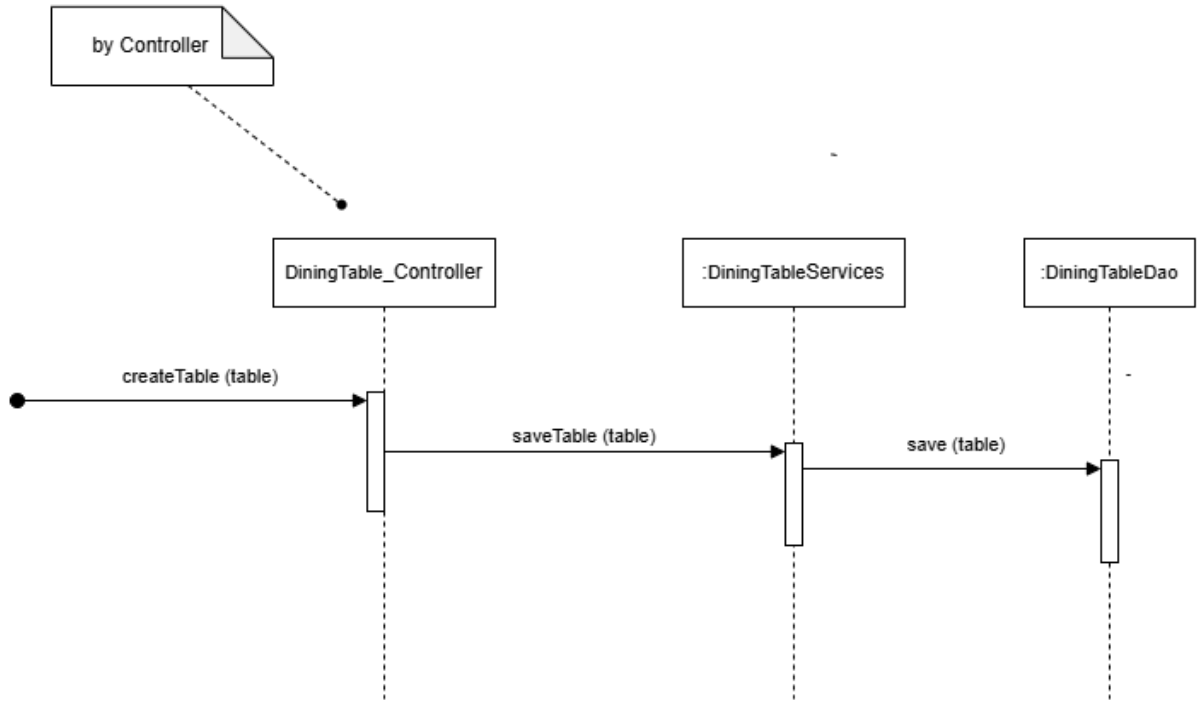


Add New Table - Manage Table

System Sequence Diagram (SSD)

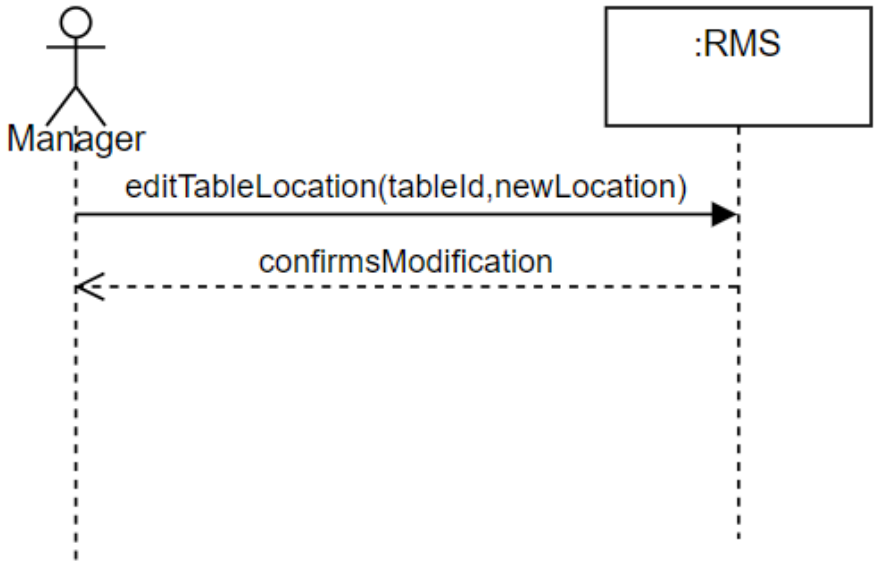


Sequence Diagram (SD) 01

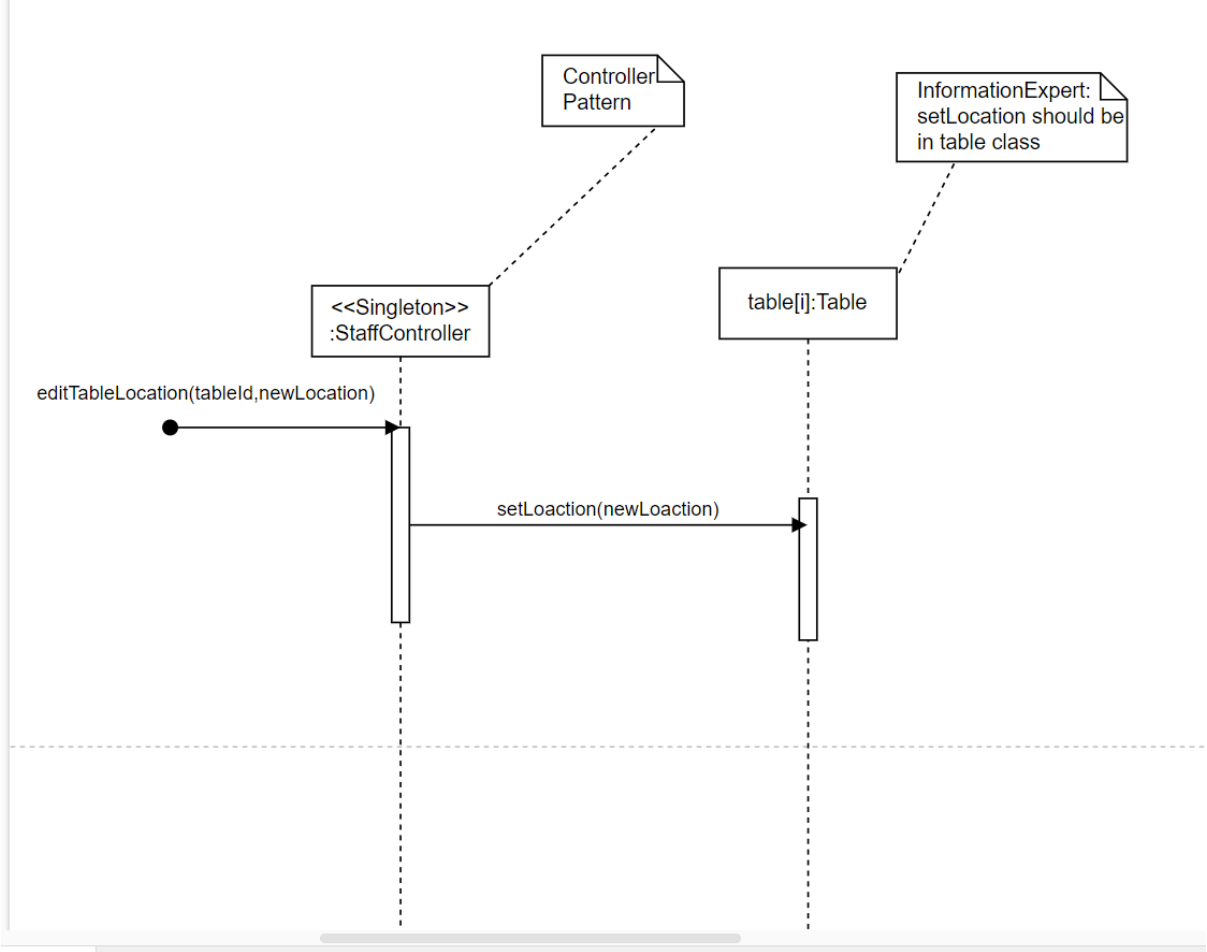


Update Table - Manage Table

System Sequence Diagram (SSD)

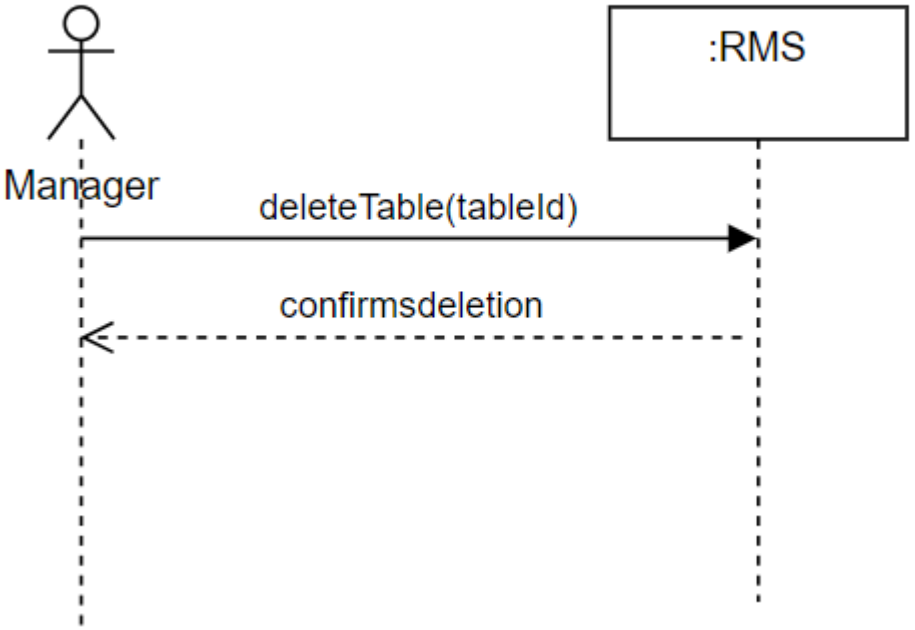


Sequence Diagram (SD) 01

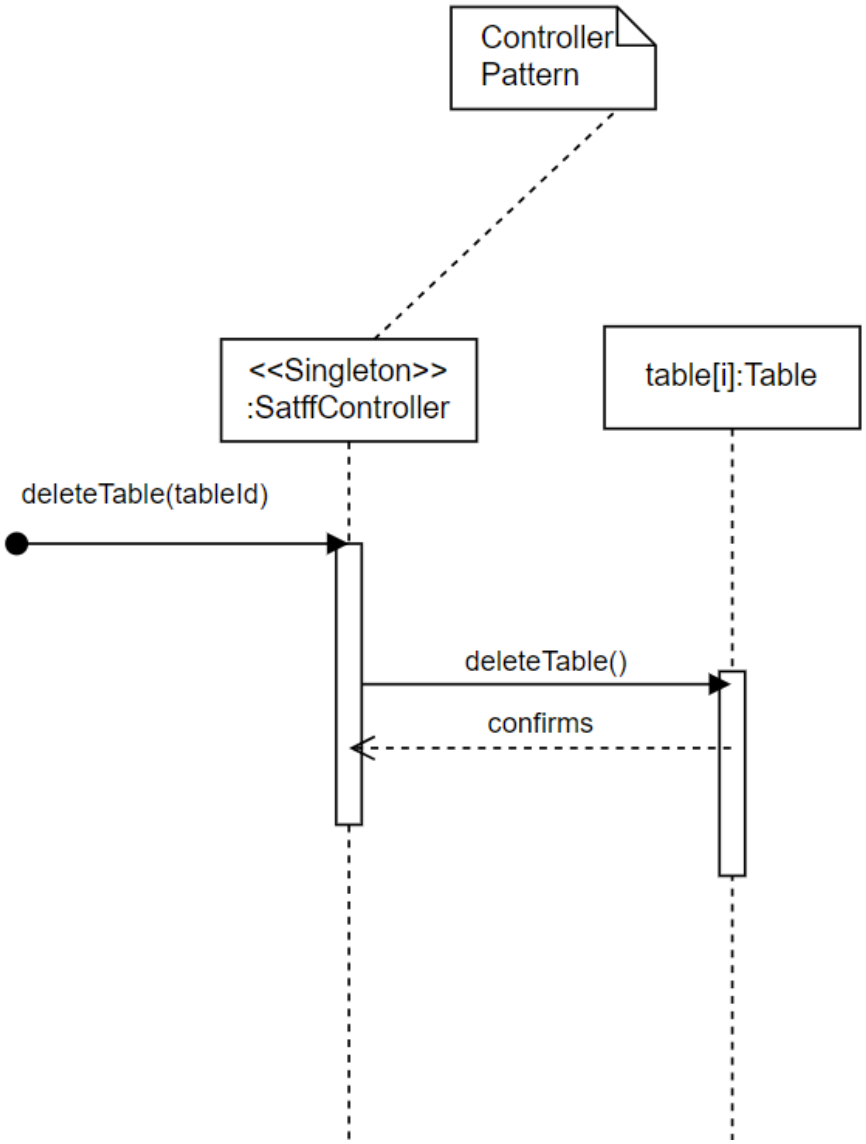


Delete Table - Manage Table

System Sequence Diagram (SSD)

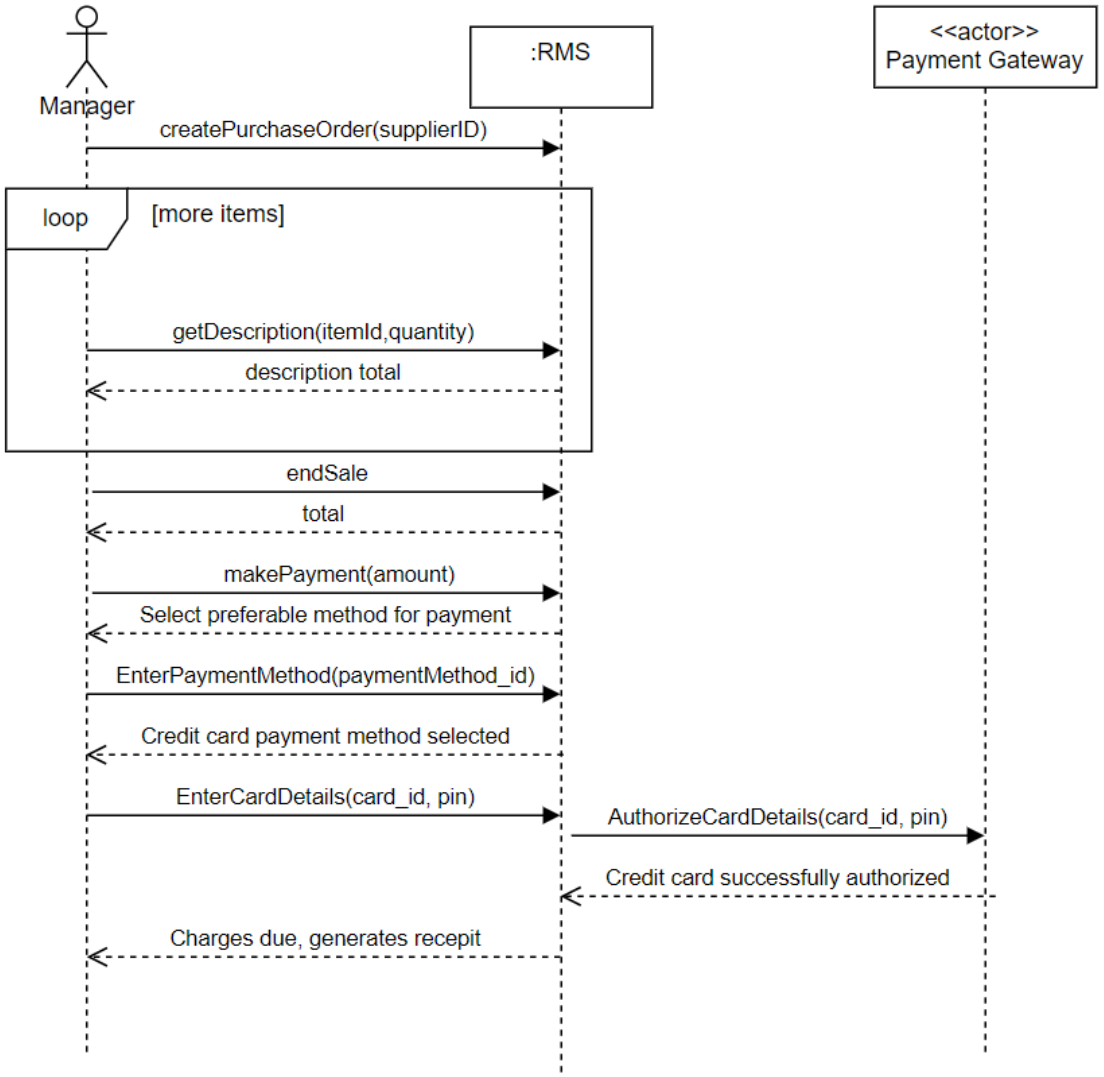


Sequence Diagram (SD)

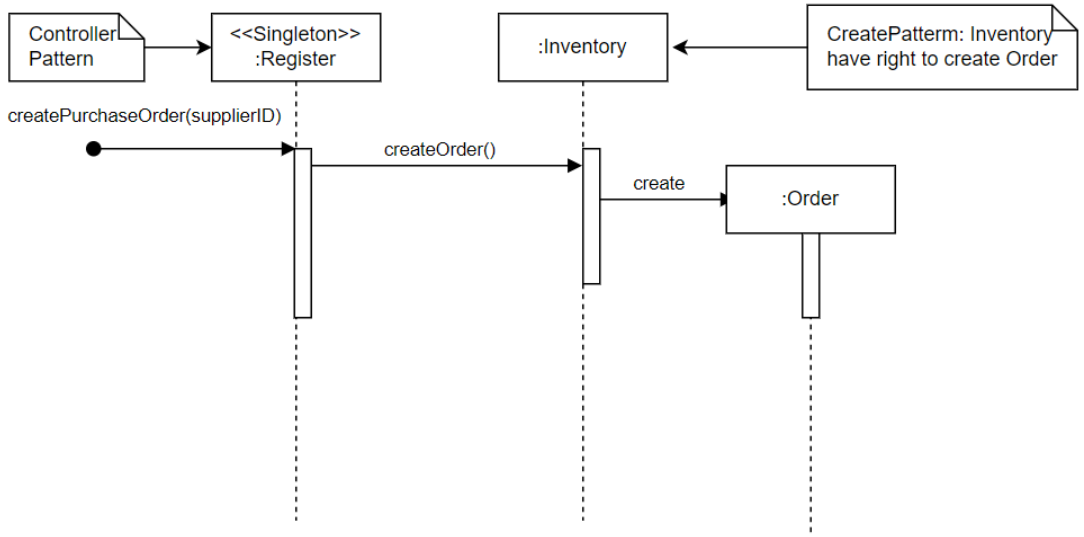


Make Supplier Order

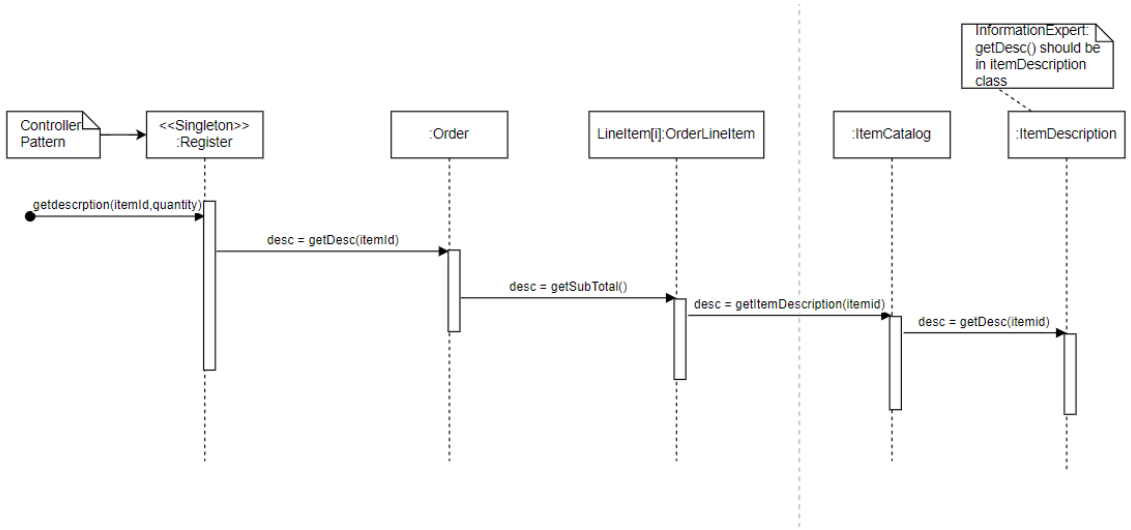
System Sequence Diagram (SSD) 01



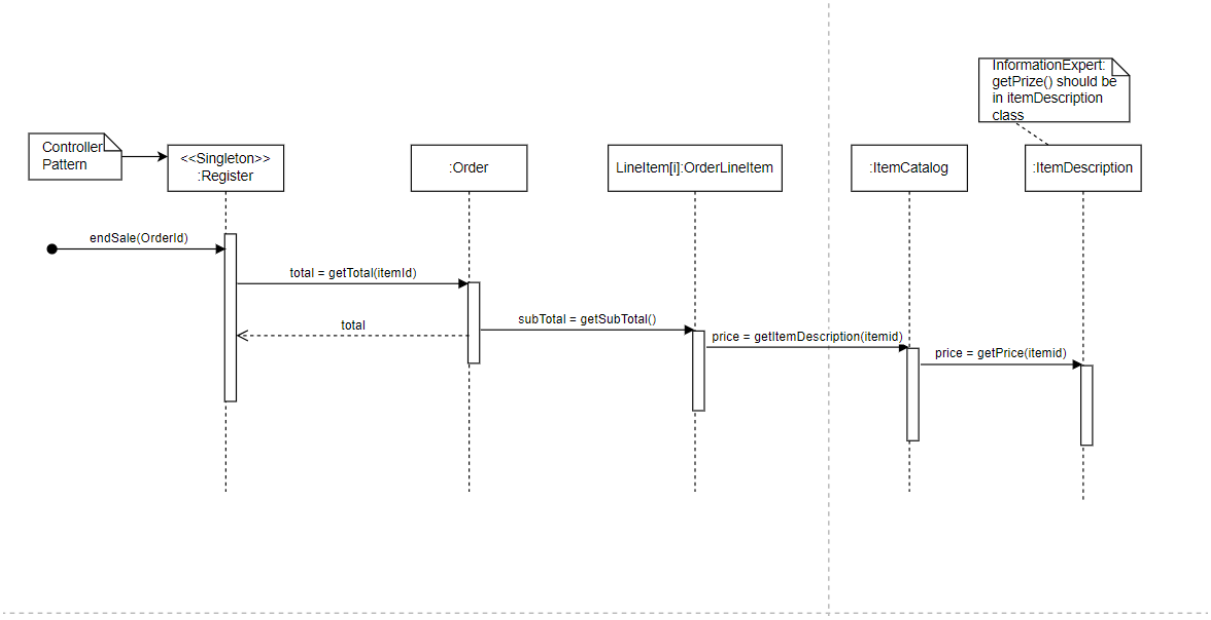
Sequence Diagram (SD) 01



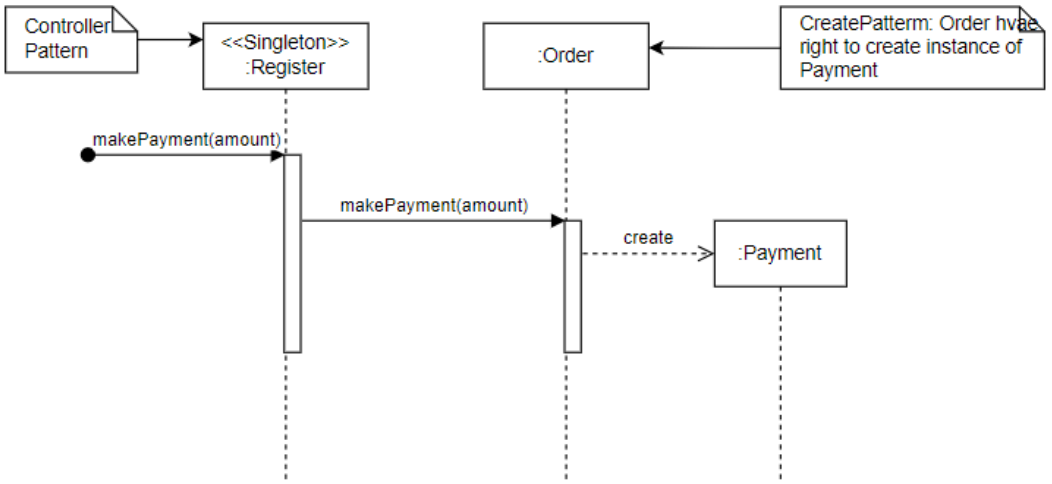
Sequence Diagram (SD) 02



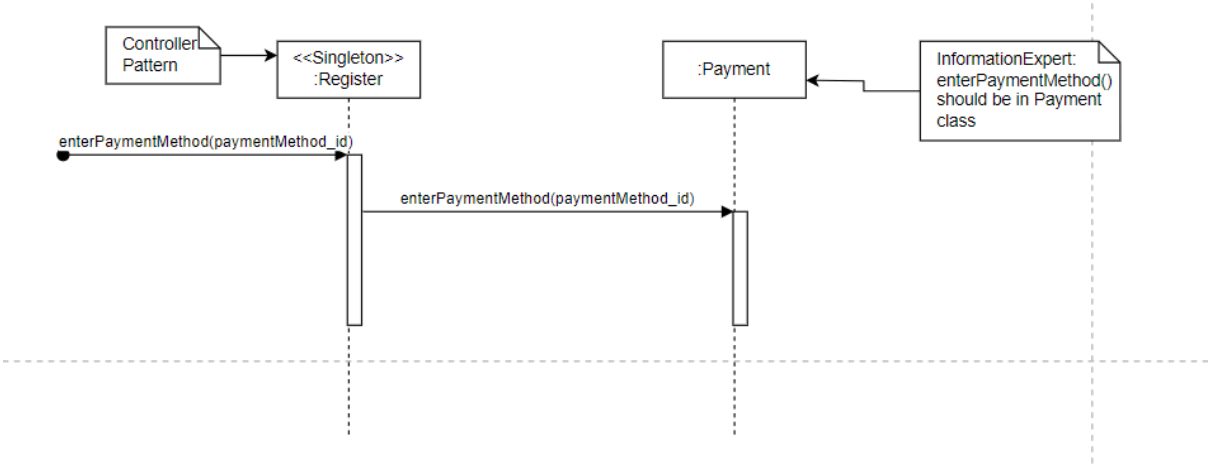
Sequence Diagram (SD) 03



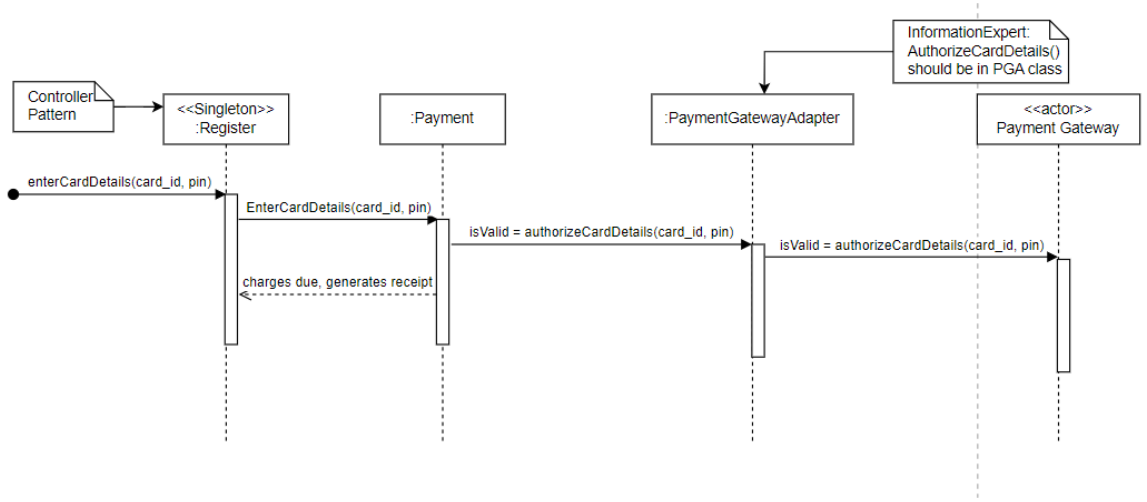
Sequence Diagram (SD) 04



Sequence Diagram (SD) 05

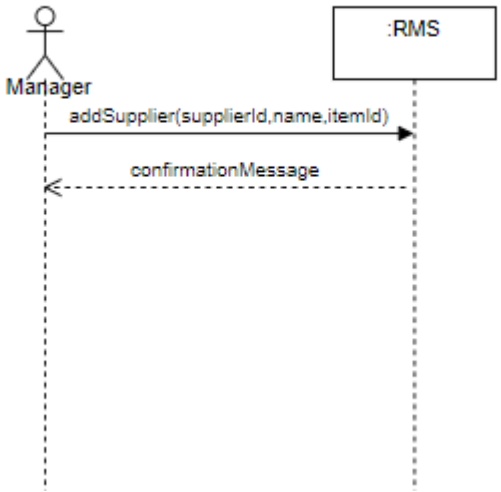


Sequence Diagram (SD) 06

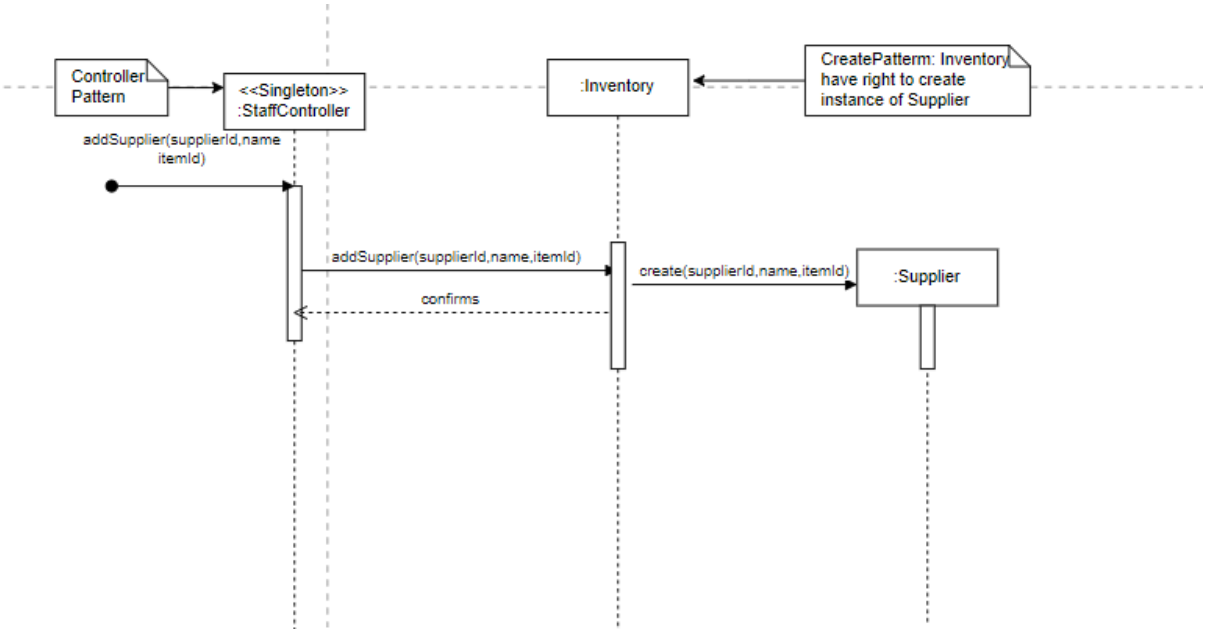


Manage Supplier

System Sequence Diagram (SSD)

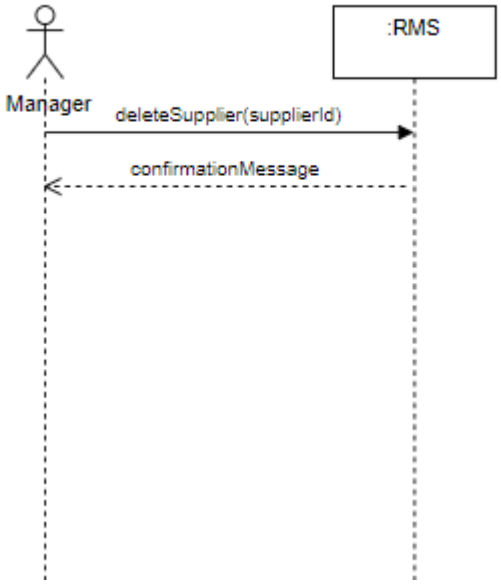


Sequence Diagram (SD) 01

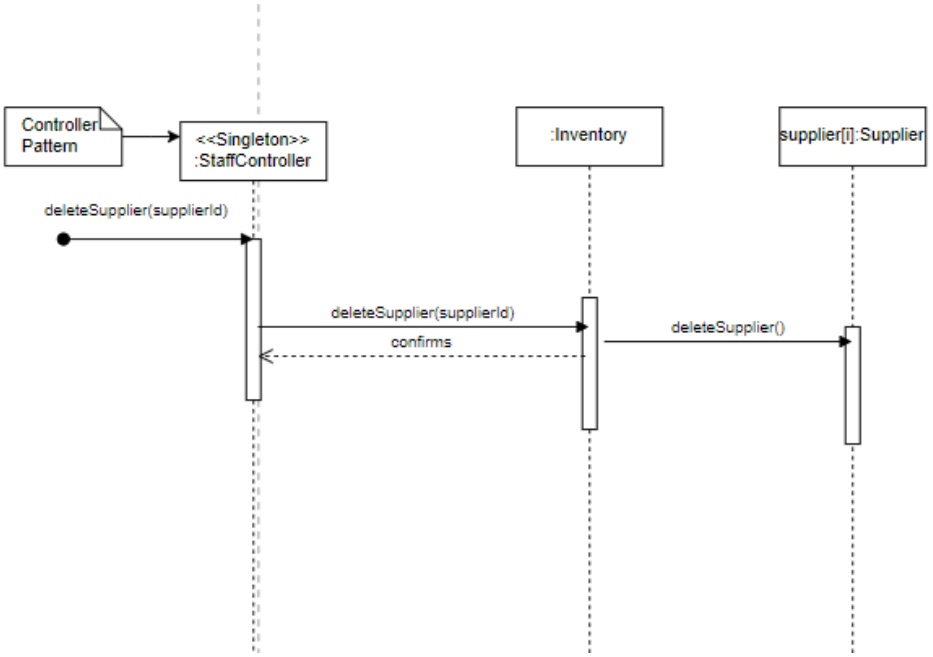


Delete Supplier - Manage Supplier

System Sequence Diagram (SSD) 02

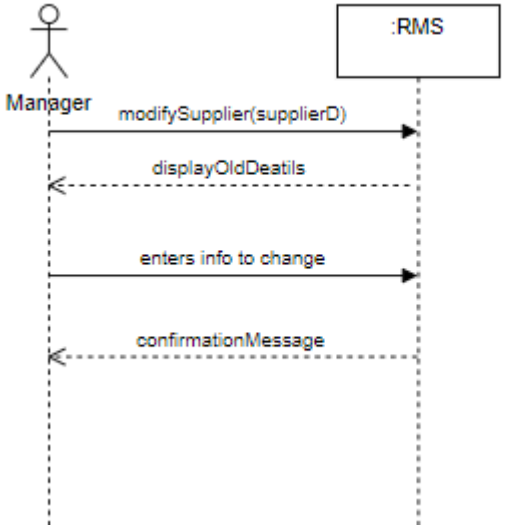


Sequence Diagram (SD) 01



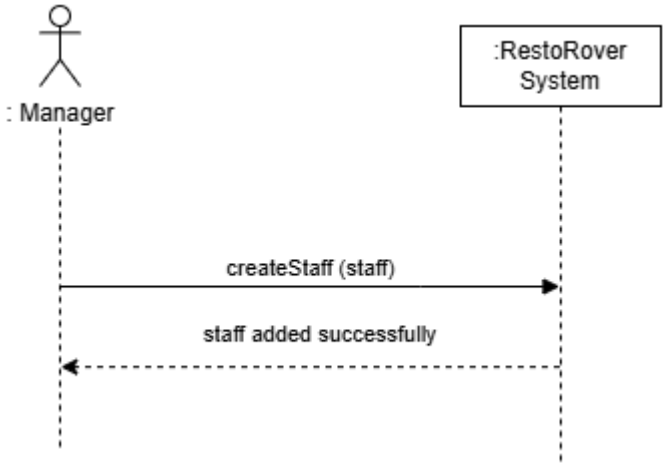
Update Supplier - Manage Supplier

System Sequence Diagram (SSD)

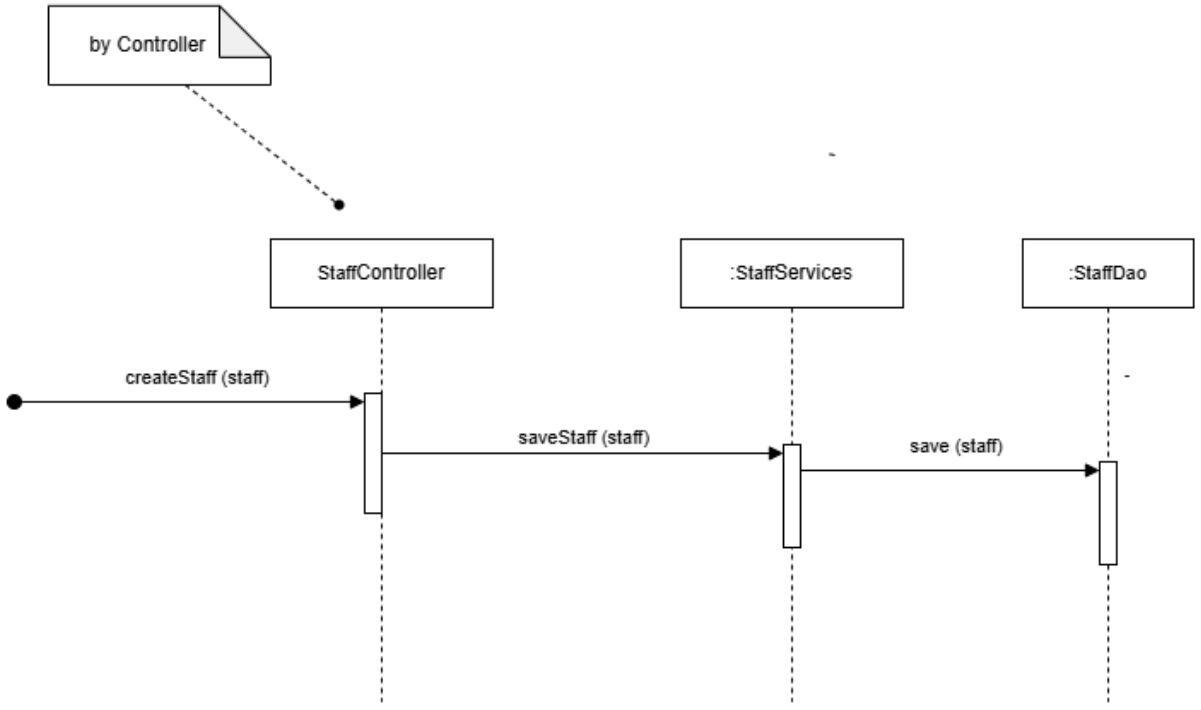


Add Staff - Manage Staff

System Sequence Diagram (SSD) 01

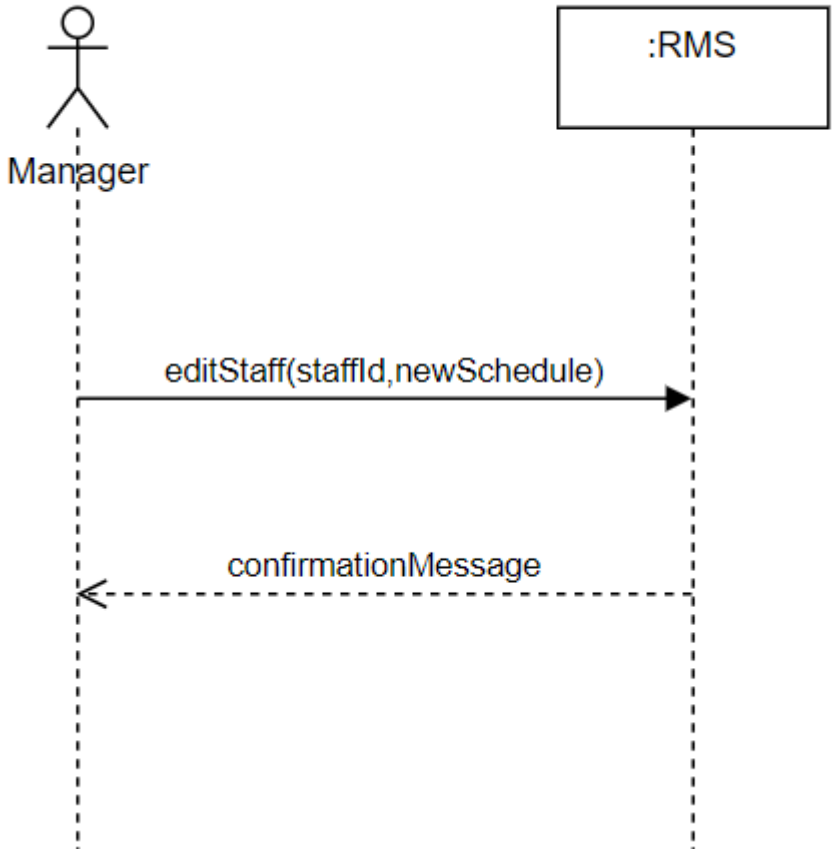


Sequence Diagram (SD) 01

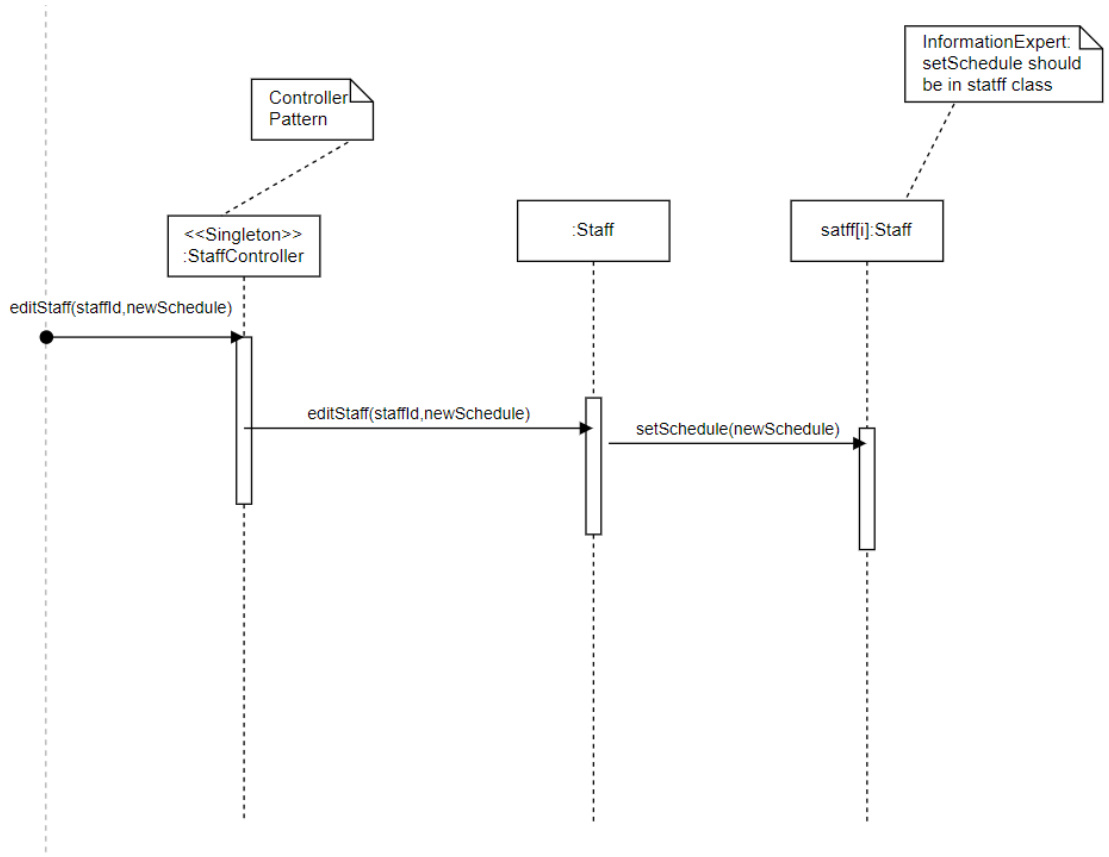


Update Staff - Manage Staff

System Sequence Diagram (SSD) 02

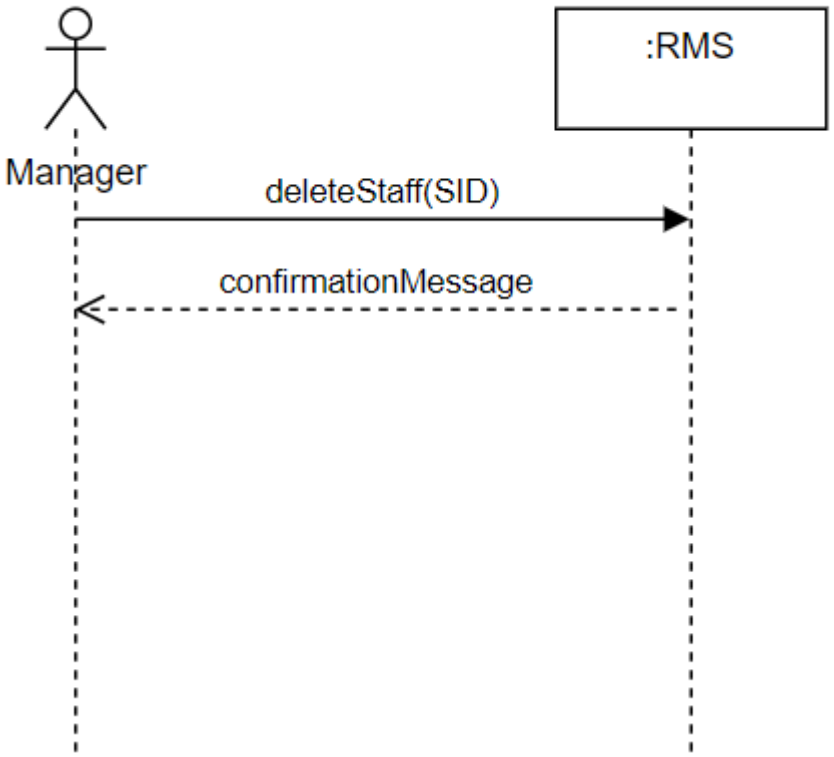


Sequence Diagram (SD) 01

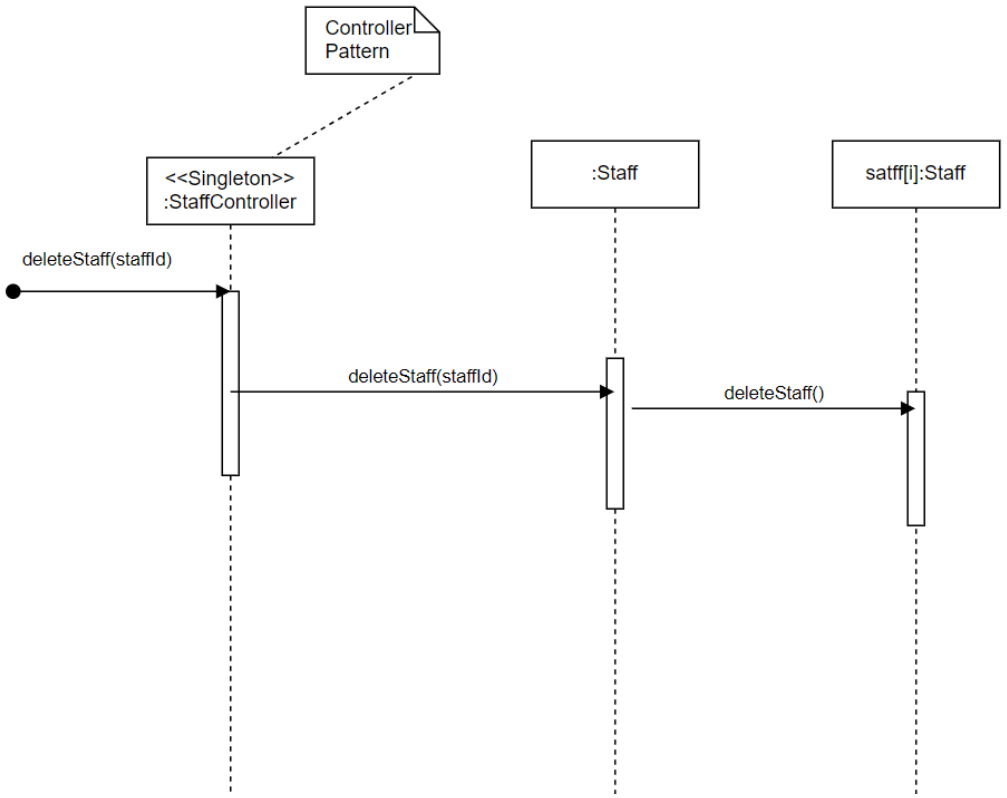


Delete Staff - Manage Staff

System Sequence Diagram (SSD)

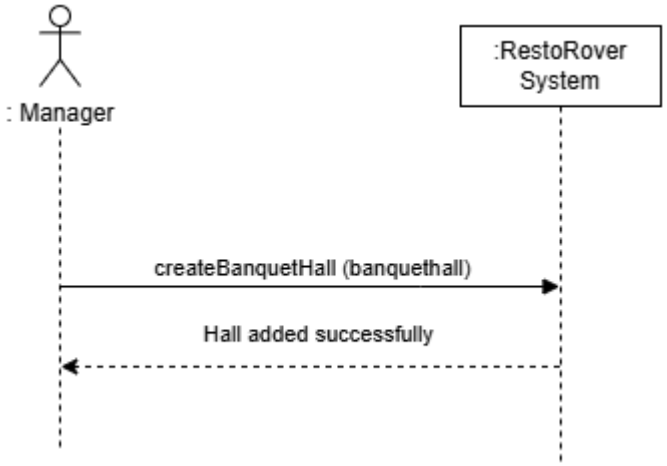


Sequence Diagram (SD) 01

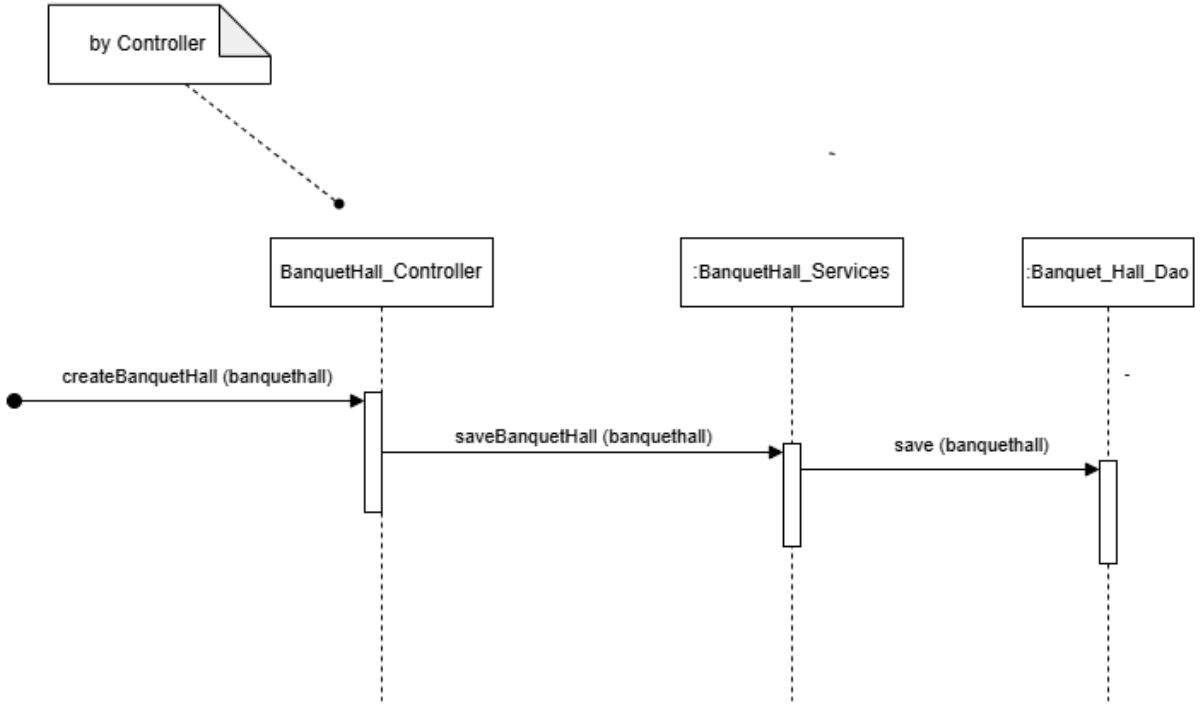


Add BanquetHall - Manage BanquetHall

System Sequence Diagram (SSD)

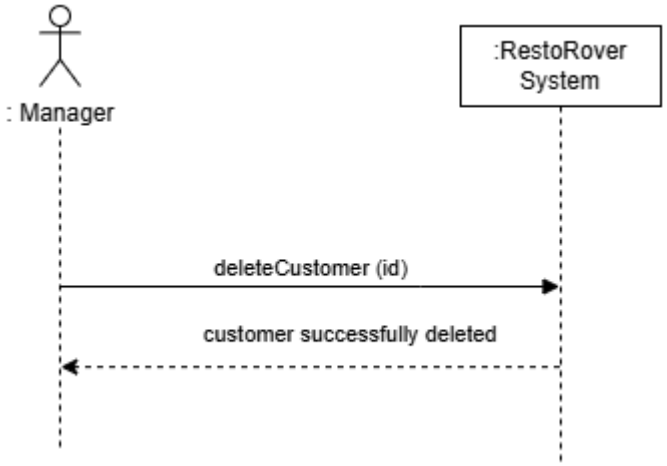


Sequence Diagram (SD)

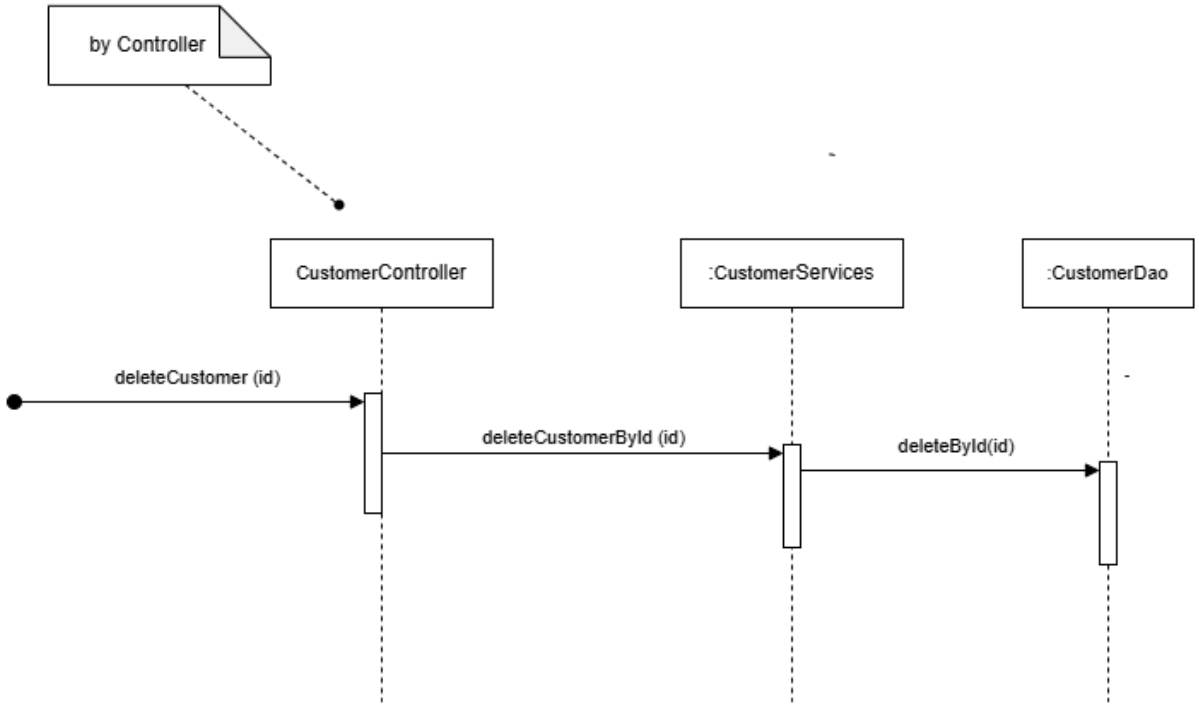


Add Customer- Manage Customer

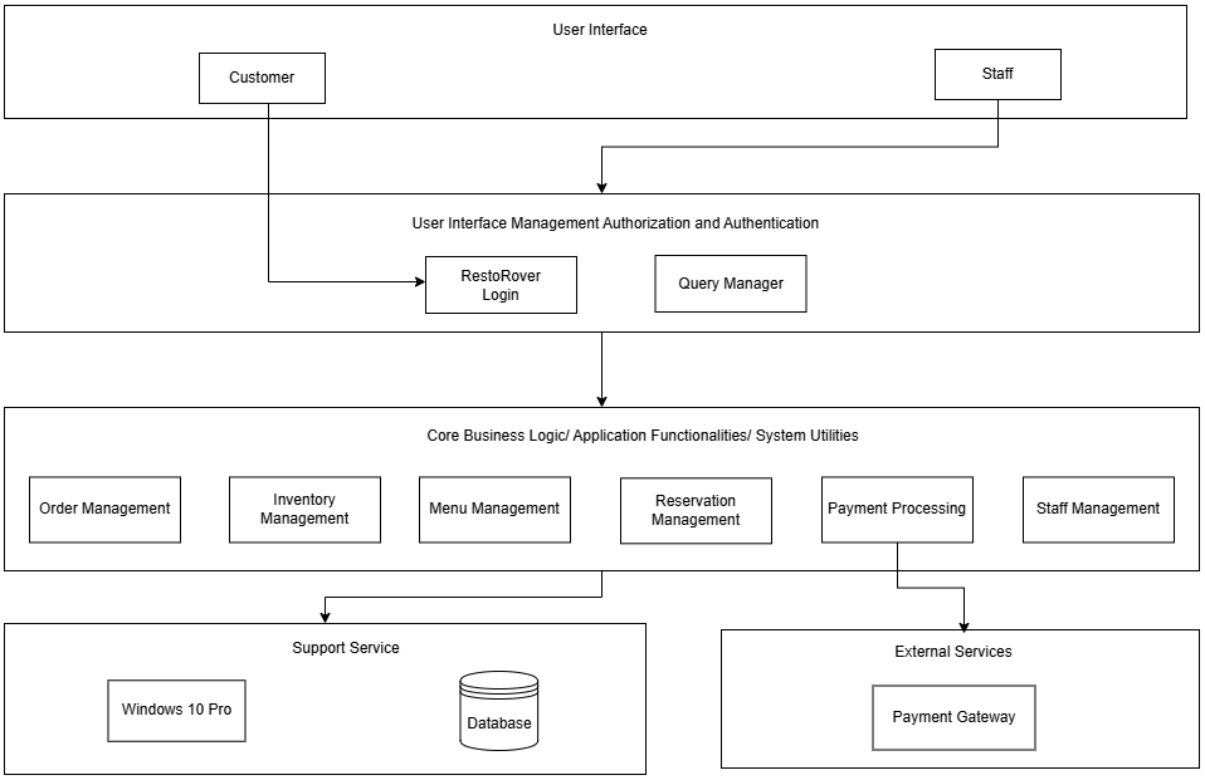
System Sequence Diagram (SSD)



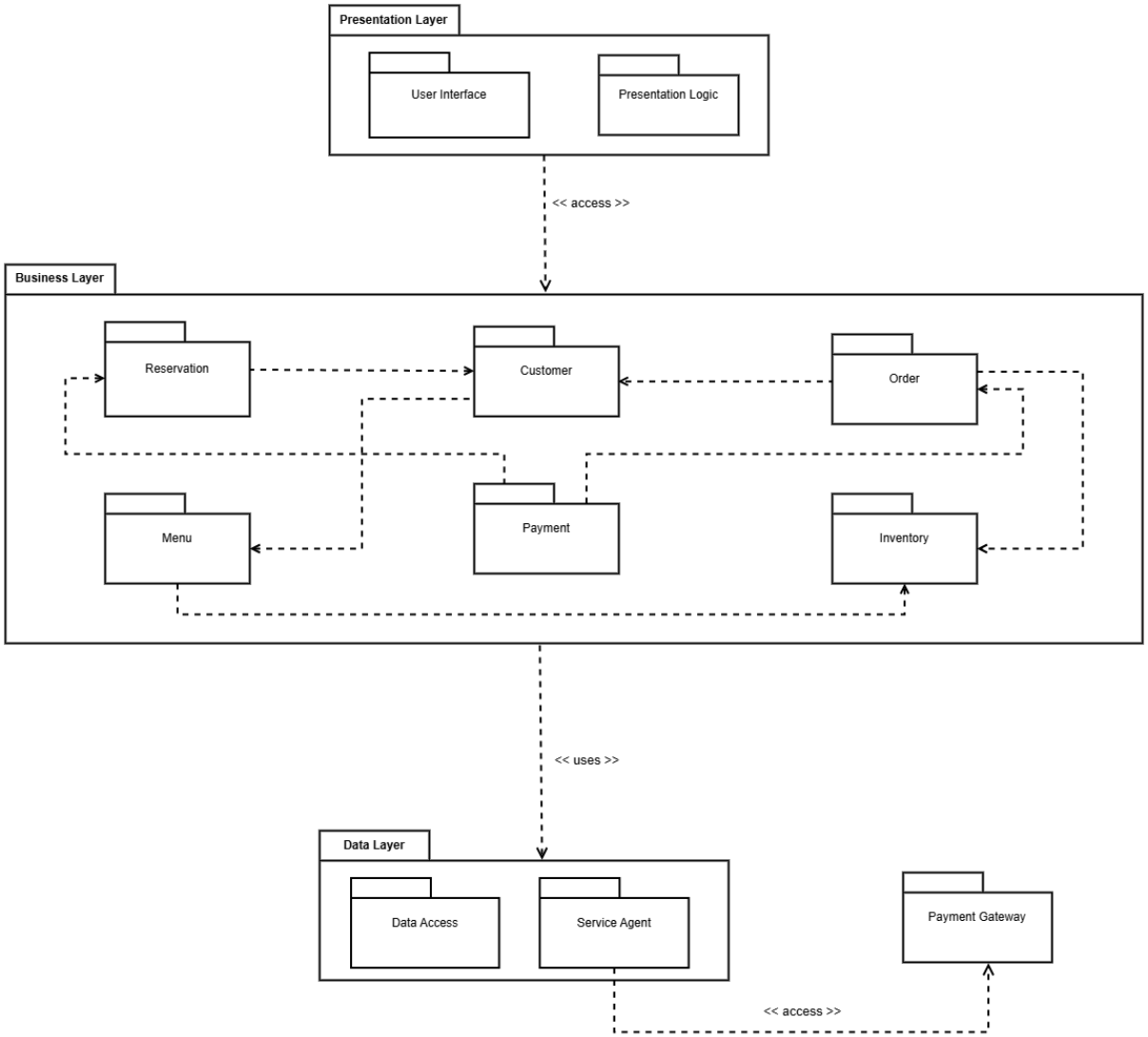
Sequence Diagram (SD)



Block Diagram



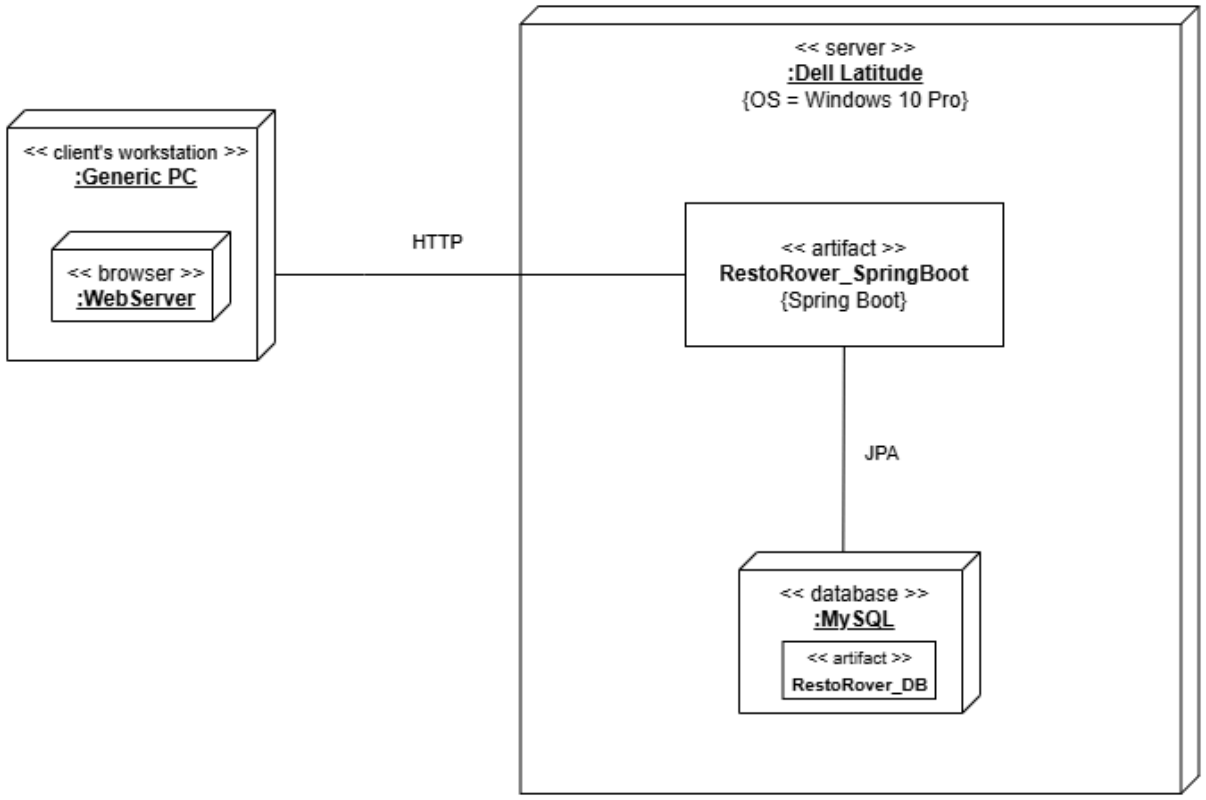
Package Diagram



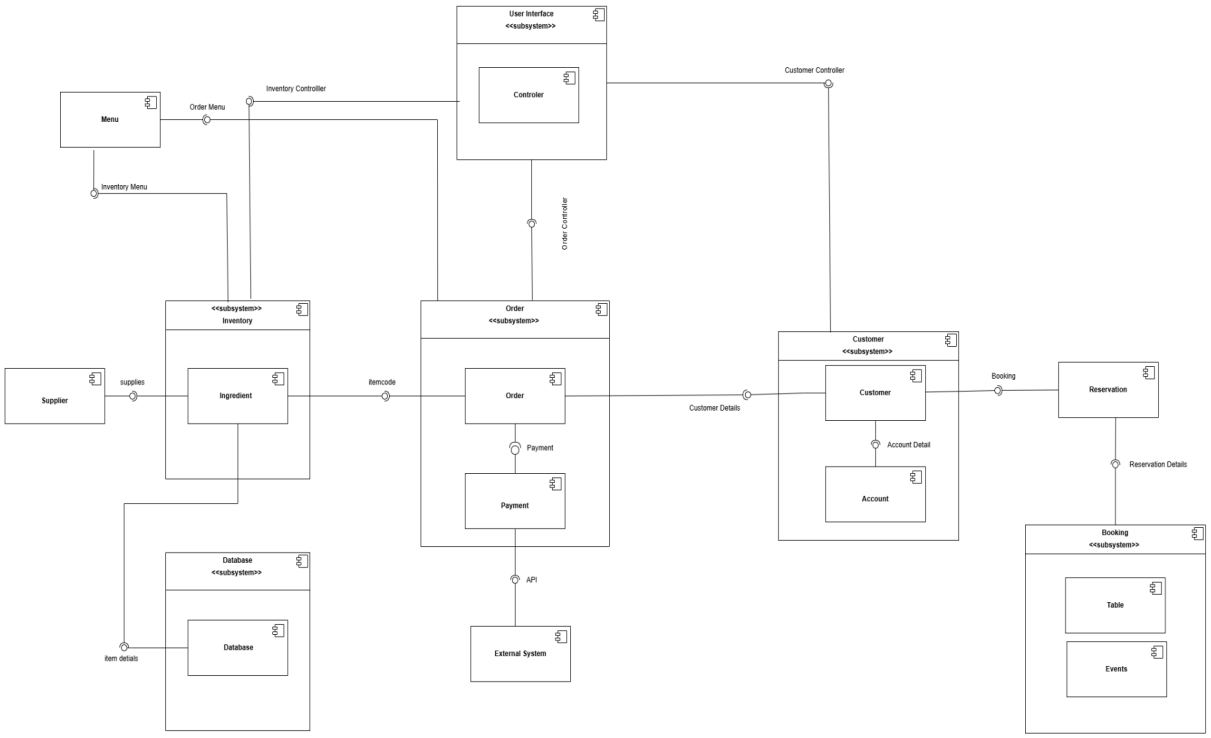
Classes in Packages:

1. Order
    - Order
    - OrderlineItem
    - OrderRegistry
    - Register
2. Menu
    - ItemDescription
    - ItemCatalog
    - Recipe
    - RecipeCatalog
    - Updater
3. Inventory
    - Ingredients
    - IngredientsCatalog
    - Inventory
    - Supplier
    - StaffConrtoller
4. Reservation
    - Table
    - Event
    - BanquetHall
    - Reservation
    - ReservationCatalog
5. Customer
    - Customer
    - Feedback
    - Member
    - MemberRegistry
    - BlackListCustomerRegistry
    - Customer_Controller
6. Payment
    - Payment
    - PaymentGateway

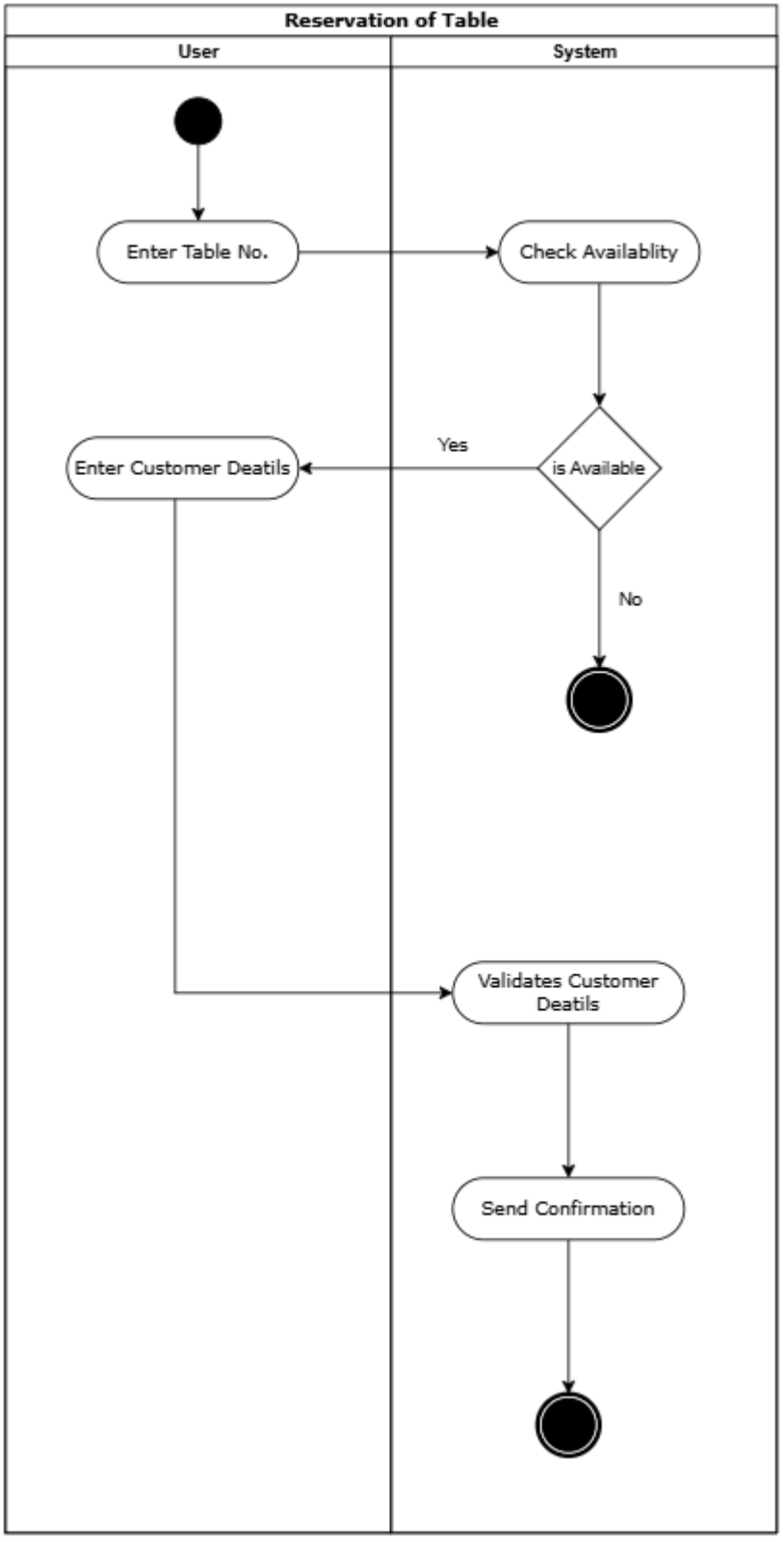
Deployment Diagram

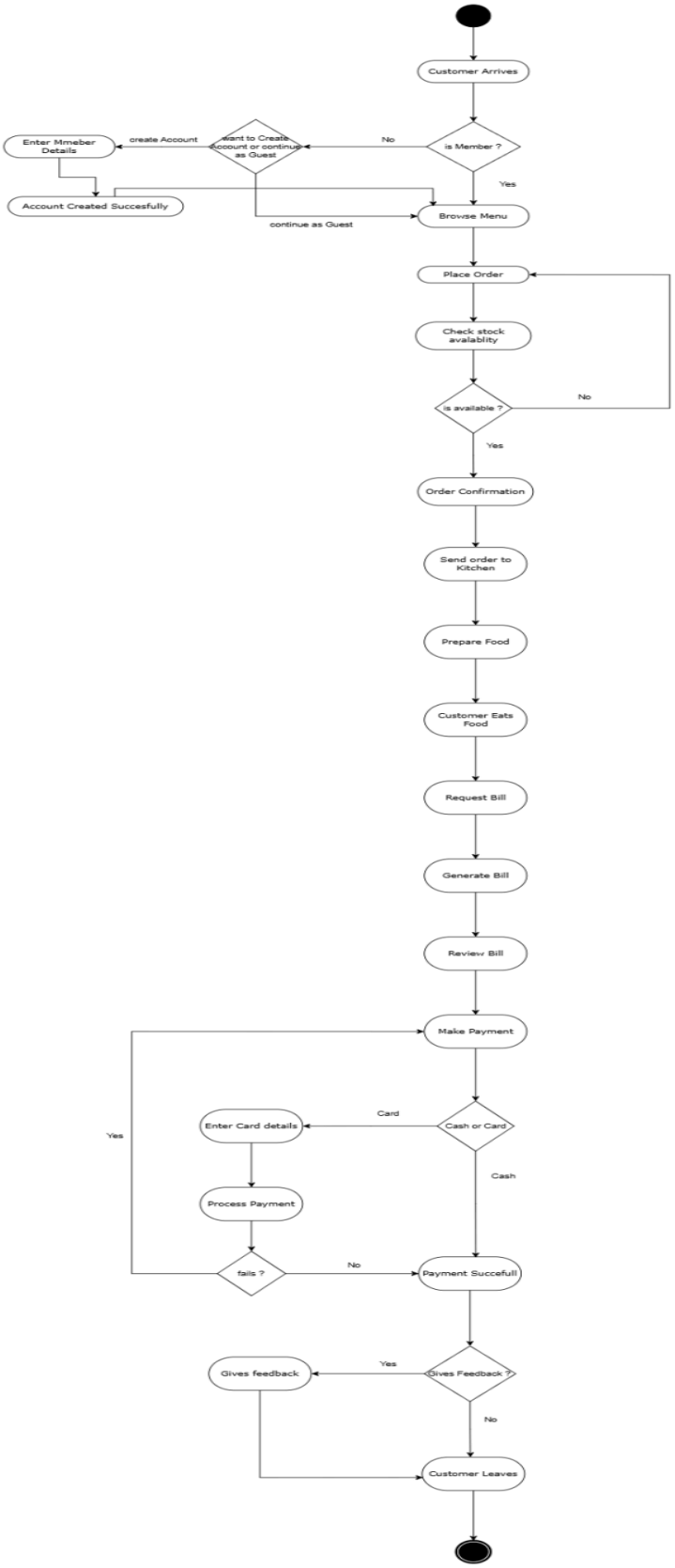


Component Diagram



Activity Diagrams




Customer Arrives
is Member ?
No
want to Create Account or continue as Guest
create Account
Enter Mmeber Details
Account Created Succesfully
continue as Guest
Yes
Browse Menu
Place Order
Check stock avalablity
is available ?
No
Yes
Order Confirmation
Send order to Kitchen
Prepare Food
Customer Eats Food
Request Bill
Generate Bill
Review Bill
Make Payment
Cash or Card
Card
Enter Card details
Process Payment
fails ?
No
Yes
Cash
Payment Succefull
Gives Feedback ?
Yes
Gives feedback
No
Customer Leaves