KTS Solutions

Smart POS (Point of Sales) application
Software Architecture Document

Version 1.0
Prepared by Group 11

Mentor - Dr. Sapumal Ahangama

Group Members

Kobinarth Panchalingam - 200307C

R. A. T. C. Kumara - 200321M

Sanjula Kumarasinghe - 200323V

Index Number	Contribution
200307C	• Introduction • Architectural Representation • Use Case View – (4.1.1) (4.1.4), (4.1.7) • Process View – (6.1.1) (6.1.4), (6.1.7) • Deployment view • Implementation view • Data view • Size and performance
200321M	• Introduction • Use Case View – (4.1.3), (4.1.6) • Process View – (6.1.3), (6.1.6) • Logical view • Data view
200323V	• Introduction • Use Case View – (4.1.2), (4.1.5) • Process View – (6.1.2), (6.1.5) • Logical view • Data view • Architectural goals and constraints • Quality

Revision History

Date	Version	Description	Author
28/08/2023	1.0	Initial document	Group 11

Table of Contents

Software Architecture Document

1. Introduction

1.1 Purpose

The purpose of the Software Architecture Document is to provide a comprehensive overview of the system's architecture. It achieves this by utilizing various architectural views to represent different aspects of the system. This document aims to capture and communicate the significant architectural decisions that have been made during the design and development of the system.

With the help of the 5 various viewpoints, the system's architecture is elaborated. (4+1 model), which considers both the system's static and dynamic behavior to aid developers during the development stage. Additionally, it will enable a variety of stakeholders to understand the SAD's strong ideology.

1.2 Scope

This Software Architecture Document pertains to the architectural design and decision-making process for a Smart POS application. It encompasses the various aspects of the system's structure, including sales processing, inventory management, employee handling, customer interaction, and multi-store support. The document directly influences the system's overall design, the allocation of responsibilities among its components, and the establishment of architectural patterns and principles that guide the development process. It serves as a reference for developers, architects, and stakeholders involved in shaping the system's architecture and ensures alignment with the project's objectives and requirements.

1.3 Definitions, Acronyms, and Abbreviations

- POS - Point of Sale
- API - Application Programming interface.
- MVC - Model-View-Controller
- SSL/TLS - Secure Sockets Layer/Transport Layer Security
- HTTP – Hyper Text Transfer Protocol
- TCP/IP – Transmission Control Protocol/Internet Protocol

1.4 References

[1] *Intelligent diagramming* (no date) *Lucidchart*. Available at: https://www.lucidchart.com/pages/ (Accessed: 28 August 2023).

1.5 Overview

The document is organized into key sections, including Architectural Representation, Architectural Goals and Constraints, Use-Case View, Logical View, Process View, Deployment View, Implementation View, Data View, Size and Performance, and Quality. Each section addresses specific facets of the system's architecture, encompassing its design rationale, components, interactions, and deployment. This comprehensive organization aims to present a clear and coherent depiction of the architectural decisions, guiding principles, and design considerations that collectively shape the supermarket management system.

2. Architectural Representation

This section expounds on the software architecture of the current system and its representation through different architectural views. The following views are considered essential for understanding the system's architecture:

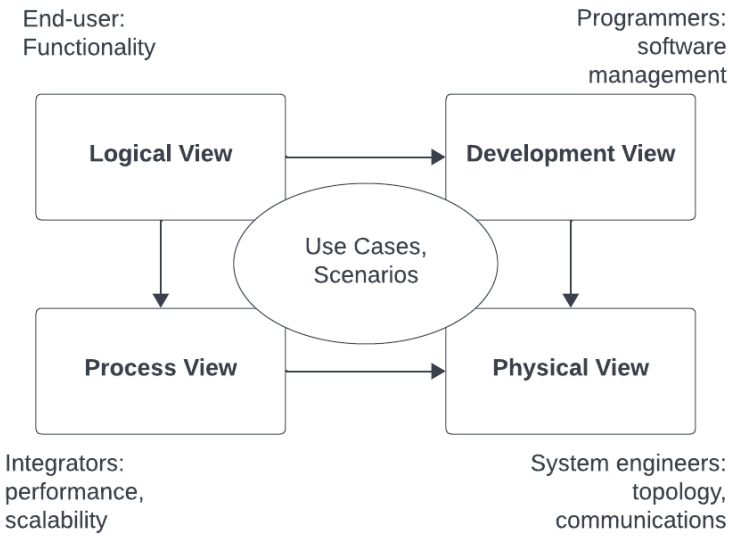

Figure 2-1 4+1 view model

Use-Case View: This view focuses on pivotal system use cases, which are integral to the system's functionality. It outlines scenarios where the system is interacted with by various actors, shedding light on how the system responds to their actions.

Logical View: The Logical View delves into the high-level organization of components within the system. It shows how various components are structured, how they interact, and how they collectively contribute to fulfilling the system's requirements.

Process View: The Process View encapsulates the dynamic behavior of the system. It emphasizes the interactions between components during runtime, showcasing the flow of data and control as well as the response to user inputs.

Deployment View: This view shows the physical arrangement of software components across hardware resources. It details the distribution of software components to different nodes, highlighting their deployment, connectivity, and communication patterns.

Implementation View: The Implementation View provides insights into the system's internal organization. It delves into the software modules, their relationships, and how they contribute to realizing the system's functionalities.

By dissecting the Smart POS application architecture into these distinct perspectives, this section facilitates a comprehensive understanding of the system's structure, behavior, and deployment, ensuring a holistic view of its architectural landscape.

3. Architectural Goals and Constraints

The Smart POS application aims to achieve several architectural goals that drive the design and development process.

3.1 Security

One of the primary goals is to ensure robust security throughout the system. Given the sensitivity of payment transactions and customer data, the architecture will implement strong encryption mechanisms, user authentication, and role-based access control to safeguard against unauthorized access.

3.2 Scalability

Ensuring the system can handle varying transaction loads without compromising performance. The architecture will incorporate scalable components and employ modern cloud technologies to dynamically allocate resources based on demand.

3.3 Persistence

The architecture aims to achieve efficient and reliable data persistence. Data integrity and consistency are paramount, with ACID-compliant transactions ensuring that critical data operations are performed accurately. The architecture must facilitate seamless integration with the chosen database management system while minimizing database bottlenecks.

Caching mechanisms will be implemented to enhance data retrieval speed, particularly for frequently accessed data. However, data caching should be balanced to prevent outdated information from being presented to users.

3.4 Performance

The architecture emphasizes high performance by minimizing latency in data retrieval and processing. Load balancing mechanisms prevent bottlenecks and resource strain. To maintain optimal performance, the architecture prioritizes resource-efficient design and proactive monitoring.

3.5 Flexibility and Modularity

The architecture will emphasize a modular and flexible design that enables easy integration of new features and enhancements. Well-defined interfaces and modular components will support incremental updates and extensions without disrupting the overall system.

3.6 Integration

The architecture will facilitate seamless integration with external systems, such as payment gateways and inventory management tools. Well-defined APIs and communication protocols will ensure smooth data exchange and interoperability.

3.7 Portability

The application should be designed to be portable across different operating systems and environments. Avoiding platform-specific dependencies will ensure the application can be deployed on various platforms without major modifications.

3.8 Maintainability

The architecture fosters maintainability through standardized coding practices and modular components. It promotes separation of concerns and avoids unnecessary complexity. Detailed documentation and automated testing facilitate future updates without introducing regressions.

4. Use-Case View

4.1 Use-Case Realizations

4.1.1 Sales Processing and Transaction Management

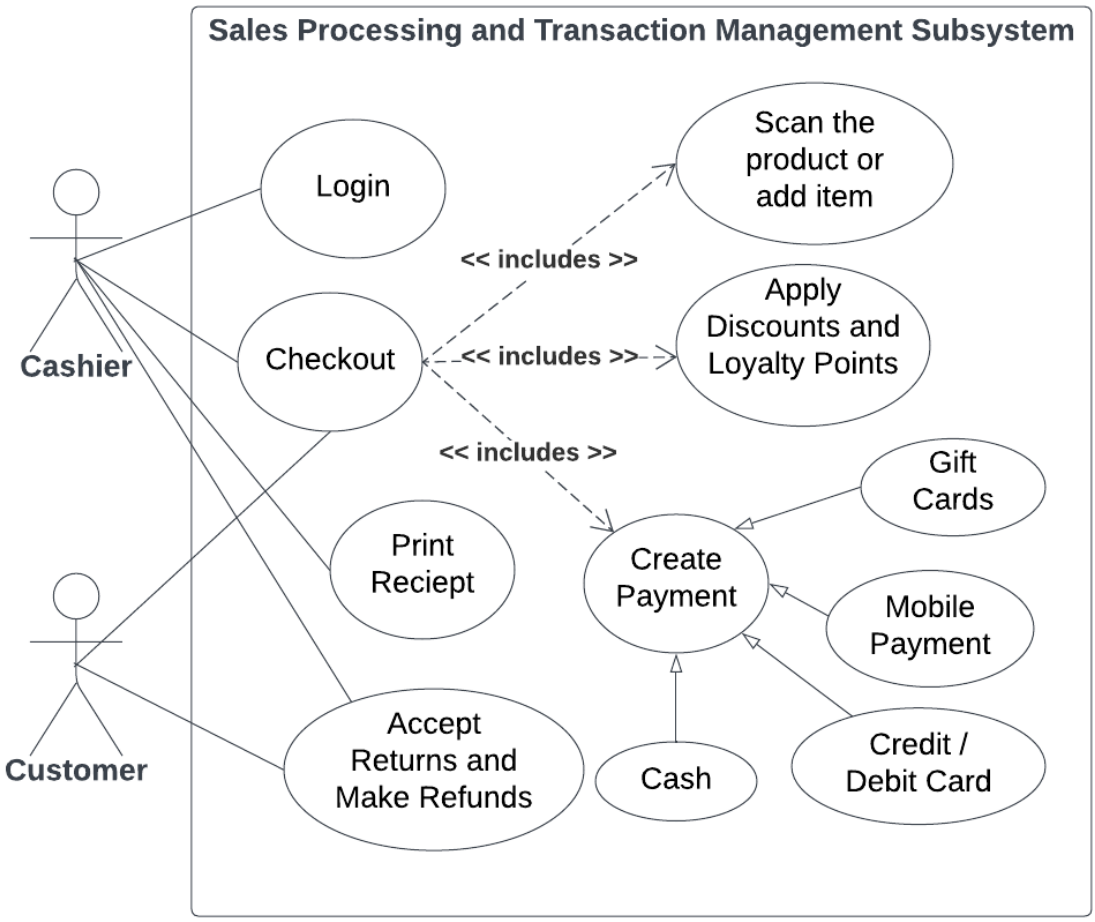

Figure 4-1

4.1.1.1 User Login

Use case name	User Login
Actor	Cashier
Description	This use case represents the process of a cashier logging into the system to begin processing sales transactions
Preconditions	The cashier should have a valid username and password.
Main flow	1. The cashier launches the application. 2. The system presents a login screen.

	3. The cashier enters their username and password. 4. The cashier submits the login credentials. 5. The system verifies the credentials. 6. If the credentials are valid, the system grants access to the cashier. 7. The system displays the main dashboard.
Successful end/ post condition	The cashier successfully logs into the system and gains access to the main dashboard.
Fail end/ post condition	If the entered credentials are invalid, the system displays an error message, and the cashier is not granted access.
Extensions	N/A

4.1.1.2 Checkout

Use case name	Checkout
Actor	Cashier, Store Customer
Description	This use case represents the process of checking out items for a customer, including scanning products, applying discounts and loyalty programs, and processing payment.
Preconditions	The cashier is logged into the system, and the items to be purchased are ready for checkout.
Main flow	1. The cashier selects the "Checkout" option from the main dashboard. 2. The cashier begins scanning each product's barcode using the barcode scanner. 3. For products without barcodes, the cashier can search for and manually add the product using the system's product database. 4. After scanning/adding each product, the system displays the product name, quantity, and price. 5. Once all products are added, the system calculates the subtotal. 6. The system prompts the cashier to apply any applicable discounts or loyalty programs. 7. The cashier selects the relevant discounts or loyalty programs for the transaction. 8. The system recalculates the total amount based on applied discounts.

	9. The system displays the final amount to be paid. 10. The cashier asks the mode of payment. 11. The cashier selects the payment method (e.g., cash, card, mobile wallet) and the system handles the payment. 12. The system calculates the change due and updates the transaction status. 13. The system generates a receipt with transaction details and items purchased. 14. The cashier provides the receipt to the customer. 15. The system updates the inventory by reducing the quantities of purchased items.
Successful end/ post condition	The transaction is successfully completed, the customer receives a receipt, and the inventory is updated accordingly.
Fail end/post condition	If there are any issues with payment processing or if the inventory update fails, the system displays an error message, and the transaction is not completed.
Extensions	None

4.1.1.3 Create Payment

Use case name	Create Payment
Actor	Cashier, Store Customer
Description	This use case represents the process of creating a payment for a transaction using various payment methods, including cash, credit/debit card, online payment, and gift cards.
Preconditions	The cashier has completed the checkout process and selected the items for purchase.
Main flow	1. After selecting the items for purchase, the cashier selects the "Create Payment" option. 2. The system presents a list of available payment methods: cash, credit/debit card, online payment, and gift cards. 3. The cashier selects the preferred payment method for the transaction. **If Payment Method is Cash:** 4. The system displays the total amount due. 5. The cashier enters the amount received from the customer.

	6. The system calculates the change due and updates the payment details. 7. The system generates a receipt with payment details and items purchased. **If Payment Method is Credit/Debit Card or Online Payment:** 1. The system prompts the cashier to enter the credit/debit card information or initiate the online payment process. 2. The cashier enters the card details or initiates the online payment. 3. The system processes the payment and updates the payment details. 4. The system generates a receipt with payment details and items purchased. **If Payment Method is Gift Card:** 1. The system prompts the cashier to scan or enter the gift card code. 2. The cashier scans the gift card barcode or enters the code manually. 3. The system verifies the gift card details and available balance. 4. If the gift card balance covers the total amount due, the system updates the payment details and deducts the amount from the gift card balance. 5. If the gift card balance is insufficient, the system prompts the cashier to enter an alternative payment method for the remaining amount. 6. The system generates a receipt with payment details and items purchased.
Successful end/ post condition	The payment is successfully created using the selected payment method, and the system generates a receipt with payment details and items purchased.
Fail end/ post condition	If there are issues with processing the payment, such as invalid card information, insufficient gift card balance, or online payment failure, the system displays an error message, and the payment creation is not completed.
Extensions	None

4.1.1.4 Accept Returns and Refund Customers

Use case name	Accept Returns and Refund Customers
Actor	Cashier, Store Customer
Description	This use case represents the system's process of accepting returns from customers and generating refund transactions in cash. The system interacts with the cashier and the transaction data.
Preconditions	The cashier initiates the return process in the system, and the return items are verified against the original transaction.
Main flow	1. The cashier selects the "Accept Returns and Refund" option in the system. 2. The system prompts the cashier to scan the barcode on the original bill. 3. The cashier scans the barcode to link the return to the original transaction. 4. The system calculates the refund amount based on the items returned and their original prices. 5. The system displays the calculated refund amount to the cashier. 6. The system confirms that the refund will be provided in cash. 7. The system updates the transaction records to reflect the refund and its amount. 8. The system generates a refund transaction with the refunded items, refund amount, and the payment method used for the refund (cash). 9. The system associates the refund transaction with the original transaction.
Successful end/ post condition	The refund transaction is generated and linked to the original transaction, indicating the refund amount and cash payment method.
Fail end/post condition	If the return items do not match the original transaction or there are issues with calculating the refund amount or generating the refund transaction, the system displays an error message, and the refund process is not completed.
Extensions	None

4.1.2 Inventory Management Subsystem

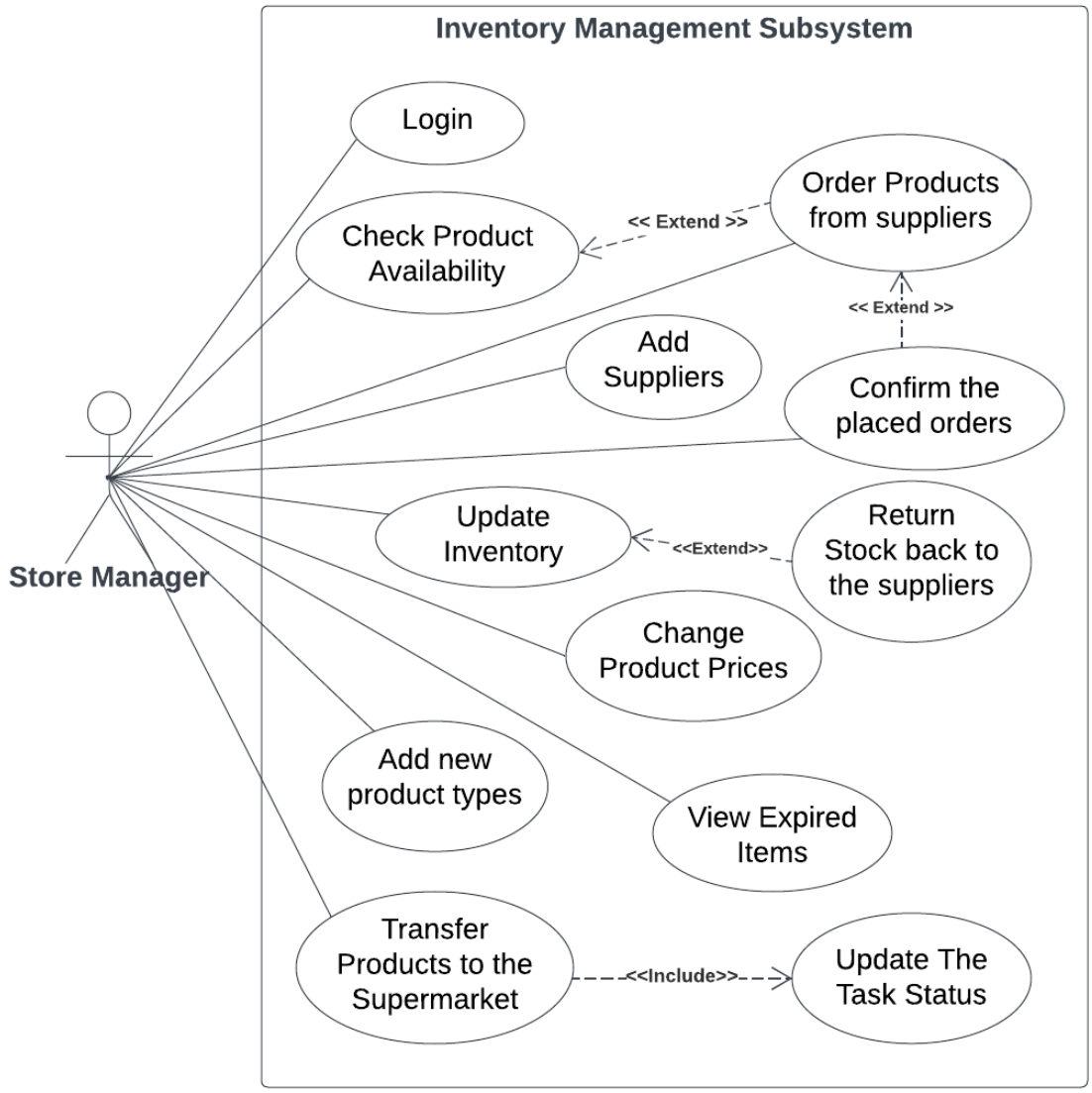

Figure 4-2

4.1.2.1 Login

Use Case Name	Login
Actor	Store Manager
Description	This use case involves the process of the Store Manager logging into the inventory management system. The user should be able to log in to the system.
Preconditions	The user must be registered in the system

Main Flow	1. The Store Manager opens the inventory management system application or navigates to the login page on the web interface. 2. The manager enters their username and password in the provided fields. 3. The system validates the entered credentials.
Successful end/ post condition	If the entered credentials are valid, the Store Manager gains access to the inventory management system. The manager is redirected to the system's dashboard or home page.
Fail End/ post condition	If the entered credentials are invalid, the system displays an error message indicating the login attempt was unsuccessful. The Store Manager can retry entering the correct credentials or may need to contact the system administrator for assistance.
Extension	None

4.1.2.2 Order Products from Suppliers

Use Case Name	Order Products from suppliers
Actor	Store Manager
Description	Need to order products from the suppliers
Preconditions	The store Manager should log into the system. The list of products to be ordered has been prepared.
Main Flow	1. Store Manager accesses the inventory system. 2. Navigate to the “Order Products” section. 3. The system presents a form to enter the details of the order. 4. The staff enters the following information: Supplier details (name, contact information) Product details (name, quantity, unit price) Delivery date preference Any additional notes 5. Place the order by clicking on the “Place Order” button
Successful end/post condition	1. Display the Success massage of the order placement. 2. Order details are saved in the system and displayed in the “Order Confirmation" section.
Fail End/ post condition	An error message along with the error will be displayed.

Extension	N/A

4.1.2.3 Check Product Availability

Use Case Name	Check Product Availability
Actor	Store Manager
Description	See the available product details and find the availability of the products based on various criteria such as type, category, brand, and amount.
Preconditions	1. The store Manager should log into the system. 2. The inventory database is up to date with the latest product information.
Main Flow	1. The Store Manager accesses the inventory management system. 2. The manager navigates to the "Check Product Availability" section. 3. The system presents a search form with various filtering options. 4. The manager enters the desired criteria to filter the available products: • Product type (e.g., electronics, groceries, clothing) • Product category (e.g., TV, vegetables, shirts) • Product brand (e.g., Sony, Apple, Nike) • Desired amount or quantity 5. Press the “Show” button and then should display the available products
Successful end/post condition	1. The system retrieves a list of products that match the specified criteria. 2. The list includes product details such as name, quantity in stock, price, and location within the store. 3. The manager can view the list of available products on the screen.
Fail End/ post condition	1. If the search criteria are too broad or contain conflicting filters, the system could prompt the manager to refine the criteria. 2. If there are technical issues, the system might display an error message. 3. If no products match the specified criteria, the system could provide a message indicating no products are available.
Extension	If you encountered a low stock before auto informing the low stocks, order those low-stock products. (In case of a special

	order etc.)

4.1.2.4 Confirm The Placed Orders

Use Case Name	Confirm The Placed Orders
Actor	Store manager
Description	All the manually auto placed orders must be confirmed before sent to the suppliers. This confirmation is handled here
Preconditions	Order must have been placed (by manually or automatically) Store Manager should log into the system
Main Flow	1. Navigates to the "Confirm Placed Order" section. 2. Placed orders are displayed by the system. 3. Confirm each order by click on the “Confirm” button. 4. If needed, cancel the order by click on the “Cancel Order” button. 5. Can view all the placed orders in details and the states of the order (not confirmed yet, confirmed, cancel)
Successful end/post condition	Display the order confirmation or the rejection. An order confirmation or reference number is generated for future tracking. An email or notification is sent to the supplier with the confirmed order details.
Fail End/ post condition	If failed to confirm or cancel the order, error message should be displayed.
Extension	N/A

4.1.2.5 Add Suppliers

Use Case Name	Add Suppliers
Actor	Store Manager
Description	Add new supplier details into the system.
Preconditions	The store manager should log in to the system.
Main Flow	1. Go to the Add Suppliers" section. 2. Enter the supplier’s details and the supplying product details. 3. Click the “Save Supplier” button and save.
Successful	Display success message. (Successfully added a supplier)

end/post condition	Update the database with the new supplier details.
Fail End/ post condition	If the supplier information is incomplete or contains errors, the system could prompt the staff to correct the information. If there are technical issues, the system might display an error message and prevent the addition of the supplier.
Extension	N/A

4.1.2.6 Update Inventory

Use Case Name	Update Inventory after Product Delivery
Actor	Store manager
Description	This use case involves the process of updating the inventory after products are delivered by suppliers, selecting items to be returned to the supplier due to damage or errors, and marking orders as completed.
Preconditions	The Store Manager has logged into the inventory management system. Products have been delivered by suppliers, and their corresponding orders are in the system.
Main Flow	1. The manager navigates to the "Update Inventory". 2. confirms the quantity received and any discrepancies. 3. Confirm the order arrivals from suppliers. 4. The system updates the inventory by adding the received quantity of the product. 5. The manager decides if any products need to be returned to the supplier due to damage or errors and mark them as returning the products.
Successful end/post condition	The inventory is updated with the received products and their quantities. The inventory is updated with the received products and their quantities. The selected items to be returned are updated in the “Return Items to Suppliers” section. The system may generate inventory-related reports, reflecting the changes.
Fail End/ post condition	If there are technical issues, the manager might encounter difficulties in updating the inventory, and the system could display an error message.

Extension	N/A

4.1.2.7 Return Stock Back to The Suppliers

Use Case Name	Return Stock Back to the suppliers
Actor	Store manager
Description	This use case involves the process of returning damaged or incorrect items received from suppliers back to them for replacement or refund.
Preconditions	The Store Manager has logged into the inventory management system. The received order contains damaged or incorrect items and then the selected orders have been sent back for returning processing when the updating inventory.
Main Flow	1. navigates to the "Return Items to Suppliers" section. 2. The system displays a list of recently received orders with damaged or incorrect items. 3. The system provides details of the order, including the products, quantities, and reasons for return (damaged or incorrect). 4. The store manager verifies the information and confirms the return for the specific items.
Successful end/post condition	Display the success message of requesting the return to the supplier. The system generates a return request to the supplier.
Fail End/ post condition	1. If the return request contains incorrect information or doesn't meet the supplier's return criteria, the system could prompt the manager to make corrections. 2. If there are technical issues, the manager might encounter difficulties in initiating the return process, and the system could display an error message.
Extension	The system could track the status of the return request and provide notifications to the manager about each step of the process.

4.1.2.8 Add New Product Types

Use Case Name	Add New Product Types
Actor	Store Manager
Description	This use case involves the process of adding new product types to the inventory management system.
Preconditions	The store manager should log into the system. Must have the access to add new item.
Main Flow	1. This use case involves the process of adding new product types to the inventory management system. 2. The system displays a list of existing product types and an option to add a new one. 3. Selects the option to add a new product type. 4. The system presents a form for entering details of the new product type and the manager enters the details. 5. The manager submits the information to create the new product type.
Successful end/post condition	Display a success message if adding a new product type is done. The Store Manager can now use this product type when adding new products to the inventory.
Fail End/ post condition	If the information entered is incomplete or contains errors, the system could prompt the manager to correct the information.
Extension	N/A

4.1.2.9 Change Product Prices

Use Case Name	Change Product Prices
Actor	Store manager
Description	This use case involves the process of changing the prices of products in the inventory management system, which is a common and frequent task for some products.
Preconditions	The Store Manager has logged into the inventory management system. The manager has the necessary authorization to update product prices.
Main Flow	1. The manager navigates to the "Change Product Prices" section.

	2. The system displays a list of products with their current prices (here the prices are). 3. The manager selects the product(s) whose price needs to be changed. 4. The system presents a form to input the new price for the selected product(s). 5. The manager enters the new price for each selected product. 6. The manager submits the changes.

4.1.2.10 Transfer Products to the supermarket

Use Case Name	Transfer Products to the Supermarket
Actor	Store Manager
Description	All the automatically created product transfer tasks when encountering low stocks should be transferred to the supermarket and updated in the system by the store manager. The Store Manager can then view the tasks, check product details, and mark tasks as completed when the transfer is done.
Preconditions	The Inventory Management System monitors inventory levels and has the capability to create tasks. The Store Manager has logged into the inventory management system.
Main Flow	1. The Store Manager accesses the inventory management system. 2. The manager navigates to the "Product Transfer Tasks" or "Inventory Tasks" section. 3. The system displays a list of pending product transfer tasks. 4. The manager selects a specific task to view the details. 5. The system provides the following details for the task: • Product details (name, count, price, expiry date, etc.) • Source location (current supermarket cell) • Destination location (supermarket stack) 6. Mark the task as “Completed” when the store manager completed the product transferring task.
Successful end/post condition	The system updates the task status to "Completed." The system adjusts inventory levels based on the completed task. The Store Manager can view completed tasks and their associated products. The Store Manager can also view the updated inventory levels

	after the transfer.
Fail End/ post condition	If the manager encounters any discrepancies or issues with the product transfer, they can choose not to mark the task as completed and take necessary actions. If there are technical issues, the manager might encounter difficulties in marking tasks as completed, and the system could display an error message.
Extension	N/A

4.1.2.11 View Expired Items

Use Case Name	View Expired Items
Actor	Store Manager
Description	This use case involves the process of viewing expired items and items that are about to expire in the inventory management system.
Preconditions	The Store Manager has logged into the inventory management system. The inventory data includes expiration date information for products.
Main Flow	1. The Store Manager accesses the inventory management system. 2. The manager navigates to the "View Expired Items" or "Expiration Status" section. 3. The system displays options to view either expired items or items that are about to expire. 4. The manager selects the desired option (expired items or items to expire). 5. For viewing expired items, the system generates a list of products that have exceeded their expiration dates. Then manager can remove the product from the inventory. 6. For viewing items to expire the system generates a list of products that are nearing their expiration dates.
Successful end/post condition	Remove the expired and item from the inventory. Update the removed expired item section. View the items to expire.
Fail End/ post condition	If there are no expired items or items about to expire, the system could display a message indicating that there are no items to

	display.
Extension	For the items to expire, the store manager can add discounts on them.

4.1.3 Employee Management Subsystem

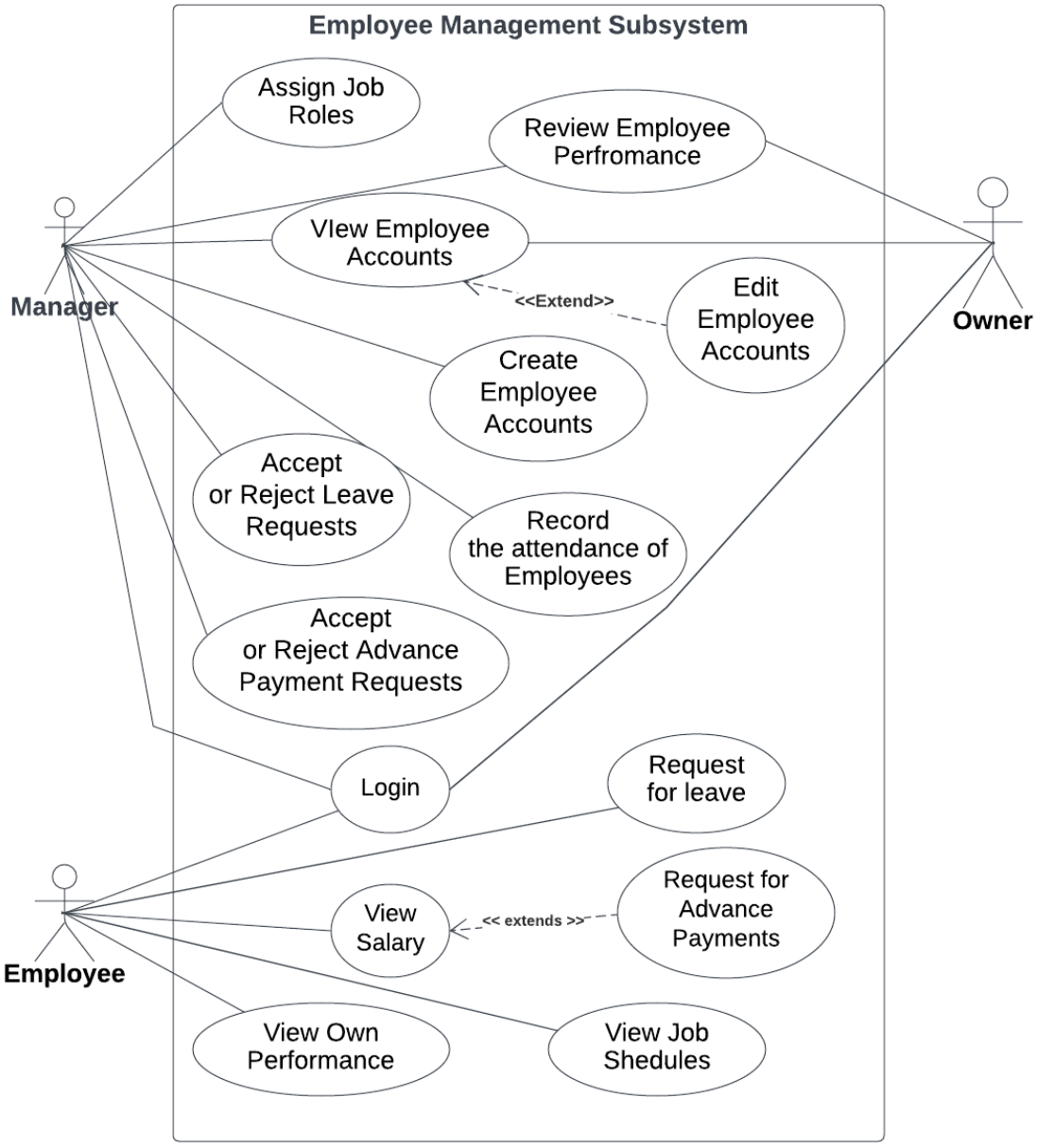

Figure 4-3

4.1.3.1 Assign Job Roles

Use case name	Assign Job Roles
Actor	Manager
Description	The manager can change the employee's job roles per week such as cashier, floor assistant, or stocker.
Preconditions	The manager has to log into the employee management system.
Main flow	1. Log into the employee management system. 2. The manager has to navigate to the Assign Job Role section. The system presents a list of employees along with their current roles. 3. Select a job role for each employee by the dropdown menu. 4. Press the ok button to update the changes
Successful end/ post condition	The manager has to navigate to the Assign Job Role section.
Fail end/ post condition	The system presents a list of employees along with their current roles.
Extensions	Select a job role for each employee by the dropdown menu.

4.1.3.2 Review Employee Performance

Use case name	Review employee performance
Actor	Manager and Owner
Description	This use case involves the process of a manager/owner reviewing the performance of each employee and assigning points based on their achievements, skills, and contributions. The points can be used for performance evaluations, recognition, and rewards.
Preconditions	The manager/owner must log into the employee management system.
Main flow	1. Log into the employee management system. 2. The manager/owner must navigate to the Performance Review section. The system presents a list of employees with profile pictures. 3. The manager/owner selects the employee they wish to review the performance from the list and based on performance metrics, the manager can assign the points to the employee.

	4. After reviewing and giving the points to the employee, the manager can save the edited details from save button
Successful end/ post condition	The employee's performance has been reviewed, points have been assigned, and the system's performance records are updated.
Fail end/ post condition	The manager encounters an issue such as a system error or incomplete performance data. The point assignment is not successfully recorded in the system.
Extensions	N/A

4.1.3.3 Create Employee Accounts

Use case name	Create Employee Accounts
Actor	Manager
Description	The manager can create an employee account for the employees who are working under the manager
Preconditions	The manager must log into the employee management system.
Main flow	1. Log into the employee management system. 2. The manager must navigate to the Create Employee Account section. The system presents a form for entering new employee information. 3. The manager has to fill in the required details for the new employee. 4. After confirming the details, press the ok button to create the account.
Successful end/ post condition	A new employee account has been successfully created in the system.
Fail end/ post condition	Error: The account creation is not successfully recorded in the system.
Extensions	N/A

4.1.3.4 View the Employee Account

Use case name	View Employee Account
Actor	Manager and Owner
Description	This use case involves the process of a manager accessing and

	viewing an employee's account to gather information about their profile, work history, performance, and other relevant details
Preconditions	The manager/owner must log into the employee management system.
Main flow	1. Log into the employee management system. 2. The manager/owner has to navigate to the Employee Record section. The system presents a list of employees with profile pictures. 3. The manager/owner selects the employee they wish to view from the list. The system displays a comprehensive account overview for the selected employee
Successful end/ post condition	The manager/owner has successfully reviewed the selected employee's account.
Fail end/ post condition	The manager encounters an issue, such as a system error or lack of proper access permissions. The employee's account information is not successfully displayed.
Extensions	Edit the Employee Account

4.1.3.5 Edit Employee Account

Use case name	Edit the Employee Account
Actor	Manager and Owner
Description	The manager/owner can edit an employee's account details, the account can include personal information, and contact information.
Preconditions	The manager/owner must log into the employee management system.
Main flow	1. Log into the employee management system. 2. The manager/owner must navigate to the Employee Record section. The system presents a list of employees with profile pictures. 3. The manager/owner selects the employee they wish to edit from the list. The system displays the current account details and an edit icon to edit the details. 4. Click the edit icon and edit details. 5. After editing the details, click the save button to save the edited details.

Successful end/ post condition	The manager/owner has successfully reviewed the selected employee's account.
Fail end/ post condition	The manager encounters an issue, such as a system error or lack of proper access permissions. The employee's account information is not successfully displayed.
Extensions	N/A

4.1.3.6 Record attendance

Use case name	Record the attendance of employees
Actor	Manager
Description	The manager must record the attendance of employees to track their working days.
Preconditions	The manager must log into the employee management system.
Main flow	1. Log into the employee management system. 2. The manager has to navigate to the Record the Attendance section. The system presents a date selection interface to select the date. 3. The system presents a list of employees. 4. The Manager has to choose present, absent, late, or excused by dropdown menu for each employee. 5. After confirming the attendance, press the ok button to update the changes.
Successful end/ post condition	Employee attendance has been successfully recorded for the chosen date, and the system's attendance records are updated accordingly.
Fail end/ post condition	The manager encounters an issue, such as incomplete data or a system error. The attendance recording is not successfully saved in the system.
Extensions	N/A

4.1.3.7 Request for leave

Use case name	Request for Leave
Actor	Employee
Description	This use case involves the process of a manager requesting a leave of absence on behalf of an employee in the supermarket.

	The manager initiates the request, and the system handles the approval workflow.
Preconditions	The employee must log into the system.
Main flow	1. Log into the system (want to discuss). 2. The employee must navigate to the Request for Leave section. The system presents a form of leave request after clicking the Request New Leave button and presents an approved leave request. 3. The employee can select a specific leave request from the dropdown menu given in the Leave Request form such as sick leave, or personal leave. 4. After specifying the leave type, dates, and any comments, the employee submits the leave request by clicking the submit button.
Successful end/ post condition	The employee's leave request has been successfully submitted through the system for manager approval.
Fail end/ post condition	The employee encounters an issue, such as a system error or incomplete information, preventing the leave request from being successfully submitted.
Extensions	N/A

4.1.3.8 Accepts or Reject Leave Request

Use case name	Accepts or Rejects Leave Requests
Actor	Manager
Description	This use case involves the process of a manager reviewing and responding to leave requests submitted by employees. The manager can either accept or reject the requests based on factors such as team workload, employee availability, and business needs.
Preconditions	The manager must log into the employee management system.
Main flow	1. Log into the employee management system. 2. The manager must navigate to the Approve Leave section. The system displays a list of leave requests. 3. The Manager must choose to accept or reject by dropdown menu for each leave request.

Successful end/ post condition	The manager has successfully reviewed and responded to the leave request, and the employee has been notified of the decision.
Fail end/ post condition	The manager encounters an issue, such as a system error or incomplete information, and the leave request status is not successfully updated.
Extensions	N/A

4.1.3.9 Request for Advance payments

Use case name	Request for Advance payments
Actor	Employee
Description	The employee requests an advance payment from the supermarket for a specific reason, such as unexpected expenses or financial needs.
Preconditions	The employee must log into the system. (Want to be discussed)
Main flow	1. Log into the system (want to discuss). 2. The employee must navigate to the Advance Request section. The system presents a form for the employee to fill required information, the history of advance that was taken by the employee and approved or rejected advance requests. 3. After filling out the form, the employee can submit the request by clicking the submit button.
Successful end/ post condition	The employee's advance payment request has been successfully submitted.
Fail end/ post condition	The employee's advance payment request is declined due to eligibility criteria not being met or other reasons.
Extensions	N/A

4.1.3.10 Accept or Reject Payment Request

Use case name	Accepts or Rejects Advance Payment Requests
Actor	Manager
Description	This use case involves the process of a manager reviewing and either accepting or rejecting payment requests made by

	employees, such as for reimbursements, expense claims, or other financial matters.
Preconditions	The manager must log into the employee management system.
Main flow	1. Log into the employee management system. 2. The manager has to navigate to the Approve Advance Payment section. The system displays a list of Payment requests. 3. The manager can select a specific payment request from the list. The system shows details of payment requests such as reason, and importance. 4. The Manager has to choose to accept or reject by dropdown menu for each payment request after considering the details of the request.
Successful end/ post condition	The manager has successfully reviewed and responded to the payment request, and the employee has been notified of the decision.
Fail end/ post condition	The manager encounters an issue, such as a system error or incomplete information, and the payment request status is not successfully updated.
Extensions	None

4.1.3.11 View Salary

Use case name	View Salary
Actor	Employee
Description	This use case involves the process of an employee accessing the system to view their salary details, including earnings, deductions, bonuses, and other related information.
Preconditions	The employee must log into the system.
Main flow	1. Log into the system. 2. The employee has to navigate to the Salary section. The system presents salary information after clicking the View Salary Button. 3. The employee can exit from the salary view by clicking the exit button after reviewing the salary.
Successful end/ post condition	The employee has successfully viewed their salary information for the current pay period.

Fail end/ post condition	The employee encounters an issue, such as a system error or incorrect login credentials. The salary information is not successfully displayed.
Extensions	Request for Advance payments

4.1.3.12 View Job Schedules

Use case name	View Job Schedules
Actor	Employee
Description	This use case involves the process of an employee in a supermarket accessing and viewing their job assignments to know whether they are assigned to inventory-related tasks or working as a floor assistant.
Preconditions	The employee must log into the system.
Main flow	1. Log into the system (want to discuss). 2. The employee has to navigate to the My Job Assignments section. The system presents the employee's job assignments, detailing whether they are assigned to inventory-related tasks or working as a floor assistant. 3. After reviewing their assignment, the employee can exit the My Job Assignments section by clicking the exit button.
Successful end/ post condition	The employee has successfully viewed their job assignments, understanding whether they are assigned to inventory work or working as a floor assistant
Fail end/ post condition	The assignment information is not successfully displayed.
Extensions	N/A

4.1.3.13 View Own Performance

Use case name	View Own Performance
Actor	Employee
Description	This use case involves the process of an employee accessing and viewing their own performance data.
Preconditions	The employee must log into the system.

Main flow	1. Log into the system. 2. The employee must navigate to the My Job Performance section. The system presents the performance under a certain time period. 3. After reviewing their performance, the employee can exit the My Performance section by clicking the exit button.
Successful end/ post condition	The employee has successfully viewed their job assignments, understanding whether they are assigned to inventory work or working as a floor assistant
Fail end/ post condition	The performance data is not successfully displayed
Extensions	None

4.1.4 Customer Management Subsystem

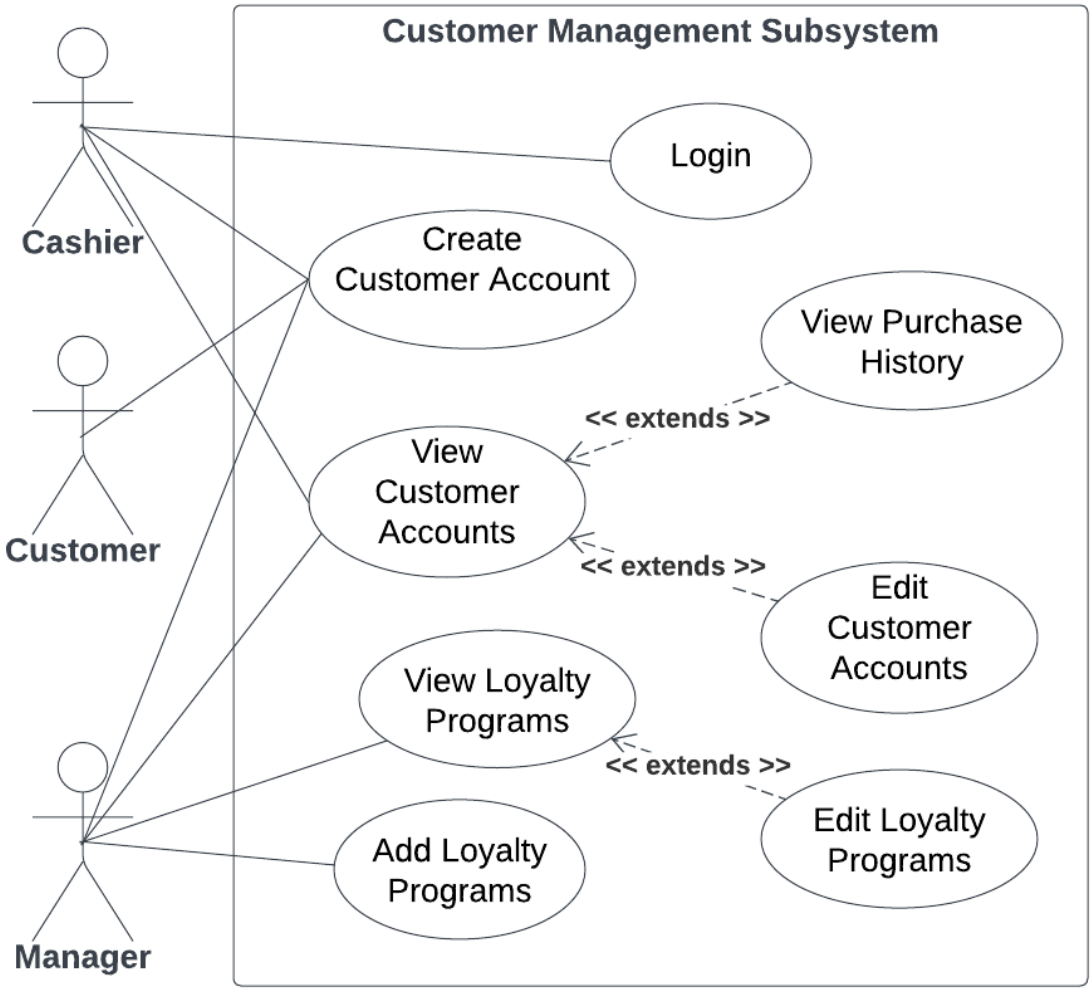

Figure 4-4

4.1.4.1 Create Customer Account

Use case name	Create customer account
Actor	Cashier, Customer
Description	This use case allows the cashier to create a new customer account in the system.
Preconditions	The cashier is logged in. The cashier has the necessary customer information.
Main flow	1. The cashier selects the "Create Customer Account" option from the customer management menu.

	2. The system displays a form for entering customer details, including name, contact information, and optional loyalty program enrollment. 3. The cashier enters the customer's information into the form. 4. The cashier submits the form to the system. 5. The system validates the entered information. 6. If the information is valid, the system creates a new customer account and assigns a unique customer ID. 7. The system stores the customer's information, including their loyalty program status, in the database. 8. The system confirms the successful creation of the customer account and displays a confirmation message.
Successful end/ post condition	If the entered information is invalid or incomplete, the system displays an error message and prompts the cashier to correct the information.
Fail end/ post condition	If the selected store location does not have sufficient data available, the system informs the store owner that there is no performance data to display.
Extensions	N/A

4.1.4.2 View Customer Accounts

Use case name	View customer accounts
Actor	Cashier, Manager
Description	This use case allows the cashier or manager to view a list of customer accounts stored in the system.
Preconditions	The actor (cashier or manager) is logged into the system.
Main flow	1. The actor selects the "View Customer Accounts" option from the customer management menu. 2. The system retrieves a list of customer accounts from the database. 3. The system displays a list of customer names and basic information.
Successful end/ post condition	The actor can see the list of customer accounts.
Fail end/ post condition	N/A
Extensions	• Edit customer account.

	• View purchase history.

4.1.4.3 View Purchase History

Use case name	View purchase history
Actor	Cashier, Manager
Description	This use case allows the cashier or manager to view the purchase history of a specific customer. It provides information about the products purchased, transaction dates, and payment methods.
Preconditions	The actor (cashier or manager) is logged into the system. The actor has already accessed the customer's account.
Main flow	1. The actor selects the "View Purchase History" option from the customer account details. 2. The system retrieves the purchase history associated with the selected customer from the database. 3. The system displays a list of past transactions, including product details, transaction dates, and payment methods.
Successful end/ post condition	The actor can see the purchase history of the selected customer.
Fail end/ post condition	If the customer has no purchase history, the system displays a message indicating that there is no purchase history available.
Extensions	N/A

4.1.4.4 Edit Customer Accounts

Use case name	Edit customer accounts
Actor	Cashier, Manager
Description	This use case allows the cashier or manager to edit the details of an existing customer account. It enables updating customer information such as contact details, loyalty program status, and personal preferences.
Preconditions	The actor (cashier or manager) is logged into the system. The actor has already accessed the customer's account.
Main flow	1. The actor selects the "Edit Customer Account" option from the customer account details. 2. The system retrieves the customer's current information from the database.

	3. The system displays the customer's information in editable fields, such as name, contact details, and loyalty program preferences. 4. The actor makes the necessary changes to the customer's information. 5. The actor confirms the changes and submits the updated information.
Successful end/ post condition	The customer's account information is updated with the new changes.
Fail end/ post condition	If there are any errors or issues during the editing process, the system displays an error message, and the customer's account information remains unchanged.
Extensions	N/A

4.1.4.5 Add Loyalty Program

Use case name	Add loyalty program
Actor	Manager
Description	This use case allows a manager to create and add a new loyalty program to the system. A loyalty program rewards customers for their repeat business and encourages them to make more purchases.
Preconditions	The manager is logged into the system.
Main flow	1. The manager selects the "Add Loyalty Program" option from the customer management menu. 2. The system presents a form to the manager to fill in details for the new loyalty program. 3. The manager enters the program name, description, rewards structure, and any applicable terms and conditions. 4. The manager specifies the criteria for customers to become eligible for the loyalty program (e.g., minimum purchase amount). 5. The manager sets the program's expiration rules if applicable. 6. The manager confirms the program details and submits the new loyalty program.
Successful end/ post condition	The new loyalty program is added to the system and is available for customers to enroll.

Fail end/ post condition	If there are any errors or missing information in the loyalty program details, the system displays an error message, and the program is not created.
Extensions	N/A

4.1.4.6 View Loyalty Programs

Use case name	View loyalty program
Actor	Cashier, Manager
Description	This use case allows cashiers and managers to view the list of available loyalty programs in the system. It provides them with information about the loyalty programs currently offered to customers.
Preconditions	The actor (cashier or manager) is logged into the system.
Main flow	1. The actor selects the "View Loyalty Programs" option from the customer management menu. 2. The system retrieves the list of available loyalty programs from the database. 3. The system displays a list of loyalty programs, including their names, descriptions, and rewards structures. 4. The actor can scroll through the list to view the details of different loyalty programs.
Successful end/ post condition	The actor is presented with the list of available loyalty programs and their details.
Fail end/ post condition	If there is an issue retrieving the loyalty program information from the database, the system displays an error message, and the actor is unable to view the list of programs.
Extensions	• Edit loyalty programs

4.1.4.7 Edit Loyalty Programs

Use case name	Edit loyalty program
Actor	Manager
Description	This use case allows managers to edit the details of existing loyalty programs. Managers can modify the program's name, description, rewards structure, or any other relevant information.
Preconditions	The manager is logged into the system.

	Manager has accessed the "View Loyalty Programs" section.
Main flow	1. The manager selects a specific loyalty program from the list displayed in the "View Loyalty Programs" section. 2. The system retrieves the detailed information of the selected loyalty program from the database. 3. The manager chooses the "Edit Loyalty Program" option. 4. The system presents a form to the manager with pre-filled fields containing the current information of the selected loyalty program. 5. The manager modifies the necessary fields such as program name, description, rewards structure, eligibility criteria, etc. 6. The manager submits the edited information.
Successful end/ post condition	If there is an issue updating the loyalty program's information, the system displays an error message, and the changes made by the manager are not saved.
Fail end/ post condition	If there is an issue retrieving the loyalty program information from the database, the system displays an error message, and the actor is unable to view the list of programs.
Extensions	N/A

4.1.5 Report and Analytics

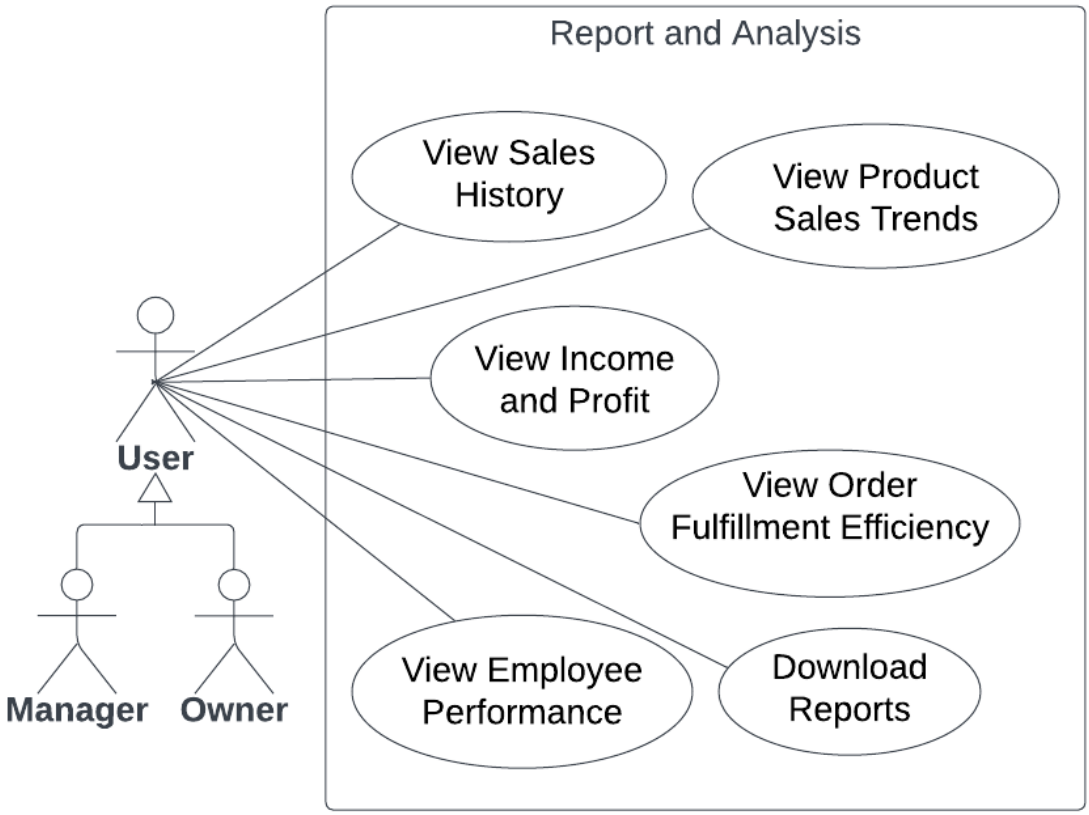

Figure 4-5

4.1.5.1 View Sales History

Use Case Name	View Sales History
Actor	Manager, owner
Description	Viewing the sales history and filter based on date, sales,
Preconditions	The user has logged into the inventory management system. The inventory system has been recording sales data.
Main Flow	1. Navigates to the "Reports" or "Sales History" section. 2. The system provides options to filter the sales history based on specific time frames, products, or other relevant criteria. 3. The system generates and displays a report showing sales history based on the selected filters. 4. The manager reviews the report, which includes details such as sales date, product names, quantities sold, and total revenue.
Successful	The Store Manager can view a comprehensive sales history

end/post condition	report based on the selected filters. The report provides insights into sales trends, popular products, and revenue generation over the specified time frame.
Fail End/ post condition	If no sales data exists within the selected time frame or filters, the system could display a message indicating that there are no sales to display.
Extension	The system could offer visualization options such as graphs or charts to better represent sales trends.

4.1.5.2 View Income and Profit

Use Case Name	View Income and Profit
Actor	Manager, Owner
Description	This use case involves the process of viewing income and profit data in the inventory management system.
Preconditions	The Store Manager has logged into the inventory management system. The system has been recording sales data and associated costs.
Main Flow	1. User navigates to the "Reports" or "Income and Profit" section. 2. The system provides options to filter the income and profit data based on specific time frames or other relevant criteria. 3. User selects the desired filters to narrow down the income and profit data. 4. The system generates and displays a report showing income, costs, and calculated profit based on the selected filters.
Successful end/post condition	The user can view a comprehensive income and profit report based on the selected filters.
Fail End/ post condition	If no relevant financial data exists within the selected time frame or filters, the system could display a message indicating that there is no data to display.
Extension	The system could offer visualization options such as graphs or charts to better represent income and profit trends. Download the reports.

4.1.5.3 View Product Sales Trends

Use Case Name	View Product Sales Trends
Actor	Manager, Owner
Description	This use case involves the process of viewing sales trends for specific products in the inventory management system. The can view the product's sales trends report, providing insights into how the product's sales have changed over time.
Preconditions	The Store Manager has logged into the inventory management system. The inventory system has been recording detailed sales data including individual product sales.
Main Flow	1. Navigates to the "Reports" or "Product Sales Trends" section. 2. The system presents a list of available products or a search feature to find the desired product. 3. User selects a specific product to view its sales trends. 4. The system generates and displays a report showing the sales history and trends for the selected product over a specified time frame.
Successful end/post condition	Display the product's sales trends report, providing insights into how the product's sales have changed over time.
Fail End/ post condition	If there are no sales data for the selected product or time frame, the system could display a message indicating that there are no sales to display. If there are technical issues, the manager might encounter difficulties in generating the sales trends report, and the system could display an error message.
Extension	Download the reports.

4.1.5.4 Download Reports

Use Case Name	Download Reports
Actor	Owner, Manager
Description	This use case involves the process of owners and managers downloading various reports in PDF format from the inventory management system.

Preconditions	Users have logged into the inventory management system. View the particular report provided by the system
Main Flow	1. Users watch the report provided by the system. 2. The system displays a list of available reports. 3. Press the “Download Button” to download the report.
Successful end/post condition	The owners and managers can download the generated report in PDF format.
Fail End/ post condition	If there are technical issues, owners and managers might encounter difficulties in generating or downloading the report, and the system could display an error message.
Extension	N/A

4.1.5.5 View Order Fulfillment Efficiency

Use Case Name	View Order Fulfillment Efficiency
Actor	Manager, Owner
Description	This use case involves the process of tracking and viewing order fulfillment efficiency in the inventory management system. The report provides insights into how effectively orders are being fulfilled, helping to identify areas for improvement.
Preconditions	The Store Manager has logged into the inventory management system. The system records data related to order fulfillment processes.
Main Flow	1. The user navigates to the "Reports" section. 2. The system provides options to filter the order fulfilment data based on specific time frames or other relevant criteria. 3. The user selects the desired filters to narrow down the order fulfilment data. 4. The system generates and displays a report showing order fulfilment efficiency metrics based on the selected filters. 5. The report includes details such as the percentage of orders fulfilled on time, average fulfilment time, and any delays.
Successful end/post condition	Display a comprehensive report on order fulfillment efficiency based on the selected filters.
Fail End/ post condition	If there is no relevant order fulfillment data within the selected time frame or filters, the system could display a message indicating that there is no data to display.
Extension	N/A

4.1.6 Ecommerce Integration Subsystem

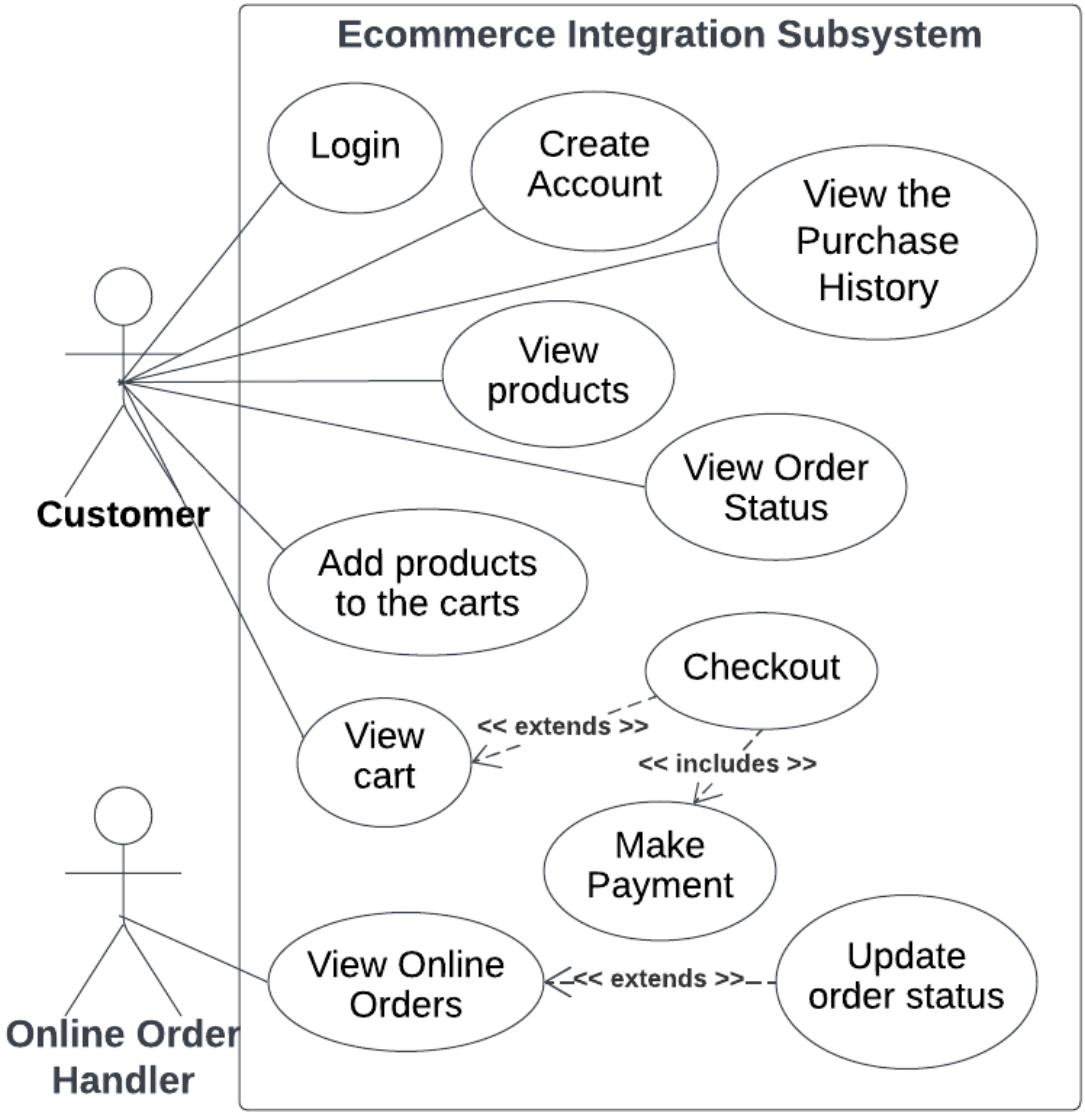

Figure 4-6

4.1.6.1 Create Customer Account

Use case name	Create Account
Actor	Customer
Description	The customer can create a customer account
Preconditions	The customer must visit the web.
Main flow	1. Visit the web. 2. The Customer has to navigate to the Create New Account section.

	The system presents a form for entering new customer information. 3. The customer has to fill in the required details for the new customer. 4. After confirming the details, press the OK button to create the account.
Successful end/ post condition	A new customer account has been successfully created in the system.
Fail end/ post condition	Error: The account creation is not successfully recorded in the system.
Extensions	None

4.1.6.2 View Products

Use case name	View the products
Actor	Customer
Description	This use case involves the process of a customer accessing the web to view products for purchase.
Preconditions	The customer must visit the web.
Main flow	1. Visit the web. 2. The customer must navigate to the Products section. The system presents products.
Successful end/ post condition	The customer has successfully viewed the products.
Fail end/ post condition	The customer encounters an issue, such as a system error or incorrect login credentials. The products are not successfully displayed.
Extensions	None

4.1.6.3 Add Products to the Cart

Use case name	Add Products to the Cart.
Actor	Customer
Description	This use case involves the process of a customer accessing the web to add the products to the cart.
Preconditions	The customer must visit the web.

Main flow	1. Visit the web. 2. The customer must navigate to the **Products** section. The system presents products. 3. Click the cart icon below the selected product.
Successful end/ post condition	The customer has successfully added the products to the cart.
Fail end/ post condition	The customer encounters an issue, such as a system error or incorrect login credentials. The products are not successfully added to the cart.
Extensions	None

4.1.6.4 View Cart

Use case name	View Cart
Actor	Customer
Description	This use case involves the process of a customer accessing the web to view the cart.
Preconditions	The customer must visit the web.
Main flow	1. Visit the web. 2. The customer must navigate to the Products section. 3. Click the Cart tab to view the cart. 4. Customers can remove products from the cart by clicking the minus icon.
Successful end/ post condition	The customer has successfully viewed the cart.
Fail end/ post condition	The customer encounters an issue, such as a system error or incorrect login credentials. The cart is not successfully viewed.
Extensions	None

4.1.6.5 View the Purchase History

Use case name	View the Purchase History
Actor	Customer
Description	This use case involves the process of a customer accessing the web to view the purchase history.
Preconditions	The customer must visit the web.

Main flow	1. Visit the web. 2. The customer must navigate to the Products section. 3. Select the Purchase History tab. The system presents the history of purchasing products.
Successful end/ post condition	The customer has successfully viewed the history of purchasing products.
Fail end/ post condition	The customer encounters an issue, such as a system error or incorrect login credentials. The history of purchasing products is not successfully displayed.
Extensions	None

4.1.6.6 View the Order Status

Use case name	View the Order Status
Actor	Customer
Description	This use case involves the process of a customer accessing the web to view the order status.
Preconditions	The customer must visit the web.
Main flow	1. Visit the web. 2. The customer must navigate to the Products section. 3. Select the Order tab. The system presents the status of ordered products.
Successful end/ post condition	The customer has successfully viewed the order status.
Fail end/ post condition	The customer encounters an issue, such as a system error or incorrect login credentials. The order status is not successfully displayed.
Extensions	None

4.1.6.7 View Online Orders.

Use case name	View Online Orders.
Actor	Online Order Handler
Description	This use case involves the process of an online order handler accessing the web to view the ordered products by customers.
Preconditions	The online order handler must visit the web.
Main flow	1. Visit the web.

	2. The online order handler must navigate to the Order section. The system presents ordered products and details of purchase.
Successful end/ post condition	The online order handler has successfully viewed the ordered products.
Fail end/ post condition	The user encounters an issue, such as a system error or incorrect login credentials. The ordered products are not successfully displayed.
Extensions	Update Order Status

4.1.6.8 Update Order Status.

Use case name	Update Order Status.
Actor	Online Order Handler
Description	This use case involves the process of an online order handler accessing the web to update the ordered product status.
Preconditions	The online order handler must visit the web.
Main flow	1. Visit the web. 2. The online order handler must navigate to the Order section. The system presents ordered products and details of purchase. 3. Click the order that needs to be updated. 4. Select status from the dropdown menu.
Successful end/ post condition	The online order handler has successfully updated the status of ordered products.
Fail end/ post condition	The user encounters an issue, such as a system error or incorrect login credentials. The status of ordered products are not successfully updated.
Extensions	None

4.1.7 Multi-Store Support Management Subsystem

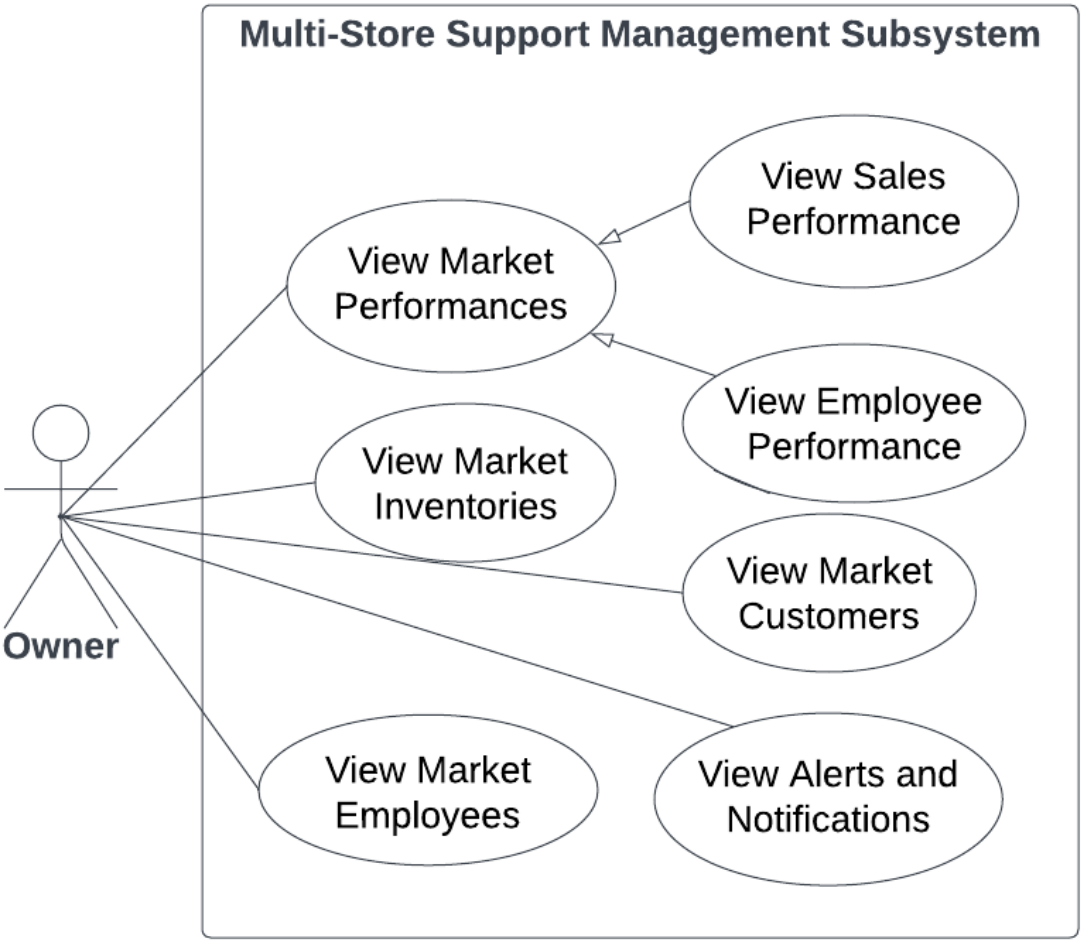

Figure 4-7

4.1.7.1 View Market Performances

Use case name	View Market Performance
Actor	Store Owner
Description	This use case allows the store owner to access and analyze the market performance data, including sales performance and employee performance, for a specific store location.
Preconditions	The store owner is logged into the system.
Main flow	1. The store owner selects the "View Market Performance" option from the multi-store management dashboard. 2. The system presents a list of available store locations.

	3. The store owner selects the desired store location for which they want to view market performance data. 4. The system retrieves and displays the sales performance metrics for the selected store location, which may include: • Total sales revenue for a specified time period. • Sales trends over time. • Best-selling products. • Average transaction value. 5. The system also presents employee performance metrics for the selected store location. 6. The store owner can customize the time period for which the performance data is displayed. 7. The system generates graphs, charts, and reports to visually represent the market performance data. 8. The store owner can drill down into specific metrics to gain deeper insights. 9. The store owner can compare the performance of different store locations if desired.
Successful end/ post condition	The store owner gains insights into the sales and employee performance of the selected store location.
Fail end/ post condition	If the selected store location does not have sufficient data available, the system informs the store owner that there is no performance data to display.
Extensions	N/A

4.1.7.2 View Market Inventories

Use case name	View Market Inventories
Actor	Store Owner
Description	This use case allows the store owner to view the inventory levels of products for each store location, providing insights into stock availability and potential restocking needs.
Preconditions	The store should logged into the system.
Main flow	1. The store owner selects the "View Market Inventories" option from the multi-store management dashboard. 2. The system presents a list of available store locations. 3. The store owner selects a specific store location to view its inventory.

	4. The system retrieves and displays the inventory data for the selected store location, including: List of products available in the store. Quantity of each product in stock. Low stock indicators for products below the reorder point. 5. The store owner can sort and filter the inventory list based on various criteria, such as product category or stock availability. 6. The system highlights products that are low in stock or have reached the reorder point. 7. The system may provide a visual representation of inventory trends over time.
Successful end/ post condition	The store owner gains visibility into the inventory levels of products for the selected store location and can identify items that require attention.
Fail end/ post condition	If the selected store location's inventory data is unavailable or there are technical issues, the system informs the store owner and suggests trying again later.
Extensions	N/A

4.1.7.3 View Market Customers

Use case name	View Market Customers
Actor	Store Owner
Description	This use case enables the store owner to view customer information and insights for a specific store location, helping them understand customer demographics, preferences, and behavior.
Preconditions	The store owner should be logged into the system.
Main flow	1. The store owner selects the "View Market Customers" option from the multi-store management dashboard. 2. The system presents a list of available store locations. 3. The store owner selects a specific store location to view customer data. 4. The system retrieves and displays customer information and insights for the selected store location, including: • List of registered customers associated with the store. • Customer demographics such as age, gender, and contact details.

	• Purchase history and preferences of customers. • Customer loyalty program participation. • Frequency of visits and average spending. 5. The store owner can filter customer data based on various criteria, such as loyalty program members or recent shoppers. 6. The system may provide visual representations of customer segments and trends over time.
Successful end/ post condition	The store owner gains valuable insights into the customer base of the selected store location, enabling targeted marketing and customer engagement strategies.
Fail end/ post condition	If the selected store location's customer data is unavailable or there are technical issues, the system informs the store owner and suggests trying again later.
Extensions	N/A

4.1.7.4 View Market Employees

Use case name	View Market Employees
Actor	Store Owner
Description	This use case allows the store owner to access and manage employee information for a specific store location, facilitating efficient employee management.
Preconditions	The store owner should be logged into the system.
Main flow	1. The store owner selects the "View Market Employees" option from the multi-store management dashboard. 2. The system displays a list of available store locations. 3. The store owner selects a specific store location to view employee data. 4. The system retrieves and presents a list of employees associated with the selected store location, including: Employee names, roles, and contact details. Work schedules, shifts, and availability. 5. The store owner can filter employee data based on criteria such as roles, shifts, or performance ratings. 6. The system may provide visual representations of employee workloads and scheduling patterns.

Successful end/ post condition	The store owner gains a comprehensive view of employees' information and schedules for the selected store location, allowing effective employee management and allocation.
Fail end/ post condition	If employee data is unavailable for the selected store location or technical issues occur, the system informs the store owner and suggests trying again later.
Extensions	N/A

5. Logical View

5.1 Overview

The logical view of the supermarket software system entails a modular decomposition into architecturally significant packages that encapsulate various functional aspects. These packages encompass Sales Processing, Inventory Management, Employee Management, Customer Management, Report and Analytics, E-commerce Integration, and Multi-Store Support. Each package contains classes responsible for specific functionalities, such as transaction processing, inventory tracking, employee profiles, customer records, and more. The system's design emphasizes the organization of responsibilities into distinct packages, fostering maintainability, scalability, and effective management of the supermarket's operations.

5.2 Architecturally Significant Design Packages

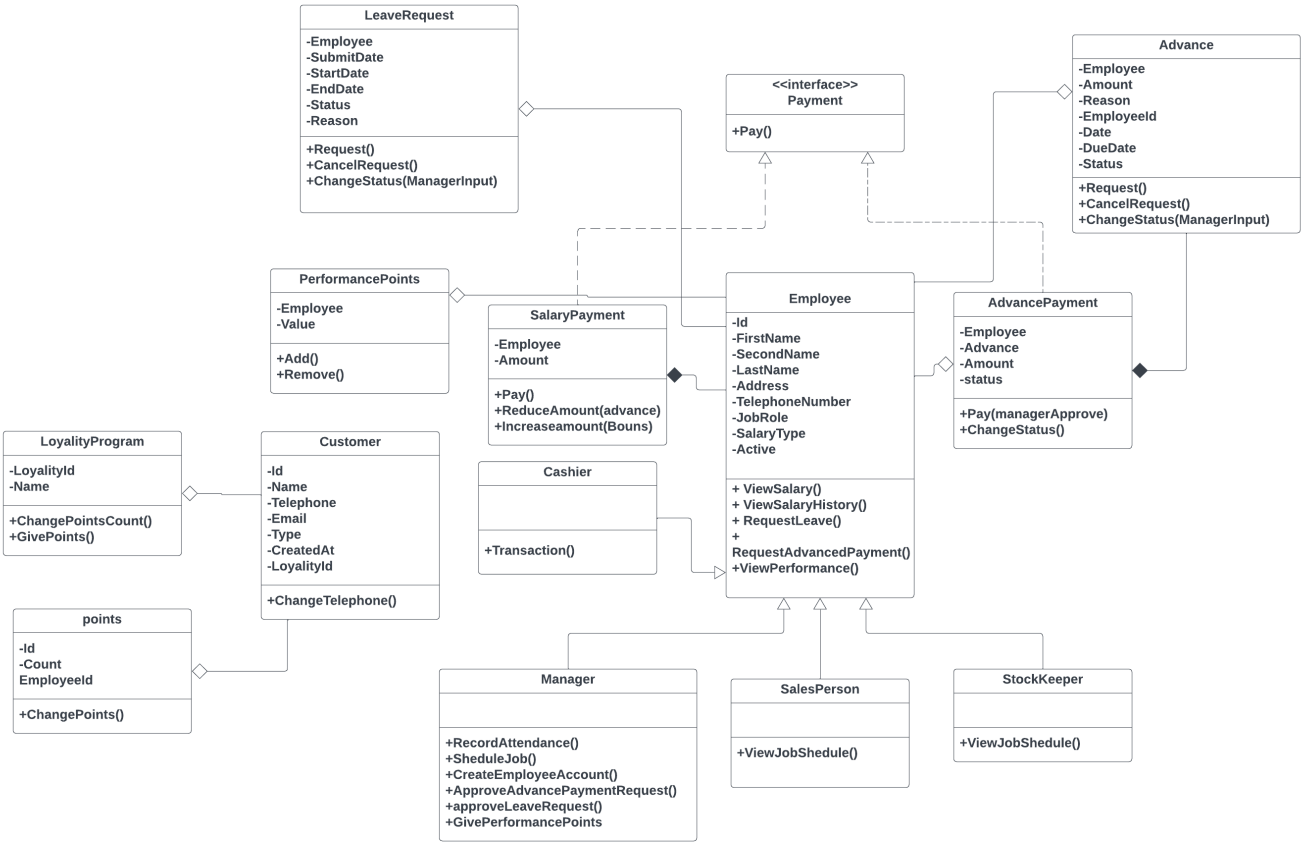

Figure 5-1 Class diagram part 1

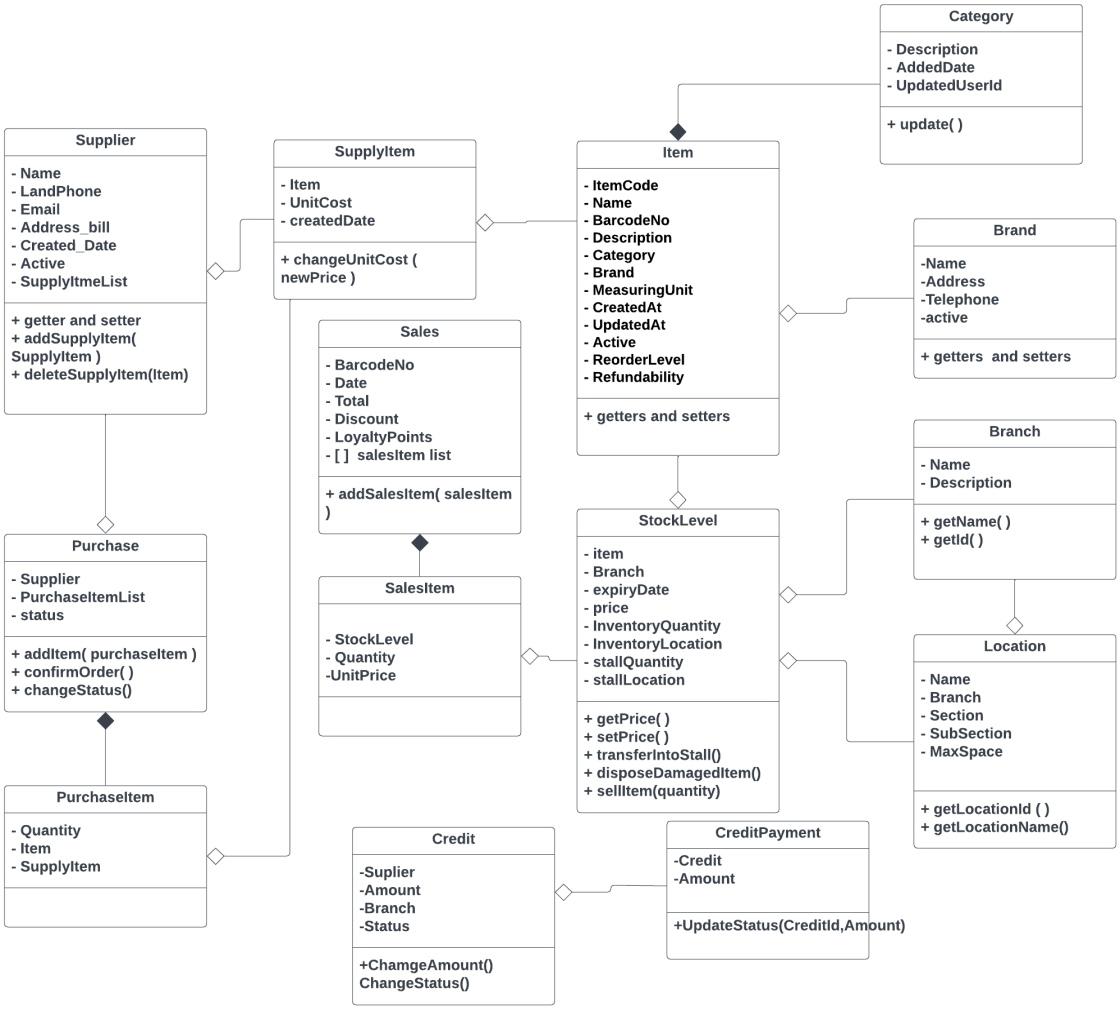

Figure 5-2 Class diagram part 2

6. Process View

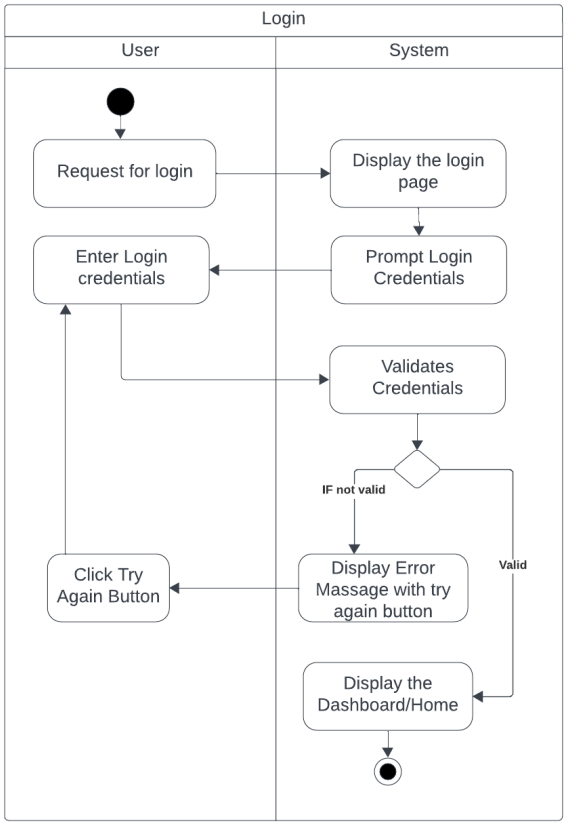

Figure 6-1

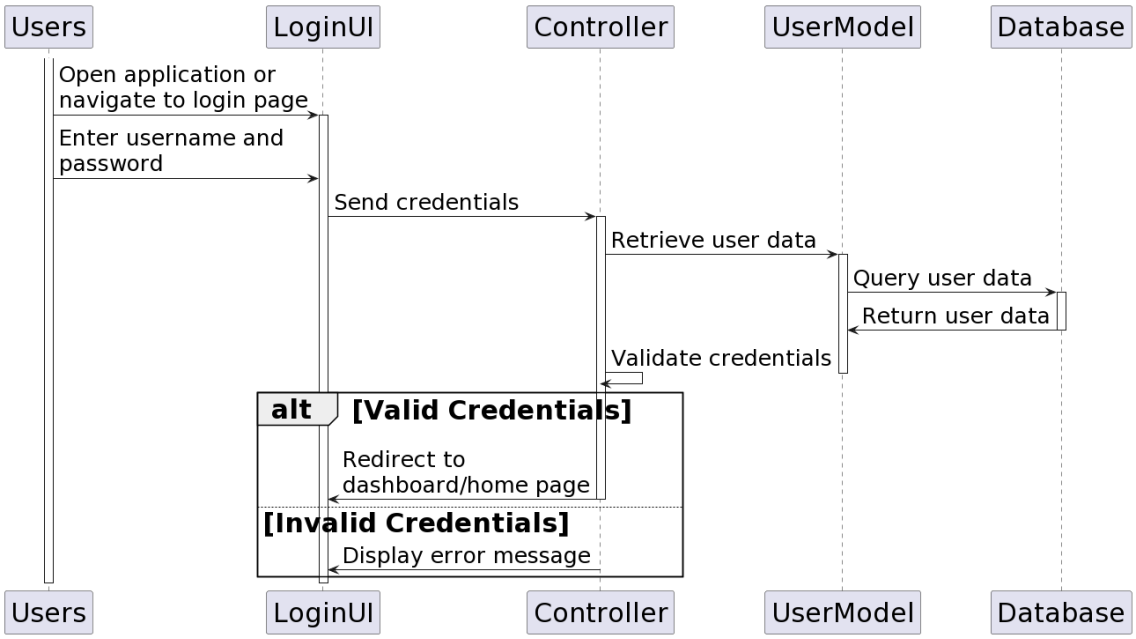

Figure 6-2

6.1 Sales Processing and Transaction Management

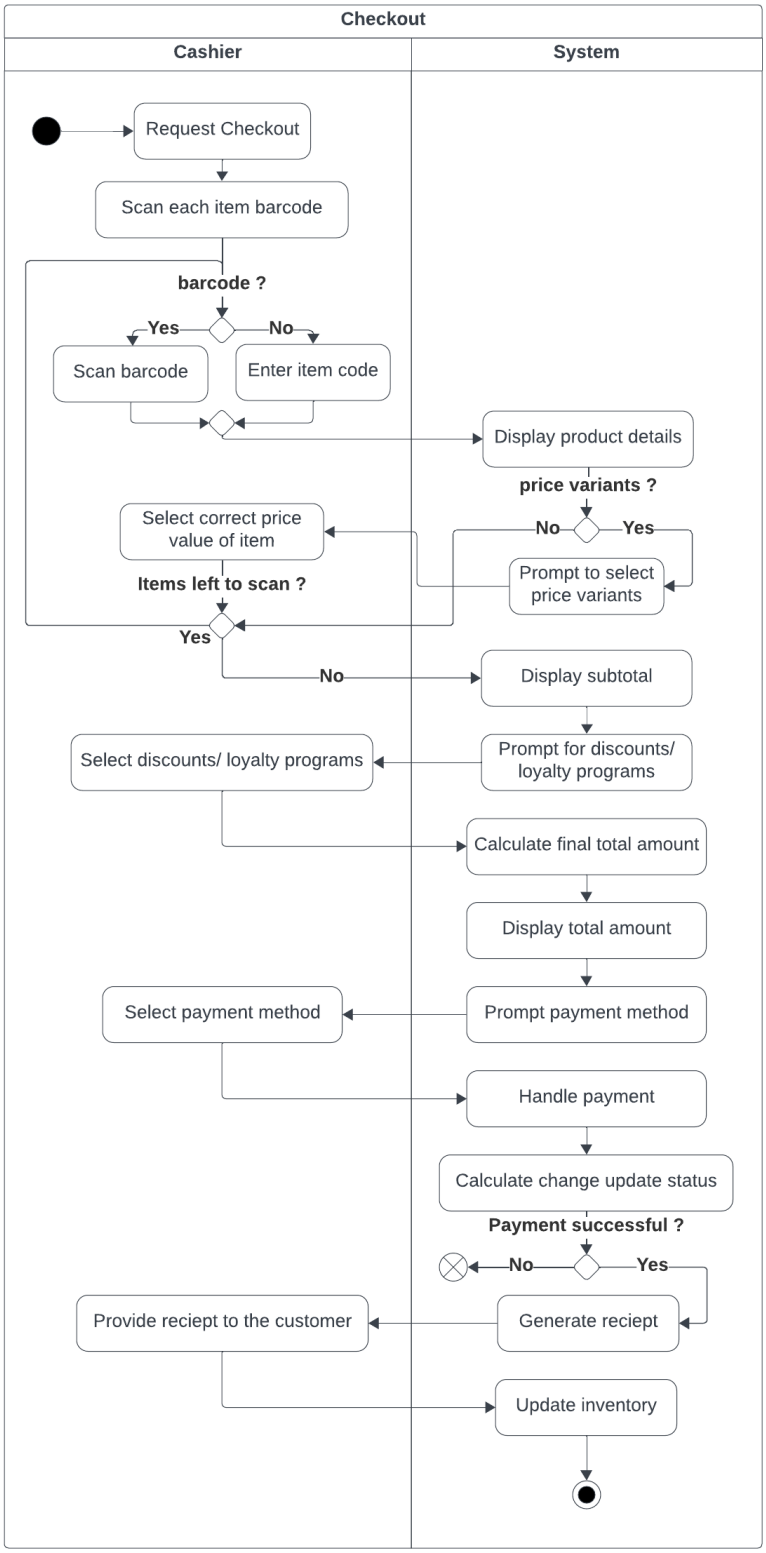

Figure 6-3

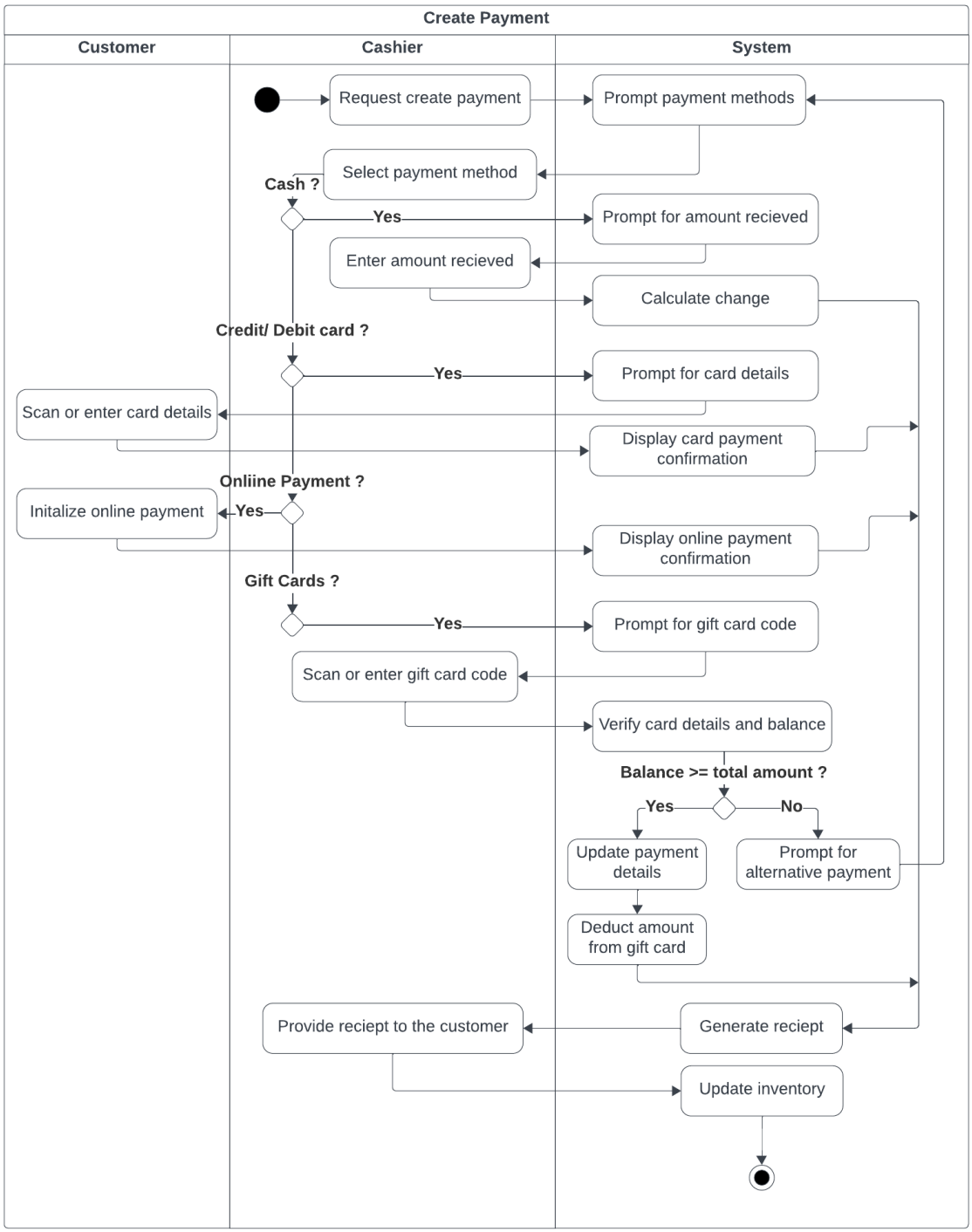
Create Payment
Customer
Cashier
System
Request create payment
Prompt payment methods
Select payment method
Cash ?
Yes
Prompt for amount recieved
Enter amount recieved
Calculate change
Credit/ Debit card ?
Yes
Prompt for card details
Scan or enter card details
Display card payment confirmation
Onliine Payment ?
Initalize online payment
Yes
Display online payment confirmation
Gift Cards ?
Yes
Prompt for gift card code
Scan or enter gift card code
Verify card details and balance
Balance >= total amount ?
Yes
No
Update payment details
Prompt for alternative payment
Deduct amount from gift card
Provide reciept to the customer
Generate reciept
Update inventory

Figure 6-4

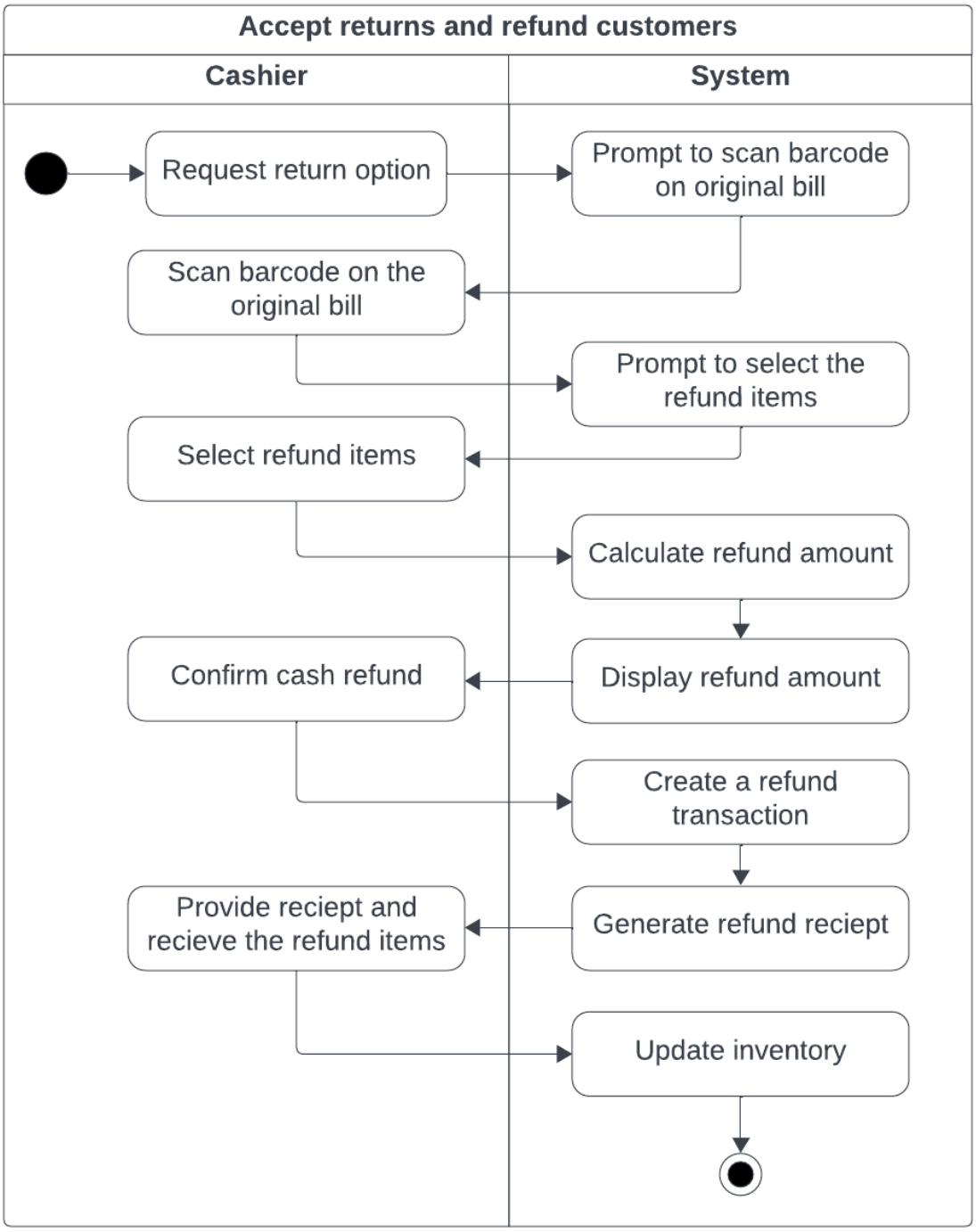
Accept returns and refund customers
Cashier
System
Request return option
Prompt to scan barcode on original bill
Scan barcode on the original bill
Prompt to select the refund items
Select refund items
Calculate refund amount
Confirm cash refund
Display refund amount
Create a refund transaction
Provide reciept and recieve the refund items
Generate refund reciept
Update inventory

Figure 6-5

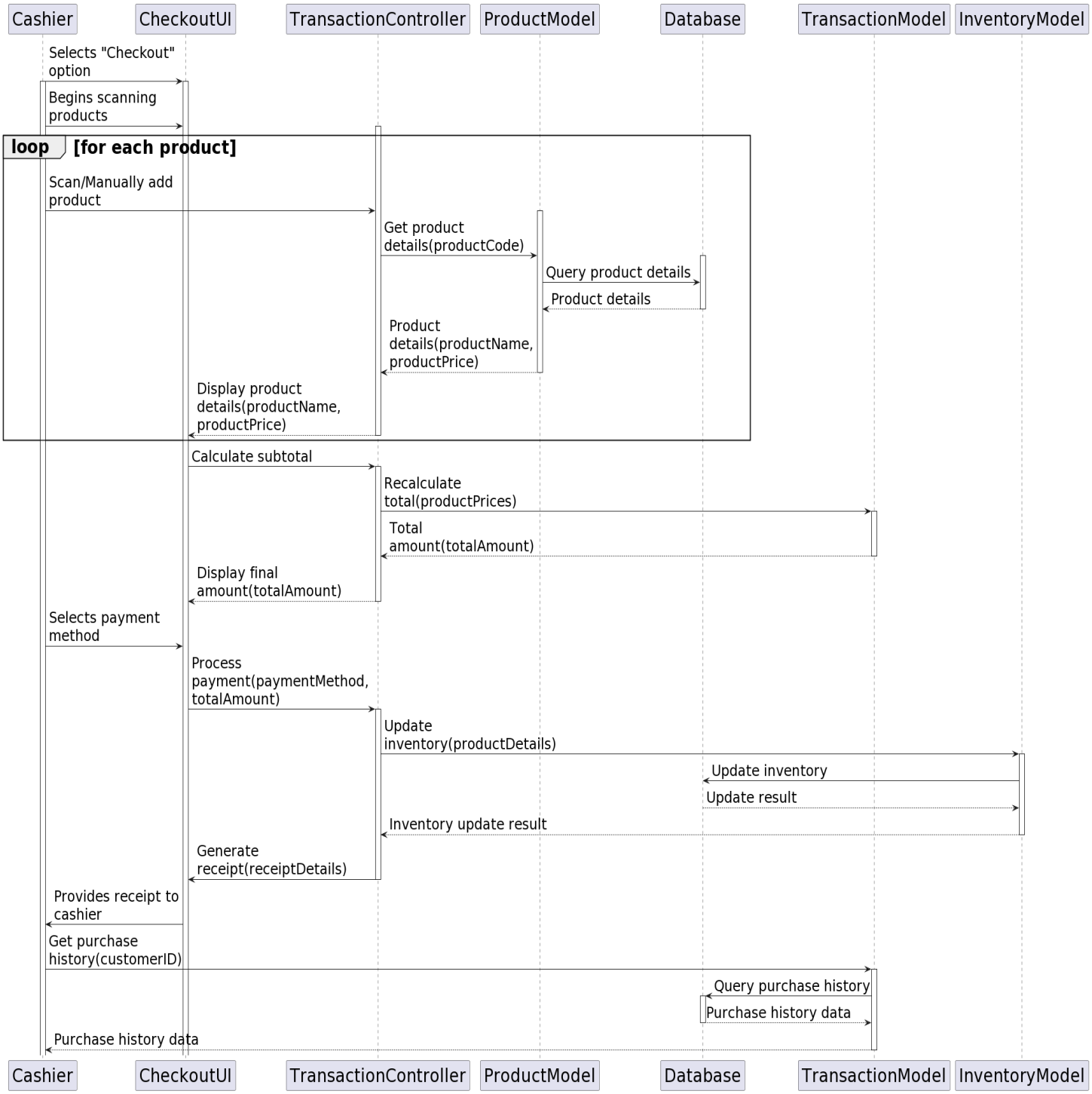

Figure 6-6 Checkout

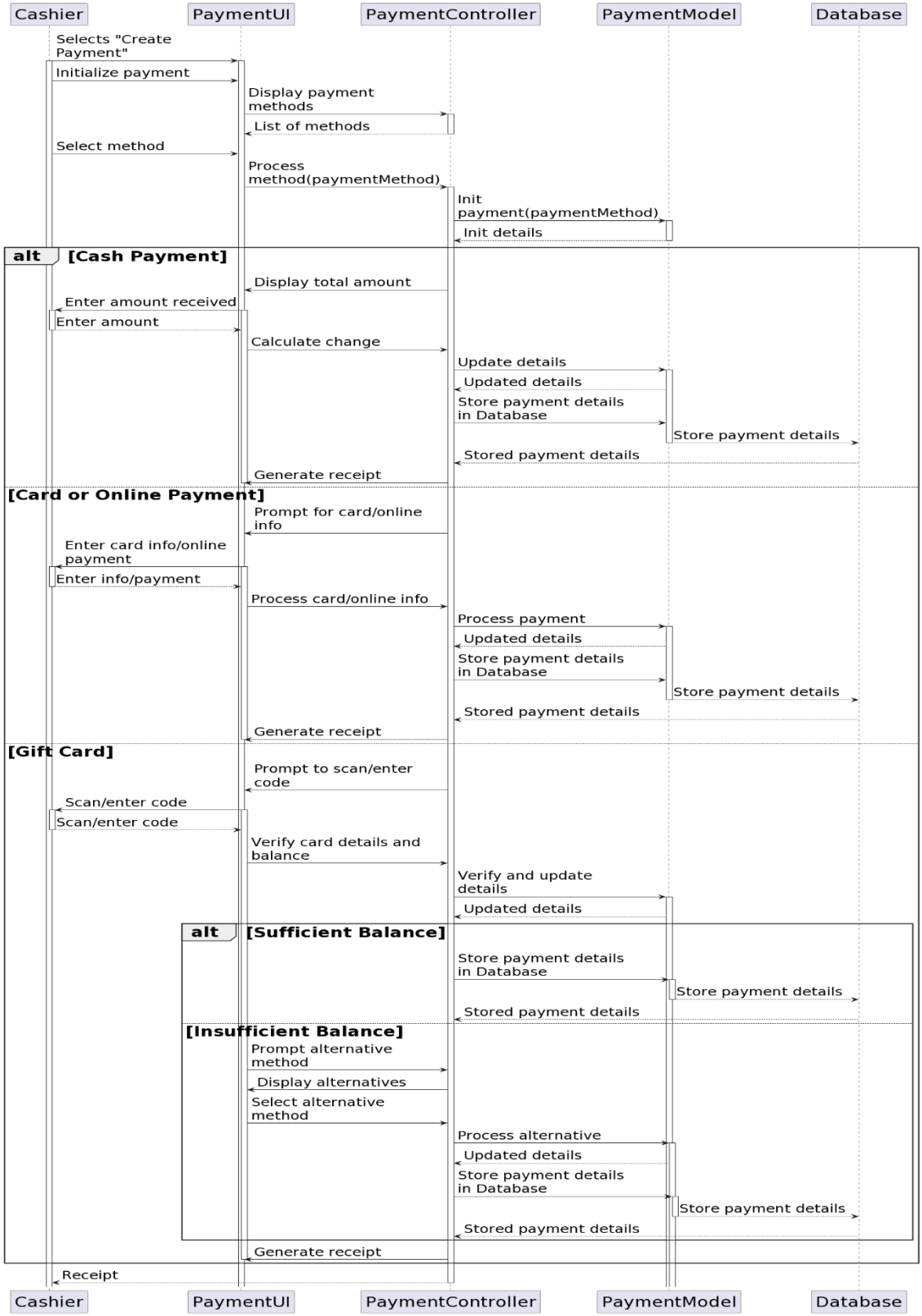

Figure 6-7 Payment

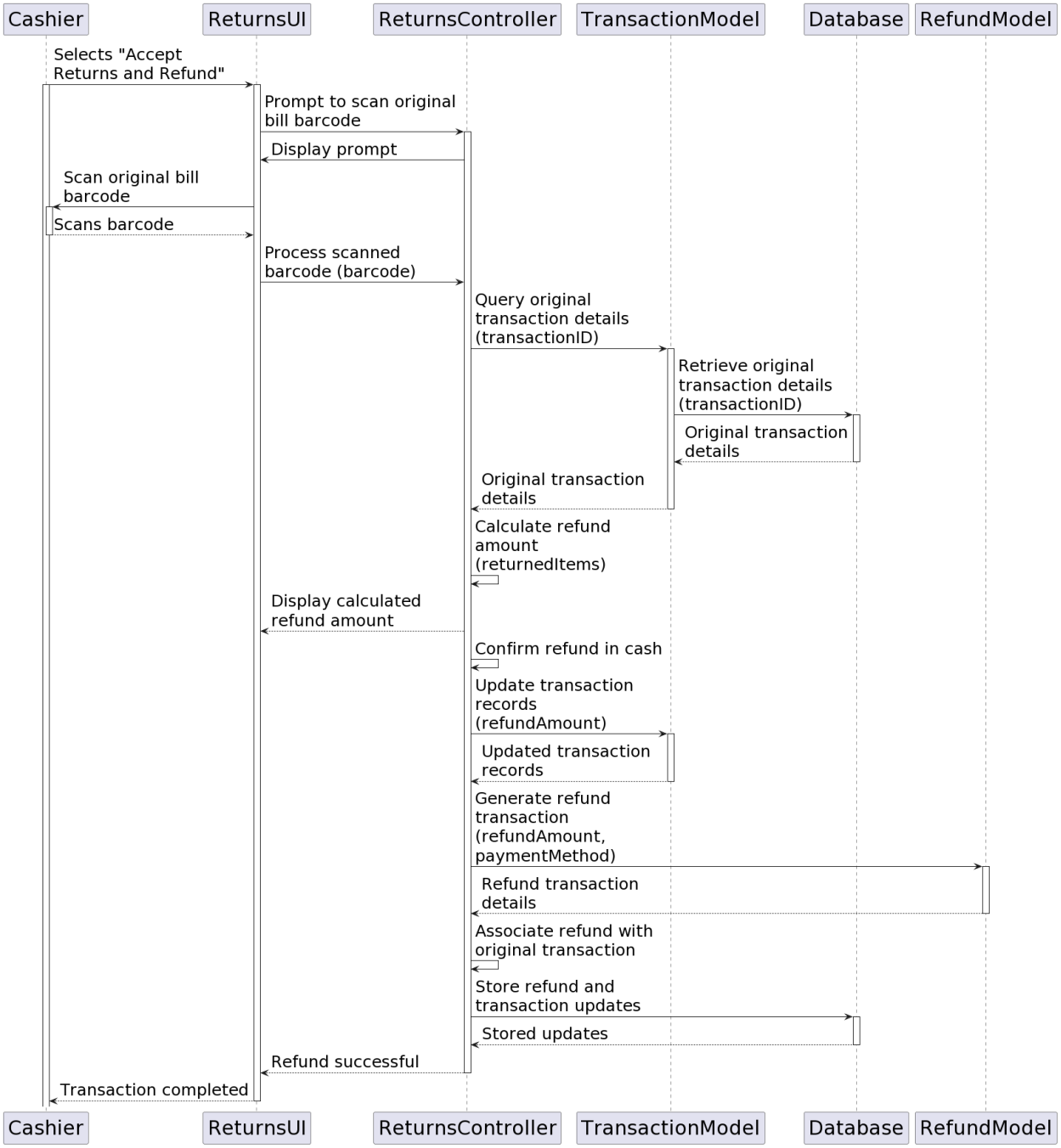

Figure 6-8 Accept Returns and Refunds

6.2 Inventory Management Subsystem

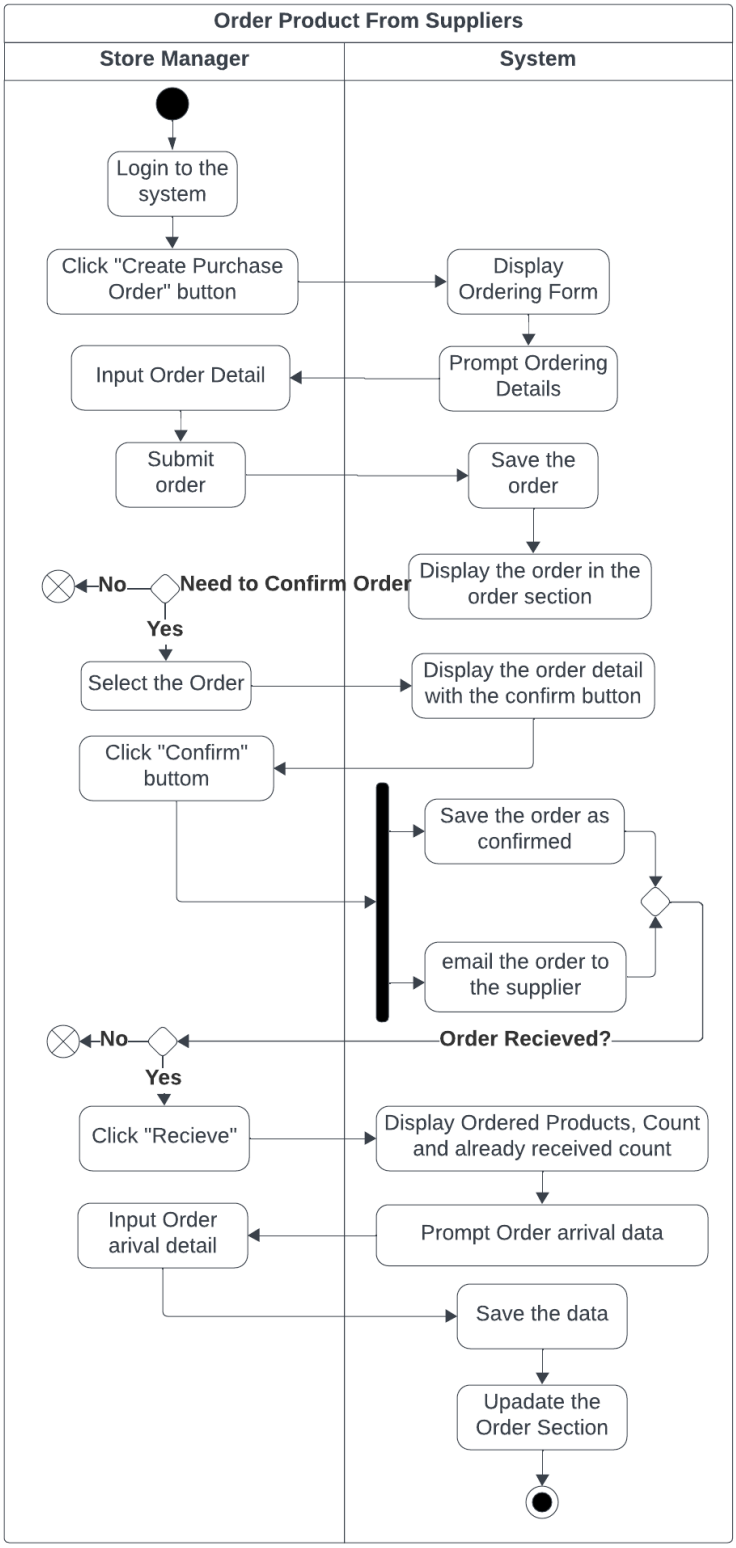

Figure 6-9

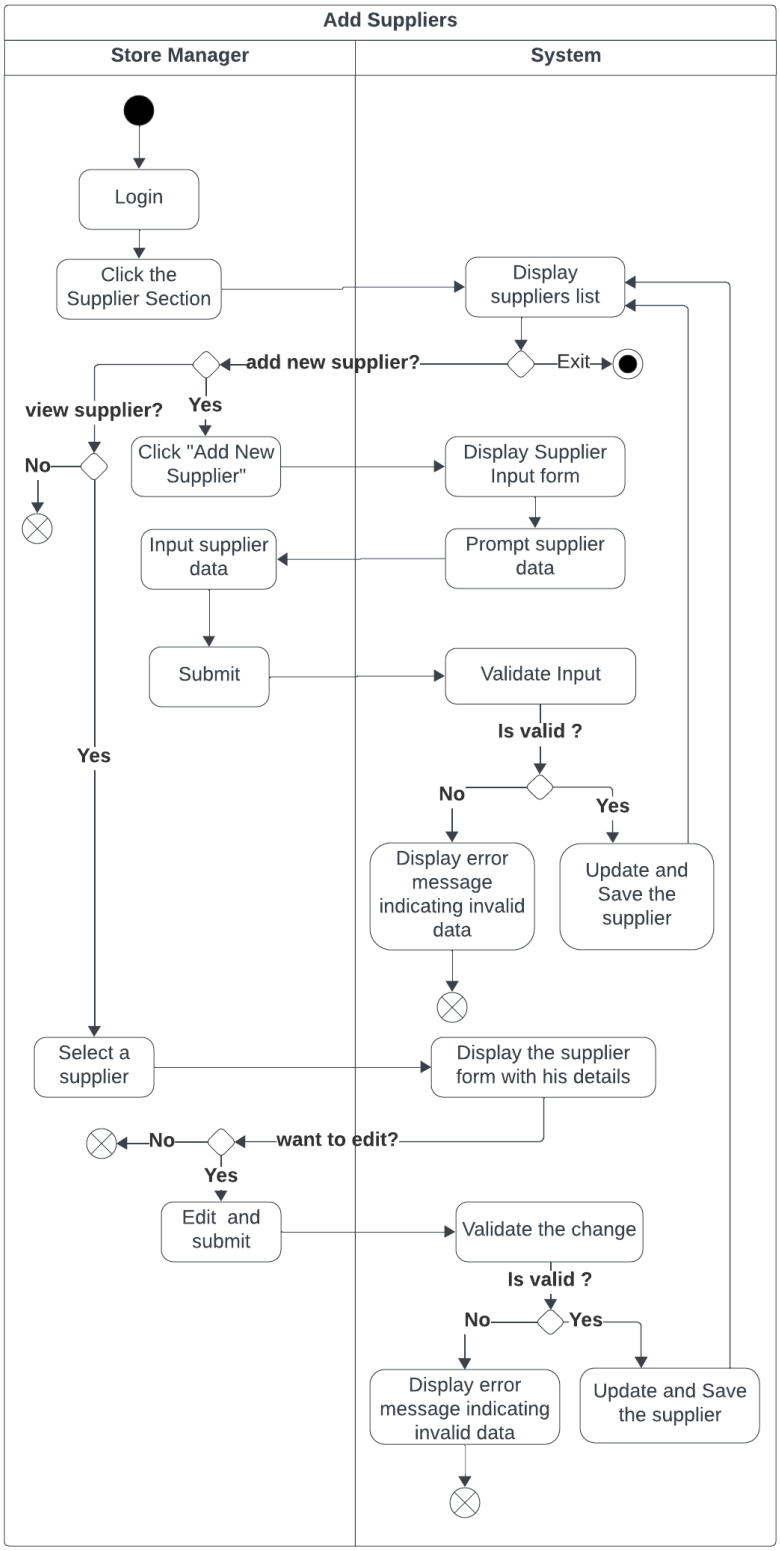
Add Suppliers
Store Manager
System
Login
Click the Supplier Section
Display suppliers list
add new supplier?
Exit
view supplier?
Yes
No
Click "Add New Supplier"
Display Supplier Input form
Input supplier data
Prompt supplier data
Submit
Validate Input
Is valid ?
Yes
No
Yes
Display error message indicating invalid data
Update and Save the supplier
Select a supplier
Display the supplier form with his details
No
want to edit?
Yes
Edit and submit
Validate the change
Is valid ?
No
Yes
Display error message indicating invalid data
Update and Save the supplier

Figure 6-10

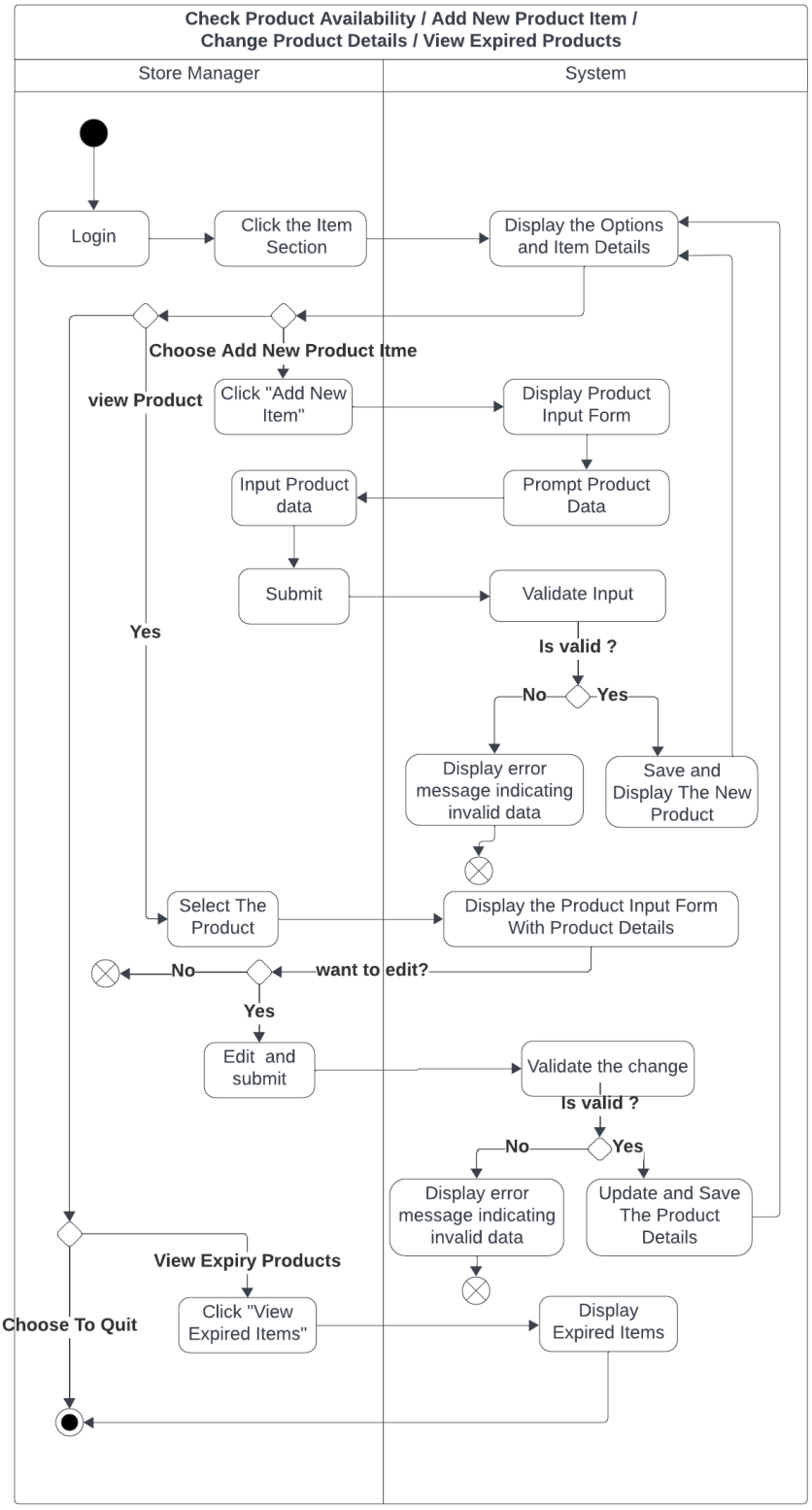
Check Product Availability / Add New Product Item /
Change Product Details / View Expired Products
Store Manager
System
Login
Click the Item Section
Display the Options and Item Details
Choose Add New Product Itme
view Product
Click "Add New Item"
Display Product Input Form
Input Product data
Prompt Product Data
Submit
Validate Input
Yes
Is valid ?
No
Yes
Display error message indicating invalid data
Save and Display The New Product
Select The Product
Display the Product Input Form With Product Details
No
want to edit?
Yes
Edit and submit
Validate the change
Is valid ?
No
Yes
Display error message indicating invalid data
Update and Save The Product Details
View Expiry Products
Choose To Quit
Click "View Expired Items"
Display Expired Items

Figure 6-11

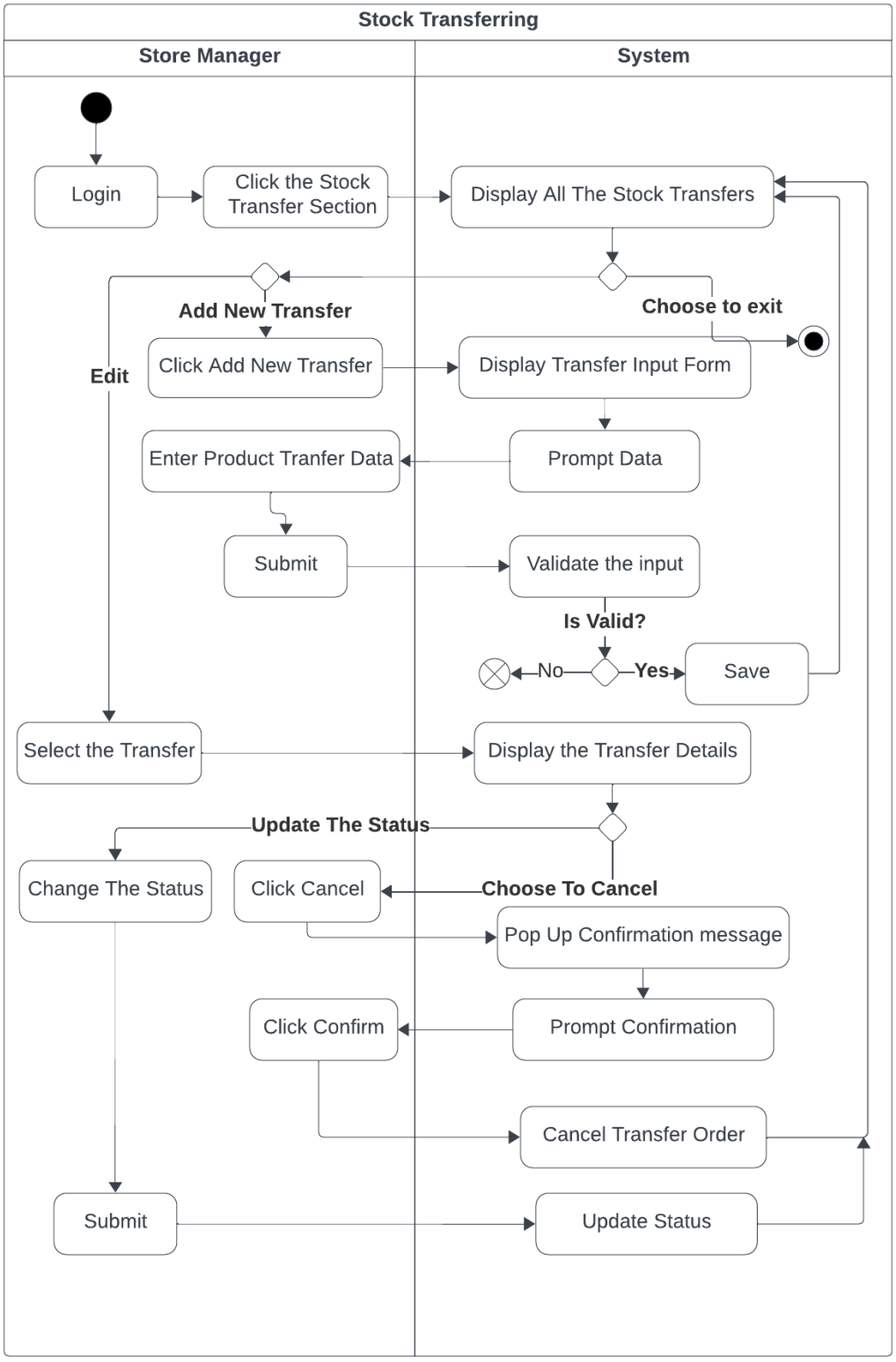
Stock Transferring
Store Manager
System
Login
Click the Stock Transfer Section
Display All The Stock Transfers
Add New Transfer
Choose to exit
Edit
Click Add New Transfer
Display Transfer Input Form
Enter Product Tranfer Data
Prompt Data
Submit
Validate the input
Is Valid?
No
Yes
Save
Select the Transfer
Display the Transfer Details
Update The Status
Change The Status
Click Cancel
Choose To Cancel
Pop Up Confirmation message
Click Confirm
Prompt Confirmation
Cancel Transfer Order
Submit
Update Status

Figure 6-12

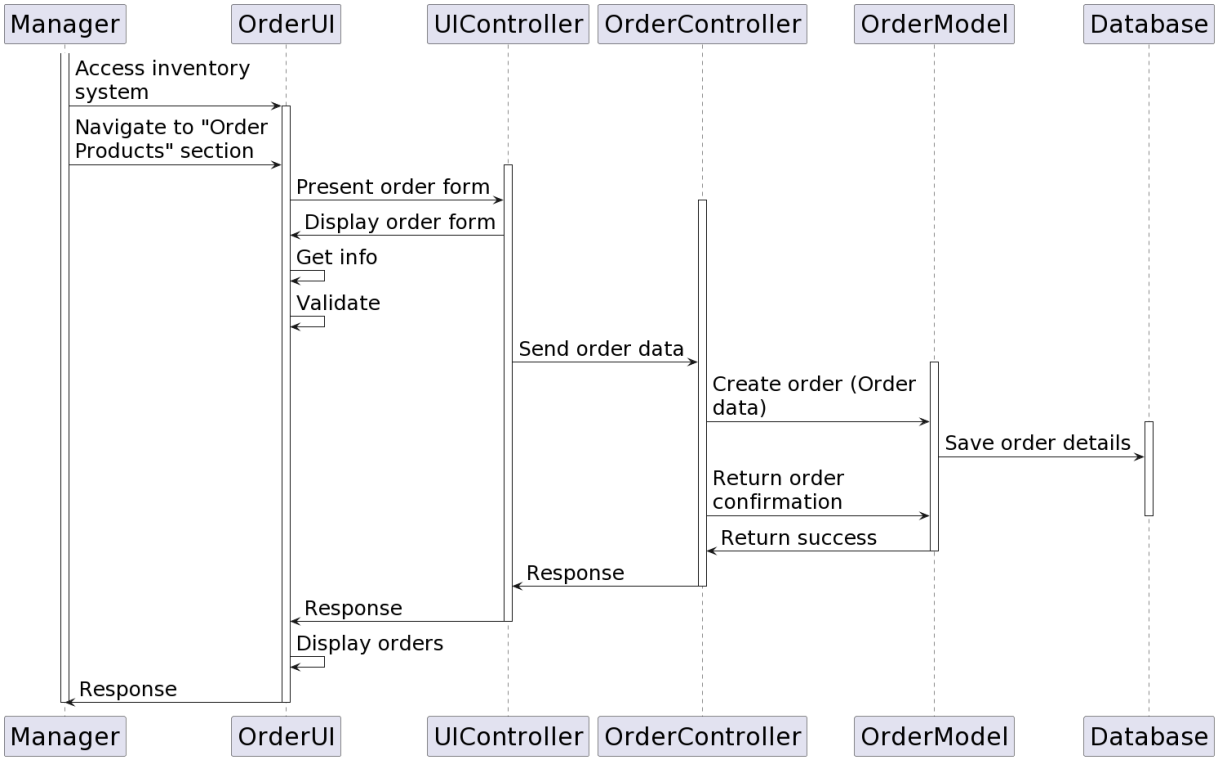

Figure 6-13

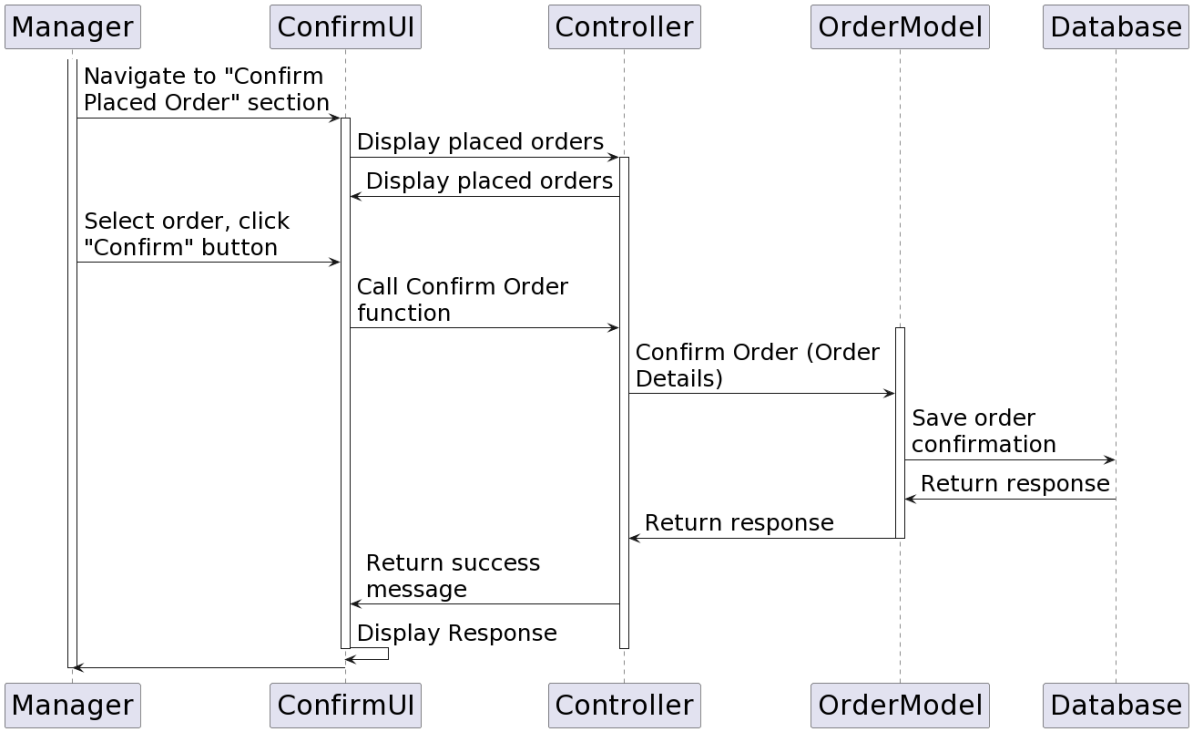

Figure 6-14

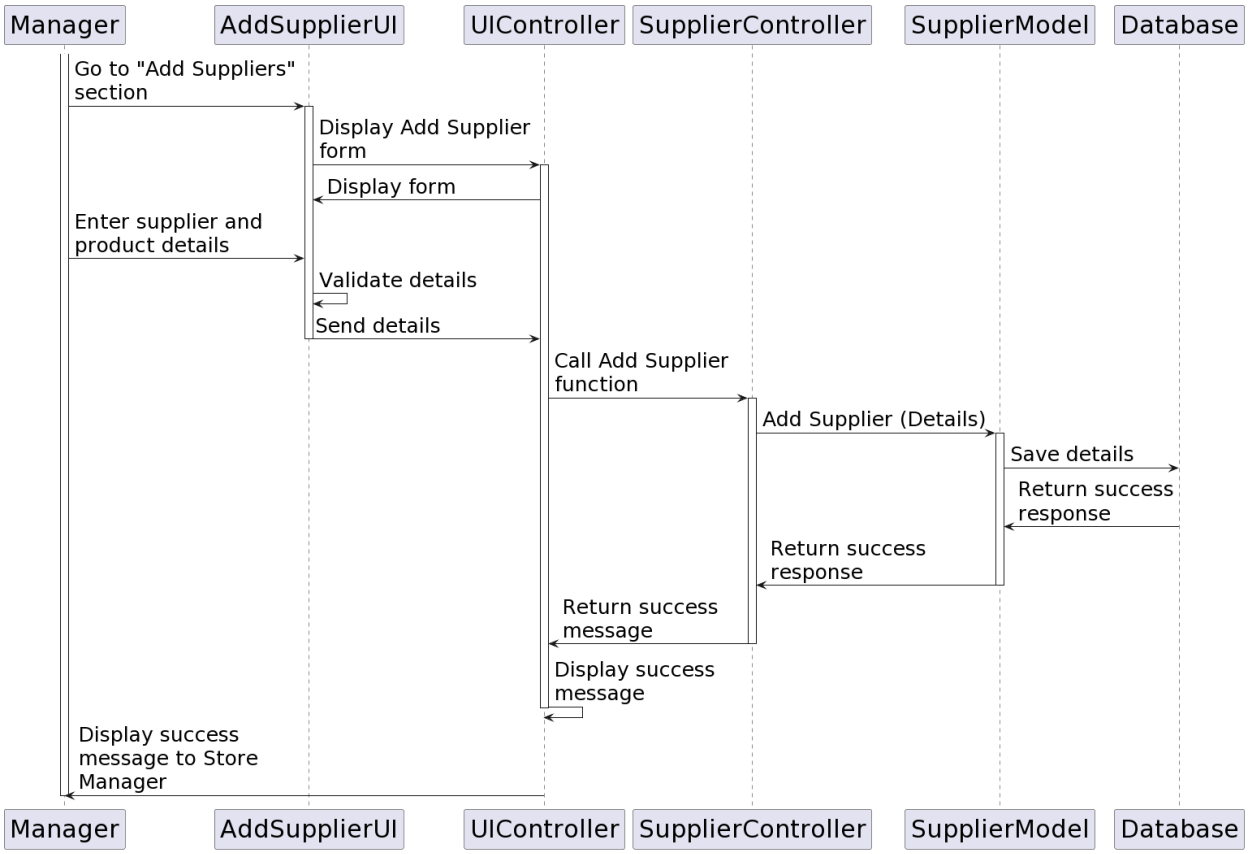
Manager
AddSupplierUI
UIController
SupplierController
SupplierModel
Database
Go to "Add Suppliers" section
Display Add Supplier form
Display form
Enter supplier and product details
Validate details
Send details
Call Add Supplier function
Add Supplier (Details)
Save details
Return success response
Return success response
Return success message
Display success message
Display success message to Store Manager
Manager
AddSupplierUI
UIController
SupplierController
SupplierModel
Database

Figure 6-15

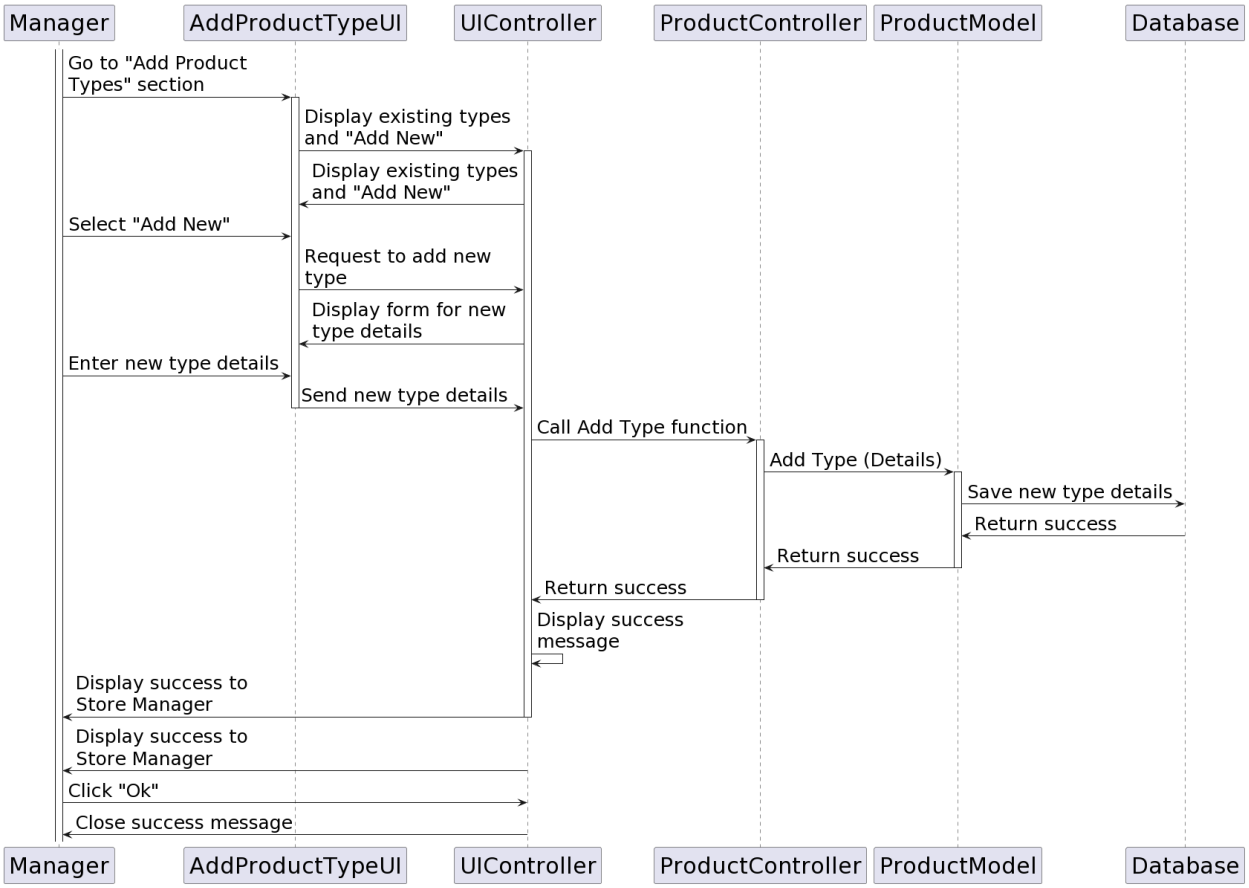

Figure 6-16

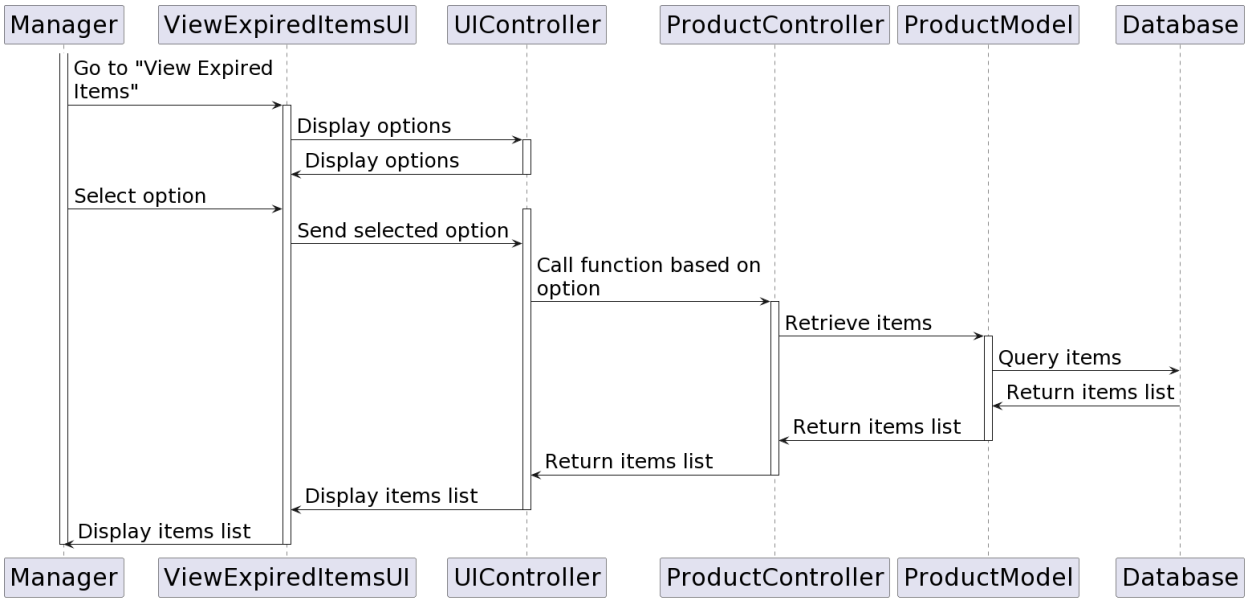

Figure 6-17

6.3 Employee Management Subsystem

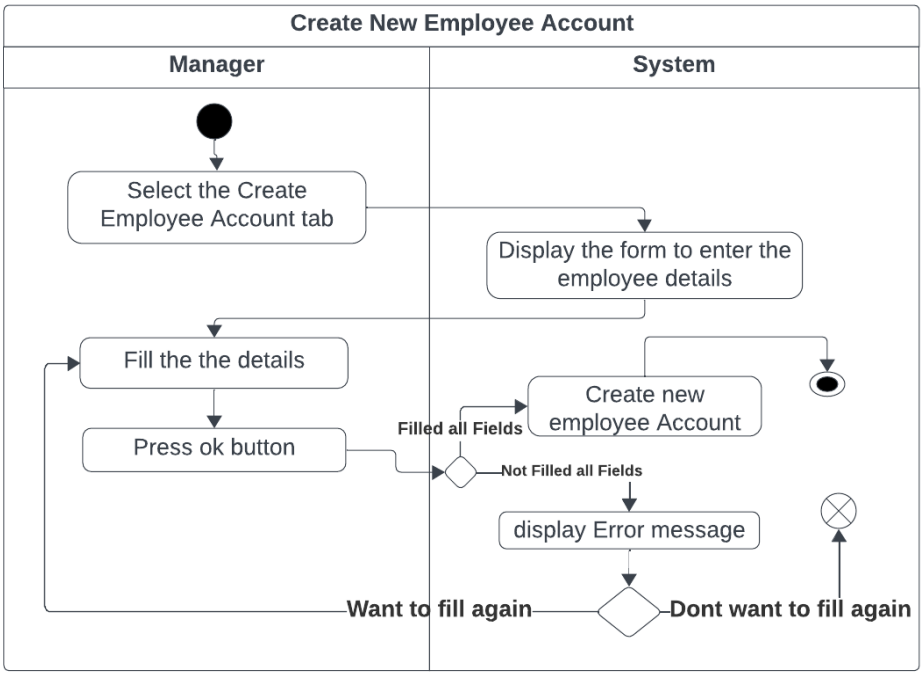

Figure 6-18

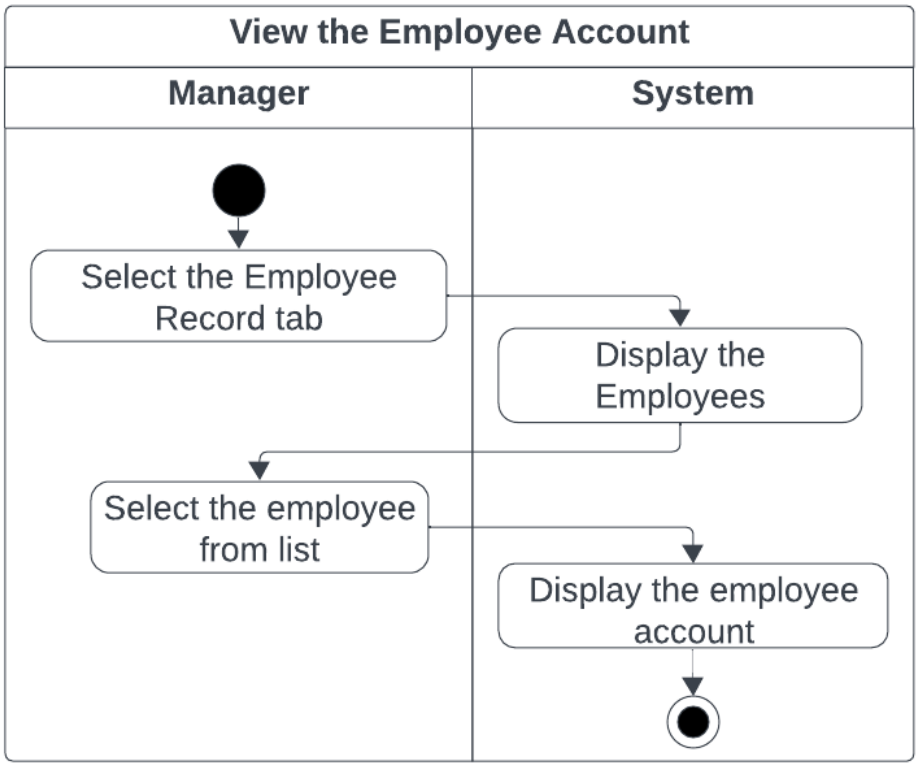

Figure 6-19

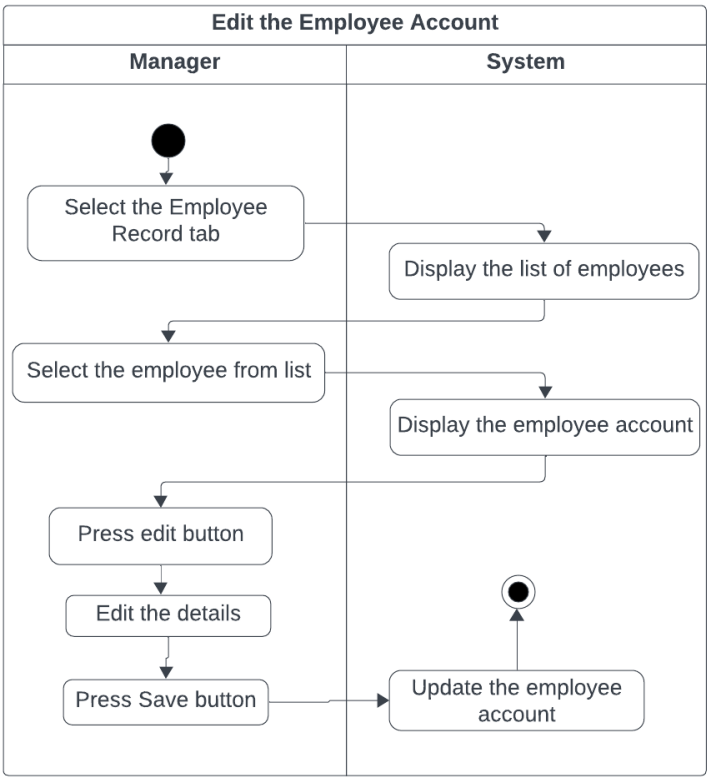

Figure 6-20

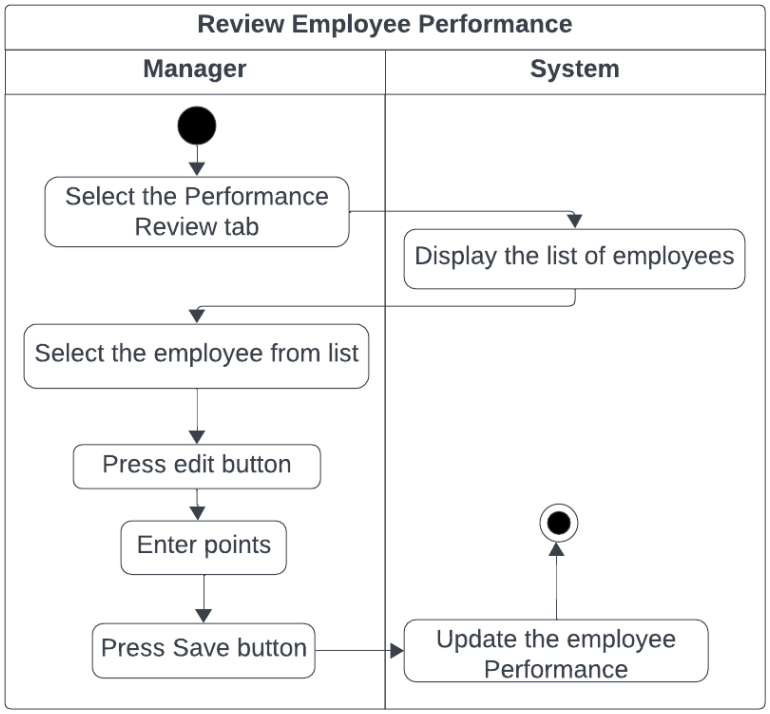

Figure 6-21

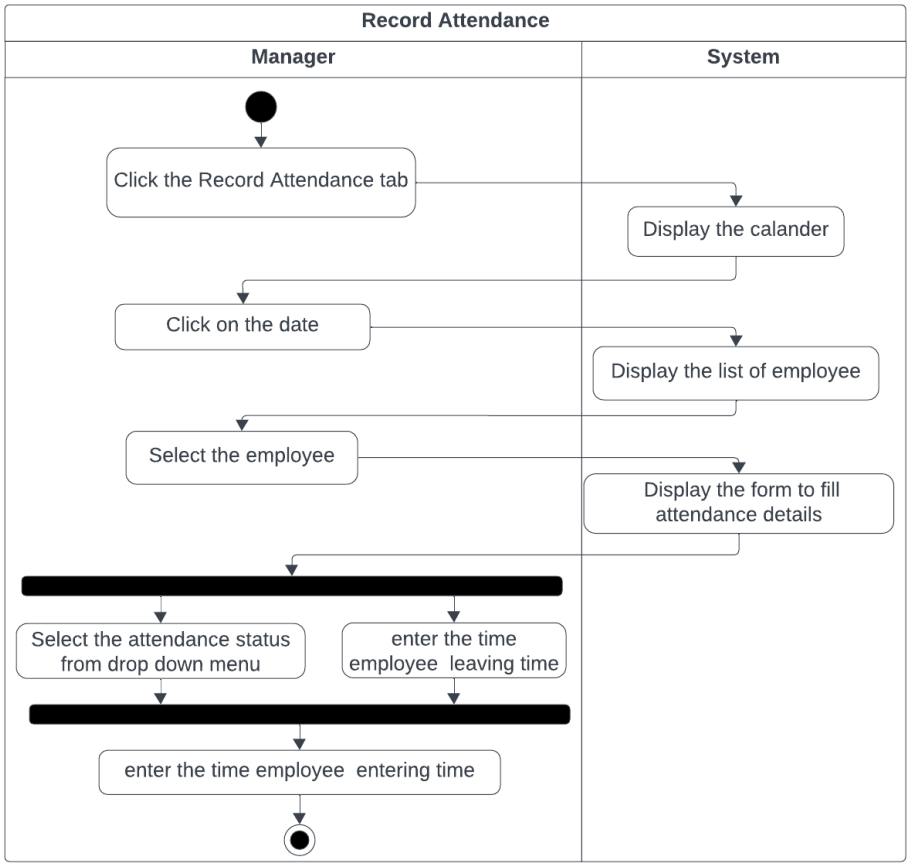
Record Attendance
Manager
System
Click the Record Attendance tab
Display the calander
Click on the date
Display the list of employee
Select the employee
Display the form to fill attendance details
Select the attendance status from drop down menu
enter the time employee leaving time
enter the time employee entering time

Figure 6-22

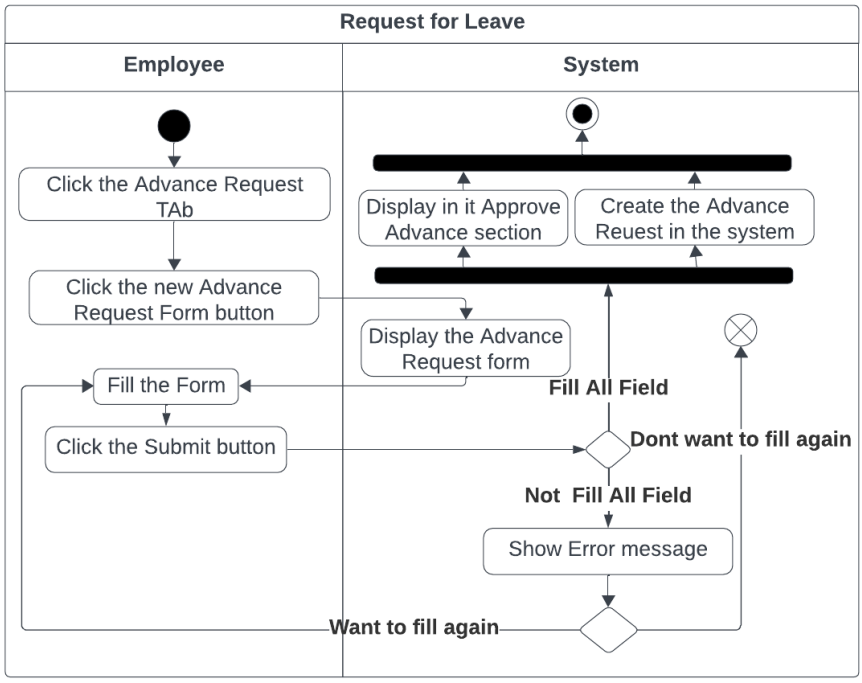

Figure 6-23

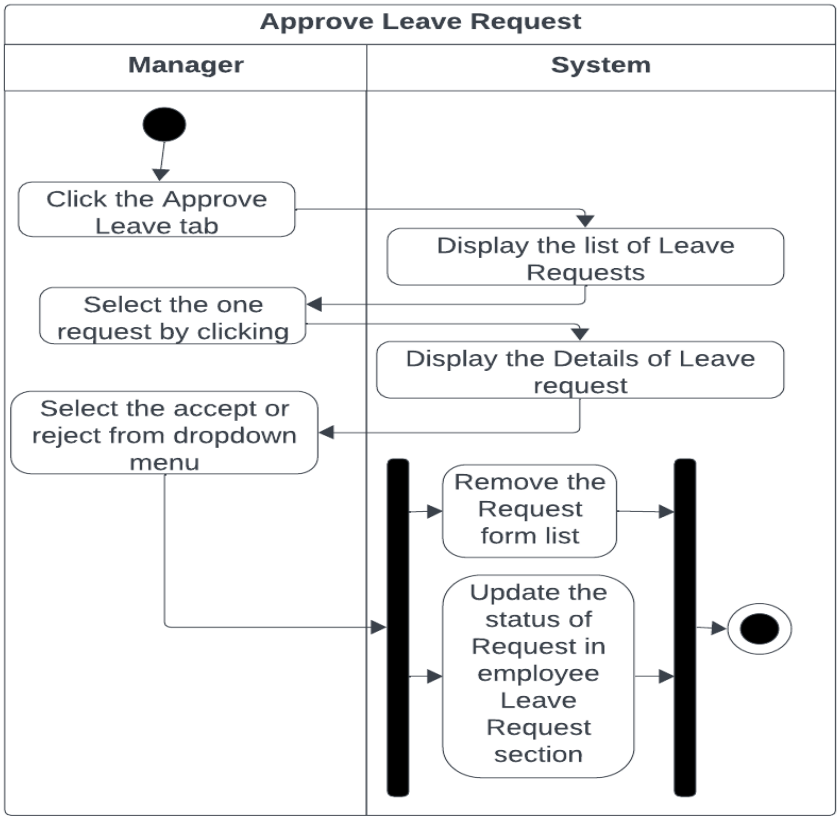

Figure 6-24

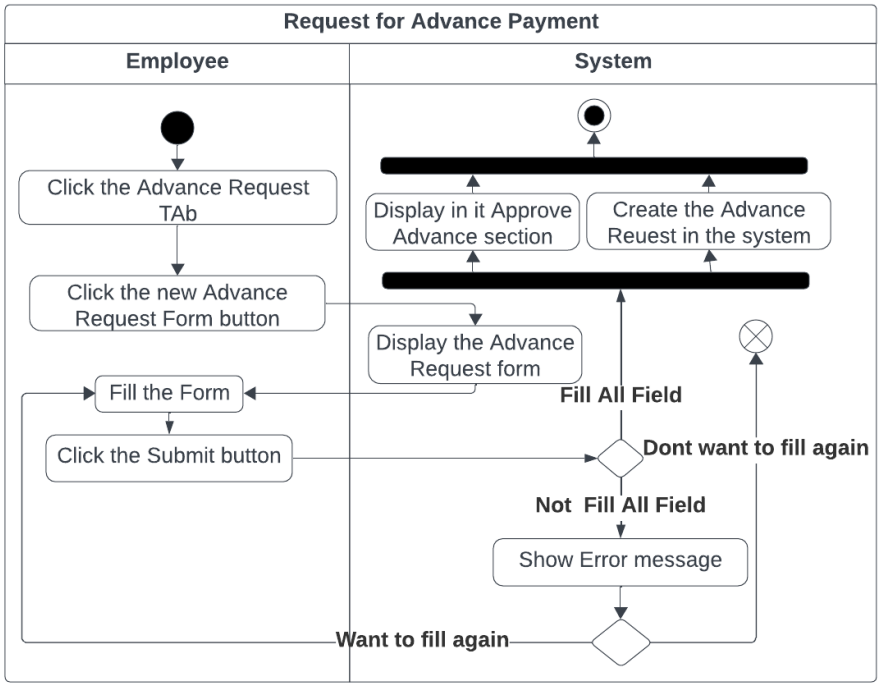
Request for Advance Payment
Employee
System
Click the Advance Request TAb
Click the new Advance Request Form button
Display the Advance Request form
Fill the Form
Click the Submit button
Display in it Approve Advance section
Create the Advance Reuest in the system
Fill All Field
Dont want to fill again
Not Fill All Field
Show Error message
Want to fill again

Figure 6-25

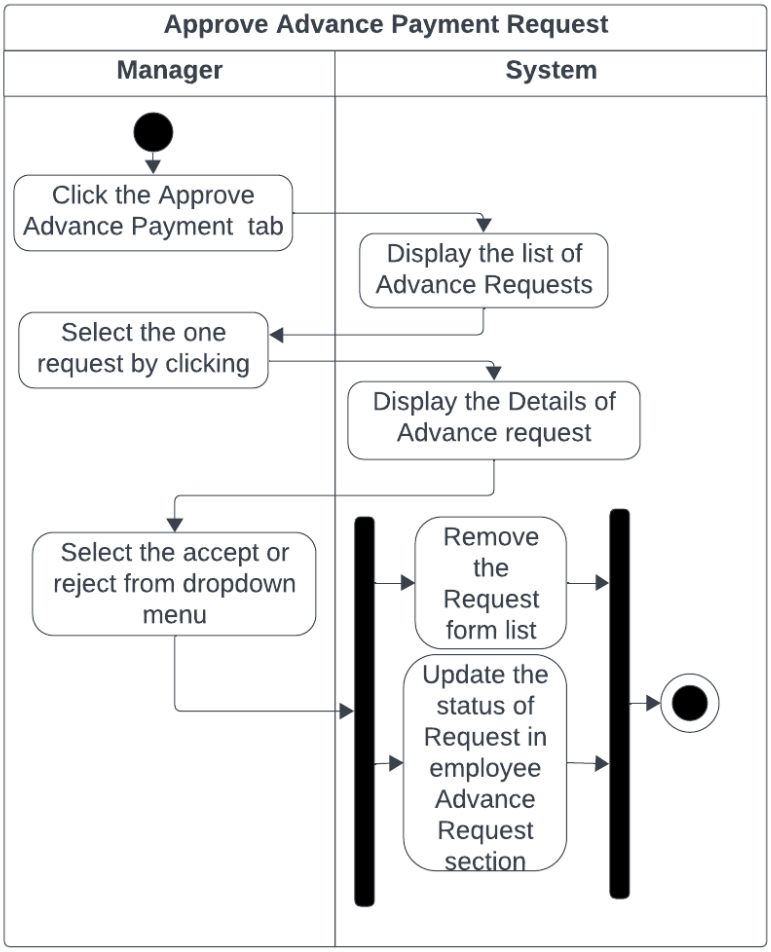

Figure 6-26

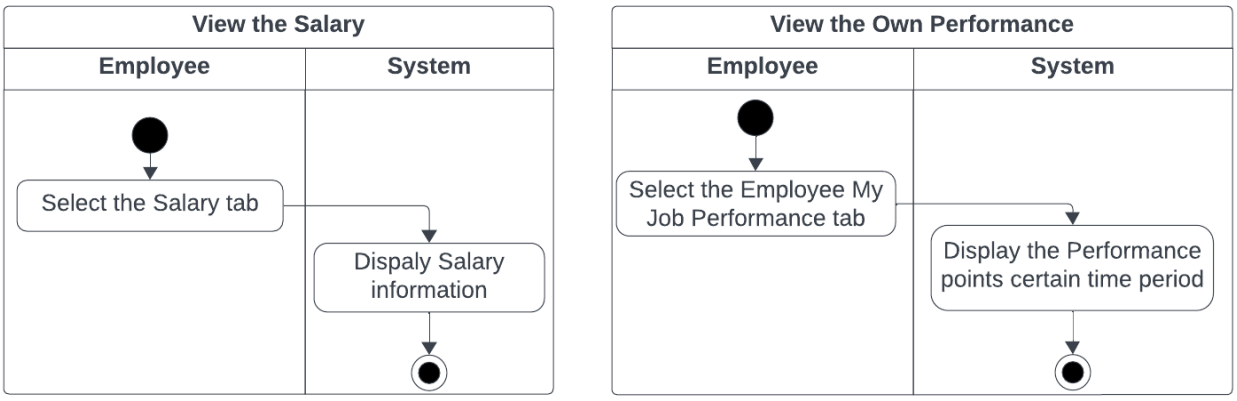

Figure 6-27

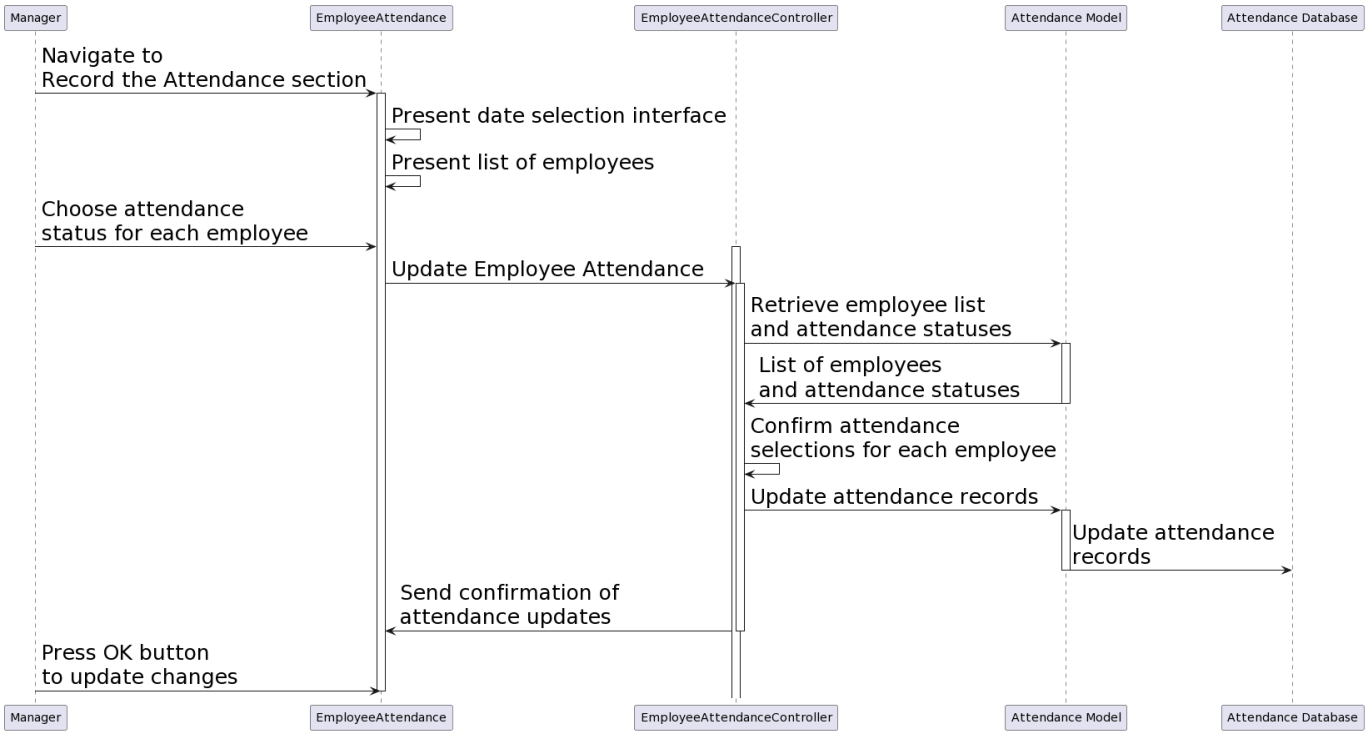

Figure 6-28

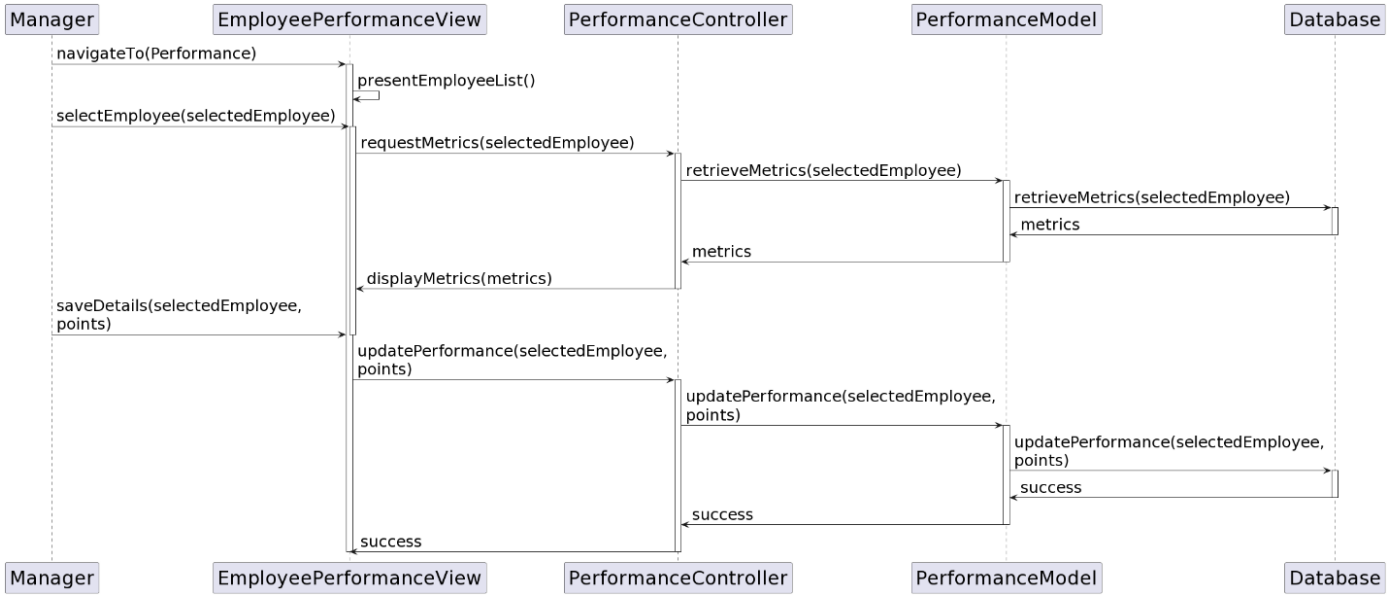

Figure 6-29

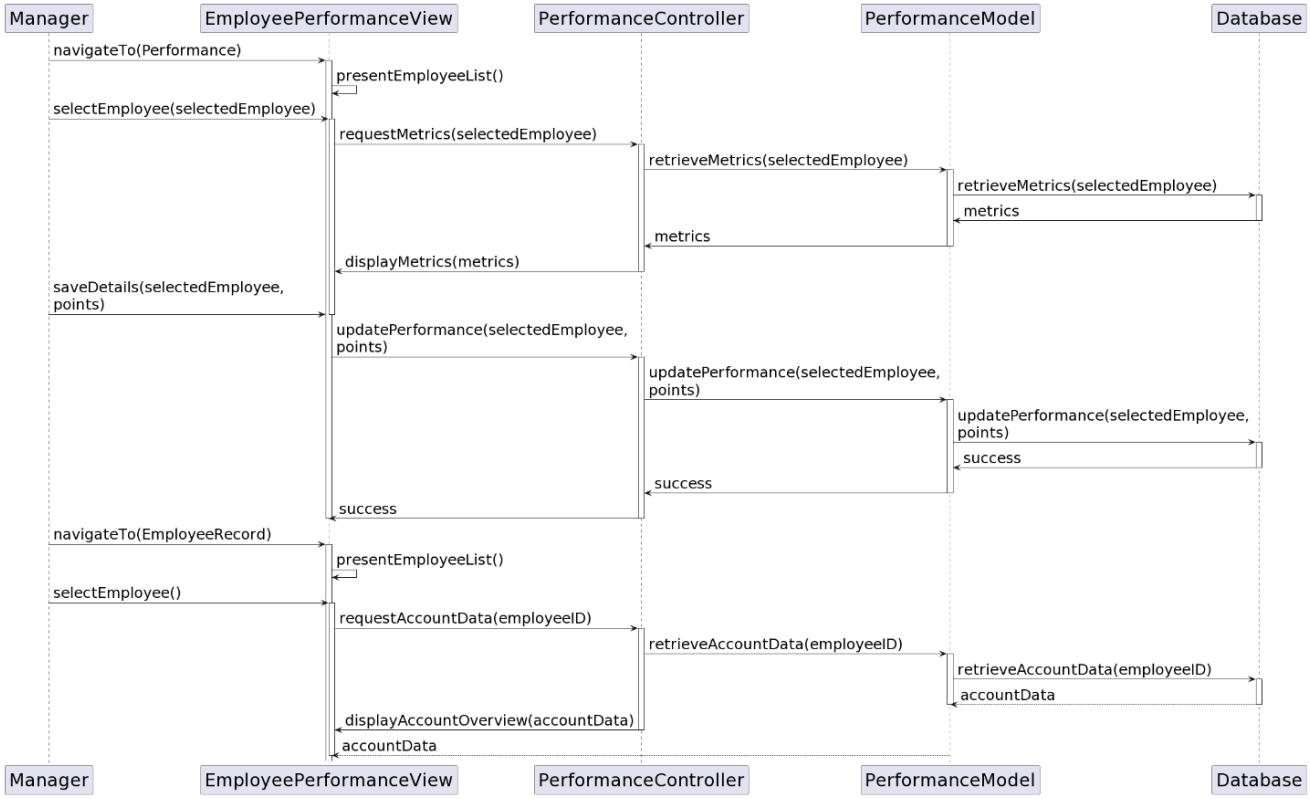

Figure 6-30

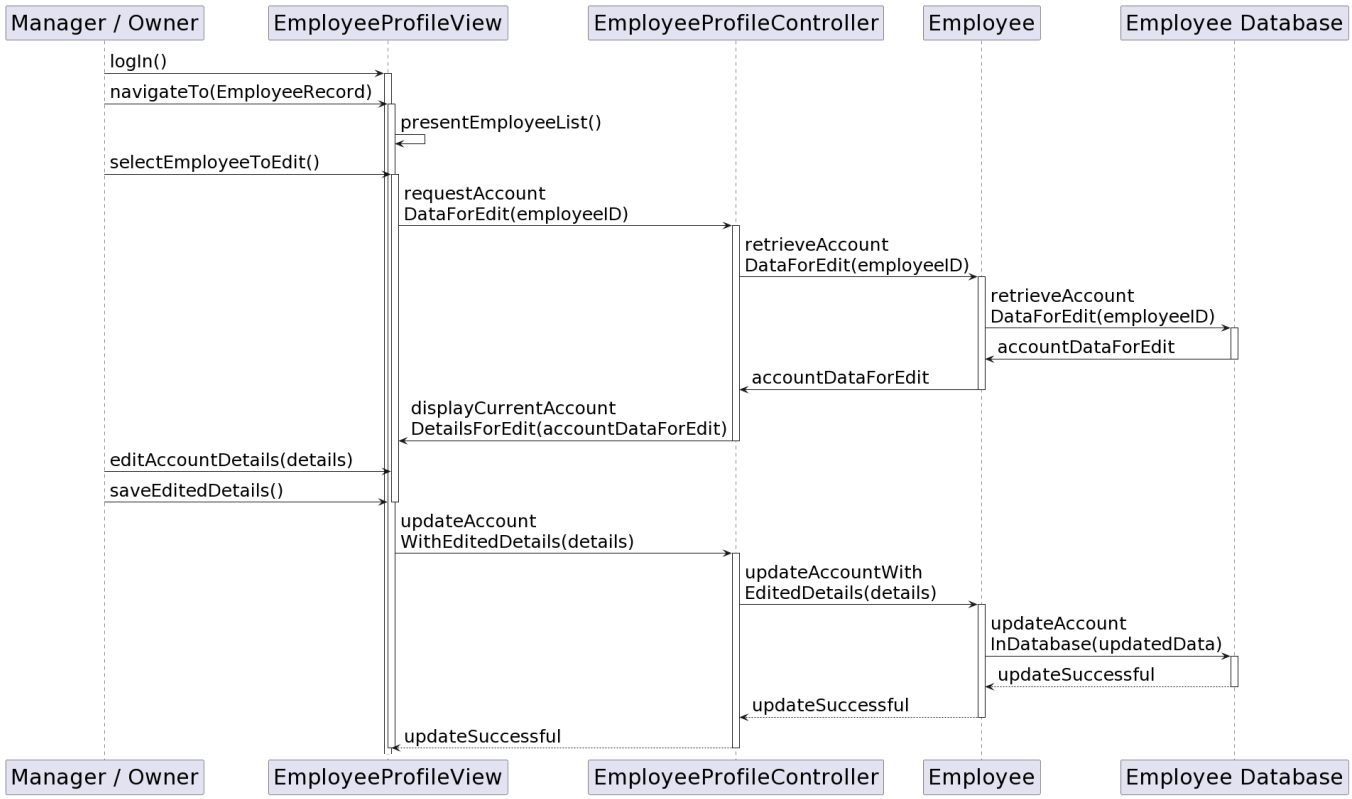

Figure 6-31

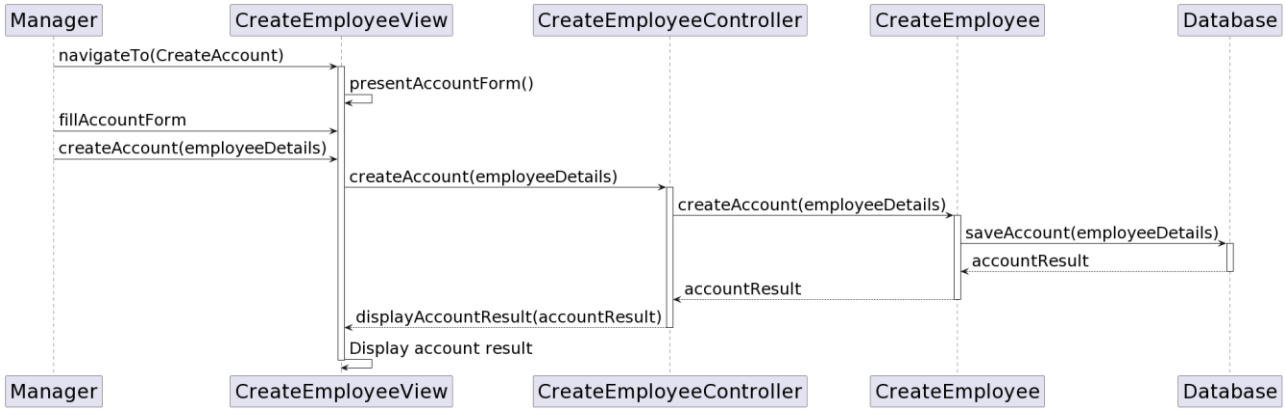

Figure 6-32

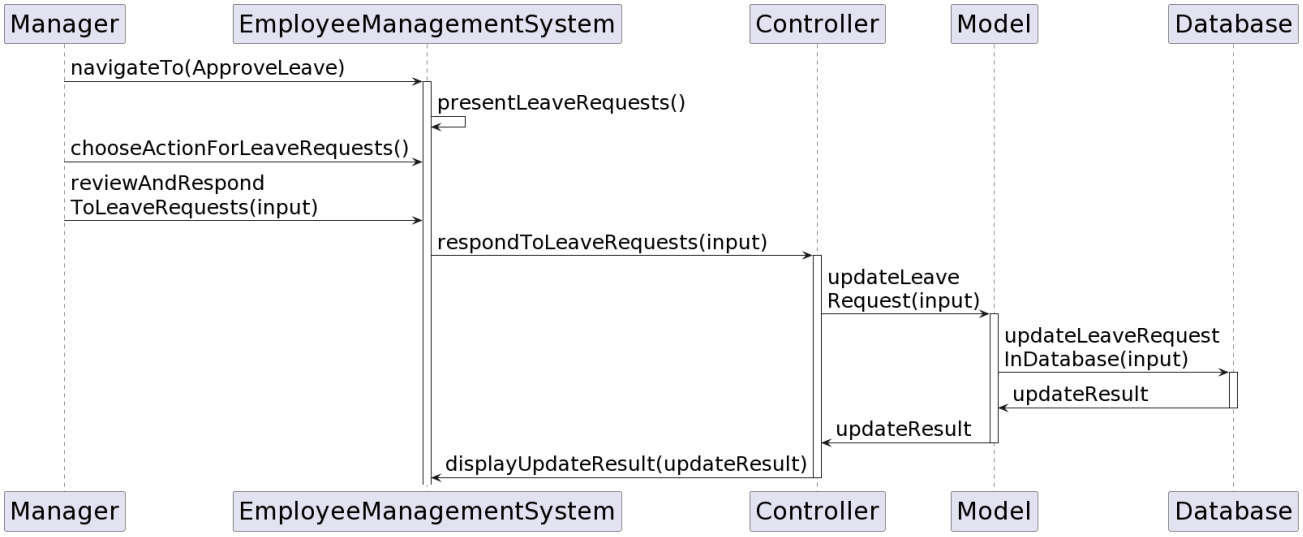

Figure 6-33

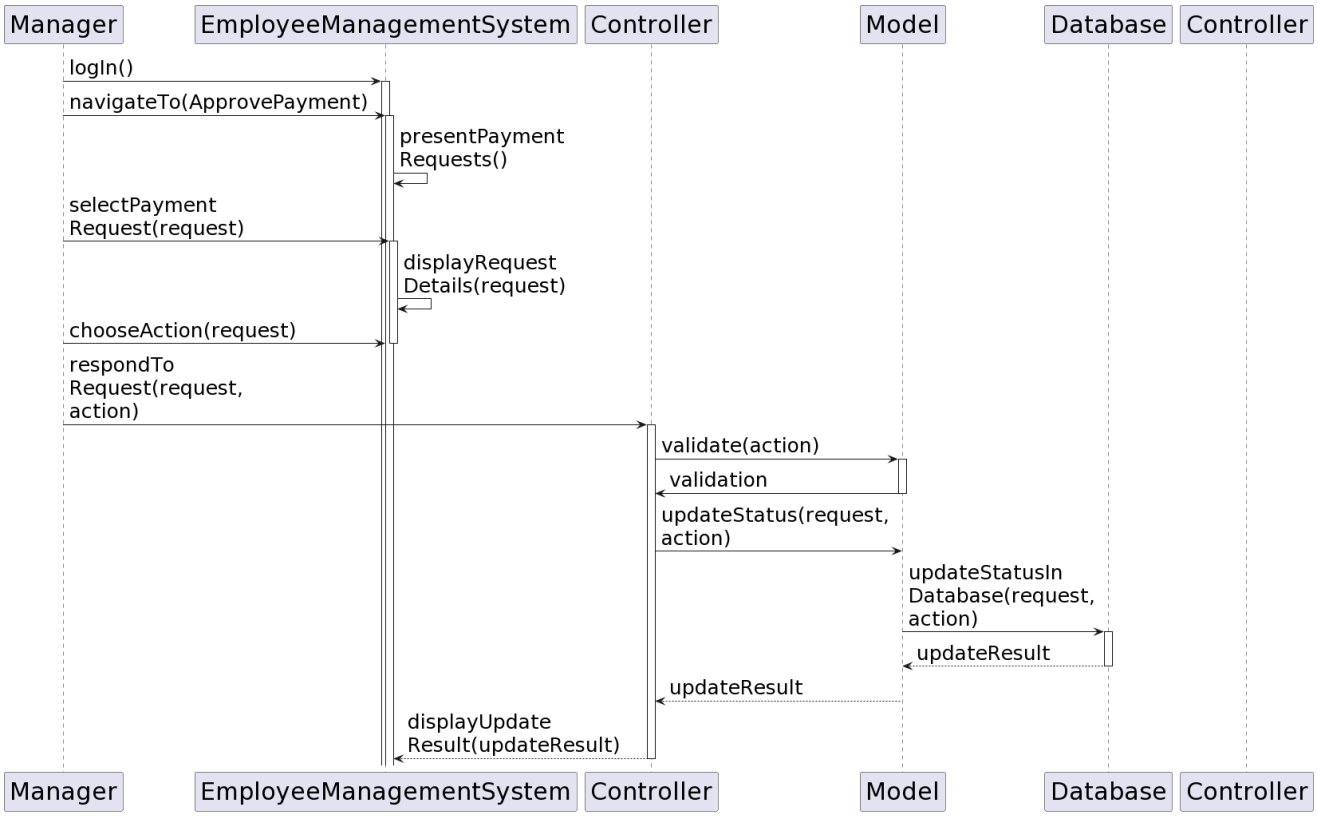

Figure 6-34

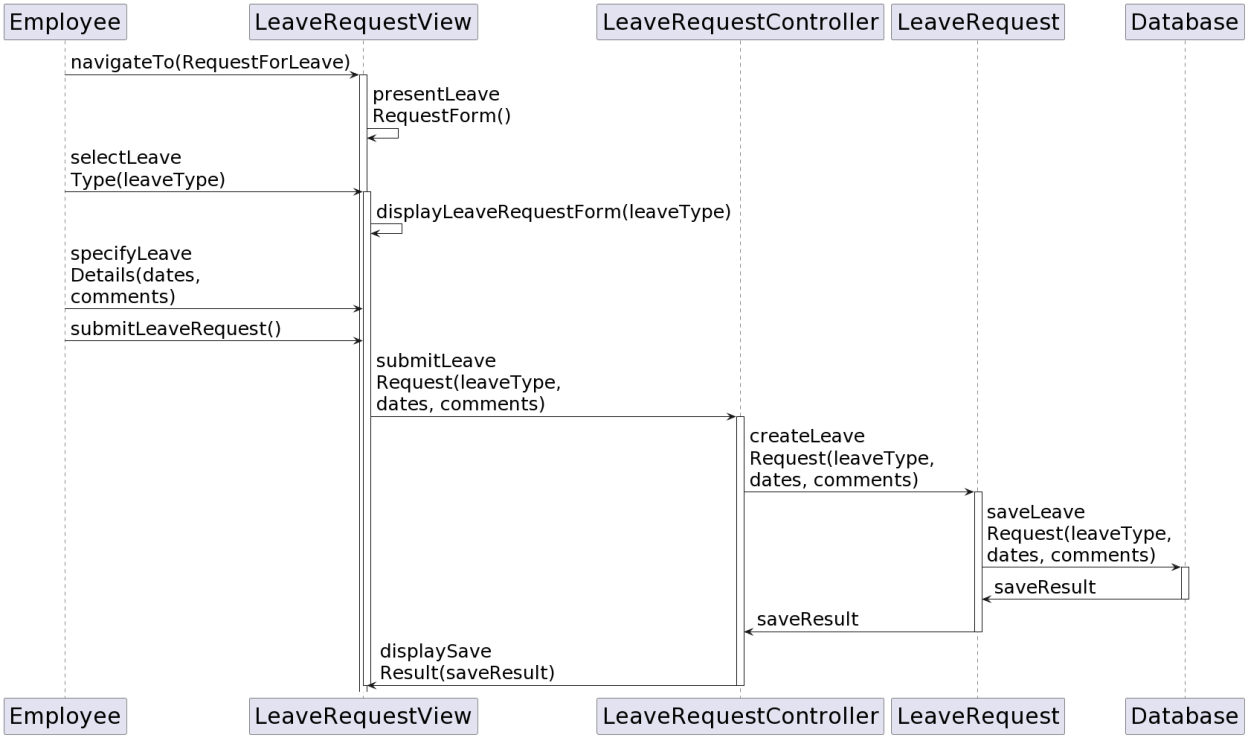

Figure 6-35

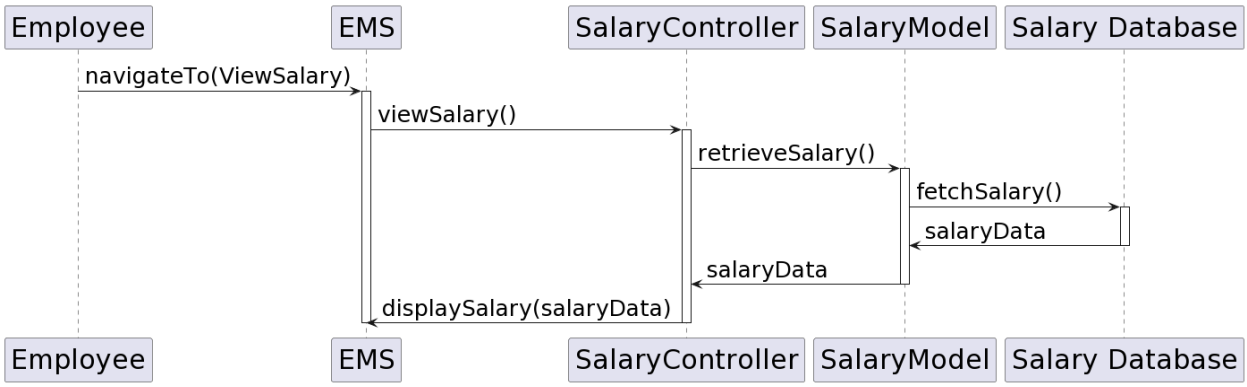

Figure 6-36

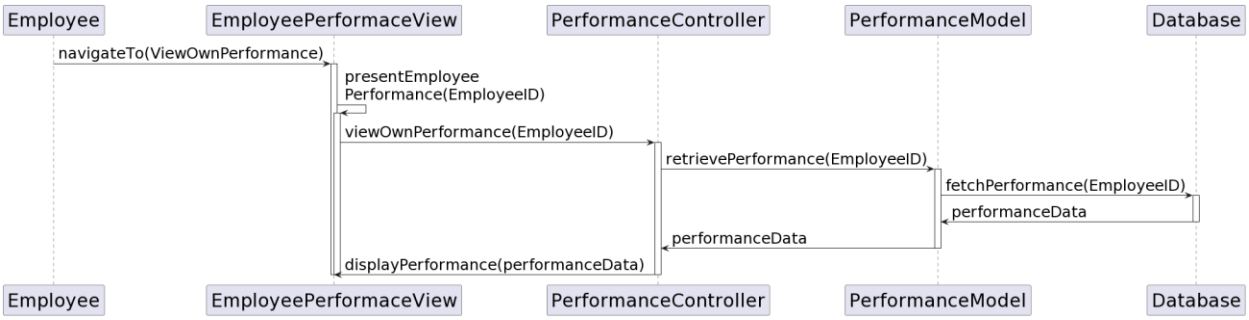

Figure 6-37

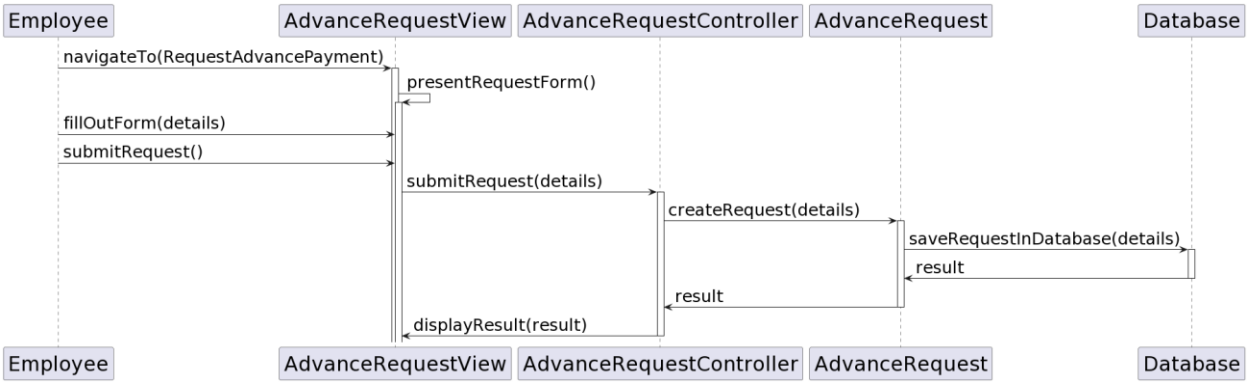

Figure 6-38

6.4 Customer Management Subsystem

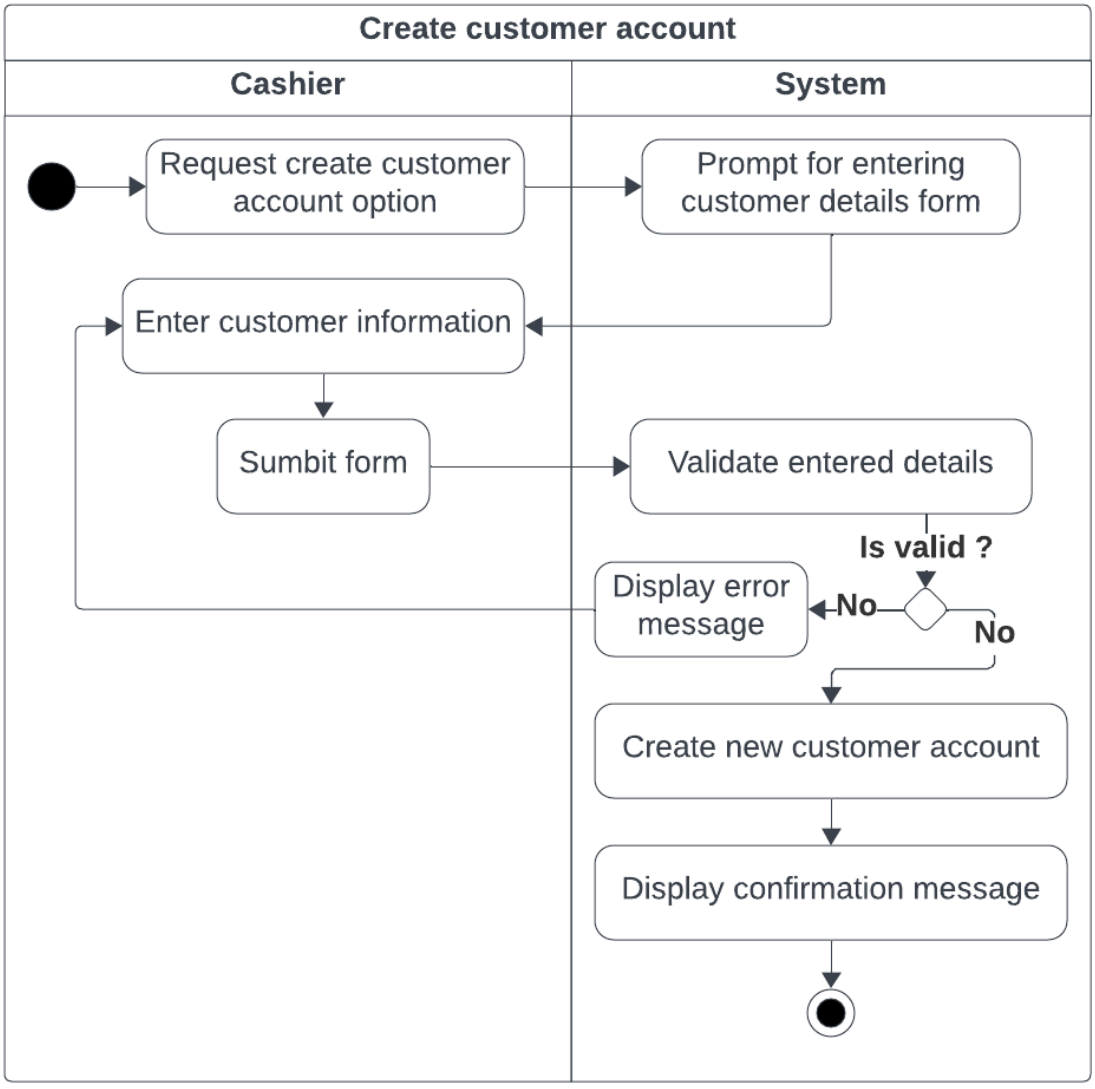

Figure 6-39

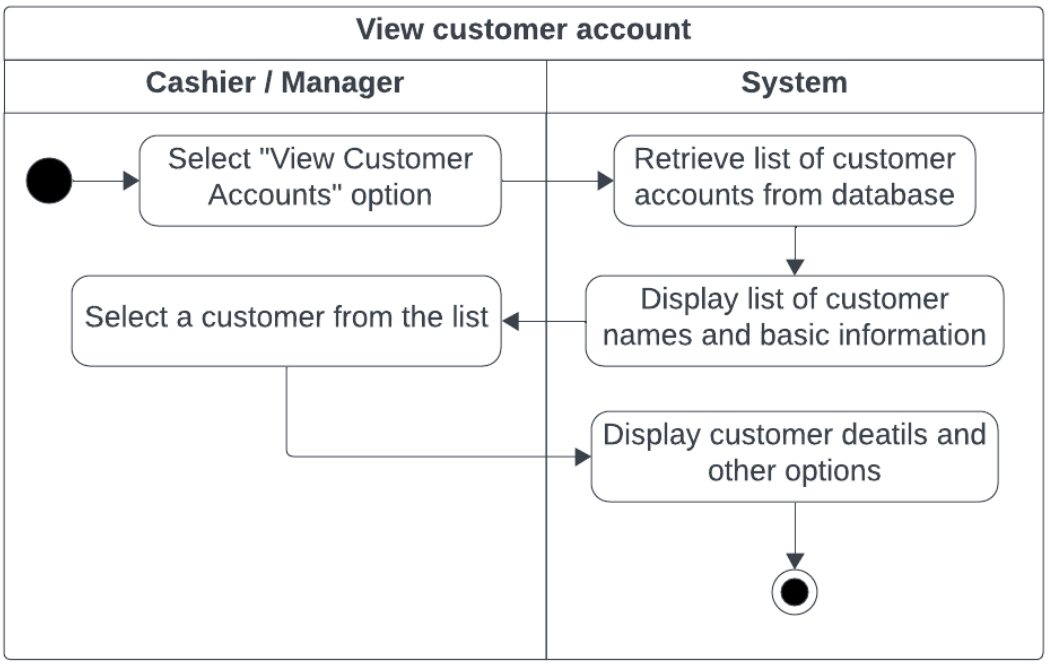

Figure 6-40

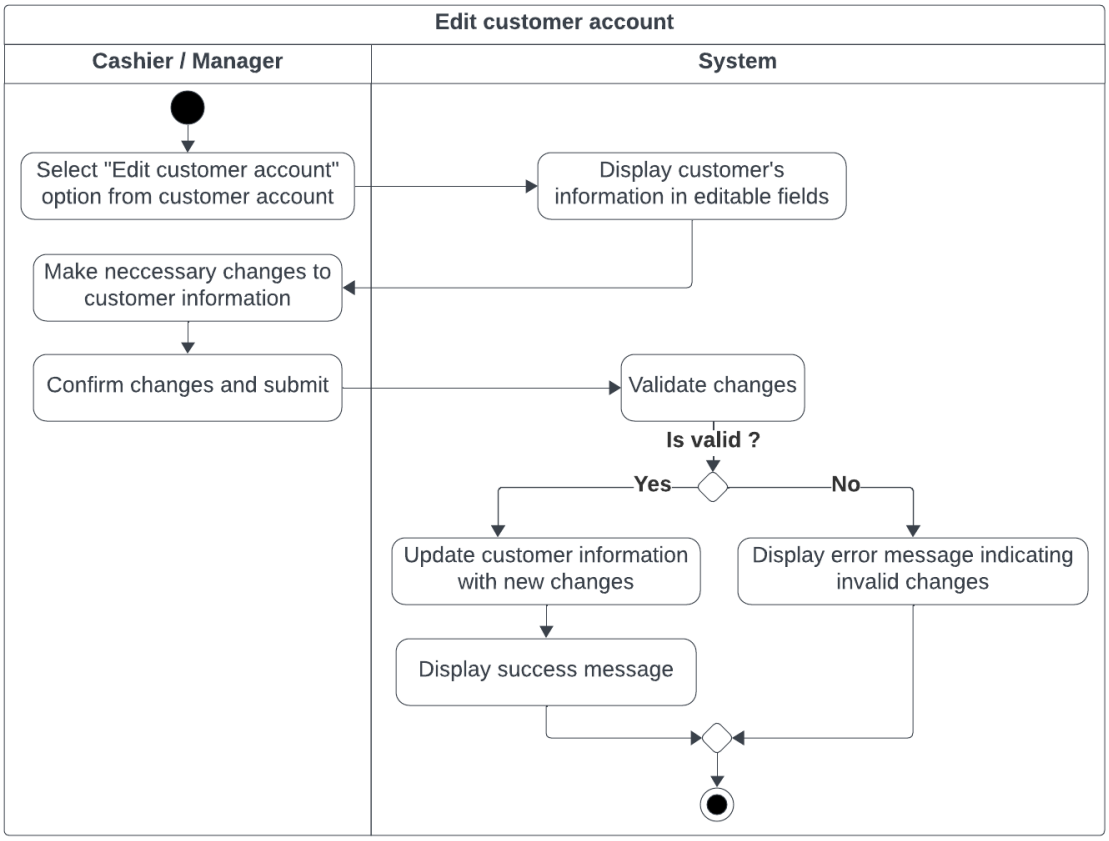

Figure 6-41

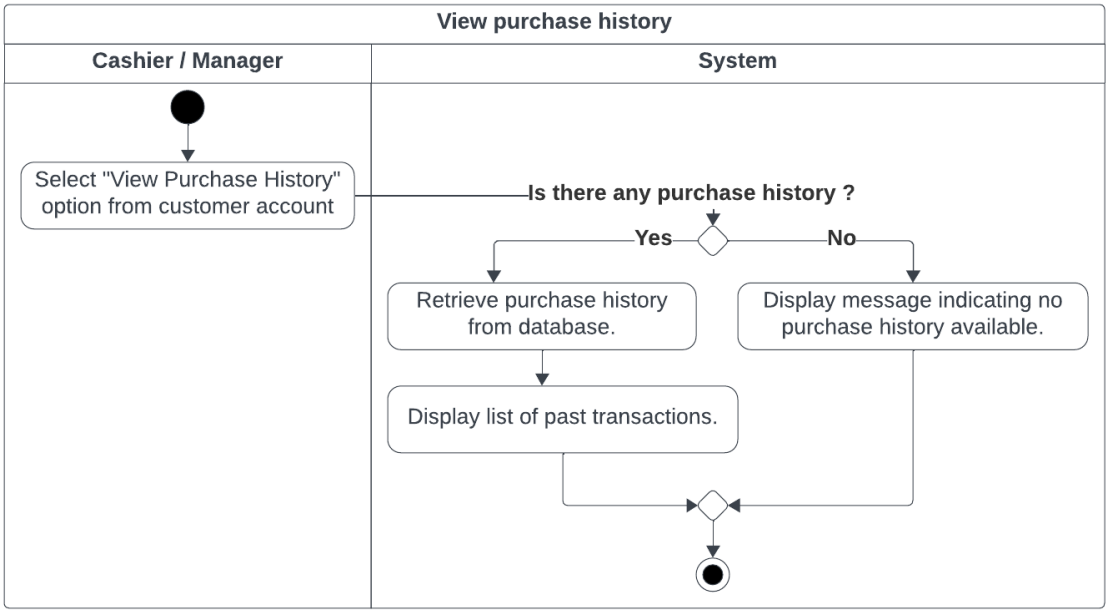

Figure 6-42

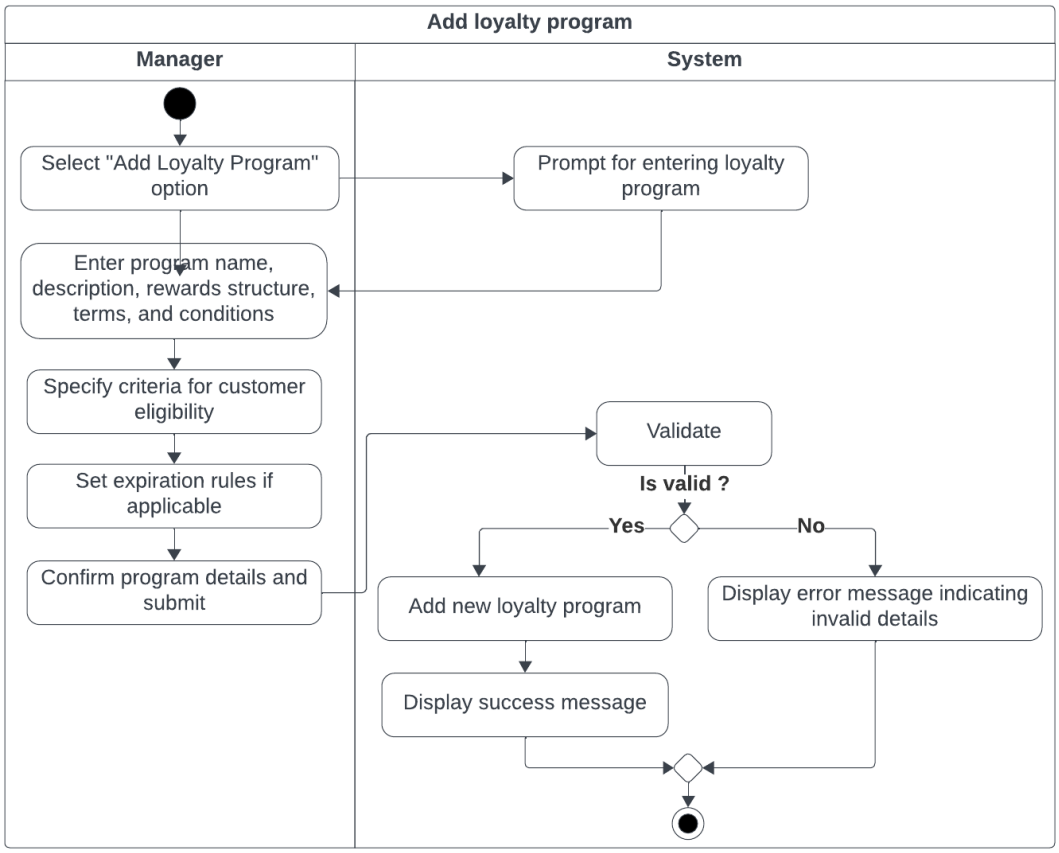

Figure 6-43

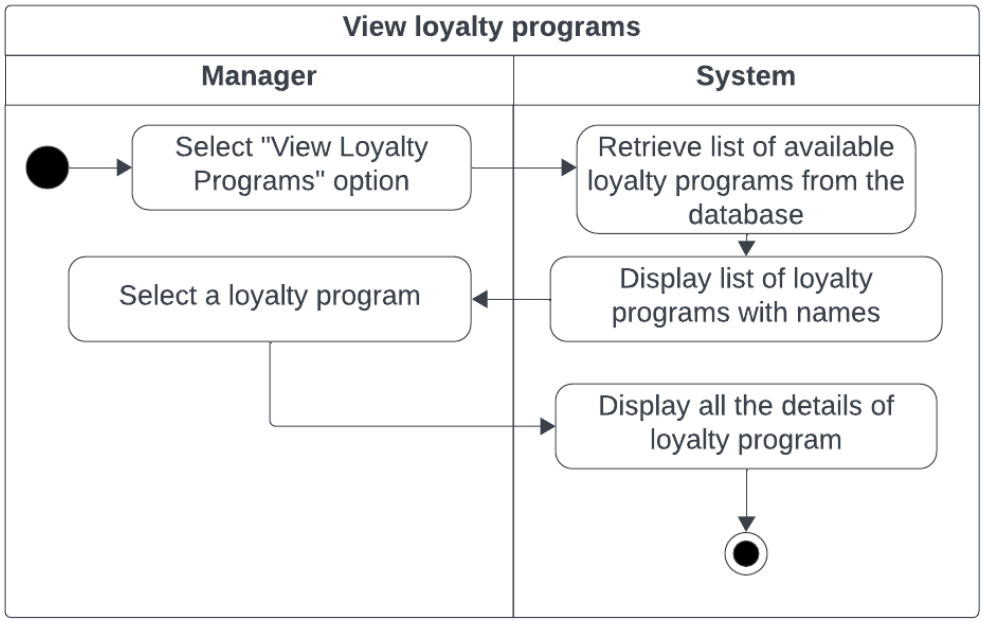

Figure 6-44

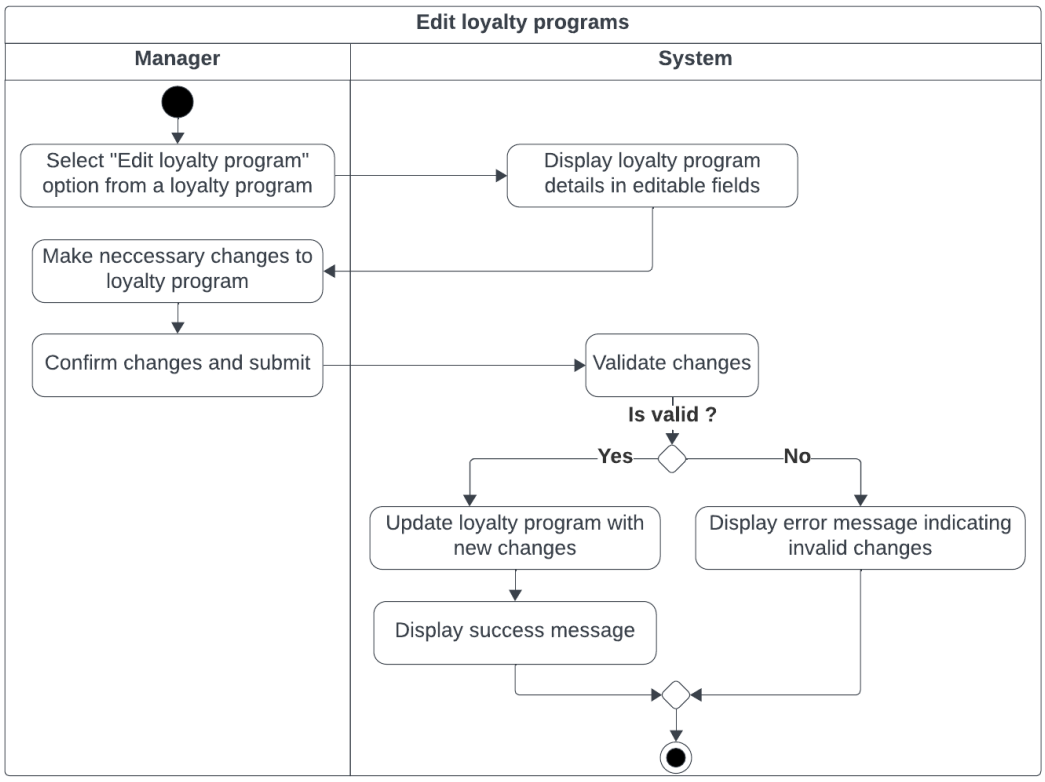

Figure 6-45

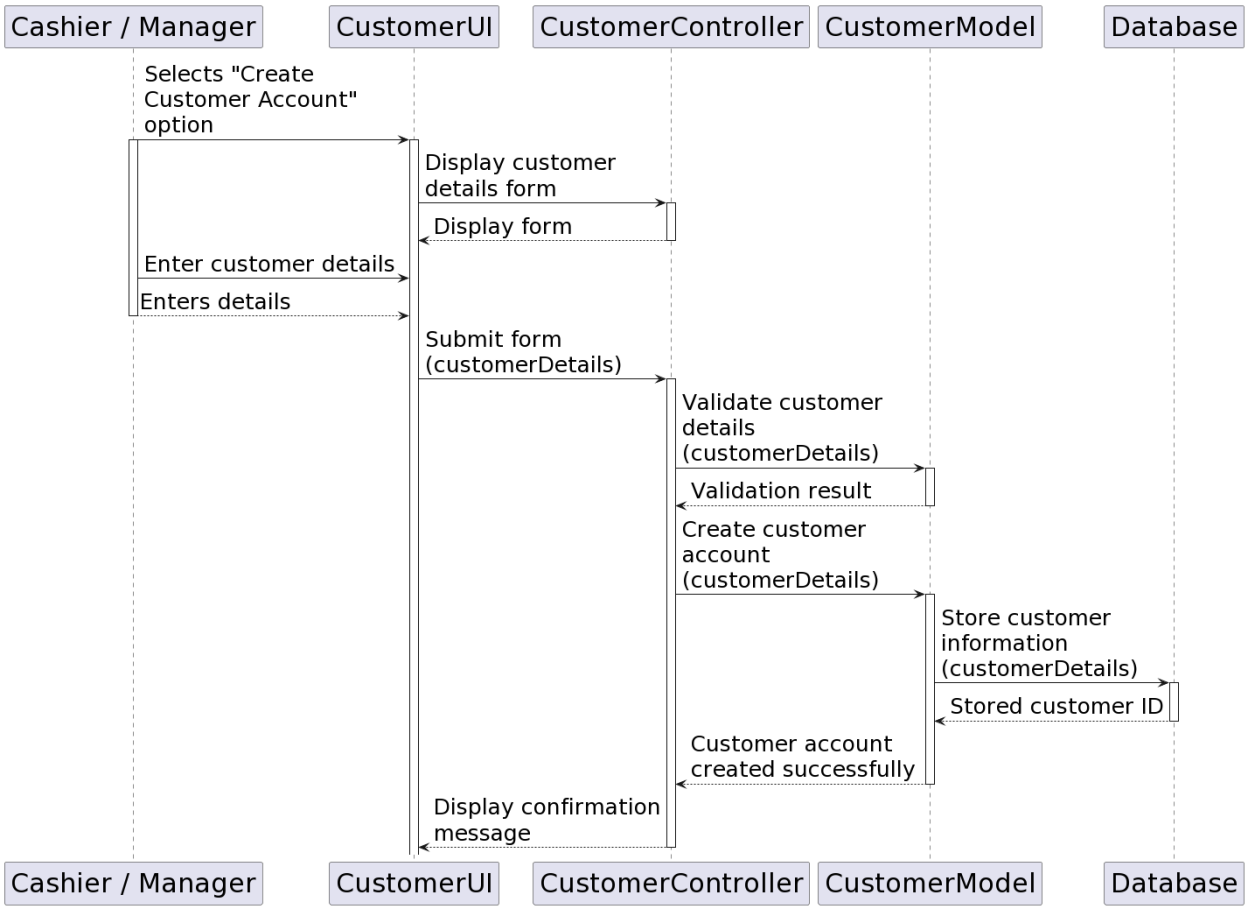

Figure 6-46 Create Customer Account

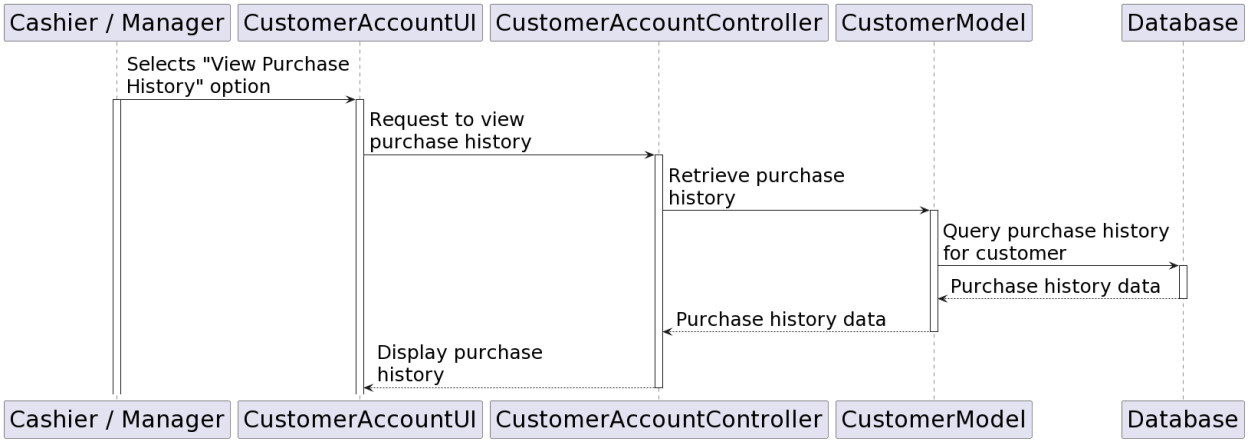

Figure 6-47 View Purchase History

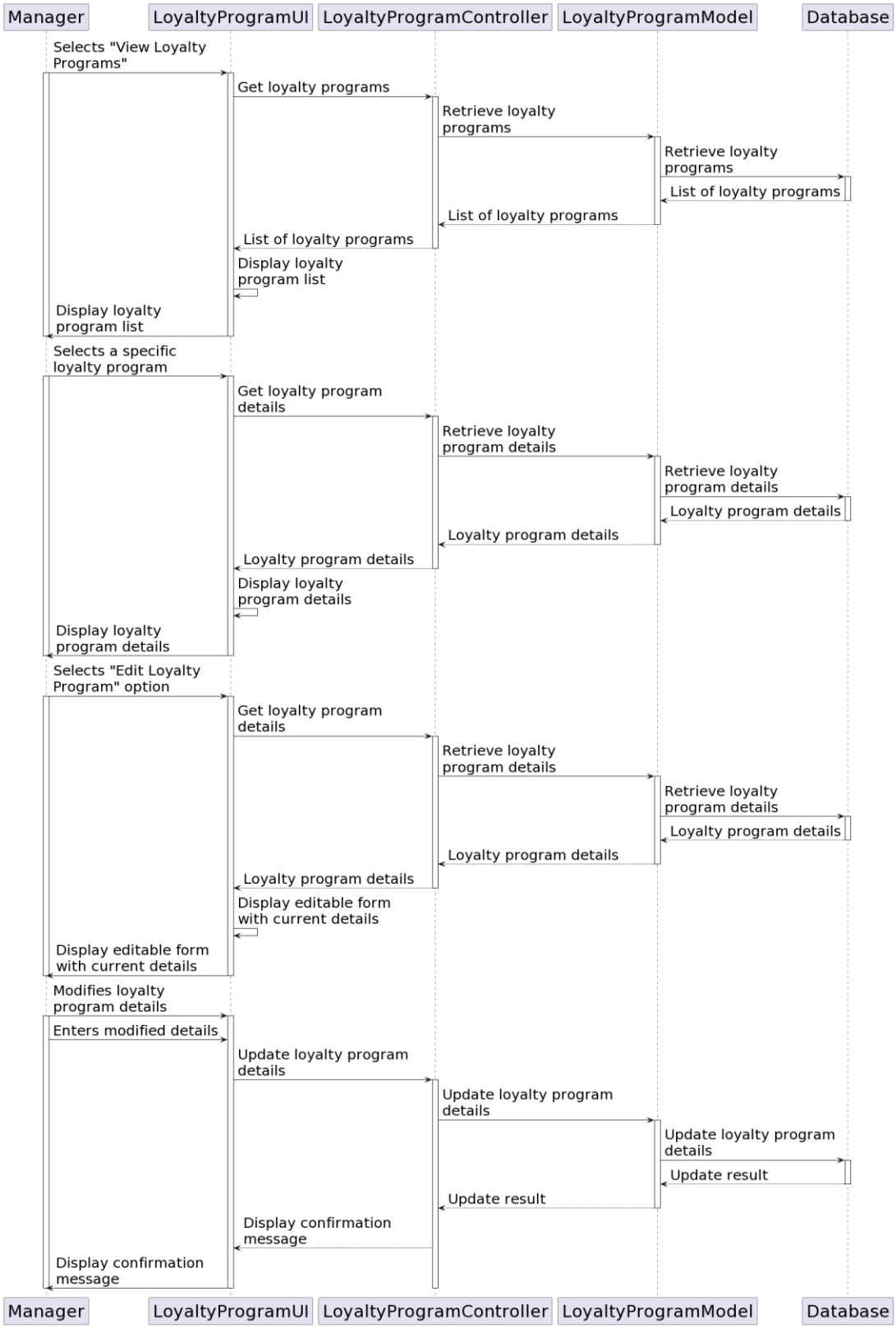

Figure 6-48 Add, View or Edit Loyalty Programs

6.5 Reports and Analytics

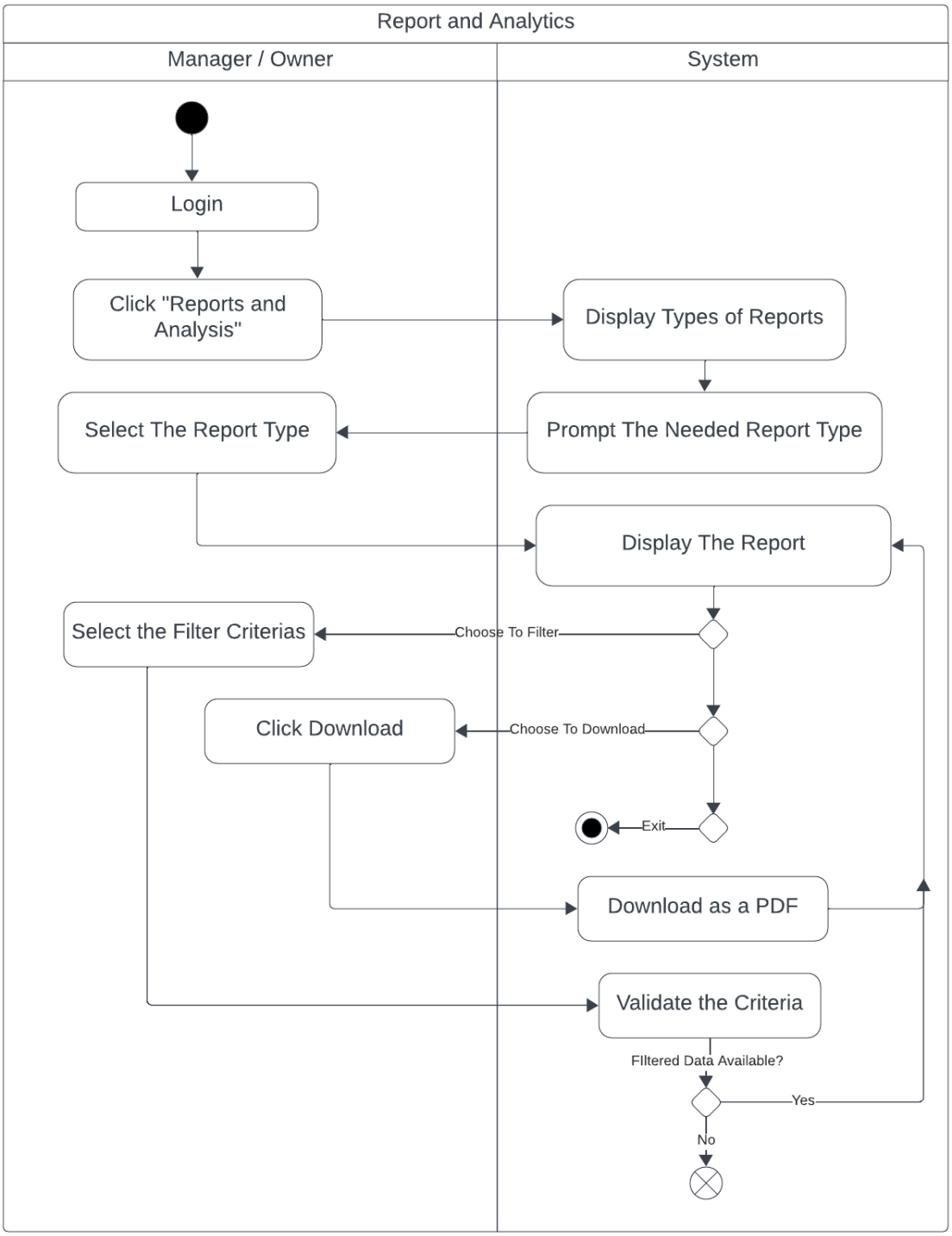

Figure 6-49

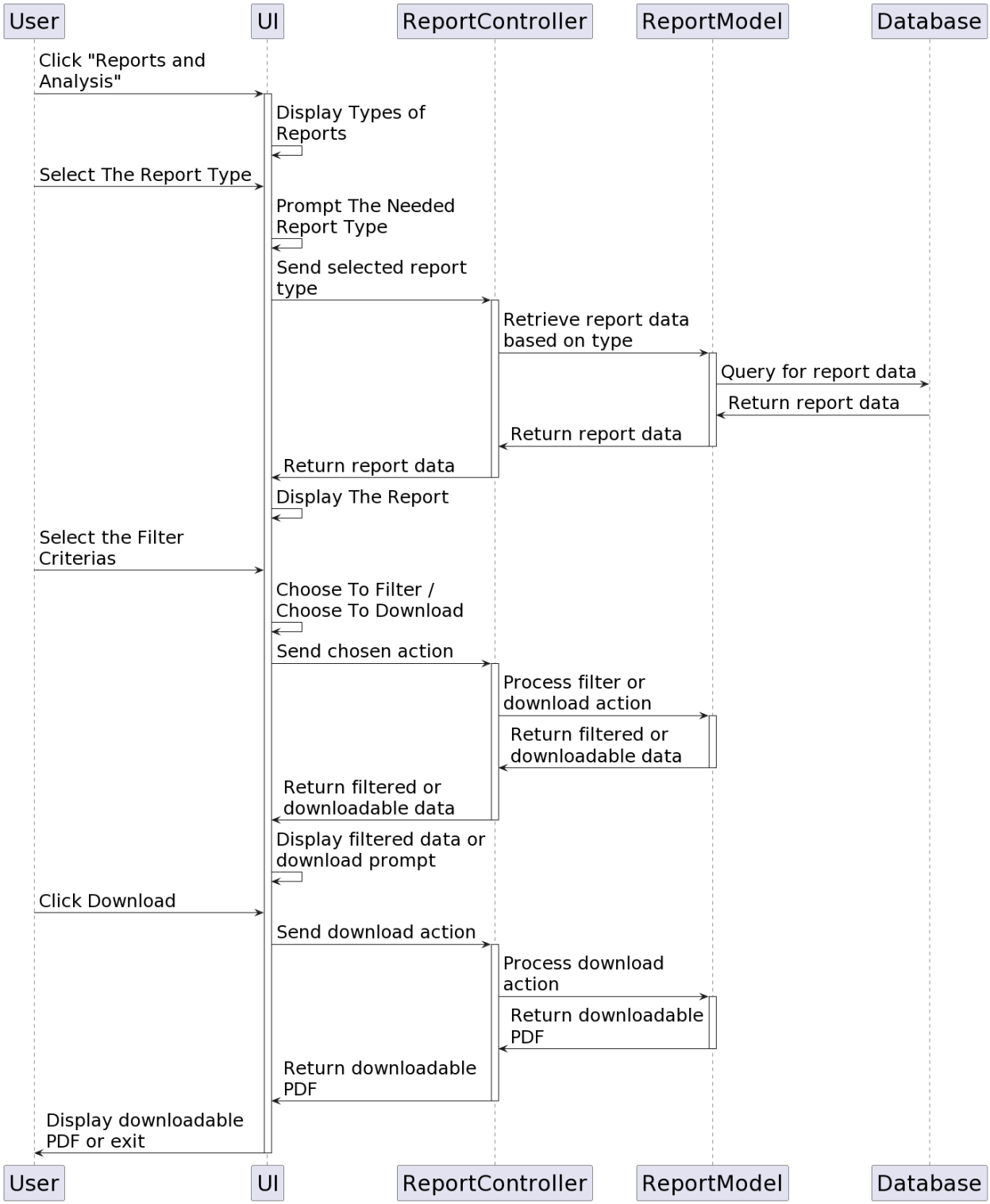
User
UI
ReportController
ReportModel
Database
Click "Reports and Analysis"
Display Types of Reports
Select The Report Type
Prompt The Needed Report Type
Send selected report type
Retrieve report data based on type
Query for report data
Return report data
Return report data
Return report data
Display The Report
Select the Filter Criterias
Choose To Filter / Choose To Download
Send chosen action
Process filter or download action
Return filtered or downloadable data
Return filtered or downloadable data
Display filtered data or download prompt
Click Download
Send download action
Process download action
Return downloadable PDF
Return downloadable PDF
Display downloadable PDF or exit
User
UI
ReportController
ReportModel
Database

Figure 6-50

6.6 Ecommerce Integration Subsystem

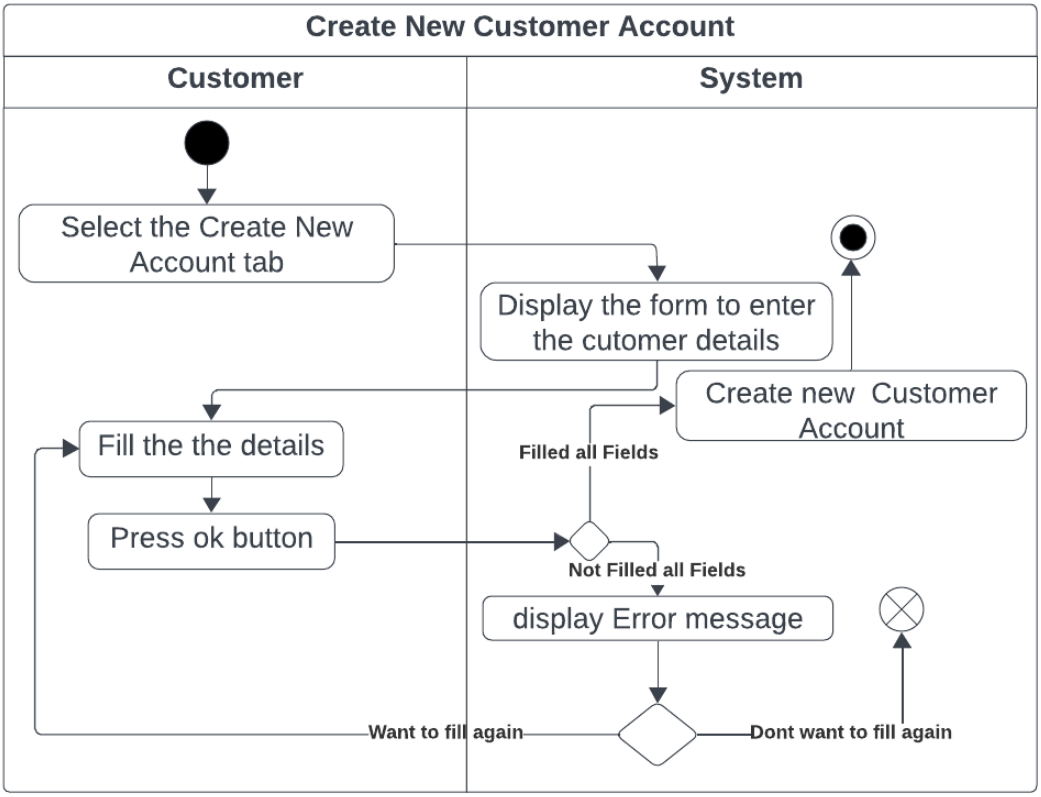

Figure 6-51

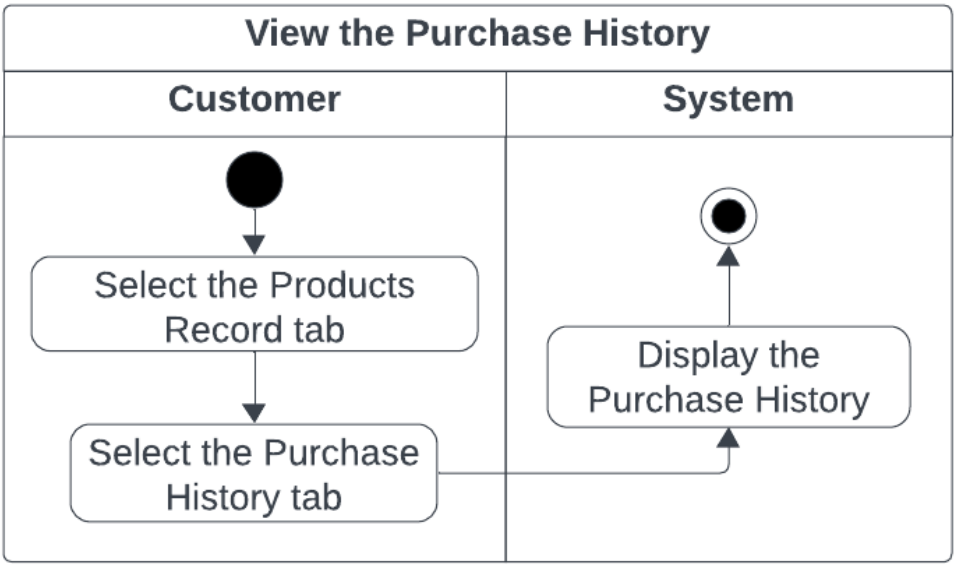

Figure 6-52

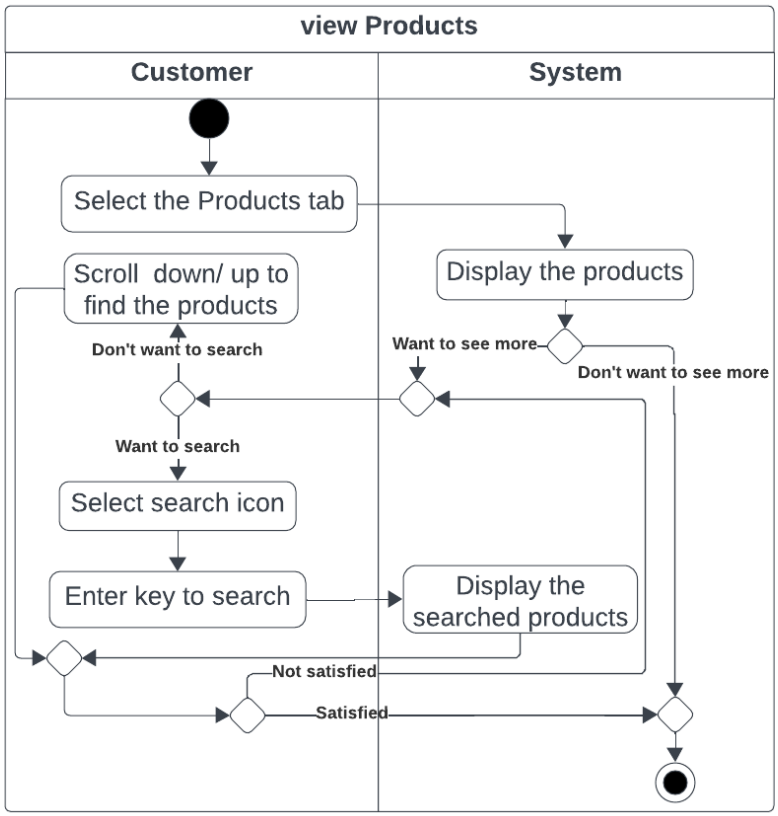

Figure 6-53

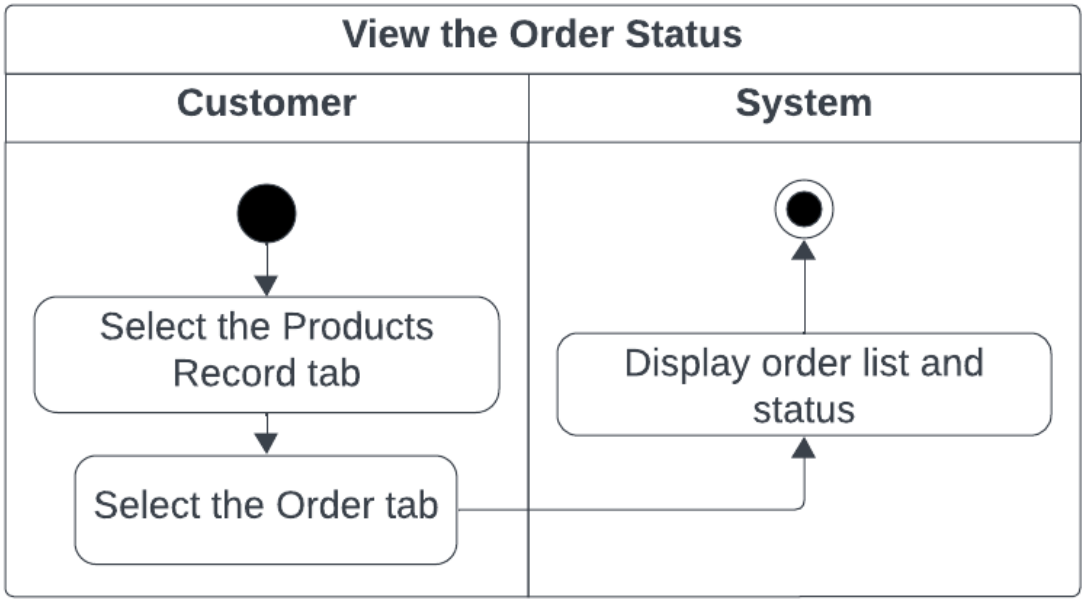

Figure 6-54

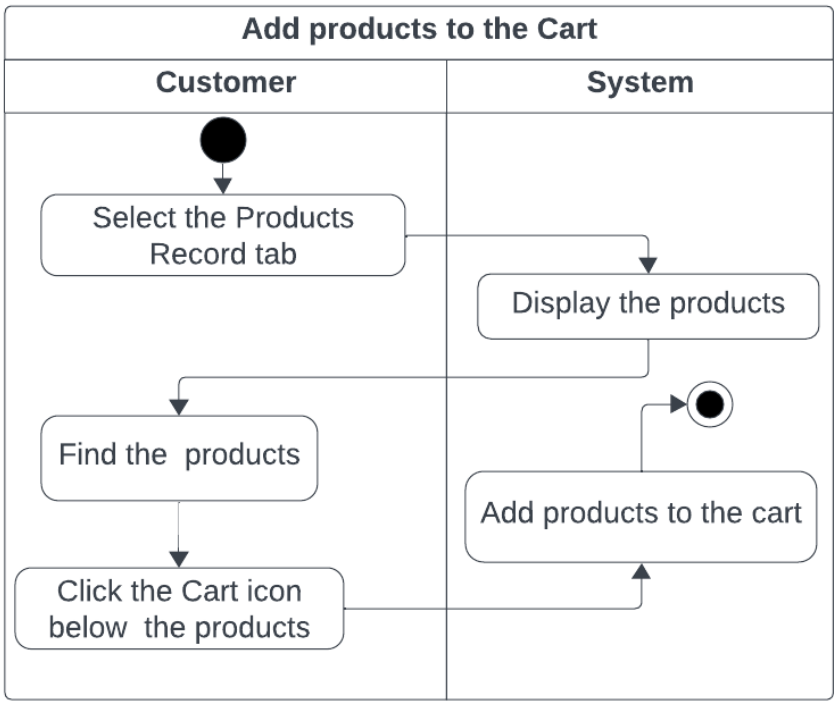

Figure 6-55

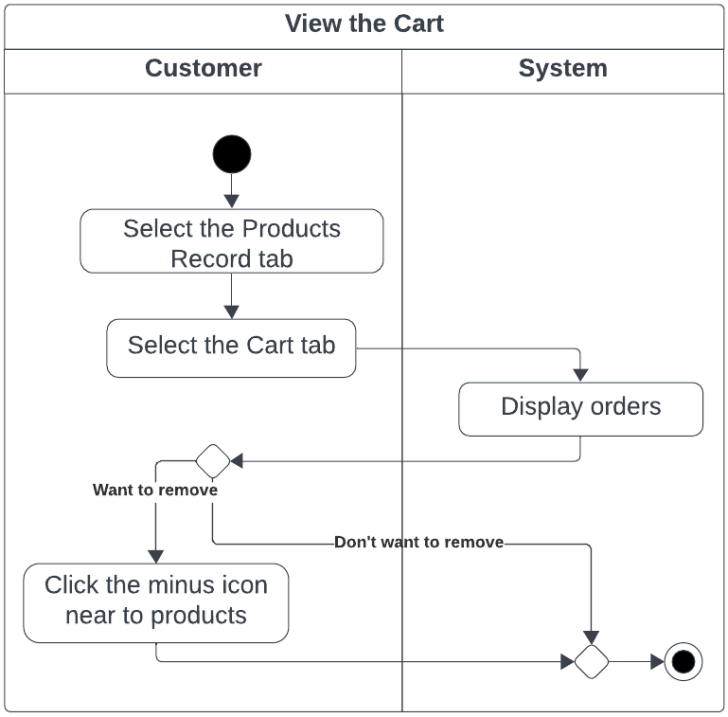

Figure 6-56

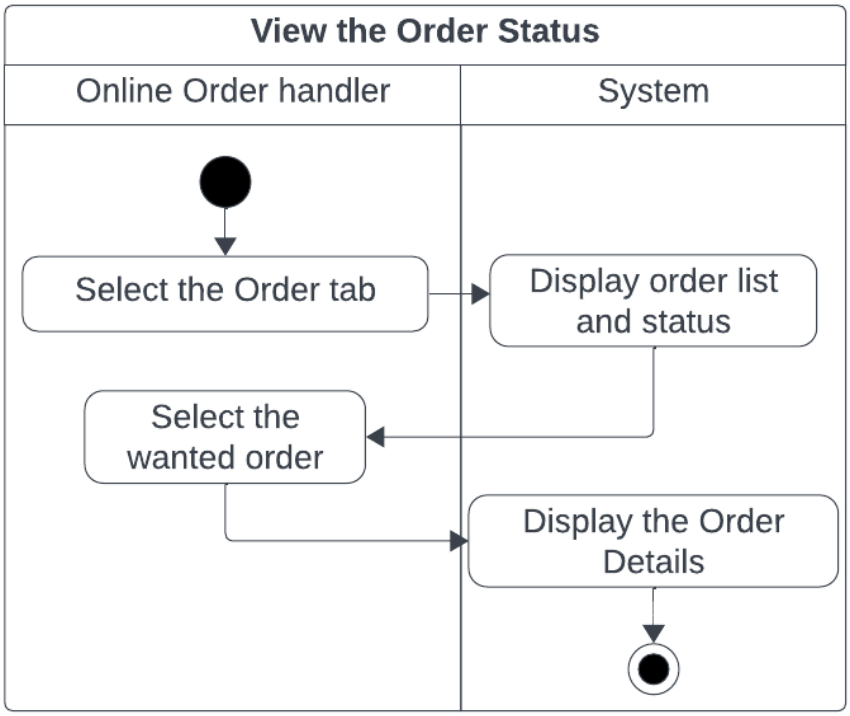

Figure 6-57

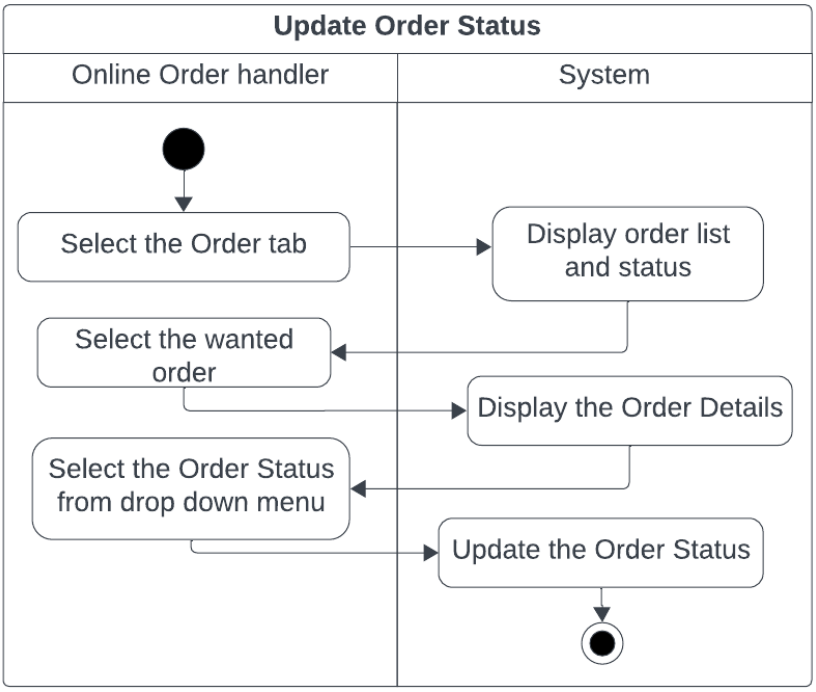

Figure 6-58

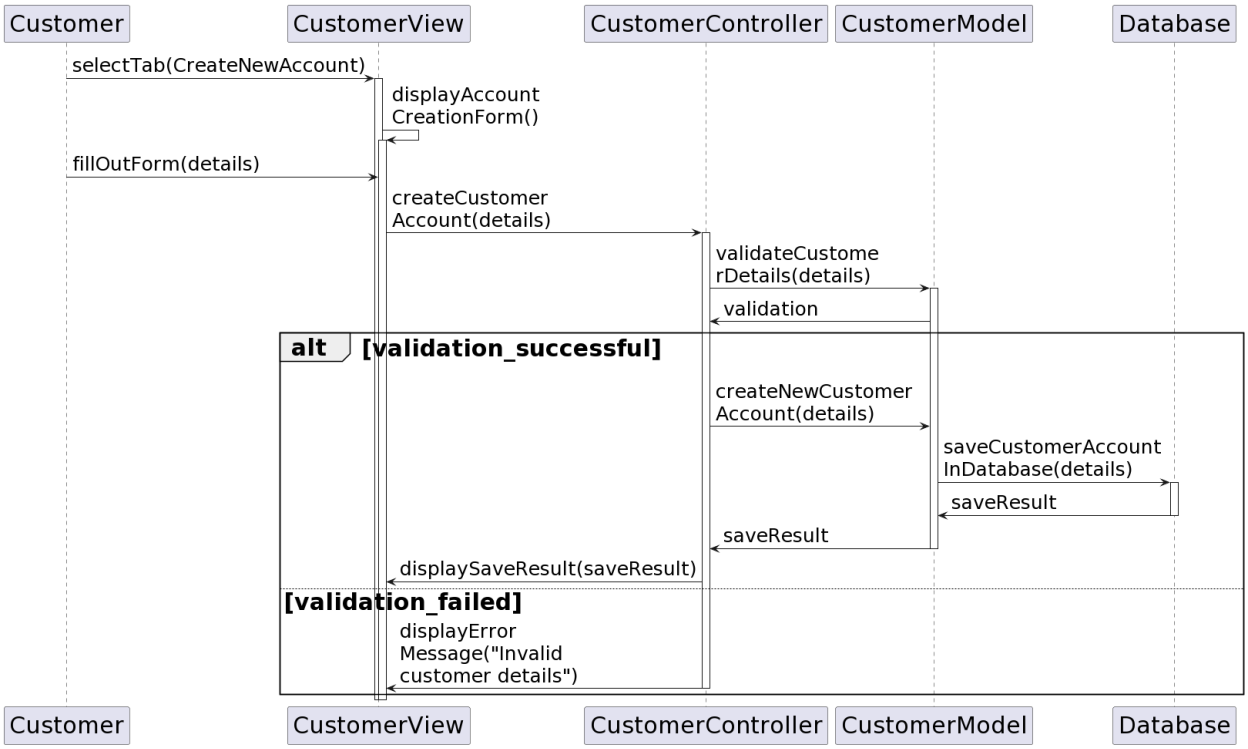

Figure 6-59

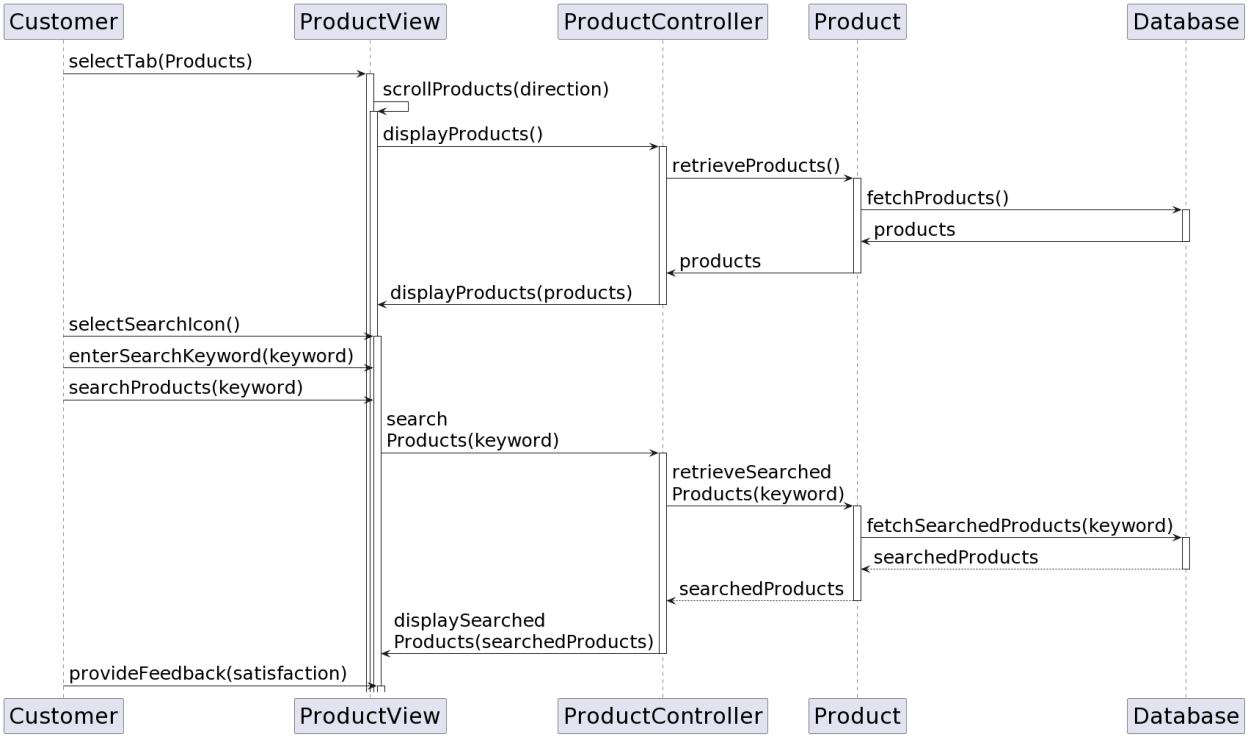

Figure 6-60

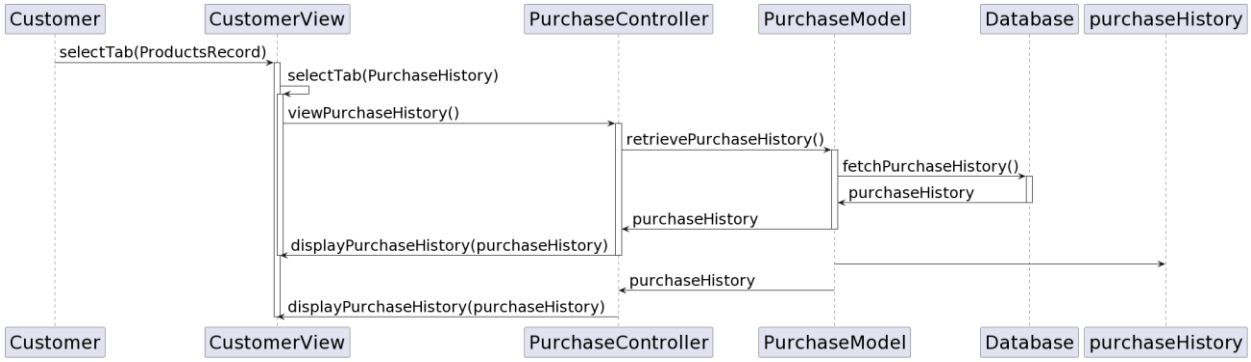

Figure 6-61

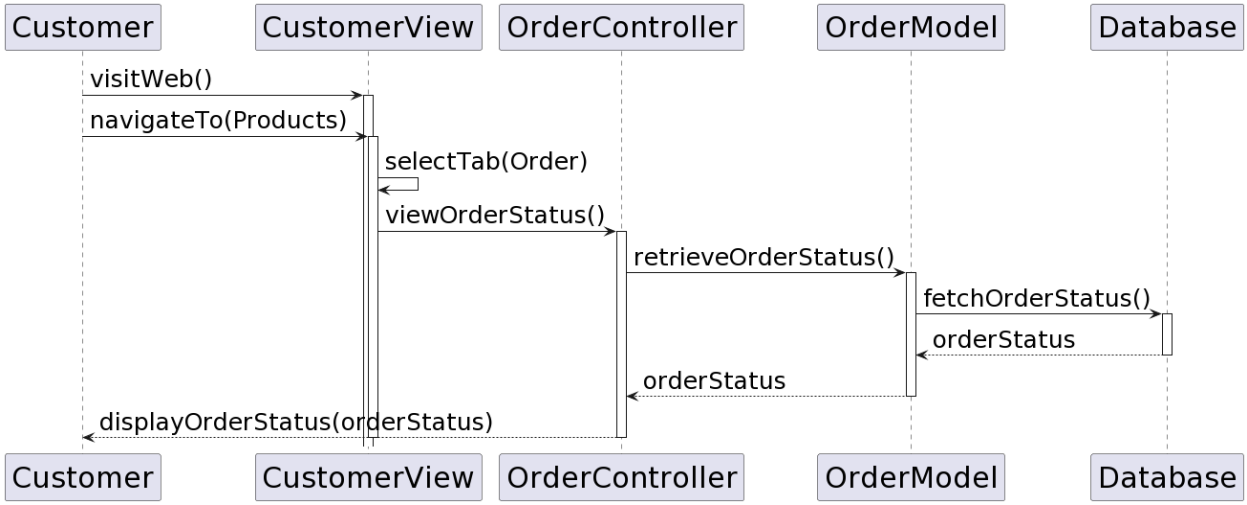

Figure 6-62

6.7 Multi-Store Support Management Subsystem

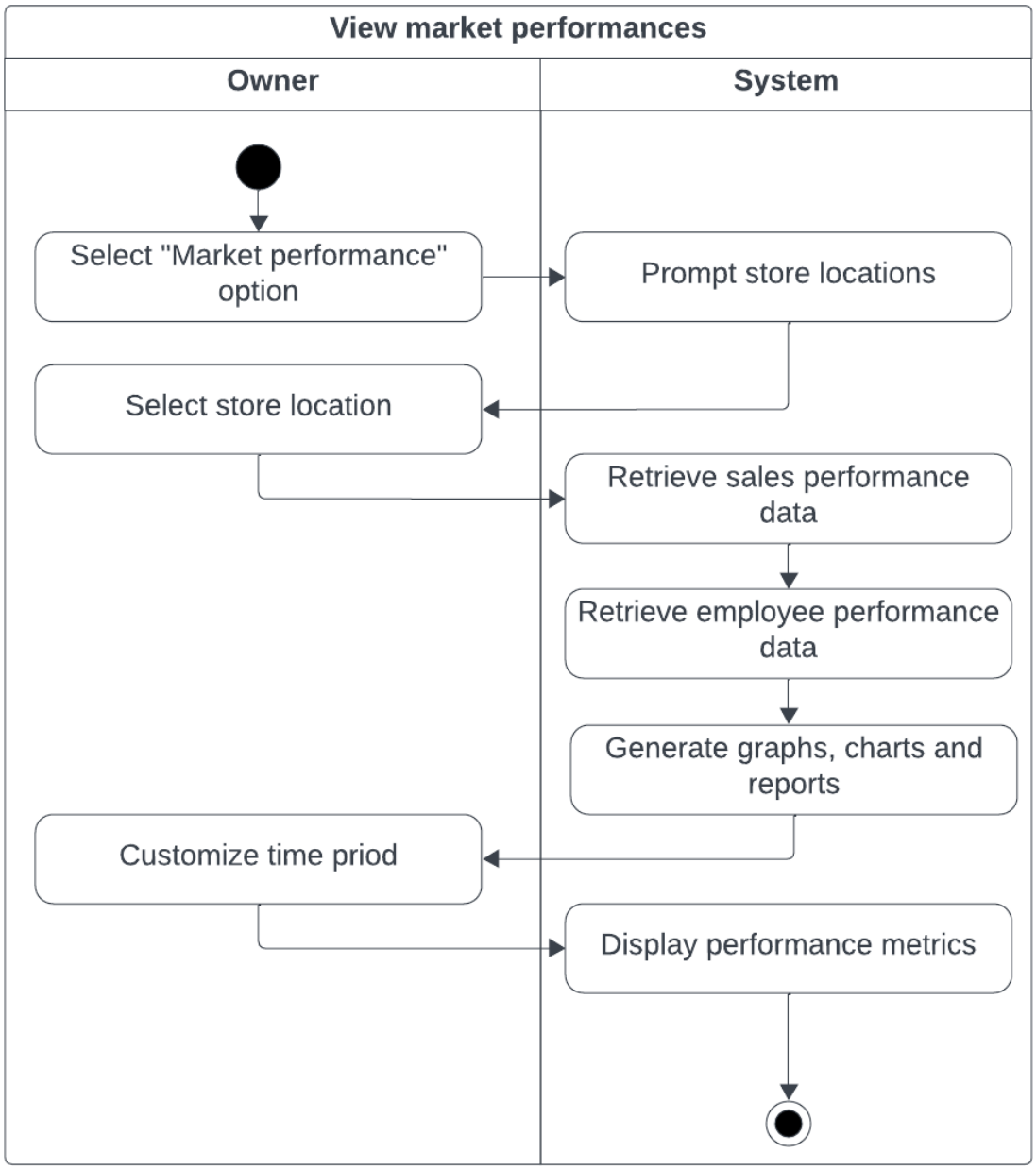

Figure 6-63

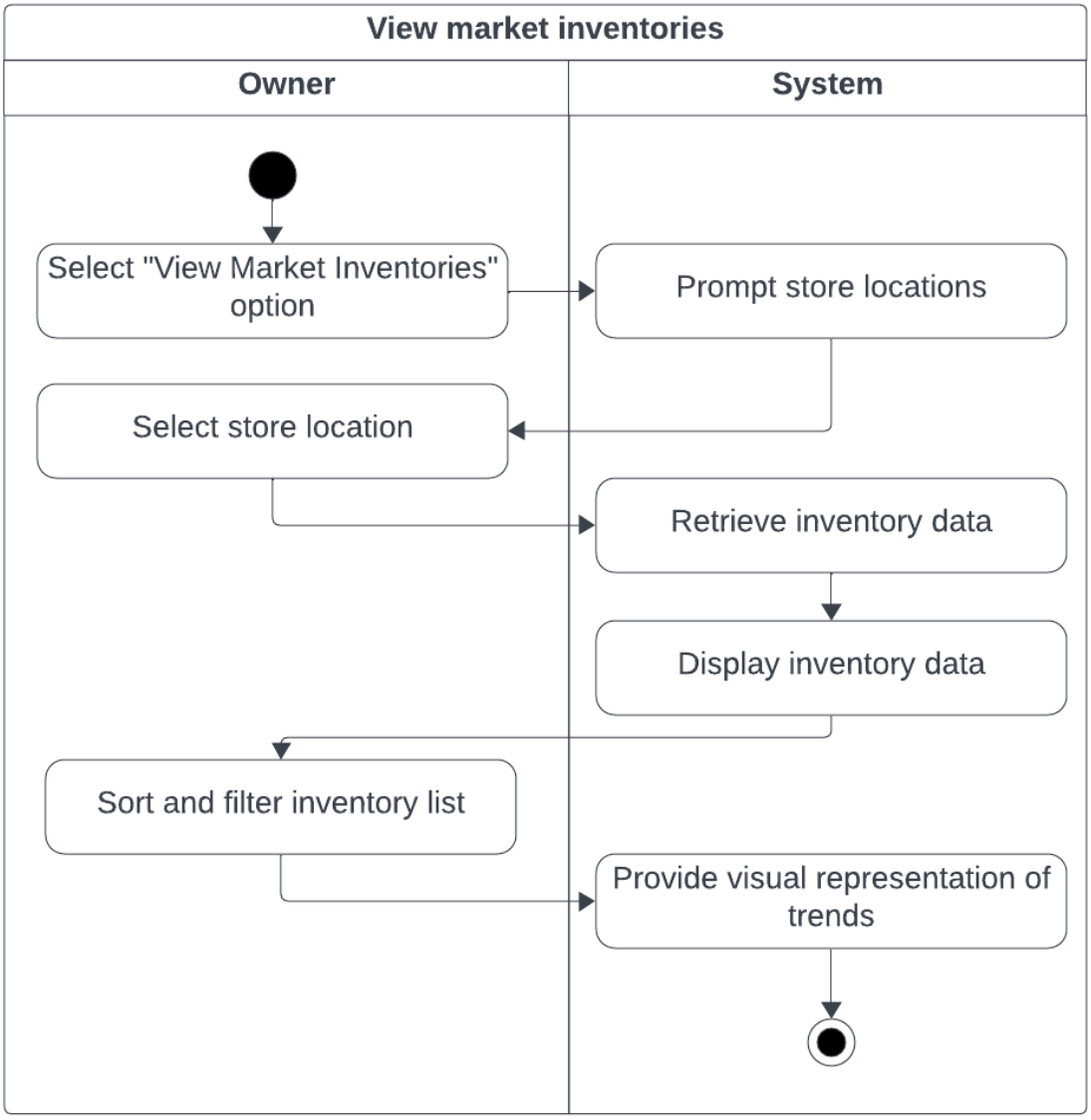
View market inventories
Owner
System
Select "View Market Inventories" option
Prompt store locations
Select store location
Retrieve inventory data
Display inventory data
Sort and filter inventory list
Provide visual representation of trends

Figure 6-64

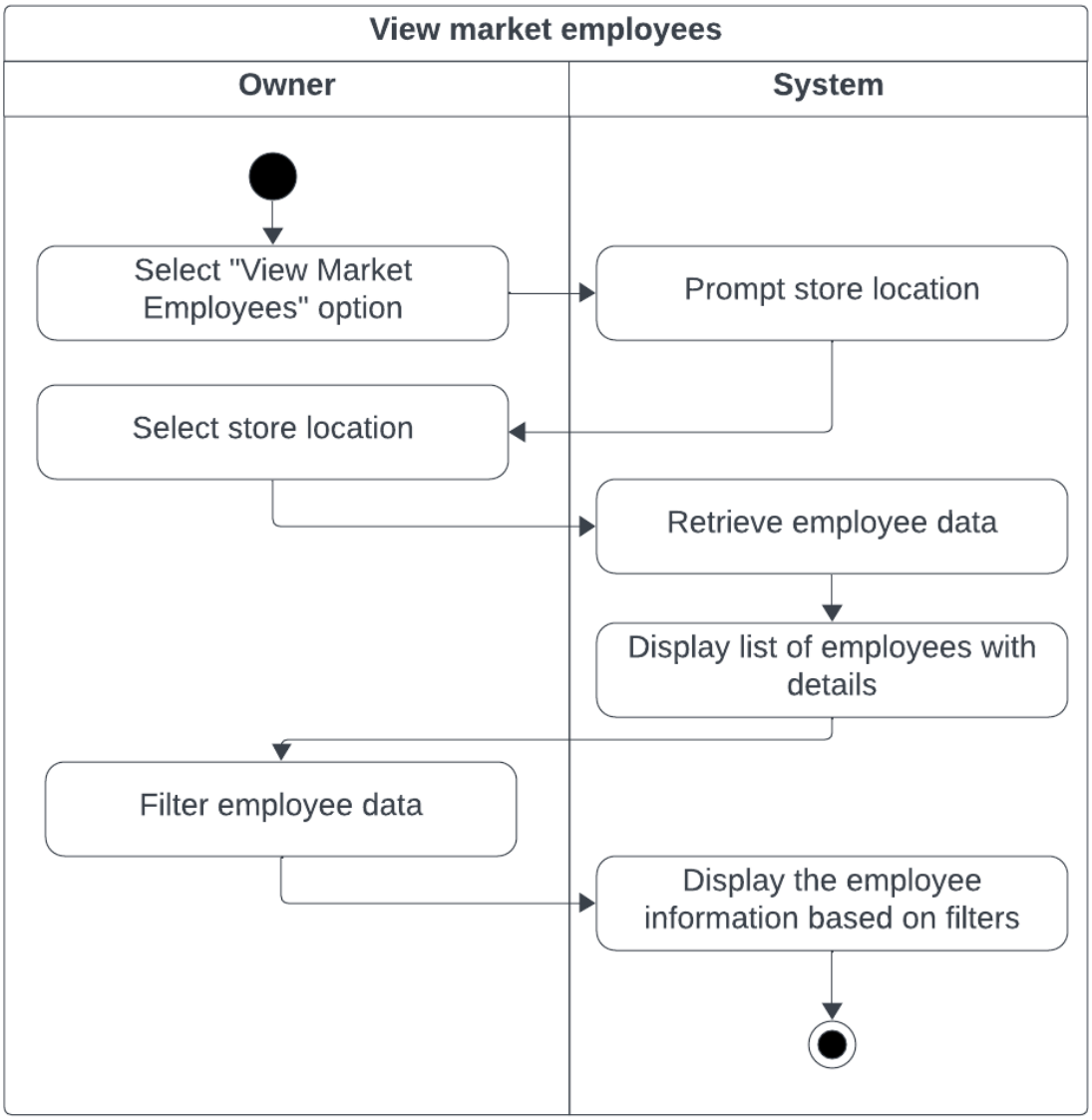
View market employees
Owner
System
Select "View Market Employees" option
Prompt store location
Select store location
Retrieve employee data
Display list of employees with details
Filter employee data
Display the employee information based on filters

Figure 6-65

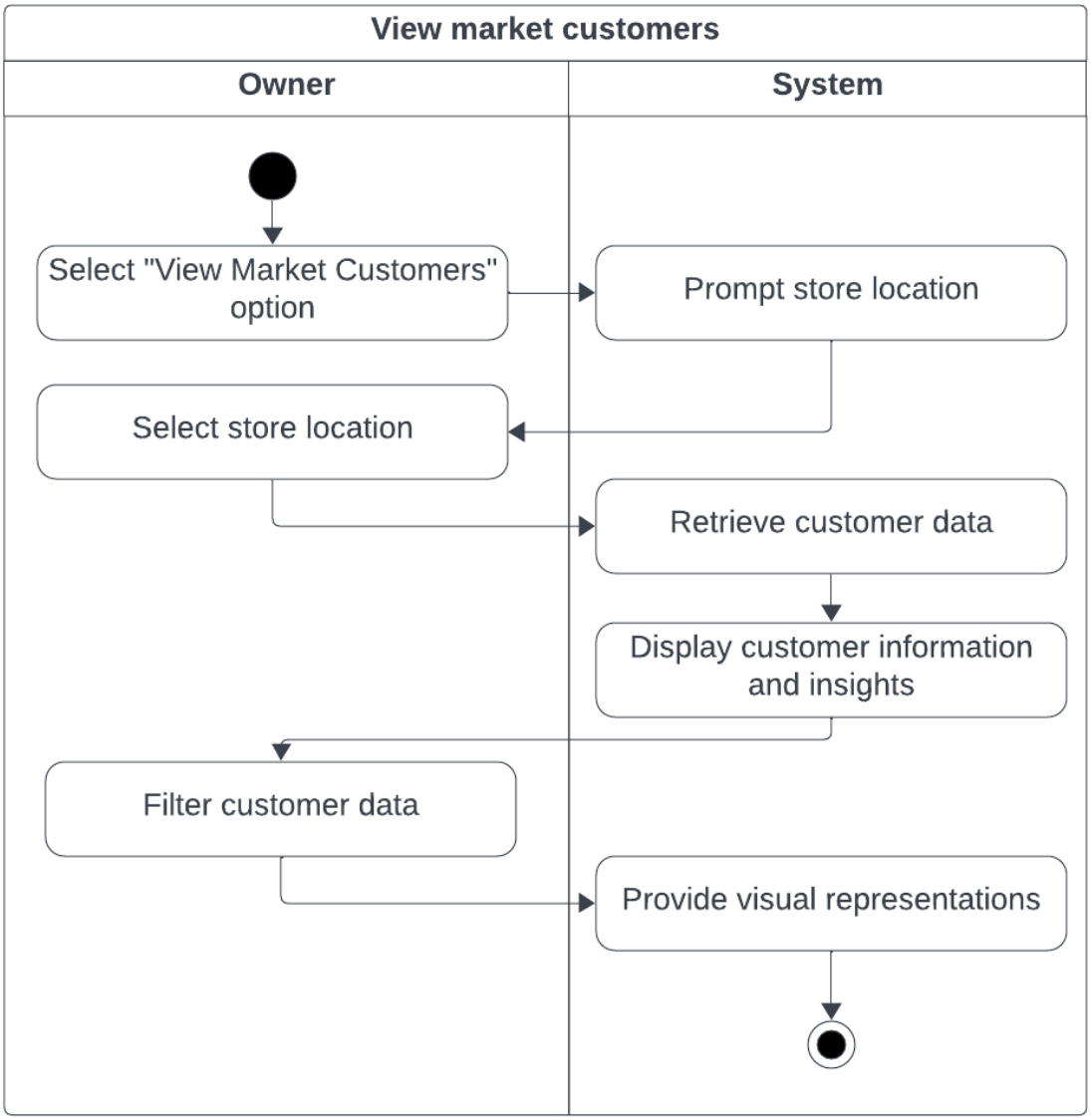
View market customers
Owner
System
Select "View Market Customers" option
Prompt store location
Select store location
Retrieve customer data
Display customer information and insights
Filter customer data
Provide visual representations

Figure 6-66

7. Deployment View

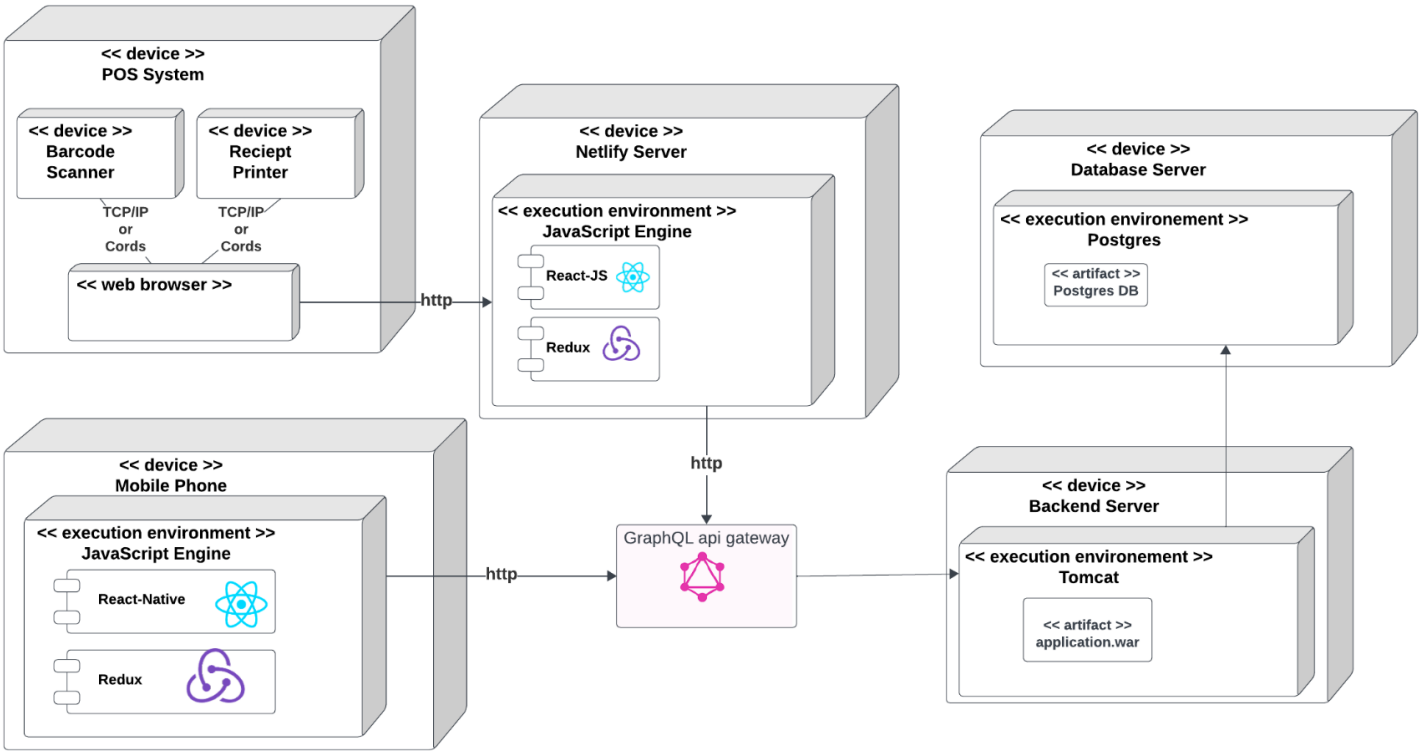

Figure 6-1 Deployment Diagram

8. Implementation View

8.1 Overview

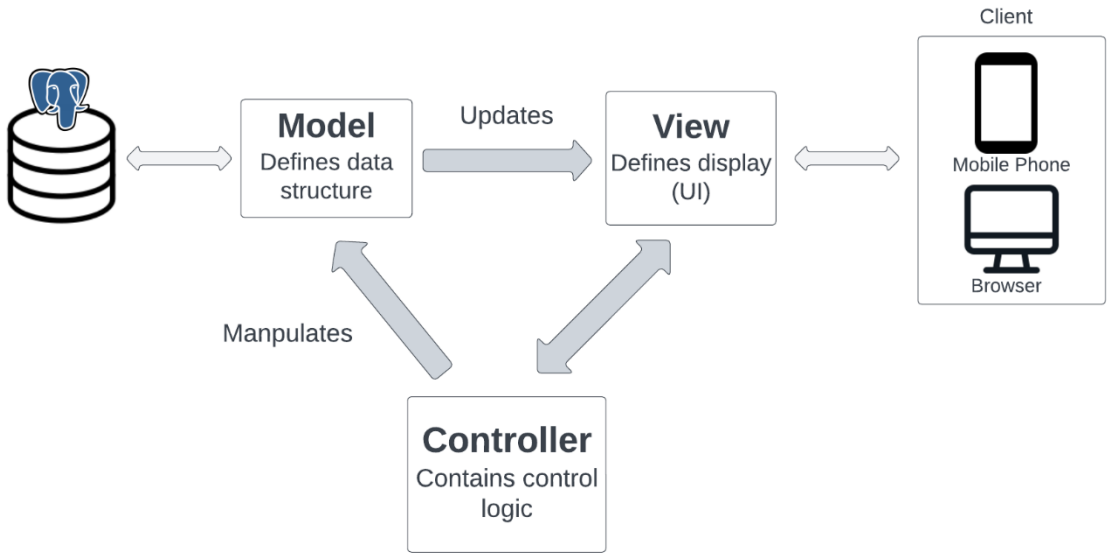

Figure 8-1 MVC architecture

The Model-View-Controller (MVC) architecture divides an application into three distinct components. By utilizing MVC, an application achieves a clear separation between its business logic and presentation layer.

Model

The Model is responsible for managing the application's data, logic, and rules. It represents the core business logic and data structures. This component handles tasks such as data validation, calculations, database interactions, and other data-related operations. The Model ensures that data remains consistent and up to date, regardless of the user interface. In essence, it encapsulates the application's underlying data and functionality.

View

The View represents the user interface components that display data to users and receive user input. Its primary role is to present the data provided by the Model in a user-friendly format. The View is responsible for layout, appearance, and how the information is presented to the user. It can include graphical elements, text, forms, and other visual components. However, the View does not handle the application's logic or data manipulation.

Controller

The Controller acts as an intermediary between the Model and the View. It processes user input and translates it into actions to be taken by the Model or the View. The Controller handles user interactions, such as clicking a button or submitting a form, and invokes the appropriate actions in the Model. It also updates the View based on changes in the Model's data. In short, the Controller coordinates the flow of data between the Model and the View.

8.2 Layers

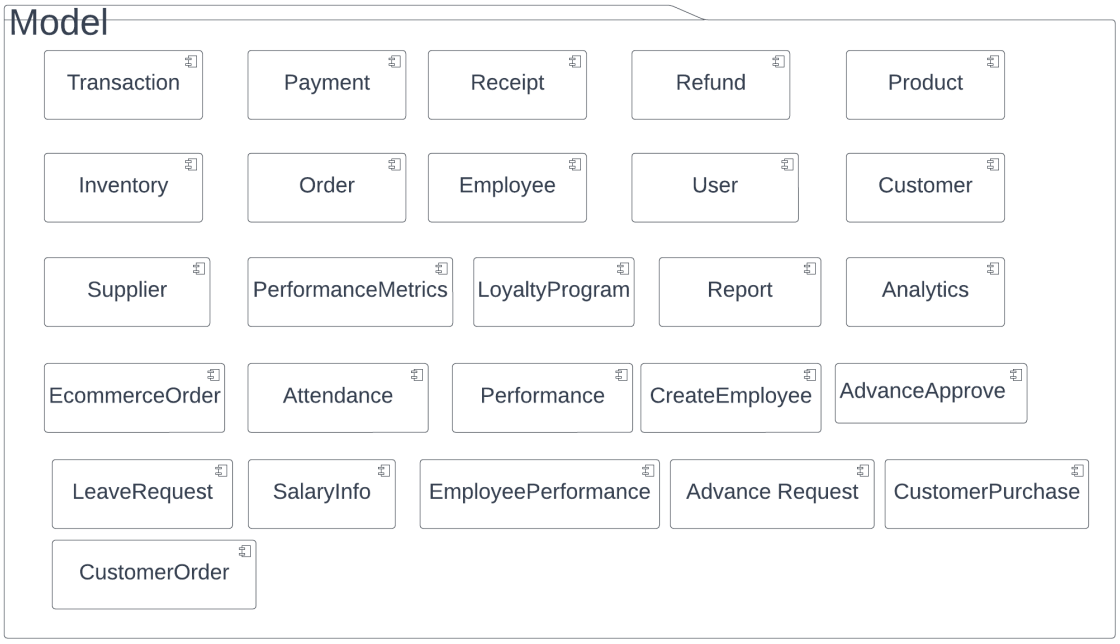

Figure 8-2 Models

Figure 8-3 Views

Figure 8-4 Controllers

9. Data View

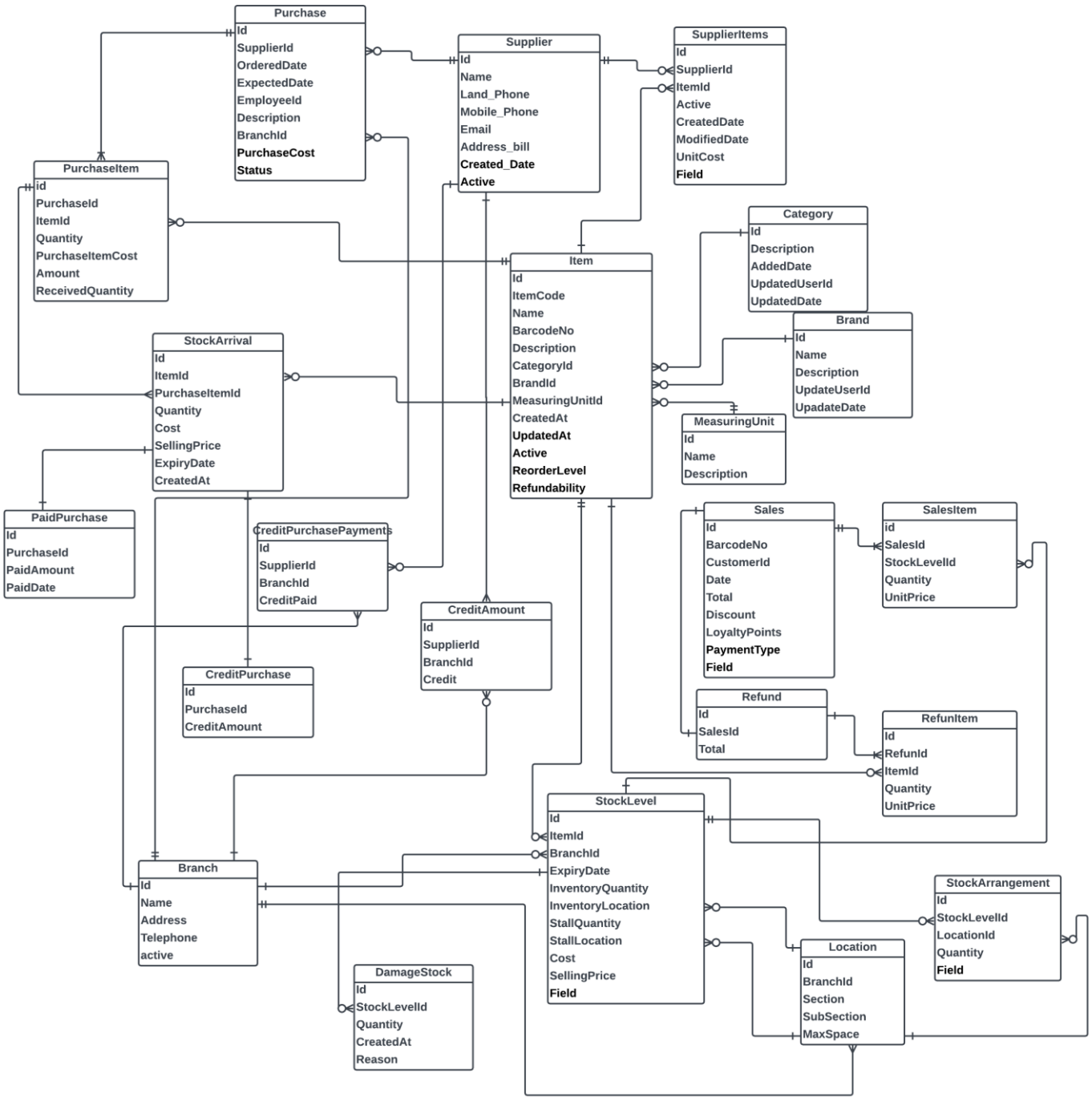

Figure 9-1 ER diagram part 1

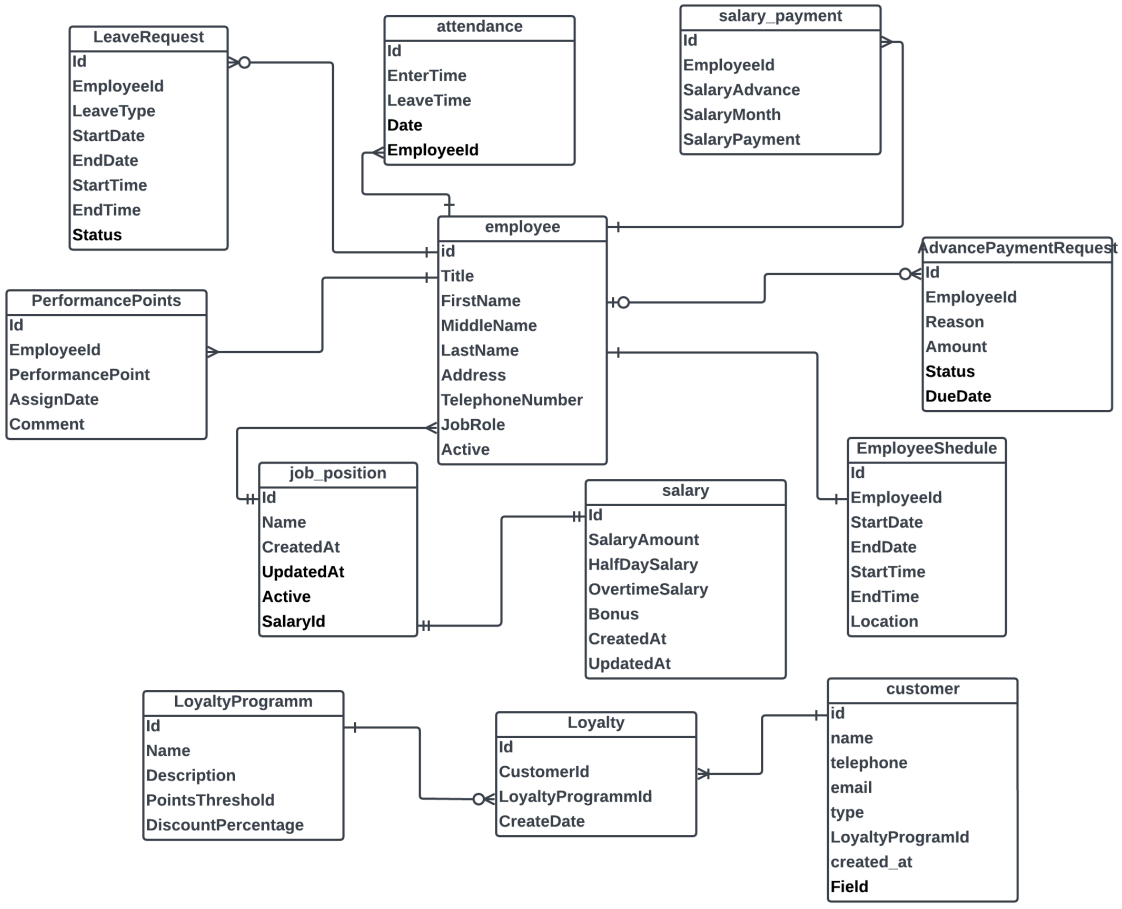

Figure 9-2 ER-diagram part 2

10. Size and Performance

The Smart POS software architecture is designed with a focus on accommodating the anticipated transactional load and data volume of a multi-store supermarket environment. The following key subtopics outline the major dimensioning characteristics and target performance constraints.

10.1 Transactional Load Dimensioning

The architecture considers the expected number of concurrent users, sales transactions, and data interactions across multiple store locations. This dimensioning ensures that the system can handle peak loads without compromising performance.

10.2 Data Volume Considerations

The architecture accounts for the volume of inventory data, customer profiles, and transaction records that are generated and maintained within the system. This consideration includes strategies for efficient data storage, retrieval, and management.

10.3 Performance Constraints

The Smart POS architecture defines specific performance constraints to ensure seamless user experience and timely data processing. The following targets have been established:

Transaction Processing Time

The system aims to achieve a response time of under 2 seconds for processing sales transactions, including payment processing and receipt generation.

Inventory Update Latency

Real-time inventory updates are targeted to ensure accurate stock levels across all store locations. The architecture seeks to minimize latency in updating inventory records.

Scalability

The architecture is designed to scale horizontally to accommodate potential increases in the number of store locations, customer transactions, and data volume. This scalability ensures consistent performance as the system grows.

11. Quality

The Smart POS software architecture encompasses various quality attributes that contribute to the overall reliability, security, maintainability, and extensibility of the system. The following subtopics highlight how these quality aspects are addressed.

11.1 Reliability and Availability

The architecture incorporates mechanisms for real-time data synchronization between store locations and failover strategies to ensure data integrity and prevent service disruptions. This approach enhances system reliability and availability.

11.2 Security and Privacy

The architecture places a strong emphasis on security by implementing user authentication, role-based access control, and encryption of sensitive data, including payment information. Compliance with data protection regulations ensures customer privacy.

11.3 Portability and Compatibility

Utilizing industry-standard technologies and frameworks, the architecture ensures that the Smart POS system can be deployed across various platforms and devices. This approach enhances portability and compatibility while reducing platform-specific dependencies.

11.4 Maintainability and Extensibility

The architecture's modular design and well-defined interfaces facilitate ease of maintenance and future enhancements. New features and functionalities can be integrated seamlessly without disrupting existing components, ensuring long-term system extensibility.

12. References

[1] "What is Unified Modeling Language (UML)?" Visual-paradigm.com, 2023. [Online]. Available: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/. [Accessed: 28-Aug-2023].

[2] "Deployment Diagram Tutorial." Lucidchart, 2023. [Online].

Available: https://www.lucidchart.com/pages/uml-deployment-diagram. [Accessed: 28-Aug-2023].

[3] "Package Diagram Tutorial." Lucidchart, 2023. [Online].

Available: https://www.lucidchart.com/pages/uml-package-diagram. [Accessed: 28-Aug-2023].