

3. Design

Supermarket charging system

Supermarket charging system

ID :

PW :

GitHub address : <https://github.com/zouzou12345678/zouzou.git>

S t u d e n t N o .	2 2 1 1 3 8 8 1
N a m e	류륜장
E - m a i l	2382131937@qq.com

[Revision history]

Revision date	Version #	Description	Author
	#1.1	Perform a macro analysis using class diagrams, sequence diagrams, and state machine diagrams.	

= Contents =

1. Introduction	
2. Class diagram	
3. Sequence diagram	
4. State machine diagram	
5. Implementation requirements	
6. Glossary	
7. References	

1. Introduction

- Summarize the contents of this document.
- Describe the important points of your design.
- 12pt, 160%.

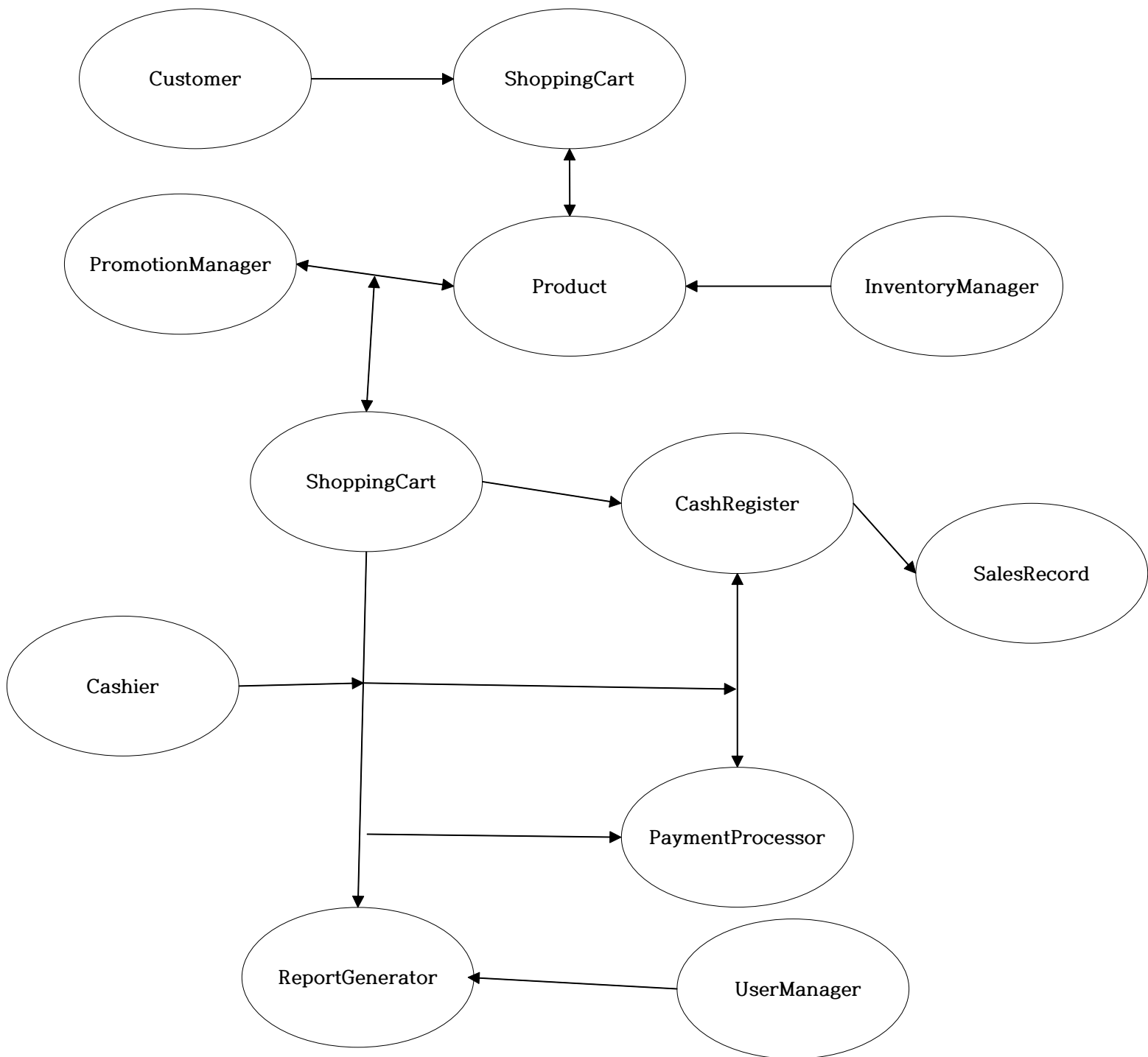
The key point of this project is that its functionality can basically realize process automation management. It reduces a large amount of manual labor, saving significant financial and human resources, as well as space and time. In addition, this system adopts the commonly used MVC design pattern under the B/S model. It has a clear division of database persistence layer, business logic functions, and interface design, enhancing its scalability and practicality. The separation of front-end and back-end management also makes the entire system more secure and user-friendly.

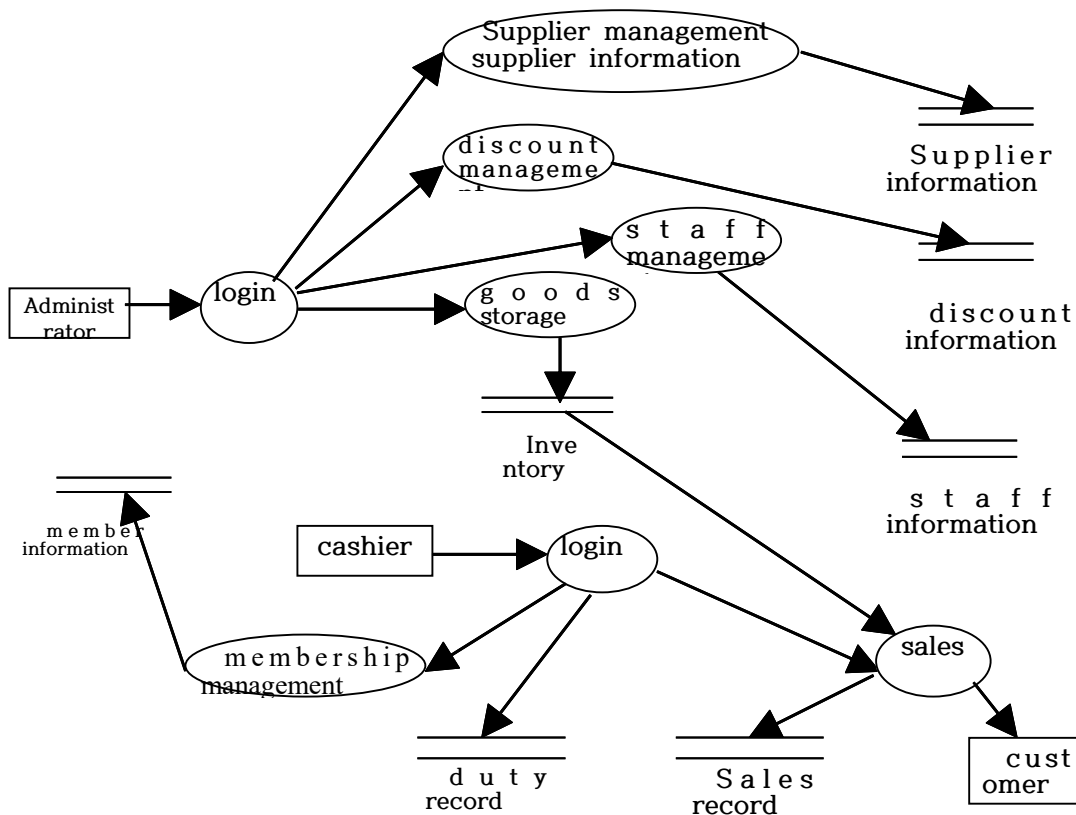
Since it is developed for regular supermarkets and not for online shopping, the C/S development model is used. This facilitates easier migration and does not require the complex server environment configuration needed by the B/S model. It allows for quick operations and rapid response times. The non-networked environment helps ensure security.

2. Class diagram

- Draw a class diagram.
- Describe each class in detail (attributes, methods, others) (table type).
- 12pt, 160%.

2.1 상속의 관점





Class Relationship Descriptions:

Customer has a direct relationship with ShoppingCart as customers interact with their shopping carts.

PromotionManager and Product have a bidirectional interaction for managing product promotions.

Product is managed by InventoryManager.

ShoppingCart is used by multiple classes: customers add products to the shopping cart, promotion management applies discounts to products in the cart, and the shopping cart information is used during checkout by CashRegister.

CashRegister processes SalesRecord to record sales.

Cashier operates the CashRegister and processes payments through PaymentProcessor.

ReportGenerator generates reports from SalesRecord and InventoryManager, and is managed by UserManager.

Class Relationships:

1.Cashier Class

Dependencies:

Relies on CashRegister to process sales.

Relies on PaymentProcessor to handle payments.

2.CashRegister Class

Relationships:

Aggregates ShoppingCart, indicating that the cash register operates the shopping cart.

Depends on SalesRecord to record each sale.

3.Product Class

Relationships:

Can be managed by InventoryManager.

Is an element within ShoppingCart.

4.InventoryManager Class

Relationships:

Manages Product.

5.SalesRecord Class

Relationships:

Records products sold from Product.

Aggregates into CashRegister.

6.Customer Class

Relationships:

Relies on ShoppingCart for shopping.

Can receive promotions through PromotionManager.

7.PromotionManager Class

Relationships:

Provides discounts to Product.

8.ShoppingCart Class

Relationships:

Contains multiple Product objects.

Depends on PromotionManager to apply promotions.

9.PaymentProcessor Class

Relationships:

Processes payments from Customer.

Called by Cashier to complete transactions.

10.ReportGenerator Class

Relationships:

Generates reports from SalesRecord and InventoryManager.

11.UserManager Class

Relationships:

Manages Cashier and other system users.

2.2 기능의 관점

Front-end Module for Cashiers

The front-end module is designed for cashiers and primarily includes checkout functionality and membership management functionality.

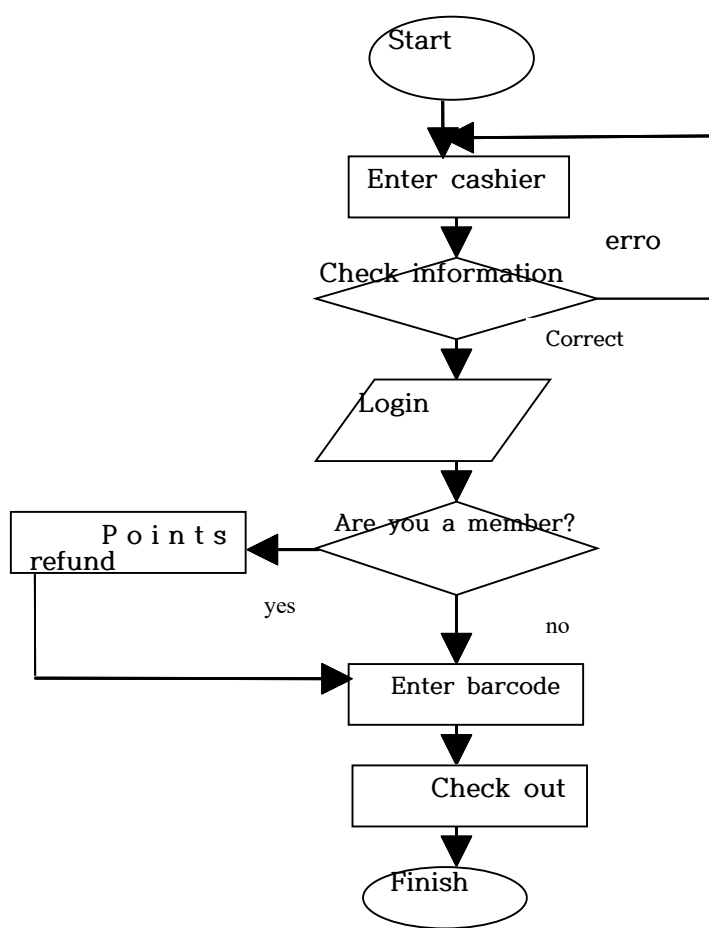
(1) Checkout Functionality

Since this is a supermarket billing system, the checkout functionality is naturally the core feature of the system as it is the most frequently used function. The highlight of the designed checkout feature in this system is the inclusion of a membership mechanism, which encourages members to make more purchases to accumulate points. The specific mechanism is as follows: for members, every 100 points accumulated through purchases will return 5 currency units to the member. By incorporating the points mechanism, the system aims to encourage repeat purchases.

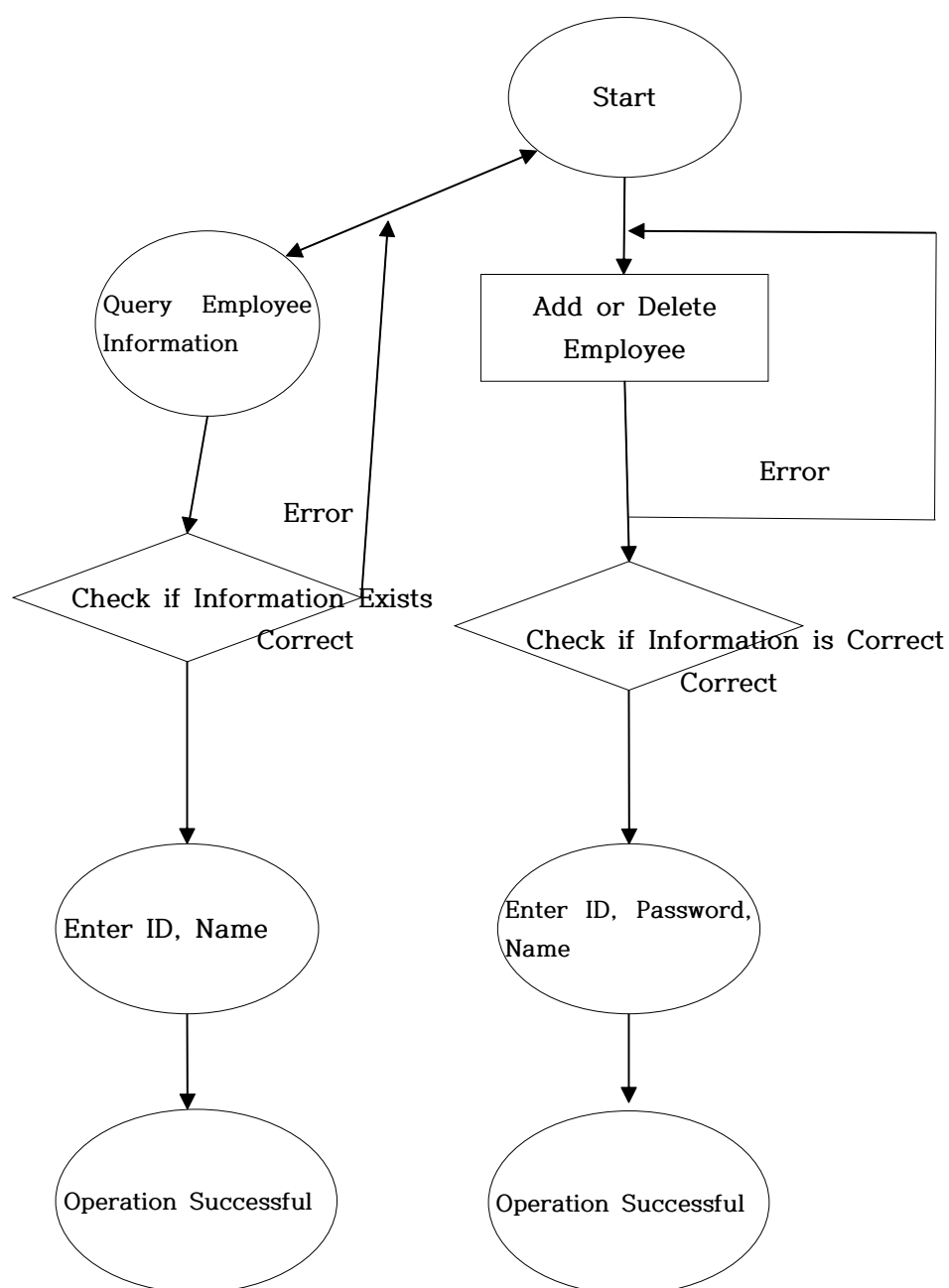
(2) Membership Management

To facilitate membership services, the membership management feature is placed at the front end. The primary functions of membership management are to query, add, and delete members. When a new member is added, their points are automatically initialized to 0. As members make more purchases, their points gradually increase. When the points reach the system's set minimum return threshold (100 points in this system), 5 currency units are returned to the member for every 100 points, and the points that have been returned are deducted from the member's total points.

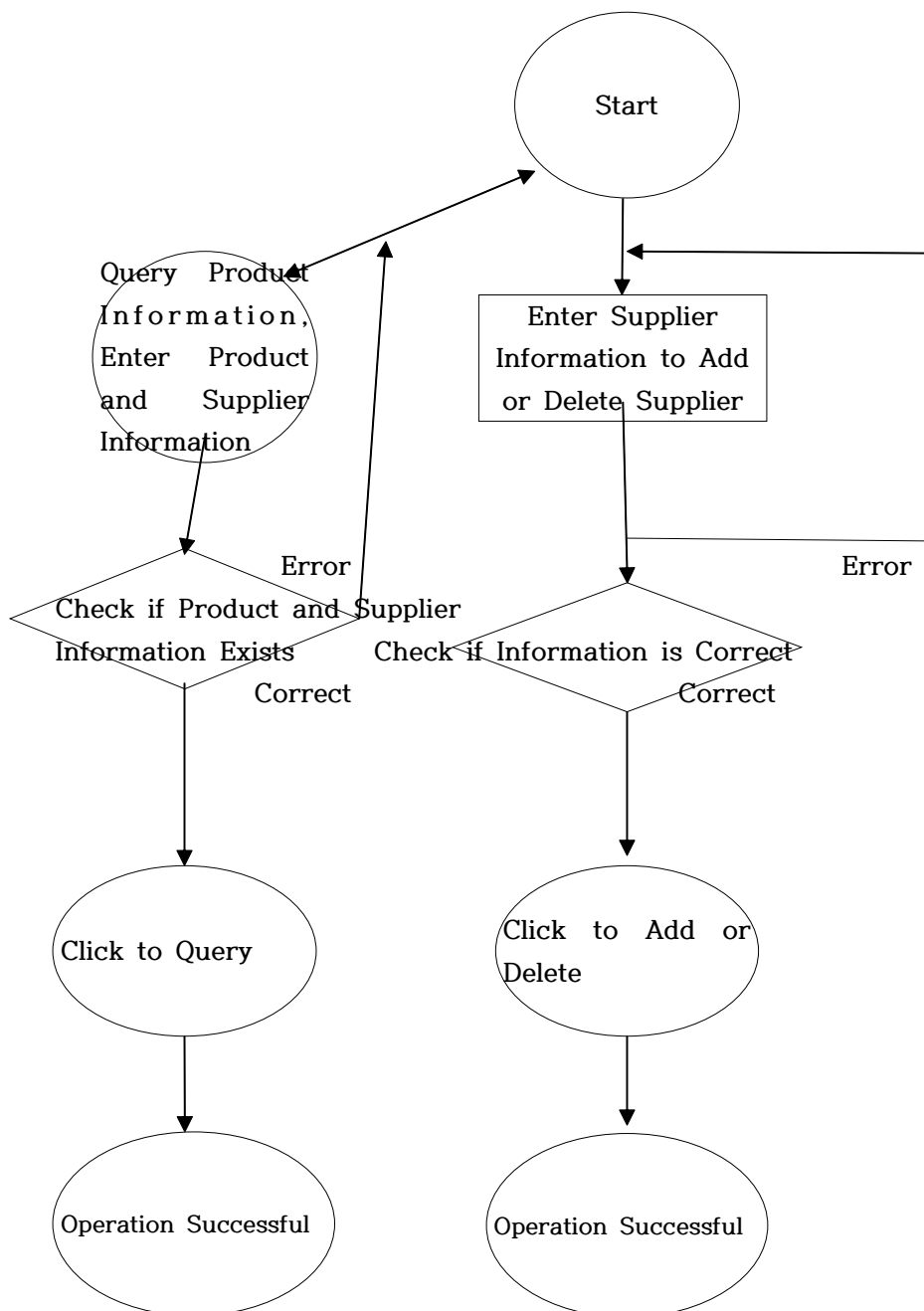
As shown in the diagram:



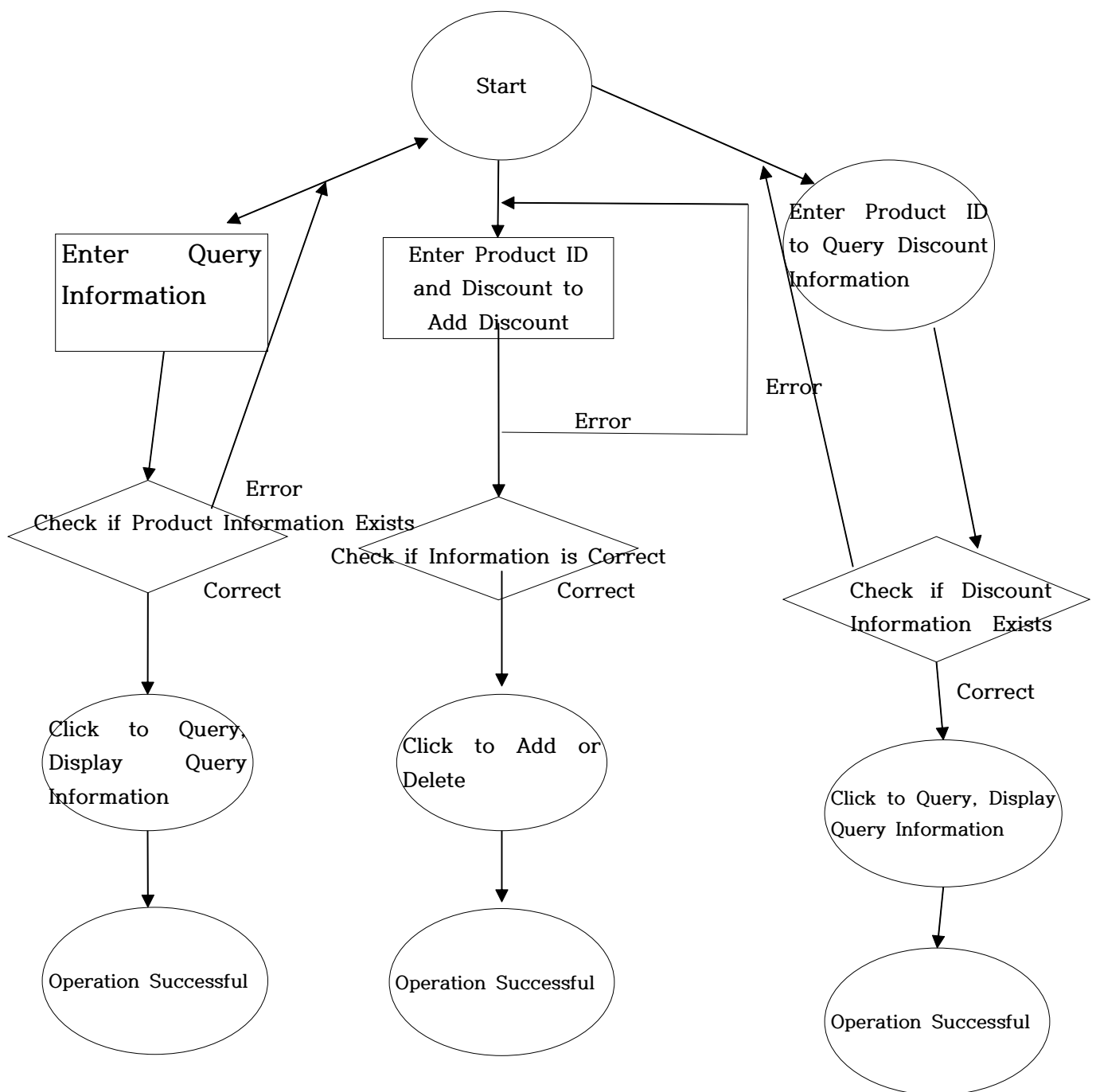
(1)."Employee Management: This section implements the management of supermarket staff (cashiers), including the addition, deletion, and querying of cashier information. Through this function, the administrator sets up exclusive usernames and passwords for each cashier, who then uses this information to log in and conduct sales. Additionally, there is a unique feature: querying duty information. Every time a cashier logs into the system, the system automatically records their login date and username. This log-like feature enables the administrator to have a clear view of the duty information for each day. This facilitates rewarding and penalizing cashiers; for instance, if a particular day's sales revenue is exceptionally high, the cashier on duty for that day can be rewarded, or if theft occurs on a certain day, the cashier on duty can be penalized. It's worth noting that employee information is linked to duty information. If a staff member is deleted, the duty information associated with that staff member will also be deleted accordingly. As shown in the figure."



(2)."Product Management: Products are the main profit drivers for supermarkets, so managing products is a key functionality of this system. This functionality primarily includes entering product information, i.e., inventorying products. It also allows for querying products, such as by ID, supplier, or name. Product management also involves managing supplier information: each product must have a supplier, and only by first entering the corresponding supplier information can the products supplied by that supplier be inventoried. In addition, product information can be deleted, but since products are related to various other information such as sales information and discount information, when deleting products, the administrator will be warned. If deletion is confirmed, sales and discount information will also be deleted accordingly."

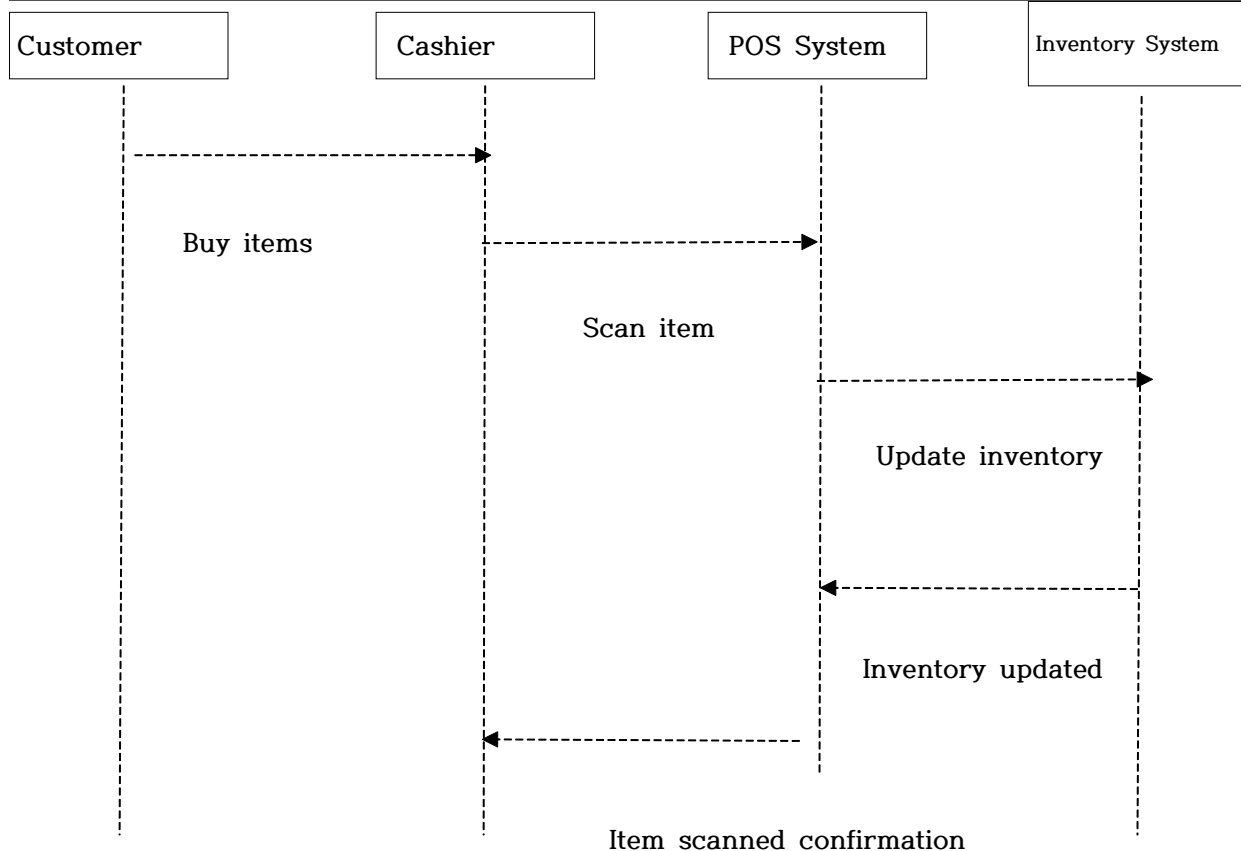


(3)."Sales Management: Sales records the total quantity of a particular product sold within a day. Through queries, the administrator can analyze the sales trends of a product over a certain period or compare the sales of various products within a day, providing reference for the formulation of operational strategies. The discount management feature allows the administrator to conveniently apply discounts to a particular product. Of course, discounts can also be cancelled at any time by simply deleting the discount information. The discount set here will be automatically taken into account during the checkout process."



3. Sequence diagram

- Draw sequence diagrams for the whole functions of your system (this is related to the use cases you made in the Conceptualization phase).
- Explain each sequence diagram.
- 12pt, 160%.



"Product Scanning and Inventory Update

This sequence diagram illustrates the process of a cashier scanning a product and updating the inventory system after a customer selects a product:

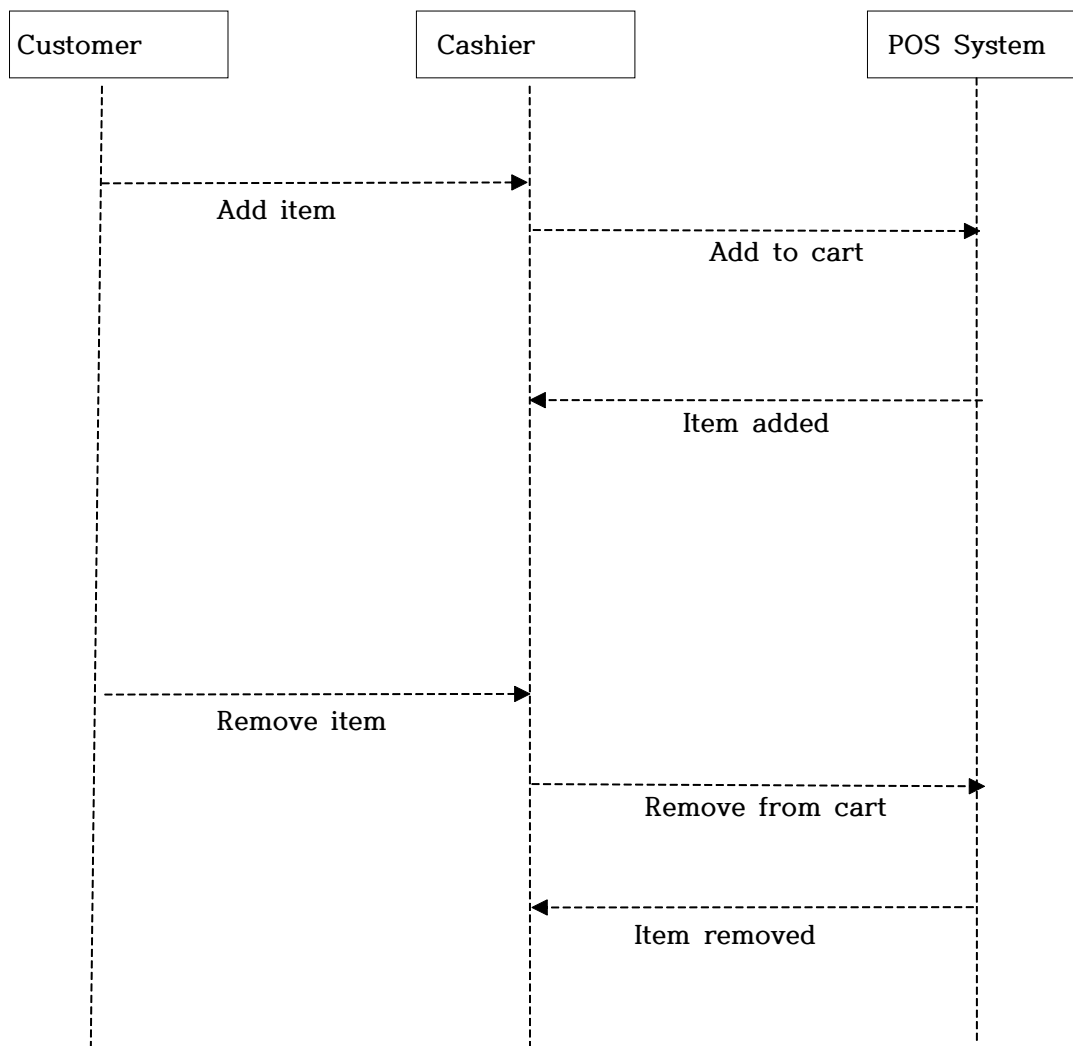
The customer expresses the desire to purchase a product to the cashier.

The cashier scans the barcode of the product.

The POS system receives the scanning information and sends a request to the inventory system for inventory update.

The inventory system updates the inventory quantity of the corresponding product and confirms to the POS system.

The POS system notifies the cashier that the product has been successfully scanned.



"Adding and Removing Items from Shopping Cart

This sequence diagram illustrates the process of a customer adding or removing items from the shopping cart before checkout:

The customer requests the cashier to add a specific item to the shopping cart.

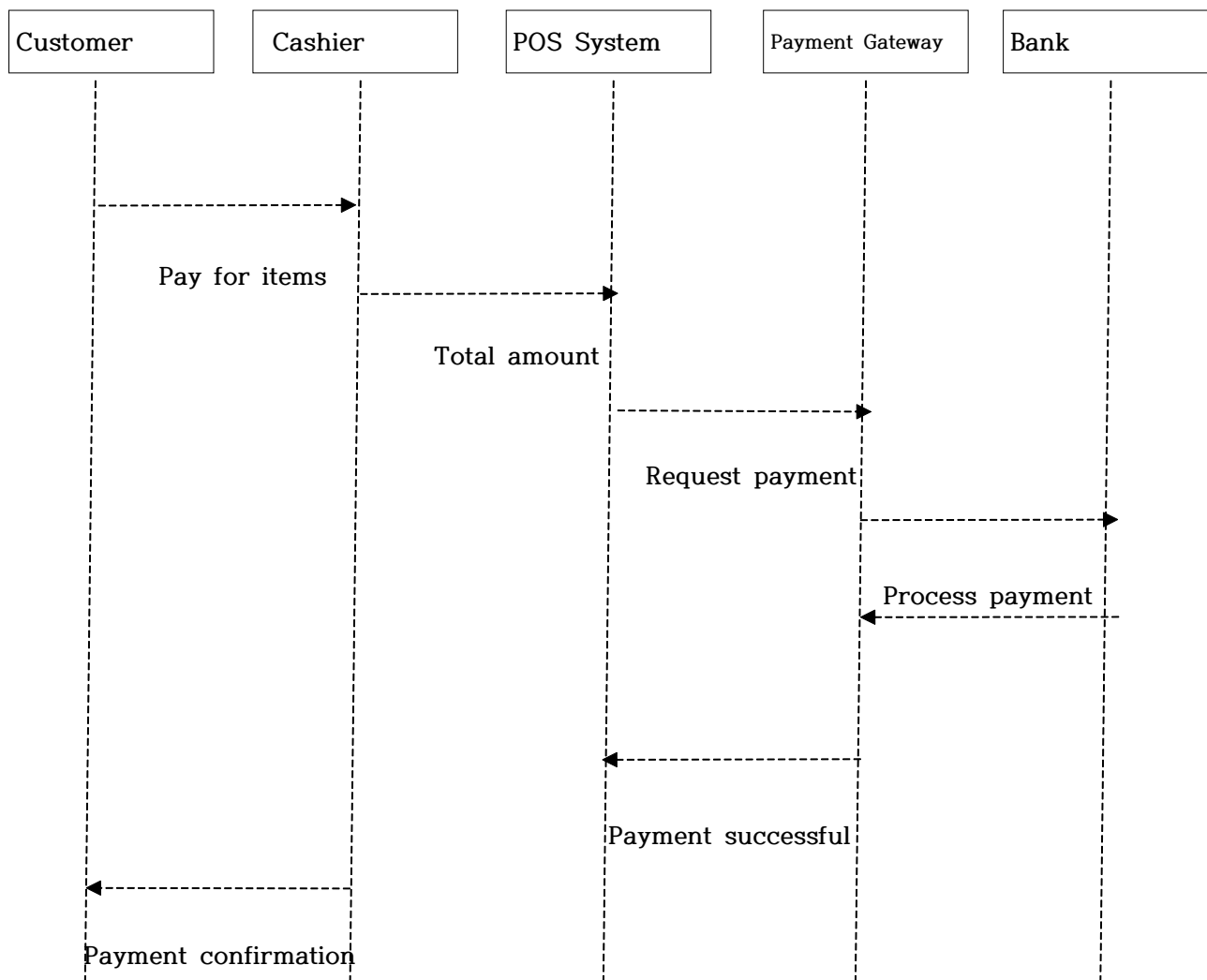
The cashier adds the item to the shopping cart through the POS system.

The POS system confirms that the item has been successfully added.

The customer requests the cashier to remove a specific item from the shopping cart.

The cashier removes the item from the shopping cart through the POS system.

The POS system confirms that the item has been successfully removed."



"Checkout and Payment Processing

This sequence diagram illustrates the process of a customer preparing to checkout and completing the payment:

The customer informs the cashier of their intention to pay for the purchased items.

The cashier calculates the total amount using the POS system.

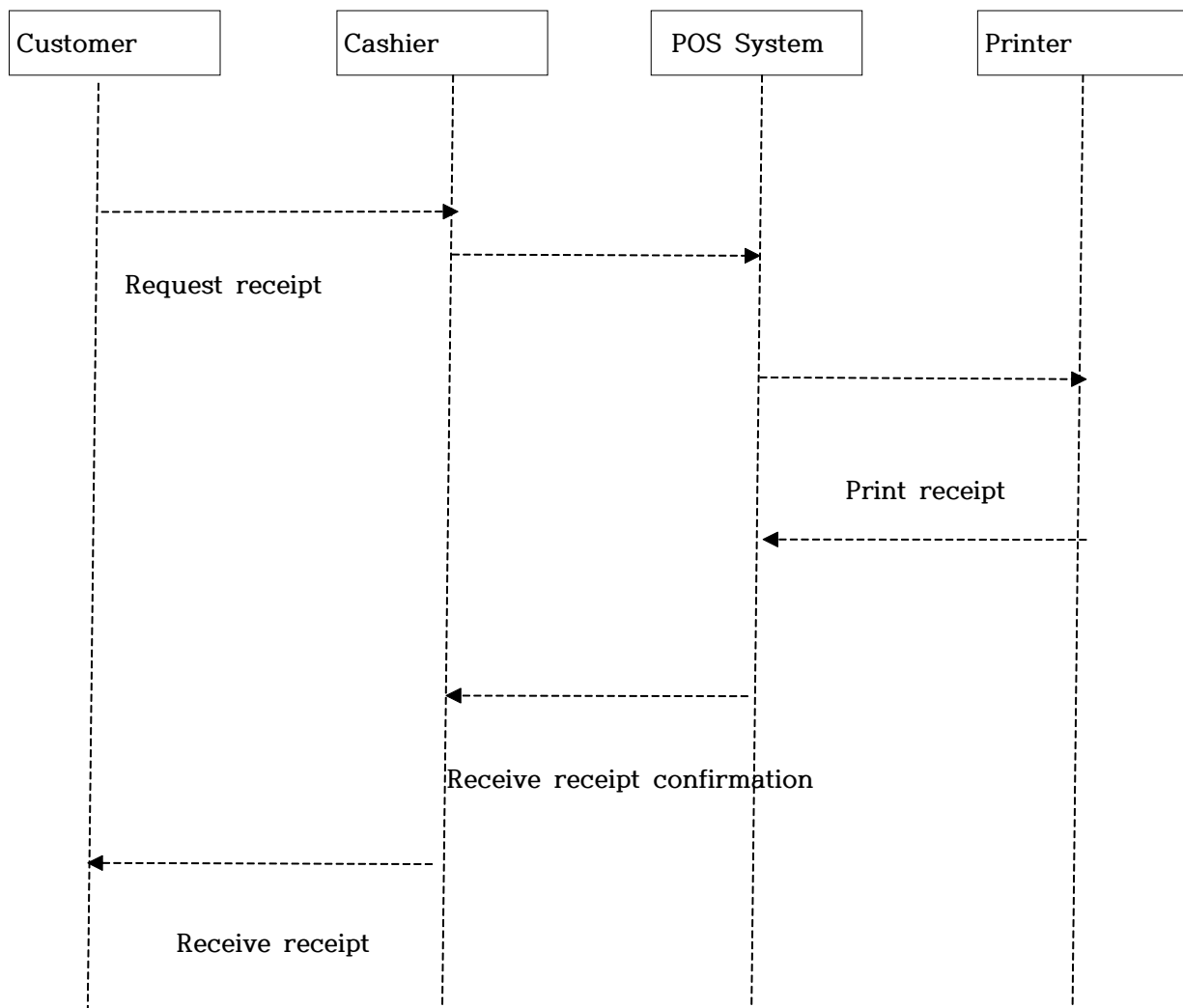
The POS system sends a payment request to the payment gateway.

The payment gateway forwards the payment request to the bank for processing.

The bank processes the payment and confirms the payment success.

The payment gateway notifies the POS system of the successful payment.

The POS system notifies the cashier that the payment has been completed."



"Printing Receipt

This sequence diagram illustrates the process of the cashier printing a receipt after the payment is completed:

The customer requests to print the receipt.

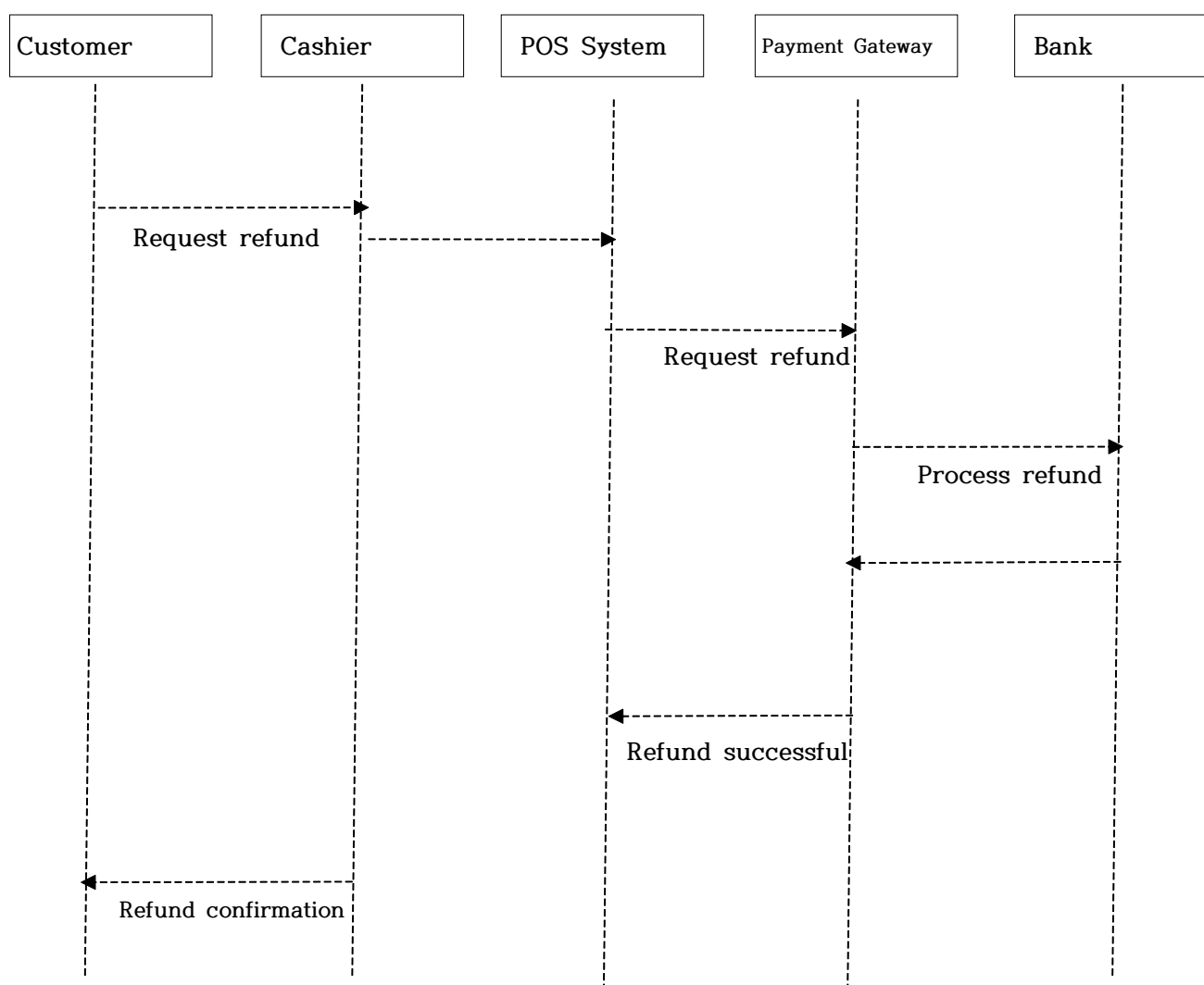
The cashier sends a print receipt request through the POS system.

The POS system sends a print command to the printer.

The printer prints the receipt.

The POS system confirms that the receipt has been printed.

The cashier hands the receipt to the customer."



"This sequence diagram illustrates the process of a customer requesting a refund and completing the refund:

The customer requests a refund.

The cashier submits the refund request through the POS system.

The POS system sends the refund request to the payment gateway.

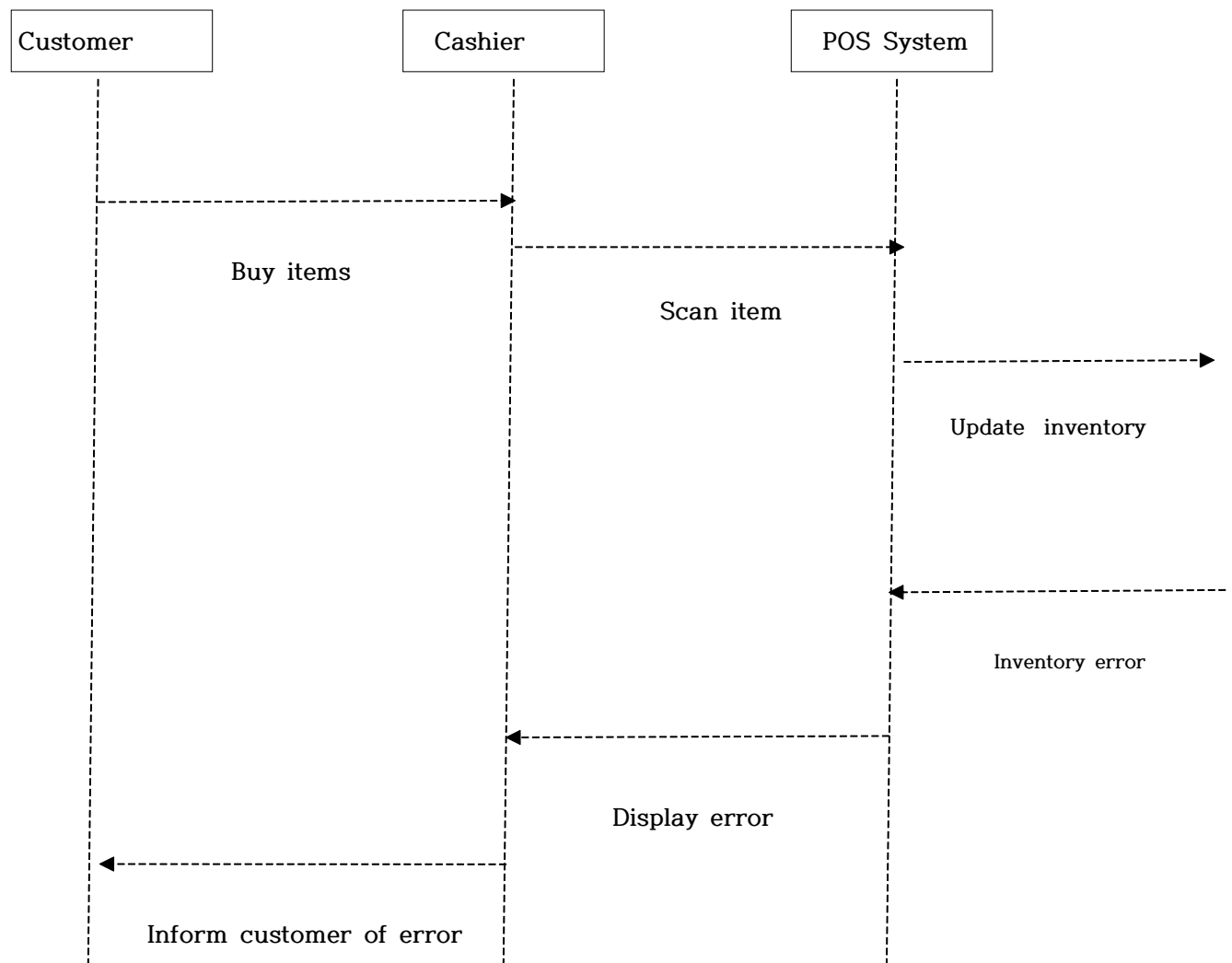
The payment gateway forwards the refund request to the bank for processing.

The bank processes the refund and confirms the refund success.

The payment gateway notifies the POS system of the successful refund.

The POS system notifies the cashier that the refund has been completed.

The cashier informs the customer that the refund has been successful."



"Error Handling in the System

This sequence diagram illustrates the process of handling errors that occur during scanning of items or updating inventory:

The customer informs the cashier of their intent to purchase items.

The cashier scans the barcode of the item.

The POS system attempts to update the inventory but encounters an error.

The inventory system returns an error message.

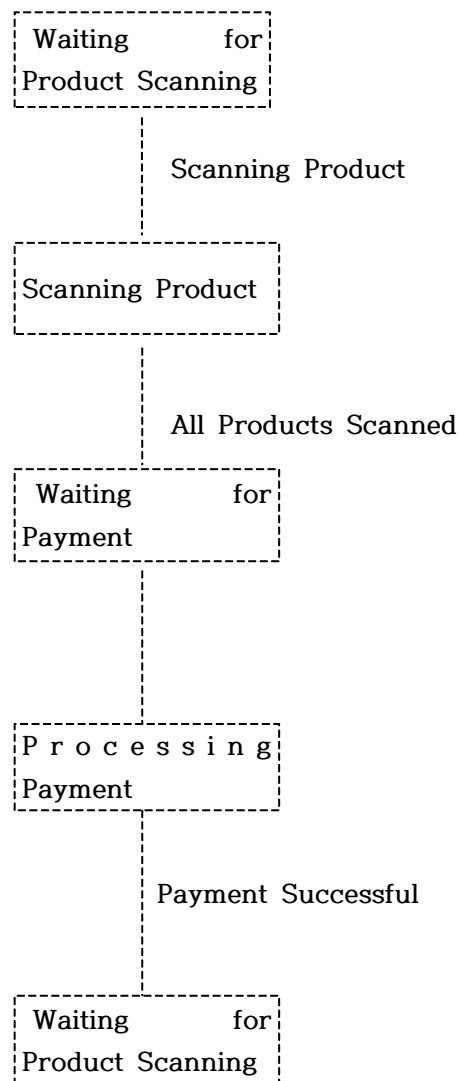
The POS system displays the error message to the cashier.

The cashier notifies the customer of the error and attempts to resolve it or seeks technical support.

In summary, these sequence diagrams depict the primary flows of different use cases in the cashier system, involving user interaction, data transmission, and page switching, among other aspects. Through these diagrams, we can see the connections and interactions between various modules of the system, understanding how different functional modules collaborate to ensure the system's normal operation and optimize user experience. In all scenarios, the POS system serves as the core module, interacting frequently with other system components to perform key functions such as item scanning, inventory updating, checkout payment, receipt printing, handling refunds, and system error handling."

4. State machine diagram

- Draw state machine diagrams for the client and the server system.
- Explain each state machine diagram.
- 12pt, 160%.



From "Waiting for Item Scanning" to "Scanning Item" state: A product barcode is scanned.

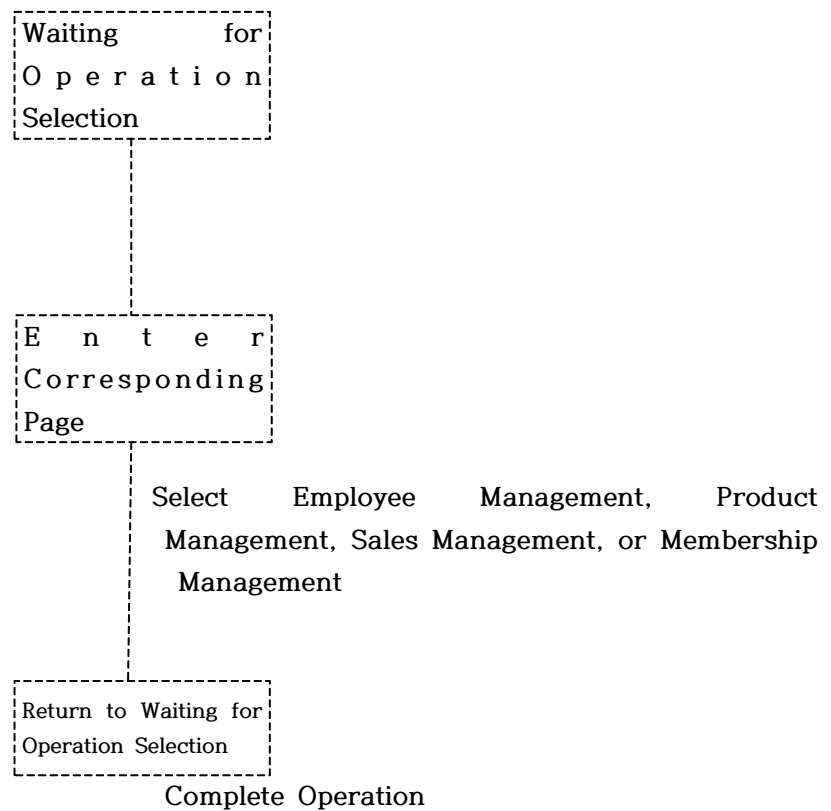
From "Scanning Item" to "Waiting for Payment" state: All items have been scanned.

From "Waiting for Payment" to "Processing Payment" state.

From "Processing Payment" to "Refund" state: Payment is successful.

From "Refund" state back to "Waiting for Item Scanning" state: Refund completed.

From "Scanning Item" state back to "Waiting for Item Scanning" state: Transaction is cancelled or items are rescanned.



This state machine diagram represents a comprehensive system, including four main functional modules: Employee Management, Product Management, Sales Management, and Member Management. Here's an explanation of the state machine diagram:

Waiting for Operation Selection: The initial state of the system, indicating that the system is ready to wait for user operation selection.

Selecting Employee Management, Product Management, Sales Management, or Member Management: In this state, users can choose to enter different functional modules. For example, Employee Management is used for managing staff, Product Management is for managing product information, Sales Management is for viewing sales data, and Member Management is for managing member information.

Entering Respective Pages: Based on the user's selection, the system will enter the corresponding functional module page for users to operate and manage.

Completing Operation: After completing the operation of the current functional module, users can choose to return to the Waiting for Operation Selection state to continue other operations or exit the system.

This state machine diagram succinctly describes the switching relationship between the functional modules of the entire system. Users can choose different functional modules for operation as needed, thereby completing various management and business processes.

5. Implementation requirements

- Describe operating environments to implement your system.
- 12pt, 160%.

For the database, we choose MySQL. MySQL has a small footprint yet ensures high speed, and it is also very cost-effective. As an open-source database, it offers great stability, making it very suitable for developing small desktop-level software applications.

The database connection will use JDBC.

For the development environment, we choose Eclipse. First of all, Eclipse is the development tool I am most familiar with. Additionally, Eclipse has a vast array of useful plugins, and as a long-standing preferred development tool for Java, it is extremely suitable.

For developing the application's interface in Java, the development toolkit used is Swing.

6. Glossary

- Specifically describe all of the terms used in this documents.
- 12pt, 160%.

Terms	Annotations
Supermarket Cashier System	Name of this software system.
Administrator	Manage users.
Cashier	Responsible for using the cashier system to check out customers.
Member	Enjoy member discount benefits.
Purchasing Management Function	Categorize and record incoming goods into the database.
Sales Management Function	Count and manage goods for sale, remove related product information from the database.
Inventory Management System	Provides the function to query goods in inventory, receive and process data from the purchasing management module, receive and process data from the shipping management module.
Ordering System	Query existing orders.
Sales Inquiry System	Product sales inquiry.

7. References

- Describe all of your references (book, paper, technical report etc).
- 12pt, 160%.

1. Yan Huipiaowu. (2012). Definition of software testing and introduction to common software testing methods. Retrieved from <https://wenku.baidu.com/view/09764f785acfa1c7aa00cc3a.html>
2. Eckel, B. (2006). Thinking in Java [M]. Upper Saddle River, New Jersey, USA: Prentice Hall.
3. Zhang Xiaoxiang. (2006). Inside Deep Java Web Development—Core Fundamentals [M]. Beijing: Electronics Industry Press.
4. Eckel, B. (2003). Thinking in Java Programming [M]. Mechanical Industry Press.
5. Wang Weidong. (2005). Introduction to Java Programming (1st Edition). Jilin Electronic Press.
6. Zhou Yahui. (2005). Practical Development of JAVA Database System Projects. Science Press.
7. Zhang Xiaoxiang. (2006). Inside Deep Java Web Development—Core Fundamentals [M]. Beijing: Electronics Industry Press.
8. Flanagan. (2002). Java Technology Manual [M]. China Electric Power Press.
9. Java Database Programmin Guide : <https://www.runoob.com/java/java-mysql-connect.html>
10. JDBC Tutorial : <https://www.w3school.cn/java/java-jdbc.html>