

# Object-Oriented Design: Public Coffee Vending Machine System.

## 1. System Requirements

### Functional Requirements:

1. **Allow users to select from a variety of beverages:** The system should present a menu of available beverages for users to choose from.
2. **Display product information and prices:** Each beverage should have associated details, such as ingredients and pricing.
3. **Accept different forms of payment:** The machine should support cash, cards, and mobile payments.
4. **Dispense the selected beverage:** Once a payment is processed, the chosen drink is dispensed.
5. **Provide change when necessary:** The system should return any excess money if the payment exceeds the price.
6. **Allow for refilling of ingredients and products:** Staff should be able to replenish supplies easily.
7. **Track inventory levels:** The machine should monitor current stock and alert when supplies are low.
8. **Generate sales reports:** The system should produce reports for sales and inventory management.

### Non-Functional Requirements:

1. **User-friendly interface with clear instructions:** The UI should be intuitive and easy to navigate.
2. **Quick response time (under 2 seconds) for user interactions:** Interactions should be swift to enhance user experience.
3. **High reliability:** The machine should have minimal downtime.
4. **Secure payment processing:** All transactions should be securely handled to protect user information.
5. **Energy-efficient operation:** The system should consume minimal power.
6. **Easy maintenance and cleaning:** The machine should be designed for straightforward upkeep.
7. **Compliance with food safety regulations:** The machine should meet all applicable health and safety standards.
8. **Accessibility features for users with disabilities:** The system should be accessible to all users.

## **2. Use Cases**

### **1. Purchase a Beverage**

Actor: Customer

Description: Customer selects a beverage, pays for it, and receives the drink.

Steps:

- a. Customer approaches the machine.
- b. Views available options and prices.
- c. Selects desired beverage.
- d. Chooses payment method.
- e. Makes payment.
- f. Machine dispenses beverage.
- g. Customer collects beverage and any change (if applicable).

### **2. Refill Machine**

Actor: Staff Member

Description: Staff member refills ingredients and restocks products.

Steps:

- a. Staff member authenticates access to machine.
- b. Opens machine compartments.
- c. Refills ingredients (coffee beans, milk, etc.).
- d. Restocks cups and other supplies.
- e. Closes and secures machine.
- f. Logs refill in system.

### **3. Generate Reports**

Actor: Administrator

Description: Administrator generates and views sales and inventory reports.

Steps:

- a. Administrator logs into system interface.
- b. Selects report type (sales, inventory, etc.).
- c. Specifies time period for report.
- d. System generates report.
- e. Administrator views or exports report.

### **4. Perform Maintenance**

Actor: Technician

Description: Technician performs routine maintenance or repairs.

Steps:

- a. Technician authenticates access to machine.
- b. Runs diagnostic tests.
- c. Performs necessary maintenance or repairs.
- d. Updates maintenance log in system.
- e. Ensures machine is operational.

### **5. Update Prices or Menu**

Actor: Administrator

Description: Administrator updates product prices or menu options.

Steps:

- a. Administrator logs into system interface.
- b. Selects menu or price management option.
- c. Makes necessary changes to prices or available products.
- d. Saves changes.
- e. System updates display and internal records.

### 3. Objects, Classes, and Relationships

#### Classes and Relationships

##### 1. VendingMachine

- **Attributes:** id, location, status
- **Methods:** dispenseBeverage(), acceptPayment(), generateReport()
- **Relationships:**
  - has-a **Inventory**
  - has-a **PaymentSystem**
  - has-a **UserInterface**
  - has-many **Product**
  - has-a **MaintenanceLog**

##### 2. Product

- **Attributes:** id, name, price, quantity
- **Methods:** updateQuantity(), getPrice()

##### 3. Inventory

- **Attributes:** stockLevels
- **Methods:** checkStock(), updateStock(), alertLowStock()
- **Relationships:**
  - manages **Product**

##### 4. PaymentSystem

- **Attributes:** supportedMethods
- **Methods:** processPayment(), giveChange(), validatePayment()

##### 5. UserInterface

- **Attributes:** currentDisplay
- **Methods:** showOptions(), getUserInput(), displayMessage()

##### 6. Report

- **Attributes:** type, dateRange, data
- **Methods:** generateSalesReport(), generateInventoryReport()

## 7. **MaintenanceLog**

- **Attributes:** entries
- **Methods:** addEntry(), getLastMaintenance()

## 8. **Administrator**

- **Attributes:** id, name, accessLevel
- **Methods:** updatePrices(), viewReports()
- **Relationships:**
  - generates **Report**

## 9. **Technician**

- **Attributes:** id, name, specialization
- **Methods:** performMaintenance(), updateLog()
- **Relationships:**
  - updates **MaintenanceLog**

# 4.CRC (Class-Responsibility-Collaboration) cards

## 1. VendingMachine

- Responsibilities:
  - Dispense beverages
  - Accept payments
  - Generate reports
- Collaborations:
  - Collaborates with Product to dispense the selected beverage.
  - Uses PaymentSystem to process payments.
  - Interfaces with UserInterface to display options and messages.
  - Works with Inventory to check stock levels.
  - Generates reports with Report.

## 2. Product

- Responsibilities:

- Store product information (ID, name, price, quantity)
- Update product quantity
- Provide pricing information
- Collaborations:
  - Managed by Inventory for stock updates and alerts.
  - Used by VendingMachine to dispense beverages and check prices.

### 3. Inventory

- Responsibilities:
  - Manage stock levels of products
  - Update and check stock quantities
  - Alert when stock is low
- Collaborations:
  - Maintains a relationship with Product to manage quantities.
  - Provides stock information to VendingMachine.

### 4. PaymentSystem

- Responsibilities:
  - Process different payment methods
  - Validate payments
  - Provide change when necessary
- Collaborations:
  - Interacts with VendingMachine for payment processing.
  - Works with UserInterface to handle payment options and feedback.

### 5. UserInterface

- Responsibilities:
  - Display product options and prices
  - Get user input
  - Display messages to users
- Collaborations:
  - Interfaces with VendingMachine for displaying and updating information.
  - Works with PaymentSystem to process user input related to payments.

## 6. Report

- Responsibilities:
  - Generate sales and inventory reports
- Collaborations:
  - Utilized by Administrator to view and export reports.
  - Retrieves data from VendingMachine and Inventory.

## 7. MaintenanceLog

- Responsibilities:
  - Record maintenance activities
  - Provide access to the latest maintenance entry
- Collaborations:
  - Updated by Technician during maintenance.
  - Associated with VendingMachine for operational logs.

## 8. Administrator

- Responsibilities:
  - Update product prices and menu options
  - Generate and view reports
- Collaborations:
  - Uses Report to generate reports.
  - Interfaces with VendingMachine to update prices and menu.

## 9. Technician

- Responsibilities:
  - Perform maintenance and repairs
  - Update maintenance logs
- Collaborations:
  - Updates MaintenanceLog during maintenance.
  - Ensures VendingMachine is operational after maintenance.

## 4. Class Diagram

