COFFEE VENDING MACHINE

By Keerthana. H [174]

Date:11/03/2025

Functional Requirements:

1.The machine should offer three Flavors: Espresso, Cappuccino, and Latte.

2.The user should be able to select a coffee flavour.

3.The machine should check if sufficient ingredients are available.

4.If ingredients are available and payment is made, dispense the selected coffee.

5.If ingredients are insufficient, notify the user.

6.The machine should allow the admin to refill ingredients.

7.The machine should accept UPI payment before dispensing coffee.

8.Payment should be processed and authenticated by a third-party system.

9.A user can drink a maximum of 4 coffees per day.

10.AI-Powered Recommendations: The machine suggests a coffee based on most preferred once.

11.Dynamic Ingredient Adjustments: Users can customize sugar/milk levels.

12. Allow users to choose their coffee temperature (Hot, Warm, Extra Hot).

13.Gamification & Loyalty System: Users earn points per coffee, leading to rewards.

14.Allow user to get one free coffee per 200 reward points.

15.Smart Payment System: Supports multiple payment options beyond UPI (e.g., wallets, NFC).

16. Allow users to cancel the order before payment is processed.

Use Cases:

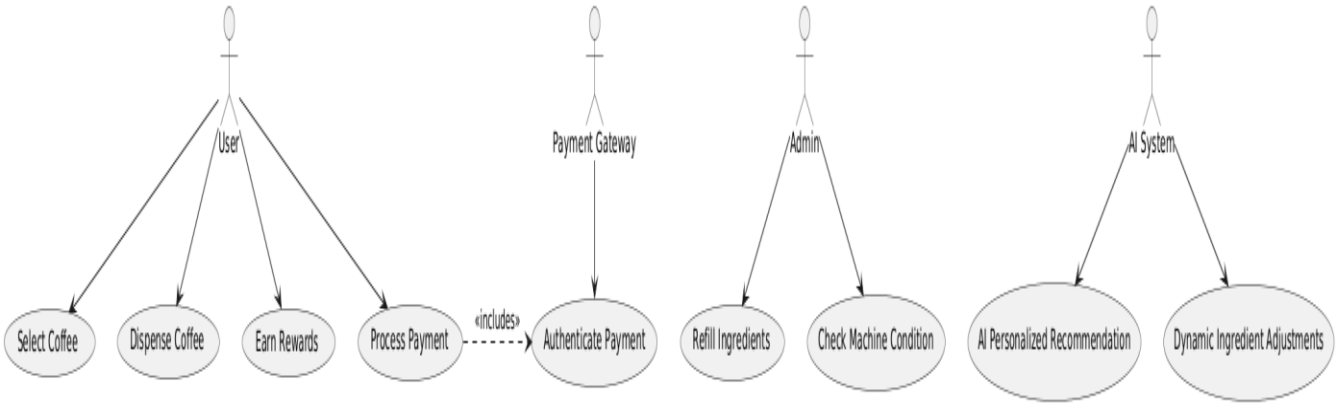
1. **Select Coffee** – User selects a coffee type.
2. **Authenticate Payment** – Payment Gateway validates payment.
3. **Process Payment** – Payment Gateway completes the transaction.
4. **Dispense Coffee** – Coffee machine prepares and serves coffee.
5. **Refill Ingredients** – Admin refills stock when needed.

6. **Track machine condition** – Admin can System tracks the condition of the machine.
7. **Recommend Coffee** – AI system suggests coffee based on past choices.
8. **Earn Rewards** – Users earn points per coffee purchase.
9. **Dynamic Ingredient Adjustments:** Users can customize sugar/milk levels.

Actors:

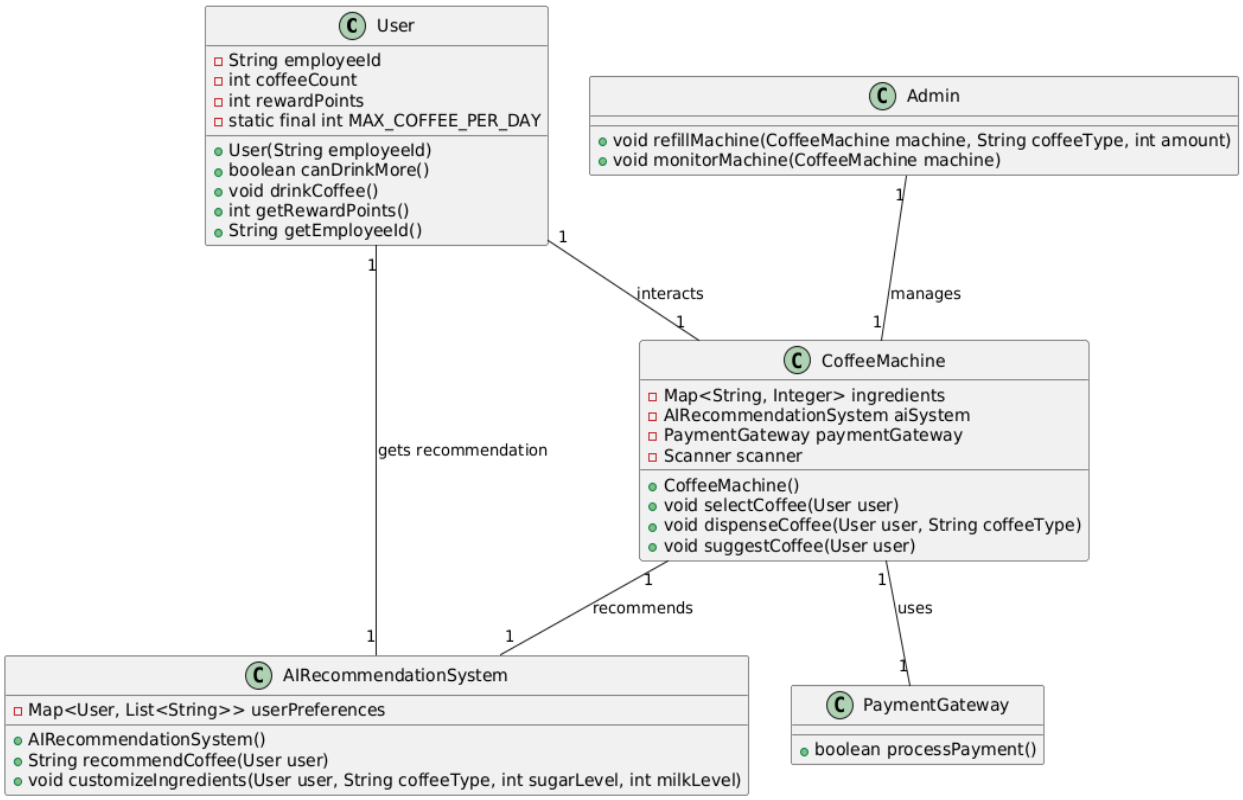
1. **User** – Selects coffee, makes payment, and receives coffee. Can also view history and track consumption.
2. **Admin** – Refills ingredients, monitors machine status, and receives low stock alerts.
3. **Payment Gateway** – Authenticates and processes payments.
4. **AI System** – Suggests coffee based on user history and consumption trends.

USECASE DIAGRAM

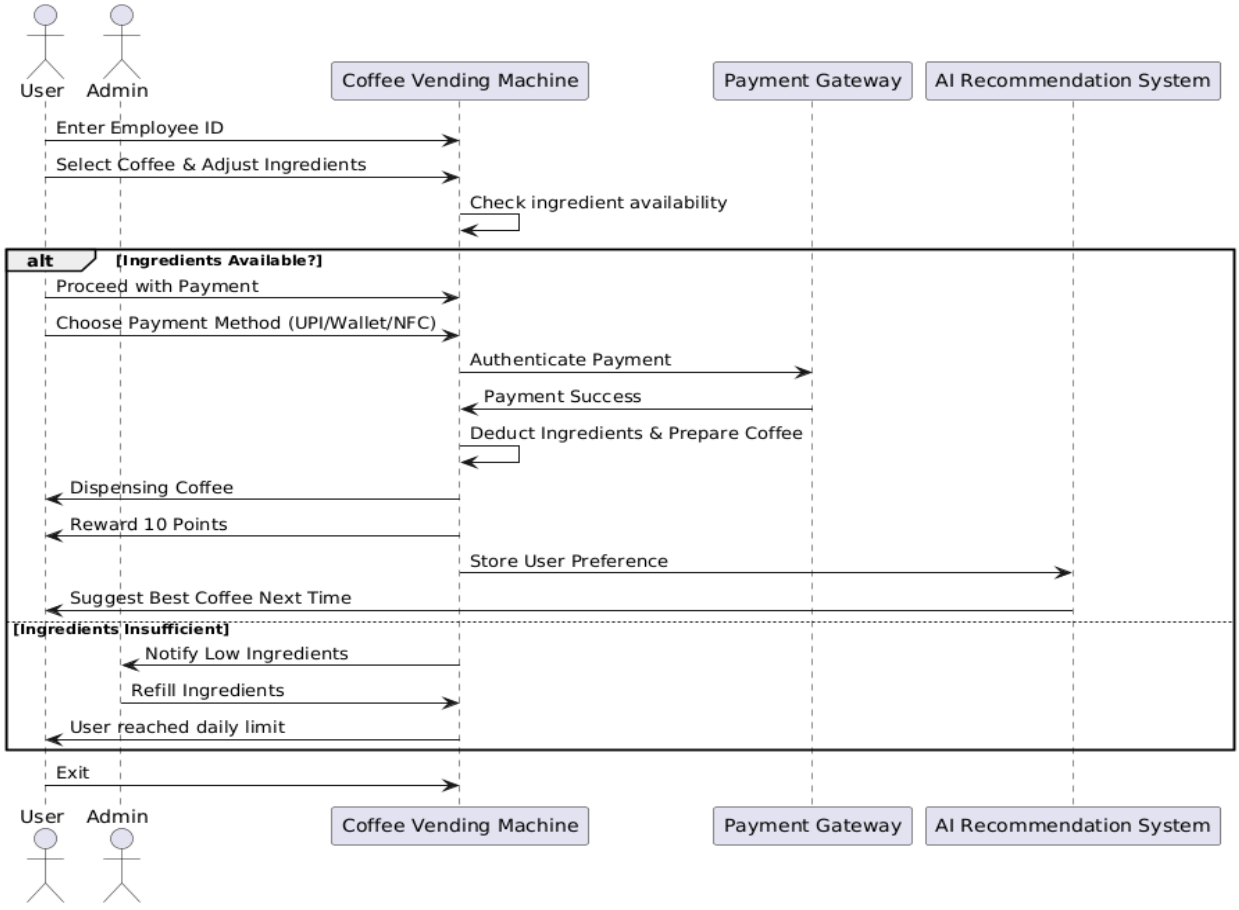


- **User:** Selects coffee, makes payments, receives coffee.
- **Admin:** Refills ingredients, manages machine maintenance.
- **Payment Gateway:** Authenticates and processes payments.
- **AI System:** Provides recommendations based on most preferred once and allow users to customize sugar/milk levels, and give rewards.

CLASS DIAGRAM



SEQUENCE DIAGRAM



The Coffee Vending Machine (CVM) allows users to enter their employee ID, select a coffee, and adjust sugar level, milk level and the temperature level. It checks ingredient availability, if sufficient the user proceeds with payment via UPI, Wallet, or NFC and allows user the to cancel their order before Payment. The Payment Gateway authenticates and processes the transaction, after which the machine deducts ingredients, prepares coffee, and dispenses it while rewarding the user with 10 points per each cup and once the points reached 200, that user can get one free coffee for 200 points. The AI System tracks preferences to suggest the best coffee next time. If ingredients are low, the admin is notified to refill them, ensuring uninterrupted service.

CODE

```
import java.util.*;

class User {

private String employeeId;

private int coffeeCount;

private int rewardPoints;

private static final int MAX_COFFEE_PER_DAY = 4;

private static final int FREE_COFFEE_POINTS = 200;


public User(String employeeId) {

this.employeeId = employeeId;

this.coffeeCount = 0;

this.rewardPoints = 0;

}


public boolean canDrinkMore() {

return coffeeCount < MAX_COFFEE_PER_DAY;

}


public void drinkCoffee(boolean isFree) {

if (canDrinkMore()) {

coffeeCount++;

if (!isFree) {

rewardPoints += 10; // Earn 10 points per coffee

} else {

rewardPoints -= FREE_COFFEE_POINTS; // Deduct points for free coffee
```

```
System.out.println("You used 200 reward points for a free coffee!");
}
} else {
System.out.println("Daily limit reached! Cannot drink more coffee today.");
}}

public boolean hasFreeCoffee() {
return rewardPoints >= FREE_COFFEE_POINTS;
}

public int getRewardPoints() {
return rewardPoints;
}

public String getEmployeeId() {
return employeeId;
}}

class CoffeeMachine {
private Map<String, Integer> ingredients;
private AIRecommendationSystem aiSystem;
private PaymentGateway paymentGateway;
private Scanner scanner;

public CoffeeMachine() {
ingredients = new HashMap<>();
ingredients.put("Espresso", 5);
ingredients.put("Cappuccino", 5);
ingredients.put("Latte", 5);
aiSystem = new AIRecommendationSystem();
paymentGateway = new PaymentGateway();
scanner = new Scanner(System.in);
```

```
}

public void selectCoffee(User user) {
if (!user.canDrinkMore()) {
System.out.println("User reached daily limit.");
return;
}

System.out.println("Select Coffee Flavor: 1. Espresso  2. Cappuccino  3. Latte");
int choice = scanner.nextInt();
String coffeeType = switch (choice) {
case 1 -> "Espresso";
case 2 -> "Cappuccino";
case 3 -> "Latte";
default -> {
System.out.println("Invalid selection! Defaulting to Espresso.");
yield "Espresso";
}};

System.out.println("Select Sugar Level (1-Less, 2-Normal, 3-More): ");
int sugarLevel = scanner.nextInt();
System.out.println("Select Milk Level (1-Less, 2-Normal, 3-More): ");
int milkLevel = scanner.nextInt();

System.out.println("Select Temperature (1-Hot, 2-Warm, 3-Extra Hot): ");
int tempChoice = scanner.nextInt();
String temperature = switch (tempChoice) {
case 1 -> "Hot";
case 2 -> "Warm";
 case 3 -> "Extra Hot";
 default -> "Hot";
 };
```

```
aiSystem.customizeIngredients(user, coffeeType, sugarLevel, milkLevel, temperature);

if (!ingredients.containsKey(coffeeType) || ingredients.get(coffeeType) <= 0) {
System.out.println("Insufficient ingredients for " + coffeeType);
System.out.println("Notifying Admin to Refill Ingredients...");
return;
}

boolean isFree = user.hasFreeCoffee();
if (isFree) {
System.out.println("You have enough reward points for a free coffee! Proceeding...");
} else {
System.out.println("Proceeding to Payment... (Press 0 to Cancel)");
int cancelChoice = scanner.nextInt();
if (cancelChoice == 0) {
System.out.println("Transaction Cancelled.");
return;
}

if (!paymentGateway.processPayment()) {
System.out.println("Payment failed!");
return;
}}

dispenseCoffee(user, coffeeType, isFree);
}

private void dispenseCoffee(User user, String coffeeType, boolean isFree) {
ingredients.put(coffeeType, ingredients.get(coffeeType) - 1);
user.drinkCoffee(isFree);
System.out.println("Dispensing " + coffeeType + " for Employee ID: " + user.getEmployeeId());
```

```
System.out.println("Reward Earned: " + (isFree ? 0 : 10) + " Points | Total Rewards: " +
user.getRewardPoints());
}

public void suggestCoffee(User user) {
String recommended = aiSystem.recommendCoffee(user);
System.out.println("Based on your preferences, we recommend: " + recommended);
}}

class Admin {
public void refillMachine(CoffeeMachine machine, String coffeeType, int amount) {
System.out.println("Admin refilled " + amount + " units of " + coffeeType);
}

public void monitorMachine(CoffeeMachine machine) {
System.out.println("Machine Status: Ingredients are being monitored.");
}}

class PaymentGateway {
public boolean processPayment() {
System.out.println("Payment Successful!");
return true; // Simulating successful payment
}}

class AIRecommendationSystem {
private Map<User, List<String>> userPreferences;

public AIRecommendationSystem() {
userPreferences = new HashMap<>();
}

public String recommendCoffee(User user) {
List<String> preferences = userPreferences.getOrDefault(user, new ArrayList<>());
```

```
if (!preferences.isEmpty()) {
return preferences.get(new Random().nextInt(preferences.size())); // Suggest a preferred coffee
 }
 return "Espresso"; // Default recommendation
 }

public void customizeIngredients(User user, String coffeeType, int sugarLevel, int milkLevel, String temperature) {
System.out.println(user.getEmployeeId() + " customized " + coffeeType +
" | Sugar Level: " + sugarLevel + " | Milk Level: " + milkLevel + " | Temperature: " + temperature);
}}

public class CoffeeVendingSystem {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
CoffeeMachine machine = new CoffeeMachine();

System.out.println("Enter Employee ID:");
String empId = scanner.nextLine();
User user = new User(empId);

while (true) {
System.out.println("\n MENU ");
System.out.println("1. Select Coffee");
System.out.println("2. View Reward Points");
System.out.println("3. Get Coffee Recommendation");
System.out.println("4. Exit");

int option = scanner.nextInt();
switch (option) {
case 1 -> machine.selectCoffee(user);
case 2 -> System.out.println("Total Reward Points: " + user.getRewardPoints());
case 3 -> machine.suggestCoffee(user);
```

```
case 4 -> {
System.out.println("Thank you for using the Coffee Vending Machine!");
return;
}
default -> System.out.println("Invalid Option! Try Again.");
}}}}
```

```
C:\MyJava>java CoffeeVendingSystem
Enter Employee ID:
174

 MENU
1. Select Coffee
2. View Reward Points
3. Get Coffee Recommendation
4. Exit
1
Select Coffee Flavor: 1. Espresso  2. Cappuccino  3. Latte
3
Select Sugar Level (1-Less, 2-Normal, 3-More):
2
Select Milk Level (1-Less, 2-Normal, 3-More):
3
Select Temperature (1-Hot, 2-Warm, 3-Extra Hot):
3
174 customized Latte | Sugar Level: 2 | Milk Level: 3 | Temperature: Extra Hot
Proceeding to Payment... (Press 0 to Cancel)
1
Payment Successful!
Dispensing Latte for Employee ID: 174
Reward Earned: 10 Points | Total Rewards: 10

 MENU
1. Select Coffee
2. View Reward Points
3. Get Coffee Recommendation
4. Exit
1
Select Coffee Flavor: 1. Espresso  2. Cappuccino  3. Latte
1
Select Sugar Level (1-Less, 2-Normal, 3-More):
3
Select Milk Level (1-Less, 2-Normal, 3-More):
3
Select Temperature (1-Hot, 2-Warm, 3-Extra Hot):
3
174 customized Espresso | Sugar Level: 3 | Milk Level: 3 | Temperature: Extra Hot
Proceeding to Payment... (Press 0 to Cancel)
1
Payment Successful!
Dispensing Espresso for Employee ID: 174
Reward Earned: 10 Points | Total Rewards: 20
```

```
Reward Earned: 10 Points | Total Rewards: 20

 MENU
1. Select Coffee
2. View Reward Points
3. Get Coffee Recommendation
4. Exit
1
Select Coffee Flavor: 1. Espresso  2. Cappuccino  3. Latte
2
Select Sugar Level (1-Less, 2-Normal, 3-More):
2
Select Milk Level (1-Less, 2-Normal, 3-More):
2
Select Temperature (1-Hot, 2-Warm, 3-Extra Hot):
2
174 customized Cappuccino | Sugar Level: 2 | Milk Level: 2 | Temperature: Warm
Proceeding to Payment... (Press 0 to Cancel)
1
Payment Successful!
Dispensing Cappuccino for Employee ID: 174
Reward Earned: 10 Points | Total Rewards: 30

 MENU
1. Select Coffee
2. View Reward Points
3. Get Coffee Recommendation
4. Exit
2
Total Reward Points: 30

 MENU
1. Select Coffee
2. View Reward Points
3. Get Coffee Recommendation
4. Exit
3
Based on your preferences, we recommend: Espresso

 MENU
1. Select Coffee
2. View Reward Points
3. Get Coffee Recommendation
4. Exit
1
Select Coffee Flavor: 1. Espresso  2. Cappuccino  3. Latte
1
Select Sugar Level (1-Less, 2-Normal, 3-More):
3
Select Milk Level (1-Less, 2-Normal, 3-More):
3
Select Temperature (1-Hot, 2-Warm, 3-Extra Hot):
1
174 customized Espresso | Sugar Level: 3 | Milk Level: 3 | Temperature: Hot
Proceeding to Payment... (Press 0 to Cancel)
0
```

```
174 customized Espresso | Sugar Level: 3 | Milk Level: 3 | Temperature: Hot
Proceeding to Payment... (Press 0 to Cancel)
0
Transaction Cancelled.

 MENU
1. Select Coffee
2. View Reward Points
3. Get Coffee Recommendation
4. Exit
1
Select Coffee Flavor: 1. Espresso  2. Cappuccino  3. Latte
3
Select Sugar Level (1-Less, 2-Normal, 3-More):
2
Select Milk Level (1-Less, 2-Normal, 3-More):
2
Select Temperature (1-Hot, 2-Warm, 3-Extra Hot):
2
174 customized Latte | Sugar Level: 2 | Milk Level: 2 | Temperature: Warm
Proceeding to Payment... (Press 0 to Cancel)
1
Payment Successful!
Dispensing Latte for Employee ID: 174
Reward Earned: 10 Points | Total Rewards: 40

 MENU
1. Select Coffee
2. View Reward Points
3. Get Coffee Recommendation
4. Exit
1
User reached daily limit.

 MENU
1. Select Coffee
2. View Reward Points
3. Get Coffee Recommendation
4. Exit
4
Thank you for using the Coffee Vending Machine!
```