Kennesaw State University
College of Computing and Software Engineering

COFFEE & CODE

Requirements

SWE 3313

Professor:
Jeff Adkisson

Group 6:
Elliot Larez
Garrett Heffner
Michael Butler
Sahan Reddy

Table of Contents

Requirements Definition

1. Screens

ID	Name	Description	Priority
1.1	*Main* **Screen**	(Winforms) A screen with **large buttons** that works as a starting point of the program.	**Must Have**
1.1.1	*Order Drink* **Button** on *Main* **Screen**	A button on the *Main Screen* that takes the user to the *Order Drink* Screen as an **anonymous** customer	**Must Have**
1.1.2	*Customer List* **Button** on *Main* **Screen**	A button on the *Main Screen* that takes the user to the *Customer List* Screen	Need to Have
1.1.3	*Management Tools* **Button** on *Main* **Screen**	A button on the *Main Screen* that takes the user to the *Management Tools* Screen	**Must Have**
1.2	*Customer List* **Screen**	(Winforms) A screen with a **table** listing the customer records - **Anonymous** customer record at the top - **Non-Anonymous** customers' records must be sorted by last name, first name, and phone number. Table must include their reward points	**Must Have**
1.2.1	*Cancel* **Button** on *Customer List* **Screen**	A button on the *Customer List Screen* that takes the user to the previous screen *(Main Screen)*	Need to Have
1.2.2	*Order Drink* **Buttons** on *Customer List* **Screen**	A set of button on the *Customer List Screen* that takes the user to the *Order Drink Screen* - There should be a button next to each customer - The customer next to the clicked button should be the one used on the next screen	**Must Have**
1.2.3	*Add New Customer* **Button** on *Customer List* **Screen**	A button on the *Customer List Screen* that takes the user to the *Add New Customer Screen*	Need to Have
1.2.4	*Edit Customer* **Buttons** on *Customer List* **Screen**	A button next to each customer on the *Customer List Screen* that takes the user to the *Add New Customer Screen* with the existing user details already filled in.	Nice to Have

1.2.5	*Sort type* **Button** on *Customer List* **Screen**	A button on the *Customer List Screen* that changes the order of the list (ascending to descending)	Nice to Have
1.3	*Add Customer* **Screen**	(Winforms) Program a screen with a **form** to receive a new customer record. The inputs will include - First name, Last name, Phone Number	Need to Have
1.3.1	*First Name* **Text-Field** on *Add Customer* **Screen**	A text field on the *Add Customer Screen* that takes the user input and store it in a local First Name variable - This field is mandatory	Need to Have
1.3.2	*Last Name* **Text-Field** on *Add Customer* **Screen**	A text field on the *Add Customer Screen* that takes the user input and store it in a local Last Name variable - This field is mandatory	Need to Have
1.3.3	*Phone Number* **Text-Field** on *Add Customer* **Screen**	A text field on the *Add Customer Screen* that takes the user input and store it in a local Phone Number variable - This field is mandatory - The input should be validated ● Has 10 Characters in total ● Doesn't appear in the customer list	Need to Have
1.3.4	*Ready* **Condition** on *Add Customer* **Screen**	A function on the *Add Customer Screen* that takes the user to the *Order Drink Screen* when all the Fields are valid	Nice to Have
1.3.5	*Cancel* **Button** on *Add Customer List* **Screen**	A button on the *Add Customer Screen* that takes the user to the initial screen *(Main Screen)*	Need to Have
1.4	*Order Drink* **Screen**	(Winforms) Program a screen with **two panes** for taking the order from the user. The screen uses a customer that can be **anonymous** if opened from the *Main Screen*, or **non-anonymous** if opened from the *Customer List Screen* or *Add Customer Screen* - The left side: Drink Creator - The right side: Drinks added to the order, subtotal, tax, and total	**Must Have**
1.4.1	*Drink Creator* ***Pane*** on *Order Drink* **Screen**	The left side of the screen displays the menu with all the items and their POSSIBLE customizations. Includes a button to send the selected option to the next pane and reset the left pane.	**Must Have**

1.4.2	*Added Drinks* ***Pane*** on *Order Drink* **Screen**	The right side of the screen displays the added drinks, including the price and other details of each other and the total. - Each drink listed on the right pane has a button that can be pressed to remove the drink from order	**Must Have**
1.4.3	*Cancel* **Button** on *Order Drink* **Screen**	A button on the *Order Drink* that takes the user to the initial screen *(Main Screen)* - The customer's order should be deleted when the button is pressed	**Must Have**
1.4.5	*Proceed to Payment* **Button** on *Order Drink* **Screen**	A button on the *Order Drink* that takes the user to the next screen *(Payment Screen)* - This button should be activated only when at least 1 item is included in the order	**Must to Have**
1.5	Payment **Screen**	(Winforms) Program a screen with a **form** to receive customer payment type	**Must Have**
1.5.1	Current Order **Pane** on *Payment* **Screen**	The customer's order is shown, including drinks, customizations, and prices - The price of each item is shown along with the subtotal, tax, and total of the order	Need to Have
1.5.2	*Credit Card Information* **Pane** on *Payment* **Screen**	The user can input credit card information as a payment type, including number and expiration date, and press the *pay with credit card* button - Proceeds to the receipt screen only when the credit card number is validated - Add 10 reward points to **non-anonymous clients** for every $1 spent. If the amount is not an integer, round down.	**Must Have**
1.5.3	*Reward Points* **Pane** on *Payment* **Screen**	The user can use reward points as a payment type if they are not **anonymous** customers. - 10 Reward Points are worth $1 - Proceeds to the receipt screen only if there are enough reward points for the order - Subtracts the reward points necessary to pay the order.	Need to Have
1.5.4	*Cancel* **Button** on *Payment* **Screen**	A button on the *Payment Screen* that takes the user to the initial screen *(Main Screen)* - Customer's order should be deleted when the button is pressed	**Must Have**

1.5.5	*Back to Order* **Button** on *Payment* **Screen**	A button on the *Payment Screen* that takes the user to the previous screen without erasing the order *(Take Order Screen)*	Need to Have
1.6	*Receipt* **Screen**	(Winforms) A screen that shows a **pane** with the customer receipt - Displays each item in the order and its customizations, the price of each item, the subtotal, tax, and total of the order - If the customer paid with a credit card, display the last 4 digits of the credit card - If the customer used rewards points, display the number of rewards points spent as well as their total rewards points remaining	**Must Have**
1.6.1	*Return* **Button** on *Customer Receipt* **Screen**	A button on the *Receipt Screen* that takes the user to the initial screen *(Main Screen)*	Need to Have
1.7	Management Screen	Create a screen where users can generate a report of sales data	Need to Have
1.7.1	*Generate Report* **Button** on *Management Screen*	Create a button that opens an Excel document of sales data as a CSV file	Need to Have
1.7.2	*Return* **Button** on *Management Screen*	A button on the *Management Screen* that takes the user to the initial screen *(Main Screen)*	**Must Have**

2. Configuration Data

ID	Name	Description	Priority
2.1	Configuration Data Loading	The software should be able to load the configuration data when the program starts running - The loaded data should be a JSON file	**Must Have**
2.2	Menu Data Loading	The file should contain the drink menu and customizations, the tax rate, and reward points per data - The loaded data should be a JSON file	**Must Have**

3. Customer Data

ID	Name	Description	Priority
3.1	Customer List	Create a **record structure** including the **fields** + Customer ID - GUID/string + First Name - string + Last Name - string + Phone - string + Reward Point - integer	**Must Have**
3.2	Anonymous Record	Create an **"Anonymous"** customer record, which has to be located at the **top** and doesn't receive **Rewards points**. + First Name = "Anonymous" + Last Name = "Anonymous" + Phone = "000-000-0000"	**Must Have**
3.3	Customer Count	The number of customers in the rewards program is calculated and displayed on the Customer List screen.	Nice to Have

4. Sales Data

ID	Name	Description	Priority
4.1	Sales Data	The data should contain the following - Customer ID - GUID/string - Date / Time - C# DateTime - Tax - C# decimal - Subtotal - C# decimal - Total - C# decimal - Payment method - Credit Card or Rewards - List of drinks - Name - string - All customizations in a single comma-separated string - Total price	**Must Have**

5. Data Storage

ID	Name	Description	Priority
5.1	Using the JSON Files	Use the three JSON files already provided in the base project. - Use the Configuration Data for any extensions needed. This will be in the **appsetting.json** file - Use **IStorageService** to read and write the Customer and sales data to one file - Use **IStorageService** once more to read Drink Menu data from another file. Write out the drink data in the menu's associated JSON file and read it into memory when the application starts	**Must Have**
5.2	Updating the Customer and	Rewrite the data with the latest information after any new data entry	**Must Have**

	Sales Data		
5.3	Loading the Configuration File and Drink Menu	Load the data into the memory when the application starts. The application will NOT modify the data.	**Must Have**

6. User Interface

ID	Name	Description	Priority
6.1	Building the UI	The user interface MUST be built using C# Winforms - The application will run on Windows only	**Must Have**
6.2	Formatting the Prices	The prices should be in US currency and round to two decimal places	**Must Have**
6.3	Validating Input Fields	Every input field must be valid - Name: Should be first and last name and the input is a string - Credit Card: Should be 16 digits and not expired	Need to Have

7. Open Source Nuget Packages

ID	Name	Description	Priority
7.1	Newtonsoft JSON	Popular high-performance JSON framework for .NET" (Json.NET) - Reading and Writing JSON Data	**Must Have**
7.2	CSVHelper	A .NET library for reading and writing CSV files. Extremely fast, flexible, and easy to use. - Generating CSV Data File	**Must Have**
7.3	Credit Card Validator	Verifies the credit card numbers	Nice to Have

Decision Tables

Customer Management Decision Table

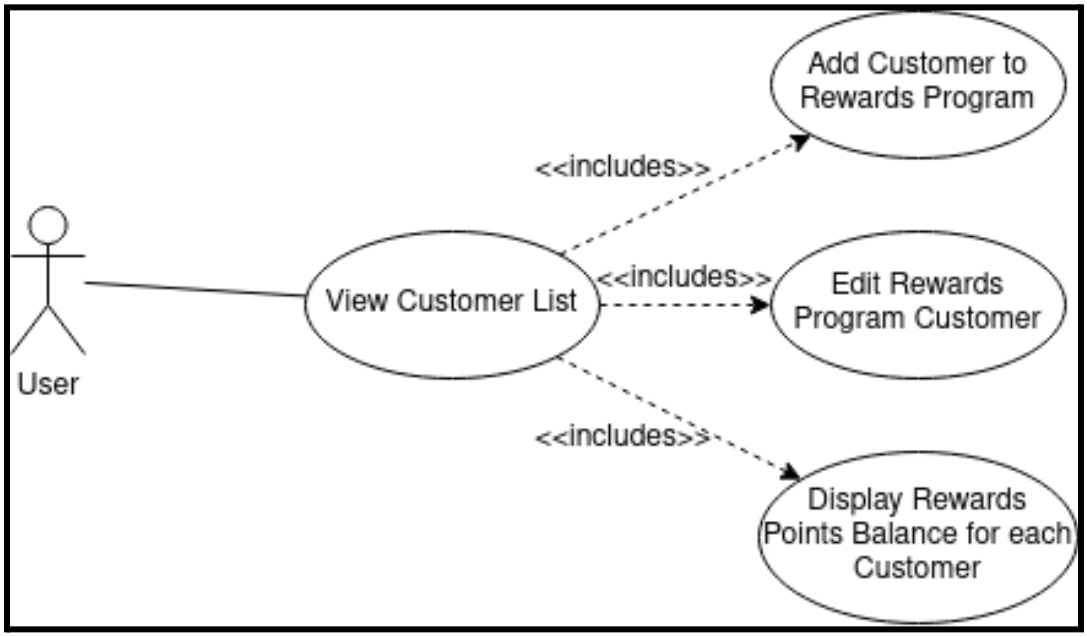
	Conditions/ Courses of Action	Rules		
Condition Stubs	Customer Type	Anonymous	Rewards	New
Action Stubs	Add customer to customer list			X
	Go to order screen with customer		X	X
	Go directly to order screen	X		

Payment Decision Table

	Conditions/ Courses of Action	Rules		
Condition Stubs	Payment type	Credit Card		Rewards
	Customer Type	Anonymous	Rewards	-
Action Stubs	Validate Credit Card Number	X	X	
	Validate Rewards Points			X
	Subtract Rewards Points			X
	Give Rewards Points		X	
	Receive Payment	X	X	X
	Generate Receipt	X	X	X

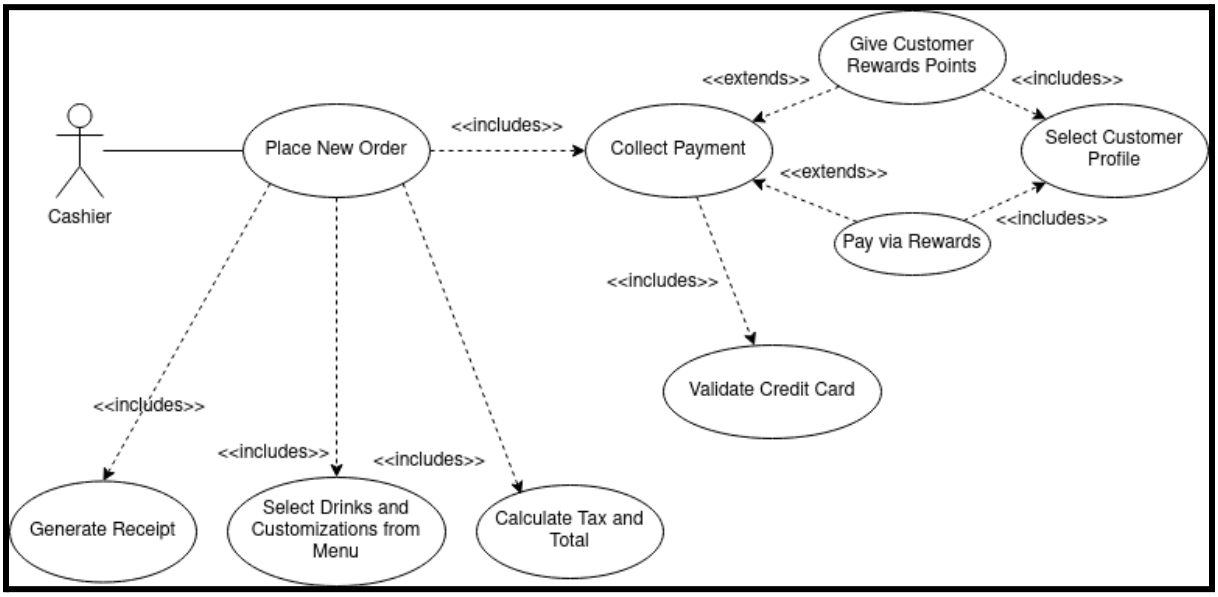
Use Case Diagrams

Customer Management



1) Flow of Events for the Customer Management Use Case
    a) Preconditions:
    The user must navigate to the customer list screen from the main menu.
    b) Main Flow:
    The customer list screen presents the user with a list of every customer in the rewards program, their rewards balance, and the "anonymous" customer (representing customers, not in the rewards program). Users can add new customers to the rewards program and edit the profiles of existing ones.
    c) Subflows:
        i) Add Customer to Rewards Program:
        The user clicks the "Add Customer" button on the customer list screen. They are presented with a new screen containing a form to input the new customer's details. After they submit this form, the customer is added to the list.
        ii) Edit Rewards Program Customer:
        In the customer list, each customer has an “Edit” button next to their entry. The user can click on this button to display a form similar to the add customer form in which they can edit the details of the user.

Ordering Flow



2) Flow of Events for the Ordering Use Case
    a) Preconditions
    For a cashier to place a new order, they first have to select a customer from the rewards list who is ordering through the customer list screen. Otherwise, if they select the place order button on the main screen, the order will be placed for an anonymous customer.

    b) Main Flow
    After the cashier presses the “Order Drink” button, they are navigated to the order drink screen. The cashier can then select drinks and customizations from the menu, calculate the tax and total, collect a payment, and generate a receipt. Before the cashier finalizes the order on the payment screen, they can cancel it at any time.

    c) Sub Flows
        i) Select Drinks and Customizations from Menu
        The cashier can add multiple drinks to the order and customize them on the order screen. All the drinks in the order will be displayed to the cashier.

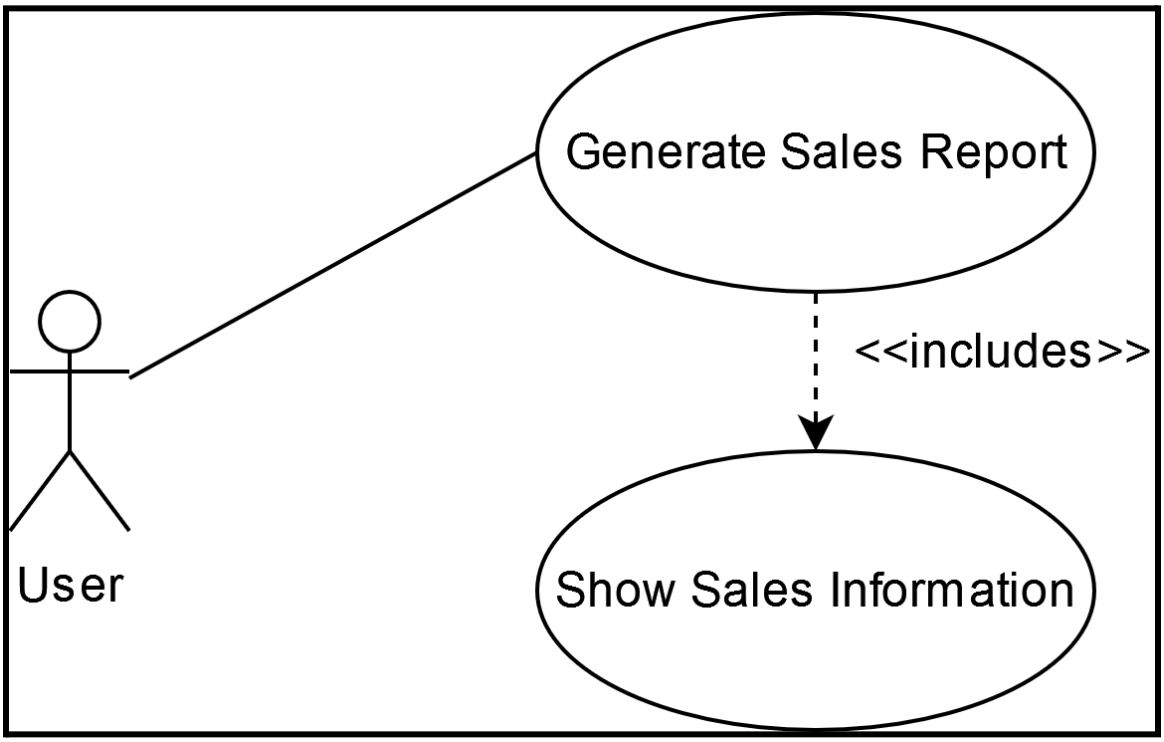
        ii) Calculate Tax and Total
        The program will add up each drink’s price and the cost of customizations for the subtotal. Then, the tax will be calculated using the rate defined in the configuration file, then added to form the total.

iii) Collect Payment
The program's payment processing will validate a credit card number inputted by the cashier and generate a receipt if the card number is valid. The program gives rewards points to customer profiles after a credit card payment according to the rate set in the configuration file. If the customer opts to pay with rewards, the program checks if their profile has enough points and subtracts points from their account.

iv) Generate Receipt
After the customer has successfully paid for their order, the program will generate a receipt that shows the following:
- The full order with all drink customizations
- The last four digits of the credit card used or the number of rewards points used
- The customer's, if not anonymous, reward points balance

d) Alternate Flows
i) If the order is empty, the cashier cannot navigate to the payment processing screen until a drink is added
ii) If the entered credit card is invalid, an error message will tell the cashier the card is invalid
iii) Suppose the selected customer does not have sufficient rewards points to pay with rewards points. In that case, an error message will tell the cashier the customer has an insufficient balance of rewards points.
iv) If the cashier goes directly to the order screen, the anonymous customer will be selected and receive no reward points.

Sales Analysis Flow



3) Flow of Events for the Generate Sales Report Use Case

a) Preconditions:
The cashier must navigate to the management screen from the main menu and select the “Generate Sales Report” button to create a report.

b) Main Flow:
This use case begins when the cashier presses a button to generate a sales report. The system creates a CSV file containing data on all of the coffee shop’s sales. The file is then opened in Excel for the cashier to view. The sales report contains specific sales information about each order.

c) Sub Flows:
   i) Show Sales Information:
   After each order is completed, the following will be entered into the sales data file. This data will be separated by order and entered into a new CSV file.
      - Customer ID
      - Date
      - Tax
      - Subtotal
      - Total
      - Payment type
      - Contents and customizations to each drink

d) Alternate Flows:
    i) If the program cannot read the sales data file, the program will tell the user an error has occurred in reading the file.