Programming III – Fall 2022

Course Project: Topic

Team Information

**Team name:** Gangnem

**Team members:**

Jayson Lartey, 2239058

Brayden Fournier, 6194481

Project Description

The application we are building is a vending machine. It will display 12 items with their respective prices. It will also display a checkout button, a clear button and a section displaying the items selected and the total. When the user clicks the clear button, it erases all snacks the user previously selected from the vending machine and erases the current total. When the user clicks the checkout button, it prompts the user for a payment method and the method is either cash or card (Credit/Debit). The user can either cancel or pay. If the user cancels, it takes them back to the vending machine. If the user pays, a receipt-like page is displayed showing the payment method, the total cost and the items bought.

Development Approach

Explain how did you prepare for the project. You can use the 5 steps of algorithmic thinking to you help build this section (you will need to elaborate on each step).

* 1. Understanding the problem.
  2. Formulating the problem.
  3. Developing the application \ algorithm.
  4. Implementing the application \algorithm.
  5. Testing.

 1. Vending machine needs items, payment methods, receipts and transaction log, figuring out inputs and outputs.

2. Building Classes and WPF as needed

3. writing an outline of what we need to code(pseudo-code, flowcharts)

4. Writing the code for each class and communicating with each other all changes made (github commits)

5. Sharing each other’s screens and running the program and debugging at the same time

OOP Design

Talk about the classes you need to create for the application and what is the purpose of each class. Include the UML class diagram in this section. The UML class diagram should include the relations between the created classes. Do not mention the WPF classes (Window, etc.)

Item Class : Class needed to initialize an item and its properties(name, price, quantity).

Vending Machine class: Class that “initializes” the vending machine… Has item instance from the item class(composition) to be used for all products in the vending machine and pricing instance to be used to get item price and item total as well general total

Pricing class: Class containing the information on item cost and item total, as well as total for all items.

Contributions

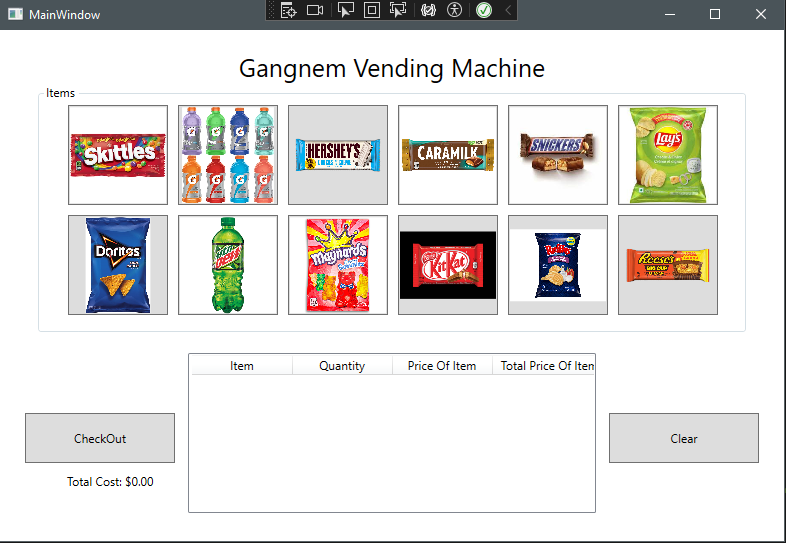
What did each team member do?

How was the work in the project divided?

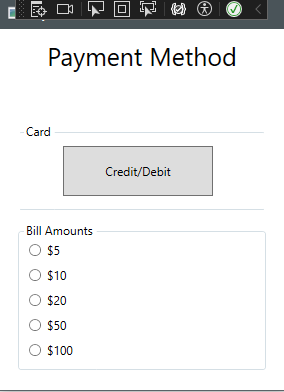
|  |  |
| --- | --- |
| Brayden | Jayson |
| -Did all wpf page design  -Did the binding and validation for wpf page code  -Did all back-end code for wpf pages, this includes all logic and validation  -created shopping cart class + logic  -created bill breakdown + logic | * Implemented Item class and Vending machine class * Implemented loading,removing, modifying and resetting quantities from file * Implemented saving transactions to file * AddProductToCart method (me and Brayden) * Image implementation (me and brayden) |

App Snapshots

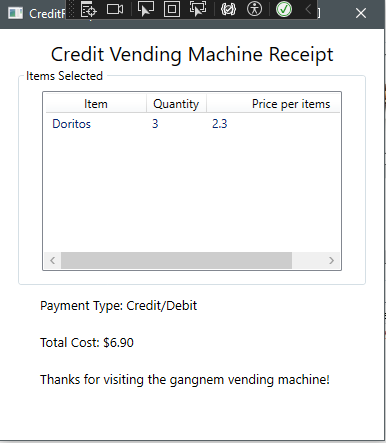
This section includes snapshots of the final application showing different features. It could be a guideline to using the application. You may include snapshots of the app while being developed. Remember to add explanatory captions to the snapshots.



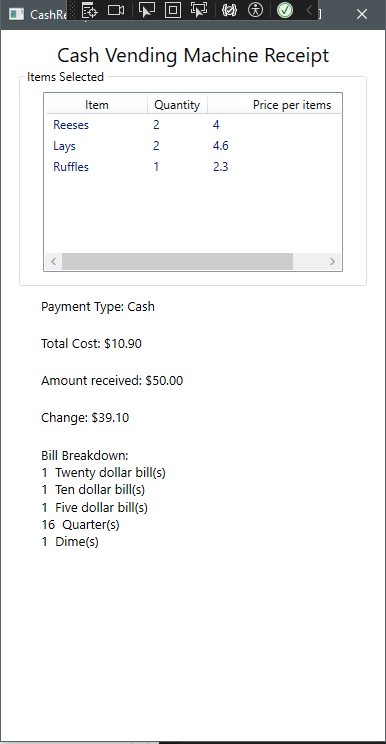
This is the Main Window, it displays all 12 buttons each with their own item, when clicked, the item is added to our observable Collection (aka the Cart) which we displayed on screen using ListView and a manual approach to updating the list for every event click. There is also a checkout button and a clear button, the clear button's purpose is to clear the cart, and allow users to view prices, while giving them the decision to erase undesired items. The checkout button would redirect the user to a pay method window.



Once the user checkouts, they are prompted with a pay method window, which asks the user whether they’d like to pay with cash or card. The credit validates that the total cost is above 5, ensuring that its above the minimum payment amount for card, then redirects them to a credit/debit receipt page. The bill amounts has validation ensuring that the user can only select a bill above the total cost amount and then redirects them to a cash receipt page.



If the user selected the credit/debit option they were then redirected to a receipt page, which displays the items they chose and the price of the items. It displays their payment type and finally the final cost, and a little thank you message.



If the user chooses to pay cash, they are prompted to the cash receipt page which display what items they bought, displays the payment type and the total cost. It then displays the bill amount chosen, and the expected change. With that information it calculates the bill breakdown of how much the user would receive and with what bills/change they’d get.

Future Work

Discuss features or improvement that can be added to application.

So, there is some logic we still wish to implement, such as finishing the stock(availability) of products taken from a file, to which once checkout the quantities in the file will be adjusted as well. Similarly, to this we also need to implement a method to disable any item buttons where the quantity is 0. We also need to finish our pricing logic which will aid greatly in displaying total costs, breakdown of bills and validation for the credit card option ensuring that the total cost is above the minimum threshold. Finally, we need to save the user's receipts to file, allowing us to have a history of billing from users.

UPDATE: Finished all above implementation, nothing else to add.

Appendix A: Team Contract

Submitted team contract goes here.

Appendix A: Team Contract Template

This is an informal contract to ensure that all team members have a common understanding of what is expected in terms of work standards, communication, division or work, and conflict resolution.

Team Members (Name & ID)

|  |  |  |
| --- | --- | --- |
|  | Name | Student ID |
| Member A: | Brayden Fournier | 6194481 |
| Member B: | Jayson Lartey | 2239058 |

Strength & Weaknesses

Within the context of this project, what are the strengths and weaknesses that each member brings to the team?

Member A:

**Strengths**: Cooperative team player, always open to constructive criticism. Ready to put in the work and time required to complete and excel at the project.

**Weaknesses**: I do struggle with procrastination so starting projects is always hard. Always overwhelmed.

Member B: Strengths: Cooperative, always open to new ideas, can brings different POVs to a problem

Weaknesses: Punctuality, sticking to one idea

Definition of “good enough” for this project

What would the team collectively consider “good enough” of an achievement for the project?

 - “Good enough” Apologies Aref, but this isn’t in our vocabulary. Jayson and I are striving to achieve our absolute best on this project and are not willing to accept anything less than perfection.

*(One response for the whole team)*

Picked Topic:

Topic 1: Simple Vending App

Division of work

How will each member contribute to the project?

Member A: We aim to divide the work as evenly as possible, 50/50 give or take.

Member B: Unclear as of now since we haven’t started but we tend to work on every part together

Frequency of communication

How often will the team be in touch and what tools will be used to communicate?

Every day! Discord

Response delays

What is a reasonable delay to reply to messages? Is it the same for weekdays and weekends?

1-3 hours

Receiving feedback

Each member must provide a sample sentence for how they would like to receive constructive feedback from their peers.

(If unsure, assume a hypothetical situation such as you have not completed your work in time or you have not replied to a message in a timely manner).

Member A: Don’t beat around the bush, ex “Great work, however I had something different planned, would you be open to something like this?”-provided example.

Member B: That’s a good way, but I would try to do it this way instead and see how it works?

In case of conflict

If a team member fails to communicate as described in this contract or does not respond to constructive feedback, what measures should the other teammate take?

 Advise the teacher, and continue trying to reach out via Mio, Teams, social media etc...

*(One response for the whole team)*

Appendix B: UML Class Diagram

* DO NOT PLACE A  LINK TO THE DIAGRAM.
* Do not include WPF created classes in the class diagram.
* The diagram should be placed in the document.

A screenshot of a computer

Description automatically generated