I. Project Title

Eco-Matic (Eco vending machine) using Console Application in C#

II. Introduction

This project aims to develop a console “Eco-Matic” Vending Machine Simulator in C#. The application will show the use of different OOP principles and file handling. The idea came from not just for commercial purposes but to also integrates a "trash to credit" recycling system, promoting awareness of Sustainable Development Goals (SDGs) like responsible consumption (SDG 12).

This project can potentially be a great idea in the future to be put on the streets of Cebu to help clean the trashes. The console program will be data-driven, with a dynamic inventory managed through a CSV file and a transaction log.

III. Objectives

- Develop a user-friendly and interactive console application with clear, separated roles for customers and administrators.
- Demonstrate inheritance and polymorphism by creating an abstract `VendingItem` class with specialized child classes (`DrinkItem`, `SnackItem`) that provide unique behaviors.
- Implement a data-driven inventory that loads and saves its state from a CSV file, allowing the machine's catalog to be modified dynamically.
- Create an admin panel, protected by a code, for managing the machine's inventory and logs.
- Integrate a "Recycle for Credit" feature, allowing users to convert simulated trash (plastic, glass, aluminum) into usable machine credit, thus promoting sustainability.
- Enforce realistic constraints, such as a maximum number of item slots and maximum stock per item, with input validation to guide the user.

IV. Scope

- The system will focus on simulating customer transactions (purchasing, recycling), managing a dynamic inventory, and providing administrative functions.
- The system will not include a GUI. It will only be purely console based.

V. Project Requirements

- **Software Requirements:**
  - IDE: Visual Studio Code or Visual Studio 2022 (Or any IDE with c# environment)
  - Language: C#.
  - Tools/Libraries: Standard .NET libraries, including System.IO for file handling and System.Linq for data manipulation. I will also be including System.Thread for sleep function.
- **Hardware Requirements:**
  - Any modern computer capable of running the .NET runtime.

VI. Functional Requirements

- **Customer Functions:**
  - Insert Money: Allow a user to add to their current balance.
  - Select Item: Allow a user to purchase an available item if they have sufficient balance.
  - Examine Item: Display descriptive "flavor text" for a selected item.
  - Recycle for Credit: Allow a user to select a type of recyclable material and quantity to add credit to their balance.
  - Get Change: Allow a user to finish their session and receive their remaining balance.
- **Administrator Functions:**
  - Restock Items: Restore the quantity of a specific item or all items to the maximum stock level (10).
  - Add Item: Add a new, unique item to the machine's inventory (up to the 6-item limit), defining its type, name, price, stock, and attributes.
  - Remove Item: Permanently remove an item from the machine's inventory.
  - View Log: Display the complete contents of the transaction log file.
  - Clear Log: Wipe all entries from the transaction log file after a confirmation prompt.
- **System Functions:**
  - Data Persistence: Automatically load and save the inventory from/to a CSV file.

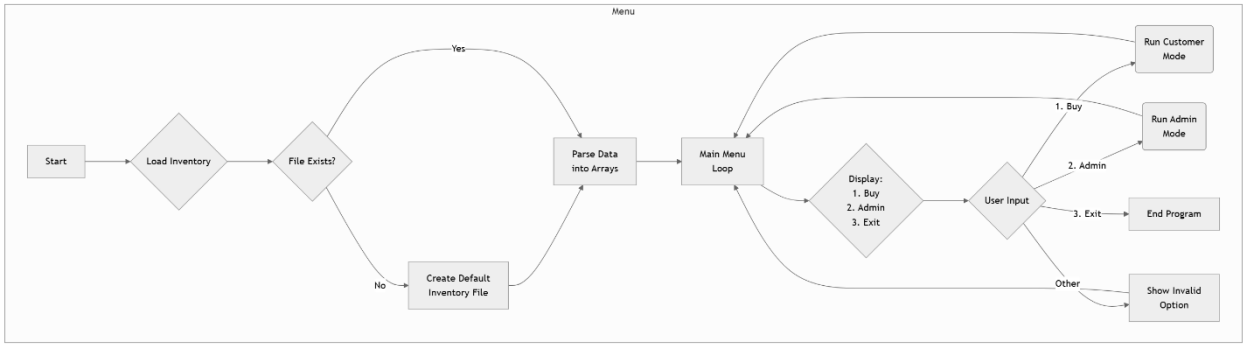
- Event Logging: Automatically log all purchases, recycling activities, and admin actions with a UTC timestamp (Using DateTime from System).
- Input Validation: Guide users to re-enter data if their input is invalid (negative prices, out-of-range stock levels).

VII. Expected Output

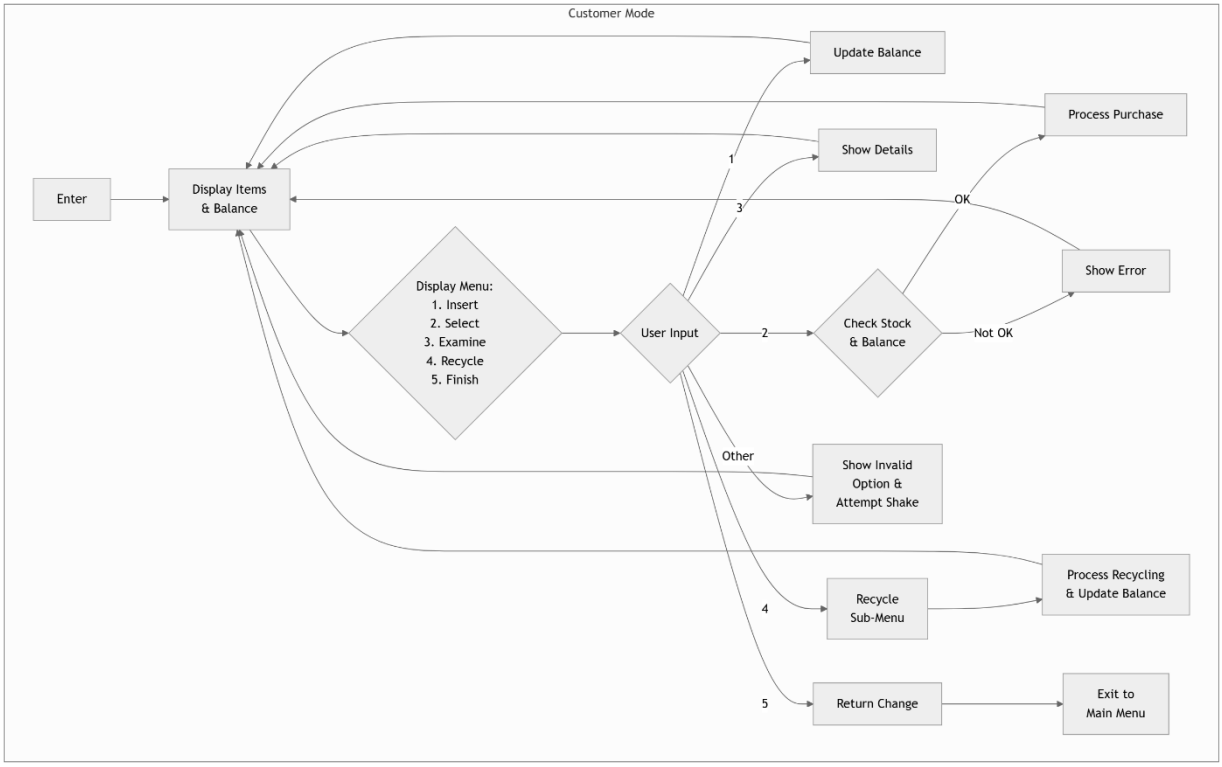
- The final deliverables for this project will include: The complete, single-file C# source code (Program.cs) for the Eco-Matic Simulator.
- The final executable console application.
- The inventory_stock.csv and transaction_log.txt files generated and used by the application during runtime.

FLOWCHART

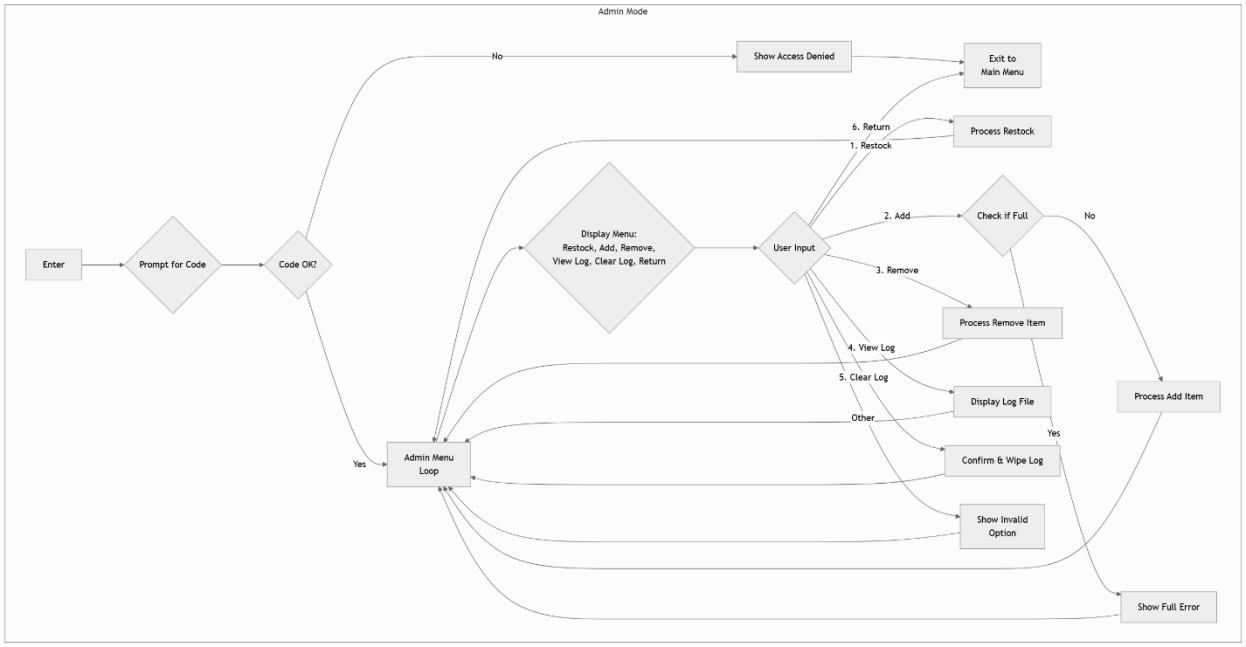
Menu



Customer


Customer Mode
Update Balance
Process Purchase
Show Details
Enter
Display Items & Balance
1
3
OK
Show Error
Display Menu:
1. Insert
2. Select
3. Examine
4. Recycle
5. Finish
User Input
2
Check Stock & Balance
Not OK
Other
Show Invalid Option & Attempt Shake
Process Recycling & Update Balance
4
Recycle Sub-Menu
5
Return Change
Exit to Main Menu

Admin


Admin Mode
No
Show Access Denied
Exit to Main Menu
6. Return
1. Restock
Process Restock
Enter
Prompt for Code
Code OK?
Display Menu:
Restock, Add, Remove,
View Log, Clear Log, Return
User Input
2. Add
Check if Full
No
3. Remove
Process Remove Item
4. View Log
5. Clear Log
Display Log File
Process Add Item
Other
Yes
Yes
Admin Menu Loop
Confirm & Wipe Log
Show Invalid Option
Show Full Error