CODELAB I

ASSESSMENT 2: Utility App

Tutor : Ms. Lavanya Mohan

Programming Fundamentals

|  |  |
| --- | --- |
| **Student’s Name:** | Frem Fredric Baring Patalinghug |
| **Student Id** | 04-24-184 |
| **Github Repository Name:** | <https://github.com/Eldeston/Creative-Computing> |
| **Github Repository Link:** | <https://github.com/Eldeston/Creative-Computing/tree/main/Year%201/03%20-%20Intro%20to%20Programming/Assessments/Assessment%2002%20-%20Intro%20to%20Python> |
| **YouTube Link** |  |

**BRIEF**

**Specifications**

The following is the basic structure of the the Vending Machine:

- Booting animation

- User sign in

- Main system

- User interface

- Deposit cash

- Purchase items

- View receipt

- Admin access (Secret trigger)

- Enter password

- Admin interface

- Withdraw profits

- Add stock

- Remove stock

- Change password

- Check logs

- Exit admin interface

- Exit user interface

- Exit main system

**System Flowchart**

**Technical Description**

To explain the technicalities of my vending machine, we need to look at it's basic form.

When you play an RPG game, what is the first thing you see when you launch the game? Or the first thing you see when you open a shop menu in-game? Or the first thing you see after booting up your BIOS menu? The first thing you will always see among these 3 things is a menu or a user interface.

So I began making my vending machine by first making a basic interface. Inside this interface it contains the following that the user could access (emphasis on could, more on that later):

User Interface (Menu):

Purchase (A separate interface for buying is found here)

Exit (Exit the program)

This is the most basic vending machine in it's most basic form, containing a basic menu and two options the user could access. "Purchase" for buying an item and "Exit" for exiting the program. So in pseudo code it would look like this:

# Run forever until break or exit

while True :

   # Ask for user input each loop

   userInput = input("\"Purchase\" to purchase coffee or \"Exit\" to exit program.")

   # If user purchases an item...

   if userInput == "Purchase" :

      purchaseInterface()

      print("Bought 1 can of coffee")

   # If user exits program...

   if userInput == "Exit"

      print("Goodbye and have a nice day.")

      exit()

   # If all checks are passed, announce invalid input

   print("Invalid input.")

The pipeline is quite simple if you look at the flowchart too. This allows for the developer to add as many options as they want with great flexibility just by adding another check statement and a function and using only one user input.

**Output**

\_\_ \_\_\_ \_ \_\_\_\_ \_\_ \_\_\_\_

/ |/ /\_\_\_ \_(\_)\_\_\_ / \_/\_\_\_ / /\_\_\_\_ \_\_\_\_\_/ \_\_/\_\_\_ \_\_\_\_\_\_\_\_\_

/ /|\_/ / \_\_ `/ / \_\_ \ / // \_\_ \/ \_\_/ \_ \/ \_\_\_/ /\_/ \_\_ `/ \_\_\_/ \_ \

/ / / / /\_/ / / / / / \_/ // / / / /\_/ \_\_/ / / \_\_/ /\_/ / /\_\_/ \_\_/

/\_/ /\_/\\_\_,\_/\_/\_/ /\_/ /\_\_\_/\_/ /\_/\\_\_/\\_\_\_/\_/ /\_/ \\_\_,\_/\\_\_\_/\\_\_\_/

--------------------------------------------------------------------------------------------------------------------------------

Current cash: AED 100.0

Receipt history: 1

--------------------------------------------------------------------------------------------------------------------------------

ID: 0, Name: Coca Cola, Stock: 16, Price: 3.0

ID: 1, Name: Pepsi, Stock: 16, Price: 3.0

ID: 2, Name: Miranda, Stock: 16, Price: 3.0

ID: 3, Name: Fanta, Stock: 16, Price: 3.0

ID: 4, Name: Mountain Dew, Stock: 16, Price: 3.0

ID: 5, Name: Water, Stock: 16, Price: 1.0

--------------------------------------------------------------------------------------------------------------------------------

Type "Deposit/D" to deposit cash, "Purchase/P" to purchase items, "Receipt/R" to open receipt, or "Exit/E" to exit machine.

P

Purchasing items...

Enter item ID: 0

Enter item amount: 10

Dispensed 10 Coca Cola for AED 30.0 with AED 70.0 left in deposit.

Type "Continue" to continue making purchases or press enter key to exit interface...

**Critical Reflection**

**Appendix**

**Reference List**