FINAL REPORT

by

Syed Nazmul kabir

FIT 9136

STUDENT ID : 32312016

date: 12th April 2021

**The vending machine program is explained by the following diagram**

Login

Restart by writing y with welcome message

Going to User interface

yes

No

Refund, Update inventory and transaction history

Maintenance done

Maintenance Stop

Factory Reset

Going for reset or go to user interface

Getting updated sales and inventory data

Valid

Quitting with goodbye

Confirmation of buying drinks or changing order or quit

Option to view menu by writing y or quit by selecting any key

More drinks

Money back

Money back or dispensing.

Invalid

Balance low

Balance ok

Is it valid coin

Insert coin.

Price check

Selecting drinks

User login ( No password need)

Customer Login by writing Red then Password required: Password: 123xyz

**Description of High-level Design of the Programs**

The vending machine program is developed for fulfilling the requirements of both user and the customer who is the owner of the machine. Few functions are being created to use those inside another functions for several times. This helps to reduce coding duplications, make it simpler and confirming high level programming.

**User Case:**

In case of user interfaces, an user can restart program writing ‘ y ’ (the word ‘ press ’ actually represents ‘ write ’ ) in the input template bypassing logging option for a customer. As soon as restart is done, **set\_machine()** takes input from the customer through a text file to set the data for drinks, price and stock. Using this function, a menu and stock dictionary is developed which can be reset with the restart of the machine. For displaying, a customized menu is developed by **displaymenu()** function which is re-called every time when user needs to see the menu. Thus high level programming is maintained. **Show\_menu()** function helps to deal with the situation when user oppose not to see menu and quite the transaction.

**Totalcost()** and **deposit()** function accumulate data related to costing and money deposit respectively for each transaction. This two functions are used in **change\_func()** to display transaction history to the user at the end of transaction.

**Low\_balance()** function take a parameter which is the chosen drinks by the user. This function is used every time when program needs to compare price with the inserted coins, thus confirming that machine got money to dispense the drinks. The function provides pathway to the user to select other drinks or for quitting.

**Milk and sugar()** helps user to select either milk or sugar or both along with only the selection of tea or coffee. The functions are checking the coin balance with the cost using low\_balance() function. Those are used every time when users select tea or coffee for their drinks, thus avoiding excessive coding and high programming level is maintained.

**Change\_func()** is the most used function. It is recalled every time when user chose to quite the transaction. The function provides transaction history and updates sales and inventory data corresponding to order time.

**Another\_drink()**  is used always when user choose for another drink. The function either helps to choose a drink recalling drink\_selection() function or to terminate the transaction using change\_func(). **No\_stock()** and **not\_in\_menu()** function deals with the situation if user choose something that is not in menu or has no stock by mistake.

**Coin\_validity()** function is recalled all times when user inserts a coin. It confirms whether the coins are the allowable one by the machine. It also update coin list so that total deposit can be traced out for a particular transaction.

**Drink\_selection(), drink\_confirmation()** and **drinks\_func**() function are integrated with each other where the first one is guided by show\_menu function. Drink\_selection() confirms the situation of ‘no stock’ , ‘not in menu’ and ‘cancel order situation’ while allowing users to select drinks by recalling corresponding functions. Then it leads the selected drinks to drink\_confirmation() function as a parameter where user can continue buying by inserting allowable coins . In this case coin\_validity is recalled and then the program is shifted to drinks\_func(). The parameter of drink\_func() is the selected drinks in drink\_selection() function. For discontinuing the order, drink\_selection() is recalled again.

Drinks\_func() asks buyer for reconfirmation before getting drinks in dispenser. It handles the situation of another drink request, low balance of money and item not menu as well by recalling corresponding function.

**Customer case**:

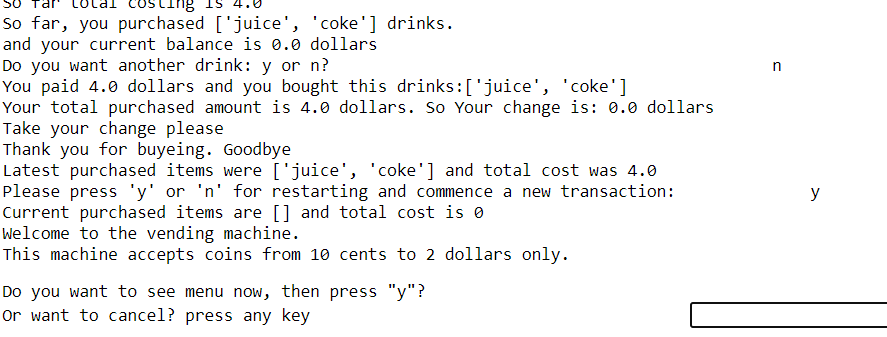
The **login()** function confirms that the owner or customer can log in the machine by simply writing ‘l’, then ‘red’ and then the password = 123. If customer wants to check the machine as user he can swap to user case, for this **restart()** function is recalled in side login() function. To do other activities, log\_in() function guides to customer() function.

**customer()** function accommodate other functions to show stock alert, to get transaction & inventory history and to reset the machine. These are done respectively by recalling **stock\_alert(), owner\_record(), and reset()** functions inside the customer function. That is the example of high-level programming.

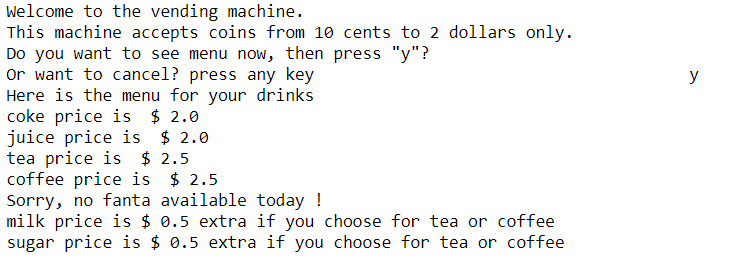
The whole program is designed to handle all kind of exceptions and errors.

Designing **User Cases** Addressing **All** Requirements:

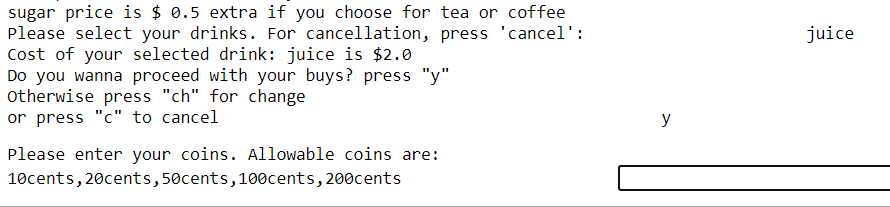
|  |  |
| --- | --- |
| Name | U-001 |
| Actors(s) | User A new transaction shall be started |
| Goal | A new transaction shall be started. Users select **restart** to reset previous transactions and conditions to clear transaction history. |
| Explanation | A full running of a transaction was done at first, then transaction finished. Machine was showing last transaction history. Then a restart was done. After that, purchased item list was empty and total costing was 0. |



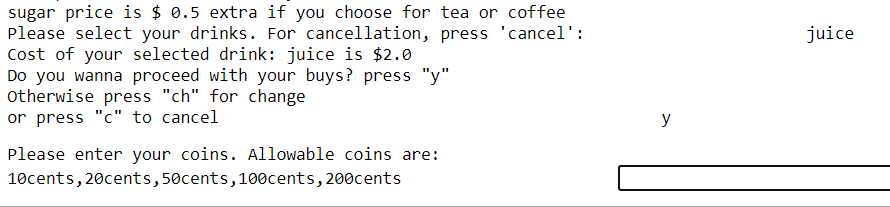
|  |  |
| --- | --- |
| Name | U-002, S-009 |
| Actors(s) | User |
| Goal | U-002: List of products must be displayed  S-009: The system shall allow a user to select products coffee, tea, Coke… |
| Explanation | Every time, after restarting, the machine asks for input from the user to have the menu on the screen. For yes response, the screen showing available drinks with pricing. User can select coke, juice, tea or coffee. Milk and sugar is only available with coffee and tea. |



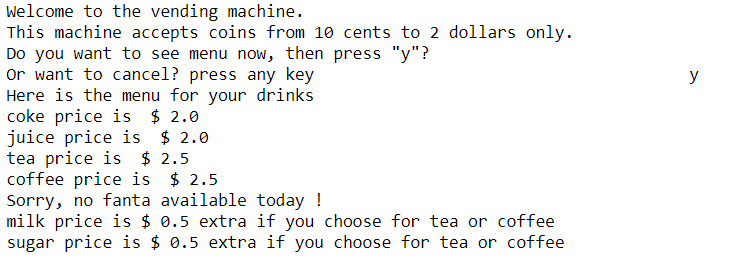
|  |  |
| --- | --- |
| Name | U-003 |
| Actors(s) | User |
| Goal | Cost of the selected item must be displayed |
| Explanation | After showing menu, the machine asks for selection of the drinks. After selecting, screen suggests whether user wants to proceed with transaction or change or cancel whole transaction. |



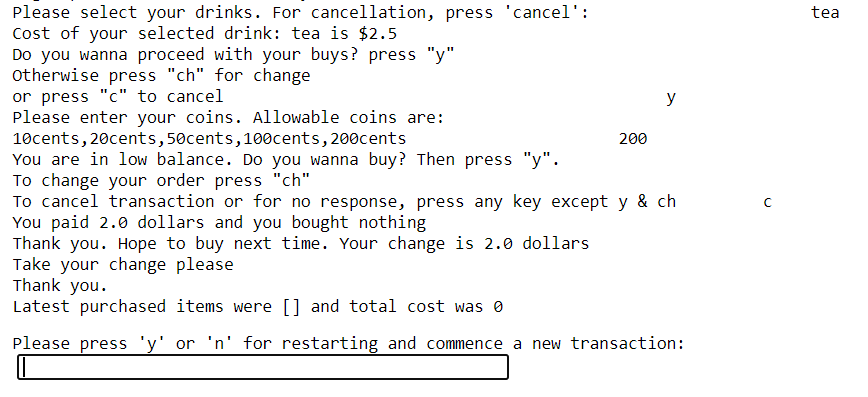
|  |  |
| --- | --- |
| Name | U-004 |
| Actors(s) | User |
| Goal | Coin must be inserted to get the product |
| Explanation | After selecting and confirming drinks, the machine asks for coin inserting |



|  |  |
| --- | --- |
| Name | S-002 |
| Actors(s) | User |
| Goal | Only readily items must be displayed as item list otherwise appropriate information to the user |
| Explanation | The menu shows the notice of any unavailable drinks due to stock shortage. |

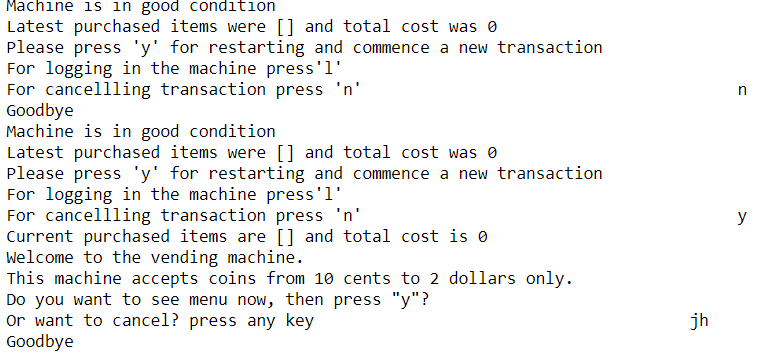


|  |  |
| --- | --- |
| Name | S-007, S-010 |
| Actors(s) | S-007: User ; S-010: User and Customer; |
| Goal | S-007: The system must dispense the selected item only if the amount coin is inserted unless it must rollback the transaction.  S-010: The system must accept **coins** of different in amount |
| Explanation | S-007: User selected tea. Pricing was 2.5 dollars. User paid 2 dollars. So the machine reported about low balance and asked for more coins. User choose other way. So The machine refunded user’s 2 dollars .  S-010: User can select different coins ranging from 0.1 to 2 dollars. |

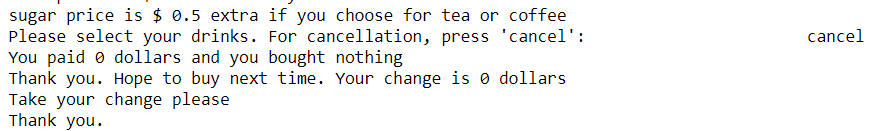


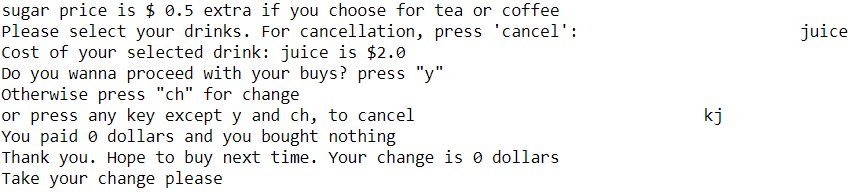
|  |  |
| --- | --- |
| Name | U-005 |
| Actors(s) | User |
| Goal | The transaction shall be cancelled as required. |
| Explanation | User can cancel order at any stage of the transaction. Cancellation can be done during restart, menu choice, after selecting drinks, during inserting coin, during changing drinks and finally while selecting no more drinks. |

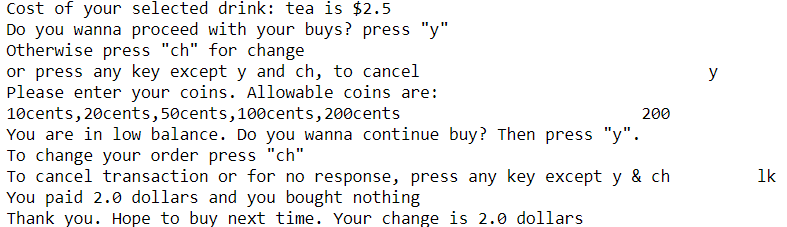
*Users can quit at the beginning before restarting. They can quit just after restarting the machine before watching menu as well:*

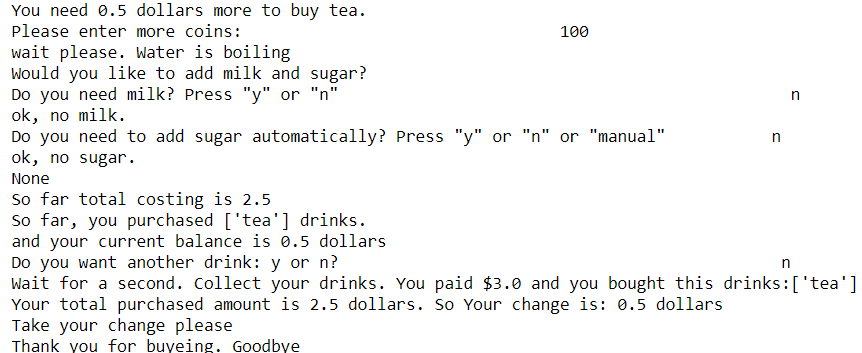


*User can cancel order after watching menu:-*

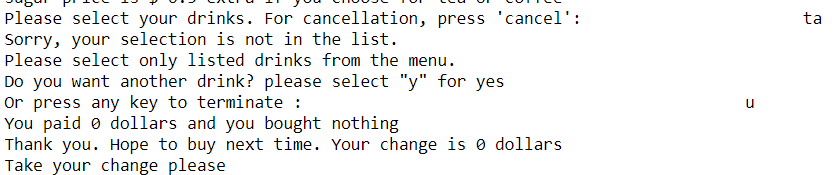


*User can cancel order after selecting drinks from menu:-*

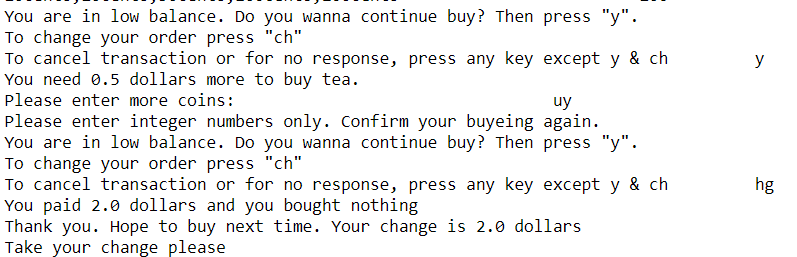
*Users can cancel order while inserting coins if they feel that item is too costly:-*

*Users can oppose extra items like milk or sugar and finally can oppose more to buy, thus quiting:*

*Users can cancel order after writing something that is not in the menu:*

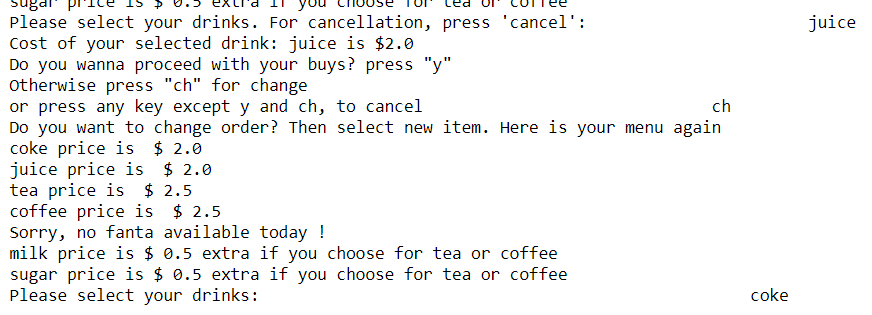


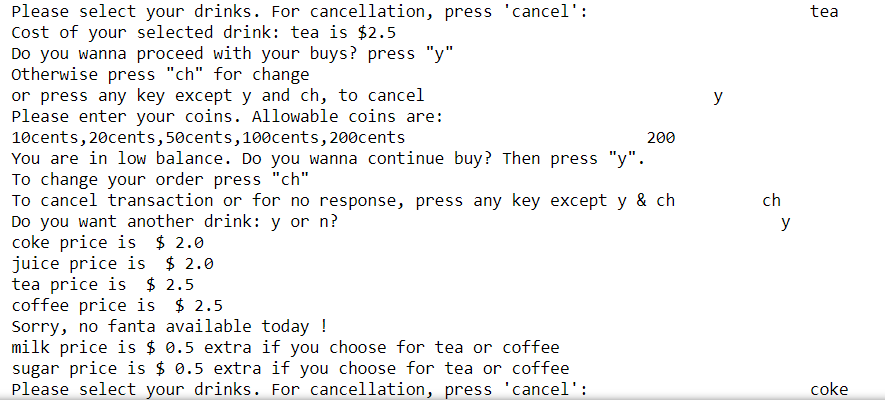
*User can cancel order while doing mis-operation at the time of inserting coin:*



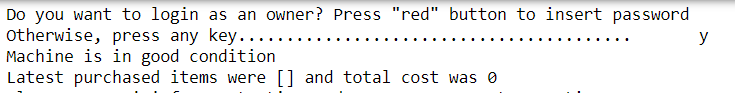
|  |  |
| --- | --- |
| Name | U-007 |
| Actors(s) | User |
| Goal | Order shall be changed as required. |
| Explanation | Order can be changed just after selection of drinks. If users are in low balance, they can change order. |

*User can change order just after selecting drinks*

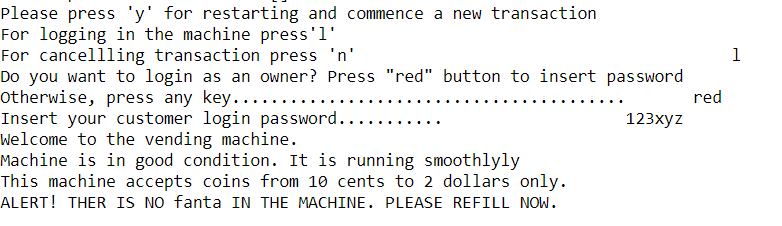


*User can change order at the time of inserting coin if they think they do not have enough coins:*

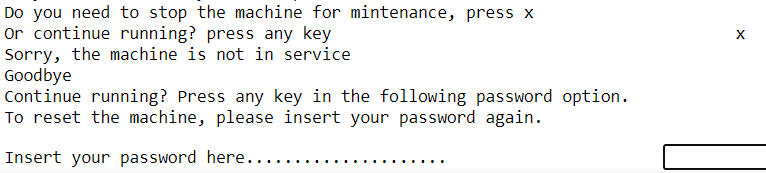
|  |  |
| --- | --- |
| Name | S-001 |
| Actors(s) | User |
| Goal | The status of the machine shall be displayed. |
| Explanation | If machine is not stopped, user and customer both can see that machine is in good condition. |

*In user mode, Machine is displaying that ‘ Machine is in good condition’.*

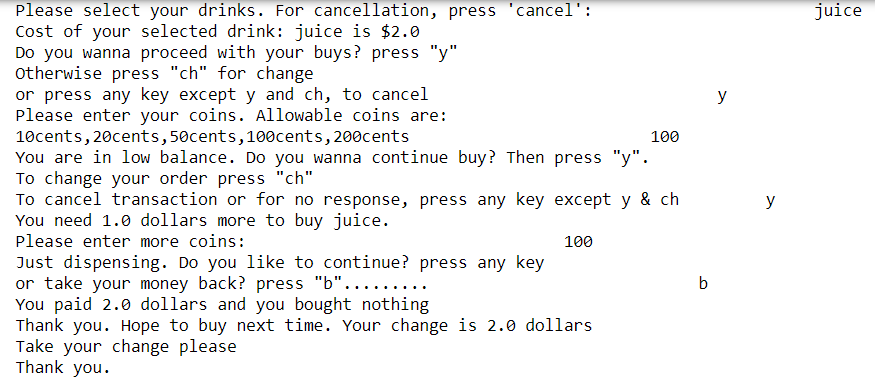
*In customer mode, while machine not in stopped, showing the same message:*



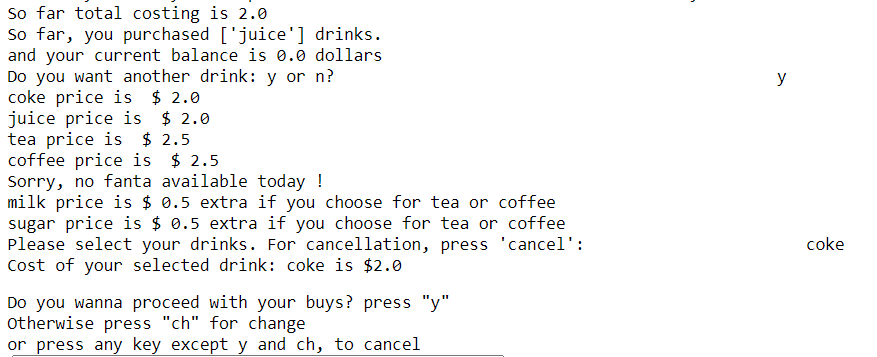
*In customer mode, while machine is stopped, showing different message:*



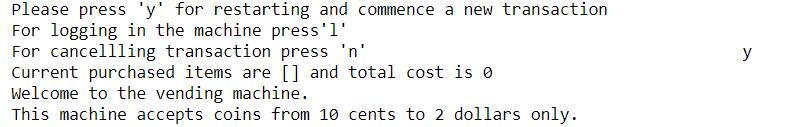
|  |  |
| --- | --- |
| Name | S-004, S-011 |
| Actors(s) | User |
| Goal | S-004: The system shall refund the coin if in need by the user before the item gets ready for dispense.  S-011: The system must compare item cost with entered coin |

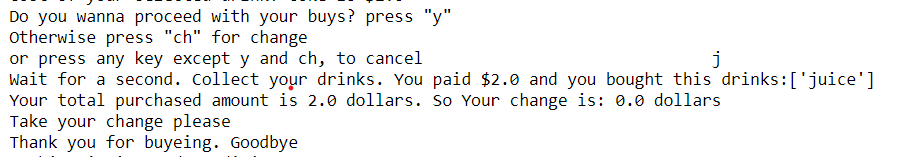


|  |  |
| --- | --- |
| Name | U-006 |
| Actors(s) | User |
| Goal | Transaction shall be continued as required without restarting all steps. |
| Explanation | User can get another drink without restarting the program. |

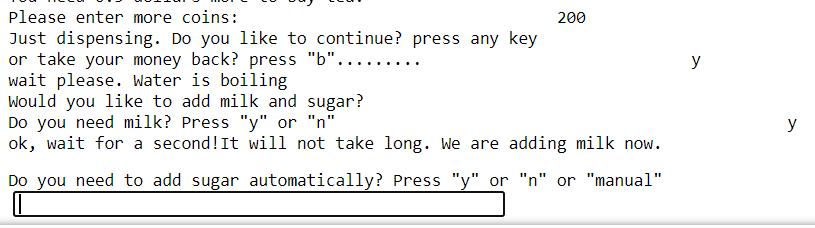


|  |  |
| --- | --- |
| Name | S-005 |
| Actors(s) | User |
| Goal | The system shall display welcome and goodbye messages at the beginning and end of the transaction respectively. |

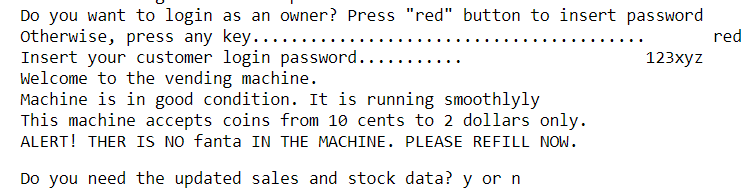




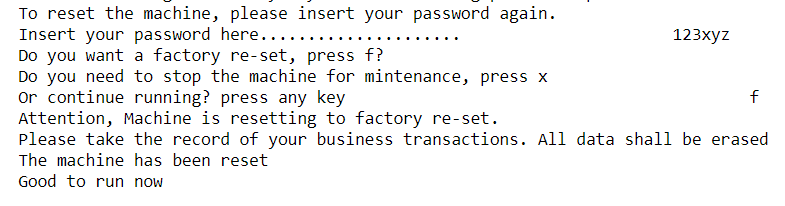
|  |  |
| --- | --- |
| Name | S-006 |
| Actors(s) | User |
| Goal | The system must display waiting time as a message to the user in preparing the items |



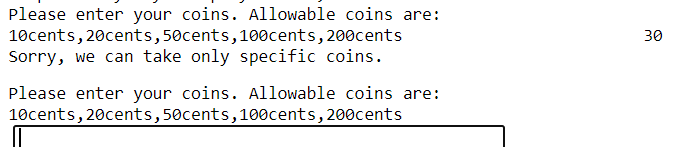
|  |  |
| --- | --- |
| Name | S-003 |
| Actors(s) | Customers |
| Goal | Alert message shall be printed automatically by the machine if any of ingredient is finished |



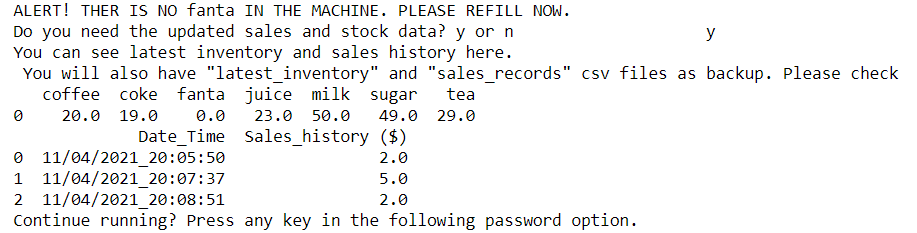
|  |  |
| --- | --- |
| Name | S-008 |
| Actors(s) | Customers |
| Goal | The system shall allow resetting operation for vending machine supplier. |



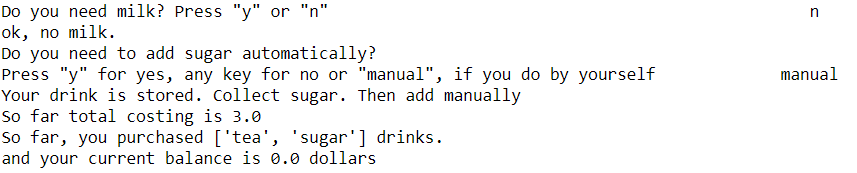
|  |  |
| --- | --- |
| Name | S-012 |
| Actors(s) | Users |
| Goal | The system must check the validity of the coin |



|  |  |
| --- | --- |
| Name | D-005 |
| Actors(s) | Users |
| Goal | The system must store the coins and record all transaction |



|  |  |
| --- | --- |
| Name | D003, D-006 |
| Actors(s) | Users |
| Goal | D003: The system shall store Coffee and allow the user to mix sugar manually or automatically.  D006: The system need to provide statistical data based on the transactions |



**Reference:**

Monash University, *Monash Moodle, FIT9136 Algorithms and Programming Foundations,* viewed on 10April 2021, https://lms.monash.edu/my/