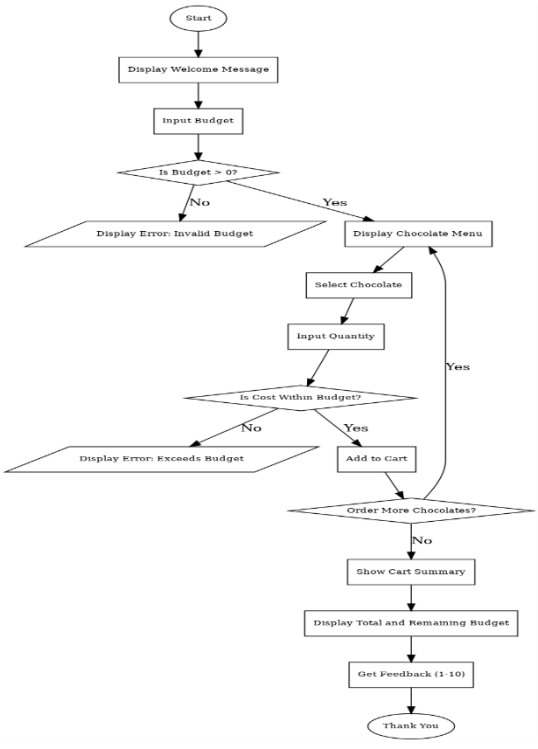
The Development Document By: Samuel Justine Sampang

SPECIFICATION : GITHUB RESPIRATORY

For this assessment, we had to develop an application for a vending machine as our final task for the first semester. Using Python code programming, I developed a Python software about a virtual chocolate vending machine that makes buying fun and easy for customers. At the start of my program, I welcome my clients and provide the chocolate menu that is available. The user can also monitor their spending by selecting how many chocolates to buy and setting their budget. The application determines the total cost, monitors the shopping cart, and ensures that the budget is adhered to. In order to make the interaction enjoyable and engaging for users, the program will ask about customer satisfaction and service at the end after receiving input.

SYSTEM FLOWCHART : PHOTO OF THE FLOWCHART



The application begins by introducing itself and inquiring about the users' budgetary resources in this flowchart system. The application also makes sure the user's budget is in order. After that, you can select the type of chocolate you like and purchase as much as you like, depending on your budget. The application will monitor your shopping cart and overall cost,

ensuring that it stays within the user's budget. Additionally, if you're in the mood for chocolate, the application will ask if you want to keep getting it. When the user is finished, the application will generate a receipt that includes the total amount paid, the items ordered, and the remaining balance. The application will then inquire about the user's opinions and level of customer satisfaction. Like a typical vending machine, this flowchart is simple to grasp and follows a step-by-step procedure.

TECHNICAL DESCRIPTION & WALKTHROUGH: VIDEO LINK

Introduction:

I'll describe how my program codes and all of the general functions of my project vending machine operate in this task. Additionally, I'll describe the special functionality I added to my program.

Functions:

Items -

Ten keys containing chocolate things are defined in the variable **chocolate_items**, which is defined in the first section of this programming code. Each key's value is a different dictionary that contains information on the chocolate's name, brand, and cost. This dictionary code is used to arrange all of the available chocolates and save the chocolate alternatives in the machine. whenever the user needs to be informed of the available chocolates by the application.

Cart –

In order for us to add all of the chocolate goods that the user chooses and indicates how many chocolates (along with their quantities) they desire, the cart will be an empty list. The chocolate that the user selected to purchase can also be tracked by the shopping cart.

Display Items –

The display items function, is to show the available chocolates from the **chocolates_item()** where the **menu_display()** function is used. This function print all the available chocolate

items in the vending machine showing both name and price. The price format with two decimal places so it will look more specific and professional, realistic at the same time.

Purchase Item –

In the variable function **select_quantity()**, the user will input to select the chocolates that they want to buy in from the **menu_display()** and specify the quantity each chocolates the users ordered. Firstly, the function will display the menu, then the user can choose a chocolate by entering the code (1-10) that the program will ask. After selecting a chocolate, the program will ask if how many of that chocolate they want to to buy. Then if the total cost of their selection surpass to the users inputted budget, the program will automatically notified the user so they can adjust their order. Lastly, the function then adds the selected items to their shopping cart and update the total cost, continuing until the user is done to shop .

Display Cart –

Once the user is ready to checking out their chocolate, the **Display Cart** function is now called. It shows the user a summary of everything the user buy and what they've added in their cart. If the cart is empty the function will automatically remind the user to select their chocolate orders before continuing. So this will ensure the users knows exactly what their buying and how much money they will pay.

Get Money -

After showing the cart, the program will asks the user to input their amount of money they have. If the money was entered enough to cover the total cost, then the program will calculate and prints it out. If the user hasn't entered enough money, the program will ask the user to enter the exact correct amount.

Item Purchased –

Once the correct amount of money the program received, the program prints out the final receipt that the user ordered, including the total cost and any change. It also has a thanks message directly to the user for their purchase.

Feedback -

The program conclude by asking the users their honest feedback from 1 to 10. Based on their feedback, the program display a response ranging from expressions, and gratitude for high

ratings and improvement for lower ones. The reason why I want to add feedback function for to ask the users their satisfaction level and hear their opinion by using numbers to improve more to my program.

CRITICAL REFLECTION:

I learned very little about programming during the creation of this assessment, and I occasionally struggle to comprehend and become familiar with it. As a result, it takes me some time to master it, but every time I ask a classmate for assistance, it greatly aids in my learning, even if it takes some time. I often believe that programming is not for me since, in addition to learning all of the functions and data types by heart, I also need to constantly practicing and pushing myself to get better. I kept giving up while I was programming the vending machine utility app because I realized how inadequate I was, but I persisted because it just takes time. Occasionally, I made a big mistake in my program, so I always asked the class for assistance. Asking questions also helped me get better and added a technique to my repertoire. Even though I was under pressure, I always saw this as a positive thing since it pushes me so much and I enjoy it. It also helps me connect with other people. However, I want to make my chocolate vending machine program even better so that it surpasses people's expectations and functions like a real machine. And I'm glad I took this course, and I aim to be a better programmer in the near future than I am now. I want to study other programming languages as a hobby and a career so I can enjoy it more.

APPENDIX:

```
print("Hi Welcome to Samuel's Chocolate's Vending Machine !")

chocolates_item = {

  1:{"name" :"M&M's","price" : "6.00"},

  2:{"name" :"Maltesers","price" : "5.50"},

  3:{"name" :"Bounty","price" : "5.00"},

  4:{"name" :"Mars","price" : "4.00"},

  5:{"name" :"Kinder Bueno","price" : "4.00"},

  6:{"name" :"Reese's","price" : "4.75"},

  7:{"name" :"Twix Top Chocolate Bar","price" : "2.00"},

  8:{"name" :"Hershey's Milk Chocolate","price" : "4.50"},

  9:{"name" :"Cadbury Dairy Milk","price" : "3.75"},

  10:{"name":"Galaxy Smooth Chocolate Bar","price" : "4.75"},

}

def menu_display():
```

```
  ### Displays the Menu ###
 print("Here are the available Chocolates")
 for code, details in chocolates_item.items():
  print(f"{code}. {details['name']} - AED {float(details['price']):.2f}")



def select_quantity(budget):
 ### It allows the user to select and specify their orders quantity ###
 shopping_cart = {}
 total_cost = 0
 while True:
  ### Display the menu ###
  menu_display()

  ### Get users chocolate selection ###
  try:
   choices = int(input("\nEnter the number of chocolate  you want to buy: "))
   if choices not in chocolates_item:
    print("Invalid choice !! Please select a valid number from the list of items.")
    continue
  except ValueError:
    print("Please enter valid number.")
    continue

  ### Get users quantity of order ###
  try:
   quantities = int(input(f"\nEnter the quantity of {chocolates_item[choices]['name']} you want to purchase: "))
   if quantities <= 0:
    print("Quantity atleast must be more than 1. ")
    continue
  except ValueError:
    print("Please enter valid number.")
    continue

  cost = float(chocolates_item[choices]['price']) * quantities

  if total_cost + cost > budget:
   print(f"I apologize, but this choice is too expensive for you! You have AED left over from your budget is {budget - total_cost:2f}. ")
   continue

  ### Shopping cart and update total cost of budget !!

  if choices in shopping_cart:
   shopping_cart[choices] += quantities
  else:
   shopping_cart[choices] = quantities
  total_cost += cost

  ### Ask user if the users want to order more ###
```

```
    more = str(input("\nWould you want to choose a different chocolate? PRESS Either yes or no: ")).strip().lower()
    if more != "yes":
      break

  return shopping_cart, total_cost

### Main program ###
try:
  users_money = float(input("Enter your money in AED: "))
  if users_money <= 0:
    print("Money must be more than 0.")
  else:
      cart, total = select_quantity(users_money)
      print("\nYour Shopping Cart:")
      for code, quantity in cart.items():
        name = chocolates_item[code]['name']
        price = float(chocolates_item[code]['price'])
        subtotal = price * quantity
        print(f"{name} x {quantity} = AED {subtotal:.2f}")
      print(f"Total: AED {total:.2f}")
      print(f"Remaining Budget: AED {users_money - total:.2f}")
except ValueError:
  print("Please enter a valid budget.")

while True:
  try:
      feedback = int(input("Rate us from 1 to 10: "))
      if 1 <= feedback <= 10:
        if feedback == 10:
          print("Thank you! We're thrilled you loved it! Please come again.")
        elif 7 <= feedback < 10:
          print("Thank you for your feedback! We're glad you had a good experience.")
        elif 4 <= feedback < 7:
          print("Thank you! We'll work on improving your experience.")
        else:
          print("We're sorry to hear that. We'll strive to do better next time.")
        break
      else:
        print("Invalid input. Please rate us between 1 and 10.")
  except ValueError:
      print("Invalid input. Please enter a number between 1 and 10.")
```