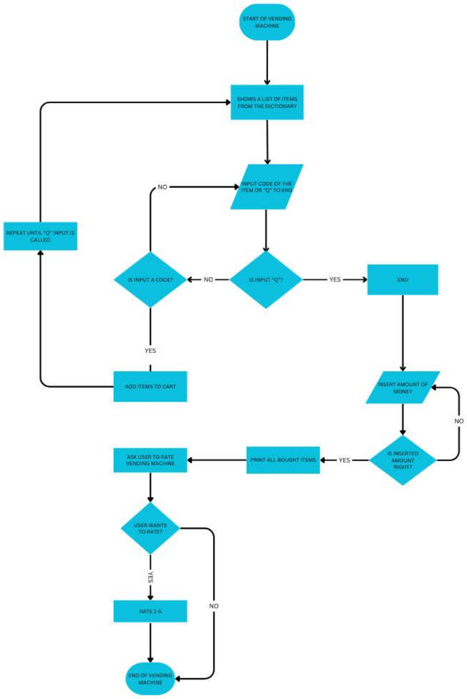
The Development Document By: Eizyah Kein B. Lomod

SPECIFICATION: LINK OF GITHUB

In this task, we were assigned by Mr. Oliver to make a vending machine with the python programming language as our final assessment for the 1st semester. The vending machine that I made is a simple vending machine that consists of 7 def functions to start the vending machine and a dictionary. There are a total of 12 items in my vending machine, 4 of them being chips, another being sandwiches, and the last being drinks. I added a cart feature into my vending machine as well as the stock feature to make my vending machine a bit more function-able.

SYSTEM FLOWCHART: LINK



In the flowchart above, I made how the system works for my vending machine. When you start the vending machine, it prints out all available items in my vending machine with codes, then it asks you for the code or if you wanna finish the machine by typing "q". Once you have finish purchasing the items, the vending machine asks you to input money and once the required money is received, the machine prints out all the items you have bought and asks if you want to rate the machine or not.

TECHNICAL DESCRIPTION & WALKTHROUGH: LINK

Introduction:

For this task, I'll be breaking down how my codes work and how overall the functions in my vending machine happens to run. I'll also be talking about what features I included in my vending machine such as the stock feature, cart feature, and the rating feature.

Functions:

- **Items:**
  Now breaking down my code. The first code we have is the Items variable, in it is a dictionary that consist of 12 keys and 12 values, but within the value is another dictionary. These keys and values are the items inside my vending machine and are shown when their value is called with a for loop.

- **Cart:**
  The cart will be an empty list so we can append all the items that are bought in the vending machine to the cart.

- **Display Items:**
  To display the items, I made a define function called display_items and inside are the print functions to print the name of my vending machine and use the for loop to print the code and the item name and stock in the items variable and left align the item['name'] within the width of 20.

- **Purchase:**
  Define purchase is where you input the code of the item you want that's in the items variable. I used while true so that it keeps repeating until "q" is inputted. If the code is in items variable, it creates 4 new variables called item, name, price, and stock. In Item is where you retrieve a specific element using a code then store it in the Item variable. In name is where you retrieve value of 'name' of the code that is stored in the Item. In price is where you retrieve the value of 'price' of the code that is stored in the Item. And in stock is where you retrieve the value of 'stock' in the code that is stored in the item. If stock is greater than 0, we will use the variable cart and append the name and price into the cart, then the stock stored in the item variable will decrease by 1 each time it is called. Then we will print "you have added: {name}(which is the name of the item you inputed) for {price}(the price of the product)AED in your cart. Remaining stock: {stock}(the available stock of the product). After that if item does not have anymore stock it will print "Sorry, {name}(the name of the product you want) is out of stock." Or if the code you have entered isn't found it will print "Invalid code. Please select a valid item".

- **Display Cart:**
  In this function, if your cart is empty it will print "Your cart is empty, please purchase items in the machine before inserting money.". Otherwise it creates a new variable called total_price. This variable will calculate all the total price of the items that is in the cart. After it calculates the items in the cart, it will use a for loop to print the items and the price and then prints the total value of the items in the cart.

- **Get Money** :
  Once the machine is done calculating all the items in the cart, the machine will ask to input the right amount of money. If cash is greater or equal to the total_price of the item it will return the change. Else if your money is less than the requested money, it will return none and ask the user to input the right money again.

- **Items Purchased:**
  After the machine received enough money, the machine will print out all purchased items, the total price, the cash received, the change if you've put a great amount of cash, and a message saying "Thank you for using Eizyah's Vending machine!!!"

- **Rating:**
  After everything is done, the machine will ask the user if they would like to rate my machine. If the user inputs "y", the machine will again ask the user how they would like to rate my machine from 1-5. If they inserted 1-3 the machine will then print a message saying "Thank you for rating my machine! I'll do my best to improve.". And if the user inputs 4-5, the machine will print another message saying "Thank you for the excellent rating. I look forward to you using my machine again!". Else if the user input "n" the machine will then end and the machine will print a message saying "Thank you for using Eizyah's vending machine!!!".

- **VendingMachine:**
  Is where every function is compiled as one and when VendingMachine is called, the machine will run.

CRITICAL REFLECTION:

During the making of my vending machine, I knew I lacked so much when I made so many errors trying to figure out how to make my vending machine work. I tried multiple times to make the code in my brain work, but in the end I still failed at doing so. I wanted to make a vending machine with categories, but I lack the brain process to make it. But, I will improve myself and try out more things using python and explore new stuff to help improve my knowledge on the python programming language. I am still relieved I was able to make my vending machine work, I was compelled on how unique it was in my eyes. I was thinking "why don't I try this?" , and I somehow made it. I thought "Oh, wow, this works." and went to the next function, failed a couple of times, but found solution to make it work. It's amazing how just this one programming language can make you think "There's so many things I wanna try" and then it fails, but the beauty in coding is when you fail then you find a solution to make it work. You keep finding solutions to make it work the way you want it to work. There will be certain errors or bug, but the debugging won't take that much time if you know where the error starts. I'm excited to learn more about this language as the time pass on, I'm glad I took this course.

APPENDIX:

```
#Creates a dictionary of items and their price and stock.
items = {
  '1001': {'name': 'Piatos', 'price': 5.00, 'stock': 7},
  '1002': {'name': 'Cheetos', 'price': 4.00, 'stock': 5},
  '1003': {'name': 'Cheepy', 'price': 4.50, 'stock': 4},
  '1004': {'name': 'Clover', 'price': 5.50, 'stock': 6},
  '2001': {'name': 'Egg Sandwich', 'price': 6.50, 'stock': 3},
  '2002': {'name': 'Ham Sandwich', 'price': 7.00, 'stock': 5},
  '2003': {'name': 'Cheese Sandwich', 'price': 5.50, 'stock': 4},
  '2004': {'name': 'Veggy Sandwich', 'price': 8.00, 'stock': 7},
  '3001': {'name': 'Coke', 'price': 3.00, 'stock': 5},
  '3002': {'name': 'Sprite', 'price': 3.00, 'stock': 7},
  '3003': {'name': 'Fanta', 'price': 3.00, 'stock': 7},
  '3004': {'name': 'Water', 'price': 1.00, 'stock': 10},
}
#creates an empty cart
cart = []

#This function will display the items in the dictionary.
def display_items():
  print("\n************WELCOME TO EIZYAH'S VENDING MACHINE************\n")
  print("****************ITEMS IN THE MACHINE*****************")
  print()
  for code, item in items.items():
    print(f" {code} - {item['name']:<20} | {item['price']:.2f}AED | stock: {item['stock']} |")


#This function is where you input the code of the item you want to purchase.
def purchase(cart):
  while True:
    print()
    #input the code for the item.
    code = input("Enter the code of the item you want to add to your cart (or 'Q' to finish): ")
    #If Q is called
    if code.upper() == 'Q':
      #breaks the input whether there's an item or not.
      break

    #creates an if statement
    if code in items:
      item = items[code]
      name = item['name']
      price = item['price']
      stock = item['stock']

      if stock > 0:

        cart.append((name, price))
        item['stock'] -= 1
        print("-------------------------------------------------------------------------------------\n")
        print(f"YOU HAVE ADDED: {name} for {price:.2f}AED in your cart.          Remaining Stock: | {stock:>4}   |")
        print("***************************************************************************************\n")
      else:
        print(f"Sorry, {name} is out of stock.")
    else:
      print("Invalid code. Please select a valid item.")

#This function will display the purchases you made within the machine.
def display_cart(cart):
  if not cart:
    print("Your cart is empty, please purchase items in the machine before inserting money.")

  else:
    total_price = sum(item[1] for item in cart)
    print("Your Cart: \n")
    for item, price in cart:
      print(f" {item:<10} - {price:.2f}AED ")
      print("---------------------------")
    print(f"Total: AED{total_price:.2f}\n")
```

```
#Will ask the user for the exact amout of money needed for the purchase.
def get_money(cart):
   total_price = sum(item[1] for item in cart)
   cash = float(input(f"Insert the total amount of {total_price:.2f}AED:  "))


   if cash >= total_price:
      return cash
   else:
      print("Insufficient funds. Please insert more cash.")
      return None

#The machine will print out all the items purchased in the machine.
def items_purchased(cart, total_price, cash):
   if not cart:
      print("No items purchased.")
   else:
      print("\nITEMS PURCHASED: \n")
      for item, price in cart:
         print(f"{item:<15} - {price:.2f}AED\n")
      print("***************************************************************\n")
      print(f"*Total: {total_price:.2f}AED")
      print(f"*Cash Received: {cash:.2f}AED\n")
      print("***************************************************************\n")
      change = cash - total_price
      print(f"*Your Change is: {change:.2f}AED\n")
      print("***************************************************************\n")
      print("Thank you for using Eizyah's Vending Machine!!!")

#this function will ask the user to rate after everything is done.
def rating():
   while True:
      user_input = input("Would you like to rate Eizyah's vending machine (y/n)?:  ")
      if user_input.lower() == "y":
         user_rating = int(input("How would you like to rate Eizyah's vending machine from 1 - 5? "))
         if user_rating == 1 or user_rating == 2 or user_rating == 3:
            print("Thank you for rating my machine! I'll do my best to improve.")
            break
         if user_rating == 4 or user_rating == 5:
            print("Thank you for the excellent rating. I look forward to you using my machine again!")
            break
         else:
            print("You have either exceeded your rate or rated below the given ratings. Please rate again.")
      elif user_input.lower() == "n":
         print("Thank you for using Eizyah's vending machine!!")
         break
      else:
         print("You didnt put a valid answer.")

#puts all the functions into 1
def vending_machine():
   #Displays all the available items.
   display_items()

   #Purchase items and adds them to the cart.
   purchase(cart)

   #Displays the purchased items in the cart.
   display_cart(cart)

   #Ask the user money for the purchased items in the cart.
   cash_received = get_money(cart)
   while cash_received is None:
      cash_received = get_money(cart)

   #Displays all the purchased items in the machine.
   total_price = sum(item[1] for item in cart)
   items_purchased(cart, total_price, cash_received)

   #shows the rating display
   rating()

#runs the vending machine
vending_machine()
```