Middlesex University Mauritius

CST1510 Programming for Data Communication and Networking

CST1510 FINAL COURSEWORK B

VENDING MACHINE

Name: AISHA RAMJAUN

Student Number: M01014015

22/07/2024

Contents

Introduction

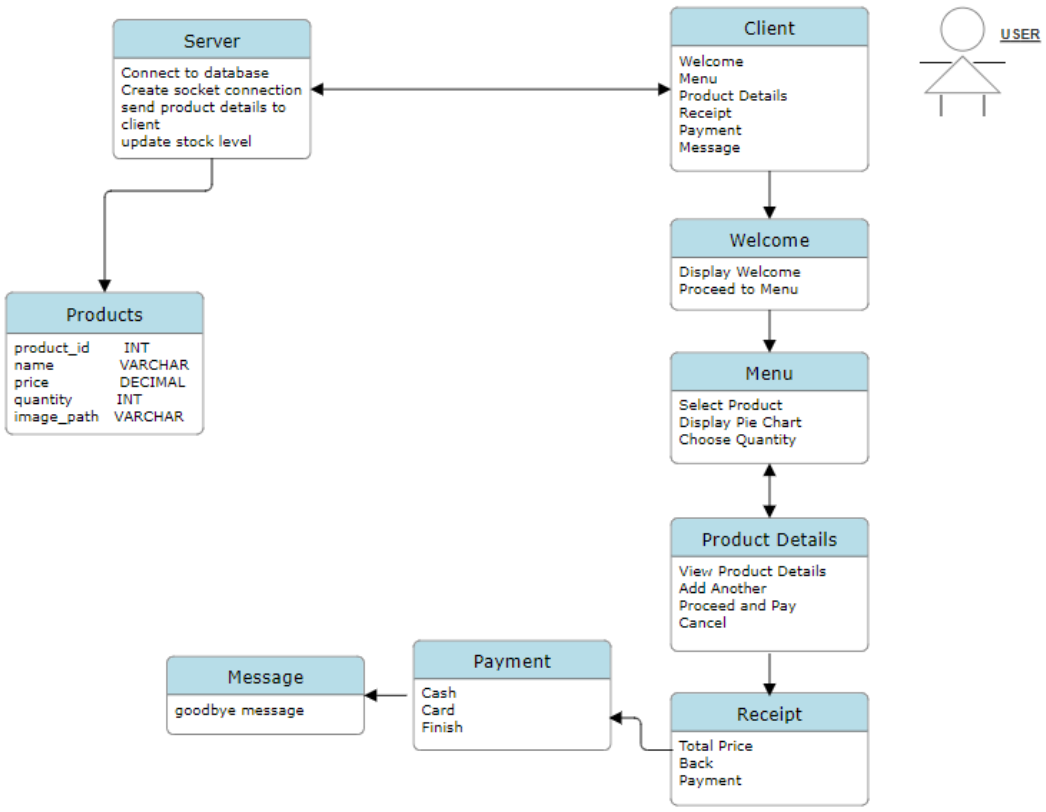
This project is based on designing and implementing a 'Vending Machine' for snacks. I have chosen to sell chocolates. It is a Client/Server-Side software system built in Python to interact with the user on the client-side and the data storage is processed on the server-side.

The vending machine carries out the following operations:

- Start with a welcome message:
- Enter the code of the item he/she wish to buy followed by a number of chosen items.
- Displays the name of the item, number of the items, individual price and total price.
- After each operation the machine allows a list of option: add another, finish and pay or cancel.
- Add another item allows the user to add a new item and again display the above operations. Finish and pay displays the final resit containing the list of items, individual price, total price and leads the user to make a payment.
- The finish and pay option allows the user to go through the payment process
- The payment process mimicks the actual process by providing the client with payment options: cash or card and calculates the change if cash payment made.
- Finally thank you and goodbye message is displayed
- The Cancel option displays a message apologising. Sorry, we could not provide you with what you would like today. We hope to be seeing you again soon. Wish you a Good day!

For this coursework, tkinter was used for the GUI development.
Tkinter is not the only GUI Programming toolkit for python, there are others such as Pygame, but Tkinter is the most commonly used one. It provides a powerful object-oriented interface to the Tk GUI toolkit.

UML Diagram



- **Client Interface:**
  - The client has an intuitive interface with modules for welcome, menu, product details, payment, and receipt handling.
  - Allows viewing available items and prices, creating an order, managing the order, and handling payment processes.

- **Server Functionality:**
  - The server handles database interactions, socket connections, sending product details, and updating stock levels.
  - The server's method to connect to the database and update stock based on transactions aligns with the project requirements.

- **Order Process:**
  - The process starts with a welcome message and allows entering item codes and quantities.
  - Displays item details and total price.
  - Provides options to add items, finish and pay, or cancel.
  - Mimics payment process with cash or card options.
  - Displays a final receipt and a thank you/goodbye message.

- **Database Management:**
    - Products database schema covers all necessary fields.
    - Server updates the stock based on client transactions.

Steps for running

1) Execute a file by the name ‘products.sql’ in the database.
2) Run the server.py file.
3) Run the client.py file to start the application.

Ensure that tkinter, pillow and matplotlib is installed

Server-Side Software System

The server will carry out the following operations:

1) The server will set up the connection to communicate to the client and also with the database.

2) It will retrieve the products details from a database stored in MySQL database service into a list and it will retrieve the quantity of products available from available_stock.txt.

3) The server will send the list to the client.

4) After the order is placed at the client-side, the quantity ordered along with the product ID will be sent back to the server-side.

5) The server will update the quantity in the available_stock.txt.

MySQL Database Service

To connect to a database from python, we have to import MySQL library as below.

```
import mysql.connector
```

In order for the server software system able to extract records from the database, a connection has to be set up.

The figure below shows a screenshot of the code to be amended in server.py. The database name to be inserted is the name of your database as shown

```
HOST = '127.0.0.1'  # Database host
PORT = 3306  # Database port (MySQL default port)

# Establishing a connection to the MySQL database
mydb = mysql.connector.connect(
    host=HOST,                  # Database host
    user='root',                # <--  the username for the database
    password='aisharamjaun',    # <--  the user password for the database
    database='vendingmachine'         # <--  the name of the database to connect to
)
```

Socket Programming

Sockets will act as a point-to-point channel of communication between the client and the server. The socket library was imported as shown below.

```
import socket
```

Threading in communication

When communication is set up, once it received or sent information from/to the client, the communication will terminate. In order, to keep the communication active, threading was used. Threading was imported as shown below.

```
from _thread import *  # For creating new threads
```

Client-side Software System

On the client-side, the GUI will be implemented which will allow the user to carry out the following operations:

1)View a welcome message

2)Choose a snack, the quantity (0 to 10) and place the order.

3) The user can choose another snack.

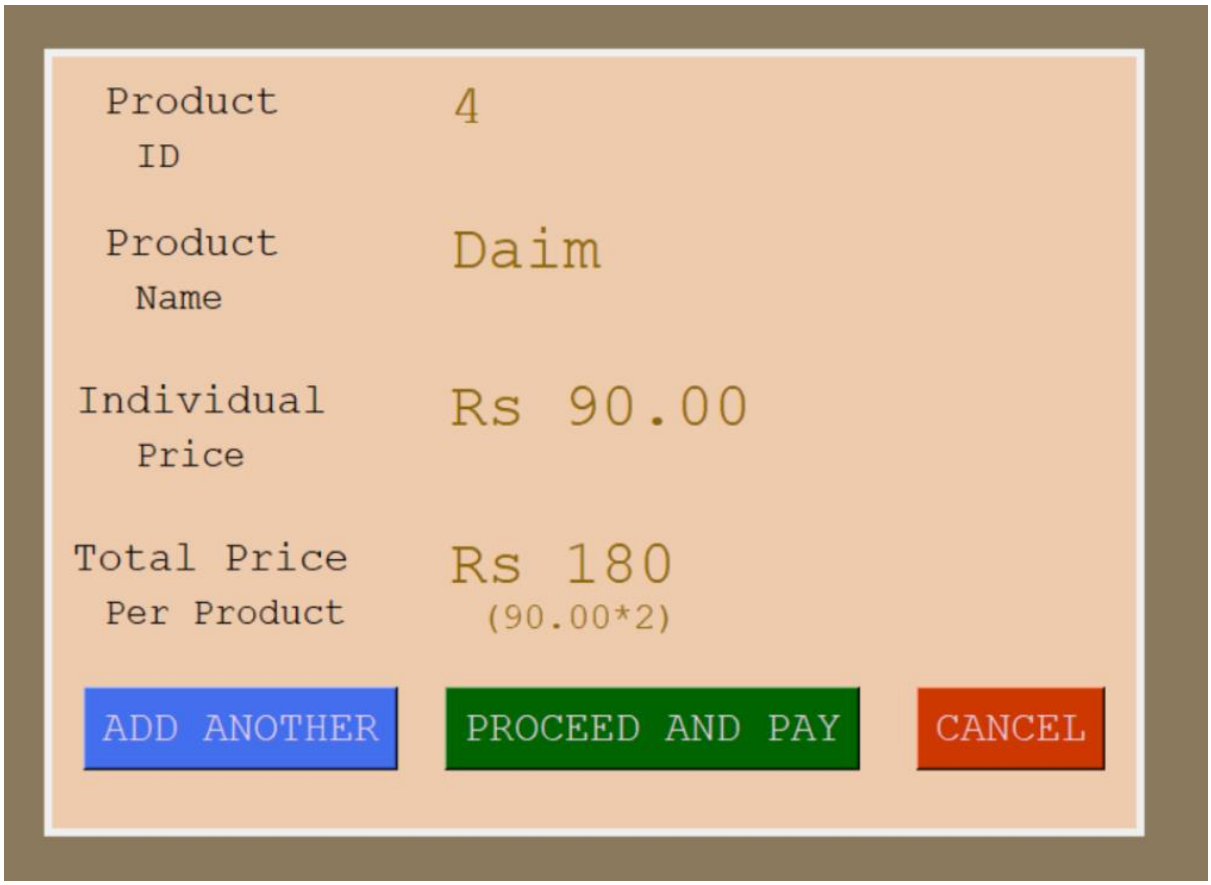
4) The final receipt will be displayed and then, proceeding to pay (either by cash or by card).

Some screenshots of the GUI

A welcome message which leads the user to the menu when clicked on “Order Now!”

The menu showing the id of the chocolates, the stock level and the product details when the user selects a snack.

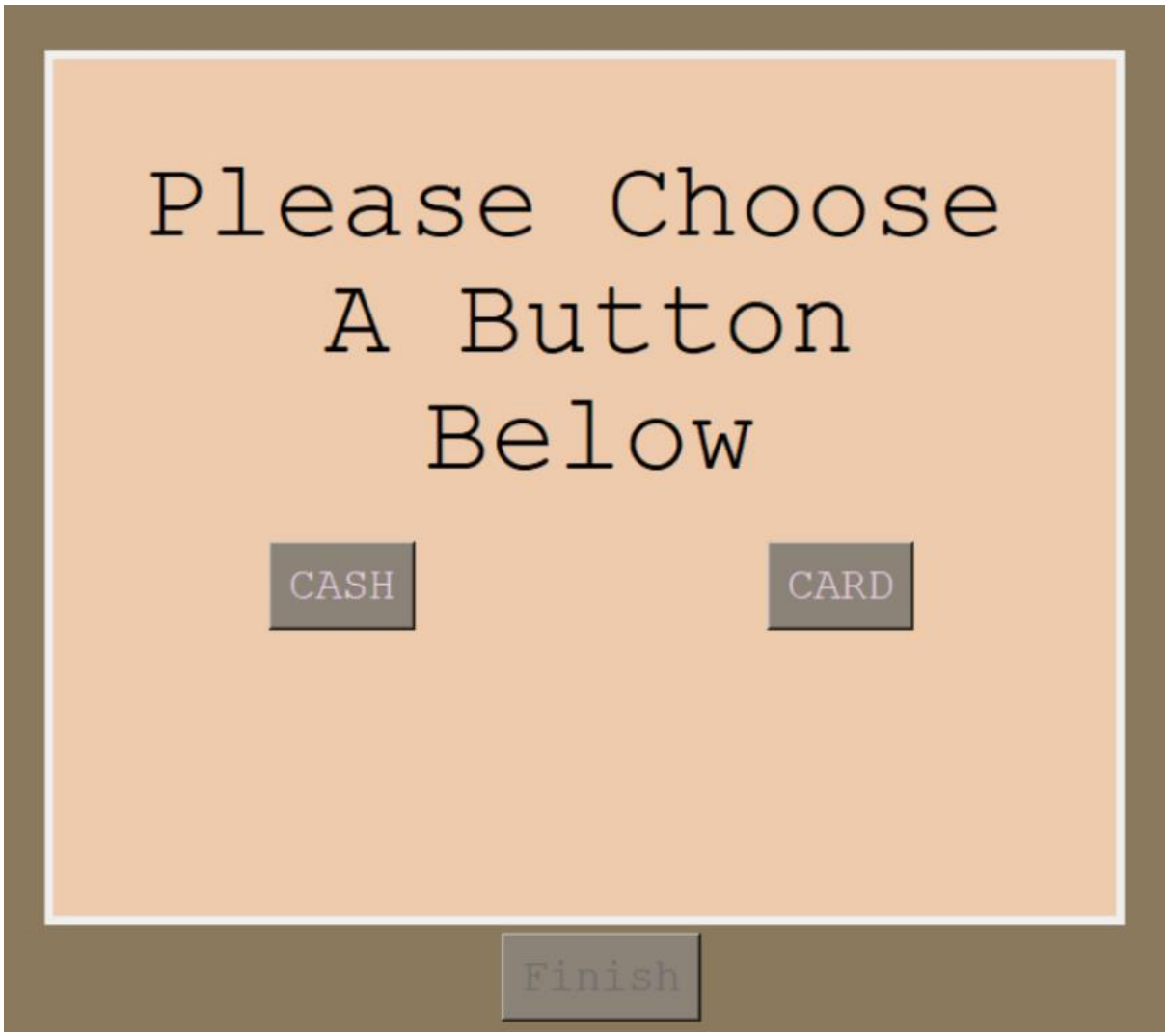
Individual product details



Receipt:


Product ID
Individual Price
Name
Total Price Per Product
4
90.00
Daim
180.0
Total Price: Rs 180.0
Back
Payment

Selecting cash or card method for payment:



If user selects card

If user selects cash

Thank you and goodbye messag

If user selects more products, the receipt will look like this:

Here's the updated pie chart after the following transactions have taken place