Mini Project – 1

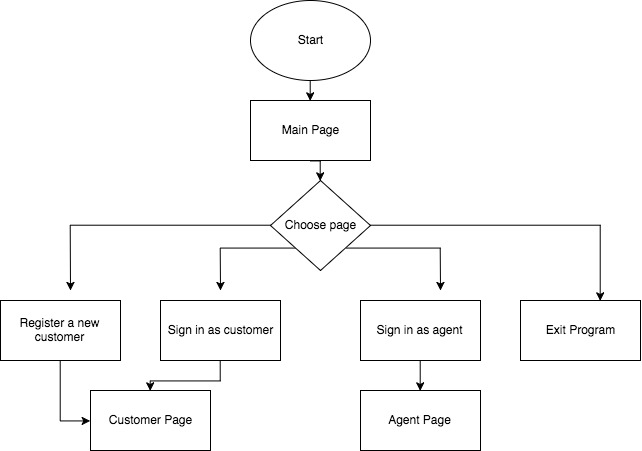
CMPUT – 291

Aman Anand - 1481975

Zhihao Zhang – 1472413

Adit Hasan - 1459800

General Overview:



The program is run at the terminal using the command: python3 MainPage.py -f /path/to/database.db

The program starts with the main page where a user can:

1. register as a new customer   2. sign in as a customer

3. sign in as an agent   4. exit the program

Once a user registers as a customer, they are logged into the customer page. Signing in as a customer also logs the user into the customer page. User is sent to the agent page when they successfully log in as an agent. The selections are done by entering the option number on the terminal and pressing enter. The customer page takes in cid of the customer and the path to the database as arguments. The agent page takes in just the path as an argument.

In the customer page, a user can:

1. Search for products  2. Place orders  3. List orders  4. Log out

Flow of data:

search\_products() : takes the path of the database as the argument and returns the basket

place\_orders() : Takes the basket,cid as arguments and places the orders in the basket

list\_orders() : Takes the cid as the argument and list all orders of the customer

In the agent page, a user can:

1. Set up a delivery  2. Update a delivery  3. Add to stock  4. Log out

Flow of data:

agents\_set\_order(): take the orderid,pickuptime,dropofftime as input and set a new order in database

agents\_add\_stock(): take the storeid,productid as input to add more stock for the product and change the unit price

agents\_update\_order(): take trackingNo as input and user can change the pickup dropoff date and remove the order with inputting extra orderid

Detailed Overview:

Main Page:

SignInCustomer()

* Takes in and validates cid and password from the user
* Matches the required information and on success calls the customer page

SignInAgent()

* Takes in and validates cid and password from the user
* Matches the required information and on success calls the agent page

RegisterCustomer()

* Collects and validates all the required information to register a customer
* Calls SaveCustomer() on success, passes a dictionary of information as argument

SaveCustomer()

* Takes in a dictionary of all required information as parameter
* Saves the customer information into the database

Customer page:

In Search, primary methods used are:

search\_products()

* Main function of Search for managing all functions
* Takes path of the database as the argument
* Calls different functions for different tasks
* Returns the basket

match\_details()

* Takes matched products as the argument
* Collects all the information needed for the product
* Returns the details of the matched products

show\_products(details,index,order)

* Takes the details, index(to ask from where to print), order (to check if orders needs to be printed or products)
* Shows products 5 at a time

Product\_Selected(details)

* Takes details of the products as the argument
* Give the option to select the product and provide details of the product
* Returns selected product details

Basket()

* Takes product information and basket as the arguments
* Adds the products to the basket
* Returns the basket

change\_quantitiy()

* Takes the basket as the argument
* Gives the option to change the quantity multiple times and changes the quantity
* Returns basket

In Place Order, primary methods used are:

check\_quantity()

* Takes the basket as an argument
* Checks if the store quantity is greater than the user quantity
* Returns all the item nos with store quantity for whichever items need to be changed

update\_quantity(item\_no,basket)

* Alerts the user to change/delete the quantity

update\_store\_quantity(basket)

* Takes the basket as the argument
* Updates the store quantity

PlaceOrder()

* Takes cid, basket as the arguments
* Places the order if all products have the required quantity in the store

In List Orders, primary methods used are:

list\_orders(cid)

* Takes cid as an argument
* List all the orders with the provided cid

selected\_order(cid,details)

* Takes the cid and details of the orders
* Prints the information of all products from the order selected

Agent page

enter\_date()

* Get valid date format(y/m/d/h/m/s)
* Keep looping until get the correct format

Agent set order()

* Get valid orderid,pickuptime , dropofftime as input
* Get the random tranckingnum without duplication

remove\_data()

* Match the data with tracking number and ordernum
* Delete the data with tracking number and ordernum

agents\_update\_order()

* Match the data with tracking number
* Update or delete the order by ordernum and tracking number

agents\_add\_stock()

* Match the data with storeid and product id
* Add the stock quantity and update the uniprice

Testing Strategy

Main page:

The main menu was tested with inputs outside the allowed range. To see if the program asks for the input again.

The functionalities in the mainpage was tested for SQL injections and irregular inputs. The inputs can only allow alphanumeric values and the program was tested for error handling.

Customer page:

Testing and bugs fixed:

The basket was tested with deleting items and then placing orders and we found a bug where the item gets deleted but the item number doesn’t gets changed.

The customer part at all places where input is required is protected from any garbage value input or any random input which is not needed.

Agent Page:

The deliveries\_arrived function was tested within different date to find the delivery is arrived or not. We found the the comparison of two date is not functioning.

Group Work Break-down Strategy

BitBucket was used as a private repository to hold the project source code. Regular twice-a-week meetings were used to coordinate efforts and update the team of each member’s progress.

The work done by each member is as follows:

Adit Hasan (~ 3 days)

1. The main page loop
2. Registering new customers
3. Log in as customer
4. Log in as agent
5. SQL injection protection
6. Password hiding

Aman Anand (~ 2 days)

Customer Functionality

1. Searching for products
2. Placing orders
3. Listing orders

Zhihao Zhang (~ 2 days)

Agent Functionality

1. Delivery setup
2. Delivery update
3. Stock adding