



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

OBJECT-ORIENTED PROGRAMMING (SECJ2154)

SEMESTER 2 2023/2024

GROUP PROJECT

GROCERY SHOPPING ASSISTANT

KUGHANRAJ A/L ARUNASALAM (A22EC0179)

NURISH SYAFIQA BINTI HAZLAN (A23CS3016)

ALIA NATASHA BINTI MOHD HAFIZAR (A23CS3007)

SYAZA SYAFIAH BINTI AHMAD AKMAL (A23CS3018)

2 SECR

SECTION 04

Lecturer:

MADAM LIZAWATI MI YUSUF

22nd JUNE 2024

SECTION A: PROJECT DESCRIPTION

1.0 Synopsis

The grocery shopping assistant system is designed for two types of users: **buyers and sellers**. The system provides three main options: **creating a grocery list (for buyers), managing the store (for sellers), and exiting the system**. Both buyers and sellers must log in or register; new users can sign up while existing users can log in with their usernames. Once logged in, buyers have six options: adding an item, editing the quantity, deleting an item, viewing the cart, changing the address, and exiting the system. Sellers, on the other hand, have five options: adding an item, deleting an item, viewing items, editing the address, and exiting the system. Additionally, if a buyer's address matches the seller's address, the **system will recommend nearby shops**. The primary distinction between buyers and sellers in this system is that buyers can edit item quantities, while sellers can edit addresses.

2.0 Objective

The objective is to develop an interface that makes the grocery shopping experience as smooth as possible by **enabling users to build and organize their grocery lists without difficulties**. The system also helps in-store management since it will facilitate **easy creation of stores for the sellers**, store modification, and even the ability to delete a product or change certain aspects of their store. It is intended to improve the user experience by creating an **efficient registration and login process** for the customers and the sellers and incorporating security into the process. Also, it will intelligently **recommend information through the creation of a recommendation system**, where buyers will be provided with relevant local stores to visit on their location and the items in their cart. Cart features will be comprehensive including **adding items to the cart, updating quantities, removing items, viewing the cart, and updating the address on the cart list**.

3.0 Scope

i. User registration and Login

Users need to be able to register or sign up for the page and then have the ability to log into an account while ensuring proper input of their data.

i. Grocery List Management

Allow the buyers to make, modify, and control their grocery lists in a way that lets them add products by categories, modify quantities of added products, and delete products from their cart.

ii. Store Inventory Management

Allow the sellers to arrange products for sale in their stores and let them add new products, remove products, display all products, and edit features such as the business address.

iii. Location-Based Store Recommendations

Develop an option to suggest stores where buyers can purchase the product, based on the provided address that will indicate the stores around the designated area.

4.0 Workflow

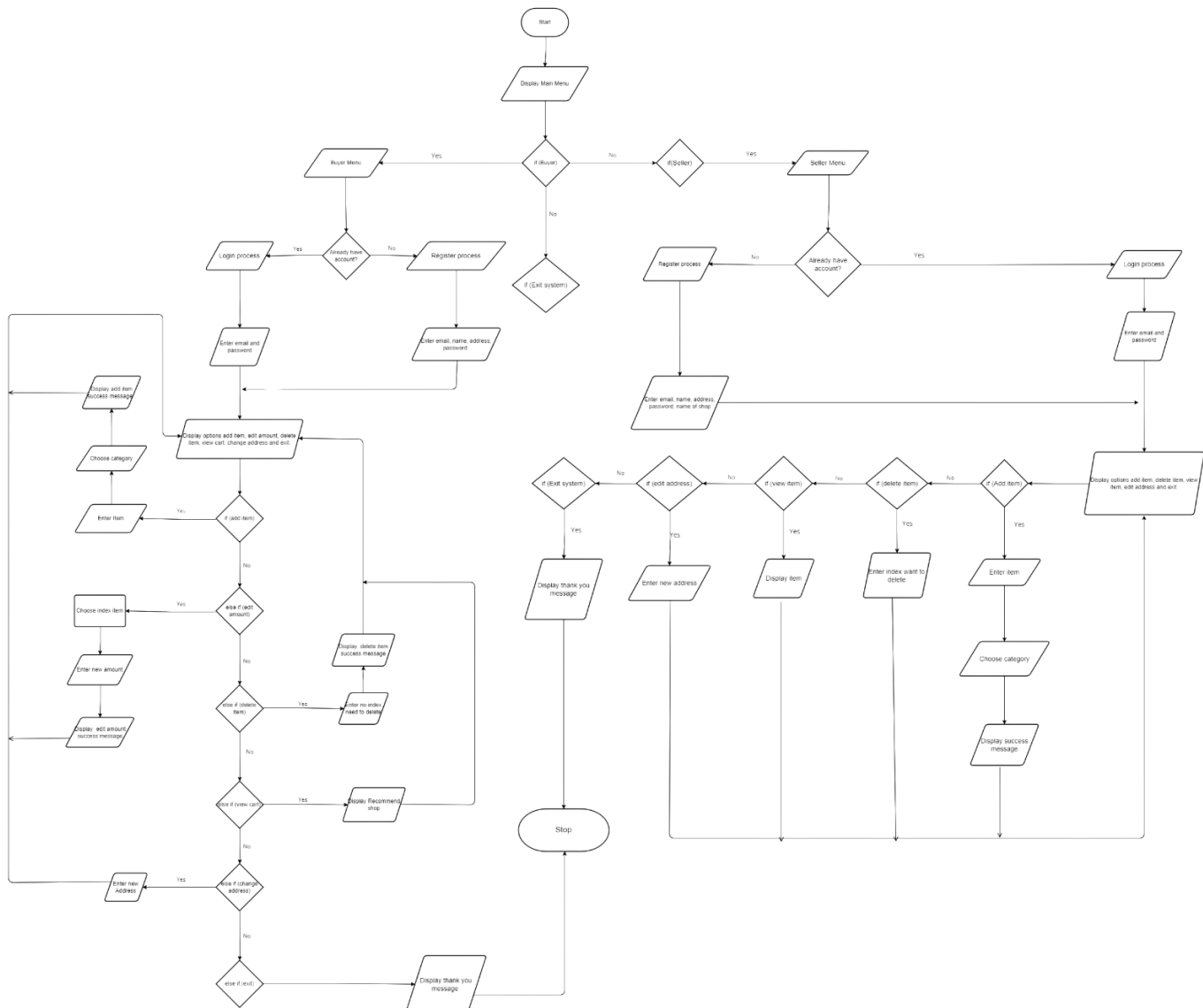


Figure 1

5.0 Object Oriented Concepts

To make a system that is successfully functional, it needs to use object-oriented concepts. Our project uses several key object-oriented concepts, including:

1. Encapsulation

The Cart class encapsulates its data by using private attributes `cart` and `amount`, which are `ArrayList` instances holding `Product` objects and their quantities, respectively.

Example:

```
private ArrayList<Product> cart;
private ArrayList<Integer> amount;
```

2. Data hiding

The attributes are declared as private and some of them have additional access modifiers, such as 'get' or 'set'. This way it is safe because the variables cannot be accessed directly from outside the class. The private attribute can only be accessed and changed by through public method namely the getter and setter method.

Example: Declare as private:

```
private String password;
private String email;
```

Access using public setter method:

```
public void setPassword(String password)
public void setEmail(String email)
```

3. Composition

The `cart` attribute is declared as private within the `Buyer` class. This ensures that the `Cart` object is encapsulated within the `Buyer` class, adhering to the principles of data hiding and encapsulation.

Example:

```
class Buyer extends User {
    private Cart cart;
    public Buyer() {
        cart = new Cart();
    }
}
```

4. Aggregation

Aggregation occurs when one class (Store class) contains a collection of objects from another class (Product). In this relationship, the Store class has a reference to multiple Product objects, but these products can exist independently of the Store.

Example:

```
private String name;

private Category category;

public Product(String name, Category cat) {
    this.name = name;
    category = cat;
}
```

5. Inheritance

Both Buyer and Seller are subclasses of User, meaning they inherit characteristics (methods and fields) from the User class. This allows both Buyer and Seller to use and override methods from User and to add their own specific functionality.

Example:

```
Buyer.java - class Buyer extends User implements
    interfaceBuyer {
    }

Seller.java - class Seller extends User {
    }
```

6. Polymorphism

When you create an ArrayList of User and add instances of Buyer to it, you are leveraging polymorphism. This allows you to treat Buyer objects as User objects, which can be useful when you want to handle different types of users in a unified way.

Example:

```
ArrayList<User> Buyers = new ArrayList<>();

Buyers.add(new Buyer(e, p, n, new Address(st, zipcode,
state, country)));
```

7. Exception handling

The throws IOException in the method signature indicates that this method may throw an IOException. Any caller of this method must handle or declare this exception. The code snippet demonstrates a method saveData that attempts to write data to a file and handles potential IOException using a try-with-resources statement.

Example:

```
public void saveData() throws IOException { try
    (PrintWriter writer = new PrintWriter(new
    FileWriter(filename))) {
    }
    catch (IOException e) { System.out.println("Error :
    File not found"); }
```

8. Enumeration

In Java, an enum belongs to a special class of data types that are used to declare sets of constants. Here, Category is an enum that defines the possible categories of items that can be stored. Category enum has each constant mapped to a string description. These are in fact invocations of the enum constructor passing the string arguments.

Example:

```
public enum Category {
    FRUITS("Fruits"), VEGETABLES("Vegetables"),
    DAIRY("Dairy"), MEAT("Meat"), BAKERY("Bakery"),
    BEVERAGES("Beverages"), SNACKS("Snacks"), OTHERS("Other
    s");

    private String desc;

    private Category(String s) {
        desc = s;
    }

    public String toString() {
        return desc;
    }
}
```

SECTION B: CLASS DIAGRAMS

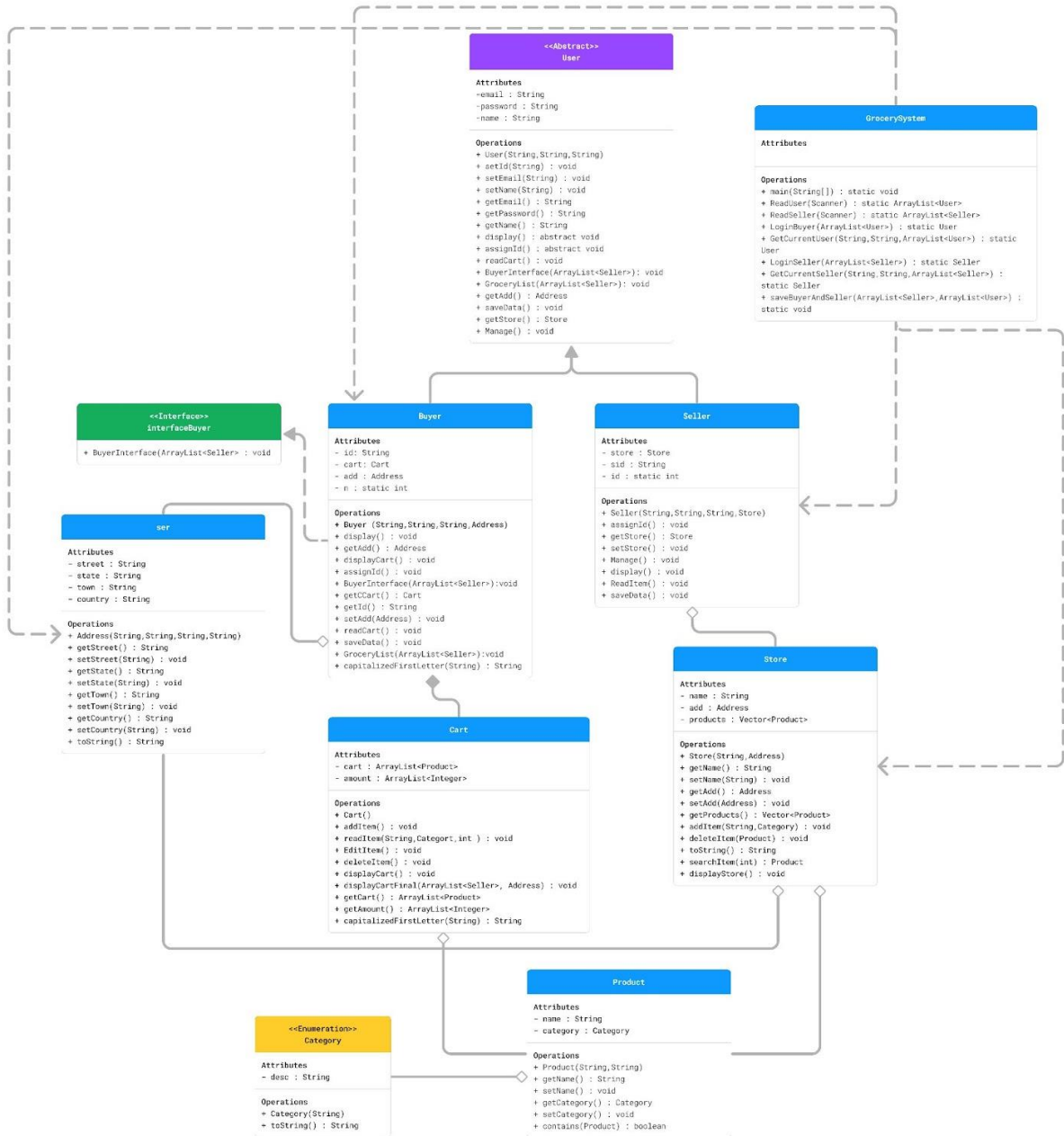


Figure 2

For a clearer view: [CLASS DIAGRAM](#)

Abstract Class User:

Attributes	Description
email	Email of the user
password	Password of the user
name	Name of the user
Methods	Description
User (String, String, String)	Constructor to initialize item with given attributes
setId (String) : void setEmail (String) : void setName (String) : void getEmail () : String getPassword () : String getName () : String	Mutator and Accessor the private attributes
display () : abstract void assignId () : abstract void	Abstract method display Abstract method to assign Id
readCart () : void BuyerInterface (ArrayList<Seller>) : void GroceryList (ArrayList<Seller>) : void getAdd () : Address saveData () : void getStore () : Store Manage () : void	Methods to be overridden later in child class

Class Buyer:

Attributes	Description
id : String	Unique identifier for the buyer files
cart : Cart	A class to keep the items of the buyer and their amounts
add : Address	A class that has the address of the buyer
n : static int	Used to assign Id to the buyer incrementally
Operations	Description
Buyer(String,String,String, Address)	Constructor to initialize the attributes
getCCart() : Cart getId() : String setAdd(Address) : void getAdd() : Address	Accessor and Mutator for the attributes
display() : void	The method to display buyer name,email,password and address
displayCart() : void	The method to display the content of the cart
BuyerInterface(ArrayList<Seller>):void	The method that has the buyer menu, where buyer choose what to do in the system
assignId(): void	The method used to assign id automatically to the user
readCart(): void	The method that read the contents of the cart from a csv file and save it to the cart of the buyer
saveData():void	The method that saves the contents of the cart to a csv file with the user id.
GroceryList(ArrayList<Seller>):void	A method that will produce a txt file containing the cart content and recommended shops.

Class Seller:

Attributes	Description
store : Store	A store that is managed by the seller
sid : String	Unique identifier for the seller files
id : static int	Used to assign Id to the seller incrementally
Operations	Description
Seller(String, String, String, Store)	Constructor to initialize the attributes
assignId()	The method used to assign id automatically to the user
getStore() : Store setStore() : void	Accessor and mutator for the store attribute.
Manage() : void	The method that has the seller menu, where seller choose what to do in the system
display() : void	The method that the seller name, email and the store he manages
ReadItem() : void	The method that read the contents of the store products from a csv file and save it to the store class of the seller
saveData():void	The method that saves the contents of the store to a csv file with the sid.

Class Store:

Attributes	Description
name : String	Name of the store
add : Address	Address of the store
products : Vector<Product>	Products that the store has
Operations	Description
Store(String, Address)	Constructor to initialize the attributes
getName() : String setName(String) : void getAdd() : Address setAdd(Address) : void getProducts() : Vector<Product>	Accessor and mutator for the store class
addItem(String, Category) : void	Method to add the product into the products vector
deleteItem(Product) : void	Method to delete the item inside the products vector
toString() : String	Method to return a string with store name and address
searchItem(int) : Product	Method that return a product inside the products vector with the given index
displayStore() : void	Method to display the contents of the products vector with their name and category

Class Cart

Attributes	Description
cart : ArrayList<Product>	An arraylist of product class that the buyer wants to buy
amount : ArrayList<Integer>	An arraylist of integer that is used to keep track of the quantity of the item the buyer wants to buy
Operations	Description
Cart()	Constructor to initialize the attributes
addItem() : void	Method that the buyer use to add product into cart and quantity into amount arraylists.
readItem(String,Category,int) : void	Method to add a product into cart and quantity ; used for file operation.
EditItem() : void	Method used to change the quantity of the product that is entered before
deleteItem() : void	Method to delete the item from the cart and amount arraylist.
displayCart() : void	Method to show all the products name, category and the amount in the arraylists
displayCartFinal (ArrayList <Seller>,Address) : void	Method to show all the products name , category and amount in the arraylist with recommended shops
getCart() : ArrayList<Product> getAmount() : ArrayList<Integer>	Accessor for the cart class.
capitalizedFirstLetter (String) : String	Method to capitalize the first letter of a string to be a better display.

Class Address:

Attributes	Description
String street	Street name
String state	State name
String town	Town name
String country	Country name
Operations	Description
Address(String, String, String, String)	Constructor to initialize the attributes
getStreet() : String setStreet(String) : void getState() : String setState(String) : void getTown() : String setTown(String) : void getCountry() : String setCountry(String) : void	Accessor and Mutator for the attributes
toString() : String	A method that returns a string containing street, town, state and country.

Class Product

Attributes	Description
name : String	Name of the product
category : Category	Category of the product
Operations	Description
Product(String, String)	Constructor to initialize the attributes
getName() : String setName() : void getCategory() : Category setCategory() : void	Mutator and Accessor for the attributes.
contains(Product) : boolean	A method to check whether the products is same as the a product

Enum Category

Attributes	Description
desc : String	The description of the category
Operations	Description
Category(String)	Constructor to initialize the attribute
toString() : String	A method to return the attribute desc

Interface interfaceBuyer

Attributes	Description
Operations	Description
BuyerInterface (Array List <Seller> : void	An abstract interface method that will be implemented by the class user.

GrocerySystem

Attributes	Description
Operations	Description
main(String[]) : static void	Main method of the system
ReadUser (Scanner) : static ArrayList<Buyer>	Method used to read the Buyers from a csv file and return an arraylist of the Buyer
ReadSeller (Scanner) : static ArrayList<Seller>	Method used to read the Sellers from a csv file and return an arraylist of the Buyer
LoginBuyer (ArrayList<Buyer> >) : static Buyer	Method to ask for the user to login or sign up (register as a new buyer)
GetCurrentUser (String, String, ArrayList<Buyer>) : static Buyer	Method to get the specific buyer from the already registered before.
LoginSeller (ArrayList<Seller> >) : static Seller	Method to ask for the user to login or sign up (register as a new seller)
GetCurrentSeller (String, String, ArrayList<Seller>) : static Seller	Method to get the specific seller from the already registered before.
saveBuyerAndSeller (ArrayList<Seller>, ArrayList<Buyer>) : static void	Save all the buyer and seller information including their cart (for buyer) and store (for seller) into predetermined csv file.

SECTION C: USER MANUAL

USER MANUAL

Main Menu

```
-----  
Hello and Welcome to Grocery Shopping Assistant!  
-----  
Press 1 is you want to make a grocery list!  
Press 2 if you want to manage your store!  
Press 3 to exit the system!  
Option: █
```

This simple main menu will show when the user runs the program Java on the main menu. These 3 options have three different purposes, and the purpose of each of these options is to:

Press 1: The buyer can log in and register to enter and make the grocery list.

Press 2: The seller can manage their store via login and register by choosing this option.

Press 3: Exit the system.

The user must choose and enter one of the options.

Enter Invalid Option

```
Press 1 is you want to make a grocery list!
Press 2 if you want to manage your store!
Press 3 to exit the system!
Option: 4
Invalid Option
-----
Hello and Welcome to Grocery Shopping Assistant!
-----
Press 1 is you want to make a grocery list!
Press 2 if you want to manage your store!
Press 3 to exit the system!
Option: █
```

If the user enters the incorrect option other than 1 to 3, it will display:

“Invalid Option”

Then, it will loop back to the main menu.

Buyer Register

```
-----  
Hello and Welcome to Grocery Shopping Assistant!  
-----  
Press 1 is you want to make a grocery list!  
Press 2 if you want to manage your store!  
Press 3 to exit the system!  
Option: 1  
  
Press 1 to login!  
Press 2 for register!  
Option: 2  
  
Please enter your email: hilman@gmail.com  
Please enter your name: hilman  
Please enter your Address  
Street: Jalan Ibrahim  
Town: Taiping  
State: Perak  
Country: Malaysia  
Please enter your password: hilman10  
Please re-enter your password: hilman10  
  
1) Add Item  
2) Edit Quantity  
3) Delete Item  
4) View Cart  
5) Change Address  
6) User Menu  
7) Exit  
Option(1-7): █
```

When the user enters the option 1 to make a grocery list, it will allow the user or buyer to login and register. Press 2 for register and the user need to enter their information details such as email, name, address, and password. Once done entered the details, it will display the choices for the user to make a grocery list. Those choices are:

1. **Add Item:** The user can add items such as food and others as they want.
2. **Edit Quantity:** Users can change the quantity by editing it.
3. **Delete Item:** The user also can delete the item that they already add.
4. **View Cart:** This will show the item details.
5. **Change Address:** The user can change their address.
6. **User Menu:** This will display the information details of the user.
7. **Exit:** Out of the program system.

Add Item

```
*****
Add Item
*****
Please enter the item: banana
Please enter the Category [FRUITS(F), VEGETABLES(V), DAIRY(D), MEAT(M), BAKERY(B), SNACKS(S), OTHERS(O)]: F
Please enter the amount: 2
Item successfully added!

1) Add Item
2) Edit Quantity
3) Delete Item
4) View Cart
5) Change Address
6) User Menu
7) Exit
Option(1-7): █
```

This will display when the user enters the add item. The user needs to enter the item and choose what category of the item they want. After enter the amount, it will show that “**Item successfully added!**”.

Edit Quantity

```
1) Add Item
2) Edit Quantity
3) Delete Item
4) View Cart
5) Change Address
6) User Menu
7) Exit
Option(1-7): 2
No.  Item          Category  Amount
1)  banana        Fruits   2
2)  butter         Dairy    2
Edit Quantity
Please choose the index of item to edit: 1
Please enter the new quantity: 4
Change has been applied!
```

When users enter the edit quantity, it will generate the item, category and amount. Then, the user needs to choose the index of the item to edit and enter the new amount. After that, it will show **“Change has been applied!”**.

Display Edit Quantity in Cart

```
1) Add Item
2) Edit Quantity
3) Delete Item
4) View Cart
5) Change Address
6) User Menu
7) Exit
Option(1-7): 4
```

No.	Item	Category	Amount	Recommended Shop
1)	Banana	Fruits	4	
2)	Butter	Dairy	2	

To see if the edit amount has already changed, the user can choose the option view cart. The amount of item “**Banana**” is changed to 4.

Delete Item

```
1) Add Item
2) Edit Quantity
3) Delete Item
4) View Cart
5) Change Address
6) User Menu
7) Exit
Option(1-7): 3
No. Item          Category Amount
1) banana         Fruits    4
2) butter         Dairy     2

Please enter the item you want to delete: 2
No. Item          Category Amount
1) banana         Fruits    4
```

The user can delete an item by choosing option number 3 which is “**Delete Item**”. They need to enter the number of items they want to delete. Then, it will only display the remaining item.

Change Address

```
1) Add Item
2) Edit Quantity
3) Delete Item
4) View Cart
5) Change Address
6) User Menu
7) Exit
Option(1-7): 5
Please enter the new address!
Street: Jalan Abdul Rahman
Town: Taiping
State: Perak
Country: Malaysia
```

The user can enter their new address by choosing the option “**Change Address**”.

User Menu

```
1) Add Item
2) Edit Quantity
3) Delete Item
4) View Cart
5) Change Address
6) User Menu
7) Exit
Option(1-7): 6

Name: hazel
Email: hazel@gmail.com
Password: hazel@gmail.com
Address: Jalan Permata, Perak, Ipoh, Malaysia
```

When the user chooses option number 6 which is the user menu, it will display the information details of the user.

Enter Exist Account in Register

```
-----  
Hello and Welcome to Grocery Shopping Assistant!  
-----  
Press 1 is you want to make a grocery list!  
Press 2 if you want to manage your store!  
Press 3 to exit the system!  
Option: 1  
  
Press 1 to login!  
Press 2 for register!  
Option: 2  
  
Please enter your email: ardean@gmail.com  
There is already a account with this email!  
Please try again!  
-----  
Hello and Welcome to Grocery Shopping Assistant!  
-----  
Press 1 is you want to make a grocery list!  
Press 2 if you want to manage your store!  
Press 3 to exit the system!  
Option: █
```

When the user enter the account that already registered, it will display:

“There is already an account with this email!”

“Please try again!”

It will loop back to the main menu.

Error Login Seller

```
-----  
Hello and Welcome to Grocery Shopping Assistant!  
-----  
Press 1 is you want to make a grocery list!  
Press 2 if you want to manage your store!  
Press 3 to exit the system!  
Option: 2  
  
Press 1 to login!  
Press 2 for register!  
Option: 1  
  
Please enter your Email: hazel@gmail.com  
Please enter your password: hazel  
Error  
Thank you for using our system!
```

The seller can manage their store by option number two. If the user wants to login that email that does not register yet, it will show an error.

Seller Register

```
-----  
Hello and Welcome to Grocery Shopping Assistant!  
-----  
Press 1 is you want to make a grocery list!  
Press 2 if you want to manage your store!  
Press 3 to exit the system!  
Option: 2  
  
Press 1 to login!  
Press 2 for register!  
Option: 2  
  
Please enter your email: kiahmart@gmail.com  
Please enter your name: rokiah  
Please enter your Address  
Street: Jalan Emas  
Zipcode: 40010  
State: Perak  
Country: Malaysia  
Please enter your password: rokiah  
Please re-enter your password: rokiah  
Please enter your shop name: Kiah Mart  
  
1) Add Item  
2) Delete Item  
3) View Item  
4) Edit Address  
5) User Menu  
6) Exit  
Option: █
```

The user needs to register the account by entering their information and shop name.

Add Seller Item

```
1) Add Item  
2) Delete Item  
3) View Item  
4) Edit Address  
5) User Menu  
6) Exit  
Option: 1  
  
Please enter the name: super ring  
Please enter the Category [FRUITS(F), VEGETABLES(V), DAIRY(D), MEAT(M), BAKERY(B), SNACKS(S), OTHERS(O)]: S  
Item successfully added!  
No.  Item      Category  
1)  lettuce    Vegetables  
2)  banana     Fruits  
3)  apple      Fruits  
4)  super ring  Snacks
```

The item and category will display when added.

Add

Exist Item Seller

```
1) Add Item
2) Delete Item
3) View Item
4) Edit Address
5) User Menu
6) Exit
Option: 1

Please enter the name: apple
Please enter the Category [FRUITS(F), VEGETABLES(V), DAIRY(D), MEAT(M), BAKERY(B), SNACKS(S), OTHERS(O)]: F
Item already exists in List!
No.  Item          Category
1)  lettuce        Vegetables
2)  banana          Fruits
3)  apple           Fruits
4)  super ring      Snacks
```

If the user enters the existing item, it will show that **“Items already exist in List!”**.

Delete Invalid Index Seller Item

```
1) Add Item
2) Delete Item
3) View Item
4) Edit Address
5) User Menu
6) Exit
Option: 2
No.  Item          Category
1)  banana          Fruits
2)  apple           Fruits
3)  super ring      Snacks
4)  lettuce        Vegetables
Please enter the index of the item: 6
Sorry the item you have searched for cannot be found!
```

When the user enters the index of the item that does not have, it will generate **“Sorry the item you have searched for cannot be found!”**

Recommend Shop for Buyer

```
Name: hazel
Email: hazel@gmail.com
Password: hazel@gmail.com
Address: Jalan Sepang, Johor Bahru, Johor, Malaysia
```

- 1) Add Item
- 2) Edit Quantity
- 3) Delete Item
- 4) View Cart
- 5) Change Address
- 6) User Menu
- 7) Exit

Option(1-7): 4

No.	Item	Category	Amount	Recommended Shop	
1)	Egg	Dairy	10	Jaya Grocer	Jalan Sepang, Johor Bahru, Johor, Malaysia
2)	Milk	Dairy	1	Jaya Grocer	Jalan Sepang, Johor Bahru, Johor, Malaysia
3)	Onion	Vegetables	8	Jaya Grocer	25 Jalan Bukit Bintang, Johor Bahru, Johor, Malaysia
4)	Donut	Bakery	4	Jaya Grocer	Jalan Sepang, Johor Bahru, Johor, Malaysia
5)	Pretzels	Snacks	2	Econsave	15 Jalan Tebrau, Johor Bahru, Johor, Malaysia
				Jaya Grocer	25 Jalan Bukit Bintang, Johor Bahru, Johor, Malaysia

The user can view their cart to see recommended shops near them for the items. It will suggest to the user which store is located that has the item they want near their address.

Exit

```
1) Add Item
2) Edit Quantity
3) Delete Item
4) View Cart
5) Change Address
6) User Menu
7) Exit
Option(1-7): 7
Thank you for using our system!
```

If the user wants to exit the program, enter option 7.