



GROCERY SHOPPING ASSISTANT

SECJ2154 - 04



Prepared By : Group 2

OUR TEAM

**KUGHANRAJ A/L
ARUNASALAM**

A22EC0179

**NURISH SYAFIQA
BINTI HAZLAN**

A23CS3016

**ALIA NATASHA BINTI
MOHD HAFIZAR**

A23CS3007

**SYAZA SYAFIAH
BINTI AHMAD AKMAL**

A23CS3018




PROJECT DESCRIPTION





SYNOPSIS

- two types of users: buyers and sellers.
 - The system provides three main options:
 - Creating a grocery list (for buyers)
 - Managing the store (for sellers)
 - Exiting the system
 - buyers have 6 options:
 - Add an item
 - Edit quantity
 - Delete an item
 - View cart
 - Change address
 - Exit system
- 




SYNOPSIS

- Seller have 5 options:

- Add an item
- Delete an item
- View cart
- Change address
- Exit system

- buyers and sellers :

- Log in
- Register

- buyer's address matches the seller's address, the system will recommend nearby shops.
- 

OBJECTIVE



- To make it easier for users to create and organize their grocery lists.
- To allow sellers to quickly set up their stores.
- To ensure the registration and login process is quick and simple.
- To develop a system that recommends shops to users based on their preferences.

SCOPE OF PROJECT

USER REGISTRATION AND LOGIN

Users able to register for an account, ensuring their data is properly entered, and then be able to log in.

GROCERY LIST MANAGEMENT

Allow buyers to create, edit, and manage grocery lists by adding products by category, adjusting quantities, and deleting items.

SCOPE OF PROJECT

STORE INVENTORY MANAGEMENT

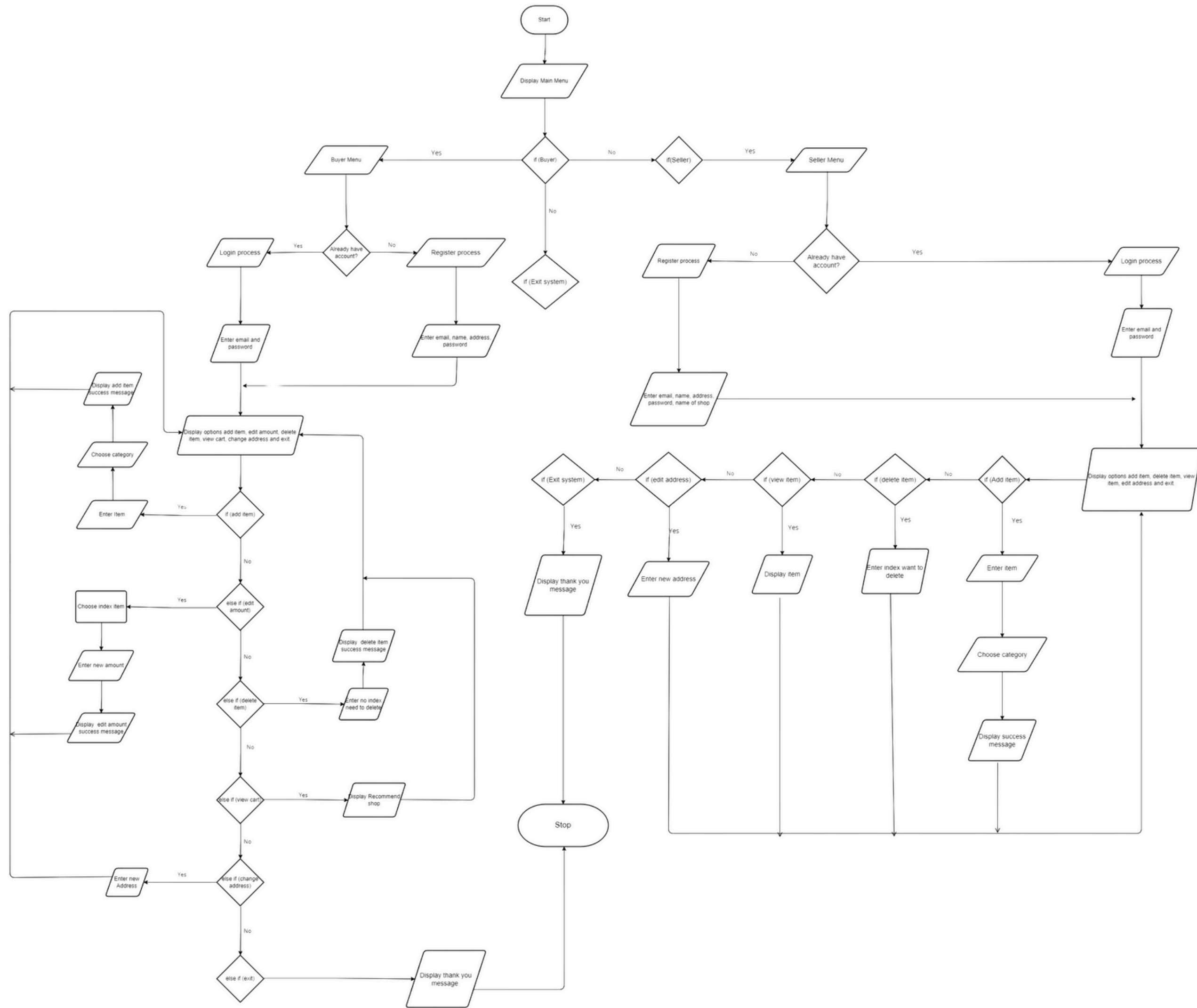
Allow sellers to manage their store by adding, removing, displaying, and editing products, and updating the business address.

LOCATION-BASED STORE RECOMMENDATIONS

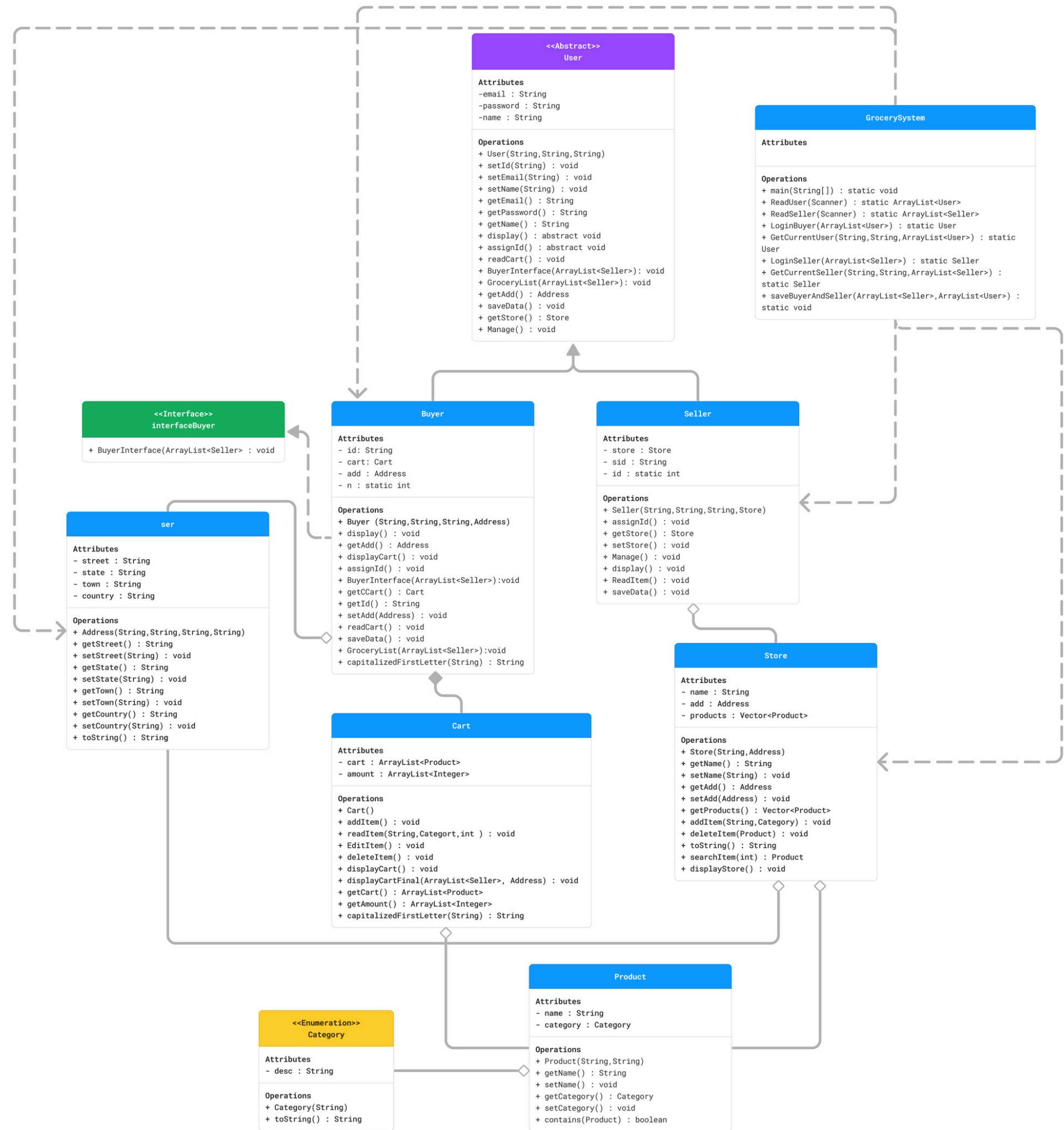
Develop a feature to suggest nearby stores for buyers based on their provided address.

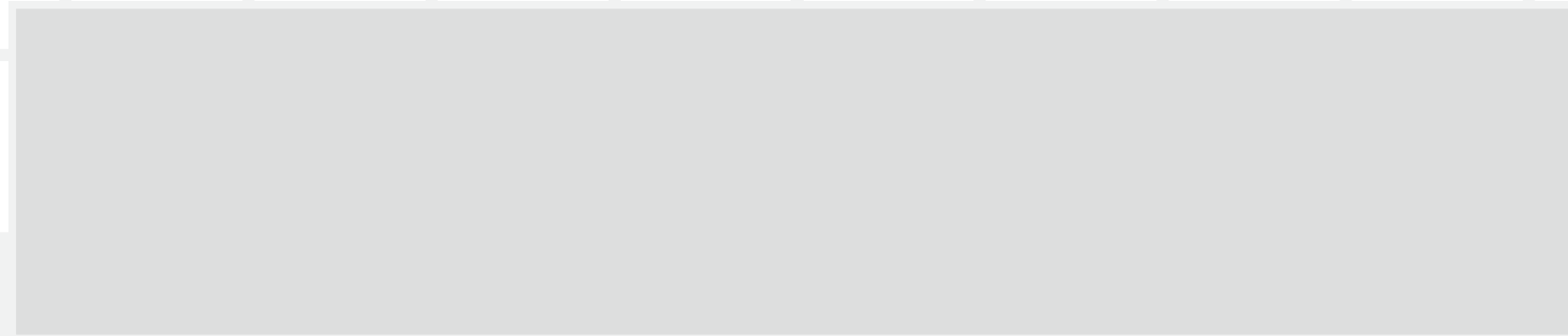


PROJECT DESIGNS



FLOWCHART





OO CONCEPTS EMPLOYED



OBJECT ORIENTED CONCEPTS



ENCAPSULATION

Cart.java

- **Private Attributes:**

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

- **Public Methods:**

```
public void readItem(String n, Category ca, int am)  
public void EditItem()
```

DATA HIDING

User.java

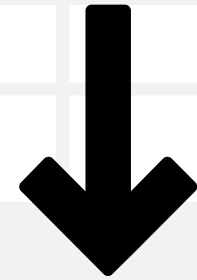
- **Declare as private:**

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

- **Access using public setter method:**

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

OBJECT ORIENTED CONCEPTS



COMPOSITION

Buyer.java

```
class Buyer extends User {  
private Cart cart;
```

```
public Buyer() {  
cart = new Cart();  
}
```



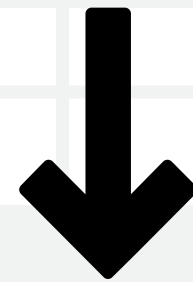
AGGREGATION

Product.java

```
private String name;  
private Category category;
```

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

OBJECT ORIENTED CONCEPTS



INHERITANCE

Buyer.java

- class Buyer **extends** User implements interfaceBuyer {

Seller.java

- class Seller **extends** User {

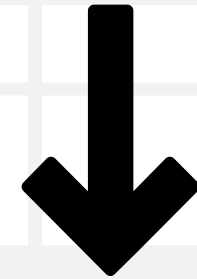


POLYMORPHISM

GrocerySystem.java

```
ArrayList<User> Buyers = new ArrayList<>();  
Buyers.add(new Buyer(e, p, n, new Address(st,  
zipcode, state, country)));
```

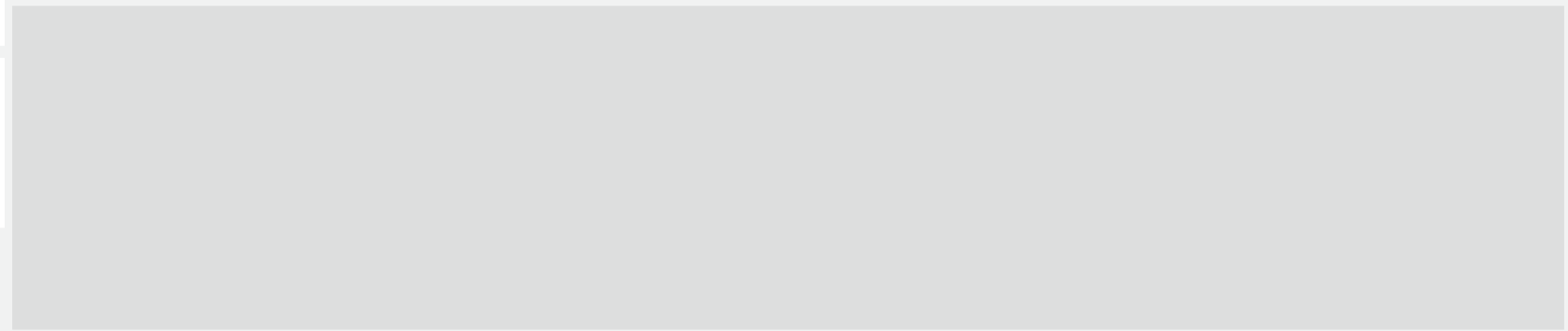
OBJECT ORIENTED CONCEPTS



EXCEPTION HANDLING

Seller.java

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

DEMONSTRATION SYSTEM





THANK YOU!

