1. Introduction and Purpose of the Project

1.1 Introduction

The idea of our project is to design and develop a supermarket management system that aims to streamline the checkout process for customers and improve the efficiency of the store's operations. The system will allow customers to collect their items and automatically calculate the total cost. It will also keep track of inventory and sales data, allowing store managers to have real-time visibility into their business.

1.2 Purpose of the Project:

The purpose of this project is to provide a convenient and efficient way for customers to pay for their purchases at the supermarket. It aims to reduce the time and effort required for the checkout process, which will ultimately improve the customer experience and increase customer satisfaction. Additionally, the system will provide valuable data for store managers, such as sales and inventory data, which can be used to make informed business decisions. Overall, the goal of the supermarket management system is to improve the store's operations and enhance the customer experience.

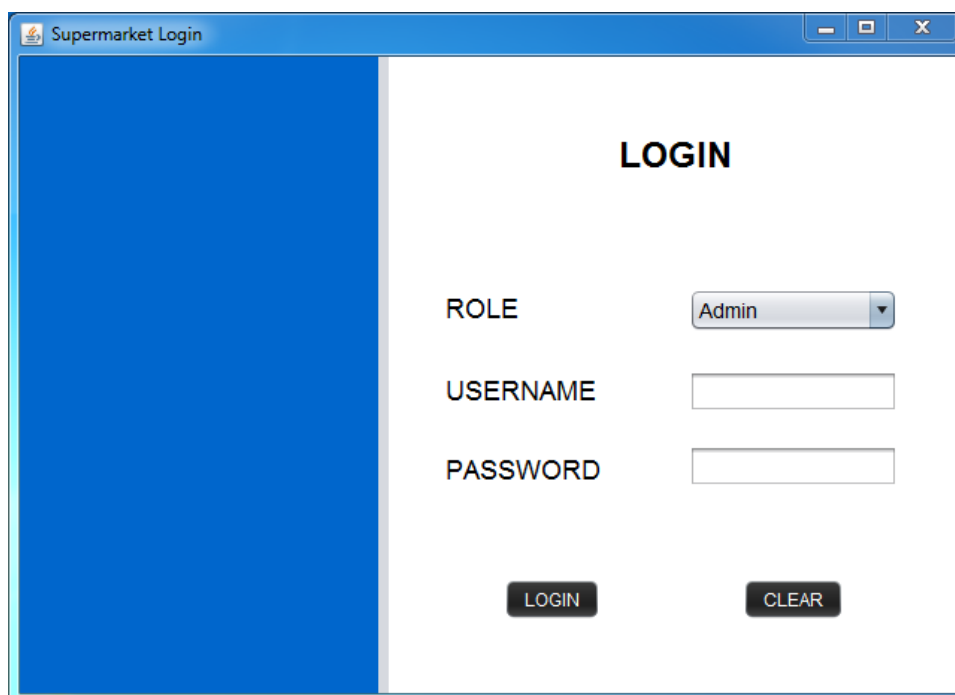## 2. Features and Functionality of the Project

2.1 Overview

- Admin can add Sellers, Categories and Products under categories
- Seller can add products to Cart as per Customer's requests
- A Bill is printed out as PDF
- A Database is maintained for all Sellers, Categories and Products

2.2 Functions in detail

2.2.1 Login Page

On the login page, the user will be able to select their role as either an admin or a seller. This will determine the features and functionality that the user has access to within the system.

The user will be required to enter their login credentials, such as their username and password, in order to access the system.
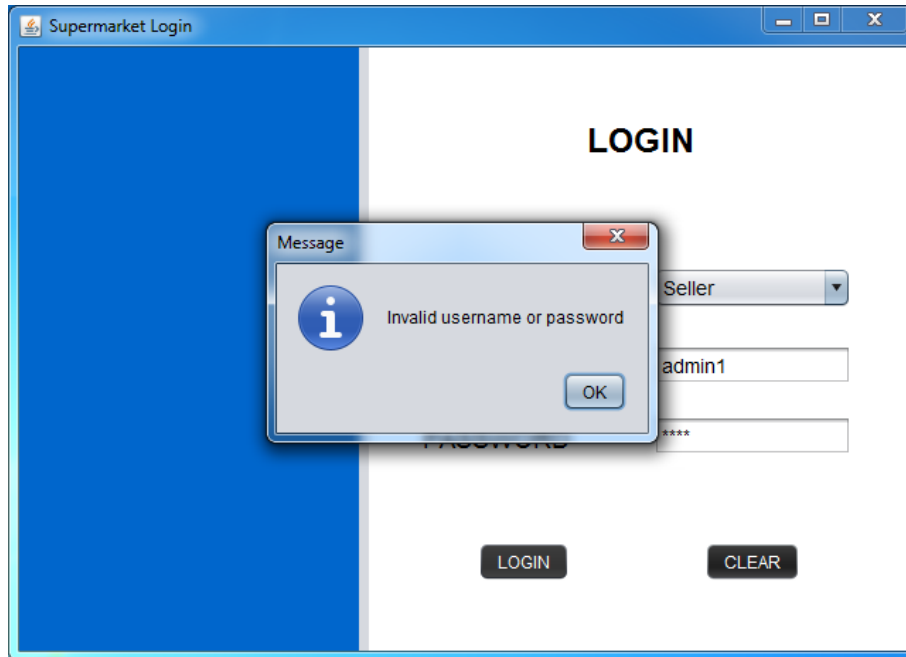
2.2.2 Admin Module
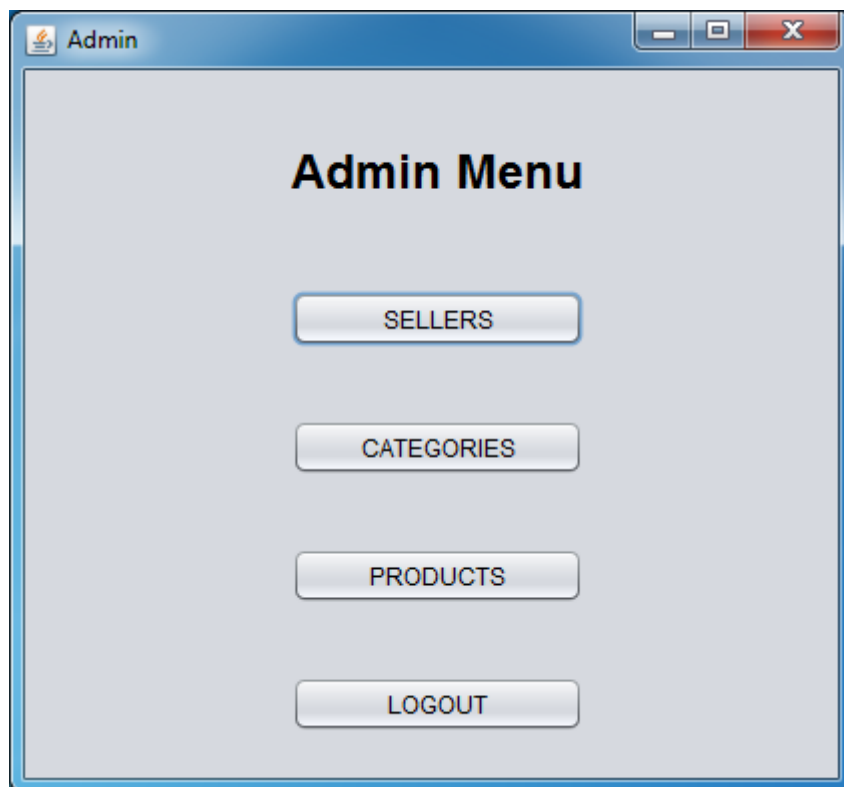
a. Manage Sellers

- View all sellers registered in the database
- Add new seller

- Edit seller's information
- Delete seller

b. manage categories

- View all categories registered in the database

- Add new category

- Edit category information

- Delete category

c. manage products under categories

- View all products registered in the database

- Add new product

- Edit product information

- Delete product

MANAGE PRODUCTS

PRODID		QUANTITY	
NAME		PRICE	
CATEGORY	Groceriess ▼		

Add	Edit	Delete	Clear

PRODUCT LISTS

PRODID	PRODNAME	PRODQTY	PRODPRICE	PRODCAT
2	Diet Coke	40	35.0	Drinks and Beverages
3	Milkshake	50	45.55	Drinks and Beverages
4	Moisturizer	8	180.0	Cosmetics
5	Face Wash	10	80.0	Cosmetics
6	Milk	15	10.0	Dairy Products
7	Orange	100	5.65	Groceries

MANAGE PRODUCTS

PRODID	2	QUANTITY	40
NAME	Diet Coke	PRICE	35.0
CATEGORY	Drinks and Beverages ▼		

WARNING

ⓘ Are you sure you want to delete the data?

Yes	No

PRODID	PRODNAME	PRODQTY	PRODPRICE	PRODCAT
2	Diet Coke	40	35.0	Drinks and Beverages
3	Milkshake	50	45.55	Drinks and Beverages
4	Moisturizer	8	180.0	Cosmetics
5	Face Wash	10	80.0	Cosmetics
6	Milk	15	10.0	Dairy Products
7	Orange	100	5.65	Groceries

MANAGE PRODUCTS

PRODID	2	QUANTITY	40
NAME	Diet Coke	PRICE	35.0
CATEGORY	Drinks and Beverages ▼		

Add	Edit	Delete	Clear

PRODUCT LISTS

PRODID	PRODNAME	PRODQTY	PRODPRICE	PRODCAT
3	Milkshake	50	45.55	Drinks and Beverages
4	Moisturizer	8	180.0	Cosmetics
5	Face Wash	10	80.0	Cosmetics
6	Milk	15	10.0	Dairy Products
7	Orange	100	5.65	Groceries

2.2.3 Seller Module

- Add Products to Cart: The seller will be able to add products to a virtual cart for the customer, by manually entering the product information.

- View and Update Cart: The seller will be able to view the contents of the cart and make any necessary updates, such as removing items or changing the quantity.

- Calculate Total Cost: The system will automatically calculate the total cost of the items in the cart.

- Print Receipt: The seller will be able to print a receipt for the customer's purchase, which will include a list of the items purchased and the total cost.

- Database Maintenance: The database is maintained and updated with the new available quantities of the products after the selling process.

Screenshot 1

Billing

LOGOUT

BILLING POINT

PROD NAME   Moisturizer

QUANTITY   3

Add to Bill   **Clear**

Groceriess ▼   **Filter**   **Refresh**

PROD ID	PROD NAME	PROD QTY	PROD PRI...	PROD CAT
3	Milkshake	50	45.55	Drinks an...
4	Moisturizer	5	180.0	Cosmetics
5	Face Wash	10	80.0	Cosmetics
6	Milk	15	10.0	Dairy Pro...

-------------------- BILL DETAILS --------------------

```
***** IDEAL POINT *****
NUM        PRODUCT      PRICE      QUANTITY      TOTAL
1          Moisturizer  180.0      3             540.0

                                                 540.0 LE
```

PRODID	PRODUCT	PRICE	QUANTITY	TOTAL
4	Moisturizer	180.0	3	540.0

Print

Remove from Bill

Screenshot 2

Billing

LOGOUT

BILLING POINT

PROD NAME   Orange

QUANTITY   20

Add to Bill   **Clear**

Groceriess ▼   **Filter**   **Refresh**

PROD ID	PROD NAME	PROD QTY	PROD PRI...	PROD CAT
4	Moisturizer	5	180.0	Cosmetics
5	Face Wash	10	80.0	Cosmetics
6	Milk	15	10.0	Dairy Pro...
7	Orange	80	5.65	Groceries

-------------------- BILL DETAILS --------------------

```
***** IDEAL POINT *****
NUM        PRODUCT      PRICE      QUANTITY      TOTAL
1          Moisturizer  180.0      3             540.0
2          Orange       5.65       20            113.0

                                                 653.0 LE
```

PRODID	PRODUCT	PRICE	QUANTITY	TOTAL
4	Moisturizer	180.0	3	540.0
7	Orange	5.65	20	113.0

Print

Remove from Bill

Screenshot 3

Billing

LOGOUT

BILLING POINT

PROD NAME   Orange

QUANTITY   20

Add to Bill   **Clear**

Drinks and Beverages ▼   **Filter**   **Refresh**

PROD ID	PROD NAME	PROD QTY	PROD PRI...	PROD CAT
3	Milkshake	50	45.55	Drinks an...
4	Moisturizer	8	180.0	Cosmetics
5	Face Wash	10	80.0	Cosmetics
6	Milk	15	10.0	Dairy Pro...

-------------------- BILL DETAILS --------------------

```
***** IDEAL POINT *****
NUM        PRODUCT      PRICE      QUANTITY      TOTAL
1          Orange       5.65       20            113.0

                                                 113.0 LE
```

PRODID	PRODUCT	PRICE	QUANTITY	TOTAL
7	Orange	5.65	20	113.0

Print

Remove from Bill

3. UML Class Diagram

Login
-clearBtn : JButton
-jLabel1 : JLabel
-jLabel2 : JLabel
-jLabel3 : JLabel
-jLabel4 : JLabel
-jPanel1 : JPanel
-jPanel2 : JPanel
-loginBtn : JButton
-password : JPasswordField
-role : JComboBox<String>
-username : JTextField

-clearBtnActionPerformed(ActionEvent evt)
-initComponents()
-loginBtnActionPerformed(ActionEvent evt)
-loginBtnKeyPressed(KeyEvent evt)
-passwordActionPerformed(ActionEvent evt)
-passwordKeyPressed(KeyEvent evt)
-usernameActionPerformed(ActionEvent evt)
+Login()
+main(String[])

AdminMenu
-jButton1 : JButton
-jButton2 : JButton
-jButton3 : JButton
-jButton4 : JButton
-jLabel1 : JLabel
-jPanel1 : JPanel

+AdminMenu()
-initComponents()
-jButton1ActionPerformed(ActionEvent evt)
-jButton2ActionPerformed(ActionEvent evt)
-jButton3ActionPerformed(ActionEvent evt)
-jButton4ActionPerformed(ActionEvent evt)
+main(String[])

Seller
-add : JButton
-clear : JButton
-delete : JButton
-edit : JButton
-gender : JComboBox<String>
-id : JTextField
-jLabel1 : JLabel
-jLabel2 : JLabel
-jLabel3 : JLabel
-jLabel4 : JLabel
-jLabel5 : JLabel
-jPanel1 : JPanel
-jPanel2 : JPanel
-jScrollPane1 : JScrollPane
-jSeparator1 : JSeparator
-jTable1 : JTable
-name : JTextField
-pass : JTextField

+Seller()
-addActionPerformed(ActionEvent evt)
+addSeller()
-clearActionPerformed(ActionEvent evt)
-deleteActionPerformed(ActionEvent evt)
+deleteSeller()
-editActionPerformed(ActionEvent evt)
+editSeller()
+fetchSellers()
-initComponents()
-jTable1MouseClicked(MouseEvent evt)
-jTable1MousePressed(MouseEvent evt)
+main(String[])

Splash
-Myprogress : JProgressBar
-Percentage : JLabel
-jLabel2 : JLabel
-jLabel3 : JLabel
-jPanel1 : JPanel

+Splash()
+main(String[])
-initComponents()

JFrame

Categories
-jButton1 : JButton
-jButton2 : JButton
-jButton3 : JButton
-jButton4 : JButton
-jLabel1 : JLabel
-jLabel2 : JLabel
-jLabel3 : JLabel
-jLabel7 : JLabel
-jMenu1 : JMenu
-jMenu3 : JMenu
-jMenu4 : JMenu
-jMenuBar2 : JMenuBar
-jMenuItem1 : JMenuItem
-jMenuItem2 : JMenuItem
-jPanel1 : JPanel
-jScrollPane1 : JScrollPane
-jSeparator1 : JSeparator
-jTable1 : JTable
-jTextField1 : JTextField
-jTextField2 : JTextField

+Categories()
+addData()
+deleteData()
+editData()
+fetchData()
-initComponents()
-jButton1ActionPerformed(ActionEvent evt)
-jButton2ActionPerformed(ActionEvent evt)
-jButton3ActionPerformed(ActionEvent evt)
-jButton4ActionPerformed(ActionEvent evt)
-jMenu1MouseClicked(MouseEvent evt)
-jMenu3MouseClicked(MouseEvent evt)
-jMenu4MouseClicked(MouseEvent evt)
-jTable1MouseClicked(MouseEvent evt)
-jTextField1ActionPerformed(ActionEvent evt)
-jTextField2ActionPerformed(ActionEvent evt)
+main(String[])
+selectData()

Products
-jButton1 : JButton
-jButton2 : JButton
-jButton3 : JButton
-jButton4 : JButton
-jComboBox1 : JComboBox<String>
-jLabel1 : JLabel
-jLabel2 : JLabel
-jLabel3 : JLabel
-jLabel4 : JLabel
-jLabel5 : JLabel
-jLabel6 : JLabel
-jLabel7 : JLabel
-jMenu1 : JMenu
-jMenu2 : JMenu
-jMenu3 : JMenu
-jMenuBar1 : JMenuBar
-jPanel1 : JPanel
-jScrollPane1 : JScrollPane
-jSeparator1 : JSeparator
-jTable1 : JTable
-jTextField1 : JTextField
-jTextField2 : JTextField
-jTextField3 : JTextField
-jTextField4 : JTextField

+Products()
+addData()
+deleteData()
+editData()
+fetchCat()
+fetchData()
-initComponents()
-jButton1ActionPerformed(ActionEvent evt)
-jButton2ActionPerformed(ActionEvent evt)
-jButton3ActionPerformed(ActionEvent evt)
-jButton4ActionPerformed(ActionEvent evt)
-jComboBox1ActionPerformed(ActionEvent evt)
-jMenu1MouseClicked(MouseEvent evt)
-jMenu2MouseClicked(MouseEvent evt)
-jMenu3MouseClicked(MouseEvent evt)
-jTable1MouseClicked(MouseEvent evt)
-jTextField1ActionPerformed(ActionEvent evt)
-jTextField2ActionPerformed(ActionEvent evt)
-jTextField3ActionPerformed(ActionEvent evt)
-jTextField4ActionPerformed(ActionEvent evt)
+main(String[])
+selectData()

Billing
+Total : double
+availQty : int
+bill_prod_id : int
+bill_prod_qty : int
+bill_row_num : int
+cartQty : int
+i : int
+itemPrice : double
-addBtn : JButton
-bill_area : JTextArea
-bill_details : JTable
-catCb : JComboBox<String>
-clearBtn : JButton
-filterBtn : JButton
-jLabel1 : JLabel
-jLabel3 : JLabel
-jLabel4 : JLabel
-jLabel5 : JLabel
-jLabel6 : JLabel
-jPanel1 : JPanel
-jPanel2 : JPanel
-jScrollPane1 : JScrollPane
-jScrollPane2 : JScrollPane
-jScrollPane4 : JScrollPane
+newAvailQty : int
+newQty : int
+prodId : int
+totalItemPrice : double
-printBtn : JButton
-prodTable : JTable
-prod_name : JTextField
-prod_qty : JTextField
-refreshBtn : JButton
-removeBtn : JButton

+Billing()
-addBtnActionPerformed(ActionEvent evt)
-bill_detailsMouseClicked(MouseEvent evt)
-clearBtnActionPerformed(ActionEvent evt)
-filterBtnActionPerformed(ActionEvent evt)
-initComponents()
-printBtnActionPerformed(ActionEvent evt)
-refreshBtnActionPerformed(ActionEvent evt)
-removeBtnActionPerformed(ActionEvent evt)
-prodTableMouseClicked(MouseEvent evt)
+all_products()
+bill_area_build()
+get_categories()
+main(String[])
+update_database()
+update_database2()