**D. Y. Patil College of College of Engineering and Technology, Kolhapur**

**Department of Computer Science & Engineering**

**Class: SY-A Subject: AOOC**

**Experiment no: 15**

**Group No. G27 Mini Project**

**Title of Mini-Project: Grocery-Management System**

**Problem Statement:**

Managing household groceries efficiently can be difficult without an organized system. Many individuals find it hard to keep track of what items are in stock, what needs to be purchased, and how much is being spent on groceries each month. This project aims to provide a simple desktop-based Grocery Management System that allows users to log grocery items, track inventory, and monitor spending. By storing data locally, it ensures easy access to grocery history, helping users make informed purchasing decisions and reduce waste.

**Introduction:**

Grocery Management System is a Java-based desktop application designed to streamline the process of managing household grocery items. In day-to-day life, keeping track of available items, quantities, prices, and overall grocery spending can become overwhelming. This application offers a structured and efficient solution, enabling users to organize their grocery inventory and monitor expenses with ease.

Developed using Java Swing for the graphical user interface, the application provides an intuitive and user-friendly experience. Users can add grocery items by specifying the name, quantity, and price per unit. The system automatically calculates the total cost per item and updates the overall bill in real-time. Features such as editing, removing items, clearing the list, and searching by item name make the application practical for everyday use.

The data is presented in a tabular format using JTable, allowing users to view, sort, and manage items efficiently. While the current implementation stores data in-memory (session-based), it is designed with modularity in mind, making it easy to extend with persistent storage in future versions.

One of the key highlights of this project is its real-time total bill calculation, which updates dynamically as items are added, modified, or removed. This helps users stay within budget and track how much is being spent on groceries at any point. The search feature further enhances usability by allowing users to quickly locate specific items in their inventory.

This project demonstrates several important Java programming concepts such as object-oriented design, event-driven programming, and GUI development using Swing components like JFrame, JPanel, JLabel, JButton, and JTable. It serves as a practical example of how desktop applications can simplify daily household management and promote better organization.

**System Architecture:**

**Presentation Layer (User Interface)**

* **Built using: Java Swing.**
* **Components: JFrame, JPanel, JTable, JLabel, JTextField, JButton, JScrollPane, JOptionPane.**
* **Responsibilities:**
  + **Handles all user interactions such as adding, editing, removing, clearing, and searching grocery items.**
  + **Displays the list of items in a table format.**
  + **Shows the dynamically calculated total bill to the user.**

**2. Business Logic Layer**

* **Contains logic for:**
  + **Adding new grocery items.**
  + **Editing selected items.**
  + **Removing selected items.**
  + **Calculating the total bill in real-time.**
  + **Searching for items by name.**
* **Manages the list of items through a DefaultTableModel, representing the in-memory data.**

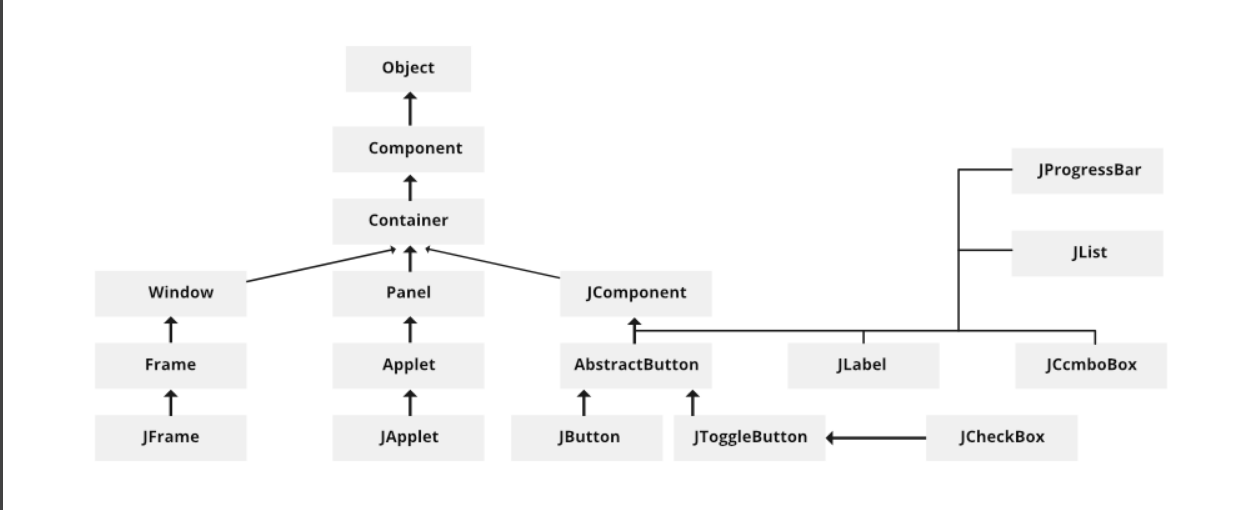
**3. Data Layer (Persistence)**

* **Current Status:**
  + **The application is session-based and does not persist data beyond runtime.**
* **Potential Enhancements:**
  + **Save and load grocery item data using local storage (e.g., text file, JSON, or database).**
  + **Implement file I/O using BufferedReader and BufferedWriter for persistence.**

**4. Model Layer**

* **(Optional/Implied) GroceryItem Class:**
  + **Fields: itemName, quantity, pricePerUnit, total.**
  + **Methods: Could include methods for computing total and converting objects to/from strings for file storage.**

**System Architecture Diagram:**

****

**Module description or working of system:**

1. **ItemEntryAndEditingSystem**

This module allows users to add new grocery items by entering item name, quantity, and price per unit. Users can also edit existing entries by selecting a row in the table and updating the fields.

**2. RealTimeBillCalculation**

Each time an item is added, edited, or removed, the system automatically calculates the total cost (quantity × price) and updates the overall bill in real-time. This provides users with an up-to-date view of their total expenses.

**3. GroceryTableDisplaySystem**

All grocery items are displayed in a JTable with columns for item name, quantity, price, and total cost. This tabular format makes it easy to view, track, and manage grocery details.

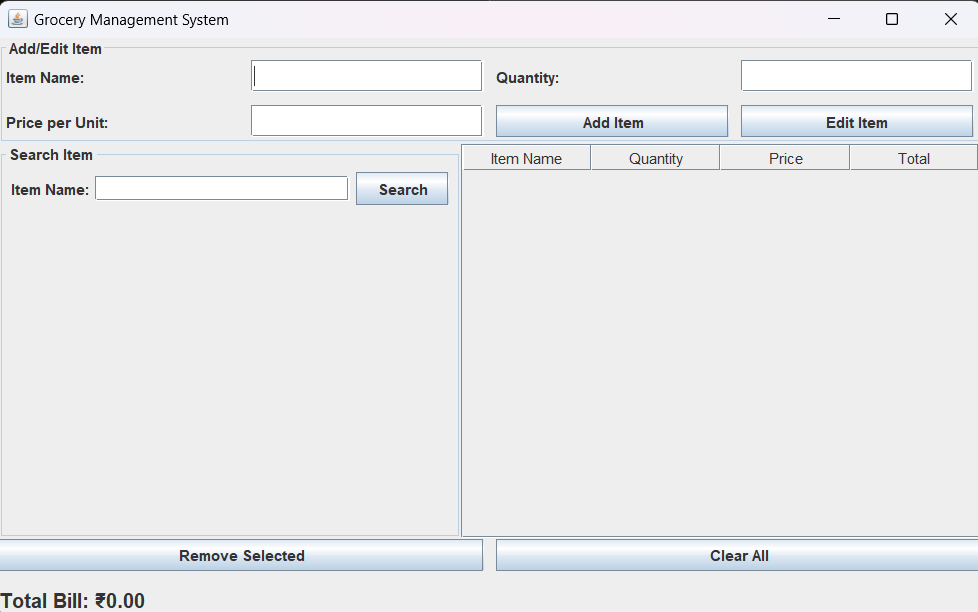
**4. SearchAndClearSystem**

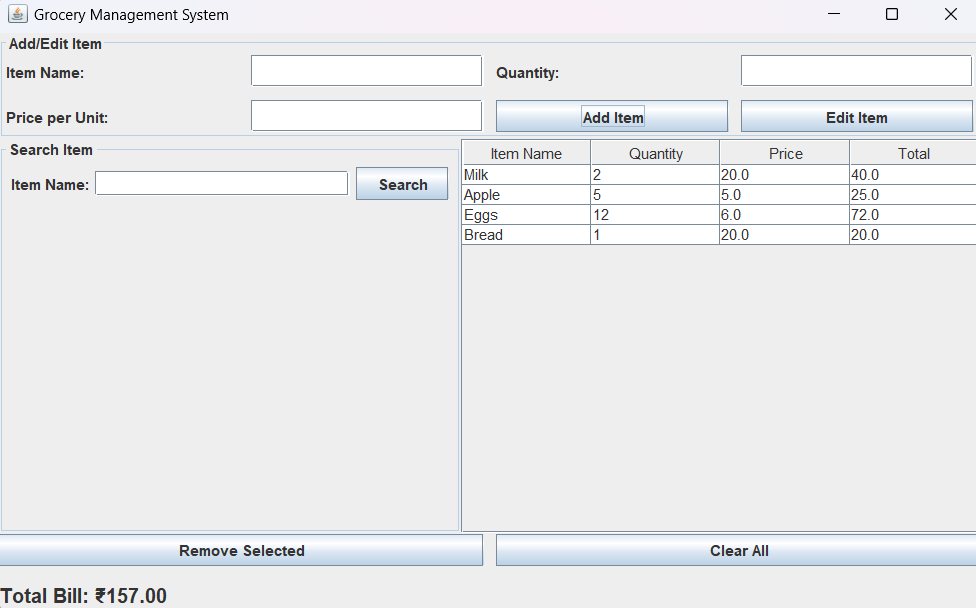
Users can search for specific items by name using a search field. Additionally, the system allows users to remove selected items or clear the entire list with dedicated buttons, offering better control over the inventory.

**5. In-MemoryDataManagement**

The application stores all data in memory during runtime using a DefaultTableModel. Although there is no file persistence in the current version, the modular design allows for easy integration of file or database storage in future enhancements.

**Screenshots:**

****

****

**Group Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unique id** | **Roll No** | **Name of Student** | **Sign** |
| **EN23187816** | **102** | **Revatee Ranjit Ghatage** |  |
| **EN23164031** | **103** | **Sanaa Khushahmad Shaikh** |  |
| **EN23208345** | **119** | **Anisha Anil Shiudkar** |  |
| **EN23303770** | **125** | **Sakshi Kiran Sambhaji** |  |
|  |  |  |  |