

**Object-Oriented Programming in Java**

**Mini Project Report**

Grocery Shopping List Management System

By

N Sanjana(1RVU23CSE292)

Prajna(1RVU23CSE339)

Under the guidance of:

Dr./Prof. Jobin Thomas

School of Computer Science and Engineering

RV University, Bangalore



**School of Computer Science and Engineering**

CERTIFICATE

Certified that the CS2024 Mini Project work titled Grocery Shopping List is carried out by N Sanjana(1RVU23CSE292) and Prajna(1RVU23CSE339) who are bonafide students of the School of Computer Science and Engineering, RV University, Bengaluru, during the year 2024–25. It is certified that all corrections/ suggestions from all the continuous internal evaluations have been incorporated into the project and in this report.

Dr./ Prof. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Dr. Sudhakar K. N

Faculty Guide Program Director

## **1 .Problem statement**

In today's busy lifestyle, managing grocery shopping efficiently is crucial for consumers who want to ensure they have the necessary items for their households while avoiding unnecessary purchases. Traditional methods of keeping track of grocery needs often involve manual lists or memory, which can lead to forgotten items or overspending. This project aims to design and implement a **Grocery Shopping List Management System** that simplifies the process of managing grocery shopping.

The objective of this project is to develop a system that allows users to effectively manage their grocery shopping lists. The system will enable users to:

1. Add Grocery Items: Users can input the names of grocery items they need, ensuring they have a comprehensive list before shopping.

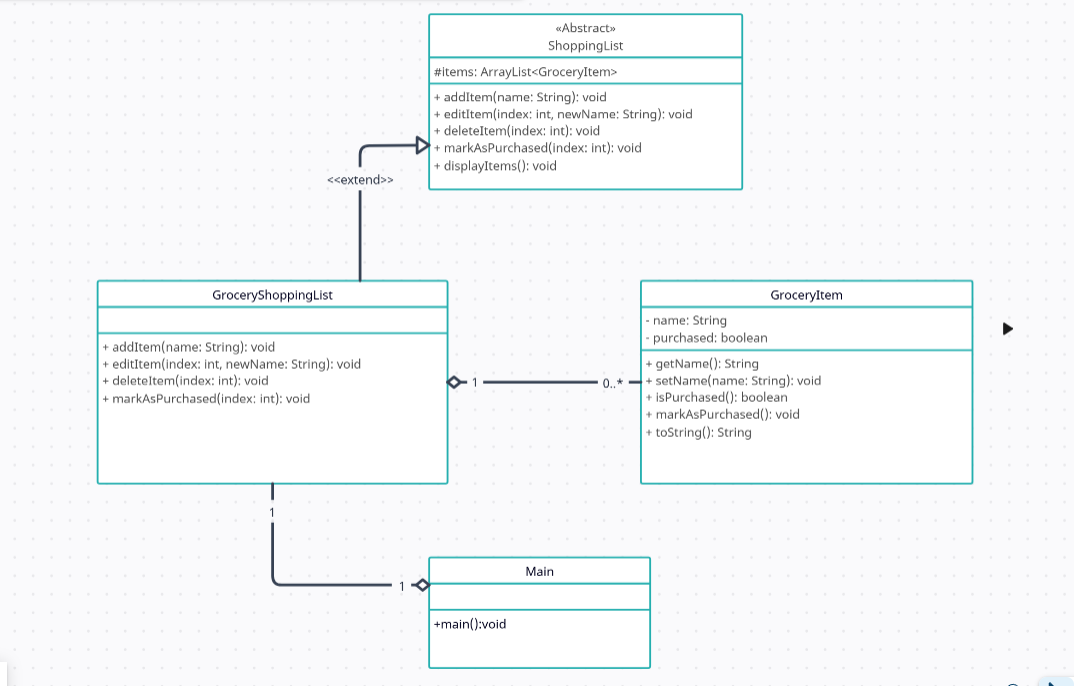
2. Edit Existing Items: If users change their minds about a particular item or need to update it, they can easily edit the item name.

3. Delete Unwanted Items: Users can remove items from their list that they no longer need, keeping their shopping list relevant and up-to-date.

4. Mark Items as Purchased: Once items are bought, users can mark them as purchased, helping them track what they have already acquired.

5. Display the Current Shopping List: Users can view all items in their shopping list, clearly indicating which items have been purchased and which are still needed.

## **2. Class Diagram**



3 .Source Codeimport java.util.ArrayList;  
import java.util.Scanner;  
  
  
abstract class ShoppingList {  
 protected ArrayList<GroceryItem> items = new ArrayList<>();  
  
 public abstract void addItem(String name);  
 public abstract void editItem(int index, String newName);  
 public abstract void deleteItem(int index);  
 public abstract void markAsPurchased(int index);  
  
 public void displayItems() {  
 if (items.isEmpty()) {  
 System.out.println("The shopping list is empty.");  
 } else {  
 System.out.println("Shopping List:");  
 for (int i = 0; i < items.size(); i++) {  
 System.out.println((i + 1) + ". " + items.get(i));  
 }  
 }  
 }  
}  
  
class GroceryItem {  
 private String name;  
 private boolean purchased;  
  
 public GroceryItem(String name) {  
 this.name = name;  
 this.purchased = false;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public boolean isPurchased() {  
 return purchased;  
 }  
  
 public void markAsPurchased() {  
 this.purchased = true;  
 }  
  
 public String toString() {  
 return name + (purchased ? " (Purchased)" : "");  
 }  
}  
  
  
class GroceryShoppingList extends ShoppingList {  
  
 public void addItem(String name) {  
 items.add(new GroceryItem(name));  
 System.out.println("Item added: " + name);  
 }  
  
 public void editItem(int index, String newName) {  
 if (isValidIndex(index)) {  
 items.get(index - 1).setName(newName);  
 System.out.println("Item updated to: " + newName);  
 } else {  
 System.out.println("Invalid item number.");  
 }  
 }  
  
 public void deleteItem(int index) {  
 if (isValidIndex(index)) {  
 System.out.println("Removing item: " + items.get(index - 1).getName());  
 items.remove(index - 1);  
 } else {  
 System.out.println("Invalid item number.");  
 }  
 }  
  
 // Mark item as purchased, demonstrating polymorphism  
 public void markAsPurchased(int index) {  
 if (isValidIndex(index)) {  
 items.get(index - 1).markAsPurchased();  
 System.out.println("Item marked as purchased.");  
 } else {  
 System.out.println("Invalid item number.");  
 }  
 }  
  
 private boolean isValidIndex(int index) {  
 return index > 0 && index <= items.size();  
 }  
}  
  
public class Main {  
 public static void main(String[] args) {  
 GroceryShoppingList shoppingList = new GroceryShoppingList();  
 Scanner scanner = new Scanner(System.in);  
 boolean running = true;  
  
 while (running) {  
 System.out.println("\nGrocery Shopping List:");  
 System.out.println("1. Add Item");  
 System.out.println("2. Edit Item");  
 System.out.println("3. Delete Item");  
 System.out.println("4. Mark Item as Purchased");  
 System.out.println("5. View Shopping List");  
 System.out.println("6. Exit");  
  
 System.out.print("Choose an option: ");  
 int choice = scanner.nextInt();  
 scanner.nextLine();  
  
 switch (choice) {  
 case 1:  
 System.out.print("Enter item name: ");  
 shoppingList.addItem(scanner.nextLine());  
 break;  
 case 2:  
 shoppingList.displayItems();  
 System.out.print("Enter item number to edit: ");  
 int editIndex = scanner.nextInt();  
 scanner.nextLine();  
 System.out.print("Enter new name: ");  
 shoppingList.editItem(editIndex, scanner.nextLine());  
 break;  
 case 3:  
 shoppingList.displayItems();  
 System.out.print("Enter item number to delete: ");  
 shoppingList.deleteItem(scanner.nextInt());  
 break;  
 case 4:  
 shoppingList.displayItems();  
 System.out.print("Enter item number to mark as purchased: ");  
 shoppingList.markAsPurchased(scanner.nextInt());  
 break;  
 case 5:  
 shoppingList.displayItems();  
 break;  
 case 6:  
 running = false;  
 System.out.println("Exiting...");  
 break;  
 default:  
 System.out.println("Invalid choice. Please try again.");  
 }  
 }  
 scanner.close();  
 }  
}

## **4. Output**