

SUPER MARKET BILLING SYSTEM

A Project by :

Sathish .B (2212055)
Mani Kiran .B (2212056)

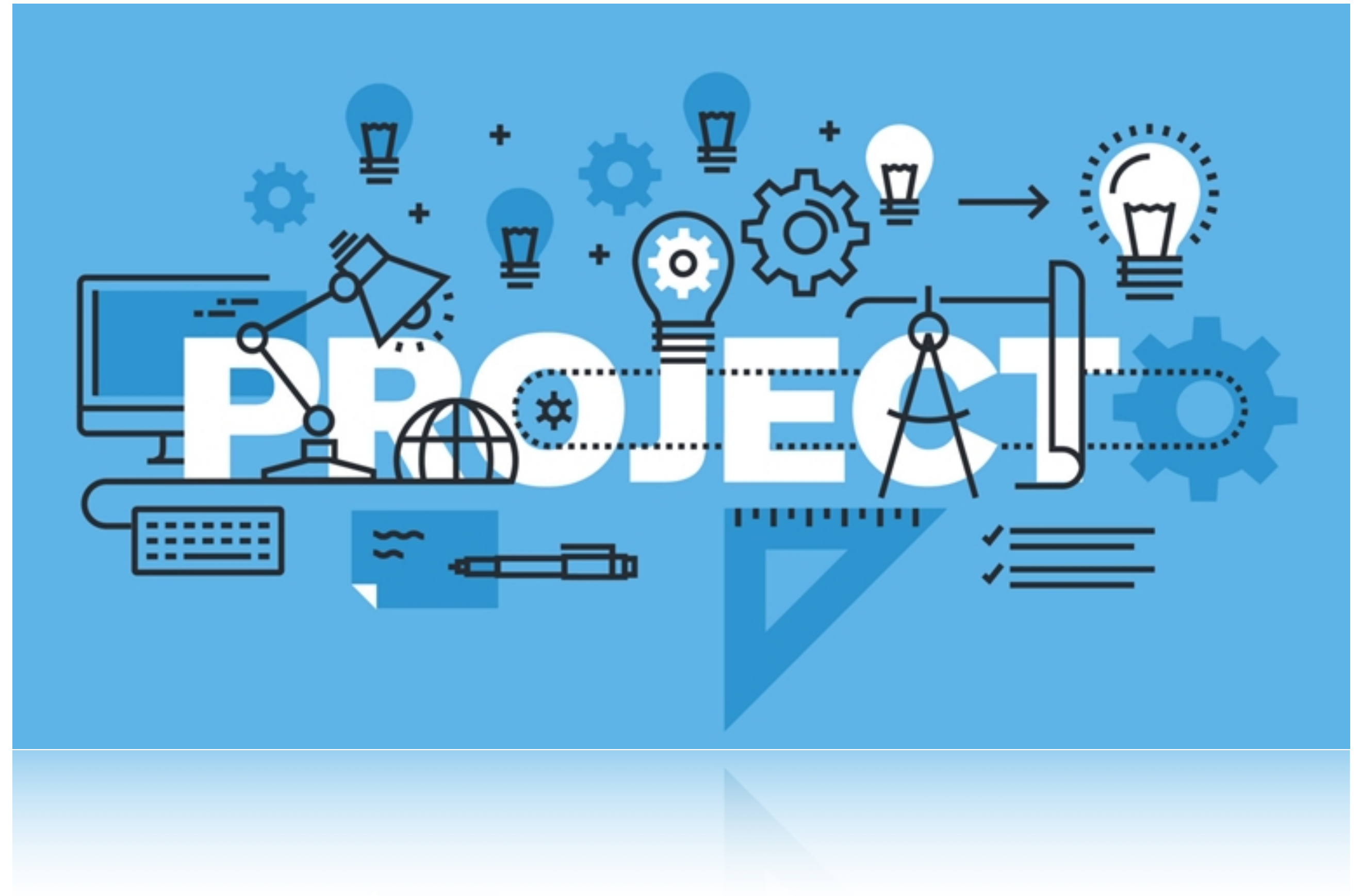
Department of Computer Science and Engineering
National Institute of Technology, Silchar

23rd April' 2024



PROJECT OUTLINE:

- Introduction.
- Problem Statement.
- Features.
- How OOPS helped?
- Flow chart.
- Functionality.
- Final consideration



INTRODUCTION



Objective :

To design and implement a supermarket billing system using C++ programming language.

Project Overview :

The project aims to create a user-friendly interface for managing stock, adding items to the cart, and generating bills for customers.

It provides functionalities such as adding items to the stock, viewing stock details, adding items to the cart, and generating bills with discounts and taxes.

Developed as a part of the CS 212 Object-Oriented Programming (OOP) course.

PROBLEM STATEMENT

Background :

In today's fast-paced world, supermarkets play a crucial role in providing a convenient shopping experience for customers.

However, managing inventory, processing transactions, and generating accurate bills can be challenging tasks for supermarket staff. To streamline these operations and enhance customer satisfaction, there is a need for an efficient supermarket billing system.

Problem Description:

You are tasked with designing and implementing a supermarket billing system using C++ programming language. The system should provide functionalities to manage stock, process customer transactions, and generate bills with itemised details.



Stock Management :

Add items to the stock with details such as name, price, and quantity.

View the current stock of items available.

Billing System :

Add items to the cart with the desired quantity, Calculate the total bill amount, including discounts and taxes. Apply festival discounts and additional charges like a carry bag.

User-Friendly Interface :

Clear and concise menu options for easy navigation, Visual feedback during item addition and bill generation for enhanced user experience.

Real-Time Updates :

Instantaneous updates to the stock upon adding or removing items.

Dynamic calculation of the bill based on current stock and user inputs.

FEATURES

Error Handling :

Informative error messages for out-of-stock items or invalid user inputs.

Robust file handling to prevent data loss or corruption.

Customisation Options :

Ability to customise bills with cashier ID, bill number, and timestamp for record-keeping. Options to include or exclude additional charges like a carry bag based on user preference.

Scalability :

Designed to accommodate future expansions or modifications in functionality. Flexible code structure allowing easy integration of new features or improvements.

Enhanced User Experience:

Incorporation of visual elements and sleep delays for a more interactive and engaging user experience. efficient stock management and billing operations.

IMPORTANT ROLE OF OOPS

Classes

Each class represents a distinct entity with its own set of properties and behaviours. Objects of these classes are created and manipulated throughout the program.



The project defines several classes such as ItemDetails, Bill, and BillingSystem. Ex:- instances of the Bill class represent individual items added to the cart.

Encapsulation

implemented by using access specifiers (public, private, protected) to control the access to class members.



Data members of classes (Item, Price, Quantity) are encapsulated. Getter and setter methods are used to encapsulate access to class properties and ensure data integrity.

Abstraction

Abstraction is achieved by hiding the internal implementation details and exposing only the essential features of an object.



Users interact with the system through high-level functions provided by the BillingSystem class, without needing to know the internal workings of the system.

Inheritance

Inheritance is utilised to create a specialised class (Bill) from a base class (ItemDetails) to inherit common attributes and behaviours.



The Bill class inherits from the ItemDetails class, demonstrating inheritance. It allows Bill to inherit the properties (Item and Price) from ItemDetails, promoting code.

IMPORTANT ROLE OF OOPS

Friend Function

A friend function in C++ is a function that is not a member of a class but has access to the class's private and protected members.

The BillingSystem class declares print_receipt() and show_cart() as friend functions of the Bill class. And these can access the private members of bill

Constructor Overloading

the concept of having more than one constructor with different parameters so that every constructor can perform a different task.

Bill class demonstrates two constructor overloading :- All parameters(Item, Price, Quantity), another one is only items and price.

Destructor

a member function that is invoked automatically when the object goes out of scope or is explicitly destroyed by a call to delete or delete[]

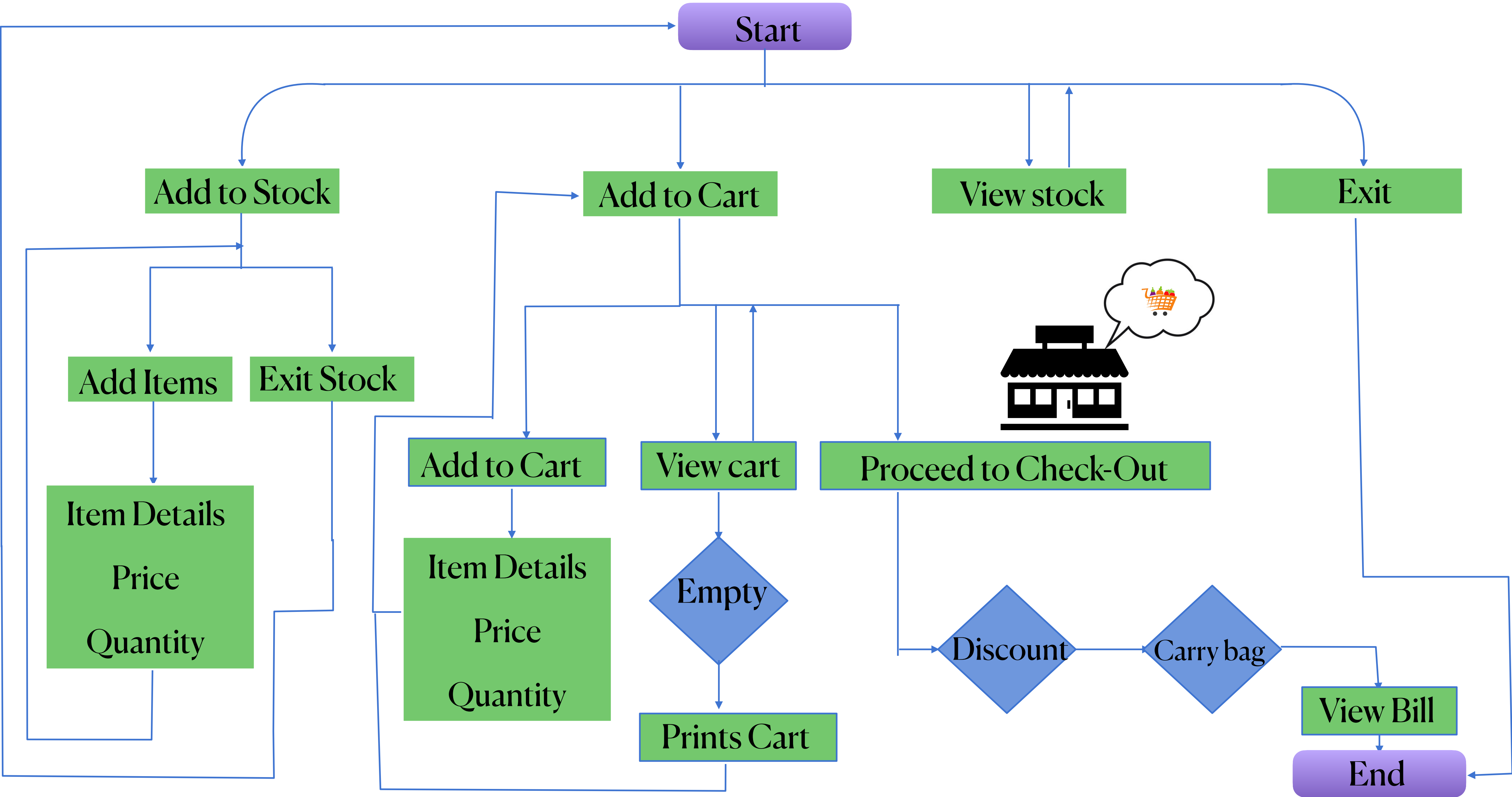
the Bill class has a destructor, although it doesn't perform any cleanup.

File Handling (I/O)

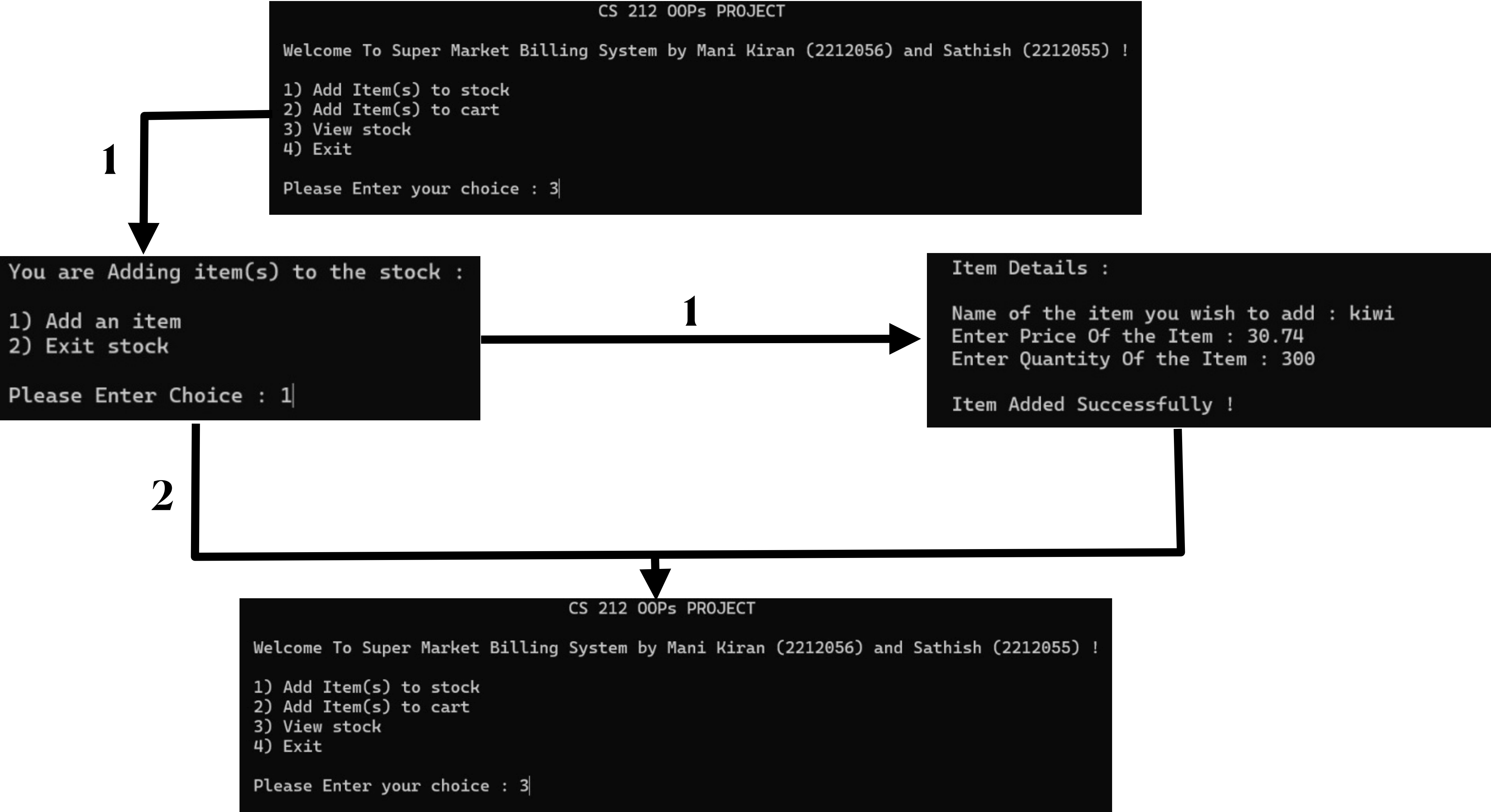
File handling is utilised to read and write data from/to external files, such as maintaining the inventory (shop.txt) of the super market.

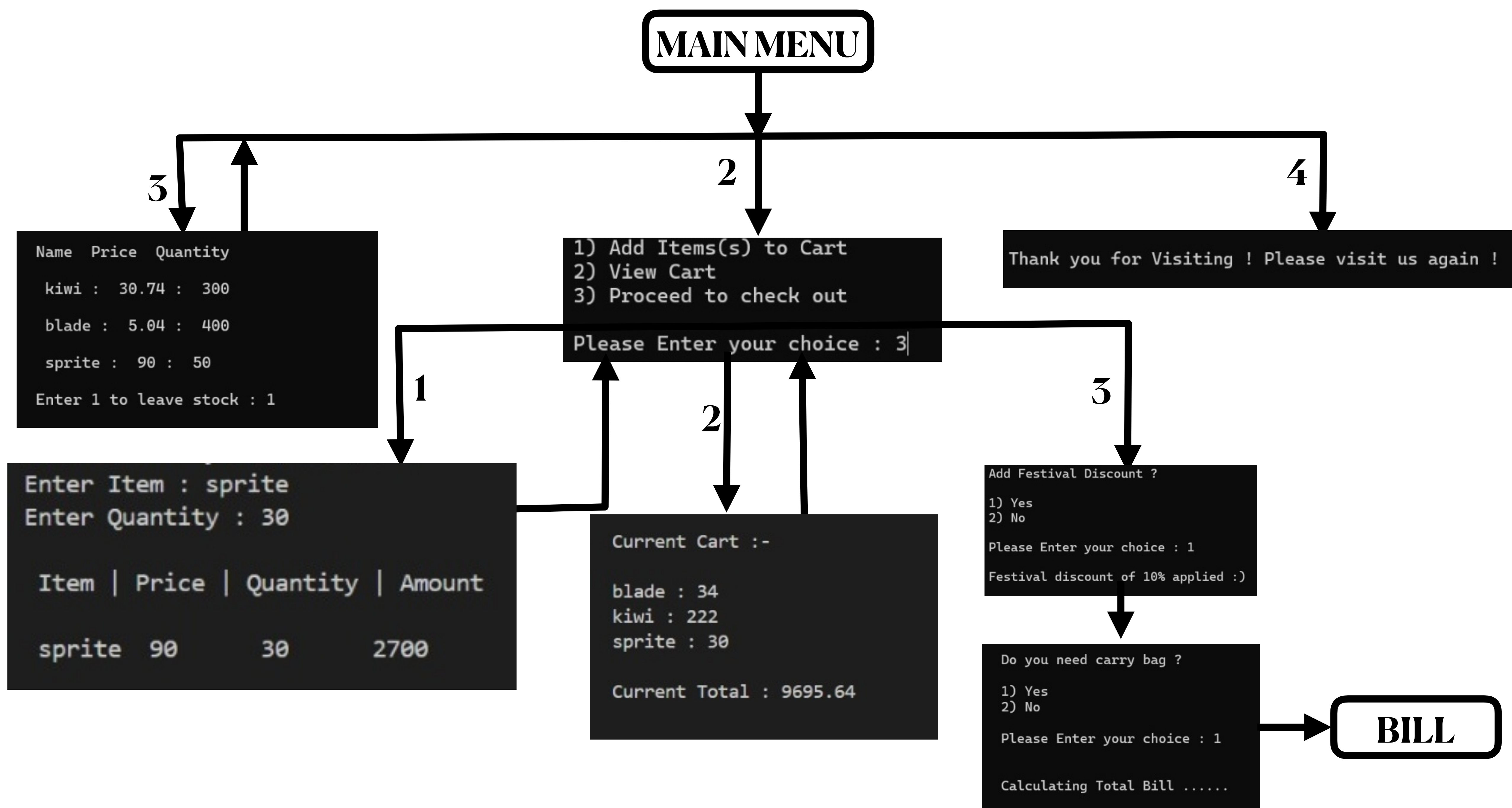
The ViewStock() function reads from the file to display available items in the stock, while the addItem() function writes new items to the file.

FLOW CHART



FUNCTIONALITY OF PROJECT





TOTAL BILL OF THE SELECTED ITEMS

SMK SHOPPING MALL
GSTIN : 38AANCA1901C1Z6

Cashier ID : 1166

Date : 22/4/2024

Bill No : 6

Time : 20:6:40

blade	:	171.36/-
kiwi	:	6824.28/-
sprite	:	2700/-

SubTotal	:	9695.64/-
Discounts	:	-969.564/-
CGST 2.5%	:	218.152/-
SGST 2.5%	:	218.152/-
Carry Bag	:	0/-

Grand Total : Rs 9162.38/-

Thank you For Shopping with us ! Please visit us again !

FINAL CONSIDERATION

Our Team Members



Thank You!

Contact us



2212055, 2212056