STORE BILLING SYSTEM

Project Report

Name	Lalan Kumar Gupta
Registration No	25BAI11005
Programme	B.Tech CSE (AIML)
Course	CSE1021 - Problem Solving and Programming
Faculty	Sasmita Padhy
Institution	Vellore Institute of Technology

Introduction

The Store Billing System is a Python-based console application that helps a small shopkeeper manage billing operations. The user can add items with their prices, remove items if entered incorrectly, view the complete bill and check the total amount to be paid by the customer.

1. Problem Statement

In many small stores, billing is still performed manually using calculators or handwritten notes. This is time-consuming and error-prone, especially when items need to be removed or edited. The goal of this project is to create a simple menu-driven program that automates basic billing operations and calculates the total amount accurately.

2. Functional Requirements

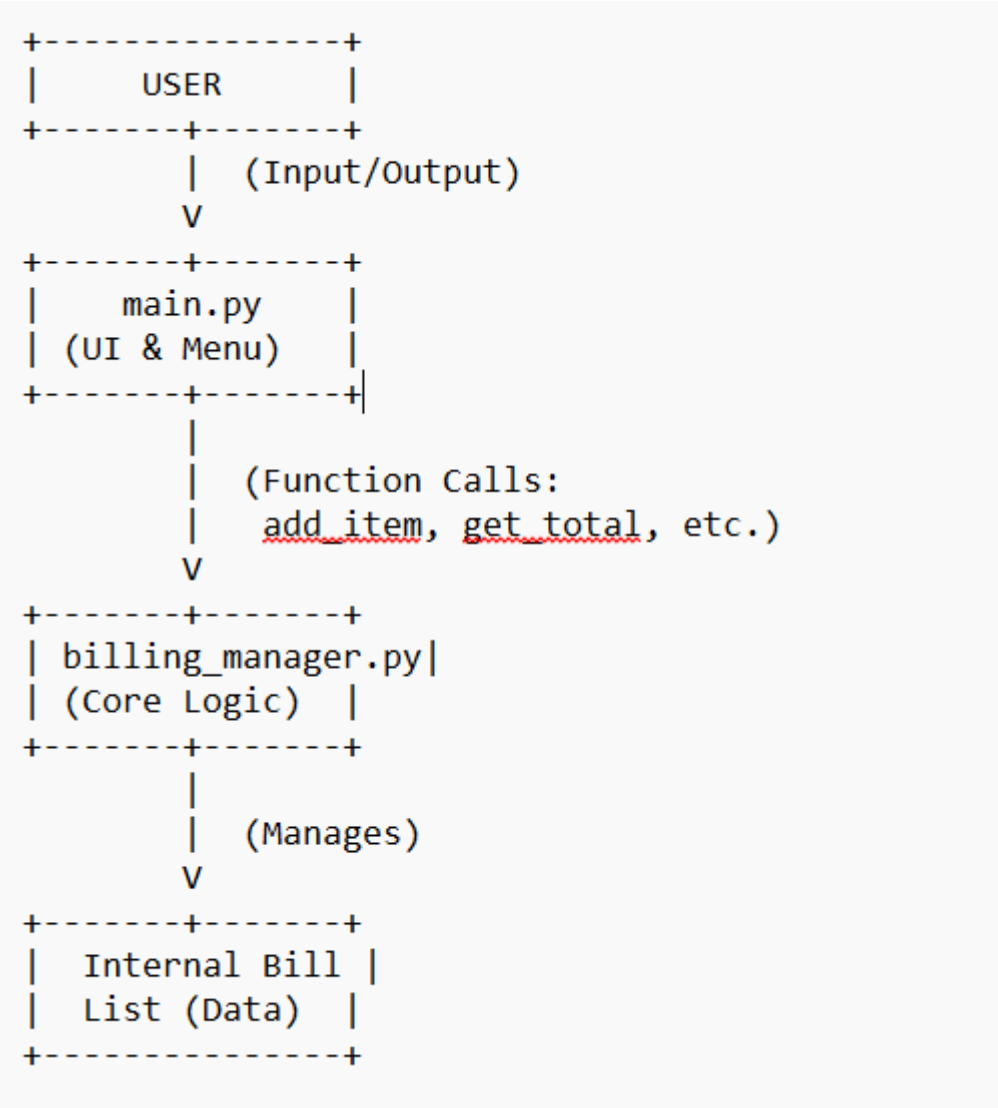
- Add Item – The user can enter an item name and its price to add it to the current bill.
- Remove Item – The user can remove a specific item by entering its name.
- View Bill – Display all items currently added along with their individual prices.
- View Total Amount – Calculate and display the total amount of the bill.
- End – Exit the program gracefully.

3. Non-functional Requirements

- Usability – The options in the menu should be easy to understand for any user.
- Accuracy – Item prices and total amount must be calculated correctly.
- Maintainability – Core billing logic is kept in a separate module (billing_manager) to keep code organised.

- Error Handling – The program validates price input and handles invalid menu choices with clear messages.

4. System Architecture



5. Implementation Details

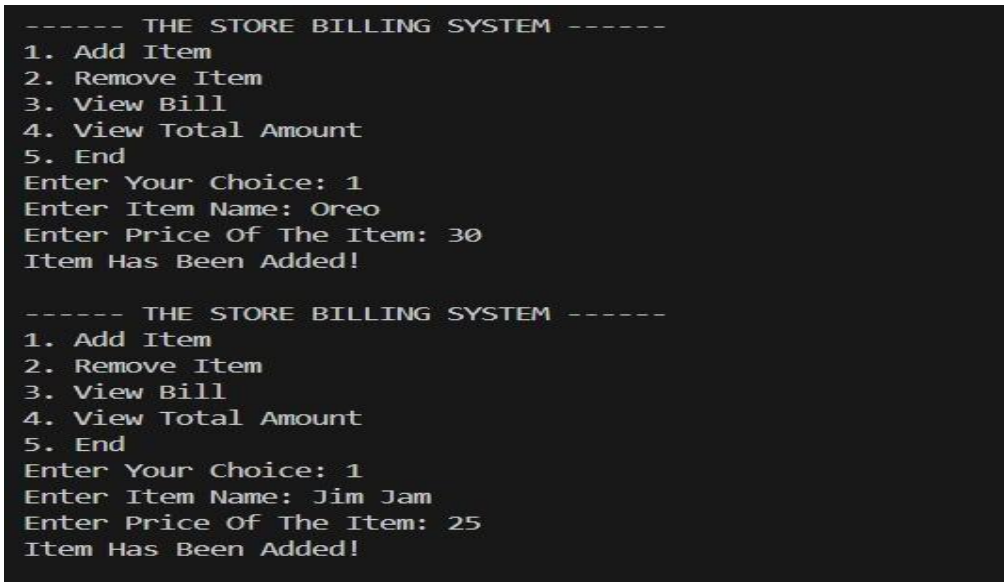
The project is implemented in Python with a focus on clean, modular code. The main.py file defines a menu() function that prints the available options. Inside an infinite while loop, the program reads the user's choice and performs the corresponding action.

When the user chooses to add an item, the program accepts the item name and price. The price is validated using string checks (allowing digits and a single decimal point). If the input is valid, it is converted to a float and passed to the add_item function in billing_manager. If invalid, an error message is displayed.

For removing an item, the program asks for the item name. The remove_item function returns True if the item was found and removed, otherwise False, which is used to print an appropriate message. The view_bill function returns the list of items, which is then printed line by line in a friendly format. The get_total function computes the sum of all item prices and returns the final amount.

6. Screenshots / Results

The screenshots below show a sample execution of the program where items are added, one item is removed, the bill is displayed and the total amount is calculated before exiting the application.



```
------ THE STORE BILLING SYSTEM ------
1. Add Item
2. Remove Item
3. View Bill
4. View Total Amount
5. End
Enter Your Choice: 2
Enter the name of the item you want to remove: Munch
Item has been successfully removed!

------ THE STORE BILLING SYSTEM ------
1. Add Item
2. Remove Item
3. View Bill
4. View Total Amount
5. End
Enter Your Choice: 3

--- Your Bill is ---
Oreo : ₹30.0
Jim Jam : ₹25.0
Kurkure : ₹10.0
```

```
------ THE STORE BILLING SYSTEM ------
1. Add Item
2. Remove Item
3. View Bill
4. View Total Amount
5. End
Enter Your Choice: 1
Enter Item Name: Kurkure
Enter Price Of The Item: 10
Item Has Been Added!

------ THE STORE BILLING SYSTEM ------
1. Add Item
2. Remove Item
3. View Bill
4. View Total Amount
5. End
Enter Your Choice: 1
Enter Item Name: Munch
Enter Price Of The Item: 10
Item Has Been Added!
```

```
------ THE STORE BILLING SYSTEM ------
1. Add Item
2. Remove Item
3. View Bill
4. View Total Amount
5. End
Enter Your Choice: 4

Total Amount: ₹65.0

------ THE STORE BILLING SYSTEM ------
1. Add Item
2. Remove Item
3. View Bill
4. View Total Amount
5. End
Enter Your Choice: 5
Thank you for using the Store Billing System!
PS C:\Users\lalan\OneDrive\Documents\Project>
```

7. Testing Approach

The Store Billing System was tested manually with several scenarios: adding multiple items, removing existing and non-existing items, viewing the bill when it is empty and when it has several items, and checking the total amount calculation. Invalid choices such as entering a non-numeric value for price or selecting a menu option outside the range 1–5 were also tested to verify error messages and program stability.

8. Challenges Faced

Handling invalid price input required careful string validation so that users could not accidentally crash the program by entering letters instead of numbers.

Ensuring that item removal works correctly even when the user types the item name with different cases or unexpected spacing was another practical challenge.

9. Learnings & Key Takeaways

This project strengthened understanding of how to use lists, loops and conditional statements to manage data in a real-world scenario.

It demonstrated the advantages of separating logic into different modules for better readability and maintenance.

The project also highlighted the importance of user input validation and clear messaging in console applications.

10. Future Enhancements

Add support for item quantities instead of assuming quantity 1 for every item.

Introduce discount and tax calculations (for example, GST) to provide a more realistic billing experience.

Allow saving and loading bills from files so that records can be stored for future reference.

Enhance the user interface by adding colours or migrating to a graphical interface.

11. References

Python Official Documentation – https://docs.python.org/3/

VITyarthi Build Your Own Project Guidelines.