Restaurant Point of Sale (POS) System

Project Report

a) Cover Page

Title:
Restaurant Point of Sale (POS) System

Team Members:

- Hamza Mahboob

Course:
OOP

Instructor:
Usman Anwer

Submission Date:
Monday, May 12, 2025

b) Abstract

This project presents a console-based Restaurant Point of Sale (POS) system developed in C++. The system facilitates menu management, order taking, checkout processing, and receipt generation. It supports multiple item orders, stock management, and secure admin authentication for menu modifications.

The design leverages core Object-Oriented Programming (OOP) concepts such as **encapsulation**, **composition**, and **abstraction**, organizing the code into logical, reusable classes like Product, Inventory, Cart, and POSSystem. These principles improve code maintainability, clarity, and extensibility.

CamScanner

c) Table of Contents

1. Cover Page
2. Abstract
3. Table of Contents
4. Introduction
5. OOP Concepts Used
6. Class Diagrams
7. Test Cases
8. Future Directions

d) Introduction

Problem Statement

Manual restaurant order and billing management is prone to errors and inefficiencies. A software solution is needed to automate these processes, improve accuracy, and speed up service.

Objectives

- Develop a user-friendly POS system to manage menu items and customer orders.
- Enable multi-item ordering with real-time stock validation.
- Calculate bills including applicable taxes and generate receipts.
- Secure admin functionalities with authentication.
- Persist menu and order data for continuity.

Motivation

The project aims to simplify restaurant operations by automating order taking and billing, reducing human error, and enhancing customer satisfaction. It also serves as a practical application of OOP principles in software development.

e) OOP Concepts Used

- **Encapsulation:**
  Each entity (Product, Inventory, OrderItem, Cart, POSSystem) encapsulates its data and behavior, promoting modularity and data hiding.
- **Composition:**
  The Cart class is composed of multiple OrderItem objects, which in turn reference Product objects, modeling real-world relationships.
- **Abstraction:**
  POSSystem abstracts the overall system workflow, providing a clean interface for user

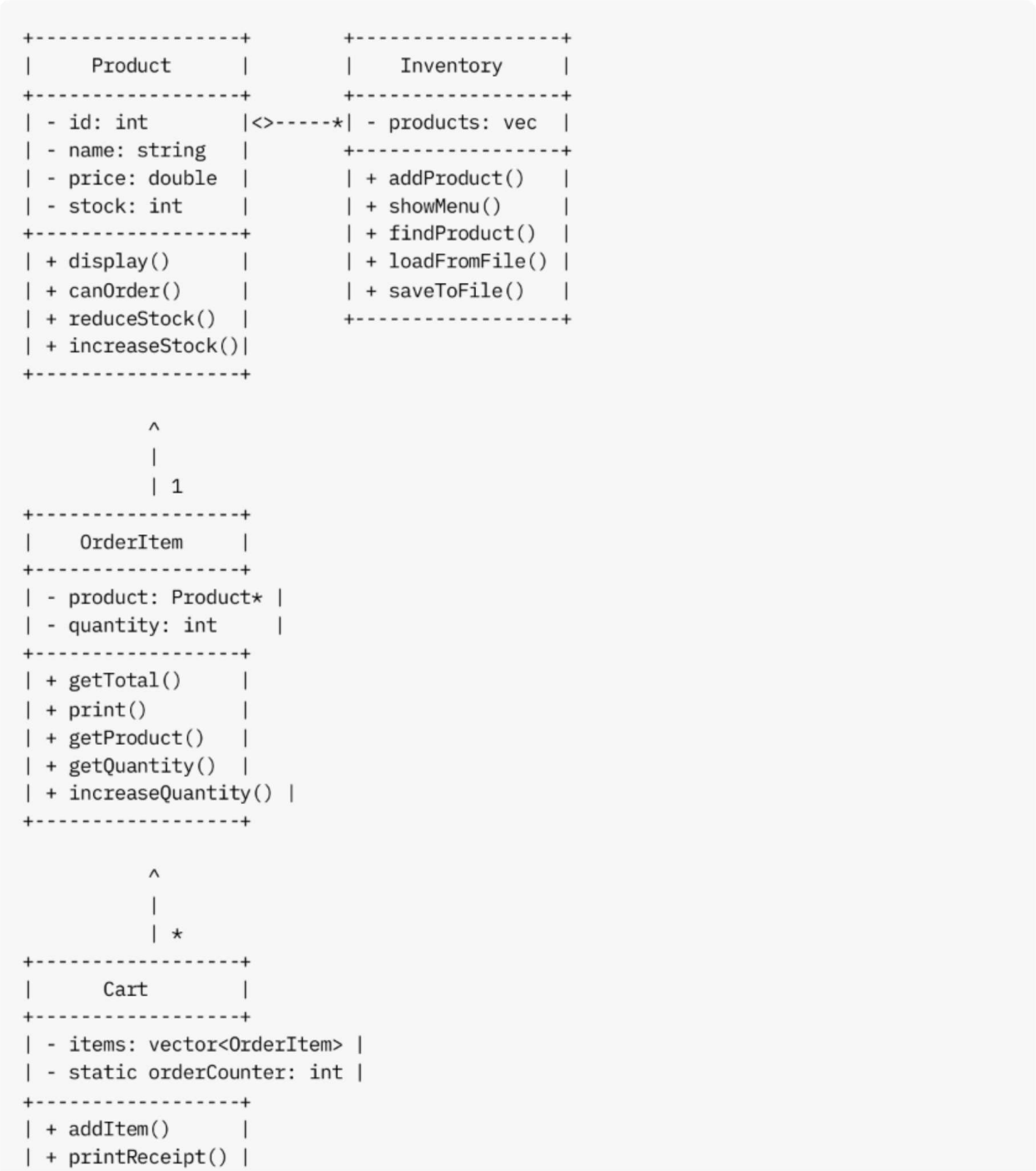
CamScanner

interaction while hiding complex internal operations.

- **Static Members:**
  Used in Cart to maintain a unique, incrementing order number for receipts.
- **Note:**
  The current system does not implement inheritance or polymorphism explicitly but is designed to allow easy future integration of these features.

f) Class Diagrams

UML Class Diagram (Textual Overview)



CamScanner

```
| + saveReceipt()  |
| + empty()        |
| + clear()        |
| + listItems()    |
| + removeItem()   |
+------------------+

          ^
          |
          | 1
+------------------+
|    POSSystem     |
+------------------+
| - menu: Inventory |
| - cart: Cart      |
+------------------+
| + start()        |
| + authenticateAdmin() |
| + adminMenu()    |
+------------------+
```

Legend:

- Solid diamond (composition) between Inventory and Product, Cart and OrderItem.
- Aggregation (hollow arrow) from OrderItem to Product.

g) Test Cases

Test Case ID	Description	Input	Expected Output	Result
TC1	View menu with default dishes	Select "View Menu"	Display list of dishes with correct details	Pass
TC2	Add valid dish to order	Dish ID=1, Quantity=2	"Added to order!" confirmation, stock reduced	Pass
TC3	Add dish with quantity > stock	Dish ID=2, Quantity=100	"Only X available" error message	Pass
TC4	Cancel order item	Cancel item #1 from order	Item removed, stock restored	Pass
TC5	Checkout with items in cart	Select "Checkout"	Receipt printed with subtotal, tax, total; receipt saved to file	Pass
TC6	Admin login with correct creds	Username=admin, Password=pass123	Access granted, admin menu displayed	Pass
TC7	Admin login with wrong creds	Username=admin, Password=wrong	"Invalid credentials" message, access denied after 3 attempts	Pass
TC8	Add new dish via admin menu	Enter dish details	Dish added and saved to menu file	Pass
TC9	Update stock via admin menu	Enter dish ID and new stock	Stock updated and saved to menu file	Pass

CamScanner

Test Case ID	Description	Input	Expected Output	Result
TC10	Exit program	Select "Exit"	Program terminates gracefully	Pass

h) Future Directions

- **GUI Implementation:** Develop a graphical user interface for improved usability.
- **Database Integration:** Replace text files with a database for better data management and scalability.
- **User Roles & Permissions:** Implement role-based access control for cashiers, managers, and admins.
- **Payment Gateway Integration:** Add support for electronic payments and invoice generation.
- **Order Editing:** Allow modification or removal of order items before checkout.
- **Table Management:** Track orders by table numbers for dine-in service.
- **Reporting & Analytics:** Generate sales reports, inventory alerts, and customer analytics.
- **Multi-terminal Support:** Enable networked POS terminals for larger restaurant setups.

CamScanner