CST-239\_StoreFront\_Assignment\_Report.docx

**Cover Page**

**CST-239 Final Assignment Report  
StoreFront Application: Design, Implementation, and Demonstration**

**Batossa Bakouma  
May 2025  
Grand Canyon University  
CST-239: Object-Oriented Programming  
Instructor: Komal Chhibber**

**1. Introduction**

This project demonstrates the application of object-oriented programming principles through the design and implementation of a console-based storefront system. The system allows users to browse products, add items to a shopping cart, cancel purchases, and complete checkout, all managed in-memory using Java collections.

**2. Objectives**

* Design class relationships using UML class diagrams.
* Visualize program logic using a system flowchart.
* Implement a modular Java application using OOP best practices.
* Use JavaDoc to document all code.
* Demonstrate full functionality through compilation, execution, and testing.

**3. UML Class Design**

The class diagram includes:

* Salable – An interface defining common behavior for all products.
* SalableProduct – Implements the Salable interface and represents a sellable item.
* InventoryManager – Manages product stock and availability.
* ShoppingCart – Tracks items added to a user's session.
* StoreFront – Serves as a façade for coordinating inventory and cart logic.
* StoreFrontApp – Provides the main menu-driven console interface.

Relationships include aggregation and composition, with a clear separation of concerns between data, business logic, and user interface.

**4. Flowchart Design**

The flowchart outlines user interactions:

* Display catalog
* Add item to cart → Check inventory → Update cart
* Cancel purchase → Validate → Remove from cart
* Checkout → Confirm → Process payment → Update inventory
* Return to menu or Exit

Each logic branch is handled via clearly labeled decision paths with Yes/No outcomes.

**5. Java Implementation**

The program is structured into reusable classes across the edu.gcu.storefront package. Java 11 compatibility was ensured by avoiding newer language features like String.formatted() and switch expressions.

All code has been documented using JavaDoc comments, and the documentation was generated using the javadoc CLI tool and stored under the docs/ folder.

**6. Compilation and Execution**

The project can be compiled and run from the terminal using:

javac -d out $(Get-ChildItem -Recurse src -Filter \*.java).FullName

java -cp out edu.gcu.storefront.StoreFrontApp

**7. Demonstration Screencast**

The full 8-minute video walkthrough including design explanation, code overview, and console demo is available here:  
 <https://www.loom.com/share/e9a1f84eda3840a59b10f9c6df61f69d>

**Appendices**

* UML Class Diagram: diagrams/uml-class.png
* System Flowchart: diagrams/flowchart.png
* JavaDoc: docs/index.html
* Source Code: src/edu/gcu/storefront/