

IT5106 - Software Development Project

Project Proposal - 2024

(Submission deadline: 25th November 2024)

Candidate Details

Index No:	2300161
Name of candidate:	Avishka Nawagamuwa
Contact Numbers:	0768913695
Email:	Avishkaanawagamuwa@gmail.com

Supervisor/Advisor Details

	Supervisor 1 (IT Related)	Supervisor 2 (Optional)
Name		
Designation		
Workplace address		

Academic/ professional qualifications and memberships		
Work experience		
Email:		
Phone number/s		

Note: Any deviation of the final project from the project described in this proposal should be explained by the candidate in the final Project Report.

1. Title of Project

Web-Based Inventory and Order Management System for Sampath Grocery Store

2. Client Information

- **Name:** Mr. Sampath Perera
- **Address:** 123 Main Street, Colombo 05, Sri Lanka
- **Contact Information:** Phone: +94 77 987 6543, Email: info@sampathgrocery.lk

3. Introduction

The Web-Based Inventory and Order Management System for Sampath Grocery Store is designed to streamline retail and wholesale operations for a medium-sized grocery business. The system integrates an e-commerce website for customers to browse products, place orders, and schedule deliveries, alongside an admin dashboard for managing inventory, orders, deliveries, and customer loyalty programs. It supports offline functionality using Service Workers and IndexedDB, ensuring usability without internet access, ideal for interview demonstrations. The system addresses inefficiencies in manual inventory tracking, order processing, and delivery management, enhancing operational efficiency and customer satisfaction through automated reorder notifications, credit sales tracking, and loyalty features for frequent delivery items like bread and string hopsers.

4. Motivation

Sampath Grocery Store currently relies on manual processes, leading to challenges such as inaccurate stock tracking, delayed order processing, and difficulty managing loyalty customers and delivery schedules. Paper-based records are prone to errors, loss, or damage, and generating sales reports is time-consuming. The inability to track supplier details and purchasing patterns further complicates operations. This project is motivated by the need to modernize these processes, providing a scalable, offline-capable solution that improves efficiency, supports loyalty programs, and enhances customer experience. As a student passionate about full-stack development, this project aligns with my skills in Spring Boot and Bootstrap, offering a practical solution to real-world retail challenges.

5. Objectives

- Develop a responsive e-commerce website for customers to browse products, manage carts, and place orders with line and grand totals.
- Implement an inventory management system with automatic reorder notifications and manager approval workflows.
- Integrate a home delivery module supporting standard, route-based, and scheduled deliveries for loyalty customers.
- Ensure offline functionality using Service Workers and IndexedDB for product browsing and order placement.
- Support role-based authentication (admin, customer, delivery) and credit sales tracking.
- Provide a dashboard for admins to manage inventory, orders, deliveries, and analytics.

6. Scope

Included:

- User registration/login with role-based access (admin, customer, delivery).
- E-commerce website for product browsing, cart management, and order placement.
- Inventory management with stock tracking and reorder notifications.
- Delivery module with driver, vehicle, and route management.
- Loyalty program with scheduled delivery requests and credit limit tracking.
- Admin dashboard for managing orders, inventory, deliveries, and analytics.
- Offline functionality for the e-commerce website using Service Workers and IndexedDB.

Excluded:

- Payment gateway integration (simulated payments only).
- Mobile app development.
- Advanced AI for predictive analytics.
- Integration with external ERP systems.

Boundaries:

- Web-based system for desktop and mobile browsers.
- Uses H2 embedded database for standalone, offline operation.

7. Methodology

The project will follow an **Iterative Model** to accommodate changing requirements and ensure flexibility. The methodology includes the following phases:

- **Requirements Gathering:** Conduct interviews with the hypothetical client (Sampath Grocery Store) and analyze user needs to define functional and non-functional requirements.
- **Design:** Create an Entity-Relationship Diagram (ERD) for the database, wireframes for the UI using Figma, and API specifications for backend services.
- **Development:** Build the backend using Spring Boot with Spring Data JPA and REST APIs, the frontend using Bootstrap 5, and the database using H2. Implement offline support with Service Workers and IndexedDB.
- **Testing:** Perform unit testing (JUnit), integration testing, and user acceptance testing to ensure functionality and reliability.
- **Deployment:** Configure a standalone setup (mvn spring-boot:run) for local demonstration without internet dependency.

The iterative approach allows early feedback through prototypes, ensuring alignment with client needs and adaptability to changes.

8. Requirements

8.1 Functional Requirements

- **High Priority:**
 - User registration and login with role-based authentication.
 - Product browsing and adding to cart on the e-commerce website.
 - Order placement with line totals and grand total calculations.
 - Inventory tracking with real-time stock updates.
- **Medium Priority:**
 - Delivery scheduling and route assignment for frequent items.
 - Loyalty card management with scheduled delivery requests.
 - Automatic reorder notifications with manager approval.
- **Low Priority:**
 - Customer feedback submission for products.
 - Analytics dashboard with sales and inventory reports.

8.2 Non-functional Requirements

- **Usability:** Simple, intuitive UI with responsive design using Bootstrap.
- **Security:** Encrypted passwords and JWT-based authentication.
- **Performance:** Page load time <2 seconds, API response time <500ms.
- **Reliability:** Offline functionality for browsing and ordering using IndexedDB.
- **Availability:** System available locally 24/7 for demo purposes.

9. Deliverables

- Web application source code (Git repository).
- E-commerce website and admin dashboard (deployable locally).
- System documentation (architecture, API details).
- User manual (for customers and admins).
- ERD and H2 database schema.
- Test cases and results.
- Gantt chart and project timeline.
- Demo video or live presentation.

10. Feasibility Study

- **Technical Feasibility:** The system is feasible using open-source technologies (Spring Boot, Bootstrap 5, H2 database). Java, HTML, and JavaScript are well-supported for web development, and H2 ensures standalone operation without internet.

- **Financial Feasibility:** Minimal cost as all tools (VS Code, Java JDK, H2, Bootstrap) are free. No licensing fees or server costs for local demo.
- **Operational Feasibility:** The system is easy to use for admins, customers, and delivery staff. Offline support ensures reliability, and the modular design allows future scalability.

11. Project Plan

Gantt Chart

The project timeline spans from September 2024 to November 2024, with key tasks and milestones:

- **Week 1-2 (Sep 1-14):** Requirements gathering and ERD design.
- **Week 3-4 (Sep 15-28):** Database schema creation and UI wireframes.
- **Week 5-8 (Sep 29-Oct 26):** Backend development (Spring Boot APIs) and frontend development (Bootstrap UI).
- **Week 9-10 (Oct 27-Nov 9):** Offline functionality (Service Workers, IndexedDB) and testing (unit, integration).
- **Week 11 (Nov 10-16):** User acceptance testing and documentation.
- **Week 12 (Nov 17-24):** Final refinements, demo preparation, and submission.

12. Resource Requirements

Software	Hardware
Java JDK 17	Laptop with 16GB RAM
Spring Boot 3.2.0	Intel Core i7 Processor
Mongo Database	512GB SSD

13. Evaluation Criteria

- **Test Plan:** Define test cases for all functional requirements (e.g., order placement, reorder notifications).
- **Unit Testing:** Achieve >90% code coverage using JUnit.
- **Integration Testing:** Verify API and database interactions.
- **System Testing:** Test offline functionality and UI responsiveness.
- **User Feedback:** Conduct mock user testing for usability (score >4/5).
- **Supervisor Feedback:** Regular meetings to ensure alignment with objectives.
- **Success Metrics:** Meet all objectives, ensure offline demo works seamlessly.

14. References

- Spring Boot Documentation. (2024). Retrieved from <https://spring.io/projects/spring-boot>
- Bootstrap 5 Documentation. (2024). Retrieved from <https://getbootstrap.com>
- H2 Database Engine. (2024). Retrieved from <https://www.h2database.com>
- Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures. University of California, Irvine.
- Sommerville, I. (2015). Software Engineering (10th ed.). Pearson.