SOFTWARE REQUIREMENTS SPECIFICATION

AI-Powered Multi-Mart Grocery Shopping & Optimization Platform

Version 1.0

Prepared by:

1. Balaga Lokesh (2023BCS0141)
2. Barukula Brijesh (2022BCS0153)
3. Hafiz Feroz (2023BCS0006)

Submitted to:

Dr. Rosebell Paul
Lecturer

Contents

1 Introduction

1.1 Purpose

The purpose of this SRS is to define the requirements for the AI-Powered Multi-Mart Grocery Shopping and Optimization Platform. The system enables users to submit shopping lists, fetch real-time inventories and offers from multiple marts, and apply AI-driven sentiment analysis for product quality. It optimizes purchases across marts based on cost, delivery efficiency, and quality, while supporting secure checkout and real-time tracking. This document ensures all stakeholders have a common understanding of the system's goals and scope.

1.2 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, testers, and stakeholders of the platform. Developers should use it to understand system requirements, APIs, and integrations. Project managers can refer to it for planning deliverables and milestones, while testers can derive test cases from the functional and non-functional requirements. Stakeholders and clients may review it to validate that the system features align with business needs.

1.3 Project Scope

The AI-Powered Multi-Mart Grocery Shopping and Optimization Platform creates a unified space for users, partner marts, and administrators. It allows shoppers to enter free-text lists, view optimized purchase options across multiple marts, and track their orders in real time. The platform integrates AI-based sentiment analysis, multi-mart APIs, payment services, and delivery tracking to provide a cost-effective, high-quality shopping experience, while admins can monitor system performance and user satisfaction.

User cycle: Shoppers provide shopping lists; the system parses items using NLP, fetches inventories and offers from multiple marts, applies sentiment analysis for product quality, and generates an optimized split basket. Shoppers can then proceed to checkout, pay securely via Stripe, and track their orders until delivery. The platform is designed to be easy to use for non-technical users.

Mart cycle: Partner marts expose APIs for product inventory, pricing, and offers. Orders are intelligently split across marts to minimize cost and delivery time. Each mart updates stock availability and order status in real time, ensuring accurate information at checkout.

Admin cycle: Administrators monitor activity through dashboards. They can track user savings, evaluate AI performance, review sentiment accuracy, analyze order statistics, and manage notifications. Admins also ensure compliance with payment standards and oversee integrations with external services such as Google Maps and Stripe.

System goals: The platform aims to reduce shopping costs for users, maximize delivery efficiency, and enhance product quality transparency using AI-driven sentiment analysis. By integrating multiple marts under one system, it eliminates the need for users to manually compare prices and offers across websites.

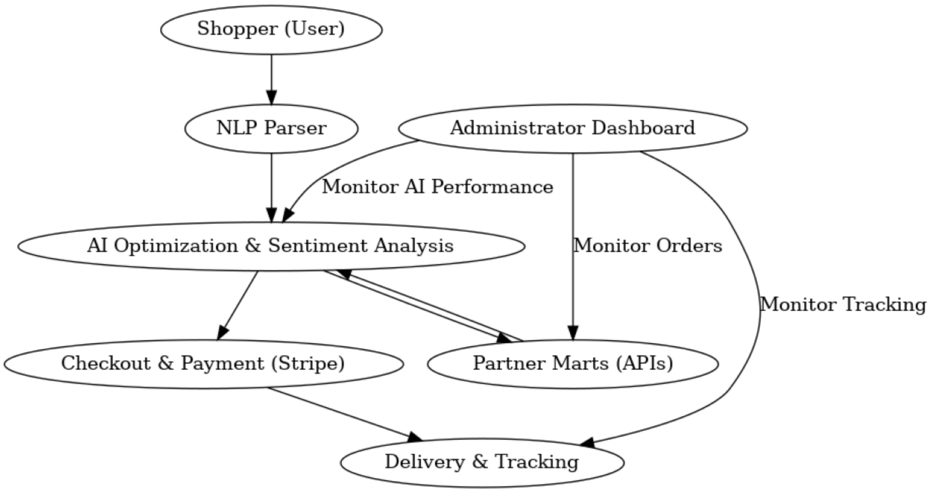

Figure 1.1: High-level workflow of the Grocery Optimization Platform

Figure 1.1 illustrates the overall workflow of the project. Shoppers interact with marts through the platform; AI services optimize decisions, payments are secured via Stripe, and delivery services handle last-mile tracking. Each entity—User, Mart, Admin, and AI Service—is interconnected, making the system both a shopping assistant and an optimization engine.

2 Overall Description

2.1 Product Perspective

The platform is designed as an intelligent middleware between users, partner marts, and delivery services. Traditional grocery shopping requires users to visit multiple marts or websites and manually compare prices. This platform replaces that manual process by fetching inventories and offers, applying AI-driven sentiment analysis to assess product quality, and generating optimized baskets based on cost and delivery efficiency. The system is a web-based application with a React frontend, Django backend, MySQL database for persistent storage, and integrations with external APIs such as Google Maps and Stripe.

2.2 User Classes and Characteristics

The platform supports three primary user classes:

- **Shoppers (End Users)**: General customers who create shopping lists, view optimized purchase options, complete checkout, and track deliveries. They are non-technical and require a simple interface.
- **Partner Marts (Vendors)**: Grocery stores that provide product data, inventory, and offers via APIs. They keep inventory updated and process orders.
- **Administrators (Admins)**: Technical staff who monitor the system, validate AI performance, manage dashboards, analyze statistics, and resolve operational issues.

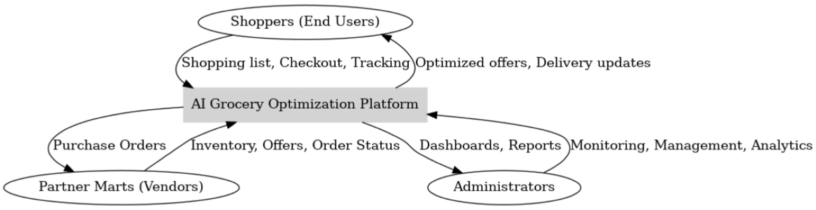
Shoppers (End Users)
Shopping list, Checkout, Tracking
Optimized offers, Delivery updates
AI Grocery Optimization Platform
Purchase Orders
Inventory, Offers, Order Status
Dashboards, Reports
Monitoring, Management, Analytics
Partner Marts (Vendors)
Administrators

Figure 2.1: User classes and their interactions with the system

2.3 Product Functions

The platform provides integrated functions that automate and optimize grocery shopping. Key functions include:

- **Shopping List Input:** Users submit free-text lists; the system parses them using NLP.
- **Inventory and Offer Retrieval:** Inventory, pricing, and offers are retrieved from the MySQL database (and optionally from partner APIs in future).
- **Sentiment-Based Quality Scoring:** AI-driven sentiment analysis computes quality scores from product reviews.
- **Basket Optimization:** Generates optimized split orders across marts to minimize cost and delivery time.
- **Checkout and Payment:** Secure payments via Stripe, showing costs, fees, and ETAs.
- **Order Finalization and Tracking:** Orders are assigned and tracked through to delivery.
- **Admin Monitoring:** Dashboards for savings, offer usage, and AI performance.

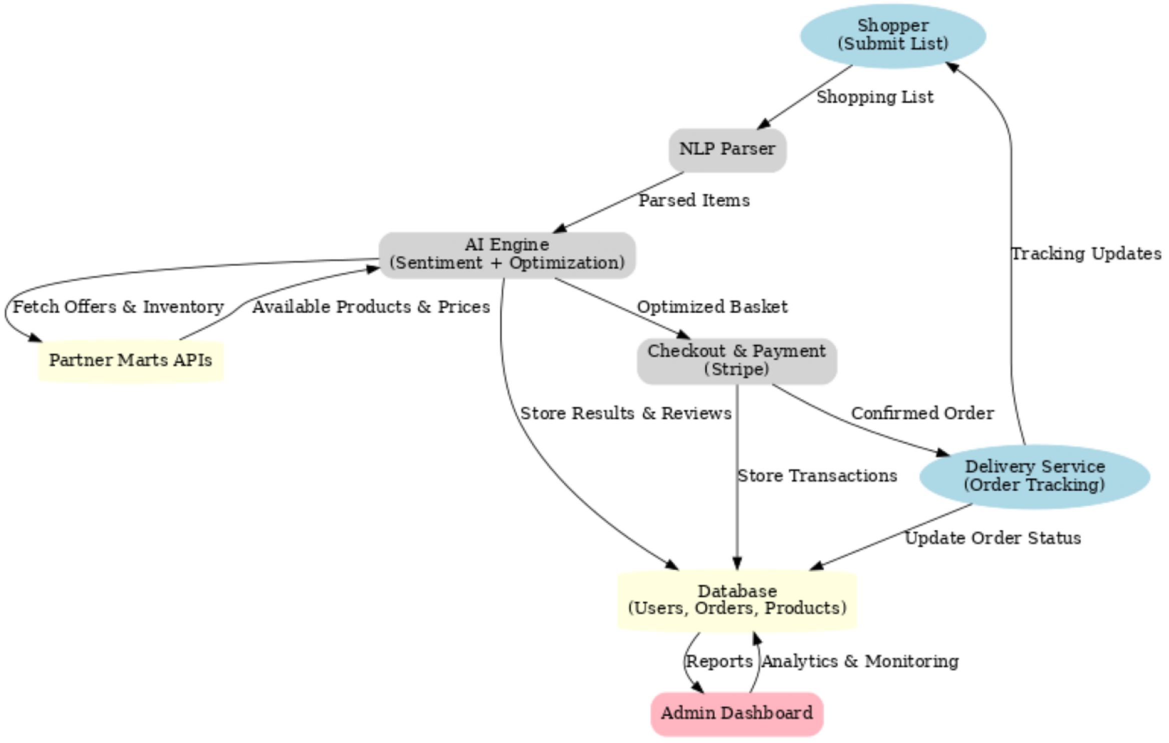

Figure 2.2: Data Flow Diagram of the Grocery Optimization Platform

2.4 Operating Environment

The platform is web-based and runs on major operating environments including Windows, macOS, and Linux. The frontend is browser-based (React) and supports modern browsers (Chrome, Firefox, Edge, Safari). Mobile responsiveness is supported. The backend (Django + MySQL) is deployable on Linux-based servers and containerized via Docker, and all services are exposed over HTTPS.

2.5 Design

The design is centered around three main actors: Shoppers, Partner Marts, and Administrators.

Shopper Activities

Shopper activities include:

- **Shopping List Input:** User enters free-text shopping list.
- **Optimization and Checkout:** System fetches inventory and offers, applies sentiment scoring and optimization, and displays the optimized basket.
- **Payment:** User completes secure payment via Stripe.
- **Order Tracking:** Real-time delivery updates until completion.

Each shopper profile stores personal information, past orders, savings history, and notifications.

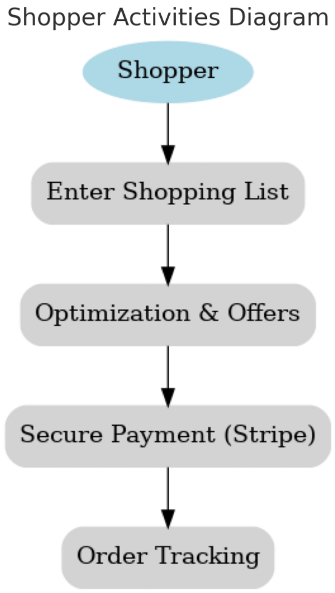

Figure 2.3: Shopper Activities

Mart Activities

Partner marts provide:

- **Inventory Updates:** Product availability, prices, and offers via APIs.
- **Order Fulfillment:** Process and confirm orders.
- **Status Updates:** Send order status for tracking.

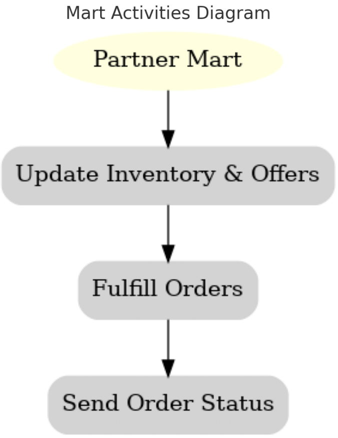

Figure 2.4: Mart Activities

Admin Activities

Administrators are responsible for:

- Monitoring system performance and AI optimization.
- Reviewing order statistics and user savings.
- Ensuring compliance with security and payment standards.
- Managing notifications, logs, and troubleshooting.

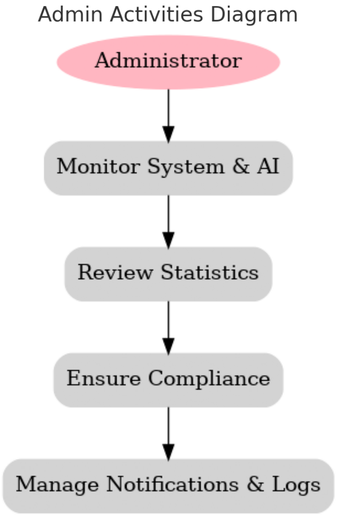

Figure 2.5: Admin Activities

3 System Features

The platform automates and optimizes grocery shopping, focusing on cost-effectiveness, quality, and convenience.

3.1 Description and Priority

Features prioritized from highest to supportive:

1. **Shopping List Input and Parsing** (High)
2. **Inventory and Offer Retrieval** (High)
3. **AI Optimization and Sentiment Scoring** (High)
4. **Checkout and Secure Payment** (High)
5. **Order Tracking** (Medium)
6. **Admin Monitoring** (Medium)
7. **Notifications and Reports** (Low/Supportive)

3.2 Functional Requirements

The platform provides the following FRs:

FR-1 **Shopping List Input**
Users shall be able to enter free-text shopping lists through the web interface. The system shall parse the input using an NLP pipeline to extract structured product items.

FR-2 **Inventory Retrieval**
The system shall fetch product availability, pricing, and offers directly from the MySQL database. (Optional: integrate partner mart APIs later.)

FR-3 **Sentiment-Based Quality Scoring**
The system shall analyze product reviews and assign sentiment-based quality scores to items, which will be used during optimization.

FR-4 **Basket Optimization**
The system shall generate an optimized basket across available marts/products to minimize cost and delivery time while ensuring product quality.

FR-5 **Checkout and Payment**
Users shall view the optimized basket with total cost, delivery fees, and estimated delivery time. Payments shall be processed securely via Stripe.

FR-6 **Order Finalization and Tracking**
The system shall confirm orders, assign them to marts/delivery services, and provide real-time tracking updates.

FR-7 **Admin Monitoring and Reports**
Administrators shall be able to view dashboards showing user savings, offer usage, AI performance, and order statistics.

4 Other Nonfunctional Requirements

4.1 Performance Requirements

The platform shall respond to optimization requests (shopping list parsing, inventory retrieval, and basket generation) within **5 seconds** under normal load. The system shall support at least **10,000 concurrent users** and keep inventory query response times under **2 seconds**.

4.2 Security Requirements

Only registered users shall access the system. Role-based access control shall enforce permissions. All sensitive data shall be encrypted (AES-256) and all communications secured with TLS 1.2+. Payments shall comply with PCI-DSS via Stripe.

4.3 Software Quality Attributes

Maintainability, reliability, usability, and testability are required:

- **Reliability:** Target uptime of 99.9%.
- **Usability:** Intuitive UI for non-technical users.
- **Maintainability:** Modular code structure for easy refactoring.
- **Testability:** Unit, integration, and UI tests with CI.

4.4 Business Rules

Primary business goals:

- Minimize shopping costs via multi-mart optimization.
- Provide transparency in product quality using AI sentiment scoring.
- Ensure reliable, secure order fulfillment with tracking.
- Save user time by removing manual price comparison.

5 Other Requirements

The platform requires ongoing maintenance and periodic updates.

- **Maintainability:** Modular architecture for easy updates and refactoring.
- **Portability:** Deployable across cloud providers using Docker containers.
- **Adaptability:** Support for future integrations (new marts, delivery services).
- **Compliance:** PCI-DSS, GDPR (as applicable).
- **Future Enhancements:** Voice shopping, loyalty programs, personalized recommendations.