Software Requirements Specification

Group 2

CompareCart

Software Engineering
Fall 2024

CS673 A1

Siming Zhao
Zichen Wang
Yifei Wang
Jingwei Ma
Sifat Singh Khalsa

1. Introduction

1.1 Purpose

The purpose of this SRS is to define the requirements for the development of the CompareCart software. CompareCart is a web application that enables users to compare products based on price and reviews from various online stores. This document outlines the functional and non-functional requirements of the system and provides a detailed overview of the system's capabilities, constraints, and user expectations.

1.2 Intended Audience

- **Developers**: To understand the functional requirements and technical constraints.
- **Product Owner/Managers**: To understand the scope and functionality of the project.
- **Testers**: To use the defined requirements for creating test cases.
- **Stakeholders**: To understand how CompareCart will operate and meet their expectations.

1.3 Scope

CompareCart aims to simplify the shopping experience by allowing users to compare prices, reviews, and product details across different online stores. The system will scrape data from multiple e-commerce platforms, aggregate the product information, and display it to users for comparison. Users will be able to search for products, view reviews, and access product details in one unified platform.

---

2. Overall Description

2.1 Product Perspective

CompareCart will be developed as a web application that acts as an aggregator of product information from various online retailers. The system will interact with APIs or web scraping tools to extract product data such as prices, reviews, and other details from external sources.

System Interfaces:

- Web scraping modules/API to gather product data from online stores.
- Integration with payment gateway (optional, future version).

User Interfaces:

- A responsive and user-friendly UI where users can search for products and view detailed comparisons.

Hardware/Software Requirements:

- A web browser for users to access the application.
- Server infrastructure to handle scraping tasks, store data, and serve the website.

2.2 Product Functions

The CompareCart software will offer the following functionalities:

- Product search and comparison: Users can search for products by name, category, or brand.
- Price comparison: The application will display product prices from various online retailers.
- Product reviews: CompareCart will display aggregated user reviews for products.
- Filtering and sorting: Users can filter products based on price, rating, or store.

2.3 User Characteristics

- **End Users**: Individuals using the platform to compare product prices and reviews.
- **Admins**: Users managing the platform, ensuring that data is accurate, and performing maintenance tasks.

2.4 Assumptions and Dependencies

- Online stores will not block scraping or API calls.
- The online & in-store prices will remain the same and would not be subjected to any further discounts

---

3. Functional Requirements

3.1 Product Search and Comparison

- **Input**: The system should allow the user to search products by name.
- **Process**: The system will fetch the product details from different online stores.
- **Output**: The system will display a list of products with their respective prices, reviews, and links to the store.

User Story: As a user, I want to search and view different products and their details like prices, reviews etc.

3.2 Product Details Page

- The system should allow users to click on any product to view more detailed information such as product specifications, reviews.

User Story: As a user, I want to view the product details.

3.3 Sorting and Filtering

- The system will allow users to sort the results based on:
    - Price
    - Rating
    - Store

User Story: As a user, I want to filter results based on pricing, rating and store from which I can make the purchase

3.4 Review Aggregation

- The system will display aggregated reviews and ratings from different platforms.
- The System will provide users with both numerical ratings and textual reviews.

User Story: As a user, I want to view the reviews and ratings of products from all platforms.

3.5 Admin Features

- Admins can:
    - Update the list of supported online stores.
    - Monitor web scraping processes to ensure data accuracy.
    - Manage user accounts and permissions

User Story: As an admin, I want to have access to be able to update supported stores, monitor scrapping and manage user accounts.

3.6 User Reviews and Ratings

- The system must allow users to leave their own reviews and rate products within CompareCart.

User Story: As a user, I want to leave remarks and review of the products on ComapreCart.

3.7 Social Sharing

- The systems should have a feature for users to share products and their comparisons.

User Story: As a user, I want to share the product details and prices.

---

4. Non-Functional Requirements

4.1 Performance Requirements

- The system should return search results within 3-5 seconds under normal load.
- The system should support up to 1,000 concurrent users without performance degradation.

4.2 Usability Requirements

- The UI will be intuitive and easy to use, with clear navigation paths for product comparison.

4.3 Reliability Requirements

- The system will maintain a 99.9% uptime guarantee, with backups performed regularly to avoid data loss.
- Error logging and notification systems should be implemented to track and report any system failures.

4.4 Backup and Disaster Recovery

- Regular backups must be taken to ensure data is recoverable in case of system failure.
- The system should be equipped with disaster recovery protocols to restore operations in case of unforeseen issues.

4.4 Data Accuracy

- The system must ensure that the data scraped from third-party websites is accurate and up to date.
- Scraped data will be validated and refreshed periodically (e.g., every 24 hours or based on website activity).

This SRS serves as a comprehensive guideline for developing the CompareCart platform and meeting user expectations while addressing system constraints and performance requirements.