Habib University

Dhanani School of Science and Engineering

Milestone: ANT-MAN (Software Specification Document)

SHOQ Mobiles

Group Members

Hammad Sajid (hs07606)
Sarim Tahir (st07112)
Muhammad Owais Waheed (ow07611)
Muhammad Khubaib Mukaddam (mk07218)

Contents

1 Introduction

1.1 Purpose

The purpose of this Software specification document is to outline the functional and non-functional requirements for the development of the Price Comparison E-commerce Application, referred to as "SHOQ Mobiles." It serves as a guide for the development team to understand the project goals, functionalities, and constraints.

1.2 Intended Audience

The intended audience for this document includes members of the development team, project stakeholders, and the quality assurance team. Development team members will use this document as a reference for implementing features, stakeholders will use it to understand project scope and progress, and the quality assurance team will refer to it for testing and validation.

1.3 Intended Use

The Price Comparison E-commerce Application is intended to provide users with a convenient platform to compare prices from various online listings and find the best deals for products they are interested in purchasing. Users will interact with the application through a mobile app interface, accessing features such as product search, price comparison, and redirection to original product listings for purchase.

1.4 Product Scope

The scope of the product includes the development of a mobile application for Android and Web platforms, a backend server to handle user requests and data processing, and a database to store user data, product information, and scraped data from third-party e-commerce websites. The application will focus on providing a seamless user experience with intuitive navigation and efficient price comparison functionalities.

1.5 Definitions and Acronyms

SRS: Software Requirements Specification

UI: User Interface

UX: User Experience

API: Application Programming Interface

CRUD: Create, Read, Update, Delete

2 Overall Description

2.1 User Needs

The application aims to address the following user needs:

Ability to search for products using keywords or filters.

Access to detailed product listings including images, descriptions, prices, and seller/external website information.

Comparison of prices and deals across multiple sources to find the best offer.

Seamless redirection to original product listings on external e-commerce websites for purchase.

2.2 Assumptions and Dependencies

The development and functionality of the Price Comparison E-commerce Application are based on the following assumptions and dependencies:

Availability of a stable internet connection for data fetching and user interaction.

Accessibility of third-party e-commerce websites for web scraping and data retrieval.

Compliance with legal and ethical guidelines regarding web scraping and data usage.

3 System Features and Requirements

3.1 Functional Requirements

3.1.1 Searching for Products

Users should be able to search for products by entering keywords or applying filters such as category, price range, and brand.

The search functionality should provide relevant and accurate results based on user input.

3.1.2 Viewing Product Listings

Users should be able to view detailed product listings including images, descriptions, prices, and seller information.

Product listings should be presented in a clear and organized manner to facilitate easy comparison.

3.1.3 Comparing Prices and Deals

Users should be able to compare prices and deals for the same product across different sources.

Price comparison should consider factors such as shipping costs, discounts, and availability to provide accurate comparisons.

3.1.4 Redirecting to Original Listings

Users should be able to click on a product listing to be redirected to the original listing on the external e-commerce website for purchase.

The redirection process should be seamless and efficient, ensuring a smooth transition for users.

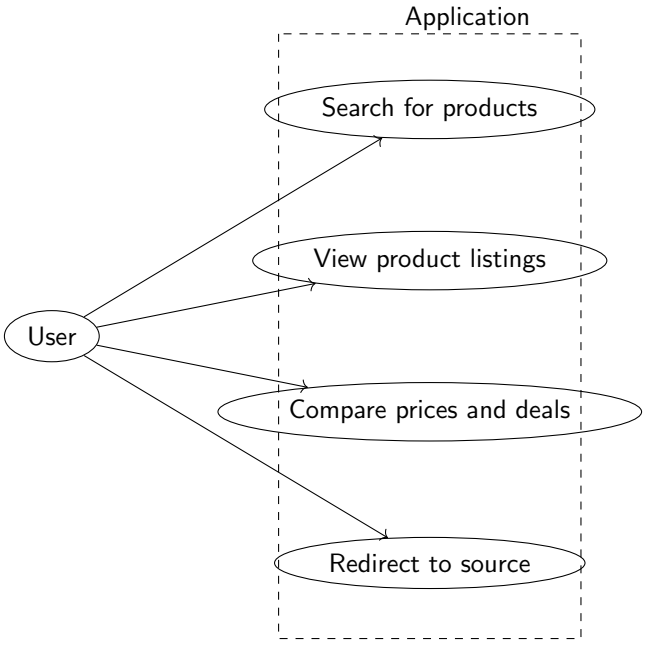

Figure 1: Use Case Diagram

3.2 External Interface Requirements

3.2.1 User Interface

The user interface should be intuitive, responsive, and visually appealing on both web and mobile platforms.

UI elements should be well-designed and organized to enhance user experience and facilitate easy navigation.

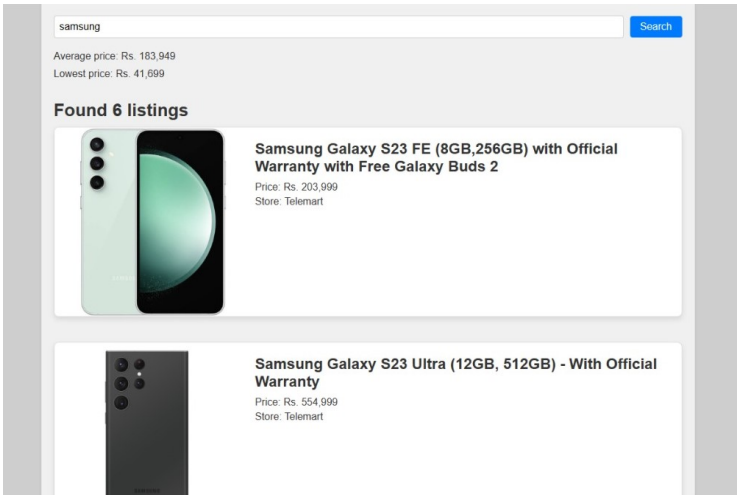

Figure 2: UI Mockup

3.2.2 API

RESTful APIs should be implemented for communication between the frontend and backend components of the application. Further information on the APIs can be found in the API documentation.

APIs should follow standard conventions for request and response formats, supporting CRUD operations for data management.

3.3 System Features

3.3.1 Frontend Web App

The Web/mobile application should be developed using the Flutter framework to ensure cross-platform compatibility.

UI components should be implemented using Flutter widgets to provide a native-like experience on Android devices and web browsers.

3.3.2 Backend Server

The backend server should be developed using Node.js, providing a scalable and efficient platform for handling user requests and data processing.

APIs should be implemented to handle user authentication, product search, price comparison, and data retrieval from the database.

3.3.3 Database

MongoDB should be used as the database management system to store user data, product information, and scraped data from third-party websites.

Data should be organized and indexed to optimize retrieval and storage operations, ensuring fast and efficient access.

3.4 Nonfunctional Requirements

3.4.1 Performance

The application should achieve fast response times and minimal latency to provide a smooth and responsive user experience.

Response times for search queries, product listings, and price comparison should be optimized to ensure timely results.

3.4.2 Scalability

The application architecture should be designed to scale horizontally to handle a growing number of users and product listings.

Load balancing and auto-scaling mechanisms should be implemented to distribute traffic evenly and maintain performance during peak usage periods.

3.4.3 Reliability

The application should minimize downtime and ensure consistent availability to users.

Error handling mechanisms should be implemented to gracefully handle exceptions and prevent system failures.

3.4.4 Security

The application should implement robust security measures to protect user data and prevent unauthorized access.

Encryption should be used to secure sensitive information such as user credentials and payment details.

3.4.5 Accessibility

The application should adhere to accessibility standards to ensure inclusivity and provide equal access to all users, including those with disabilities.

UI elements should be designed with accessibility features such as screen reader compatibility and keyboard navigation support.