Software Requirements Specification

for

<RecycleBin>

Version 1.0 approved

Prepared by

<Mrinal Kanti Saha Joy, Md. Julker Nayon, Foysal Hasan Rafi>

<ULAB>

<25.10.2023>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

<The purpose of the RecycleBin e-commerce app project is to develop a comprehensive and user-friendly online platform that facilitates the buying, selling, and borrowing of products with a focus on sustainability and responsible consumption.>

1.2 Document Conventions

<This document follows the IEEE standard for SRS. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3 Intended Audience and Reading Suggestions

<The document is intended for developers, project managers, users, and testers. Describe what the rest of this SRS contains and how it is organized. The sequence for reading the document begins with the overview sections and proceeds through the most pertinent sections.>

1.4 Product Scope

<RecycleBin. The app will work as an e-commerce site. There will be two sections for user, one is who uploads and sell products and another is who purchases or borrow products. There will be a payment system for paying bills for products online and offline. It will save money.>

2. Overall Description

2.1 Product Perspective

<The RecycleBin e-commerce app is a standalone system designed to provide a user-friendly and sustainable marketplace for the buying, selling, and borrowing of products. The app operates independently but may integrate with external services and APIs to enhance its functionalities.>

2.2 Product Functions

<The app will have different types of functions. User Registration and Authentication, product Listing and Management, product Search and Filtering, order Management, payment Processing, user Profile Management, Push Notifications, help and Support.>

2.3 User Classes and Characteristics

<The various user classes that we anticipate will use this product. User classes may be differentiated based on frequency of use, and subset of product functions used. One user can upload products and another user can borrow them or purchase them. one will do a payment and the owner will receive payment and deliver the item to the user's address within a limited time.>

2.4 Operating Environment

<It will work on Android or iOS, websites.>

2.5 Design and Implementation Constraints

<There will be any items or issues that will limit the options available to the developers. These might include corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 User Documentation

<The user documentation components (such as user manuals, online help, and tutorials) will be delivered along with the software.>

2.7 Assumptions and Dependencies

<We will depend on some third-party e-banking payment systems like Bkash, Nagad, and Online bank system payment.>

3. External Interface Requirements

3.1 User Interfaces

< First user will have a login or sign-up screen, and then the user will get two types of options for entering another page, is he/she going into there as a product uploader or product buyer or borrower? They will select a product and make payment >

3.2 Hardware Interfaces

<The app supports a range of mobile devices, including smartphones and tablets, running on Android and iOS operating systems. It is also accessible through web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.

The app is optimized for various screen sizes and resolutions commonly found on mobile devices and desktop/laptop screens. Users can interact with the app through touchscreens, mouse clicks, and keyboard inputs. The app integrates with the cameras of mobile devices for features like product image uploads and scanning capabilities.>

3.3 Software Interfaces

<software interfaces refer to the interactions and integrations with other software components, systems, and external services. These interfaces encompass data exchange with databases, communication with third-party APIs, and dependencies on specific software systems and libraries. The software interfaces define the protocols, data formats, and methods by which the app interacts with these external software elements, ensuring a seamless and functional user experience.>

3.4 Communications Interfaces

<The communication interface for the RecycleBin e-commerce app involves the use of internet connectivity and local area networks (LANs) to enable real-time data exchange between the application and external systems, including servers and third-party services. This interface allows for the seamless transfer of information, including product listings, order data, and payment transactions, ensuring a responsive and interactive user experience while maintaining secure and efficient communication channels.>

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 Functional Requirements

4.1.1. User Registration and Authentication

- *Users can create accounts with unique usernames and passwords.*
- *Email verification and password recovery functionality.*
- *Secure authentication mechanisms to protect user data.*

4.1.2. Product Listing and Management

- *Sellers can create, edit, and manage product listings.*
- *Upload product images and descriptions.*
- *Specify product details like price, condition, and category.*
- *Include location information for products.*
- *Real-time preview and editing of listings.*

4.1.3. Product Search and Filtering

- *Comprehensive search functionality to find products by keywords.*
- *Advanced filters for refining search results by category, price range, location, and more.*
- *Sorting options for search results.*
- *Product recommendations based on user preferences.*

4.1.4. Order Management

- *Users can place orders for products they wish to buy or borrow.*
- *Manage and track order status.*
- *In-app messaging for communication between buyers and sellers.*
- *Review and rating system for products and sellers.*

4.1.5. Payment Processing

- *Integration with various online payment gateways for secure online transactions.*
- *Support for offline payment methods, including cash-on-delivery and in-person card payments.*
- *Payment tracking and status updates for users.*
- *Refund and dispute resolution processes.*

4.1.6. User Profile Management

- *Users can update their profiles, including contact information, profile picture, and payment details.*
- *View transaction history and order details.*
- *Configure privacy settings and notifications.*

4.1.7. Location-Based Services

- *Utilize GPS and location services to provide location-specific product listings.*
- *Find products available nearby.*
- *Display seller locations and directions for product pickup.*
- *Location-aware notifications and alerts.*

4.1.8. Security and Privacy

- *Encryption of sensitive data, including user credentials and payment information.*
- *Secure socket layer (SSL) for data transmission.*
- *Privacy controls for user data, including profile information and transaction history.*
- *Compliance with relevant data protection regulations.*

4.1.9. Push Notifications

- *Real-time notifications for order updates, messages, and new product listings.*
- *Customizable notification preferences for users.*
- *In-app and email notifications.*

4.1.10. Reporting and Moderation

- *Reporting system for users to flag inappropriate content, products, or users.*

- Moderation tools for administrators to review and take action on reported content.
- Implementation of community guidelines and terms of service.

4.1.11. Help and Support
- Access to a knowledge base and FAQs.
- In-app chat support for user inquiries and assistance.
- Reporting and resolution of technical issues.

These system features cover the core functionalities of the RecycleBin e-commerce app, ensuring a comprehensive and user-friendly platform for buying, selling, and borrowing products while promoting sustainability and responsible consumption.

4.2 System Feature 2 (and so on)

5. Other Nonfunctional Requirements

5.1 Performance Requirements

<Response Time: The app should provide quick response times, with pages and actions loading within a reasonable timeframe to ensure a smooth user experience.
Scalability: The system should be able to handle a growing number of users and product listings without significant performance degradation.
Data Throughput: The app should efficiently process data to support a high number of simultaneous transactions, including product listings, searches, and orders.
.>

5.2 Safety Requirements

<Secure Payment Processing: All payment transactions must be secured and comply with industry standards to protect user financial data.
Data Encryption: Sensitive user data, including personal information and payment details, should be encrypted both in transit and at rest to prevent unauthorized access.
User Safety: The app should include safety measures for user interactions, including chat moderation and reporting mechanisms for inappropriate content or behavior.>

5.3 Security Requirements

<Authentication and Authorization: Ensure robust user authentication and authorization mechanisms to protect user accounts and data.
Data Security: Implement secure coding practices and measures to protect against common security threats, such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).
Session Management: Manage user sessions securely to prevent session hijacking and unauthorized access.>

5.4 Software Quality Attributes

<Usability: The app should have an intuitive and user-friendly interface to ensure ease of use for both sellers and buyers.
Reliability: The system should be highly available and reliable, with minimal downtime and data integrity.
Maintainability: The codebase should be well-structured and documented to facilitate future maintenance and updates.
Portability: The app should be compatible with a wide range of devices and browsers, ensuring accessibility for users.>

5.5 Business Rules

<Product Listing Guidelines: Define rules for product listings, including content standards, image quality, and acceptable product categories.
Payment Processing Rules: Specify rules for processing payments, including transaction fees, refund policies, and payment confirmation procedures.
Order Management Rules: Define the order handling process, including cancellation policies, return policies, and dispute resolution mechanisms.
These nonfunctional requirements ensure that the RecycleBin e-commerce app not only functions effectively but also provides a safe, secure, and user-friendly experience for both buyers and sellers. They address performance, safety, security, software quality, and compliance with business rules to create a robust and reliable platform.>

6. Other Requirements

<Develop and maintain a comprehensive privacy policy that informs users about data collection, storage, and usage practices. Create and enforce terms of service that outline user responsibilities and guidelines for using the app. The app should support multiple languages, including translation of user interfaces and product listings to cater to a diverse user base.
Cultural Adaptation: Ensure that the app's content and design are culturally sensitive and appropriate for users from different regions. Adhere to accessibility standards (e.g., WCAG) to make the app accessible to individuals with disabilities, including support for screen readers and keyboard navigation. Perform regular accessibility testing to identify and address accessibility issues. Implement effective error-handling mechanisms to provide informative error messages to users in case of system errors or user input issues. Maintain detailed system logs for debugging and monitoring purposes, ensuring that logs are securely stored and accessible to authorized personnel.
Regularly backup user data, including product listings, orders, and user profiles, to prevent data loss in case of system failures. Develop a disaster recovery plan to ensure system availability in the event of unforeseen disasters or server failures. Implement performance monitoring tools and practices to identify and address performance bottlenecks and optimize system efficiency. Conduct load testing to assess system performance under high traffic conditions and make necessary adjustments. Provide user support through multiple channels, such as email, chat, or a dedicated support portal, to address user inquiries and issues. Offer training materials or online resources to help users effectively utilize the app's features. Implement a feedback mechanism for users to provide suggestions, report issues, and share their experiences with the app. Use user feedback to drive ongoing app improvements and updates.>

Appendix A: Glossary

<- API (Application Programming Interface): A set of rules and protocols that allow different software applications to communicate with each other.

- Authentication: The process of verifying the identity of a user or system to ensure secure access to specific resources or services.

- Backend: The server-side of a software application, responsible for data storage, processing, and business logic.

- E-commerce: Electronic commerce, the buying and selling of products or services over the internet.

- Frontend: The client-side of a software application, responsible for the user interface and user interactions.

- GPS (Global Positioning System): A satellite-based navigation system that provides location and time information anywhere on Earth.

- HTTPS (Hypertext Transfer Protocol Secure): A secure version of HTTP, used to encrypt data transmission between a web browser and a web server.

- Mobile Application: A software application designed for use on mobile devices such as smartphones and tablets.

- Payment Gateway: A service that processes payment transactions, securely transmitting data between a customer, merchant, and the payment processor.

- QR Code (Quick Response Code): A two-dimensional barcode that can be scanned with a mobile device to access information, such as URLs or product details.

- SSL (Secure Sockets Layer): A protocol for establishing secure and encrypted connections on the internet.

- User Profile: An account or digital representation of a user within the app, containing personal information, settings, and transaction history.

- UX (User Experience): The overall experience and satisfaction a user has when interacting with a software application.

- UI (User Interface): The visual elements and design of an application that users interact with.

- WebSocket: A communication protocol that provides full-duplex, bidirectional communication channels over a single TCP connection.
>

Appendix B: To Be Determined List

<1. Integration with Third-Party Services: The specific third-party services and APIs that will be integrated into the app for features such as social media sharing, product recommendations, and location services.

2. Payment Gateway Selection: The choice of payment gateways and processors to be used for online payment transactions and associated fees.

3. Data Backup and Recovery Procedures: The development of a detailed data backup and disaster recovery plan, including backup frequencies and restoration processes.

4. Performance Metrics: Determining specific performance metrics, such as maximum response times and data throughput requirements, for the app under various load conditions.

6. User Support Channels: The establishment of channels and workflows for user support, including response times and support team availability.

7. Security Audits: Conducting security audits and vulnerability assessments to ensure the app's robustness against security threats.

8. Release Schedule: Establishing a detailed release schedule for updates and new features based on user feedback and development milestones.>