

---

# **Software Requirements Specification**

**for**

**Campus Barter: A Point-Based Exchange System for Students**

**Version 1.0 approved**

**Prepared by Deepesh Bhardwaj-SE21UCSE049**

**Gnan Reddy-SE21UCSE064**

**Rachana Vinjamur-SE21UCSE159**

**Sanskar Nagda-SE21UCSE191**

**Ramya Swarna-SE21UARI113**

**Jahnavi Gali-SE21UARI053**

**Aditya Bhardwaj-SE21UARI007**

**Rishith N V-SE21UCSE163**

**08/03/2024**

# Table of Contents

<b>Table of Contents .....</b>	<b>ii</b>
<b>Revision History .....</b>	<b>Error! Bookmark not defined.</b>
<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions .....	1
1.4 Product Scope .....	1
1.5 References.....	2
<b>2. Overall Description .....</b>	<b>2</b>
2.1 Product Perspective.....	2
2.2 Product Functions .....	2
2.3 Operating Environment.....	2
2.4 Design and Implementation Constraints .....	3
2.5 User Documentation .....	3
2.6 Assumptions and Dependencies .....	3
2.7 Functional Requirements Specification .....	
<b>3. External Interface Requirements .....</b>	<b>4</b>
3.1 User Interfaces .....	4
3.2 Hardware Interfaces .....	5
3.3 Software Interfaces .....	5
3.4 Communications Interfaces .....	5
<b>4. System Features .....</b>	<b>6</b>
4.1 Functional Requirements .....	6
<b>5. Other Nonfunctional Requirements.....</b>	<b>7</b>
5.1 Performance Requirements .....	7
5.2 Safety Requirements .....	7
5.3 Security Requirements .....	7
5.4 Software Quality Attributes .....	7
<b>6. Other Requirements .....</b>	<b>8</b>
<b>Appendix A: Glossary.....</b>	<b>8</b>

# 1. Introduction

## 1.1 Purpose

*This Software Requirements Specification (SRS) document specifies the software requirements for the Campus Barter platform, version 1.0. Campus Barter is designed as a point-based exchange system catering to students at educational institutions, aiming to facilitate the borrowing, lending, or exchanging of items like books, sports equipment, and ride shares without monetary transactions. This SRS covers the entire system, detailing the functionalities and specifications required to develop and implement the platform.*

## 1.2 Document Conventions

*This SRS follows standard documentation conventions to ensure clarity and consistency. Important terminologies and acronyms are defined in the glossary section. Requirements are presented in a structured format, using:*

- **Bold** for key features and requirements,
- *Italics* for emphasis on certain aspects or considerations, and
- **Monospaced font** for system outputs and code snippets. Each requirement statement is assigned a unique identifier and priority level, ensuring clear traceability and prioritization throughout the development process.

## 1.3 Intended Audience and Reading Suggestions

*This document is intended for a broad audience, including:*

- **Developers**, who will implement and test the specified requirements,
- **Project Managers**, overseeing the project's progression,
- **Marketing Staff**, responsible for promoting the platform,
- **Users (Students)**, who will interact with the system,
- **Testers**, ensuring the system meets quality standards, and
- **Documentation Writers**, creating user guides and help materials.

*The SRS is organized into sections starting with an overview, followed by detailed descriptions of system functionalities, interfaces, and performance requirements. Readers are suggested to begin with the Introduction (Section 1) and Product Scope (Section 1.4), before moving onto sections most relevant to their role.*

## 1.4 Product Scope

*Campus Barter is a digital exchange platform that introduces a unique, point-based transaction system for students. It aims to promote sustainability, community engagement, and equitable sharing of resources without direct cash transactions. The platform's key features include a monthly rewards system, a user-friendly interface, and mechanisms for security, trust, and environmental impact tracking. By fostering a strong, engaged community, Campus Barter seeks not just to transform how students interact with their resources but to encourage a sustainable, collaborative lifestyle. This project aligns with corporate goals of innovation in educational environments and business strategies focusing on social responsibility and sustainability.*

## 1.5 References

- *Project Statement of Work: Campus Barter Platform, Version 1.0.* This document provides the initial project overview, vision, and objectives.
- *User Interface Style Guide:* Specifies the design standards and guidelines for the platform's interface.
- *System Requirements Specifications for Similar Systems:* Offers insight into standard practices and benchmarks.
- *Vision and Scope Document:* Further details the project's goals, benefits, and intended impact within the target community.
- *Environmental Sustainability Reports:* Provides data and methodologies for calculating the environmental impact of platform participation.

## 2. Overall Description

### 2.1 Product Perspective

*Campus Barter is a new, self-contained product designed to introduce a sustainable, community-driven exchange system among students within educational institutions. It stands as a novel platform distinct from traditional buy/sell marketplaces and informal borrowing/lending practices by offering a point-based transaction system without direct cash exchanges. As a standalone system, it focuses on promoting sustainability, community engagement, and equitable sharing of resources. While Campus Barter is initially developed as an independent product, its design allows for future integration into larger educational or community platforms, facilitating broader adoption and scalability.*

### 2.2 Product Functions

*Campus Barter will provide a range of functions to facilitate the sharing of goods and services among students:*

- **Point-Based Transaction System:** Enables users to lend, borrow, or exchange items using a system of barter points, ensuring equitable transactions.
- **Monthly Rewards System:** Rewards the most active users with additional points, encouraging continuous engagement.
- **User-Friendly Interface:** Offers an intuitive platform with search filters, item categories, and a user and item rating system to enhance the user experience.
- **Community Forums and Sustainable Living Tips:** Supports community building by offering forums for discussion, success stories, and tips for sustainable living.
- **Security Features:** Includes a robust user verification process, a secure messaging system for communication, and a transparent rating system for users and items.
- **Environmental Impact Tracking:** Provides users with insights into the environmental benefits of their participation through savings in money, carbon footprint, and waste reduction.

### 2.3 Operating Environment

*The software will be designed to operate across a wide range of devices and platforms to ensure maximum accessibility for students. It will be compatible with:*

- *Major operating systems: Windows, macOS, Linux, iOS, Android.*
- *Web browsers: Chrome, Firefox, Safari, Edge.*
- *Hardware: Smartphones, tablets, laptops, desktop computers.*

## 2.4 Design and Implementation Constraints

- **Regulatory Policies:** *Compliance with data protection regulations (e.g., PDP Bill).*
- **Technology Stack:** *Use of specific frameworks and languages (e.g., React for front-end, Node.js for back-end) as determined by preliminary project decisions.*
- **Accessibility Standards:** *Adherence to WCAG 2.1 for ensuring accessibility to all users.*
- **Security Protocols:** *Implementation of secure authentication and data encryption standards.*
- **Integration Capabilities:** *Designed to allow future integration with existing educational or social platforms without significant overhaul.*

## 2.5 User Documentation

- **User Manual:** *Detailed guide on using the platform.*
- **Online Help:** *Context-sensitive help accessible within the platform.*
- **Tutorials:** *Step-by-step video and text tutorials for key features.*

*Documentation will be provided in accessible formats (PDF, HTML) to ensure usability across devices.*

## 2.6 Assumptions and Dependencies

- **Assumption:** *Availability of a reliable internet connection for users.*
- **Dependency:** *Use of third-party services for email verification and environmental impact calculations.*
- **Assumption:** *User willingness to engage in a point-based exchange system rather than traditional monetary transactions.*

## 2.7 Functional Requirement Specifications

### 1. FRS1: User Registration and Authentication

- *Users must be able to register and authenticate using email and password.*
- *System must perform email verification to ensure authenticity.*
- *Error handling for failed authentication attempts or invalid inputs.*

### 2. FRS2: Item Listing and Searching

- *Users can list items or services for lending, borrowing, or exchange.*
- *Items/services must be categorizable and searchable by various filters (e.g., category, location, point value).*

### 3. FRS3: Transaction Management

- *System must track and manage point transactions between users.*
- *Includes error handling for insufficient points or incomplete transactions.*

### 4. FRS4: Community Engagement

- *Implementation of forums and success stories section for user interaction and community building.*

### 5. FRS5: Environmental Impact Tracking

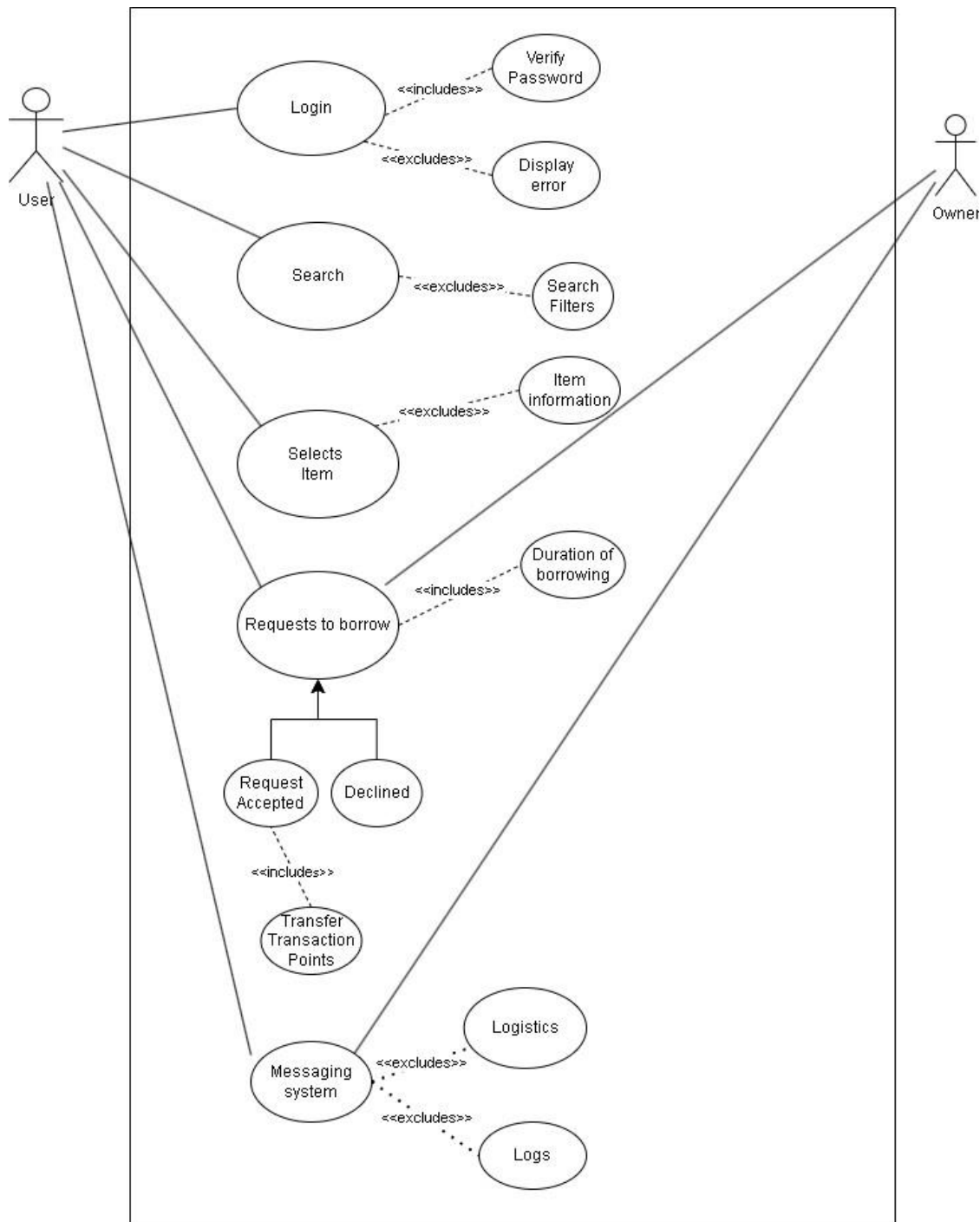
- *Feature to calculate and display the estimated environmental savings of user transactions.*

Each requirement is crucial for fulfilling the comprehensive services offered by the feature, aimed at ensuring a seamless, engaging, and sustainable exchange platform for students.

### 2.7.1 Role use case 1

**Use case:** *Borrowing an Item*

**Diagram:**



**Brief Description:** *This use case describes the process by which a user can search for, request, and borrow an item using barter points.*

### Initial Step-By-Step Description

1. **User logs in** and navigates to the search page.
2. **User searches** for an item, applying necessary filters.
3. **User selects** an item and views detailed information.
4. **User requests** to borrow the item, specifying the duration.
5. **Item owner receives the request** and approves or denies it.
6. If approved, **transaction points are transferred** from the borrower to the lender.
7. **Borrower and lender coordinate** pickup/return logistics through the secure messaging system.
8. Upon return, both parties **confirm the transaction's completion**, and the system logs the transaction.

*This use case exemplifies a typical interaction within the Campus Barter platform, highlighting its functionality and user-focused design.*

## 3. External Interface Requirements

### 3.1 User Interfaces

*The Campus Barter platform will prioritize a user-friendly, intuitive interface that adheres to the following guidelines:*

- **GUI Standards:** *Follows the Material Design guidelines for a clean, modern look and feel.*
- **Sample Screen Images:** *Key screens will include the Home Page, Search Results, Item Listing, User Profile, and Transaction History. Detailed UI designs will be documented separately.*
- **Screen Layout Constraints:** *Responsive design to accommodate various screen sizes from mobile devices to desktop monitors.*
- **Standard Buttons and Functions:** *Each screen will feature consistent navigation elements, including a Home button, Search bar, User Profile access, and a Help option.*
- **Keyboard Shortcuts:** *For desktop users, shortcuts for navigation and common actions (e.g., Ctrl + S for search) will be implemented.*
- **Error Message Display Standards:** *Error messages will be clear, concise, and provide guidance for resolution. They will appear in a consistent format across the platform.*
- **Components Needing UI:** *Major components include the user registration/login module, item listing and search module, transaction management system, community forum, and environmental impact tracker.*

### 3.2 Hardware Interfaces

*Campus Barter is designed to operate with minimal hardware interface requirements, focusing on broad compatibility:*

- **Supported Device Types:** *Smartphones, tablets, laptops, and desktop computers.*
- **Data and Control Interactions:** *Primarily through touch or mouse/keyboard inputs, with outputs displayed visually on screen. No specialized hardware interactions are planned.*

- **Communication Protocols:** Standard internet protocols (HTTP/HTTPS) for data transmission.

### 3.3 Software Interfaces

- **Operating Systems:** Windows, macOS, Linux for desktops; iOS and Android for mobile devices.
- **Web Browsers:** Chrome, Firefox, Safari, Edge.
- **Database:** MongoDB for storing user data, item listings, transactions, and forum posts.
- **Back-end Framework:** Node.js for server-side operations.
- **Front-end Framework:** React for dynamic user interface rendering.
- **Authentication Service:** Integration with OAuth for secure user authentication.
- **Email Service:** SMTP protocol for sending notifications and verifications.
- **APIs and Libraries:** Use of Google Maps API for location-based services and Stripe API for optional future monetary transactions.

### 3.4 Communications Interfaces

- **E-mail Communications:** Automated email notifications for registration verification, transaction updates, and monthly rewards announcements.
- **Web Browser Communication:** Utilizes standard HTTP/HTTPS protocols for all user interactions with the platform.
- **Network Server Communications:** RESTful API architecture for client-server communication, ensuring scalability and flexibility.
- **Electronic Forms:** Utilized for user registration, item listing, and feedback submission.
- **Communication Standards:** HTTP/HTTPS for web traffic, SMTP for email, and WebSocket for real-time updates in the community forum.
- **Security:** SSL/TLS encryption for all data transmission to ensure user data privacy and integrity.
- **Data Transfer Rates:** Optimized for low latency and high throughput to ensure a responsive user experience even on slower connections.
- **Synchronization Mechanisms:** Real-time data synchronization for user profiles, item listings, and transactions using WebSocket for live updates.

## 4. System Features

### 4.1 Item Listing and Borrowing

<b>Use Case Name</b>	<i>Item Listing and Borrowing</i>
<b>XRef</b>	<i>Section 4.1, Item Listing and Borrowing. SDD, Section 7.2</i>
<b>Trigger</b>	<i>The User accesses the Campus Barter platform intending to list an item for borrowing or to borrow an item.</i>
<b>Precondition</b>	<i>The platform is accessible with options for users to list items or search for items to borrow.</i>
<b>Basic Path</b>	<i>1. The User chooses to either list an item or search for items to borrow. The choices for searching are by Category, by Item Condition, and by Keyword.</i>



	<ol style="list-style-type: none"> <li>2. <i>If listing an item, the user fills out a form including item details (name, category, condition, point value) and submits it for listing.</i></li> <li>3. <i>If the search is by Category, the system presents a categorized list of items available for borrowing.</i></li> <li>4. <i>The User selects a category.</i></li> <li>5. <i>The system presents a list of items in that category.</i></li> <li>6. <i>The User selects an item to view details.</i></li> <li>7. <i>The system displays detailed information including item condition, point value, and borrowing terms.</i></li> <li>8. <i>The User selects to borrow the item, initiating a transaction, or to return to the item list or the previous list.</i></li> </ol>
<b>Alternative Paths</b>	<p><i>In step 3, if the User selects to search by Item Condition, the system presents items filtered by the selected condition. Return to step 5.</i></p> <p><i>In step 3, if the User selects to search by Keyword, the system presents a dialog box to enter the keyword or phrase.</i></p> <ul style="list-style-type: none"> <li>• <i>The User enters a keyword or phrase.</i></li> <li>• <i>The system searches item descriptions for the keyword or phrase and presents a list of matching items. Return to step 5.</i></li> </ul>
<b>Postcondition</b>	<i>The transaction for borrowing or listing an item is initiated, with system tracking and managing point exchanges as necessary.</i>
<b>Exception Paths</b>	<i>The User may abandon the listing or search at any time.</i>
<b>Other</b>	<i>The categories and item conditions list is generated from the information provided by users when items are listed, not predefined in the Campus Barter database.</i>

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

- **Response Time:** *The system should display search results within 2 seconds under normal load conditions, ensuring a smooth and efficient user experience.*
- **Concurrent Users:** *The platform must support up to 10,000 concurrent users without significant degradation in performance, catering to the potential high demand during peak hours.*
- **Database Transactions:** *Database read and write operations should complete within 100 milliseconds, ensuring timely updates and retrievals of data for a responsive system.*

### 5.2 Safety Requirements

- **Data Integrity:** *Measures must be implemented to prevent data loss or corruption resulting from system errors or failures, including regular backups and transaction logging.*
- **User Safety:** *Features such as secure messaging and user verification are required to ensure the safety and privacy of users when interacting and transacting on the platform.*

- **Regulatory Compliance:** The platform must comply with applicable data protection regulations (e.g., PDP Bill) to protect user data and privacy.

### 5.3 Security Requirements

- **Authentication:** Users must authenticate using a secure method (e.g., OAuth, two-factor authentication) to access the platform and perform transactions.
- **Data Encryption:** All data transmitted over the network and stored in the database must be encrypted using industry-standard protocols (e.g., TLS for data in transit, AES for data at rest).
- **Access Control:** The system shall implement role-based access control (RBAC) to restrict access to sensitive features and data according to user roles.

### 5.4 Software Quality Attributes

- **Usability:** The platform must feature an intuitive interface and navigational structure, allowing new users to perform basic tasks without instruction.
- **Reliability:** The system should be operational 24/7, with downtime limited to scheduled maintenance windows communicated in advance to users.
- **Maintainability:** The codebase should follow industry-standard practices and documentation, enabling efficient updates and troubleshooting.
- **Portability:** The web application should be compatible with the latest versions of major browsers (Chrome, Firefox, Safari, Edge) and adaptable to various screen sizes.

## 6. Other Requirements

- **Internationalization:** The platform should support internationalization to accommodate users from different geographical locations, including options for multiple languages and time zones.
- **Legal Requirements:** Compliance with all relevant laws and regulations pertaining to online marketplaces and digital platforms, including copyright and privacy laws.
- **Reuse Objectives:** Where possible, the project should leverage existing components and libraries to speed development and ensure reliability.

## Appendix A: Glossary

- **API:** Application Programming Interface. A set of routines, protocols, and tools for building software applications, specifying how software components should interact.
- **AES:** Advanced Encryption Standard. A symmetric encryption algorithm widely used to secure data.
- **GUI:** Graphical User Interface. A user interface that includes graphical elements, such as windows, icons, and buttons, for interaction with the device.
- **HTTP/HTTPS:** Hypertext Transfer Protocol/Secure. An application protocol used for transmitting hypermedia documents, such as HTML. HTTPS is the secure version of HTTP.
- **iOS/Android:** Operating systems for mobile devices. iOS is developed by Apple Inc., while Android is developed by Google.
- **JSON:** JavaScript Object Notation. A lightweight data-interchange format that is easy for humans to read and write and for machines to parse and generate.
- **Node.js:** An open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a web browser.

- **OAuth:** Open Authorization. An open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords.
- **PDP Bill:** Personal Data Protection Bill. Proposed legislation in India aimed at protecting personal data of individuals and establishing a Data Protection Authority for the same.
- **React:** A JavaScript library for building user interfaces, developed by Facebook.
- **RESTful API:** Representational State Transfer. An architectural style for designing networked applications, using stateless communications and standard web methods.
- **RBAC:** Role-Based Access Control. A method of regulating access to computer or network resources based on the roles of individual users within an enterprise.
- **SRS:** Software Requirements Specification. A document that describes the software system to be developed, outlining functional and nonfunctional requirements, and setting the foundation for the project design.
- **SSL/TLS:** Secure Sockets Layer/Transport Layer Security. Cryptographic protocols designed to provide communications security over a computer network.
- **UI/UX:** User Interface/User Experience. UI refers to the graphical layout of an application. UX involves the user's overall experience using that application.
- **WebSocket:** A computer communications protocol, providing full-duplex communication channels over a single TCP connection.