Software Requirements Specification

for

Point-Based Sharing System

Version <X.X>

Prepared by

Group Name: codeX crew

Srikruth	SE22UARI054	SE22UARI054@mahindrauniversity.edu.in
Nikita	SE22UARI112	SE22UARI112@mahindrauniversity.edu.in
Sanjay	SE22UARI006	SE22UARI006@mahindrauniversity.edu.in
Abhiram	SE22UARI155	SE22UARI155@mahindrauniversity.edu.in
Yashwanth	SE22UARI100	SE22UARI100@mahindrauniversity.edu.in
Harshitha	SE22UARI204	SE22UARI204@mahindrauniversity.edu.in

Instructor: *Avinash Arun*

Course: *Software Engineering*

Lab Section: *<place your lab section here>*

Teaching Assistant: *Nartkannai K*

10/03/25

Date:

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

1 Introduction

Point-Based Sharing System is an online system for the lending and borrowing of books and other resources for both students and teachers. It is based on a point system where individuals accumulate points when they lend something and use points when they borrow something. It ensures accountability and trust by maintaining digital inventory, reputation scores, and electronic monitoring of transactions. By offering a trusted and transparent exchange process, the platform fosters sharing of resources among social and academic communities.

This document presents an overall description of the software requirements of the Point-Based Sharing System. It identifies the purpose, scope, and audience of the system, as well as key definitions, so that all concerned understand its purpose and functionality. This section will assist readers in understanding the essential concepts and aims of the system before proceeding with the detailed requirements.

1.1 Document Purpose

This report outlines the software requirements of the Point-Based Sharing System. It gives a detailed description of the functional and non-functional requirements of the system, which allows students and instructors to exchange necessary resources effectively through a point-based exchange system.

1.2 Product Scope

The Point-Based Sharing System is a people - oriented platform that aims to promote responsible lending and borrowing of educational a nd leisure items like books, sporting goods, and vehicles. As opposed to conventional lending practices or current platforms like OLX and Facebook Marketplace, our system promotes accountability through a systematic point-based system and electronic monitoring.

1.3 Intended Audience and Document Overview

This document is intended for:

- End users, including students and teachers
- Developers and designers implementing the system
- Project managers overseeing the development
- Quality assurance testers ensuring functionality

It details system requirements, functionalities, constraints, and quality attributes.

1.4 Definitions, Acronyms and Abbreviations

- **Digital Inventory:** A centralized list of available items.
- **Point-Based System:** A system where users earn and spend points for borrowing and lending items.
- **Reputation Score:** A reliability rating based on user behaviour.

1.5 Document Conventions

This paper conforms to IEEE style of writing. Arial, 12 in size, fonts have been utilized with hierarchical numbering in the sections to promote easy readability and quick reference. Wherever applicable, diagrams, tables, and figures shall be incorporated for easier reading and understanding. The paper has been arranged for smooth accessibility for everyone, which would clearly facilitate system requirement and functionali ty comprehension.

2 Overall Description

2.1 Product Overview

The system offers a cost-effective way for students and instructors to exchange resources, cutting expenses and fostering cooperation. It features an interactive web-based interface for borrowing and lending transactions management.

Product Perspective:
The Point-Based Sharing System is in an academic setting, supplementing student and faculty current resource-sharing programs. It coordinates with institutional authentication networks for secure login, so only authenticated users can join. The system is unlike marketplaces because it reinforces accountability using a structured point-based framework. It discourages abuse and promotes equitable lending practices.

Below is a high-level diagram illustrating the system's interaction within its environment:

2.2 Product Functionality

- **Point-Based Transactions:** Users earn/spend points based on sharing history.
- **Inventory Tracking:** Monitors available and borrowed items.
- **Reputation System:** Rates users based on past interactions.
- **Automated Notifications:** Reminders for due dates and penalties for late returns.

2.3 Design and Implementation Constraints

Technology Stack: The system must be implemented using web technologies such as HTML, CSS, JavaScript, and a backend framework like Django.
Database Requirements: Uses a relational database (e.g., PostgreSQL) to store user information, transaction history, and inventory data.
Authentication: Secure login via institutional single sign-on (SSO) or OAuth authentication.
Performance Constraints: Must handle concurrent users without significant delays.
Security Considerations: Data encryption for sensitive information, role-based access control, and audit logs.

Design Standards: The system must adhere to the COMET method for software design and use UML modelling language for system diagrams.

2.4 Assumptions and Dependencies

1.Users will have stable internet access for interacting with the system.
2.The institution's authentication system will be accessible for user verification.
3.Notifications will be sent via email and SMS, assuming integration with a third-party messaging service.
4.The system will operate on institutional servers or a secure cloud-based infrastructure.
5.Future enhancements may integrate AI-based recommendation systems for lending suggestions.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The Point-Based Sharing System will have a web-based interface through which users can access the main functionalities. The user interface will comprise:

- Login/Registration Screen: Users will log in using their credentials or register.
- Dashboard: Shows points balance, recent activity, and available items to share.
- Item Listing Page: Enables users to list items, filter by category, and view availability.
- Point Transaction Screen: Enables users to process point transactions, view their point history, and monitor earned/spent points.

3.1.2 Hardware Interfaces

The system will communicate with the following hardware interfaces:

- Web Server: Host of the Point-Based Sharing System.
- Database Server: User profiles, item listings, and point transactions are stored in it.
- End-User Devices: Desktop computers, laptops, and tablets having web browsers to access the system.

All the user devices will need only an internet connection and a compatible web browser to use the platform.

3.1.3 Software Interfaces

The system will be integrated with the following software interfaces:

- Payment Gateway API: To process point purchases and wallet replenishments.
- Email Notification System: Sends notification of successful transactions, pending requests, and reminders.
- Database Management System (DBMS): Responsible for data storage for points, transactions, and user profiles.

3.2 Functional Requirements

3.2.1 F1: The system will permit users to register, log in, and edit their profiles.

3.2.2 F2: The system will permit users to list items for sharing, with item information and point values.

3.2.3 F3: The system will permit users to request items and alert item owners for approval.

3.2.4 F4: The system will handle point transactions, with correct point deductions for borrowing and point additions for lending.

3.2.5 F5: The system will keep a history of transaction, which will be visible to users via their profile dashboard.

3.3 Use Case Model

3.3.1 Use Case #1 (use case name and unique identifier – e.g. U1)

TO DO: Provide a specification for each use case diagram

Author – Identify team member who wrote this use case

Purpose - What is the basic objective of the use-case. What is it trying to achieve?

Requirements Traceability – Identify all requirements traced to this use case

Priority - What is the priority. Low, Medium, High. Importance of this use case being completed and functioning properly when system is depolyed

Preconditions - Any condition that must be satisfied before the use case begins

Post conditions - The conditions that will be satisfied after the use case successfully completes

Actors – Actors (human, system, devices, etc.) that trigger the use case to execute or provide input to the use case

Extends – If this is an extension use case, identify which use case(s) it extends

Flow of Events

1. Basic Flow - flow of events normally executed in the use-case
2. Alternative Flow - a secondary flow of events due to infrequent conditions
3. Exceptions - Exceptions that may happen during the execution of the use case

Includes (other use case IDs)

Notes/Issues - Any relevant notes or issues that need to be resolved

3.3.2 Use Case #2

…

4 Other Non-functional Requirements

4.1 Performance Requirements

The following performance expectations shall be met by the system:

1. Page Load Time:
The website will load within 3 seconds using typical internet access (4G or broadband).
The item list and search result will be loaded within 2 seconds upon query.
2. Reminder Notifications:
Reminder notifications on borrowing due-date should fire up at the prescribed time with up to 1 second delay.
Overdue return penalty reminder notifications need to be issued within 2 seconds when a deadline is missed.
3. Media and Inventory Loading:
Images of items available for sharing should be loaded in less than 2 seconds on viewing.
Description of an item, its borrowing history, and the profiles of borrowers must be retrieved within 2 seconds.
4. Concurrent User Handling:
At least 100 concurrent users must be supported by the system without affecting the performance.
The response of the server should be within 1.5 seconds in normal usage scenarios.
5. Performance of Database Queries:
Retrieval of a user's borrowing/lending history should be under 1 second.
Inventory updates system-wide should be done in 5 seconds to reflect real-time availability tracking.

4.2 Safety and Security Requirements

1. Authentication of Users and Protection of Data:
Users need to authenticate through a secure login mechanism (OTP-based or password-based).
User data (borrowed items, reputation scores) must be encrypted with AES-256 prior to storage.
Data transmission needs to be done through HTTPS to avoid unauthorized access.
2. Access Control:
Administrators must have privileged access for penalty enforcement and system monitoring.
Users must only be able to view and manage their own borrowing/lending history.

3. Privacy Compliance:
The system must be compliant with data privacy laws (e.g., GDPR, India's IT Act 2000).
Users must be able to delete their account and transaction history permanently on request.
4. Protection Against Cyber Threats:
CAPTCHA must be used to stop bots from creating spurious lending/borrowing requests.
Input validation should be implemented to avoid SQL injection and cross-site scripting (XSS) attacks.
5. Fail-Safe Mechanisms:
The system must restart automatically in 5 seconds if the system crashes.
In the event of server down time, reminders of borrowing due dates must be sent via email/SMS to users.

4.3 Software Quality Attributes

4.3.1 Usability
The interface must be user-friendly and easy to use, with clear navigation for borrowing and lending items.
There should be a tutorial section that informs new users on how to spend/earn points and utilize the system.
The system must display visual cues (e.g., color-coded status) for items that are borrowed, available, and overdue.
4.3.2 Reliability
The system must have 99.9% uptime to ensure there is no loss of access.
Transaction logs must be backed up every 24 hours to avoid data loss.
The system must be capable of automatically resuming operations in the event of unforeseen internet disconnections.
4.3.3 Maintainability
The codebase must be modular so that it is simple to update and debug.
Third-party integrations (such as authentication, notification services) must be loosely coupled so that they can be easily replaced if necessary.
Security patches and software updates must be deployed every 3 months.
4.3.4 Adaptability
The system must run flawlessly on mobile, tablet, and desktop platforms.
The interface must be multilingual with a minimum of 3 regional languages.
Future updates should allow integration with voice assistants (e.g., Alexa, Google Assistant) for easy navigation.

Appendix A – Data Dictionary

Name	Type	Description	Possible Values	Related Operations/Requirements

user_role	State Variable	Determines whether the user is a borrower or lender	Borrower, Lender	Used to control interface visibility
points_balance	Variable	Tracks the number of points a user has	Positive integers	Checked before approving item requests
Item_status	State Variable	Tracks if an item is available or borrowed	Available, Borrowed	Displayed on the item listing page
Request_status	State Variable	Tracks the state of a borrowing request	Pending, Approved, Rejected	Triggers notification events
Transaction type	Constant	Identifies the type of point transaction	Earned, Spent	Logged in transaction history
email	Input	User's email address	Valid email format	Used for account creation and notifications
password	Input	User's password for login	Alphanumeric string	Required for authentication
Confirmation code	Output	Code sent to user email for verification	6-digit numeric code	Used for email confirmation

Appendix B - Group Log

Meeting 1: Project Kickoff
Date & Time: 20-02-2025, 5:30 pm

Attendees: All

Agenda:

- Understanding the problem statement
- Defining the scope of the website for sharing resources
- Identifying initial functional requirements
- Assigning roles and responsibilities

Decisions Made:

- The website will provide services like item sharing, point transactions, and user accountability
- Each member will research relevant sharing platforms
- Next meeting scheduled for requirement gathering

Meeting 2: Requirement Gathering

Date & Time: 01-03-2025, 3:30 pm

Attendees: All

Agenda:

- Reviewing research findings
- Listing key functionalities for the system
- Discussing user-friendly design considerations

Decisions Made:

- Major functionalities finalized (user authentication, item browsing, point management, etc.)
- System will have a simple UI with clear navigation and notifications
- Next meeting scheduled for SRS documentation

Meeting 3: Drafting the SRS

Date & Time: 03-03-2025, 3:30 pm

Attendees: All

Agenda:

- Structuring the Software Requirements Specification (SRS)
- Assigning sections to team members for writing

Decisions Made:

- Each member assigned a section (Product Overview, Functionalities, Constraints, etc.)
- Use UML diagrams for design representation
- Next meeting scheduled for SRS review

Meeting 4: Review & Finalization

Date & Time: 04-03-2025, 4:30 pm

Attendees: All

Agenda:

- Reviewing the SRS document
- Making final edits based on feedback
- Ensuring completeness and accuracy

Decisions Made:

- Document formatted and reviewed for clarity
- Final version submitted

Group Activities Summary:

- Conducted research on resource-sharing platforms
- Defined key features and user requirements
- Developed UML diagrams and wireframes
- Drafted, reviewed, and finalized the SRS document

This log provides a detailed record of the team's collaboration and effort in producing the document.