Software Requirements Specification for
CampusLoop

CSE 328: Software Engineering Lab.

SUBMITTED BY

Name	ID	Intake
Geshan Molla	22235103561	51
Md.Mehedi Hasan	22235103558	51
Masum Ahmed	22235103573	51
Shorafat Hoshen	22235103574	51
Niloy Sarker	22235103564	51
Rakib Hossain	22235103335	51

SUPERVISED BY

Rubaiya Reza Sohana
Lecturer,
Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY
(BUBT)

Chapter 1

1. Introduction

1.1 Purpose

This document defines the software requirements for ***CampusLoop***, a mobile-based rental and community service platform designed for university students and beyond. Unlike traditional renting platforms, CampusLoop not only facilitates renting physical items (such as books, electronics, or sports gear) but also extends to skill-based services (such as academic tutoring, assignment help, editing, photography, and extracurricular coaching).

Additionally, the system provides a News & Events feature, enabling students and campus communities to stay updated on university happenings.

The mission of CampusLoop is to:

- Simplify access to goods and services for students.
- Promote peer-to-peer skill sharing and collaboration.
- Foster a connected and resourceful campus environment.
- Extend the idea of “rental” to include time, skills, and experiences, beyond physical items.

This SRS outlines functional and non-functional requirements, user interactions, system constraints, and performance expectations to ensure a seamless and scalable platform.

1.2 Intended Audience

The Software Requirements Specification is created to meet the needs of all stakeholders of CampusLoop. The primary stakeholders include:

- Administrators - Manage users, monitor transactions, enforce rules.
- Developers - Implement platform features as per specifications.
- QA/QC Engineers - Test and validate functionality, usability, and performance.
- End Users (Renters, Service Providers, Skill Seekers, Event Followers) - Interact with the platform to rent, share, or participate in services/events.
- Marketing Team - Promote the platform.
- Investors/University Stakeholders - Understand project goals, benefits, and risks.

1.3 Intended Use

The intended audience of this SRS has been identified in the previous section. In this section, we will describe how each group can use this document for a better understanding of their responsibilities and perspectives.

1.3.1 Administrators

Administrators will use this SRS to gain a clear understanding of the system requirements, deployment considerations, and backend functionalities of CampusLoop. They will rely on this document to determine their tasks, such as managing user accounts, verifying rental/service listings, moderating event posts etc. By following the requirements outlined here, administrators will be able to maintain smooth operations and ensure compliance with system policies.

1.3.2 Project Managers (PMs)

Project Managers will use this document to understand the scope of CampusLoop, stakeholder expectations, and development priorities. They will rely on the SRS to identify key deliverables, milestones, and constraints, ensuring that the project remains aligned with both the timeline and budget. The requirements described here will also help them oversee coordination between developers, testers, and analysts.

1.3.3 Developers

Developers will use the SRS to gain understanding of required functionalities, technical constraints, and user interactions. The document provides guidance for implementing rental listings, skill-service booking systems, live chat, and event updates. By following the specifications, developers can ensure that the app meets both functional and non-functional requirements.

1.3.4 QA/QC Engineers

Quality Assurance and Quality Control Engineers will use this SRS as a foundation for designing test cases and validation procedures. They will evaluate whether the implemented system meets the defined quality standards for performance, usability, and security.

1.3.5 Users (Renters, Service Providers, Skill Seekers, Event Followers)

For end-users, this SRS provides the system functionalities they will interact with. Renters and service seekers can expect clear flows for browsing, booking, and payments. Service providers and item lenders can rely on structured processes for listing and managing offerings. Event followers can stay informed about how campus news and events will be integrated into the app.

1.3.6 Stakeholders (Marketing, University Partners)

Stakeholders such as marketing staff, and supporting organizations will use the SRS to understand the system's scope and objectives. They will rely on this document to ensure the platform fulfills goals and provides opportunities for promotion and collaboration. By understanding system requirements, stakeholders can also adjust their strategies for marketing and outreach.

1.3.7 Testers

Testers will rely on the SRS to design validation scenarios and test workflows. The requirements specified here will help them confirm whether the platform functions as expected under different conditions. They will use the SRS to ensure that all critical user actions like renting, booking services, and browsing events perform smoothly and securely.

1.4 Product Scope

CampusLoop is a campus-focused mobile application where students and community members can rent both **physical items** and **skill-based services**. The system acts like a smart rental marketplace that not only allows people to lend and borrow items but also enables users to hire or offer academic and extracurricular skills. In addition, it provides a **news and events feature** that keeps students updated with everything happening around their campus.

1.4.1 Purpose

This system is designed to make campus life easier by creating a platform where students can rent what they need without buying, and also earn money by offering what they already have—whether it is physical goods or personal skills. It provides a single hub for sharing, collaboration, and campus engagement by combining **item rentals, skill-based services, and event updates** in one place.

1.4.2 Benefits and Objectives

- **Helping Students Earn:** It allows students to earn money by renting out their items or offering services such as tutoring, editing, or photography.

- **Making Renting Simple:** It makes it easy for students to find items or skills they need for a short time, without the cost of ownership.

- **Building Trust:** By using ratings and reviews, the system helps create trust between users, ensuring reliable transactions.

- **Strengthening Campus Community:** By offering both services and events, it encourages collaboration and active participation within the campus.

- **Keeping Up with Trends:** It adapts to modern lifestyles where temporary use and skill-sharing are often preferred over permanent ownership.

1.5 Risk Definition

The Software Requirements Specification identifies potential risks that may arise in the operation of CampusLoop. Users (both renters and service providers) may not always follow platform policies, rental/service terms, or ethical guidelines. Such non-compliance could result in disputes, administrative overload, or a disrupted user experience. These risks, if not managed properly, could compromise the overall reliability and success of the CampusLoop system.

1.5.1 User Inactivity

Risk that students may not actively participate in the platform by listing items, offering skills, or renting services. This lack of engagement could limit the growth of the CampusLoop community and affect its overall success.

1.5.2 Trust Issues

Risk that users may doubt the quality of listed skills or services, or the condition of rental items. This could discourage new users from participating and reduce long-term retention.

1.5.3 Security & Privacy Vulnerabilities

Risk that user data, financial transactions, or communication channels may be exposed to unauthorized access. Without strong encryption and compliance measures, this could harm trust and platform integrity.

1.5.4 Administrator Workload

Risk that administrators may face a heavy workload in verifying listings, moderating events, resolving disputes, and handling reports. This could slow down system processes and negatively impact user satisfaction.

1.5.5 Changing Student Needs

Risk that user demand may fluctuate depending on academic schedules, semesters, or events. This inconsistency could affect platform usage and engagement, requiring continuous updates to remain relevant.

Chapter 2 – Overall Description

2.1 User Classes and Characteristics
The CampusLoop platform serves three distinct user classes, each with specific roles and needs within the peer-to-peer ecosystem.

2.1.1 User Class: Borrower (Renter)

Characteristics:

- **Primary Goal:** To find and temporarily access needed items or skills cost-effectively.
- **Activities:** Browsing and searching listings, initiating rental or booking requests, and communicating with lenders/providers to clarify details.
- **Responsibilities:** Maintaining an accurate user profile, adhering to rental agreements, and returning items on time and in good condition.

2.1.2 User Class: Lender/Provider (Rentee)

Characteristics:

- **Primary Goal:** To earn income and reduce waste by monetizing underutilized assets or skills.
- **Activities:** Creating and managing detailed listings (for products or skills), setting availability and pricing, responding to booking requests, and coordinating handovers or sessions.
- **Responsibilities:** Ensuring listed items are functional and descriptions are accurate, maintaining a calendar of availability, and providing high-quality services.

2.1.3 User Class: Administrator

Characteristics:

- **Primary Goal:** To ensure the platform operates as a secure, trustworthy, and efficient marketplace.
- **Activities:** Verifying user university affiliations, moderating listings, monitoring transactions for fraud, resolving disputes between users, and managing platform policies.
- **Responsibilities:** Upholding community guidelines, protecting user data, and ensuring the overall integrity and reliability of the system.

2.2 User Needs

CampusLoop has multiple categories of users, and each category has its own expectations and requirements from the system. Addressing these needs is essential to ensure platform adoption, usability, and long-term success.

2.2.1 Renters (Item Borrowers)

Renters need a **simple and efficient way** to browse available items on the platform. They should be able to:

- Search for items by category, price range, or availability.
- View detailed product descriptions, images, and rental conditions.
- Communicate with providers to clarify terms before confirming a rental.
- Manage their rental history, track ongoing rentals, and receive timely reminders for returns.
- Ensure that transactions are **secure and transparent**, reducing the risk of disputes.

2.2.2 Rentees (Item Providers)

Rentees, who list items for others to borrow, need tools to effectively manage their offerings. They require:

- A **user-friendly listing interface** to upload product details, images, and pricing.
- Visibility so that their listings can reach a wide audience.
- Options to update or remove items as needed, based on availability.
- A secure payment process that ensures they receive compensation.
- Features to build **trust and credibility** through reviews and ratings.

2.2.3 Skill Providers

Skill Providers offer academic or extracurricular services through the platform. Their needs include:

- A structured way to **create service listings** with descriptions, pricing (hourly/daily rates), and schedules.
- Opportunities for **exposure and visibility** to students in need of services.
- Tools for managing bookings, confirming availability, and tracking service history.
- A system for feedback and reviews to strengthen their reputation.
- Secure and timely payments for their services.

2.2.4 Skill Seekers

Skill Seekers are students who need academic or extracurricular support. Their needs include:

- Easy **search and filtering options** to find the right provider.

- Transparent service details, including pricing and provider qualifications.
- A secure and straightforward **booking and payment process**.
- Assurance of service quality through ratings and reviews.
- Notifications and reminders about scheduled services.

2.2.5 Event and News Followers

These users rely on CampusLoop for updates on campus events. Their needs are:

- A **centralized events/news feed** with verified announcements.
- The ability to save, share, or register for events.
- **Notifications** to stay informed about upcoming activities.
- Real-time updates to avoid missing important events.

2.2.6 Administrators

Administrators manage the entire platform. Their needs are:

- A powerful **admin dashboard** to monitor user activity, transactions, and event listings.
- Tools to verify new listings and prevent fraudulent activity.
- Reporting and analytics to track platform performance.
- Features to resolve disputes between users.
- Control over system settings, ensuring security and compliance with policies.

2.3 Operating Environment

The CampusLoop application will operate under the following environment:

2.3.1 Hardware Platform

- Client-side: Android smartphones/tablets.
- Server-side: Vercel-hosted.

2.3.2 Operating System and Versions

- Minimum Android required.
- Backend server running Vercel.
- Future plan: iOS compatibility.

2.3.3 Software Components and Applications

- Frontend: React Native mobile application
- Backend: NodeJs.
- Push Notifications: Firebase Cloud Messaging
- Payment Processing: SSLCommerz/bKash/Nagad.

2.3.4 Database Compatibility

- Database: MongoDB.
- Vercel-hosted, optimized for scalability and high availability.

2.3.5 Interoperability

- Integration with payment gateways.
- Integration with university email systems for user verification.

2.3.6 Network Requirements

- Requires stable internet connection.
- Real-time features depend on active connectivity.

2.3.7 Security Considerations

- Secure storage of sensitive data (passwords, payments).
- Mandatory university email verification.

2.4 Constraints

The CampusLoop system will face the following constraints:

2.4.1 Technical Constraints

- Limited to Android devices during initial launch.
- System performance depends on stable internet connection.

2.4.2 Time Constraints

- MVP development and deployment deadline: **28 October 2025**.
- Three months of post-launch support until **30 December 2025**.

2.4.3 Budget Constraints

- Fixed budget of **30,000 BDT**, covering all development, hosting, payment, and outreach expenses.

2.4.4 Regulatory and Compliance Constraints

- Must comply with Bangladesh's financial transaction regulations.
- Must adhere to university's data privacy and security policies.

2.4.5 Resource Constraints

- Development team limited to six members.
- No additional human resources available outside the assigned team.

2.5 Assumptions

The development and operation of CampusLoop are based on several assumptions that define expected conditions for smooth functioning:

2.5.1 User Participation

It is assumed that users will actively participate by listing items, offering services, and engaging in rentals or event activities. Without active participation, the platform's value may decline.

2.5.2 User Proficiency

It is assumed that the majority of users (students and staff) are familiar with mobile applications, digital transactions, and online communication tools. This will minimize the need for extensive training or tutorials.

2.5.3 Administrator Authority

It is assumed that administrators will have sufficient authority and skills to enforce rules, moderate listings, and resolve conflicts to maintain platform integrity.

Chapter 3 – Requirements

3.1 Functional Requirements

3.1.1 User Registration

As a new user,
I want to sign up by providing my full name, university email/ID, gender, and password,
So that I can create a verified account and securely access the CampusLoop platform.

Success

- User provides all required details (full name, university email/ID, gender, and password) → System creates a verified account.
- User chooses "Sign up with Google" and authenticates successfully → System creates an account using their Google profile (linked to university email if applicable).
- System verifies the provided university email/ID → Account is activated and user can log in.

Failure

- Required fields are missing → System prevents submission and shows appropriate error messages.
- User tries to sign up with an already registered email/ID → System denies registration and shows: **"This email/ID is already registered. Please log in."**

3.1.2 Log in

As a registered user,
I want to log in using my verified university email/ID and password,
So that I can securely access my CampusLoop account and use platform features like browsing, renting, and listing items.

Success

- User enters a valid university email/ID and correct password → System grants access to their CampusLoop account.
- User selects "Login with Google" and authenticates successfully → System grants access to their CampusLoop account.
- After login, user navigates through the platform → They remain logged in until they log out

Failure

- User enters incorrect email/ID or password → System denies access and shows: **“Invalid email/ID or password.”**
- User tries to log in with an unregistered university email/ID → System denies access and shows: **“Account not found. Please sign up.”**

3.1.3 Search

As a user,
I want to search for products/services/content,
So that I can quickly find the exact information I need.

Success

- User enters a query in the search bar.
- System returns relevant products/services/content matching the search.

Failure

- If the search query is empty, system alerts user to enter a valid keyword.
- If no results are found, system shows: **“No items found.”**

3.1.4 Damage Waiver (Renter’s/Buyer Perspective)

As a renter,
I want the option to purchase a damage waiver during checkout,
So that I can reduce liability if the rented product is accidentally damaged.

Success

- Renter selects a product and proceeds to checkout.
- System displays an option to add a damage waiver with cost and coverage details.
- Renter chooses the waiver and confirms booking.
- System validates and stores renter’s choice in the order summary.
- Both renter and owner are notified that the waiver is included.

Failure

- If renter ignores the waiver option, system processes the order without waiver but shows a risk reminder.
- If renter tries to add waiver but payment fails, system alerts and guides renter to fix payment.
- After correction, renter successfully includes waiver and completes booking.

3.1.5 Security Deposit (Owner's/Seller Perspective)

As an owner,
I want to require a refundable deposit for my listed products,
So that I have compensation if the item is returned late or damaged.

Success

- Owner lists an item and sets deposit amount.
- Renter selects item and system shows deposit requirement.
- Renter pays rental fee + deposit at checkout.
- System securely holds deposit until item is returned.
- Upon return in good condition, system refunds deposit to renter.

Failure

- If owner forgets deposit, system prompts before listing goes live.
- If renter skips deposit payment, system prevents booking confirmation and shows error.
- If owner reports damage, system verifies claim and deducts from deposit.
- If claim invalid, system refunds full deposit to renter.

3.1.6 Post

As a seller,
I want to post about a product (sell, rent, or skill),
So that I can offer it to other users.

Success

- User selects 'Post' from menu.
- User selects category.
- User fills required information.
- User submits post.
- System validates and confirms with success message.

Failure

- If required fields are missing, system alerts and guides user for correction.

3.1.7 Add to Cart

As a buyer,
I want to add products or services to my cart,
So that I can buy or rent them easily.

Success

- User selects 'Add to Cart'.
- User fills required details (name, description, quantity, delivery method).
- User submits.
- System validates and confirms with success message.

Failure

- If required fields are missing, system alerts about missing information.

3.1.8 Chat

As a seller,
I want to chat with buyers inside the platform,
So that I can answer questions and close deals faster.

As a buyer,
I want to chat with sellers before renting an item,
So that I can clarify details and feel secure before payment.

Success

- Seller and buyer can initiate chat from item listing or booking page.
- System provides real-time messaging with history.
- Notifications are sent for new messages.
- Messages are stored securely.

Failure

- If chat service is down, system shows clear error message.
- If messages fail (network issues), system retries and notifies user.
- Unauthorized access to chat is blocked.

3.1.9 Discount

As a seller,
I want to provide discounts or promo codes,
So that I can attract more buyers.

As a buyer,
I want to apply discount codes,
So that I can save money.

Success

- Seller sets discount or promo code during listing.
- Buyer enters code at checkout.
- System validates code and recalculates price.
- Discount details appear in order summary.

Failure

- If seller sets discount incorrectly, system prompts correction.
- If buyer enters invalid/expired code, system shows error.
- If discount calculation fails, system prevents checkout until fixed.

3.1.10 Refund

As a buyer,
I want to request a refund,
So that I can get my money back for defective or undelivered items.

Success

- User navigates to "Refund Request".
- User selects product/service to refund.
- User provides details (order ID, reason, proof).
- User submits request.
- System validates and confirms submission.
- Seller/Platform is notified.

Failure

- If details are missing/invalid, system alerts and guides correction.
- If refund conditions not met, system rejects and notifies user.

3.1.11 Report

As a user (buyer or seller),
I want to report a product, service, or user,
So that I can flag inappropriate, fraudulent, or harmful activities.

Success

- User selects "Report" option.
- User chooses what to report (product, service, user).
- User provides reason and supporting details.
- User submits report.
- System validates and confirms submission.
- Admin/Moderation team is notified.

Failure

- If details/reason are missing, system alerts and guides correction.
- If report submission fails due to technical issues, system notifies to retry later

3.2 Non-Functional Requirements

Non-functional requirements define the **quality attributes** of the CampusLoop system that ensure usability, security, and performance.

3.2.1 Performance Requirements

- The system should respond to user actions (search, booking, messaging) within **2 seconds** under normal operating conditions.

- The platform should scale to handle a **20% increase in users** during peak times, such as exam weeks or event seasons, without major performance degradation.

3.2.2 Safety Requirements

- The system must implement measures to ensure **transaction integrity**. In case of sudden failures (server crash, app closure), ongoing transactions must not be lost or corrupted.

- User data must be backed up regularly to prevent loss during unexpected outages.

3.2.3 Security Requirements

- All communication between client and server must be encrypted using **TLS/SSL**.

- User accounts must support **multi-factor authentication** for increased security.
- Sensitive data (payment details, personal information) must be securely stored and comply with data protection regulations.

3.2.4 Usability Requirements

- The app must provide an **intuitive and user-friendly interface**, ensuring accessibility for students with different levels of technical expertise.
- The platform should be optimized for **mobile-first usage** with a clean design and responsive layouts.
- At least **90% of surveyed users** should report satisfaction with ease of use.

3.2.5 Reliability Requirements

- The system should maintain **99.9% uptime**, ensuring availability throughout academic and extracurricular cycles.
- Critical operations like payments, booking confirmations, and notifications should be protected against failure through redundancy and failover mechanisms.