

Software Requirements Specification

CampGround Site

Team Details

CS TY - Div 1, Batch: T3

- Khush Gandhi - 612303058
 - Hardik Mutha - 612303066
 - Suswan Joglekar - 612303074
 - Tanmay Karad - 612303083
-

Table of Contents

1. [Introduction](#)
 - [Purpose](#)
 - [Scope](#)
 - [Intended Audience and Document Conventions](#)
 - [Definitions, Acronyms, and Abbreviations](#)
 - [References](#)
 - [Overview](#)
2. [Overall Description](#)
 - [Product Perspective](#)
 - [Product Functions](#)
 - [User Characteristics](#)
 - [Constraints](#)
 - [Assumptions and Dependencies](#)
3. [Specific Requirements](#)
 - [Functional Requirements](#)
 - [Non-Functional Requirements](#)
 - [External Interface Requirements](#)
4. [Appendices](#)
 - [Glossary](#)
 - [Assumptions](#)
 - [References](#)
5. [Index](#)

Introduction

Purpose

The purpose of this Software Requirements Specification (SRS) is to define the complete functional and non-functional requirements of the **CampGround Booking Platform**, developed as part of the **Software Engineering** course.

The project demonstrates the **Software Development Life Cycle (SDLC)** using the **Waterfall Model**, emphasizing systematic requirement gathering, analysis, design, implementation, testing, and maintenance. This document acts as a formal reference for developers, evaluators, and testers throughout the project lifecycle.

Scope

The CampGround Platform is an online system that enables users to explore, book, and review verified campgrounds while allowing owners to list and manage them.

The platform aims to:

- Provide a **centralized system** for campground discovery and booking.
- Simplify **management of listings, approvals, and reviews**.
- Implement core software engineering concepts such as **modular design, system modeling, and requirement traceability**.
- Demonstrate the **application of SDLC phases** through planning, analysis, design, and development of a functional software system.

The final system includes modules for user authentication, campground listings, bookings, reviews, notifications, and administrative control.

Intended Audience and Document Conventions

Intended Audience:

- Faculty instructors and evaluators assessing software engineering methodology.
- Developers and project team members.
- Testers validating functional and non-functional requirements.

Conventions:

- **Shall** denotes a mandatory requirement.
- **May** denotes an optional or future requirement.
- Roles such as Admin, Owner, and User are capitalized throughout.

Definitions, Acronyms, and Abbreviations

Term	Definition
SDLC	Software Development Life Cycle — a structured approach that defines phases like requirement analysis, design, development, testing, and maintenance for systematic software creation.
UML	Unified Modeling Language — a standardized diagrammatic language used to model and visualize software system structures and behaviors.
CRUD	Create, Read, Update, Delete — the core operations used for handling persistent data within a system.
SRS	Software Requirements Specification — a document that describes the complete behavior and characteristics of a system before development.
API	Application Programming Interface — a set of rules and protocols enabling communication between software components.

References

1. Pressman, R. S. (2020). Software Engineering: A Practitioner's Approach (9th Edition). McGraw-Hill Education.
2. Sommerville, I. (2015). Software Engineering (10th Edition). Pearson Education.
3. IEEE Computer Society (1998). IEEE Std 830-1998: Recommended Practice for Software Requirements Specifications.
4. Tripathi, J. P., & Singh, A. (2021). "Design and Implementation of an Online Campground Reservation System." International Journal of Computer Applications, 174(32), 10–15.
5. W3C. (2024). Web Content Accessibility Guidelines (WCAG) 2.2. World Wide Web Consortium (W3C).

Overview

This document provides a structured specification of requirements for the CampGround Platform, including system functionalities, user roles, and design constraints.

The SRS is aligned with **Waterfall Model** principles, focusing on clear requirement definition before progressing to design and development.

Overall Description

Product Perspective

The CampGround Site is a **standalone web-based application** following a **3-tier architecture**:

1. **Presentation Layer:** Web interface accessible via browsers on desktops and mobile devices.
2. **Application Layer:** Server-side logic handling booking operations, approvals, and notifications.
3. **Data Layer:** Backend database managing campground, booking, and user information.

The system follows a **modular architecture** that supports separation of concerns, promoting maintainability and scalability.

Product Functions

The main functionalities of the system include:

1. **User Management:** Registration, login, and authentication.
2. **Campground Management:** Owners add, edit, and delete campground listings.
3. **Admin Approval:** Admin reviews and approves/rejects new listings.
4. **Booking Module:** Users book available campgrounds for specific dates.
5. **Review and Rating:** Campers share experiences after their stay.
6. **Notification Module:** Updates users and owners regarding booking and approval status.
7. **Search and Filter:** Allows searching by location, price, or amenities.

User Characteristics

User Type	Description	Skill Level
Admin	Manages system operations, reviews listings, and ensures policy compliance.	Advanced
Campground Owner	Manages own listings, views bookings, and responds to user feedback.	Intermediate
User/Camper	Browses, books, and reviews campgrounds with minimal technical knowledge.	Basic

Constraints

- The system shall be developed using the **Waterfall SDLC model**.
- The system must run on a web browser supporting modern web technologies.
- Real payment integration is not required; simulated transaction logic suffices.
- Response time for user interactions shall be under 2 seconds for optimal experience.

Assumptions and Dependencies

- The user has access to stable internet.
- Payment system simulated (not real).
- Browser-based access assumed (desktop/mobile).

- The server hosting environment supports web and API services.
- All project artifacts (UML diagrams, design documents) will align with SE methodologies.

Specific Requirements

Functional Requirements

ID	Module	Description
FR-01	User Management	The system shall allow user registration and authentication.
FR-02	Role Management	The system shall assign roles (Admin, Owner, User) upon registration.
FR-03	Campground Listing	Owners shall be able to add and modify campground details.
FR-04	Admin Approval	The Admin shall review and approve or reject listings.
FR-05	Booking System	Users shall be able to book available campgrounds for specified dates.
FR-06	Review System	Users shall rate and review campgrounds after booking.
FR-07	Notification System	The system shall notify users and owners of bookings or approvals.
FR-08	Search and Filter	The system shall provide filters for price, amenities, and location.
FR-09	Favorites	Users shall be able to mark and view favorite campgrounds.
FR-10	Report Generation	The system shall generate activity summaries for admin review.

Non-Functional Requirements

ID	Category	Requirement
NFR-01	Performance	The system shall respond to all user actions within 2 seconds.
NFR-02	Security	User data shall be stored securely, with role-based access control.
NFR-03	Usability	The interface shall be intuitive and easy to navigate.
NFR-04	Reliability	The system shall ensure availability of core modules during operation.
NFR-05	Maintainability	The architecture shall be modular for easy maintenance.
NFR-06	Portability	The application shall be deployable on any modern browser or OS.
NFR-07	Scalability	The system shall allow addition of new modules with minimal rework.

External Interface Requirements

Type	Description
User Interface	Web-based, responsive design with clear navigation for all user roles.
Hardware Interface	Compatible with standard computers and mobile devices.
Software Interface	Works with standard web browsers, server hosting, and data storage.
Communication Interface	HTTP/HTTPS for all client-server communication.

Appendices

Glossary

Term	Description
CampGround	The web-based software that allows users to discover, book, and manage campgrounds, while owners can list and maintain their sites.
Platform	
Stakeholder	Any individual or entity involved in or affected by the software, such as users, owners, administrators, or developers.
Use Case	A scenario that describes how a user interacts with the system to achieve a particular goal.
Requirement Engineering	The process of identifying, analyzing, documenting, and validating the software requirements.
Waterfall Model	A sequential SDLC model where each phase must be completed before moving to the next stage.

Assumptions

- Users have basic familiarity with online booking systems.
- All stakeholders agree on requirements before design and implementation phases.
- The development process adheres to Waterfall SDLC principles.

References

1. Pressman, R. S. (2020). Software Engineering: A Practitioner's Approach (9th Edition). McGraw-Hill Education.
2. Sommerville, I. (2015). Software Engineering (10th Edition). Pearson Education.
3. IEEE Computer Society (1998). IEEE Std 830-1998: Recommended Practice for Software Requirements Specifications.
4. Tripathi, J. P., & Singh, A. (2021). "Design and Implementation of an Online Campground Reservation System." International Journal of Computer Applications, 174(32), 10–15.
5. W3C. (2024). Web Content Accessibility Guidelines (WCAG) 2.2. World Wide Web Consortium (W3C).

Index

Keywords: Campground, Booking, Admin, Owner, Waterfall Model, SDLC, Requirements, Software Engineering, UML.