Software Requirements Specification

For

Room Radar

Version 1.0

Prepared by

Group Name: Team ULTRON

Sujit Subhash Mali	211101053	Sujitmali939@gmail.com
Manoj Pandharinath Chaudhari	211101041	Manoj53@gmail.com

Instructor:	Prof. Mrs. Pallavi Agrawal
Course:	B.Tech (CSE)
Lab Section:	CE-L5
Teaching Assistant:	Prof. Mrs. Pallavi Agrawal
Date:	3/9/2024

Contents

CONTENTS II

REVISIONS III

1 INTRODUCTION 1

1.1 DOCUMENT PURPOSE 1
1.2 PRODUCT SCOPE 1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW 1
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS 1
1.5 DOCUMENT CONVENTIONS 2
1.6 REFERENCES AND ACKNOWLEDGMENTS 2

2 OVERALL DESCRIPTION 3

2.1 PRODUCT OVERVIEW 3
2.2 PRODUCT FUNCTIONALITY 3
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS 4
2.4 ASSUMPTIONS AND DEPENDENCIES 4

3 SPECIFIC REQUIREMENTS 5

3.1 EXTERNAL INTERFACE REQUIREMENTS 5
3.2 FUNCTIONAL REQUIREMENTS 7
3.3 USE CASE MODEL 7

4 OTHER NON-FUNCTIONAL REQUIREMENTS 9

4.1 PERFORMANCE REQUIREMENTS 9
4.2 SAFETY AND SECURITY REQUIREMENTS 13
4.3 SOFTWARE QUALITY ATTRIBUTES 14

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	1. Sujit .S. Mali 2. Manoj .P. Chaudhari	"Roommate Matching" feature. This feature would allow users to find compatible roommates based on shared interests, habits, and preferences, enhancing the overall roommate selection process and for better living arrangements.	3/9/2024

1 Introduction

Welcome to Room Radar, the go-to Android app for students seeking vacant rooms effortlessly. Room Radar aims to simplify the accommodation search process, offering a seamless solution that connects students with available rooms. In this section, readers will discover an introduction to the Room Radar project, outlining its objectives and the value it brings to students.

1.1 Document Purpose

This Software Requirements Specification (SRS) document outlines the requirements for Room Radar, an Android application aimed at assisting students in finding vacant rooms. The purpose of this document is to provide a comprehensive understanding of the functionality, features, and specifications of Room Radar, ensuring clarity and alignment among stakeholders involved in the development process. Room Radar serves as a digital platform connecting students with available accommodations, streamlining the housing search experience. This SRS covers the entire scope of Room Radar, detailing the requirements for the entire system rather than specific subsystems. It serves as a guiding blueprint for developers, designers, testers, and other project stakeholders, facilitating effective communication and collaboration throughout the development lifecycle.

1.2 Product Scope

The key objectives of Room Radar include providing users with accurate and up-to-date information about vacant rooms, facilitating communication between students and landlords, and ultimately enhancing the overall experience of securing housing for students. With Room Radar, students can enjoy the benefits of a seamless and hassle-free housing search process, empowering them to focus more on their academic pursuits and less on the challenges of finding accommodation.

1.3 Intended Audience and Document Overview

The intended audience for this Software Requirements Specification (SRS) document includes developers, project managers, users, testers, and the client, who in this case is the individual or organization overseeing the development of Room Radar. Additionally, this document may also be of interest to professors or academic advisors involved in the project's evaluation or supervision.
The SRS contains detailed information about the requirements, functionalities, and specifications of Room Radar. It is organized into sections that address different aspects of the project, including an introduction, product scope, functional requirements, non-functional requirements, user interface design, and more.

1.4 Definitions, Acronyms and Abbreviations

Abbreviation/Acronym	Definition
SRS	Software Requirements Specification
UI	User Interface
API	Application Programming Interface

Abbreviation/Acronym	Definition
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
XML	Extensible Markup Language
CRUD	Create, Read, Update, Delete
MVC	Model-View-Controller
SQL	Structured Query Language

1.5 Document Conventions

Formatting Conventions:

Font: Throughout the document, Arial font size 11 is used for all textual content. This ensures consistency and readability in accordance with IEEE standards.

Typographical Conventions:

Section Titles: Section and subsection titles follow the IEEE template for consistency and ease of navigation.

Other Conventions:

Lists and Tables: Information is presented using lists and tables where appropriate to enhance organization and readability.

1.6 References and Acknowledgments

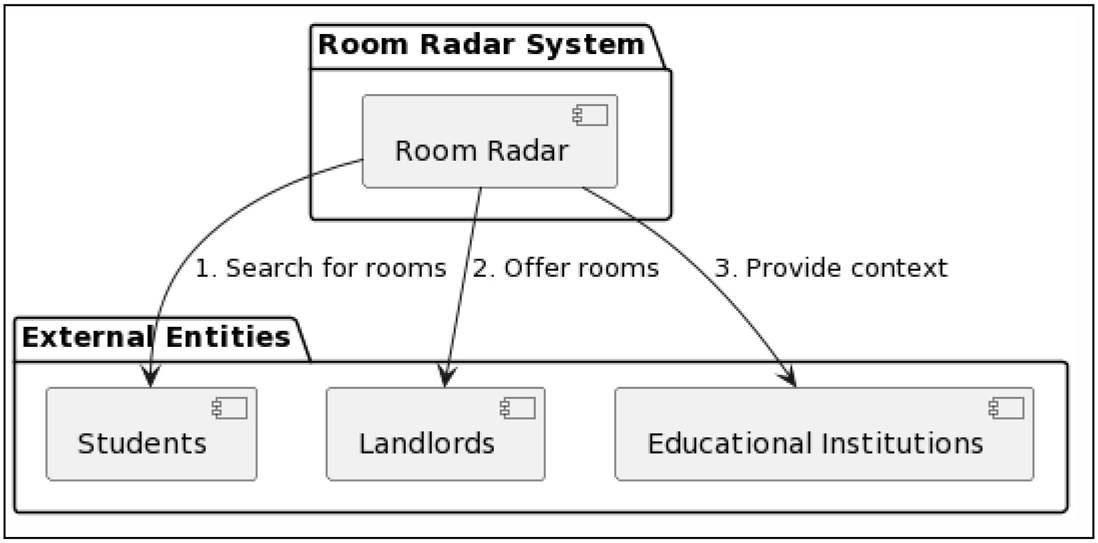
User Interface Style Guide: A document outlining the design principles, layout guidelines, and visual elements used in the user interface (UI) of Room Radar.

System Requirements Specifications: Detailed specifications outlining the functional and non-functional requirements of the Room Radar application, including system architecture, data storage requirements, and performance criteria.

2 Overall Description

2.1 Product Overview

Room Radar is a novel solution designed to address the challenges faced by students when searching for vacant rooms. Unlike existing systems that primarily focus on general housing listings, Room Radar is specifically tailored to the needs of students, providing a dedicated platform for finding accommodations near educational institutions. This app serves as a standalone product, offering a self-contained solution for students seeking housing options. It is not a replacement for existing systems but rather fills a gap in the market by providing a specialized and user-friendly interface for locating vacant rooms.



2.2 Product Functionality

The Room Radar Android application is designed to perform the following major functions:

Enable students to search for vacant rooms near their educational institutions.
Allow landlords to list and offer vacant rooms for rent.
Provide comprehensive information about available rooms, including location, rent, amenities, and contact details.
Facilitate communication between students and landlords for room inquiries and booking arrangements.
Allow students to filter search results based on preferences such as location, rent range, and room amenities.
Enable students to view and compare multiple room listings to make informed decisions.
Provide a user-friendly interface for easy navigation and interaction with the system.

2.3 Design and Implementation Constraints

Hardware Limitations: The app should be optimized to operate efficiently on a variety of Android devices, considering factors such as processing power, memory, and screen size.

Interfaces to Other Applications: Room Radar may need to integrate with external services or APIs for features such as mapping and location services. Compatibility with these interfaces must be ensured.

Specific Technologies and Tools: The development of Room Radar will utilize the COMET (Concurrent Object Modeling and Architectural Design) method for software design. Additionally, UML (Unified Modeling Language) will be employed for modeling and documenting the system's architecture and design.

2.4 Assumptions and Dependencies

Third-Party Components: The development of Room Radar assumes the availability and reliability of third-party components, such as mapping APIs and communication libraries. Any changes or disruptions to these components could affect the functionality and performance of the app.

Operating Environment: It is assumed that Room Radar will be deployed and operated in a stable and secure environment. Any issues or vulnerabilities in the operating environment could impact the app's reliability and security.

Data Availability: The functionality of Room Radar relies on the availability and accuracy of data sources such as room listings and user information. Assumptions are made regarding the availability and consistency of this data.

External Services: Room Radar may depend on external services for features such as push notifications or payment processing. Assumptions are made regarding the availability, reliability, and compatibility of these services.

Software Reuse: The project may depend on the reuse of software components or modules from other projects. Assumptions are made regarding the compatibility and integration of these components with Room Radar.

Regulatory Compliance: Assumptions are made regarding compliance with relevant regulations and standards, such as data protection laws and industry-specific regulations. Any changes in regulatory requirements could affect the development and operation of Room Radar.

User Behavior: Assumptions are made regarding user behavior and preferences, such as frequency of app usage, preferred search criteria, and willingness to share personal information. Any discrepancies between assumptions and actual user behavior could impact the app's effectiveness and user satisfaction.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

For the Room Radar project, the main user interface is the mobile app interface. However, if we were to consider an alternative example such as a thermostat interface, the following description can be provided:

The user interface of the thermostat consists of a digital display screen with touch-sensitive controls. Users interact with the thermostat by touching the screen to access menus and adjust settings such as temperature, mode (cooling, heating, or auto), fan speed, and scheduling. The interface may also include buttons for quick access to commonly used functions, such as setting the temperature to predefined comfort levels or activating energy-saving modes. Additionally, the thermostat interface may display relevant information such as current temperature, humidity levels, and system status indicators. Overall, the interface is designed to be intuitive and user-friendly, allowing users to easily control and monitor their heating and cooling systems.

3.1.2 Hardware Interfaces

For the Room Radar Android application, the hardware interfaces primarily involve communication with the mobile device's built-in hardware components. These interfaces include:

GPS Receiver: The app interacts with the device's GPS receiver to obtain the user's current location for location-based services, such as searching for rooms near the user's location.

Internet Connectivity: Room Radar requires an internet connection to access external services, retrieve room listings, and communicate with the server-side components. It interfaces with the device's networking hardware to establish and maintain internet connectivity.

3.1.3 Software Interfaces

For Room Radar, the connections between the mobile app and other software components primarily involve communication with backend servers and databases. The mobile app sends commands and requests to the backend server to perform various tasks, such as retrieving room listings, updating user preferences, and facilitating communication between students and landlords.

3.2.1 F1: User Registration and Authentication

The system shall allow users to register for an account by providing necessary information such as username, email, and password.
The system shall verify user credentials during registration to ensure uniqueness and validity.
The system shall provide authentication mechanisms, such as email verification or password-based login, to authenticate registered users.

3.2.2 F2: Room Search

The system shall allow users to search for vacant rooms based on various criteria such as location, rent range, and amenities.
The system shall display search results listing available rooms matching the user's search criteria.
The system shall provide filters to refine search results based on user preferences.

3.2.3 F3: Room Listing

The system shall allow landlords to list vacant rooms for rent by providing details such as location, rent amount, amenities, and contact information.
The system shall validate room listings to ensure completeness and accuracy of information provided by landlords.
The system shall enable landlords to manage and update their room listings as needed.

3.2.4 F4: Room Booking

The system shall facilitate communication between students and landlords for room inquiries and booking arrangements.
The system shall allow users to send booking requests to landlords for specific rooms.
The system shall notify users and landlords about booking requests and updates via email or in-app notifications.

3.2.5 F5: User Profile Management

The system shall allow users to manage their profiles by updating personal information, preferences, and account settings.
The system shall provide options for users to upload profile pictures and additional information to personalize their profiles.
The system shall ensure the security and privacy of user profile information.

3.2.6 F6: Messaging

The system shall provide messaging functionality for communication between students and landlords regarding room inquiries, booking arrangements, and other related matters.
The system shall maintain message history and conversations between users for reference and record-keeping purposes.
The system shall support real-time messaging and notifications to facilitate timely communication between users.

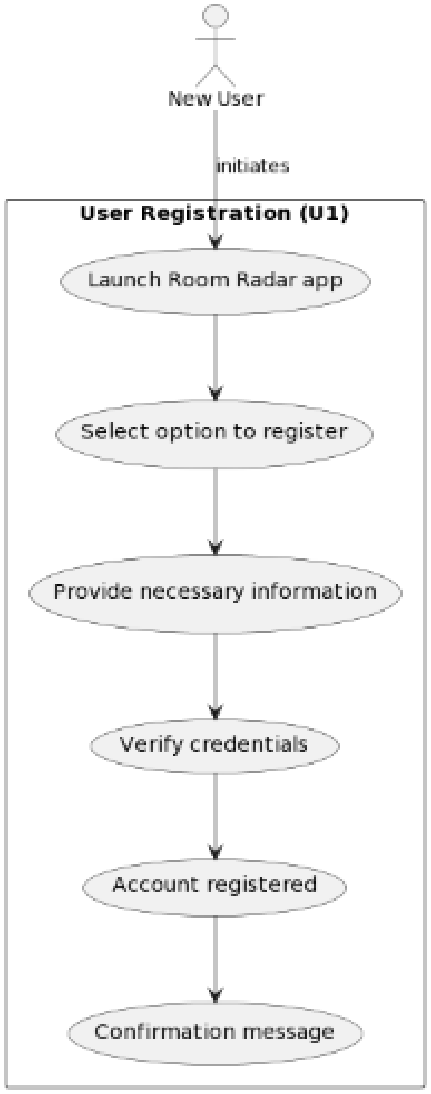
3.2.7 F7: Reviews and Ratings

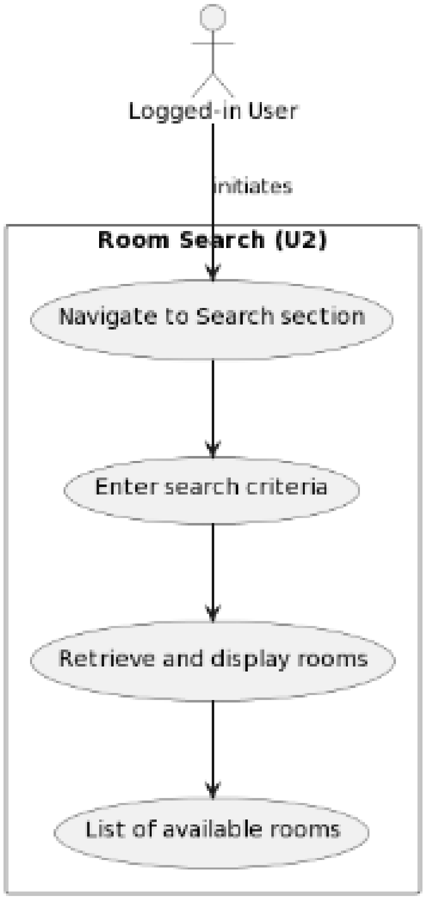
The system shall allow users to leave reviews and ratings for rooms and landlords based on their experiences.
The system shall display reviews and ratings on room listings to help users make informed decisions.
The system shall ensure the authenticity and integrity of reviews and ratings to maintain credibility and trustworthiness.

3.2 Use Case Model

3.3.1 Use Case: User Registration (U1)

Author: Sujit Subhash Mali
Purpose: The basic objective of this use case is to allow new users to register for an account on the Room Radar platform, enabling them to access the app's features and functionalities.
Requirements Traceability: F1 (User Registration and Authentication)
Priority: High
Preconditions: The user must have access to a compatible device with internet connectivity.
Postconditions: The user successfully registers for an account and gains access to the app's features.
Actors: New User
Flow of Events:





3.3.2 Use Case: Room Search (U2)

Author: Sujit Subhash Mali
Purpose: The basic objective of this use case is to allow users to search for vacant rooms based on various criteria such as location, rent range, and amenities.
Requirements Traceability: F2 (Room Search)
Priority: High
Preconditions: The user must be logged into the Room Radar app.
Post conditions: The user receives a list of available rooms matching the specified search criteria.
Actors: Logged-in User
Flow of Events:
Basic Flow:
The user navigates to the "Search" section of the Room Radar app.
The user enters search criteria such as location, rent range, and amenities.
The system retrieves and displays a list of available rooms matching the specified criteria.
Alternative Flow: None
Exceptions: None
Includes: None
Notes/Issues: None

4 Other Non-functional Requirements

4.1 Performance Requirements

P1. Response Time for Room Search:

- The system shall respond to user-initiated room searches within 2 seconds under normal load conditions.
    - Rationale: Quick response times are crucial for providing a seamless user experience and ensuring user satisfaction. Users expect instant feedback when searching for room listings, and a delay in response time may lead to frustration and abandonment of the search process.

P2. Messaging Delivery Time:

- The system shall deliver messages between users (students and landlords) within 5 seconds of being sent.
    - Rationale: Timely communication is essential for facilitating smooth interactions between users. Delays in message delivery may lead to misunderstandings or missed opportunities, impacting the user experience negatively.

P3. Server Uptime:

- The system shall maintain a minimum uptime of 99.9% per month, excluding scheduled maintenance windows.
    - Rationale: Continuous availability of the system is critical to ensure uninterrupted access for users. Downtime can disrupt users' ability to search for rooms, communicate with landlords, and perform other essential tasks, leading to dissatisfaction and potential loss of users.

P4. Concurrent User Handling:

- The system shall support a minimum of 1000 concurrent users without degradation in performance.
    - Rationale: As the user base grows, the system must be capable of handling increased traffic and maintaining responsiveness to ensure a positive user experience. Failure to handle concurrent users adequately can result in slow response times, timeouts, and system crashes.

P5. Data Retrieval Time:

- The system shall retrieve room listings and user information from the database within 1 second.
    - Rationale: Efficient data retrieval is essential for providing real-time updates and search results to users. Slow data retrieval times can lead to delays in displaying information, hindering the user experience and potentially discouraging users from using the app.