

Software Requirements Specification Template

January 2020

The following annotated template shall be used to complete the Software Requirements Specification (SRS).

Template Usage:

Text contained within angle brackets ('<', '>') shall be replaced by your project-specific information and/or details. For example, <Project Name> will be replaced with either 'Smart Home' or 'Sensor Network'.

Italicized text is included to briefly annotate the purpose of each section within this template. This text should not appear in the final version of your submitted SRS.

This cover page is not a part of the final template and should be removed before your SRS is submitted.

Acknowledgements:

Sections of this document are based on the IEEE Guide to Software Requirements Specification (ANSI/IEEE Std. 830-1984). The SRS templates of Dr. Orest Pilskalns (WSU, Vancouver) and Jack Hagemester (WSU, Pullman) have also been used as guides in developing this.

CSE Automation 1

Software Requirements Specification

2.0

30-05-2020

Dudipala Sathvikreddy (180003017)

Rapolu Pulakitha (180001041)

Shravya Ramasahayam (180001052)

Prepared for
Software Engineering
Spring 2020

Revision History

Date	Description	Author	Comments
<date>	<Version 1>	<Your Name>	<First Revision>

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	<Your Name>		

Table of Contents

REVISION HISTORY	III
DOCUMENT APPROVAL	III
1. INTRODUCTION	I
1.1 PURPOSE	I
1.2 SCOPE	I
1.3 DOCUMENT CONVENTIONS	II
1.4 REFERENCES	II
1.5 OVERVIEW	II
2. GENERAL DESCRIPTION	III
2.1 PRODUCT PERSPECTIVE	III
2.2 PRODUCT FUNCTIONS	III
2.3 USER CHARACTERISTICS	IV
2.4 GENERAL CONSTRAINTS	V
2.5 ASSUMPTIONS AND DEPENDENCIES	V
3. SPECIFIC REQUIREMENTS	V
3.1 EXTERNAL INTERFACE REQUIREMENTS	V
3.1.1 <i>User Interfaces</i>	v
3.1.2 <i>Hardware Interfaces</i>	v
3.2 FUNCTIONAL REQUIREMENTS	VI
3.2.1 <i>Login Page</i>	vi
3.2.2 <i>Application Form</i>	vi
3.2.3 <i>Page to view Previous Records</i>	vii
3.2.4 <i>Filters</i>	vii
3.2.5 <i>Cancellations</i>	vii
3.3 NON-FUNCTIONAL REQUIREMENTS	VIII
3.3.1 <i>Availability</i>	viii
3.3.2 <i>Security</i>	viii
3.3.3 <i>Maintainability</i>	viii
3.3.4 <i>Safety</i>	viii
3.3.5 <i>Usability</i>	viii
3.3.6 <i>Confidentiality</i>	viii
3.4 LOGICAL DATABASE REQUIREMENTS	VIII

1. Introduction

This SRS (Software Requirements Specification) provides a complete idea on how to design and implement the software as well as a basic idea of its usage and benefits. It provides a complete insight into the classes of users who will be using this software and how they will be benefited by it.

There are two subparts to this project.

1) *Automation of Room booking system:*

The existing procedure for managing leaves of the faculty at IIT Indore is manual and tedious. The faculty who apply for leave does not know the status of their leave and have to wait for long periods to know if the admin approved it or not. While applying for leave, one needs to have an idea of his previous leave records which becomes essential to abide by the institute's leave policy. However, searching for all of his previous records manually is more often than not results in errors. And hence there arises a need to have a proper system to manage these leave records. Thus automation of the leave management system is required.

2) *Automation of leave management system:*

IIT Indore has several classrooms and lecture halls accessible to its faculty and students. Booking of these rooms involves mailing the concerned authority and waiting for their response to confirm it. It is often time-consuming and is also difficult for the administrator to check the availability of rooms. In case the room is already booked, spontaneous change of plans has to be made. We thus need software that makes room booking simpler for its users as well as the administrator.

1.1 Purpose

The SRS (Software Requirements Specification) gives a comprehensive idea on the design, implementation and usage of the software. This SRS document provides a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. Any software developer can design and implement the software using this document. Also, regular users can get a basic understanding of its usage.

1.2 Scope

- 1) *ROOM BOOKING SYSTEM:* *The purpose of the room booking management system is to ease room booking and to create a convenient and user-friendly application for faculty to book rooms and cancel booked rooms online as well as managing the enormous database in an effective manner. The system is based on a relational database with its room management and booking functions. It is free, fast and reliable software. This project is a prototype of the website, and it is restricted within college premises. However, it can be further expanded to work on a larger scale.*

- a) The website provides information on the availability of all the classrooms at a given time slot on a given date.*
- b) Registered users can also make a booking as per their requirements.*

- c) *There is also a provision for cancelling a booking made by them (only for registered users).*
 - d) *Admin has the exclusive privilege of registering new users and adding new rooms.*
 - e) *Admin can also delete an existing user.*
- 2) **LEAVE MANAGEMENT SYSTEM:** *The purpose of the leave management system is to override the problems prevailing in practising manual system. This system is developed as per the requirements of IITI to carry out operations smoothly and effectively. This convenient and user-friendly application eases the process of applying for leave. Registered faculty can apply for leave when necessary.*
- a) *Admin approves and disapproves the leaves waiting for approval.*
 - b) *Admin also has a special privilege to view the leaves of all participants.*

1.3 Document Conventions:

<i>SRS:</i>	<i>Software Requirements Specifications</i>
<i>PHP:</i>	<i>Hypertext Preprocessor</i>
<i>HTML:</i>	<i>HyperText Markup Language</i>
<i>CSS:</i>	<i>Cascading Style Sheets</i>
<i>JS:</i>	<i>Java Script</i>
<i>DBMS:</i>	<i>Database Management System</i>
<i>SQL:</i>	<i>Structured Query Language</i>

1.4 References

- (1) <https://senior.ceng.metu.edu.tr/2014/such/documents/SRS.pdf> .
- (2) *IEEE STD 1233-1998, IEEE Guide for Developing System Requirements Specifications*
- (3) *IEEE STD 830-1998, IEEE Recommended Practice for Software Requirements Specifications*

1.5 Overview

The Second section of this document gives an outline of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification. The third section, Specific Requirements, is written primarily for the developers and describes in technical terms the details of the functionality of the product. Both

sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

2. General Description

2.1 Product Perspective

- 1) Room Booking System: The current manual system of room booking is quite vexing. The problem with this is the user is unaware of the availability status of the room; he wants to book. Contacting the admin to know the current status of his request for booking and altering plans as per it (i.e. changing time slots or room) is time consuming both for the admin and user.*
- 2) Leave Management System: The current system for applying leave is manual. It is challenging to apply for a leave manually and wait for its approval. Keeping track of all his leaves is not a simple task. But with this software, the user can view the record of his leaves.*

2.2 Product Functions

The application functions vary depending on whether the user is using it or the Admin.

- 1) Room Booking System: On opening the website, there appears a chart depicting all the classrooms in a building, corresponding time slots with their availability status. For a booked room, the purpose of booking can be viewed in addition to the details of the person who made the booking.*

To continue further, one has to sign in.

- a. User: On clicking the user login button, a sign in form is displayed. Once the user has signed in, he can perform the following:*
 - Book a room: By clicking on a particular time slot of a room, the user can book it if it is vacant. The date of a booking can be selected from the given calendar.*
 - Cancel a booking: By clicking on an already booked slot, the user finds an option to cancel the booking he has made.*
 - Log out: The user can end his session using the log out button.*
 - b. Admin: On clicking the user login button, a sign in form is displayed. Once the admin has signed in, he has the following options:*
 - Add a classroom: The admin can add a new class into the database.*
 - Add a new user: The admin can register a new user.*
 - Delete user: The admin can also delete an existing user.*
 - Log out: The admin can end his session (log out of his account) using this button.*
- 2) Leave Management System: As soon as the website is opened, two login options (one for the user and one for the admin) appear.*

- c. *User: On clicking the user login button, a sign in form is displayed. Once the user has signed in, the entire list of leaves, applied so far, is presented. The static bar to the left of the window provides the user with two options:*
- *Apply for leave: A form appears on clicking this button with auto-filled user ID. One can apply for leave by filling out the details of date, reason and type of leave. After the user applies for the leave, a message appears on-screen indicating that the leave is applied successfully.*
 - *My leaves: The entire list of User's leaves is displayed on clicking this button. These leaves can be further filtered by the date on which the leave was applied or status of the applied leave.*
 - *Log out: The admin can end his session (log out of his account) using this button.*
- d. *Admin: On clicking the admin login button, a sign in form is displayed. After the admin has signed in, the entire list of leaves, applied so far, is presented. Since navigation through this set of leaves is difficult, on the left, you have a static bar which helps to filter out certain leaves based on their status. This static bar has the following options:*
- *All leaves: Set as the default, the entire list of leaves, applied so far, is displayed.*
 - *Approved: The list of leaves that have been approved so far is displayed.*
 - *Disapproved: The list of leaves that have been disapproved so far is displayed.*
 - *Waiting for Approval: The list of leaves that have been applied, and are supposed to be approved or disapproved are displayed.*
- We also have two buttons in addition:*
- *Filter: The leaves of a particular page can be further filtered by the date on which the leave was applied or status of the applied leave.*
 - *Log out: The admin can end his session (log out of his account) using this button.*

Note: The list of leaves being displayed implies all the leaves along with their details such as the applicant, date, reason etc. are shown.

2.3 User Characteristics

1. User:

- *The user should have basic knowledge of using a website.*

2. Admin:

- *The admin should have necessary experience of using a website.*
- *The admin should be able to add, delete, and modify things in an existing database.*

2.4 General Constraints

- *The website is expected to perform in a real-time environment and is developed, keeping that in mind.*
- *This website is designed in such a way that it provides the necessary information to book a classroom (Room booking) and to apply for a leave (Leave Management).*
- *The website made does not include any payment or booking options.*
- *The website will function only when the system has a local area network.*

2.5 Assumptions and Dependencies

Server-side dependencies: My SQL Server, Php interpreter

Client-side dependencies: A web browser

Client-side assumption: Active internet connection.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

1) User Interfaces

- *Log-In Page: Room Booking System and Leave Management System- The system allows only authenticated users and admin to login. To use more functionalities, the individual must login.*
- *Application Form: Room booking System: A logged-in user can book a room by selecting a suitable slot of a given room. A logged-in admin can fill a simple form to add a new user, delete a new user or to add a new room.*
Leave Management System: A logged-in user can apply for leave
- *Page to view previous records: Room booking system: Any person visiting the website can view the current status of all the classrooms of the various building on this page.*
Leave Management System: A logged-in user can view his previous leave records, and an admin can see the previous leave records of all the registered users.
- *Filters: Room Booking System: Bookings can be filtered by the date of the event, which is by default the current date.*
Leave Management System: A logged-in user can filter the leave records by date of application and status of the leave. Similarly, an admin can filter the leave records likewise in addition to who has applied for the leave.
- *Cancellation: Room Booking System: A user can cancel his active bookings.*
Leave Management System: A logged-in user can cancel his leave application if there is any change in plan.

3.1.2 Hardware Interfaces

A device with a web browser

3.2 Functional Requirements

3.2.1 Login Page

Room Booking System and Leave Management System:

- *A user or an admin needs to fill a simple form in order to log in.*
- *User: A user has to enter his username and password in the form.*
- *Admin: An admin has to enter his adminid and password in the form*
- *The system checks if the credentials are valid or not.*
- *If the credentials are correct, then an alert message is displayed saying the login was successful.*
- *If the login fails Due to Invalid credentials, a warning is displayed stating "incorrect credentials" and is redirected to where the person has left.*

3.2.2 Application Form:

1) Room Booking System

User:

- *A logged-in user can book a room by selecting a suitable slot of a given room.*
- *When a free slot is selected, the user is prompted to fill the purpose of the booking in a form.*
- *The system adds the booking into the database and emails the user at his email the details of the booking.*
- *The user receives an alert message saying the booking was confirmed, and email was sent successfully.*
- *If a user selects a slot that's already booked by someone else, then the form is deactivated, and the details of the booking are displayed.*

Admin:

- *An admin can do three tasks after logging in:
Add a room, add a user, delete a user.*
- *To add a room, the admin has to fill a form containing the room number, name of the building and capacity of the room.*
- *To add a user, the admin has to fill a form containing the username, password and email of the new user.*
- *To delete a user, the admin has to fill the username of that user in a form.*
- *During any of the following, corresponding changes are made in the database.*
- *After a successful change, the admin is sent a confirmation message in an alert box.*

2) Leave Management System

User:

- *A logged-in user can apply for leave by filing a simple form.*
- *In the form, the user needs to fill the date and the reason for applying for the leave*
- *The system adds the leave proposal into the database, and its status is set to "waiting for approval".*
- *The user receives an alert message saying the application was successful.*

Admin:

- *An admin change status of leaves that are waiting for approval by selecting corresponding buttons.*
- *On clicking the change button, the admin is prompted with the approve and disprove buttons.*
- *Based on the action, the status of the leave is updated in the database.*

3.2.3 Page to view previous records:

1) Room Booking System

User and Admin:

- *A user or an admin can see the prior bookings of anyone in the table.*
- *When you click on a slot that has been booked, the details of his bookings can be viewed in a pop up that appears on the screen.*

2) Leave Management System

User: A logged-in user can view the records of his previous leaves and their status. This data is retrieved from the database.

Admin: An admin can view previous leave records of all the users and their status.

3.2.4 Filters:

1) Room Booking System

User and Admin:

- *One can filter the bookings of various rooms of different buildings by using the date filter. Once a date is selected, the table of bookings updates itself by retrieving data from the database.*

2) Leave Management System

User:

- *A logged-in user can filter his previous leave records by their status or the date of leave.*
- *If a new date is picked from the calendar or if a different status is chosen from the drop-down list, the table of leave records updates itself by retrieving data from the database.*

Admin:

- *An admin can filter previous leave records of various users by their status, the date of leave or the username.*
- *On clicking the status from the list, the table updates itself.*
- *The filter button, when clicked displays date and username filter options.*
- *The table of leave records updates itself with a change in the filter by retrieving data from the database.*

3.2.5 Cancellation:

1) Room Booking System

User:

- *A user can cancel his active bookings. When the user clicks on a slot he has booked, he can view the details along with the cancel button.*
- *On clicking the cancel button, the active booking is transferred into the expired bookings along with the date of cancellation (current date).*
- *An alert message is displayed confirming the cancellation.*

2) Leave Management System

User:

- *A logged-in user can cancel his leave if it is still under waiting for approval.*
- *On clicking the cancel button, the applied leave is deleted from the database.*
- *An alert message is displayed confirming the cancellation.*

3.3 Non-functional Requirements

Both projects have to be built on IITI cloud. So, most of the Non-functional requirements are in common.

3.3.1 Availability

- *The system is available 24/7. Admin has the complete authority to give accessibility to users.*
- *All the requirements have already been written in Section 3 and should be revisited for further clarity.*

3.3.2 Security

- *All the data is in encrypted form. Private Key is only known to the owner. So, hacking the data is not possible.*
- *The system is safeguarded against deliberate and intrusive faults from internal and external sources.*
- *The security concern is for the user account and admin account. So, the proper mechanism has been followed to avoid hacking.*
- *Passwords shall never be viewable at the point of entry or at any other time.*

3.3.3 Maintainability

- *Software is built such that it allows incorporating new requirements.*
- *The software is created by collaborating using a free and open-source distributed version control system, Git, and hence the contributions of various other programmers and thereby remains updated.*

3.3.4 Safety

- *There is no risk of data loss because of local system crash as all the data is automatically backed up on the IITI cloud.*

3.3.5 Usability

- *Software is user friendly, easy to learn and operate. The product shall be self-explanatory and intuitive.*

3.3.6 Confidentiality

- *The software system protects sensitive data and allows only authorized access to the data.*

3.4 Logical Database Requirements

Room Booking System: A database is required to store details of authenticated users, admins, rooms available for booking and record of expired and active room bookings.

Tables with attributes:

Admin (adminid, password)

Users (username, password, email)

Rooms (roomid, roomno, building, capacity)

Activebooking (dateofevent, dateofbooking, roomid, username, purpose, starttime, endtime)

Expiredbooking (dateofevent, dateofbooking, roomid, username, purpose, starttime, endtime, dateofcancellation)

the current day.

Leave Management System: A database is required to store details of authenticated users, admins and previous leave records of all the users.

Tables with attributes:

Admin (admin ID, user_name, password)

Users (user ID, password, user_name)

Leaves (date, User_ID, type, status, remarks, reason)