

# Software Requirements Specification

for

## Hall Management System

Shagor(363), Omar(364), Uosmoy(383), Nafiz(386), Babor(392)

*Jahangirnagar University, Savar*  
*Software Quality Assurance*  
*Requirements Engineering (RENG)*

8 December, 2023

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	System Overview . . . . .	4
1.2	Benefits of Using HMS . . . . .	4
1.3	Target Audience . . . . .	4
1.4	Documentation Structure . . . . .	4
1.5	Version Control and Updates . . . . .	5
1.6	Feedback and Support . . . . .	5
<b>2</b>	<b>Overall Description</b>	<b>5</b>
2.1	System Perspective . . . . .	5
2.1.1	System boundaries: . . . . .	5
2.2	Product Functions . . . . .	5
2.3	User Classes and Characteristics . . . . .	6
2.4	Operating Environment . . . . .	6
2.5	Design . . . . .	6
2.6	Implementation and Coding Standard . . . . .	6
<b>3</b>	<b>External Interface Requirements</b>	<b>7</b>
3.1	User Interfaces . . . . .	7
3.2	Hardware Interfaces . . . . .	7
3.3	Software Interfaces . . . . .	7
3.4	Communications Interfaces . . . . .	7
<b>4</b>	<b>System Features</b>	<b>7</b>
4.1	User Authentication (Login/Register) . . . . .	7
4.1.1	Description and Priority . . . . .	7
4.1.2	Stimulus/Response Sequences . . . . .	7
4.1.3	Functional Requirements . . . . .	7
4.2	Seat Search . . . . .	8
4.2.1	Description and Priority . . . . .	8
4.2.2	Stimulus/Response Sequences . . . . .	8
4.2.3	Functional Requirements . . . . .	8
4.3	Seat Booking . . . . .	8
4.3.1	Description and Priority . . . . .	8
4.3.2	Stimulus/Response Sequences . . . . .	8
4.3.3	Functional Requirements . . . . .	8
4.4	Profile Management . . . . .	8
4.4.1	Description and Priority . . . . .	8
4.4.2	Stimulus/Response Sequences . . . . .	9
4.4.3	Functional Requirements . . . . .	9
4.5	Feedback and Ratings . . . . .	9
4.5.1	Description and Priority . . . . .	9
4.5.2	Stimulus/Response Sequences . . . . .	9
4.5.3	Functional Requirements . . . . .	9

4.6	Seat Fee Payment . . . . .	9
4.6.1	Description and Priority . . . . .	9
4.6.2	Stimulus/Response Sequences . . . . .	10
4.6.3	Functional Requirements . . . . .	10
<b>5</b>	<b>Other non-functional Requirements</b>	<b>10</b>
5.1	Performance Requirements . . . . .	10
5.2	Safety Requirements . . . . .	10
5.3	Scalability . . . . .	11
5.4	Readability . . . . .	11
5.5	Security . . . . .	11
<b>A</b>	<b>Glossary</b>	<b>11</b>
<b>B</b>	<b>Analysis Model</b>	<b>11</b>
<b>C</b>	<b>Follow Up</b>	<b>11</b>

# 1 Introduction

This documentation serves as a comprehensive guide for the Hall Management System (HMS). It provides information on the system's functionalities, usage, and technical specifications for both administrators and users.

## 1.1 System Overview

The HMS is a software application designed to streamline the management of halls within an organization. It eliminates the need for manual processes and paperwork, offering a centralized platform for managing bookings, reservations, payments, and overall hall operations.

## 1.2 Benefits of Using HMS

- **Improved Efficiency:** Automates manual tasks, reduces administrative burden, and increases operational efficiency.
- **Enhanced Transparency:** Provides online access to hall information, bookings, and availability for easy reference and management.
- **Optimized Resource Utilization:** Facilitates efficient scheduling and allocation of halls, ensuring optimal utilization of resources.
- **Data-Driven Decision Making:** Generates accurate reports on hall usage and revenue, enabling informed decision-making and resource management.
- **Improved User Experience:** Offers a user-friendly interface for both administrators and users, simplifying interaction with the system.
- **Reduced Costs:** Saves time and resources by automating tasks and eliminating paper-based processes.

## 1.3 Target Audience

The HMS caters to two primary user groups:

- **Hall Administrators:** Manage bookings, reservations, payments, user accounts, and overall system operations.
- **Hall Users:** Individuals or groups who require booking halls for events, meetings, or other activities.

## 1.4 Documentation Structure

This documentation is organized into sections that provide comprehensive information about the HMS, including:

- **System Features:** Describes the functionalities available in the system.
- **User Guide:** Provides step-by-step instructions for using the system for administrators and users.
- **Technical Specifications:** Outlines the hardware and software requirements for running the system.
- **Data Security and Privacy:** Explains how user data is protected and secured within the system.
- **Troubleshooting:** Offers solutions to common problems encountered while using the system.
- **FAQ:** Answers frequently asked questions about the HMS.

## 1.5 Version Control and Updates

This documentation will be updated regularly to reflect any changes or improvements made to the HMS. All updates will be documented and communicated to users through appropriate channels.

## 1.6 Feedback and Support

We encourage users to provide feedback on the HMS documentation and suggest improvements. For any technical issues or support inquiries, please contact the system administrator. By following the information provided in this documentation, users can effectively utilize the HMS to manage hall bookings, reservations, and operations efficiently and seamlessly.

# 2 Overall Description

The overall description of a Hall Management System provides an understanding overview of the system. Generally, it includes the following key components:

## 2.1 System Perspective

### 2.1.1 System boundaries:

Clarifies if it's an independent system or part of a larger system. Also, define the system's interaction with a user and ecosystem.

## 2.2 Product Functions

- Presents an overview of what the system is desired to complete in general terms.

- Updates the current status of each seat in a hall whether it is booked or free.
- Save all information of staff and users of a hall.
- Facilitates all services associated with a hall.

## 2.3 User Classes and Characteristics

Provides a brief description of all classes, their tasks, and their interaction with the system.

- Administrators Class should provide all authentications and verifications.
- Staff and managers should perform their day-to-day responsibilities.
- There should be classes for technical support.

## 2.4 Operating Environment

Determines the environment in which the system will operate, including hardware, software, and network requirements.

- There should be maintained minimum software requirements
- There should be maintained cost-free hardware requirements
- An adaptation network will be established based on the ecosystem
- Security issues should be ensured.
- System environment should be more scalable according to user

## 2.5 Design

- Specifies the system's design, which should be user-friendly.
- Determine the database design.
- Design an user-friendly interface.

## 2.6 Implementation and Coding Standard

- The coding language such as Python, Java, C++, or any. Variables, functions, classes, and interface names should be meaningful. Coding indentation should be formatted.
- Should be precise.
- The naming convention should be meaningful.
- Logical error should be handled perfectly.

## **3 External Interface Requirements**

### **3.1 User Interfaces**

- Intuitive selection screens
- Review submission forms
- Secure login and registration screens

### **3.2 Hardware Interfaces**

Compatible with any device with internet access.

### **3.3 Software Interfaces**

Integrates with University database for real-time updates.

### **3.4 Communications Interfaces**

Uses secure HTTPS for communication.

## **4 System Features**

### **4.1 User Authentication (Login/Register)**

#### **4.1.1 Description and Priority**

Securely authenticates users for personalized experiences. Medium priority.

#### **4.1.2 Stimulus/Response Sequences**

- User selects login or registration.
- User provides credentials or registers with personal information.
- App verifies credentials or registers new users.
- Authenticated users gain access to personalized features.

#### **4.1.3 Functional Requirements**

1. User data must be securely stored and transmitted.
2. Users can recover passwords through email verification.
3. Authenticated users gain access to personalized features.

## **4.2 Seat Search**

### **4.2.1 Description and Priority**

User searches for available seats. High priority.

### **4.2.2 Stimulus/Response Sequences**

- User selects preferred hall, floor and accomodation type.
- App shows the user if any seat matching these criteria are available.

### **4.2.3 Functional Requirements**

1. The app must access university hall database.
2. The app must return available options that matches the required criteria.

## **4.3 Seat Booking**

### **4.3.1 Description and Priority**

Allows users to book seats in a hall. High priority.

### **4.3.2 Stimulus/Response Sequences**

- User selects a room/seat they would want to book.
- App prompts the user to give necessary information.
- User provides information.
- App books the selected seat for the user.

### **4.3.3 Functional Requirements**

1. App must access the database in real time to modify it.
2. Users must be prompted to input their data.

## **4.4 Profile Management**

### **4.4.1 Description and Priority**

User can view their own profile, update it as per necessary. Medium priority.



#### **4.4.2 Stimulus/Response Sequences**

- User selects a specific section of a profile they would like to modify.
- App prompts the user to provide the updated information.

#### **4.4.3 Functional Requirements**

1. The app must access the database in realtime and modify it accordingly.
2. The app must provide input forms for the user.

### **4.5 Feedback and Ratings**

#### **4.5.1 Description and Priority**

Allow students to provide feedback on the seats, facilities, and overall experience. Implement a rating system for seats or study areas. Medium priority.

#### **4.5.2 Stimulus/Response Sequences**

- User selects a hall/facility for review.
- App displays existing reviews and ratings.
- User submits a new review with an optional rating.
- Admin moderates the review before publication.

#### **4.5.3 Functional Requirements**

1. Users can submit reviews with optional ratings.
2. Reviews should include text comments.
3. Admins can moderate and remove inappropriate content.
4. The app should display the average rating for each hall.

### **4.6 Seat Fee Payment**

#### **4.6.1 Description and Priority**

The user can pay the monthly/yearly seat fee through the system.

#### 4.6.2 Stimulus/Response Sequences

- User selects the month/year
- App calculates the total fee.
- User provides payment details.
- App processes the payment.
- User receives a confirmation.

#### 4.6.3 Functional Requirements

1. The app must integrate with a secure payment gateway.
2. Users should receive information in their profile about payment confirmation.

## 5 Other non-functional Requirements

Non-functional requirements are aspects of a system that describe how it performs its functions rather than what functions it performs. In the context of a Hall Management System, various non-functional requirements may be considered to ensure the system meets certain criteria for performance, reliability, security, and other qualities. Here are some examples:

### 5.1 Performance Requirements

- Response Time: Define the maximum acceptable response time for different operations within the system, such as retrieving student information or generating reports.
- Throughput: Specify the number of transactions the system should handle within a given time frame.

### 5.2 Safety Requirements

- Shutdown Database: The emergency halt feature must respond within 2 seconds.
- Password: The student database will be password-protected.

### 5.3 Scalability

- Load Handling: Describe how the system should behave under varying loads, and specify the maximum number of concurrent users it should support.
- Database Scalability: Outline how the system's database should scale as the amount of data increases.

### 5.4 Readability

- Availability: Specify the percentage of time the system should be available for use.
- Fault Tolerance: Describe how the system should handle errors or failures, ensuring minimal downtime and data loss.

### 5.5 Security

- Authentication: Define the authentication mechanisms used to ensure that only authorized users can access the system.
- Authorization: Specify the levels of access and permissions for different user roles.
- Data Encryption: Define encryption standards for sensitive data in transit and at rest.

## A Glossary

- HMS: Hall Management System
- HTTPS: Hyper Text Transfer Protocol

## B Analysis Model

- Use Case Diagram
- Sequence Diagram
- Flow Diagram
- Class Diagram

## C Follow Up

Changes will be made as per necessary throughout the whole development process.