
Software Requirements Specification

for

CHS – The Complete Hostel Solutions

Prepared by

Batch: B (Comps B)

Manish Jadhav
Mayur Solankar

2023301005
2023301018

manish.jadhav23@spit.ac.in
mayur.solankar23@spit.ac.in

Instructor: Dr. Prasenjit Bhavathankar

Course: Software Engineering

Date: 26-08-2024

Contents

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	1
1.6 REFERENCES AND ACKNOWLEDGMENTS.....	2
2 OVERALL DESCRIPTION	3
2.1 PRODUCT PERSPECTIVE	3
2.2 PRODUCT FUNCTIONALITY	3
2.3 USERS AND CHARACTERISTICS.....	3
2.4 OPERATING ENVIRONMENT	3
2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS.....	4
2.6 USER DOCUMENTATION	4
2.7 ASSUMPTIONS AND DEPENDENCIES	4
3 SPECIFIC REQUIREMENTS	5
3.1 EXTERNAL INTERFACE REQUIREMENTS	5
3.2 FUNCTIONAL REQUIREMENTS	6
3.3 BEHAVIOUR REQUIREMENTS	6
4 OTHER NON-FUNCTIONAL REQUIREMENTS	7
4.1 PERFORMANCE REQUIREMENTS	7
4.2 SAFETY AND SECURITY REQUIREMENTS.....	7
4.3 SOFTWARE QUALITY ATTRIBUTES.....	7
5 OTHER REQUIREMENTS	8
APPENDIX A – DATA DICTIONARY	9
Appendix B - Group Log	10

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Manish Jadhav, Mayur Solankar	Our Hostel Management System (HMS) project is a user-friendly and feature-rich software solution to meet the specific needs of hostel administrators, staff, and residents. Whether managing a student hostel, our HMS offers a robust platform to streamline operations, improve communication, provide shared accommodation facility, and enhance security.	26-08-2024

Introduction

1.1 Document Purpose

The purpose of this document is to define the software requirements for the development of a web-based hostel management system, named "CHS." CHS aims to provide students with a platform to make billing efficiently, check for available rooms and daily check-in & check-out process.

1.2 Product Scope

CHS will allow users to perform the following tasks:

1. User authentication and access control.
2. Managing room occupancy and vacancies.
3. Managing payment records, including due dates and late fees.
4. Ensuring the security of personal and financial data.

1.3 Intended Audience and Document Overview

The intended audience for the Hotel Management System (HMS) includes Hostel administrators, Students and residents, Staff members and Parents. Additionally, developers and technical support staff involved in the system's setup, maintenance, and security of the HMS are part of the target audience.

1.4 Definitions, Acronyms and Abbreviations

HMS - Hostel Management System

SIS - Student Information System

GUI - Graphical User Interface

UI - User Interface

DFD – Data Flow Diagram

API - Application Programming Interface

IEEE – Institute of Electrical and Electronics Engineers

1.5 Document Conventions

The document follows IEEE formatting standards, using Times New Roman font size 11 or 12 throughout for text. Italics are used for comments. The text is single-spaced with 1” margins. Section and subsection titles adhere to the provided template. Any specific standards or typographical conventions are detailed, including formatting, naming conventions, and other special conventions used within the document.

1.6 References and Acknowledgments

IEEE standard -830-1998 ,Pankaj Jalote Software Engineering book.

Overall Description

2.1 Product Perspective

CHS, our Hostel Management System, is a pivotal tool within the educational institution's administrative framework. Operating within the context of the organization, it plays a central role in managing hostel facilities, enhancing efficiency, and improving the experience for administrators, students, and staff. CHS interacts with key internal entities, including administrators responsible for hostel operations and students who utilize the system for room bookings, payments, and maintenance requests.

2.2 Product Functionality

1. User Registration and Authentication

- 1.1. Allow users to register for accounts.
- 1.2. Authenticate users securely for system access.

2. Room Allocation and Management:

- 2.1. Administrators can assign rooms to students and manage room allocations efficiently.
- 2.2. Room details, including type, occupancy status, and maintenance history, are easily accessible.

3. Billing and Payments:

- 3.1. Process payments securely through integrated payment gateways, supporting various payment methods.

4. Check-In and Check-Out::

- 4.1. Streamline the check-in and check-out processes for students, ensuring accurate records and minimal manual effort.

5. User Profile Management:

- 5.1. Users can update their profiles, including contact information, preferences, and emergency contacts, ensuring up-to-date records.

6. Security and Data Protection:

- 6.1. Implement robust security measures, including encryption, secure authentication, and audit trails, to protect user data and privacy.

7. Role-Specific Dashboards:

7.1. Offer customized dashboards for administrators and students, displaying relevant information and tasks based on their roles.

8. Reporting and Analytics:

8.1. Access historical data to identify trends and optimize hostel operations.

2.3 Users and Characteristics

1. Hostel Administrators:

- **Pertinent Characteristics:**
 - Authorized Personnel: Trained staff members responsible for managing hostel operations.
 - Access to System: Have administrative privileges allowing them to perform a wide range of tasks.
 - Familiarity with Hostel Policies: Knowledgeable about the institution's hostel policies and procedures.
- **Importance:**
 - Hostel administrators are key users responsible for configuring and maintaining the system, making their role critical to the system's proper functioning.

2. Students:

- **Pertinent Characteristics:**
 - Residents: Individuals residing in the hostel facilities.
 - Varied Technical Proficiency: Students with varying levels of technical expertise and familiarity with the system.
 - Need for Accessibility: May require ease of use and accessibility features to accommodate diverse needs
- **Importance:**
 - Students are primary end-users who rely on the system for room bookings, maintenance requests, and payments.

3. Student Information System (SIS) Users:

- **Pertinent Characteristics:**
 - Academic Staff: Personnel managing student academic records and enrollment.

- Data Accuracy: Require accurate and timely synchronization of student data.
- **Importance:**
 - SIS users play a crucial role in maintaining the accuracy of student information, which is essential for room allocation and billing within CHS.

4. Developers and IT Support:

- **Pertinent Characteristics:**
 - Technical Experts: IT professionals responsible for system setup, maintenance, and security.
 - Security Focus: Prioritize system security, data protection, and performance.
 - Access Control: Have privileged access to system configuration and security settings.
- **Importance:**
 - They are responsible for system maintenance, updates, backups, and monitoring to prevent downtime and security breaches.

2.4 Operating Environment

The operating environment for "CHS," the Hostel Management System (HMS) project, encompasses a robust hardware and software infrastructure. It relies on a server infrastructure, responsive user interfaces for various devices, and a stack of software components, including operating systems, web servers, and a relational database system. Security measures include firewalls, intrusion detection, and data encryption. Networking components ensure reliable access, and integration with external systems like the Student Information System (SIS) and payment gateways is vital. Scalability, redundancy, and data backup mechanisms are in place, forming a dependable foundation for efficient hostel management within an educational institution.

2.5 Design and Implementation Constraints

1. **Budget Constraints:**
 - The project may have budget limitations that impact the choice of technology stack, hardware, and the extent of custom development
2. **Time Constraints:**
 - There may be strict project timelines, such as academic semesters or regulatory deadlines, which could affect the development and deployment schedule.
3. **Technology Stack Compatibility:**
 - The chosen technology stack for CHS must be compatible with existing systems and infrastructure within the educational institution.

4. Integration Challenges:

- Integration with external systems, such as the Student Information System (SIS) and payment gateways, may present technical challenges and require coordination with third-party providers

5. Resource Constraints:

- Resource limitations, such as server capacity and bandwidth, could impact system performance, particularly during peak usage periods.

6. User Accessibility:

- Ensuring accessibility for users with disabilities (compliance with WCAG standards) may influence the system's design and user interface.

2.6 User Documentation

For the CHS Hostel Management System (HMS), comprehensive user manuals and help resources are essential to ensure that users can effectively utilize its features and capabilities. The following types of manuals and help materials are needed:

1. **User Guides:** User guides will provide instructions on various aspects of the HMS, such as booking available room, billing transaction. These guides should be accessible from within the HMS interface and cover both basic and advanced features.
2. **Administrator Manuals:** Separate manuals tailored to administrators should detail system setup, configuration, security management, and advanced customization options. These manuals will guide administrators in effectively managing and maintaining the HMS.
3. **Frequently Asked Questions (FAQs):** A comprehensive FAQ section should be available to address common queries and troubleshoot common issues. This resource can help users find quick solutions to their problems.
4. **Online Support and Community Forums:** An online support portal and community forums will enable users to seek help from support staff and interact with other users for insights and solutions to specific challenges.

2.7 Assumptions and Dependencies

2.7.1 Assumptions:

- **Compliance with Data Protection Regulations:** The assumption is made that the hostel management will take necessary steps to ensure that the HMS complies with data protection and privacy regulations applicable in their region. Failure to comply could lead to legal and operational issues.

- **Hardware Availability:**

It is assumed that the hostels will have access to necessary hardware and infrastructure, including servers or cloud hosting, to support the HMS software. Availability of hardware can impact system performance and scalability.

2.7.2 Dependencies:

- **Data Backup and Recovery Services:** Dependency exists on external data backup and recovery services to ensure data integrity and availability. Regular backups and reliable recovery mechanisms are crucial.

Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

1. Login Page:

- The login page is the entry point for users. It includes fields for entering usernames and passwords. Users select their role (administrator or student) and access the system.

2. Dashboard:

- The dashboard is the central hub where users can access various features and information. It typically includes widgets and panels displaying room occupancy, billing information, maintenance requests, and important notifications.

3. Room Management Interface:

- Administrators use this interface to manage rooms within the hostel. It includes features for adding, editing, or deleting rooms, viewing room details, and assigning rooms to students.

4. Billing and Payment Interface:

- In this section, users can view their bills, make payments, and set up payment methods. It displays billing details, payment history, and due dates.

5. Profile Management Interface:

- Users can update their profiles through this interface, including personal information, contact details, and emergency contacts. It may also allow for password changes

6. Room Reservation and Booking Interface:

- Students can reserve rooms in advance for future semesters or academic years. This interface displays room availability and allows users to select and book rooms.

3.1.2 Hardware Interfaces**1. Server Hardware:**

- CHS requires dedicated server hardware to host the application and the database. The server hardware includes:
- Central Processing Unit (CPU): A multi-core processor to handle concurrent user requests.
- Random Access Memory (RAM): Sufficient RAM to ensure smooth system performance.
- Storage: Adequate disk space for storing application files and database data.
- Network Interface: A network interface card (NIC) for network connectivity.

2. Client Devices:

- Users access CHS through various client devices, including:
- Desktop Computers
- Laptops
- Tablets
- Smartphones
- These devices should have standard hardware components, including input devices (keyboard, mouse, touchscreen), display screens, and network connectivity.

3. Networking Hardware:

- Networking hardware components are essential for communication between the server and client devices, as well as for internet connectivity. These components include:
- Routers: To manage network traffic within the institution.
- Switches: For local network connections.
- Firewalls: To protect the network and server from unauthorized access.
- Internet Modems: To provide external internet access.

3.1.3 Software Interfaces**1. Student Information System (SIS) Integration:**

- CHS must interface with the institution's SIS to synchronize student data, including personal details, enrollment status, and room assignments. This integration ensures accuracy and consistency of data between systems.

2. Payment Gateway Integration:

- To facilitate secure financial transactions, CHS integrates with payment gateway services. This allows students to make payments for accommodation, meal plans, and other charges within the system.

3. Database Management System (DBMS):

- CHS relies on a DBMS (e.g., MySQL, MongoDB) to store and manage data efficiently. The system communicates with the DBMS through SQL queries and commands.

4. Web Server:

- A web server (e.g., Apache, Nginx) serves as an interface between the application's frontend and the server's backend. It handles HTTP requests and responses, routing user interactions to the appropriate components.

5. Operating System (OS):

- The software interfaces with the server's operating system (e.g., Linux, Windows Server) to manage system resources, security, and permissions.

6. Third-Party Libraries and APIs:

- CHS may use third-party libraries and APIs for various functionalities, such as charting libraries for generating reports or geolocation APIs for mapping and location-based services.

7. Web Browsers::

- Users access CHS through web browsers (e.g., Chrome, Firefox, Safari). The system must be compatible with various browser versions and handle browser-specific behaviors and features.

3.1.4 Communications Interfaces

- **Web-based User Interface (UI):**

CHS's primary communication interface is its web-based user interface, accessible through web browsers. Users interact with the system by sending HTTP requests and receiving responses over the internet

- **APIs:**

CHS provides a set of APIs that allow external systems or applications to programmatically interact with the system. These APIs can be RESTful, using HTTP as the communication protocol, or use other communication protocols (e.g., GraphQL).

- **HTTP Communication:**

CHS utilizes HTTP (Hypertext Transfer Protocol) for communication between the client (web browsers) and the web server. This protocol defines how requests and responses are formatted and transmitted, ensuring the exchange of data between the user interface and the application server.

3.2 Functional Requirements**1. User Registration:****a. Process:**

- **Input:**

- Full Name
- Email Address
- Password

- **Output:**

- Successful account creation
- Error messages if any validation fails
- Verification email sent to the user

- **Description:**

- Users can register by providing their full name, email address, and password. The system validates the input to ensure accuracy (e.g., proper email format, strong password). Upon successful registration, a verification email is sent to the provided email address. The user must verify their email to activate the account.

b. Validation:

- **Input:**

- User-provided full name, email address, and password.

- **Output:**

- Valid input leads to successful registration.
- Invalid input results in error messages specifying the issues (e.g., incorrect email format, weak password).

- **Description:**

- The system validates the registration data to ensure correctness and security. Clear error messages guide users in correcting any mistakes.

c. Email Verification:

- **Input:**

- User's email address.

- **Output:**

- Verification email containing a link.

- **Description:**

- After registration, the system sends a verification email to the user. The user must click the link to activate their account.

2. Room Management:**a. Add, Edit, or Delete Rooms:**

- **Input:**

- Room details (type, availability, price, etc.).

- **Output:**

- Updated room inventory reflecting additions, modifications, or deletions.

- **Description:**

- Administrators can manage room inventory by adding new rooms, editing details of existing rooms, or deleting rooms from the system.

b. Categorize Rooms by Type:

Input:

- Room type (single, double, suite, etc.).

- **Output:**

- Categorized list of rooms.

- **Description:**

- Administrators can categorize rooms based on their type for easy management and searchability.

c. Assign and Change Room Occupants:

- **Input:**

- Occupant details (name, check-in date, check-out date).

- **Output:**

- Updated room assignment with current occupant details.

- **Description:**

- Administrators can assign rooms to occupants and manage check-in and check-out processes, updating the room's status accordingly.

3. Billing and Payments:

a. Generate Bills:

- **Input:**

- Room occupancy details, meal plan selections, and any additional charges.

- **Output:**

- Detailed bill for the student, including accommodation, meal plans, and other charges.

- **Description:**

- The system generates bills based on the student's room occupancy, selected meal plans, and any other services used.

b. View and Pay Bills Online:

- **Input:**

- Student account access.

- **Output:**

- Display of current and past bills, payment options.
- **Description:**
 - Students can view their bills online and choose from multiple payment methods (credit/debit cards, online banking) to pay their dues.
- c. **Payment Reminders and Notifications:**
 - **Input:**
 - Payment due dates, payment status.
 - **Output:**
 - Automated reminders and notifications for upcoming payments or overdue bills.
 - **Description:**
 - The system automatically sends reminders and notifications to students to ensure timely payments.

3.3 Behaviour Requirements

3.3.1 Use Case View

1. User Registration and Login

- **Actors:** Students, Administrators
- **Overview:** Students and administrators can register new accounts or log in using their credentials.

2. Room Management

- **Actors:** Administrators
- **Overview:** Administrators can perform actions related to room management, including adding, editing, and deleting rooms, as well as assigning rooms to students

3. Billing and Payments

- **Actors:** Students
- **Overview:** Students can view bills, make payments, and set up payment methods. The system generates bills and sends payment reminders.

4. Reporting and Analytics

- **Actors:** Administrators
- **Overview:** Administrators can generate reports on room occupancy, financials, and maintenance history.

5. Check-In and Check-Out

- **Actors:** Students, Administrators

- **Overview:** Students complete the check-in and check-out processes, including room inspection and key return (if applicable). Administrators oversee the process.

6. Integration with SIS and Payment Gateway

- **Actors:** External Systems (SIS, Payment Gateway)
- **Overview:** CHS integrates seamlessly with the institution's Student Information System (SIS) for data synchronization and with a payment gateway for secure financial transactions.

Other Non-functional Requirements

4.1 Performance Requirements

1. Response Time:

- Any user action, such as loading a page, performing a transaction, or saving a record, should have a response time of less than 2 seconds, under normal system load conditions.

2. Concurrent User Support:

users) without significant performance degradation.

3. Database Performance::

- Database queries and transactions should be optimized for speed and efficiency to prevent bottlenecks.

4. API Response Time:

- Any API requests made to the HMS should have an average response time of less than 100 milliseconds, allowing for quick integration with third-party applications and tools.

5. Scalability:

- The system should be scalable to accommodate an increasing number of users and hostels without a significant drop in performance.

4.2 Safety and Security Requirements

1. Safety Requirements for Hostel Management System (HMS):

1.1.Data Backup and Disaster Recovery:

- Maintain regular data backups and establish a disaster recovery plan to ensure data integrity and availability in case of system failures or disasters.

1.2.Physical Access Control:

- Restrict physical access to server rooms and data centers to authorized personnel only to prevent tampering or unauthorized access.

1.2.Electrical Safety:

- Ensure that electrical systems and equipment used in the hostel, including servers and networking equipment, meet safety standards and are regularly inspected for safety.

2. Security Requirements for Hostel Management System (HMS):

2.1. Data Encryption: The HMS must use strong encryption protocols (e.g., SSL/TLS) to secure data transmission and storage, including user login credentials and sensitive content.

2.2. User Authentication:

- Users must authenticate their identity using secure and industry-standard methods (e.g., username/password, multi-factor authentication) before accessing the system.
- Passwords must adhere to complexity and length requirements, and password policies should be customizable by administrators.

2.3. Access Control:

- Prefer role-based access control (RBAC) to restrict user access to specific functionalities and data.

2.4. Secure API Access:

- Any APIs provided by the HMS must require authentication and authorization, and they should follow industry best practices for security.

2.4. Database Security:

- Secure the database with appropriate access controls and encryption for sensitive data.

4.3 Software Quality Attributes

1. Usability:

- The system should be user-friendly and intuitive, making it easy for both administrators and residents to navigate and perform tasks

2. Reliability:

- The HMS should be highly reliable, minimizing system downtime and data loss.

3. Performance::

- Ensure the system's speed and responsiveness meet user expectations, with fast loading times for pages and quick response to user actions

4. Maintainability::

- Develop clean and modular code to simplify maintenance and future enhancements.

Appendix A – Data Dictionary

1. Hostel Information (Table: hostels)

hostel_id (Primary Key): Unique identifier for each hostel.

name: Name of the hostel (e.g., "North Wing Hostel").

location: Location or address of the hostel.

capacity: Total number of rooms available in the hostel.

occupancy: Current number of occupied rooms.

warden_id (Foreign Key): ID of the hostel warden responsible for this hostel.

description: Brief description or additional information about the hostel.

2. Room Information (Table: rooms)

room_id (Primary Key): Unique identifier for each room.

hostel_id (Foreign Key): ID of the hostel to which the room belongs.

room_number: Room number or identifier within the hostel.

occupant_id (Foreign Key): ID of the student currently occupying the room (if applicable).

status: Current status of the room (e.g., "Occupied," "Vacant," "Under Maintenance").

room_type: Type of room (e.g., "Single," "Double," "Suite").
floor: Floor on which the room is located.
description: Additional information about the room.

3. Student Information (Table: students)

student_id (Primary Key): Unique identifier for each student.
first_name: Student's first name.
last_name: Student's last name.
email: Student's email address.
contact_number: Student's contact number.
gender: Student's gender (e.g., "Male," "Female," "Other").
date_of_birth: Student's date of birth.
nationality: Student's nationality.
room_id (Foreign Key): ID of the room assigned to the student (if applicable).
check_in_date: Date when the student checked into the hostel.
check_out_date: Date when the student checked out of the hostel (if applicable).

4. Billing Information (Table: billing)

billing_id (Primary Key): Unique identifier for each billing transaction.
student_id (Foreign Key): ID of the student associated with the billing transaction.
billing_date: Date of the billing transaction.
description: Description of the billing item (e.g., "Accommodation Fee," "Meal Plan").
amount: Amount charged for the billing item.
payment_status: Payment status of the billing item (e.g., "Paid," "Unpaid," "Late Payment").

