

# Software Requirements Specification

## Smart Hostel Management System



Prepared for:  
Ms. Nida Sultan Nahra  
Hostel Warden, Namal University

Prepared by:

Muhammad Ahmad	NUM-BSCS-2024-44
Asad Ullah Khan	NUM-BSCS-2024-17
Maryam Rashid	NUM-BSCS-2024-33

Department of Computer Science  
Namal University, Mianwali

December 28, 2025

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Scope . . . . .	3
1.2.1	System Name and Components . . . . .	4
1.2.2	What the Software Will Do . . . . .	4
1.2.3	Benefits and Objectives . . . . .	5
1.2.4	What the Software Will NOT Do . . . . .	6
1.3	Definitions, Acronyms, and Abbreviations . . . . .	7
1.4	References . . . . .	8
1.5	Overview . . . . .	9
1.5.1	Document Structure . . . . .	9
1.5.2	Organization of the Document . . . . .	9
<b>2</b>	<b>General Description</b>	<b>10</b>
2.1	Product Perspective . . . . .	10
2.1.1	Background and Current System Limitations . . . . .	10
2.1.2	System Interfaces . . . . .	10
2.1.3	User Interfaces . . . . .	10
2.1.4	Hardware Interfaces . . . . .	10
2.1.5	Software Interfaces . . . . .	10
2.1.6	Communications Interfaces . . . . .	11
2.1.7	System Context . . . . .	11
2.2	Product Functions . . . . .	11
2.2.1	Attendance Management . . . . .	11
2.2.2	Biometric Entry and Exit Tracking . . . . .	11
2.2.3	Room Management . . . . .	11
2.2.4	Complaint Management . . . . .	11
2.2.5	Day Scholar Tracking . . . . .	11
2.3	User Characteristics . . . . .	11
2.3.1	Hostel Warden . . . . .	12
2.3.2	Assistant Warden . . . . .	12
2.3.3	Resident Students . . . . .	12
2.3.4	Security Personnel . . . . .	12
2.4	General Constraints . . . . .	12
2.4.1	Regulatory and Policy Constraints . . . . .	13
2.4.2	Hardware Limitations . . . . .	13
2.4.3	Interface Requirements . . . . .	13
2.4.4	Development Constraints . . . . .	13
2.4.5	Operational Constraints . . . . .	14
2.4.6	Security and Safety Constraints . . . . .	14
2.5	Assumptions and Dependencies . . . . .	14
2.5.1	Assumptions . . . . .	14
2.5.2	Dependencies . . . . .	15
<b>3</b>	<b>Specific Requirements</b>	<b>15</b>
3.1	External Interfaces . . . . .	15
3.1.1	User Interfaces . . . . .	15

3.1.2	Hardware Interfaces . . . . .	15
3.1.3	Software Interfaces . . . . .	15
3.1.4	Communications Interfaces . . . . .	15
3.2	Attendance Management Module . . . . .	16
3.2.1	Functional Requirements . . . . .	16
3.2.2	Performance Requirements . . . . .	18
3.3	Biometric Entry and Exit Tracking Module . . . . .	18
3.3.1	Functional Requirements . . . . .	18
3.3.2	Performance Requirements . . . . .	20
3.4	Room Management Module . . . . .	20
3.4.1	Functional Requirements . . . . .	21
3.4.2	Performance Requirements . . . . .	23
3.5	Complaint Management Module . . . . .	23
3.5.1	Functional Requirements . . . . .	23
3.5.2	Performance Requirements . . . . .	24
3.6	Day Scholar Tracking Module . . . . .	24
3.6.1	Functional Requirements . . . . .	24
3.6.2	Performance Requirements . . . . .	25
<b>4</b>	<b>Other Non-Functional Requirements</b>	<b>26</b>
4.1	Security Requirements . . . . .	26
4.2	Reliability Requirements . . . . .	26
4.3	Usability Requirements . . . . .	27
4.4	Supportability Requirements . . . . .	27
4.5	Portability Requirements . . . . .	28
4.6	Maintainability Requirements . . . . .	28
<b>5</b>	<b>Meeting Minutes</b>	<b>29</b>
5.1	Meeting 1: Initial Requirements Discussion . . . . .	29
5.2	Meeting 2: Requirements Refinement and Validation . . . . .	30
<b>A</b>	<b>Use Case Diagram</b>	<b>32</b>
<b>B</b>	<b>Context Diagram</b>	<b>33</b>

# 1 Introduction

## 1.1 Purpose

This Software Requirements Specification (SRS) document provides a detailed description of all functions and specifications of the Smart Hostel Management System (SHMS). This (SRS) document is intended for:

- **Hostel Administration:** Ms. Nida Sultan Nahra (Hostel Warden) and hostel management staff who will use the system to manage daily hostel operations, and those operations are like monitoring student attendance, getting online complaints by students and allocation of the rooms.
- **Development Team:** The software development team (Muhammad Ahmad, Asad Ullah Khan, and Maryam Rashid) who will design, implement, test, and maintain the system based on the requirements specified in this document and discussed with their RP.
- **Students (End Users):** Hostel residents who will interact with the system to mark attendance, submit online reason before leaving Hostel/University, register complaints, group wise room selection and manage their hostel-related activities.  
On the other Hand Day Scholars (student) will also included in the attendance, in order to make proper check in that who enter hostel or when exit Hostel/University.
- **System Administrators:** Technical staff will be responsible for deploying, configuring, and maintaining the system infrastructure in the Hostel as well as in the university.
- **Quality Assurance Team:** Testing will be done by the team to verify that the implemented system meets all specified requirements and quality standards.
- **Stakeholders:** Hostel Management, University management, security personnel, maintenance staff who interact with or are affected by the system.

The purpose of SHMS is to automate and digitize hostel management operations at Namal University to replace the manual and inconsistent processes with a unified, efficient, and transparent digital platform.

## 1.2 Scope

The Smart Hostel Management System (SHMS) is a comprehensive web-based application designed to manage all aspects of hostel operations at Namal University. The system is designed for female hostel facilities and operates within the girls hostel environment and as well as in the university. It enforces strict security and access controls to ensure that only authorized users can access system functions and student information. All data related to hostel residents is handled securely to maintain privacy and safety at all times.

### 1.2.1 System Name and Components

**System Name:** Smart Hostel Management System (SHMS)

#### Major Components:

- Attendance Management Module
- Biometric Entry and Exit Tracking System
- Room Allocation and Management Module
- Day Scholar Tracking Module
- Centralized Student Database
- Reporting and Analytics Dashboard
- Complaint Management System

### 1.2.2 What the Software Will Do

The SHMS will accomplish the following objectives:

#### 1. Automate Attendance Tracking:

- Integrate with the existing biometric systems to automatically record student attendance for hostelers using biometric devices placed inside the hostel
- Provide real-time attendance dashboards for wardens
- Automatically identify and flag absent students
- Generate daily, weekly, and monthly attendance reports
- Send automated notifications (through mail or SMS) to wardens at 10:05 PM about unauthorized absences and absentee list
- Link attendance with the last online exit application submitted by the student

#### 2. Biometric Entry and Exit Tracking:

- Enable students to submit online reason/application before exiting hostel or university (no approval needed except for restricted times after 5 PM)
- Provide hostel warden with a centralized interface to monitor and view entry/exit logs in real time.
- Automatically records each student during entry/exit without the need for manual gate passes using biometric devices at main gate
- Allow security personnel to make sure that students are using the biometric system during entry/exit points.
- Track student return status and send automated notification for late returns or violations (immediate for restricted time violations)
- For short leaves (shopping, medical, etc.), exit allowed only before 5 PM and entry compulsory before 5 PM, with immediate notifications for violations

- For day scholars, track entry/exit from hostel and university (application required for non-routine university exit and no approval needed)
- Send daily notifications at 10:05 PM to warden about attendance, day scholars in hostel, and any security events

### 3. Streamline Room Management:

- Maintain real-time inventory of approximately 50 rooms with capacities of 2, 4, or 8 beds
- Facilitate group-wise room allocation where exactly 2, 4, or 8 students agree and one submits the form with all details; no duplicate registrations allowed
- Track room occupancy history and maintenance schedules
- Manage room change requests with reason and vacancy viewing
- Handle room clearance processes when students vacate
- Generate room occupancy reports and statistics

### 4. Handle Complaints:

- Allow students to submit online complaints with details and attachments
- Send immediate notifications to warden
- Track status and resolution
- Generate complaint reports

### 5. Monitor Day Scholar Access:

- Register and manage day scholar profiles
- Track day scholar entry and exit from hostel premises using biometrics
- Send real-time notifications when day scholars enter hostel
- Maintain logs for day scholars
- Include current day scholar count in daily 10:05 PM notification

### 6. Centralize Student Information:

- Maintain unified student profiles with hostel-related information
- Synchronize with university student information systems
- Provide role-based access to student data
- Generate comprehensive student hostel history reports

#### 1.2.3 Benefits and Objectives

The implementation of SHMS will provide the following benefits:

- **Efficiency:** Reduce manual paperwork and administrative load by 70-80% and allowing wardens to focus on student welfare rather than administrative tasks.
- **Transparency:** Provide real-time visibility into all hostel operations for authorized stakeholders and ensuring that everyone has equal access to system.

- **Security:** Enhance hostel security through digital verification on gate entry/exit through biometric, automated absence tracking, and day scholar monitoring, with focus on girls' security.
- **Accuracy:** Eliminate human errors in attendance marking or finding who was absent and help a lot in record keeping.
- **Accessibility:** Enable 24/7 access to hostel services and information through web interfaces.
- **Data-Driven Decision Making:** Offer detailed reports and analytics on student activities and hostel operations to guide effective management and improving overall hostel efficiency.

#### 1.2.4 What the Software Will NOT Do

To set clear boundaries, the following are explicitly outside the scope of SHMS:

- The system will not handle academic performance tracking or course registration (these remain with the university's academic management system)
- The system will not process financial aid or scholarship applications (handled by university financial services)
- The system will not manage library operations, sports facilities, or other non-hostel university services
- The system will not replace or integrate with existing university ERP for financial accounting
- The system will not provide medical services management or health records (it will be handled by university health center)
- The system will not manage faculty or staff housing (scope limited to girls student)
- The system will not handle disciplinary actions or conduct violations (these remain administrative decisions documented separately)

### 1.3 Definitions, Acronyms, and Abbreviations

This section provides clear definitions of important terms and abbreviations used throughout the Smart Hostel Management System (SHMS) documentation. These explanations ensure that all readers understand the terminology consistently.

Term	Definition
<b>SHMS</b>	Smart Hostel Management System the software platform designed to manage hostel operations such as attendance, gate passes, and security activities.
<b>Warden</b>	The hostel administrator who oversees daily operations, approves requests, monitors attendance, and manages students.
<b>Biometric System</b>	A biometric device used to mark student attendance and connected with the SHMS for automated updates.
<b>Absentee</b>	A student who fails to mark attendance before the deadline and does not hold an approved gate pass.
<b>Role-Based Access</b>	A security method in which each user role (student, warden, security) has access only to the features relevant to their responsibilities.
<b>SRS</b>	Software Requirements Specification this document, which describes how the system should function.
<b>IEEE</b>	Institute of Electrical and Electronics Engineers the organization whose documentation standards are followed in this SRS.
<b>UI</b>	User Interface the visual and interactive elements of the system that users interact with.
<b>API</b>	Application Programming Interface enables communication between SHMS and other systems such as biometric machines or student databases.
<b>HTTPS</b>	Hypertext Transfer Protocol Secure used for secure communication between the system and the server.
<b>JSON</b>	JavaScript Object Notation a lightweight format used for exchanging data between system components.
<b>CSV</b>	Comma-Separated Values a file format used for importing or exporting hostel-related data.
<b>SMTP</b>	Simple Mail Transfer Protocol used by the system to send automated email notifications.
<b>SMS</b>	Short Message Service used for sending text alerts such as approval messages or warnings.

## 1.4 References

This section lists the key documents, standards, and resources that were consulted during the preparation of this Software Requirements Specification (SRS). These materials support the structure, technical decisions, and background of the Smart Hostel Management System.

1. **IEEE Standard 830-1984:** Guide for preparing Software Requirements Specifications.
2. **IEEE Standard 29148-2011:** International standard for systems and software requirements engineering processes.
3. **Namal University Hostel Rules and Regulations (2025):** Official policies regarding hostel operations, attendance, and entry or exit procedures.
4. **React.js Documentation:** Official reference for frontend development — <https://reactjs.org>.
5. **Node.js Documentation:** Backend development reference — <https://nodejs.org>.
6. **MongoDB Documentation:** Database and data model reference — <https://www.mongodb.com/docs>.
7. **Biometric Device Vendor Manual (2024):** Technical documentation for the fingerprint attendance hardware integrated with the system.

## 1.5 Overview

This section provides a brief summary of how the Software Requirements Specification (SRS) is arranged. It helps readers understand the flow of the document and easily locate information related to the Smart Hostel Management System (SHMS).

### 1.5.1 Document Structure

This SRS document is organized into several major sections, each focusing on a different aspect of the system:

- **Introduction:** Presents the purpose, scope, key definitions, references, and an overall overview of the SHMS project.
- **General Description:** Describes the system at a broader level, including main functions, user characteristics, constraints, and assumptions.
- **Specific Requirements:** Provides detailed functional requirements, external interface descriptions, and behavior of key system modules such as attendance, entry or exit tracking, room management, and security verification.
- **Other Non-Functional Requirements:** Covers system qualities such as security, reliability, usability, supportability, portability, and maintainability.
- **Appendices:** Contains diagrams, tables, and additional supporting material that further explains different system components.

### 1.5.2 Organization of the Document

The document is arranged in a clear and hierarchical format to ensure easy navigation. Each section is numbered to show how topics are connected and how the content flows from general descriptions to detailed specifications. Definitions and references are included near the beginning to provide context before technical requirements are introduced. Cross-references are used where needed to help readers trace related requirements across different parts of the document.

## 2 General Description

### 2.1 Product Perspective

The Smart Hostel Management System (SHMS) is a new custom-developed web-based application which will be used for Namal University's hostel operations. It addresses limitations in the current minor digital system, such as marking attendance but doesn't know who is missing, time-consuming verifications, and lack of real-time reporting. SHMS represents an evolution from previous tools (e.g., biometric devices for girls' hostel to track in/out) to an integrated platform.

#### 2.1.1 Background and Current System Limitations

Currently, hostel management relies on manual processes: attendance is recorded via biometrics for girls but manually checked who marked attendance, gate passes are paper-based, room allocations are tracked in spreadsheets, and day scholar tracking is ad-hoc. These actions lead to errors, delays, and security risks. SHMS aims to centralize these into a digital system, reducing workload and improving accuracy.

#### 2.1.2 System Interfaces

SHMS will integrate with:

- **Biometric Attendance Systems:** Real-time data import from fingerprint devices.
- **University Student Database:** API access for student profiles, enrollment, and updates.
- **Email/SMS Gateways:** For alert notifications.

#### 2.1.3 User Interfaces

Responsive and user-friendly web UI with dashboards, forms, and reports. Supports accessibility features like keyboard navigation.

#### 2.1.4 Hardware Interfaces

- Biometric devices on main gates.
- Server hardware or cloud (e.g., AWS/EC2 equivalents).
- Client devices: Desktops/Laptops, mobiles with browser support.

#### 2.1.5 Software Interfaces

- OS: Windows OR Mobile browser.
- DBMS: MongoDB for data storage.
- Framework: Node.js/Express for backend, React.js for frontend.
- Browsers: Chrome, Firefox, etc.

### 2.1.6 Communications Interfaces

- HTTPS for secure data.
- SMTP for emails.
- RESTful APIs for internal/external communication.
- Optional SMS via third-party gateway.

### 2.1.7 System Context

SHMS operates within hostel boundaries, aligned with university calendars, rules, and privacy policies. It interacts with external entities like students, staff, and IT systems.

## 2.2 Product Functions

SHMS has main modules to handle daily hostel operations.

### 2.2.1 Attendance Management

This module takes care of daily attendance to keep track of student presence and safety. It supports biometric scanning, manual entry when needed, leave requests, automatic absentee lists, and helpful reports and alerts for staff.

### 2.2.2 Biometric Entry and Exit Tracking

This module handles student entry and exit from the hostel and university using digital verification. Students can apply online, upload documents if needed, get approvals from staff for restricted times, and security personnel verify at the gate to record actual exit and return times. The system also sends alerts for overdue returns and provides reports.

### 2.2.3 Room Management

This module helps assign and manage hostel rooms. It keeps a record of all rooms, suggests or assigns rooms to new students, handles transfer requests, tracks maintenance issues, and manages room clearance when students leave.

### 2.2.4 Complaint Management

This module looks after student complaints. Students can file complaints (with photos), and the system tracks each complaint, sending notifications to warden.

### 2.2.5 Day Scholar Tracking

This module controls who enters the hostel. Security staff can log day scholars, get approvals if needed, issue time-limited passes, record entry and exit times, and send alerts if someone stays too long.

## 2.3 User Characteristics

Detailed profiles ensure the system meets diverse user needs.

### 2.3.1 Hostel Warden

- **Technical Expertise:** Moderate; familiar with web apps but not coding.
- **Primary Tasks:** Approvals, reports, oversight.
- **Usage Frequency:** Daily, high volume.
- **Interface Needs:** Advanced dashboard with alerts, customizable views, and quick navigation.
- **Additional Notes:** May require training on advanced features like report customization.

### 2.3.2 Assistant Warden

- **Technical Expertise:** Basic to moderate.
- **Primary Tasks:** Initial approvals, verifications.
- **Usage Frequency:** Daily.
- **Interface Needs:** Task-focused UI with checklists.
- **Additional Notes:** Support for mobile access during inspections.

### 2.3.3 Resident Students

- **Technical Expertise:** High
- **Education:** Undergraduates.
- **Primary Tasks:** Requests, views, submissions.
- **Usage Frequency:** Daily.
- **Interface Needs:** Intuitive, mobile-first with push notifications.

### 2.3.4 Security Personnel

- **Technical Expertise:** Basic
- **Education:** Basic.
- **Primary Tasks:** Make sure Biometric Entry.
- **Additional Notes:** Voice guidance or Urdu interface as Biometric is done

## 2.4 General Constraints

This section explains the limitations and conditions that may affect how the Smart Hostel Management System (SHMS) is designed, developed, and used. These constraints must be considered to ensure that the system functions effectively within the university environment.

#### **2.4.1 Regulatory and Policy Constraints**

- The system shall comply with all university rules and policies related to attendance, hostel operations, and student privacy.
- The system shall maintain female hostel data in such a way to ensure privacy and security.
- The system shall protect all stored student information according to data protection and privacy guidelines.

#### **2.4.2 Hardware Limitations**

- The system shall operate with the existing biometric devices installed in the hostels, even if these devices have limited features or outdated APIs.
- The system's performance may depend on the speed and stability of the university's network infrastructure.
- The system shall run effectively on a variety of staff devices, including desktops, laptops, tablets, and mobile phones.
- The system's availability may be affected by power interruptions, and it must rely on the university's UPS or backup systems.

#### **2.4.3 Interface Requirements**

- The system shall integrate with the existing student information database through approved APIs or supported data formats.
- The system shall use standard web communication protocols to ensure compatibility across different devices and networks.
- The system shall follow basic accessibility guidelines to support users with different abilities.

#### **2.4.4 Development Constraints**

- The system shall be fully developed, tested, and deployed within the limited academic semester timeline.
- The system shall be developed by a small student team, requiring the design to match their technical skill level.
- The system shall be built using free or open-source technologies to avoid licensing costs.
- The system shall use the predefined tech stack (React.js, Node.js, MongoDB) for consistency across all modules.

#### **2.4.5 Operational Constraints**

- The system shall remain available during major hostel operating hours, with occasional maintenance windows when needed.
- The system shall support a large number of users at the same time, especially during busy evening periods.
- The system shall provide limited offline support to ensure essential features remain accessible during slow or unstable internet conditions.
- The system's performance may vary during peak usage times, such as semester start or hostel room allocation days.

#### **2.4.6 Security and Safety Constraints**

- The system shall use role-based access control to ensure that only authorized users can perform sensitive operations.
- The system shall maintain secure audit logs of important user actions, such as approvals and data updates.
- The system shall implement measures to protect against unauthorized access, data breaches, and cyber threats.
- The system shall support emergency procedures by allowing wardens quick access to attendance and gate pass information when needed.

### **2.5 Assumptions and Dependencies**

#### **2.5.1 Assumptions**

- Infrastructure provision by university.
- Biometric API access.
- Student tech access.
- Training provided.
- IT support available.
- Policies stable.
- Connectivity reliable.
- User adoption high.

### 2.5.2 Dependencies

- Vendor cooperation for biometrics.
- IT for database access.
- Network stability.
- RP availability.
- Tech stack updates.
- Email/SMS services.
- Budget for hosting.

## 3 Specific Requirements

This section details requirements, organized by external interfaces and functional modules. Performance requirements are included per module.

### 3.1 External Interfaces

#### 3.1.1 User Interfaces

The system shall have a clean and responsive web interface built with React.js. It will show different dashboards based on the user's role, easy forms (for example, an entry/exit request form with fields for dates and reason), and reports that can be downloaded as PDF or CSV. The interface will be simple to use, with helpful hints, clear error messages, and full support for mobile devices.

#### 3.1.2 Hardware Interfaces

The system shall connect to biometric fingerprint devices through network APIs. The server needs at least 8GB RAM and a 4-core CPU (can be university server or cloud). Users can access the system from any computer, tablet, or phone with a modern web browser.

#### 3.1.3 Software Interfaces

The system shall use MongoDB to store all data with proper structure for students, rooms, attendance, etc. The backend will run on Node.js with Express to handle APIs (for example, POST requests for saving attendance). The frontend will use React.js for smooth and fast user interaction. It will work on Linux servers and be compatible with all major web browsers.

#### 3.1.4 Communications Interfaces

The system shall always use secure HTTPS for all connections. It will communicate using REST APIs with JSON data format. Emails will be sent via SMTP. The system will work over the university's normal internet or any Wi-Fi connection.

## 3.2 Attendance Management Module

This module ensures accurate tracking of student presence, integrating biometrics for automation. It includes workflows for leaves and reports to support warden decision-making.

### 3.2.1 Functional Requirements

**ATT-REQ-001: Biometric Data Import** The system shall automatically import attendance records from biometric devices every hour between 6:00 PM and 11:59 PM daily, validating data for duplicates and logging errors.

Description: This feature pulls data via API, maps fingerprints to student IDs, and updates the database.

**ATT-REQ-002: Manual Attendance Entry** The system shall allow wardens and assistant wardens to manually mark attendance for students when biometric systems are unavailable.

Description: UI form with student search, date selection, and confirmation.

**ATT-REQ-003: Attendance Deadline** The system shall consider all students who have not marked attendance by 10:00 PM as absent for that day, triggering automated processes.

Description: Cron job runs at deadline to flag absentees.

**ATT-REQ-004: Absentee Identification** The system shall automatically generate a list of absent students at 10:01 PM each day, excluding students with approved leave applications, and sort by room/floor.

Description: Query database for unmarked records.

**ATT-REQ-005: Absentee Notification** The system shall send automated notifications to wardens and assistant wardens within 5 minutes of generating the absentee list, including list attachments.

Description: Email with links to dashboard.

**ATT-REQ-006: Leave Request Submission** The system shall allow students to submit leave requests at least 24 hours in advance, specifying departure date, return date, reason, and optional attachments (max 5MB).

Description: Form validation for dates, reason length.

**ATT-REQ-007: Leave Request Approval Workflow** The system shall route leave requests through a two-level approval process: first to assistant warden, then to warden, with email notifications at each step.

Description: State machine for status (pending, approved1, approved2).

**ATT-REQ-008: Emergency Leave Request** The system shall provide a mechanism for emergency leave requests that require only warden approval and can be submitted with less than 24 hours notice, flagged as high priority.

Description: Separate form with urgency field.

**ATT-REQ-009: Leave Status Notification** The system shall notify students via email within 5 minutes when their leave request status changes (approved, rejected, or pending), including reasons for rejection.

Description: Webhooks for real-time updates.

**ATT-REQ-010: Attendance History Viewing** The system shall allow students to view their own attendance history for the current semester and previous semesters, with filters and export options.

Description: Paginated table with dates, status.

**ATT-REQ-011: Attendance Report Generation** The system shall allow wardens to generate attendance reports filtered by date range, hostel wing, floor, room number, or individual student, in PDF/CSV format.

Description: Custom queries with aggregation.

**ATT-REQ-012: Attendance Statistics Dashboard** The system shall display real-time attendance statistics including total present, total absent, leave count, and attendance percentage on the warden dashboard, with charts.

Description: Live updates via WebSockets.

**ATT-REQ-013: Repeated Absence Alerts** The system shall automatically flag students who have been absent for 3 or more consecutive days without approved leave and send alerts to the warden, with student details.

Description: Daily check job.

**ATT-REQ-014: Attendance Data Retention** The system shall retain attendance records for a minimum of 5 years for historical reference and compliance purposes, with archiving for old data.

Description: Database partitioning.

**ATT-REQ-015: Bulk Attendance Import** The system shall support bulk import of attendance data from CSV or Excel files for migration or backup restoration purposes, with validation.

Description: Admin-only feature.

**ATT-REQ-016: Attendance Correction** The system shall allow wardens to correct attendance records within 48 hours, requiring justification and audit logging.

Description: Edit form with history.

**ATT-REQ-017: Integration with University Calendar** The system shall integrate with university holidays to adjust attendance expectations automatically.

Description: API pull for dates.

**ATT-REQ-018: Student Attendance Reminders** The system shall send reminders to students at 9:00 PM if attendance is not marked.

Description: Scheduled notifications.

**ATT-REQ-019: Export Attendance Data** The system shall allow export of attendance data for external analysis.

Description: Multiple formats.

**ATT-REQ-020: Attendance Audit Logs** The system shall maintain logs of all attendance changes for security.

Description: Immutable records.

### 3.2.2 Performance Requirements

**ATT-PERF-001** The system shall process biometric imports for 1000 students within 1 minute.

**ATT-PERF-002** Absentee generation shall complete within 10 seconds for 500 students.

**ATT-PERF-003** Reports shall generate within 10 seconds for 1 month data.

**ATT-PERF-004** Dashboard updates shall occur in real-time ( $\pm 2$  seconds lag).

**ATT-PERF-005** Notifications shall send within 30 seconds of trigger.

## 3.3 Biometric Entry and Exit Tracking Module

This module provides secure, real-time tracking of student entry and exit using biometrics and online applications, with approval for restricted times and automated alerts for security.

### 3.3.1 Functional Requirements

**BEET-REQ-001: Entry/Exit Request Submission** The system shall allow students to submit entry/exit requests specifying destination, purpose, departure time, expected return time, and contact details.

Description: Form with date pickers, validation for return departure.

**BEET-REQ-002: Supporting Document Upload** The system shall allow students to upload supporting documents (e.g., invitation letters, medical prescriptions) in PDF or image format, with maximum file size of 5MB per document, and virus scanning.

Description: Cloud storage integration.

**BEET-REQ-003: Entry/Exit Approval Workflow** The system shall route entry/exit requests first to the assistant warden for initial approval, then to the warden for final approval.

Description: Email links for approval.

**BEET-REQ-004: Entry/Exit Rejection** The system shall allow wardens and assistant wardens to reject exit requests with (optional) reason explaining the rejection, and optional comments.

Description: Dropdown for common reasons.

**BEET-REQ-005: Emergency Entry/Exit Request** The system shall provide an emergency entry/exit option that is routed directly to the warden and marked with high priority for immediate attention, with phone call option.

Description: Red-flagged in dashboard.

**BEET-REQ-006: Entry/Exit Expiry** The system shall automatically expire approved requests 1 hour after the specified return time and flag the student as overdue, updating status.

Description: Job for expiry check.

**BEET-REQ-007: Entry/Exit Verification** The system shall provide security personnel with a verification interface that validates requests using biometric, showing student info.

Description: Mobile app or web for gates.

**BEET-REQ-008: Exit Activation** The system shall activate an approved request only when the student physically exits through the gate and security personnel scan or enter the request ID, recording exit time.

Description: Geo-location optional.

**BEET-REQ-009: Return Time Recording** The system shall record the actual return time when security personnel verify the student's return using the request ID, calculating duration.

Description: Auto-alert if late.

**BEET-REQ-010: Overdue Alerts** The system shall send automated alerts to wardens when students exceed their approved return time by more than 1 hour.

Description: Multi-channel alerts.

**BEET-REQ-011: Entry/Exit History** The system shall maintain a complete history of all entry/exit requests issued to each student.

Description: Filtered views.

**BEET-REQ-012: Bulk Requests for Events** The system shall allow wardens to create bulk entry/exit requests for university-organized events, specifying event name, date, and list of participating students from CSV.

Description: Group approval.

**BEET-REQ-013: Request Cancellation** The system shall allow students to cancel pending exit requests and wardens to revoke approved but unused requests, with notifications.

Description: Status update.

**BEET-REQ-014: Entry/Exit Statistics** The system shall generate statistics showing total requests issued, average processing time, rejection rate, and overdue rate for administrative review, with charts.

Description: Monthly reports.

**BEET-REQ-015: Concurrent Request Limit** The system shall enforce a maximum of 3 pending entry/exit requests per student at any given time.

Description: Check on submission.

**BEET-REQ-016: Extension Request** The system shall allow students to request extensions for active requests, send for approval.

Description: Limited to once per request.

**BEET-REQ-017: Integration with Attendance** The system shall link approved entry/exit requests to attendance, marking as on-leave if approved.

Description: Auto-sync.

**BEET-REQ-018: QR Code Generation** The system shall generate secure QR codes for approved requests, printable or mobile-displayable.

Description: Secure encoding.

**BEET-REQ-019: Verification Logs** The system shall log all verifications for audit.

Description: Timestamped.

**BEET-REQ-020: Custom Request Types** The system shall support different request types (e.g., medical, family), with specific workflows.

Description: Configurable.

### 3.3.2 Performance Requirements

**BEET-PERF-001** Verification shall complete within 2 seconds.

**BEET-PERF-002** Notifications within 1 minute.

**BEET-PERF-003** Form load within 3 seconds.

**BEET-PERF-004** Statistics generation within 10 seconds.

**BEET-PERF-005** Bulk processing for 100 students within 1 minute.

## 3.4 Room Management Module

Manages accommodations with automation for efficiency and maintenance.

### 3.4.1 Functional Requirements

**RM-REQ-001: Room Database** The system shall maintain a database of all hostel rooms containing room number, wing, capacity, current occupancy, amenities list, and status (available, maintenance).

Description: Admin editable.

**RM-REQ-002: Automatic Room Allocation** The system shall automatically suggest available rooms for new students based on gender, program, room capacity, and preferences (e.g., floor), prioritizing vacancy.

Description: Algorithm for matching.

**RM-REQ-003: Manual Room Assignment** The system shall allow wardens to manually override automatic suggestions and assign students to specific rooms, with conflict checks.

Description: Drag-drop UI optional.

**RM-REQ-004: Room Capacity Enforcement** The system shall prevent allocation of students to rooms that have reached their maximum capacity, displaying errors.

Description: Real-time check.

**RM-REQ-005: Room Transfer Request** The system shall allow students to request room transfers, specifying reasons, preferred destination, and roommate consents if needed.

Description: Form with approvals.

**RM-REQ-006: Room Transfer Approval** The system shall route room transfer requests to the warden for approval and require confirmation from both current and destination roommates via email links.

Description: Workflow tracking.

**RM-REQ-007: Vacancy Display** The system shall display real-time vacancy information showing available rooms, partially occupied rooms, and fully occupied rooms organized by floor, wing.

Description: Interactive dashboard.

**RM-REQ-008: Room Clearance Initiation** The system shall allow students to initiate room clearance requests when preparing to vacate, specifying expected checkout date and inventory list.

Description: Checklist form.

**RM-REQ-009: Room Inspection Scheduling** The system shall allow assistant wardens to schedule room inspection appointments with students requesting clearance, integrating with calendars.

Description: Availability slots.

**RM-REQ-010: Room Condition Documentation** The system shall provide a checklist interface for documenting room condition during inspection including furniture status, cleanliness, damages, with photo uploads.

Description: Mobile-friendly.

**RM-REQ-011: Clearance Approval** The system shall allow wardens to issue final clearance approval after reviewing inspection reports and confirming no outstanding issues or dues.

Description: Digital signature optional.

**RM-REQ-012: Maintenance Request Submission** The system shall allow students to submit maintenance requests for their rooms, specifying issue type and optional photo attachments (max 5MB).

Description: Categorized (plumbing, electrical, etc).

**RM-REQ-013: Maintenance Request Tracking** The system shall track maintenance requests through statuses: Submitted, Acknowledged, In Progress, Completed, and allow students/wardens to view current status and history.

Description: Notifications on updates.

**RM-REQ-014: Room Occupancy Reports** The system shall generate occupancy reports showing utilization rates by floor, wing, overall hostel, with trends over time for planning.

Description: Graphical reports.

**RM-REQ-015: Roommate Information** The system shall display roommate names, contact information, and profiles to occupants of shared rooms for communication, with privacy controls.

Description: Consent-based sharing.

**RM-REQ-016: Room Amenity Management** The system shall allow admins to update room amenities and reflect in allocation suggestions.

Description: Database fields.

**RM-REQ-017: Bulk Allocation for New Semester** The system shall support bulk room allocations for incoming students from lists.

Description: Import feature.

**RM-REQ-018: Room History Logs** The system shall log all allocations, transfers for audit.

Description: Searchable.

**RM-REQ-019: Preference Matching** The system shall match rooms based on student preferences (e.g., quiet floor).

Description: Custom fields.

**RM-REQ-020: Integration with Student DB** The system shall sync room data with university student records.

Description: Auto-updates.

### 3.4.2 Performance Requirements

**RM-PERF-001** Vacancy updates within 5 seconds.

**RM-PERF-002** Reports within 10 seconds.

**RM-PERF-003** Allocation suggestions within 3 seconds.

**RM-PERF-004** Maintenance tracking loads in 2 seconds.

**RM-PERF-005** Bulk operations for 200 students within 2 minutes.

## 3.5 Complaint Management Module

Handles complaint tracking for quality improvement.

### 3.5.1 Functional Requirements

**COMP-REQ-001: Complaint Submission** The system shall allow students to submit complaints selecting type (general, maintenance, etc), description, and rating (1-5 stars).

Description: Anonymous option.

**COMP-REQ-002: Complaint Photo Attachment** The system shall allow students to attach up to 3 photos with each complaint for evidence, max 5MB per photo, with compression.

Description: Storage optimization.

**COMP-REQ-003: Complaint Acknowledgment** The system shall automatically acknowledge complaint receipt and assign a unique tracking number within 1 minute of submission, emailing confirmation.

Description: Auto-response.

**COMP-REQ-004: Complaint Assignment** The system shall route complaints to appropriate staff based on type.

Description: Assignment rules.

**COMP-REQ-005: Complaint Status Updates** The system shall allow staff to update status (Received, Under Review, Resolved, Closed) with comments and evidence.

Description: History log.

**COMP-REQ-006: Complaint Resolution Notification** The system shall notify students when complaints are resolved or closed with details of actions taken and satisfaction survey link.

Description: Feedback loop.

**COMP-REQ-007: Complaint Reports** The system shall generate reports on complaints, with filters and exports.

Description: Analytics.

### 3.5.2 Performance Requirements

**COMP-PERF-001** Complaint ack within 60 seconds.

**COMP-PERF-002** Reports within 15 seconds.

**COMP-PERF-003** Notifications within 30 seconds.

## 3.6 Day Scholar Tracking Module

Ensures secure access control with logging and alerts.

### 3.6.1 Functional Requirements

**DS-REQ-001: Day Scholar Entry Logging** The system shall allow security to log day scholar entries by Biometric, recording time, purpose, visited resident.

Description: Quick form.

**DS-REQ-002: Day Scholar Registration** The system shall capture day scholar details: name, contact, address, relationship, purpose, ID upload.

Description: Form with validation.

**DS-REQ-003: Approval Workflow** The system shall route requests for approval if required, with instant for low-risk.

Description: Configurable rules.

**DS-REQ-004: Entry Verification** The system shall generate entry with Biometric for verification.

Description: Scan-based.

**DS-REQ-005: Time-Limit Enforcement** The system shall enforce time limits, flag overdues.

Description: Timer job.

**DS-REQ-006: Exit Logging** The system shall log exits using Biometric, calculate duration.

Description: Auto-close.

**DS-REQ-007: Unauthorized Entry Alert** The system shall alert on unauthorized attempts in real-time.

Description: Siren integration optional.

**DS-REQ-008: Pre-Approved Requests** The system shall allow pre-approvals for recurring visits.

Description: Schedule-based.

**DS-REQ-009: History Viewing** The system shall show history for students/wardens.

Description: Filtered.

**DS-REQ-010: Overdue Notifications** The system shall notify after 15 minutes overdue.

Description: Multi-user.

**DS-REQ-011: Blacklist Management** The system shall maintain blacklists, block entries.

Description: Admin editable.

**DS-REQ-012: Statistics Dashboard** The system shall display stats: visits, durations, peaks.

Description: Charts.

**DS-REQ-013: Reports Generation** The system shall generate filtered reports.

Description: Exports.

**DS-REQ-014: Emergency Contact Linkage** The system shall link to student emergencies.

Description: Quick access.

**DS-REQ-015: Offline Entry Support** The system shall support offline logging, sync later.

Description: Local storage.

**DS-REQ-016: Photo Capture** The system shall capture photos at entry.

Description: Webcam integration.

**DS-REQ-017: Group Support** The system shall handle groups.

Description: Linked records.

**DS-REQ-018: Purpose Categories** The system shall categorize purposes for reporting.

Description: Dropdowns.

### 3.6.2 Performance Requirements

**DS-PERF-001** Logging within 3 seconds.

**DS-PERF-002** Alerts within 10 seconds.

**DS-PERF-003** Reports within 15 seconds for 6 months.

**DS-PERF-004** Offline sync within 1 minute online.

**DS-PERF-005** Dashboard load within 5 seconds.

## 4 Other Non-Functional Requirements

### 4.1 Security Requirements

The following security requirements ensure that the Smart Hostel Management System remains safe and protects all user data:

- **SEC-REQ-001:** The system shall restrict access based on user roles so that each user can only view or use the features allowed for their position.
- **SEC-REQ-002:** The system shall protect all user accounts with secure login and strong password rules to prevent unauthorized access.
- **SEC-REQ-003:** The system shall encrypt sensitive information, including student records, attendance data, and entry or exit tracking, both during storage and transmission.
- **SEC-REQ-004:** The system shall maintain logs of important actions, such as approvals or data updates, to help track suspicious or incorrect behavior.
- **SEC-REQ-005:** The system shall include protections against common security threats such as brute-force attacks, unauthorized page access, and harmful code injections.

### 4.2 Reliability Requirements

The following reliability requirements ensure that the Smart Hostel Management System performs consistently and remains dependable during everyday use:

- **REL-REQ-001:** The system shall remain available and fully operational during standard hostel hours (6:00 AM to 11:00 PM) with an uptime target of 99%.
- **REL-REQ-002:** The system shall recover from unexpected server failures within a few minutes without losing any previously saved or committed data.
- **REL-REQ-003:** The system shall handle high user load smoothly, ensuring stable performance even when many students or staff use the system at the same time.
- **REL-REQ-004:** The system shall safely store user actions such as attendance updates so that no data is lost during temporary network interruption.
- **REL-REQ-005:** The system shall automatically create regular data backups to ensure that information can be restored in case of technical failure.

- **REL-REQ-006:** The system shall operate reliably across multiple device types (mobile, desktop, tablet) without unexpected crashes or errors.

### 4.3 Usability Requirements

The following usability requirements ensure that the Smart Hostel Management System is simple, comfortable, and easy for all types of users:

- **USA-REQ-001:** The system shall support both English and Urdu languages so users can choose the language they are most comfortable with.
- **USA-REQ-002:** The system shall allow new users to perform basic tasks, such as submitting attendance, without requiring prior training.
- **USA-REQ-003:** The system shall provide tooltips and help icons on detailed or complex features to guide users when needed.
- **USA-REQ-004:** The system shall display error messages in simple, user-friendly language along with suggestions on how to fix the issue.
- **USA-REQ-005:** The system shall maintain a consistent color scheme, layout, and visual design across all pages to avoid user confusion.
- **USA-REQ-006:** The system shall clearly highlight important actions such as “Submit,” “Save,” and “Approve” to help users complete tasks easily.

### 4.4 Supportability Requirements

The following supportability requirements ensure that the Smart Hostel Management System can be easily maintained, updated, and supported after deployment:

- **SUP-REQ-001:** The system shall be easy to update so that new features or fixes can be added without affecting other modules.
- **SUP-REQ-002:** The system shall include clear documentation that explains how each part of the system works and how it can be maintained.
- **SUP-REQ-003:** The system shall provide helpful error messages and maintain proper logs to assist technical staff in troubleshooting issues quickly.
- **SUP-REQ-004:** The system shall be designed in a modular structure so different modules (attendance, gate pass, rooms, etc.) can be maintained separately.
- **SUP-REQ-005:** The system shall support regular data backups to ensure easy recovery of important information if any failure occurs.
- **SUP-REQ-006:** The system shall be designed to support future improvements and upgrades with minimal changes.

## 4.5 Portability Requirements

The portability requirements ensure that the Smart Hostel Management System can run easily across different devices, browsers, and hosting environments. The system should remain usable even if the university changes its technical setup in the future.

- **PORT-REQ-001:** The system shall run smoothly on all major web browsers including Chrome, Firefox, Edge, and Safari without requiring special configurations.
- **PORT-REQ-002:** The system shall be usable on multiple device types such as desktops, laptops, tablets, and smartphones with consistent functionality.
- **PORT-REQ-003:** The system shall be operating-system independent and function properly on Windows, macOS, and Linux platforms.
- **PORT-REQ-004:** The system shall be easily transferable to another server or hosting environment if required by the university.
- **PORT-REQ-005:** The system shall support cloud-based deployment on platforms like AWS, Azure, or any university-approved cloud service.
- **PORT-REQ-006:** The system shall use standard web technologies so that future updates or migrations can be performed without major code modifications.

## 4.6 Maintainability Requirements

The maintainability of the Smart Hostel Management System is important to ensure that the system can be updated, improved, and fixed easily over time. The following non-functional requirements define how maintainability will be supported:

- **MAIN-REQ-001:** The system shall be developed using a modular structure so that each major feature (attendance, Biometric Tracking system, room management, complaints) can be maintained independently.
- **MAIN-REQ-002:** The system shall be designed in a way that allows new features or policy changes to be added with minimal changes to the existing code.
- **MAIN-REQ-003:** The system shall include clear and simple documentation that explains modules, workflows, and important functions to help future developers maintain the system easily.
- **MAIN-REQ-004:** The code-base shall follow consistent naming conventions and folder structures so developers can quickly locate and update required sections.
- **MAIN-REQ-005:** The system shall maintain readable logs and organized error messages to support quick debugging and problem resolution.
- **MAIN-REQ-006:** The system shall store adjustable settings (such as attendance deadlines, visiting hours) in configuration files instead of hard-coded values.
- **MAIN-REQ-007:** The database design shall be flexible enough to allow the addition of new fields, tables, or relationships as hostel needs grow over time.
- **MAIN-REQ-008:** The system shall support maintenance operations with minimal downtime to ensure that updates or fixes do not disturb daily hostel activities.

## 5 Meeting Minutes

This appendix contains records of all meetings held with the Requirement Provider (Ms. Nida Sultan Nahra) throughout the requirements gathering and validation process.

### 5.1 Meeting 1: Initial Requirements Discussion

**Date:** December 12, 2025

**Time:** 2:00 PM - 3:30 PM

**Location:** Namal University, Mianwali

**Attendees:**

- Ms. Nida Sultan Nahra (Requirement Provider - Hostel Warden)
- Muhammad Ahmad (Team Lead)
- Asad Ullah Khan (Team Member)
- Maryam Rashid (Team Member)

**Agenda:**

1. Introduction and project overview
2. Discussion of current hostel management challenges
3. Identification of key stakeholders
4. Initial requirements gathering

**Discussion Summary:**

Ms. Nida Sultan Nahra outlined the major pain points in current hostel operations:

- Manual attendance verification is time-consuming and error-prone
- Paper-based gate slip system lacks traceability and security
- Room allocation is managed through spreadsheets leading to conflicts
- No centralized student database for hostel-specific information

Key functional requirements identified:

- Automated attendance tracking integrated with biometric devices
- Online application method before exiting hostel/university
- Comprehensive room management module
- Day scholar tracking for security purposes

**Next Meeting:** December 23, 2025

## 5.2 Meeting 2: Requirements Refinement and Validation

**Date:** December 23, 2025

**Time:** 10:00 PM

**Location:** Online meeting through Zoom

**Attendees:**

- Ms. Nida Sultan Nahra (Requirement Provider)
- Muhammad Ahmad (Team Lead)
- Asad Ullah Khan (Team Member)
- Maryam Rashid (Team Member)

**Agenda:**

1. Review of information provided by RP
2. Detailed discussion of each functional module
3. Clarification of user roles and permissions
4. Discussion of integration requirements
5. Validation of non-functional requirements

**Discussion Summary:**

**Attendance Management:**

- Biometric devices will be used
- Attendance window: 6:00 PM to 10:00 PM daily
- Absent student list must be generated by 10:05 PM
- Manual override needed for biometric failures

**Gate Exit System:**

- Current paper forms include: student details, destination, departure/return times, parent contact
- Average approval time: 2-4 hours currently
- Security checkpoint verification is manual, needs digital system solution
- Online short application and Biometric will be used to exit gate

**Room Management:**

- Total rooms: approximately 50 with capacities 2,4,8
- Group wise allocation with mutual agreement
- Room change requests handled case-by-case, need systematic workflow with vacancy viewing

- Clearance process (student absent for consecutive 6 months will be automatically removed from room allocation)

**Day Scholar Tracking:**

- Approximately 50-80 day scholars visit daily
- Entry/exit must be logged for security
- Warden notification needed when day scholars enter

**Non-Functional Requirements Validation:**

- System must support 100+ concurrent users
- Response time: under 3 seconds for most operations
- Data backup: daily automated backups essential
- Security: strict role-based access control required
- Mobile responsiveness: critical as most students use smartphones

**Clarifications:**

- Integration with university student information system is read-only
- SMS notifications limited to critical events due to cost
- Email notifications preferred for routine communications

## A Use Case Diagram

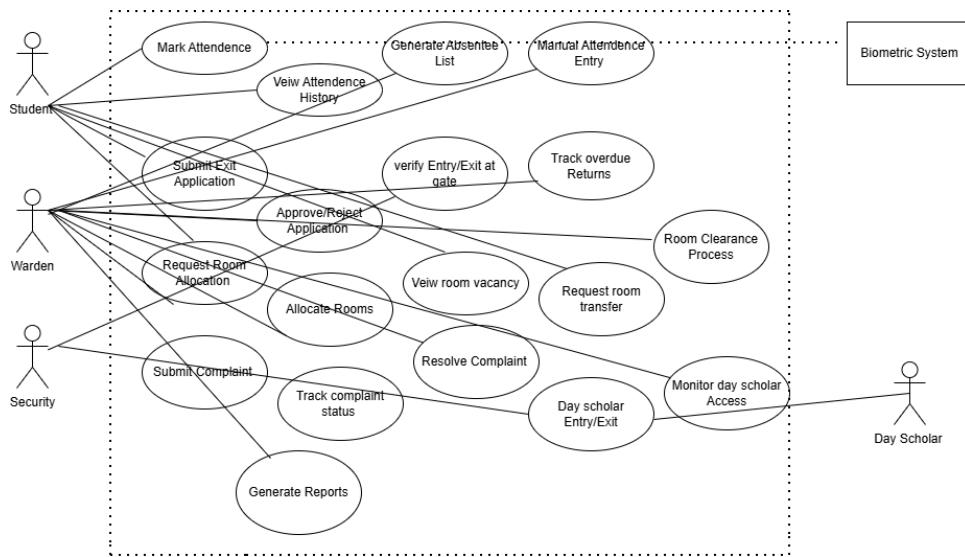


Figure 1: Use Case Diagram for SHMS

## B Context Diagram

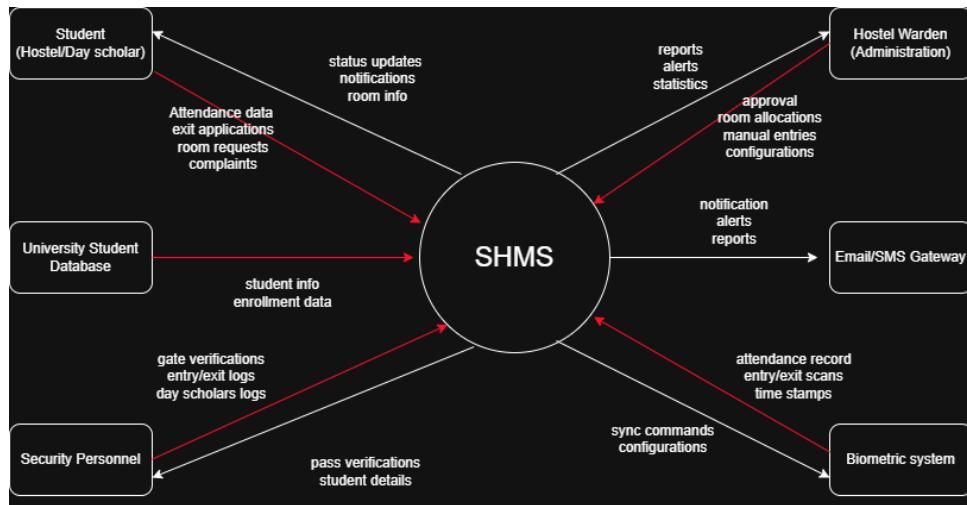


Figure 2: Context Diagram for SHMS