Software Requirements Specification
for
Smart Hostel Management System

Semester project 2025-2026

Version 1.0

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

1 Introduction **4**
- 1.1 Purpose . . . 4
  - 1.1.1 Overview of the Document . . . 4
  - 1.1.2 Purpose of the SRS . . . 4
  - 1.1.3 Intended Audience . . . 4
- 1.2 Scope . . . 5
  - 1.2.1 Product Identification . . . 5
  - 1.2.2 Product Features . . . 5
  - 1.2.3 Application of the Software . . . 5
  - 1.2.4 Goals and Benefits . . . 5
- 1.3 Definitions, Acronyms, and Abbreviations . . . 6
- 1.4 References . . . 7
- 1.5 Overview . . . 7
  - 1.5.1 Document Structure . . . 7
  - 1.5.2 Organization of the Document . . . 7

2 General Description **8**
- 2.1 Product Perspective . . . 8
  - 2.1.1 Background and Current System Limitations . . . 8
  - 2.1.2 System Interfaces . . . 8
  - 2.1.3 User Interfaces . . . 8
  - 2.1.4 Hardware Interfaces . . . 8
  - 2.1.5 Software Interfaces . . . 8
  - 2.1.6 Communications Interfaces . . . 9
  - 2.1.7 System Context . . . 9
- 2.2 Product Functions . . . 9
  - 2.2.1 1. Attendance Management . . . 9
  - 2.2.2 2. Gate Pass Management . . . 9
  - 2.2.3 3. Room Management . . . 10
  - 2.2.4 4. Mess Management . . . 10
  - 2.2.5 5. Visitor and Day Scholar Tracking . . . 10
- 2.3 User Characteristics . . . 10
  - 2.3.1 Hostel Warden . . . 11
  - 2.3.2 Assistant Warden . . . 11
  - 2.3.3 Resident Students . . . 11
  - 2.3.4 Mess Committee Members . . . 11
  - 2.3.5 Security Personnel . . . 12
- 2.4 General Constraints . . . 12
  - 2.4.1 Regulatory and Policy Constraints . . . 12
  - 2.4.2 Hardware Limitations . . . 12
  - 2.4.3 Interface Requirements . . . 12
  - 2.4.4 Development Constraints . . . 12
  - 2.4.5 Operational Constraints . . . 13
  - 2.4.6 Security and Safety Constraints . . . 13
- 2.5 Assumptions and Dependencies . . . 13
  - 2.5.1 Assumptions . . . 13

2.5.2 Dependencies 13

3 Specific Requirements **14**
3.1 External Interfaces 14
3.1.1 User Interfaces 14
3.1.2 Hardware Interfaces 14
3.1.3 Software Interfaces 14
3.1.4 Communications Interfaces 14
3.2 Attendance Management Module 14
3.2.1 Functional Requirements 14
3.2.2 Performance Requirements 16
3.3 Gate Pass Management Module 17
3.3.1 Functional Requirements 17
3.3.2 Performance Requirements 18
3.4 Room Management Module 19
3.4.1 Functional Requirements 19
3.4.2 Performance Requirements 20
3.5 Mess Management Module 21
3.5.1 Functional Requirements 21
3.5.2 Performance Requirements 22
3.6 Visitor and Day Scholar Tracking Module 23
3.6.1 Functional Requirements 23
3.6.2 Performance Requirements 24

4 Other Non-Functional Requirements **25**
4.1 Security Requirements 25
4.2 Reliability Requirements 25
4.3 Usability Requirements 25
4.4 Supportability Requirements 25
4.5 Portability Requirements 25
4.6 Maintainability Requirements 25

A Use Case Diagram **26**

B Context Diagram **26**

1 Introduction

1.1 Purpose

1.1.1 Overview of the Document

This Software Requirements Specification (SRS) document provides a detailed description of the Smart Hostel Management System (SHMS), including its functionalities, performance criteria, design constraints, and external interfaces. It serves as a comprehensive guide for the development, testing, and deployment phases, ensuring that the system aligns with stakeholder expectations and university needs.

1.1.2 Purpose of the SRS

The purpose of this document is to:

- Define the functional and non-functional requirements for SHMS in detail.
- Provide a clear roadmap for the development team to implement the system effectively.
- Ensure all stakeholders have a shared understanding of the system's objectives, scope, and constraints.

1.1.3 Intended Audience

The SRS is intended for a diverse group of stakeholders: **Stakeholders**

- University Administration: To ensure the system supports institutional goals like efficiency, security, and student welfare.
- Requirement Provider (Ms. Nida Sultan Nahra, Hostel Warden): To validate and refine requirements based on operational needs.
- Hostel Staff (Wardens, Assistant Wardens): To understand how the system will streamline daily tasks.

Development Teams

- Software Developers: To guide implementation using specified requirements and technologies.
- System Architects: To design robust, scalable architecture.

Testers

- QA Specialists: To create test cases and verify compliance with requirements.

End-Users

- Students and Security Personnel: To confirm the system meets usability and functional needs for daily interactions.

This document promotes collaboration among all parties for successful project delivery.

1.2 Scope

1.2.1 Product Identification

The product is named Smart Hostel Management System (SHMS), a web-based application for automating hostel operations at Namal University.

1.2.2 Product Features

What SHMS Will Do:

- Automate attendance tracking with biometric integration and leave management.
- Manage gate passes with online requests, approvals, and QR code verification.
- Handle room allocations, transfers, clearances, and maintenance requests.
- Oversee mess operations including menu publication, complaints, and account tracking.
- Track visitors and day scholars with logging, approvals, and alerts.

What SHMS Will Not Do (Initial Release):

- Manage academic records or grading.
- Handle university-wide finances beyond mess accounts.
- Book campus facilities outside hostels.
- Track library or transportation resources.
- Manage medical records.
- Integrate with unspecified external services.

1.2.3 Application of the Software

SHMS addresses inefficiencies in manual hostel management by providing:

- Digital automation to reduce paperwork and errors.
- Real-time monitoring for improved security and compliance.
- User-friendly interfaces for diverse users, enhancing operational efficiency.
- Scalable design for future university integrations.

1.2.4 Goals and Benefits

- Efficiency: Reduce administrative workload by 60% through automation.
- Security: Enhance safety with tracking and alerts.
- Transparency: Provide dashboards and reports for better oversight.
- Accessibility: Ensure mobile-responsive access for all users.
- Scalability: Support growth and additional features.

1.3 Definitions, Acronyms, and Abbreviations

This section provides a comprehensive list of terms used throughout the document to ensure clarity and consistency. Additional terms have been included to cover all aspects of the system.

Term	Definition
SHMS	Smart Hostel Management System - The software application being specified in this document.
RP	Requirement Provider (Ms. Nida Sultan Nahra) - The primary stakeholder responsible for providing and validating requirements.
Warden	Hostel administrator responsible for overall management, student welfare, and system oversight.
Biometric System	Fingerprint or facial recognition device for attendance, integrated with SHMS.
Mess	Hostel dining facility providing meals, managed through the system.
Absentee	Student failing to mark attendance by deadline without approved leave.
Role-Based Access	Security model restricting features by user role.
SRS	Software Requirements Specification - This document.
IEEE	Institute of Electrical and Electronics Engineers - Standards body.
UI	User Interface - The system's interactive elements.
API	Application Programming Interface - For system integrations.
HTTPS	Hypertext Transfer Protocol Secure - For secure communications.
JSON	JavaScript Object Notation - Data format for APIs.
QR Code	Quick Response Code - For gate pass verification.
CSV	Comma-Separated Values - File format for data import/export.
SMTP	Simple Mail Transfer Protocol - For email notifications.
SMS	Short Message Service - For text notifications.
PKR	Pakistani Rupee - Currency for mess transactions.

1.4 References

This section lists all referenced materials, expanded for completeness.

1. IEEE Std 830-1984, IEEE Guide to Software Requirements Specifications.
2. IEEE Std 29148-2011, Systems and software engineering — Life cycle processes — Requirements engineering.
3. Namal University Hostel Rules and Regulations, 2025.
4. React.js Documentation, `https://reactjs.org`.
5. Node.js Documentation, `https://nodejs.org`.
6. MongoDB Documentation, `https://www.mongodb.com/docs`.
7. Data Protection and Privacy Laws of Pakistan, 2021.
8. Biometric System Vendor Documentation (Fingerprint Devices), 2024.
9. Software Engineering: A Practitioner's Approach, Roger S. Pressman, 8th Edition, 2014.

1.5 Overview

1.5.1 Document Structure

This SRS document is organized as follows: 1. Introduction: Provides purpose, scope, definitions, references, and overview. 2. General Description: Covers product perspective, functions, user characteristics, constraints, and assumptions. 3. Specific Requirements: Details external interfaces and module-specific functional/performance requirements. 4. Other Non-Functional Requirements: Addresses security, reliability, usability, etc. Appendices: Include diagrams and additional materials.

1.5.2 Organization of the Document

Sections are hierarchically numbered for easy navigation. The document progresses from high-level overviews to detailed specifications, with cross-references for traceability. Definitions and references are placed early for context.

2 General Description

2.1 Product Perspective

The Smart Hostel Management System (SHMS) is a new, custom-developed web-based application tailored for Namal University's hostel operations. It addresses limitations in the current manual system, such as paper-based records, time-consuming verifications, and lack of real-time reporting. SHMS represents an evolution from fragmented tools (e.g., biometric devices for boys' hostels, manual logs for gate passes) 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 boys but manually for girls, gate passes are paper-based, room allocations are tracked in spreadsheets, mess operations involve physical complaint books, and visitor tracking is ad-hoc. These 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 notifications.

2.1.3 User Interfaces

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

2.1.4 Hardware Interfaces

- Biometric scanners via USB/network.
- Server hardware or cloud (e.g., AWS/EC2 equivalents).
- Client devices: Desktops, mobiles with browser support.

2.1.5 Software Interfaces

- OS: Linux server, cross-platform client.
- DBMS: MongoDB for data storage.
- Framework: Node.js/Express for backend, React.js for frontend.
- Browsers: Latest versions of 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 offers five core modules, each with detailed functions to ensure comprehensive coverage.

2.2.1 1. Attendance Management

This module automates attendance to ensure student safety and compliance.

- Automatic biometric import and manual entry options.
- Deadline enforcement, absentee lists, and notifications.
- Leave workflows with approvals and emergencies.
- Reports, statistics, and history viewing.
- Alerts for patterns like repeated absences.
- Data retention and bulk imports for administration.

Detailed scenarios include nightly processing and user interactions.

2.2.2 2. Gate Pass Management

Handles permissions for leaving premises, enhancing security.

- Request submission with documents.
- Workflows for approval/rejection.
- Verification, activation, and tracking.
- Alerts for overdues and statistics.
- Bulk and cancellation features.

Includes integration with security devices for QR scanning.

2.2.3 3. Room Management

Optimizes accommodation allocation and maintenance.

- Database maintenance and auto-allocation.
- Transfers, vacancies, and clearances.
- Inspections, maintenance tracking.
- Reports and roommate info.

Ensures capacity enforcement and real-time updates.

2.2.4 4. Mess Management

Manages dining operations for efficiency.

- Menu publication and notifications.
- Complaint handling with attachments.
- Account management, transactions, payments.
- Reports and balance alerts.

Supports financial transparency with debits/credits.

2.2.5 5. Visitor and Day Scholar Tracking

Controls access to premises for security.

- Logging, registration, approvals.
- Time limits, exits, alerts.
- History, statistics, blacklists.
- Offline support for reliability.

Integrates with gate security for verification.

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.
- **Education:** Advanced degree; administrative role.
- **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.
- **Education:** Bachelor's level.
- **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; tech-savvy generation.
- **Education:** Undergraduates.
- **Primary Tasks:** Requests, views, submissions.
- **Usage Frequency:** Weekly.
- **Interface Needs:** Intuitive, mobile-first with push notifications.
- **Additional Notes:** Multilingual support if needed (English/Urdu).

2.3.4 Mess Committee Members

- **Technical Expertise:** Basic.
- **Education:** Varied.
- **Primary Tasks:** Menu posts, complaint resolutions.
- **Usage Frequency:** Daily.
- **Interface Needs:** Simple forms, calendars.
- **Additional Notes:** Bulk upload features for menus.

2.3.5 Security Personnel

- **Technical Expertise:** Basic; possible literacy challenges.
- **Education:** Basic.
- **Primary Tasks:** Verifications, logging.
- **Usage Frequency:** Shift-based.
- **Interface Needs:** Minimalist, visual UI with QR scanners.
- **Additional Notes:** Voice guidance or Urdu interface.

2.4 General Constraints

Expanded to cover all aspects per rubric.

2.4.1 Regulatory and Policy Constraints

- Compliance with university rules, data laws.
- Gender segregation in access.
- Privacy for student data.
- Audit requirements for financial transactions.

2.4.2 Hardware Limitations

- Existing biometrics with limited APIs.
- University network bandwidth.
- Device compatibility for security.
- Power backup for servers.

2.4.3 Interface Requirements

- API integration with student DB.
- Standard web protocols.
- Accessibility standards (WCAG).

2.4.4 Development Constraints

- 12-month timeline.
- Small team, limited experience.
- Free tools only.
- Tech stack: React, Node, MongoDB.

2.4.5 Operational Constraints

- 24/7 availability.
- Support 500+ concurrent users.
- Offline modes for key functions.
- Peak load during evenings.

2.4.6 Security and Safety Constraints

- RBAC, encryption.
- Audit logs.
- Breach prevention.
- Emergency protocols.

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. Each functional requirement uses "The system shall..." format, with unique IDs for traceability. Performance requirements are included per module.

3.1 External Interfaces

As per IEEE standard, this subsection specifies all interfaces.

3.1.1 User Interfaces

The system shall provide a responsive web UI using React.js, with role-based dashboards, forms (e.g., gate pass request with fields for date, reason), and reports (PDF/CSV export). UI shall be intuitive, with tooltips, validation, and mobile optimization. Colors and layouts shall follow university branding.

3.1.2 Hardware Interfaces

The system shall interface with biometric devices via API or file import (e.g., USB for data transfer). Server hardware shall support at least 8GB RAM, 4-core CPU. Client devices: Any with browser.

3.1.3 Software Interfaces

The system shall use MongoDB for data (schemas for students, rooms, etc.). Backend: Node.js/Express for APIs (e.g., POST /api/attendance). Frontend: React for dynamic updates. OS compatibility: Linux server, cross-client.

3.1.4 Communications Interfaces

The system shall use HTTPS for all traffic. APIs: REST with JSON. Notifications: SMTP for email, optional Twilio for SMS. Network: Ethernet/WiFi, with fallback for offline.

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 or malfunctioning, requiring reason entry. 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/SMS 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 or SMS within 10 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 5 minutes.

ATT-PERF-002 Absentee generation shall complete within 60 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 (¡2 seconds lag).

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

3.3 Gate Pass Management Module

This module digitalizes permissions, with workflows and security integrations for traceability.

3.3.1 Functional Requirements

GP-REQ-001: Gate Pass Request Submission The system shall allow students to submit gate pass requests specifying destination, purpose, departure time, expected return time, and contact details. Description: Form with date pickers, validation for return ¿ departure.

GP-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.

GP-REQ-003: Gate Pass Approval Workflow The system shall route gate pass requests first to the assistant warden for initial approval, then to the warden for final approval, with timeouts for escalation. Description: Email links for approval.

GP-REQ-004: Gate Pass Rejection The system shall allow wardens and assistant wardens to reject gate pass requests with mandatory reason entry explaining the rejection, and optional comments. Description: Dropdown for common reasons.

GP-REQ-005: Emergency Gate Pass The system shall provide an emergency gate pass 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.

GP-REQ-006: Gate Pass Expiry The system shall automatically expire gate passes 1 hour after the specified return time and flag the student as overdue, updating status. Description: Job for expiry check.

GP-REQ-007: Gate Pass Verification The system shall provide security personnel with a gate pass verification interface that validates passes using QR codes or pass ID numbers, showing student photo. Description: Mobile app or web for gates.

GP-REQ-008: Gate Pass Activation The system shall activate an approved gate pass only when the student physically exits through the gate and security personnel scan or enter the pass ID, recording time. Description: Geo-location optional.

GP-REQ-009: Return Time Recording The system shall record the actual return time when security personnel verify the student's return using the gate pass ID, calculating duration. Description: Auto-alert if late.

GP-REQ-010: Overdue Alerts The system shall send automated alerts to wardens when students exceed their approved return time by more than 1 hour, with escalation after 2 hours. Description: Multi-channel alerts.

GP-REQ-011: Gate Pass History The system shall maintain a complete history of all gate passes issued to each student, including timestamps for request, approval, exit, and return, with search. Description: Filtered views.

GP-REQ-012: Bulk Gate Pass for Events The system shall allow wardens to create bulk gate passes for university-organized events, specifying event name, date, and list of participating students from CSV. Description: Group approval.

GP-REQ-013: Gate Pass Cancellation The system shall allow students to cancel pending gate pass requests and wardens to revoke approved but unused gate passes, with notifications. Description: Status update.

GP-REQ-014: Gate Pass Statistics The system shall generate statistics showing total gate passes issued, average processing time, rejection rate, and overdue rate for administrative review, with charts. Description: Monthly reports.

GP-REQ-015: Concurrent Gate Pass Limit The system shall enforce a maximum of 3 pending gate pass requests per student at any given time, preventing abuse. Description: Check on submission.

GP-REQ-016: Gate Pass Extension Request The system shall allow students to request extensions for active passes, routed for approval. Description: Limited to once per pass.

GP-REQ-017: Integration with Attendance The system shall link gate passes to attendance, marking as on-leave if approved. Description: Auto-sync.

GP-REQ-018: QR Code Generation The system shall generate QR codes for approved passes, printable or mobile-displayable. Description: Secure encoding.

GP-REQ-019: Pass Usage Logs The system shall log all verifications for audit. Description: Timestamped.

GP-REQ-020: Custom Pass Types The system shall support different pass types (e.g., medical, family), with specific workflows. Description: Configurable.

3.3.2 Performance Requirements

GP-PERF-001 Verification shall complete within 2 seconds.

GP-PERF-002 Notifications within 5 minutes.

GP-PERF-003 Form load within 3 seconds.

GP-PERF-004 Statistics generation within 10 seconds.

GP-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, floor, 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, with maps. 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, urgency level, description, and optional photo attachments (max 5MB). Description: Categorization (plumbing, electrical).

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 Mess Management Module

Handles dining operations with financial and complaint tracking for quality improvement.

3.5.1 Functional Requirements

MESS-REQ-001: Menu Publication The system shall allow mess committee members to publish daily and weekly mess menus specifying breakfast, lunch, dinner, snack items with descriptions, calories, allergens. Description: Calendar interface.

MESS-REQ-002: Menu Scheduling The system shall support scheduling menus up to 2 weeks in advance and automatically display the current day's menu to students, with revisions tracked. Description: Version control.

MESS-REQ-003: Menu Notifications The system shall send daily menu notifications to all hostel students at 8:00 AM each morning via email or in-app notification, with opt-out. Description: Personalized.

MESS-REQ-004: Complaint Submission The system shall allow students to submit mess complaints selecting type (food quality, cleanliness, service, billing, other), description, and rating (1-5 stars). Description: Anonymous option.

MESS-REQ-005: 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.

MESS-REQ-006: 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.

MESS-REQ-007: Complaint Assignment The system shall route complaints to appropriate mess committee members based on type, with load balancing for multiple members. Description: Assignment rules.

MESS-REQ-008: Complaint Status Updates The system shall allow mess committee members to update status (Received, Under Review, Resolved, Closed) with comments and evidence. Description: History log.

MESS-REQ-009: 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.

MESS-REQ-010: Student Mess Account The system shall maintain a mess account for each student showing current balance, transaction history, outstanding dues, with monthly statements. Description: Secure viewing.

MESS-REQ-011: Meal Transaction Recording The system shall allow recording meal transactions automatically debiting standard rates from accounts, with barcode scanning option. Description: Batch processing.

MESS-REQ-012: Custom Transaction Entry The system shall allow adding custom transactions (special items, penalties, refunds) with description, amount, and approval if over threshold. Description: Audit required.

MESS-REQ-013: Balance Notifications The system shall send notifications when balance falls below PKR 500 or negative, with payment reminders. Description: Weekly summaries.

MESS-REQ-014: Payment Recording The system shall record payments, update balances, generate receipts, supporting cash, bank transfer, online modes. Description: Integration with payment gateway.

MESS-REQ-015: Mess Reports The system shall generate reports on revenue, dues, complaints, meal patterns, with filters and exports. Description: Analytics.

MESS-REQ-016: Menu Feedback Collection The system shall collect daily feedback on menus via quick polls. Description: Ratings aggregated.

MESS-REQ-017: Vendor Management The system shall track mess vendors and supplies (optional extension). Description: Inventory basics.

MESS-REQ-018: Dues Recovery Workflow The system shall flag overdue accounts and restrict services if needed. Description: Escalation to warden.

MESS-REQ-019: Menu Customization The system shall allow special menus for events/holidays. Description: Approval required.

MESS-REQ-020: Integration with Room Module The system shall link mess accounts to room occupants for billing. Description: Auto-sync.

3.5.2 Performance Requirements

MESS-PERF-001 Menu display within 2 seconds.

MESS-PERF-002 Complaint ack within 60 seconds.

MESS-PERF-003 Balance updates within 1 second.

MESS-PERF-004 Reports within 15 seconds.

MESS-PERF-005 Notifications within 30 seconds.

3.6 Visitor and Day Scholar Tracking Module

Ensures secure access control with logging and alerts.

3.6.1 Functional Requirements

VDS-REQ-001: Day Scholar Entry Logging The system shall allow security to log day scholar entries by ID scan, recording time, purpose, visited resident. Description: Quick form.

VDS-REQ-002: Visitor Registration The system shall capture visitor details: name, contact, address, relationship, purpose, ID upload. Description: Form with validation.

VDS-REQ-003: Visit Approval Workflow The system shall route requests for approval if required, with instant for low-risk. Description: Configurable rules.

VDS-REQ-004: Entry Verification The system shall generate digital pass with QR for verification. Description: Scan-based.

VDS-REQ-005: Time-Limit Enforcement The system shall enforce 2-hour limit, flag overdues. Description: Timer job.

VDS-REQ-006: Exit Logging The system shall log exits using pass ID, calculate duration. Description: Auto-close.

VDS-REQ-007: Unauthorized Entry Alert The system shall alert on unauthorized attempts in real-time. Description: Siren integration optional.

VDS-REQ-008: Pre-Approved Visit Requests The system shall allow pre-approvals for recurring visits. Description: Schedule-based.

VDS-REQ-009: Visit History Viewing The system shall show history for students/wardens. Description: Filtered.

VDS-REQ-010: Overdue Visit Notifications The system shall notify after 15 minutes overdue. Description: Multi-user.

VDS-REQ-011: Blacklist Management The system shall maintain blacklists, block entries. Description: Admin editable.

VDS-REQ-012: Visit Statistics Dashboard The system shall display stats: visits, durations, peaks. Description: Charts.

VDS-REQ-013: Visitor Reports Generation The system shall generate filtered reports. Description: Exports.

VDS-REQ-014: Emergency Contact Linkage The system shall link to student emergencies. Description: Quick access.

VDS-REQ-015: Offline Entry Support The system shall support offline logging, sync later. Description: Local storage.

VDS-REQ-016: Visitor Photo Capture The system shall capture photos at entry. Description: Webcam integration.

VDS-REQ-017: Group Visitor Support The system shall handle groups (e.g., family). Description: Linked records.

VDS-REQ-018: Visit Purpose Categories The system shall categorize purposes for reporting. Description: Dropdowns.

VDS-REQ-019: Integration with Security Cameras The system shall log with camera timestamps (future). Description: API ready.

VDS-REQ-020: Audit for Visits The system shall log all actions. Description: Compliance.

3.6.2 Performance Requirements

VDS-PERF-001 Logging within 3 seconds.

VDS-PERF-002 Alerts within 10 seconds.

VDS-PERF-003 Reports within 15 seconds for 6 months.

VDS-PERF-004 Offline sync within 1 minute online.

VDS-PERF-005 Dashboard load within 5 seconds.

4 Other Non-Functional Requirements

4.1 Security Requirements

The system shall implement RBAC with JWT authentication, encrypt data (AES-256), maintain audit trails (retained 1 year), and comply with GDPR-like standards. Penetration testing required. Measurable: No unauthorized access in tests.

4.2 Reliability Requirements

99.9% uptime, daily backups, 4-hour recovery. Fault tolerance with redundancies. Measurable: MTBF ¿ 1000 hours.

4.3 Usability Requirements

Intuitive UI, ¡2 hours training, 95% task completion rate in user tests. Responsive design, accessibility (screen readers).

4.4 Supportability Requirements

Modular code, documentation, API docs. Easy updates, logging for debugging.

4.5 Portability Requirements

Cloud-agnostic, browser-compatible.

4.6 Maintainability Requirements

Code coverage ¿80%, modular design.

A Use Case Diagram

The Use Case Diagram illustrates actors (Warden, Student, Security, etc.) and use cases (Mark Attendance, Request Gate Pass, etc.). It reflects all functional requirements.

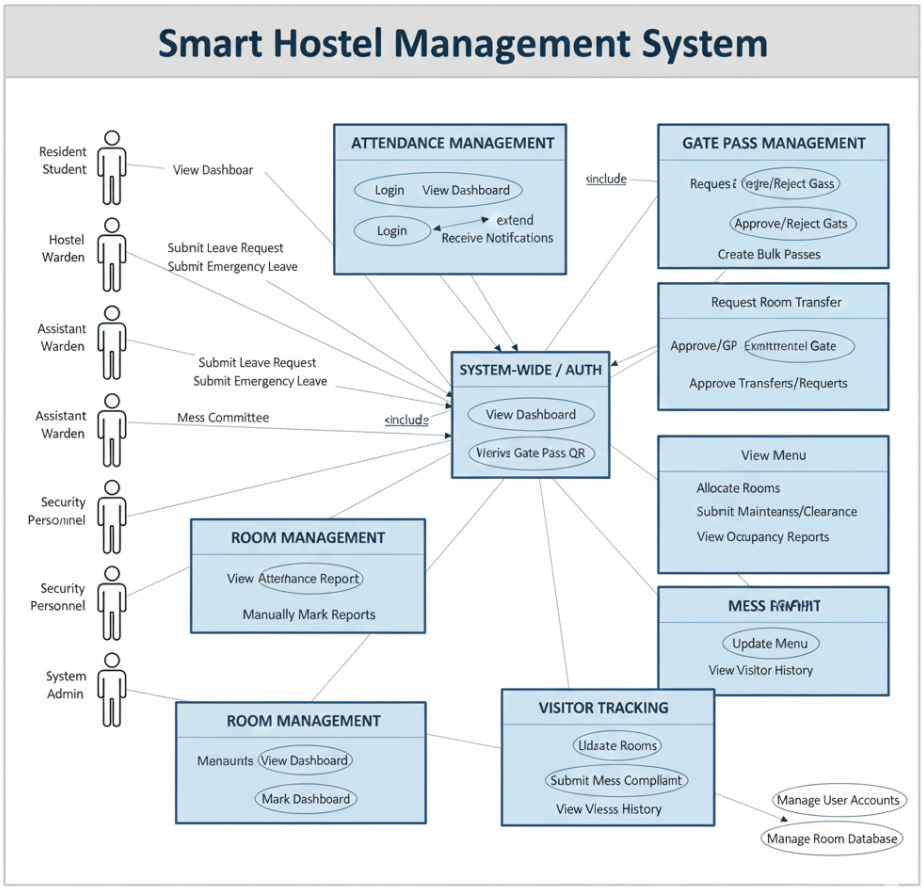

Figure 1: Use Case Diagram for SHMS

Detailed description: Actors interact with modules via extend/include relationships.

B Context Diagram

The Context Diagram shows system boundary with external entities (Biometric System, Student DB, Users).

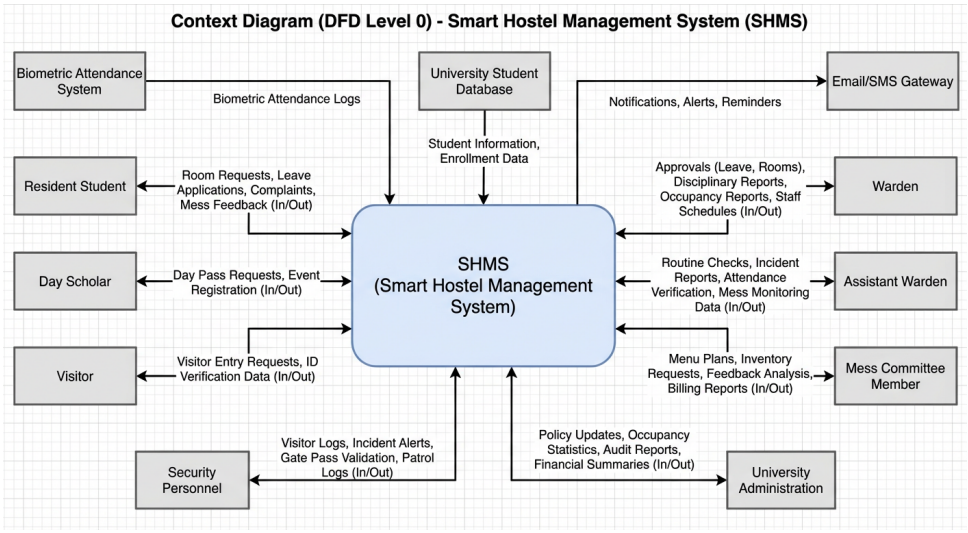

Figure 2: Context Diagram for SHMS

Detailed description: Data flows like attendance import, notifications.