

Software Design 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

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Project Overview	4
1.3	Design Approach	4
1.4	Document Organization	5
2	Design Assumptions and Constraints	5
2.1	Design Assumptions	5
2.1.1	Infrastructure Assumptions	5
2.1.2	User Assumptions	5
2.1.3	Data and Integration Assumptions	5
2.2	Design Constraints	6
2.2.1	Technical Constraints	6
2.2.2	Performance Constraints	6
2.2.3	Security Constraints	6
2.2.4	Operational Constraints	6
3	Key Design Decisions	7
3.1	Architectural Decisions	7
3.1.1	Three-Tier Architecture	7
3.1.2	RESTful API Design	7
3.1.3	Modular System Design	7
3.2	Database Design Decisions	8
3.2.1	MongoDB Selection	8
3.2.2	Collection Design	8
3.3	Security Design Decisions	8
3.3.1	JWT Authentication	8
3.3.2	Role-Based Access Control	9
3.4	User Interface Design Decisions	9
3.4.1	Mobile-First Approach	9
3.4.2	Dashboard-Centric Design	9
4	System Design Diagrams	10
4.1	Use Case Diagram	10
4.2	Data Flow Diagrams	11
4.2.1	DFD Level 0 - Context Diagram	11
4.2.2	DFD Level 1	11
4.2.3	DFD Level 2 - Attendance Management	13
4.2.4	DFD Level 2 - Entry/Exit Management	13
4.2.5	DFD Level 2 - Room Management	15
4.2.6	DFD Level 2 - Day Scholar Tracking	17
4.3	Sequence Diagrams	18
4.3.1	Attendance Marking via Biometric	18

4.3.2	Entry/Exit Request and Approval	20
4.3.3	Gate Verification and Return Logging	22
4.3.4	Room Allocation	24
4.3.5	Day Scholar Entry and Exit	25
4.4	Activity Diagrams	27
4.4.1	Attendance Workflow	27
4.4.2	Entry/Exit Workflow	28
4.4.3	Room Allocation or Transfer Workflow	30
4.4.4	Complaint Handling Workflow	31
4.4.5	Day Scholar Visit Workflow	33
4.5	Class Diagram	35
4.6	Component Diagram	36
5	Requirements-Design Traceability	37
6	Figma Prototype and Validation	39
6.1	Prototype Overview	39
6.2	Prototype Coverage	39
6.2.1	Student Portal	39
6.2.2	Warden Portal	39
6.2.3	Security Portal	40
6.3	Prototype Screenshots	41
6.3.1	Login Screen	41
6.3.2	Student Dashboard	41
6.3.3	Entry/Exit Application Form	42
6.3.4	Warden Dashboard	42
6.3.5	Warden Assistant	43
6.3.6	Admin Dashboard	43
6.3.7	Security Personnel	44
7	Meeting Documentation (Milestone 3)	45
7.1	Meeting 1: Paper-Based Prototype Validation	45
7.1.1	Prototype Demonstration	45
7.1.2	Discussion Summary	45
7.1.3	Feedback and Observations	46
7.2	Meeting 2: Interactive Prototype Review and Design Validation	46
7.2.1	Meeting Agenda	46
7.2.2	Prototype Demonstration	47
7.2.3	Discussion Summary	47
7.2.4	Key Feedback	47
7.2.5	Final Approval	48
8	Conclusion	49
8.1	Design Completeness	49
8.2	Stakeholder Validation	49

8.3 Next Steps	49
8.4 Acknowledgments	50
A Glossary	51
B References	51
C Design Tools Used	51
D GitHub Repository Link	52

1 Introduction

1.1 Purpose

This document presents the complete system design for the Smart Hostel Management System (SHMS). It translates the functional and non-functional requirements from the approved SRS into a detailed architectural and behavioral design that will guide the implementation phase.

This document is intended for:

- Development team to understand system architecture and component interactions
- Requirement Provider (Ms. Nida Sultan Nahra) to validate design completeness
- Quality assurance team to create test cases based on design specifications
- Future maintainers to understand system structure

1.2 Project Overview

SHMS is a web-based application designed to automate hostel operations at Namal University. The system replaces manual processes with an integrated digital platform handling:

- Automated biometric attendance tracking
- Online entry/exit application and approval workflow
- Room allocation and management
- Complaint submission and tracking
- Day scholar access control

1.3 Design Approach

Our design follows these key principles:

- **Requirements-Driven:** Every design element traces back to SRS requirements
- **Modular Architecture:** Independent modules with clear interfaces
- **User-Centered:** Interface designs prioritize ease of use
- **Iterative Validation:** Design validated through prototyping and stakeholder feedback

1.4 Document Organization

- Section 2: Design assumptions and constraints
- Section 3: Key architectural decisions
- Section 4: System design diagrams (UML models)
- Section 5: Requirements-design traceability
- Section 6: Figma prototype and validation
- Section 7: Meeting documentation
- Section 8: Conclusion and next steps

2 Design Assumptions and Constraints

2.1 Design Assumptions

2.1.1 Infrastructure Assumptions

- University network provides stable connectivity (99% uptime during 6 AM - 11 PM)
- Server infrastructure available (on-premises or cloud: 8GB RAM, 4-core CPU, 100GB storage)
- Biometric devices support API integration
- Students have access to modern web browsers on smartphones/computers

2.1.2 User Assumptions

- Students possess basic smartphone usage skills
- Staff will receive system training before deployment
- Users will actively adopt the digital system
- All students have enrolled fingerprints in biometric system

2.1.3 Data and Integration Assumptions

- University provides API access to student information system
- Student data from university is accurate and updated
- Email server (SMTP) access available

- Budget allocated for critical SMS notifications

2.2 Design Constraints

2.2.1 Technical Constraints

- Technology stack: React.js (frontend), Node.js/Express (backend), MongoDB (database)
- Must support all major browsers without browser-specific code
- Fully responsive design (320px to 1920px screen width)
- Limited by existing biometric device API capabilities

2.2.2 Performance Constraints

- All user interactions must complete within 3 seconds
- System must handle 100+ concurrent users
- Biometric import must complete hourly (6 PM - 11:59 PM)
- Reports must generate within 15 seconds
- Dashboard updates must reflect within 2 seconds

2.2.3 Security Constraints

- Strict role-based access control required
- All sensitive operations must be logged
- Female hostel data requires special security measures
- Strong password policies must be enforced

2.2.4 Operational Constraints

- Development timeline: One academic semester (4 months)
- Small team: 3 developers
- Limited budget: Use open-source technologies
- System updates during non-peak hours only

3 Key Design Decisions

3.1 Architectural Decisions

3.1.1 Three-Tier Architecture

Decision: Adopt a three-tier architecture separating Presentation (React.js), Business Logic (Node.js/Express), and Data (MongoDB) layers.

Justification:

- Separation of concerns enables independent development and testing
- Supports multiple client types through shared API
- Facilitates maintenance by isolating changes to specific layers
- Aligns with team expertise and industry best practices

3.1.2 RESTful API Design

Decision: Implement RESTful APIs for all client-server communication using JSON format.

Justification:

- Industry-standard with extensive documentation
- Stateless communication simplifies scalability
- Enables potential future mobile app integration
- JSON provides lightweight, readable data exchange

3.1.3 Modular System Design

Decision: Organize system into five independent modules: Attendance, Entry/Exit, Room Management, Complaint Management, Day Scholar Tracking.

Justification:

- Each module addresses distinct SRS requirements
- Enables parallel development
- Facilitates testing and maintenance
- Reduces risk by isolating failures

3.2 Database Design Decisions

3.2.1 MongoDB Selection

Decision: Use MongoDB (NoSQL document database) over relational database.

Justification:

- Flexible schema accommodates evolving requirements
- JSON-like documents align with JavaScript stack
- Horizontal scalability for future growth
- Team familiarity reduces learning curve

3.2.2 Collection Design

Decision: Create separate collections for major entities (Students, Attendance, EntryExitApplications, Rooms, Complaints, DayScholarLogs) with embedded sub-documents for tightly coupled data.

Justification:

- Separate collections enable efficient querying and indexing
- Embedding reduces joins for common queries
- Balances normalization with MongoDB best practices

3.3 Security Design Decisions

3.3.1 JWT Authentication

Decision: Implement JWT (JSON Web Token) based authentication with role information in token.

Justification:

- Stateless authentication supports RESTful design
- Role embedding enables efficient authorization
- Industry-standard with mature libraries
- Scalable across multiple servers

3.3.2 Role-Based Access Control

Decision: Implement RBAC with four roles: Student, Security, Assistant Warden, Warden. Permissions checked at both API and UI levels.

Justification:

- Directly implements SRS security requirements
- Simple model matches organizational hierarchy
- Dual enforcement (API + UI) provides defense in depth

3.4 User Interface Design Decisions

3.4.1 Mobile-First Approach

Decision: Design UI with mobile devices as primary target, then adapt for larger screens.

Justification:

- Most students primarily use smartphones
- Ensures critical workflows work on smallest screens
- Progressive enhancement for desktop experience

3.4.2 Dashboard-Centric Design

Decision: Provide role-specific dashboards as landing pages.

Justification:

- Reduces navigation overhead
- Presents important information immediately
- Supports situational awareness for wardens

4 System Design Diagrams

This section presents all UML diagrams modeling the system's structure and behavior.

4.1 Use Case Diagram

The Use Case Diagram provides a high-level view of system functionality and actor interactions.

Key Actors:

- **Student:** Marks attendance, submits entry/exit applications, requests room changes, submits complaints
- **Hostel Warden:** Approves requests, views reports, manages rooms, monitors operations
- **Assistant Warden:** view attendace, assists in room management
- **Security Personnel:** Verifies biometric entry/exit, logs day scholar access
- **Biometric System:** Automated attendance and gate entry/exit recording

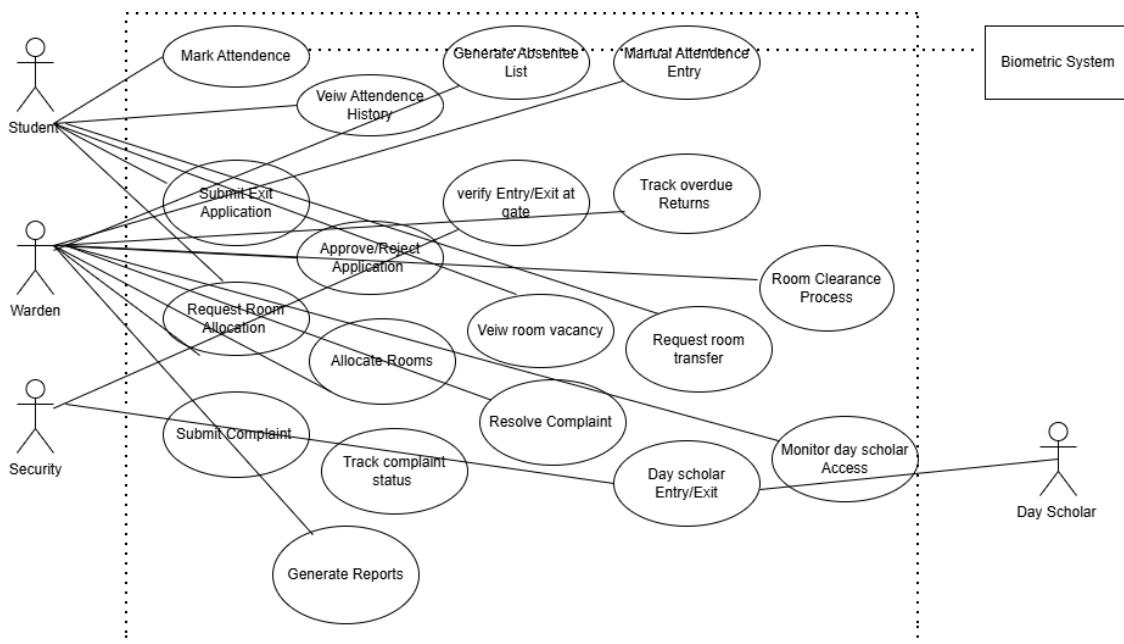


Figure 1: System Use Case Diagram

4.2 Data Flow Diagrams

Data Flow Diagrams show how data moves through the system at multiple abstraction levels.

4.2.1 DFD Level 0 - Context Diagram

The Context Diagram shows SHMS as a single process with all external entities.

Major Data Flows:

- Students ↔ SHMS: Applications, queries, notifications, confirmations
- Warden ↔ SHMS: Approvals, manual entries, reports, alerts
- Security ↔ SHMS: Verifications, day scholar logs
- Biometric System → SHMS: Attendance data, entry/exit logs
- University DB → SHMS: Student profiles, enrollment data

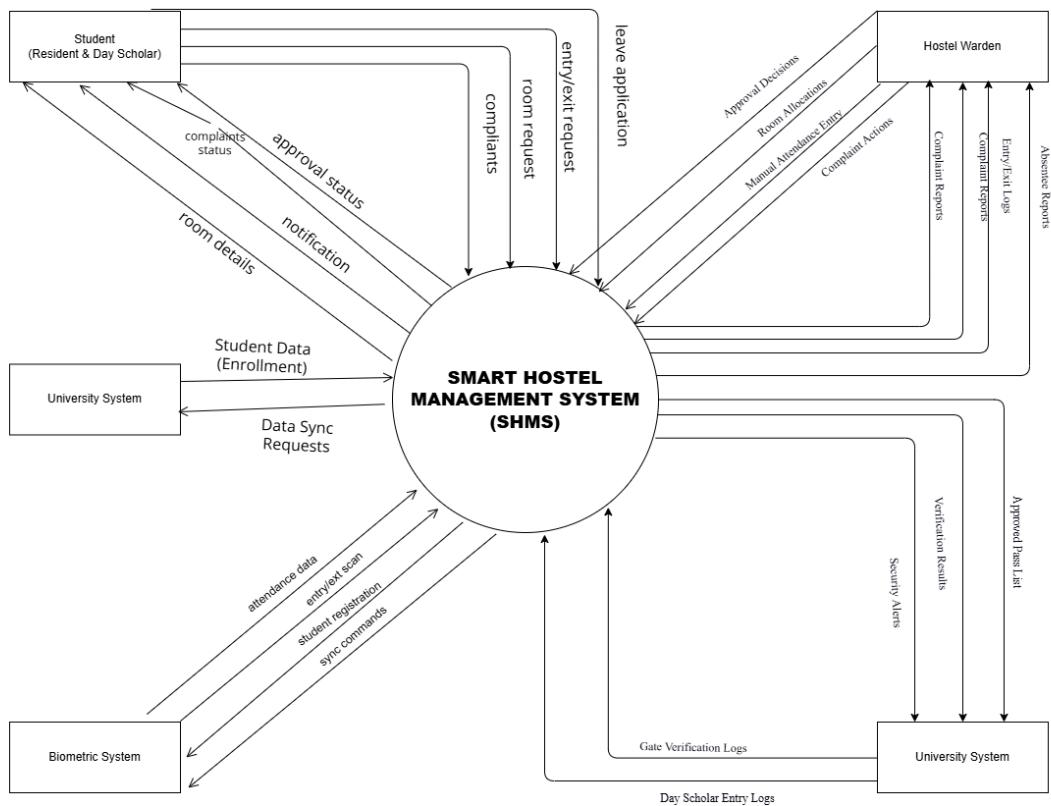


Figure 2: DFD Level 0, Context Diagram

4.2.2 DFD Level 1

Level 1 DFD decomposes the system into five major processes.

Major Processes:

- 1.0 Manage Attendance
- 2.0 Manage Entry/Exit
- 3.0 Manage Rooms
- 4.0 Manage Complaints
- 5.0 Manage Day Scholars

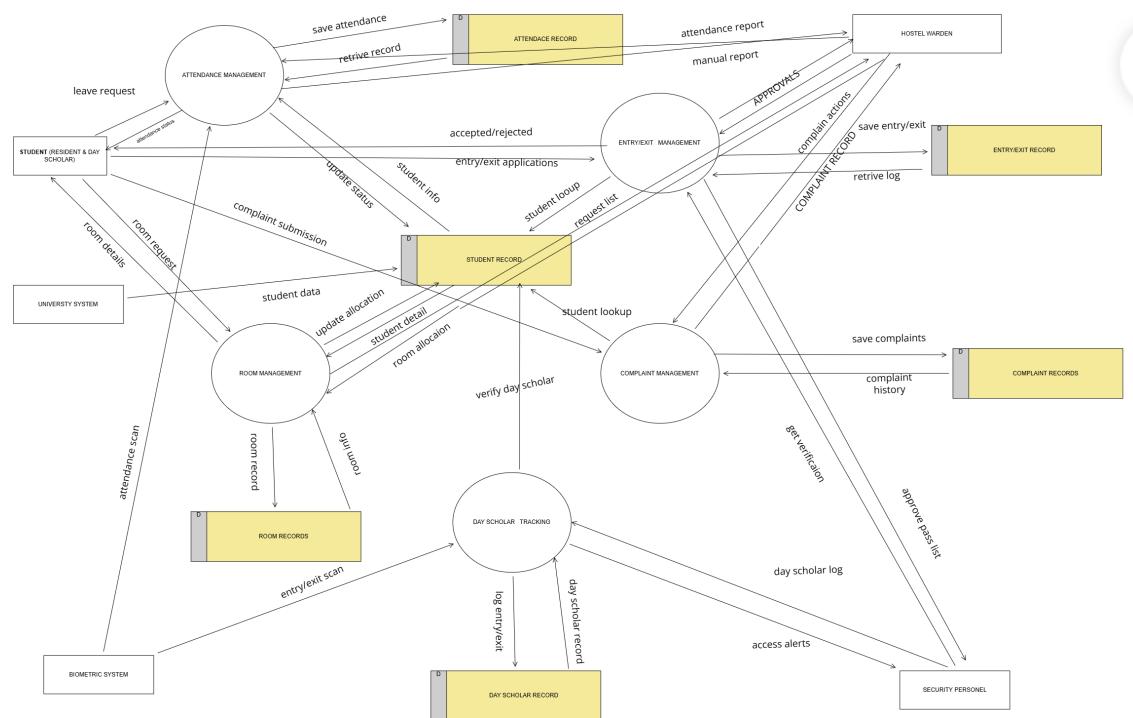


Figure 3: DFD Level 1

4.2.3 DFD Level 2 - Attendance Management

Sub-Processes:

- 1.1 Import Biometric Data
- 1.2 Validate Attendance
- 1.3 Generate Absentee List
- 1.4 Send Notifications
- 1.5 Process Leave Requests
- 1.6 Generate Reports

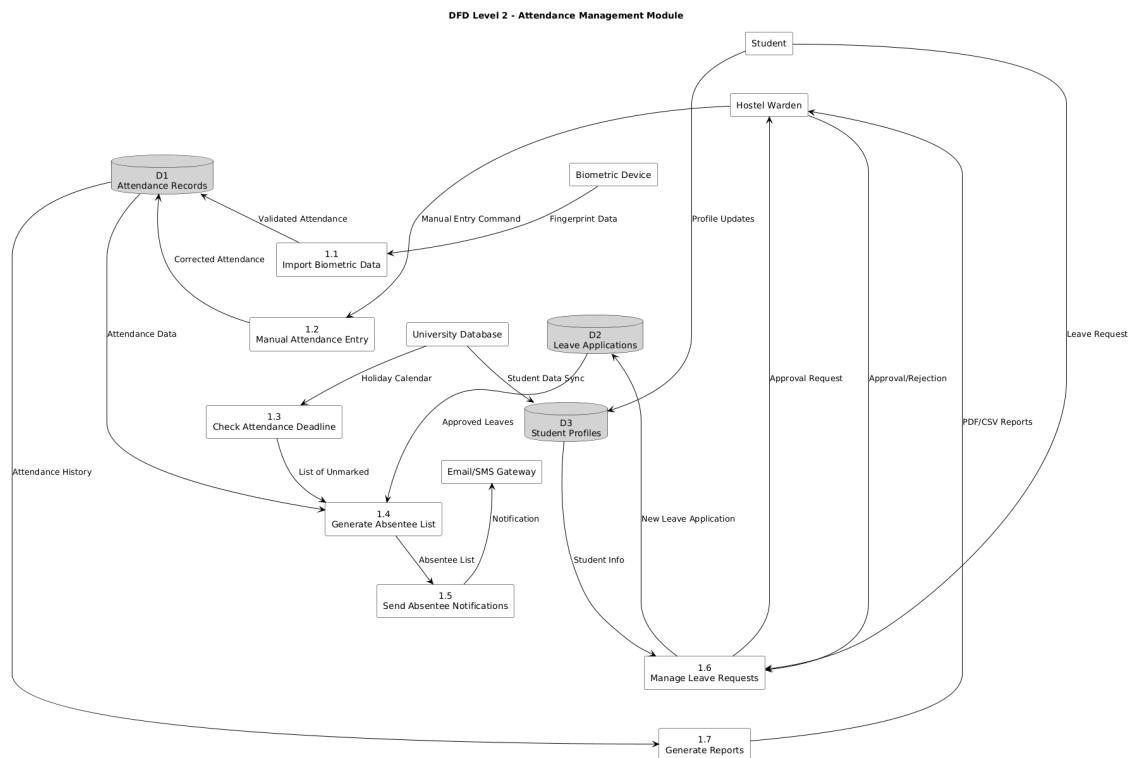


Figure 4: DFD Level 2 - Attendance Management

4.2.4 DFD Level 2 - Entry/Exit Management

Sub-Processes:

- 2.1 Submit Application
- 2.2 Route for Approval

- 2.3 Process Approval
- 2.4 Verify at Gate
- 2.5 Record Exit/Entry
- 2.6 Monitor Return Time
- 2.7 Send Alerts

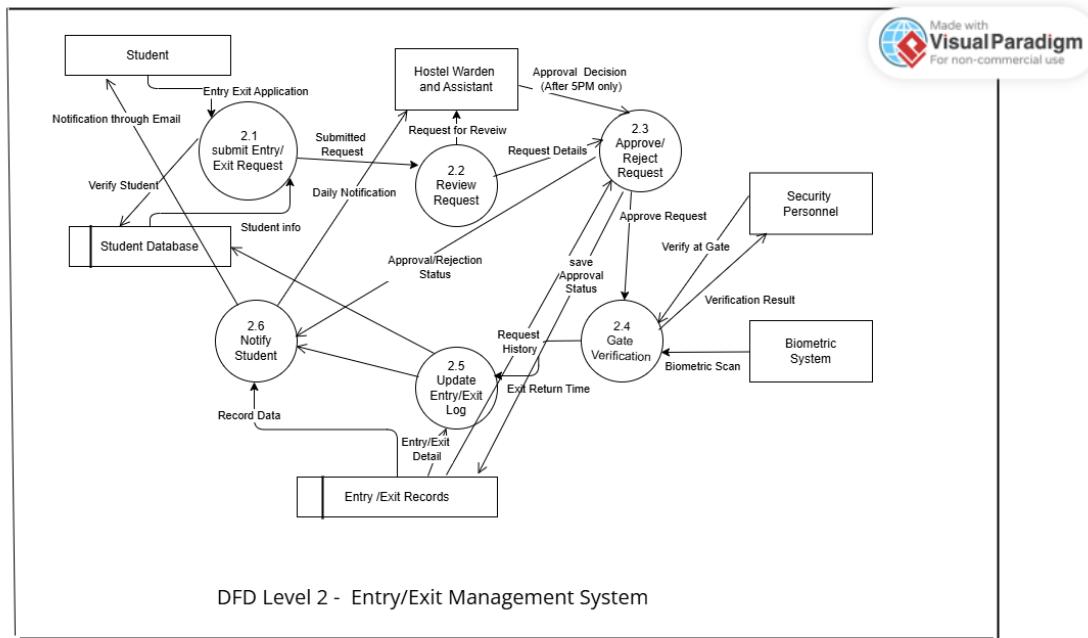


Figure 5: DFD Level 2 - Entry/Exit Management

4.2.5 DFD Level 2 - Room Management

Sub-Processes:

- 3.1 Process Room Allocation
- 3.2 Handle Room Transfers
- 3.3 Process Clearance
- 3.4 Handle Maintenance Requests
- 3.5 Update Room Status

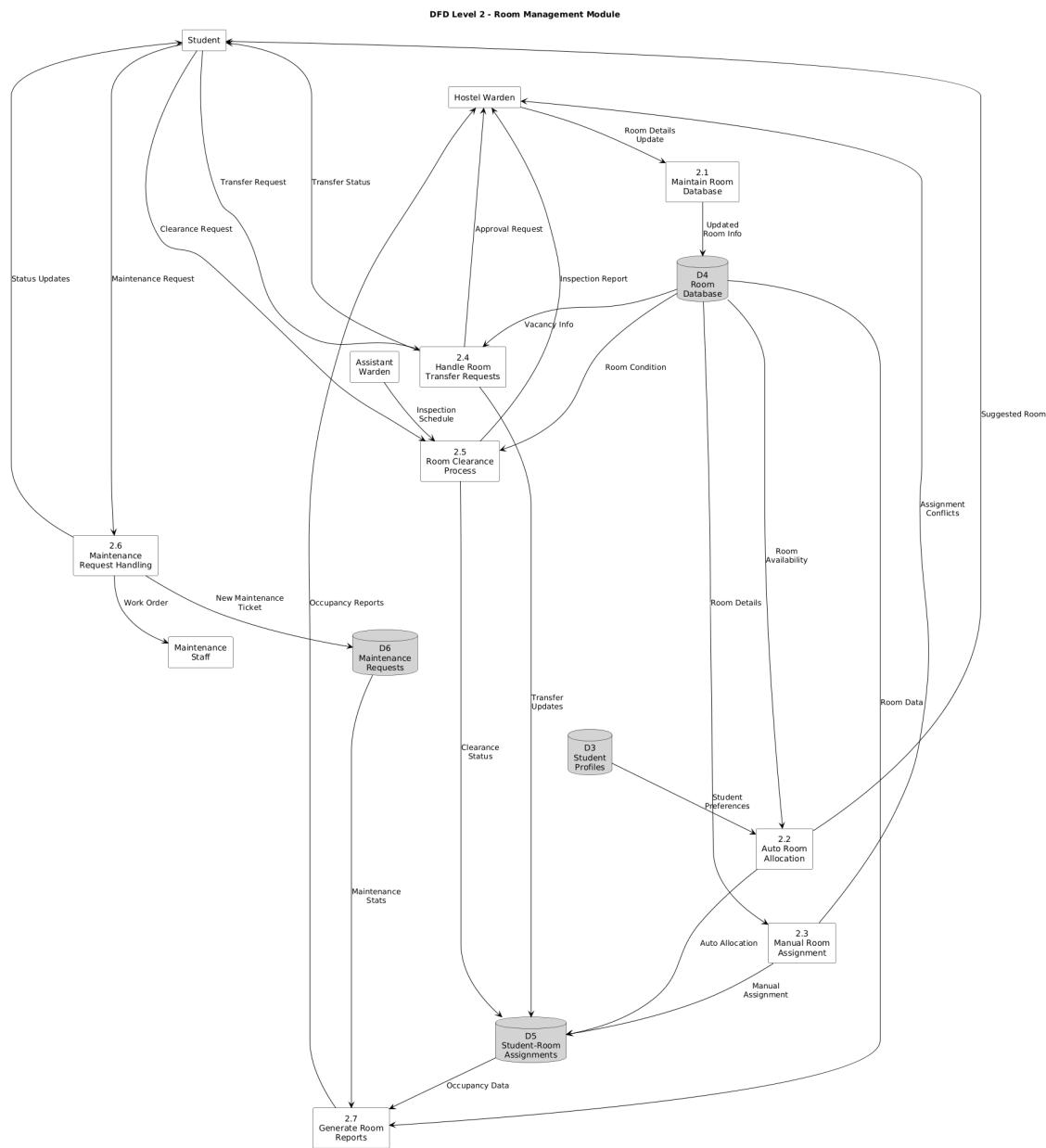


Figure 6: DFD Level 2 - Room Management

4.2.6 DFD Level 2 - Day Scholar Tracking

Sub-Processes:

- 5.1 Log Entry
- 5.2 Verify Entry
- 5.3 Monitor Duration
- 5.4 Send Alerts
- 5.5 Log Exit

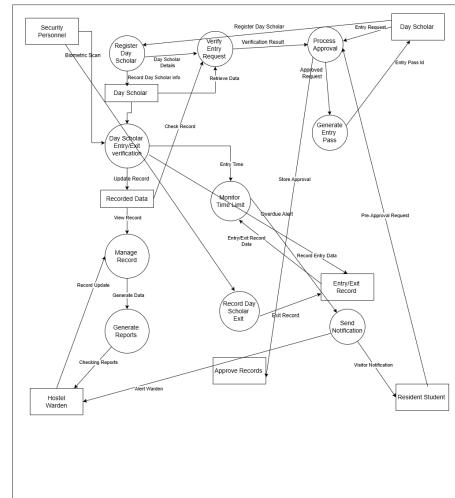


Figure 7: DFD Level 2 - Day Scholar Tracking

4.3 Sequence Diagrams

Sequence diagrams model object interactions over time for key scenarios.

4.3.1 Attendance Marking via Biometric

Main Flow:

1. Student scans fingerprint on biometric device
2. Biometric system sends data to Attendance Controller
3. Controller validates student ID and timestamp
4. Service saves attendance record to database
5. System returns confirmation

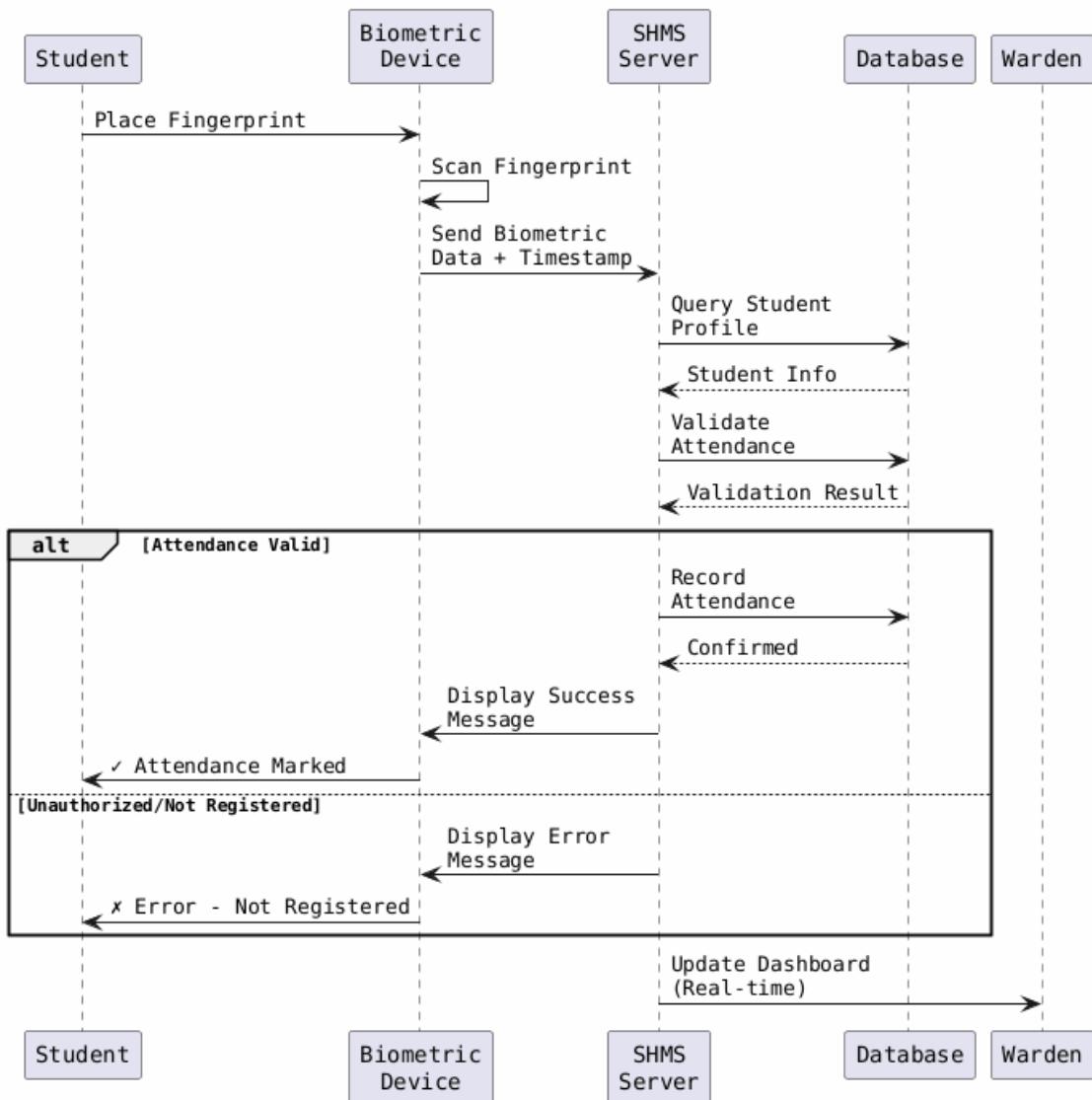


Figure 8: Sequence Diagram - Biometric Attendance Marking

4.3.2 Entry/Exit Request and Approval

Main Flow:

1. Student submits entry/exit application
2. System creates application and routes to Assistant Warden
3. Assistant Warden approves
4. System routes to Warden for final approval
5. Warden approves
6. System notifies student of approval

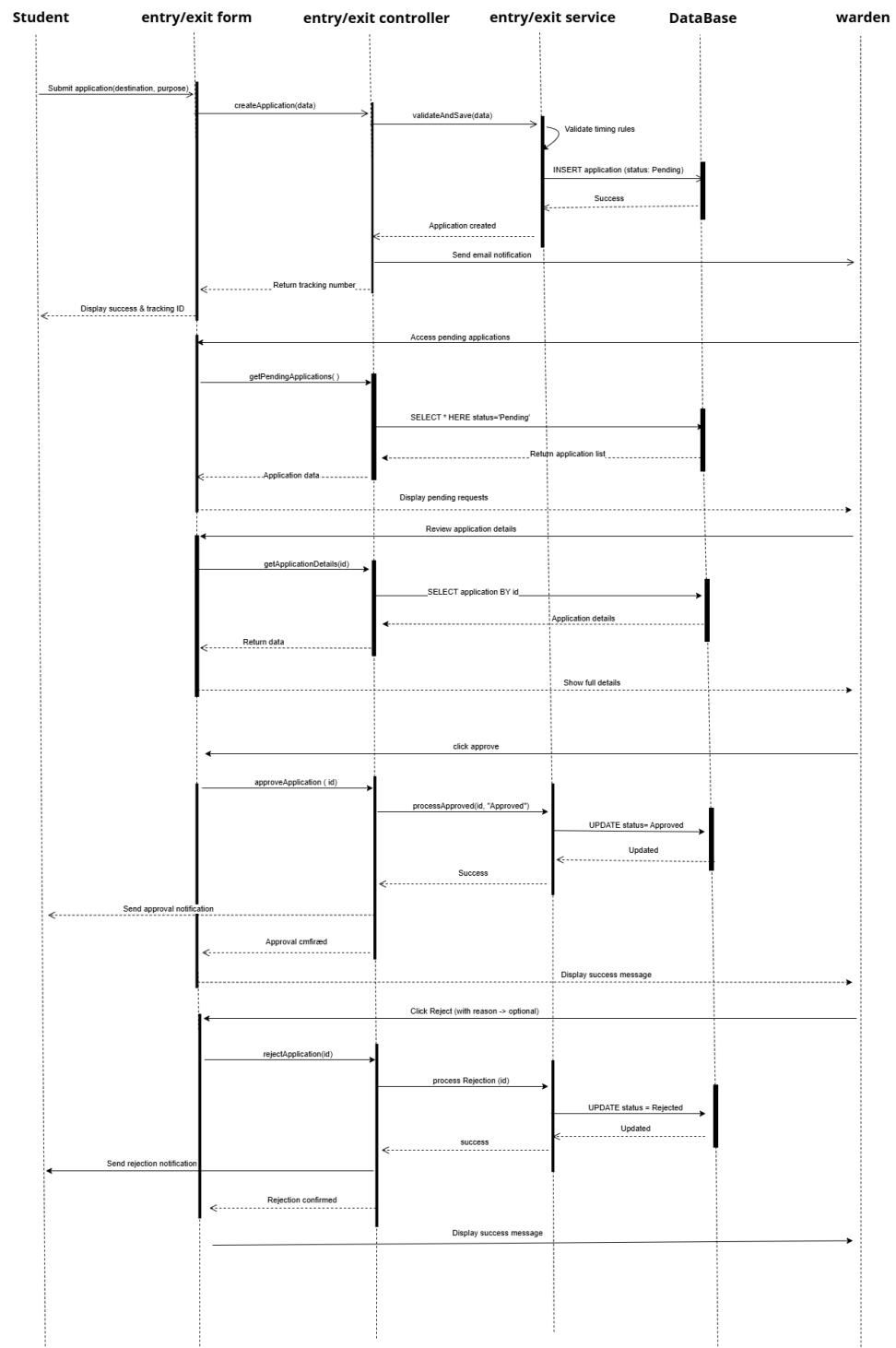


Figure 9: Sequence Diagram - Entry/Exit Request and Approval

4.3.3 Gate Verification and Return Logging

Main Flow:

1. Security enters application ID
2. System retrieves and displays application details
3. Student scans fingerprint for verification
4. Biometric system confirms identity
5. System records exit/entry time
6. Security receives confirmation

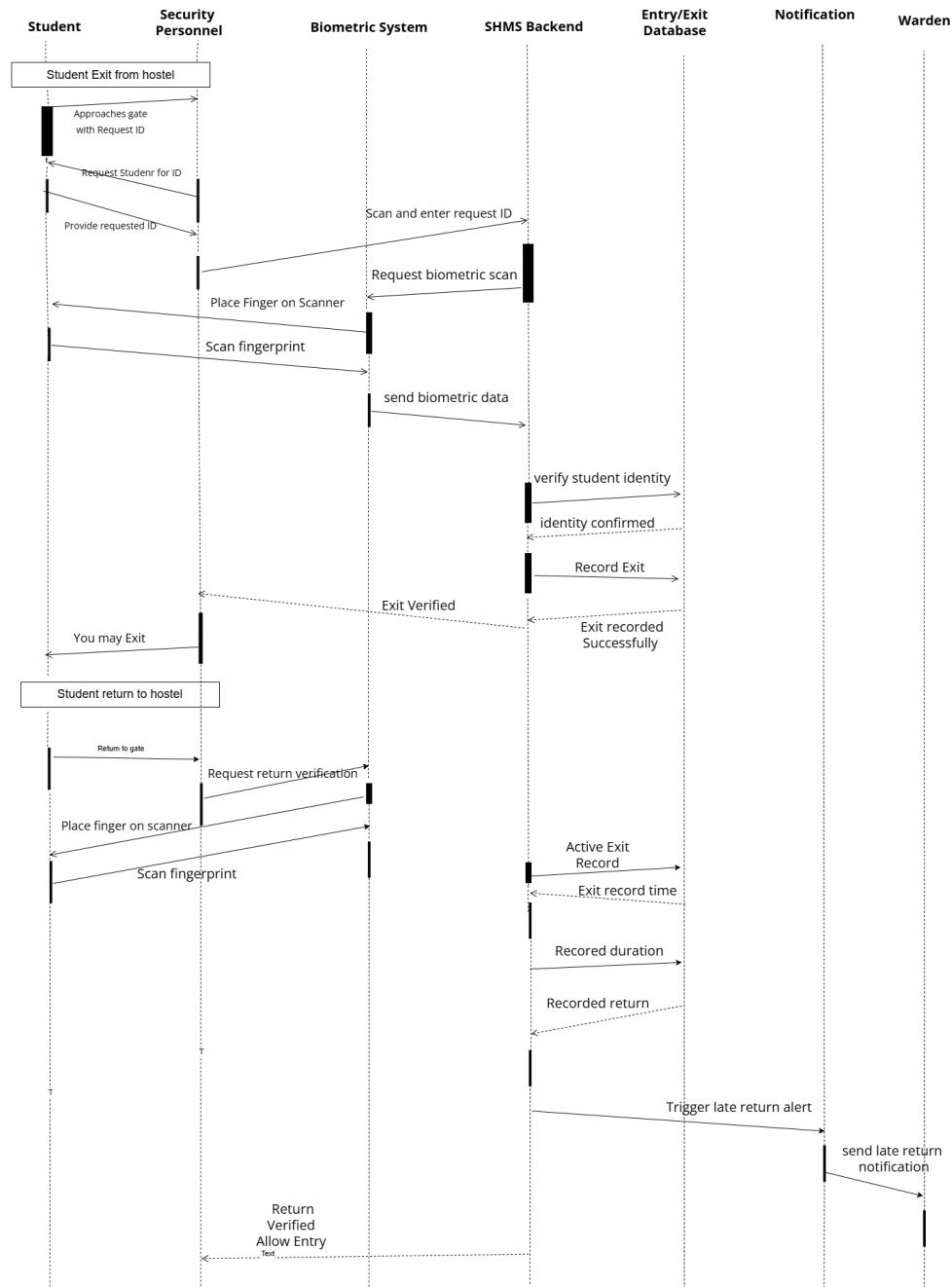


Figure 10: Sequence Diagram - Gate Verification and Return Logging

4.3.4 Room Allocation

Main Flow:

1. Student submits group room allocation request
2. Controller validates group (capacity match, no duplicates)
3. Service checks available rooms
4. System displays matching rooms
5. Student selects preferred room
6. Service assigns room to all group members
7. System sends confirmations to all students

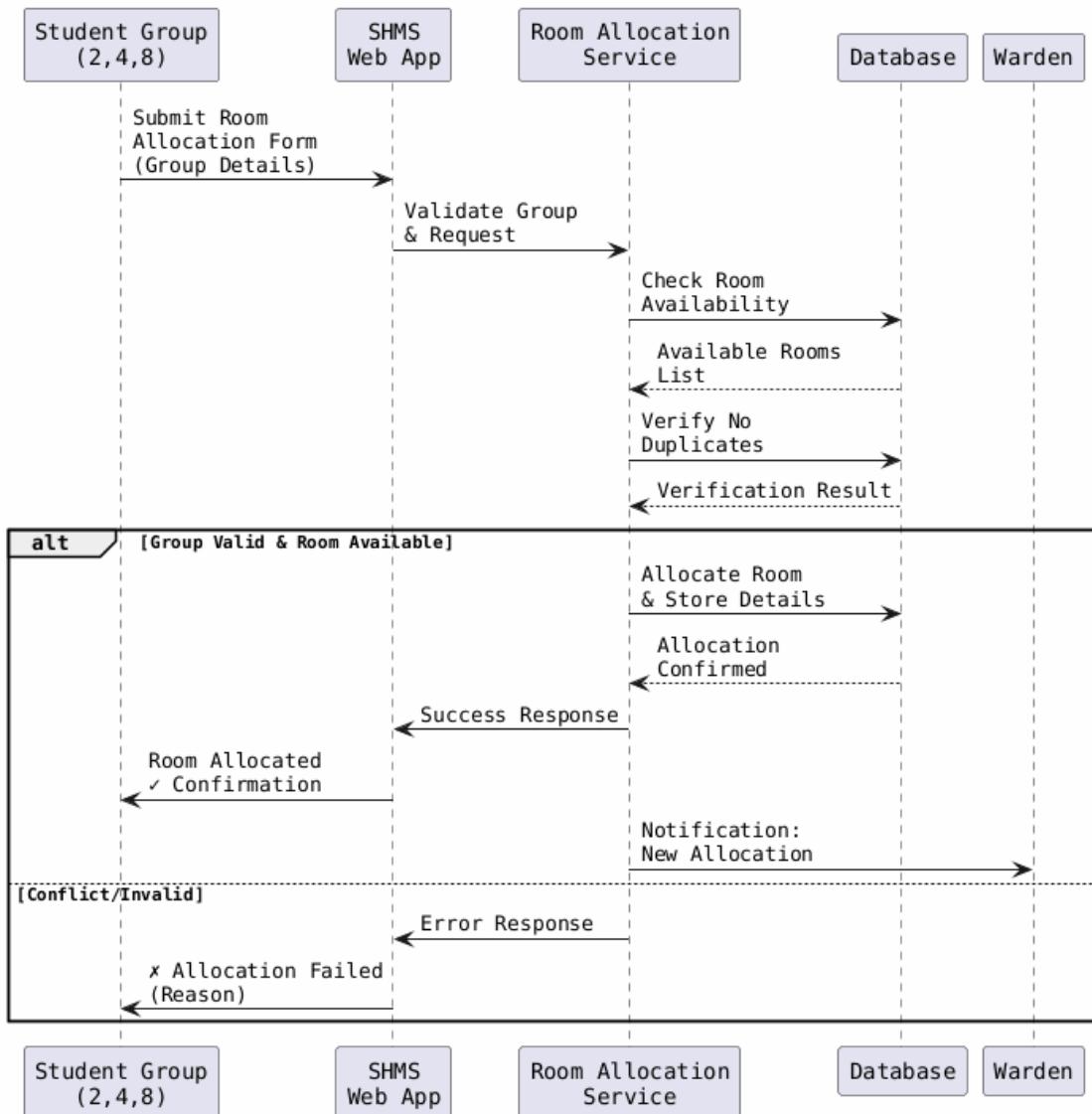


Figure 11: Sequence Diagram - Room Allocation

4.3.5 Day Scholar Entry and Exit

Main Flow:

1. Day scholar arrives at hostel
2. Security logs entry with biometric
3. System records entry time and notifies warden
4. System monitors duration
5. Day scholar exits via biometric scan

6. System logs exit and calculates duration

Made with
 Visual Paradigm
 For non-commercial use

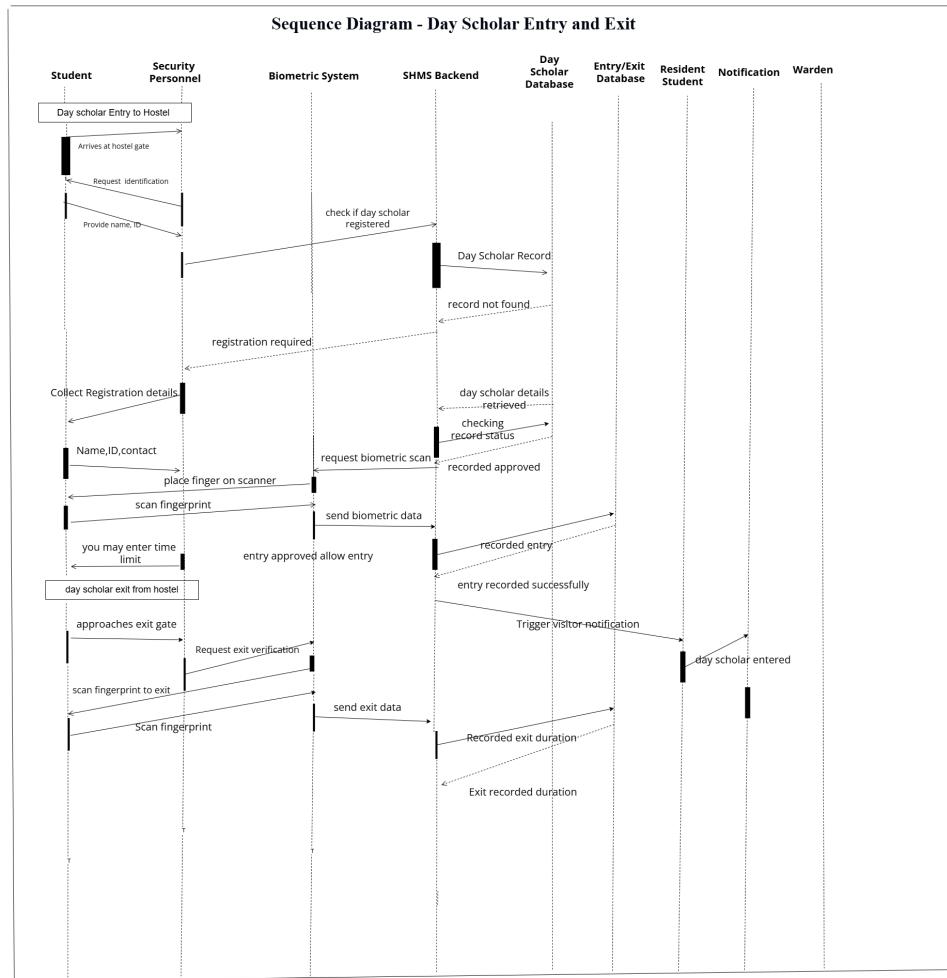


Figure 12: Sequence Diagram - Day Scholar Entry and Exit

4.4 Activity Diagrams

Activity diagrams illustrate workflow control and decision logic.

4.4.1 Attendance Workflow

Swimlanes: Student, Biometric System, SHMS, Warden

Main Steps:

1. Students mark attendance throughout evening (6 PM - 10 PM)
2. System imports biometric data hourly
3. At 10:00 PM, system checks all students
4. System generates absentee list (excludes approved leaves)
5. At 10:05 PM, system sends notification to warden

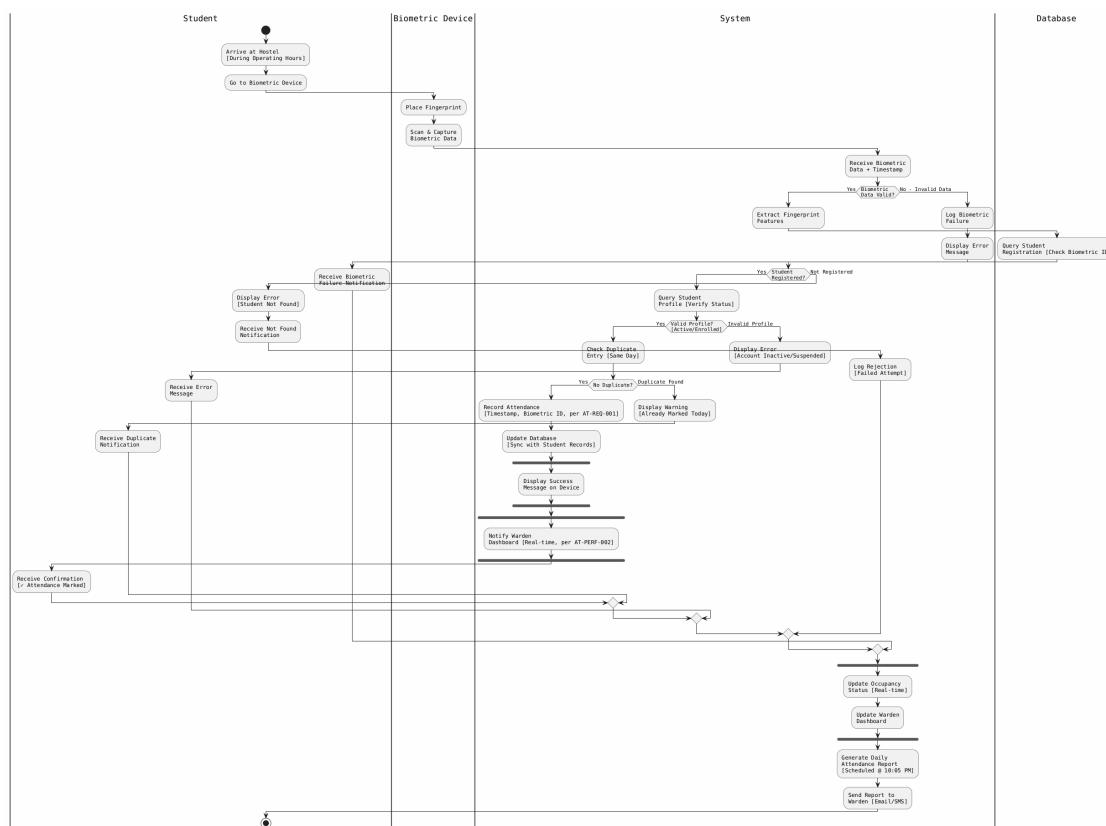


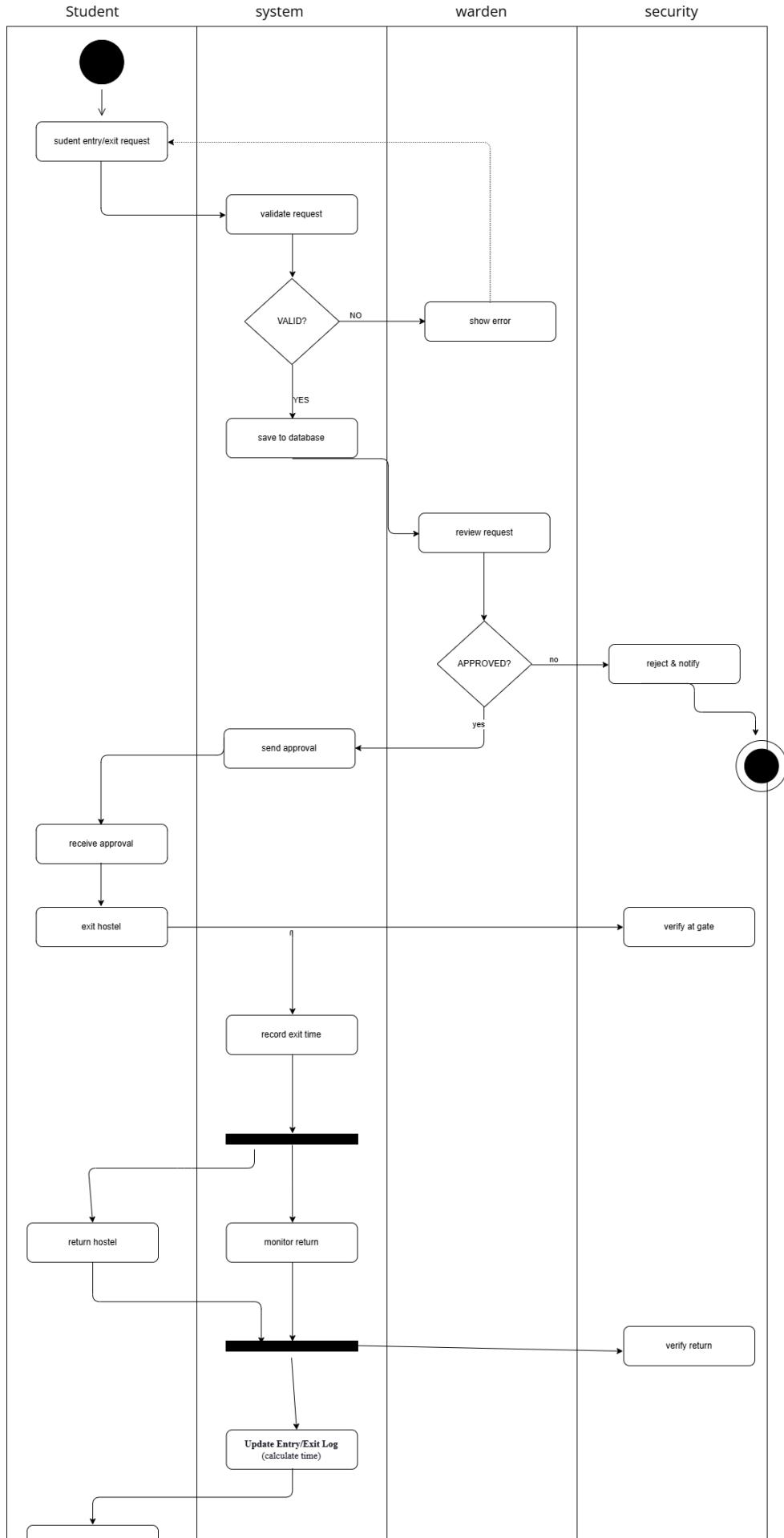
Figure 13: Activity Diagram - Attendance Workflow

4.4.2 Entry/Exit Workflow

Swimlanes: Student, System, Assistant Warden, Warden, Security

Main Steps:

1. Student submits application
2. System validates and routes to Assistant Warden
3. Assistant Warden reviews (approve/reject decision)
4. If approved, routes to Warden
5. Warden final review (approve/reject)
6. If both approved, student notified
7. Student goes to gate for biometric verification
8. Security verifies and system records exit
9. System monitors return and sends alert if overdue



4.4.3 Room Allocation or Transfer Workflow

Swimlanes: Students, System, Warden

Main Steps:

1. Students form group agreeing on room capacity
2. One student submits allocation form with all member details
3. System validates (capacity match, no duplicates)
4. System displays available rooms
5. Student selects room
6. System checks real-time availability
7. System assigns room to all members
8. System sends confirmations and updates occupancy

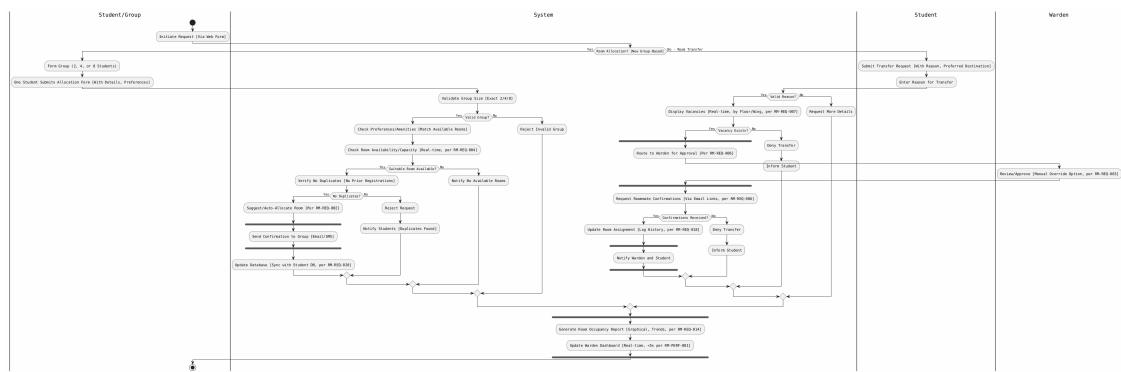


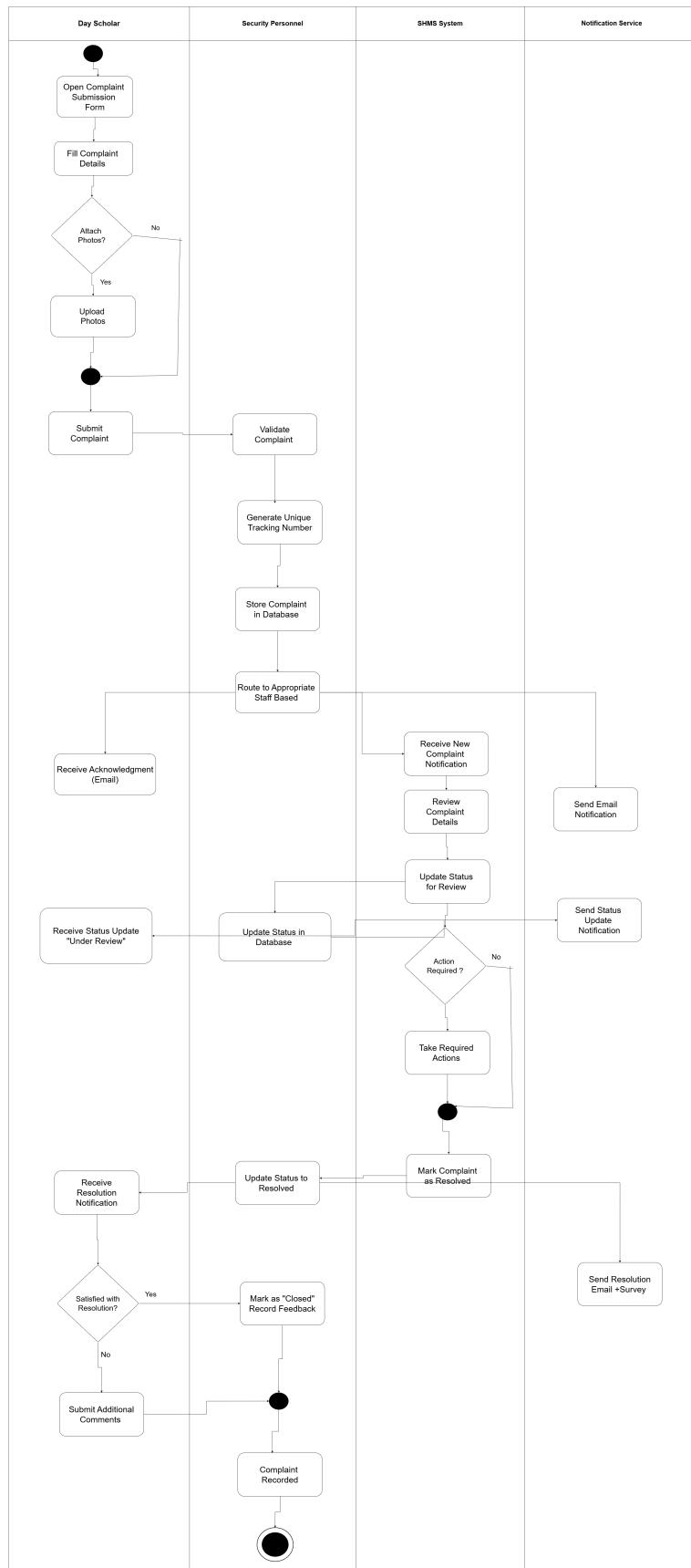
Figure 15: Activity Diagram - Room Allocation or Transfer

4.4.4 Complaint Handling Workflow

Swimlanes: Student, System, Warden, Maintenance Staff

Main Steps:

1. Student submits complaint with category and photos
2. System assigns tracking number
3. System notifies warden immediately
4. Warden reviews and assigns to staff
5. Staff updates status as work progresses
6. Warden marks as resolved
7. System notifies student with resolution details
8. Student can provide feedback



4.4.5 Day Scholar Visit Workflow

Swimlanes: Day Scholar, Security, System, Warden

Main Steps:

1. Day scholar arrives at hostel
2. Security checks if pre-approved
3. If yes, scans biometric and allows entry
4. If no, security submits request
5. System may require warden approval for certain cases
6. If approved, generates time-limited pass
7. Day scholar scans biometric to enter
8. System logs entry and notifies warden
9. System monitors duration and alerts if exceeded
10. Day scholar exits via biometric
11. System records exit and calculates duration

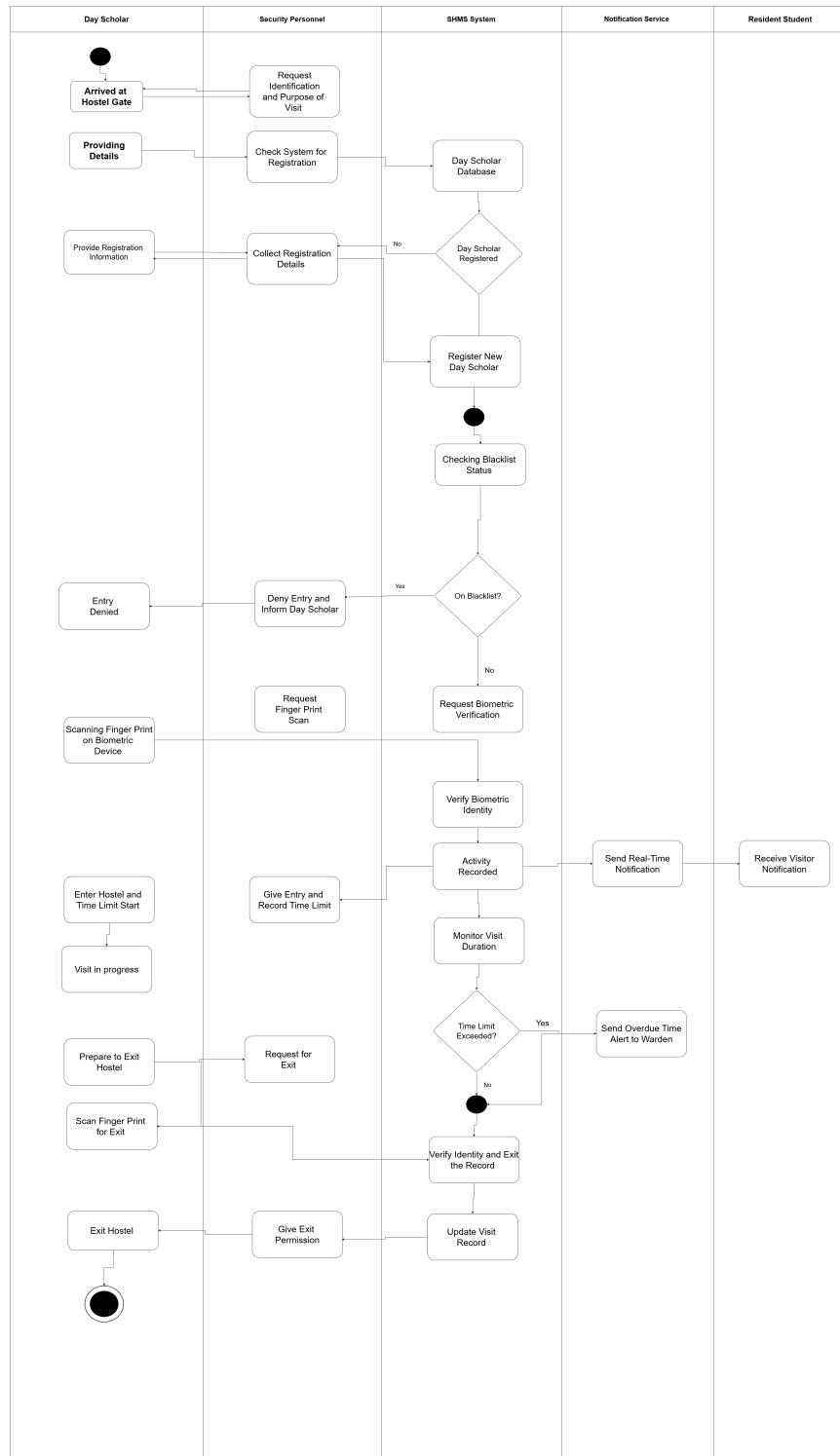


Figure 17: Activity Diagram - Day Scholar Visit Workflow

4.5 Class Diagram

The Class Diagram shows the system's static structure including classes, attributes, operations, and relationships.

Key Packages:

- User Package:** User (abstract), Student, Warden, AssistantWarden, SecurityPersonnel
- Attendance Package:** AttendanceRecord, LeaveRequest, BiometricData
- EntryExit Package:** EntryExitApplication, ApprovalRecord, GateVerification
- Room Package:** Room, Bed, RoomAllocation, TransferRequest, MaintenanceRequest
- Complaint Package:** Complaint, ComplaintCategory, ComplaintStatus
- DayScholar Package:** DayScholarLog, VisitorPass
- Notification Package:** Notification, EmailNotification, SMSNotification

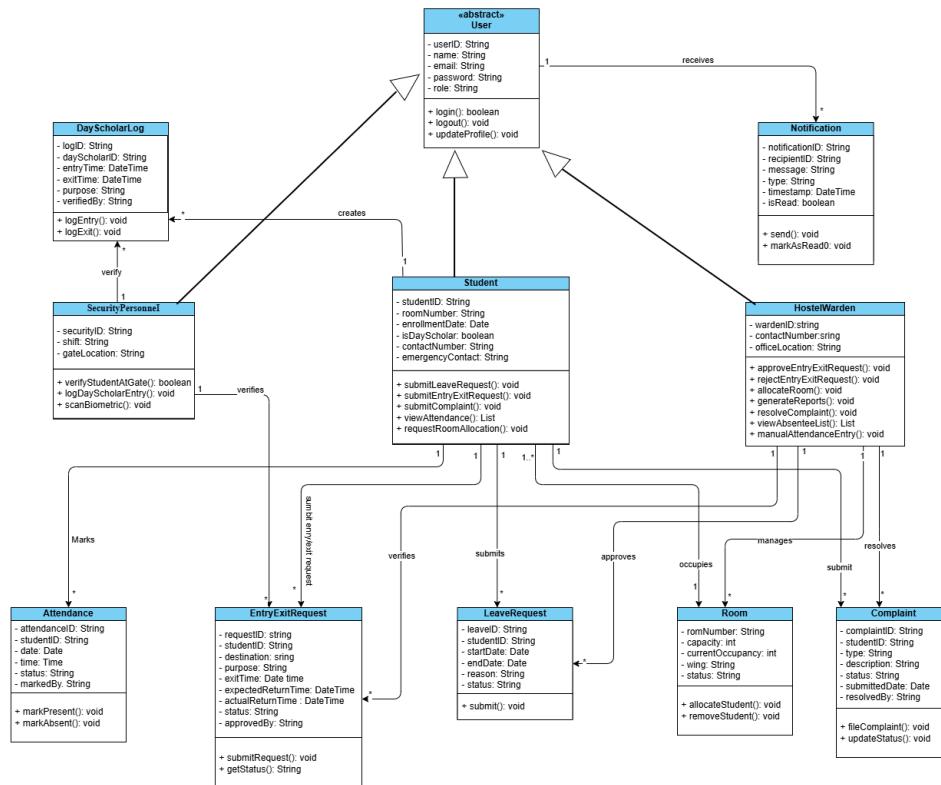


Figure 18: System Class Diagram

4.6 Component Diagram

The Component Diagram illustrates system architecture showing major components and dependencies.

Key Components:

- Presentation Layer:** Dashboard Component, Forms Component, Reports Component
- Business Logic Layer:** Authentication Service, Attendance Service, EntryExit Service, Room Service, Complaint Service, DayScholar Service
- Integration Layer:** Biometric Adapter, Email Service, SMS Service, University DB Adapter
- Data Layer:** Data Access Component, MongoDB Database

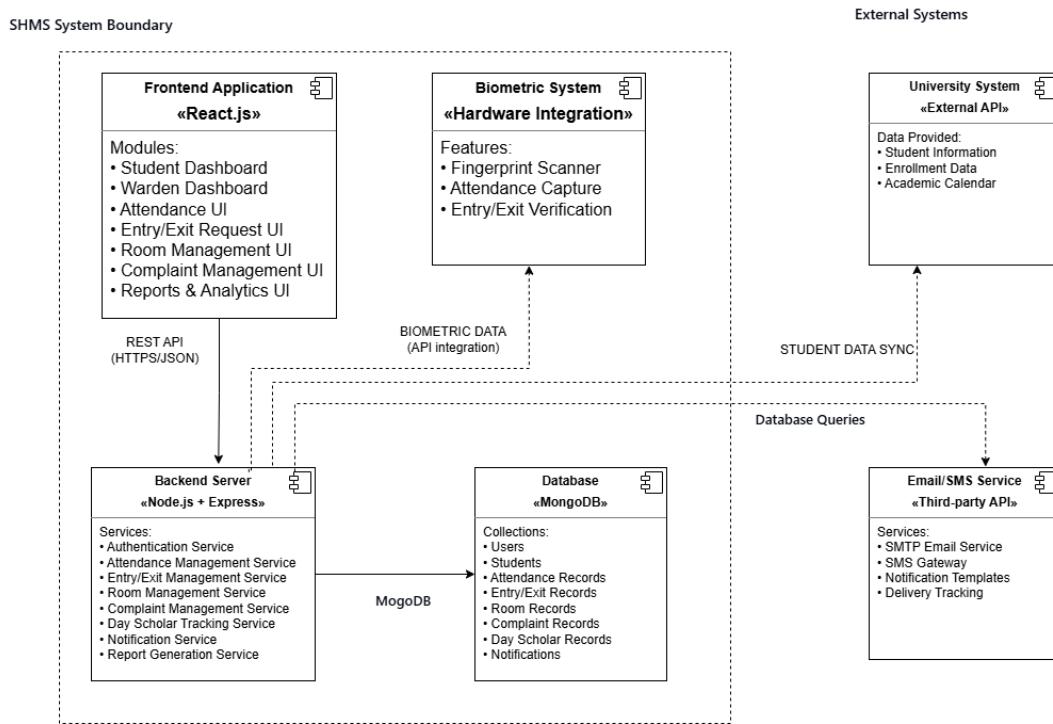


Figure 19: System Component Diagram

5 Requirements-Design Traceability

This section maps SRS requirements to specific design elements, ensuring complete coverage.

Requirement ID	SRS Requirement	Design Element	Diagram Reference
FR-01	Biometric Attendance Marking	Attendance Controller, Biometric Adapter	Fig. 8, 4
FR-02	Hourly Data Import	BiometricData Import Service	Fig. 4
FR-03	Daily Absentee List Generation	Attendance Report Generator	Fig. 13
FR-04	Warden Notification	Notification Service	Fig. 13
FR-05	Entry/Exit Application Submission	EntryExitApplication Form	Fig. 9, 14
FR-06	Dual-Level Approval Workflow	ApprovalRouting Service	Fig. 9, 14
FR-07	Gate Verification System	GateVerification Controller	Fig. 10
FR-08	Return Time Monitoring	ReturnMonitor Service	Fig. 14
FR-09	Late Return Alerts	AlertService	Fig. 14
FR-10	Room Allocation System	RoomAllocation Service	Fig. 11, 15
FR-11	Group-Based Allocation	GroupValidation Logic	Fig. 11
FR-12	Vacancy Viewing	RoomRepository	Fig. 6
FR-13	Room Transfer Requests	TransferRequest Form	Fig. 15
FR-14	Complaint Submission	Complaint Form	Fig. 16
FR-15	Photo Upload Support	FileUpload Component	Fig. 16
FR-16	Warden Notification	ComplaintNotification Service	Fig. 16
FR-17	Status Tracking	ComplaintStatus Tracker	Fig. 16
FR-18	Resolution Notification	ResolutionNotifier	Fig. 16
FR-19	Day Scholar Logging	DayScholarLog Service	Fig. 12, 17
FR-20	Biometric Verification	VisitorVerification Service	Fig. 12

Requirement ID	SRS Requirement	Design Element	Diagram Reference
FR-21	Duration Monitoring	DurationMonitor	Fig. 17
FR-22	Duration Alerts	OverstayAlert Service	Fig. 17
FR-23	Reports Generation	ReportGenerator Service	Fig. 4, 5
FR-24	Email Notifications	EmailService	Fig. 18
FR-25	SMS Alerts	SMSService	Fig. 18
NFR-01	3-Second Response Time	Performance Optimization	Architecture Decisions
NFR-02	100+ Concurrent Users	LoadBalancing, Caching	Component Diagram
NFR-03	Mobile Responsive Design	Responsive CSS Framework	Figma Prototype
NFR-04	99.9% Uptime	Redundant Architecture	Component Diagram
NFR-05	Role-Based Security	JWT, RBAC Implementation	Architecture Decisions
NFR-06	Data Backup	Automated Backup Service	Architecture Decisions
NFR-07	Audit Logging	AuditLog Service	Class Diagram

6 Figma Prototype and Validation

6.1 Prototype Overview

An interactive prototype was developed in Figma covering all major system functionalities with realistic workflows and navigation.

Figma Desktop Prototype Link: <https://www.figma.com/proto/NShhDD17HBESNAPUMK2dFZ/Untitled?node-id=59-7&t=dsS40dtozuPHHvJ9-0&scaling=min-zoom&content-scaling=fixed&page-id=0%3A1&starting-point-node-id=59%3A81>

Figma Android Prototype Link: <https://www.figma.com/proto/hlz7vYwG3SDnBhdV1urny1/Untitled?node-id=3-1099&t=Cijof00ZqVsER2wG-0&scaling=min-zoom&content-scaling=fixed&page-id=0%3A1&starting-point=5%3A1346&show Proto-sidebar=1>

6.2 Prototype Coverage

6.2.1 Student Portal

- Login and authentication
- Dashboard with attendance status
- Entry/exit application submission (complete flow)
- Room allocation request (group-based)
- Room transfer request
- Complaint submission with photo upload
- History viewing (attendance, applications, complaints)

6.2.2 Warden Portal

- Dashboard with key metrics and alerts
- Pending approvals (leave requests, entry/exit, room transfers)
- Attendance monitoring and absentee lists
- Room management (allocation, transfers, clearances)
- Complaint tracking and resolution
- Report generation
- Manual attendance entry

6.2.3 Security Portal

- Gate verification interface
- Day scholar entry logging
- Current entry/exit status dashboard
- Alert viewing

6.3 Prototype Screenshots

6.3.1 Login Screen

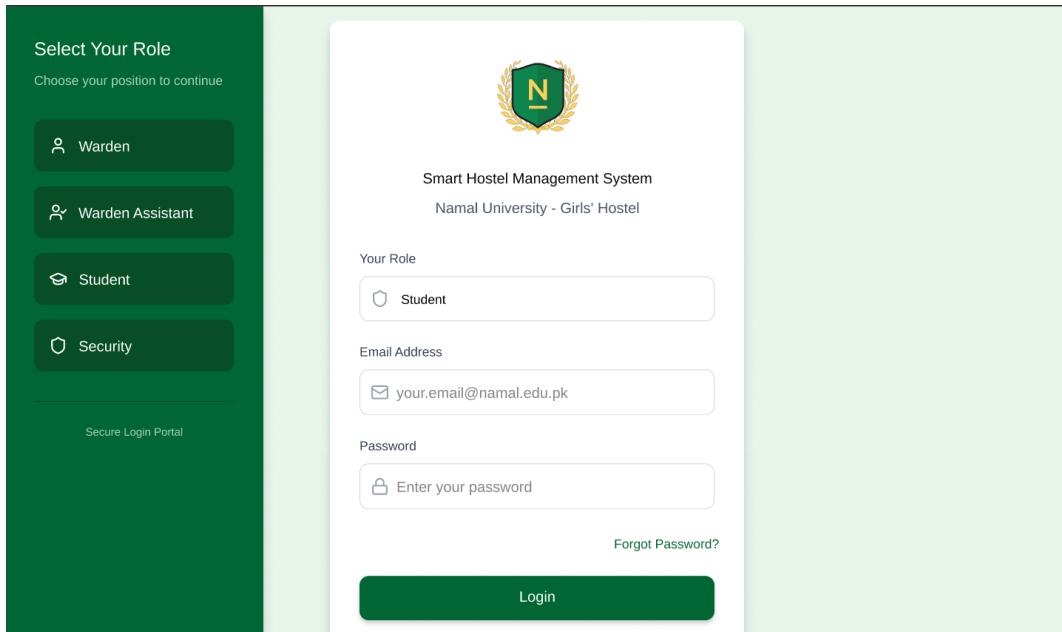


Figure 20: Login Screen

6.3.2 Student Dashboard

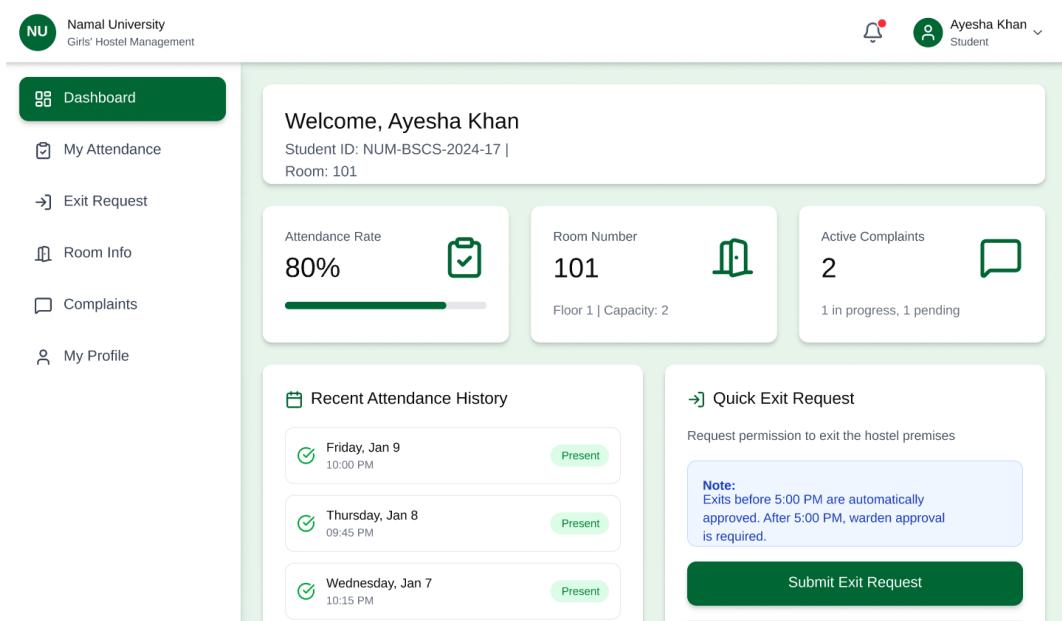


Figure 21: Student Dashboard

6.3.3 Entry/Exit Application Form

The screenshot shows the 'Entry/Exit Tracking' section of the application. On the left sidebar, there are links for 'Dashboard', 'Exit Request' (which is highlighted in green), and other options like 'Attendance', 'Entry/Exit Logs', etc. The main area has a title 'Entry/Exit Tracking' with a sub-instruction 'Request permission to exit and track your entries/exits'. Below this is a 'Quick Exit Request' section with a 'New Request' button. It includes an 'Exit Policy' box with rules: 'Exits before 5:00 PM are automatically approved', 'Exits after 5:00 PM require warden approval', 'Expected return time must be before 10:00 PM', and 'Late returns (after 1 hour) will be flagged'. At the bottom is a table titled 'My Entry/Exit History' showing one record:

Date	Time	Type	Reason	Destination	Return Time	Status
09/01/2026	03:30 PM	Exit	Shopping	City Mall	06:15 PM	approved

Figure 22: Entry/Exit Application Form

6.3.4 Warden Dashboard

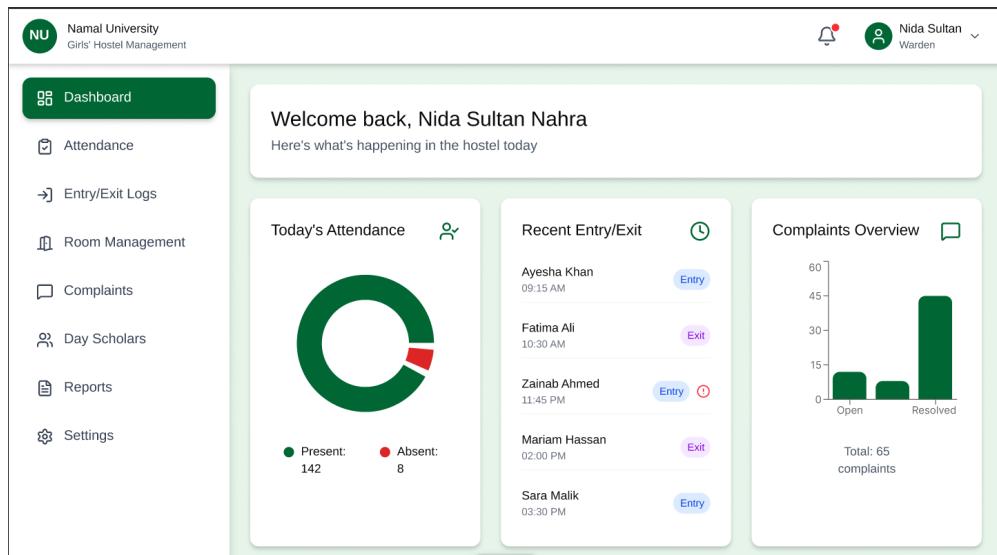


Figure 23: Warden Dashboard

6.3.5 Warden Assistant

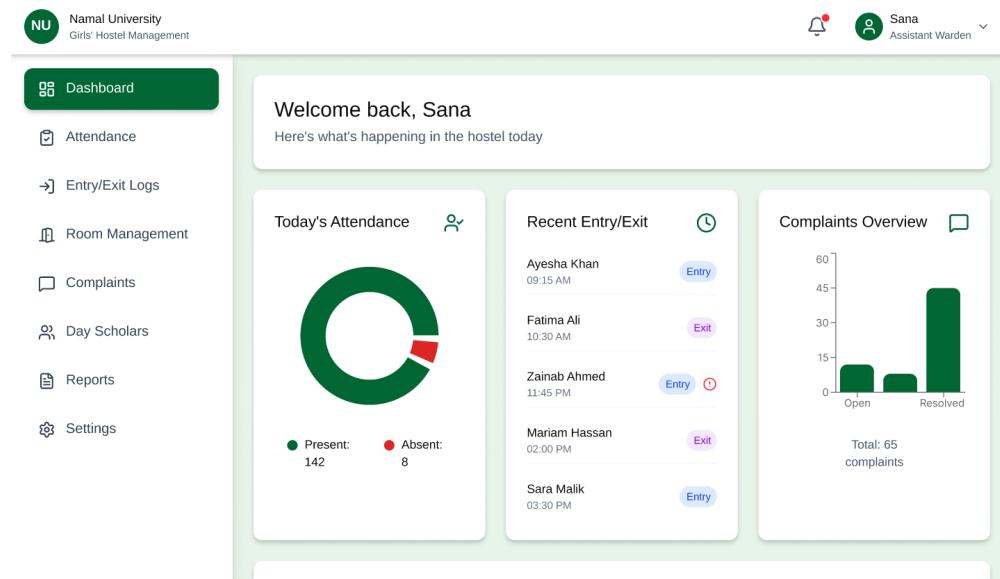


Figure 24: Warden Assistant Dashboard

6.3.6 Admin Dashboard

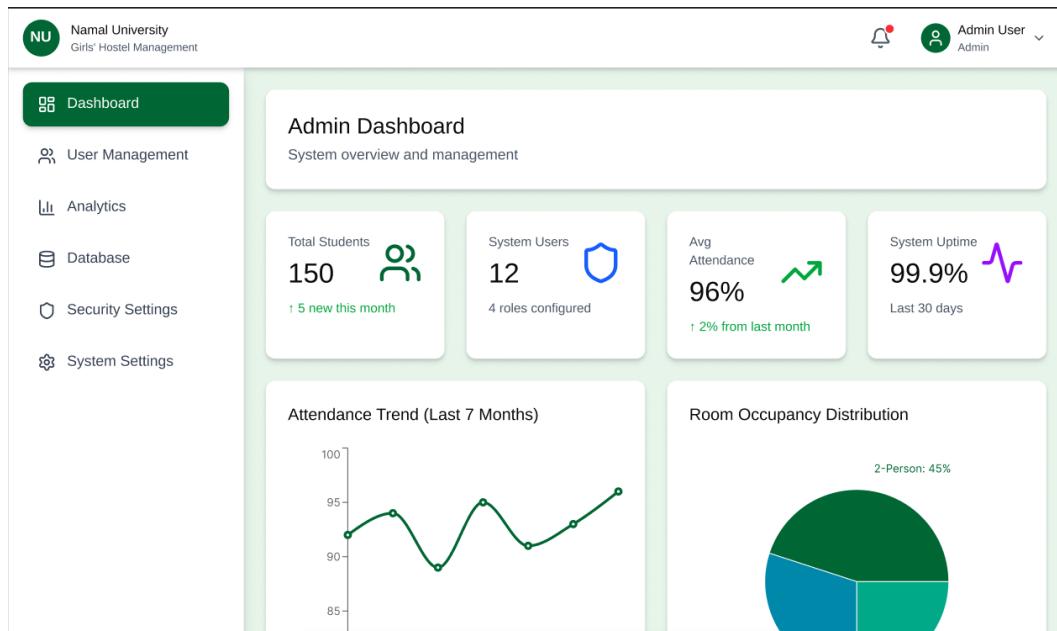


Figure 25: Admin Dashboard

6.3.7 Security Personnel

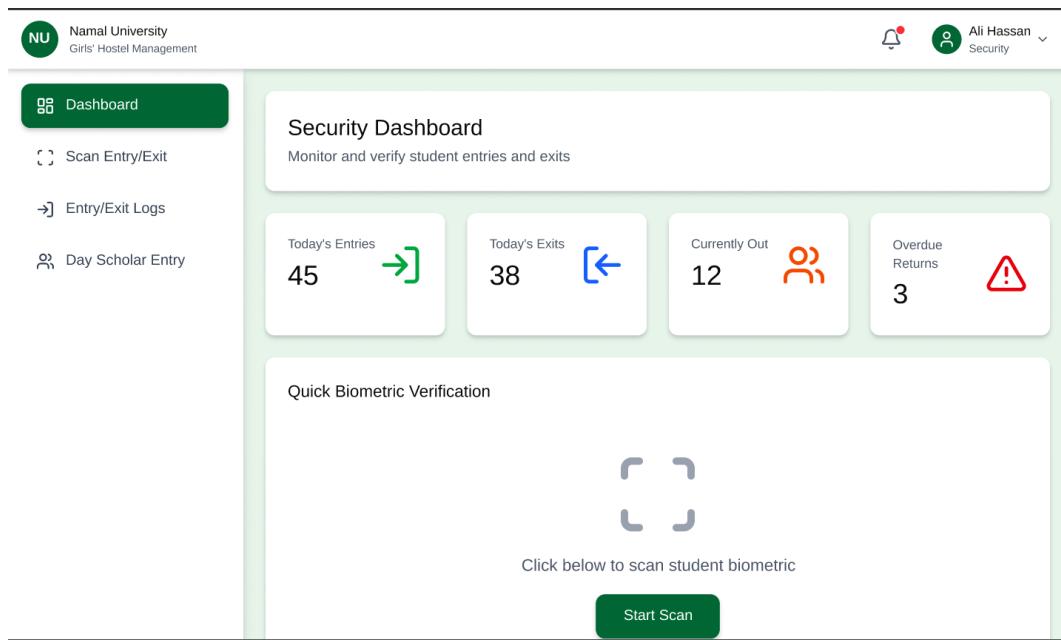


Figure 26: Security Personnel Dashboard

7 Meeting Documentation (Milestone 3)

7.1 Meeting 1: Paper-Based Prototype Validation

- **Date:** January 08, 2024
- **Time:** 3:00 PM – 4:00 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

Meeting Objective: The objective of this meeting was to present and validate the paper-based prototype of the Smart Hostel Management System (SHMS) with the Requirement Provider, ensuring alignment with the approved SRS before proceeding to detailed system design and interactive prototyping.

7.1.1 Prototype Demonstration

The project team presented hand-drawn paper-based screens representing the main system interfaces derived directly from the SRS. The prototype included:

- Student dashboard (attendance status, entry/exit requests, complaints)
- Hostel warden dashboard (attendance overview, approvals, alerts)
- Entry/exit request form with approval workflow
- Room allocation and vacancy overview screens
- Complaint submission and tracking interface

Each screen was explained by the team, highlighting how it maps to specific functional requirements defined in the SRS.

7.1.2 Discussion Summary

- The team explained how the attendance workflow integrates biometric data and manual overrides.

- Entry/exit request flow was discussed in detail, including restricted time approvals.
- Room management logic, including group-wise allocation and vacancy viewing, was clarified.
- Complaint management and notification flow was explained.
- The Requirement Provider verified that all presented screens reflected real hostel operations.

7.1.3 Feedback and Observations

- Overall structure of the system interfaces was found consistent with hostel procedures.
- The paper-based prototype clearly represented all major system functionalities.
- The Requirement Provider emphasized clarity in approval status visibility for students.

Outcome: The paper-based prototype was approved as a correct representation of system requirements. The project team was advised to proceed with creating a complete interactive prototype in Figma based on this structure.

7.2 Meeting 2: Interactive Prototype Review and Design Validation

- **Date:** January 16, 2024
- **Time:** 1:30 PM – 02:00 PM
- **Location:** Online (Google Meet)

Attendees:

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

7.2.1 Meeting Agenda

1. Presentation of complete system design artifacts
2. Demonstration of Figma interactive prototype
3. Walkthrough of major user flows

4. Validation of usability and design consistency
5. Collection of final feedback
6. Design approval confirmation

7.2.2 Prototype Demonstration

The project team presented a complete interactive Figma prototype covering all major system functionalities defined in the SRS and required by Milestone 3. The demonstration included:

- Role-based dashboards for students, wardens, and security staff
- Attendance marking and reporting interfaces
- Entry/exit request submission, approval, and tracking workflow
- Room allocation, vacancy viewing, and transfer request screens
- Complaint submission, status tracking, and resolution views
- Day scholar entry/exit monitoring interface

Navigation flow, screen transitions, and interactive elements were demonstrated live.

7.2.3 Discussion Summary

- The Requirement Provider confirmed understanding of the system workflows.
- Entry/exit approval logic and late return alerts were discussed in detail.
- Attendance dashboard statistics and daily notification timing were reviewed.
- Room allocation constraints and group registration logic were validated.

7.2.4 Key Feedback

Positive Feedback:

- The interface design was clear, simple, and easy to understand.
- Mobile-friendly layout was appreciated due to high student mobile usage.
- Approval workflows were well visualized and easy to track.
- Dashboard summaries provided useful operational insights.

Suggestions and Improvements:

- Add quick attendance percentage widget on the student dashboard.
- Show notification badges for pending approvals on warden dashboard.
- Include emergency contact field in entry/exit request form.
- Add date-range filters in attendance and complaint reports.
- Use color indicators for overdue requests and unresolved complaints.

7.2.5 Final Approval

The Requirement Provider expressed strong appreciation and satisfaction with the complete system design and interactive prototype presented by the project team. She stated that the proposed system accurately reflects real hostel operations and addresses long-standing administrative and security challenges.

The Requirement Provider further shared that if this system is successfully implemented in the future, it would significantly improve hostel management efficiency, transparency, and student safety. She particularly appreciated the clarity of workflows, role-based dashboards, and the overall usability of the system.

Although the system is currently a prototype and not yet implemented, the Requirement Provider conveyed high confidence in its practical value and expressed enthusiasm about its potential real-world adoption. The design was validated as fully compliant with the approved SRS and Milestone 3 objectives, with only minor suggestions recommended for future implementation.

Conclusion: Both meetings were conducted by the project team to validate system design and prototypes. All feedback was documented, and the final design reflects stakeholder input and confirmed requirements.

8 Conclusion

8.1 Design Completeness

This Design Specification presents a comprehensive system design for the Smart Hostel Management System that fully addresses all requirements specified in the approved SRS document. The design includes:

- Complete UML Model: Use case, DFD, sequence, activity, class, and component diagrams
- Full Traceability: Every functional requirement mapped to specific design artifacts
- Validated Design: Stakeholder feedback incorporated through interactive prototype
- Implementation-Ready: Sufficient detail for development team to begin implementation

8.2 Stakeholder Validation

The design has been validated through formal meeting with the Requirement Provider. The interactive Figma prototype received approval, confirming design completeness and suitability for implementation.

8.3 Next Steps

Following approval of this design, the project will proceed to:

1. Future Implementation:

- Set up development environment with Node.js, React.js, and MongoDB
- Implement database schema based on class diagram
- Develop RESTful APIs following component design specifications
- Build frontend interfaces based on approved Figma prototype
- Integrate biometric device APIs for attendance and gate verification
- Implement JWT-based authentication and role-based access control
- Develop notification services for email and SMS alerts

2. Testing and Quality Assurance:

- Unit testing of individual components and services
- Integration testing of module interactions

- API testing using Postman or similar tools
- User acceptance testing with hostel staff and sample student group
- Performance testing against specified requirements
- Security testing for authentication and authorization
- Cross-browser and responsive design testing

3. Deployment and Training:

- Production environment setup on university server or cloud platform
- Data migration from existing spreadsheets and systems
- User training sessions for warden, assistant warden, and security personnel
- Student orientation for system usage
- Phased rollout starting with attendance module
- Monitoring and support during initial weeks of operation

8.4 Acknowledgments

We extend our sincere gratitude to:

- **Ms. Nida Sultan Nahra**, Hostel Warden, for her guidance, detailed feedback, and commitment to improving hostel operations
- Our **course instructor**, for teaching software engineering principles and providing feedback on our design
- **Namal University**, for providing the opportunity to work on a real-world problem

A Glossary

API Application Programming Interface, used for Inter-component communication

Biometric System Fingerprint identification system used at hostel and main gate

DFD Data Flow Diagram - shows how data moves through system processes

RBAC Role-Based Access Control - security model restricting access based on user roles

REST Representational State Transfer - architectural style for web services

SHMS Smart Hostel Management System

SRS Software Requirements Specification

UML Unified Modeling Language - standard notation for software diagrams

B References

1. Software Requirements Specification (SRS) - Smart Hostel Management System, December 2023
2. IEEE Standard 830-1998: Recommended Practice for Software Requirements Specifications

C Design Tools Used

- **Diagrams:** draw.io and visual paradigm for creating use case, DFD, sequence, activity, class, and component diagrams
- **Prototyping:** Figma for interactive UI/UX design and mockups
- **Documentation:** LaTeX for professional document preparation
- **Version Control:** Git and GitHub for source code and document management
- **Collaboration:** Google Drive for file sharing and Google Sheets for meeting minutes
- **Communication:** WhatsApp for team coordination and quick discussions
- **Project Management:** Trello for task tracking and milestone planning

D GitHub Repository Link

Repository URL: <https://github.com/Muhammad-Ahmad-99/Smart-Hostel-Management-System.git>