Software Design Specification

HOSTEL MANAGEMENT SYSTEM

Project Code: ___________________

Internal Advisor: ________________

External Advisor: Prof. Muhammad Ilyas

Project Manager: Prof. Muhammad Ilyas

Project Team: Alishba Azhar, Arsheen Zahra, Samiya Shahzad

Submission Date: ______________

____

Project Manager's Signature

Document Information

Customer	UOS – Department of Computer Science
Project	Hostel Management System (HMS)
Document	Software Design Specification
Document Version	1.0
Identifier	
Status	Draft
Author(s)	Alishba Azhar, Arsheen Zahra, Samiya Shahzad
Approver(s)	
Issue Date	Oct 3, 2025
Document Location	
Distribution	1. Advisor 2. PM 3. Project Office

Definition of Terms, Acronyms and Abbreviations

This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.

HMS	Hostel Management System
DD	Design Specification
SRS	Software Requirement Specification
UI	User Interface
DFD	Data Flow Diagram

Table of Contents

1. Introduction

1.1 Purpose of Document

- This document describes the design of the Hostel Management System (HMS) using structural design methodology (breaking down into modules, components, or entities).
- It is intended for project advisors, developers, and testers to understand system components and data flow.
- Design is illustrated using DFDs, Use Case diagrams, and database models.

1.2 Project Overview

- The system manages hostel operations like registration, room allocation, fee records, etc.
- The design includes classes, modules, and diagrams showing how the system works internally.

1.3 Scope

- The scope of this document includes the architectural design of the system, module breakdown, and relationship between entities/modules.
- Everything included here is meant to guide the development team during the developmental process of the software.
- This document focuses only on design-related details and does not cover the coding phase, testing procedures, or deployment steps.

2. Design Considerations

2.1 Assumptions and Dependencies

- The system is designed to assume that the database will remain stable and accessible for all modules.
- All users who operate the system (warden, accountant, mess staff, students) will have access to working computers.
- The user roles defined earlier in the SRS will remain the same during design and implementation.
- The system depends on a central database for storing hostel records.
- The smooth working of the system depends on the proper functioning of hardware devices (computers, network devices) used by staff.

2.2 Risks and Volatile Areas

Some parts of the system are more likely to change and can affect the design.

- **Changing hostel policies:** If the university updates rules (room allocation policy, fee structure, visitor rules), the system modules may need to redesign.

- **Dependency on database structure:** Change in database tables or relationships can affect multiple modules.
- **Integration with future modules:** Adding new features (biometric attendance, online payment, or mobile app); the current design may need to be modified.

Regular backups and clear documentation help handle unexpected changes smoothly.

3. System Architecture

This section explains the overall structure of the Hostel Management System (HMS). It describes how the system is divided into different modules, how these modules interact with each other, and how data flows between users and the system.
The architecture of the system is shown using:

- Context Diagram (DFD Level 0)
- Use Case Diagram

These diagrams help in understanding the interaction between users and the system clearly.

3.1 System Level Architecture

At the system level, the Hostel Management System **is divided into the following main modules:**

1. Registration Module
2. Student Module
3. Warden Module
4. Room Module
5. Accountant Module
6. Mess Incharge Module
7. Admin Module

Relationship Between the Elements:

All modules communicate indirectly through the central database.
For Example:

- The Room Allocation Module needs student data from the Student Module.
- The Accountant Module uses data from the Student Module to calculate fees.
- The Accountant Module interacts with the Mess Module to calculate financial reports.

- Modules are loosely coupled; changing one module does not break the others.

Interfaces to External Systems:

The system may interact with a few outside services, such as:

- Network/Internet connectivity
- Gmail

If any of these are unavailable, certain features may not work, even though the internal modules remain fine.

Major Physical Design Issues (Where Elements Execute)

- The frontend (user interface) runs on the student or admin computer.
- The backend server runs the core modules like room allocation, fees, and student records.
- The database stores all information such as room details, student profiles, fee records, etc.

Global Design Strategies

- Future Strategy
- Reusability Strategy
- User Interface Strategy
- Data Management Strategy
- Error-Handling Strategy
- Concurrency Strategy

3.2 Sub-System / Component / Module Level Architecture

The Hostel Management System is organized into several major components. Each component handles a specific set of features, but all of them share data through a common database to keep the system connected.

1. Registration Module

- Handles the initial registration process for new students, manages **login**, and allows users to **update** their basic profile information.

2. Student Module

- Allows students to **submit admission forms**, **request** rooms, **check** their allocated room, view fee details, submit **complaints,** and check the daily mess menu.

3. Warden Module

- Enables the warden to **review** student applications, **approve or reject** admissions, **allocate** rooms, mark **attendance**, **manage** complaints, and **update notices** or hostel rules.

4. Room Module

- Manages the **structure of the hostel**, including floors, rooms, room types, and occupancy status. It **keeps track** of vacant and occupied rooms.

5. Accountant Module

- Handles all **fee-related tasks** such as recording payments, updating fee structures, and generating fee reports.

6. Mess Incharge Module

- Uploads the daily **mess menu**, manages the list of **enrolled students**, and generates **mess-related reports.**

7. Admin Module

- **Manages all user accounts and roles**, maintains the hostel structure (floors, rooms, capacities), performs **system backups** and data restoration, and monitors logging activity.

3.3 Sub-Component / Sub-Module Level Architecture (1…n)

Here we are breaking each module into its internal mini features.

Registration Module (Sub-Modules)

- **Account Creation:** Collect basic info (name, email, CNIC, phone).
- **Credentials Management:** Generates username/password, stores login details.
- **Profile Update Handler:** Lets students edit or update info.
- **Admin View Handler:** Allows admin to see all registered users.

Student Module (Sub-Modules)

- **Admission Form Submission**: Handles hostel admission form.
- **Room Request Handler:** Sends room request to warden module.
- **Room Status Viewer**: Shows allocated room details.
- **Fee Payment Interface:** View and pay hostel fees.
- **Mess Menu Viewer:** View monthly mess menu.
- **Complaint Submission:** Submit new complaints.
- **Complaint Tracking:** View complaint status.

Warden Module (Sub-Modules)

- **Application Review System:** Views student applications.
- **Request Approval Handler:** Approves or rejects student admissions.
- **Room Allocation Manager:** Assigns rooms based on availability.
- **Attendance Manager:** Marks and stores student attendance.
- **Complaint Handler:** Reviews and resolves student complaints.
- **Notice & Rules Manager:** Updates notices, announcements, and hostel policies.

Room Module (Sub-Modules)

- **Floor Management System:** Stores floor data and structure.
- **Room Information Manager:** Stores room ID, room type, and capacity.
- **Occupancy Tracker:** Tracks empty/occupied rooms.
- **Room Type Classifier:** Categorizes rooms (Single, Double, Quad).

Accountant Module (Sub-Modules)

- **Fee Payment Recorder:** Stores all fee transactions.
- **Fee Structure Updater:** Updates and maintains the fee structure.
- **Financial Report Generator:** Generates hostel financial reports.

Mess Incharge Module (Sub-Modules)

- **Mess Menu Manager:** Uploads and updates daily mess menu.
- **Mess Attendance System:** Records of student mess attendance.
- **Mess Enrolment Manager:** View enrolled mess students.
- **Mess Report Generator:** Generates attendance and consumption reports.

Admin Module (Sub-Modules)

- **User Account Manager:** Create, update, delete user accounts
- **Hostel Structure Manager:** Manages floors, rooms, and overall capacity.
- **Backup & Restore Manager:** Backup or restore system data.
- **Activity Log Monitor:** Track system activity logs.

4. Design Strategies

The Hostel Management System is designed with strategies that make it easy to use, maintain, and extend in the future.

4.1. Future Strategy:

The system is built in separate modules like Student, Warden, Accountant, Mess, Room Management, Registration, and Admin. This makes it easy to add new features later, such as online payments, mobile access, without affecting existing functionality.

4.2. Reusability Strategy:

Common functions like login, notifications, and database access are shared across modules. This avoids writing the same code again and keeps the system simple.

4.3. User Interface Strategy:

The interface is designed to be simple and easy to use. All modules follow a similar layout.

4.4. Data Management Strategy:

All important information about student details, rooms, fees, and mess records is stored in a central database. The system ensures data is updated correctly.

4.5. Error-Handling Strategy:

If something goes wrong, like a network or database error, the system prevents loss of information and keeps operations smooth and reliable.

4.6. Concurrency Strategy:

The system manages multiple users working at the same time. For example, it prevents two students from booking the same room at once.

5. Detailed System Design

6.References

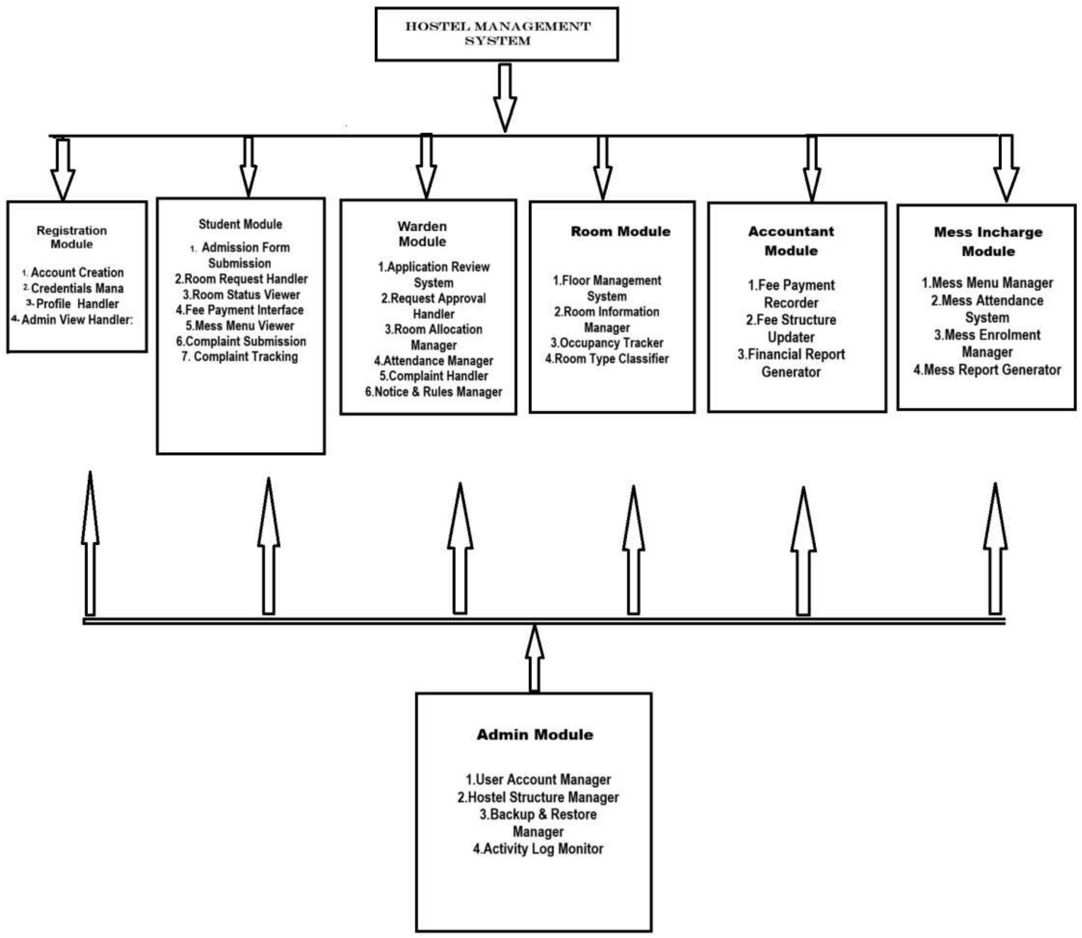
This Design document is made by using following references:

PGBH01-2025-SRS	SRS	Dec 5, 2025	https://github.com/samiyashahzad/SRS-Project-Group--4---Hostel-Management-system-/blob/main/Software_Requirements_Specifications.pdf
PGBH01-2025-Design Notes	Software Design	Oct 9, 2025	https://docs.google.com/presentation/d/1qNpIcgmx3g4xqXb7QT2ZyW7dPZN6rvdo/edit?slide=id.p1#slide=id.p1

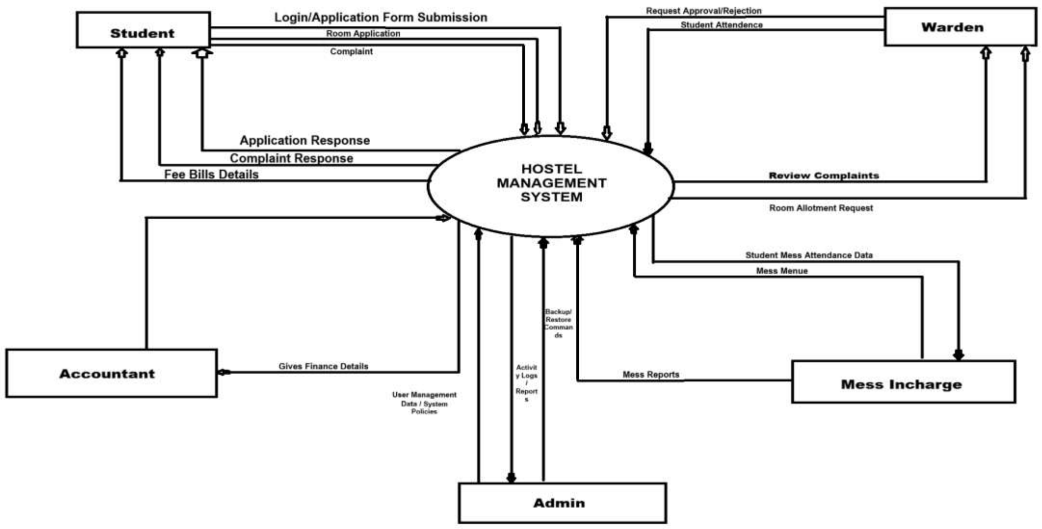
6. Appendices

The diagrams for the design document are:

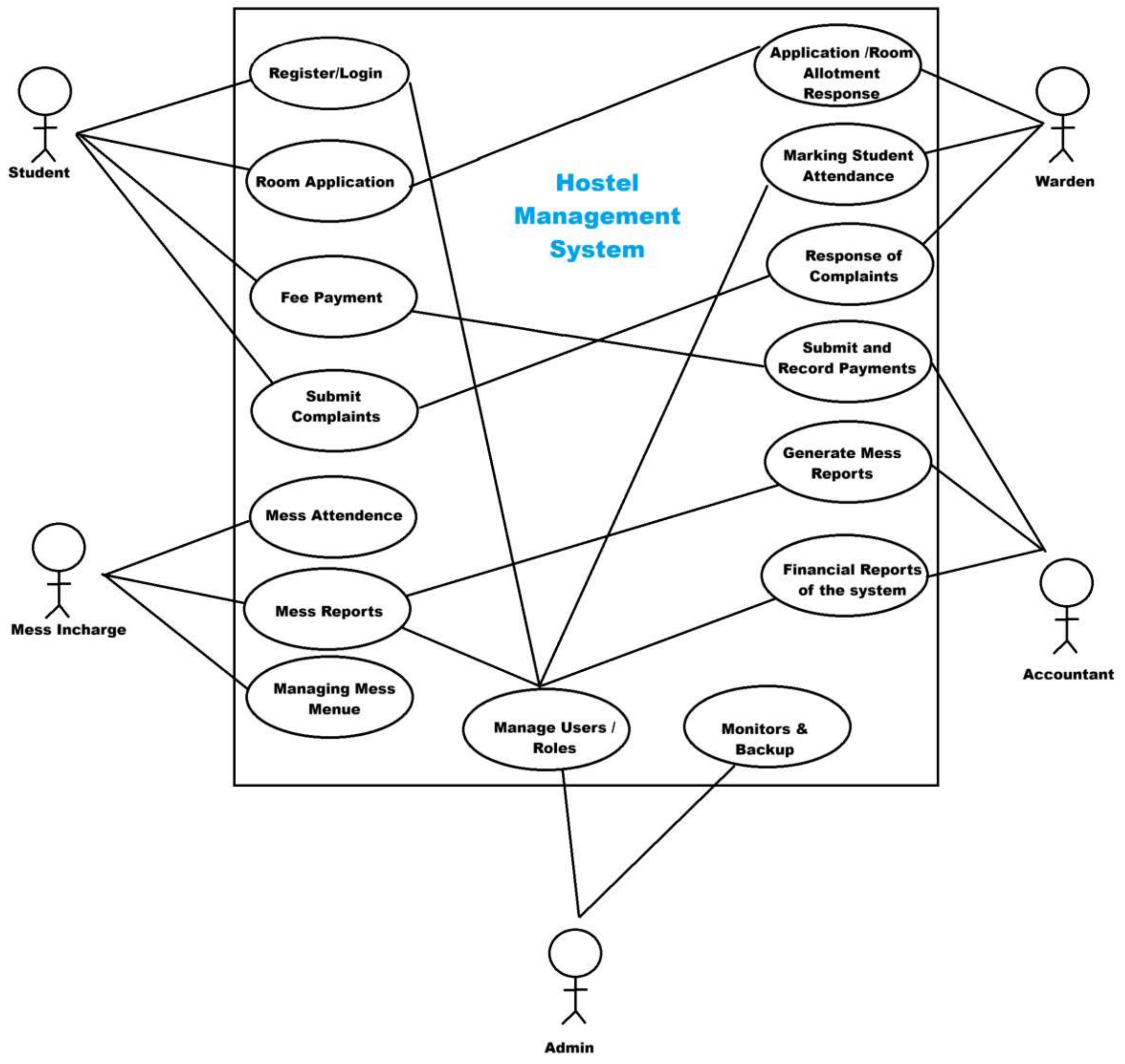
Sub-Module Level Architecture Diagram:



Data Flow Diagram:



User Case Diagram:


Hostel Management System
Student
Warden
Mess Incharge
Accountant
Admin
Register/Login
Room Application
Fee Payment
Submit Complaints
Mess Attendence
Mess Reports
Managing Mess Menue
Manage Users / Roles
Monitors & Backup
Application /Room Allotment Response
Marking Student Attendance
Response of Complaints
Submit and Record Payments
Generate Mess Reports
Financial Reports of the system