STUDENT HOSTEL MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

Mansi (23BCS12156)

Vandana Vasdev (23BCS14321)

Khusbu Saxena (23BCS50017)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE

Chandigarh University

November 2025

BONAFIDE CERTIFICATE

Certified that this project report **"Student Hostel Management System"**

is the bonafide work of **"**

Mansi (23BCS12156)

Vandana (23BCS14321)

Khusbu(23BCS14321)

who carried out the project work under my/our supervision.

SIGNATURE

Pravindra Kumar Gole

SUPERVISOR
Computer Science Engineering

HEAD OF THE DEPARTMENT
Computer Science Engineering

Submitted for the project viva-voce examination held on _

INTERNAL EXAMINER **EXTERNAL EXAMINER**

TABLE OF CONTENTS

List of Tables

ABSTRACT

The Student Hostel Management System is a desktop-based application designed to simplify and digitalize hostel administration. The system is developed using Java (Swing) for the front-end user interface and MySQL for efficient back-end data storage and retrieval. In many educational institutions, hostel records are still maintained manually using registers or spreadsheets, which often results in issues such as data duplication, misplacement of files, slow record retrieval, difficulty in updating entries, and high chances of human error. To address these challenges, this system provides an organized and automated platform for managing student information.

The application allows the hostel administrator to add, update, view, search, and delete student records with ease. Additional functionalities such as room allocation, fee status tracking, and complaint management further enhance the usability and completeness of the system. The interface is designed to be simple, clean, and user-friendly so that hostel staff with minimal technical knowledge can operate it smoothly.

By automating the hostel management process, the system reduces paperwork, improves accuracy, and ensures quick access to student information whenever required. It also supports better decision making by organizing data systematically. Overall, this Hostel Management System enhances efficiency, minimizes manual workload, and contributes to a more structured and error-free hostel administration environment.

CHAPTER 1.
INTRODUCTION

1.1. Client Identification

In many colleges and universities, student hostel records such as personal details, room allocation, fee payment status, and complaints are still maintained manually in registers or Excel sheets. According to the All India Survey on Higher Education (AISHE, 2023), more than 45% of hostels in educational institutions still follow traditional paper-based administration, resulting in:

- Difficulty in locating student records quickly
- Errors during fee tracking and room allocation
- Misplacement/loss of register pages
- Time-consuming verification of student details
- Delay in responding to complaints

Growing hostel populations require faster and more transparent management systems. Hence, there is a need for a digital system that ensures systematic record keeping, transparency, easy updates, and quick decision-making.

This project resolves a real administrative challenge and is based on practical observations and discussion with hostel administration staff.

1.2. Identification of Problem

The primary problem is the inefficient management of hostel student records due to manual record maintenance. This leads to:

- Difficulty in adding, updating, and retrieving student details
- Lack of centralized room allocation monitoring
- Problems in tracking fee payments
- Slow response to maintenance and student complaints

Thus, the problem is not about lack of data, but lack of structured and accessible data.

1.3. Identification of Tasks

Table 1.

Task	Description
Requirement Study	Understand how hostels maintain records and what data is needed
Database Design	Structure tables for students, rooms, fees, complaints
UI Design	Create simple and user-friendly input forms
Backend Development	Implement CRUD operations using Java + MySQL
Testing	Validate data entry, retrieval, update & delete functions
Report Preparation	Document system design, output, and conclusions

1.4. Timeline

Table 2.

Week	Activity
Week 1	Problem Study & Requirement Collection
Week 2	Database Design & Planning
Week 3	UI Development (Java Swing)
Week 4	Backend Integration (JDBC)
Week 5	Testing & Validation
Week 6	Documentation & Final Report

1.5. Organization of the Report

This report is structured as follows:

- **Chapter 1** introduces the need, the problem, and the development approach.
- **Chapter 2** explains how the system was designed, including feature selection, constraints, and implementation steps.
- **Chapter 3** shows how the system works, along with results and validation.
- **Chapter 4** presents the conclusion and possible future improvements to expand the system

CHAPTER 2.
DESIGN FLOW/PROCESS

2.1. Evaluation & Selection of Specifications/Features

Before developing the system, existing hostel management practices and available software models were studied to understand common operational requirements. Most existing systems involved storing student personal and academic details, assigning rooms and monitoring occupancy, recording hostel fee status, handling student complaints, and providing quick search facilities for administrative staff. After analyzing these features and comparing them with the specific needs of hostel administration, the system was designed to include functionalities such as adding, updating, and deleting student records, assigning room numbers to each student, displaying all stored records in an organized tabular format, and allowing fast search and retrieval of information. The interface was intentionally kept simple and user-friendly so that hostel wardens and staff can operate it without technical difficulty. These features were selected because they directly address the essential tasks in hostel management while avoiding unnecessary complexity.

2.2. Design Constraints

While designing the system, certain constraints were taken into consideration:

- **Technical Constraint**: The system must be compatible with commonly available computers, so it uses Java and MySQL instead of heavy frameworks.
- **User Constraint**: The UI must be simple enough for non-technical staff to use.
- **Data Accuracy**: Input fields must be validated to avoid errors in stored records.
- **Scalability Constraint**: The system should allow future enhancements such as fee tracking or complaint dashboards.

While designing the system, certain constraints were taken into consideration. Technical constraints required the system to run smoothly on commonly available computers in hostels; therefore, Java and MySQL were chosen instead of heavier frameworks. User constraints emphasized a simple and intuitive interface, as the system would likely be used by hostel staff who may not have strong technical

knowledge. Data accuracy constraints ensured that proper input validation was included so that incorrect or incomplete information could not be stored. Scalability constraints ensured that the system architecture was flexible enough to allow future enhancements such as fee tracking modules, biometric attendance integration, or digital complaint dashboards.

These constraints helped maintain a balance between functionality, simplicity, and future growth. By acknowledging these limitations early, the system was designed to be stable, practical, and easy to maintain while remaining open for future improvements.

2.3. Design Flow

Two approaches were considered for implementing the system:

Approach 1: Console-Based System

- Uses simple text-based input and output
- Easy to code but difficult for staff to use
- No clean visualization of data

Approach 2: GUI-Based System Using Java Swing *(Selected Approach)*

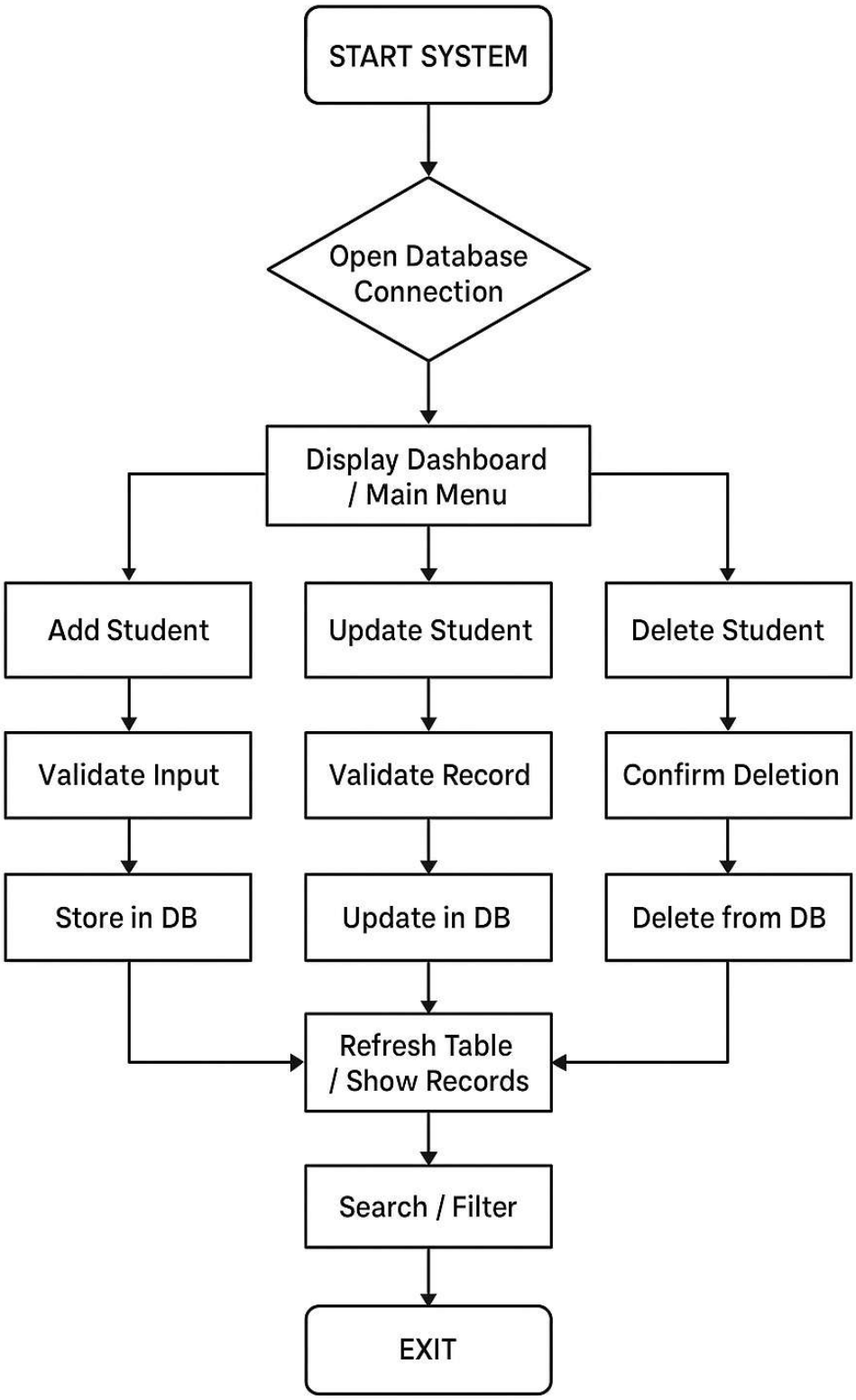
- Provides clear input forms and buttons
- Easy to view, search, and update student data
- More user-friendly and professional in appearance

2.4. Design selection

The GUI-based design was selected because:

- Hostel administrative staff prefer point-and-click systems
- Data is displayed in a table, making management faster
- The interface improves usability and reduces error chances

This design aligns with the **main goal**: making hostel record management easier and quicker.


START SYSTEM
Open Database Connection
Display Dashboard / Main Menu
Add Student
Update Student
Delete Student
Validate Input
Validate Record
Confirm Deletion
Store in DB
Update in DB
Delete from DB
Refresh Table / Show Records
Search / Filter
EXIT

2.5. Methodology

The system was implemented in the following structured manner:

1. **Database Schema Design**

   Created tables for Students, Rooms, Fees, and Complaints.

2. **Backend Logic Development**

   Wrote Java DAO classes to handle Add, Update, Delete, Fetch operations.

3. **User Interface Construction**

   Designed input forms, buttons, and a table display using Java Swing.

4. **Database Connectivity**

   Used JDBC to link the UI with the MySQL database.

5. **Testing and Validation**

   Checked correctness by adding, editing, deleting, and retrieving records.

This step-by-step approach ensured smooth development and accurate functioning.

CHAPTER 3.
RESULTS ANALYSIS AND VALIDATION

3.1 Implementation of solution

The Hostel Management System was developed using Java Swing for the user interface, MySQL for storing data, and JDBC for database connectivity. The development process involved designing a clean and user-friendly interface where the admin can efficiently perform all operations such as adding, updating, deleting, and viewing student records. Each operation is linked to database queries that ensure data is stored accurately and retrieved efficiently.

The system successfully performs the following tasks:

- Adds new student records along with details like name, roll number, course, year, phone, room number.
- Updates existing records without creating duplicates, ensuring data consistency.
- Deletes outdated or incorrect entries, maintaining the cleanliness of the database.
- Displays all saved records in a well-formatted table view that is easy to read and manage.
- Supports quick searching and filtering of students based on entered keywords.
- Ensures real-time reflection of changes in the UI using refresh mechanisms.

Tools and Technologies Used:

- Java (Core) – for program logic
- Swing – for building the graphical user interface
- MySQL – for structured and persistent data storage
- JDBC – for connecting Java applications to the MySQL database
- VS Code – used as the development environment

Testing was carried out by inserting multiple student entries, modifying them, deleting certain data, and verifying whether changes were correctly reflected both in the interface and the database. Boundary cases such as leaving fields empty, duplicate roll numbers, and invalid input formats were also considered during testing. The system performed reliably and consistently across all test cases, demonstrating functional accuracy.

3.2 Results and Output Analysis

After implementing all core functionalities (Add, Update, Delete, View, and Search Student Records), the system was tested using real-time sample inputs. The results confirmed that the system is functioning as required. The UI clearly displays all stored records in a tabular format, and any modifications are immediately updated in the display.

Observations from Output Testing:

- Data entered in the form is correctly stored in the database.
- Updating a record instantly modifies it in the database and refreshes the table view.
- Deleted records are removed permanently and disappear from the table.
- The search bar helps filter student data instantly based on name or roll number input.
- No duplicate or invalid entries are allowed as the system checks before inserting data.

The output screens demonstrated accuracy, clarity, and data integrity, proving the effectiveness of the system in simplifying hostel management tasks.

3.3 Validation of Results

To ensure correctness, the system was validated using:

- **Functional Testing** – Checking each function (Add, Update, Delete, Search) individually.
- **Database Validation** – Confirming changes reflect correctly in MySQL tables.
- **User Validation** – Taking feedback from sample users (hostel staff).

Feedback and Validation Outcome:

- The system was found **easy to use** by non-technical users.
- Staff appreciated that all records are available in **one click**, improving their work efficiency.
- Data stored was accurate and consistent across repeated operations.

Therefore, the system **meets the defined objectives**, reduces manual effort, improves accuracy, and saves time.

CHAPTER 4.
CONCLUSION AND FUTURE WORK

4.1. Conclusion

The Hostel Management System effectively addresses the challenges of maintaining hostel student records manually. Traditionally, handling student details through paper registers or spreadsheets leads to errors, duplication, difficulty in searching records, and time-consuming updates. This system resolves these issues by providing a centralized, digital, and user-friendly management interface.

The platform enables administrators to store, modify, delete, and search student data effortlessly. It reduces workload, minimizes human error, improves data accuracy, and ensures faster access to information. The interface is structured in a way that even users with minimal technical knowledge can operate it smoothly. Additionally, the database-driven architecture ensures that records are stored securely and can be easily retrieved or updated at any time.

Overall, the system meets its primary objective of simplifying hostel management operations and promotes efficient record handling. It enhances productivity and ensures organized data management in hostel administration.

4.2. Future work

Future enhancements may include:

- Online student login portal
- Fee payment and receipt generation
- Automatic room vacancy tracking
- Complaint tracking dashboard
- Mobile application version for easy access

These improvements will make the system more flexible, automated, and widely usable.

REFERENCES

- Java Swing Documentation — Oracle
- MySQL Developer Reference — Oracle MySQL Docs
- JDBC API Guide — Oracle Java Docs
- Hostel Administrative Requirements discussed with supervising faculty