A Project Report on

# COMILLA UNIVERSITY HALL MANAGEMENT SYSTEM

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science (Engineering) in **Information and Communication Technology.**


**Submitted By:**

ID: 12009044

ID: 12009045

Session: 2019-2020

**Department of Information & Communication Technology,**

**Comilla University**


**Date of Submission: February 19, 2025**

# ABSTRACT

Comilla University Hall Management System is a software developed for managing colorful conditioning in the residence. There are five resident halls for the scholars of Comilla University. As the number of halls are adding up, the number of scholars abiding in those caravansaries are also adding. As a result, the people managing the halls are under a lot of stress. Software or applications aren't generally used in this environment. Generally, in Bangladesh perspective no digital systems or applications are used, most of the halls or residences are managed manually. In early days, managing manually was not very tough. But at present manual system is tedious. This specific design addresses the challenges of hall management and circumvents the issues that arise when carried out by hand. When the drawbacks of the current system are recognized, lead us to the design of a motorized system that will work with the existing system and the system that is additionally easily grasped and more GUI acquainted. We can ameliorate the effectiveness of the system and therefore overcome the downsides of the being system. Comilla University Hall Management System targeted for sale operation of the hall for better control and timely response. This eliminates time day and paper deals being marked. The Warden is handed with a better control over the deals like adding the details of new scholar in the hall, modifying the details of the scholars, deleting the scholars, viewing the pupil's details abiding in the hall, adding or removing rooms, checking available seats. scholars can also check if there's any room available or not, look for better seats. As the main target of this application is to do efficient management by storing scholar's data, MongoDB is used as database as it offers numerous benefits, especially its ability to manage unstructured data, its straightforward process for horizontal scaling, and its quicker performance in insert and update tasks, which makes it perfect for applications that require rapidly evolving data structures and handle large volumes of data. In the future, we can implement online fee payments, integrate various load balancers, develop a mobile application, and establish a backup database for improved security. This design's main aphorism is to reduce the effect of wardens and give better service to the scholars. The main thing of this design is to develop a system for the robotization of the hall affiliated information.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

## 1.1 Overview

The Comilla University Hall Management System is a web application designed to provide a straightforward and attractive interface for managing processes. Traditionally, hostel management relies on physical paperwork for documentation and various tasks. Our goal is to streamline the control system, making it more user-friendly for both students and the room authorities.

The primary aim of creating the Comilla University Hall Management System is to oversee information related to both new and returning students, handle student payments, allocate rooms, and manage room assignments through a web portal. Presently, all rooms at Comilla University are handled manually within the hall. The verification of registration forms for various data processes is conducted manually as well. Consequently, there are numerous repetitions that could be easily eliminated, leading to significant pressure on those who manage the hostel, especially since software is rarely used in this context.

This project specifically addresses the challenges associated with hostel management and seeks to eliminate the issues that arise from manual operations. Recognizing the shortcomings of the current system has prompted the design of a computerized solution that will integrate with the existing system while being more user-friendly and oriented towards a graphical user interface (GUI). This can enhance system efficiency and bridge the gaps found in current systems.

## 1.2 Motivation

The objective of the Comilla University Hall Management System is to organize all hall-related activities systematically. It is a web-based application designed to facilitate student accommodation in the university hostel more effectively. The system is overseen by the Warden, who acts as the administrator. This project maintains records of both hostellers and applicants. The primary goal of this project is to reduce manual tasks and simplify hostel operations. This system offers an online application for hostel accommodation, automatically selects students from a

waiting list, manages mess calculations, handles complaint submissions, and maintains a notice board, among other features. Students will receive approval notifications, be able to view the notice board, and check hostel fees by logging into the online system.

## 1.3 Existing System

Our university has a total of five hostels, comprising three for boys and two for girls. Currently, all these halls are managed by the hall administration through manual processes. The verification of registration forms and other data processing tasks are performed manually as well. As a result, there are numerous repetitive tasks that could be easily eliminated.

This leads to significant pressure on the individuals managing the hostels, and software solutions are seldom utilized in this area. This particular project addresses the challenges associated with hostel management and aims to eliminate the problems that arise from manual processes.

Recognizing the shortcomings of the current system has led to the development of a computerized system that will be compatible with the existing setup while also being more easily grasped and easily adapted. Thus, the system's efficiency can be enhanced and address the limitations of the current system.

## 1.4 Problem Statement

The Comilla University Hall Management System was developed to improve the management of the hostel. A key benefit of this software is that it replaces the outdated manual processes. Since most hostels are overseen by a single manager, information such as student occupancy and outstanding payments is often recorded on paper or receipts.

If these documents were to be lost or stolen, it would be nearly impossible to retrieve that information. Additionally, staff members may not always be aware of how many students are currently occupying a room or if it has reached capacity. This initiative will provide significant support to the staff, making the hostel's operations more efficient.

## 1.5 Proposed System

The anticipated results of creating the ‘Comilla University Hall Management System’ include:

- A fully digital management platform for Comilla University Hall.
- Electronic storage of student information.
- Safe and digital communication channels between the administration and students.
- Decreased reliance on human resources and lower additional expenses.
- Enhanced student experience through a more organized and effective system for overseeing dormitories.
- Possible financial savings through decreased administrative expenses and better management of dormitories.

## 1.6 Objectives

The goals of creating the 'Comilla University Hall Management System' are:

- To establish a comprehensive system for managing hostels.
- To offer an online application for students wishing to apply for hostel accommodation.
- To evaluate the effectiveness of system design for a small IT initiative.
- To keep a separate record for students residing in the hostel and those on the waiting list.
- Admin can send approval notifications to all students whose applications have been accepted via email.
- Automatically update the hosteller's records with student details when the admin confirms allotment, and remove them once vacation is confirmed or after the course concludes.
- Students are able to submit their complaints.
- Admin can modify the notice board, and all students can access it.

.

# CHAPTER 2

# LITERATURE REVIEW

Managing student accommodations or dormitories can be a demanding and intricate responsibility. Dormitories play a crucial role in shaping students' academic and social experiences, making them an essential component of campus life. However, overseeing these facilities entails a variety of administrative responsibilities, such as handling room assignments and maintaining precise occupancy records, which can be both labor-intensive and challenging to execute effectively. Additionally, conventional manual methods for managing dormitories can be clunky and prone to errors, potentially leading to issues like incorrect records, misplaced documents, and bottlenecks in administration.

To tackle these difficulties, numerous universities and colleges are adopting digital solutions like hall management system websites to optimize and automate the management of dormitories. These platforms offer a centralized hub for overseeing accommodations and can enhance the efficiency, accuracy, and transparency of hall management. Generally, these websites are designed to offer automated tools for building management, room assignment oversight, and occupancy tracking. Moreover, they may include features such as online student registration, communication avenues between students and administrators, and other functionalities that aid in streamlining the hall management process.

Considering the advantages of a hall management system website or web application, many educational institutions are looking into ways to integrate these systems on their campuses. Creating a hall management system website for our university can enhance the management of residences, reduce administrative expenses, and elevate the overall student experience. In this project, our goal is to develop a hall management system website tailored to address the specific challenges encountered by our hall management team, providing a practical and efficient platform for managing our dormitories.

## 2.1 Associated Work

### 2.1.1 Dhaka University Hall

Information regarding the university's residence halls, such as Dr. Muhammad Shahidullah Hall, Jagannath Hall, and Salimullah Muslim Hall, is available on this page. Along with helpful connections for staff and students, the website offers details about the university's governing structure, leadership, and history. There are also connections to the university's employment openings and tender notices, as well as login pages for staff and students [1].

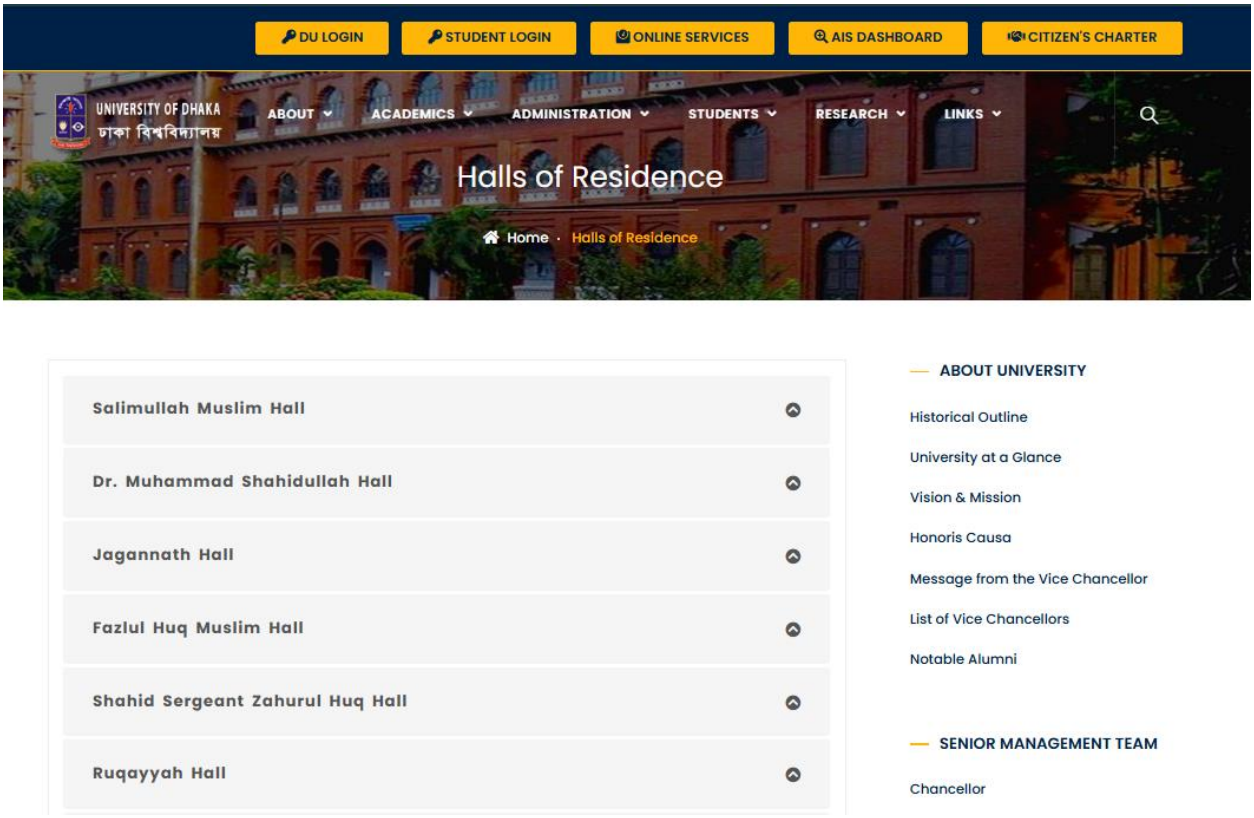


Figure 2.1.1 Dhaka University Hall Home Page

### 2.1.2 Shahjalal University of Science & Technology Hall

Utilizing information technology, such as several web-based automation systems, the "digital world" is being digital. Shahjalal University of Science & Technology is responsible for designing and implementing a Web-based Hostel or Hall Management System [2].

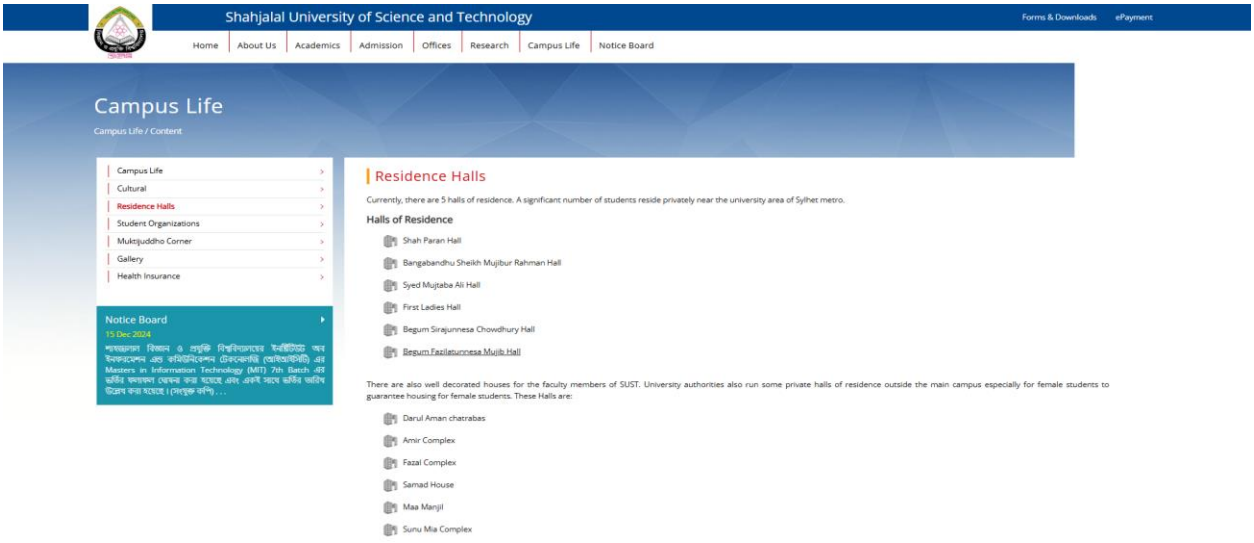


Figure 2.1.2 Shahjalal University of Science & Technology Hall Home Page

### 2.1.3 Khulna University of Engineering &Technology Hall

Khulna University of Engineering and Technology also uses Web-based Dormitory Management System for easy maintenance of their residences [3].

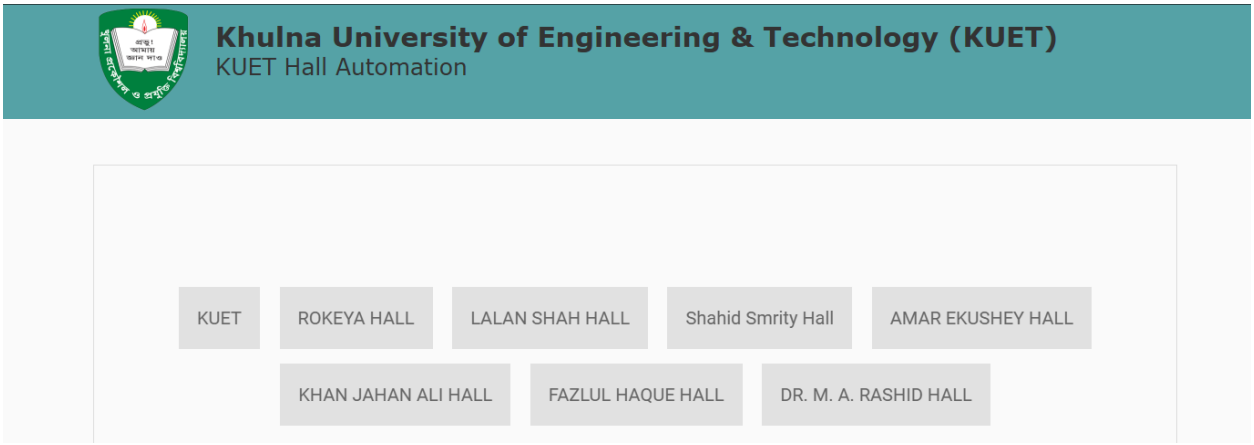


Figure 2.1.3 Khulna University of Engineering and Technology Hall Home Page

# CHAPTER 3

# METHODOLOGY

## 3.1 Project Plan

This document outlines the project plan for the development of a Comilla University Hall Management System. This system aims to streamline hostel operations, improve communication, and enhance overall management efficiency.

### 3.1.1 Project Overview

Project Title: Comilla University Hall Management System

Project Objective: To develop a comprehensive software solution that automates and streamlines the management of hostel operations, including room allocation, student registration, fee management, and maintenance requests.

Target Audience: Hostel administrators and students.

Key Features:

- Student registration and profile management
- Hostel building management
- Room management
- Seat allocation and cancellation
- Notice update
- Student's problem and solution

### 3.1.2 Project Scope

In-Scope:

- Development of a web-based application.
- User roles: Admin, Student, and Maintenance Staff.

Out-of-Scope:

- Mobile application development (can be considered in future phases).
- Payment system.

### 3.1.3 Project Deliverables

Phase 1: Requirements Gathering and Analysis

- Document functional and non-functional requirements.
- Conduct stakeholder interviews and surveys.

Phase 2: System Design

- Create system architecture diagrams.
- Design database schema.
- Develop UI/UX prototypes.

Phase 3: Development

- Set up development environment.
- Implement core features (student registration, room allocation, etc.).
- Conduct unit testing.

Phase 4: Testing

- Perform integration testing.
- Conduct user acceptance testing (UAT).
- Fix bugs and optimize performance.

Phase 5: Deployment

- Deploy the system to a production server.
- Provide training to end-users.
- Prepare user documentation.

Phase 6: Maintenance and Support

- Monitor system performance.
- Collect user feedback and issues.
- Plan for future enhancements.

### 3.1.4 Risk Management

Risk 1: Delays in development due to unforeseen technical challenges.

Mitigation: Regular code reviews and agile sprints to identify and address issues early.

Risk 2: Insufficient user training leading to low adoption rates.

Mitigation: Comprehensive training sessions and detailed user manuals.

Risk 3: Data security and privacy concerns.

Mitigation: Implement robust security measures and conduct regular security audits.

### 3.1.5 Quality Assurance

- Regular peer reviews to ensure code quality.
- Comprehensive testing including unit, integration, and user acceptance testing.
- Automated builds and tests to catch issues early.

### 3.1.6 Post-Implementation Review

- Assess the success of the project against initial objectives.
- Gather feedback from users to identify areas for improvement.
- Update project documentation with lessons learned and best practices.

## 3.2 Software and Hardware Requirements

### 3.2.1 Software Requirements

Software required for the development of our project Comilla University Hall Management System are:

- HTML
- CSS
- Java Script
- Jquery
- Node js
- Tailwind CSS
- Text Editor
- MongoDB

#### 3.2.1.1 HTML

Web documents can be described using HTML, a markup language. On our website, HTML5 is used. HTML code is used on all websites, including this one, to help format and display text and images in a readable manner. Without HTML, a browser would only show plain text without any formatting and without any links because it would not know how to format a page.

#### 3.2.1.2 CSS

Cascading Style Sheets (CSS) is a language for style sheets that specifies the appearance of documents written in markup languages like HTML. HTML, JavaScript, and CSS are the core technologies of the World Wide Web.

### 3.2.1.3 JAVA SCRIPT

Programming languages that adhere to the ECMAScript specification include JavaScript. JavaScript is multi-paradigm, high-level, and frequently just-in-time compiled. It features first-class functions, prototype-based object orientation, dynamic typing, and curly-bracket syntax.

### 3.2.1.4 JQuery

To simplify HTML DOM tree traversal and manipulation, event handling, CSS animation, and Ajax, a JavaScript library named jQuery was developed.

### 3.2.1.5 Node JS

Node.js (Node) serves as an open-source, cross-platform runtime environment for running JavaScript code. It is widely utilized for server-side programming, allowing developers to write both client-side and server-side code in JavaScript without the need to learn a separate language. While some may describe Node as a programming language or software development framework, that is not accurate; it is solely a runtime for JavaScript.

Node employs the V8 JavaScript engine, which is also utilized by Google Chrome and other web browsers. Written in C++, it is compatible with macOS, Linux, Windows, and various other systems. This engine is responsible for parsing and executing JavaScript code. It can function outside of a browser environment, either integrated into a C++ application or used as a standalone program. The V8 engine internally compiles JavaScript through just-in-time (JIT) processes to enhance execution speed.

### 3.2.1.6 Tailwind CSS

Tailwind CSS is a CSS framework that helps developers build responsive and customizable web interfaces. It's a utility-first framework, which means it provides a set of pre-defined utility classes that can be used to style elements.

### 3.2.1.7 TEXT-EDITOR

You can effectively design, create, and manage standards-based websites and applications with Sublime Text, a web development tool. Sublime Text offers a potent blend of code editing capabilities, application development tools, and visual layout tools.

### 3.2.1.8 MONGODB DATABASE

MongoDB is an open-source, non-relational database management system (DBMS) that processes and stores many kinds of data using flexible documents as opposed to tables and rows.

As a NoSQL database solution, MongoDB provides a flexible data storage paradigm that enables users to store and query a variety of data types with ease, hence removing the requirement for a relational database management system (RDBMS). In addition to making database administration easier for developers, this strategy creates an environment that is highly scalable for cross-platform services and apps.

Documents or sets of documents are the basic data units of MongoDB. These documents can hold a range of data kinds and can be distributed across different platforms because they are formatted in Binary JSON (Java Script Object Notation). Users have remarkable freedom when it comes to creating data records, querying document collections using MongoDB aggregation, and analyzing large volumes of data because to MongoDB's dynamic schema design.

## 3.2.2 Hardware Requirements

| **Component** | **Specification** |
|---|---|
| Processor | Pentium IV Processor |
| RAM | 8 GB |
| HDD | 40 GM or more |
| Monitor | SVGA Color |

Table 3.2.2 Hardware Requirements

## 3.3 Software Process Model

In order to address a specific issue within an industry, a software developer or a group of developers must adopt a development strategy that encompasses the process, methods, tools layer, and generic phases. This approach is commonly known as a process model or a software development paradigm. Our project utilizes the waterfall model.

The phases of the waterfall model include:

- Requirement Definition
- System and Software Design
- Implementation
- Integration and System Testing
- Operation and Maintenance

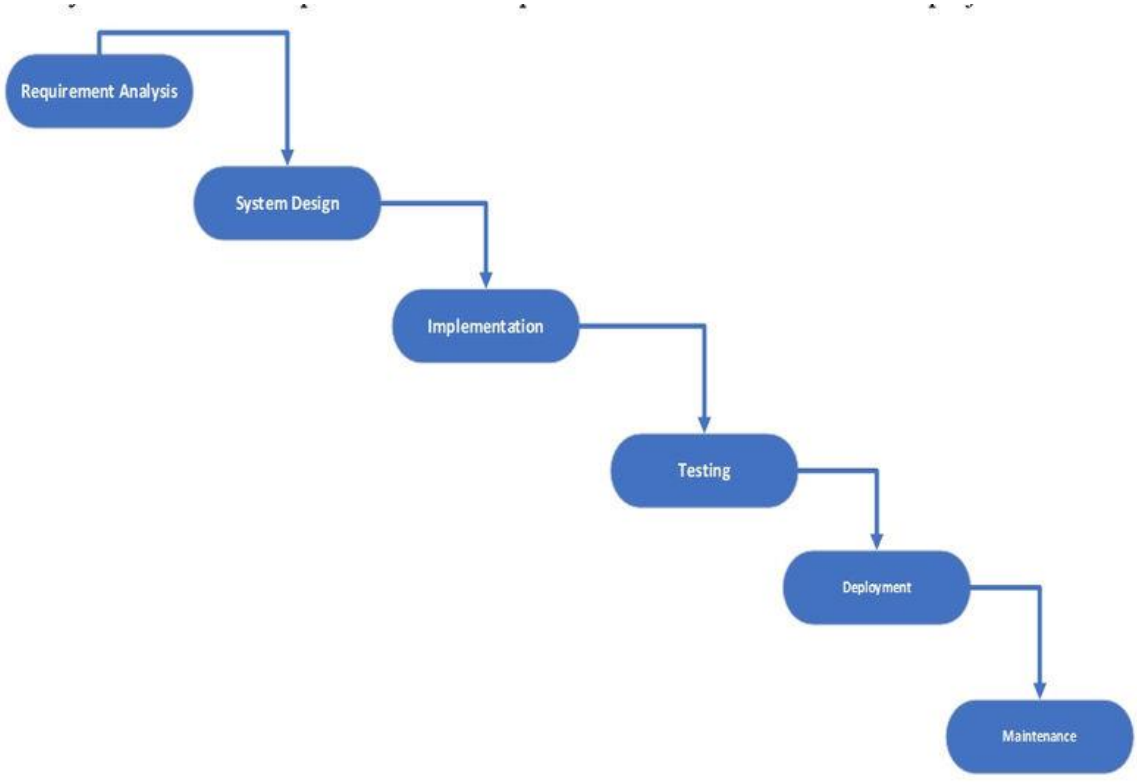


Figure 3.3 Waterfall Model

### 3.3.1 Front End Design

We utilized various markup languages, style sheets, programming languages, and libraries for both the front-end and back-end.

- HTML 5
- CSS
- Tailwind CSS
- JavaScript

### 3.3.2 Back End Design

We employed various programming languages and frameworks to finalize the back-end design of the website.

- Node JS
- Express JS
- MongoDB

### 3.3.3 Input Design

The system architecture is split into two sections: the Administrator section and the User (student) section.

- The Administrator has the ability to assign various students to different hostels.
- He is also able to discharge students from the hostels.
- He can modify the information of the students.
- He has the authority to change their accommodations, as well as to amend student records.

Changing over user-generated inputs into a computer-readable organize is a basic prepare. Input plan plays a crucial part in the advancement prepare, as erroneous input informations are the essential source of blunders in information handling. Input plan can help moderate incorrect passages. It includes making details and strategies for information passage into a framework, and

must be user-friendly. The objective of input information plan is to encourage simple, consistent, and error-free information passage. In input information plan, we make the source reports that capture the data and at that point select the medium through which they will be entered into the computer.

### 3.3.4 Process Design

Process design is crucial in the development of a project. To grasp the operational procedure, it is essential to implement process design.

The tools utilized for process design include the Data Flow Diagram and the System Flow Chart. A System Flow Chart provides a visual representation of the system, demonstrating the total control flow during processing at the job level; it details the tasks necessary to shift from a physical model to a logical one.

In the meantime, a Data Flow Diagram acts as a conceptual representation of the data transfer within the project. Different symbols are used to construct the DFD, featuring a source and a destination. The circles represent the processing, squares illustrate the source and destination, and arrows denote the data flow. Readers can readily understand the project via the Data Flow Diagram.

## 3.4 Architecture

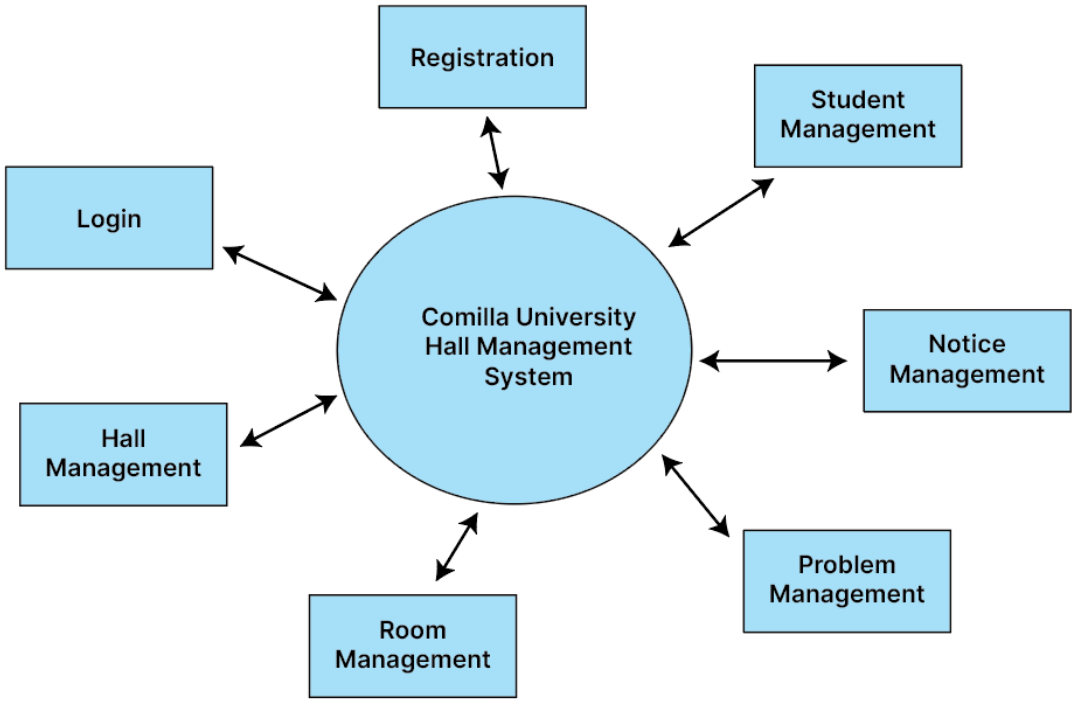


Figure 3.4 System Architecture

The Hall Management System is a web-based application providing a seamless experience for both administrators and users.

Administrator controls the overall system. For that admin needs to register and login at first. After login admin can see and control every section in the application. Admin can perform actions like add, update od delete any particular building or any room of a building. Admin can allocate seat to students taking necessary information and can also cancel remove any allocation on necessary conditions. Admin can update the notice section on daily basis. Admin will also be able to check if there is any complaint or problem and give required solution.

Students after registering and login, can check if there is any room or seat available or not. On that basis, he/she can apply for the seat.  Students can also enquiry about any other students in which building and room he/she resides in. Also, can get detailed information about a particular student. Students can check important notice, can complain about a particular problem and get immediate solutions.

## 3.5 Data Flow Diagram (DFD)

### 3.5.1 Context Level DFD

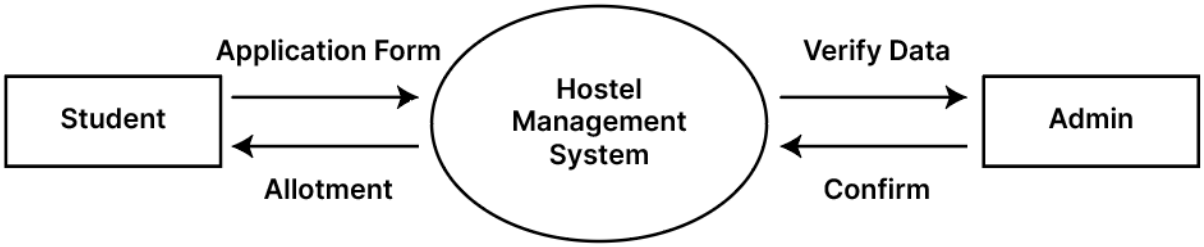


Figure 3.5.1 DFD Context Level

### 3.5.2 Student Module

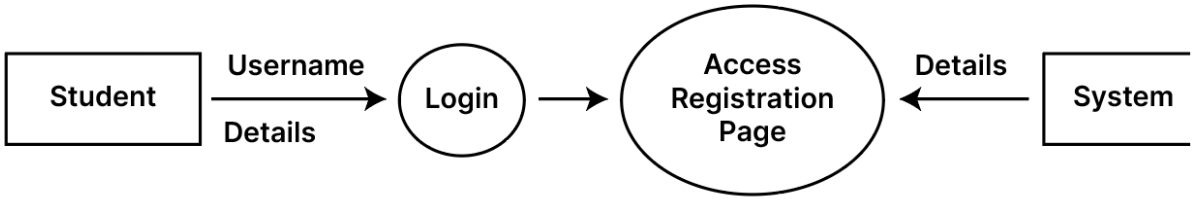


Figure 3.5.2 DFD Student Module

### 3.5.3 Registration Process


Figure 3.5.3 DFD Registration Process

### 3.5.4 Admin Module

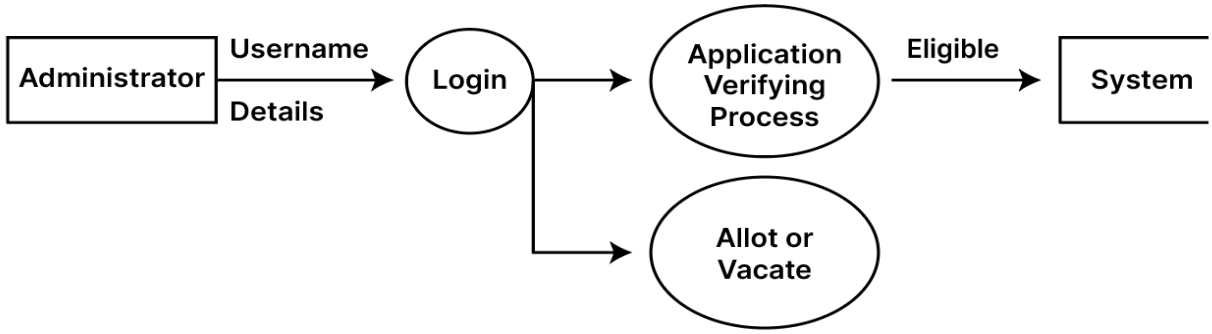


Figure 3.5.4 DFD Admin Module

### 3.5.5 Allotment Process

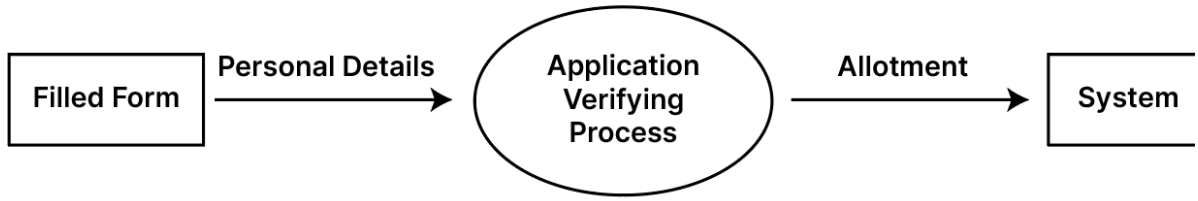


Figure 3.5.5 DFD Allotment Process

### 3.5.6 Vacating Process

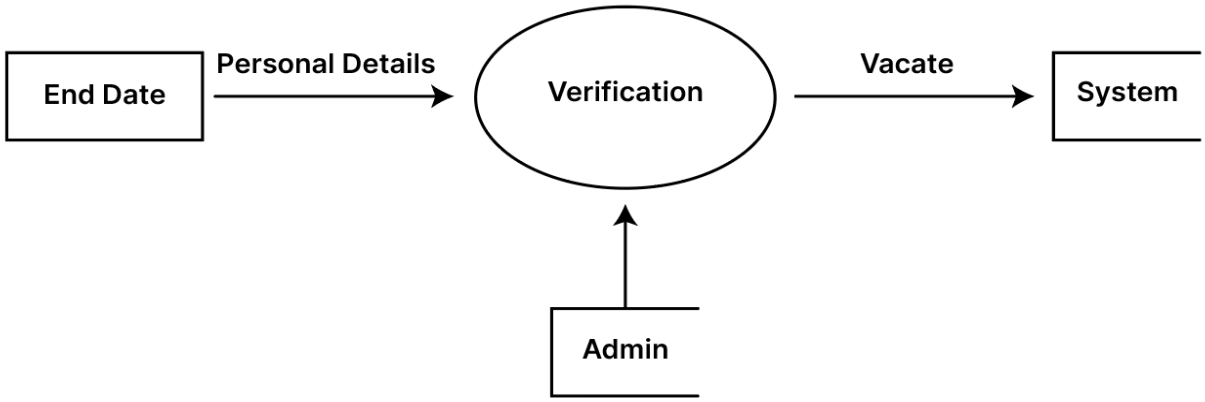


Figure 3.5.6 DFD Vacating Process

## 3.6 Flow Chart

### 3.6.1 Student Activity

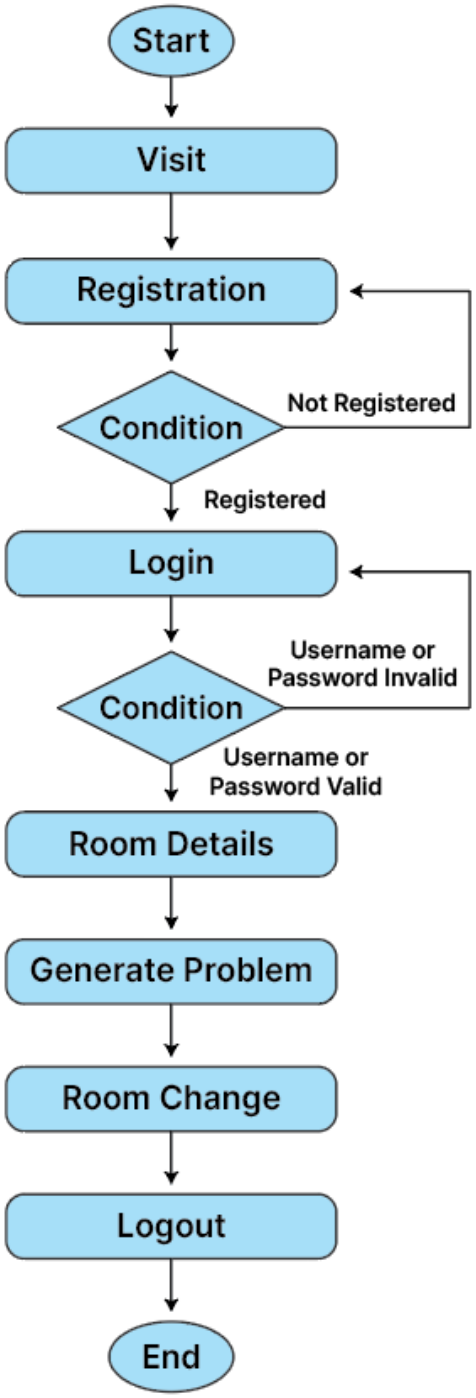


Figure 3.6.1 Student Activity

### 3.6.2 Admin Activity

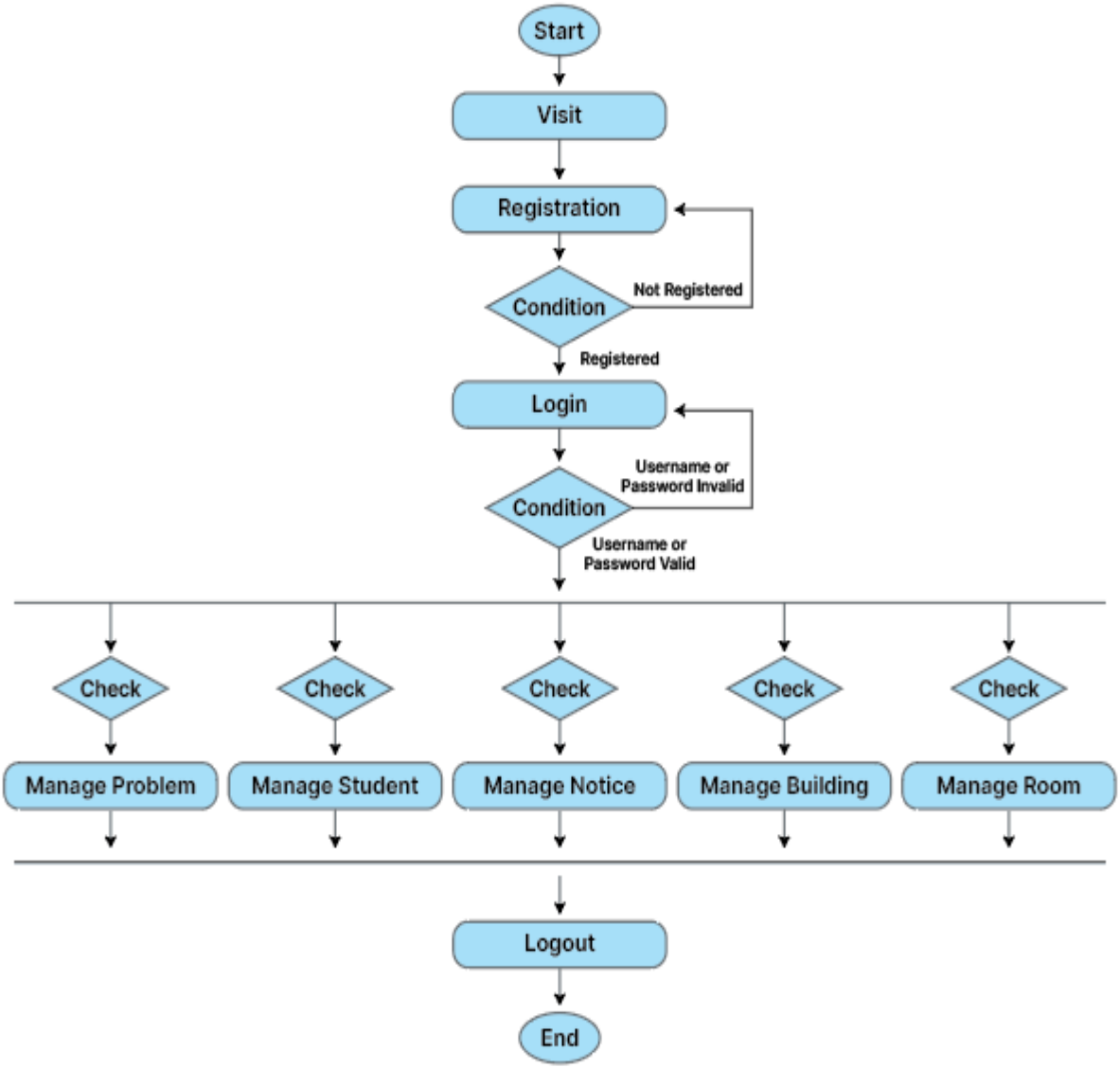


Figure 3.6.2 Admin Activity

## 3.7 Use Case Modeling

### 3.7.1 Use Case Diagram

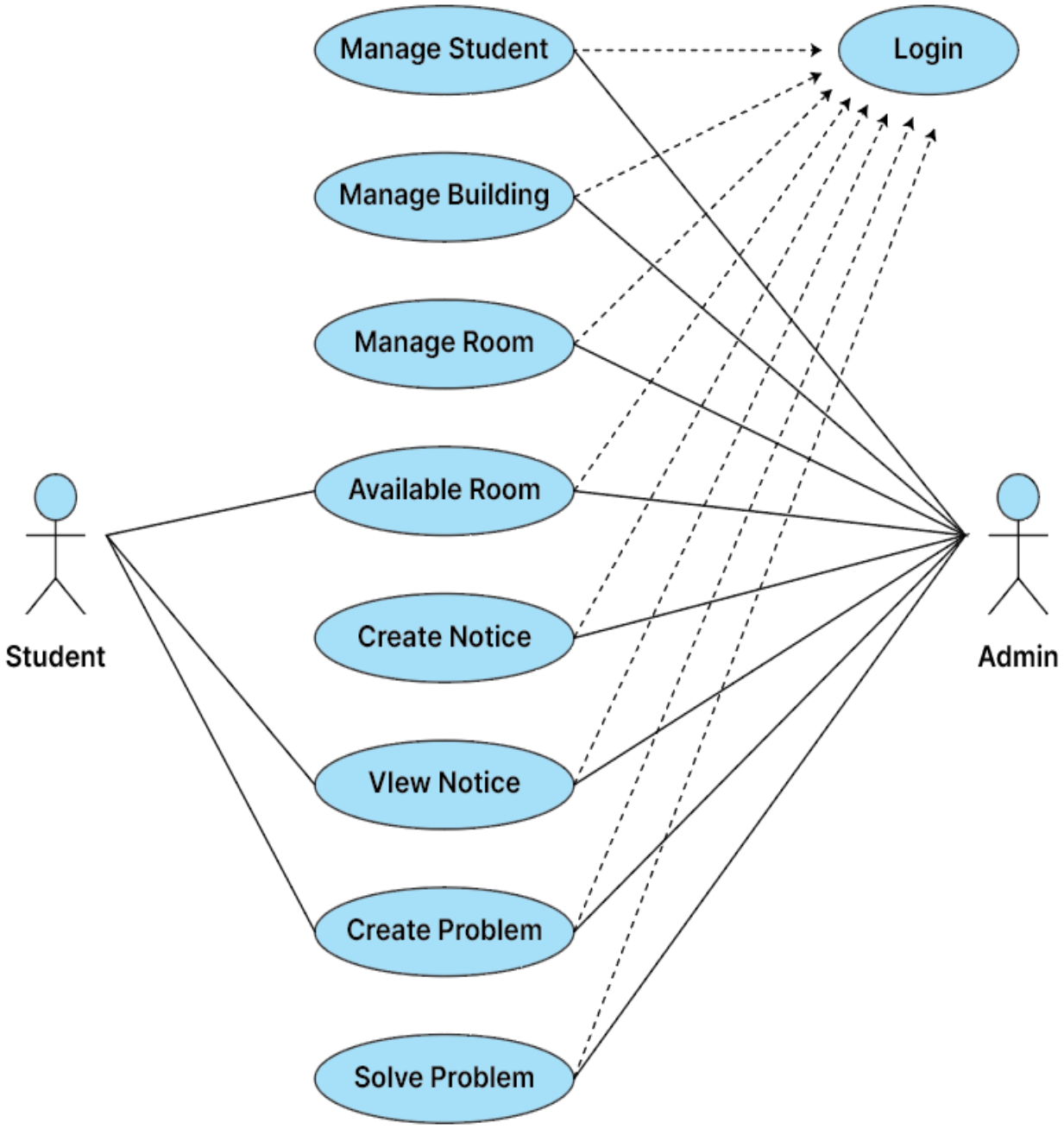


Figure 3.7.1 Use Case Diagram

### 3.7.2 Use Case Description

### 3.7.2.1 Student Manage

| | |
|---|---|
| **Use Case Term** | Student Manage |
| **Actors** | Admin |
| **Flow of Events** | • Add Student<br>• Remove Student<br>• View Student Details |
| **Substitute Flows** | • No student found<br>• Do not add student<br>• Invalid information |
| **Pre-Condition** | Login |
| **Post Condition** | • Confirm student<br>• Delete student |

Table 3.7.2.1 Student Manage

### 3.7.2.2 Room Manage

| | |
|---|---|
| **Use Case Term** | Room Manage |
| **Actors** | Admin |
| **Flow of Events** | • Add room<br>• Remove room<br>• Update room |
| **Substitute Flows** | • Chosen wrong room<br>• Delete incorrect room<br>• Invalid input |
| **Pre-Condition** | Login |
| **Post Condition** | Select right room |

Table 3.7.2.2 Room Manage

### 3.7.2.3 Available Room

| | |
|---|---|
| **Use Case Term** | Available Room |
| **Actors** | Admin |
| **Flow of Events** | • Check available room<br>• Upgrade room |
| **Substitute Flows** | • Chosen wrong room<br>• Don’t update available room<br>• Invalid input |
| **Pre-Condition** | Login |
| **Post Condition** | • Select building<br>• Select room |

Table 3.7.2.3 Available Room

## 3.8 Sequence Diagram

### 3.8.1 Login Operation of Admin

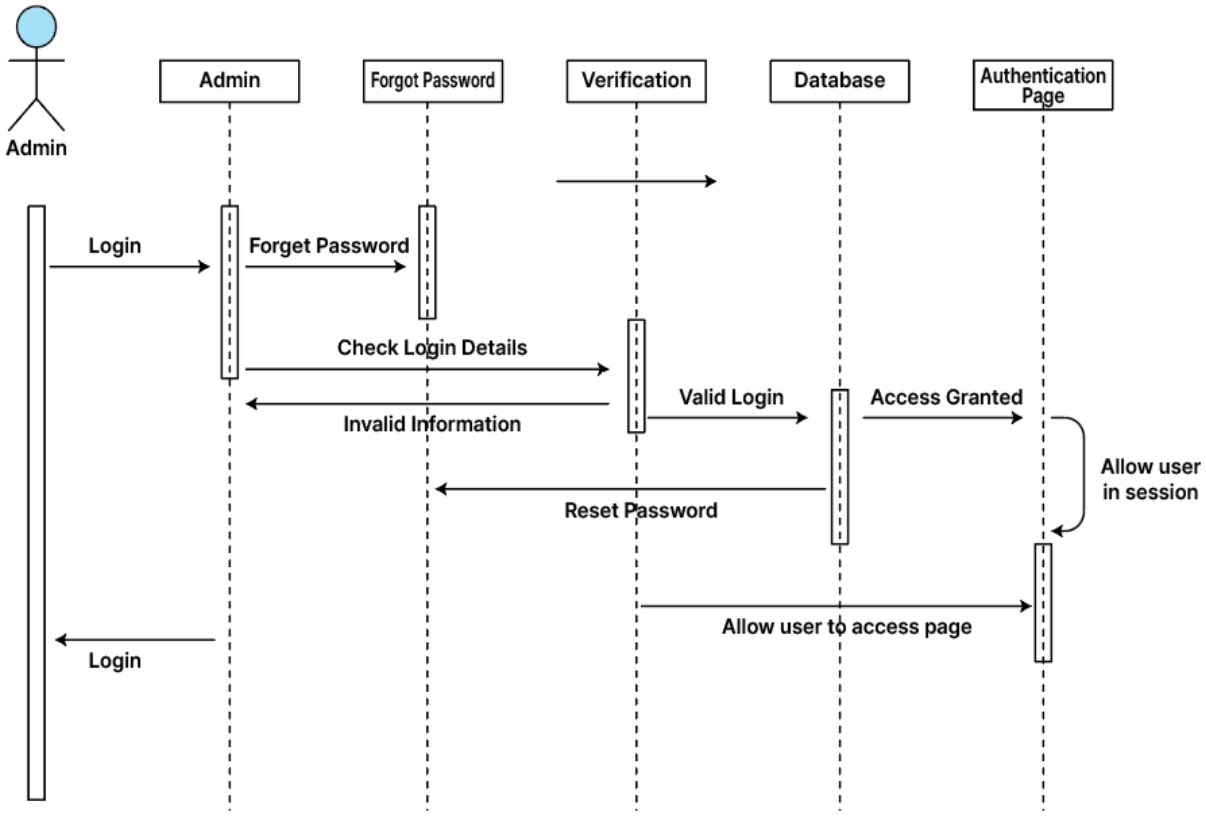


Figure 3.8.1 Login Operation of Admin

### 3.8.2 Admin Operation Process

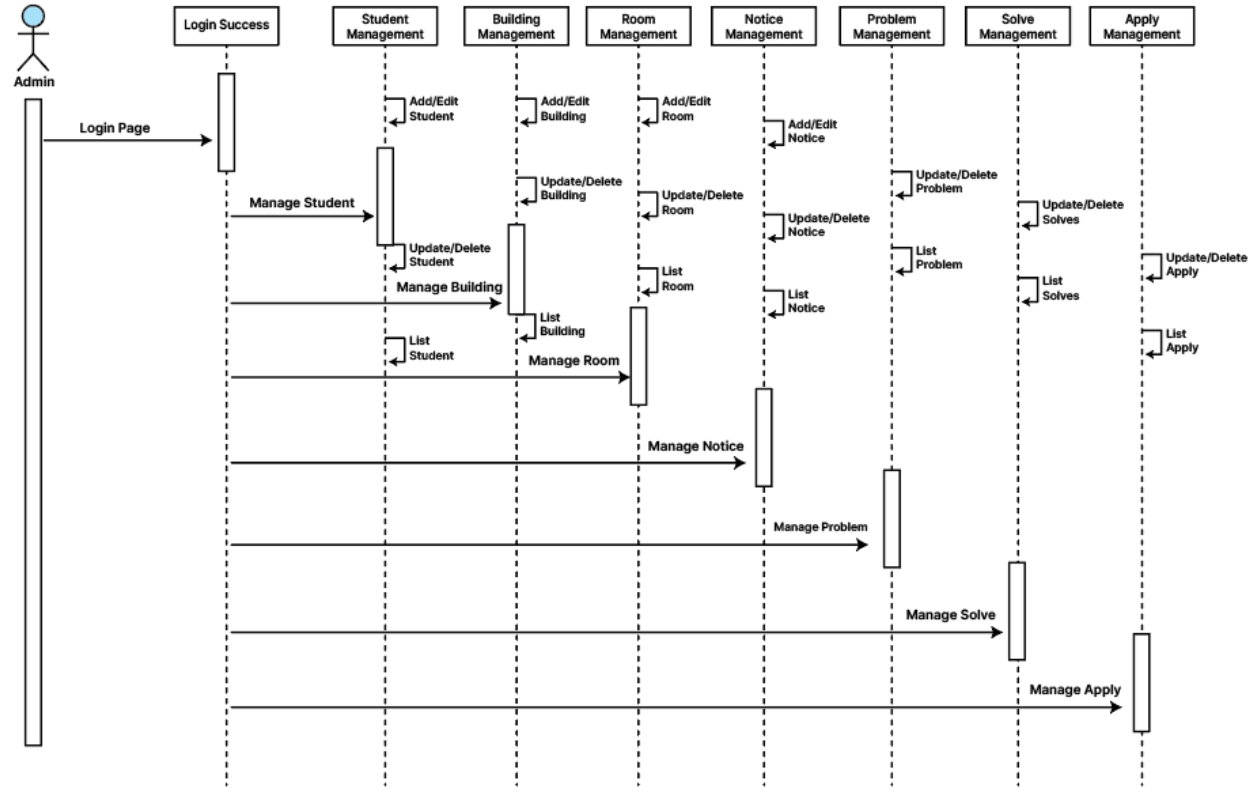


Figure 3.8.2 Admin Operation Process

### 3.8.3 Student Operation Process

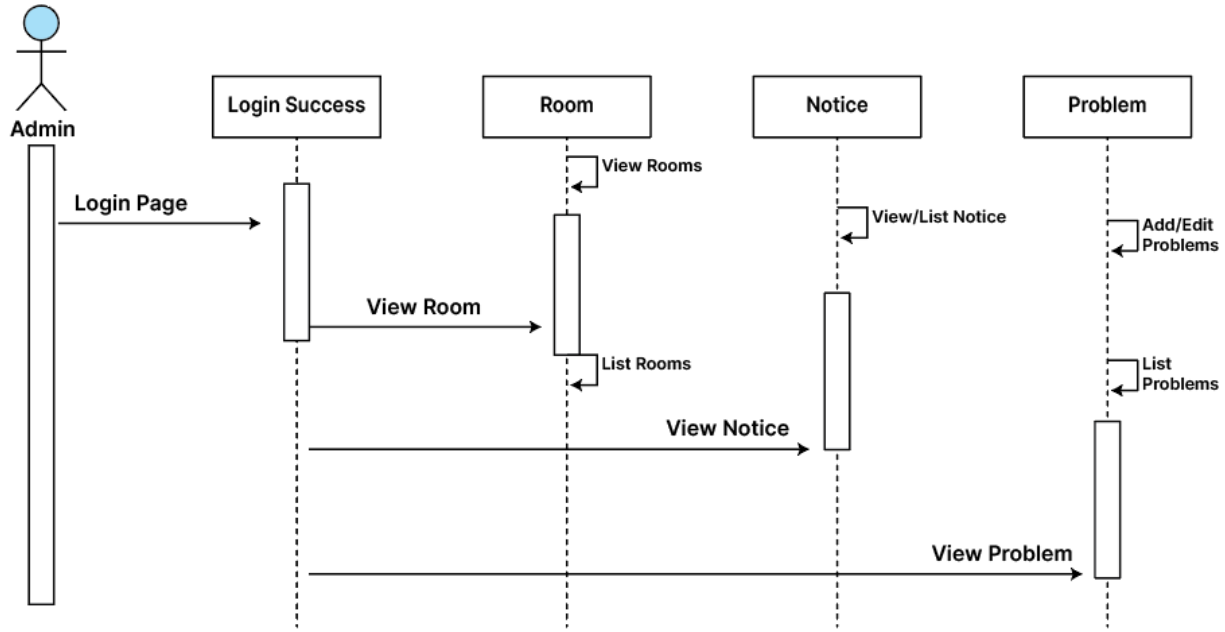


Figure 3.8.3 Student Operation Process

## 3.9 Design of Databases

The information within the system must be kept and accessed from a database. Part of the system design process involves designing the database. During the analysis phase, the necessary data elements and data structures were determined. These elements are organized to establish the system for data storage and access. A database consists of data that is linked together and stored with little duplication to effectively accommodate many users.

The primary goal is to ensure that access to the database is straightforward, rapid, cost-effective, and adaptable for users. Connections are formed between data elements, and any superfluous data points are eliminated. Normalization is applied to achieve internal reliability of data, reduce redundancy, and enhance stability. This process helps to decrease the required data storage, lower the risk of data discrepancies, and optimize updates. MS Access has been selected for the development of the necessary databases.

### 3.9.1 Student Database

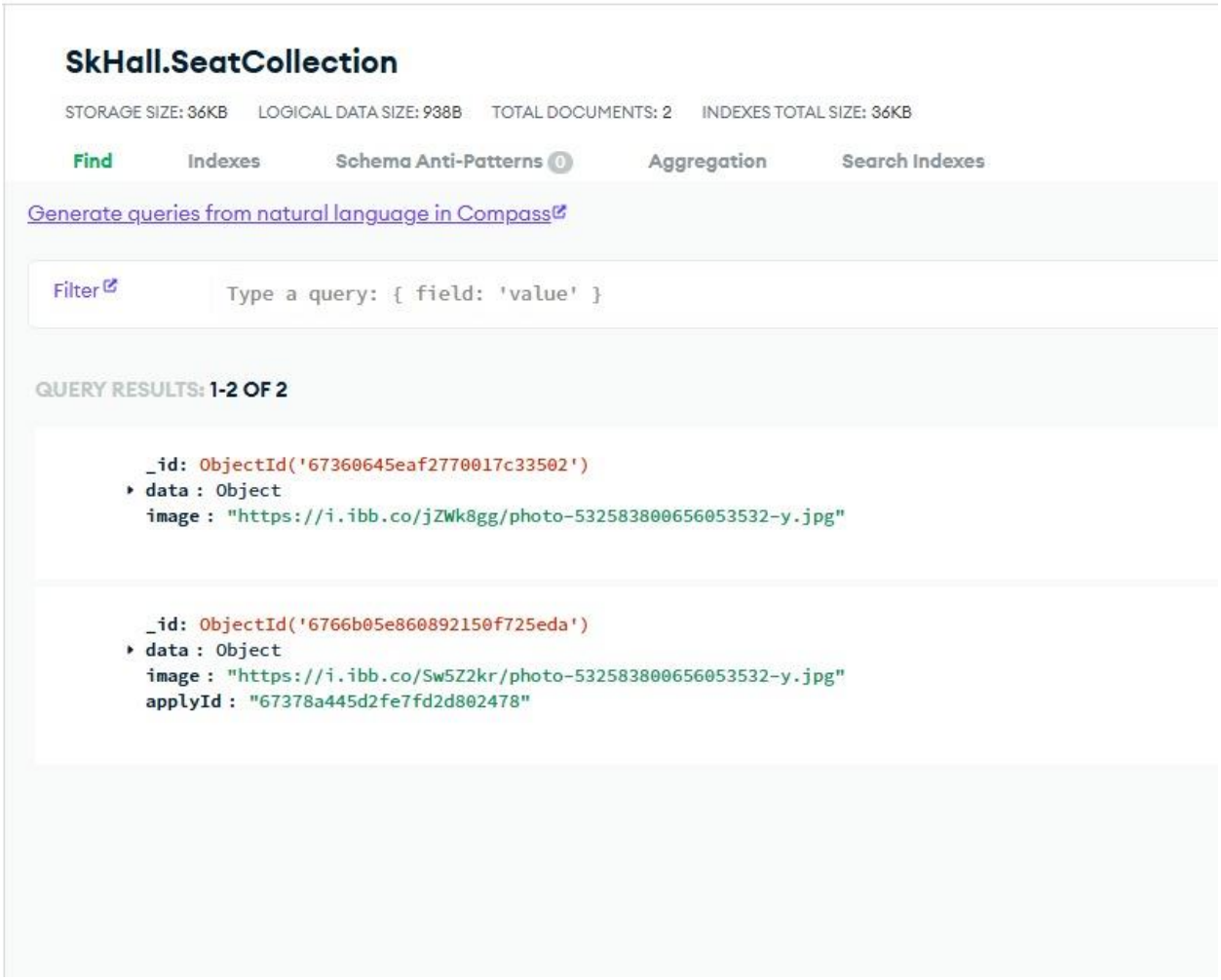


Figure 3.9.1 Student Database

### 3.9.2 Admin Database

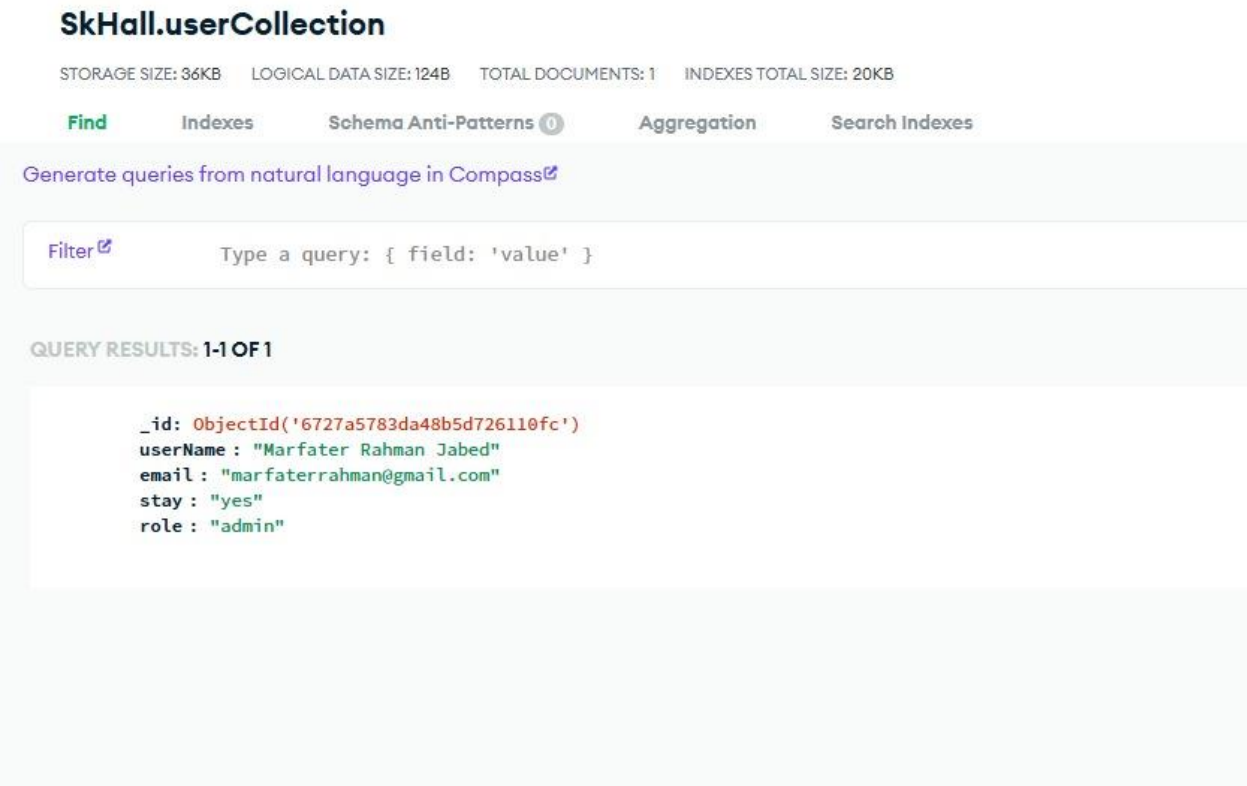


Figure 3.9.2 Admin Database

### 3.9.3 Hall Room Database

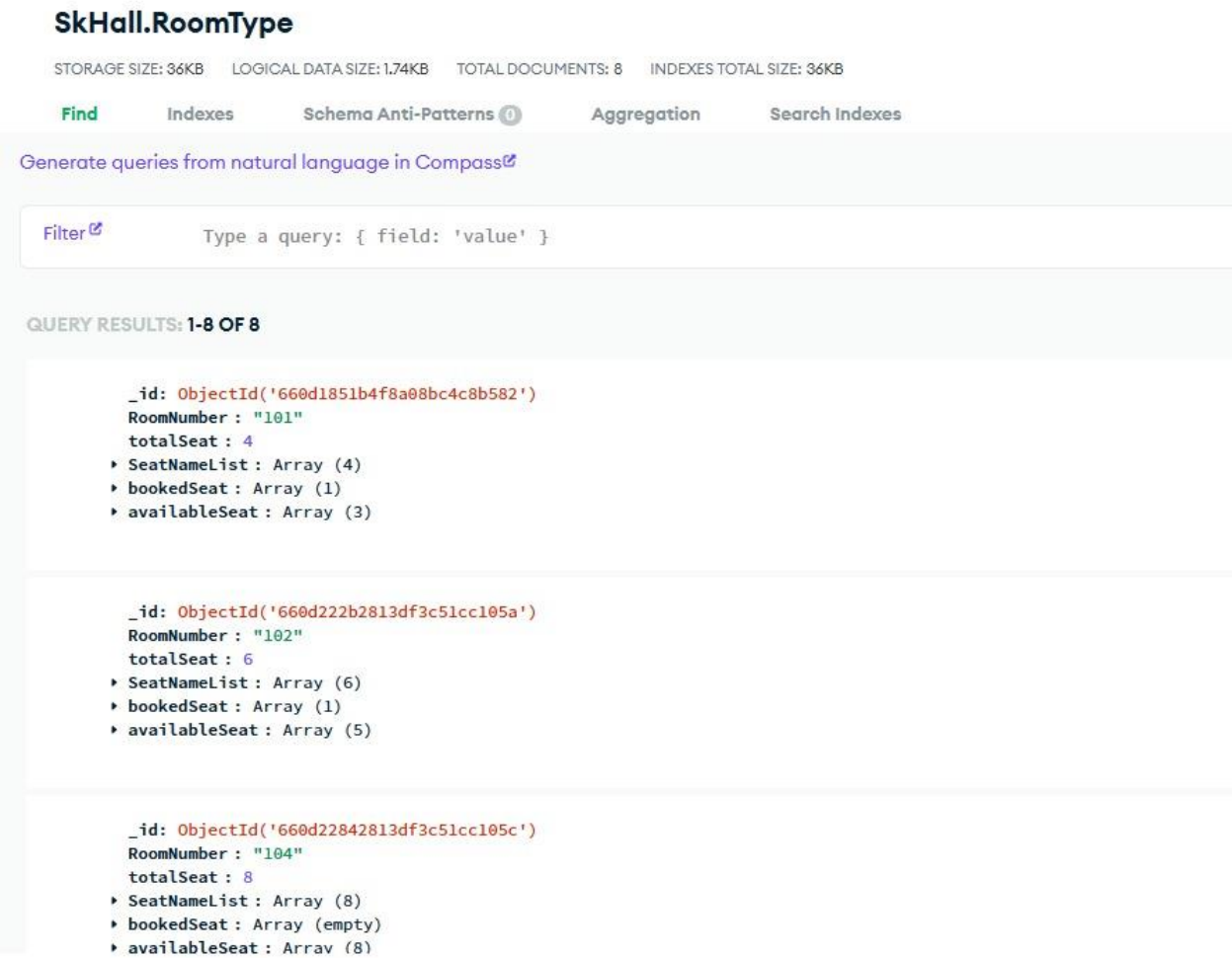


Figure 3.9.3 Hall Room Database

# CHAPTER FOUR

# RESULT, SYSTEM TESTING AND IMPLEMENTATION

## 4.1 Result

The Hall Management System is a web application created using the MERN (MongoDB, Express.js, React.js, Node.js) technology stack. This application enables an administrator to effectively allocate, manage, and cancel hall seats. The completed implementation of the system successfully fulfills the project's objectives, offering a smooth experience for both users and administrators.

The end result of the Hall Management System is a fully operational web application that automates the processes of seat allocation, cancellation, and management. It removes the necessity for manual record-keeping and provides a consolidated platform for hall managers. The system is built to be scalable, allowing for the addition of more features in the future as required.

## 4.2 Output Screen

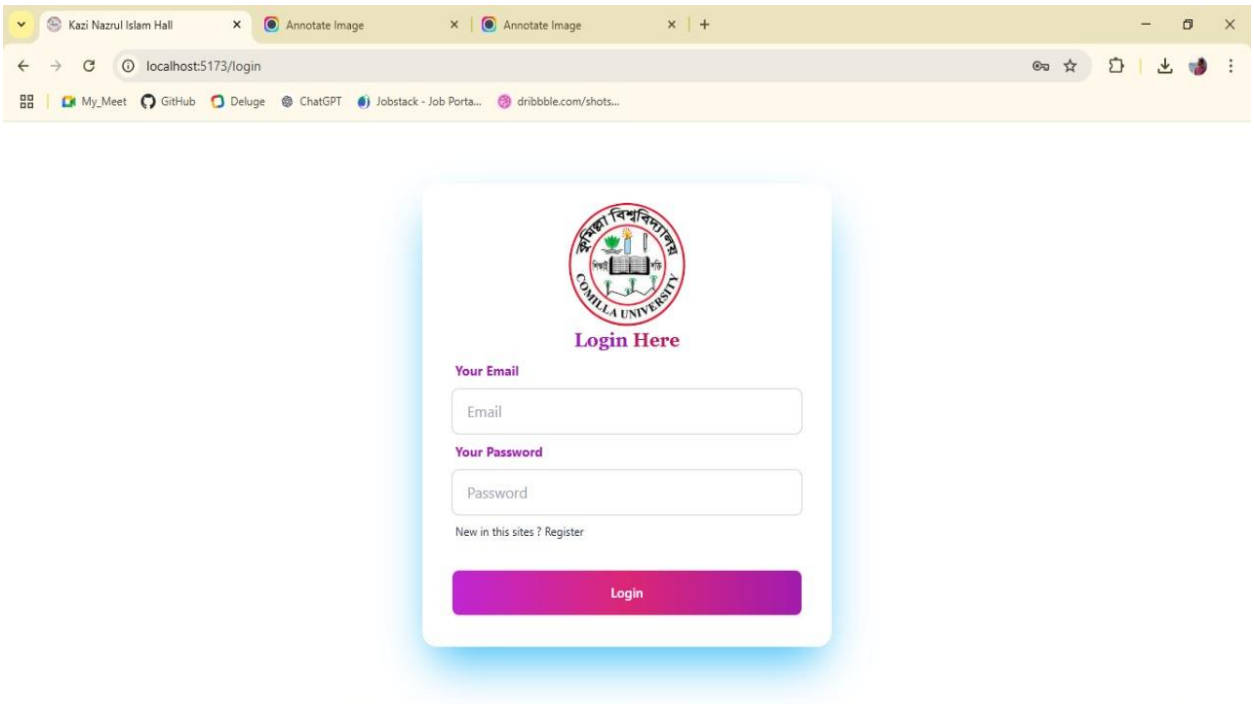


Figure 4.2.1 Login Page

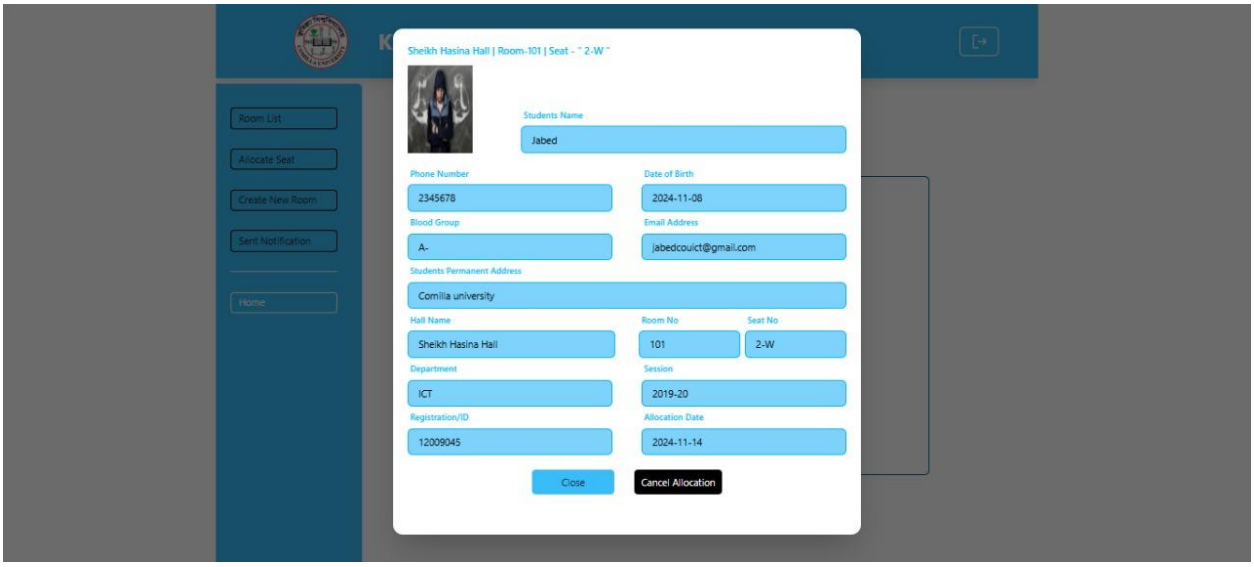


Figure 4.2.2 Residing Student Details

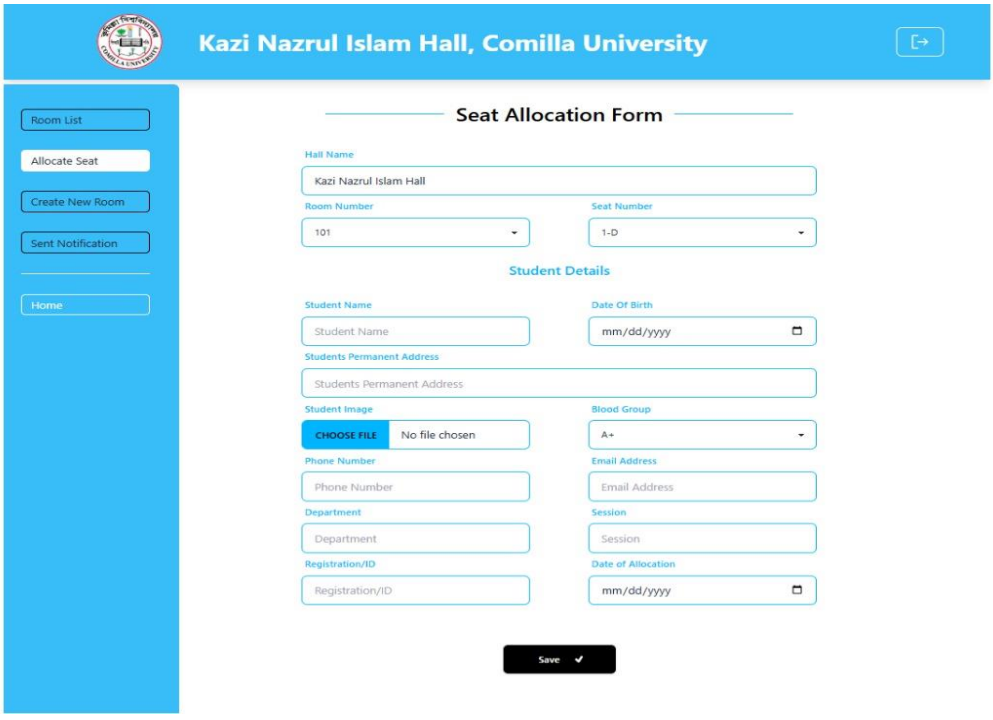


Figure 4.2.3 Seat Allocation Form

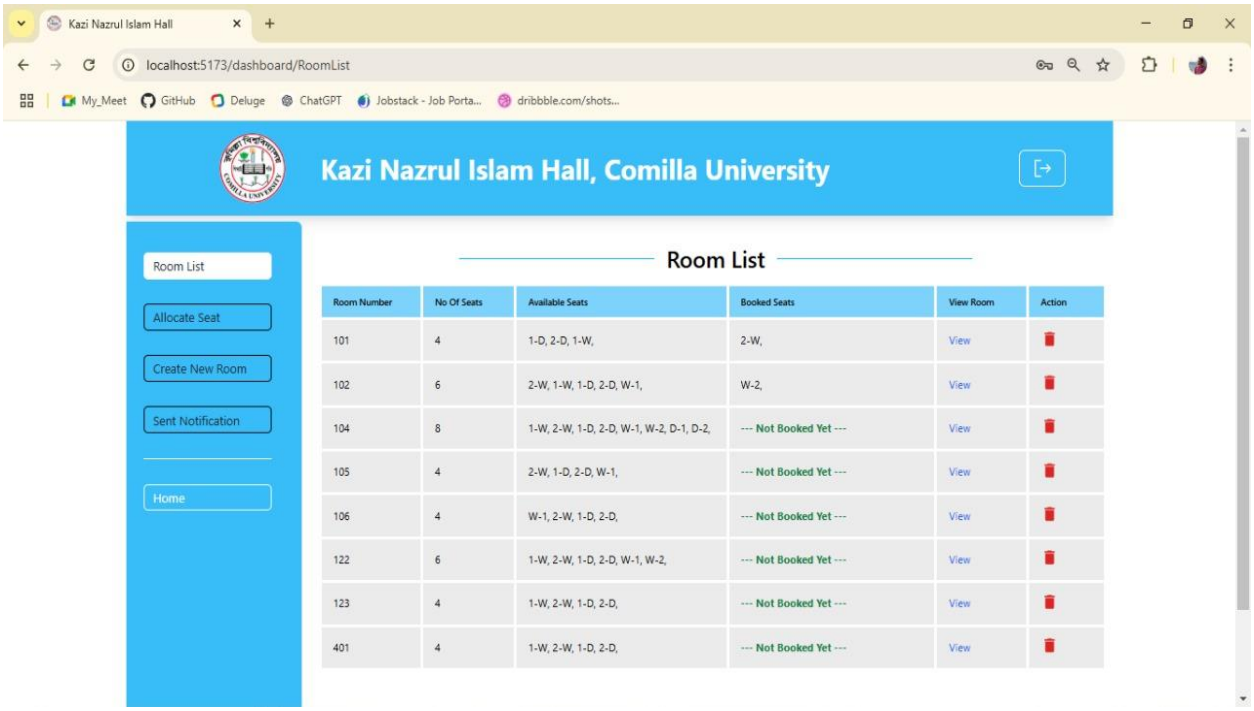


Figure 4.2.4 Room List with Details

## 4.3 System Testing

System testing represents the implementation phase focused on verifying that the system operates correctly and efficiently prior to its live launch. Testing involves running the program with the goal of identifying errors and omitted functions, as well as thoroughly validating that the goals are achieved and user requirements are fulfilled.

The primary objective is to ensure quality. Tests are conducted, and the outcomes are evaluated against the expected documentation. If discrepancies are found, debugging is performed. Through the use of comprehensive testing strategies, a test plan is executed for each module. The different types of tests conducted within the "Network Backup System" include unit testing, integration testing, and user acceptance testing.

### 4.3.1 Testing Individual Modules

The modules and procedures that make up a system's software components are put together and merged to carry out certain tasks. The main goal of unit testing is to find flaws by testing individual components. This allows for the detection of logical and coding errors in every module. The testing process includes inputting data and determining if the value corresponds to the size and type that Java supports. Analysis is done on several controls to make sure they all work as intended.

### 4.3.2 Integration Testing

Data loss can happen through any interface; one module may adversely affect another, and when sub functions are merged, they might fail to provide the anticipated main functions. Integration testing is a systematic approach designed to detect faults in the interface.

The goal is to integrate unit-tested elements into a comprehensive program structure. Together, all of the modules are evaluated as a single unit. In this case, the client and server components are integrated and assessed. A seamless data flow is made possible by this assessment, which ensures that the application functions as a single unit.

### 4.3.3 User Acceptance Testing

The success of any system largely depends on how well users accept it. To assess user acceptance for the system in question, it is essential to maintain regular communication with users throughout the development process and to implement changes as needed.

## 4.4 Implementation

Implementation denotes the stage in a project where the conceptual design evolves into a working system, fostering users' trust in the new system's performance and efficacy. This phase requires careful planning, evaluation of the current system and its implementation limitations, formulation of transition strategies, and examination of the changeover plans. Along with planning, a key responsibility is to instruct and prepare the users.

The process of implementation begins with creating a strategy for the system's launch. This plan details the tasks to be performed, conversations about essential tools and resources, and the acquisition of any extra resources needed for the implementation of the new system. For a network backup system, no additional resources are needed. Execution is the last stage, and it is also the most important one.

Making sure users have faith in a new system's efficacy and performance is a critical component of a successful launch. Only after extensive testing and confirmation that it meets the specified requirements can the system be put into use. Because the previous system may be restored in the event that issues are discovered or certain transactions cannot be handled while utilizing the new system, this approach also guarantees the highest level of security.

### 4.4.1 User Training

Once the system has been successfully implemented, a key responsibility of the developer is to provide education to the users. To facilitate this, user manuals are created and distributed to assist users in navigating the newly developed system. As a result, users receive training on how to effectively utilize the new system. Measures for both hardware and software protection are established to ensure the ongoing functionality of the developed systems in the future. The following steps were undertaken to integrate the new application system:

- Development of user and system documentation, along with the organization of user training sessions that feature demonstrations and practical experience.
- Conducting a test run for a designated period to ensure a seamless transition to the system.
- Training users on the newly introduced features.

- Distribution of user manuals, which detail the procedures for accessing the functions available in the menu, to each user.

It has been verified that the system has been implemented in alignment with user requirements and expectations.

### 4.4.2 Security and Maintenance

System resources are used for maintenance, which also involves the software industry. It proposes returning something to its original state. After conversion, maintenance is carried out because modifications are required to ensure effective operations in response to modifications in the user's environment. Small improvements or fixes for issues that arise while the system is operating are usually included.

In addition, the software itself is improved, reported problems are fixed, and the interface with other hardware or software is adjusted. It is necessary to safeguard any established system from potential threats. Security measures are implemented at different levels to prevent unwanted access to the database. It is necessary to set up a continuous power supply to prevent data loss from voltage fluctuations or power outages. Password protections and basic protocols have been put in place to stop users from gaining unauthorized access.

# CHAPTER FIVE

# CONCLUSION AND FUTURE WORK

## 5.1 Conclusion

The project, crafted with PHP and MySQL, is built according to the user's requirements and the assessment of the current system while allowing for potential future upgrades. The increasing complexity of modern software necessitates an effective approach to software development. Hostel management software is tailored for those looking to oversee various operations within a hostel.

In recent years, the rapid growth of educational institutions has led to an increase in hostels to accommodate students attending these schools. Consequently, this has placed significant pressure on individuals managing the hostels, and software solutions are often not utilized in this setting. This project specifically addresses the challenges of hostel management and mitigates the issues that arise from manual processes.

Recognizing the limitations of the current system paves the way for designing a computerized solution that is compatible with the existing framework while being more user-friendly.

## 5.2 Future Work

It is straightforward to expand the system we have designed. A user could access any of the rooms—whether issued, unissued, or all—according to their preference. If it proves beneficial for users in the future, we aim to develop a more versatile system. Moreover, the challenges we have encountered are something we have plans to address down the line. We have a well-structured approach for this purpose since establishing a DIU Hall is crucial for the educational landscape in our nation. Most importantly, we intend to incorporate a payment gateway using an API method in our project.

Future possibilities:

- We could implement an online transaction or fee payment system.

- The application could be transformed into a website and linked to Comilla University's official website.
- We could incorporate multiple load balancers to alleviate server loads.
- A backup system for the database could be established for enhanced security.
- To enhance the efficiency and precision of room allocations, we might create more sophisticated algorithms that consider factors like student preferences, availability, and compatibility.
- To increase accessibility and convenience for users, we could create a mobile application version of the web application that enables students and administrators to utilize the system while on the go.
- In the future, we might look into incorporating the hall management system with smart building technologies, such as sensors and automation systems, to establish a more energy-efficient and environmentally sustainable campus.

## 5.3 Limitations

Despite our best efforts to design the application to be flexible, intuitive, and user-friendly, there are still some limitations present. While the application offers a wide variety of features for users, certain complex options could not be included due to logistical challenges and a lack of sophistication. Significant effort has been made to ensure the application is easy to use, even for individuals who are not familiar with computers, though it is recognized that a novice may encounter some difficulties initially. Users receive assistance at each stage to facilitate their interaction with the application.

Some limitations include:

- Not available as a website.
- Not integrated with the official Comilla University website.
- Online fee submission not included.

# REFERENCE

[1] “Dhaka University Hall Management Web Portal” du.ac.bd, 2024. [Online]. Available: http://www.du.ac.bd/home/hall_admin/hall. [Accessed: Dec. 06, 2024]

[2] “Shahjalal University of Science and Technology Hall” sust.edu, 2024. [Online]. Available: http://www.sust.edu/campus-life/residence-hall. [Accessed: Dec. 06, 2024]

[3] “Khulna University of Engineering and Technology” kuet.ac.bd, 2024. [Online]. Available: http://www.kuet.ac.bd/index.php/welcome/khajahalldetails. [Accessed: Dec. 08, 2024]

[4] "Design and Development of Hostel Management System using Web Technology." By R. Rajeswari and K. R. Sridevi, International Journal of Computer Science and Information Technologies, Volume 6, Number 3 (2015). [Accessed: Jan. 09, 2025]

[5] "Development of a Web-Based Hostel Management System for Tertiary Institutions." by G. O. Okegbemi and S. O. Oyebisi, Journal of Engineering and Applied Sciences, Volume 12, Number 1 (2017). [Accessed: Jan. 26, 2025]

[6] "Design and Implementation of a Smart Dormitory Management System Based on the Internet of Things." by Yanhui Zhu, International Journal of Future Computer and Communication, Volume 8, Number 1 (2019). [Accessed: Jan. 26, 2025]

[7] "Design and Implementation of a Dormitory Management System Based on Web Services." by Hongbin Wang, Journal of Physics: Conference Series, Volume 1156, Number 4 (2019). [Accessed: Jan. 29, 2025]

[8] "A Web-Based Dormitory Management System for Higher Education Institutions." By C. L. Bautista and M. F. L. Asistio, International Journal of Computer Applications, Volume 179, Number 45 (2018). [Accessed: Feb. 06, 2025]