

HOSTEL MANAGEMENT SYSTEM

Project submitted to the
SRM University - AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology In
Computer Science and Engineering
School of Engineering and Sciences

AUTHORED BY

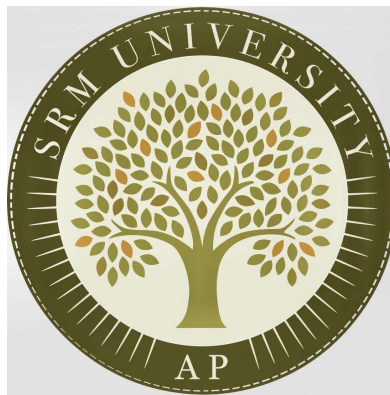
Harshith - [AP20110010002]

Narasimha - [AP20110010017]

Vignesh - [AP20110010057]

Siva Kumar - [AP20110010047]

Jayanth - [AP20110010049]



SRM University-AP
Neerukonda, Mangalagiri, Guntur
Andhra Pradesh - 522 240
[May 2023]

Contents

1 INTRODUCTION	3
1.1 Purpose:	3
1.2 Scope:	3
1.3 Objectives:	4
1.4 Definition:	4
2 SYSTEM OVERVIEW	4
2.1 Architectural Style:	4
2.2 DFD:	5
2.2.1 Context diagram-	5
2.3 Architectural Mapping with DFD:	5
2.3.1 Level-0-	6
2.3.2 Level-1-	7
2.4 System components:	7
2.5 System Interfaces:	8
3 FUNCTIONAL REQUIREMENTS	8
3.1 ACTORS:	9
Admin-	9
User-	10
4 USE CASE DIAGRAM	11
4.1 USE CASE DESCRIPTION	12
4.2 ACTIVITY DIAGRAM	14
4.3 CLASS DIAGRAM	15
4.4 SEQUENCE DIAGRAM	16
5 ER-DIAGRAM	17
6 NON-FUNCTIONAL REQUIREMENTS	18
6.1 Performance requirements:	18
6.2 Usability Requirements:	18
6.3 Security Requirements:	18
6.4 Reliability Requirements:	19
6.5 Scalability Requirements:	19
6.6 Maintainability Requirement:	19
7 CONCLUSION	20
8 FUTURE SCOPE	20
9 REFERENCES	21

1 INTRODUCTION

The hostel management system project maintains and stores information regarding the students that are present in the hostel along with their room details and block numbers which meet the needs of the end user. This system allows both the users and admins to keep the track of all the rooms which are available in the hostel. A Hostel Management System with room booking, registration, and leave request functionality is a software application specifically designed to assist in the efficient management of hostels or similar accommodation facilities. It combines essential features related to room reservation, guest registration, and handling leave requests, enabling hostel administrators to streamline operations and enhance the overall experience for guests.

1.1 Purpose:

A hostel management system software aims to automate and streamline the administrative tasks involved in managing a hostel, including registration of guests, room allocation, fee management, attendance tracking, and other related tasks. Hostel management system software can reduce the amount of time and help maintain accurate records of guest details and help maintain a secure and safe environment.

1.2 Scope:

The main objective of a hostel management system is to efficiently manage the day-to-day operations of a hostel, enhancing the overall experience for both guests and administrators. By automating manual tasks and providing centralized control, it enables hostel managers to save time, reduce errors, and improve productivity.

This software enables user to store all the data about the students that are present in the hostel and also contains the block and room information of all the students that are present in the hostel. In this software admin has access to all the information about the students and their room allocation details.

Students can also apply for leave through the website by specifying the reason the required dates where as Admin can see the leave details and approve the leave of the students through the website itself.

1.3 Objectives:

The objectives of the library management system are as follows:

- To provide an efficient and effective way of managing the student information and their data
- To improve the overall user experience by making it easy for users to find and borrow books
- To efficiently manage the day-to-day operations of a hostel
- To enhance the overall experience for both guests and administrators by automating manual tasks and providing centralized control

1.4 Definition:

User-The person accessing the Hostel management system.

Admin-One who controls the Hostel management system.

DFD: Data Flow Diagram

ERD: Entity-Relationship Diagram

2 SYSTEM OVERVIEW

The library management system will follow a client-server architecture. The client will be a web-based application that can be accessed through a web browser, while the server will be responsible for managing the database and other backend services.

2.1 Architectural Style:

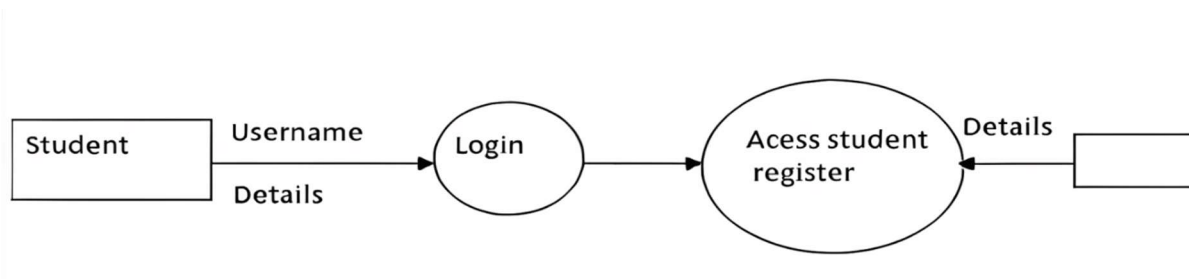
A hostel management system was created to provide a computerised process that is stress-free, dependable, and quick to both students and staff in charge of the registration and hostel management processes by utilising the PHP computer programming language and the MySQL database application. The front-end would be HTML, which would give the graphical user interface that interacts with the user, while the back-end would be the MySQL database, which would manage the data storing process.

2.2 DFD:

The Data Flow Diagram (DFD) is a graphical representation of the system's data flows. The DFD will be used to illustrate the flow of data between the system components.

2.2.1 Context diagram-

- The Context Diagram provides an overview of the entire library management system as a single process.
- It shows the external entities interacting with the system, such as students, librarians, and the library database.



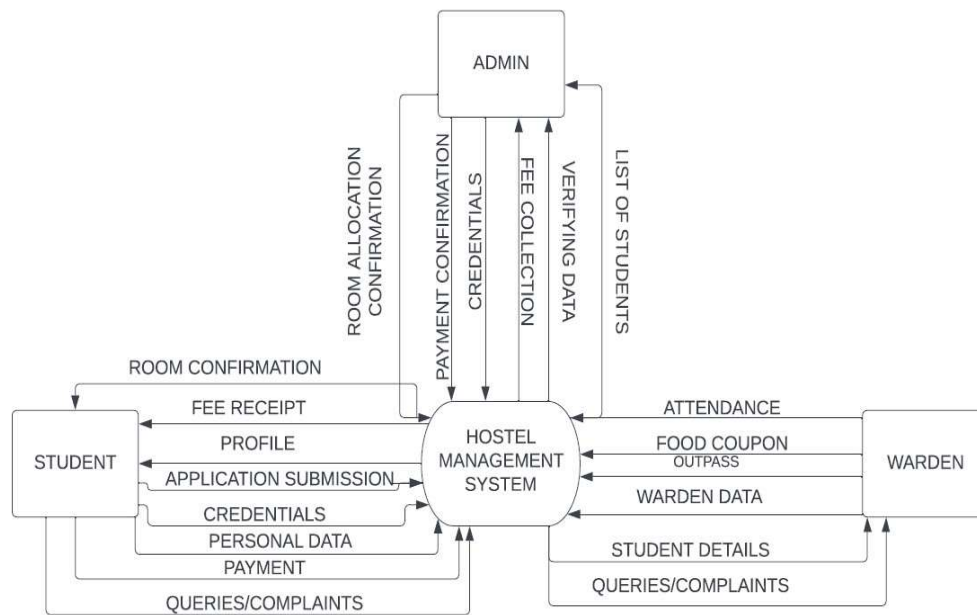
2.3 Architectural Mapping with DFD:

The DFD will be mapped to the system's architectural components as follows:

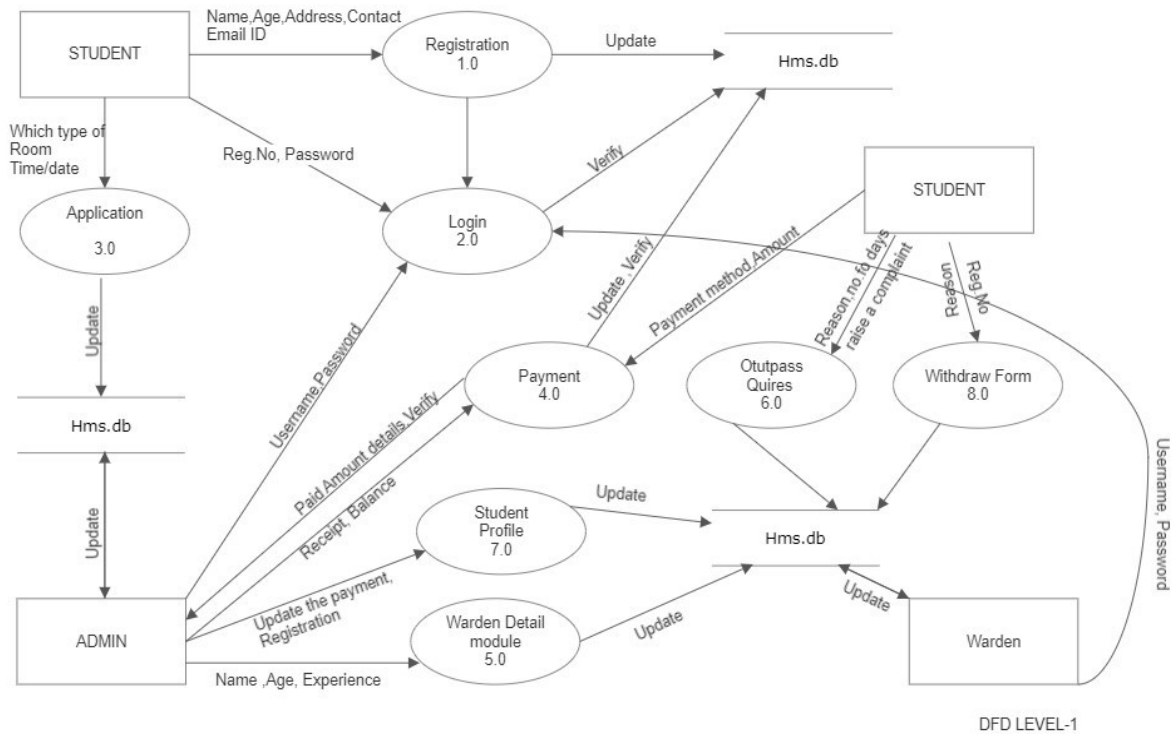
Level 0 DFD: client (student) and Server (Administrator)

Level 1 DFD: User Management, Room Allocation, Room Identification

2.3.1 Level-0-



2.3.2 Level-1-



2.4 System components:

The Hostel management system will consist of the following components:

- Client: Web-based user interface
- Server: Backend services, including the database and controllers
- Database: Stores the student data, students room allocation details, room details and leave requests
- Administrator: Manages all the student information and their room details

2.5 System Interfaces:

The Hostel management system will have the following interfaces:

- Web-based user interface: Allows users (students) to interact with the system through a web browser
- Database interface: Allows the system and the admin to interact with the database and perform data operations

3 FUNCTIONAL REQUIREMENTS

Login page: The user (student) must login with the credentials in the webpage to enter into the software and can access all the functions

Room search: Users can search their room details by accessing into the portal and can also see the information about the fellow roommates

Registration Page: New user should register with their registration number and by entering password with correct requirements to access into the software

Admin: The admin should login into the website with the different login portal with the admin credentials and can access the student details

Applying leave: The Student/User can apply for the leave through the portal by specifying the reason and the dates of the leave

Leave Granting: The admin can approve the leave by logging in into the portal by admin login page and see the information about the leave details and can grant them .

3.1 ACTORS:

3.1.1 Admin-

Admin has full access to the system which means he is able to manage any activity with regard to the system. He is the highest privileged user who can access the system.

KEY FUNCTIONS:

- Access student records, and warden Records.
- Add new warden/student entry in the system database.
- Confirm Payment and Generate Bill.
- View Student Info. (Total no of students added/removed, fee collections).

1. Admin logs in the system.
2. Admin may add/remove the students/warden.
3. Admin view student data.
4. Admin Verify the payment submitted by the students.
 - 4.1 Generate Bill/receipt and confirmation message for the Student

3.1.2 User-

- **STUDENT:**

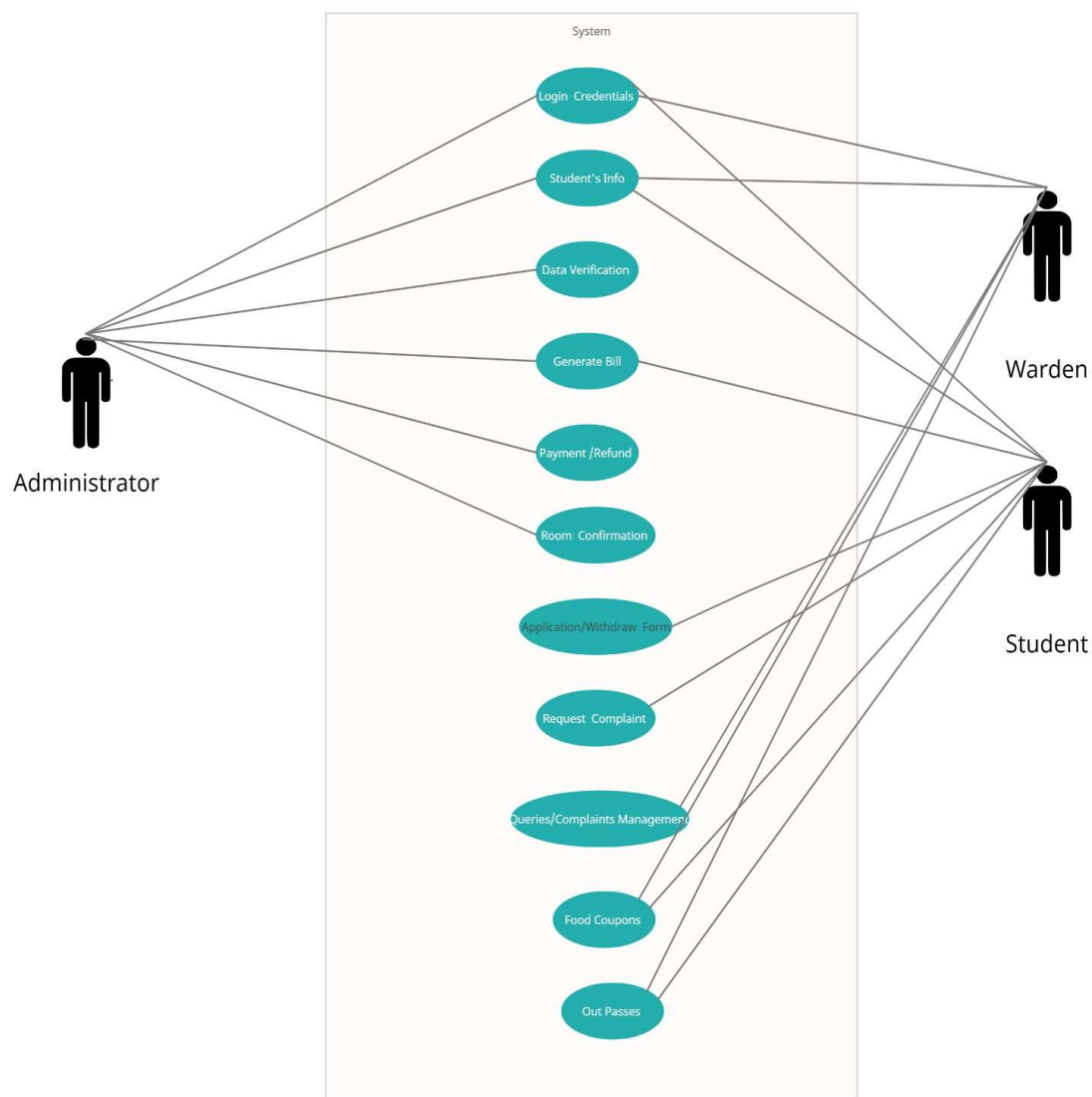
DESCRIPTION – The new student can register themselves and add their details like name, age, gender, etc. The student entry will be made in the Hms database. They can use the system to book a room, check in and check-out, view their room and roommate details, request maintenance or cleaning services, make payments, and submit feedback or complaints. They can also use the system to access important information related to the hostel, such as the hostel rules and regulations, events and activities, emergency contact information, and resources such as laundry facilities or study rooms. The system may also have a communication feature that allows students to connect with hostel management staff or other students in the hostel.

MAIN FLOW OF EVENTS

1. Student selects sign up in the login module.
2. A registration form gets displayed.
3. Student fills in the required details.

4 USE CASE DIAGRAM

A use case diagram is a type of visual representation of a system that illustrates the interactions between system components (actors) and their specific use cases. It is a high-level diagram that provides an overview of the system's functionality from a user's perspective.



4.1 USE CASE DESCRIPTION

- **STUDENT:**

DESCRIPTION – The new student can register themselves and add their details like name, age, gender, etc. The student entry will be made in the Hms database.

PRE-CONDITION – The student must be a new student, If necessary fields are left by the user then prompt the user to fill in the necessary fields.

POST CONDITION – Student record is added to hms database.

MAIN FLOW OF EVENTS

1. Student selects sign up in the login module.
2. A registration form gets displayed.
3. Student fills in the required details.

- **ADMIN:**

DESCRIPTION- The admin add the warden/student, updates the student's details and verifies the payment, and generates Bill/receipt for the same.

PRE-CONDITION- Admin must first log in with his/her credentials

POST CONDITION-The hms database is updated

MAIN FLOW OF EVENTS

5. Admin logs in the system.
6. Admin may add/remove the students/warden.
7. Admin view student data.
8. Admin Verify the payment submitted by the students.
 - 4.2 Generate Bill/receipt and confirmation message for the Student

- **WARDEN:**

DESCRIPTION- The warden can provide the students the out passes and food coupons. And wardens will sort out the queries/complains

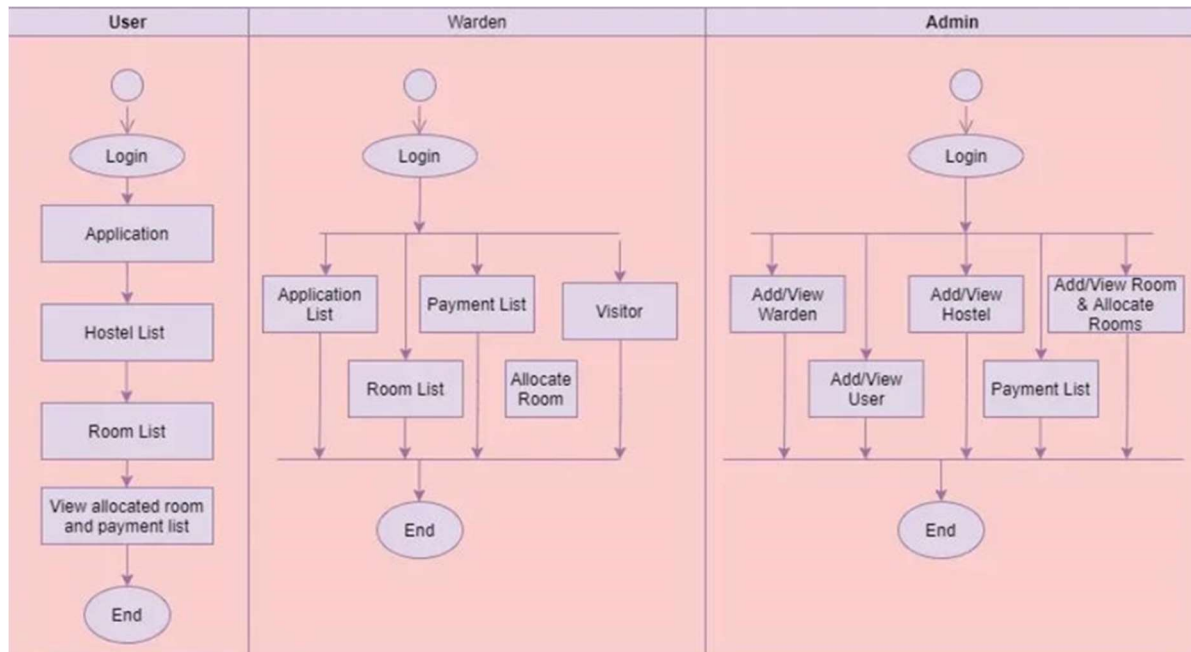
PRE-CONDITION- The warden must first log in with his/her credentials.

POST CONDITION-The hms database is updated.

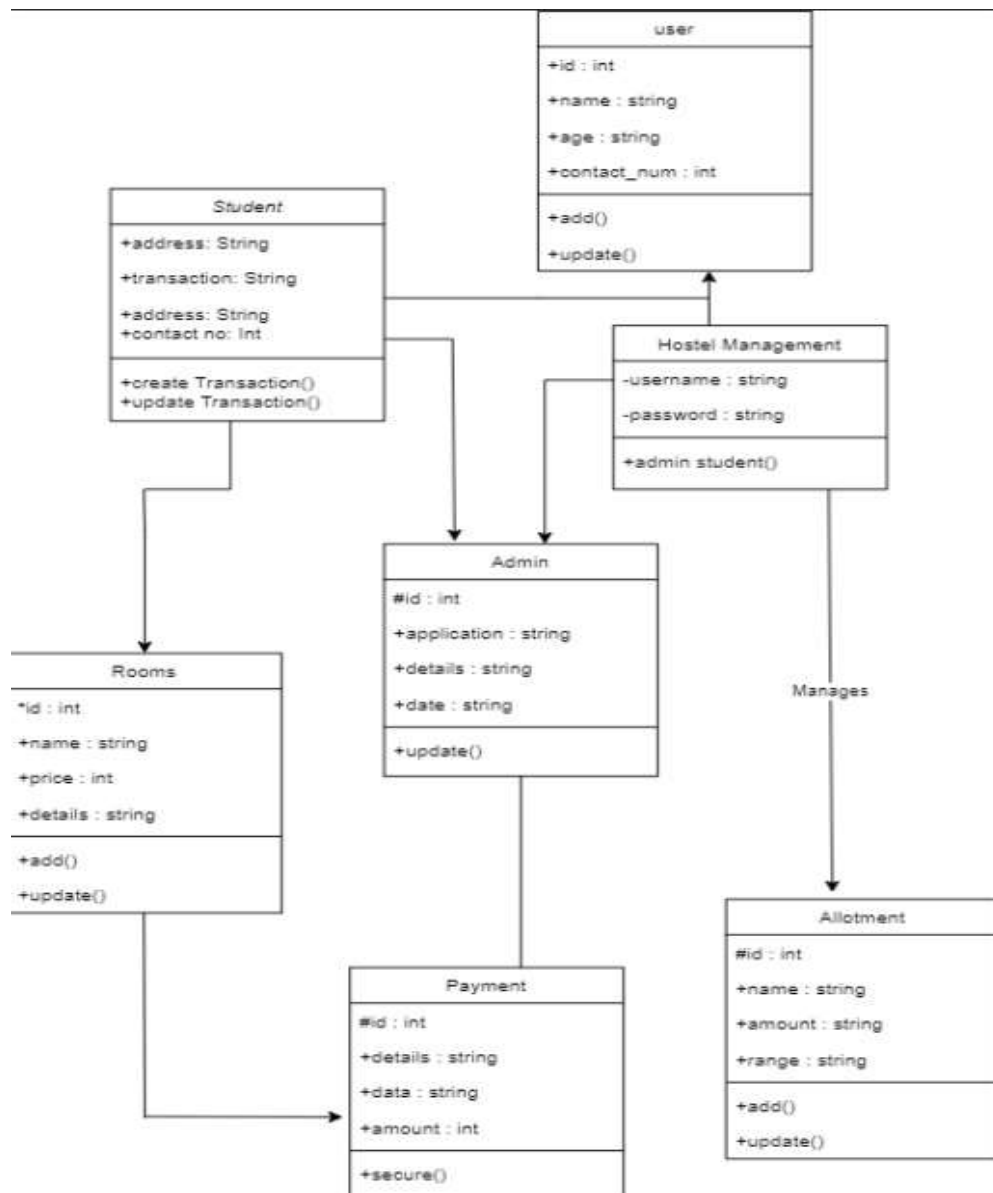
MAIN FLOW OF EVENTS

1. Warden logs in the system.
2. Warden views student data.
3. Warden provides out passes/food coupons.

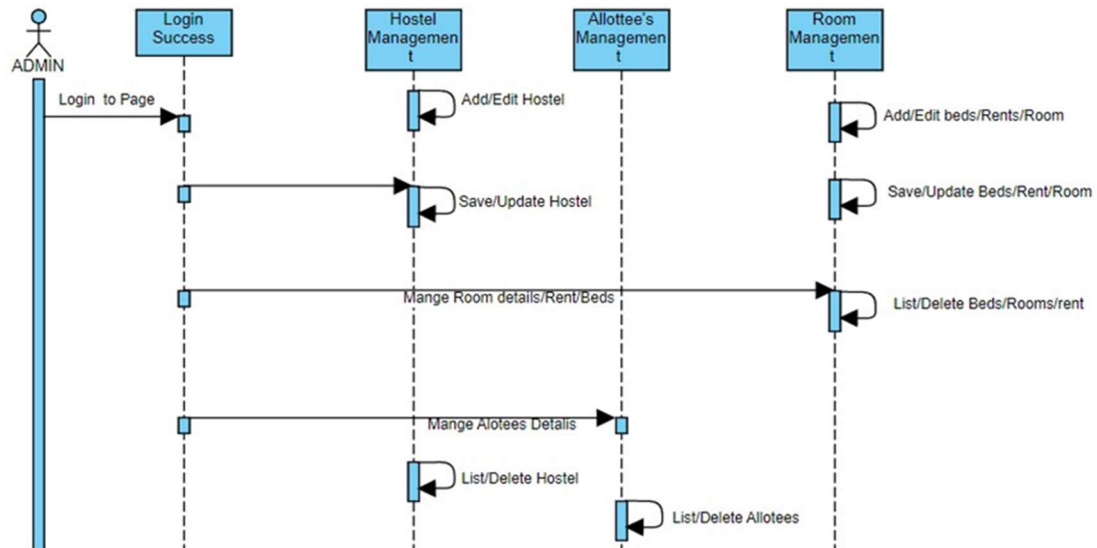
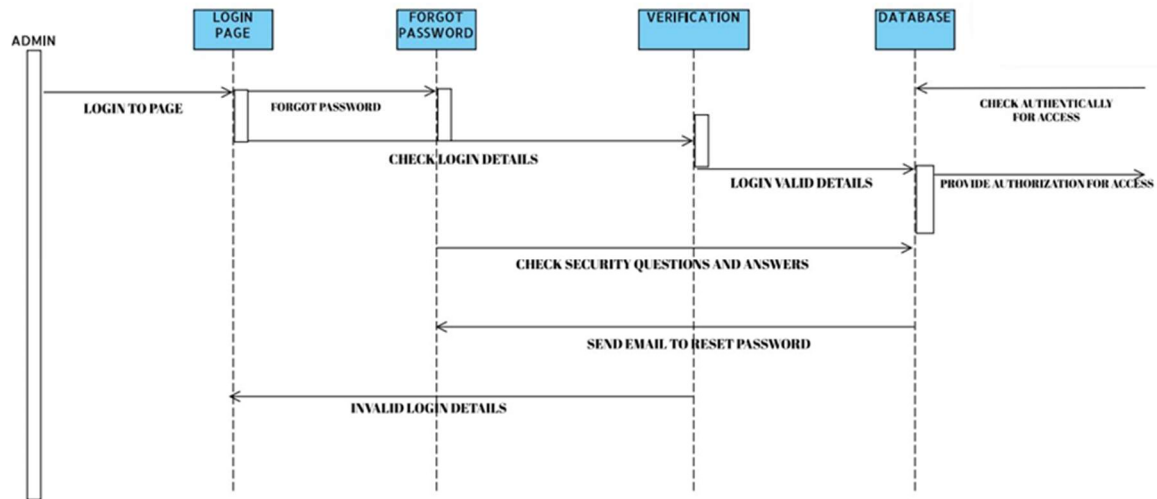
4.2 ACTIVITY DIAGRAM



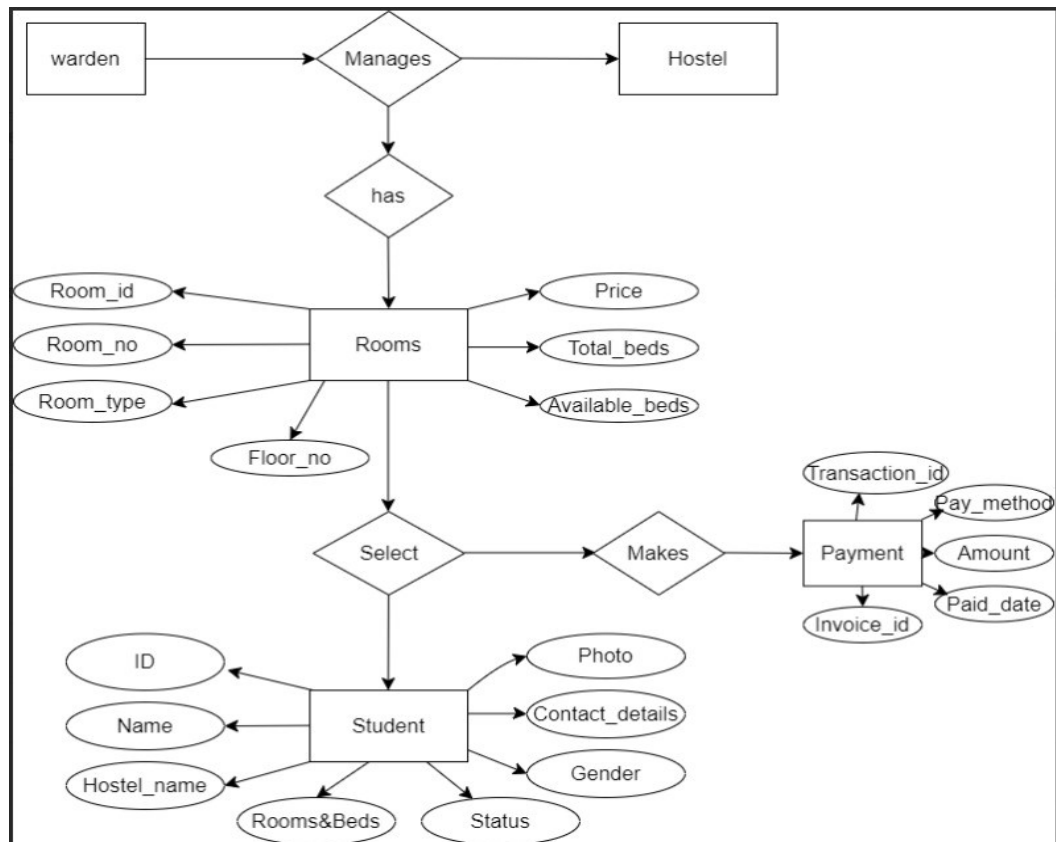
4.3 CLASS DIAGRAM



4.4 SEQUENCE DIAGRAM



5 ER-DIAGRAM



6 NON-FUNCTIONAL REQUIREMENTS

6.1 Performance requirements:

- Speed: The software should be able to handle large amounts of data quickly during peak usage hours of the software
- Reliability: The software should be available at all the time without breakdowns
- Security: The software contains a database with user information so security should be maintained
- Scalability: The software should also handle the increasing number of users and their data

6.2 Usability Requirements:

- User Interface: The user interface should be easy to use and visually appealing to user
- Input validation: Input validation is required to validate the user inputs (login details)
- Error Handling: The system should provide meaningful Error messages for invalid inputs

6.3 Security Requirements:

Database is secured with all the necessary requirements and technologies.

- Authentication: The software should have strong authentication to allow only authorized users
- Access Control: Better access control mechanisms are required to cut access to sensitive data based on their roles and permissions
- Backup – Recovery: The database which contains all the information about users should be stored for backup and recovery

6.4 Reliability Requirements:

- Availability: The system should be available to users at all times, with minimal downtime for maintenance
- Recovery: The system should be able to recover from errors or failures quickly and efficiently
- Maintenance: The system should be maintained regularly, with updates and upgrades applied as needed to ensure that it remains reliable and up-to-date.
- to ensure reliable operations.

6.5 Scalability Requirements:

- Performance: The software should have good performance, even when there are large amounts of data or heavy user traffic
- Elasticity: The software should adapt the changing User demands and increase of users
- Data Management: The system should have efficient data management mechanisms that allow for the easy storage, retrieval, and manipulation of large amounts of data.
- Load Balancing: The system should maintain excessive load of multiple users

6.6 Maintainability Requirement:

- The system should be designed and implemented in a modular and well-documented manner to facilitate easy maintenance and updates.
- Code should be structured and organized, following best practices and standards to ensure readability and ease of modification.

7 CONCLUSION

The project is based on the user's requirement specification and an analysis of the existing system, with flexibility for future extension.

The increased functionality of today's software necessitates a more systematic approach to software development. This hostel management software is for those who want to keep track of a variety of activities in the hostel. The number of educational institutions has been significantly expanding in recent years. As a result, the number of hostels available for students studying at this university is expanding. As a result, the person in charge of the hostel is under a lot of stress, and software is rarely employed in this situation. This solution addresses the challenges of running a hostel and avoids the issues that arise when done manually.

The identification of the existing system's flaws leads to the creation of a computerized system that is compatible with the existing system while being more user-friendly and GUI oriented.

8 FUTURE SCOPE

There are various possible areas for additional expansion and improvement of the Student Hostel Management System project in the future. Integrating artificial intelligence (AI) skills to automate operations such as room allocation, anticipating occupancy trends, and selecting roommates based on preferences and compatibility may be among them. In addition, integrating a comprehensive financial management module allows students to examine and control their expenses, such as rent, utilities, and food plans, all within the system. In addition, incorporating a sophisticated reporting and analytics tool can give hostel administrators with useful insights into occupancy rates, student satisfaction levels, and resource utilisation, enabling for data-driven decision-making and ongoing improvement. Integrating the system with mobile apps can improve accessibility and convenience for students by allowing them to check room availability, submit maintenance requests, and more.

9 REFERENCES

- <https://www.ijraset.com/research-paper/hostel-management-system-hms>
- <https://sourceforge.net/projects/hostel-management-system/>
- <https://www.codementor.io/projects/hostel-management-system-atx32n08vq>
- <https://www.php.net/manual/en/index.php>
- <https://www.php.net/manual/en/getting-started.php>
- <https://www.youtube.com/playlist?list=PL4cUxeGkcC9gksOX3Kd9KPo-O68ncT05o>