ANALYSIS AND DESIGN

1. CLASS DIAGRAM

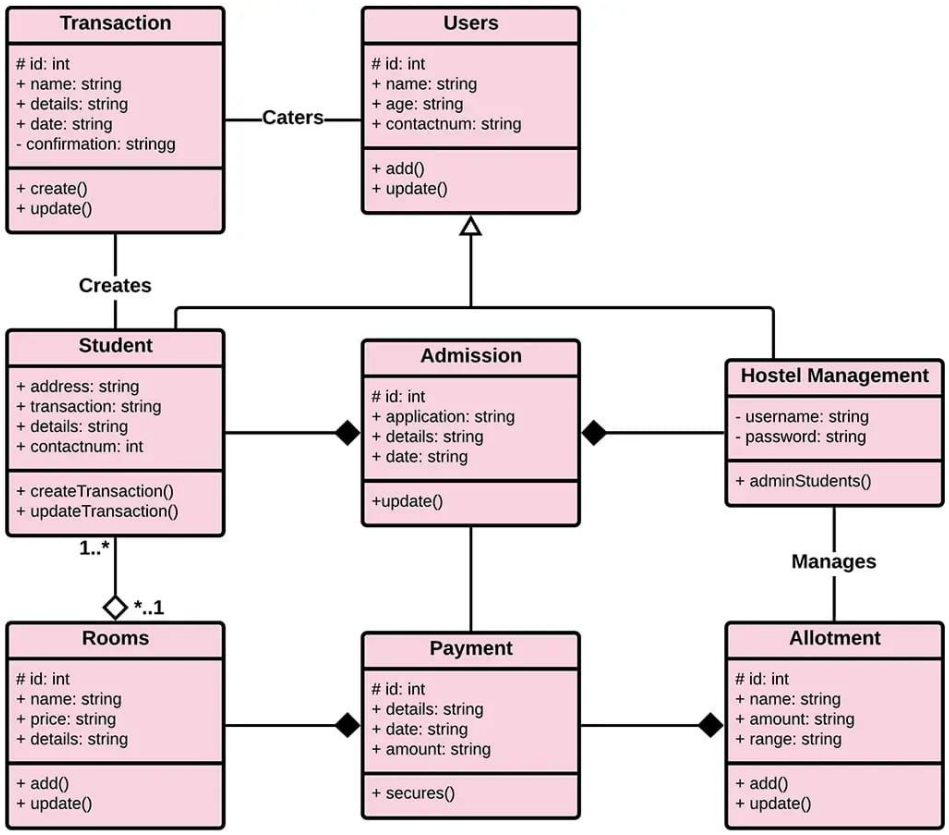

Figure 3.1 Class Diagram

1.1 Use Case Description

The hostel management system class diagram represents the interaction between different components involved in managing hostel activities. At the core, the Users class serves as the base class, from which the Student and Hostel Management classes derive. Students undergo the admission process, represented by the Admission class, which stores details such as application information and admission dates. Once admitted, students are allotted rooms through the Allotment class, which works alongside the *Rooms* class that maintains room details like pricing and availability. Financial activities are handled through the Transaction and Payment classes, where students' payments create secure transactions linked to their accounts. The Hostel Management class oversees the entire process, managing student records, admissions, allotments, and updates. The relationships between these classes ensure smooth coordination between admission, room allotment, and financial management, forming a complete and efficient hostel administration workflow.

2. USE CASE DIAGRAM

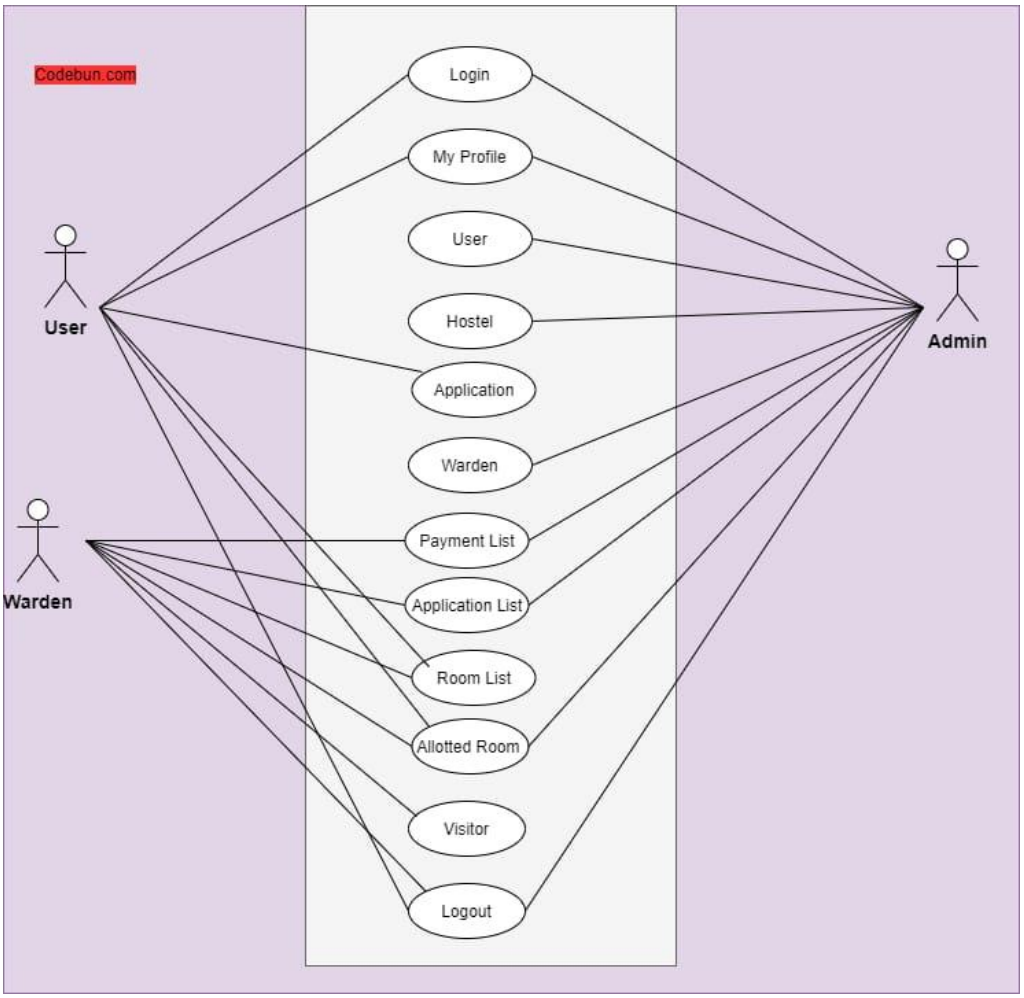

Figure 3.2 Use Case Diagram

2.1 Use Case Description

The use case diagram represents the overall functionality of the Hostel Management System and the interactions between its three main actors: User, Warden, and Admin. The User can perform basic operations such as logging in, viewing or updating their profile, applying for hostel accommodation, and checking allotted room details or visitor information. The Warden has additional responsibilities and can access the payment list, application list, room list, allotted rooms, and manage visitor entries, along with basic login and profile functions. The Admin has full system control and can manage users, hostels, wardens, applications, rooms, payments, and all allotment processes. The central system contains all use cases like Login, Application, Hostel, Room List, Payment List, Visitor, and Logout, which different actors access based on their roles. This diagram clearly shows how each actor interacts with the system and highlights the administrative hierarchy, where the Admin has the highest privileges, followed by the Warden, and then the regular User.

3. SEQUENCE DIAGRAM

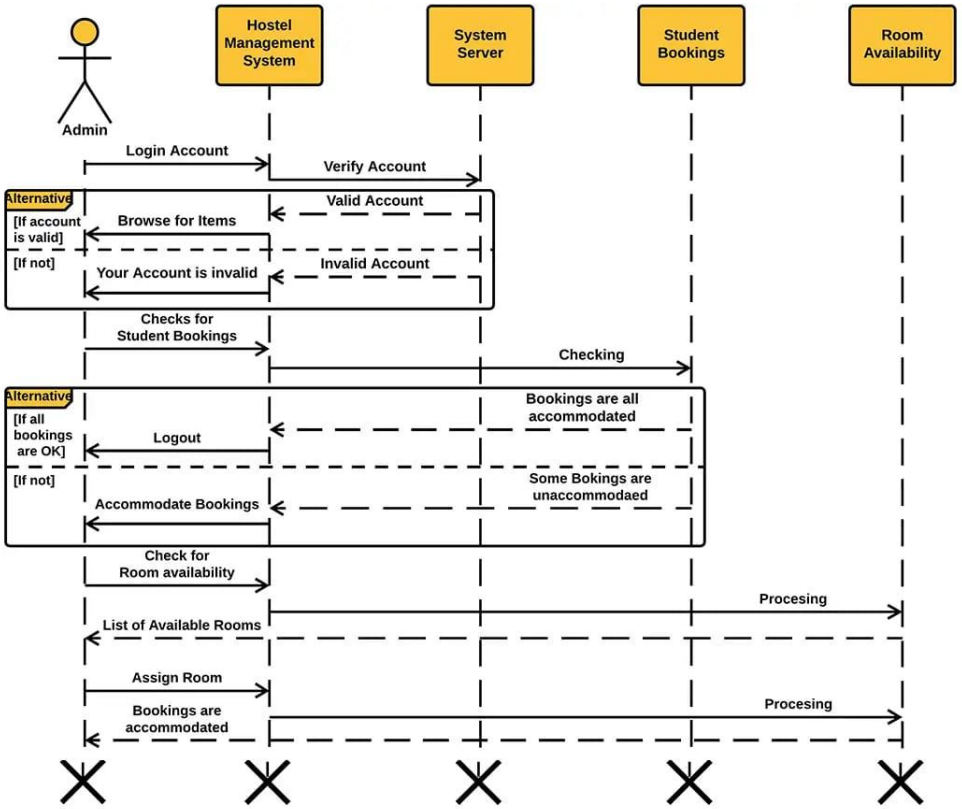

Figure 3.3 Sequence Diagram

3.1 Sequence Diagram Description

The sequence diagram illustrates the workflow of the hostel management system when an admin interacts with the system to manage student bookings and room availability. The process begins with the admin logging into the system, after which the Hostel Management System sends the login details to the System Server for verification. If the account is valid, the system allows the admin to browse items; otherwise, it displays an invalid account message. Once the admin is logged in, the system checks the Student Bookings to identify whether all students are properly accommodatedNext, the system checks room availability by communicating with the Room Availability module, retrieves the list of available rooms, and assigns a suitable room to each pending booking. After assigning rooms, the system updates the status to confirm that all bookings have been accommodated. This sequence diagram clearly shows the interaction flow between the admin and the various system components involved in verifying accounts, managing bookings, and assigning rooms efficiently.

4. ACTIVITY DIAGRAM

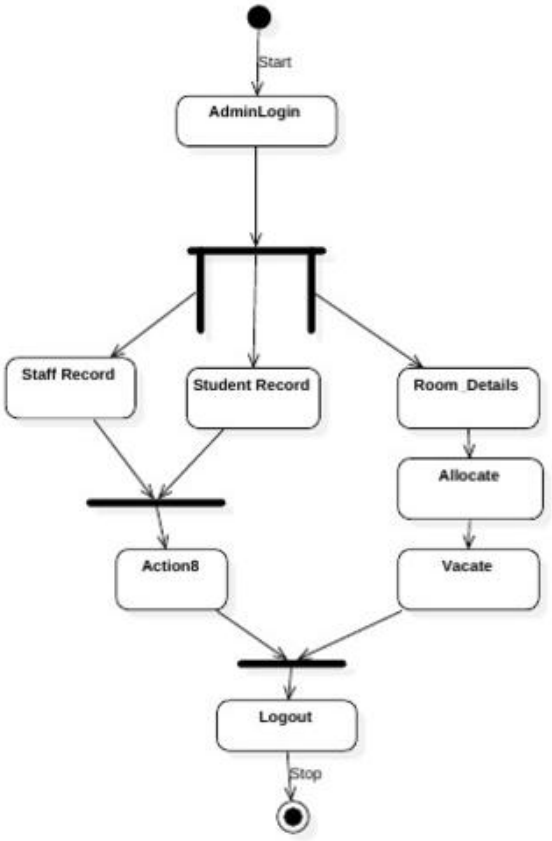

Figure 3.4 Activity Diagram

4.1 Activity Diagram Description

The activity diagram illustrates the step-by-step flow of activities performed by the Admin in the Hostel Management System. It shows how the admin navigates through different tasks such as managing staff records, student details, room allocation, and vacating rooms before finally logging out..The first activity is AdminLogin, where the admin enters credentials to gain access to the system. This is the entry point to all administrative functionalities. If the admin selects Staff Record, they can view, update, or manage hostel staff details.After completing this, the workflow joins with other activities through a join node, meaning the next step depends on completing the task.

5. COMPONENT DIAGRAM

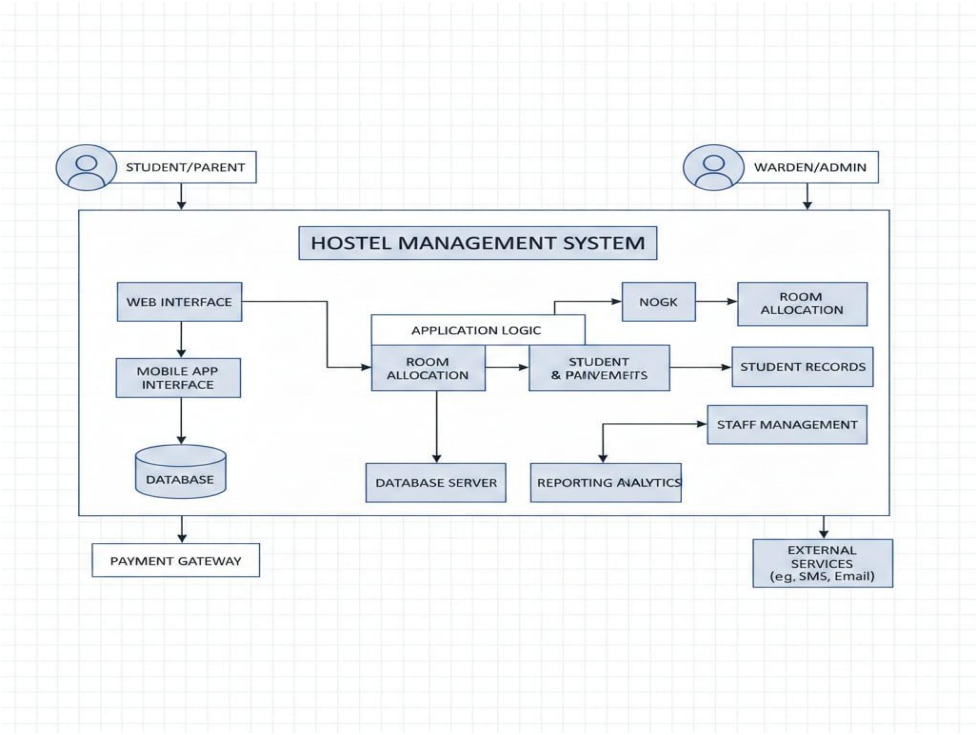

Figure 3.5 Component Diagram

5.1 Component Diagram Description

The component diagram illustrates the overall architecture of the Hostel Management System (HMS) by showing how different functional modules, users, and external services interact with each other. It highlights the major components such as interfaces, application logic, database layers, and integrated external systems that work together to deliver a complete hostel management solution. The system has two primary users: Students/Parents and Warden/Admin. Students access the system mainly for room booking, fee payments, and information updates, while the admin manages student records, room allocation, and staff-related activities. Overall, this component diagram shows a structured hostel management system where user interfaces, application logic, database layers, and external integrations work together. Each component has a specific responsibility, creating an efficient, automated, and user-friendly environment for both students and administrators.

6. DEPLOYMENT DIAGRAM

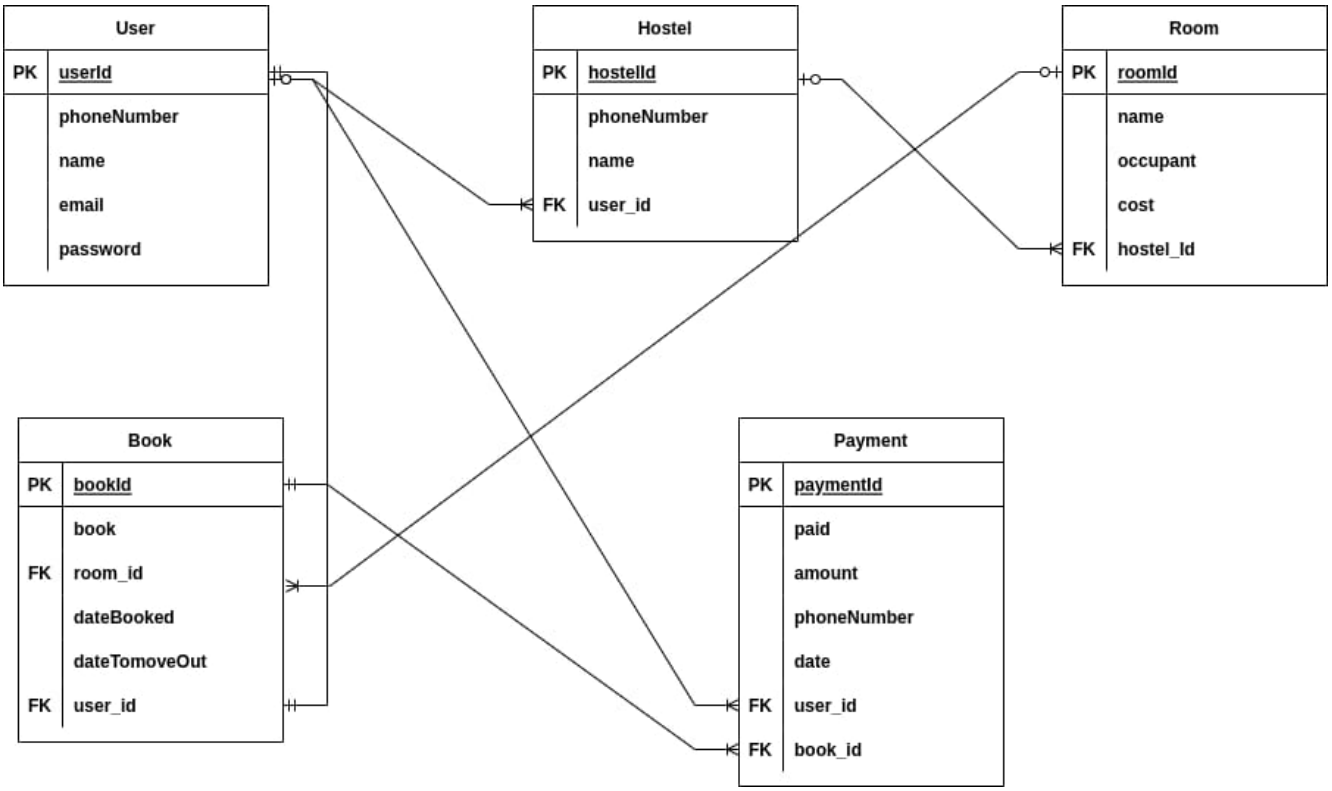

Figure 3.6 Deployment Diagram

6.1 Deployment Diagram Description

The deployment diagram illustrates the physical architecture of the Hostel Management System by showing how software components are deployed across different hardware nodes. The system is primarily divided into three major nodes: the Client Device, the Application Server, and the Database Server. Each node hosts specific components required for the system to function efficiently.The Client Device node represents the devices used by students, parents, wardens, and administrators to access the system. This includes desktops, mobile phones, and tablets through which users interact using the web or mobile interface. These devices communicate with the backend server using standard internet protocols. The client node mainly runs the browser or the mobile application that sends requests to the server.

7. PACKAGE DIAGRAM

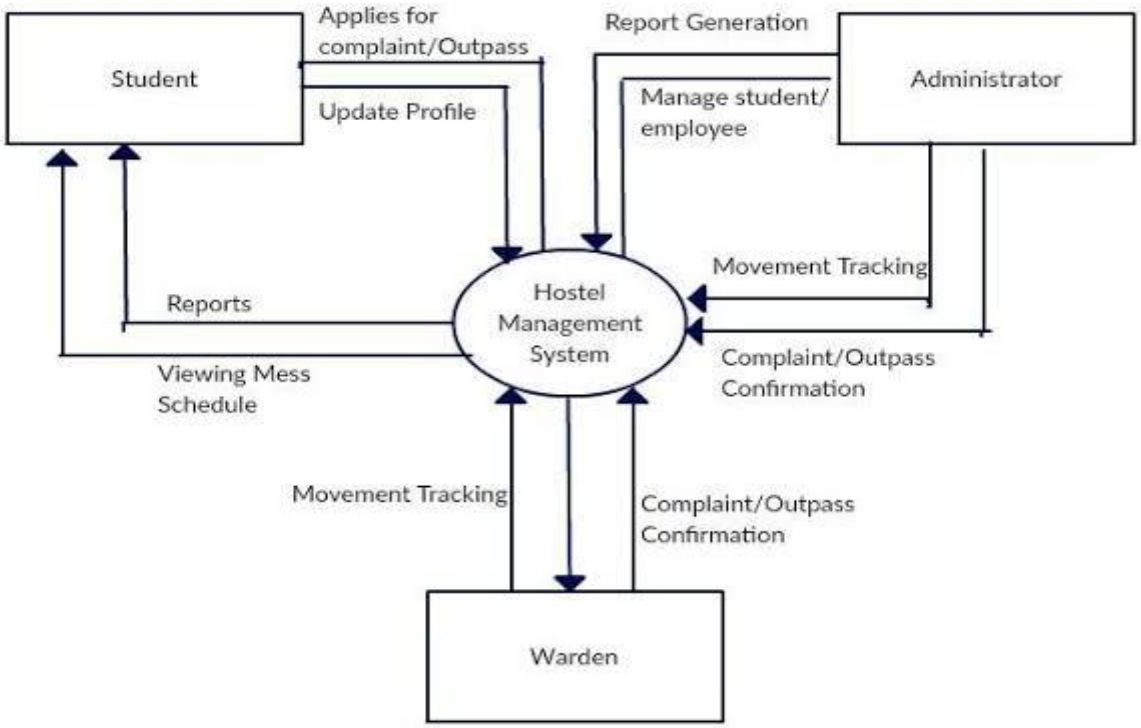

Figure 3.7 Package Diagram

7.1 Package Diagram Description

A package diagram for this Hostel Management System would organize the listed features into logical, modular groups based on user roles and shared functionality. The system can be structured into three main packages: Student, Warden, and Common. The Student package would contain modules for profile updates, complaint or outpass applications, mess schedule viewing, movement tracking, and report generation—all tailored for student access. The Warden package would include features for managing students and employees, confirming complaints or outpasses, overseeing hostel operations, generating administrative reports, and monitoring student movements. This modular approach ensures separation of concerns, reduces duplication, and maintains clear dependencies, with the Student and Warden packages both relying on the Common package for core functionalities while operating independently within their respective domains.

8. COLLABORATION DIAGRAM

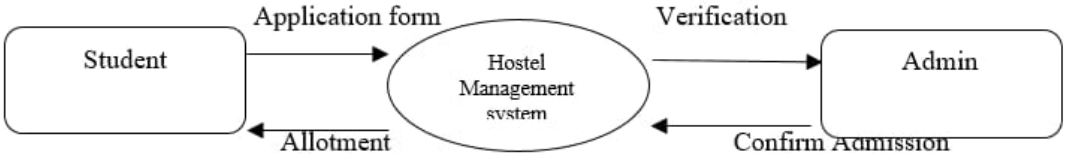

Fig 1 DFD For Allotment Process

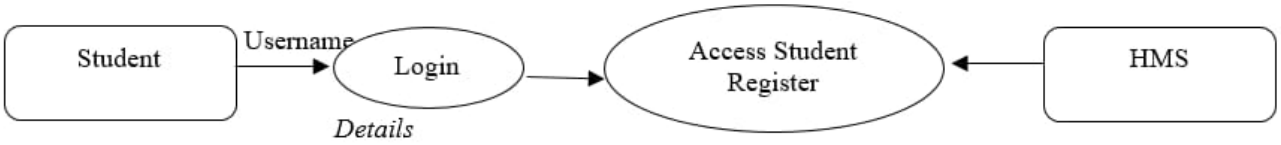

Fig 2 DFD For Student Login

Fig 3 DFD For Student Registration

Figure 3.8 Collaboration Diagram

8.1 Collaboration Diagram Description

Hostel Management System (HMS), and Admin during different hostel-related processes. In the allotment process, the student submits an application form to the Hostel Management System. The system forwards the details to the admin for verification, and after confirmation of admission by the admin, the hostel allotment information is sent back to the student. In the student login process, the student provides a username and login details, which are validated by the system. Upon successful authentication, the HMS allows access to the student register. In the student registration process, the student enters personal details through the application filling module, which is reviewed and approved by the admin. Overall, this collaboration diagram shows how different components communicate and coordinate with each other to ensure smooth registration, login, verification, and hostel allotment within the Hostel Management System.