Sardar Patel Institute of Technology

Bhavan's Campus, Munshi Nagar, Andheri(W), Mumbai:400058, India

(Autonomous College of Affiliated to University of Mumbai)

Team Member Name:	Team Member UID:
Manish Jadhav	2023301005
Mayur Solankar	2023301018

EXPERIMET NO:	3
AIM:	Class Diagram for Hostel Management System

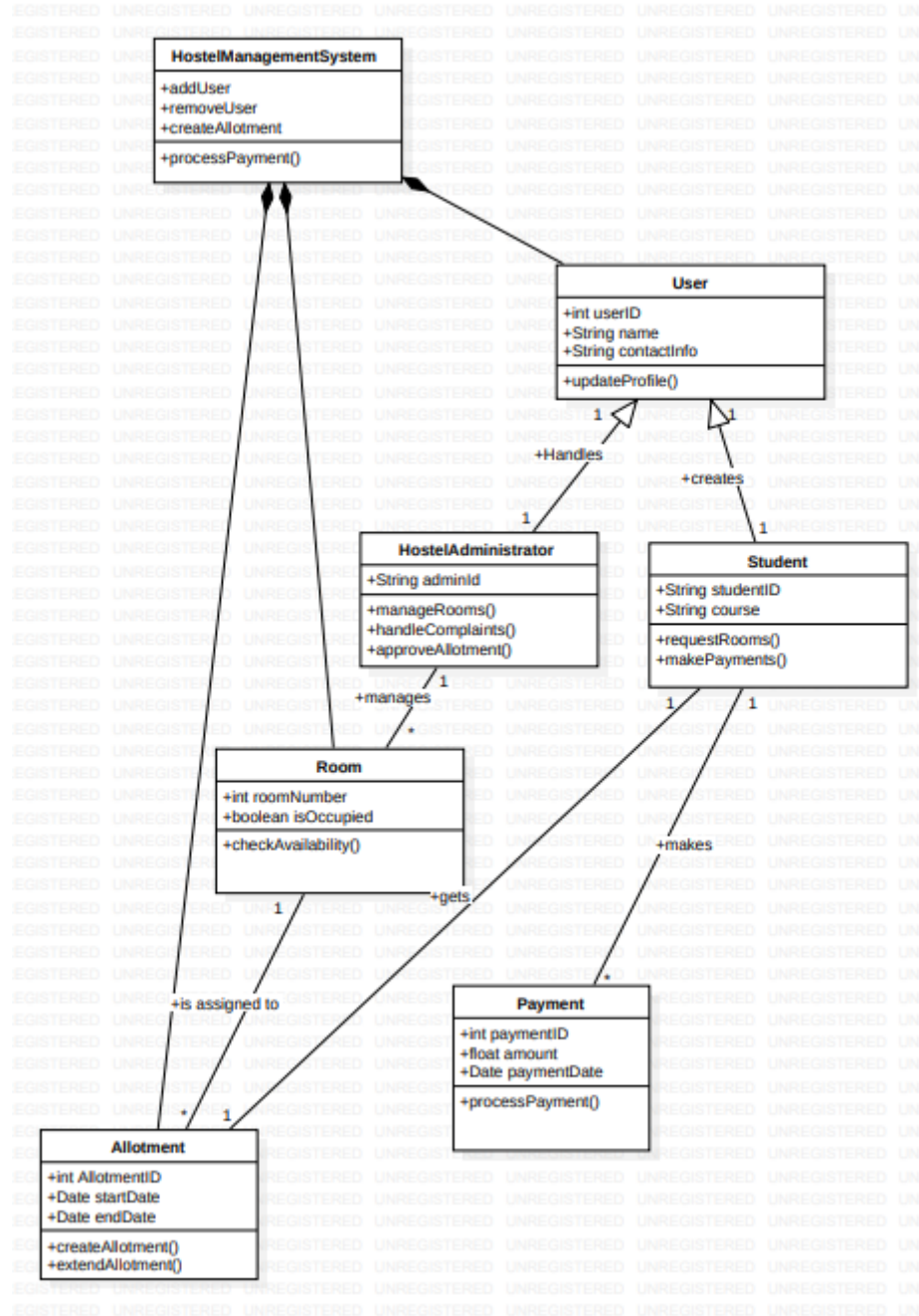|	
Problem Statement:	The Hostel Management System (HMS) project addresses the challenges faced in efficiently managing hostels in today's world. With a focus on enhancing student satisfaction, the HMS aims to provide a comprehensive solution. It offers user management for administrators, students and visitors, simplifies room booking, facilitates smooth check-in/check-out processes, manages billing and payments, monitors room availability, maintains student profiles, and provides reporting and analytics tools. The system ensures data security and privacy compliance while offering a user- friendly interface accessible via a web app. By automating administrative tasks and optimizing room management, the HMS benefits hostel owners, while also improving the student experience and modernizing hostel operations.
Noun/Noun Phrases:	Hostel Management System (HMS), challenges, hostels, focus, student satisfaction, solution, user management, administrators, students, visitors, room booking, check-in, check-out processes, billing, payments, room availability, student profiles, reporting, analytics tools, system, data security, privacy compliance, interface, web app, tasks, room management, hostel owners, experience, operations.
Classes:	HostelManagementSystem , User, HostelAdministrator, Student, Room, Allotment, Payment
Verb Phrases:	- Hostel administrator approves the student allotment. - User can check the room occupancy. - Students make payment for the hostel. - Allotments have start date and end date. - Hostel administrator handles complaints of the user. - Students creates student ID. - Payment have a payment ID. - User can update their profile information. - Students requests for room.

	• Hostel administrator manages room.
Relations:	1. **Payment:** This class records financial transactions for payments, including attributes like paymentID, amount, and paymentDate.
	2. **User:** Represents all system users, serving as a base class for both Student and HostelAdministrator. It includes common attributes like userID, name, and contactInfo.
	3. **Student:** Inherits from User and contains additional student-specific information, including studentID and course. It can request rooms and make payments.
	4. **Allotment:** Manages student accommodations, connecting students to rooms. It includes attributes like AllotmentID, startDate, and endDate.
	5. **HostelAdministrator:** Inherits from User and interacts with the system for room management, complaint handling, and allotment approval.
	6. **Room:** Represents individual hostel rooms with attributes like roomNumber and isOccupied status. It can check for availability.
	7. **HostelManagementSystem:** This is the main class that oversees the entire system. It can add and remove users, create allotments, and process payments.

Diagram:

Java Code Snippets:

HostelManagementSystem Class	
| **HostelManagementSystem Class** | ```java
import java.util.List;
import java.util.ArrayList;

public class HostelManagementSystem {
    private List<User> users;
    private List<Allotment> allotments;
    private List<Payment> payments;

    public HostelManagementSystem() {
        users = new ArrayList<>();
        allotments = new ArrayList<>();
        payments = new ArrayList<>();
    }

    public void addUser(User user) {
        users.add(user);
    }

    public void removeUser(User user) {
        users.remove(user);
    }

    public void createAllotment(Allotment allotment) {
        allotments.add(allotment);
    }

    public void processPayment(Payment payment) {
        payments.add(payment);
        payment.processPayment();
    }
}
``` |
| **User Class** | ```java
public class User {
    protected int userID;
    protected String name;
    protected String contactInfo;

    public User(int userID, String name, String contactInfo) {
        this.userID = userID;
        this.name = name;
        this.contactInfo = contactInfo;
    }

    public void updateProfile() {
        // Implementation for updating user profile
    }
}
``` |

| **HostelAdministrator Class** | ```java
public class HostelAdministrator extends User {
    private String adminId;

    public HostelAdministrator(int userID, String name, String contactInfo, String adminId) {
        super(userID, name, contactInfo);
        this.adminId = adminId;
    }

    public void manageRooms() {
        // Implementation for managing rooms
    }

    public void handleComplaints() {
        // Implementation for handling complaints
    }

    public void approveAllotment() {
        // Implementation for approving allotments
    }
}
``` |
| **Room Class** | ```java
public class Room {
    private int roomNumber;
    private boolean isOccupied;

    public Room(int roomNumber) {
        this.roomNumber = roomNumber;
        this.isOccupied = false;
    }

    public boolean checkAvailability() {
        return !isOccupied;
    }
}
``` |
| **Student Class** | ```java
public class Student extends User {
    private String studentID;
    private String course;

    public Student(int userID, String name, String contactInfo, String studentID, String course) {
        super(userID, name, contactInfo);
        this.studentID = studentID;
        this.course = course;
    }
    public void requestRooms() {
        // Implementation for requesting rooms
    }

    public void makePayments() {
        // Implementation for making payments
    }
}
``` |

Allotment Class	
| **Allotment Class** | ```java
import java.util.Date;

public class Allotment {
    private int allotmentID;
    private Date startDate;
    private Date endDate;
    private Student student;
    private Room room;

    public Allotment(int allotmentID, Date startDate, Date endDate,
Student student, Room room) {
        this.allotmentID = allotmentID;
        this.startDate = startDate;
        this.endDate = endDate;
        this.student = student;
        this.room = room;
    }

    public void createAllotment() {
        // Implementation for creating an allotment
    }

    public void extendAllotment(Date newEndDate) {
        // Implementation for extending an allotment
    }
}
``` |
| **Payment Class** | ```java
import java.util.Date;

public class Payment {
    private int paymentID;
    private float amount;
    private Date paymentDate;

    public Payment(int paymentID, float amount, Date paymentDate)
{
        this.paymentID = paymentID;
        this.amount = amount;
        this.paymentDate = paymentDate;
    }

    public void processPayment() {
        // Implementation for processing payment
    }
}
``` |

Sardar Patel Institute of Technology

Bhavan's Campus, Munshi Nagar, Andheri(W), Mumbai:400058, India
(Autonomous College of Affiliated to University of Mumbai)

Conclusion:	In summary, the Hostel Management System class diagram, incorporating classes such as," "Users," "Student," "Admission," "Hostel Administrator," "Rooms," "Payment," and "Allotment," serves as a visual blueprint for designing a robust and efficient hostel management system. It defines the core entities and their relationships, enabling effective management of student admissions, room allocations, payments, and user interactions within the system. This class diagram lays the foundation for building a comprehensive and user-friendly Hostel Management System to streamline hostel operations and enhance the overall hostel experience.