Practical:- 3 to 10

“Hostel Grievance System”

Team members -

Shailesh - 009

Paras - 005

Vishwajit - 046

Practical:-3

Aim:- Prepare the Software Project Management Plan (SPMP) including following:

 Estimation of Size, Cost, Duration, Effort

 Prepare the Schedule, Milestones

 Identification of Risk, Risk Management

Solution:-

1)Estimation of Size, Cost, Duration, Effort:-

1.1. Size Estimation

- **Lines of Code (LOC)**: Based on the project's scope, we estimate approximately 15,000 - 20,000 lines of code for the system, including frontend (web), backend (APIs), database scripts, and testing scripts.
- **Number of Features**:
  - Complaint Submission
  - Complaint Tracking
  - Complaint Management (by hostel management)
  - User Authentication (student, management, staff)
  - Report Generation

1.2. Cost Estimation

- **Development Cost**:
  - **Frontend Development** (React): Rs.10,000

  - **Backend Development** (Node.js): Rs.12,000
  - **Database Setup** (MongoDB or MySQL): Rs.5,000
  - **Testing**: Rs.3,000
  - **Deployment and Hosting**: Rs.2,000
  - **Total Estimated Cost**: Rs.30,000

1.3. Duration and Effort Estimation

- **Duration**: Estimated time for the project is around 5-6 months.
- **Effort**: We use the COCOMO (Constructive Cost Model) approach for calculating the effort:
  - **Basic Formula**:

$$\text{Effort (Person-Months)}=a\times(\text{Size (KLOC)})b$$

For a system of this complexity (Semi-Detached):

  - **Effort Constant** a=2.94
  - **Exponent** b=1.05

Estimated KLOC: 20 Effort=2.94×(20)1.05≈63.7 person-months

Therefore, approximately **64 person-months** of effort are required, which can be distributed among a team of 4 developers working for 5 months.

2) Schedule and Milestones:-

2.1. Schedule

Phase	Start Date	End Date	Duration
Project Initiation	Week 1	Week 2	2 weeks
Requirement Analysis	Week 3	Week 5	3 weeks
System Design	Week 6	Week 8	3 weeks
Development	Week 9	Week 18	10 weeks
Testing	Week 19	Week 20	1 week
Deployment	Week 21	Week 23	2 weeks
Post-Deployment Support	Week 24	Ongoing	Ongoing

2.2. Milestones

- **Project Plan Approved** (End of Week 2)
- **Requirements Freeze** (End of Week 5)
- **System Design Complete** (End of Week 8)
- **Backend API Development Complete** (End of Week 12)
- **Frontend Interface Complete** (End of Week 14)
- **Integration of Modules Complete** (End of Week 18)
- **Testing Complete** (End of Week 20)
- **System Live** (End of Week 23)

3. Risk Identification and Management:-

3.1. Risk Identification

Risk	Likelihood	Impact	Description
Requirement Changes	Medium	High	Changes in user needs or government policies could lead to rework.
Technical Risks	Low	High	Issues with integrating third-party tools for complaint tracking or delays in backend API development.
Underestimation of Effort	Medium	High	Potential underestimation of development effort, especially for backend and database integration.
Data Privacy and Security Risks	High	High	Mishandling of sensitive user data (complaints, personal information).
Availability of Resources	Low	Medium	Resource availability (developers, testers) could affect the project schedule.

Risk	Likelihood	Impact	Description
Inadequate Testing	Medium	Medium	Lack of comprehensive testing could result in undetected bugs in the live system.
User Adoption	Medium	Low	Difficulty getting hostel students and staff to adopt the new system.
Server Downtime/Technical Failures	Medium	High	Issues with the server hosting the grievance system could cause interruptions.

3.2. Risk Management Plan

Risk	Mitigation Strategy
Requirement Changes	Implement a formal change request process, ensuring clear communication with stakeholders.
Technical Risks	Ensure proper research and prototyping before selecting third-party tools or libraries for the system.

Risk	**Mitigation Strategy**
Underestimation of Effort	Use agile methodology and iterative development cycles to track progress and adjust as needed.
Data Privacy and Security Risks	Implement robust encryption, secure authentication mechanisms (e.g., JWT, OAuth), and data anonymization where needed.
Availability of Resources	Maintain buffer periods in the schedule and consider hiring additional contractors if necessary.
Inadequate Testing	Establish a dedicated testing phase and invest in both automated and manual testing processes.
User Adoption	Conduct training sessions and tutorials for students and hostel staff on using the new system.
Server Downtime/Technical Failures	Set up automated backups and redundancy in the server infrastructure to minimize downtime.

Practical:-4

Aim:- Prepare the following components of Data Flow Model:

□ Data Dictionary

□ Data Flow Diagram

□ Structure Chart

Solution:-

1) Data Dictionary:-

Entities:

- **Student**: Represents the hostel student who can file grievances.
  - Attributes:
    - StudentID: Unique identifier (Integer)
    - Name: Full name (String)
    - Email: Contact email (String)
    - PhoneNumber: Contact number (String)
    - RoomNumber: Room number in the hostel (String)

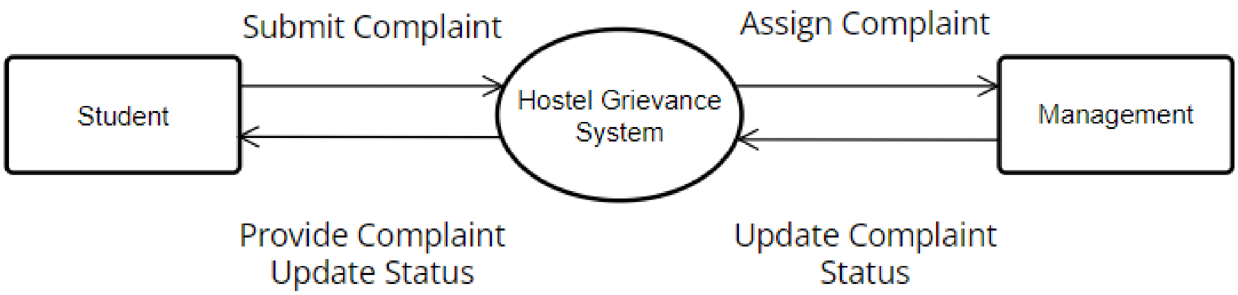
- **Complaint**: Represents a grievance raised by a student.
  - Attributes:
    - ComplaintID: Unique identifier for the complaint (Integer)
    - ComplaintType: Type of issue (e.g., Furniture, Sanitation) (String)
    - Description: Detailed description of the issue (String)

        - DateRaised: Date the complaint was filed (Date)
        - Status: Current status (e.g., Pending, In Progress, Resolved) (String)
        - AssignedStaff: Staff member assigned to resolve the issue (String)

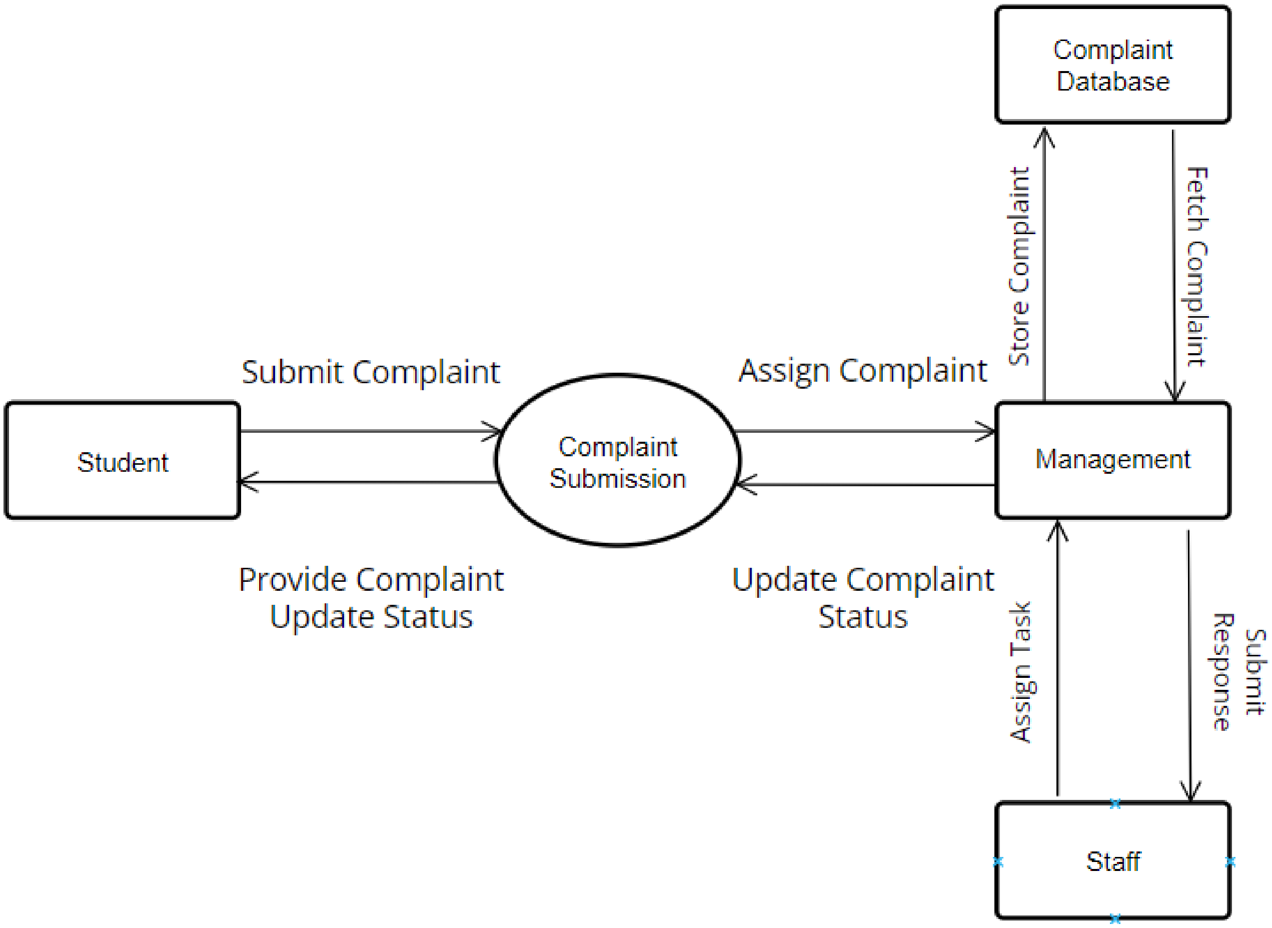
- **Management**: Represents hostel management responsible for resolving complaints.
    - Attributes:
        - StaffID: Unique identifier for the staff (Integer)
        - Name: Full name (String)
        - Role: Staff role (e.g., Maintenance, Cleaner) (String)

2) Data Flow Diagram:-

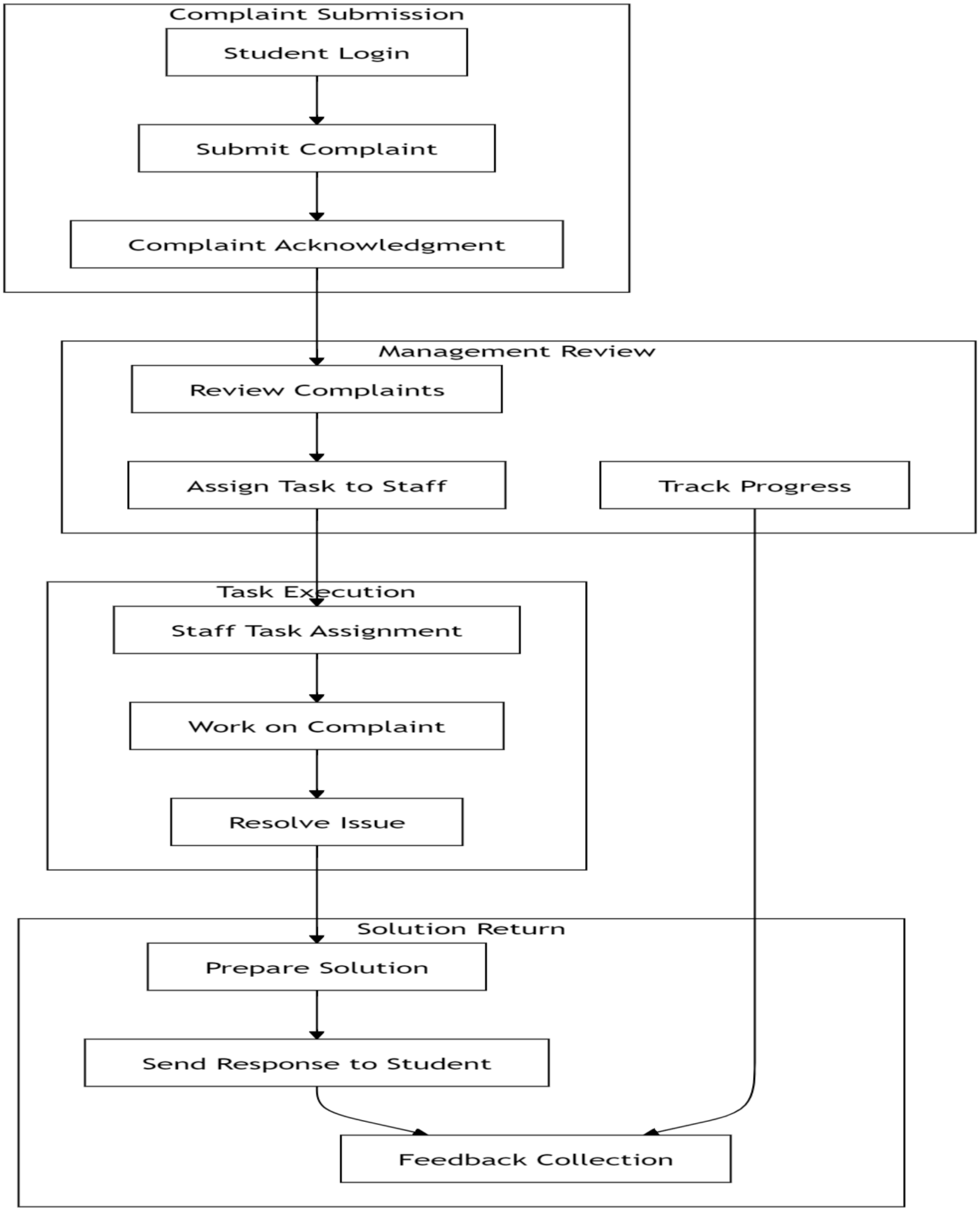
Level 0 :-



Level 1 :-



3) Structured Chart :-



Practical:- 5

Aim :- Prepare the user’s view analysis : Describe different scenarios and Draw Use case diagrams using UML.

Solution:-

Scenarios:-

Scenario 1: Student Submitting a Grievance

- **Preconditions**: The student is logged in to the hostel grievance system.
- **Actions**:
    1. The student selects the option to submit a new complaint.
    2. They choose the type of issue (e.g., Furniture, Sanitation).
    3. The student fills out the details (e.g., description, room number).
    4. The system confirms receipt of the complaint and assigns a complaint ID.
- **Postconditions**: The complaint is submitted, and the student can track its status.

Scenario 2: Hostel Management Reviewing and Assigning Complaints

- **Preconditions**: Hostel management is logged into the system.
- **Actions**:
    1. Management views the list of newly submitted complaints.
    2. They filter or sort complaints based on urgency or category.
    3. They assign complaints to specific staff members for resolution.
    4. The system updates the complaint status to 'In Progress'.

- **Postconditions**: The complaint is assigned, and the staff member receives the details.

Scenario 3: Staff Member Resolving a Complaint

- **Preconditions**: A staff member is logged in and has been assigned a complaint.
- **Actions**:
  1. The staff member reviews the assigned complaint details.
  2. After addressing the issue, they update the complaint's status (e.g., 'Resolved').
  3. The system records the resolution and notifies the student.
- **Postconditions**: The complaint status is updated to 'Resolved,' and the student is notified.

Scenario 4: Student Checking Complaint Status

- **Preconditions**: The student has submitted a complaint.
- **Actions**:
  1. The student logs in to the system.
  2. They navigate to the “My Complaints” section.
  3. The system displays the list of complaints along with their statuses (e.g., Pending, In Progress, Resolved).
- **Postconditions**: The student can see the current status of their submitted complaints.

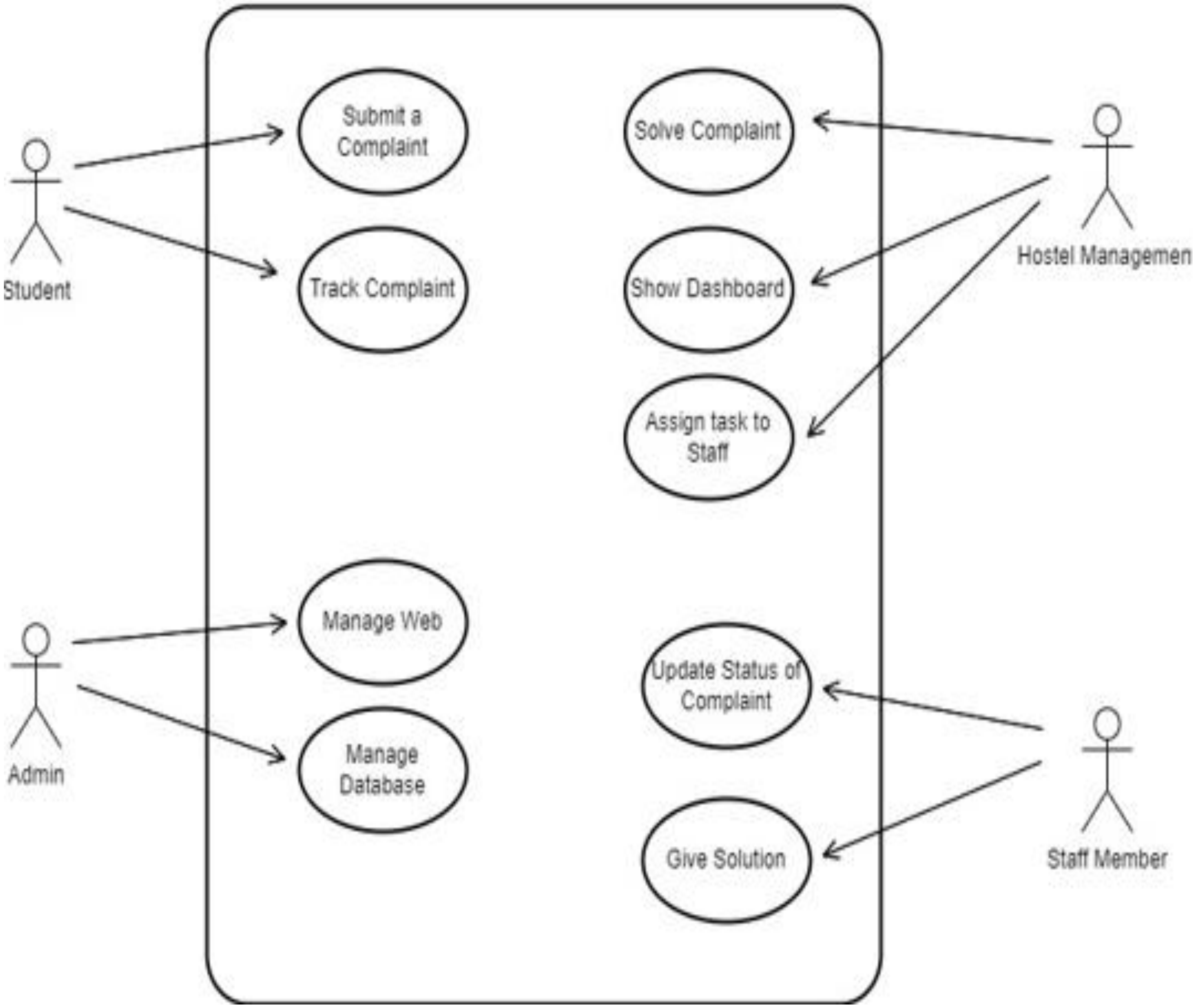
Scenario 5: Management Tracking and Prioritizing Complaints

- **Preconditions**: Management is logged in to the system.
- **Actions**:
  1. Management accesses the “Complaint Dashboard.”
  2. They filter complaints based on priority, category, or student.

3. They assign new complaints or check on ongoing ones.
4. Management can generate reports based on complaints resolved or pending.

- **Postconditions**: Efficient handling and prioritization of complaints by management.

2) Use Case Diagram :-

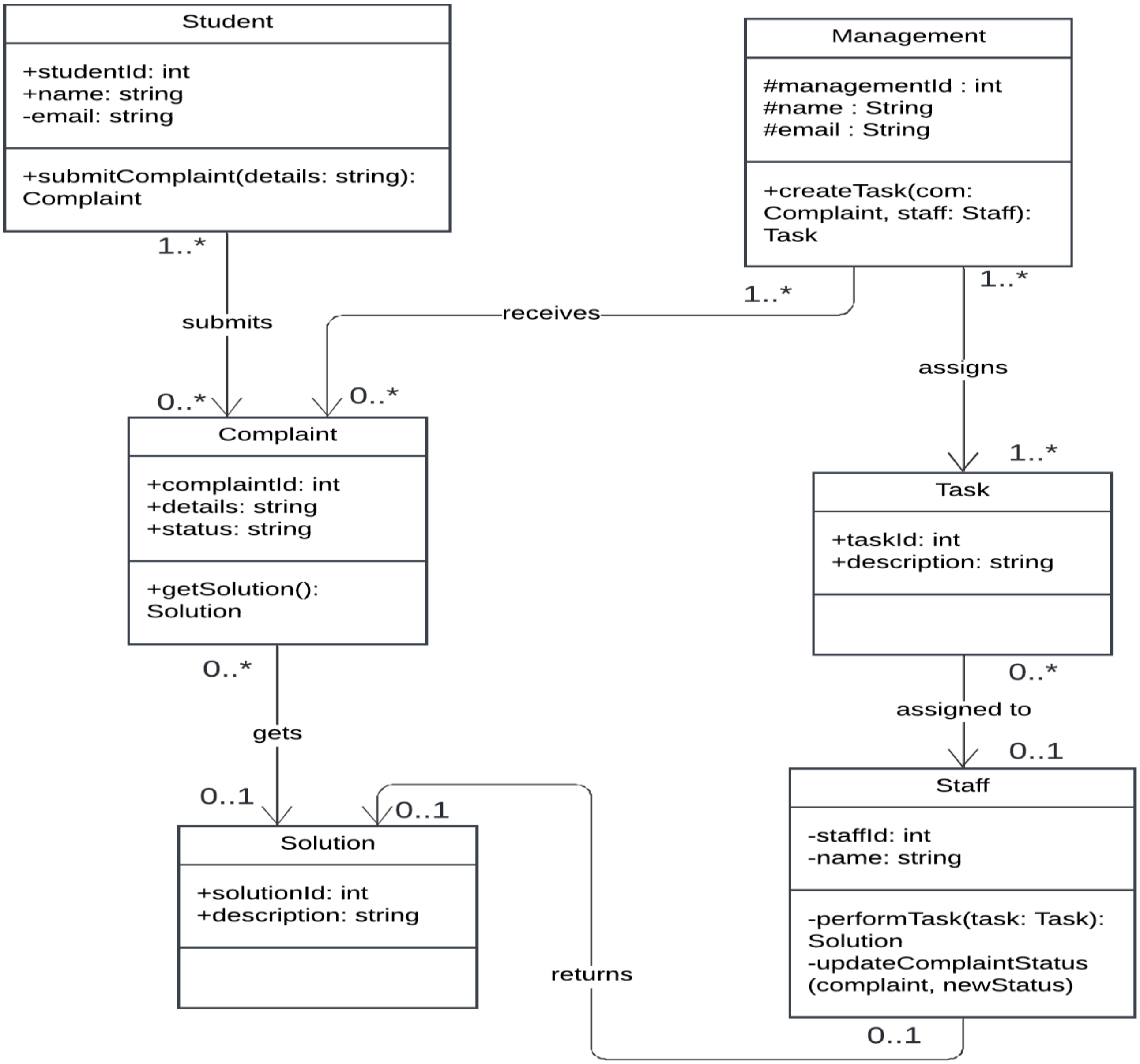


Practical:- 6

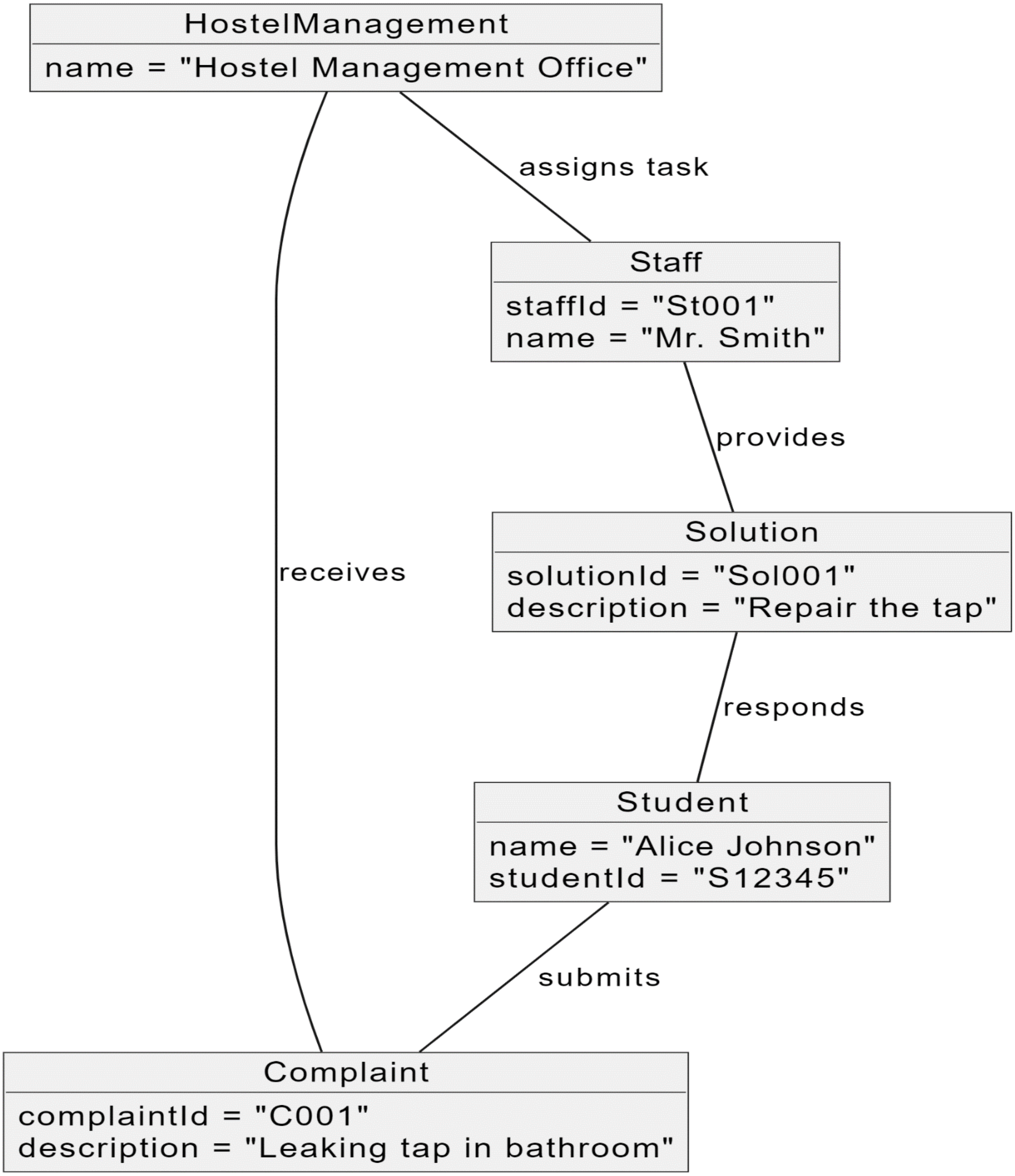
Aim :- Prepare the structural view : Draw Class diagram and object diagram

Solution:-

1) Class Diagram :-



2) Object Diagram :-

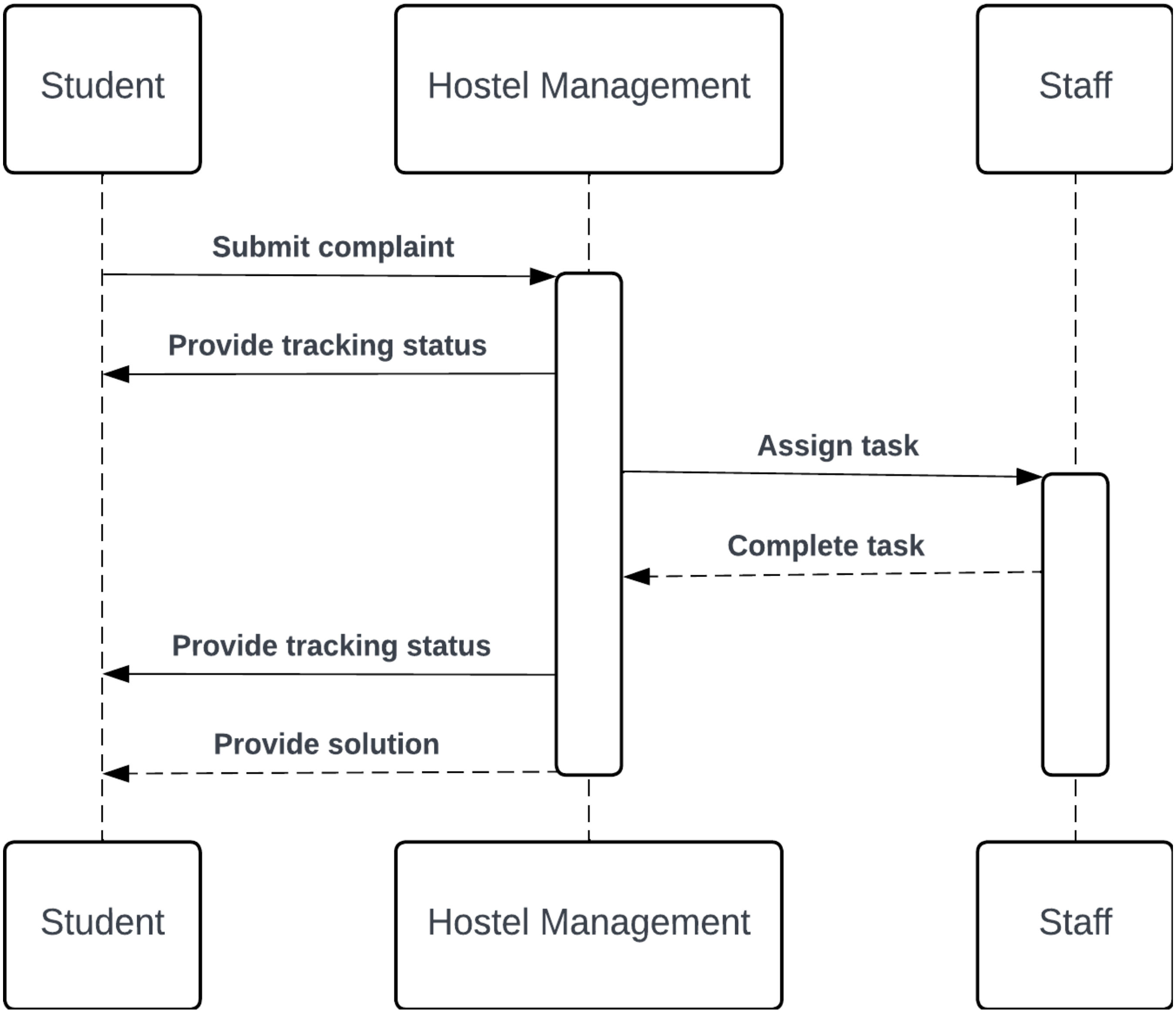


Practical:- 7

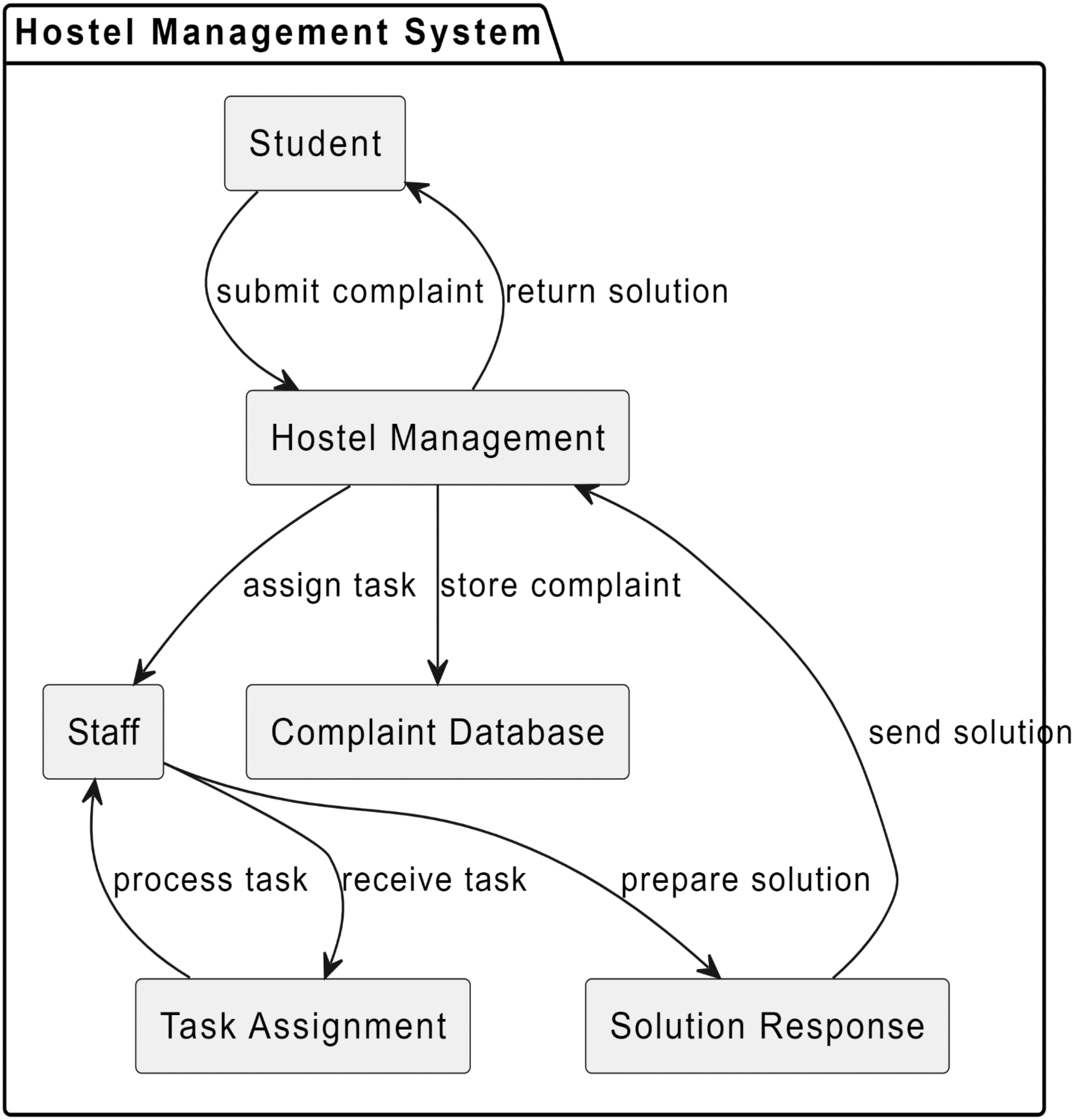
Aim :- Prepare the behavioral view : Draw Sequence diagram and Collaboration diagram

Solution:-

1) Sequence Diagram :-



2) Collaboration Diagram :-

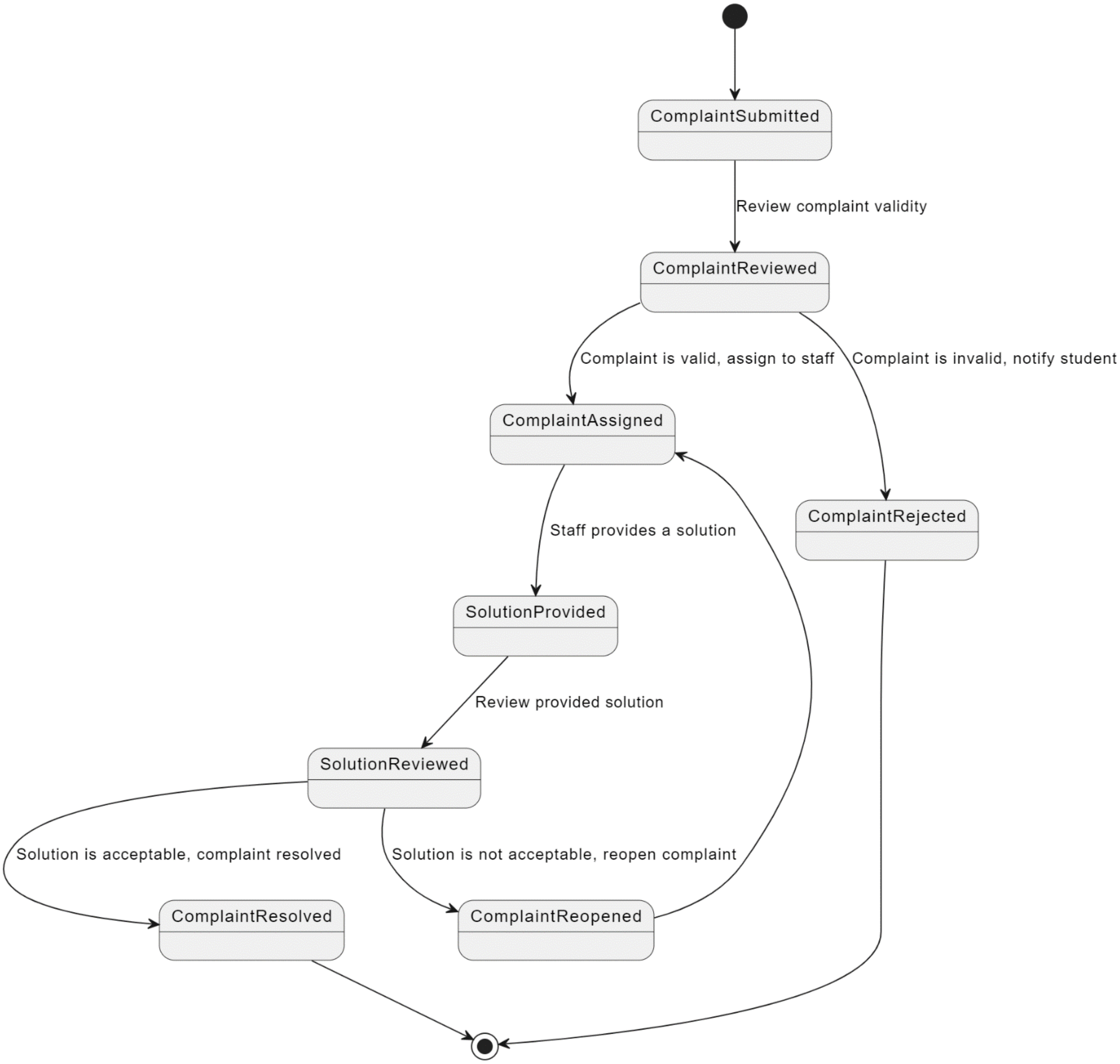


Practical:- 8

Aim :- Prepare the behavioral view : Draw State-chart diagram

Solution:-

1) State-Chart diagram :-

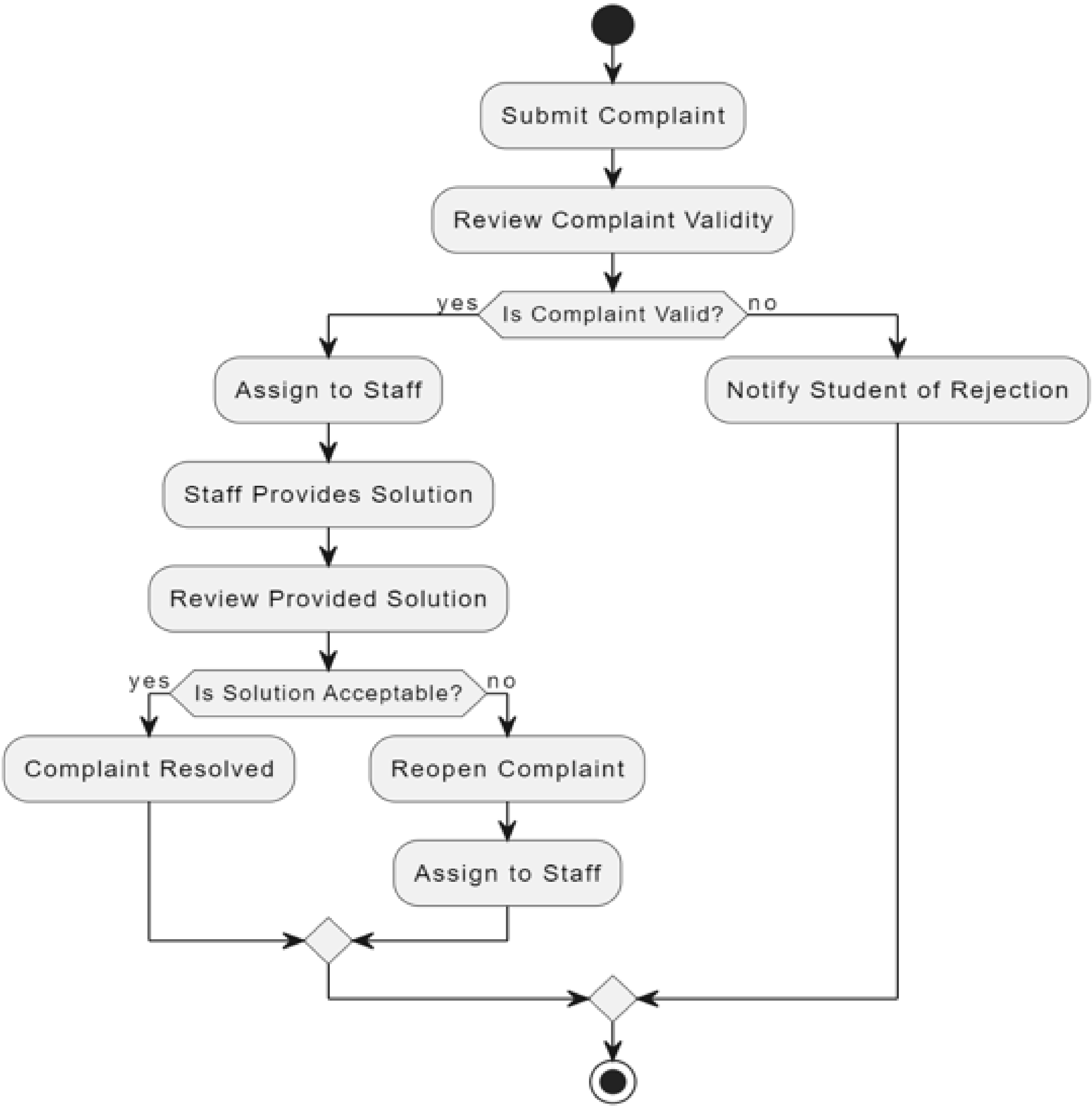

ComplaintSubmitted
Review complaint validity
ComplaintReviewed
Complaint is valid, assign to staff
Complaint is invalid, notify student
ComplaintAssigned
ComplaintRejected
Staff provides a solution
SolutionProvided
Review provided solution
SolutionReviewed
Solution is acceptable, complaint resolved
Solution is not acceptable, reopen complaint
ComplaintResolved
ComplaintReopened

Practical:- 9

Aim :- Prepare the behavioral view : Draw Activity diagram

Solution:-

1) Activity diagram :-

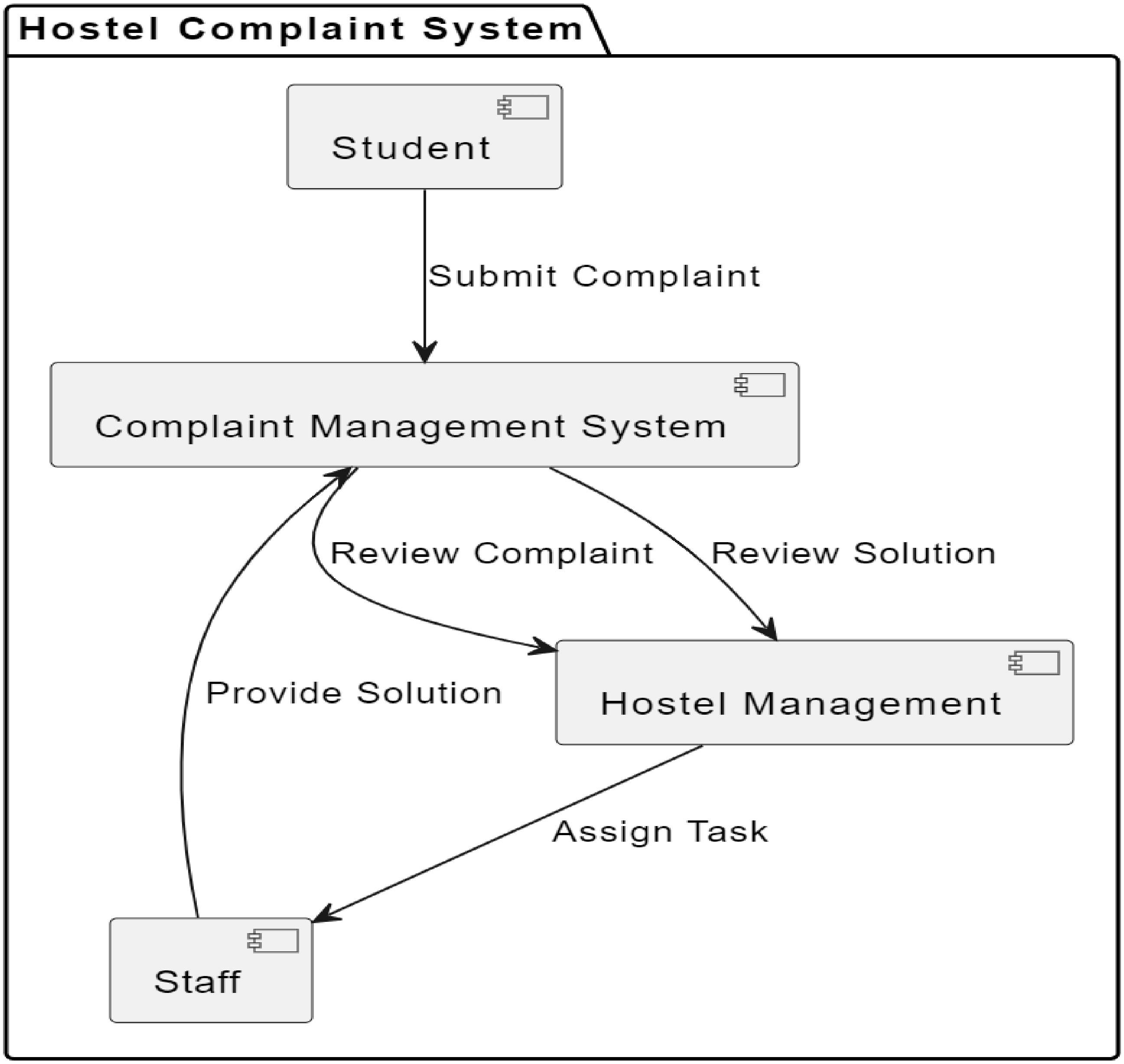


Practical:- 10

Aim :- Prepare the implementation view: Draw Component diagram & Deployment diagram.

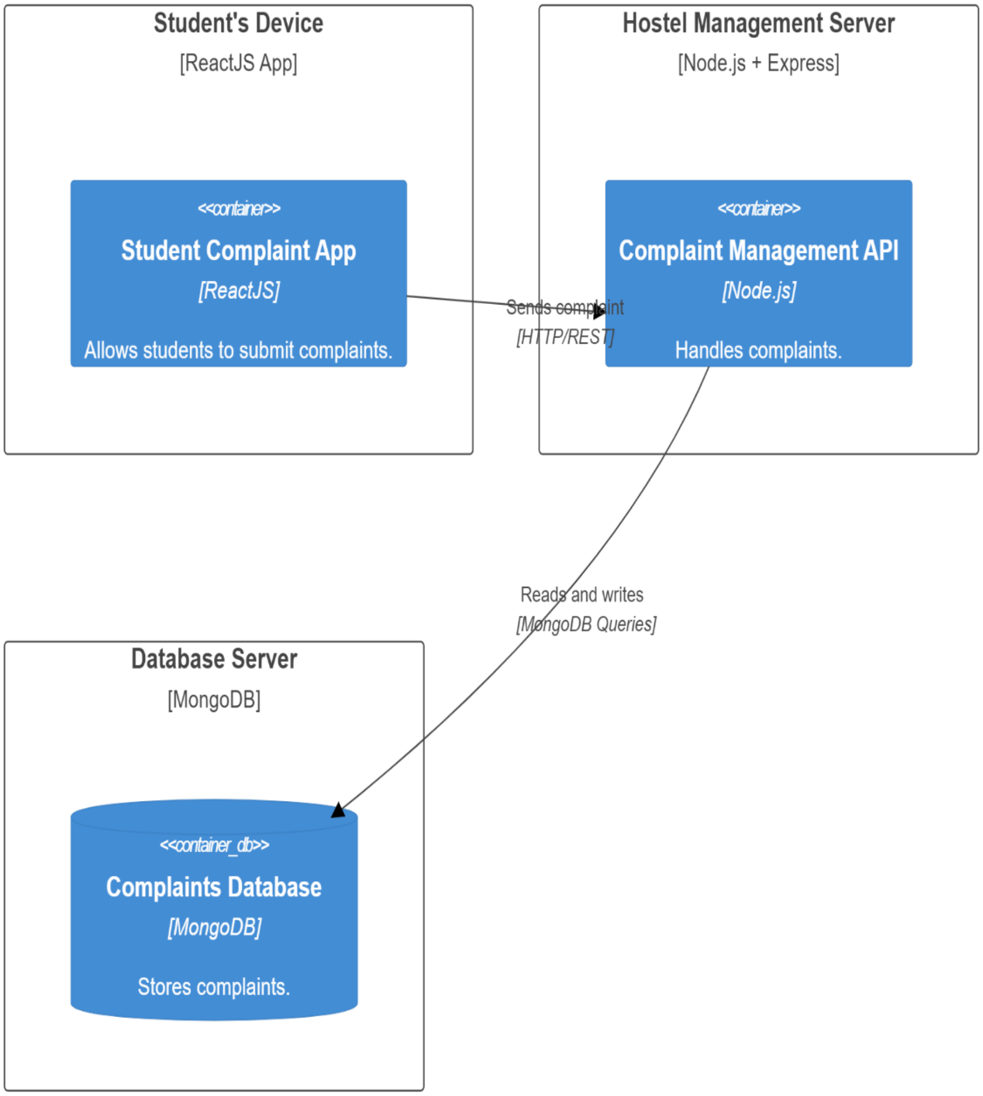
Solution:-

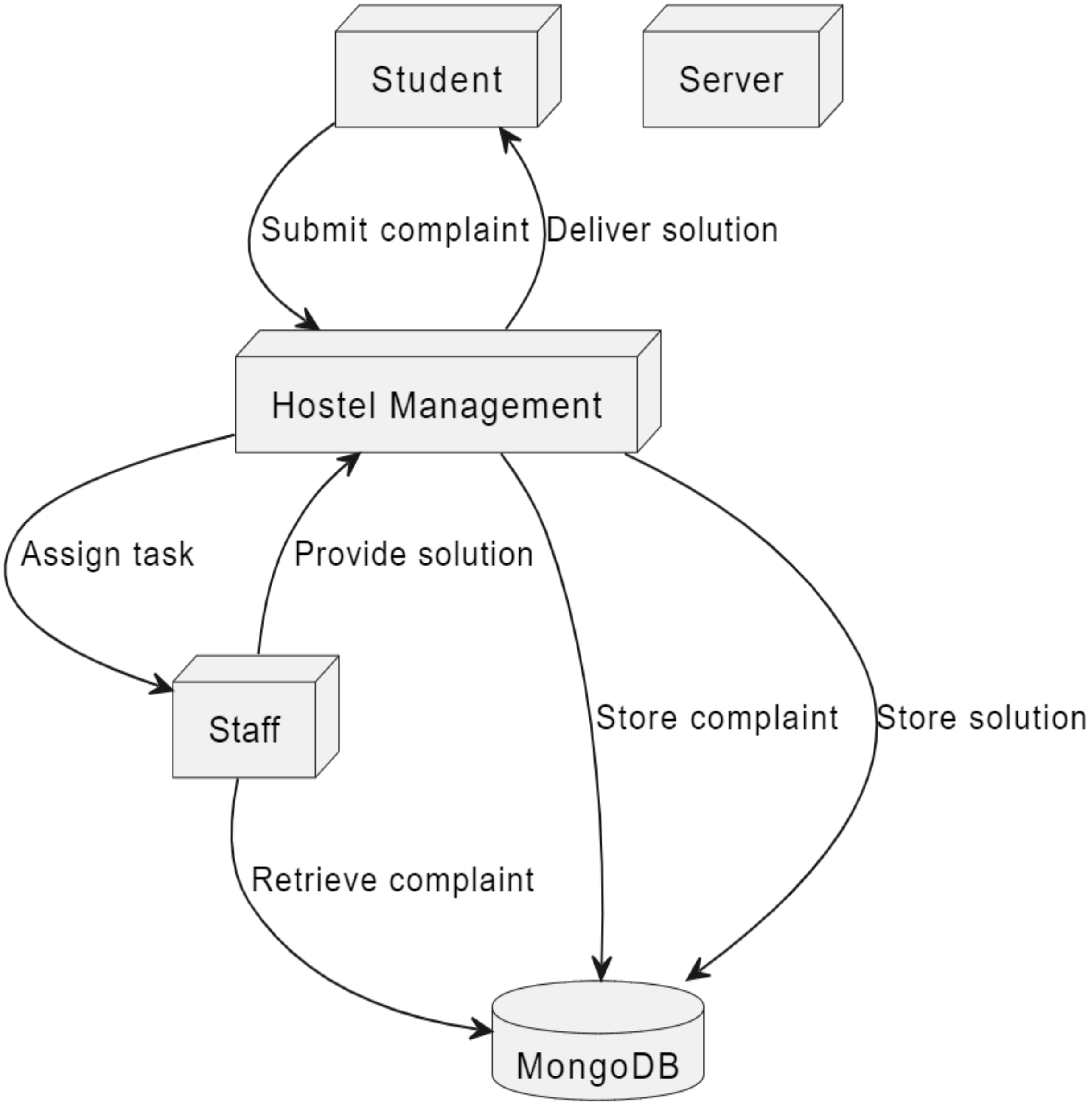
1)Component diagram :-



2) Deployment Diagram :-

Deployment Diagram for Hostel Complaint System




Student
Server
Submit complaint
Deliver solution
Hostel Management
Assign task
Provide solution
Staff
Store complaint
Store solution
Retrieve complaint
MongoDB