

Hostel Complaint Management System

Software Design and Requirement Specification



Session: 2021 – 2025

Submitted By:

Afaq Bin Asghar (2021-CS-615)

Muhammad Numan (2021-CS-633)

Sana Maqbool (2021-CS-647)

Supervised by:

Mr. Hafiz Muhammad Danish

Department of Computer Science,
University of Engineering and Technology, Lahore, Pakistan

Contents

1 Software Requirement Specification	4
1.1 Functional Requirements	4
1.2 Non-Functional Requirements	5
1.3 Use Case	6
2 Use Case Tables	7
3 Design Specification	12
3.1 Wireframes	12
3.2 Activity Diagram	21
3.3 State Diagram	22
3.4 Sequence Diagram	23
3.5 Collaboration Diagram	30
3.6 Class Diagram	31
3.7 Component Diagram	32
3.8 Deployment Diagram	33
3.9 ER Diagram	34
4 References	35

List of Figures

1.1	Use-Case Diagram	6
3.1	Login page	12
3.2	Forgot Password	13
3.3	Verification	13
3.4	Student's Dashboard	14
3.5	Complaints Form	14
3.6	Complaints List	15
3.7	Track Complaints	15
3.8	RT's Dashboard	16
3.9	Rejection Reason	16
3.10	Graphical View	17
3.11	Warden's Dashboard	17
3.12	End User List	18
3.13	Munshi Dashboard	18
3.14	Complaint Resolved	19
3.15	CC Dashboard	19
3.16	RT List	20
3.17	Warden list	20
3.18	Graphical View	21
3.19	Activity Diagram	21
3.20	State Diagram	22
3.21	Sequence Diagram	29
3.22	Collaboration Diagram	31
3.23	Class Diagram	31
3.24	Component Diagram	32
3.25	Deployment Diagram	33
3.26	ER Diagram	34

List of Tables

2.1	Use Case 1: User Authentication and Sign In	7
2.2	Use Case 2: Submit Complaint	8
2.3	Use Case 3: Track Complaint Status	8
2.4	Use Case 4: Accept/Reject Complaint	9
2.5	Use Case 5: Forward Complaint to Warden or Munshi	9
2.6	Use Case 6: Assign Complaint to End User	10
2.7	Use Case 7: Resolve Complaint	10
2.8	Use Case 8: View Graphical Overview of Complaints	11
2.9	Use Case 9: Manage Users and Hostels	11

Chapter 1

Software Requirement Specification

The project, titled **Hostel Complaint Management System**, is a web application designed to streamline the management of complaints raised by students residing in hostels. This system facilitates seamless communication between students and hostel authorities, ensuring efficient resolution of issues. Key features include complaint submission, status tracking, and a graphical overview of complaints for Resident Tutors (RT) and Campus Coordinators (CC). The application also incorporates user roles such as students, RTs, wardens, and end users, each with specific functionalities tailored to their responsibilities. By centralizing complaint management, this system enhances the responsiveness of hostel staff while providing students with a user-friendly interface to report and monitor their complaints. The system aims to foster a supportive living environment through effective issue resolution and management, ultimately promoting student satisfaction and well-being within the hostel community.

1.1 Functional Requirements

The functional requirements define the system's core behavior and operations. The Hostel Complaint Management System has the following functional requirements:

- **User Authentication:** The system must allow users (students, RTs, wardens, munshis, end users, and campus coordinators) to log in using their email and password. The system must provide a password recovery option for users.
- **Role-Based Access Control:** The system must identify and grant access to functionalities based on user roles (Student, RT, Warden, Munshi, End User, CC).
- **Complaint Management:** Students must be able to submit complaints with relevant details (title, description, attachments). RTs must be able to accept, reject, or forward complaints to the warden or munshi. Wardens must be able to assign complaints to end users and view/manage end users. Munshis and end users must be able to resolve complaints.
- **Complaint Tracking:** Users must be able to track the status of their submitted complaints.
- **Graphical Overview of Complaints:** The system must provide a graphical representation of complaints for RTs and CCs, displaying the number of accepted, pending, and rejected complaints.

- **Hostel Management:** Campus Coordinators must be able to add and manage hostels, RTs, wardens, and end users. The system must provide functionalities to view and manage graphical statistics related to complaints.
- **Notifications:** The system must notify users about updates on their complaints, such as status changes or resolutions.

1.2 Non-Functional Requirements

Non-functional requirements describe the quality attributes of the system. The Hostel Complaint Management System has the following non-functional requirements:

- **Performance:** The system must handle a minimum of 100 concurrent users without degradation in performance. The average response time for any user action must be less than 2 seconds.
- **Security:** User data must be encrypted during transmission and storage. The system must implement role-based access control to restrict access to sensitive functions based on user roles.
- **Usability:** The user interface must be intuitive and easy to navigate for all user roles. Help documentation must be available within the application to assist users.
- **Scalability:** The system must be scalable to accommodate future growth, allowing for the addition of new users and functionalities without significant changes to the existing architecture.
- **Reliability:** The system must have a downtime of less than 5% annually. Backup and recovery mechanisms must be in place to prevent data loss in case of system failure.
- **Maintainability:** The codebase must follow industry standards and best practices to ensure ease of maintenance and updates. The system must allow for easy integration of new features as requirements evolve.
- **Compatibility:** The web application must be compatible with major browsers (Chrome, Firefox, Safari, Edge) and mobile devices. The system must support various operating systems, including Windows, macOS, and Linux.

1.3 Use Case

Use case diagrams give an overview of actors, functions, and interactions in the system. They are useful for:

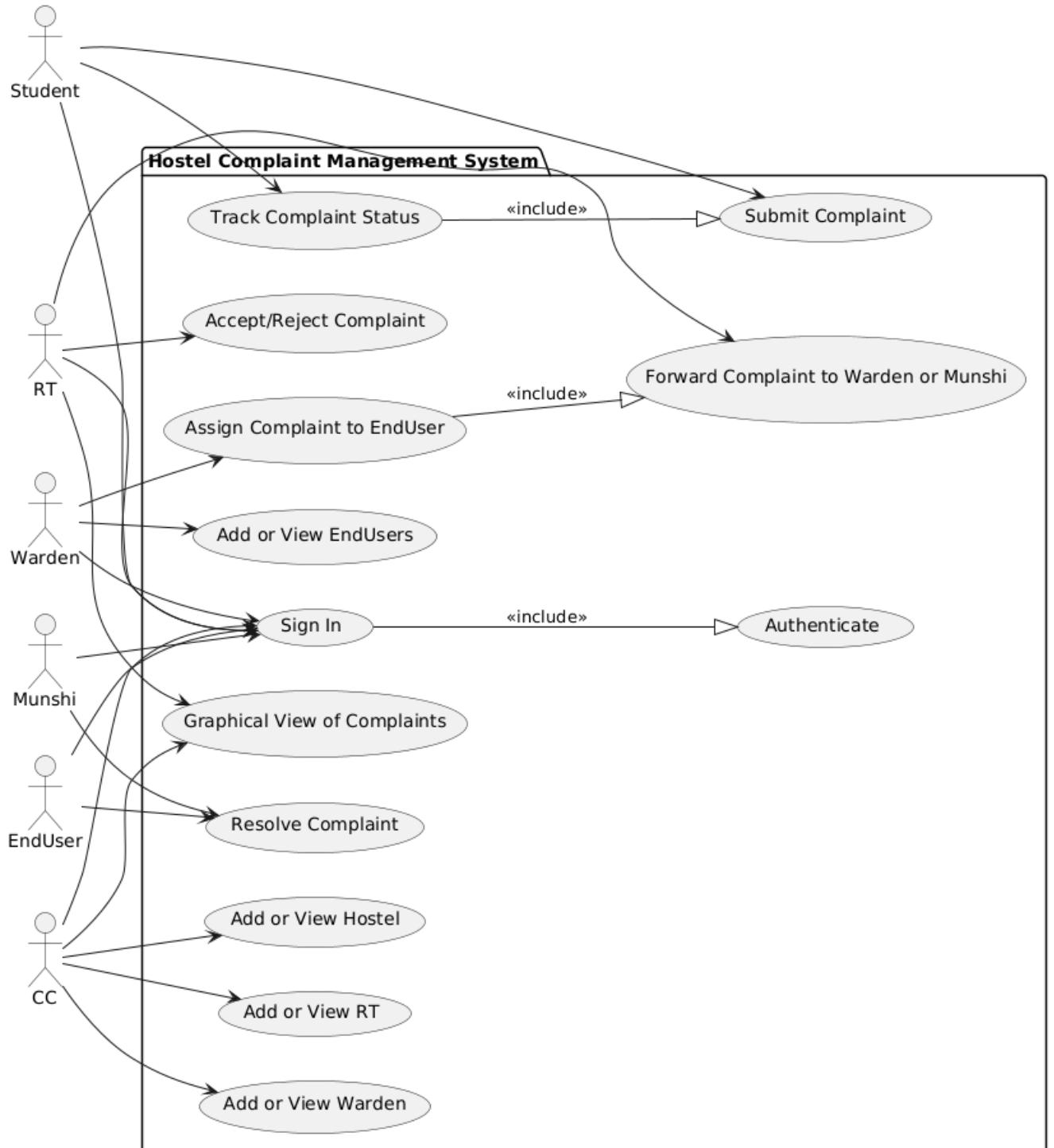


Figure 1.1: Use-Case Diagram

Chapter 2

Use Case Tables

Use Case Name	User Authentication and Sign In
Actor(s)	All Users
Description	Users can sign in using their credentials to access the system.
Precondition(s)	User must have valid credentials.
Postcondition(s)	User is authenticated and logged in.
Flow of Events	<ol style="list-style-type: none">1. User navigates to the login page.2. User enters credentials.3. System verifies credentials and grants access.
Alternate Flow	If the credentials are incorrect, the system shows an error.
Exceptions	Network failure during authentication.

Table 2.1: Use Case 1: User Authentication and Sign In

Use Case Name	Submit Complaint
Actor(s)	Student
Description	Students can submit complaints by providing necessary details like title, description, and any attachments.
Precondition(s)	Student must be authenticated and logged in.
Postcondition(s)	Complaint is submitted and stored in the system.
Flow of Events	<ol style="list-style-type: none"> 1. Student navigates to the "Submit Complaint" section. 2. Student fills out the complaint form. 3. Student submits the complaint.
Alternate Flow	If the form is incomplete, the system shows an error message.
Exceptions	Network failure during submission.

Table 2.2: Use Case 2: Submit Complaint

Use Case Name	Track Complaint Status
Actor(s)	Student
Description	Students can view the current status of complaints they have submitted.
Precondition(s)	Student must be authenticated and logged in.
Postcondition(s)	Status of the complaint is displayed to the student.
Flow of Events	<ol style="list-style-type: none"> 1. Student navigates to "Track Complaint Status". 2. The system retrieves the complaint details. 3. The system displays the current status of the complaint.
Alternate Flow	If the complaint ID is not found, the system shows an error message.
Exceptions	System error during status retrieval.

Table 2.3: Use Case 3: Track Complaint Status

Use Case Name	Accept/Reject Complaint
Actor(s)	RT, Warden
Description	RT or Warden can accept or reject a complaint based on its relevance and details.
Precondition(s)	Complaint must be submitted by a student.
Postcondition(s)	Complaint is either accepted for further processing or rejected.
Flow of Events	<ol style="list-style-type: none"> 1. RT/Warden views the complaint. 2. RT/Warden accepts or rejects the complaint.
Alternate Flow	If the complaint does not have sufficient details, the system notifies the student for additional information.
Exceptions	System error during acceptance/rejection.

Table 2.4: Use Case 4: Accept/Reject Complaint

Use Case Name	Forward Complaint to Warden or Munshi
Actor(s)	RT
Description	RT can forward complaints to the Warden or Munshi for further handling.
Precondition(s)	Complaint must be accepted by the RT.
Postcondition(s)	Complaint is forwarded to the appropriate authority.
Flow of Events	<ol style="list-style-type: none"> 1. RT views accepted complaints. 2. RT selects a complaint to forward. 3. RT forwards the complaint to the Warden or Munshi.
Alternate Flow	If the system cannot forward, it queues the complaint for retry.
Exceptions	System failure during forwarding.

Table 2.5: Use Case 5: Forward Complaint to Warden or Munshi

Use Case Name	Assign Complaint to End User
Actor(s)	Warden, Munshi
Description	Warden or Munshi assigns a complaint to an End User for resolution.
Precondition(s)	Complaint must be forwarded to Warden/Munshi.
Postcondition(s)	Complaint is assigned to an End User.
Flow of Events	<ol style="list-style-type: none"> 1. Warden/Munshi views forwarded complaints. 2. Warden/Munshi selects an End User. 3. Warden/Munshi assigns the complaint.
Alternate Flow	If no End User is available, the system notifies the RT.
Exceptions	System error during assignment.

Table 2.6: Use Case 6: Assign Complaint to End User

Use Case Name	Resolve Complaint
Actor(s)	End User
Description	End User resolves the assigned complaint and updates its status.
Precondition(s)	Complaint must be assigned to the End User.
Postcondition(s)	Complaint status is updated to "Resolved".
Flow of Events	<ol style="list-style-type: none"> 1. End User views assigned complaint. 2. End User resolves the complaint. 3. End User marks the complaint as "Resolved".
Alternate Flow	If the complaint cannot be resolved, End User marks it for further assistance.
Exceptions	System error during status update.

Table 2.7: Use Case 7: Resolve Complaint

Use Case Name	View Graphical Overview of Complaints
Actor(s)	RT, Warden
Description	RT or Warden can view a graphical dashboard showing the status and statistics of complaints.
Precondition(s)	Complaints must be present in the system.
Postcondition(s)	Dashboard with relevant statistics is displayed.
Flow of Events	<ol style="list-style-type: none"> 1. RT/Warden navigates to the dashboard. 2. The system retrieves complaint statistics. 3. The system displays a graphical overview.
Alternate Flow	If there is no data, the system shows an appropriate message.
Exceptions	System error during data retrieval.

Table 2.8: Use Case 8: View Graphical Overview of Complaints

Use Case Name	Manage Users and Hostels
Actor(s)	Admin
Description	Admin can add, update, or remove hostels, RTs, Wardens, and End Users.
Precondition(s)	Admin must be authenticated.
Postcondition(s)	Changes to users or hostels are reflected in the system.
Flow of Events	<ol style="list-style-type: none"> 1. Admin navigates to the user or hostel management section. 2. Admin adds, updates, or removes entries.
Alternate Flow	If any operation fails, the system notifies the Admin.
Exceptions	System error during modification.

Table 2.9: Use Case 9: Manage Users and Hostels

Chapter 3

Design Specification

3.1 Wireframes

Wireframes are visual guides suggesting the structure of an interface, created before design work starts.

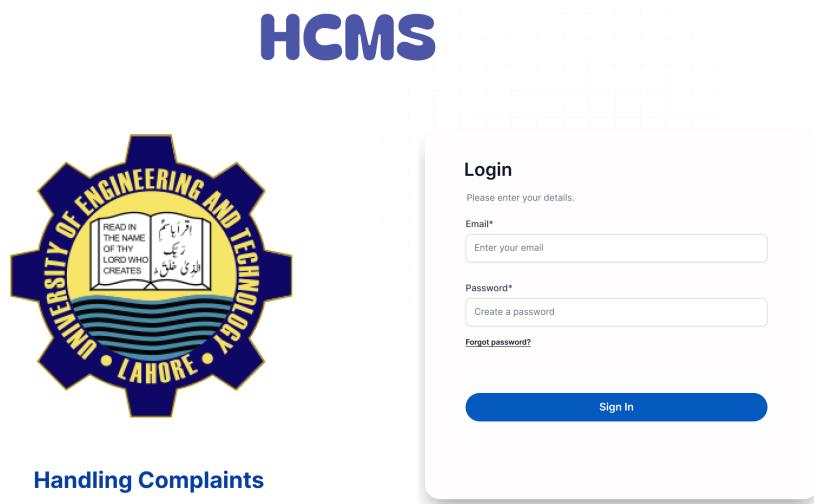


Figure 3.1: Login page



Handling Complaints

Figure 3.2: Forgot Password

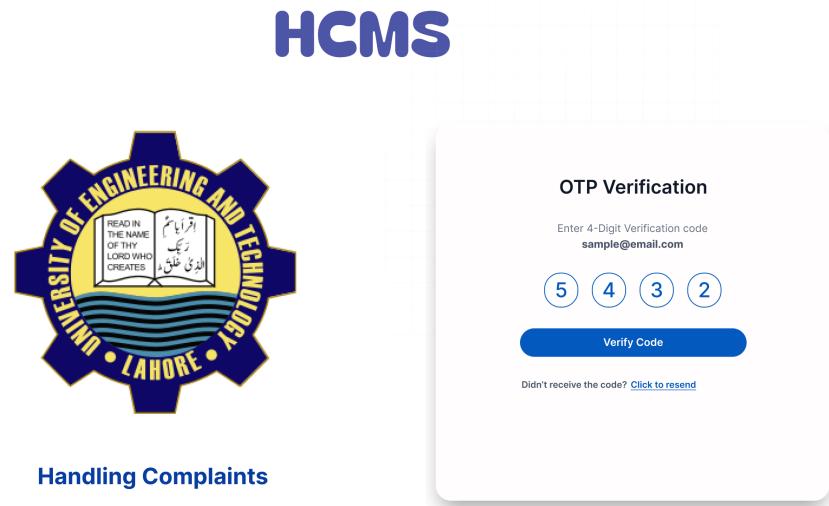


Figure 3.3: Verification

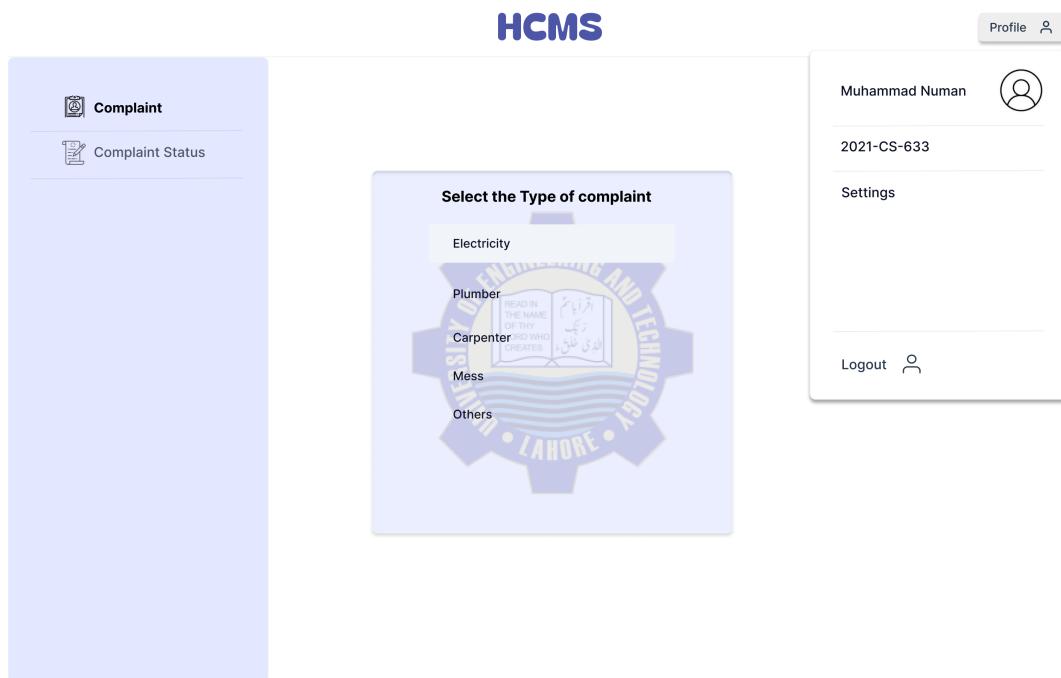


Figure 3.4: Student's Dashboard

The form is titled "Complaint Form" and includes fields for "Complaint title" (set to "Electricity"), "Registration number" (set to "2021-CS-633"), "Room number" (set to "13"), "Date" (set to "11/11/2021"), and a large "Description" area. There is also a section for "Attachment file here" with a "Submit" button at the bottom. The central logo is identical to the one in Figure 3.4.

Figure 3.5: Complaints Form

The screenshot shows the HCMS Complaint Status interface. On the left, there's a sidebar with icons for Complaint and Complaint Status. The main area displays a table titled "Complaint list" with columns for "Complaint list", "Date", and "Status". The table contains the following data:

Complaint list	Date	Status
Electricity	13-05-2024	<button>Check</button>
Mess	12-08-2024	<button>Check</button>
Plumber	24-03-2024	<button>Check</button>
Mess	23-03-2024	<button>Check</button>
others	09-02-2024	<button>Check</button>
Electricity	08-01-2024	<button>Check</button>
Carpenter	09-11-2023	<button>Check</button>

In the center of the page is a large, circular emblem of the University of Engineering and Technology Lahore, featuring a gear design with the university's name in English and Urdu.

Figure 3.6: Complaints List

The screenshot shows the HCMS Track Complaints interface. On the left, there's a sidebar with icons for Complaint and Complaint Status. The main area displays a "Status" section with a horizontal timeline showing five stages: Student, RT, Warden/Munshi, End User, and Done. The "RT" stage is highlighted in blue, indicating the current status of the complaint. Below the timeline is a text box containing placeholder text about a student's labor rights. The text box also includes the university's logo at the bottom.

Figure 3.7: Track Complaints

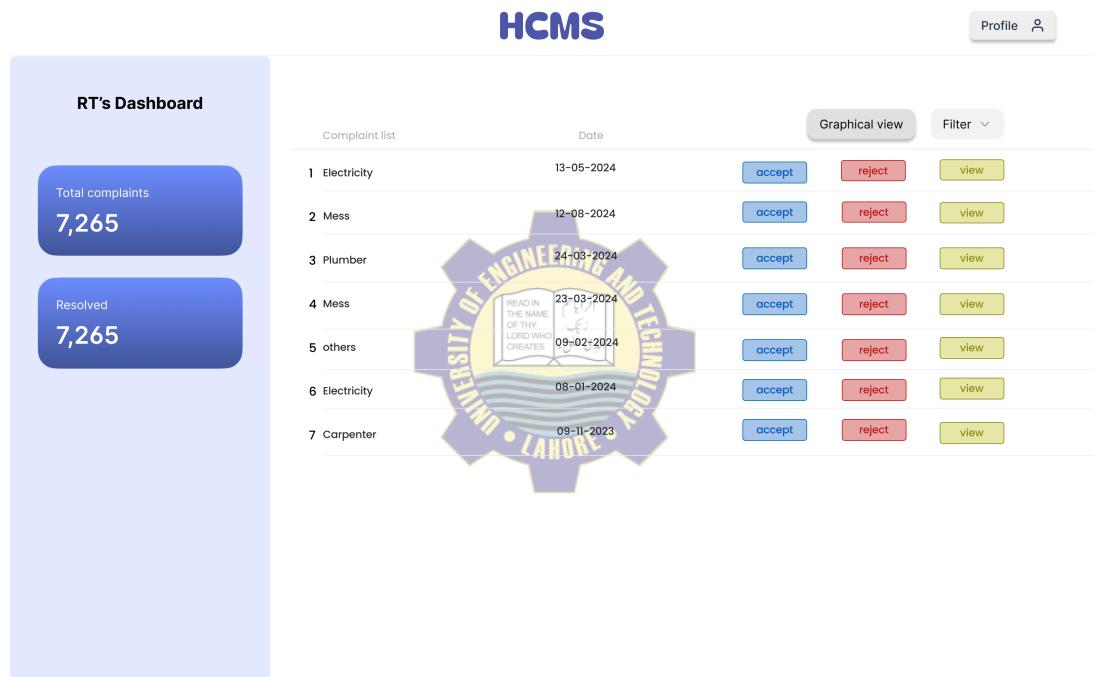


Figure 3.8: RT's Dashboard

A modal window titled "Reason:" is displayed. It contains a large text input field for entering the reason. At the bottom right of the input field is a blue rounded rectangular button labeled "Done".

Figure 3.9: Rejection Reason

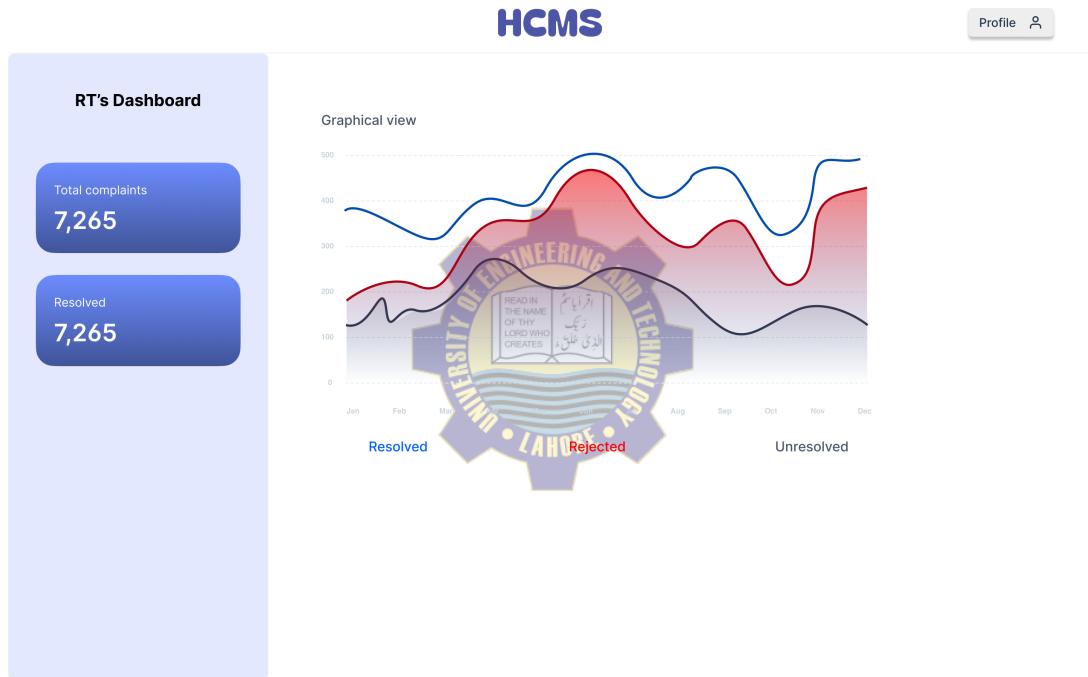


Figure 3.10: Graphical View

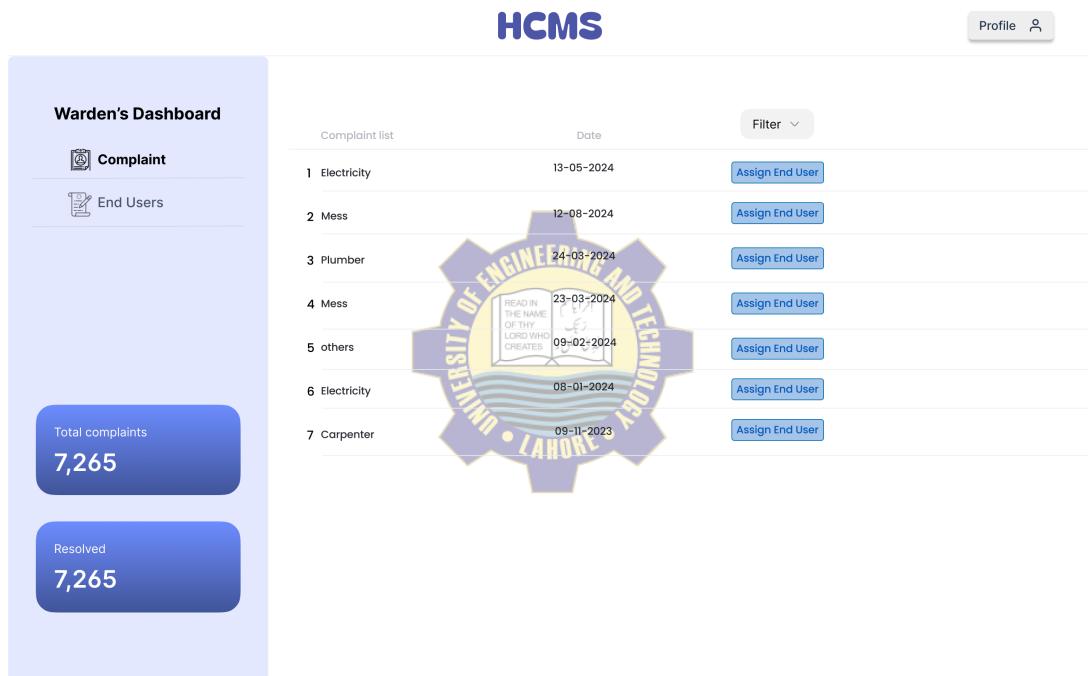


Figure 3.11: Warden's Dashboard

HCMS

Warden's Dashboard

Complaint

End Users

Total complaints
7,265

Resolved
7,265

Carpenter list	Status	Filter
1 Carpenter1	Free	Assign End User
2 Carpenter2	busy	Assign End User
3 Carpenter3	Free	Assign End User
4 Carpenter4	Free	Assign End User
5 Carpenter5	Free	Assign End User
6 Carpenter6	Busy	Assign End User
7 Carpenter7	Free	Assign End User



Add new Remove

Figure 3.12: End User List

HCMS

Munshi's Dashboard

Total complaints
7,265

Resolved
7,265

Complaint list	Date	Filter
1 Complaint1	13-05-2024	Done
2 Complaint2	12-08-2024	Done
3 Complaint3	24-03-2024	Done
4 Complaint4	23-03-2024	Done
5 Complaint12	09-02-2024	Done
6 Complaint13	08-01-2024	Done
7 Complaint13	09-11-2023	Done



Figure 3.13: Munshi Dashboard

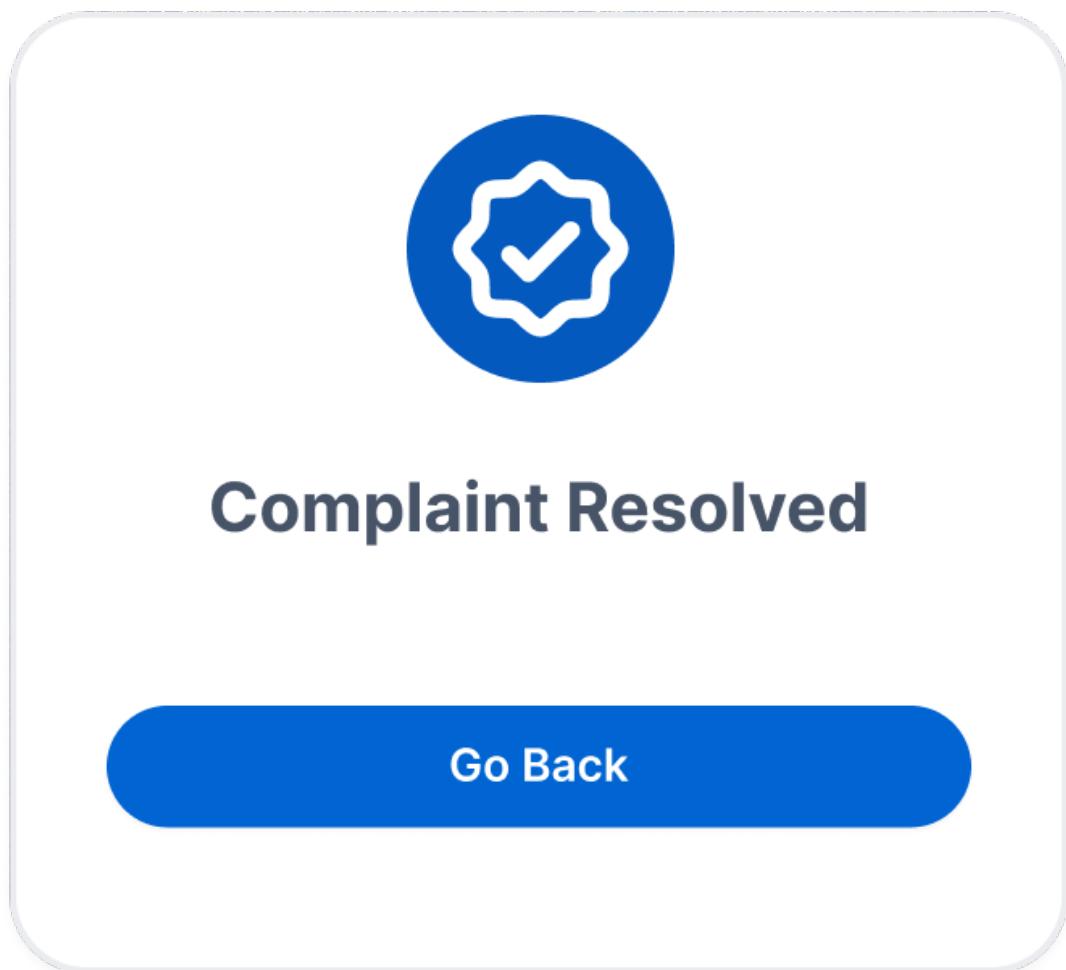


Figure 3.14: Complaint Resolved

A screenshot of the CC Dashboard. At the top right is the "HCMS" logo and a "Profile" button. The main area is titled "CC's Dashboard" and shows a table of hostels. The table has two columns: "Hostel list" and "Status". The data is as follows:

Hostel list	Status
1 Hostel1	Furnished
2 Hostel2	Unfurnished
3 Hostel4	Furnished
4 Hostel3	Furnished
5 Hostel5	Furnished
6 Hostel8	Unfurnished
7 Hostel6	Furnished

A watermark of the University of Engineering and Technology Lahore logo is overlaid on the dashboard. On the left, there is a sidebar with a purple background and the title "Hostel". It contains three items: "RT", "Warden", and "Graphical View", each with an icon.

Figure 3.15: CC Dashboard

HCMS

Profile

CC's Dashboard

RT list Hostel list Filter

1 RT1 Hostel1
2 RT12 Hostel2
3 RT2 Hostel3
4 RT13 Hostel2
5 RT3 Hostel15
6 RT4 Hostel7
7 RT15 Hostel9

Add new Remove

Hostel RT Warden Graphical View

Figure 3.16: RT List

HCMS

Profile

CC's Dashboard

Warden List

1 Warden1
2 Warden1
3 Warden1
4 Warden1
5 Warden1
6 Warden1
7 Warden1

Add new Remove

Hostel RT Warden Graphical View

Figure 3.17: Warden list



Figure 3.18: Graphical View

3.2 Activity Diagram

Activity diagrams represent workflows in a graphical way and are useful for describing business or operational workflows.

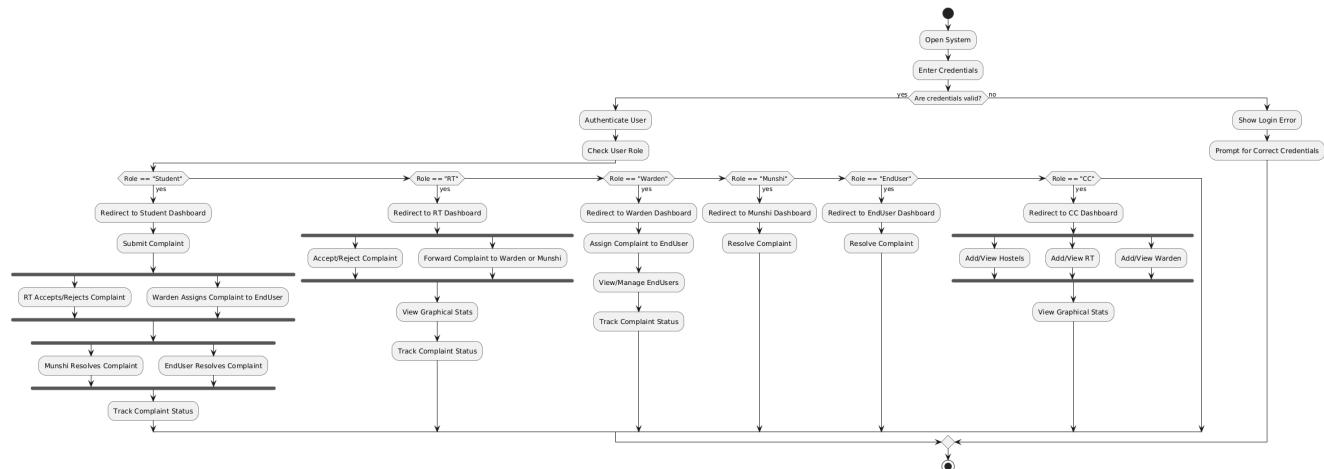


Figure 3.19: Activity Diagram

3.3 State Diagram

State diagrams show how objects behave based on their state at any given moment.

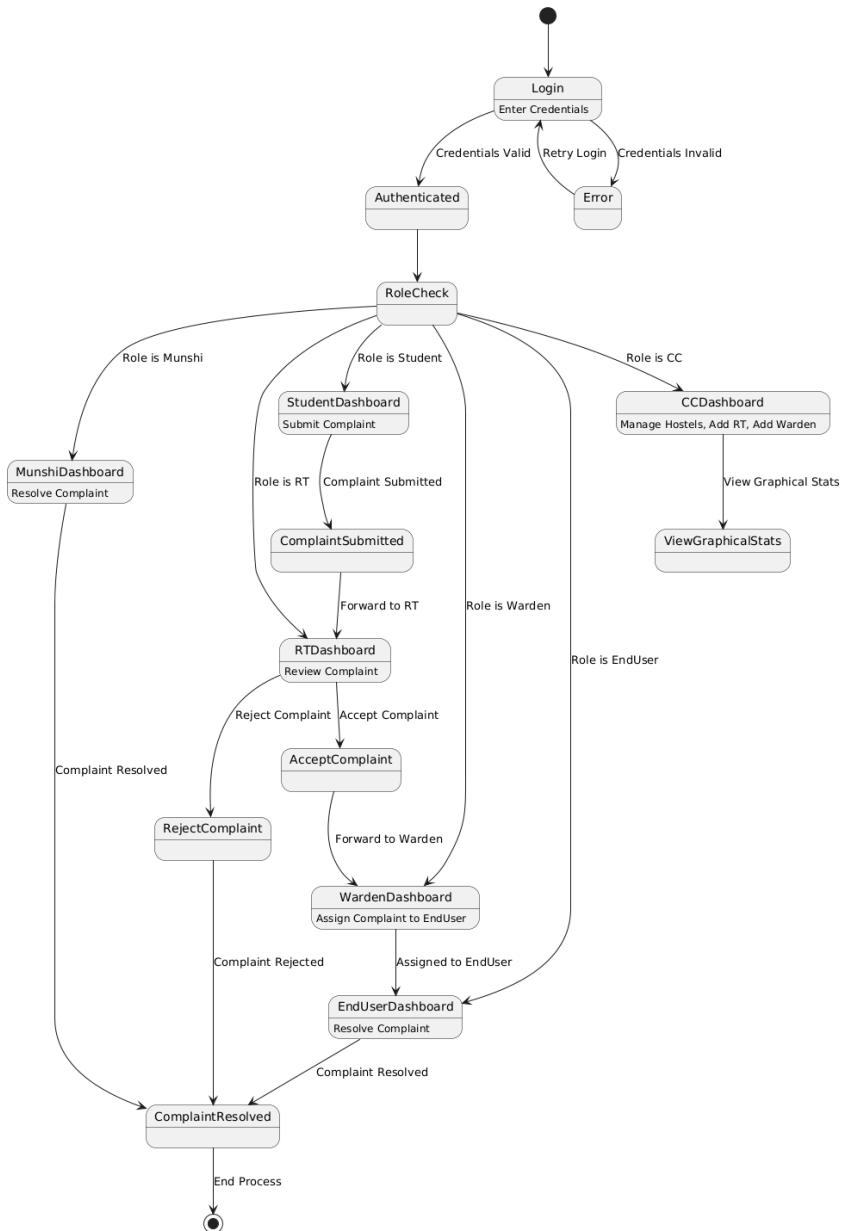
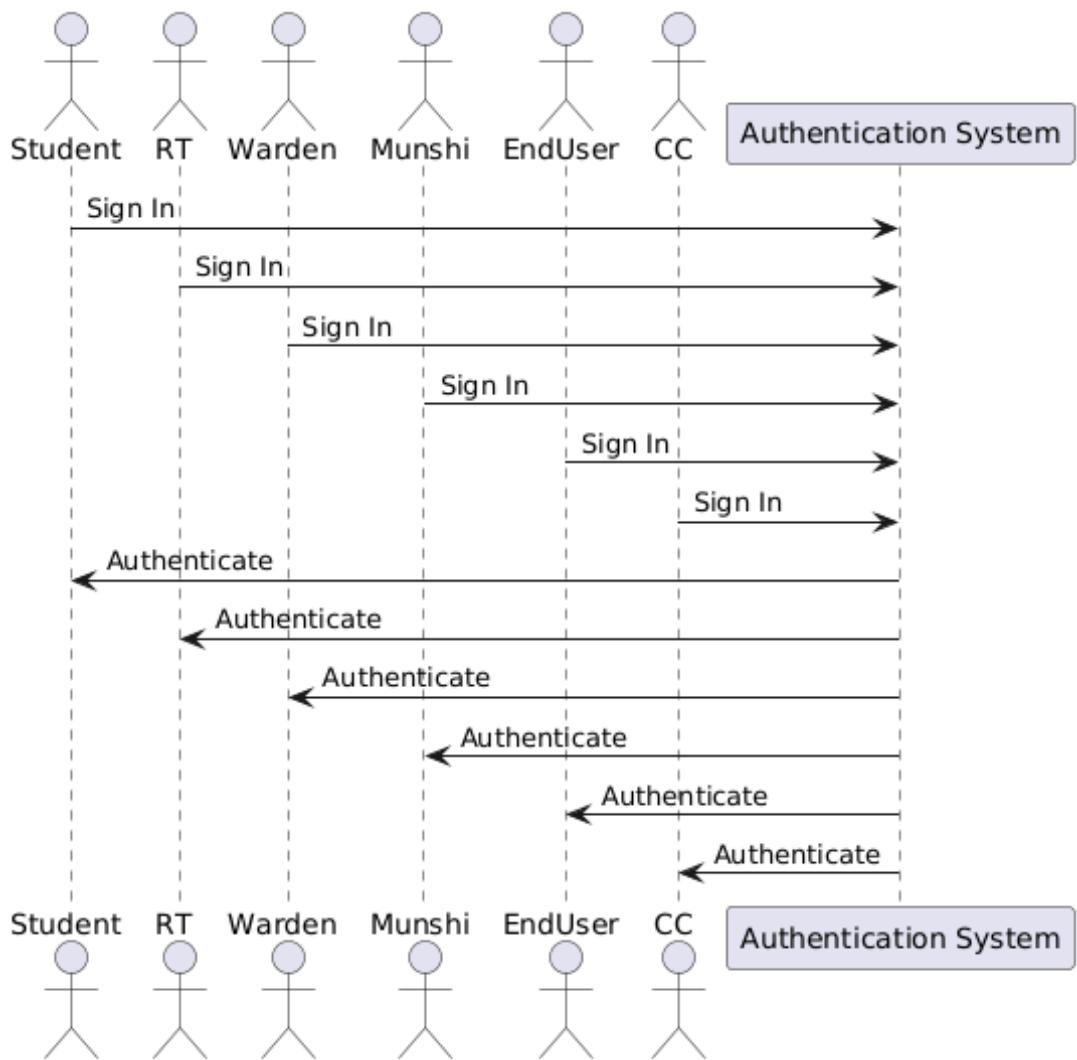
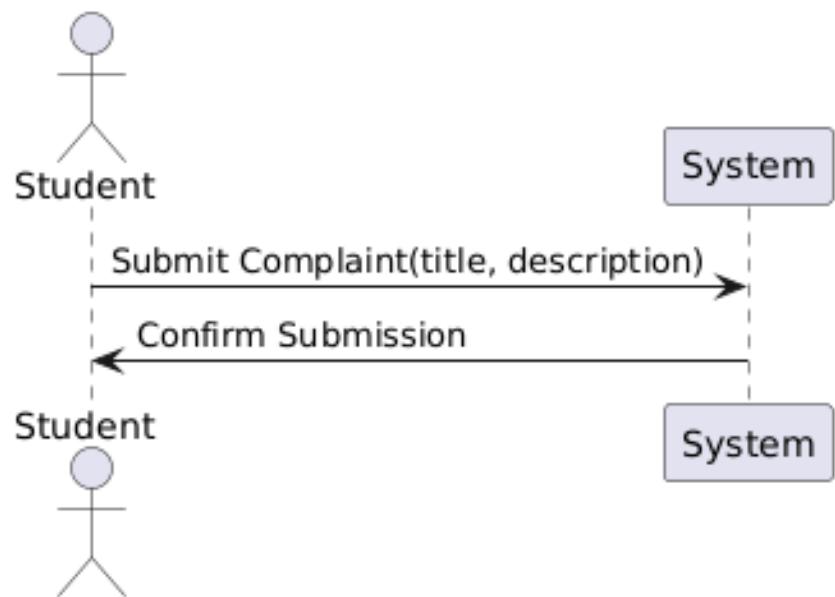
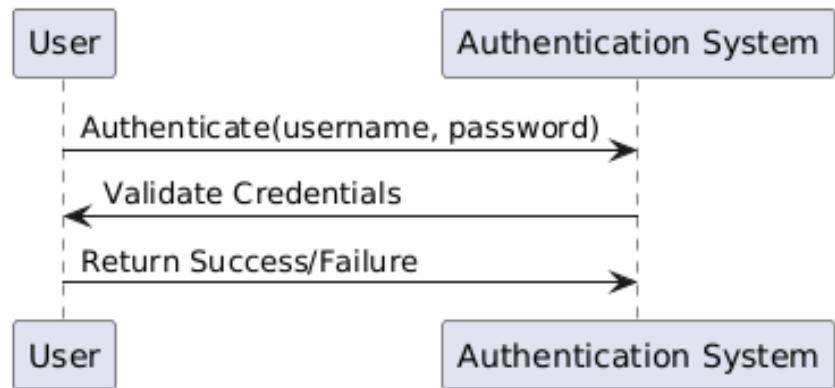


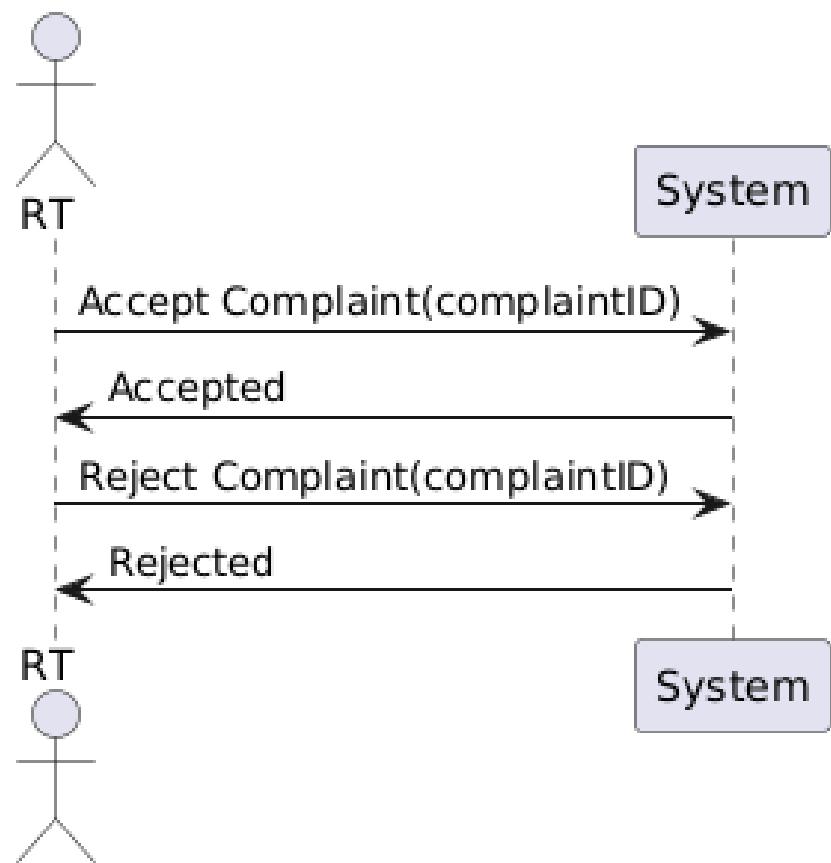
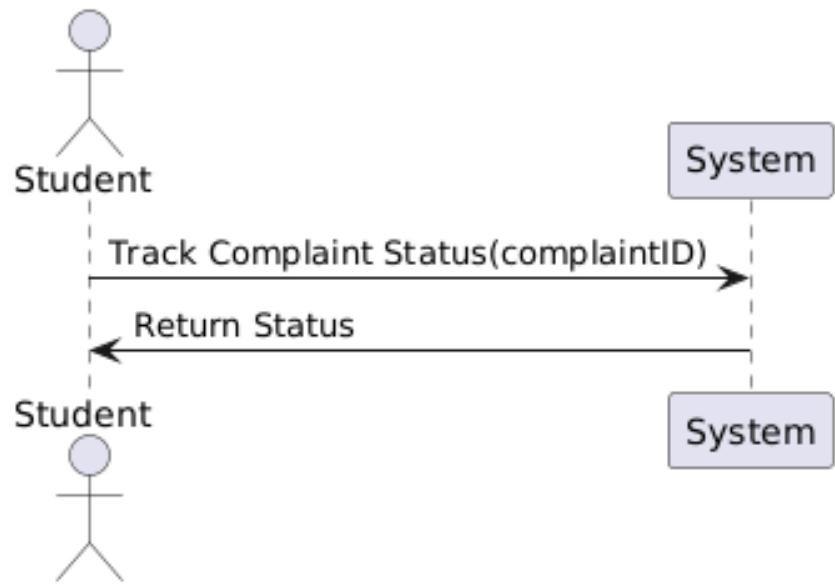
Figure 3.20: State Diagram

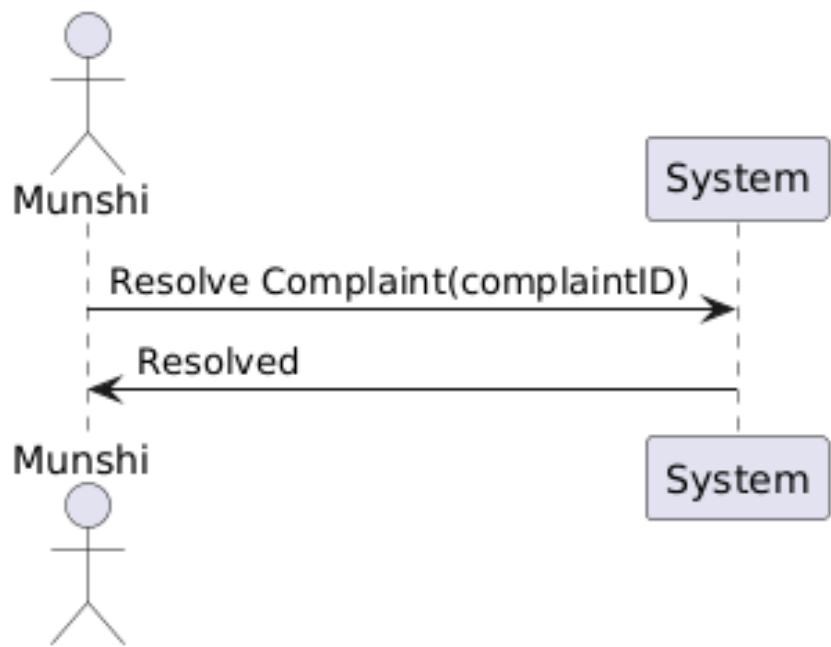
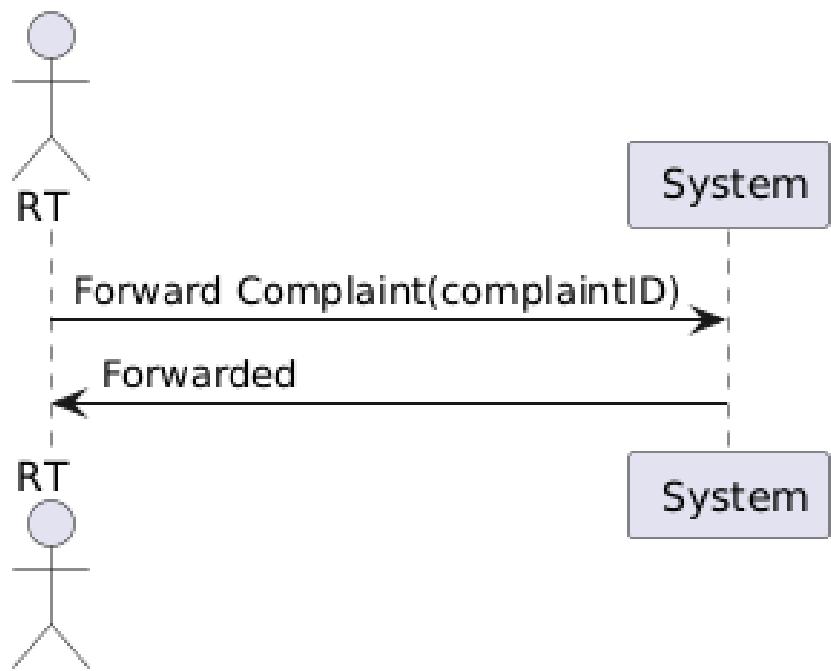
3.4 Sequence Diagram

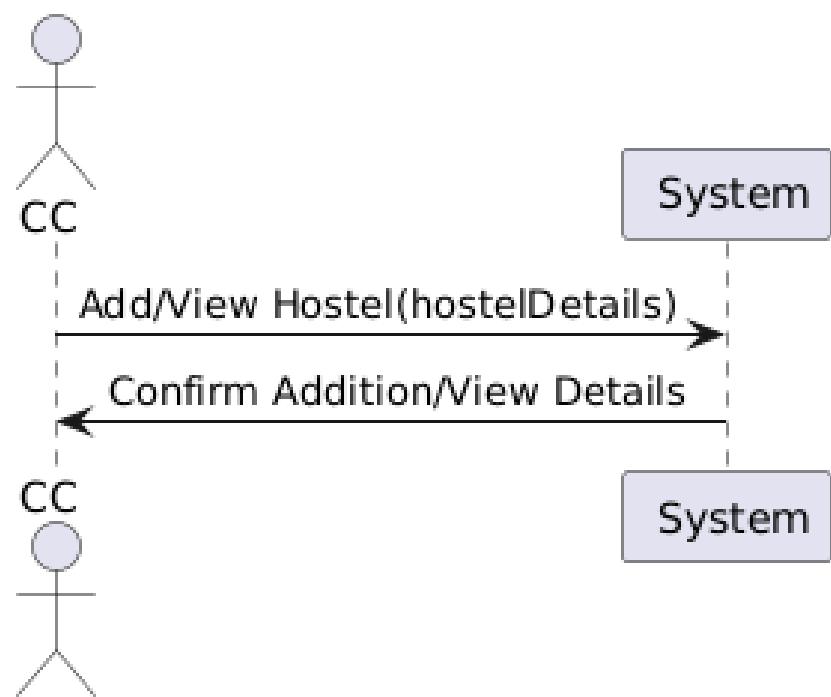
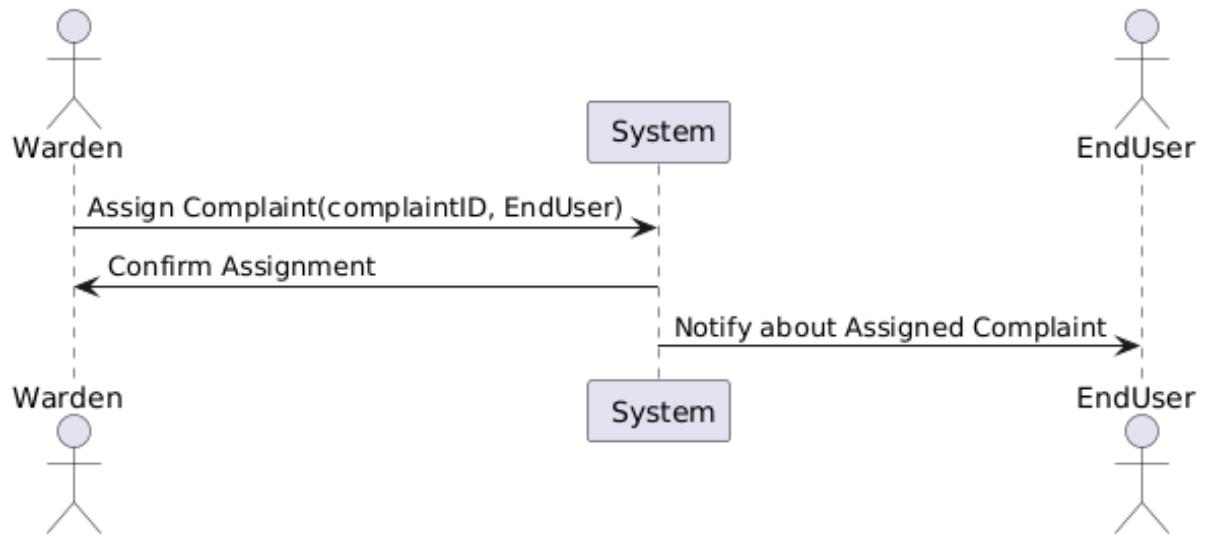
Sequence diagrams depict object interactions in a specific scenario, with processes represented vertically and interactions shown as arrows.

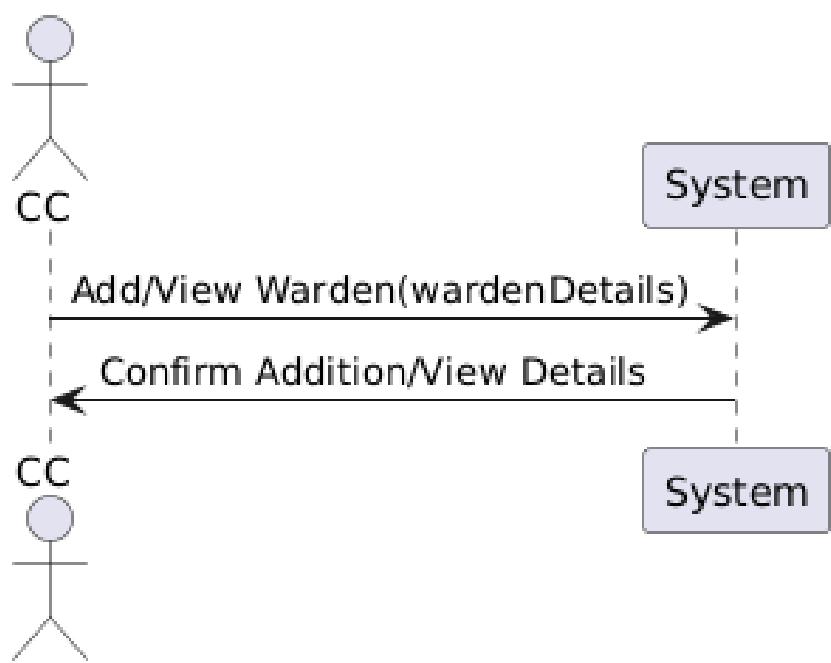
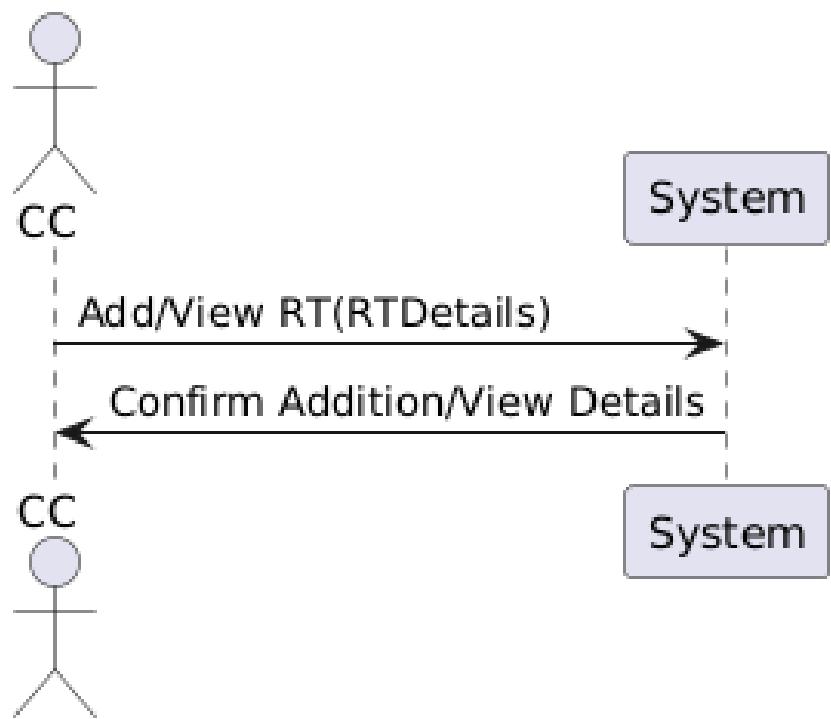












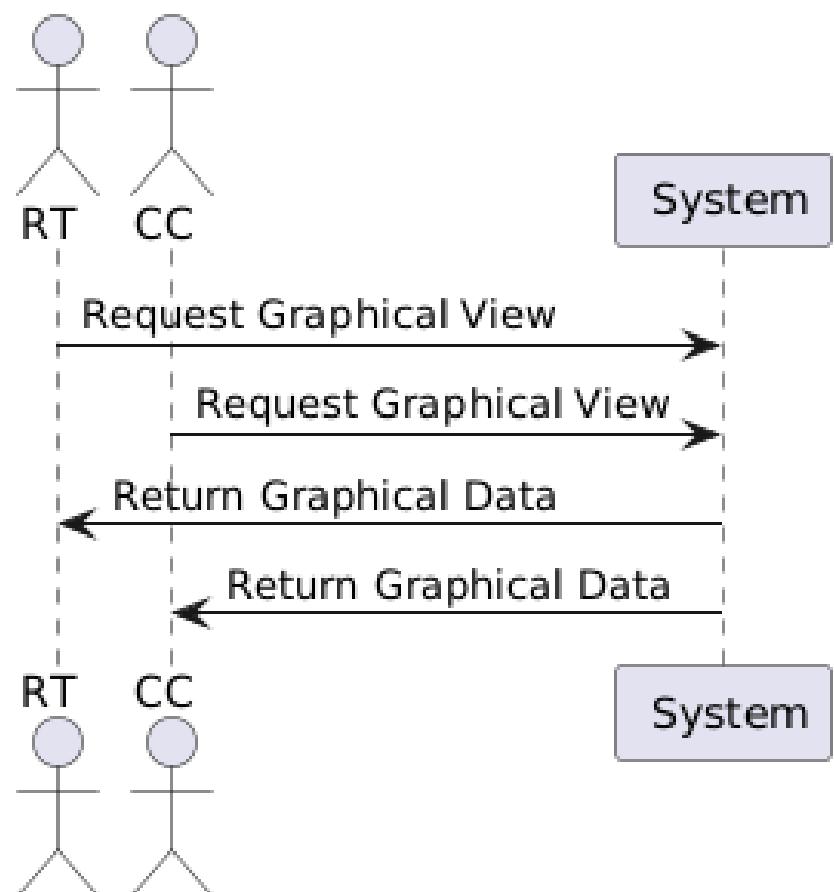
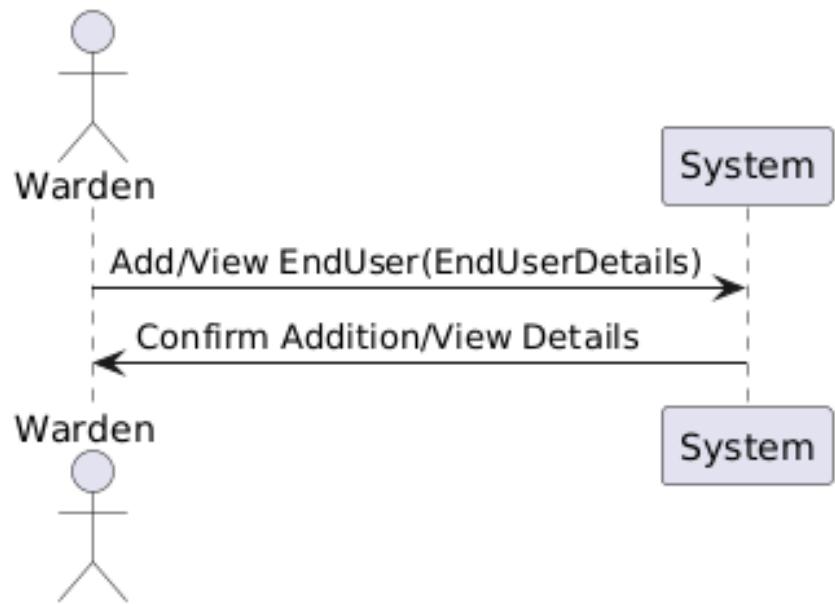
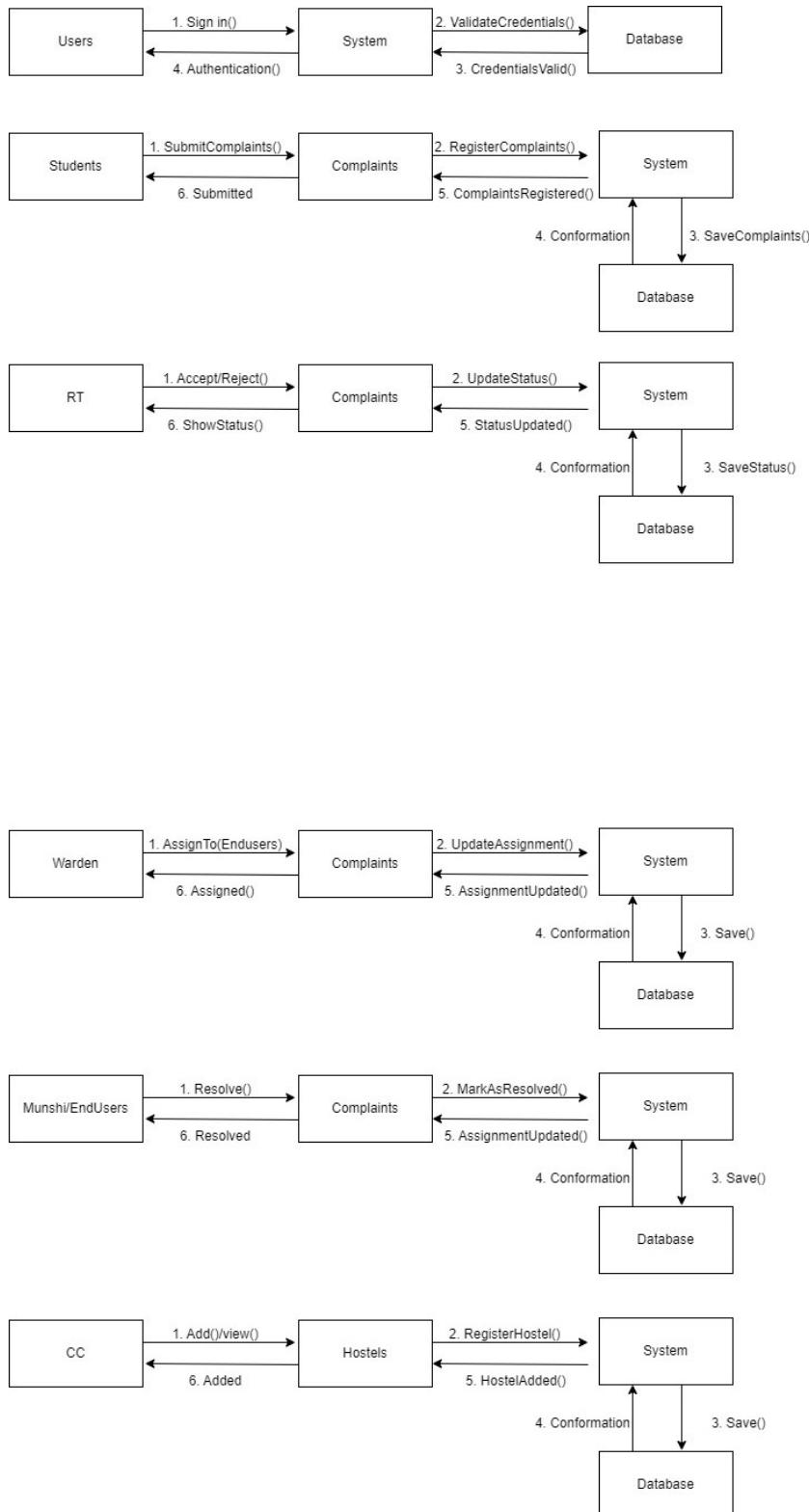


Figure 3.21: Sequence Diagram

3.5 Collaboration Diagram

Collaboration diagrams (or Communication Diagrams) show how objects interact for a use case or part of it, helping to clarify roles and object responsibilities.



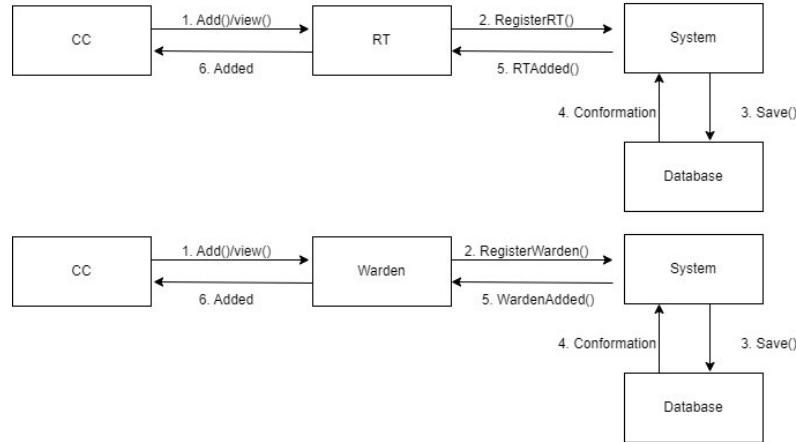


Figure 3.22: Collaboration Diagram

3.6 Class Diagram

Class diagrams are the backbone of object-oriented systems, showing classes, their attributes, methods, and the relationships between them.

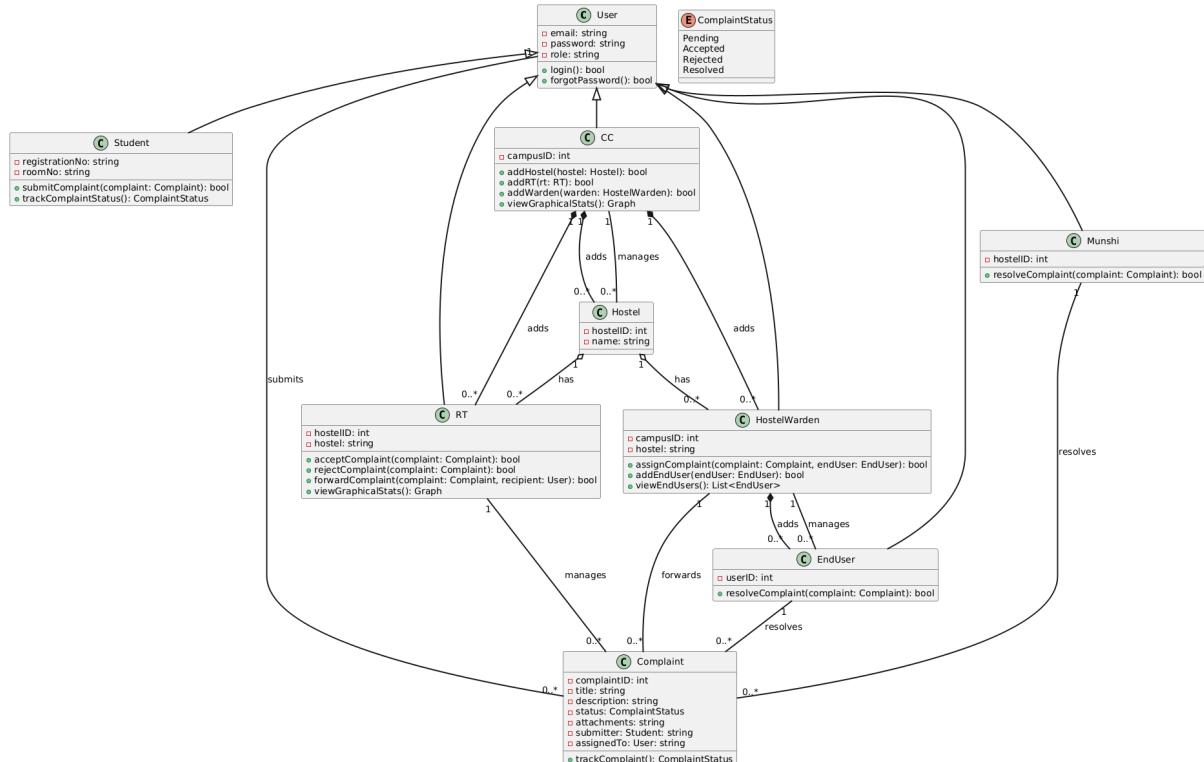


Figure 3.23: Class Diagram



3.7 Component Diagram

Component diagrams show the structural relationship between components in a software system, often used for complex systems with many components.

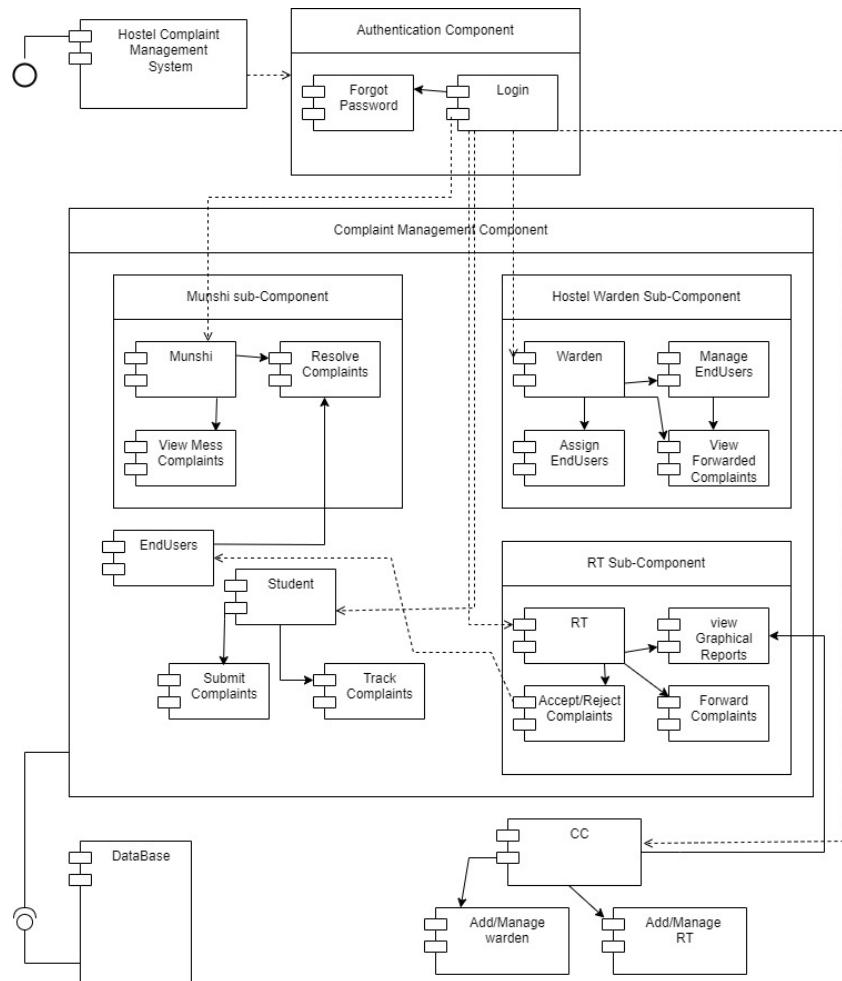


Figure 3.24: Component Diagram

3.8 Deployment Diagram

Deployment diagrams show hardware configurations and software deployment across multiple machines.

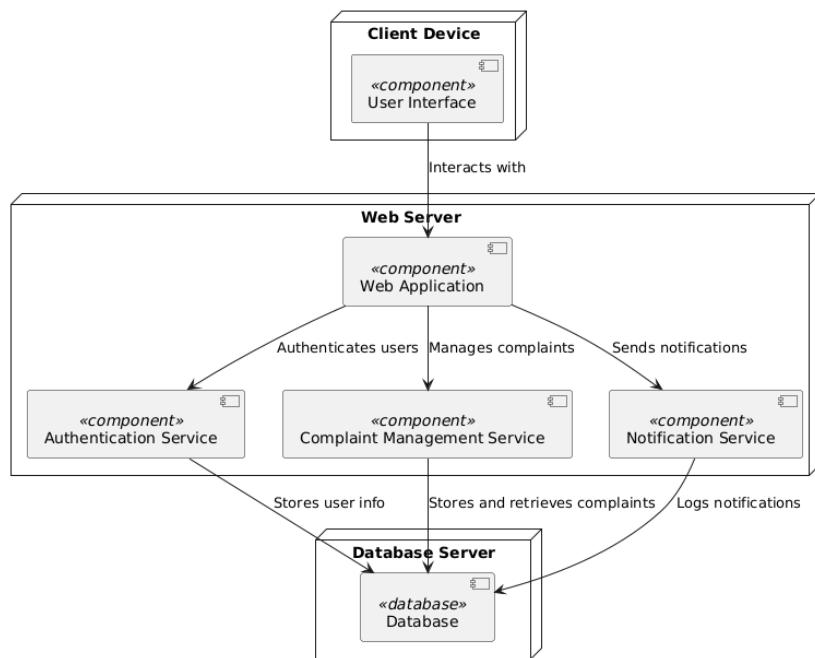


Figure 3.25: Deployment Diagram

3.9 ER Diagram

ER diagrams represent relationships among database entity sets and attributes, forming the logical structure of a database.

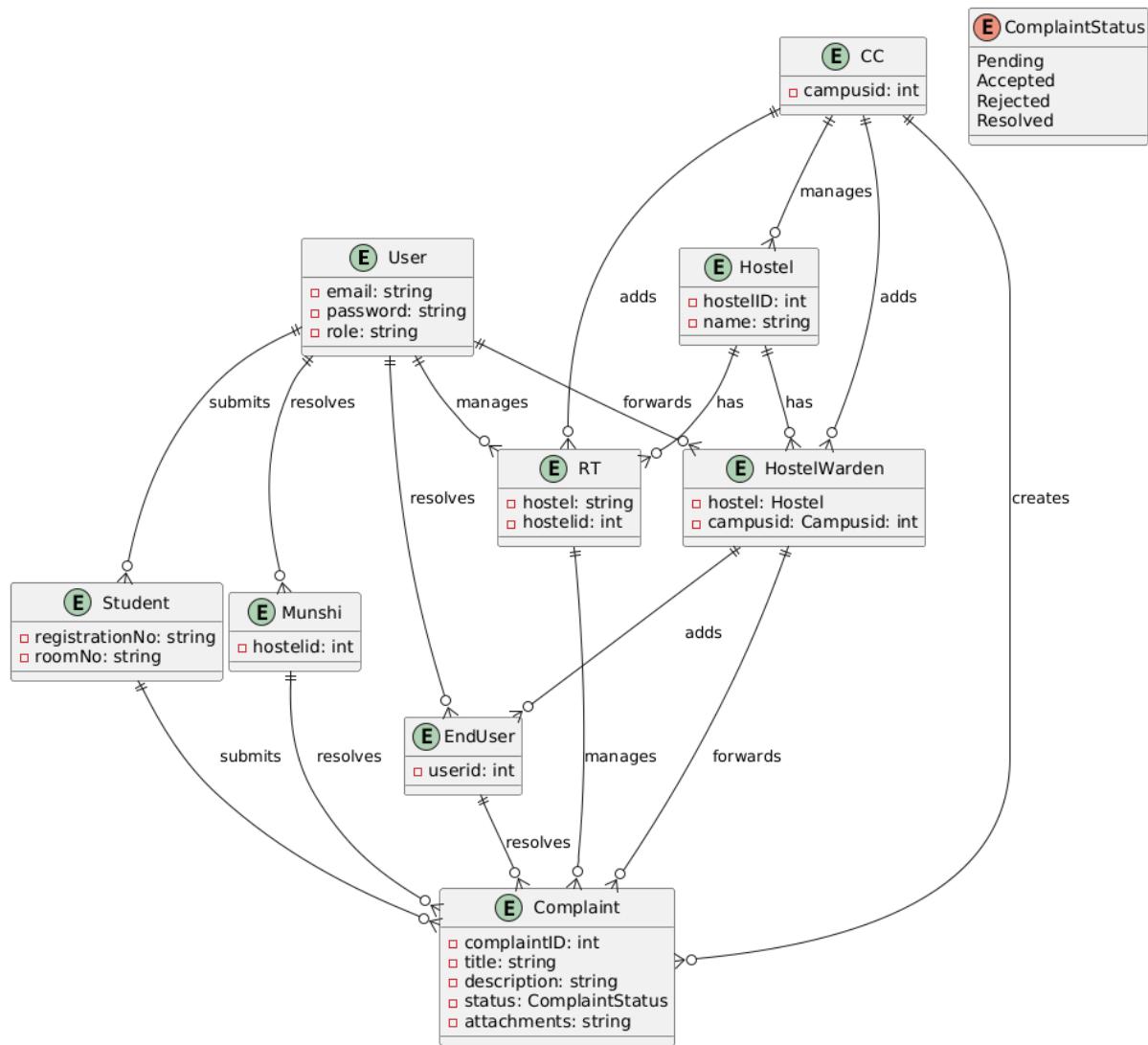


Figure 3.26: ER Diagram

□ ○
 Private Public

Chapter 4

References

- 1 Kaur, Simran. "Effective Complaint Management Systems for Hostels." International Journal of Management and Humanities, vol. 5, no. 2, 2020, pp. 45-58.
- 2 Smith, John. "Improving Hostel Living: A Case Study on Complaint Management." Hostel Management Magazine, May 2021, pp. 20-25.
- 3 "Hostel Complaint Management System." Hostel Solutions Inc., 2023. Accessed March 21, 2024. URL: <https://www.hostelsolutions.com/complaint-management>
- 4 Patel, Ravi. "Streamlining Hostel Complaints: Best Practices." Hostel Weekly, June 2022, pp. 10-15. Reproduced in Academic Search Complete. University Library, New York. Accessed March 21, 2024.