HOSTEL COMPLAINT SYSTEM

UCS503 Software Engineering Project Report

Mid-Semester Evaluation

Submitted By

101883023	Paras Garg
101703307	Lavish Jindal
101703011	Lokesh Arora
101703312	Lovish Jindal

Submitted To

Ms. Deepali Bhagat

Computer Science and Engineering Department, TIET

Thapar Institute of Engineering and Technology

August- December 2019

TABLE OF CONTENTS

1. Project Overview

This system will help the students residing in hostels to lodge their complaints online (anywhere from their rooms etc) instead of looking for the caretaker and manually lodge their complaints . This system provides ease to the hostel students and also helps in maintaining the record of complaints in the hostel.

Features :

- Ease to the students in lodging complaints.
- Report automatically sent to the warden after a particular time.
- Maintenance of record of previous complaints.

Implementation :

We will use MySQL for the database and Front End Web Development to design the entire front-end of the application. We can use CSS and BOOTSTRAP to strengthen our system and provide each and every Option/Button related to Complaints and MySQL backend can store previous records in form of tables.

Requirements for complaint registration :

- Student Roll No. / email-id
- Student password
- Room No. and Details

Types of Complaints :

- Electrical appliances fault
- Furniture related complaints
- Mosquito related problems
- Others

Goal :

The main goal of this system is to completely resolve the problems of the students (residing in hostels) related to any hostel appliances or anything in the hostel.

2. Project Requirements

2.1 Functional

1. This application will be available for both android and IOS ecosystems. Anyone using an iPhone or an android device will be able to download it from Apple AppStore and Google Play respectively.

2. Anyone having this application on his/her phone would be able to access the services from anywhere in the hostels.

3. On every complaint there will run a counter that will update the caretaker as well as the student that how much days have passed since the complaint is being registered.

4. Our system will ensure that only genuine complaints are being registered by making the students to first login through their thapar email-id's and their passwords.

5. The system will provide the ease to students as if there is no action against their complaint , then system will automatically send report to the warden.

6. The application will support three login ids namely: Login as caretaker , Login as the student and Login as Warden.

7. The app will also have two step verification if the student wants additional security for his ID and also an OTP will be sent to the registered email address if the student forgets his password.

2.2 Non-Functional

1. The system shall have 2 step verification if the student wants additional security of his account/ID.

2. The system shall be linked to user's email address so that he may be contacted as and when required.

3. The student shall be able to change his password of the account with the help of email address provided.

4. The application's interface will be very simple and easy to use so that students does not find any difficulty in using it..

5. The application shall not take more than 30 megabytes of memory.

6. The application's backend and frontend shall be designed in such a way that it uses minimal RAM in the user's device.

3. Structured Analysis

3.1 Data Flow Diagrams

3.1.1 Level 0 DFD

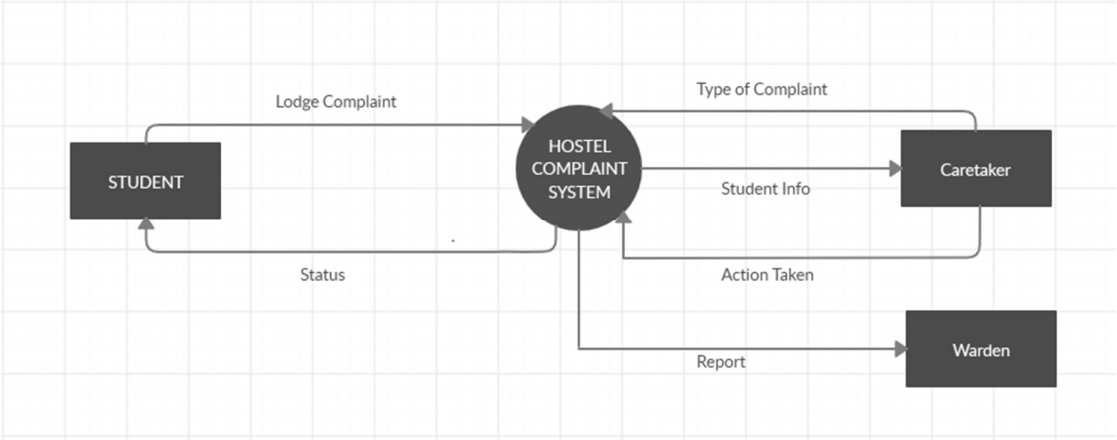

Figure 1. Level 0 DFD

3.1.2 Level 1 DFD

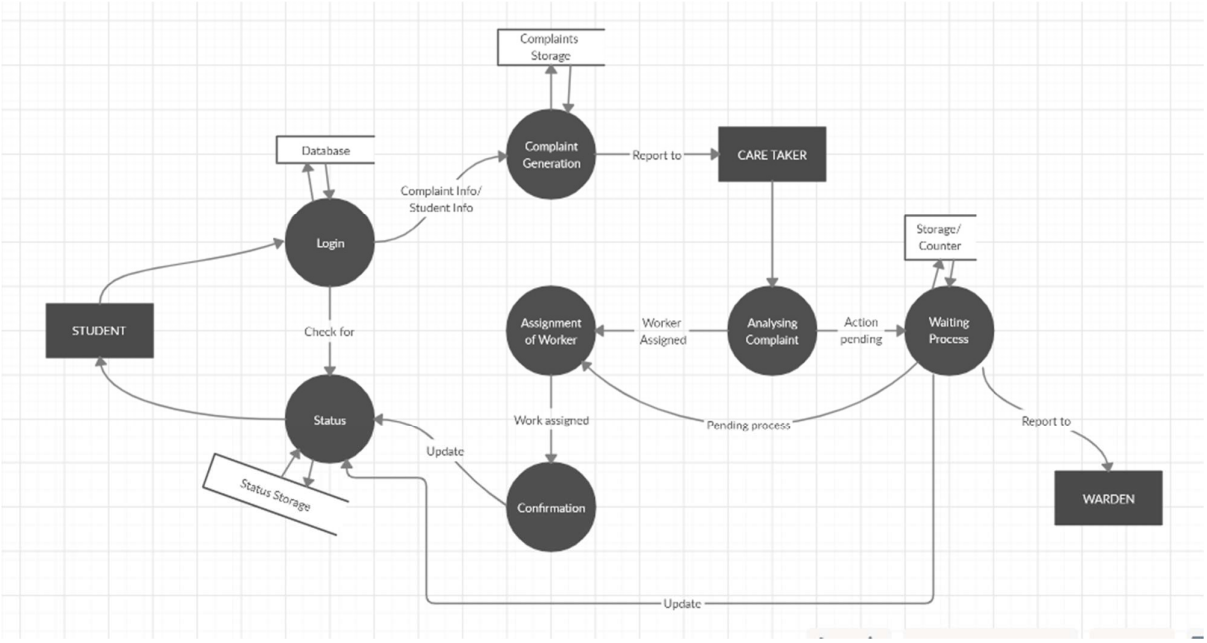

Figure 2. Level 1 DFD

3.1.3 Data Dictionary

Stores :

1. **Name of the Data Store:** Student Details

   - Description: Stores the details of the student such as Room Number, Name, Email id etc.

2. **Name of the Data Store:** Complaint Details

   - Description: Stores the type of complaint such as electrical complaints, carpenter complaints etc.

Data Processes :

1. **Name of the Process:** Complaint Generation

   - Inward Dataflow: Complaint Info/Student Info ,Store .
   - Outward Dataflow: Report to Caretaker , Retrieve .
   - Description:
     - Input –

       Complaint Info :

       Information regarding the complaint will be forwarded to the caretaker.

       Store :

       Data will be stored to the database .
     - Output –

       Report To :

       - The information of the complaint will be forwarded to the caretaker and he further analyze the complaint .

       Retrieve :

       A request will be retrieved from the database .

2. **Name of the Process:** Analyzing Complaint

- Inward Dataflow: Action taken
- Outward Dataflow: Action pending ,Worker assigned
- Description:
    - Input – Action taken
    - Output – Assignment of Workers
        - The caretaker after receiving the complaint will assign the worker for the complaint . For example if the complaint is to change the tube light of the room then the caretaker will assign a electrician for the work .
        - In this way he will assign a particular worker .
        - In the same time the data of the pending process will be stored .

3. **Name of the Process:** Waiting Process

- Inward Dataflow : Action pending , Retrieve
- Outward Dataflow: Store ,Report to Warden
- Description:
    - Input –
    - Output – Invoices, statements
        - This process will store the counter variable . After specific value of the counter variable a report will be sent to the warden highlighting that the complaint is not resolved in the specified time.
        - After that the particular worker will also be assigned regarding the complaint .

Data Flows :

1. **Name of DF:** Student info

   - Starting Point: Student
   - Ending Point: Complaint generation
   - Description: Sends the details of the student along with the complaint.

2. **Name of DF:** Worker assigned

   - Starting Point: Analyzing complaints
   - Ending Point: Assignment of worker
   - Description: Worker assigned according to type of complaint

3. **Name of DF:** Action pending

   - Starting Point: Analyzing complaints
   - Ending Point: Waiting process
   - Description: Describes that the complaint is in pending and in progress state.

4. **Name of DF:** Update

   - Starting Point: Confirmation
   - Ending Point: Status
   - Description: Updates the current status of the complaint.

3.2 ER Diagram

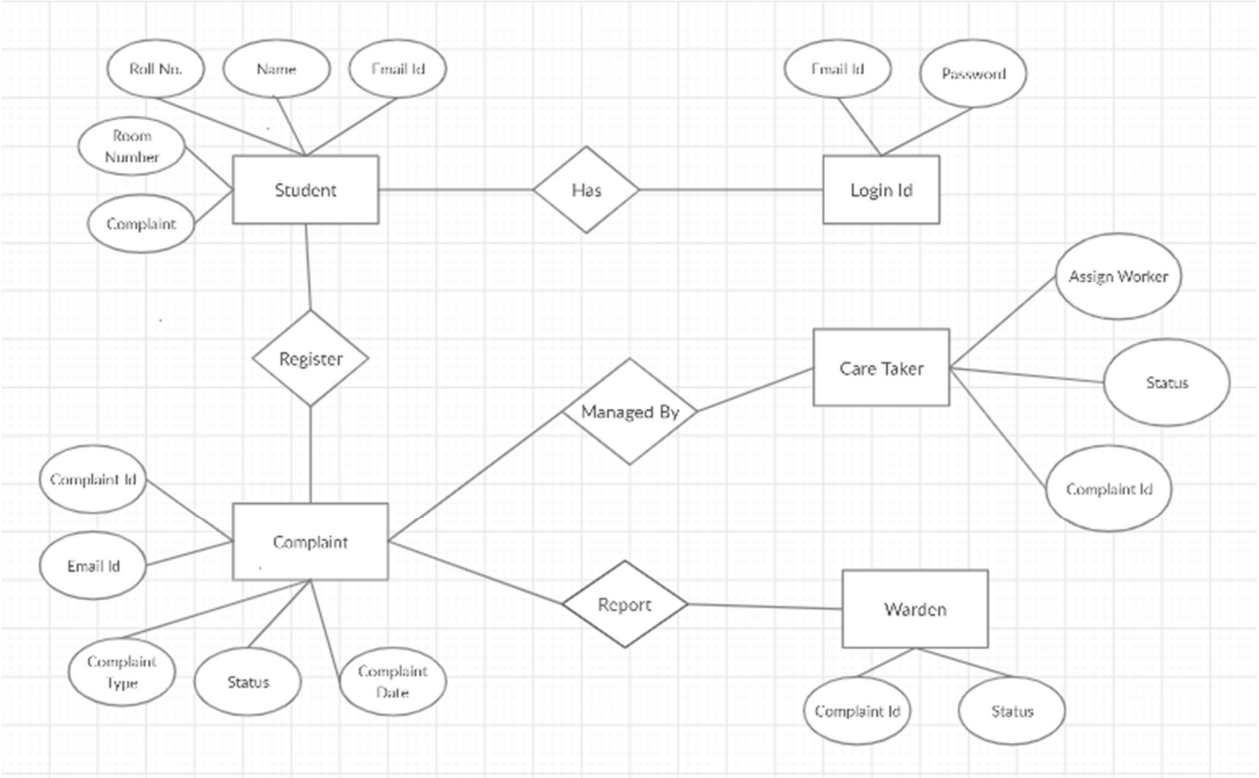

Figure 3. ER Diagram

4. Object Oriented Analysis

4.1 Use Case Diagram

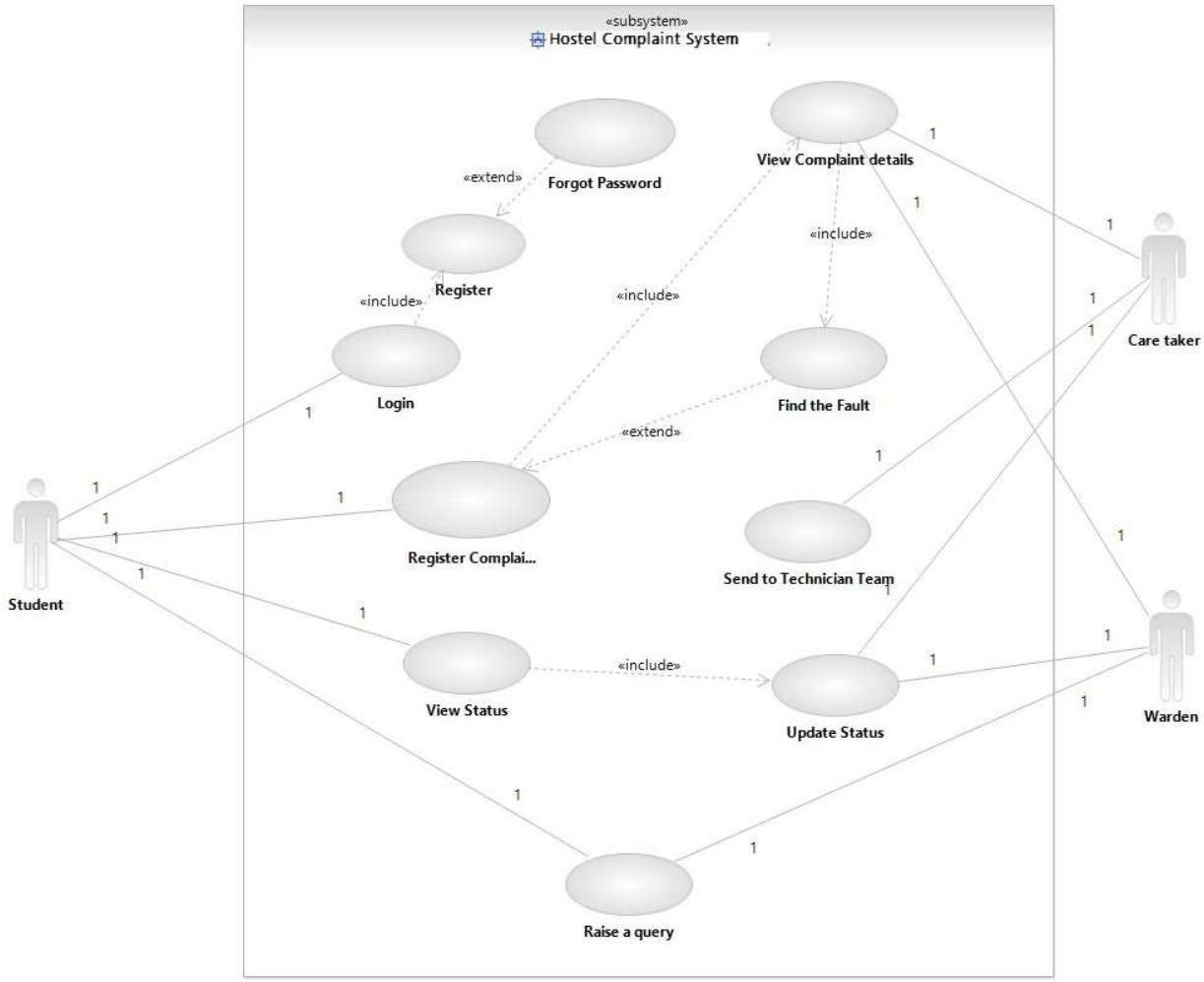

Figure 4. Use Case Diagram

4.2 Use Case Template

Use Case Name	Register a Complaint
Primary Actor	Student
Secondary Actor	Care Taker, Warden
Pre-Conditions	Register (Student should already have a account), Login (Student should be logged in)
Main Scenario	1. Student register a complaint. 2. Caretaker analyzes the complaint. 3. Counter maintains the number of days
Post Conditions	1. Caretaker assigns the worker. 2. Report is send to warden after counter passes the limit. 3. Student gives the feedback.
Error Condition	1. Counter does not work correctly. 2. Warden does not receive report due to some error.

4.3 Use Case Scenario

1. (AA) Student logins into the system and register complaint. It can also view the status of the complaint registered.
2. (SR) Complaint gets saved into the database.
3. (AA) Caretaker analyses the complaint and assign the worker for that complaint.
4. (SA) After the complaint is solved the status is changed from pending to solved.
5. (AA) Warden can see all the complaints which are solved and which are pending.
6. (SA) Complaints get highlighted when they are pending after a specific time.
7. (AA) Student also send feedback which can be seen by warden and caretaker.

4.4 Activity Diagram

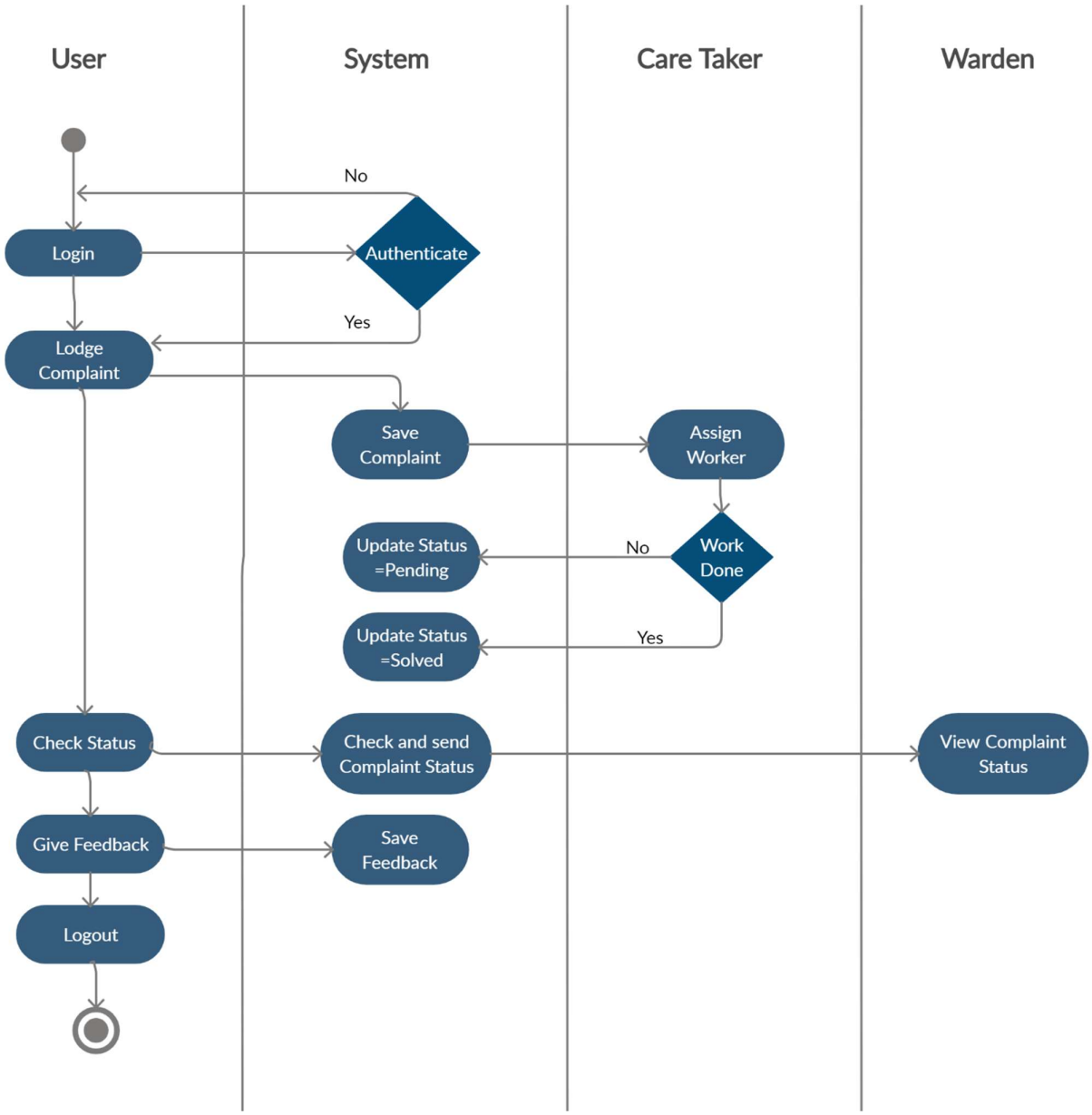

Figure 5. Activity Diagram

4.5 Class Diagram

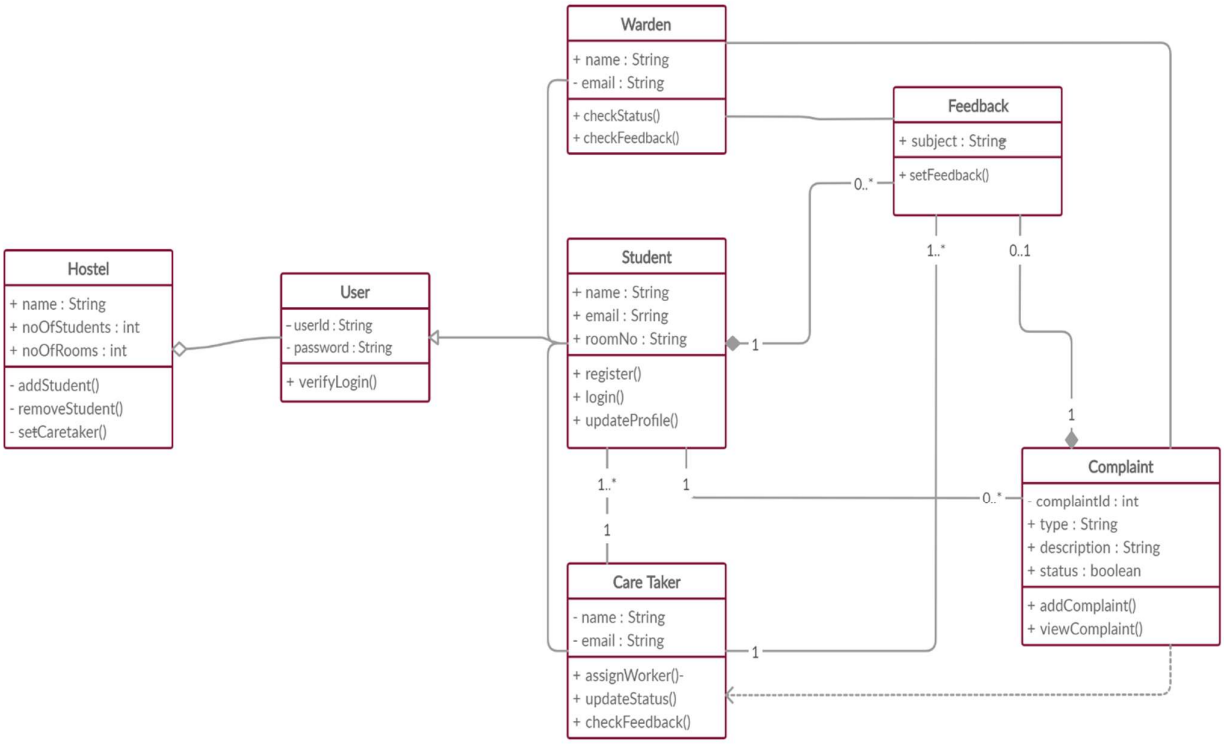

Figure 6. Class Diagram

4.6 State Diagram

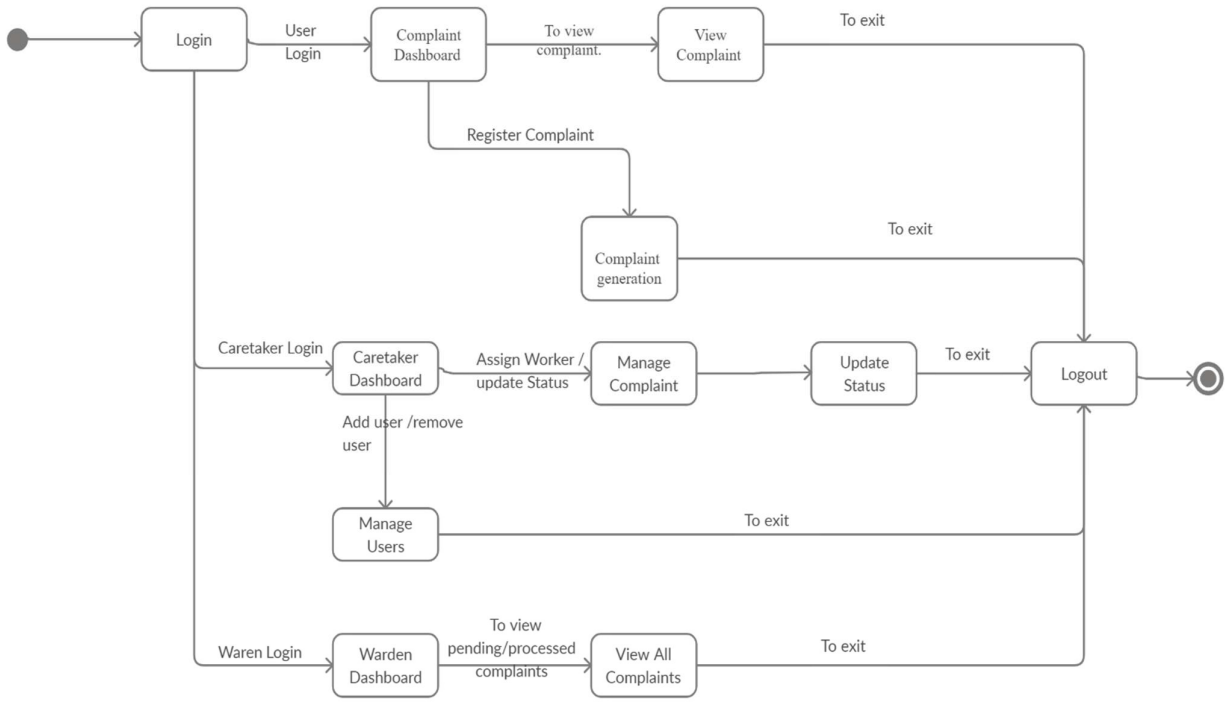

Figure 7. State Diagram

4.7 Sequence Diagram

4.7.1 Sequence Diagram of Student

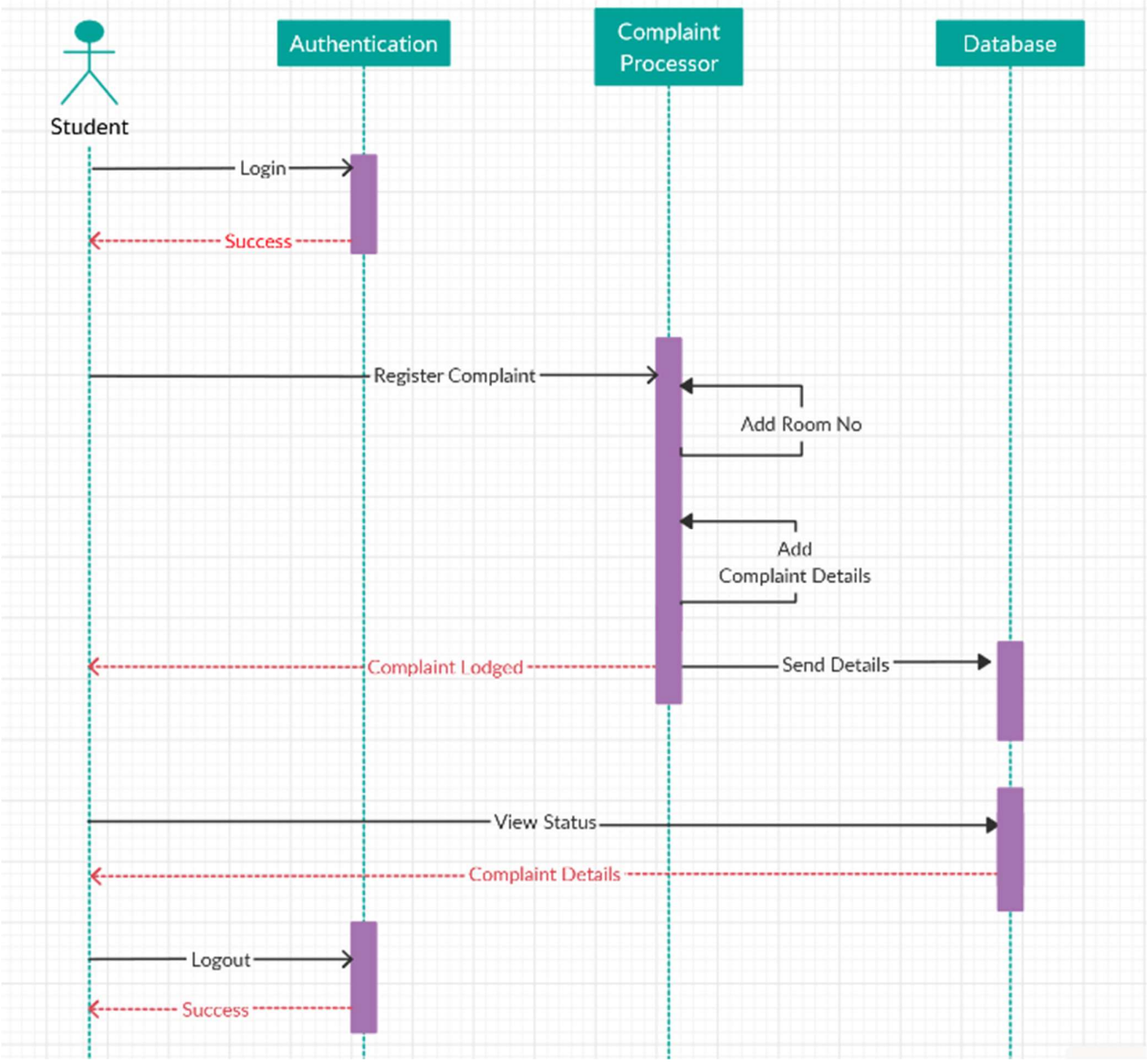

Figure 8. Sequence Diagram of Student

4.7.2 Sequence Diagram of Care Taker

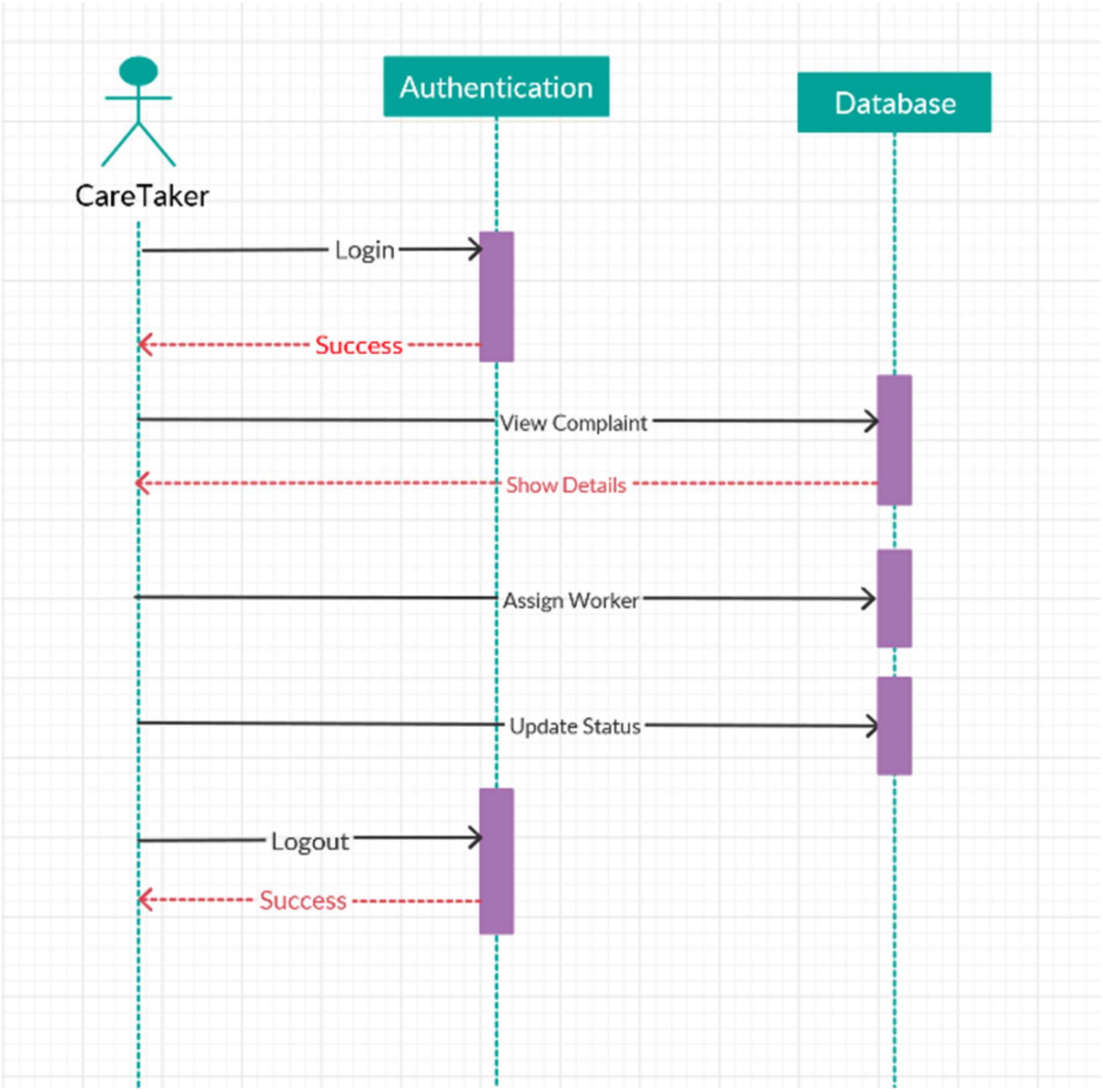

Figure 9. Sequence Diagram of Care Taker

4.8 Component Diagram

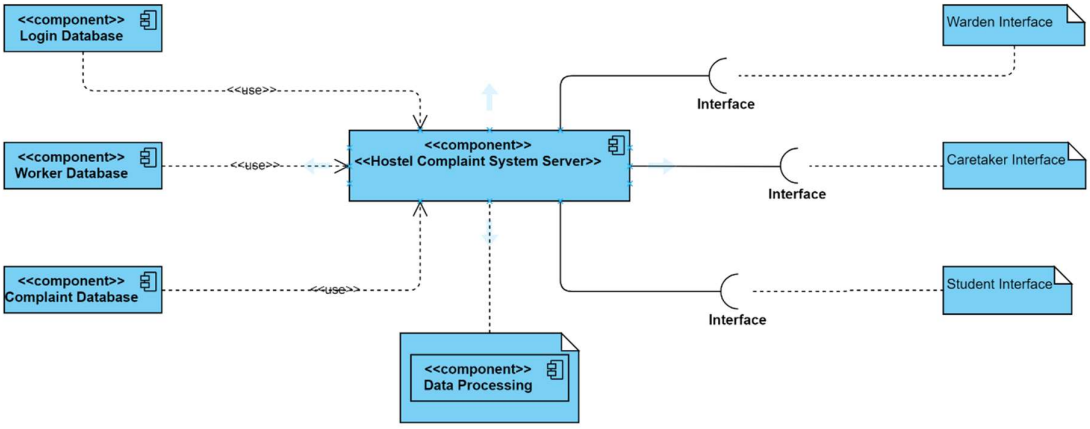

Figure 10. Component Diagram

4.9 Deployment Diagram

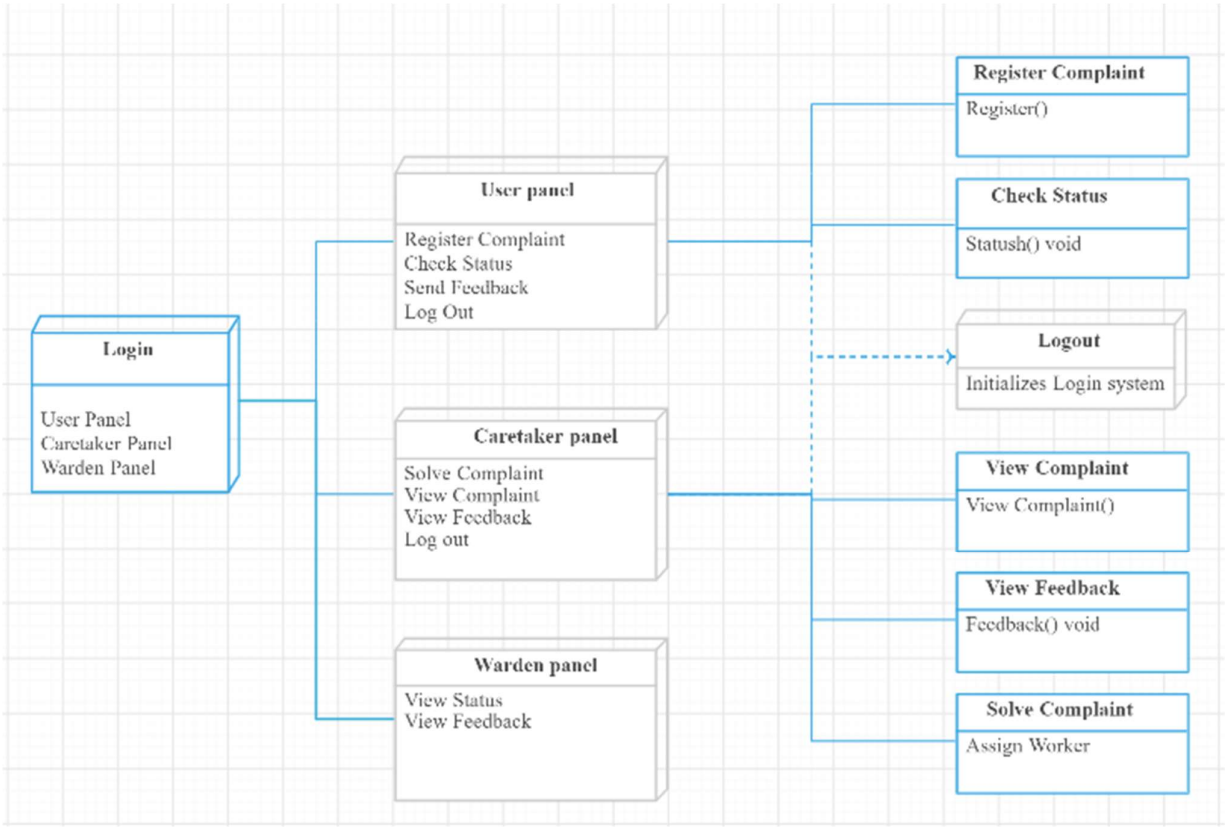

Figure 11. Deployment Diagram

4.10 Collaboration Diagram

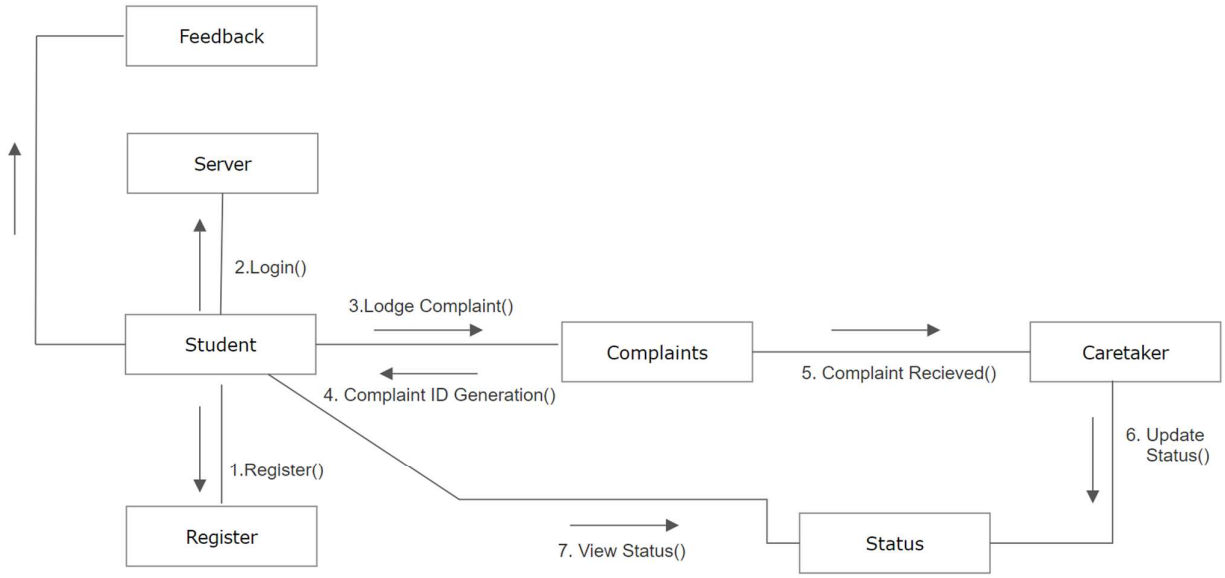

Figure 12. Collaboration Diagram

5. Testing

5.1 Test Case Report

Test Case #: 1.1

Test Case Name: Change Password

System: CMS

Subsystem: Password

Designed by: Paras , Lokesh

Design Date: 8/11/2019

Executed by: Lovish, Lavish

Short Description: Test the CMS Change Password

Pre-conditions:
The user has a valid account - The user has accessed the account by successful authentication.
The current password is lovish .
The system displays the main menu

Step	Action	Expected System Response	Pass/ Fail
1	Click the Change Password button	The system displays a form to enter the details.	Pass
2	Enter Current Password	The system asks the user to enter the correct current password.	Pass
3	Enter New Password	The system asks the user to enter the new Password.	Pass
4	Enter Confirm Password	The system asks the user to enter the same New Password.	Pass
5	Click Submit button	The system displays the confirmation message	Pass
	Check post Condition 1		

6	Repeat steps 1,2,3,4 using another say ‘paras’	The system displays the message to enter the correct Current password.	Pass
	Check post-condition 2		
7	Repeat steps 2,3,4 using new password say ‘lovishjindal’	The system asks the user to confirm (re-enter) the confirm password.	Pass
8	Enter a wrong confirmation (say ‘jindal’)	The system displays a message of **unsuccessful** operation and asks the user to confirm the correct password.	Pass
9	Re-enter ‘lovishjindal’	The system displays a message of successful operation.	Pass

Post-conditions:

1. The current is ‘lovish’ stored in the database.
2. The new password ‘lovishjindal’ stored in the database.

5.2 Screenshots

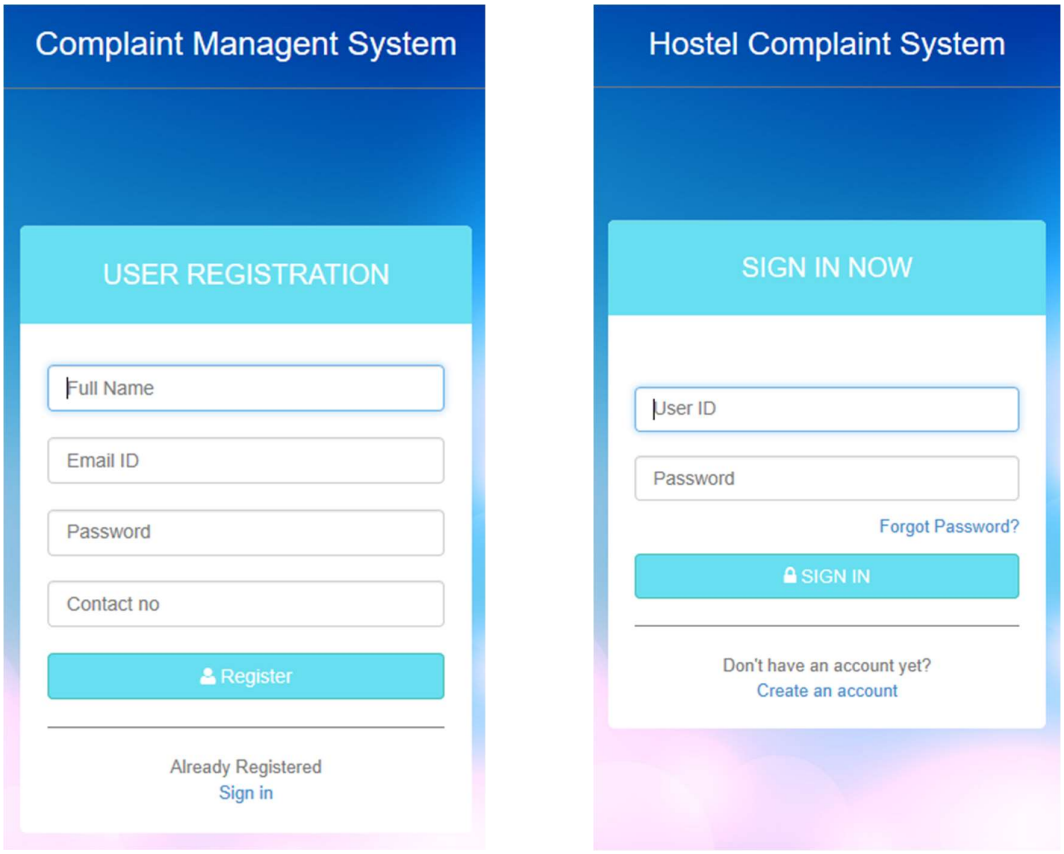

Figure13. Screenshots of Registration and Sign In

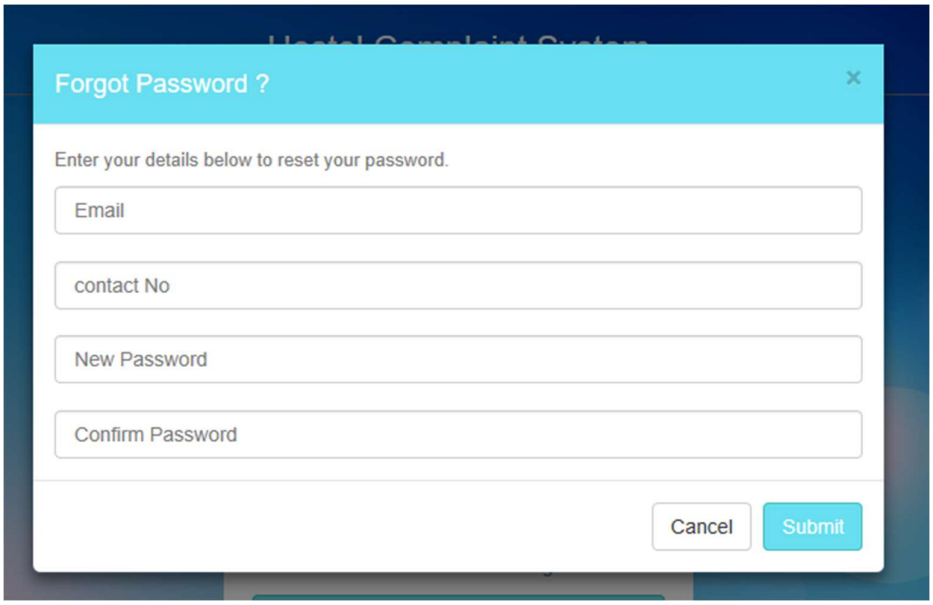

Figure14. Screenshot of Forgot Password

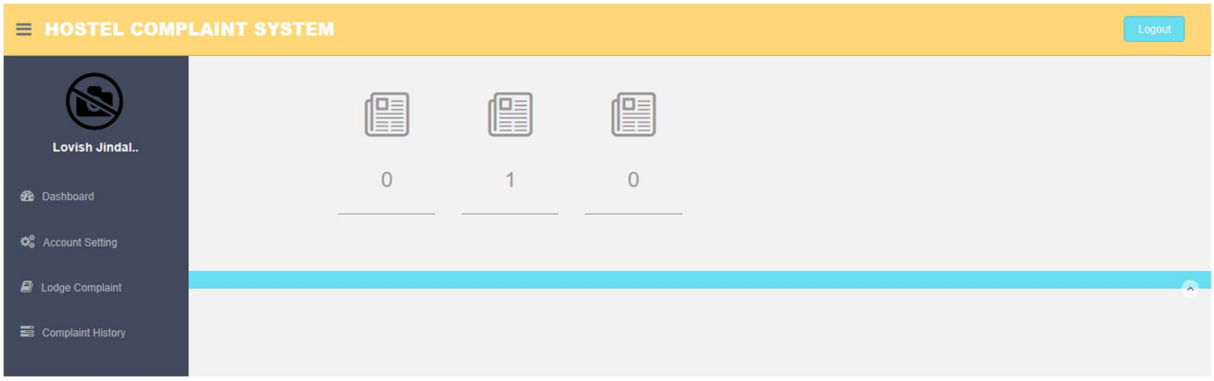

Figure15. Screenshot of Student Dashboard

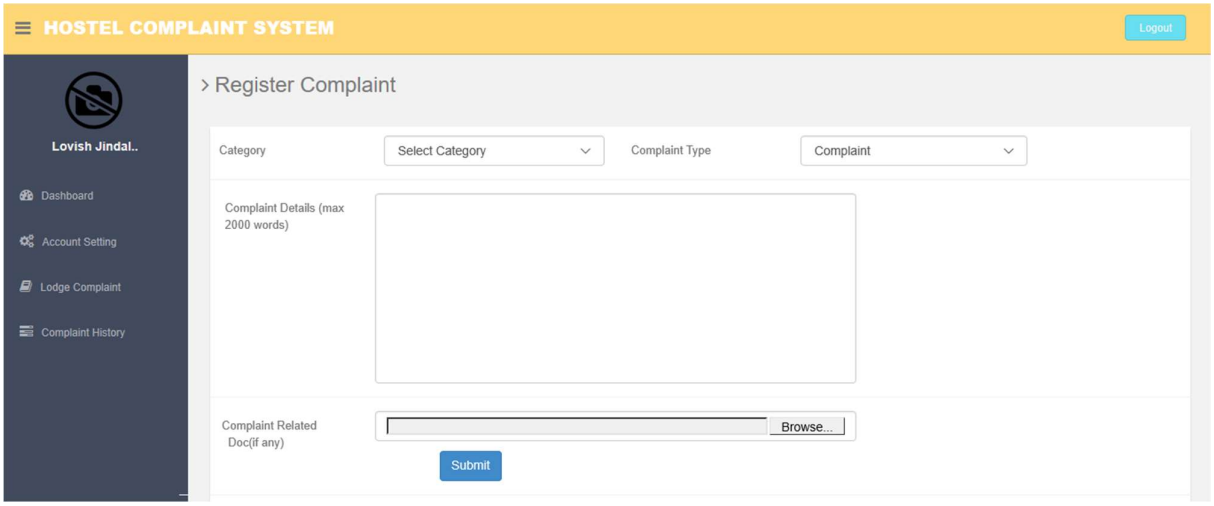

Figure16. Screenshot of Lodge Complaint

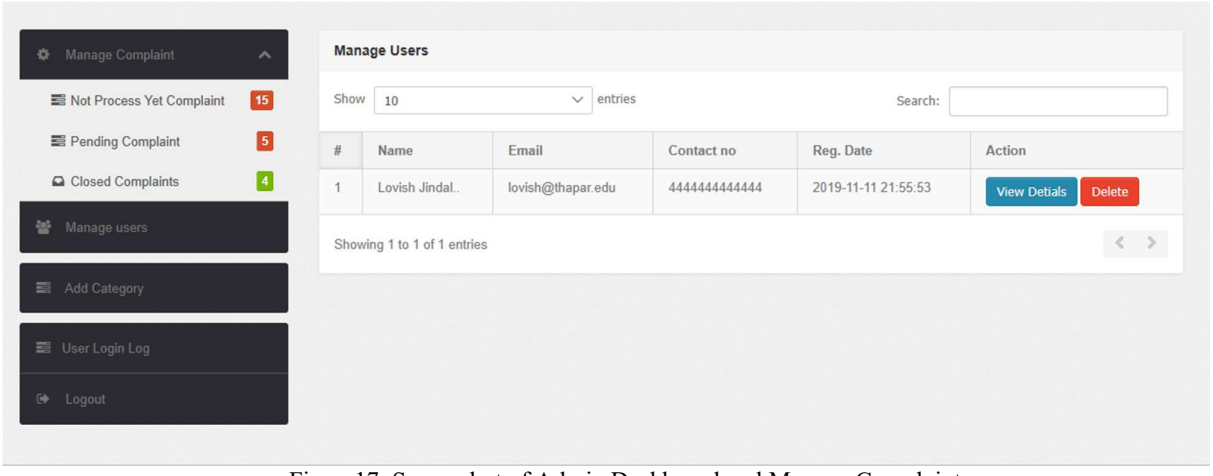

Figure17. Screenshot of Admin Dashboard and Manage Complaints