

B.M.S. COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU



OOMD Mini Project Report

College Complaint Management System

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

Chadive Muralidhar Reddy
1BM23CS072

Hemanth Kumar R
1BM23CS110

Eashan Jain V
1BM23CS098

Dhanush S
1BM23CS089

Department of Computer Science and Engineering
B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
2025-2026

B.M.S. COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



DECLARATION

We, **CHADIVE MURALIDHAR REDDY (1BM23CS072), HEMANTH KUMAR R(1BM23CS110), EASHAN JAIN V(1BM23CS098) and DHANUSH S(1BM23CS089)** students of 5th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this OOMD Mini Project entitled "**College Complaint Management System**" has been carried out in Department of CSE, B.M.S. College of Engineering, Bangalore during the academic semester August - December 2025. I also declare that to the best of our knowledge and belief, the OOMD mini Project report is not from part of any other report by any other students.

Signature of the Candidate

CHADIVE MURALIDHAR REDDY (1BM23CS072)

HEMANTH KUMAR R (1BM23CS110)

EASHAN JAIN V (1BM23CS098)

DHANUSH S (1BM23CS089)

B.M.S. COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING



CERTIFICATE

This is to certify that the OOMD Mini Project titled “**College Complaint Management System**” has been carried out by **CHADIVE MURALIDHAR REDDY (1BM23CS072)**, **HEMANTH KUMAR R(1BM23CS110)**, **EASHAN JAIN V(1BM23CS098)** and **DHANUSH S (1BM23CS089)** during the academic year 2025-2026.

Signature of the Faculty in Charge (Dr. Roopashree S)

Table of Contents

Sl No	Title	Page No
1	Ch 1: Problem statement	4
2	Ch 2: Software Requirement Specification	5
3	Ch 3: Class Diagram	13
4	Ch 4: State Diagram	17
5	Ch 5: Interaction diagram	23
6	Ch 6: UI Design with Screenshots	32
7	Ch 7: Conclusion	38

Chapter 1: Problem Statement

1.1 Introduction

Most colleges still handle student complaints through manual processes such as paper forms, physical visits to departments, or informal messaging. These outdated methods cause delays, lack of transparency, and no proper tracking of complaint progress. Students often do not know whether their complaint is received, being processed, or ignored.

Similarly, departments struggle to manage and prioritize complaints due to the absence of a centralized system. This leads to miscommunication, unresolved issues, and decreased student satisfaction.

To overcome these limitations, a digital **College Complaint Management System** is proposed to allow students to submit complaints online and track updates in real time, while departments can manage and resolve complaints efficiently.

1.2 Problem Statement

Colleges currently lack a unified, structured, and transparent mechanism for complaint reporting and resolution. Students face difficulties in registering grievances, and authorities have no proper system to monitor complaint status, assign tasks, or track resolution time.

This results in delays, unaddressed issues, and no accountability. A system is required that allows students to submit complaints easily, enables departments to manage them efficiently, and provides administrators with visibility into the entire process.

1.3 Motivation

A smooth and transparent complaint management process is essential for maintaining a positive academic environment. With increasing student population and campus activities, manual processes are no longer scalable.

A digital platform ensures faster resolution, proper documentation, accountability, and improved communication between students and authorities.

The proposed system aims to modernize grievance handling, reduce delays, improve satisfaction, and support better decision-making through proper records and analytics.

Chapter 2: Software Requirement Specification

2.1 Introduction

2.1.1 Purpose of this Document

This SRS defines the functional and non-functional requirements for the **College Complaint Management System (CCMS)**. It serves as the authoritative reference for designers, developers, testers, project managers, and college stakeholders who will build, validate, and operate the system. The document clarifies system features, constraints, interfaces, and success criteria.

2.1.2 Scope of this Document

This SRS covers requirements for a web and mobile-enabled complaint management platform that allows students to submit grievances, administrators to manage and assign complaints, and departmental officers to resolve them. The system supports tracking, notifications, reporting and basic analytics. It targets small-to-medium colleges and must be suitable for low-resource administrative environments.

1.3 Overview

The CCMS is a lightweight, role-based application that replaces manual complaint registers and ad-hoc communications with a centralized, auditable workflow. The system provides role-specific portals (Student, Admin, Department Officer), an admin dashboard for oversight and reporting, and basic mobile-friendly interfaces for students to file and track complaints.

2.2 General Description

2.2.1 Product Perspective

CCMS is a standalone system that may integrate later with the college's authentication (LDAP/SSO), SMS gateway, and email services. Initially it will use a local user store with role-based access control and optional single sign-on integration points.

2.2.2 Product Features (high-level)

- Student complaint submission with category, attachments and priority
- Complaint tracking and history view for students
- Admin dashboard: view, filter, assign, escalate, close complaints
- Department portal: view assigned complaints, update progress, mark resolved
- Notifications: Email and SMS (or in-app) on status changes
- Simple reporting: counts by category, SLA breaches, average resolution time
- Role-based security and audit trail

2.2.3 User Classes and Characteristics

- **Student:** Files complaints, views status, provides feedback. Basic computer/smartphone user.
- **Admin (Registrar/Helpdesk):** Reviews new complaints, assigns to departments, oversees workflows.
- **Department Officer / Faculty:** Receives assigned complaints, updates status, resolves.
- **System Administrator:** Manages users, roles, system settings, and backups.
- **Auditor/Management:** Views reports and analytics.

2.2.4 Operating Environment

- **Server:** Linux-based web server or cloud VM (Node/Java/Python backend)
- **Database:** Relational (Postgres/MySQL) or managed DB
- **Clients:** Modern browsers (Chrome/Firefox/Safari) and Android/iOS mobile browsers or simple hybrid app
- **Integration:** Optional SMS/email gateways (Twilio/MSG91/SMTP)

2.2.5 Design and Implementation Constraints

- Minimal dependencies to ensure easy deployment on low-cost hosting.
- App must support low-bandwidth and older mobile devices (responsive, minimal assets).
- Attachment upload size limits (e.g., 5 MB per file).
- Role-based access controls required.
- Audit trail for all status changes and assignments.

2.2.6 Assumptions and Dependencies

- Students provide valid credentials (college ID/email).
- Departments will actively use the system for assigned tasks.
- SMS/email gateway costs and privacy requirements are managed by the college.
- Internet connectivity available for essential functions; low-bandwidth mode provided.

2.3. Specific Requirements

2.3.1 Functional Requirements (FR)

FR1 — Authentication & Authorization

- System shall provide secure login for Students, Admins, Department Officers, and System Admin.
- Role-based permissions:
Students (create/view), Admin (view/assign/manage), Officers (view/update/resolve), System Admin (manage roles).
- Optional SSO/LDAP integration point.

FR2 — Complaint Registration

- Students shall create a complaint with: title, category (Hostel/Academics/Fees/Maintenance/Transport/Other), description, attachments (images/doc), priority (Low/Medium/High), optional location (room/department).
- System shall assign unique Complaint ID and timestamp.

FR3 — Complaint Tracking & Status

- The system shall maintain and display status: Draft → Submitted → Under Review → Assigned → In Progress → Resolved → Closed → Reopened.
- Students shall be able to view their complaint history and current status, including comments and timestamps.

FR4 — Admin Dashboard & Assignment

- Admin shall view all complaints with filters (date, category, priority, status).
- Admin shall assign complaints to departments or specific officers and set expected SLA/resolution time.

FR5 — Department Officer Actions

- Assigned officers shall receive tasks, add remarks, upload action evidence, change statuses (In Progress, Resolved), and request additional information.

FR6 — Notifications

- System shall notify users on critical events via email/SMS/in-app: complaint submitted, assigned, status changed, resolved, closed.
- Notification templates configurable by admin.

FR7 — Commenting & Communication

- Students, Admins, and Officers shall post comments on complaints.
- Comments recorded in timeline with author and timestamp.

FR8 — Escalation & SLA Management

- Admin shall configure SLA per category.
- If unresolved past SLA, complaint escalates to higher authority and triggers a notification.

FR9 — Reporting & Analytics

- Admin shall generate reports: total complaints (time range), complaints by category, average resolution time, pending complaints, officer performance, SLA breaches.
- Exportable as CSV/PDF.

FR10 — Feedback & Closure

- After closure, students shall provide optional satisfaction rating and comments.

FR11 — Audit & Logging

- All actions (status change, assignment, login attempts) shall be logged for auditing.

FR12 — Data Retention & Export

- Admin shall archive/restore complaint records and export reports.
- Retention policy configurable.

2.3.2 External Interface Requirements

User Interface (Mobile & Web)

- Clean responsive UI; mobile-first design.
- Student flow: Login → Dashboard → New Complaint → Attach file → Submit → View status.
- Admin flow: Login → Dashboard (filters) → View complaint → Assign/Escalate → Close → Reports.
- Department flow: Login → Assigned complaints → Update → Resolve.

API Interface

- RESTful APIs for complaint operations, user management, reporting.
- Endpoints secured with token-based auth (JWT/OAuth2).
- Webhook endpoints for SMS/email gateway integrations.

Hardware Interfaces

- Standard file storage for attachments; optional integration with institutional file servers or cloud storage.

Communication Interfaces

- SMTP or third-party email service.
- SMS gateway integration.
- Optional push notifications.

2.3.3 Performance Requirements

- Page/API responses ≤ 2 seconds under normal load; ≤ 5 seconds under peak.
- Support at least 5,000 concurrent users.
- Data optimized for low bandwidth.
- Uptime target: 99.5%.
- Attachments compressed

2.3.4 Security Requirements

- Use HTTPS/TLS for all communication.
- Passwords salted & hashed.
- Role-based access control.
- Rate-limiting & account lockout.
- Data privacy compliance.
- Regular backups and secure storage of attachments.

2.3.5 Reliability & Availability

- Queue messages if dependent services fail.
- Recoverable with backups and replication.

2.3.6 Maintainability & Extensibility

- Modular architecture.
- Well-documented APIs and DB schema.
- Config-driven SLAs, categories, notification templates.

2.4 Interface Requirements

2.4.1 User Interface Requirements

- Dashboard widgets: pending complaints, SLA breaches, new complaints.
- Form validation.
- Accessibility support.
- Low-bandwidth mode: text-only list view.

2.4.2 Admin Dashboard Requirements

- Filters: date, category, status, priority, officer.
- Bulk actions: assign multiple complaints, export list.
- Notification logs.

2.4.3 Reporting Interface

- Pre-built and custom reports.
- Export to CSV/PDF.
- Scheduled report emails.

2.5 Design Constraints

- Minimal third-party dependencies.
- Storage cap per file/account.
- No offline support for MVP.
- Deployable on a single VM.

2.6 Non-Functional Attributes

Performance

- Efficient indexing.
- Caching frequently used data.

Security

- Regular patching.
- Encrypted backups.

Usability

- Critical actions within 2 taps.
- Simple wording.

Scalability

- Modular monolith or microservice-ready architecture.

Portability

- Should run on Linux servers and be deployable as Docker containers.

2.7 Preliminary Schedule & Budget

Schedule (7 weeks):

- Requirements & Design: 1 week
- Backend API & DB: 2 weeks
- Frontend (student + admin): 2 weeks
- Integration & Testing: 1 week
- Deployment & Pilot: 1 week

Budget (Estimated for MVP):

- Development & testing: ₹30k–₹60k
- Hosting: ₹5k–₹15k
- SMS/Email/API fees: ₹5k–₹20k
- Others: variable

2.8 Data Retention & Privacy

- Default retention: 3 years.
- Users can request deletion of personal data.
- Minimal data collection.
- Explicit consent for notifications.

2.9 Glossary

- SLA – Service Level Agreement

- Complaint – Reported issue
- Admin – Manages complaints
- Officer – Resolves complaints
- MVP – Minimum Viable Product

2.10 Appendix — Data Model Summary

- Users(userId, name, email, role, passwordHash, deptId, createdAt)
- Complaints(compId, title, description, category, priority, studentId, status, assignedTo, createdAt, updatedAt)
- Comments(commentId, compId, authorId, text, createdAt)
- Attachments(attId, compId, filename, path, uploadedBy, uploadedAt)
- SLA Logs

Chapter 3: Class Modelling

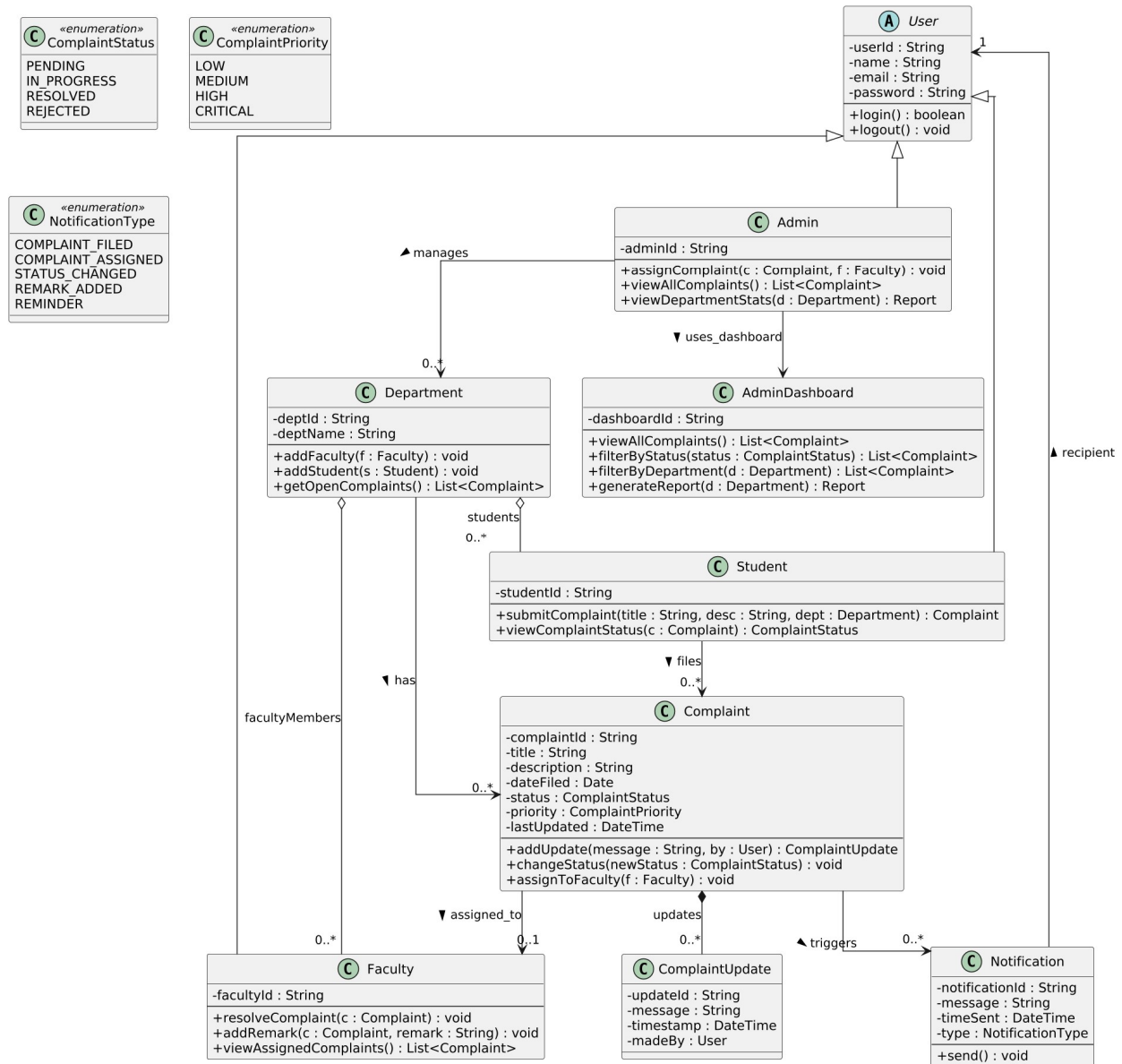


Figure 3.0 Class Diagram

1. Student Class

The Student class represents college students who raise complaints through the system. A student can file a complaint, track its status, and view updates added by faculty or admins.

Responsibilities

File new complaints

Track complaint progress

View complaint updates

Maintain personal academic details

2. Faculty Class

The Faculty class represents faculty members or officers responsible for handling and resolving complaints assigned to them by the admin.

Responsibilities

View assigned complaints

Resolve complaints

Add remarks or updates

Communicate progress to students

3. Admin Class

The Admin class represents the system administrator who manages the entire complaint lifecycle. They act as the controller between students and faculty.

Responsibilities

View all complaints

Assign complaints to faculty

Monitor system activity

Manage faculty and complaint distribution

4. Complaint Class

The Complaint class represents an issue submitted by a student. It contains all essential information such as title, description, date, and current status.

Responsibilities

Store complaint details

Hold current status

Maintain list of updates

Support modifications during the lifecycle

5. ComplaintUpdate Class

The ComplaintUpdate class tracks every update added to a complaint, allowing complete visibility of complaint progress.

Responsibilities

Store update message

Store timestamp of change

Maintain history of actions

Support transparency for students and faculty

6. Department Class

The Department class represents academic departments in the college. Each department contains multiple students and faculty members.

Responsibilities

Store department details

Link students to departments

Link faculty to departments

7. Notification Class

The Notification class manages system-generated alerts sent to students, faculty, or admins whenever a major event occurs (e.g., assignment, remark, resolution).

Responsibilities

Store notification message

Capture the time of sending

Notify respective recipients

Ensure communication inside the system

8. ComplaintStatus Enum

This enum represents the various stages a complaint can be in, ensuring standardization across the system.

Possible States

PENDING – When the complaint is freshly submitted

IN_PROGRESS – When faculty starts working on it

RESOLVED – When the complaint is marked as resolved

REJECTED – When admin rejects invalid complaint

Chapter 4: State Modeling

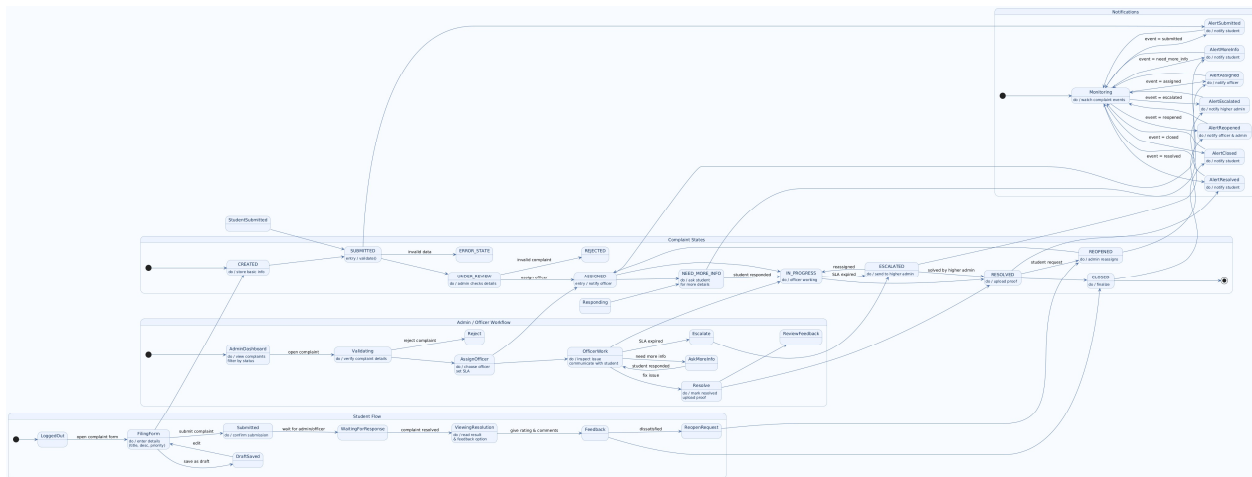


Figure 4.0 State Diagram

4.1 Student Flow States

i. LoggedOut

The student is not logged into the system. They only have access to basic features like viewing FAQs or opening the login page.

ii. FilingForm

The student begins filling out a complaint. They enter details such as title, description, category, department, and priority.

iii. DraftSaved

The student saves the complaint as a draft. This allows them to edit or complete the complaint later before submitting.

iv. Submitted

The student successfully submits the complaint. It is now forwarded to the admin for initial review.

v. WaitingForResponse

The student waits for updates from the admin or officer. No changes can be made at this stage.

vi. ViewingResolution

The complaint has been resolved, and the student opens the resolution details to see the officer's remarks or attachments.

vii. Feedback

The student provides a satisfaction rating and additional comments after viewing the resolution.

viii. ReopenRequest

The student requests the complaint to be reopened if the issue was not resolved satisfactorily.

4.2 Student Flow Events

i. open complaint form

The student initiates the complaint creation process.

ii. save as draft

Triggered when the student chooses to save incomplete complaint details.

iii. submit complaint

The user submits the complaint, moving it to the Submitted state.

iv. complaint resolved

Notification arrives from officer/admin indicating resolution.

v. give feedback

The student provides their final feedback.

vi. dissatisfied / request reopen

Student chooses to reopen the complaint due to dissatisfaction.

4.3 Complaint States (Core Life-Cycle)

i. CREATED

Initial state after the student begins the complaint process. Basic details are stored.

ii. SUBMITTED

The complaint is registered in the system and awaits admin review.

iii. UNDER REVIEW

Admin verifies details, checks evidence, and determines validity.

iv. ASSIGNED

The complaint is assigned to a specific faculty member or officer. SLA timers begin.

v. NEED MORE INFO

The officer requests extra details, images, or clarifications from the student.

vi. IN PROGRESS

The officer is actively working on resolving the complaint.

vii. ESCALATED

The complaint crosses the SLA time limit or officer fails to resolve, so it is escalated to a higher authority.

viii. RESOLVED

The officer successfully resolves the complaint and uploads proof.

ix. CLOSED

Admin verifies resolution and formally closes the complaint.

x. REOPENED

The student requests to reopen the complaint, restarting investigation.

xi. REJECTED

Admin rejects the complaint due to invalid details or insufficient information.

xii. ERROR STATE

Occurs if complaint submission or validation fails due to missing fields, invalid data, or technical issues.

4.4 Complaint Events

i. validate()

Admin verifies accuracy and completeness of complaint details.

ii. assign officer

Complaint is assigned to responsible faculty.

iii. student responded

Student provides required info when officer asks for clarification.

iv. SLA expired

Triggers escalation to higher authority.

v. solved by higher admin

Complaint resolved after escalation.

vi. student request

Student requests complaint reopening.

vii. finalize

The admin closes the complaint.

4.5 Admin/Officer Workflow States

i. AdminDashboard

Admin views all complaints and filters them by category and status.

ii. Validating

Admin checks complaint content for correctness and legitimacy.

iii. AssignOfficer

Admin selects an officer/faculty and assigns the complaint.

iv. OfficerWork

Officer investigates the issue, communicates with the student, and performs necessary actions.

v. AskMoreInfo

Officer requests additional information from the student.

vi. Resolve

Officer marks the complaint as resolved and uploads proof or reports.

vii. ReviewFeedback

Admin reviews the student's feedback after resolution.

viii. Reject

Admin rejects invalid complaints.

4.6 Admin/Officer Events

i. open complaint

Admin opens the complaint for validation.

ii. reject complaint

Admin rejects invalid submissions.

iii. set SLA / choose officer

Admin configures SLA and assigns complaint.

iv. need more info

Officer asks student for additional data.

v. escalate

Officer escalates due to SLA breach or inability to resolve.

vi. upload proof

Officer submits evidence for resolution.

4.7 Notification States

i. Monitoring

Notification system monitors complaint transitions.

ii. AlertSubmitted

Triggered when a complaint is submitted.

iii. AlertAssigned

Alert sent to the officer once assigned.

iv. AlertMoreInfo

System triggers an alert asking student to provide more details.

v. AlertResolved

Notification sent when officer resolves the complaint.

vi. AlertClosed

Admin has closed the complaint; final notification is sent.

vii. AlertReopened

Complaint reopened → Notify officer/admin.

viii. AlertEscalated

Complaint SLA expired, higher admin notified.

4.8 Notification Events

i. event = submitted

Complaint submitted → Notify student.

ii. event = assigned

Complaint assigned → Notify officer.

iii. event = need_more_info

Officer needs info → Notify student.

iv. event = resolved

Officer resolved → Notify student.

v. event = closed

Final closure notification.

vi. event = reopened

Student reopened → Notify team.

vii. event = escalated

Escalation alert to higher authority.

Chapter 5: Interaction Modeling

5.1 USE CASE DIAGRAM

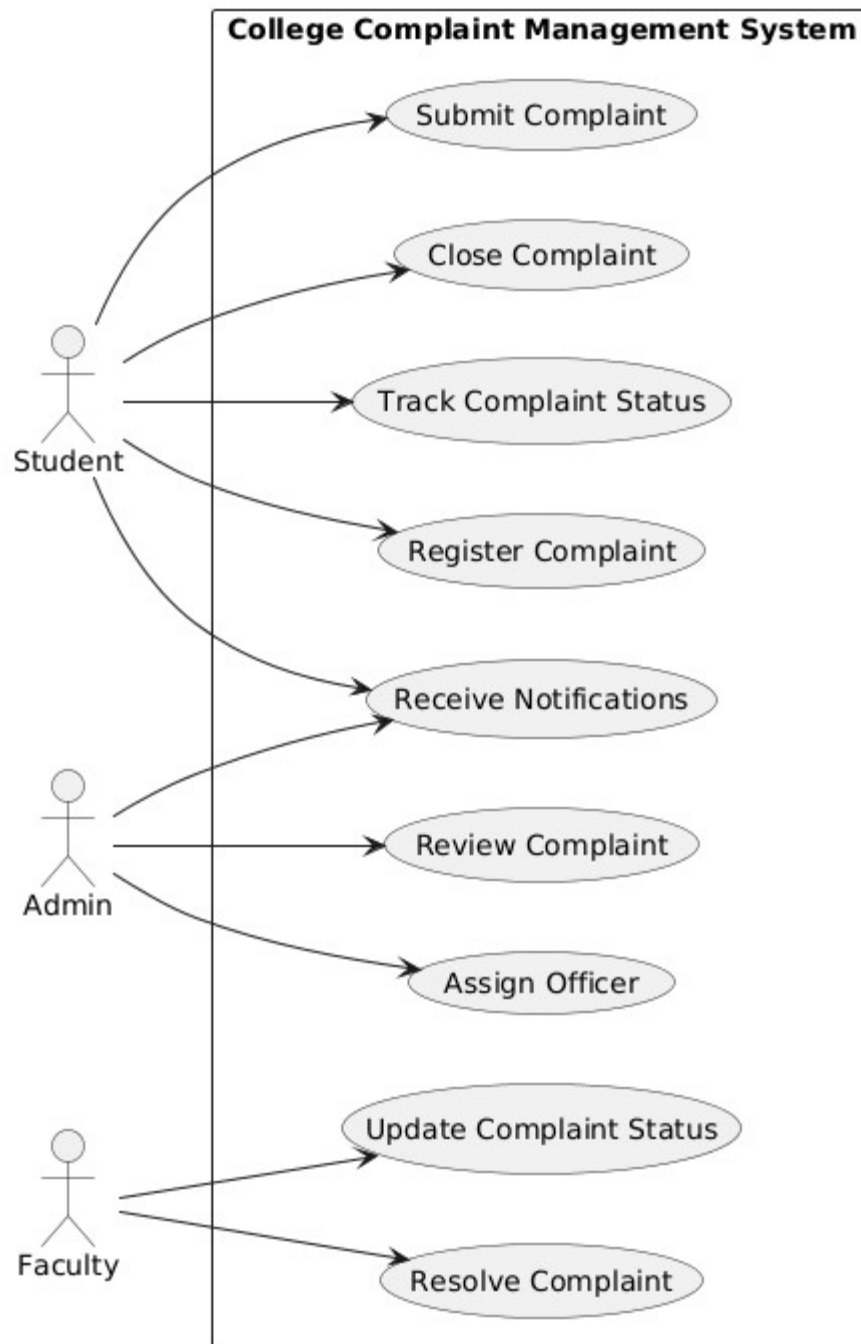


Figure 5.0 Use Case Diagram

5.1.1 Actors and Their Relevance

i. Student

The Student is the primary user of the system. Students raise complaints related to academics, hostel, fees, maintenance, transport, or any college service. They also track the complaint status, receive notifications, and provide feedback after resolution. This actor is essential because the entire purpose of the system is to simplify and digitalize the complaint-handling process for students.

ii. Admin

The Admin represents the college authorities or grievance redressal cell staff. They have complete control over incoming complaints — reviewing them, assigning them to the correct department or officer, closing them, or managing escalations. This actor ensures the system is functioning properly and maintains accountability in resolving issues.

iii. Officer / Department Staff

This actor belongs to a specific department (Hostel, Academics, Transport, IT, etc.). Officers are responsible for resolving the complaints assigned to them by the Admin. They update complaint status, add remarks, and upload evidence of resolution. They are crucial for bridging real-world issue handling with the digital complaint system.

iv. Notification Service

This represents the automated system that sends alerts to Students, Admins, and Officers. Notifications include complaint submission, status change, requests for more information, resolution, closure, and escalations. It ensures users stay updated without manually checking the portal.

5.1.2 Use Case Relevance

i. Register Complaint

This allows students to create a new complaint by entering details such as title, description, category, department, priority, and attachments. This is the core function of the system because it digitizes the problem-reporting process.

ii. Submit Complaint

After entering details, the student submits the complaint.
This action forwards the complaint to the Admin for review.
It marks the official beginning of the complaint lifecycle.

iii. Track Complaint Status

Students can check the live status of their complaints (Submitted, Under Review, Assigned, In Progress, Resolved, Closed).
This increases transparency and removes the need for manual follow-ups.

iv. Receive Notifications

Students and officers receive alerts at key stages, including assignment, remarks added, resolution, or if additional information is needed.
Notifications improve communication and reduce delays.

v. Review Complaint (Admin)

Admins verify the complaint details and determine the next action.
They can accept, reject, or escalate the complaint for clarification.

vi. Assign Officer (Admin)

Admins assign complaints to the correct department or officer.
This ensures the complaint is handled by the right authority.
The system records assignment time and calculates SLA.

vii. Resolve Complaint (Officer)

Officers perform the required action to fix the issue.
They update the status, add remarks, and upload proof (image/document).
This step progresses the complaint towards closure.

viii. Update Complaint Status (Officer)

Officers modify the complaint's progress as the investigation continues (In Progress, Need More Information, Resolved).
This helps maintain accountability and properly document the resolution workflow.

ix. Close Complaint (Admin)

Once the officer resolves an issue, the admin reviews it and closes the complaint officially.
Admins ensure that the resolution meets institutional standards.

5.2 SEQUENCE DIAGRAM

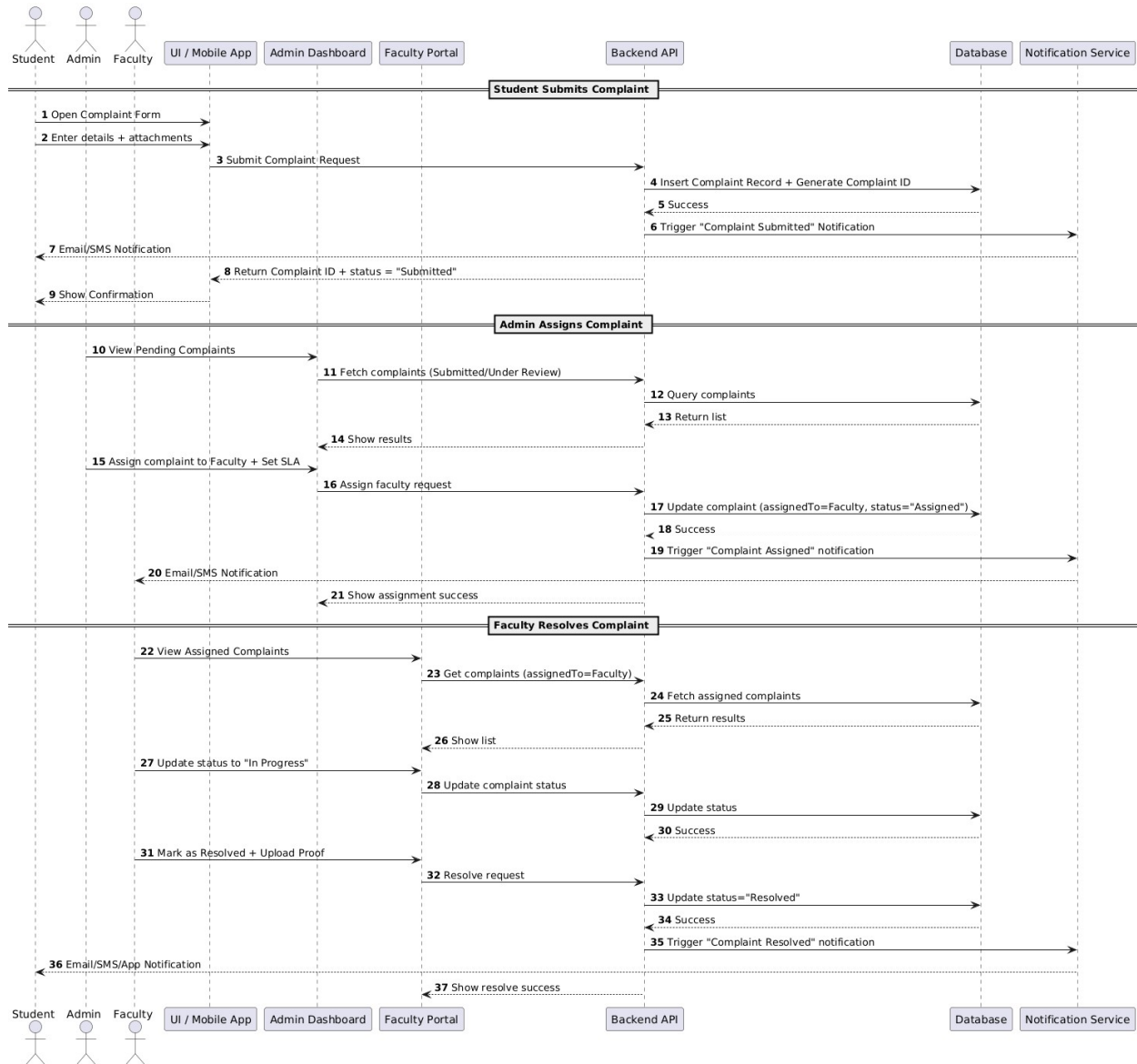


Figure 5.2 Sequence Diagram

5.2.1 Lifeline Descriptions

i. Student

The Student lifeline represents the primary user of the complaint management system. The student initiates the sequence by opening the complaint form, entering complaint details, attaching evidence, and submitting the request. This actor triggers the entire complaint lifecycle and interacts only through the UI/Mobile App. The Student also receives system-generated notifications once the complaint is submitted successfully.

ii.Mobile App:

The UI/Mobile App lifeline represents the front-end interface used by students. It collects user input such as complaint title, category, description, and attachments. The mobile app is responsible for packaging these inputs and sending them to the Backend API as a structured complaint submission request. After receiving a response, the app displays confirmation messages, complaint ID, and status to the student. It also handles incoming notifications and updates the UI accordingly.

iii.Backend Server:

The Backend API lifeline acts as the central processing unit for the entire system. It receives complaint submission requests from the mobile app and performs validation, storage coordination, and service orchestration. The API forwards the complaint data to the database and triggers notifications after successful insertion. Finally, it returns the generated Complaint ID and initial status ("Submitted") back to the mobile app for user confirmation.

iv.Database:

The Database lifeline represents the data storage layer of the system. It stores all complaint-related information, including student ID, title, category, description, attachments, timestamps, and generated Complaint ID. In this sequence, the database inserts a new complaint record and returns a success acknowledgment to the Backend API. It is a critical component, ensuring reliable persistence of all complaint details.

V.Notification Service

The Notification Service lifeline represents the automated messaging system responsible for sending alerts to students and staff. Upon receiving a trigger from the Backend API, it sends an email or SMS confirming that the complaint was successfully submitted. This lifeline ensures timely communication and enhances transparency in the complaint-handling

process.

Its role continues throughout later stages of the complaint lifecycle (e.g., assigned, in progress, resolved).

5.2.2 Interactions

- The **Student lifeline** initiates the entire process by opening the complaint form within the mobile application. This action prompts the app to load the interface for entering complaint details.
- The **UI/Mobile App lifeline** records the student's inputs, including the complaint title, category, priority, description, and attachments. Once all required fields are filled, the app prepares a structured "Submit Complaint" request.
- The **Mobile App** sends this complaint submission request to the **Backend API**, which acts as the central controller responsible for validating and processing the data.
- Upon receiving the request, the **Backend API lifeline** forwards the complaint details to the **Database** so that a new complaint record can be stored. The database generates a unique Complaint ID and stores all associated metadata.
- The **Database lifeline** returns a success acknowledgment along with the generated Complaint ID to the Backend API, indicating that the complaint has been successfully recorded.
- After receiving confirmation from the database, the **Backend API** triggers the **Notification Service** to send an alert to the student. This notification confirms that the complaint has been submitted and logged in the system.
- The **Notification Service lifeline** sends an SMS or email informing the student that the complaint has been successfully submitted and assigned a unique tracking ID.
- Once the notification is processed, the **Backend API** returns the Complaint ID and initial status ("Submitted") back to the Mobile App.
- The **Mobile App lifeline** displays a confirmation message to the Student, including the Complaint ID and submission status. This marks the completion of the complaint submission interaction.
- Finally, the **Student lifeline** receives visual confirmation on the screen, signaling that the complaint has entered the workflow and will now proceed to review by admin staff.

5.3 ACTIVITY DIAGRAM

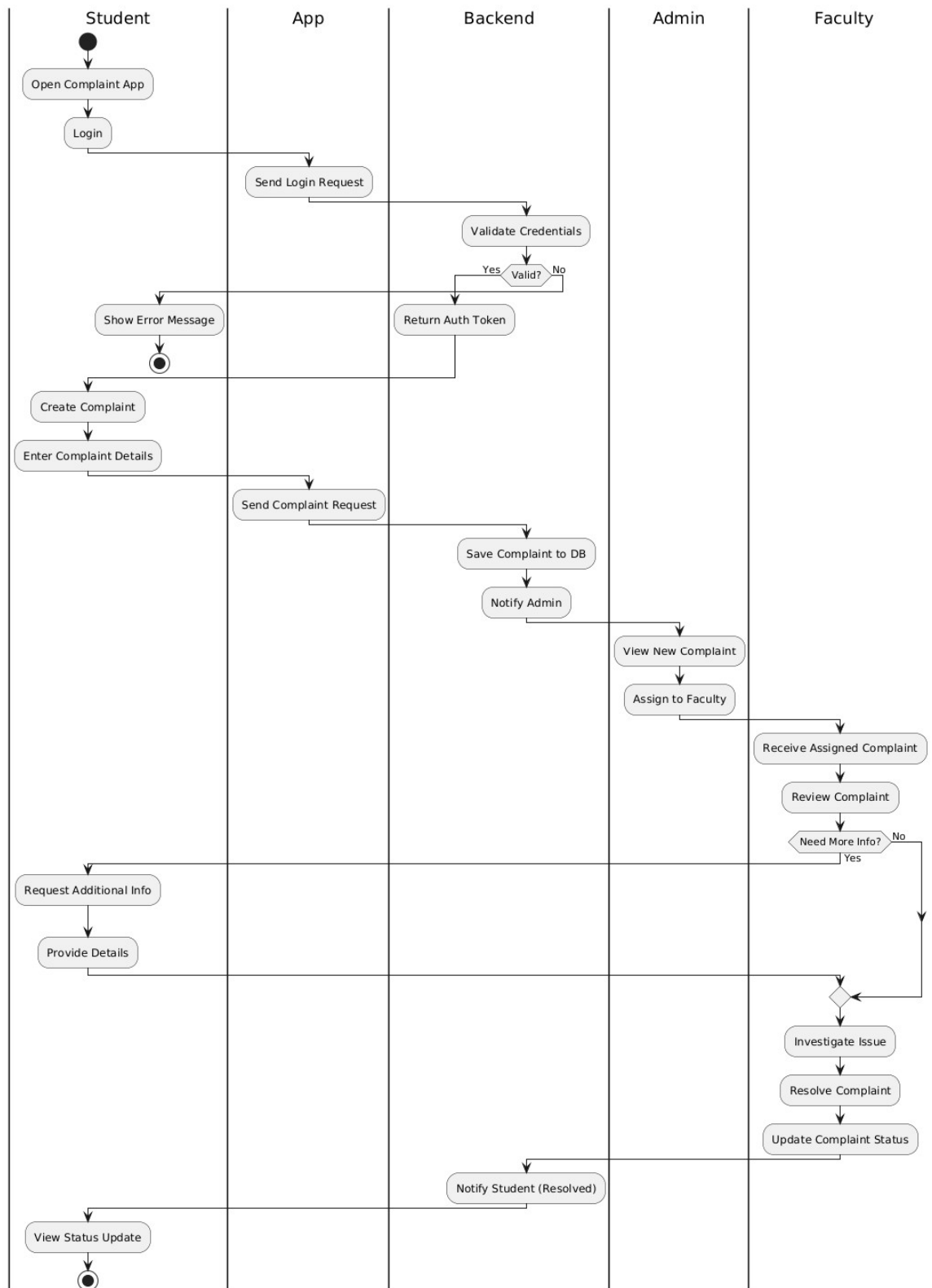


Figure 5.2 Activity Diagram using Swimlanes

Open Complaint App

The “Open Complaint App” activity represents the starting point where the student launches the mobile application to access the complaint management features. This initializes the interface and prepares the app for login or complaint submission.

Login

The login activity allows the student to authenticate themselves before accessing the system. This ensures that only authorized users can submit and track complaints.

Send Login Request

The mobile app sends the entered login credentials to the backend server for verification. This enables secure authentication.

Validate Credentials

The backend checks whether the entered credentials are valid. The system verifies them against stored user data to confirm identity.

Valid / Invalid Decision

This decision determines whether the authentication was successful. If valid, the user continues to the next step. If invalid, the system redirects the student to an error screen.

Show Error Message

If authentication fails, the system displays an error message informing the student that the login attempt was unsuccessful and prompting them to try again.

Create Complaint

After logging in successfully, the student initiates the complaint submission process by selecting the option to create a new complaint.

Enter Complaint Details

The student enters details such as title, description, category, department, urgency level, and any attachments. This step captures complete information about the issue.

Send Complaint Request

Once details are entered, the mobile app sends the complaint data to the backend as a submission request.

Save Complaint to Database

The backend server stores the complaint in the database. A unique complaint ID is generated, and the complaint status is set to “Submitted.”

Notify Admin

After storing the complaint, the backend triggers a notification to the admin, informing them that a new complaint is waiting for review.

View New Complaint (Admin)

The admin logs in and views newly submitted complaints. They check the provided details and evaluate the complaint.

Assign to Department Officer

The admin forwards the complaint to the appropriate department officer for resolution. This step updates the complaint status to “Assigned.”

Receive Assigned Complaint (Officer)

The department officer receives the assigned complaint in their task list. They open and examine the details.

Review Complaint

The officer reviews the complaint thoroughly, checks the provided evidence, and prepares to investigate the matter.

Need More Info Decision

The officer evaluates whether additional information is required to proceed. If more clarity is needed, the officer requests more details from the student.

Request Additional Info

The officer sends a request to the student asking for additional documents, explanations, or clarification.

Provide Details (Student)

The student responds to the officer’s request by submitting additional information through the app. Investigate Issue

The officer proceeds to investigate the complaint on-site or through communication with relevant departments.

Resolve Complaint

After verifying and addressing the issue, the officer resolves the complaint and prepares the resolution summary or proof.

Update Complaint Status

The officer updates the system with the resolution details, marking the complaint as “Resolved.”

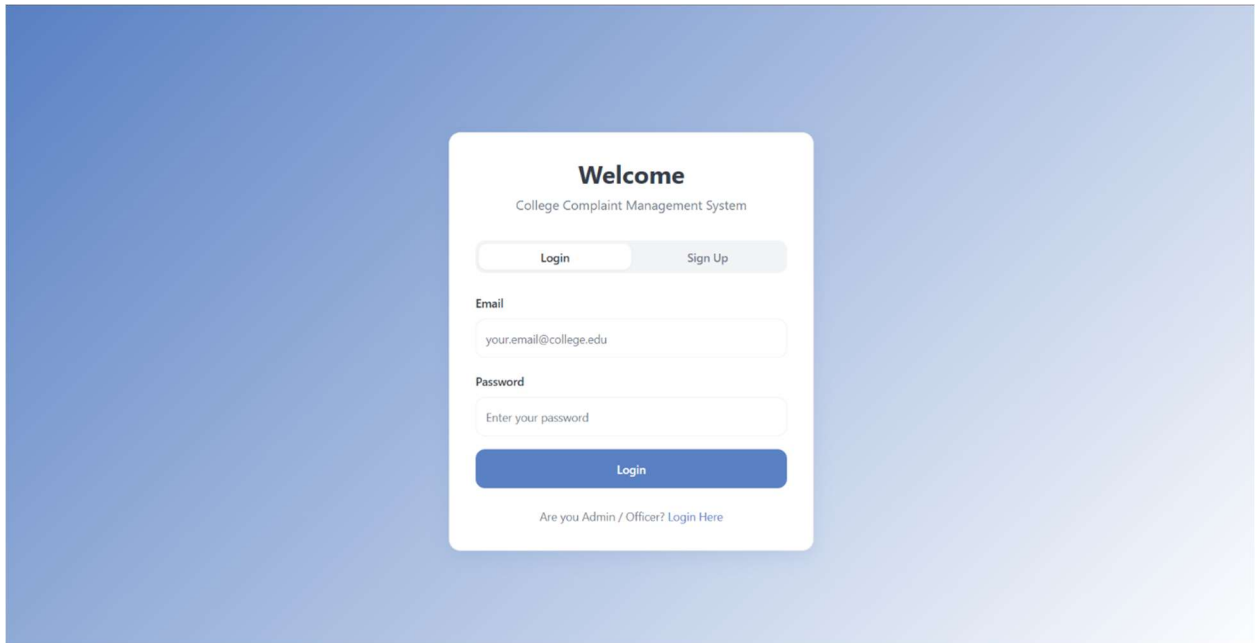
Notify Student (Resolved)

The backend system automatically notifies the student that their complaint has been resolved.

View Status Update (Student)

The student reviews the updated status and resolution details. This marks the completion of the complaint process unless the student wishes to reopen it.

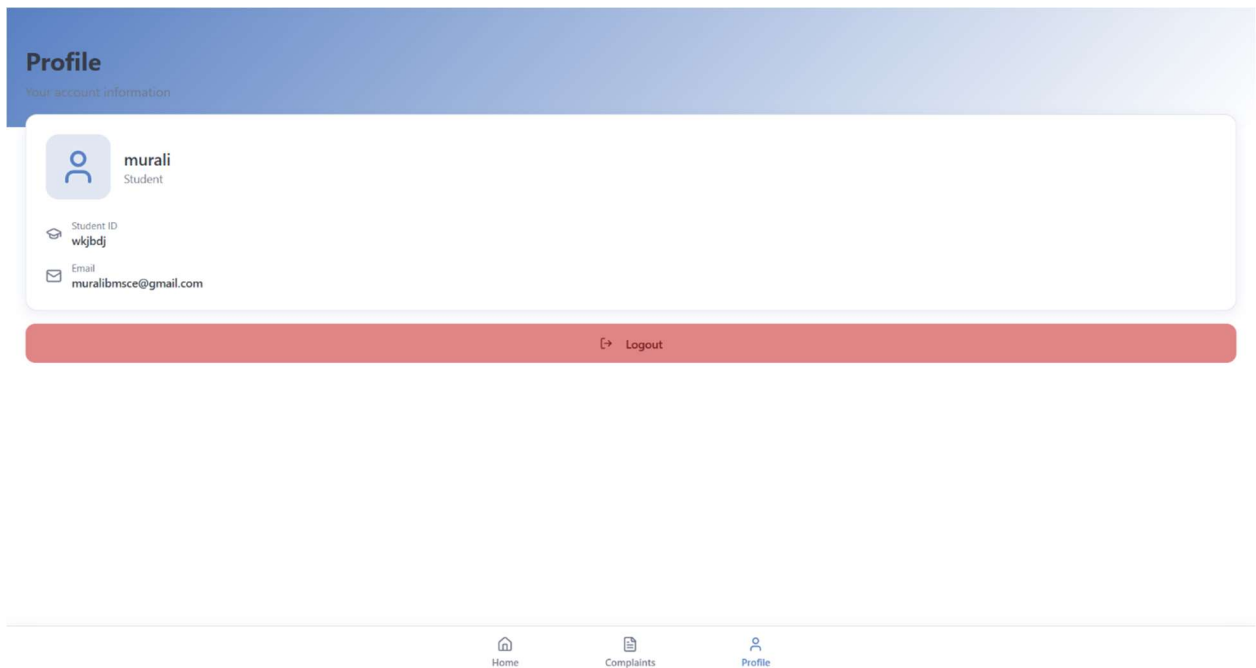
Chapter 6: UI Design with Screenshots



As shown in Figure 6.0, the Student Login Page allows users to securely access the system using their credentials.

i. Login Page

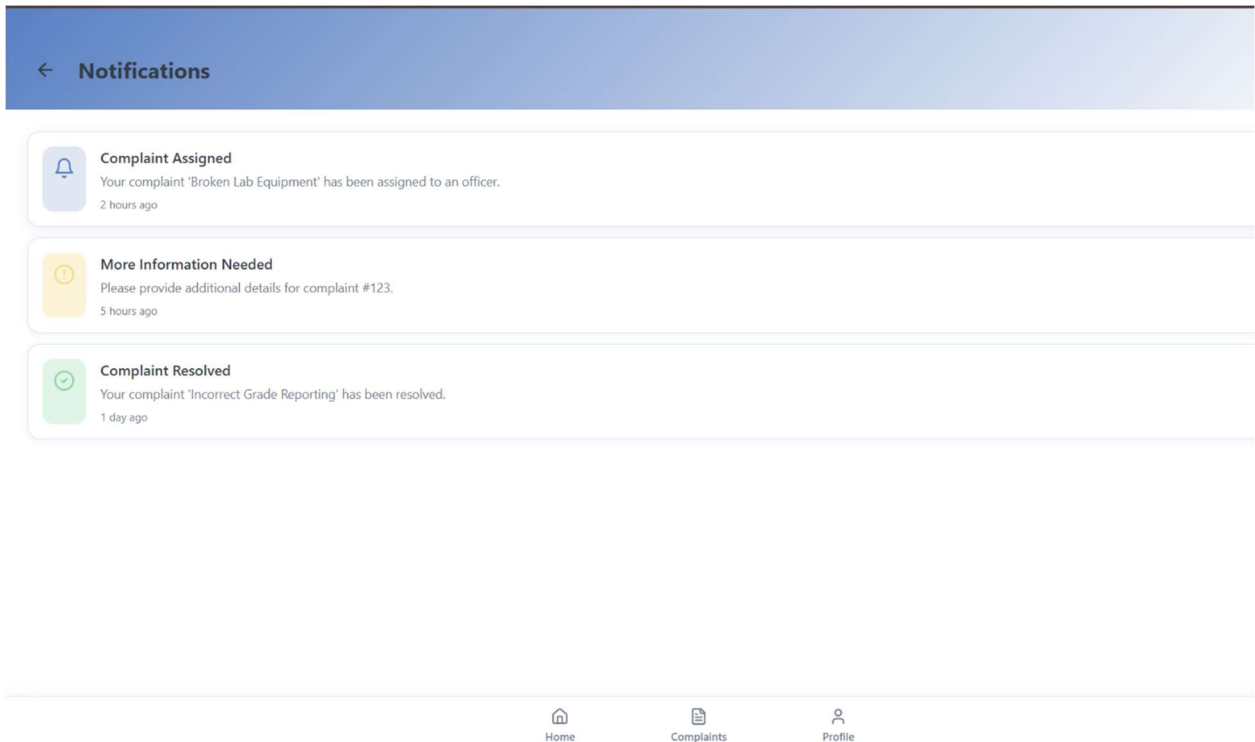
This screen represents the login interface of the College Complaint Management System, serving as the entry point for students. The design follows a clean, modern layout with a soft gradient background that creates a calm and welcoming impression. At the top, a bold “Welcome” heading is used to greet the user, followed by the system title which helps identify the purpose of the application immediately. The login panel contains two tabs—“Login” and “Sign Up”—allowing students to switch between authentication options easily. Beneath this, input fields are provided for email and password, both styled with rounded corners for better visual appeal. A centrally placed blue “Login” button allows the student to authenticate and enter the system. At the bottom, an additional link is provided for Admins and Officers to access their respective login interfaces. Overall, this page is designed to be simple, intuitive, and user-friendly, offering a smooth starting point for users accessing the system.



As shown in Figure 6.1, the Profile Page allows users to view and update their personal information.

ii. Profile Page

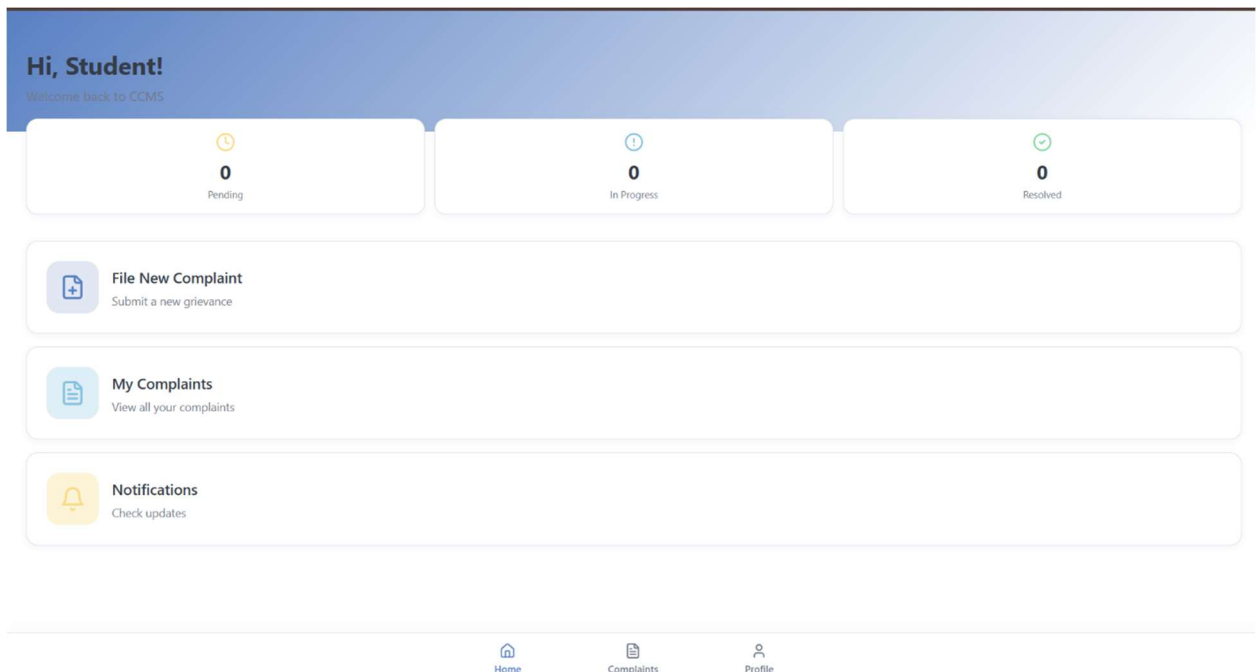
This screen displays the student's profile information. At the top, the user's name and role (Student) are shown along with profile icons for visual clarity. Key details such as Student ID and registered email are displayed in a dedicated card with rounded edges for a modern look. Below the profile information, a prominent red "Logout" button allows the student to sign out securely from the system. The bottom navigation bar remains consistent with other screens, providing quick access to Home, Complaints, and Profile. This page enables students to verify their account information and manage their session easily.



As shown in Figure 6.2, the Notifications Page displays important alerts and updates to keep users informed.

iii. Notifications Page

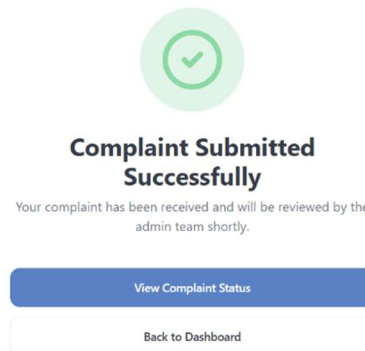
This screen displays the Notifications panel, where students can view important updates regarding their complaints. Each notification is presented in a card-style layout with a distinct background color and icon that indicates the type of alert—such as assignment updates, requests for more information, or resolution status. The first notification informs the student that a complaint has been assigned to an officer, while the second indicates that additional information is required. The last notification confirms that a complaint has been resolved, helping students stay informed throughout the complaint lifecycle. Each card also displays a timestamp, allowing the user to understand how recent each update is. The entire interface is structured for quick scanning and clarity, enabling students to stay updated without navigating through multiple pages. This page ensures transparency and continuous communication between the system and the student.



As shown in Figure 3.6, the Student Dashboard provides an organized interface that allows students to easily access and manage their complaints information.

iv. Dash Board

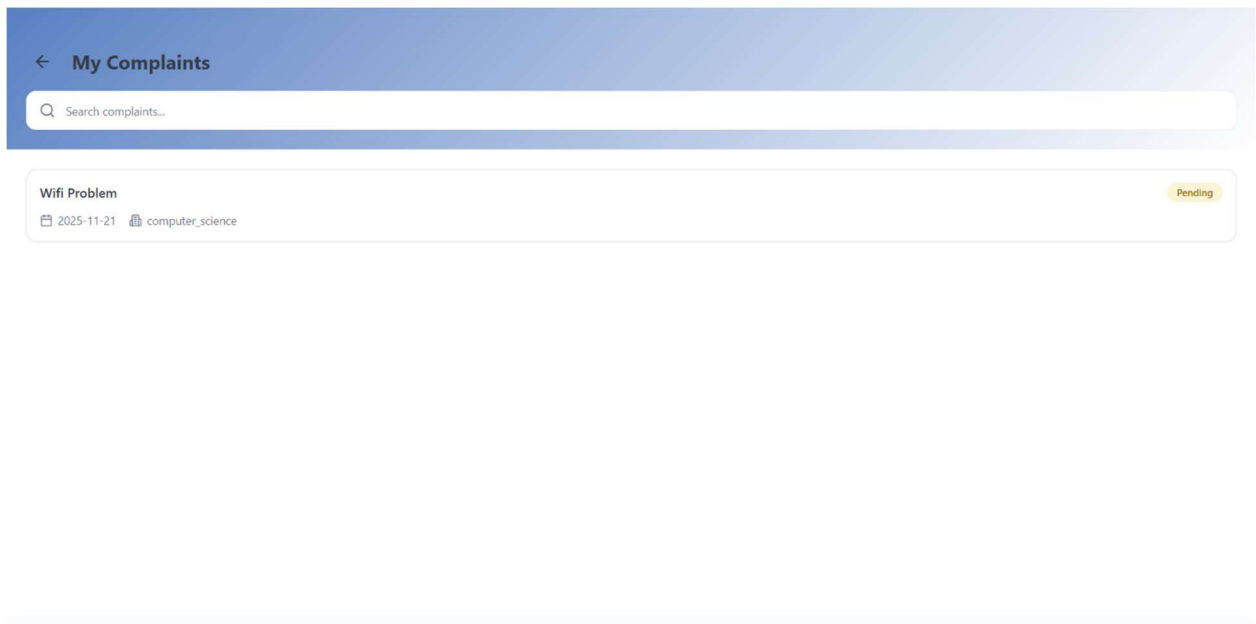
This screen shows the main Student Dashboard, which acts as the central hub after login. The top section greets the student with a personalized welcome message and displays a quick overview of complaint statistics such as Pending, In Progress, and Resolved totals. These counters allow students to monitor the overall status of their complaints at a glance. Below the summary, three large navigation tiles are provided: “File New Complaint,” “My Complaints,” and “Notifications.” Each tile includes an icon and a short description, making navigation extremely intuitive. The bottom of the screen contains a persistent navigation bar with icons for Home, Complaints, and Profile, ensuring quick movement between essential sections. This dashboard is designed to be efficient, visually appealing, and user-friendly, helping students perform actions quickly and conveniently.



As shown in Figure 6.4, the Complaint Submitted Confirmation Page informs the user that their complaint has been successfully submitted.

v. Complaint Submitted Confirmation Page

This screen represents the confirmation interface shown after a student successfully submits a complaint. A large green checkmark at the center visually communicates success and reassurance. Beneath it, the message “Complaint Submitted Successfully” confirms that the complaint has been received by the system. A short description informs the student that the admin team will review the complaint shortly. Two buttons are provided: “View Complaint Status,” which directs the student to the complaint timeline, and “Back to Dashboard,” which returns them to the home page. The minimalist white background and centered layout keep the user focused on the message while offering clear next steps, ensuring the experience is smooth and satisfying.



As shown in Figure 6.5, the My Complaints Page displays all the complaints submitted by the user along with their current status.

vi. Complaints Page

This screen represents the detailed view of a particular complaint. At the top, the system displays the complaint number, department, priority level, and the title of the issue—for example, “Broken Lab Equipment in Computer Lab.” A full description section explains the issue in detail, allowing officers and admins to understand the context clearly. Below this, a vertical “Status Timeline” is shown, illustrating each stage of the complaint lifecycle, including Submitted, Under Review, Assigned, In Progress, and Resolved. Each stage includes a timestamp to provide complete transparency about the progress of the complaint. This timeline-based layout helps the student understand exactly what has happened and what remains to be done. The interface is clean, structured, and focused on information clarity.

Chapter 7: Conclusion

The Public Complaint Management System makes it easy for students to submit complaints and track their status online. It reduces delays, removes manual work, and makes the overall process more organized. Admins and officers can review, assign, and resolve complaints more efficiently, while students receive clear updates through notifications.

Overall, the system improves communication, speeds up problem-solving, and creates a transparent way to handle complaints in colleges.