RAJALAKSHMI ENGINEERINGCOLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

RAJALAKSHMI
ENGINEERING COLLEGE

Laboratory Record Notebook

Name: . . . . . .

Year / Branch / Section: . . . . . .

University Register No. : . . . . . .

College Roll No. : . . . . . .

Semester: . . . . . .

RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

BONAFIDE CERTIFICATE

Name:_____

Academic Year:_____Semester:_____Branch :_____

Register No: | |

Certified that this is the bonafide record of work done by the above student

in the CS19442 – Software Engineering Laboratory during the year 2023-2024

Signature of Faculty in-charge

Submitted for the Practical Examination held on .....

Internal Examiner External Examiner

HOSTEL COMPLAINT MANAGEMENT

SYSTEM

Table of Contents

Software Requirements Specification

1. INTRODUCTION

Purpose:

The purpose of the hostel complaint management system is to provide a structured and efficient platform for students to raise and resolve issues they encounter within their hostel environment. This system is designed to enhance the quality of life for students by ensuring their concerns are promptly addressed, thus fostering a more responsive and communicative relationship between the student body and the hostel administration. By streamlining the complaint process, the system aims to reduce the administrative burden on staff, ensure timely resolution of issues, and improve overall student satisfaction.

Scope:

The hostel complaint management system will be highly beneficial as it streamlines the process of lodging and tracking complaints, making it easier for students to report issues and for administrators to manage them effectively. By providing an accessible platform for students to voice their concerns from any internet-enabled device, the system ensures all complaints are promptly recorded and addressed. It enhances transparency by allowing students to track the status of their complaints in real-time, fostering trust between students and administration. Additionally, the system helps administrators categorize and prioritize complaints, ensuring that urgent issues are swiftly dealt with while maintaining a record of all reported issues. Improved communication between students and hostel management is facilitated through automated email notifications, keeping all parties informed

about the status and resolution of complaints. Furthermore, the system enables data analysis and reporting, allowing hostel management to identify recurring issues and areas for improvement, thus making data-driven decisions to enhance hostel facilities.

2.Intended Audience:

The intended audience for this system includes:

- **Students:** Who need a reliable and straightforward way to report issues and track the progress of their complaints.
- **Hostel Wardens:** Who are responsible for managing student welfare and ensuring a high standard of living within the hostel.
- **College Management:** Who oversee hostel operations and require insights into common issues and areas that need attention to maintain and improve hostel facilities.

Document Overview:

This document outlines the detailed requirements for developing the hostel complaint management system, including its features, functionality, and constraints. It covers functional requirements, non functional requirements, and provides a comprehensive overview of how the system will benefit all stakeholders.

3. FUNCTIONAL REQUIREMENTS

3.1. User Authentication:

- The system shall provide user authentication to ensure secure access to the application. •

Users shall be able to login with their college mail id.

- Registered users shall be able to log in securely using their credentials.

3.2. Complaint Submission:

- The system shall allow students to submit new complaints.
- Users shall have the option to choose from various complaint categories such as electricity, food, sanitation, etc.
- Users shall be able to provide a detailed description of their complaint.

3.3. Complaint Tracking:

- Students shall be able to view the status of their submitted complaints.
- The system shall display the status updates such as "Pending", "In Progress", and "Resolved".

3.4. Administrative Dashboard:

- Administrators shall have access to a dashboard to view all submitted complaints. •

The system shall allow admins to update the status of each complaint.

- Admins shall be able to categorize and prioritize complaints.

3.5. Notification System:

- The system shall send automated email notifications to administrators when new complaints are submitted.
- Studentsshall receive email notifications about the status updates of their complaints.

3.6. Reporting and Analytics:

- The system shall provide tools for generating reports on complaint statistics.

• Administrators shall be able to analyze trends to identify recurring issues and areas for improvement.

4. NON-FUNCTIONAL REQUIREMENTS

4.1. Performance:

• The system shall be responsive and provide quick loading times, even with large amounts of data.

• The application shall support multiple simultaneous users without significant degradation in performance.

4.2. Security:

• User data shall be encrypted and stored securely to prevent unauthorized access or data breaches.

• The system shall implement measures to protect against common security threats.

4.3. Usability:

• The user interface shall be intuitive and easy to navigate, catering to users with varying levels of technical expertise.

• The application shall provide tooltips and contextual help to guide users through the complaint submission process.

4.4. Compatibility:

• The system shall be compatible with popular web browsers such as Chrome, Firefox, and Safari.

• The application shall be responsive and adapt to different screen sizes, including desktops, tablets, and mobile devices.

5. CONSTRAINTS

• The hostel complaint management system shall comply with relevant data protection regulations, such as GDPR, to ensure the privacy and security of user data.

• The system shall rely on third-party APIs or services, such as email services for notifications, which may introduce dependencies and constraints on system performance and availability.

6. CONCLUSION

In conclusion, the Hostel Complaint Management System outlined in this Software Requirements Specification (SRS) aims to provide a robust and user-friendly platform for students to report and track issues within their hostel. By enhancing communication and streamlining the complaint resolution process, the system will significantly improve the efficiency of hostel administration and contribute to a better living environment for students.

USER STORIES

Student User Stories

• As a student, I want to be able to submit a complaint online, so that I can report issues without having to visit the administration office.

• As a student, I want to receive an acknowledgment after submitting my complaint, so that I

know it has been received and is being processed.

- As a student, I want to be able to track the status of my complaint, so that I can see if it is being addressed.
- As a student, I want to provide feedback on the resolution of my complaint, so that I can express my satisfaction or dissatisfaction with the outcome.

Faculty User Stories

- As a faculty member, I want to be notified when a complaint is assigned to me, so that I can take appropriate action.
- As a faculty member, I want to view the details of complaints related to my department, so that I can address issues promptly.
- As a faculty member, I want to update the status of complaints, so that students can be informed about the progress.
- As a faculty member, I want to be able to escalate complaints if I cannot resolve them, so that higher authorities can take over.

Administrative Staff User Stories

- As an administrative staff member, I want to manage all incoming complaints, so that I can ensure they are directed to the appropriate departments.
- As an administrative staff member, I want to generate reports on the types and frequencies of complaints, so that the college can identify recurring issues and address them systematically.
- As an administrative staff member, I want to send notifications to students and faculty regarding updates on complaints, so that all parties are kept informed.
- As an administrative staff member, I want to archive resolved complaints, so that the system remains organized and efficient.

Management User Stories

- As a college administrator, I want to view analytics on complaint data, so that I can make informed decisions to improve college services.
- As a college administrator, I want to set and modify complaint resolution policies, so that the system can adapt to changing needs and standards.
- As a college administrator, I want to have access to all complaint records, so that I can review and audit the handling of complaints for quality assurance.
- As a college administrator, I want to receive alerts for unresolved complaints that exceed a certain time threshold, so that I can ensure timely resolutions.

Technical User Stories

- As a system administrator, I want to ensure the complaint management system is secure, so that sensitive information is protected.
- As a system administrator, I want to perform regular backups of complaint data, so that we can recover information in case of a system failure.
- As a system administrator, I want to manage user access levels, so that only authorized personnel can view or modify complaint information.
- As a system administrator, I want to integrate the complaint management system with other college systems, so that data can be shared seamlessly.

Agile Scrum

1. Project Overview:

The College Hostel Complaint Management System is a web-based platform designed to streamline the process of handling complaints raised by college students regarding hostel facilities and services. The system consists of two primary user roles: students and administrators (wardens). Students can log in to the system to raise complaints, while administrators can access and manage complaints.

2. Scrum Team Setup:

- **Product Owner (PO):** Represents college administration and stakeholders, responsible for defining project objectives, prioritizing features, and ensuring alignment with user needs.
- **Scrum Master (SM):** Facilitates Scrum ceremonies, ensures adherence to Agile principles, and supports the development team in overcoming obstacles.
- **Development Team:** Comprises developers, designers, testers, and other stakeholders involved in building and maintaining the system.

3. Product Backlog Creation:

- **User Stories:** The Product Owner maintains a backlog of user stories representing features and functionalities required by students and administrators. These stories include tasks related to complaint submission, complaint status tracking, and admin management.
- **Prioritization:** User stories are prioritized based on urgency, importance, and stakeholder feedback. Priority is given to critical features such as complaint submission and status tracking.
- **Estimation:** The development team estimates the effort required for each user story using techniques like story points or time-based estimates.

4. Sprint Planning:

- **Sprint Goals:** At the beginning of each sprint, the team defines sprint goals based on the highest-priority user stories from the product backlog. Sprint goals may include implementing specific features or addressing technical debt.
- **Sprint Backlog:** The team selects user stories from the product backlog and breaks them down into smaller, actionable tasks. Tasks related to student complaint submission, admin dashboard development, and database integration are included in the sprint backlog.
- **Capacity Planning:** The Scrum Master ensures that the development team has the capacity to complete the selected tasks within the sprint duration, considering factors like team velocity and resource availability.

5. Sprint Execution:

- **Student Login:**
    - Implement user authentication and authorization mechanisms for student login.
    - Develop a user-friendly interface for students to submit complaints, including form validation and error handling.
    - Integrate with external services or APIs (such as Gmail API) to send complaint notifications to administrators via email.
    - Ensure secure storage of complaint data in the database, adhering to data protection regulations.

- **Admin Login:**
    - Implement user authentication and role-based access control for admin login.
    - Develop an admin dashboard with features for viewing, sorting, and managing student complaints.
    - Enable administrators to update complaint statuses, assign priorities, and communicate with students.
    - Implement data visualization tools or reports to provide insights into complaint trends and resolution metrics.
- **Complaint StatusInterface/Page:**
    - Design a user interface for students to track the status of their complaints, displaying relevant information such as submission date, current status, and actions taken.
    - Retrieve complaint data from the database and dynamically update the interface based on status changes or user interactions.
    - Ensure responsiveness and accessibility of the interface across different devices and screen sizes.

6. Sprint Review:

- At the end of each sprint, the team conducts a sprint review meeting to showcase the completed features and functionalities to stakeholders.
- Stakeholders provide feedback on the delivered increment, highlighting areas ofsatisfaction and areas for improvement.
- The team collects insights from stakeholders to inform future sprint planning and product development decisions.

7. Sprint Retrospective:

- Following the sprint review, the team holds a retrospective meeting to reflect on the sprint process and identify opportunities for improvement.
- The team discusses what went well, what could be improved, and action items for implementing changes in the next sprint.
- Retrospective outcomes are documented and shared with the team to foster continuous learning and improvement.

8. Repeat Sprints:

- The Agile Scrum process continues iteratively, with the team executing multiple sprints to incrementally build and enhance the complaint management system.
- Each sprint follows a similar cycle of planning, execution, review, and retrospective, enabling continuous delivery of value to stakeholders.
- Feedback from stakeholders and end-users is incorporated into subsequent sprints, ensuring that the system evolves to meet changing requirements and preferences.

9. Conclusion:

- Agile Scrum provides a robust framework for developing the College Hostel Complaint Management System, enabling collaborative teamwork, flexibility, and responsiveness to stakeholder needs.

- By embracing Agile principles and practices, the project team can deliver a high-quality, user-centric solution that improves the complaint handling process for both students and administrators.
- Continuous feedback, adaptation, and iteration are key to the success of the project, allowing the system to evolve iteratively and address emerging challenges and opportunities.

NON-FUNCTIONAL REQUIREMENTS

Non-Functional Requirements (NFRs) are essential for ensuring that a complaint management system in a college operates efficiently, securely, and reliably. Here are some key NFRs for such a system:

Performance

- **Response Time:** The system should respond to user actions (e.g., submitting a complaint, viewing status) within 3 seconds.
- **Scalability:** The system should handle up to 10,000 concurrent users without degradation in performance.
- **Throughput:** The system should process at least 100 complaints per minute during peak hours.

Reliability

- **Uptime:** The system should have an uptime of 99.9%, ensuring high availability.
- **Error Handling:** The system should provide meaningful error messages and guide users to resolve common issues.

Security

- **Data Encryption:** All sensitive data, including complaint details and user information, should be encrypted in transit and at rest.
- **Authentication:** The system should use multi-factor authentication (MFA) for all users.
- **Authorization:** Access control mechanisms should ensure that only authorized personnel can view or modify complaint information.
- **Audit Logging:** All actions within the system should be logged for audit purposes, including who accessed or modified data and when.

Usability

- **User Interface:** The system should have an intuitive and user-friendly interface, accessible to users with varying levels of technical proficiency.
- **Accessibility:** The system should comply with WCAG 2.1 guidelines to ensure accessibility for users with disabilities.
- **Mobile Compatibility:** The system should be fully functional on mobile devices and tablets.

Maintainability

- **Code Quality:** The system should follow best practices for code quality, including modularity and documentation, to facilitate easy maintenance and updates.
- **Configuration Management:** The system should allow easy configuration changes without requiring code modifications.

Availability

- **Backup and Recovery:** The system should perform daily backups and have a disaster recovery plan to restore services within 4 hours of a major failure.
- **Redundancy:** Critical components of the system should have redundancy to preventsingle points of failure.

Interoperability

- **Integration:** The system should seamlessly integrate with existing college systems (e.g., student information system, email) through standardized APIs.
- **Data Import/Export:** The system should support data import/export in standard formats (e.g., CSV, XML) for easy data exchange with other systems.

Compliance

- **Data Privacy:** The system should comply with relevant data protection regulations,such as GDPR or CCPA.
- **Legal Compliance:** The system should ensure that all complaint data handling complies with legal and regulatory requirements.

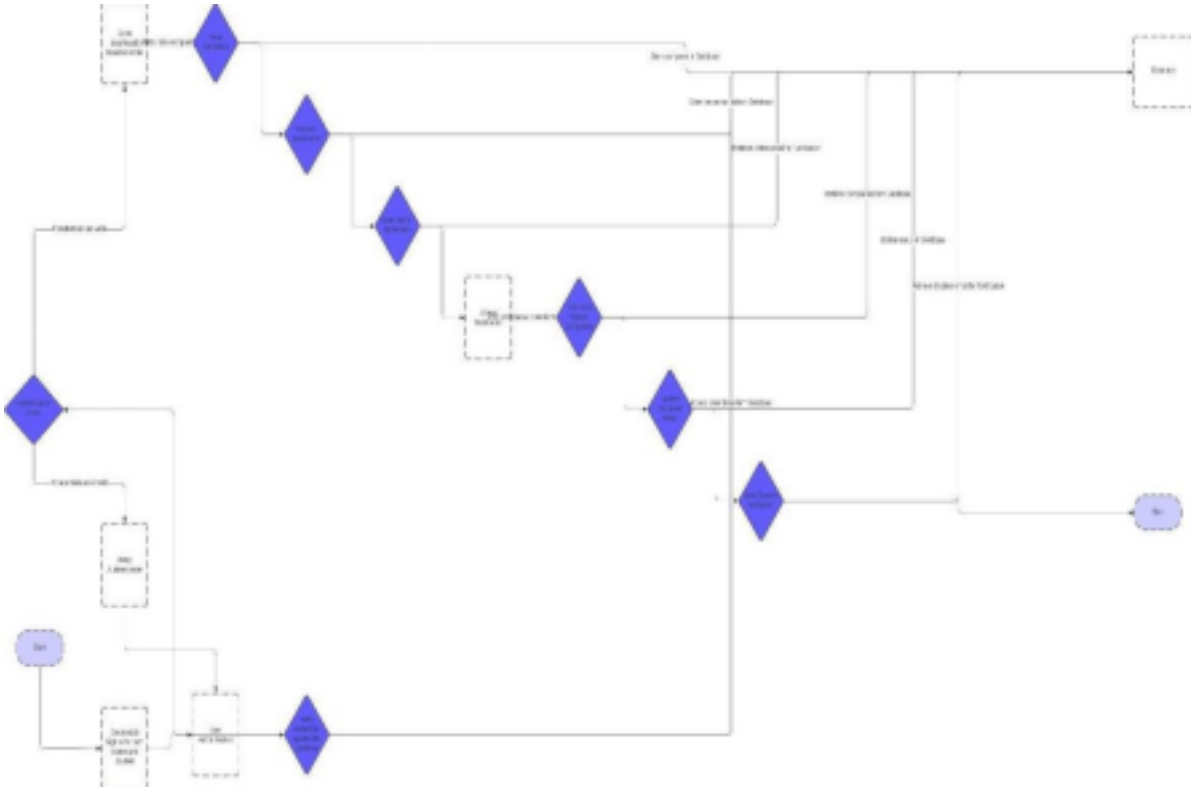
Scalability

- **Horizontal Scaling:** The system should support horizontal scaling to accommodate increasing loads by adding more servers.
- **Vertical Scaling:** The system should support vertical scaling to enhance performance by upgrading existing hardware.

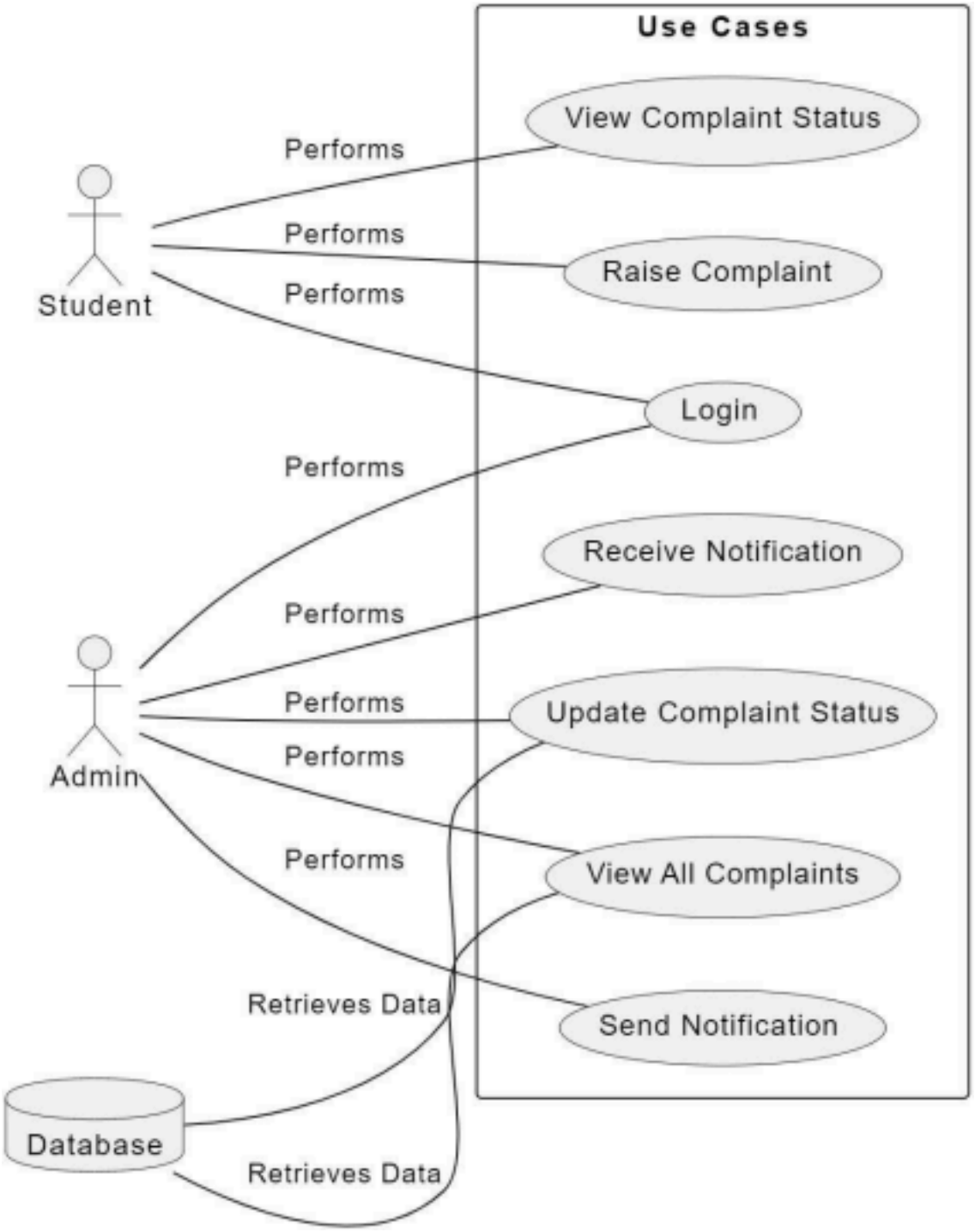
Extensibility

- **Modularity:** The system should be designed in a modular manner to allow for easy addition of new features and components without significant changes to the existing system.
- **Plugin Architecture:** The system should support a plugin architecture to enable third party developers to add functionalities.

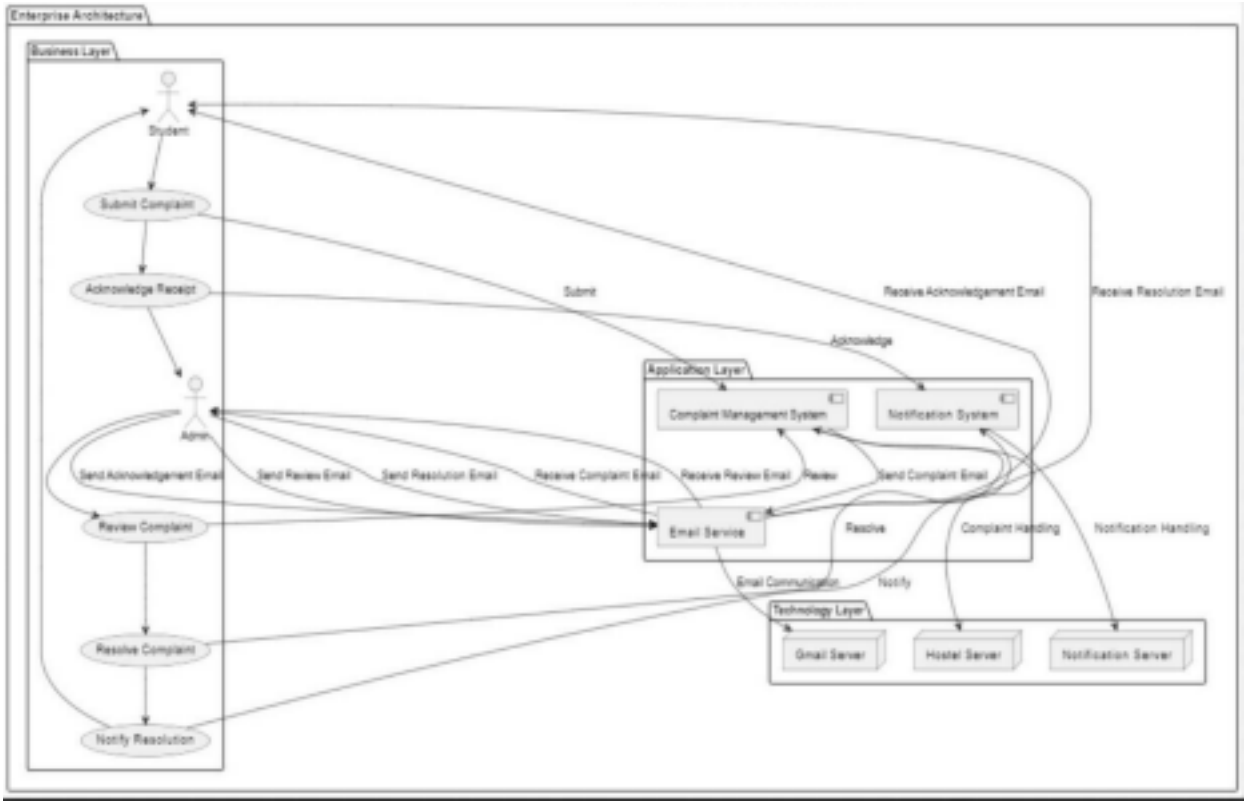
Overall Architecture Diagram



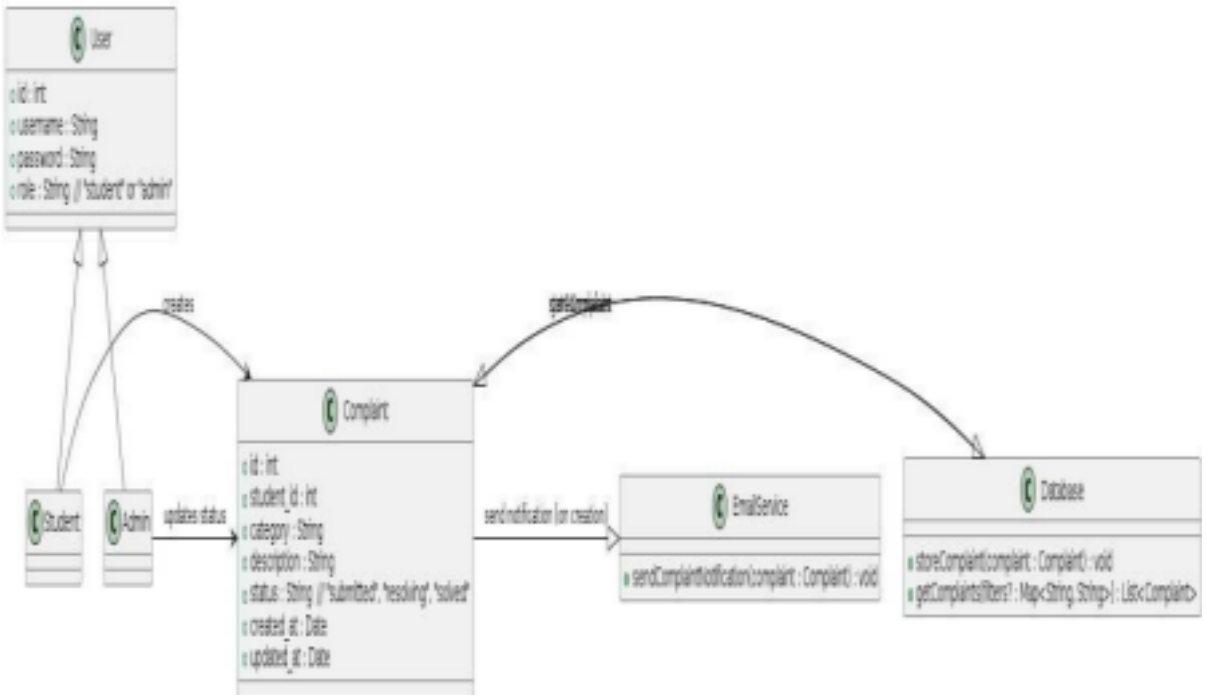
Use Case Diagram



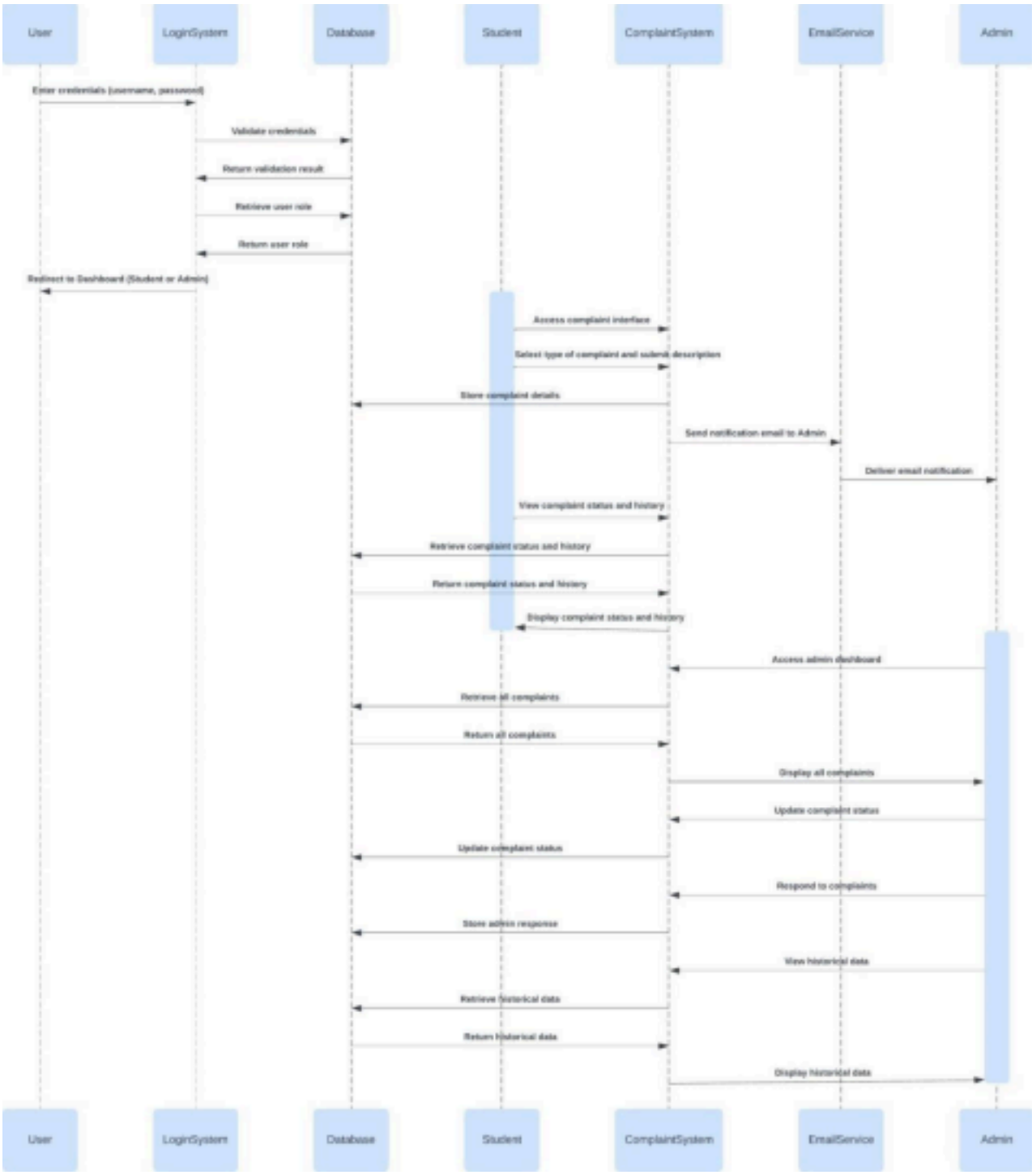
Business Architecture Diagram



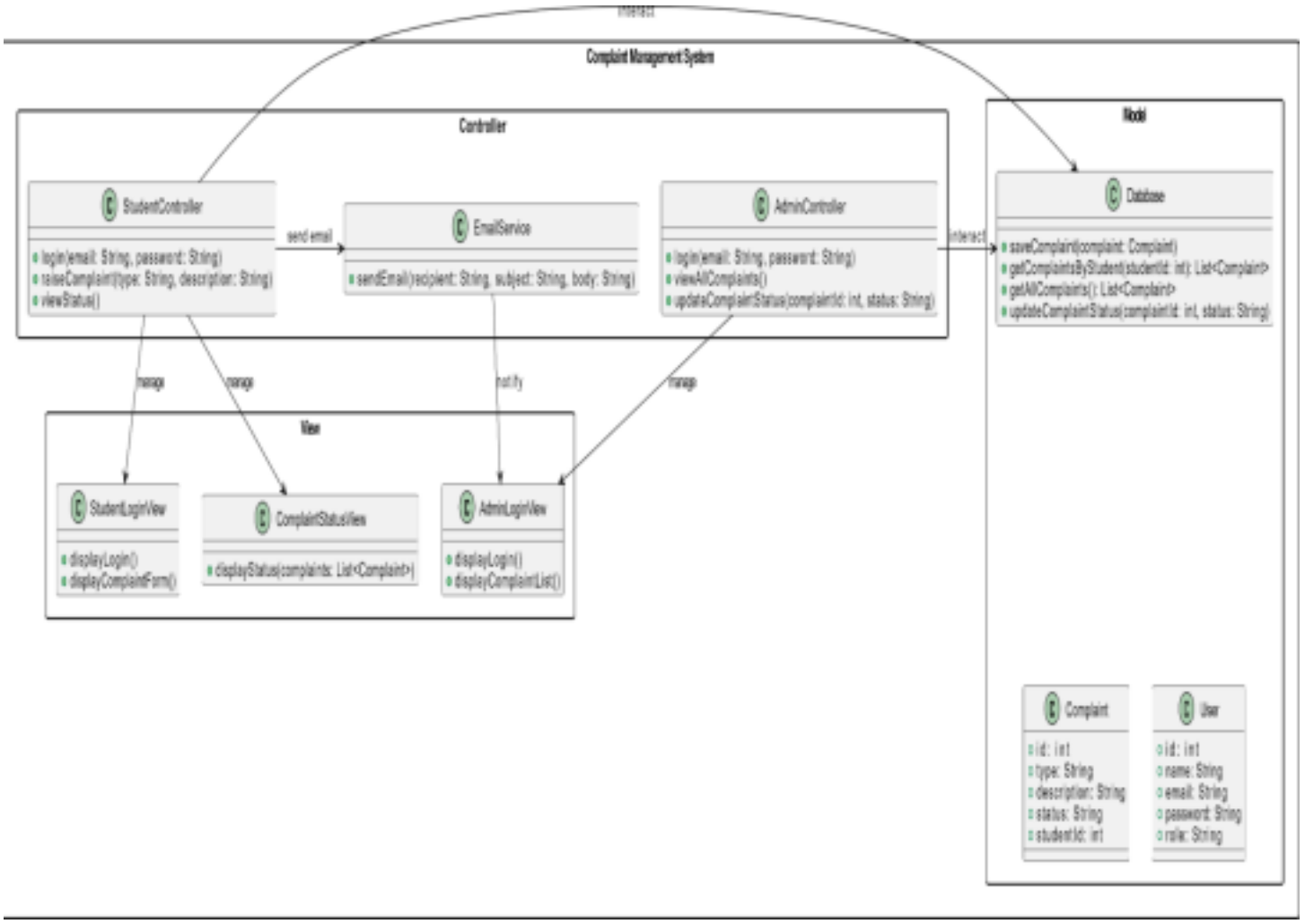
Class Diagram



Sequence Diagram



ARCHITECTURAL PATTERN(MVC)


interact
Complaint Management System
Controller
Model
StudentController
login(email: String, password: String)
raiseComplaint(type: String, description: String)
viewStatus()
send email
EmailService
sendEmail(recipient: String, subject: String, body: String)
AdminController
login(email: String, password: String)
viewAllComplaints()
updateComplaintStatus(complaintId: int, status: String)
interact
Database
saveComplaint(complaint: Complaint)
getComplaintsByStudent(studentId: int): List<Complaint>
getAllComplaints(): List<Complaint>
updateComplaintStatus(complaintId: int, status: String)
manage
manage
notify
manage
View
StudentLoginView
displayLogin()
displayComplaintForm()
ComplaintStatusView
displayStatus(complaints: List<Complaint>)
AdminLoginView
displayLogin()
displayComplaintList()
Complaint
id: int
type: String
description: String
status: String
studentId: int
User
id: int
name: String
email: String
password: String
role: String