SITS/Computer Engineering/Projects/UG/2025-26/B15

A PRILIMINARY REPORT ON

Cloud based Public Complaint System

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

Somashekhar Arabali.	Exam No.: 72247604D
Vaishnavi Kathar.	Exam No.: 72247925F
Karan Keche.	Exam No.: 72247748B
Vitthal Kendre.	Exam No.: 72247750D

GUIDE

Ms.R.T.Waghmode

DEPARTMENT OF COMPUTER ENGINEERING
SINHGAD INSTITUTE OF TECHNOLOGY AND SCIENCE
PUNE, 411041

SAVITRIBAI PHULE PUNE UNIVERSITY
2025-2026

Sinhgad Institutes

CERTIFICATE

This is to certify that the project report entitled
"Cloud Based Public Complaint System"
submitted by

Somashekhar Arabali.	**Exam No.: 72247604D**
Vaishnavi Kathar.	**Exam No.: 72247925F**
Karan Keche.	**Exam No.: 72247748B**
Vitthal Kendre.	**Exam No.: 72247750D**

are bonafide students of this institute and the work has been carried out by them under the supervision of Guide Name and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of Bachelor of Engineering (Computer Engineering).

(Ms.R.T.Waghmode)
Guide
Department of Computer Engineering

(Mrs. A. R. Kamble)
Head
Department of Computer Engineering

(Dr. S. D. Markande)
Principal
Sinhgad Institute of Technology and Science, Narhe

Date:
Place: Pune

ACKNOWLEDGEMENT

We express our gratitude to my guide Ms.R.T.Waghmode for her competent guidance and timely inspiration. It is our good fortune to complete our Project under her able competent guidance. This valuable guidance, suggestions, helpful constructive criticism, keeps interest in the problem during the course of presenting this CLOUD BASED PUBLIC COMPLAIN SYSTEM project successfully.

We would like to thank our Project Coordinator Ms. R. T. Waghmode and all the Teaching, Non-Teaching staff of our department.

We are very much thankful to Mrs. A. R. Kamble, Head, Department of Computer Engineering and also Dr. S. D. Markande, Principal, Dr. V. M. Rohokale, Vice principal, Sinhgad Institute of Technology and Science, Narhe for their unflinching help, support and co- operation during this project work.

We would also like to thank the Sinhgad Technical Educational Society for providing access to the institutional facilities for our project work.

Somashekhar Arabali. (72247604D)
Vaishnavi Kathar.(72247925F)
Karan Keche (72247748B)
Vitthal Kendre.(72247750D)

Date:
Place: Pune

ABSTRACT

The Cloud-Based Public Complaint System is an advanced grievance redressal platform designed to ensure secure, anonymous, and efficient handling of public and institutional complaints. Developed using react microservices architecture, the system enables users to file complaints through multiple channels such as web portals, mobile applications, chatbots, and IVR interfaces. It integrates AI/ML models for intelligent complaint categorization, prioritization, and duplicate detection, ensuring faster resolution and improved resource management.

The system emphasizes security and anonymity by implementing AES/RSA encryption and protected user data storage, thereby preventing retaliation against complainants. Additionally, it promotes transparency and citizen engagement through real-time status tracking, map-based visualization of complaint hotspots, and community voting features. Deployed on a scalable cloud infrastructure using Docker and Kubernetes, the platform ensures high availability, auto-scaling, and performance monitoring through tools like Prometheus and Grafana.

Overall, this project aims to modernize traditional grievance systems by combining cutting-edge technologies such as AI, cloud computing, and microservices to create a transparent, secure, and user-friendly complaint management ecosystem that supports smart city and institutional initiatives.

Keywords: Cloud Computing,Public Complain System

List of Figures

List of Tables

Contents

Chapter 1

Introduction

With the rapid growth of digital communication and the increasing shift towards online governance, the number of complaints and grievances submitted by citizens, students, and employees has risen significantly. However, most existing complaint management systems remain inefficient, non-transparent, and fail to ensure the anonymity and security of complainants. Traditional setups typically rely on manual submissions or unstructured online forms, which makes it challenging for authorities to categorize, track, and resolve complaints in an organized and timely manner.

Many individuals—particularly those reporting sensitive issues such as harassment, corruption, or discrimination—hesitate to raise their concerns due to the fear of exposure or retaliation. This lack of confidentiality discourages open communication and undermines accountability within organizations and institutions. Furthermore, conventional systems often lack real-time tracking, automated prioritization, and data-driven analytics, preventing effective monitoring of complaint trends and identification of recurring issues.

To address these limitations, the Cloud-Based Public Complaint Management System is designed to provide a secure, anonymous, and intelligent grievance

redressal platform that ensures efficiency and transparency throughout the entire complaint lifecycle. The system is developed as a fully web-based solution using React.js and TypeScript for the frontend, while Supabase serves as the cloud-hosted backend for database management, authentication, and secure file storage. This architecture ensures modularity, scalability, and seamless interaction across all user devices.

The platform incorporates AI/ML models to automatically classify complaints according to their type and severity, identify duplicate submissions, and predict estimated resolution times. Each complaint and user-related data is secured using advanced cryptographic methods such as AES and RSA algorithms, maintaining complete confidentiality and data integrity for the complainant.

In addition to the web interface, the system supports multi-channel accessibility—enabling complaints to be registered through a web portal, mobile interface, or integrated chatbot (WhatsApp/Telegram). An IVR (Interactive Voice Response) mechanism can also be connected for telephonic complaint registration. The administrative panel offers real-time dashboards, automated notifications, and map-based visualization of complaint hotspots. These features allow authorities to monitor departmental performance, analyze trends, and take proactive measures for improvement.

By leveraging the capabilities of cloud computing, artificial intelligence, and secure communication technologies, this project delivers a comprehensive and transparent approach to grievance management. It ensures that every individual can report issues safely, that every complaint is tracked and resolved efficiently, and that every contribution strengthens accountability and responsiveness in governance.

1.1 Relevance

The proposed project is intended for citizens, students, employees, and institutional users who need a secure and efficient platform to report grievances, complaints, or issues without fear of retaliation. The output of the system will be a transparent and structured complaint management workflow, allowing users to submit complaints, track their status in real-time, and receive timely updates from concerned authorities.

The system ensures confidentiality and data protection by implementing AES/RSA encryption, allowing users to lodge complaints anonymously while maintaining the integrity of the information. For administrative authorities, the platform provides a centralized dashboard that helps them view, categorize, and prioritize complaints based on severity or urgency. It also includes analytics and reporting tools that highlight complaint trends, response efficiency, and area-wise issue concentration through map-based visualization.

If a user wants to check the progress or resolution details of a particular complaint, the system retrieves only the relevant records corresponding to that complaint ID or category from the database, ensuring accuracy and responsiveness. In such cases, the output will be a status update or detailed complaint report, presented through the user dashboard or via notification alerts.

1.2 Motivation

1. **Many individuals, including students and employees, hesitate to report sensitive issues such as harassment, misconduct, or corruption due to fear of exposure or retaliation. A secure and anonymous complaint management system can encourage victims to come forward without hesitation and ensure that their identity remains protected.**

2. **Traditional complaint systems are often slow, unorganized, and lack transparency. By developing a cloud-based platform integrated with AI-driven complaint classification and real-time tracking, users can easily register and monitor complaints, while authorities can efficiently manage and resolve them, improving accountability and trust.**

1.3 Problem Definition

"Develop a secure, cloud-based platform that allows users to submit and track complaints anonymously, ensuring transparency, accountability, and data protection through AI-based categorization and encryption."

Problem	Solution
Lack of confidentiality discourages users from reporting sensitive issues.	**Provide anonymous complaint submission with AES/RSA encryption to ensure data security and user privacy.**
Inefficient manual complaint handling leads to delays and poor tracking.	**Implement AI-based complaint classification and a real-time tracking dashboard for faster and transparent resolution.**
Difficulty in managing high complaint volumes and ensuring scalability.	**Deploy system on cloud using microservices architecture with Docker and Kubernetes for scalability and reliability.**

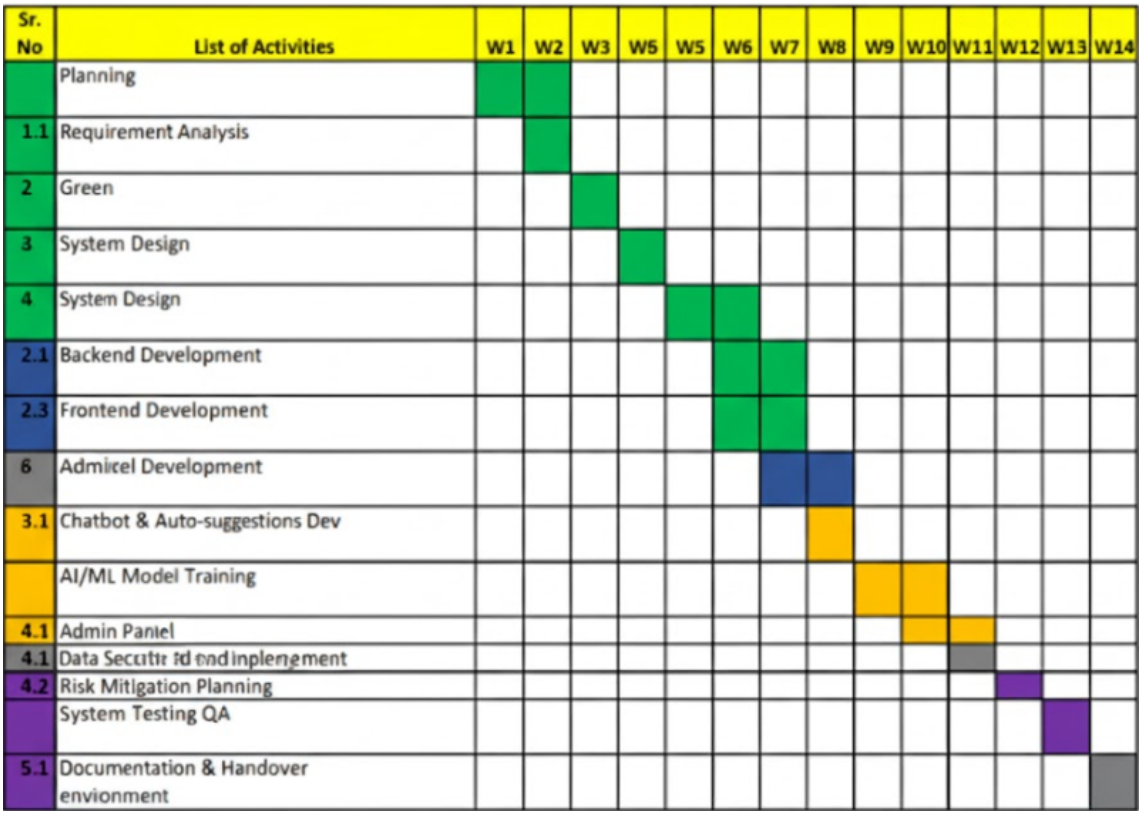
Figure 1.1: Problem Definition for Cloud-Based Public Complaint Management System

1.4 Objective Statements

The project is aimed to design and develop a secure, cloud-based platform with the following primary objectives:

1. **To enable secure and anonymous complaint submission, ensuring protection of the complainant's identity using AES/RSA encryption.**
2. **To implement AI/ML-based complaint classification and prioritization for efficient and intelligent grievance handling.**
3. **To provide real-time complaint tracking and status updates through an interactive dashboard for both users and administrators.**
4. **To support multi-channel accessibility, allowing users to submit complaints via web application, mobile app, chatbot, or IVR system.**
5. **To deploy the system using a scalable microservices architecture on the cloud, ensuring high availability and performance.**

1.5 Schedule of Project Work



Sr. No	List of Activities	W1	W2	W3	W5	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
	Planning														
1.1	Requirement Analysis														
2	Green														
3	System Design														
4	System Design														
2.1	Backend Development														
2.3	Frontend Development														
6	Admicel Development														
3.1	Chatbot & Auto-suggestions Dev														
	AI/ML Model Training														
4.1	Admin Paniel														
4.1	Data Secutir id and inplenement														
4.2	Risk Mitigation Planning														
	System Testing QA														
5.1	Documentation & Handover envionment														

Figure 1.1: Schedule of Project Work

1.6 Budget of the Project

COCOMO Model (Constructive Cost Estimation Model) was proposed by Barry Boehm. COCOMO predicts the effort, time, and cost required to develop a software product based on its estimated size. The model helps in understanding the economic feasibility and overall budget requirements of the project.

COCOMO stands for "Constructive Cost Model". According to Boehm, software cost estimation should be done through three stages: Basic COCOMO, Intermediate COCOMO, and Complete COCOMO.

For this project, we are using the Basic COCOMO model, which categorizes projects into three types:

1. **Organic: Suitable for small projects with experienced teams and flexible requirements.**
2. **Semidetached: Suitable for medium-sized projects with mixed experience levels and moderate complexity.**
3. **Embedded: Suitable for large projects with tight hardware, software, and operational constraints.**

Since our system involves multiple integrated modules such as user authentication, AI-based complaint classification, encryption, and cloud deployment, it falls under the Semidetached category.

The Basic COCOMO formula takes the form:

$$E = a_b \times (KLoC)^{b_b} \quad \textbf{[person-months]}$$

$$D = c_b \times (E)^{d_b} \quad \textbf{[months]}$$

$$P = \frac{E}{D} \quad \textbf{[persons]}$$

Where:

- E = **Effort applied in person-months**
- $KLoC$ = **Estimated thousands of delivered lines of code**
- D = **Development time in months**
- P = **Number of persons required**

The coefficients a_b, b_b, c_b, and d_b are given below:

Project Type	a_b	b_b	c_b	d_b
Organic	**2.4**	**1.05**	**2.5**	**0.38**
Semidetached	**3.0**	**1.12**	**2.5**	**0.35**
Embedded	**3.6**	**1.20**	**2.5**	**0.32**

Calculation:

For the Cloud-Based Public Complaint Management System, we assume the estimated code size as $KLoC = 8.0$ (approximately 8000 lines of code).

$$E = a_b \times (KLoC)^{b_b}$$

$$E = 3.0 \times (8.0)^{1.12}$$

$$E = 3.0 \times 9.36 = 28.08 \textbf{ person-months}$$

$$D = c_b \times (E)^{d_b}$$

$$D = 2.5 \times (28.08)^{0.35}$$

$$D = 2.5 \times 2.97 = 7.43 \textbf{ months}$$

$$P = \frac{E}{D}$$

$$P = \frac{28.08}{7.43} = 3.78 \approx 4 \textbf{ persons}$$

Cost Estimation:

Each developer earns Rs. 25,000 per month. Hence, cost per person per month = Rs. 25,000.

Therefore, The estimated development cost for the Cloud-Based Public Complaint Management System is **approximately Rs. 6,50,000 to Rs. 7,00,000, requiring **4 team members** and a development duration of **around 7.5 months**.**

Hence, this chapter provides a clear idea about the project's problem definition, objectives, schedule, and budget estimation using the COCOMO model.

Chapter 2

Literature Survey

Existing grievance redressal systems have been widely implemented across government and institutional frameworks to manage citizen complaints. However, traditional systems often depend heavily on manual operations, resulting in inefficiency, lack of transparency, and limited data security. Many earlier models focused primarily on providing a basic web interface for complaint submission without ensuring anonymity, encryption, or intelligent routing mechanisms.

2.1 Literature Survey

2.1.1 Literature Survey on Existing Grievance Redressal Systems

I

In the paper [1], Mawar et al. proposed a whistleblowing-based complaint system using the MD5 cryptographic method for identity confidentiality. Although it emphasized secure complaint submission, the use of MD5 made the system vulnerable to modern cryptographic attacks and limited its scalability

in large-scale environments.

Saha et al. [2] developed an NLP and machine learning-based complaint classification system that automatically categorizes complaints and routes them to relevant departments. This work successfully demonstrated the use of artificial intelligence in complaint prioritization but lacked secure data handling and encryption mechanisms, leaving sensitive user data exposed.

Deepika et al. [3] introduced a petition analyzer system with a chatbot interface that enabled users to file and track complaints interactively. While the integration of conversational AI enhanced user experience, the prototype did not incorporate robust data security measures or cloud-based scalability.

Wadkar et al. [4] presented an AI-driven complaint management system to automate classification and response generation. Their approach utilized machine learning algorithms for intelligent routing but primarily focused on algorithmic design rather than end-to-end implementation in a real-world cloud environment.

Hiremath et al. [5] proposed a public online complaint registration and management system using web technologies. The system provided a structured form-based submission process and basic tracking functionality. However, it lacked multi-channel accessibility and real-time analytics, which limited its usability for large organizations or government bodies.

2.1.2 Literature Survey on Cloud Computing and Microservices

Cloud computing has revolutionized the way software systems are designed, deployed, and maintained. It enables on-demand access to computing resources such as servers, storage, and networking over the internet, thereby reducing infrastructure costs and improving scalability. Microservices architecture, on the other hand, promotes modularization of large applications into smaller,

independently deployable services that communicate through APIs. The combination of cloud computing and microservices has become the foundation for building scalable, resilient, and distributed systems in modern software engineering.

In their study, Mell and Grance (NIST, 2011) defined cloud computing as a model that allows convenient, on-demand network access to shared pools of configurable computing resources. This model provides essential characteristics such as scalability, elasticity, measured service, and resource pooling, which have since become the backbone of most web-based and enterprise applications.

In the paper by Zhang et al. (2018), the authors discussed the evolution of cloud computing from traditional service-oriented architecture (SOA) toward microservices. They emphasized that microservices enable agility and faster deployment cycles, especially when combined with containerization technologies such as Docker and orchestration tools like Kubernetes. Their findings highlight that distributed microservices can independently scale, improving both performance and fault tolerance.

2.1.3 Literature Survey on Security and Data Protection

Security and data protection are critical aspects of any modern information system, particularly those handling personal or sensitive user data. With the increasing adoption of cloud-based infrastructures and web applications, ensuring confidentiality, integrity, and availability of information has become a major concern for developers, organizations, and governments alike. Modern systems must therefore incorporate strong encryption mechanisms, secure authentication protocols, and privacy-preserving data management strategies to maintain user trust and system reliability.

According to Stallings (2017), encryption is one of the fundamental building blocks of information security, providing confidentiality by transforming

plaintext into unreadable ciphertext. Symmetric encryption algorithms such as the Advanced Encryption Standard (AES) and asymmetric algorithms like the Rivest–Shamir–Adleman (RSA) have become global standards due to their robustness and computational efficiency. These techniques are essential in securing sensitive transactions, communication, and database records in web applications.

2.1.4 Literature Survey on Artificial Intelligence for Complaint Classification

Artificial Intelligence (AI) and Machine Learning (ML) have become essential tools in automating data-driven decision-making processes across various domains, including grievance management. In traditional systems, complaints are manually categorized by administrative staff, which often leads to delays, inconsistencies, and human bias. To address these issues, researchers have explored AI-based approaches that can automatically classify and prioritize complaints based on content, tone, and urgency.

In the study conducted by Pandey et al. [7], Natural Language Processing (NLP) techniques were employed to analyze complaint texts and extract semantic meaning. The authors applied the Naïve Bayes and Support Vector Machine (SVM) algorithms to classify complaints into predefined categories such as "infrastructure," "safety," and "harassment." Their results demonstrated an accuracy of over 87

2.1.5 Literature Survey on Transparency and User Engagement

Transparency and user engagement are fundamental principles in the design of modern digital governance and feedback systems. A transparent system

promotes accountability, builds user trust, and ensures that decisions or actions taken by authorities are visible and verifiable. Similarly, user engagement enhances participation, improves service delivery, and fosters a two-way interaction between users and administrators. In grievance management systems, these two aspects are crucial for encouraging individuals to report issues confidently and for enabling authorities to respond effectively.

In the study by Fung et al. (2007), the authors introduced the concept of *"participatory transparency"* in governance systems, emphasizing that citizens are more likely to engage with institutions when they can see how their input influences outcomes. The paper argued that transparency should not merely involve the publication of data, but also the provision of feedback loops that demonstrate accountability and responsiveness.

Harrison and Sayogo (2014) explored open government data initiatives and identified that transparency enhances trust when users are provided with clear, accessible, and timely information. They highlighted that real-time updates, open dashboards, and traceable workflows increase public confidence in digital systems. Their work supports the idea that grievance redressal platforms should make complaint statuses and administrative actions visible to users at all stages.

In a study by Grimmelikhuijsen et al. (2013), the impact of transparency on citizen trust was analyzed empirically. The findings indicated that perceived transparency significantly improves public confidence, particularly when users receive personalized and continuous communication about their interactions with institutions. This suggests that complaint management systems benefit from real-time notifications and visible progress indicators.

2.2 Summary of Literature Review

The literature review highlights that traditional grievance redressal systems suffer from inefficiency, lack of transparency, and weak data security. Existing studies have focused on automating complaint submission and classification but often ignored anonymity, encryption, and real-time analytics.

Research on cloud computing and microservices emphasizes scalability, modularity, and flexibility in system design. These findings influenced the proposed system's architecture, which uses React.js for the frontend and Supabase as a cloud-based backend for secure data storage and real-time updates.

Studies on security and data protection recommend hybrid encryption (AES/RSA) and anonymization to safeguard sensitive data. This guided the integration of AES/RSA encryption and Row Level Security (RLS) in the proposed system to ensure confidentiality and controlled access.

Literature on transparency and user engagement stresses the importance of open dashboards, feedback mechanisms, and user notifications to build trust. These insights shaped the inclusion of real-time dashboards, chatbot interaction, and map-based visualization in the project.

Overall, the reviewed works reveal that no single system effectively combines security, scalability, intelligence, and transparency. The proposed Cloud-Based Public Complaint Management System bridges this gap by offering a secure, anonymous, AI-powered, and cloud-integrated platform for efficient grievance handling.

Table 2.1: Literature Review on Complaint Management Systems

Author / Year of Publication	**Title**	**Strength**	**Weakness**
Mawar M., Assiddiq M., Qashlim A., 2021, Dept. of Information Systems, University Al Asyariah Mandar (Indonesia)	The Complaint System Based on Whistleblowing Concept and Message Digest 5 Cryptographic Method for Regency Inspectorate Office in Polewali Mandar	Implements whistleblowing concept and MD5 cryptographic hashing to ensure complainant identity confidentiality	Uses MD5, which is weak cryptographically; limited scalability for large-scale deployment
Saha A., Chaure A., Aayush, Bodhe A., Bhagat T.D., 2024, Dept. of Information Technology, Sinhgad College of Engineering, Pune (India)	Automated Complaint Classification and Routing Using NLP and Machine Learning	Automates categorization and routing of complaints using NLP and ML models, reducing manual workload	Focuses primarily on text classification; lacks complete complaint lifecycle management
Deepika R., Shravan V., Sai Harish R., 2025, Dept. of Computer Science and Engineering, Sri Venkateswara College of Engineering, Sriperumbudur (India)	Petition Analyzer: Grievance Management System with Chatbot	Integrates chatbot for real-time complaint interaction and tracking; supports automated categorization using NLP	Prototype stage only; not yet tested for high traffic or multi-department usage
Anonymous Authors, 2025, Dept. of Computer Science, Hindusthan College of Engineering and Technology, Coimbatore (India)	A Digital Governance Framework for Intelligent Complaint/Grievance Management System	Cloud-based public grievance portal with photo upload, complaint tracking, and admin dashboard features	Limited field testing; lacks scalability evaluation and predictive analytics integration

Gap Analysis

- **Limited large-scale deployment studies for anonymous complaint systems.**
- **Most systems lack automated notifications and real-time tracking features.**
- **Few systems combine anonymity with efficient workflow and categorization.**
- **Evaluation metrics for complaint resolution effectiveness are not standardized.**
- **Mobile accessibility and usability for diverse users are often not addressed.**

Chapter 3

Software Requirements Specification

In the previous chapter, the literature survey was summarized. The Software Requirements Specification (SRS) forms the foundation of the entire project. It provides detailed information about the system to various stakeholders, including the development team, quality assurance team, operations, and maintenance teams.

The SRS ensures that all functional and non-functional requirements are clearly defined, helping to avoid misunderstandings during development and implementation. It also serves as a reference for future maintenance and enhancement of the system. By clearly specifying the requirements, the SRS helps in making informed decisions about the system's design, implementation, and lifecycle management.

3.1 Introduction

Organizations often face challenges in managing complaints effectively while maintaining the confidentiality of the complainants. Traditional complaint submission methods may discourage users from reporting issues due to fear of retaliation or lack of anonymity. To address this problem, our project focuses on developing an **Anonymous Complaint Management System that allows users to submit complaints securely and confidentially. The system ensures that the identity of the complainant is protected while enabling administrators to efficiently track, categorize, and resolve complaints. Additionally, the system incorporates features like complaint categorization, search, and filtering, which help administrators focus on relevant issues quickly. This reduces manual effort, improves response times, and enhances overall transparency and accountability within the organization.**

3.1.1 Project Scope

The project provides a secure platform for users to submit complaints anonymously. It allows administrators to efficiently manage, categorize, and track complaints. The system ensures confidentiality, reduces manual effort, and enables quick resolution through features like search, filtering, and status tracking.

1. **User : - A user can submit a complaint anonymously through the web portal without revealing personal details. The user can track the status of their complaint if needed and access notifications about updates.**

   Properties:

   (a) **Can submit complaints anonymously.**

   (b) **Can view the status of submitted complaints.**

   (c) **Can categorize complaints or attach relevant details.**

2. **Administrator : - An admin manages all submitted complaints. They can view, categorize, assign, and update the status of complaints. The admin ensures timely resolution and maintains confidentiality.**

   Properties:

   (a) **Can view and manage all complaints in the system.**

   (b) **Can assign complaints to responsible authorities.**

   (c) **Can generate reports and track complaint resolution.**

3.1.2 Assumptions and Dependencies

Assumptions:

1. **Users have a stable internet connection to access the web application and submit complaints through the platform.**
2. **Supabase cloud services remain available to handle authentication, database operations, and real-time updates.**
3. **The system ensures complete anonymity for users submitting complaints and performs automatic recovery in case of backend or service failure.**
4. **All users access the system using modern web browsers compatible with React-based applications.**

Dependencies:

1. **Supabase (PostgreSQL-based backend) is used for storing complaint records, user information, and audit logs.**
2. **The system relies on React.js and TypeScript for building a responsive and interactive web interface for users and administrators.**

3. **Tailwind CSS and Lucide React are used for styling and icons, ensuring a clean and accessible UI.**

4. **Supabase Authentication and Row Level Security (RLS) are essential for managing secure user access and maintaining data privacy.**

5. **The system depends on secure HTTPS/SSL communication for safe transmission of data between client and cloud services.**

3.2 External Interfaces Requirements

3.2.1 User Interfaces

1. **Login Screen**

   - The Login Screen allows both users and administrators to securely log into the system using Supabase Authentication. - The system supports both registered and anonymous logins, ensuring secure session handling through JSON Web Tokens (JWT).

2. **Complaint Submission & Options Screen**

   - After login, users can submit new complaints anonymously and provide relevant details such as category, description, and attachments. - The options available on this screen include:

   (a) **Submit a new anonymous complaint**

   (b) **View or track the status of previously submitted complaints**

   (c) **Search or filter complaints (admin functionality)**

3. **Complaint Status and Management Screen**

   - Users can check the real-time status of their complaints through interactive dashboards. - Administrators can view, categorize, assign, and update complaints using secure Supabase database queries. - The admin dashboard provides visual analytics, heatmaps, and reporting features for monitoring trends and departmental performance.

3.2.2 Software Interfaces

- The system can operate on Windows, Linux, or macOS environments. - Supports modern web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari. - Frontend Technologies: React.js with TypeScript, Tailwind CSS for styling, Lucide React for icons, and Vite for fast builds. - Backend and Database: Supabase (PostgreSQL-based cloud backend) for data storage, authentication, and real-time updates. - Communication: RESTful API and WebSocket integration for real-time synchronization between client and database. - Notifications: Integrated email or push notifications via Supabase functions or third-party services (e.g., Twilio or Firebase Cloud Messaging).

3.2.3 Communication Interfaces

- **Web browser interface for both users and administrators to access the system securely.**
- **Internet or Wi-Fi connection is required for complaint submission and real-time tracking.**
- **Secure HTTPS/SSL communication is enforced for all client–server data exchanges.**
- **WebSocket channels (via Supabase Realtime) are used to deliver instant updates to users and administrators.**

3.3 Nonfunctional Requirements

The following non-functional requirements ensure that the Cloud-Based Public Complaint Management System is secure, reliable, efficient, and user-friendly.

3.3.1 Performance Requirements

- The system should process complaint submissions, updates, and status retrievals within 3–5 seconds. - Supabase real-time architecture should handle multiple concurrent users efficiently, maintaining at least 99.5% uptime. - Dashboard analytics, reporting, and map visualizations should be optimized for fast rendering and minimal latency. - The system should automatically scale backend resources to maintain performance during peak loads.

3.3.2 Reliability Requirements

- The system must remain operational even if a minor service or component fails, ensuring modular recovery due to its microservice-inspired design. - In case of any service outage, automatic recovery or failover mechanisms should restore normal operations. - Regular automated database backups should be scheduled in Supabase to prevent data loss. - All user and admin actions should be logged for audit and debugging purposes, with redundancy to ensure traceability.

3.3.3 Security Requirements

Data Transfer:

- **All communications between users and the server shall be encrypted using HTTPS/SSL.**
- **Users shall be automatically logged out after a defined period of inactivity.**
- **No sensitive information, such as passwords or complaint content, shall be stored in cookies or displayed on the client side.**

Data Storage:

- **User passwords shall be securely hashed and never displayed.**

- **Complaint data shall be encrypted in the database to ensure confidentiality.**
- **Backend access shall be restricted to authenticated administrators only.**
- **Audit logs shall be maintained for all administrative actions on complaints.**

3.3.4 Software Quality Attributes

1. **Efficiency: The system should perform its functions using resources optimally.**
2. **Portability: The system should be compatible with various operating systems and web browsers.**
3. **Testability: The system should allow easy testing of features and tracking of execution.**
4. **Usability: Interfaces should be intuitive, and logs and reports should be clear for both users and administrators.**
5. **Maintainability: The system should allow easy updates, feature additions, and bug fixes.**

sectionSystem Requirements

3.3.5 Database Requirement

- **Supabase (PostgreSQL-based cloud database) for storing complaint data, user information, and audit logs.**
- **Real-time synchronization and Row Level Security (RLS) for data integrity and access control.**

3.3.6 User Interfaces

The system will provide a web-based user interface (WUI) using interactive forms, buttons, and dashboard components. The interface will be responsive, supporting both desktop and mobile devices. Notifications will alert users about submission confirmations, complaint status updates, and administrative actions.

1. **Front-End Design:**
   (a) **React.js**
   (b) **TypeScript**
   (c) **Tailwind CSS**
   (d) **Lucide React (for modern and responsive icons)**
2. **Backend / Server-Side:**
   (a) **Supabase (for authentication, database, and file storage)**
   (b) **Vite (as a fast development and build tool)**
   (c) **useLocalStorage Hook (for persistent local client-side data storage)**

3.3.7 Hardware Requirements

Minimum hardware requirements for development and deployment:

- **Processor: Intel Core i3 or higher**
- **RAM: 4 GB or more**
- **Hard Disk Space: 10 GB or more**

3.3.8 Software Requirements

- **Operating Systems: Windows 10 or later, macOS, Linux**
- **Development Tools: Visual Studio Code (recommended), Node.js (v18+), Git**
- **Web Browsers: Google Chrome, Mozilla Firefox, Microsoft Edge**
- **Supabase CLI and NPM (for project setup and deployment)**

3.4 Analysis Models: SDLC Model to be applied

Iterative Model

The iterative model focuses on building an initial simplified version of the system, which is progressively enhanced with additional features and refinements until the final system is complete.

Technologies Used:

- **React: A JavaScript library for building fast, modular, and reusable user interfaces.**
- **TypeScript: A typed superset of JavaScript that enhances code quality, readability, and maintainability.**
- **Tailwind CSS: A utility-first CSS framework for rapidly styling components and layouts with responsive design.**
- **Vite: A fast build tool providing instant development server and optimized production builds.**
- **Lucide React: A modern collection of customizable, open-source icons for a clean and consistent UI.**

- **Supabase: Cloud-based backend as a service for database operations, authentication, and file storage.**
- **useLocalStorage Hook: A custom React hook for persistent client-side data storage, ensuring offline accessibility.**

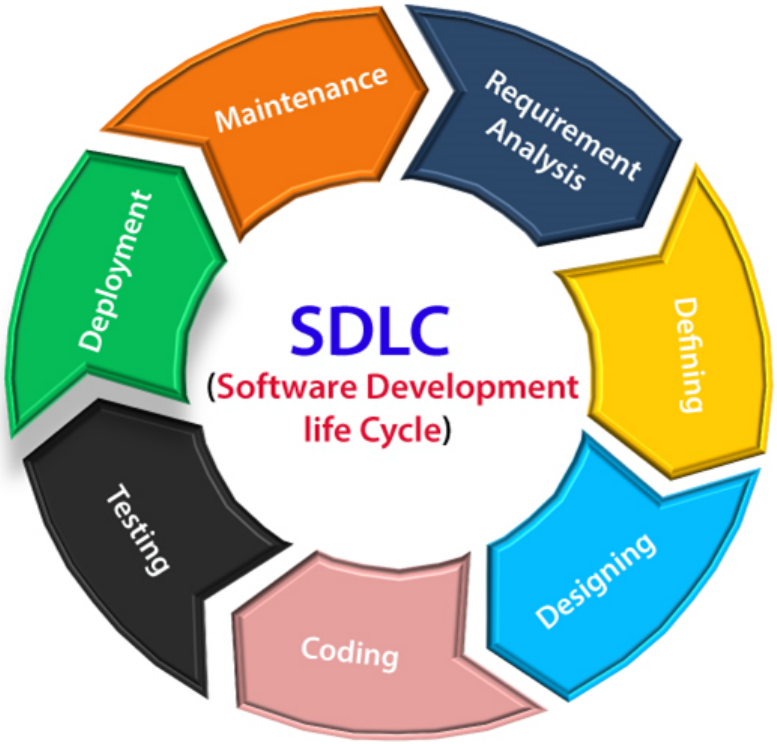


Figure 3.1: Iterative Model

The iterative model is suitable for the Anonymous Complaint Management System because the system modules (user complaint submission, admin management, reporting) can be developed, tested, and validated independently before final integration.

Steps include:

1. **Planning and requirement analysis**
2. **Defining system requirements**

3. **Designing the software**
4. **Developing the project**
5. **Testing modules independently**
6. **Deployment**
7. **Maintenance and updates**

The process of building the Complaint Management System consists of:

1. **Developing individual modules (complaint submission, admin management, reporting) and testing them independently.**
2. **Integrating all modules and performing system-wide testing to ensure functionality, reliability, and security.**
3. **Deploying the fully integrated system for real-time usage and continuous monitoring.**

3.5 System Implementation Plan

The Anonymous Complaint Management System will be implemented according to the schedule and requirements described in the previous sections. All functional and non-functional requirements, along with system and hardware/software specifications, will guide the development.

The Iterative Model will be followed for system development. Individual modules—such as anonymous complaint submission, admin complaint management, reporting, and search/filter features—will be developed, tested, and integrated incrementally.

Once all modules are integrated, the complete system will undergo final testing to ensure functionality, reliability, security, and performance meet the specified requirements before deployment.

Chapter 4

System Design

In the previous chapters, we discussed the project background, analyzed existing systems, and specified the functional and non-functional requirements for the Anonymous Complaint Management System. This chapter presents the system design, including architecture, workflow, and data flow diagrams (DFDs).

4.1 System Architecture

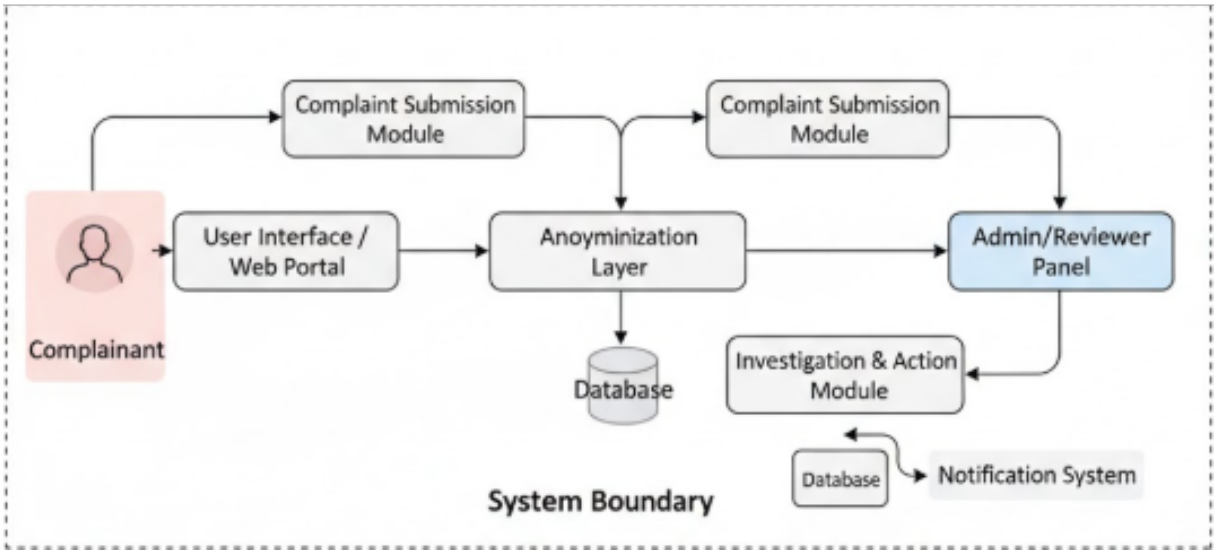


Figure 4.1: Block Diagram for Anonymous Complaint Management System

The block diagram in Fig. 5.1 gives an overview of the system architecture for complaint management.

The workflow for the system works as follows:

1. **User Registration/Login: Users or administrators log into the system to access their functionalities.**
2. **Complaint Submission: Users submit complaints anonymously with relevant details and category.**
3. **Complaint Storage: Complaints are securely stored in the database with encryption.**

4. **Admin Review: Administrators access the complaints, categorize them, assign responsible personnel, and update the status.**
5. **Status Tracking: Users can check the status of their complaints without revealing their identity.**
6. **Notifications: Users and admins receive notifications for complaint updates or required actions.**
7. **Reports Generation: Admins can generate reports for monitoring, analytics, and decision-making.**

4.2 Data Flow Diagrams

A Data Flow Diagram (DFD) represents the flow of data within the system, showing how inputs are transformed into outputs through processes.

1. **DFD (Level 0)**

   The Level-0 DFD (context diagram) provides an abstract view of the system as a single process with its interaction with external entities.

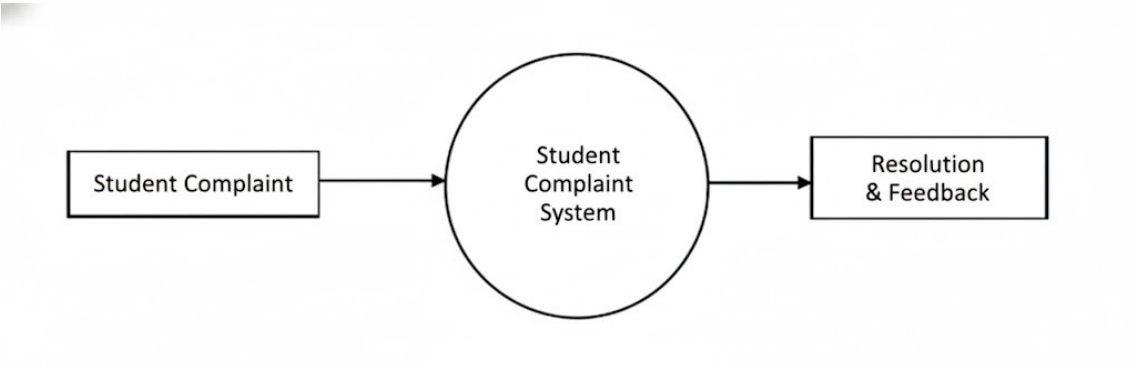


Figure 4.2: Level 0 Data Flow Diagram for Complaint Management System

Description: Users submit complaints and receive status updates. Administrators receive complaints as input and generate reports as output.

2. DFD (Level 1)

Level-1 DFD decomposes the system into multiple processes to show major functionalities in detail.

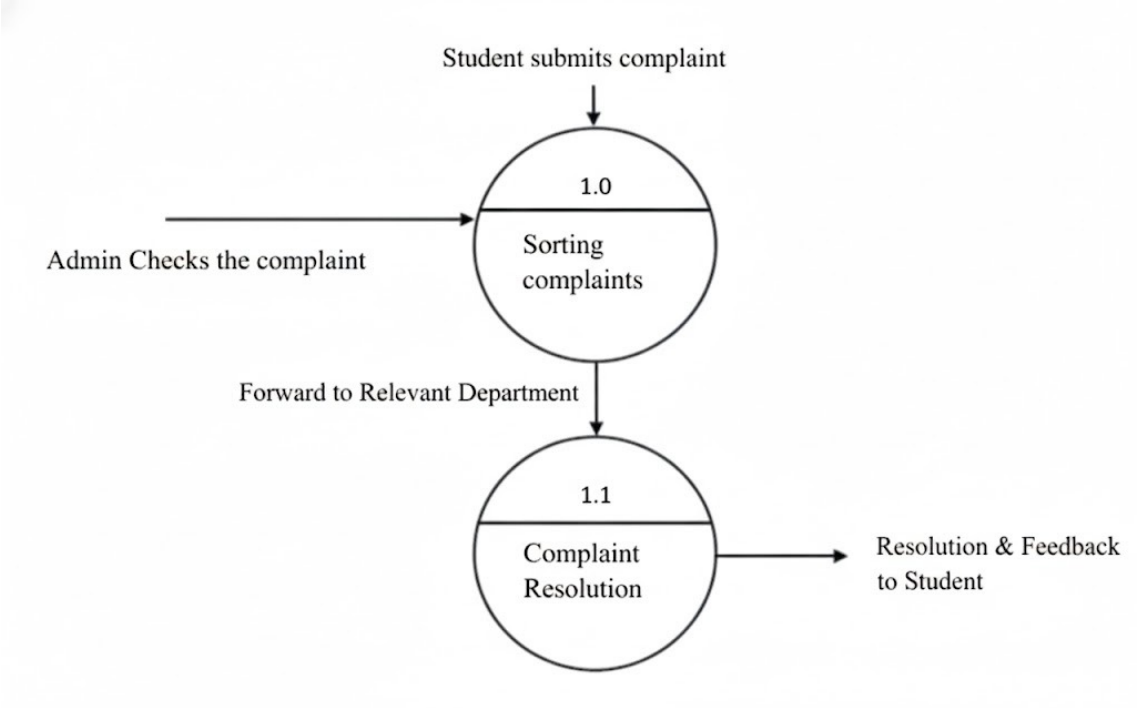


Figure 4.3: Level 1 Data Flow Diagram for Complaint Management System

Processes include: - User registration/login - Complaint submission - Admin review and assignment - Status tracking - Report generation

3. DFD (Level 2)

Level-2 DFD provides detailed information about each process in Level-1, showing sub-processes and data stores.

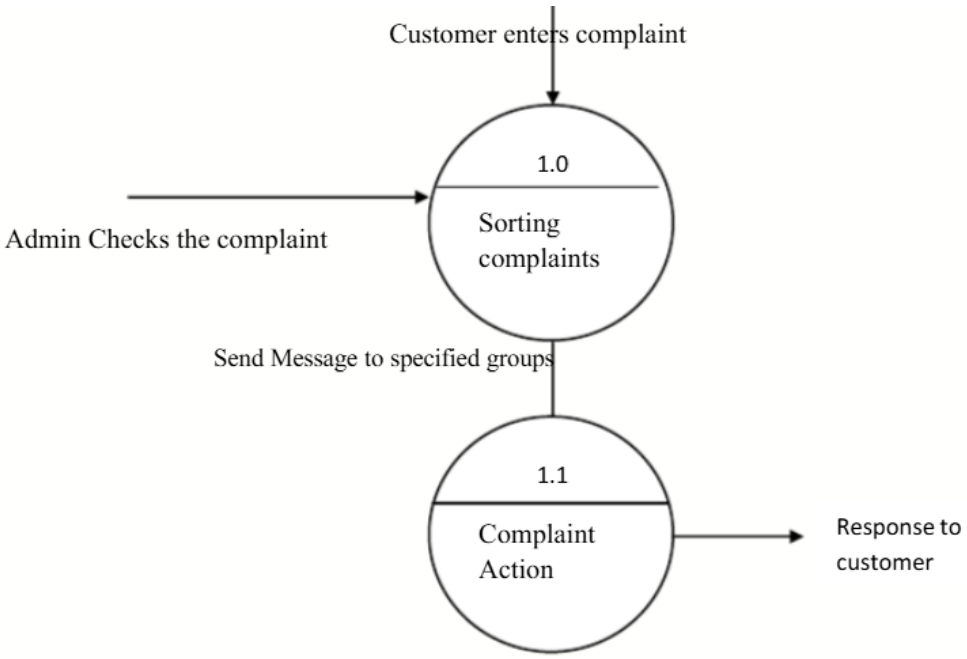


Figure 4.4: Level 2 Data Flow Diagram for Complaint Management System

Description: This level illustrates detailed operations such as validating anonymous complaint submission, updating complaint status, sending notifications, and generating reports for administrators.

4.3 UML Diagrams

4.3.1 Use Case Diagram

Figure 4.5 depicts the use case diagram, showing the interaction between actors and the system.

Actors in the use case diagram:

- **User (submits anonymous complaints)**
- **Administrator (manages complaints, generates reports)**

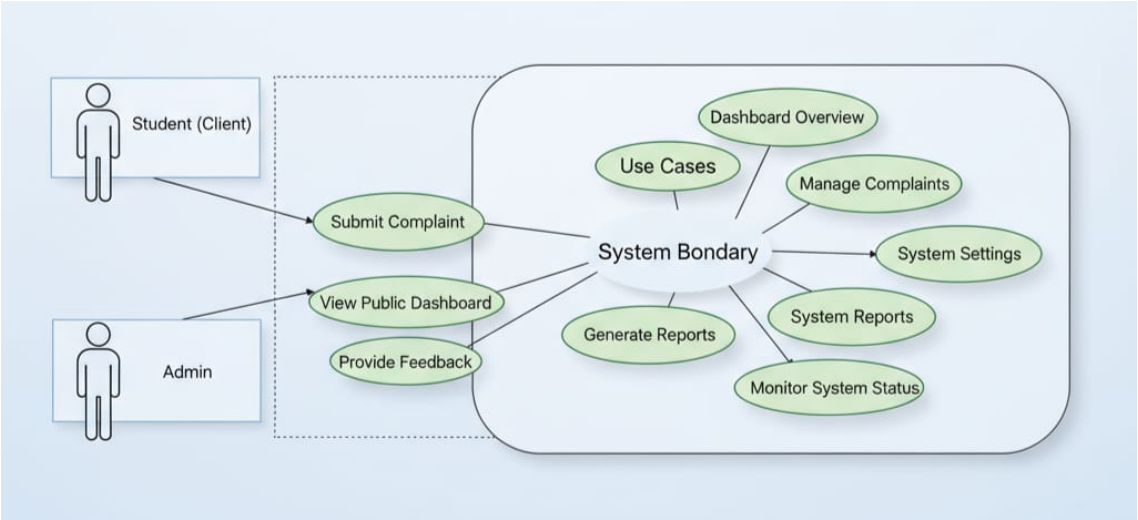


Figure 4.5: UML Use Case Diagram for Complaint Management System

4.3.2 Activity Diagram

Figure 4.6 illustrates the flow of control in the system and the steps involved in complaint submission and management. Activities follow a predefined flow based on user actions and system responses.

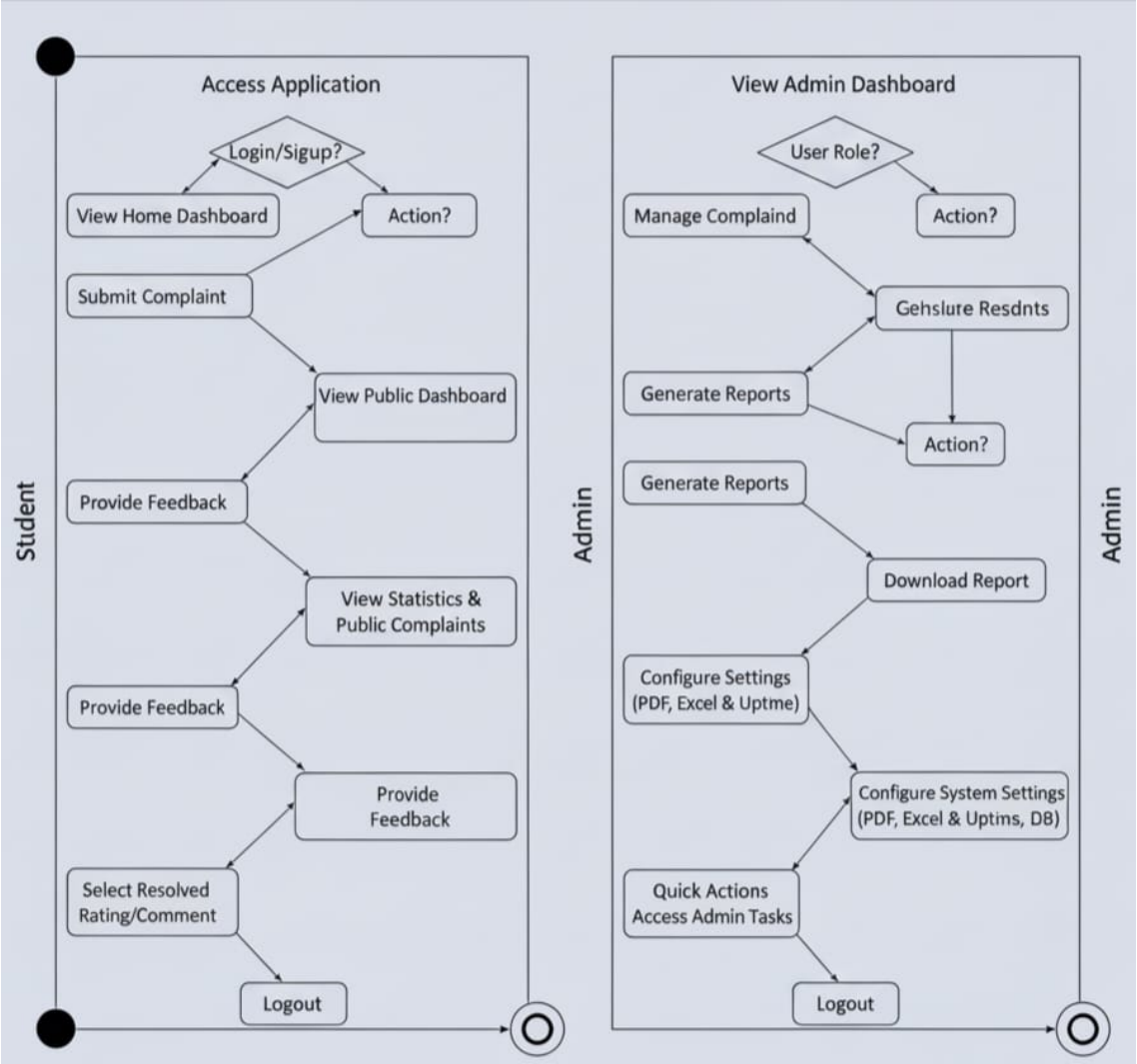


Figure 4.6: UML Activity Diagram for Complaint Management System

4.3.3 Sequence Diagram

The sequence diagram shows the sequential flow of interactions between active objects in the system. Key actors and objects include User, Web Portal, Database, and Administrator.

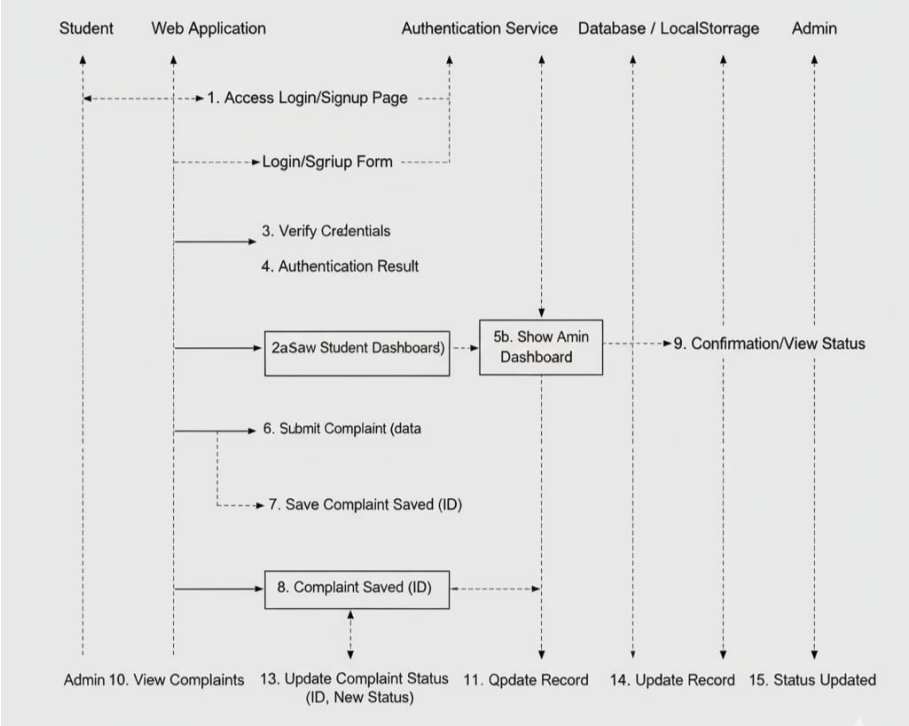


Figure 4.7: UML Sequence Diagram for Complaint Management System

4.3.4 Class Diagram

Figure 5.8 represents the class diagram, showing relationships between classes. Each class contains attributes and functions necessary for complaint submission, management, and reporting.

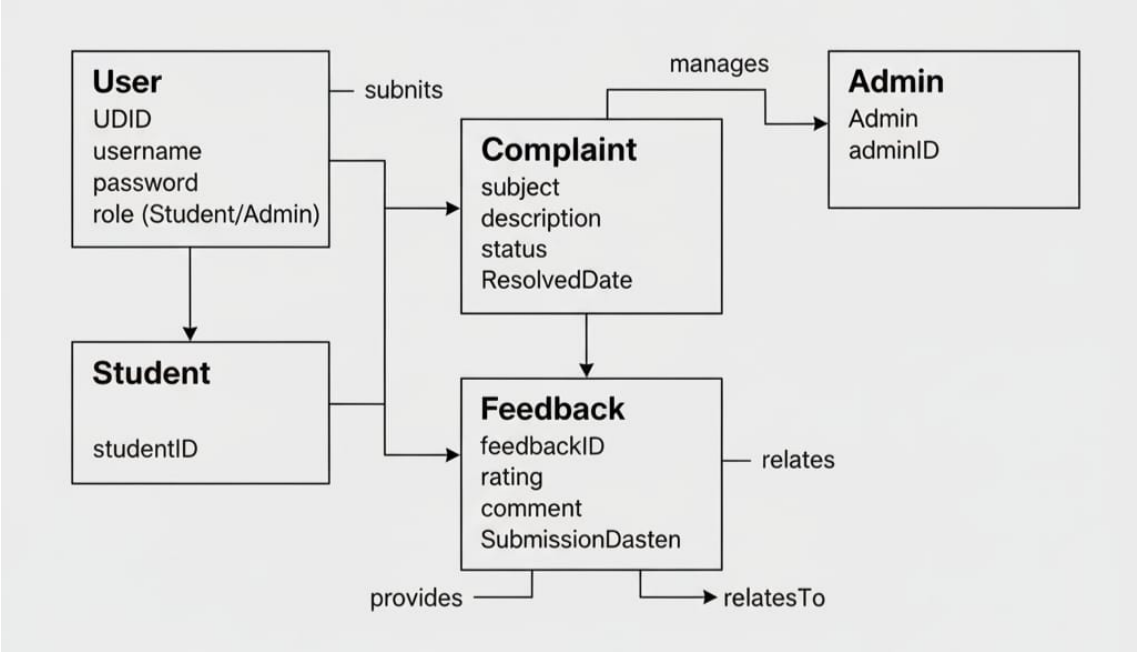


Figure 4.8: UML Class Diagram for Complaint Management System

Chapter 5

Other Specifications

5.1 Advantages

- **Allows anonymous complaint submission, ensuring user privacy and protection.**
- **Centralized system for tracking and managing complaints efficiently.**
- **Administrators can generate reports and monitor complaint resolution in real time.**
- **Notifications and status updates improve transparency and accountability.**

5.2 Limitations or Challenges

- **Ensuring anonymity while maintaining accountability for sensitive complaints.**
- **Security risks related to unauthorized access or data breaches.**
- **Proper categorization and assignment of complaints require intelligent handling.**

- **High system load may affect performance if many complaints are submitted simultaneously.**

5.3 Applications

- **Educational institutions for handling student complaints anonymously.**
- **Corporate or organizational grievance management systems.**
- **Government portals for public complaints and feedback.**
- **Any platform requiring secure and anonymous issue reporting.**

Chapter 6

Summary

This report studied various systems and research related to complaint management and anonymous reporting. The highlights and observations are reported in Chapter 2. The gap analysis has been conducted, based on which the problem statement is designed along with its objectives. The detailed plan of all the activities is mentioned in Section 1.5.

This report addresses the problem of managing complaints anonymously. The main goal of the system is to allow users to submit complaints without revealing their identity while enabling administrators to securely track, categorize, assign, and resolve these complaints. The system ensures data privacy, secure storage, and controlled access for administrators.

With the increasing number of complaints in organizations and institutions, an automated and secure complaint management system provides an efficient way to manage, monitor, and resolve complaints. The system also generates reports and updates to improve transparency and accountability. Overall, the individual components of the system—such as complaint submission, admin management, notifications, and reporting—are designed to function efficiently.

REFERENCES

[1] **M. Mawar, M. Assiddiq, A. Qashlim, Department of Information Systems, University Al Asyariah Mandar, Indonesia. *"The Complaint System Based on Whistleblowing Concept and Message Digest 5 Cryptographic Method for Regency Inspectorate Office in Polewali Mandar"* In *International Conference on Health Informatics (ICHI)*, 2021.**

[2] **A. Saha, A. Chaure, A. Aayush, A. Bodhe, T. D. Bhagat, Department of Information Technology, Sinhgad College of Engineering, Pune, India. *"Automated Complaint Classification and Routing Using NLP and Machine Learning"* In *International Journal for Innovative Research in Technology (IJIRT)*, 2024.**

[3] **D. R. Deepika, V. Shravan, R. Sai Harish, Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Sriperumbudur, India. *"Petition Analyzer: Grievance Management System with Chatbot"* In *International Journal of Creative Research Thoughts (IJCRT)*, 2025.**

[4] **P. Wadkar, A. Raorane, S. Bushra, S. Shedge, Department of Computer Engineering, Mumbai University, India. *"AI-Driven Complaint Management System"* In *4th International Conference on Advances in Science and Technology (ICAST)*, 2021.**

[5] **D. Hiremath, H. Patil, R. Patil, V. Hiremath, C. R. Shivanagi, Department of Computer Science, B. V. Bhoomaraddi College of Engineering and Technology, India.** ***"Public Online Complaint Registration and Management System"*** **In** ***International Journal of Scientific Development and Research (IJSDR)*****, 2024.**

[6] **K. K. Wagh Polytechnic, Nashik, Maharashtra, India.** ***"Anonymous Complaint System for Smart City"*** **In** ***International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*****, 2023.**

[7] **C. Venkatesh, H. Oberoi, A. K. Pandey, A. Goyal, N. Sikka, Department of Computer Science, India.** ***"RE-GrievanceAssist: Enhancing Customer Experience through ML-Powered Complaint Management"*** **In** ***arXiv preprint arXiv:2404.18963*****, 2024.**

[8] **P. Bhand, L. Chaudhari, N. Nannaware, B. Nandre, Department of Computer Engineering, India.** ***"Secure Complaint Resolver"*** **In** ***International Research Journal of Engineering and Technology (IRJET)*****, 2020.**

[9] **Anonymous Authors, Department of Computer Science, Hindusthan College of Engineering and Technology, Coimbatore, India.** ***"A Digital Governance Framework for Intelligent Complaint/Grievance Management System"*** **In** ***International Journal of Scientific Research and Engineering Development (IJSRED)*****, 2025.**

[10] **Anonymous Authors, Department of Information Technology, India.** ***"Privacy-Preserving Public Complaint Platform Using Certificateless Blind Signature and Cloud Backend"*** **In** ***International Journal for Innovative Research in Technology (IJIRT)*****, 2025.**

Appendix A

Appendix

Feasibility assessment of the complaint management system using satisfiability analysis and NP-Hard, NP-Complete, or P-type classifications based on modern algebra and relevant mathematical models.

Title:

Feasibility analysis of the Anonymous Complaint Management System using NP-Hard, NP-Complete, or P-type problem classification with modern algebra and mathematical modeling.

Theory:

1. **What is P?**
   - **P is the set of all decision problems that can be solved in polynomial time by a deterministic algorithm.**
   - **Problems that can be solved in polynomial time can also be verified in polynomial time.**
   - **Therefore, P is a subset of NP.**

2. **What is N?**
   - **The "N" in NP stands for Non-deterministic, which represents a theoretical computer that can explore multiple solutions simultaneously.**
   - **Such a system can solve certain problems faster than a standard deterministic computer, including problems that are not in P.**
   - **NP refers to problems that can be verified in polynomial time, even if finding the solution may require non-deterministic approaches.**
3. **What is NP?**
   - **NP is the class of decision problems where a proposed solution can be verified in polynomial time.**
   - **These problems may or may not be solvable in polynomial time, but verification is efficient.**
4. **Project Status -**
   - **The main challenge in the complaint management system is to ensure anonymous submission, secure storage, and efficient retrieval of complaints while maintaining confidentiality.**
   - **The feasibility analysis shows that the problem can be modeled as an NP-Complete problem due to constraints like anonymous access, encrypted storage, and real-time complaint processing.**
   - **Conceptual solutions include database encryption, secure authentication, and optimized data retrieval algorithms to handle complaints efficiently.**

Problem:

The main problem is to allow users to submit complaints anonymously while ensuring secure storage, proper categorization, and efficient retrieval of complaints by administrators. The system must also prevent unauthorized access and maintain confidentiality.

Solution:

During the Feasibility Study stage, the project requirements were analyzed, and the goals, constraints, and system specifications were identified. The solution includes:

- **Secure anonymous complaint submission for users.**
- **Encrypted storage of complaints in the database.**
- **Role-based access control for administrators to view, update, and manage complaints.**
- **Efficient retrieval and filtering of complaints to ensure quick response times.**

Based on the constraints of anonymity, security, and efficient real-time processing, this problem can be considered NP-Complete.

Set Theory :-

S = s, I, O, F, E, V

where

s = Start of program

I = I1

I1 = Complaint submitted by user anonymously,

O = O1, O2

O1 = Complaint stored securely in database

O2 = Complaint status updated and retrieved by administrator

F = F1, F2

F1 = Validate and categorize complaint

F2 = Assign and update complaint status

E = End of program

V = Success and failure conditions

Success if :

- **User can submit a complaint anonymously.**
- **Complaint is securely stored and encrypted.**
- **Administrator can view, manage, and resolve the complaint.**
- **System maintains data integrity and user confidentiality.**

Failure if :

- **More time consumption by the system during submission or retrieval.**
- **Hardware failure (e.g., server or database crash).**
- **Software failure (e.g., bug in the application logic, encryption failure).**
- **Improper network connection preventing access.**
- **Security breach or compromise of anonymous data.**

Space Complexity :

- **The space complexity depends on the number of complaints stored (n) and the data stored per complaint (e.g., text, attachments).**

- **More complaints, detailed logs, and attached files will increase the space complexity, likely $O(n)$ where n is the number of complaints.**

Time Complexity :

- **If the system has n complaint records:**
- **Best Case: Accessing a complaint by its unique ID (with database indexing) is $O(1)$ or $O(\log n)$.**
- **Worst Case: Searching for a complaint based on text content without proper indexing could be $O(n)$.**
- **Listing all complaints for an admin is $O(n)$.**