

**A**  
**Preliminary Project Report**  
**on**  
**Civic Complaints Registration Application**  
**for Citizens of Rural/Urban Areas.**

Submitted for the Course of BE in Computer Engineering by

Mr. Rohit Vasant Bava [B191204216]  
Mr. Tejas Arun Bhandare [B191204218]  
Mr. Ishan Sachin Ahirrao [B191204204]

Under the guidance of

**Prof. Sandeep.G. Shukla**



**Department of Computer Engineering**  
**Guru Gobind Singh College of Engineering and**  
**Research Center**

Nashik-422009

2023-24

# GURU GOBIND SINGH COLLEGE OF ENGINEERING AND RESEARCH CENTER

Nashik-422009

2023-2024

Department of Computer Engineering



## CERTIFICATE

This is to certify that the PRELIMINARY PROJECT REPORT entitled

# Civic Complaints Registration Application for Citizens of Rural/Urban Areas.

is submitted as partial fulfilment of the

Project Examination BE in Computer Engineering

BY

Mr. Rohit Vasant Bava [B191204216]

Mr. Tejas Arun Bhandare [B191204218]

Mr. Ishan Sachin Ahirrao [B191204204]

-----  
Prof. S.G. Shukla

*Project Guide*

-----  
Prof. P. K. Bachhav

Prof. P. C. Patil

*Project Coordinator*

-----  
Prof. S. G. Shukla

*Head of the Department*

-----  
Dr. N. G. Nikam

*Principal*

# SAVITRIBAI PHULE PUNE UNIVERSITY



## CERTIFICATE

This is to certify that,

Mr. Tejas Arun Bhandare [B191204218]

Mr. Rohit Vasant Bava [B191204216]

Mr. Ishan Sachin Ahirrao [B191204204]

of BE in Computer Engineering was examined in the

Project Examination entitled

**Civic complaints registration application for  
citizens of Rural/Urban areas.**

**on 21/11/2023**

**At**

**Department of Computer Engineering**

**GURU GOBIND SINGH COLLEGE OF ENGINEERING**

**AND RESEARCH CENTER**

**Nashik-422009**

**2023-2024**

-----  
*Internal Examiner*

-----  
*External Examiner*

# Acknowledgement

It is a great pleasure to acknowledge those who extended their support, and contributed time for this project work so far.

While the project is still in progress, I would like to thank my project guide **Mr. Sandeep.G. Shukla**, for his valuable and skillful guidance, assessment and suggestions from time to time improved the quality of work in all respects. I would like to take this opportunity to express my deep sense of gratitude towards him, for his invaluable contribution.

I am also thankful to **Mr. Sandeep. G. Shukla**, Head of Computer Engineering Department for his timely guidance, inspiration and administrative support without which my work would not have been completed.

I am also thankful to the all staff members of Computer Engineering Department and Librarian, Guru Gobind Singh College of Engineering and Research Center, Nashik.

Also I would like to thank my colleagues and friends who helped me directly and indirectly during this course of work.

I look forward to completing the project to the best of my ability and continuing to seek your support throughout the process.

**Mr. Rohit Vasant Bava**  
**Mr. Tejas Arun Bhandare**  
**Mr. Ishan Sachin Ahirrao**

# Abstract

Citizen engagement and technology usage are two emerging trends driven by smart city initiatives. Typically, citizens report issues, such as broken roads, garbage dumps, etc. through web portals and mobile apps, in order for the government authorities to take appropriate actions. Several mediums – text, image, audio, video – are used to report these issues. Through a user study with 13 citizens and 3 authorities, we found that image is the most preferred medium to report civic issues. We aim Through this platform, we aim to achieve the following: Enable citizens to easily submit complaints, particularly via image-based submissions, Automate the complaint handling process, from Geo-fencing tagging to department allocation, Utilize AI, ML, and Data Science to enhance efficiency and accuracy in complaint resolution, Foster a sense of community engagement and participation in local governance. Method which implemented are Image Processing, Computer Vision and Adversarial Scene Graph Model: To capture and interpret complaint images, Geo-fencing: To precisely determine the location of each issue, AI and ML: To automatically categorize complaints and allocate them to the relevant departments, Data Science: To analyze patterns and trends, aiding in efficient resource allocation, User-Centric Design: Ensuring an intuitive and accessible interface for citizens. Citizens can report issues promptly through images, reducing reporting barriers. The Geo-fencing and automated department allocation result in faster response times. AI and ML technologies ensure accurate tagging and categorization of complaints. By harnessing the power of automation and technology, we enable citizens to actively participate in improving their local communities. This application not only streamlines the process of addressing complaints but also promotes transparency and civic engagement.

**Keywords:-** *Civic Issues, Image Processing, Complaint Redressing, Geo-tagging, Geo-fencing.*

# Abbreviation

Following are Abbreviations used in report:

Sr No.	Abbreviation	Full Form
1	AI	Artificial Intelligence
2	ML	Machine Learning
3	GPS	Global Positioning System
4	CNN	Convolutional Neural Network
5	RNN	Recurrent Neural Network
6	SVM	Support Vector Machine
7	NLP	Natural Language Processing
8	UI	User Interface
9	RAM	Random Access Memory
10	SSD	Solid State Drive
11	IDE	Integrated Development Environment
12	API	Application Programming Interface
13	OWASP ZAP	Open Web Application Security Project Zed Attack Proxy.
14	SRS	Software Requirement Specification
15	QA	Quality assurance Drive
16	GDPR	General Data Protection Regulation
17	HIPPA	Health Insurance Portability and Accountability Act
18	FAQs	Frequently Asked Questions
19	UML	Unified Modeling Language
20	SDLC	System Development Life Cycle

# List of Figures

1.1	Issue Reporting . . . . .	1
4.1	SDLC : Agile Model . . . . .	27
5.1	System Architecture . . . . .	30
5.2	Data Flow Diagram (Level 0) . . . . .	31
5.3	Data Flow Diagram (Level 1) . . . . .	32
5.4	Entity Relationship Diagram . . . . .	33
5.5	Activity Diagram for User . . . . .	34
5.6	Activity Diagram for Admin . . . . .	35
5.7	Activity Diagram for Super Admin . . . . .	36
5.8	Use Case Diagram . . . . .	37
5.9	Class Diagram . . . . .	38
5.10	Object Diagram . . . . .	39
5.11	Sequence Diagram . . . . .	40
5.12	Component Diagram . . . . .	41
7.1	Gantt Chart . . . . .	45
8.1	Plagiarism Certificate . . . . .	49

# List of Tables

7.1	System Implementation Plan . . . . .	44
-----	--------------------------------------	----



# Contents

<b>Acknowledgement</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Abbreviation</b>	<b>iii</b>
<b>List of Figures</b>	<b>iv</b>
<b>List of Tables</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Project Idea . . . . .	1
1.2 Motivation of the Project . . . . .	2
<b>2 Literature Survey</b>	<b>3</b>
2.1 Conclusion From Literature Survey . . . . .	5
<b>3 Problem Definition and Scope</b>	<b>6</b>
3.1 Problem Statement . . . . .	6
3.1.1 Goals and objectives . . . . .	6
3.1.2 Statement of scope . . . . .	7
3.2 Software Context . . . . .	7
3.3 Major Constraints . . . . .	8
3.4 Methodologies of Problem Solving and Efficiency Issues . . . . .	9
3.5 Scenario in Which Multi-Core, Embedded and Distributed Computing Used	10
3.6 Outcome . . . . .	11
3.7 Applications . . . . .	11
3.8 Hardware Resources Required . . . . .	12
3.9 Software Resources Required . . . . .	12

<b>4</b>	<b>Software Requirement Specification</b>	<b>14</b>
4.1	Introduction . . . . .	14
4.1.1	Purpose and Scope of Document . . . . .	14
4.1.2	Overview of responsibilities of Developer: . . . . .	14
4.2	Functional Requirements . . . . .	15
4.2.1	Complaint Registration using Image Processing and Geo Tagging for Automating auto fill issue details : . . . . .	15
4.2.2	Real-time Priority and Severity assignment using Image Processing and Computer Vision : . . . . .	16
4.2.3	Real-time Complaint Tracking : . . . . .	17
4.2.4	An escalation of complaints in the authority hierarchy : . . . . .	18
4.3	External Interface Requirements . . . . .	18
4.3.1	User Interfaces : . . . . .	18
4.3.2	Hardware Interfaces : . . . . .	19
4.3.3	Software Interfaces : . . . . .	19
4.3.4	Communication Interfaces : . . . . .	19
4.4	Non Functional Requirements . . . . .	19
4.4.1	Performance Requirements : . . . . .	19
4.4.2	Safety Requirements : . . . . .	20
4.4.3	Security Requirements : . . . . .	21
4.4.4	Availability : . . . . .	22
4.4.5	Functionality : . . . . .	23
4.4.6	Software Quality Attributes : . . . . .	24
4.5	System Requirements . . . . .	26
4.5.1	Database Requirements : . . . . .	26
4.6	Analysis Models: SDLC model . . . . .	27
4.6.1	Model Implementation : . . . . .	27
<b>5</b>	<b>System Design</b>	<b>30</b>
5.1	System Architecture . . . . .	30
5.2	Data Flow Diagrams . . . . .	31
5.3	Entity Relationship Diagram . . . . .	33
5.4	UML Diagrams . . . . .	34
5.4.1	Activity Diagrams . . . . .	34
5.4.2	Use Case Diagram . . . . .	37
5.4.3	Class Diagram . . . . .	38
5.4.4	Object Diagram . . . . .	39

5.4.5	Sequence Diagram . . . . .	40
5.4.6	Component Diagram . . . . .	41
<b>6</b>	<b>Other Specifications</b>	<b>42</b>
6.1	Advantages . . . . .	42
6.2	Limitations . . . . .	43
6.3	Applications . . . . .	43
<b>7</b>	<b>Project Plan</b>	<b>44</b>
7.1	System Implementation Plan . . . . .	44
7.2	Gantt Chart . . . . .	45
<b>8</b>	<b>Conclusion</b>	<b>46</b>
	<b>References</b>	<b>48</b>
	<b>Plagiarism Report For this Report</b>	<b>49</b>

# Chapter 1

## Introduction

### 1.1 Project Idea

”Civil Complaint Registering Application” designed for both urban and rural areas. By harnessing the power of Image Processing technology, we capture images of issues, automate key details such as address, pincode, and geotagging, and categorize complaints based on image analysis. This streamlined process ensures that each complaint is efficiently routed to the respective department for swift resolution, revolutionizing the way community issues are addressed.



Figure 1.1: Issue Reporting

## 1.2 Motivation of the Project

- **Addressing Civic Needs:** A Civil Complaint Registration Application that directly addresses civic needs by resolving issues in both rural and urban areas.
- **Positive Impact:** It promises to have a positive and lasting impact on both rural and urban areas, improving the quality of life for citizens and enhancing the efficiency of municipal services.
- **Community Engagement:** By facilitating citizen engagement and feedback, the project fosters a sense of community involvement and ownership.

## Chapter 2

# Literature Survey

Kumar Prasad, Ritesh Patil, Sagar Beldare, Prof. Anita Shinde **Civic Complaint Application under Smart City Project** Published on (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (1) ,ISSN: 0975-9646 2016, 402-406

In this paper, the paper proposes the development of an online web application, accessible through mobile devices, that allows citizens to report problems in their city to the relevant authorities. The application utilizes GPS sensors to pinpoint the exact location of the issue, and users can provide visual proof through images. The system then generates a comprehensive report that includes user-entered data, location information, and visual evidence, which is sent to a central server for notification to the concerned authority.[8]

Shanu Kumar, Anjali Singh, Mohit Jain **Adversarial Adaptation of Scene Graph Models for Understanding Civic Issues** Published on (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.ACM ISBN 978-1-4503-6674-8/19/05.

In this paper, the Adversarial graphs contain a set of objects and semantic relations between them, specifically representative of the underlying civic issue. The paper also releases two multi-modal datasets containing text and images to facilitate further research in this area. To address the lack of labeled training data for civic issues, the paper uses adversarial adaptation techniques to repurpose existing scene graph models for this new task. It proposes an approach to train the model using partially annotated and unpaired data.[4]

Pengfei Xu, Xiaojun Chang, Ling Guo, Poyao Huang, Xiaojiang Chen, Alexander G. Hauptmann **A Survey of Scene Graph: Generation and Application** Published on IEEE Trans. Neural Netw.Learn. Syst. 2020.

In this paper, Authors introducing scene graphs as data structures used to describe objects, their attributes, and relationships in a scene. It highlights the importance of scene graphs in various visual tasks, such as image retrieval, image/video captions, Visual Question Answering (VQA), image generation, and specific relationship detection. The paper categorizes scene graph generation methods based on the inference models for visual relationship detection and discusses the applications of scene graphs in specific visual tasks. It also highlights some of the challenges and future research directions in the field of scene graph generation.[3]

Anjaly Antony, Minla K.S **Civic Complaint Reporting Using Image Processing** Published on IJCRT — Volume 9, Issue 6 June 2021 — ISSN: 2320-2882

In this paper, the key advantages of this system include reducing the effort and time required for registering complaints, tracking the location of the issue using GPS, generating deadlines for issue resolution, and allowing citizens to vote on complaints to prioritize them. By involving citizens in the process, it encourages active participation in improving the city's infrastructure and services.[2]

Dnyanesh Walwadkar, Saurav Yadav **Smart Civic Issue Reporting System** Published on International Journal of Advanced Research in Science Communication and Technology DOI:10.48175/IJARSC-2659

In this paper, the architecture of the system involves multiple components, including the user interface (Android app), a central server, and the involvement of citizens and officials. The app allows users to register complaints and track their status. When a complaint is registered, it is analyzed using the hybrid CNN + RNN algorithm for severity, and the SVM-NLP model processes the text description of the issue. The GPS sensor on mobile devices pinpoints the location, and the camera is used to capture visual evidence. The system then sends the data and evidence to a central server, which notifies the relevant authorities for action.[1]

## **2.1 Conclusion From Literature Survey**

In conclusion, the literature survey demonstrates a growing interest in leveraging technology to empower citizens and enhance the efficiency of civic complaint reporting and resolution. These systems offer the potential to improve urban infrastructure and services by actively involving citizens in the process and providing authorities with the necessary data and evidence to address civic issues promptly. Future research in this area can focus on further enhancing these systems and addressing specific challenges related to civic issue reporting and resolution in both urban and rural areas.



# Chapter 3

## Problem Definition and Scope

### 3.1 Problem Statement

The current process of handling Civic Complaints in Urban and Rural areas is marred by inefficiencies and lack of transparency. Citizens face challenges in reporting issues, and government departments often struggle to allocate resources effectively. To address these issues, we propose the development of a "Civic Complaint Registering Application" powered by Image Processing. The problem at hand is the need for a streamlined, automated system that can capture issue images, extract essential information, and categorize complaints, thereby enabling efficient routing to the respective departments for resolution.

#### 3.1.1 Goals and objectives

- **Enhance civic engagement:** Enhances civic engagement by providing an accessible platform for citizens to report and track complaints, fostering a stronger sense of community involvement and shared responsibility for addressing local issues.
- **Improve administrative efficiency:** Improves administrative efficiency by automating complaint handling, reducing manual workloads, and ensuring rapid complaint routing to the appropriate departments, resulting in streamlined operations, cost savings, and quicker issue resolution.
- **Contribute to the overall betterment of communities:** Contributes to the overall betterment of communities by creating cleaner, safer, and more responsive environments, thus improving the quality of life and citizen satisfaction.
- **Streamline Complaint Reporting:** Develop a user-friendly platform that simplifies the process of reporting civic issues, enabling citizens from all backgrounds,

including those in rural areas, to submit complaints effortlessly.

- **Automated Complaint Evaluation:** Implement image processing and machine learning algorithms to automatically categorize and assess the severity of complaints based on uploaded images, allowing for faster and more accurate issue identification.
- **Priority Assignment:** Utilize machine learning models to prioritize complaints based on factors such as severity, location, historical data, and potential safety hazards, ensuring that critical issues receive immediate attention.

### 3.1.2 Statement of scope

- **Data-Driven Decision Making:** Collect, Analyze, and Visualize complaint data to provide local authorities with valuable insights for better resource allocation, urban planning, and evidence-based decision-making.
- **Transparency and Accountability:** Transparency in local governance by providing real-time updates to citizens on the status of their complaints, ensuring they remain informed throughout the resolution process.
- **Collaboration with Local Authorities:** Collaborate closely with municipal authorities.
- **User Adoption and Community Impact:** With a focus on reaching underserved areas, to empower citizens and create a positive impact on their communities.

## 3.2 Software Context

The software context of this project involves the development of a user-friendly mobile and web application designed to streamline the process of registering and addressing civil complaints in both urban and rural areas. It encompasses several essential components, including the integration of image processing, machine learning, and AI technologies to automate complaint categorization based on submitted images. Geotagging and geofencing features are crucial for associating complaints with specific locations, ensuring they are routed to the relevant government departments for resolution. The software context also emphasizes robust security measures, regulatory compliance, and the need for efficient data management, as well as user-friendly interfaces for citizens and administrative personnel. The project aims to enhance civic engagement, improve administrative

efficiency, and ultimately contribute to the overall betterment of communities by fostering transparency and prompt issue resolution. Additionally, the application should provide valuable reporting and analytics tools for assessing system performance and user engagement while complying with relevant legal and regulatory requirements.

### 3.3 Major Constraints

1. **User Interface (UI):** The application's user interface plays a critical role in ensuring ease of use and accessibility for citizens, administrators, and super administrators. A user-friendly interface should be designed for mobile devices and web browsers, allowing users to register complaints, track their status, and access administrative functionalities.
2. **Mobile and Web Development:** To cater to a diverse user base, the project will involve both mobile and web development. The mobile app should be available on major platforms (iOS and Android), while the web application should be accessible via standard web browsers.
3. **Database Management:** A robust database management system is crucial for storing complaint data, user information, and system configurations. The database should be designed to handle large volumes of data efficiently and securely.
4. **Image Processing:** The heart of the project, image processing technology, is required for capturing and analyzing complaint images. Integration of image recognition and processing libraries or APIs is essential to extract information and classify the nature of complaints based on the images.
5. **Machine Learning and AI:** To automate the tagging and categorization of complaints, machine learning and artificial intelligence algorithms should be employed. These technologies enable the system to identify issues from images, assign them to the appropriate departments, and track resolution progress.
6. **Geotagging and Geofencing:** Geotagging and geofencing technologies are necessary to associate complaint locations with geographical coordinates. This ensures that complaints are routed to the correct department based on their geographical relevance and enables users to track complaints in real-time on a map.
7. **Security Measures:** Given the sensitive nature of complaint data, robust security measures should be implemented to protect user information and ensure the con-

fidentiality and integrity of the system. This includes secure user authentication, data encryption, and access controls.

8. **Integration with Government Departments:** The system should establish interfaces or APIs to integrate with various government departments, enabling automated complaint assignment and resource allocation. This requires collaboration and cooperation with local authorities.
9. **Feedback Mechanism:** A feedback mechanism should be integrated into the application to collect user feedback on the resolution process. This data can be valuable for system improvement and assessing user satisfaction.
10. **Reporting and Analytics:** The software should provide reporting and analytics tools for administrators and super administrators to assess system performance, complaint resolution times, and overall user engagement.
11. **Mobile Device Features:** Utilize device features such as GPS, cameras, and notifications to enhance the application's functionality and user experience.
12. **Scalability and Performance:** The application should be designed to handle increased user load and data as it scales. This requires a scalable architecture and optimization for performance to maintain responsiveness.
13. **Regulatory Compliance:** Ensure compliance with relevant legal and regulatory requirements, including data privacy laws, accessibility standards, and any local government regulations.
14. **Testing and Quality Assurance:** Rigorous testing and quality assurance processes should be in place to identify and rectify bugs, performance issues, and security vulnerabilities.
15. **User Support and Training:** Provide user support and training materials to ensure that citizens, administrators, and super administrators can effectively use the application.

### 3.4 Methodologies of Problem Solving and Efficiency Issues

1. **Agile Development:** Implement an Agile methodology to allow for iterative development and quick adaptation to changing requirements, ensuring efficient problem-solving.

2. **Root Cause Analysis:** Use root cause analysis techniques to identify the underlying issues causing complaints, addressing them at their source for long-term efficiency.
3. **Continuous Improvement:** Establish a culture of continuous improvement to regularly assess and optimize the application's performance and efficiency in addressing complaints.
4. **Six Sigma:** Utilize Six Sigma methodologies to identify and reduce defects or errors in the complaint resolution process, ensuring high-quality outcomes.
5. **User Feedback Loop:** Establish a feedback loop with users to gather insights and address any issues promptly, ensuring a user-centric approach to problem-solving.
6. **Data Analytics:** Leverage data analytics to identify trends and patterns in complaint data, allowing for proactive issue resolution and resource allocation.
7. **Process Automation:** Implement automation where possible to reduce manual tasks and enhance efficiency in complaint registration, routing, and resolution.

### 3.5 Scenario in Which Multi-Core, Embedded and Distributed Computing Used

#### 1. Multi-Core Computing:

- **Parallel Processing:** Multi-core processors can be employed to handle concurrent tasks within the application. For instance, while one core manages user authentication, another core can process incoming complaints, thus improving the system's responsiveness.
- **Optimized Image Processing:** Multi-core architecture can be leveraged to speed up image processing tasks, such as complaint image analysis, geo-tagging, and categorization. This parallel processing enhances the application's efficiency in handling a large volume of image-based complaints.

#### 2. Embedded Computing:

- **Efficient Mobile App Development:** Since the application needs to run on mobile devices, embedded computing technology is vital for creating a compact and power-efficient mobile app. This ensures that users can easily access the system while conserving their device's battery life.

### 3. Distributed Computing:

- **Scalability and Load Distribution:** Distributed computing can be used to deploy the application across multiple servers or cloud resources. This allows for efficient load distribution, ensuring that the application can handle a high volume of complaints and users simultaneously.
- **Fault Tolerance:** In a distributed environment, fault tolerance mechanisms can be implemented to ensure that the application remains operational even in the event of server failures, guaranteeing uninterrupted service for users.

## 3.6 Outcome

1. **Citizen Empowerment:** Enhanced civic engagement and active participation in community improvement.
2. **Efficiency:** Streamlined complaint handling and resource allocation, reducing administrative workload.
3. **Transparency:** Real-time tracking, accountability, and a culture of openness in local governance.
4. **Cost Savings:** Reduced expenses through automation and optimized processes.
5. **Community Improvement:** Cleaner, safer environments, and better quality of life.
6. **Data Insights:** Valuable analytics for performance assessment and continuous improvement.
7. **User Satisfaction:** Convenient, responsive services leading to higher citizen satisfaction.
8. **Scalability:** Ability to handle growing demand and expand as needed.

## 3.7 Applications

1. **Urban and Rural Complaint Management:** Streamlining and automating the handling of complaints related to waste, infrastructure, and more in both urban and rural areas.

2. **Local Governance Enhancement:** Improving the efficiency and transparency of local governance through citizen engagement.
3. **Community Well-Being:** Contributing to cleaner, safer, and more responsive communities.
4. **Data-Driven Decision-Making:** Providing data insights for informed decision-making and performance assessment.

### 3.8 Hardware Resources Required

1. **Development Workstation:**
  - **Processor:** A multi-core processor (e.g., Intel Core i5 or equivalent) for faster code compilation and running development tools.
  - **Memory (RAM):** At least 8 GB of RAM to handle Java development environments, integrated development tools, and web browsers.
2. **Storage:**
  - **Storage Drive:** A solid-state drive (SSD) or traditional hard drive with at least 256 GB of storage for storing development files, IDEs, and project resources.
3. **Display:** A monitor with a resolution of at least 1366 x 768 pixels to comfortably work with development tools.
4. **Input Devices:** A standard keyboard and mouse for coding and navigating development environments.

### 3.9 Software Resources Required

1. **Integrated Development Environment (IDE):**
  - **Java Development Kit (JDK):** The latest version of the JDK for Java development.
  - **Integrated Development Environment:** Popular Java IDEs include Eclipse, IntelliJ IDEA, and NetBeans.
2. **Version Control:** Version Control System: Git for source code version control, and a Git hosting service like GitHub or GitLab.

3. **Database Management:** Database Management System: MySQL, PostgreSQL, or another relational database system to store complaint data.
4. **Image Processing:** Image Processing Libraries: Java libraries for image processing, such as OpenCV or Java Advanced Imaging (JAI).
5. **Geo Tagging and Geo Fencing:** Geospatial Libraries: Libraries or APIs for geotagging and geofencing, like Google Maps API or Leaflet for web mapping.
6. **Machine Learning:** Machine Learning Libraries: Java-based ML libraries like TensorFlow or Weka for complaint image analysis.
7. **Adversarial Scene Graph Module:** Specific module or libraries that incorporate adversarial scene graph techniques, tailored to your project's requirements.
8. **Mobile Development (Optional):** Mobile Development Tools: Android Studio for Android app development.
9. **Development and Debugging Tools:** Code editors, debugging tools, and plugins or extensions specific to your chosen IDE.
10. **Security Tools:** Security testing tools to ensure the application's security, such as OWASP ZAP or Nessus.
11. **Quality Assurance and Testing:** Testing frameworks, such as JUnit for unit testing and Selenium for automated testing.
12. **Deployment and Continuous Integration:** Tools for deploying the application and setting up continuous integration and continuous deployment (CI/CD) pipelines, such as Jenkins or Travis CI.
13. **Cloud Services (Optional):** If you plan to host your application in the cloud, consider cloud service providers like Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform.



# Chapter 4

## Software Requirement Specification

### 4.1 Introduction

#### 4.1.1 Purpose and Scope of Document

1. **Purpose :** The purpose of this Software Requirement Specification (SRS) is to define the detailed requirements for the development of the Smart Civic Issue Reporting System. It serves as a comprehensive guide for all stakeholders involved in the project, including developers, testers, project managers, and clients. The SRS outlines the functional and non-functional requirements, system architecture, and constraints for the successful development and deployment of the system.
2. **Scope :** This document defines the requirements and constraints for the development of the Smart Civic Issue Reporting System. It outlines the functional and non-functional requirements, system architecture, and constraints. The scope encompasses user registration, complaint reporting, image processing, natural language processing, GPS location services, server integration, and adherence to regulatory and technological constraints.

#### 4.1.2 Overview of responsibilities of Developer:

Developers play a pivotal role in the successful implementation of the Smart Civic Issue Reporting System. Their responsibilities include, but are not limited to:

- **System Development:** Developers are responsible for implementing the software system based on the requirements outlined in this SRS. This involves coding, testing, and integrating various components to create a functional application.

- **Adhering to Best Practices:** Developers should follow industry best practices, coding standards, and design patterns to ensure the system is robust, maintainable, and secure.
- **Integration:** They are responsible for integrating the system with external services and APIs, ensuring data exchange and communication.
- **Testing:** Developers should perform unit testing and collaborate with QA teams to identify and resolve any bugs or issues in the code.
- **Security:** Ensuring the security of user data and system functionality is a critical responsibility. Developers must implement appropriate security measures to protect against threats.
- **Documentation:** Developers should maintain technical documentation to facilitate system maintenance, updates, and troubleshooting.
- **Collaboration:** Effective collaboration with other team members, including testers, designers, and project managers, is essential for project success.
- **Compliance:** Developers need to ensure that the system complies with regulatory requirements and follows ethical development practices.

## 4.2 Functional Requirements

### 4.2.1 Complaint Registration using Image Processing and Geo Tagging for Automating auto fill issue details :

The system will provide a feature that allows users to register complaints related to civic issues using image processing capabilities. This feature will automate the process of filling in complaint details by extracting information from images of the issue.

#### Requirements :

- **Image Capture :** Users should be able to capture images of the civic issue directly within the mobile application.
- **Image Processing :** The system shall employ an image processing algorithm to analyze and extract relevant details from the captured images, such as the nature of the issue (e.g., road damage, street cleaning, etc.).

- **Geo Tagging :** The system shall utilize the Global Positioning System (GPS) sensor on the user's device to geo-tag the exact location of the issue at the time of registration.
- **Auto-fill Details :** Extracted details from the images shall be used to automatically populate complaint registration fields, reducing the need for manual data entry.
- **User Verification :** Users may need to verify or modify the auto-filled complaint details and provide additional information, if necessary.
- **Multi-Platform Support :** This functionality should be available on both Android and web platforms to ensure broader accessibility.

#### 4.2.2 Real-time Priority and Severity assignment using Image Processing and Computer Vision :

The system will provide a real-time feature that automatically assigns priority and severity levels to registered complaints using image processing and computer vision technologies. This assignment will enable the system to prioritize and address higher-severity issues promptly.

##### Requirements :

- **Image Analysis :** Upon complaint registration, the system shall analyze the images associated with the complaint using computer vision techniques.
- **Severity Detection :** The image analysis algorithm shall detect the severity of the issue, categorizing it into predefined severity levels (e.g., low, medium, high).
- **Priority Assignment :** Based on the detected severity level and other relevant criteria, the system shall assign a priority to the complaint (e.g., low, normal, high).
- **Real-time Processing :** The assignment of priority and severity levels shall occur in real-time to ensure immediate attention to high-severity issues.
- **User Review :** Users and administrators should have the ability to review and modify the assigned priority and severity levels if needed.
- **Notification :** The system shall notify relevant authorities about newly registered complaints, along with their assigned priority and severity levels.
- **Logging and Reporting :** The system shall maintain a log of priority and severity assignments for reporting and analysis purposes.

### 4.2.3 Real-time Complaint Tracking :

The system shall provide a real-time complaint tracking feature that allows users, authorities, and administrators to monitor the status and progress of registered complaints. This functionality ensures transparency and accountability in addressing civic issues.

#### **Requirements :**

- **Complaint Status Updates :** Users and authorities shall have access to real-time updates on the status of registered complaints, including whether they are under review, in progress, or resolved.
- **Geo-Tagging:** The system shall display the geographical location of the reported issues on an interactive map, allowing users to visually track the distribution of complaints across the city.
- **Communication Channel:** Users should be able to communicate with the relevant authorities and officials regarding their registered complaints through the system.
- **Administrator Access:** Administrators shall have a comprehensive view of all registered complaints, enabling them to monitor overall progress and identify trends.
- **Progress Updates :**For complaints that are under review or in progress, the system shall provide regular updates to the users on the steps taken and expected resolution times.
- **Filtering and Sorting:** Users and authorities should be able to filter and sort complaints based on various criteria, such as severity, priority, date of registration, and location.
- **Notification:** The system shall send notifications to users when there are updates on their registered complaints, ensuring they stay informed.
- **Data Analytics :** The system shall gather and analyze complaint tracking data to identify patterns and areas that require specific attention.
- **History Logging :** The system shall maintain a history log of complaint tracking data for auditing and reporting purposes.

#### 4.2.4 An escalation of complaints in the authority hierarchy :

The system shall support the automatic escalation of complaints within the officer hierarchy based on predefined criteria. This functionality ensures that high-severity or unresolved complaints are brought to the attention of higher-ranking officers for prompt resolution.

##### **Requirements :**

- **Escalation Criteria :** The system shall define specific criteria for escalating complaints, which may include severity, response time, or the nature of the complaint.
- **Automatic Escalation :** When a registered complaint meets the predefined escalation criteria, the system shall automatically escalate it to the next level of officers within the hierarchy.
- **Notification :** The officer to whom the complaint is escalated shall receive a notification indicating the escalated complaint, its details, and the reason for escalation.
- **Priority Handling :** Higher-ranking officers handling escalated complaints shall prioritize them, ensuring a swift response and resolution.
- **Tracking Escalations :** The system shall maintain a record of all escalated complaints, including the reason for escalation and the actions taken by higher-ranking officers.
- **User Notification :** Users who have registered the complaint that gets escalated shall be informed about the escalation and any changes in the handling of their complaint.
- **Audit Trail :** The system shall maintain an audit trail of all escalation activities for transparency and accountability.

### 4.3 External Interface Requirements

#### 4.3.1 User Interfaces :

- **Mobile Application :** The user interface for citizens will be through a mobile application running on Android devices. Citizens will utilize this interface to register complaints, provide details, and upload images.

- **Web Portal :** A web-based interface will be available for government officials and administrators to access the system, view complaints, and manage resources. It will be accessible through standard web browsers.

#### 4.3.2 Hardware Interfaces :

- **Smartphone Hardware :** The system will utilize the hardware capabilities of smartphones, including GPS sensors and cameras, to capture and transmit location data and images of reported issues.
- **Server Hardware :** The image processing server and computer vision processing servers will require suitable hardware configurations to efficiently handle image data and perform real-time processing.

#### 4.3.3 Software Interfaces :

- **Image Processing Server :** The system shall interface with one or more image processing servers equipped with software for analyzing and processing images. This interaction is crucial for automated complaint assessment.
- **Computer Vision Software :** The system shall interact with computer vision software for real-time priority and severity assignment. This software is responsible for processing live camera feeds and assessing the severity of reported complaints.

#### 4.3.4 Communication Interfaces :

- **Internet Connectivity :** The system will rely on standard internet communication protocols to transmit data between the mobile application, web portal, image processing server, and computer vision processing servers. Secure communication protocols will be employed to protect user data and maintain system integrity.

### 4.4 Non Functional Requirements

#### 4.4.1 Performance Requirements :

1. **Response Time :** The system should respond to user interactions within 2 seconds for complaint registration and tracking.
2. **Throughput :** The system should be able to handle a minimum of 1000 complaint registrations per hour.

3. **Scalability :** The system should be designed to handle an increasing number of users and complaints as the user base grows.
4. **Resource Utilization :** The application should not consume more than 80% of the available server resources during peak usage.
5. **Availability :** The system should have an uptime of at least 99% to ensure users can register complaints at any time.
6. **Data Retrieval Speed :** Complaint information should be retrieved from the database in under 1 second.
7. **Image Processing Time :** The image processing algorithm should analyze images and assign severity within 5 seconds.
8. **Accuracy :** The system should have an accuracy rate of 95% or higher in assigning complaint severity based on images.
9. **Security Response Time :** Security features, such as login and data encryption, should not add more than 1 second to the response time.
10. **Mobile Application Performance:** The mobile application should load on a standard smartphone within 5 seconds

#### 4.4.2 Safety Requirements :

1. **Data Privacy :** The system should comply with data privacy regulations and ensure the protection of personal information provided by users during complaint registration.
2. **User Authentication :** To ensure safety, the system should implement a secure user authentication mechanism to prevent unauthorized access.
3. **Backup and Recovery :** Regular data backups should be performed to prevent data loss in case of system failures. A disaster recovery plan should be in place.
4. **Security Testing :** The system should undergo regular security testing, including vulnerability assessments and penetration testing, to identify and address potential security issues.
5. **Image Processing Security :** Ensure that the image processing and computer vision components are protected from potential threats and intrusions.

6. **Error Handling** : The system should have effective error handling mechanisms to prevent the disclosure of sensitive information in error messages.
7. **Access Control** : Different user roles should have different levels of access to the system. Access should be restricted based on roles to prevent unauthorized actions.
8. **Secure Communication** : Data exchanged between the mobile application and the server should be encrypted to prevent eavesdropping.
9. **User Training** : Safety also involves training users on safe and responsible use of the application and the potential risks involved in sharing images and data.
10. **Compliance** : Ensure that the system complies with relevant safety standards and regulations, especially those related to image processing and data handling.

#### 4.4.3 Security Requirements :

1. **User Authentication** : Implement a robust user authentication system that ensures that only authorized users can access the system. This may include multi-factor authentication for added security.
2. **Data Encryption** : All data transmitted between the mobile application and the server should be encrypted to protect it from unauthorized access during transmission.
3. **Access Control** : Define user roles and permissions within the system to restrict access to sensitive information. Users should only be able to access the functionalities and data relevant to their roles.
4. **Secure Image Storage** : Images uploaded by users should be stored securely, and access to these images should be restricted to authorized personnel only.
5. **Vulnerability Assessment** : Regularly conduct vulnerability assessments and penetration testing to identify and address security weaknesses in the system.
6. **Audit Trail** : Maintain an audit trail of all user activities within the system to track and investigate any unauthorized or suspicious activities.
7. **Secure APIs** : If your system interacts with external systems or services, ensure that the APIs used are secure and properly authenticated.
8. **Password Policy** : Enforce a strong password policy that requires users to create complex passwords and change them at regular intervals.



9. **Session Management :** Implement secure session management to protect user sessions from hijacking or session fixation attacks.
10. **Security Updates :** Keep all system components and third-party libraries up to date with the latest security patches and updates.
11. **Incident Response Plan :** Develop an incident response plan to address security breaches or vulnerabilities promptly and effectively.
12. **Compliance:** Ensure that the system complies with relevant security standards and regulations, such as GDPR, HIPAA, or any industry-specific security requirements.
13. **Regular Training :** Provide security training and awareness programs for all users and staff to educate them about potential security risks and best practices.
14. **Secure Data Transmission :** Use secure communication protocols and encryption for data transmission between the mobile application and the server.

#### 4.4.4 Availability :

1. **Uptime Requirement :** The system should have a minimum uptime of 99.9% (or as per your organization's standards) to ensure it is available to users almost continuously.
2. **Redundancy :** Implement redundancy at critical points in the system to ensure that if one component fails, the system can continue to operate. This includes redundant servers, databases, and network connections.
3. **Disaster Recovery Plan :** Develop a comprehensive disaster recovery plan to minimize downtime in case of unexpected events, such as hardware failures, natural disasters, or cyber-attacks.
4. **Load Balancing :** Use load balancing techniques to distribute traffic evenly across multiple servers to prevent overload on any single server and ensure consistent performance.
5. **Scalability :** Ensure the system is designed to scale horizontally or vertically to accommodate increased loads as the number of users and data grows.
6. **Monitoring and Alerts :** Implement real-time monitoring of system components and set up alerts for system administrators to be notified of any performance or availability issues.

7. **Scheduled Maintenance** : Plan and communicate scheduled maintenance windows during off-peak hours to minimize disruption to users.
8. **High Availability Architecture** : Implement a high availability (HA) architecture with failover mechanisms, so if one server or component fails, another can take over seamlessly.
9. **Backup and Restore** : Regularly back up data and system configurations to enable quick restoration in case of data corruption or system failures.
10. **Content Delivery Networks (CDNs)** : Utilize CDNs to distribute content, such as images and videos, to reduce the load on your servers and improve availability and load times.
11. **Geographical Redundancy** : Consider setting up geographically redundant data centers or cloud regions to ensure availability in case of regional outages.
12. **Response Time** : Define maximum acceptable response times for critical functions and ensure the system consistently meets these requirements.
13. **User Access** : Implement measures to allow users to access the system even during maintenance or failure situations, such as a maintenance mode with limited functionality.

#### 4.4.5 Functionality :

1. **User Registration and Authentication** : The system should allow users to register, log in, and authenticate securely. User roles, such as citizens and government officials, should be defined.
2. **Complaint Submission** : Users should be able to submit complaints, including images and descriptions, through the mobile application. The system must validate and store these complaints.
3. **Image Processing** : Implement image processing to automatically extract information from images, such as location and severity, to assist in complaint handling.
4. **Complaint Tracking** : Users should be able to track the status and progress of their submitted complaints in real time.
5. **Priority and Severity Assignment** : The system should automatically assign priorities and severity levels to complaints based on image analysis and descriptions.

6. **Notification System** : Implement a notification system to inform users about the status of their complaints and updates on problem resolution.
7. **Escalation Mechanism** : Define a process for escalating complaints to higher authorities if they are not resolved within a specified time frame.
8. **Geo-Tagging** : Use GPS to geotag the exact location of the reported issue for efficient problem resolution.
9. **Data Storage and Retrieval** : Ensure efficient and secure storage and retrieval of complaint data, images, and user information.
10. **Reporting and Analytics** : Provide tools for generating reports and analytics to help government authorities monitor and improve the city's infrastructure.
11. **Security Measures** : Implement security features such as encryption, access control, and data protection to safeguard user data and system integrity.
12. **Admin Dashboard** : Government officials should have access to an administrative dashboard to manage and respond to complaints efficiently.
13. **Feedback Mechanism** : Allow users to provide feedback on the resolution of their complaints, ensuring continuous improvement.
14. **Multilingual Support** : Support multiple languages to cater to a diverse user base.
15. **User Support** : Offer user support options, including FAQs, chat support, or helplines, to assist users with issues and inquiries.
16. **Mobile Compatibility** : Ensure the mobile application is compatible with a variety of devices and operating systems.
17. **User Training** : Provide user training or guides for using the application effectively.

#### 4.4.6 Software Quality Attributes :

1. **Usability** : The system should be user-friendly, with an intuitive interface that is easy for citizens and government officials to navigate. It should provide clear instructions for complaint submission and tracking.

2. **Reliability** : The system should be highly reliable, ensuring that complaints are not lost or mishandled. It should also maintain data integrity and provide consistent performance.
3. **Scalability** : The platform should be scalable to accommodate an increasing number of users and complaints as the project grows. It should handle both small and large cities efficiently.
4. **Performance** : The system should be responsive, with fast complaint registration and tracking. It should handle concurrent users and large data volumes without slowing down.
5. **Availability** : The system should have minimal downtime and ensure that users can access complaint registration and tracking services 24/7.
6. **Security** : Implement robust security measures to protect user data, complaint information, and the system itself. This includes encryption, access control, and protection against cyber threats.
7. **Maintainability** : The software should be easy to maintain and update. Ensure that it is modular, well-documented, and can be enhanced as needed.
8. **Interoperability** : The system should be able to interface with other relevant systems or applications used by government agencies to streamline complaint resolution.
9. **Portability** : Ensure that the mobile application is compatible with various devices and operating systems to reach a broader user base.
10. **Compliance** : The system should comply with relevant laws and regulations regarding data privacy and complaint management.
11. **Performance Monitoring** : Implement tools for monitoring system performance, including response times and user satisfaction, to make continuous improvements.
12. **Fault Tolerance** : The system should handle unexpected failures gracefully, ensuring that it can recover from errors without losing data or compromising usability.

## 4.5 System Requirements

### 4.5.1 Database Requirements :

1. **Database Management System (DBMS):** Choose a reliable and scalable DBMS that can efficiently handle a large amount of data. Consider using popular options like MySQL, PostgreSQL, or MongoDB.
2. **Data Storage :** The system should be capable of storing various types of data, including user profiles, complaint details, images, and location information. Define the schema and data structure for each data type.
3. **Data Backup and Recovery :** Regularly back up the database to prevent data loss in case of system failures or disasters. Define a backup and recovery strategy to ensure data availability.
4. **Data Retention Policy :** Determine how long complaint data will be retained in the database. Define archiving and purging processes for managing historical data.
5. **Data Accessibility :** Define user roles and permissions for accessing and modifying data within the database. Users should have appropriate access levels based on their roles.
6. **Backup Storage :** Identify storage solutions for backup data, considering off-site or cloud storage options for disaster recovery.
7. **Testing and Quality Assurance :** Develop test cases and perform thorough testing of the database to verify its functionality, security, and performance.
8. **Documentation :** Maintain comprehensive documentation for the database schema, data models, and data flow diagrams.

## 4.6 Analysis Models: SDLC model

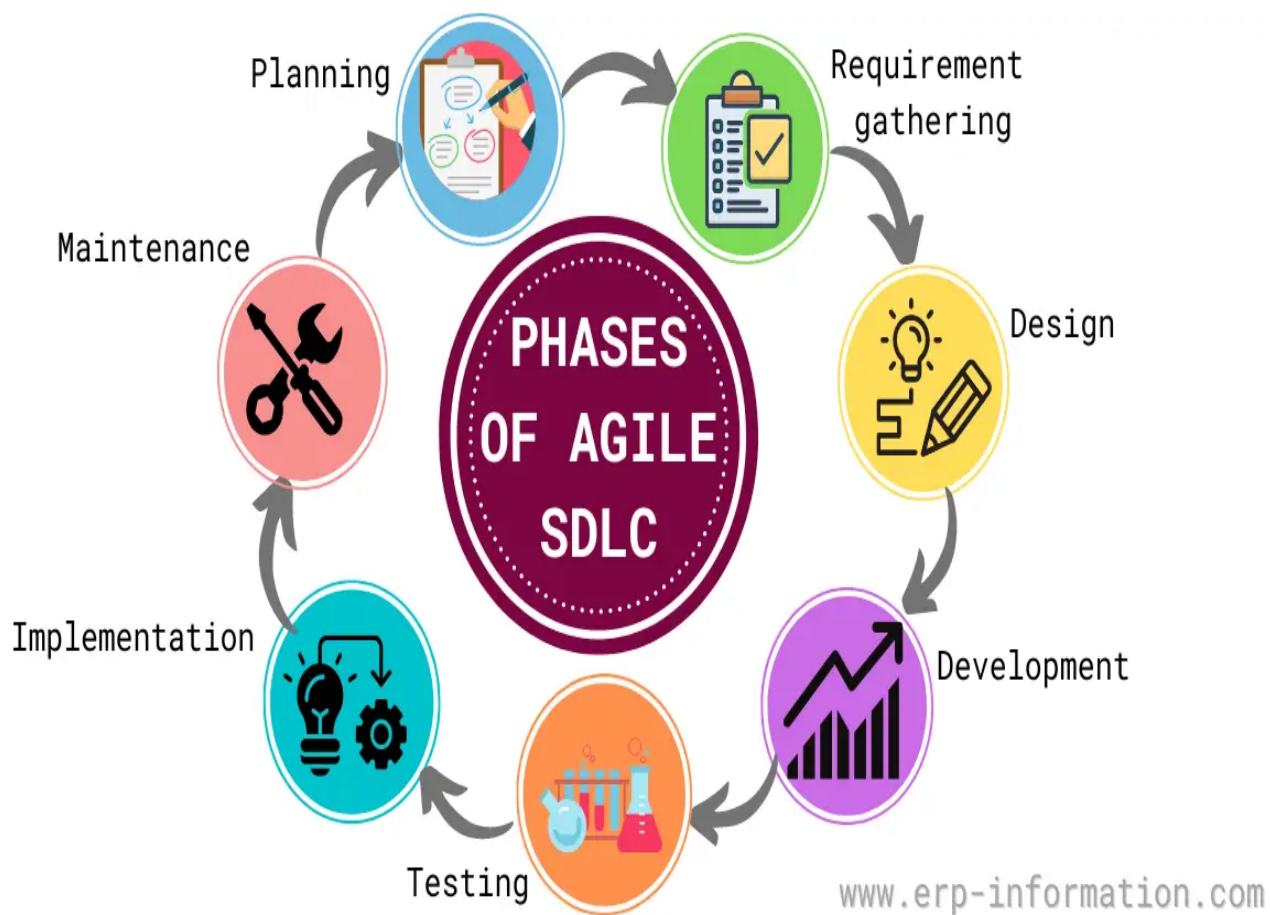


Figure 4.1: SDLC : Agile Model

### 4.6.1 Model Implementation :

1. **Project Initiation** :Begin by defining the project's vision and goals. Identify the core objectives, such as improving civic issue reporting and resolution through a user-friendly application. Assemble a cross-functional Agile team that includes developers, testers, designers, and a product owner or manager.
2. **Product Backlog Creation** :Develop a product backlog, which is a prioritized list of features and user stories. These should be derived from the project vision and

stakeholder input. The product backlog will serve as a dynamic repository of all the work that needs to be done. It should be continuously refined and reprioritized based on changing requirements and feedback.

3. **Sprint Planning** :Divide the project into time-boxed iterations called "sprints." Typically, sprints are 2-4 weeks long. For each sprint, select a subset of user stories from the product backlog based on priority and complexity. These stories should be small enough to be completed within the sprint. Set sprint goals that align with the project's overall objectives.
4. **Daily Stand-up Meetings** :Conduct daily stand-up meetings (or daily scrum) with the development team. Each team member shares their progress, challenges, and plans for the day. These short, focused meetings promote communication, identify impediments, and help keep the team aligned.
5. **Sprint Execution** :During the sprint, the development team works on the selected user stories. This includes design, development, testing, and potentially documentation. The team should collaborate closely, and developers and testers should work in parallel to ensure the work is completed efficiently.
6. **Continuous Integration and Testing** :Implement continuous integration and testing to ensure that code changes are integrated and tested frequently. This helps catch issues early and maintain product quality.
7. **Sprint Review** :At the end of each sprint, conduct a sprint review meeting with stakeholders. Demonstrate the completed user stories and receive feedback. This feedback is crucial for making adjustments and ensuring that the project remains aligned with user expectations.
8. **Sprint Retrospective** :Also at the end of each sprint, hold a sprint retrospective meeting with the development team. Reflect on what went well and what could be improved in the development process. Use the insights from retrospectives to make continuous improvements in the team's processes and collaboration.
9. **Iterative Development** :Continue with subsequent sprints, each one building on the work from the previous sprints. The process is iterative, allowing for flexibility and adaptation to changing requirements or priorities.
10. **User Involvement and Feedback** :Actively involve end-users or stakeholders throughout the project. Encourage their feedback and engage them in the develop-

ment process. Agile principles emphasize customer collaboration and responsiveness to change.

11. **Quality Assurance and Testing** :Integrate testing and quality assurance activities throughout the development process to ensure that each feature is thoroughly tested and meets quality standards.
12. **Incremental Deployment and Release** :Consider releasing the application incrementally. This means delivering valuable features to end-users as they become available, rather than waiting for the entire project to be completed.
13. **Continuous Improvement** :Agile promotes a culture of continuous improvement. Use the insights from sprint retrospectives and user feedback to refine the development process, enhance team collaboration, and improve the product.
14. **Adaptation to Change** :One of the core principles of Agile is to welcome changing requirements, even late in development. Be prepared to adapt to new insights and evolving needs.
15. **Project Monitoring and Metrics** :Use project management and development metrics to track progress, such as the velocity of the development team, sprint burn-down charts, and user story completion rates.



# Chapter 5

## System Design

### 5.1 System Architecture

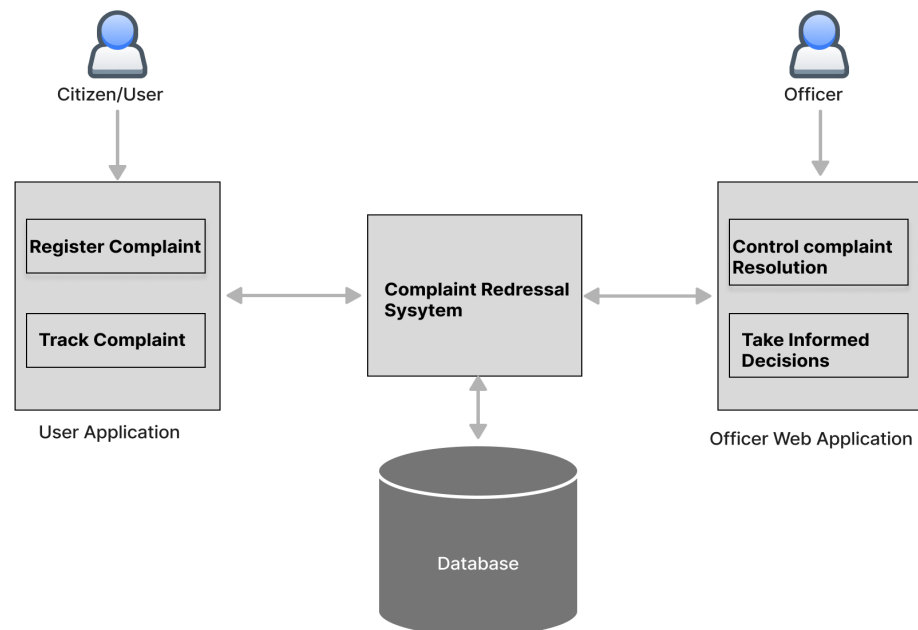


Figure 5.1: System Architecture

## 5.2 Data Flow Diagrams

A data flow diagram (DFD) maps out the flow of information for any process or system.

### Data Flow Diagram (Level 0)

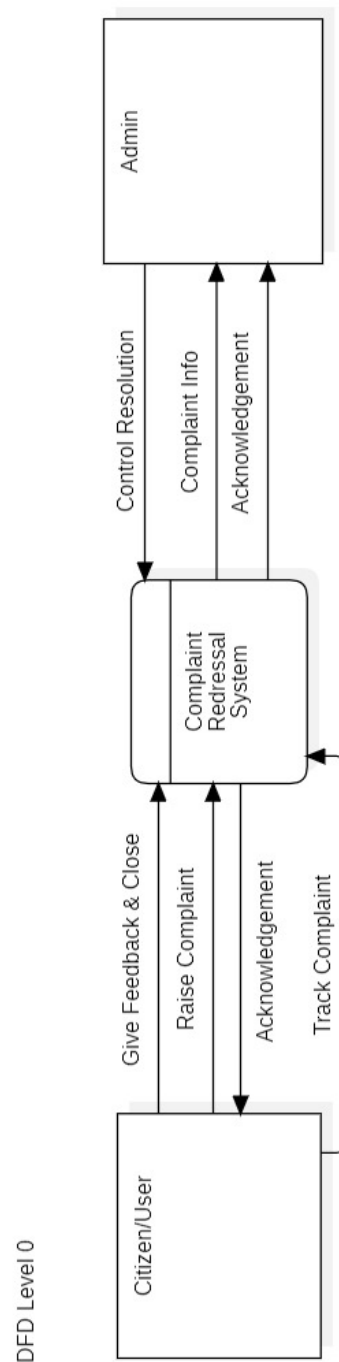


Figure 5.2: Data Flow Diagram (Level 0)

## Data Flow Diagram (Level 1)

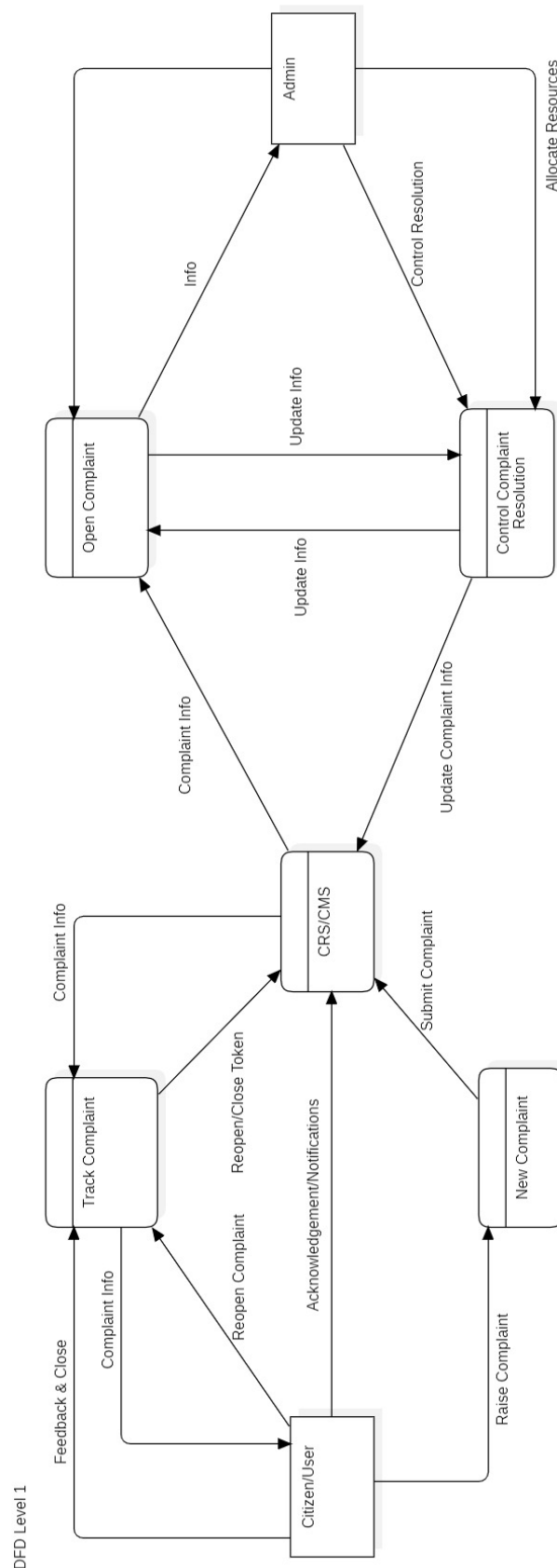


Figure 5.3: Data Flow Diagram (Level 1)  
Guru Gobind Singh College of Engineering and Research Center, Nashik

## 5.3 Entity Relationship Diagram

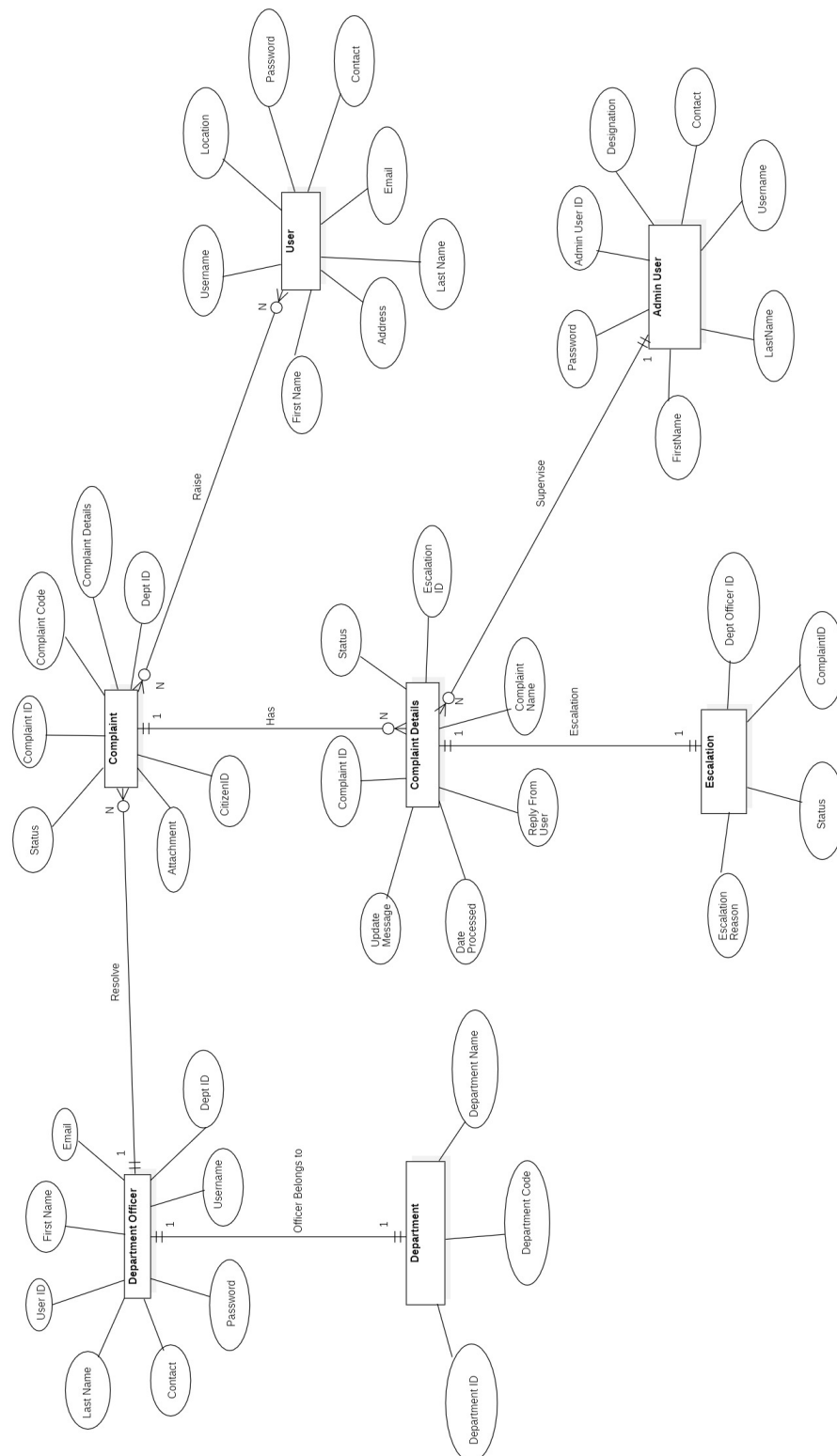


Figure 5.4: Entity Relationship Diagram

## 5.4 UML Diagrams

### 5.4.1 Activity Diagrams

We use Activity Diagrams to illustrate the flow of control in a system.

**Activity Diagram for User:**

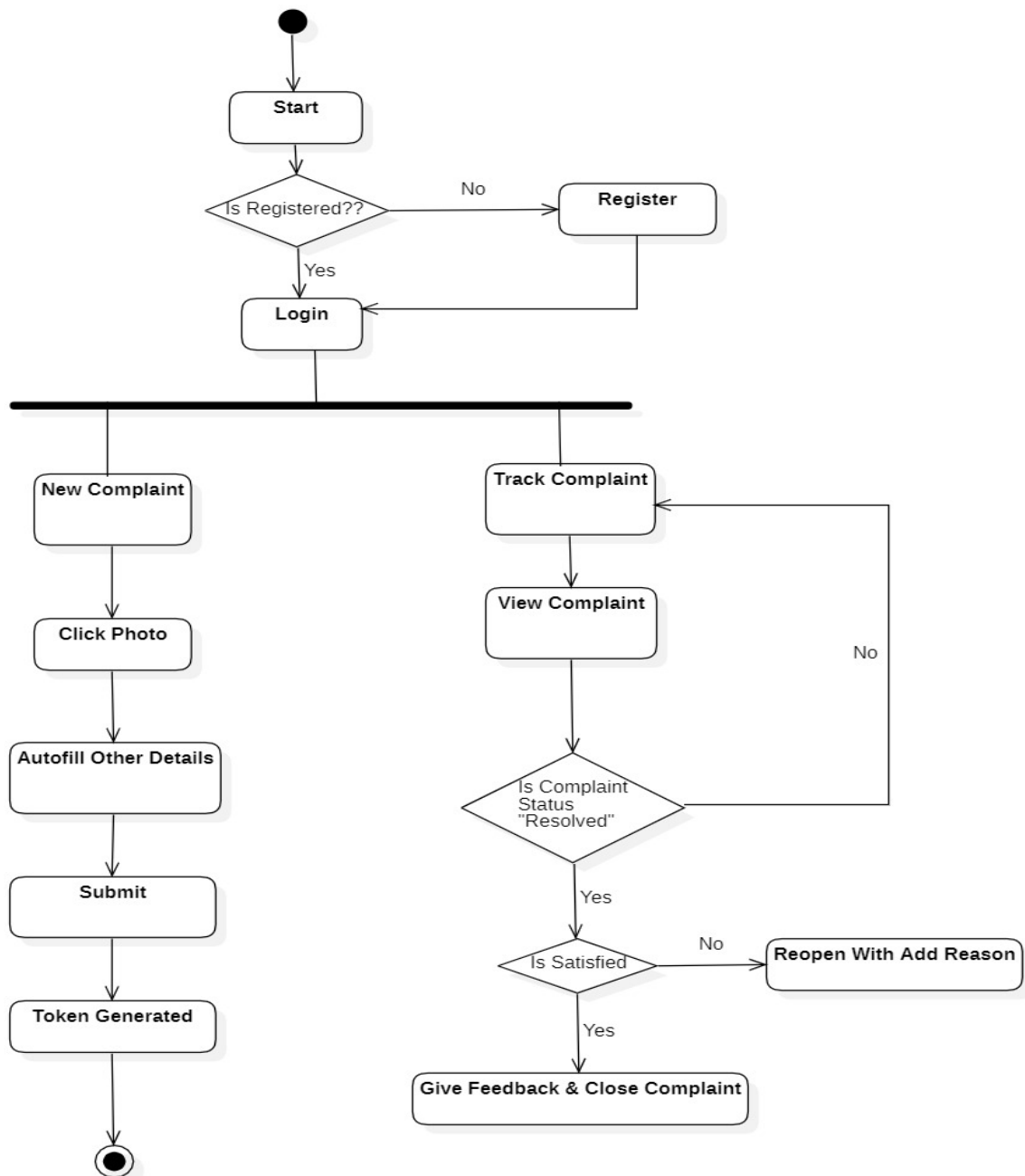


Figure 5.5: Activity Diagram for User

### Activity Diagram for Admin:

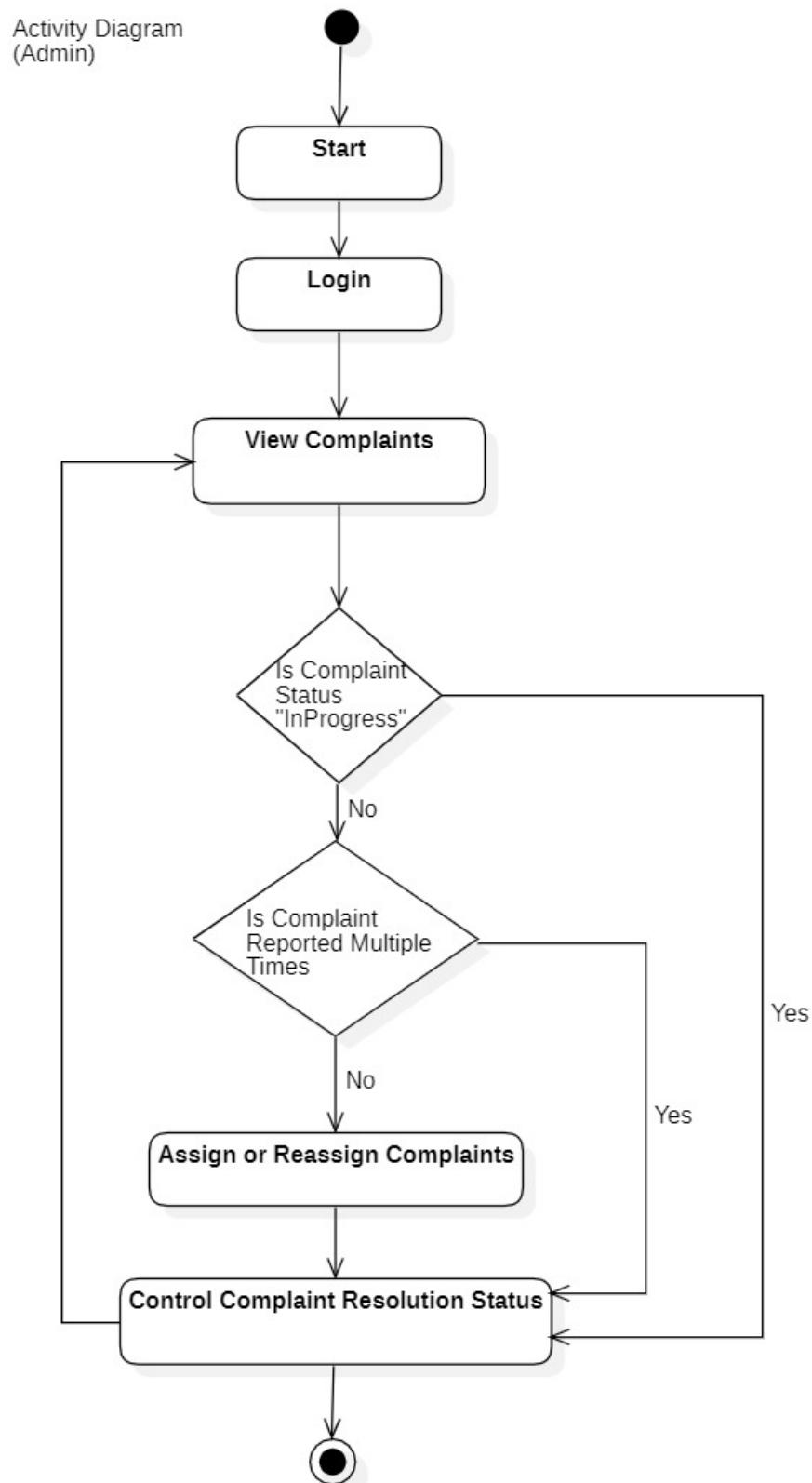


Figure 5.6: Activity Diagram for Admin

**Activity Diagram for Super Admin:**

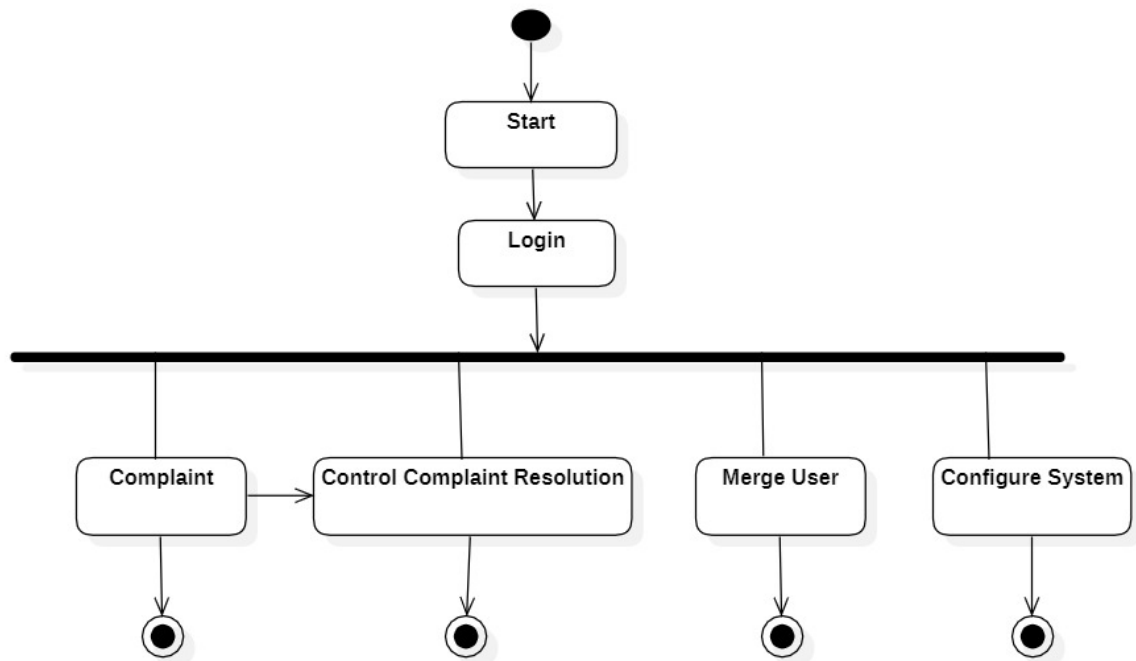


Figure 5.7: Activity Diagram for Super Admin

### 5.4.2 Use Case Diagram

Use Case Diagrams depict the functionality of a system or a part of a system.

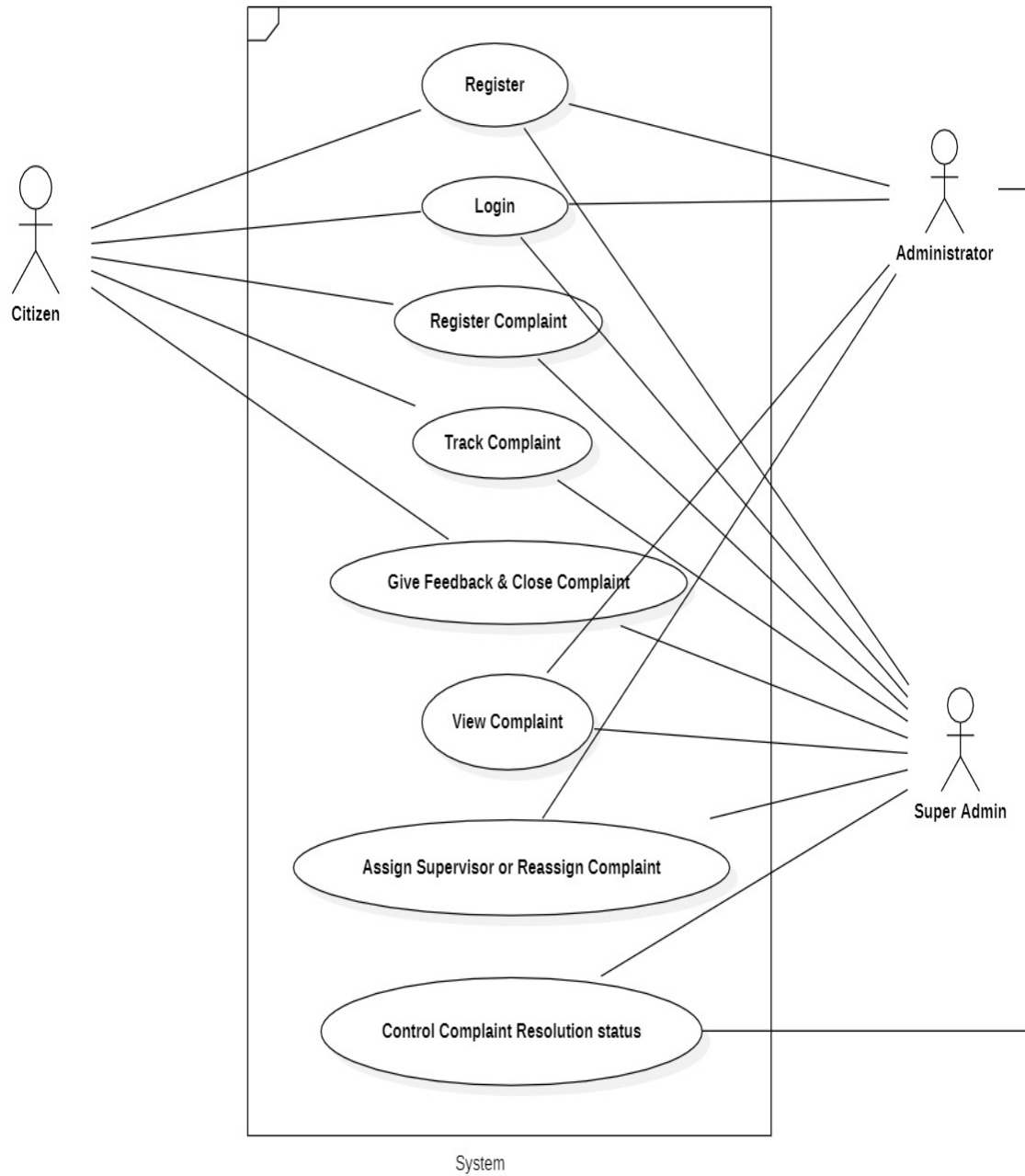


Figure 5.8: Use Case Diagram



### 5.4.3 Class Diagram

Explain class diagram

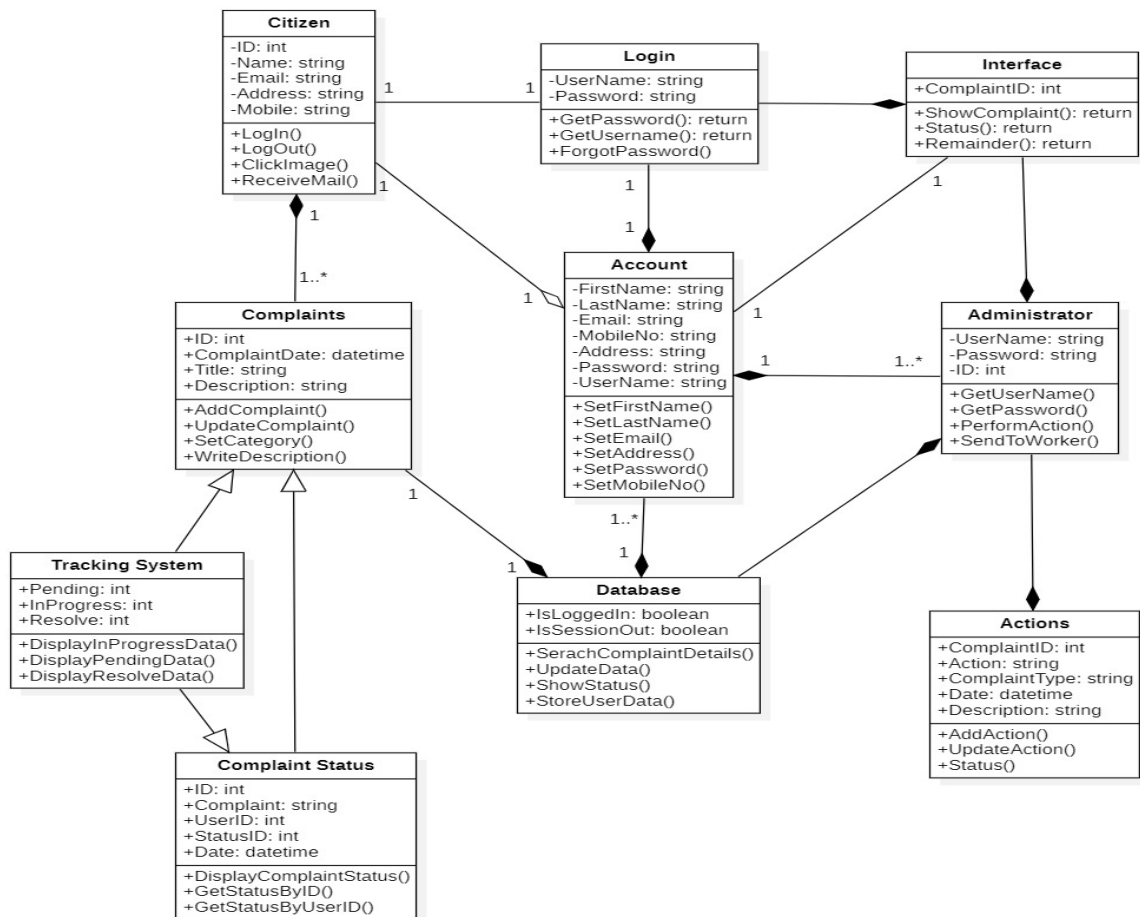


Figure 5.9: Class Diagram

#### 5.4.4 Object Diagram

Object Diagrams.

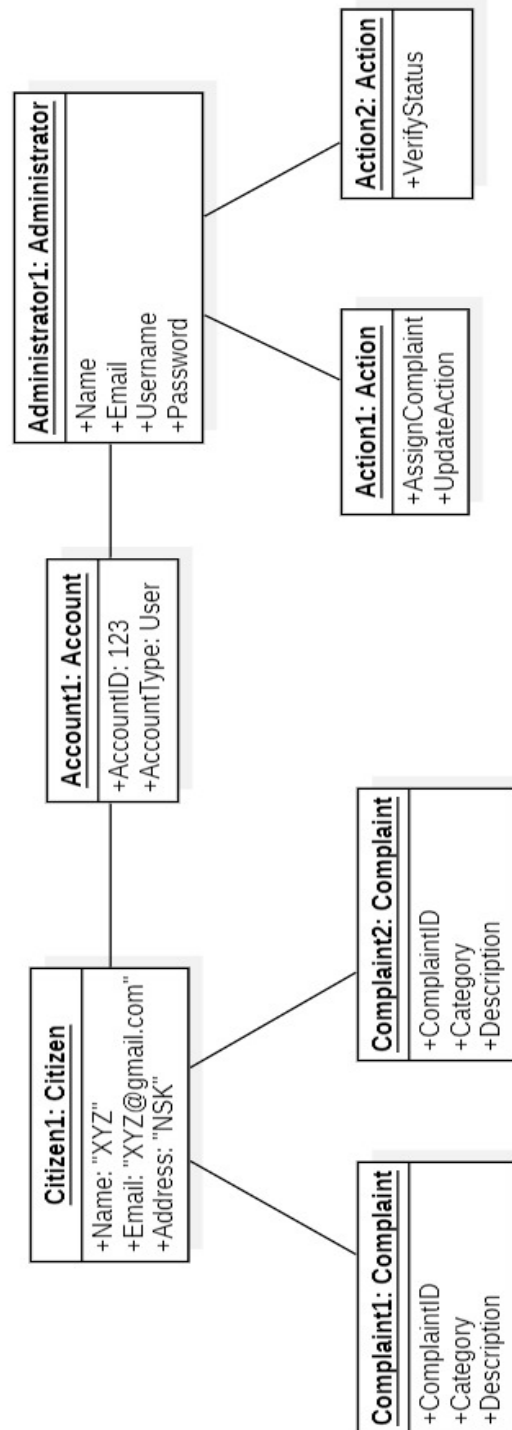


Figure 5.10: Object Diagram

### 5.4.5 Sequence Diagram

A sequence diagram depicts the interaction between objects in a sequential order i.e. the order in which these interactions take place.

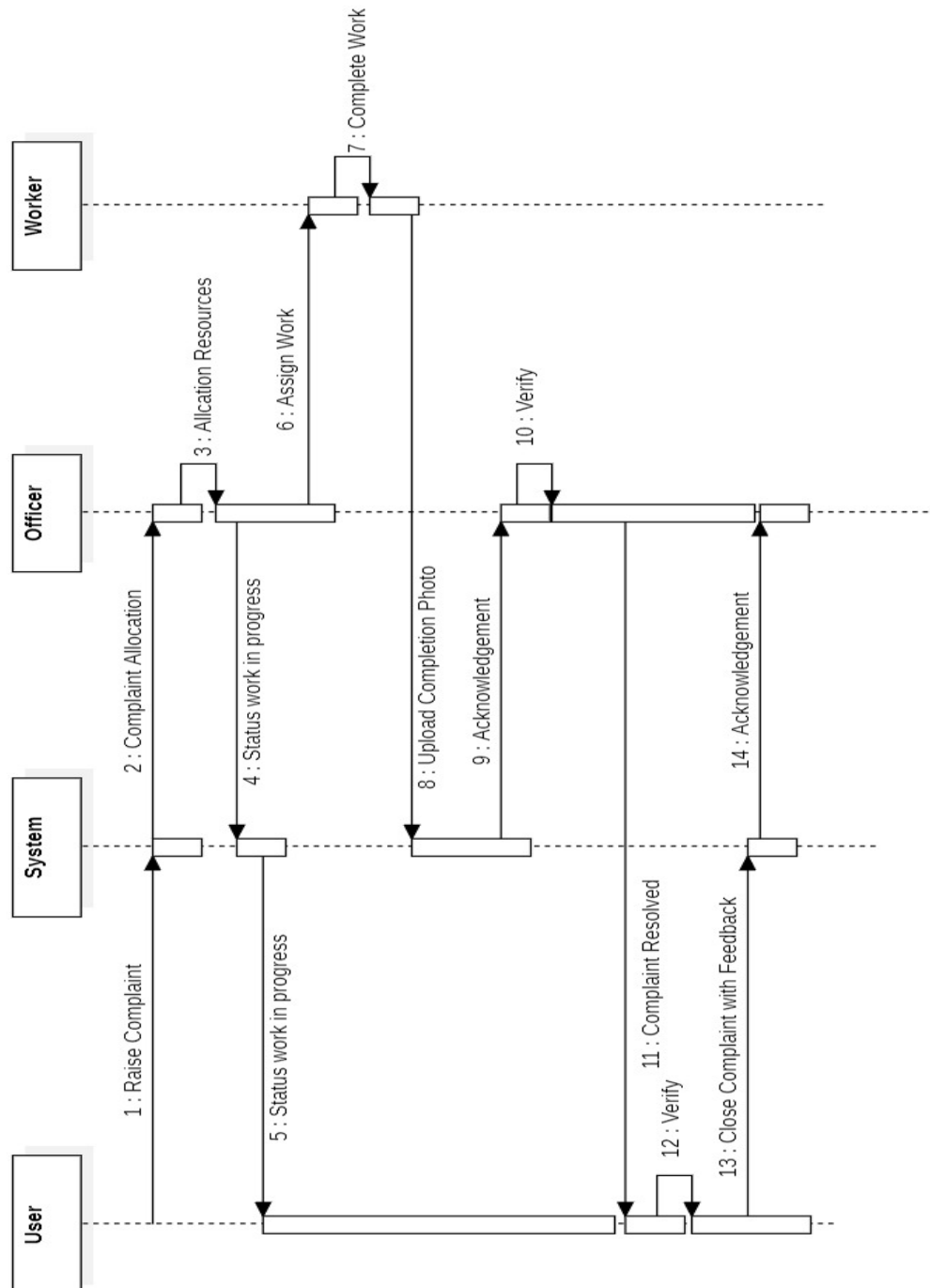


Figure 5.11: Sequence Diagram

### 5.4.6 Component Diagram

Component diagrams are used to represent how the physical components in a system have been organized. We use them for modeling implementation details.

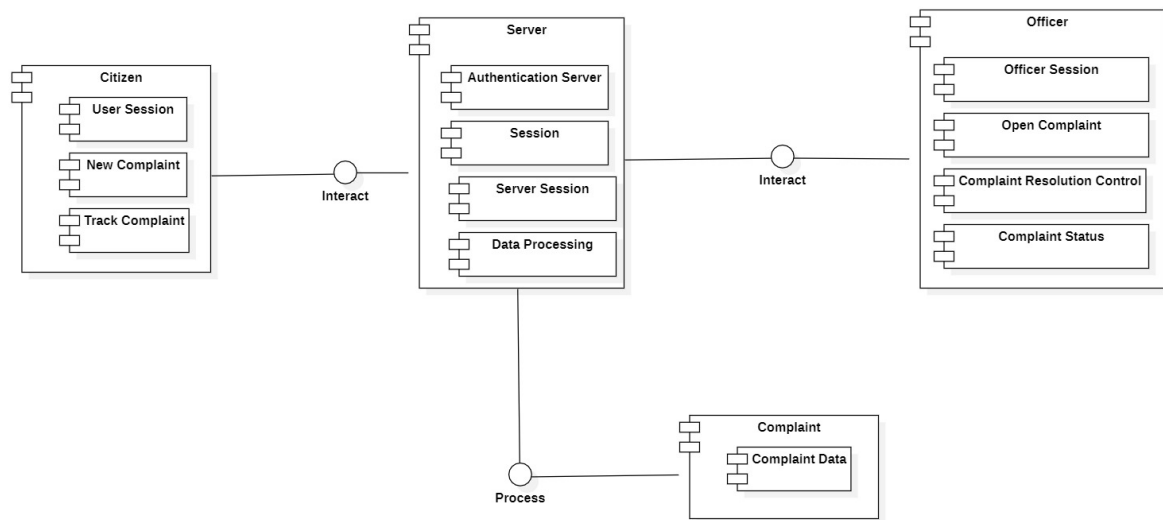


Figure 5.12: Component Diagram

# Chapter 6

## Other Specifications

### 6.1 Advantages

1. **Enhanced Civic Engagement** : Encourages citizen participation and community involvement in addressing local issues.
2. **Improved Administrative Efficiency** : Streamlines complaint handling and resource allocation, reducing administrative workload and costs.
3. **Transparency and Accountability** : Promotes transparency in local governance through real-time tracking and accountability mechanisms.
4. **Cost Savings** : Reduces administrative expenses by automating processes and optimizing resource allocation.
5. **Community Betterment** : Contributes to cleaner, safer, and more responsive environments, enhancing the quality of life.
6. **User Satisfaction** : Provides a convenient and responsive platform, leading to higher user satisfaction.
7. **Efficient Resource Allocation** : Ensures that complaints are promptly routed to the appropriate departments, optimizing government resources.
8. **Reduces Response Times** : Accelerates complaint resolution by automating issue categorization and resource allocation.

## 6.2 Limitations

1. **Digital Divide:** The application's effectiveness may be limited by the digital divide, as not all citizens in rural or urban areas have equal access to smartphones or the internet. Some individuals, especially in remote or underserved areas, may face barriers to using the platform, potentially excluding them from participating in civic engagement.
2. **Security and Privacy Concerns:** Handling sensitive citizen data and images raises significant security and privacy concerns. The project must implement robust security measures to protect user information, and there's a risk of potential data breaches or misuse of personal information, which requires careful management and adherence to data protection regulations.

## 6.3 Applications

1. **Local Governance Enhancement:** Municipalities and local government bodies can use this application to efficiently manage and address citizen complaints related to infrastructure, waste management, and other local issues.
2. **Urban and Rural Complaint Management:** Streamlining and automating the handling of complaints related to waste, infrastructure, and more in both urban and rural areas.
3. **Rural Development Agencies:** Rural development agencies can utilize the application to facilitate citizen engagement in rural areas, focusing on agriculture, infrastructure, and basic amenities.
4. **Environmental Protection Agencies:** Environmental protection agencies can employ the application to address and manage complaints related to pollution, illegal waste disposal, and environmental hazards in urban and rural settings.
5. **Community Organizations:** Non-governmental organizations (NGOs) and community-based organizations can adapt the application to encourage community members to report issues and participate in community development projects.
6. **Emergency Services:** The application can be used during emergencies or natural disasters to allow citizens to report urgent issues and help emergency services respond quickly to crises in both urban and rural areas.

# Chapter 7

## Project Plan

### 7.1 System Implementation Plan

Sr.No	Task	Start Date	End Date	Status
1	Introduction and Problem Definition	25/07/2023	05/08/2023	Completed
2	Literature Survey	06/08/2023	16/08/2023	Completed
3	System Requirement Gathering	17/08/2023	12/09/2023	Completed
4	Feasibility Study	13/09/2023	23/09/2023	Completed
5	System Analysis	24/09/2023	02/10/2023	Completed
6	System Design	03/10/2023	24/10/2023	Completed
7	Conclusion	25/10/2023	30/10/2023	Completed

Table 7.1: System Implementation Plan

## 7.2 Gantt Chart

Prepare Gantt chart it typically includes two sections: the left side outlines a list of tasks, while the right side has a timeline with schedule bars that visualize work.(Weekly)

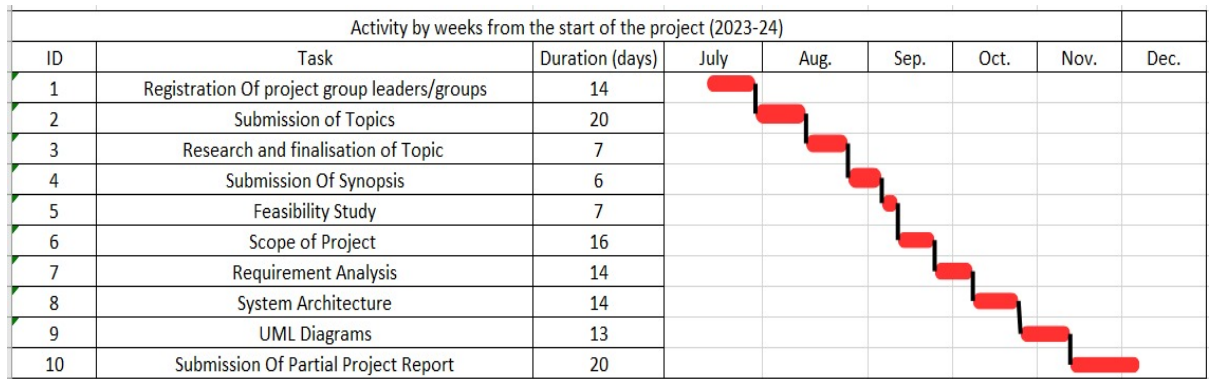


Figure 7.1: Gantt Chart



## Chapter 8

### Conclusion

The "Civic Complaint Registering Application for Urban and Rural Areas" offers an innovative solution to enhance citizen engagement, administrative efficiency, and community betterment. It empowers citizens, streamlines government processes, and fosters transparency, ultimately contributing to more responsive and accountable local governance. While challenges exist, the project's potential for positive change is evident, bridging technology and public participation to advance community well-being.

# References

- [1] Dnyanesh Walwadkar, Saurav Yadav **Smart Civic Issue Reporting System** Published on International Journal of Advanced Research in Science Communication and Technology DOI:10.48175/IJAR SCT-2659
- [2] Anjaly Antony, Minla K.S **Civic Complaint Reporting Using Image Processing** Published on IJCRT — Volume 9, Issue 6 June 2021 — ISSN: 2320-2882
- [3] Pengfei Xu, Xiaojun Chang, Ling Guo, Poyao Huang, Xiaojiang Chen, Alexander G.Hauptmann **A Survey of Scene Graph: Generation and Application** Published on IEEE Trans. Neural Netw.Learn. Syst. 2020.
- [4] Shanu Kumar, Anjali Singh, Mohit Jain **Adversarial Adaptation of Scene Graph Models for Understanding Civic Issues** Published on (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.ACM ISBN 978-1-4503-6674-8/19/05.
- [5] Qiwei Yin, Ruixun Zhang, XiuLi Shao **CNN and RNN mixed model for image classification** Published on MATEC Web of Conferences 277, 02001 (2019) JCMME 2018
- [6] Vishesh K. Kandhari, Keertika D. Mohinani **GPS based Complaint Redressal System** Published on 2014 IEEE Global Humanitarian Technology Conference - South Asia Satellite (GHTC-SAS) — September 26-27, 2014 — Trivandrum
- [7] Wenpeng Yin, Katharina Kann, Mo Yu and Hinrich Schutze **Comparative Study of CNN and RNN for Natural Language Processing** Published on February 2017 CIS, LMU Munich, Germany IBM Research, USA wenpeng,kann@cis.lmu.de, yum@us.ibm.com
- [8] Kumar Prasad, Ritesh Patil, Sagar Beldare, Prof. Anita Shinde **Civic Complaint Application under Smart City Project** Published on (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (1) ,ISSN: 0975-9646 2016,402-406

- [9] Soman Kp, Sowmya Vishvanathan, Deepika Jaswa **Image Classification Using Convolutional Neural Networks** Published on June 2014 International Journal of Scientific and Engineering Research 5(6):1661-1668 DOI:10.14299/ijser.2014.06.002
- [10] Mrs. Dhanashree B.Mote Dr.Deepak Gupta **Smart Control Complaint System for Municipal Corporations** Published on International Journal of Current Engineering and Technology E-ISSN 2277 – 4106, P-ISSN 2347 – 5161 Available at <http://inpressco.com/category/ijcet>

# Annexure A

## Plagiarism Report For this Report

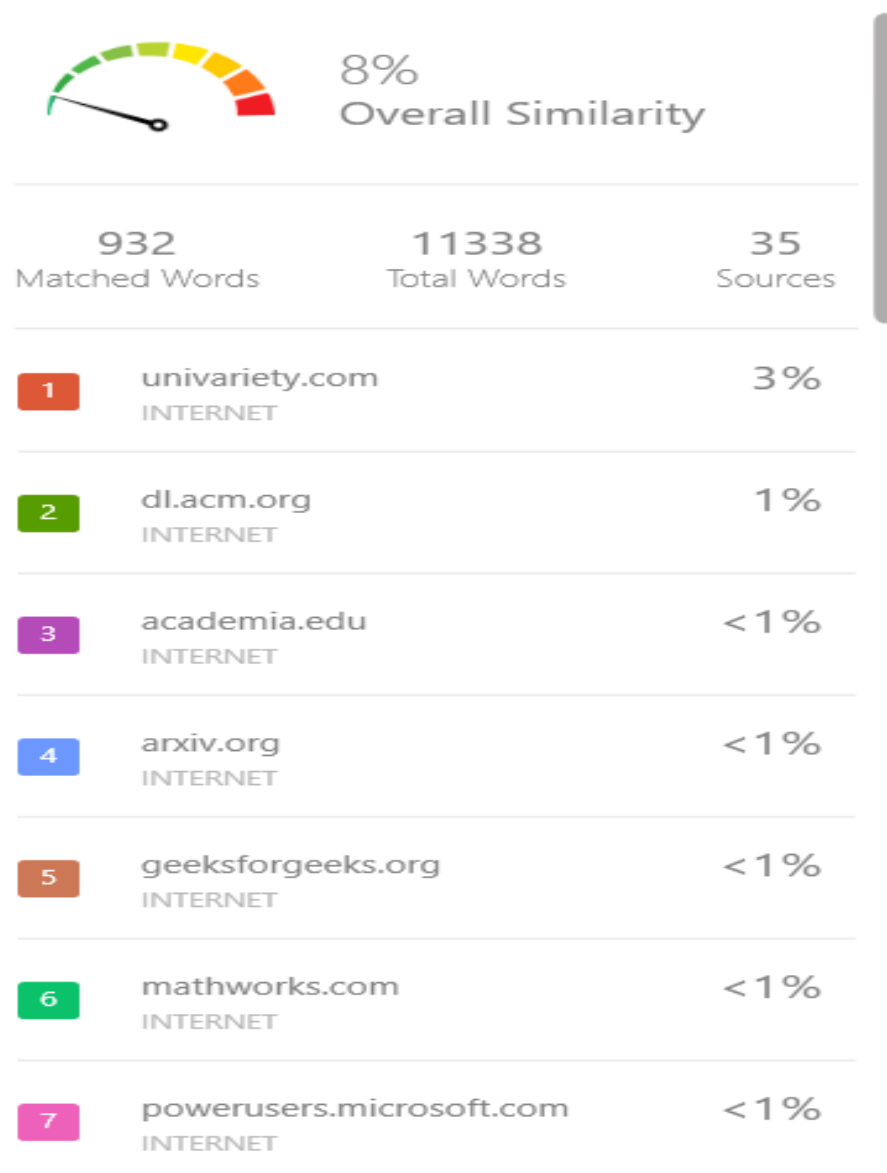


Figure 8.1: Plagiarism Certificate