SMART CIVIC ISSUES REPORTING SYSTEM

By

TINASHE PRESLEY CHAITEZVI

(H190592P)

HIT400 Capstone project Submitted in Partial Fulfillment of the

Requirements of the degree of

Bachelor of Technology

In

Software Engineering

In the

School of Information Sciences and Technology

Harare Institute of Technology

Zimbabwe

Supervisor

MR MAKONDO

05/2023

Certificate of Declaration

This is to certify that work entitled “Smart civic issues reporting system for local authorities “is *submitted in partial fulfillment of the requirements for the award of Bachelor of Technology (Hons) in Software Engineering, Harare Institute of Technology. It is further certified that no part of research has been submitted to any university for the award of any other degree.*

HIT Harare Institute of Technology

success through innovation

(Supervisor) Signature……………………………..

Date……………………….

(Mentor) Signature……………………………..

Date………………………

(Chairman) Signature………………………………..

Date………………………..

HIT 400 /200 Project Documentation Marking Guide

ITEM	TOTAL MARK /%	ACQUIRED/%
PRESENTATION- Format-Times Roman 12 for ordinary text, Main headings Times Roman 14, spacing 1.5. Chapters and sub-chapters, tables and diagrams should be numbered. Document should be in report form. Range of document pages. Between 50 and 100.Work should be clear and neat	**5**	
Pre-Chapter Section Abstract, Preface, Acknowledgements, Dedication & Declaration	**5**	
Chapter One-Introduction Background, Problem Statement, Objectives – smart, clearly measurable from your system. Always start with a TO… Hypothesis, Justification, Proposed Tools Feasibility study: Technical, Economic & Operational Project plan –Time plan, Gantt chart	**10**	
Chapter Two-Literature Review Introduction, Related work & Conclusion	**10**	
Chapter Three –Analysis Information Gathering Tools, Description of system Data analysis –Using UML context diagrams, DFD of existing system Evaluation of Alternatives Systems, Functional Analysis of Proposed System-Functional and Non-functional Requirements, User Case Diagrams	**15**	

Chapter Four –Design Systems Diagrams –Using UML Context diagrams, DFD, Activity diagrams Architectural Design-hardware, networking Database Design –ER diagrams, Normalized Databases Program Design-Class diagrams, Sequence diagrams, Package diagrams, Pseudo code Interface Design-Screenshots of user interface	**20**	
Chapter Five-Implementation & Testing Pseudo code of major modules /Sample of real code can be written here Software Testing-Unit, Module, Integration, System, Database & Acceptance	**20**	
Chapter Six –Conclusions and Recommendations Results and summary, Recommendations & Future Works	**10**	

Table of Contents

Table of Figures

Abstract

Reporting infrastructure issues in cities is a time-consuming and often frustrating process for citizens. This is due to the lengthy steps and procedures involved in reporting problems, as well as the lack of guarantee that complaints will be processed in a timely manner. As a result, many complaints go unheard, unanswered, and unresolved.

To address this issue, we propose a novel mobile application that allows citizens to report infrastructure issues to relevant authorities in a quick and easy manner. The application uses a hybrid CNN-RNN image processing algorithm to detect the severity of reported problems, so that higher-severity issues can be addressed first. The application also uses the GPS sensor on smart mobile devices to pinpoint the exact location of the problem zone and use the camera to record the problem area as visual evidence. The system then generates a form containing all the data and location and visual evidence entered by the user and sends it to a central server. The central server then notifies the relevant agencies, who are able to take action to resolve the issue.

We believe that this mobile application will make it easier for citizens to report infrastructure issues and will help to improve the quality of life in cities

Keywords: CNN-RNN Hybrid Image Processing Algorithm, GPS, Android Application, Camera, Civic Complaints, Smart City.

Preface

Cities are the epicenters of growth and development. They are where people come to seek opportunities, pursue their dreams, and build their lives. However, with the rapid increase in urbanization, cities are facing a plethora of civic issues that are affecting the quality of life for its citizens. Delayed and inefficient resolution of these issues is a significant concern for local government bodies, as it directly impacts citizens' satisfaction and trust in their services.

The proposed platform, Mushandirapamwe, aims to address this issue by providing citizens with a convenient and user-friendly way to report issues and allowing city officials to be notified of them as they occur. This survey paper presents an overview of the system's objectives, proposed functionality, and justification for its implementation.

Mushandirapamwe, which means "together we can" in Shona, is a platform that emphasizes collaboration and communication between citizens and their local government. It seeks to empower citizens to take an active role in their city's improvement and ensure that their issues are resolved efficiently.

This documentation outlines the objectives and proposed functionality of the Mushandirapamwe platform, which includes a mobile application for citizens to report issues and a web application for city officials to manage and track these complaints. It also incorporates features such as sentiment analysis and complaint severity classification to aid in decision-making and generate intelligent analysis reports.

We believe that the implementation of Mushandirapamwe will not only improve the quality of life for citizens but also strengthen the relationship between citizens and their local government. It is our hope that this platform will contribute to the betterment of cities.

Acknowledgements

I would like to express my sincere gratitude to Almighty God for the success of this project. I am also grateful to Mr. Makondo, my project supervisor, for his patient guidance and valuable suggestions during the design and implementation of this project. I appreciate his willingness to give his time so generously.

I would also like to thank the staff at the department of Software Engineering for their help in providing me with the resources I needed to run the program. I am grateful for their support and assistance.

Finally, I would like to thank my friends and classmates for their opinions and useful critiques of this work. I appreciate their feedback and support.

Dedication

The proposed platform, Mushandirapamwe, is a collaborative effort of a team of dedicated individuals who have contributed their time, expertise, and resources to bring this project to fruition. Therefore, we would like to dedicate this project to our families and friends who have supported us and encouraged us throughout this journey.

We would also like to express our gratitude to our project supervisor, who has provided valuable guidance and support, and without whom this project would not have been possible. We would like to thank our colleagues and classmates who have provided feedback and insights that have helped us improve and refine our work.

Finally, we dedicate this project to the citizens of our city, who have inspired us to create a platform that can help make our city a better place to live. We hope that Mushandirapamwe will help bridge the gap between citizens and city officials, and contribute to a more efficient and effective management of civic issues.

Declaration

All sentences or passages quoted in this dissertation from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). I understand that failure to do this amount to plagiarism and will be considered grounds for failure in this dissertation and the degree examination as a whole.

This page was intentionally left blank

1. CHAPTER ONE – Proposal

1.1. Introduction

Today's smartphones have the potential to revolutionize the way we engage with our communities. By harnessing the power of crowdsourcing, we can empower citizens to solve complex problems and improve our cities in ways that were never before possible.

One of the most important ways that crowdsourcing can be used to improve our cities is through citizen complaints. Complaints are a valuable source of feedback that can help us identify and address problems that we might not otherwise know about. However, traditional grievance systems can be time-consuming and inconvenient for citizens.

To address this challenge, we have developed an Android application that makes it easy for citizens to report complaints with just two clicks. The application is easy to use and does not require any background knowledge of the complaint process. This makes it accessible to everyone, regardless of their age or technical expertise.

In addition to making, it easier for citizens to report complaints, our application also provides a transparent and efficient way for civil servants to address them. Once a complaint is submitted, it is immediately routed to the appropriate department for review. Our application is just one example of how crowdsourcing can be used to improve our cities. By empowering citizens to participate in the decision-making process, we can create more vibrant and equitable communities for everyone.

1.2. Background

Each and every country have civic bodies that are the local governing bodies which help in maintaining general functionalities of cities. These civic bodies are called as Municipal

Corporations. These municipal cooperations also known as local authorities manage civic issues from day to day.

Today we are facing many civic issues, especially in populated cities. Illegal waste dumping is a serious environmental concern in many countries. These illegal dumps decrease the quality of human life in surrounding areas. Reporting problems such as road damage, street cleaning, dents, trash can overflow, street light post damage, or anything under the supervision of a municipality requires lengthy steps and procedures. Reported complaints will be processed by the competent authority. If daily 1000 complaints get registered then because of this, most complaints are unheard, unanswered, and unresolved because the company is too large to handle small complaints from one person. Reporting road and citizen issues is no longer an easy process for citizens.

1.3.Problem Statement

a. Citizens are often dissatisfied with the surrounding or urban infrastructure, but may not prefer traditional grievance systems that have to go through lengthy procedures such as going to the office and waiting in line for hours.
b. This can lead to a lack of communication between citizens and government agencies, which can make it difficult to identify and address problems.
c. In addition, traditional grievance systems can be time-consuming and inefficient, which can delay the resolution of problems.

1.4.Aim

To develop a mobile application that makes it easy for citizens to report complaints and participate in the decision-making process and web portal where officials will manage the reported issues. The application will be easy to use and accessible to everyone, regardless of their age or technical expertise. It will also provide a transparent and efficient way for civil servants to address complaints.

1.5.Objectives.

- To automatically geo-tag the reported complaints using mobile application

- To manage civic complaints and update their resolution status via web application
- To highlight the areas of the reported issues on a map using map markers
- To classify feedback from citizens to gain an overview of the damage severity using image processing techniques
- To generate data-driven analysis reports to provide intelligent insights for decision making.

1.6.Significance of the Project

World is going through the urban development phase. Experts predict that the population will be double by 2050 meaning 70 percent of the total population will be living in major town or city. With the ever-increasing population it is necessary for local government body to improve their services because the city as only as smart as the services provided by it, as the services directly impact the quality of the citizens life. So, the urban apps are one of the requirements for smarter cities. So, the proposed system would be in which people can take pictures of the neighborhoods issues and submit the picture along with details to local government body, the complaint would be lodged along with the address of the issue with the local government which in turn can forward the problem to the appropriate department and the department will look after the issue. The citizens will be made available with the current status of the complaint and what action is being taken against it and how quick the problem can be solved. The main purpose of this project is to help the public by providing them a system that can be used for betterment of the city and can make the city clean and problem free as well as the citizen life easier

1.7.Methodology

The proposed system will be developed using a combination of agile and waterfall methodologies. Agile methodologies are well-suited for projects that are complex and require frequent changes, while waterfall methodologies are better suited for projects that are well-defined and have a clear timeline.

The first step in the development process will be to gather requirements from stakeholders. This will involve conducting interviews, surveys, and focus groups. Once the requirements have been gathered, they will be analyzed and prioritized.

The next step will be to design the system. This will involve creating a system architecture, user interface design, and database design. Once the design has been finalized, it will be implemented.

The implementation phase will involve coding, testing, and deploying the system. The system will be tested by both developers and stakeholders. Once the system has been deployed, it will be monitored and maintained.

The proposed system will be developed using a variety of tools and technologies. These tools and technologies will be selected based on their suitability for the project and the needs of the stakeholders.

The following are some of the tools and technologies that will be used in the development of the proposed system:

- Programming languages: PHP, Python, Javascript
- Databases: MySQL
- Web development frameworks: Django, React Native

The proposed system will be developed in a secure and compliant manner. This will involve following industry best practices and using security tools and technologies and will be tested thoroughly to ensure that it is free of bugs and security vulnerabilities.

1.8.Scope

The scope of this project is to develop a system that can handle and resolve civic complaints effectively and efficiently. The system will consist of a mobile application and a web application that will enable citizens to report issues and track their progress. The system will also use various techniques to analyze the data and provide insights for better governance.

1.9.Definition of Key Variables

- Number of complaints reported: This is the total number of complaints that are reported by citizens using the mobile application.
- Resolution status of complaints: This is the status of each complaint, such as open, in progress, or closed.
- Location of complaints: This is the location of each complaint, such as a street address, a building, or a park.
- Severity of complaints: This is the severity of each complaint, such as minor, moderate, or major.
- Sentiment of feedback: This is the sentiment of the feedback that is provided by citizens, such as positive, negative, or neutral.

1.10. Conclusion

This chapter has discussed the current state of affairs so far as civic issues and their resolution processes is concerned. The remaining chapters detail the feasibility study of the project, the proposed design of solution, the implementation strategy used and the testing methodologies applied. Further chapters ahead test the assumptions, attempt to answer the research questions and the overall evaluation of the project`s success.

2. CHAPTER TWO – *Literature Review*

2.1.Introduction

This section reviews the details around similar and related research topics which may or may not have been implemented.

2.2.Synthesis

The following topics are to do with activities and projects previously undertaken which are related to this paper's field of study (Smart Civic Issues Reporting System For Local Authorities).

2.3.Review of Existing Systems

a. A comprehensive citizen engagement framework for effective resolution of public complaints in cities, Prassanna Pathmanathan, *Informatics Institute of Technology*

There are programs referred to as "citizen engagement." According to the Open Government Guide, in order for open government efforts to be effective, citizen involvement is essential. Governments should prioritize citizen involvement because it guarantees that people have access to their fundamental human rights through fostering a just and equitable society. [4]

The focus of this study is on whether or not the general public has access to an efficient public complaint management system for bringing concerns to governmental agencies and keeping track of them in a way that meets their needs. As a result, the study's goals are to determine how notions of citizen involvement affect public complaint handling, to identify the variables influencing citizen participation in this process, and to provide suggestions based on the findings.

Only a small portion of Sri Lankan complaints filed using the current methodology were effectively resolved [5]. Additionally, the flaws listed below have made individuals unsatisfied with the government's capacity to respond to their concerns, and there is also a "disconnect" between the government and citizens as a result of the absence of transparency and accountability [6]. Since Sri Lanka also participates in the Open Government Partnership, it was practical for them to establish a full public complaint management solution [7].

Understanding how citizens participate in public complaint management and validating the aforementioned elements were accomplished via the use of a mixed technique that combined the collection of both qualitative and quantitative data. All Sri Lankans who reside in, work in, or often travel through its major cities make up the study's population. From this, 300 millennials were randomly selected as a sample who reside in, work in, or frequently visit the capital of Sri Lanka. A structured questionnaire was then tested using an online survey platform. Interviews with the Colombo Municipal Council (CMC), Central Environmental Authority (CEA), Road Development Authority (RDA), Urban Development Authority (UDA), and Ministry of Health were conducted in order to collect qualitative data (MOH). The conceptual framework and IT solution were developed using the acquired data, mappings, and findings.

The Spotlight IT solution was created to improve the efficacy and efficiency of the whole complaint management process and makes it simple for individuals to become involved. Therefore, the "gap" between the actual and anticipated standards of the complaint management process is "bridged" by the IT solution. The implemented IT solution is the only one in the world to integrate all features of current solutions in one interface while it also has many innovative features as a result of a thorough examination of existing IT solutions and creativity on the part of the project team. A smartphone app and a web app are used for the front-end implementation, which enables users to file complaints quickly and simply along with images, videos, and GPS coordinates. Every person has access to every complaint, which is organized by issue kind, location, date of filing, etc. Once a complaint has been resolved, residents may verify the resolution using photos, videos, and ratings. If they are not satisfied with the outcome, they can resubmit or escalate the complaint to a different government official. Points and badges are used to reward citizens. Government institutions are currently supported by a web application that makes it simple to track all complaints and gives authorities the means to escalate issues between departments.

An efficient complaint management solution for use by the government has been developed as a result of this project. The IT solution offers a strategic perspective of how enhancing citizen participation through the use of IT may be done, and any government agency can utilize the conceptual framework presented above as a guide to create a complaint management system that works.

b. Natural Language Mobile Interface to Register Citizen Complaints by Sunil Kumar Kopparapu, TCS Innovation Lab - Mumbai

The functionality of the web portal-based system for reporting complaints is modeled in the proposed mobile natural language-based interface system. Users can register complaints using their mobile device rather than immediately using the online portal interface. On his smartphone, the user installs an application. The welcome screen appears when the user launches the app on his phone. The system allows the user to enter 160 characters for his or her complaint. The system's Natural Language Complaint Processing System Block receives the complaint in natural English via the SMS channel. The NL-based complaint processing system then uses natural language processing (NLP) techniques to interpret the user's complaint to determine the nature of the complaint (the Department and the Complaint Type) and other details (location, land mark, ward number) that are mandatory to lodge a complaint at [2]. The system provides the information in a format that is compatible with the MC complaint management system when it has deduced these details. To find the complaint number, the answer from the MC complaint management portal is fetched and processed. The user then receives this complaint number on his cell phone. The system enters an interaction mode if the complaint is incomplete in the sense that it lacks the mandatory information needed to produce a complaint number at the MC portal. In this mode, the system creates a series of dynamically produced questions to ask the user for the necessary information via menu selections or as text. When the system has collected the necessary data, it sends the information to the MC complaint handling portal. [2]. The user can submit his or her complaint using standard English thanks to the technology. English text in free format is provided as the system's input. To assess if the complaint has all the necessary information, the system internally examines the complaint content using a variety of natural language processing techniques. If the complaint sent by the user is such that a complaint number cannot be generated using the MC web portal interface system, the application intelligently generates queries and collects the necessary information from the user. If not, the information received from the portal (complaint ID generated by system) is then sent back to the reporter. The user replies to questions that the system generates. This exchange could take place more than once. In addition to the free form complaint submitted by the user, all the information acquired is utilized to file the complaint with the web-based MC complaint registering system, and the complaint number produced is handed back to the user for his reference and future complaint monitoring.

An example of this is a user complaint that is fully written in free English format (for example, Garbage has not been lifted from under the flyover in town). The complaint is examined by the system for completeness. A complaint is judged complete if the user's composition of the complaint contains all the essential fields that must be filled out in order to file the complaint.This often leads to one of two extreme outcomes: either the complaint is true or it isn't.

There is no user involvement in Scenario 1 after the user has made the original complaint. The user can submit his complaint all at once. During the scenario 2, the system engages the user in communication to get the missing details needed to finish the complaint before submitting a complaint using the portal [2]. In the most severe scenario, where the user provides no input, the system asks the user for information in the form of options and text boxes for each of the eight necessary fields.

If the complaint is incomplete, the system actively asks the user for the necessary information by posing specific questions (through menu choice). In this instance, the system files the complaint using all the data provided by the user (the initial free format complaint and the subsequent responses to the system generated queries). The option to submit photos of the incident together with the complaint strengthens its legitimacy and, in a way, completes the complaint filing process. The minimal processing system [1] by design allows the system to be configured to other native languages and will require the design of screens in native language on the mobile phone.

c. Complaint Management System that can classify complaints intelligently using Machine Learning

A complaint is a client's way of expressing displeasure with a good or service. The communication of the complaint to the appropriate party may be oral or written. [3]. Customer Complaint Management may have an impact on how satisfied customers are, thus many businesses often have a procedure in place to deal with complaints in order to optimize customer satisfaction. The company must automatically classify complaints in order to transmit them to the appropriate department due to the volume of complaints that they receive. Time and labor savings are hence advantages.

Additionally, this service offers back-end services for categorizing complaints to the appropriate departments and for locating concerns that are similar to prevent repeated complaints.

The scope of SCMS would be restricted to maintenance problems. for instance, maintaining the facilities, classrooms, and laboratories. Because SCMS was created to improve the effectiveness of the complaint management process, it has the ability to automatically classify complaints and deliver them straight to the department in charge of resolving the issue.

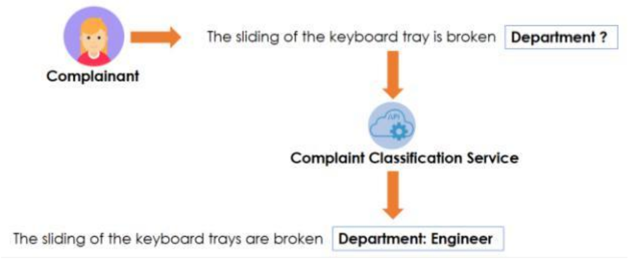

Complainant
The sliding of the keyboard tray is broken
Department ?
Complaint Classification Service
The sliding of the keyboard trays are broken
Department: Engineer

Figure 1: Complaint Classification Service for research

As shown in Fig. 1, after the complainant presents their complaint to the system, the cloud-based complaint categorization service will predict the department that will be handling it. For instance, if someone complains that the keyboard tray's sliding mechanism is malfunctioning, the service would likely assume the engineer is to blame.

By utilizing both a web application and a mobile application, SCMS was created to improve the existing complaint management system. In order to make it simpler for consumers to file complaints, SCMS was able to provide a variety of channels for doing so. These channels include a channel for progress tracking via a mobile application. Additionally, SCMS had the ability to categorize the complaint and deliver it straight to the proper department, which allowed the system to cut down on the time and labor costs associated with employing people. By advising users to file a comparable complaint, SCMS might also reduce the number of duplicate complaints. Additionally, SCMS enables the employees to manage the complaint automatically using the online application rather than manually via paper form. The system then creates the data visualization for the complaint data summary.

Additionally, the researcher selected the supervised learning algorithms—J48 Decision Tree, Sequential Minimal Optimization, and Naive Bayes—that were appropriate for our training set. Each supervised learning example demonstrates a different machine learning type’s characteristic. For example, J48 operates on decision trees, Sequential Minimal Optimization is comparable to Space Vector Machines, which support functional equations, and Naive Bayes computes Bayes' rule, which operates on the basis of probability.

d. **A comprehensive citizen engagement framework for effective resolution of public complaints in cities, Prassanna Pathmanathan, *Informatics Institute of Technology***

There are programs referred to as "citizen engagement." According to the Open Government Guide, in order for open government efforts to be effective, citizen involvement is essential. Governments should prioritize citizen involvement because it guarantees that people have access to their fundamental human rights through fostering a just and equitable society. [4]

The focus of this study is on whether or not the general public has access to an efficient public complaint management system for bringing concerns to governmental agencies and keeping track of them in a way that meets their needs. As a result, the study's goals are to determine how notions of citizen involvement affect public complaint handling, to identify the variables influencing citizen participation in this process, and to provide suggestions based on the findings.

Only a small portion of Sri Lankan complaints filed using the current methodology were effectively resolved [5]. Additionally, the flaws listed below have made individuals unsatisfied with the government's capacity to respond to their concerns, and there is also a "disconnect" between the government and citizens as a result of the absence of transparency and accountability [6]. Since Sri Lanka also participates in the Open Government Partnership, it was practical for them to establish a full public complaint management solution [7].

Understanding how citizens participate in public complaint management and validating the aforementioned elements were accomplished via the use of a mixed technique that combined the collection of both qualitative and quantitative data. All Sri Lankans who reside in, work in, or often travel through its major cities make up the study's population. From this, 300 millennials were randomly selected as a sample who reside in, work in, or frequently visit the capital of Sri Lanka. A structured questionnaire was then tested using an online survey platform. Interviews with the Colombo Municipal Council (CMC), Central Environmental Authority (CEA), Road Development Authority (RDA), Urban Development Authority (UDA), and Ministry of Health were conducted in order to collect qualitative data (MOH). The conceptual framework and IT solution were developed using the acquired data, mappings, and findings.

The Spotlight IT solution was created to improve the efficacy and efficiency of the whole complaint management process and makes it simple for individuals to become involved. Therefore, the "gap" between the actual and anticipated standards of the complaint management process is "bridged" by the IT solution. The implemented IT solution is the only one in the world to integrate all features of current solutions in one interface while it also has many innovative features as a result of a thorough examination of existing IT solutions and creativity on the part of the project team. A smartphone app and a web app are used for the front-end implementation, which enables users to file complaints quickly and simply along with images, videos, and GPS coordinates. Every person has access to every complaint, which is organized by issue kind, location, date of filing, etc. Once a complaint has been resolved, residents may verify the resolution using photos, videos, and ratings. If they are not satisfied with the outcome, they can resubmit or escalate the complaint to a different government official. Points and badges are used to reward citizens. Government institutions are currently supported by a web application that makes it simple to track all complaints and gives authorities the means to escalate issues between departments.

An efficient complaint management solution for use by the government has been developed as a result of this project. The IT solution offers a strategic perspective of how enhancing citizen participation through the use of IT may be done, and any government agency can utilize the conceptual framework presented above as a guide to create a complaint management system that works.

REFERENCES

[1]. Sunil Kopparapu, Akhilesh Srivastava, and PVS Rao. Minimal parsing question answering system. In International Conference on HCI, PR of China, 2007.

[2]. MCGM. In http://portal.mcgm.gov.in/irj/portal/anonymous/qlcomplaintreg, accessed Feb 2008.

[3]. Customer Complaint. [cited 2017 2]; Available from: http://www.financepractitioner.com/dictionary/customer-complaint.

[4]. M. Pimbert and T. Wakeford, "Overview - deliberative democracy and citizen empowerment", Participatory Learning and Action, vol. 40, pp. 23-28, 2001.

[5]. "Citizen engagement through eparticipation", GOV.LK, 2013. [Online]. Available:http://www.labour.gov.lk/web/index.php?option=com_content&view=ar ticle&id=286&Itemid=270&lang=en. [Accessed: 27- Jun- 2017].

[6]. I. Mergel, "Distributed Democracy: SeeClickFix.Com for crowdsourced issue reporting", SSRN Electronic Journal, 2012.✓

[7]. "Sri Lanka joins the Open Government Partnership", mfa.gov.lk, 2015. [Online]. Available: http://www.mea.gov.lk/index.php/en/media/media- releases/6234-sri-lanka-joins-the-open-government-partnership.[Accessed: 27- Jun- 2017].

2.4. Conclusion

Online complaint management systems enable to eradicate corruption and provides every citizen the assurance that complaints are monitored, tracked, resolved and problem areas are targeted. Mobile application creates an interface for the authority officials and people to have a proper communication and also reduces time and distance barrier. These all also promotes accountability for the municipal bodies involved

3. CHAPTER THREE - *Requirements Analysis*

3.1.Introduction

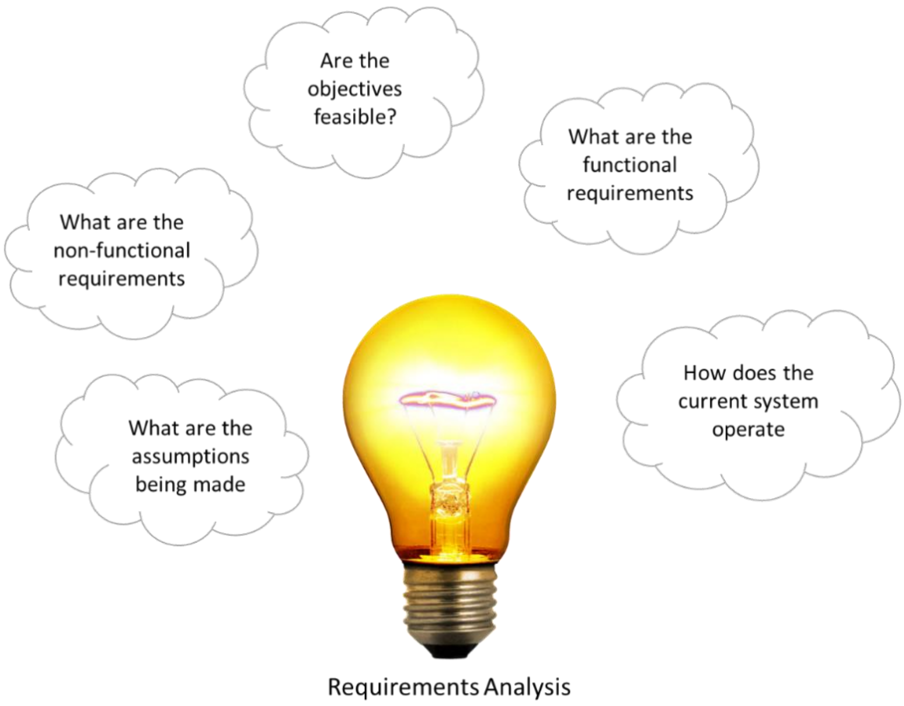

Are the objectives feasible?
What are the functional requirements
What are the non-functional requirements
How does the current system operate
What are the assumptions being made
Requirements Analysis

Figure 2: Requirements Analysis

The following chapter documents the concerns denoted in the infographic above, which are:

- Current system setup
- Feasibility of the objectives
- Functional and non-functional requirements
- Assumptions being about the system

3.2.Current System

Currently, Mutare City Council has a website for getting any information regarding the city and for any complaint registration. This module also acts as the interface between council and its clients. It is used to log all the queries brought to the attention of the council by its clients. The module will allow the PR personnel to capture all the queries reported through various channels such as:

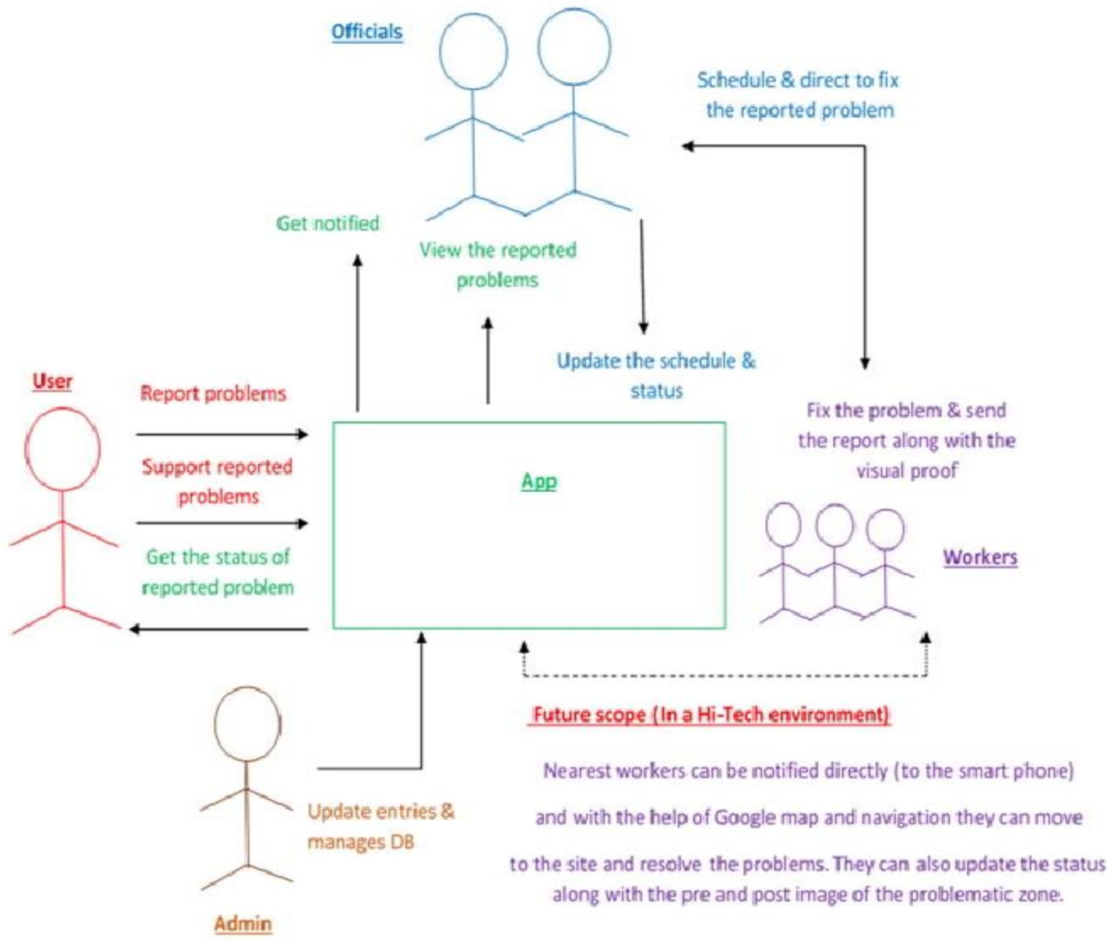
1.**Phone-ins**: These are queries received via telephone calls to the call center.

2. **Walk-ins**: These are reported by walk-in customers who physically visit council offices.

3. **Electronic media**: These are queries received through digital channels such as email, social media pages (Facebook & Twitter) and through the chatbot. This module will allow the PR personnel to dispatch the query to the relevant departments. The dispatch process basically notifying the line supervisor responsible for resolving the reported query. The dispatch process will notify both the supervisor and the department head. The time of dispatch automatically logged in the system. A successful dispatch process is when the intended supervisor acknowledges receipt of the query by responding to the dispatcher. The dispatcher will use walkie-talkies or telephones to notify the supervisors.

3.2.1.1.1. Context Level Diagram

The context level diagram for the above project is a high-level view of the system. It shows the system as a single process and the relationships between the system and external entities. The external entities are the people, organizations, or systems that interact with the system. The arrows represent the flow of data between the system and the external entities.



Officials
Schedule & direct to fix the reported problem
Get notified
View the reported problems
Update the schedule & status
User
Report problems
Support reported problems
Get the status of reported problem
App
Fix the problem & send the report along with the visual proof
Workers
Future scope (In a Hi-Tech environment)
Nearest workers can be notified directly (to the smart phone)
and with the help of Google map and navigation they can move
to the site and resolve the problems. They can also update the status along with the pre and post image of the problematic zone.
Update entries & manages DB
Admin

Figure 3: Mushandirapamwe context level diagram

3.2.2. Process Flow Diagram

This process flow diagram denotes the process though which citizens files complaint.

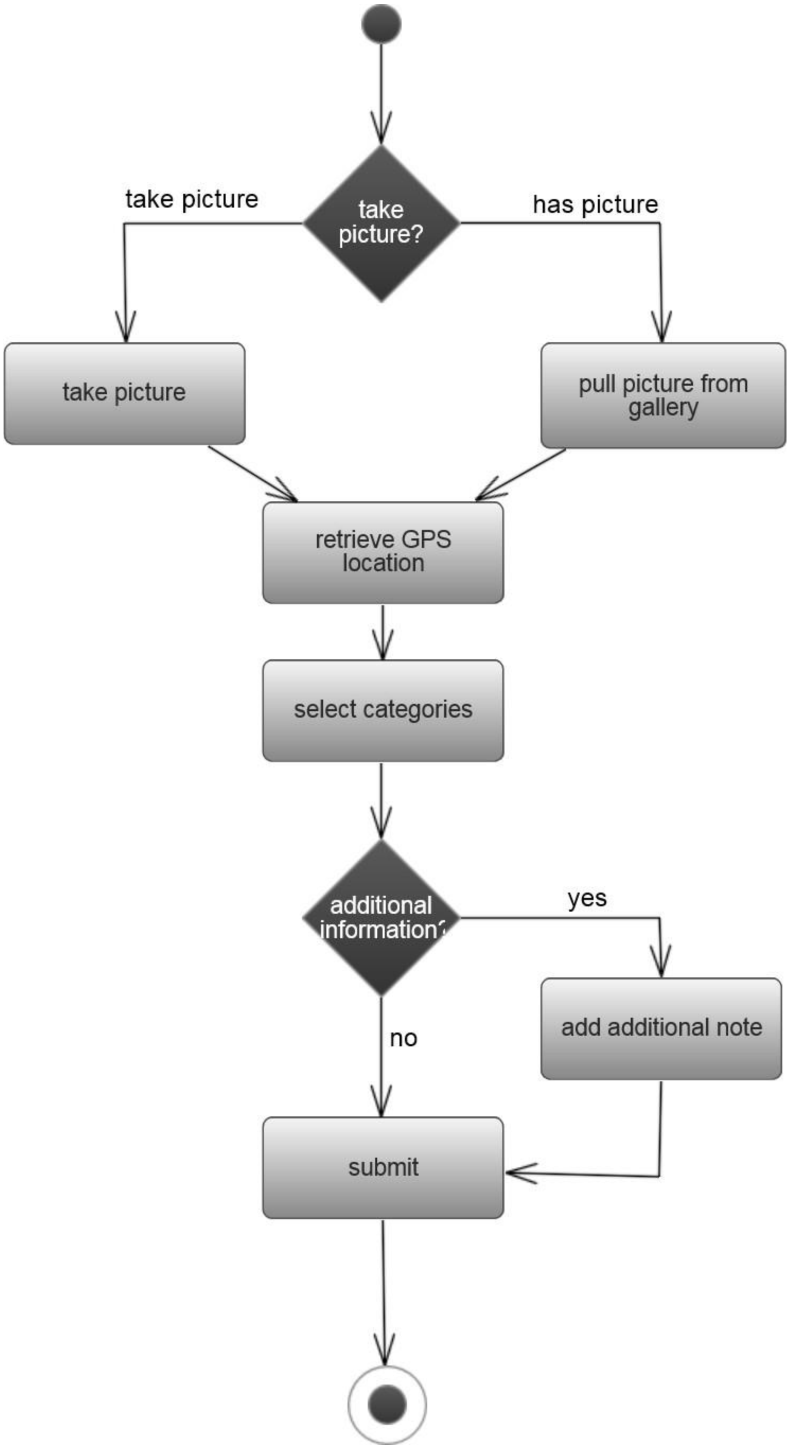


Figure 4: Process Flow diagram (Illustrating citizen logging a complaint)

3.2.3. Use Case

The Use Case diagram below denotes from an abstracted point of view, the interaction a citizen makes with the reporting system in respect to the process of filling a complaint and the complaint being managed by a city official.

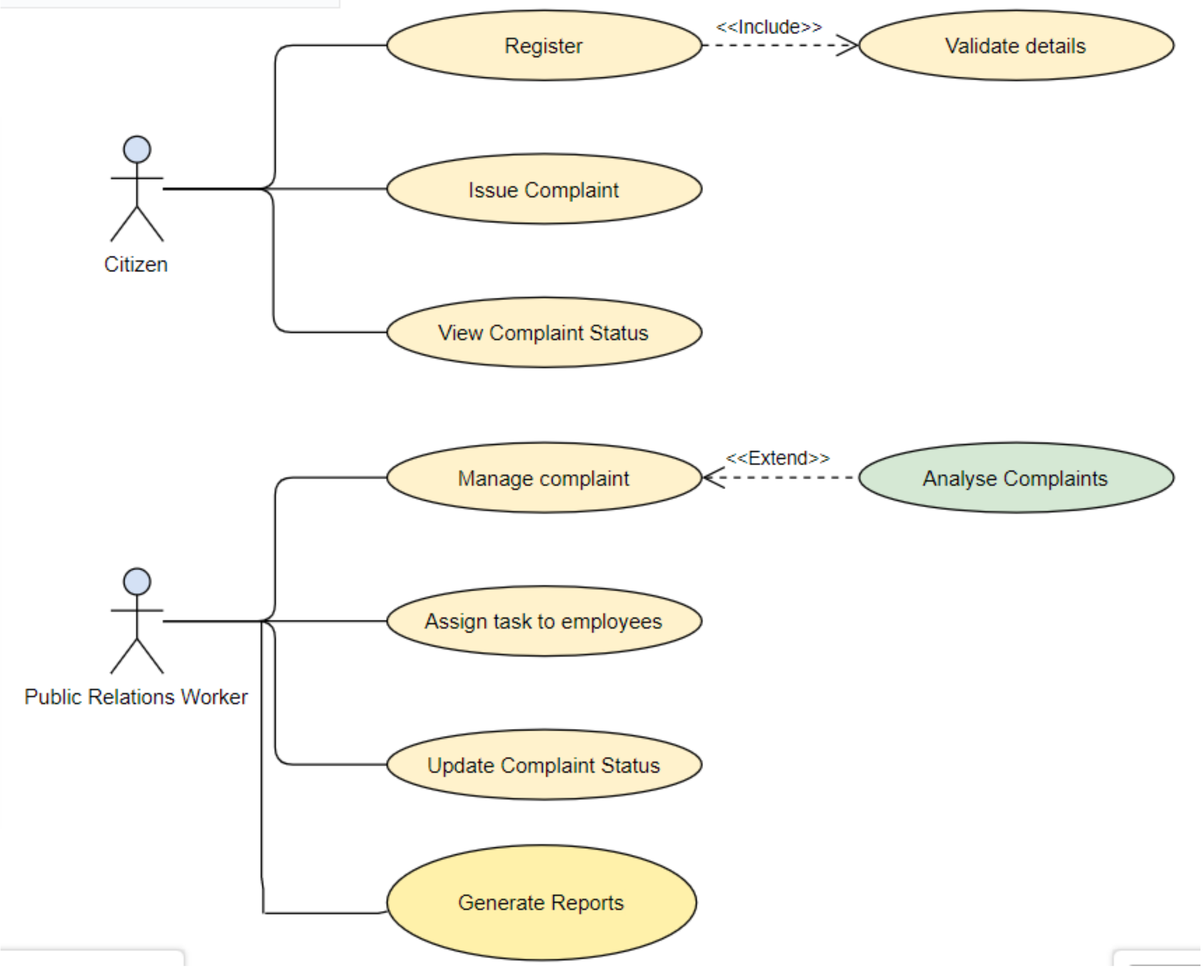


Figure 5: Use Case diagram of current system involving a citizen and public works work (from local authorities)

3.3.Feasibility Study

The purpose of this feasibility study is to analyze how successfully the project can be achieved, bringing into account internal and external influences that affect it. We will consider the technical feasibility and the economic feasibility. The study will largely be centered on the following areas:

- **Technical feasibility**
    - How accessible is the technical hardware for prototyping?
    - What skill will be required to avail a prototype
- **Economic feasibility**
    - The cost of acquiring such hardware
    - Will the outcome of the project assist in coming up with a useful and viable product?

3.3.1. Technical Feasibility

For the developers, it is worth checking if the current technology allows the development of the project to move forward and how we intend to deliver a product or service to customers .Also focus here is much on the availability of tools to be used and technical expertise. The tools we require are readily available and some are even open source. The IDEs for the languages to be used are available. Hardware and software requirements for development to supply adequate performance include the following:

Hardware

- A computer with minimum specifications Processor: Intel(R) Core (TM) i5-6200U CPU@ 2.30GHz (4 CPUs), ~2.4GHz
- Memory: 8192MB RAM
- Operating System: Windows 10 Pro 64-bit (10.0, Build 14393)
- Card name: AMD Radeon (TM) R5 M330)

Software

- Web front technologies (HTML, CSS, JAVASCRIPT)
- PHP with MVC coding convention
- MYSQL
- Visual Studio Code
- Front end web libraries using Bootstrap 4 and jQuery

No other special skills are required besides that which are already possessed by the researcher.

3.3.2. Economic Feasibility

This section evaluates feasibility of the project in respect to the cost-benefit factor of the project. Assessment will be done based on the following point:

- How much capital is required?
- Can the project outcome translate to profit?

Capital required for the project

All the software components required for this project are open source and thus do not need funding.

The purpose of an Economic Feasibility Study (EFS) is to demonstrate the net benefit of a proposed project for accepting or disbursing electronic funds/benefits, taking into consideration the benefits and costs to the agency, other state agencies, and the general public as a whole.

This system will be targeting every citizen that is willing to better the quality of life of a city that he or she lives in and the local authority's agency itself.

The development of a civic complaints reporting system will greatly improve service delivery and cultivate a healthy council-client relationship.

A local authority staff members will cut off unnecessary expenses due to driving vehicles (expending fuel and incurring vehicle wear/tear) to the location described attempting to locate the problem reported since a map will guide them.

As the civic reporting smartphone application becomes more and more ubiquitous, municipalities will see a dramatic shift in operational efficiencies. Issues being reported by phone, email (which require data entry, hence added costs) and walk-ins will begin to be replaced by smartphone reports and web-based forms (although web-based forms are a bit more cumbersome to the customer). The costs are far outweighed by the benefits, even in the short term.

The other focus here is on checking if the cost of developing the whole system won't surpass the proposed budget. Cost–benefit analysis showed that it was possible for the development of the application to progress. Most developing software I will be using are open source hence they will be easy and cheap to acquire. The risks foreseen during the development of project is significantly lesser compared with the benefits of developing it. The hardware with required processing power is also already available hence the cost is covered.

Cost Benefit Analysis

Cost-Benefit Analysis (CBA) is an important tool for evaluating the costs and benefits of a potential project or decision. The goal of a CBA is to compare the costs and benefits associated with a project or decision to determine if it is worth pursuing or not. In this case, the project is the development of a software system platform whereby citizens of a city can report issues they face and allow the city officials to be notified of civic issues as they occur for faster resolution and strengthen customer relationship.

	Costs	**Benefits**
Direct Tangible	• Equipment (hardware and software) • maintaining / updating costs • cost of hiring software engineers, designers, and other personnel • online web hosting solutions	• Reduced operating costs (decreased time by administrative offices providing forms and explaining procedures) • improved customer relationships due to better communication between citizens and city officials
Indirect tangible	• Anxiety related to use of information technology • Necessary updates to cloud-based resource database	• Universal of information to all citizens • Nurture relationships

		• Increased communication across citizens • Greater clarity for citizens regarding civic issues

Return on Investment

Investing in a software system platform that enables citizens to quickly report civic issues and allows city officials to be notified in real-time can have a significant return on investment. This system can help to identify and resolve issues faster, resulting in cost savings due to reduced man-hours and resources needed to address them. Additionally, it can help to strengthen the relationship between citizens and city officials, as citizens have less difficulty reporting issues and have a greater sense of trust in the council. As a result, the city can improve the overall quality of life for its citizens, as well as its reputation as a responsive and reliable governing body.

Apps can make anywhere from $10 up to $200 a day in average revenue from ads, and that's a conservative estimate for an app with only 1000 active users. However, the road to ad revenue is paved with bad ads, annoyed customers, and weak monetization strategies.

As of 2022, the population of Zimbabwe is 15.99 million according to world bank. Zimbabwe's internet penetration rate stood at 30.6 percent of the total population at the start of 2022, so we are expecting 4.8 million users, resulting in a expected revenue of $48000 in ads revenue.

To calculate anticipated ROI, I estimated the profits that will be generated by the project. This is done by taking the expected revenue minus the expected costs, and then dividing the result by the expected costs. For example, if this project is expected to generate $48000 in profits and cost $1000, then the anticipated ROI would be calculated as follows: ($48,000 - $1000) / $1000 = 470%.

3.3.3. Operational feasibility

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. It answers question “will the system work?” and how well the system would fit into the current business structures and once implemented how useable would it be. Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will 4 be used if it is developed and implemented.

Smartphone and mobile apps are here to stay, there’s very little doubt to this trend. Residents are part of a mobile society and they’ve become accustomed to interacting with organizations through mobile apps. Current trends indicate that mobile app usage will continue to outpace online browsing, both desktop and mobile browser combined. But mobile is not just another channel to reach residents; in many cases it may be the only manner by which your residents can digitally connect with you, particularly in regards to racial minorities, those without college education, and those of lower socioeconomic status. Mobile apps are bridging the digital divide.

After analyzing these operational requirements, we see that the proposed system is operationally feasible.

3.4.Requirements Analysis

3.4.1. Functional Requirements

Livewire library

It is a Laravel library that allows the chat function to refresh contents on the page without refreshing the page. It allows for live communication between the mobile app and the web app.

Web Application

This is the section where the municipal official views and change the resolution status of the reported issue. Each reported issue will have a unique ID that can be used by the citizen to search the issue he or she reported. The municipal official also sees the location of the reported issues.

Android application

The system will use the android application will be used to create an account for the user of the system. Information about the accounts is then saved in the database.

The user logs and reports using the Citizen Mobile Application and can attach an image using the camera module. There is a also a section that has the contact details of the citizen local authority/council so that if citizen wants to make a phone call to the civic agency, it is just one click away.

Critical:

1. Submit photo with GPS and inputted information
2. Option to submit a photo from phone gallery
3. Database retrieves photo, GPS, and any other inputted information
4. Filter/sort options for database
5. Search through database
6. Escalate repeats
7. Tag/label items that have been fixed

Recommended:

1. Show how many people have reported the same problem
2. Filter spam
3. Export data from database

Suggested:

1. Provide numbers to call

2. Notify the person who submitted the problem when it is fixed

3. Provide a log in for administrative use

4. Create a way for citizens to volunteer to fix problems

3.4.2. Context Level Diagram (DFD)

The diagram below shows the interaction of the citizen and public relations worker, and the system.

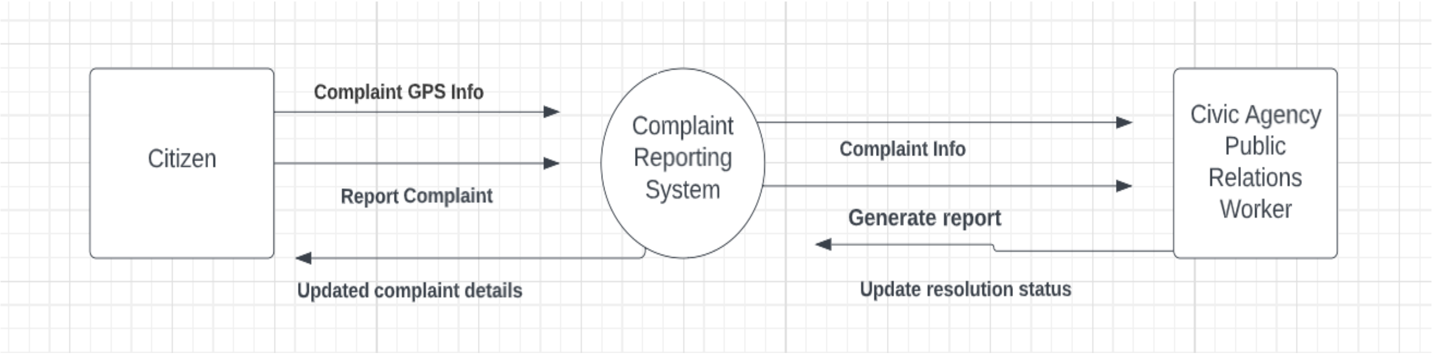


Figure 6: Context Level Diagram showing the interaction of the citizen and public relations worker

3.4.3. DFD Level 1

The use-case diagram below shows interactivity between the consumer, household and system administrator with the LEUM system. Household through the microcontroller sends usage data, whilst receiving control instructions. The usage data will be interpreted and presented to the consumer and system administrator in the appropriate format.

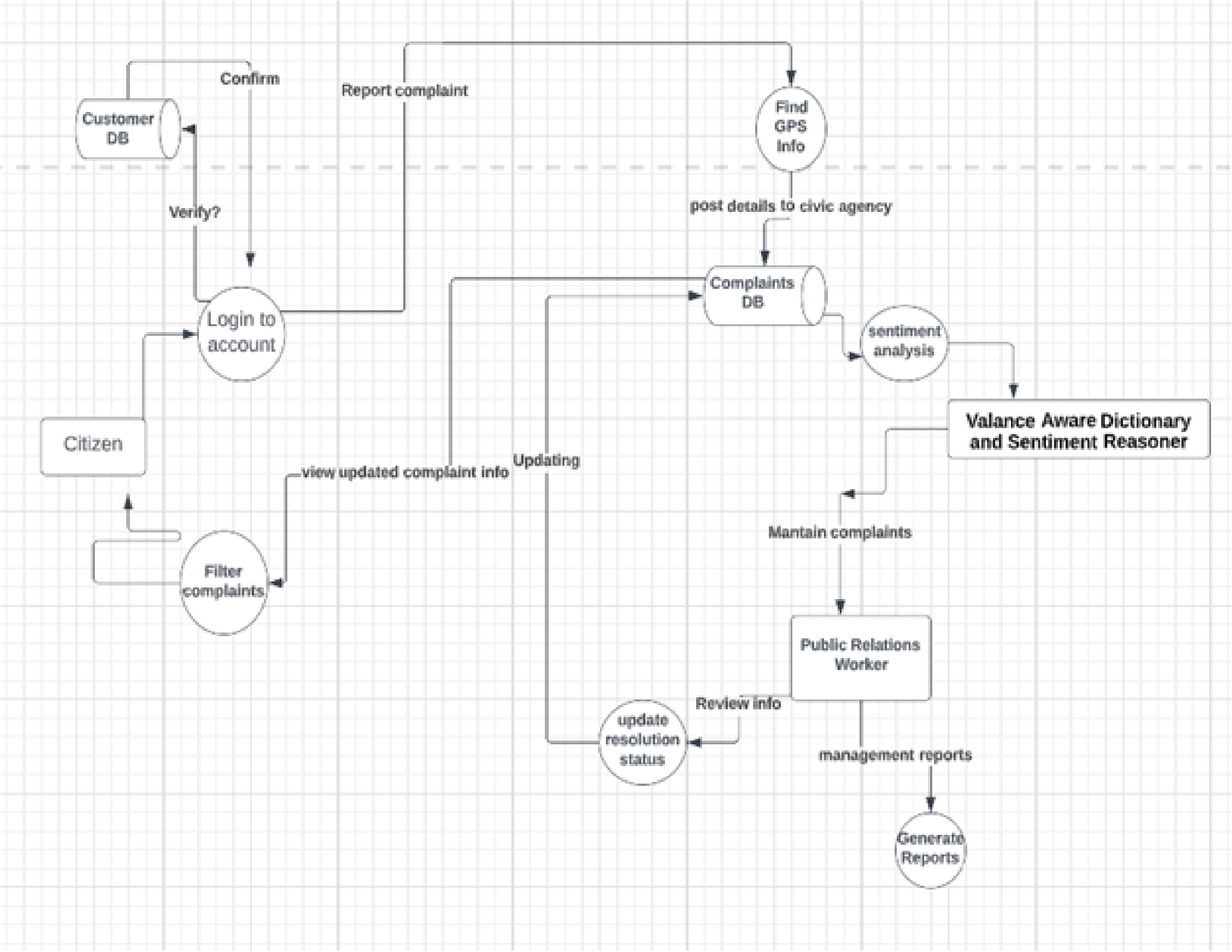


Figure 7: Level 1 DFD Context diagram

3.4.4 Use Cases

The mobile application is designed to be used by one main group of users, the general public. The public can populate the database with issues in their community. We planned for the web application to have both an admin user and a general user. Though both users would be able to query the database, only the admin user would be able to make changes to it, as shown in figure 1

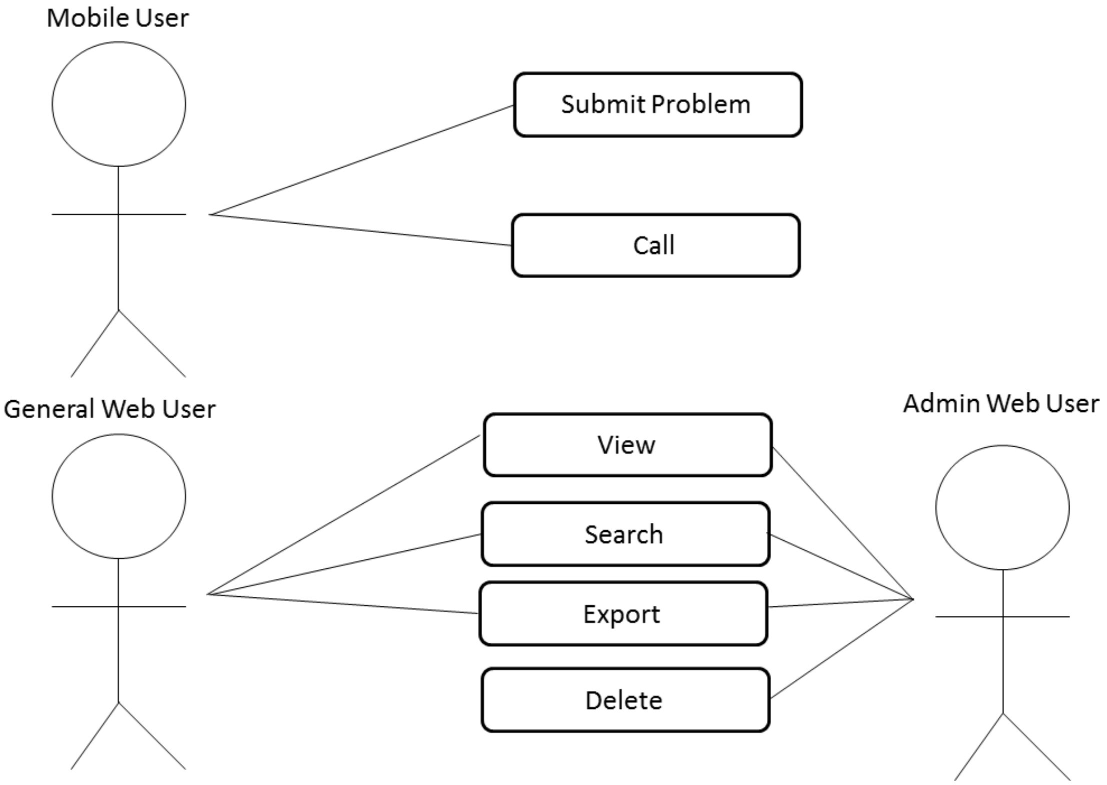


Figure 8: Use Case Diagram

3.5.Non-functional requirements

3.5.1. Performance

Performance in this regard refers to how well the system reads, interprets, and transmits usage data as well as parsing control instructions from the users. The system should have reasonable response time. The system is also supposed to be scalable in terms of management of numerous virtual households concurrently.

3.5.2. Usability

Usability is a measure of how easily a user can manipulate the system through its web interface to get a desired output. The proposed system should make use of modern methods of web development which offer amenities such as good aesthetics. The user should feel good when they interact with the system and not get frustrated or intimidated.

3.5.3. Security

Security refers resilience against potential harm which bears potential to mar the user's experience with the system. All complaints and information will be separated and accessed through the user providing a valid username and password. The password must be encrypted and hashed within the database so that even if the user's authentication details are intercepted, the interceptor will not be able to make sense of it.

3.6.`Interface requirements

The interface requirements specify how the proposed system is to interact and transmit information to its users. The proposed system should have a simplistic, consistent user-friendly interface that is easy to use, that is reasonably responsive. The main user-interface requirements for the proposed system are listed below.

- A minimalistic design theme that is precise and simplified.
- The imagery and iconography should resemble real life entities that the user is likely to be already familiar with.
- The interface should be highly responsive, appropriately providing feedback after every user`s action or request.
- The web interface should be responsive to the user`s device screen size, it should be optimized for mobile devices as well as the traditional desktop or laptop display resolution.

3.7.Technical Requirements

HARDWARE REQUIREMENTS

- Computer with at least 2GB RAM and 10GB Hard Disk Space

SOFTWARE REQUIREMENTS

- Windows Operating system
- Web browser
- Apache Web Application Server
- MySQL Database Server
- PHP 8.0
- Python 3.1

3.8.Assumptions

The implementation of this research holds the following assumptions

I. Assumption: A significant number of citizens in Zimbabwe have access to smartphones and are familiar with using mobile applications.

II. Assumption: The local government body in Zimbabwe has the necessary infrastructure and resources to receive and manage a large volume of civic complaints.

III. Assumption: The citizens are willing to participate and provide feedback on civic issues through the Mushandirapamwe platform.

IV. Assumption: The image processing algorithms used to determine complaint severity are accurate and reliable.

V. Assumption: The sentiment analysis algorithm used to classify user feedback accurately represents the opinions and sentiments of citizens.

VI. Assumption: The central server that receives and manages complaints is secure and protected against unauthorized access or data breaches.

VII. Assumption: The relevant agencies notified by the central server have the resources and ability to address and resolve the reported issues promptly.

VIII. Assumption: The citizens using the Mushandirapamwe platform are providing genuine and accurate information and not misusing the platform for personal gains or spreading false information.

3.9. Conclusion

This chapter has discussed mainly the feasibility of this research as well as the various requirements for implementation of the prototype. The remaining chapters include the design process, implementation strategies, evaluation of the implemented solution and finally the conclusion of the research.

4. CHAPTER FOUR - *Design*

4.1.Introduction

This Chapter describes the component setup and function of the proposed system. It demonstrates how the components will be put integrated in order to fulfil the project objectives. The chapter reports on the following areas:

- Proposed Solution
- Solution Architecture
- Constraints
- Security Design
- Systems Design Models
- Database Modelling
- Algorithm Design
- Interface Design

4.2.Proposed Solution

The app is built on normal client - server architecture with Android App as client and Web App as server part. This App is divided into two sections: Citizens (Android App), Web Application (Cooperator). An android app where end user gets registered and logins to app. Here the end user can access details of nearby corporator details and upload an image for grievances in the area. This uploaded image will be reported to city corporation office and it will be accessed by web application(portal) and approve it after confirming the image is fake or real.Corporator can take immediate action by communicating with the respective people and after the issue has been resolved, he has to update the compliant by uploading issue solved pictures. This information can be viewed by end users in their android apps. The end user app also provides information like nearest corporator access using GPS technology.The suggested system's goal is to provide a system with better facilities. The proposed solution can overcome all of the existing system's flaws. The method ensures sufficient security while also reducing manual labour. The current system has various flaws and many more challenges to overcome in order to function properly. The suggested solution aims to remove or mitigate these issues to some degree. The proposed system will assist the user in reducing effort and mental stress. The proposed system enables the user to operate in a user-friendly manner, allowing him to complete his tasks without wasting time.

4.3.Solution Architecture

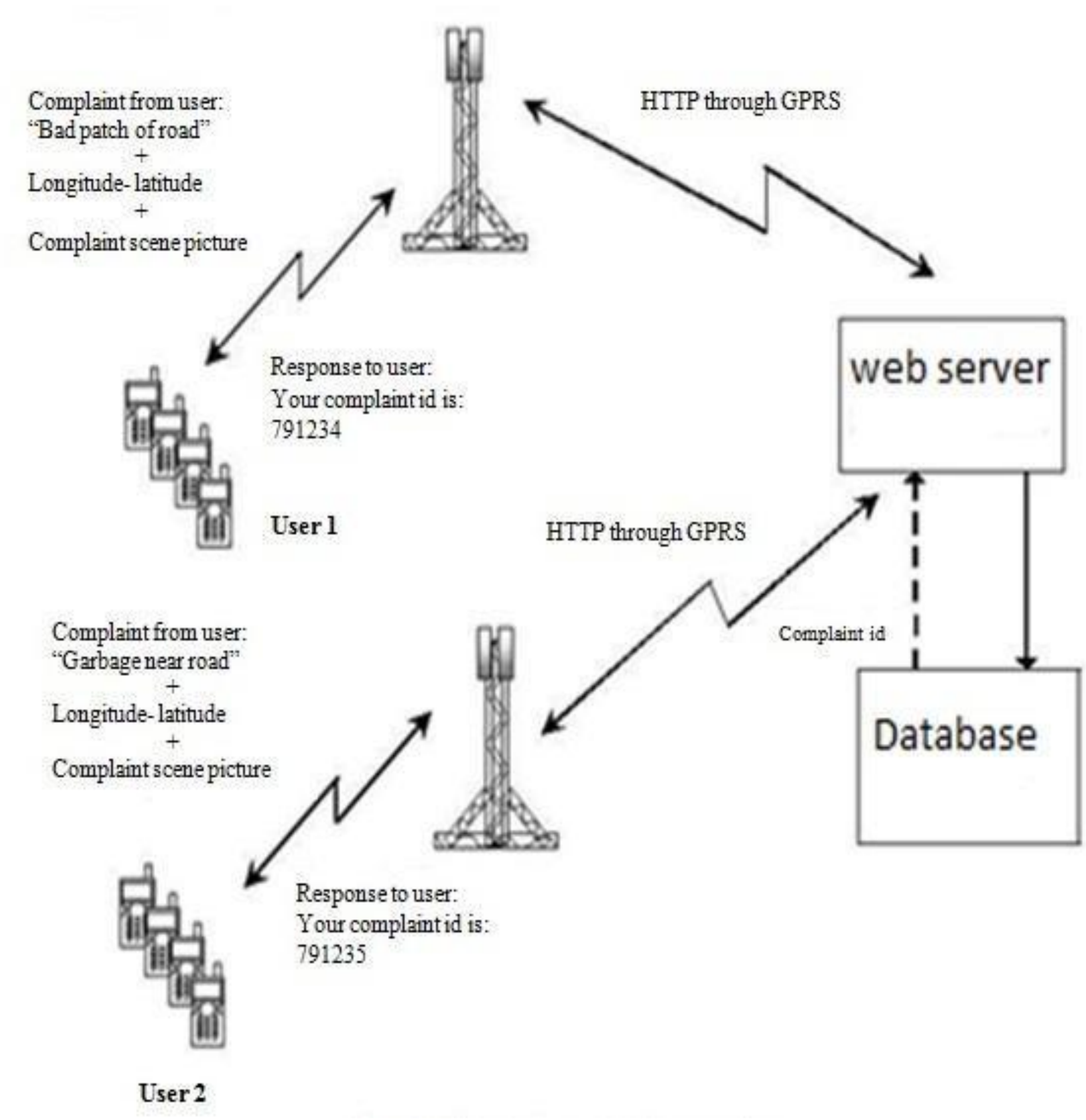


Figure 9: Architectural Diagram

4.4.Constraints

Constraints are factors that can limit or affect the development and implementation of a project. Several constraints that needed to be considered. Firstly, internet connectivity and smartphone ownership may be limited in some areas of the city, which could limit the accessibility and use of the application by citizens. Additionally, the project may face challenges in terms of getting buy-in and participation from citizens, as not everyone may be willing or able to report issues through the application. Another potential constraint is the availability and capacity of the civic agencies to effectively respond to and address the reported issues, which could slow down the resolution process and reduce the effectiveness of the application. Finally, there may be legal and regulatory constraints related to data privacy, ownership, and security that need to be considered and addressed to ensure the safety and protection of user data. Despite these constraints, the project team will work to address these challenges to ensure the successful implementation and adoption of the Mushandirapamwe application.

4.5.Security Design

In the Mushandirapamwe project, security is a top priority to ensure that user data and information are safe from unauthorized access, theft, or tampering. To achieve this, several security measures have been implemented.

Firstly, the user data is encrypted using industry-standard encryption protocols during transmission and storage to prevent unauthorized access. Additionally, the user's identity is verified during the registration process using authentication, including the user's phone number and email address.

Furthermore, the application has built-in mechanisms to prevent SQL injection attacks, cross-site scripting attacks, and other forms of web application attacks. Regular security audits are conducted to identify and address potential vulnerabilities in the system.

Lastly, the project team adheres to industry-standard security best practices, including frequent security updates, data backups, and disaster recovery plans to ensure the continuity of the service in the event of a security breach or system failure.

4.6.System Design Models

4.6.1. Activity Diagram

The activity diagram illustrates the dynamic behavior of the system showing the information flow from one activity to another and the workflows of stepwise actions, decisions, concurrency, and iterations.

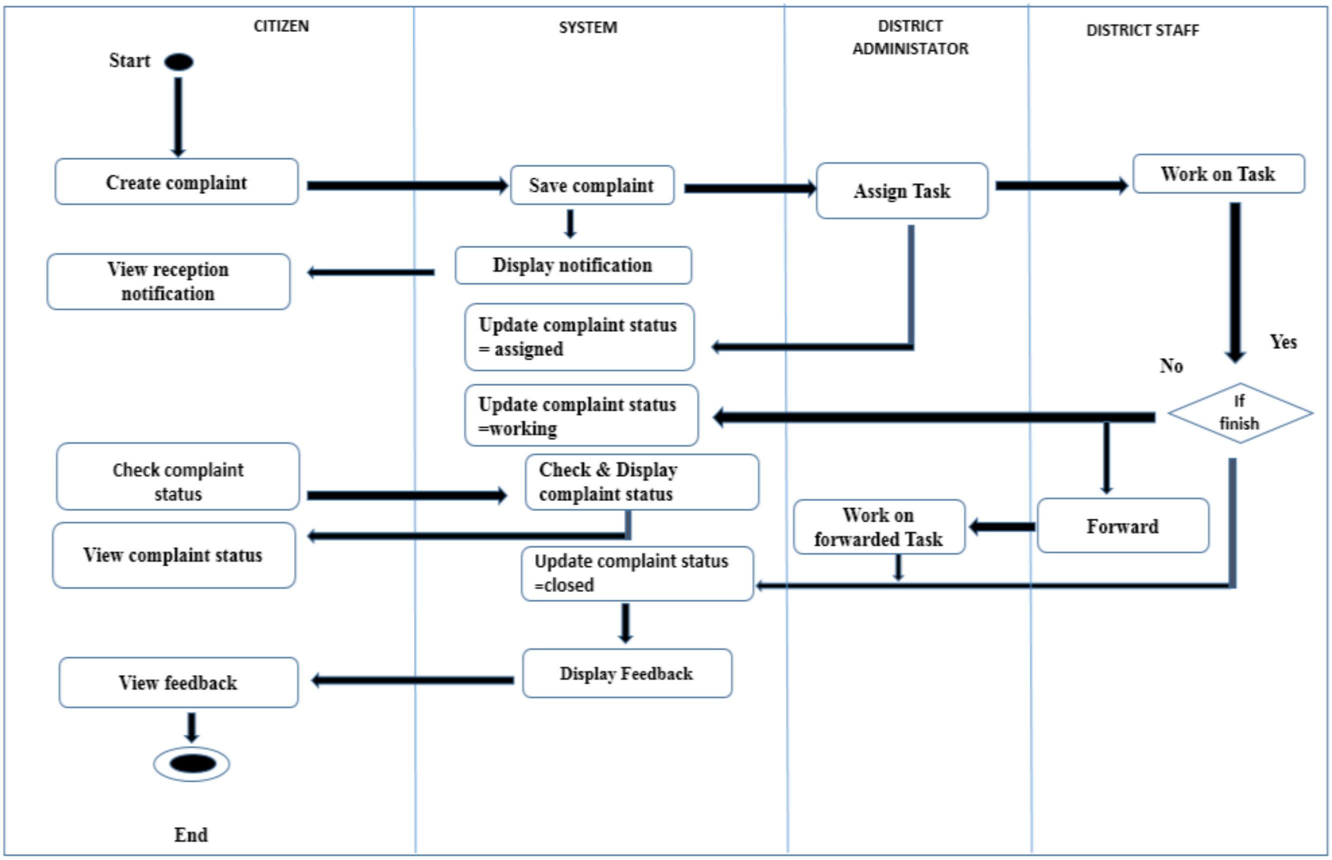


Figure 10: Activity Diagram

4.6.2. Class Diagram

The class diagram is a static diagram which represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application [7]. The class diagram of the proposed system is shown in the figure below.

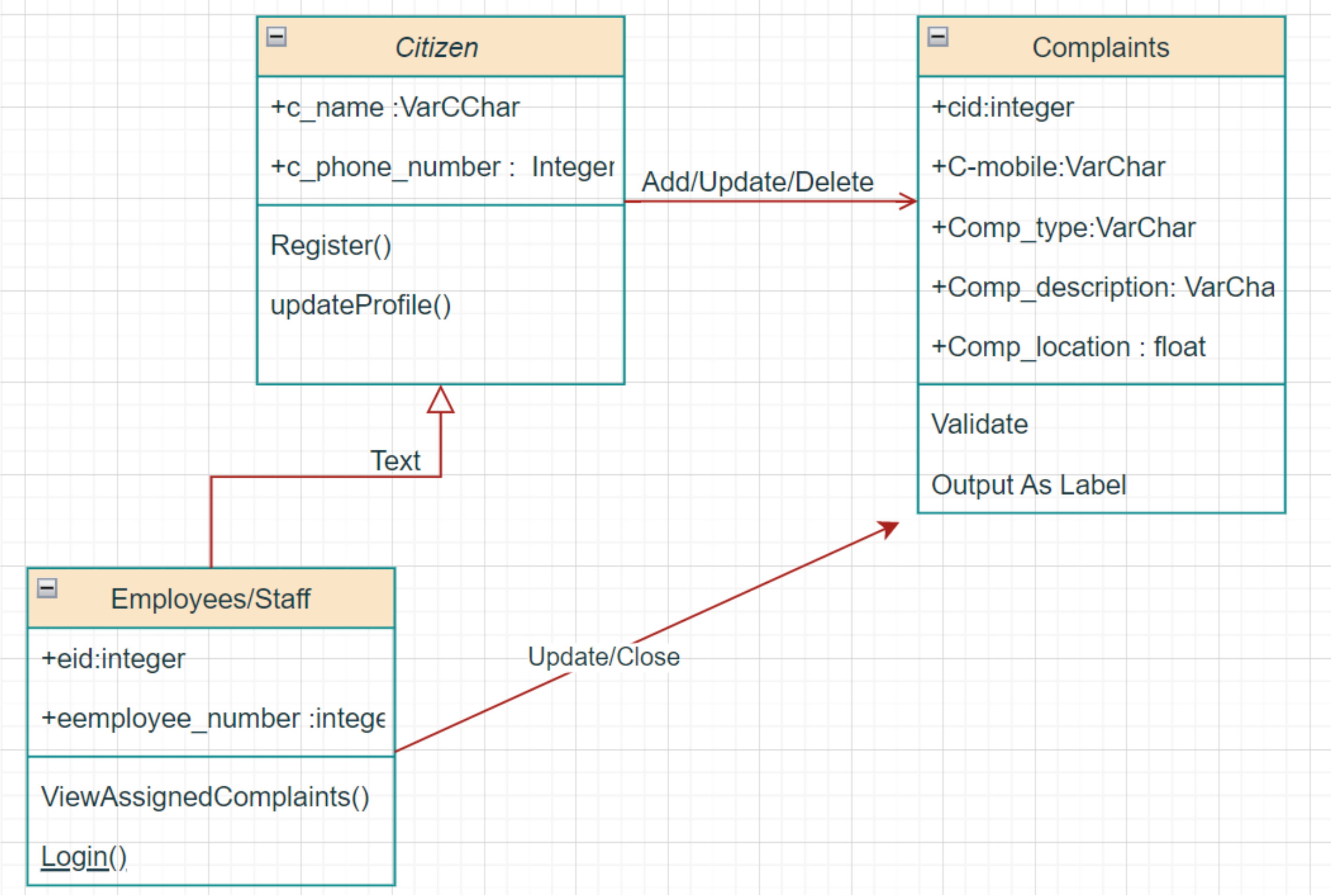


Figure 11: Proposed System Class diagram

4.6.3. Sequence Diagram

The sequence diagram below denotes the activities which take place from the time the user logs in a complaint.

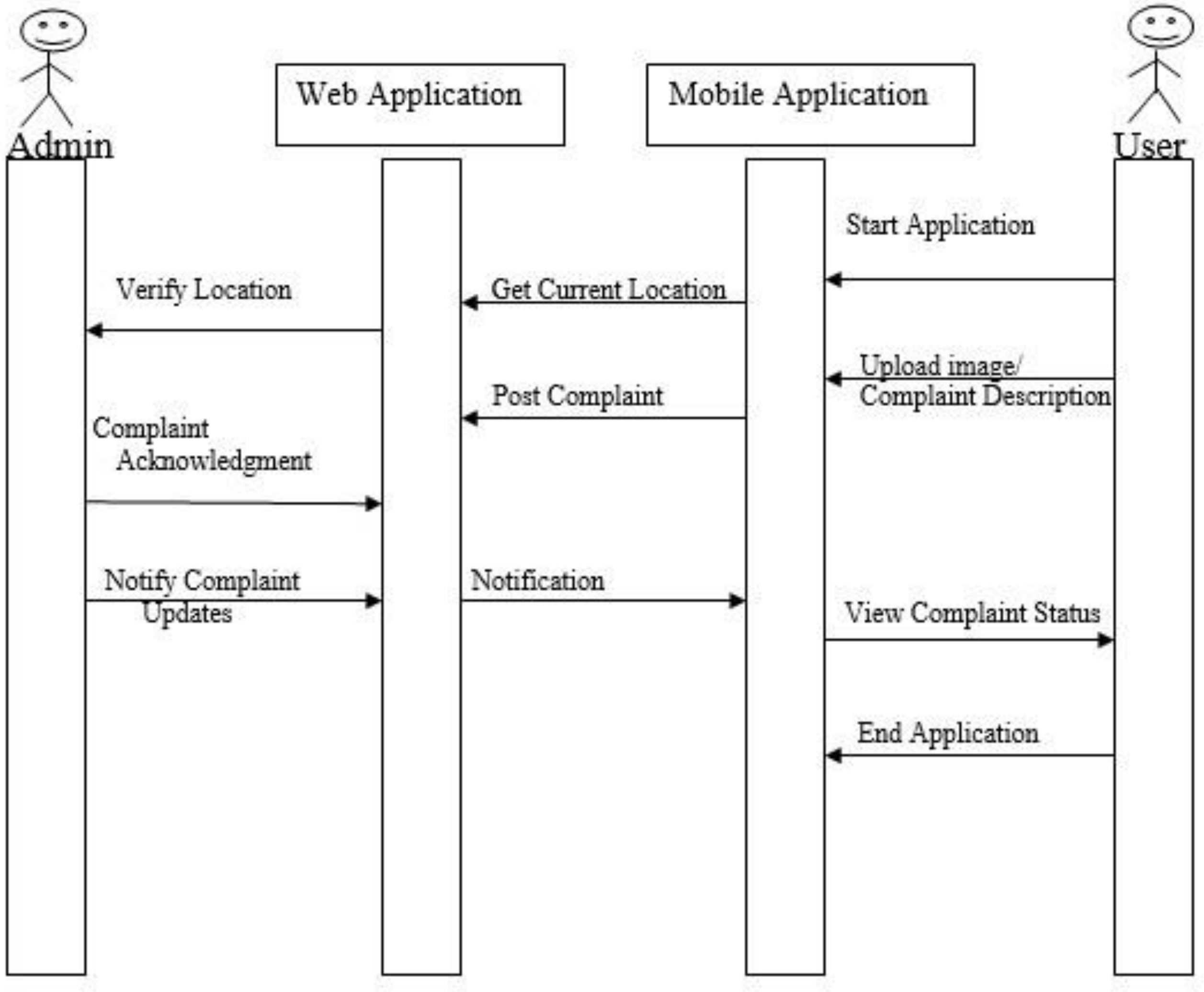


Figure 12: System Sequence Diagram

4.6.4. Deployment Diagram

The deployment diagram illustrates the hardware topology of the physical components of the proposed system. It describes the static deployment view of the system in the instance of running processing nodes. Below is deployment diagram for the design solution

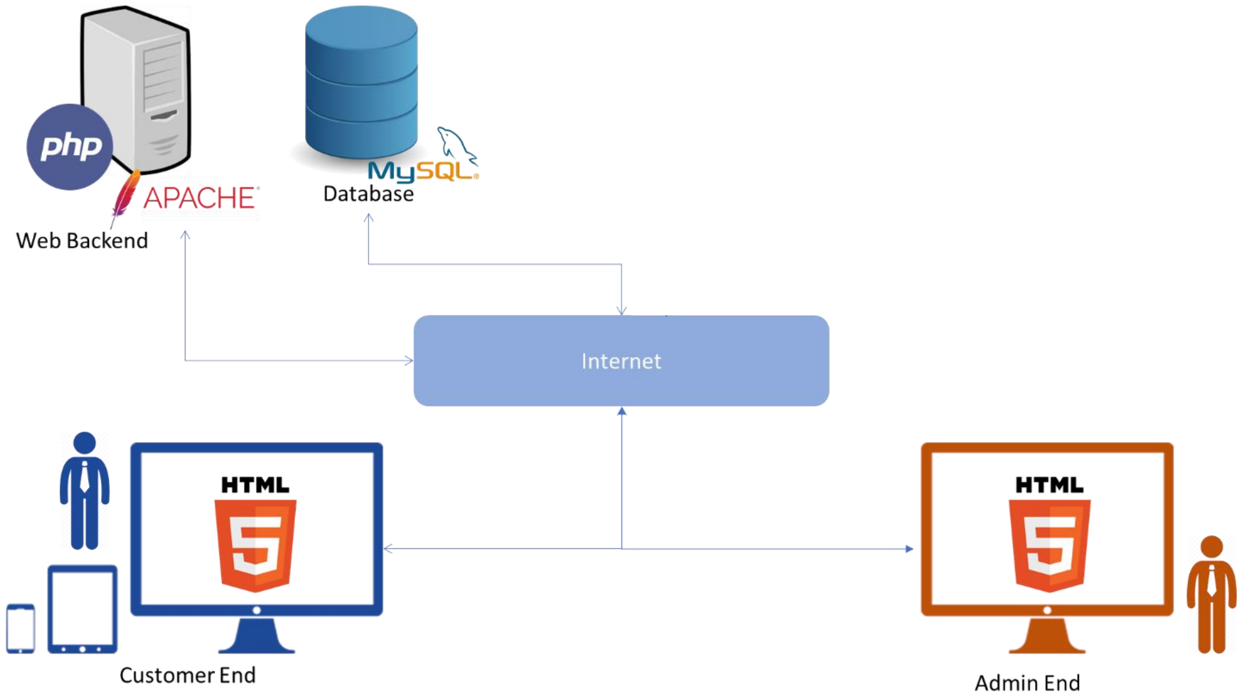


Figure 13: Deployment Diagram

4.7.Database Modelling

The database modeling process is when the databases for the proposed system are modeled a designed. These databases are central repositories of all the data and information. To model the databases, entity-relationship models are first designed. Entity-relationship diagrams illustrate the relationship between the entities that are involved in the system.

The conceptual entity-relationship models are implemented and the designed databases are created onto the database server, which in this case MYSQL was used

4.7.1. Entity-Relationship Diagram

The entity-relationship (ER) diagram is a specialized graphical illustration of the relationships between entities of the designed database. ER diagrams often make use of symbols to represent three different types of information:

- Boxes are used to represent database entities.
- Diamonds are to represent relationships.
- Ovals represent attributes.

The diagram below shows an entity relationship diagram for the proposed system.

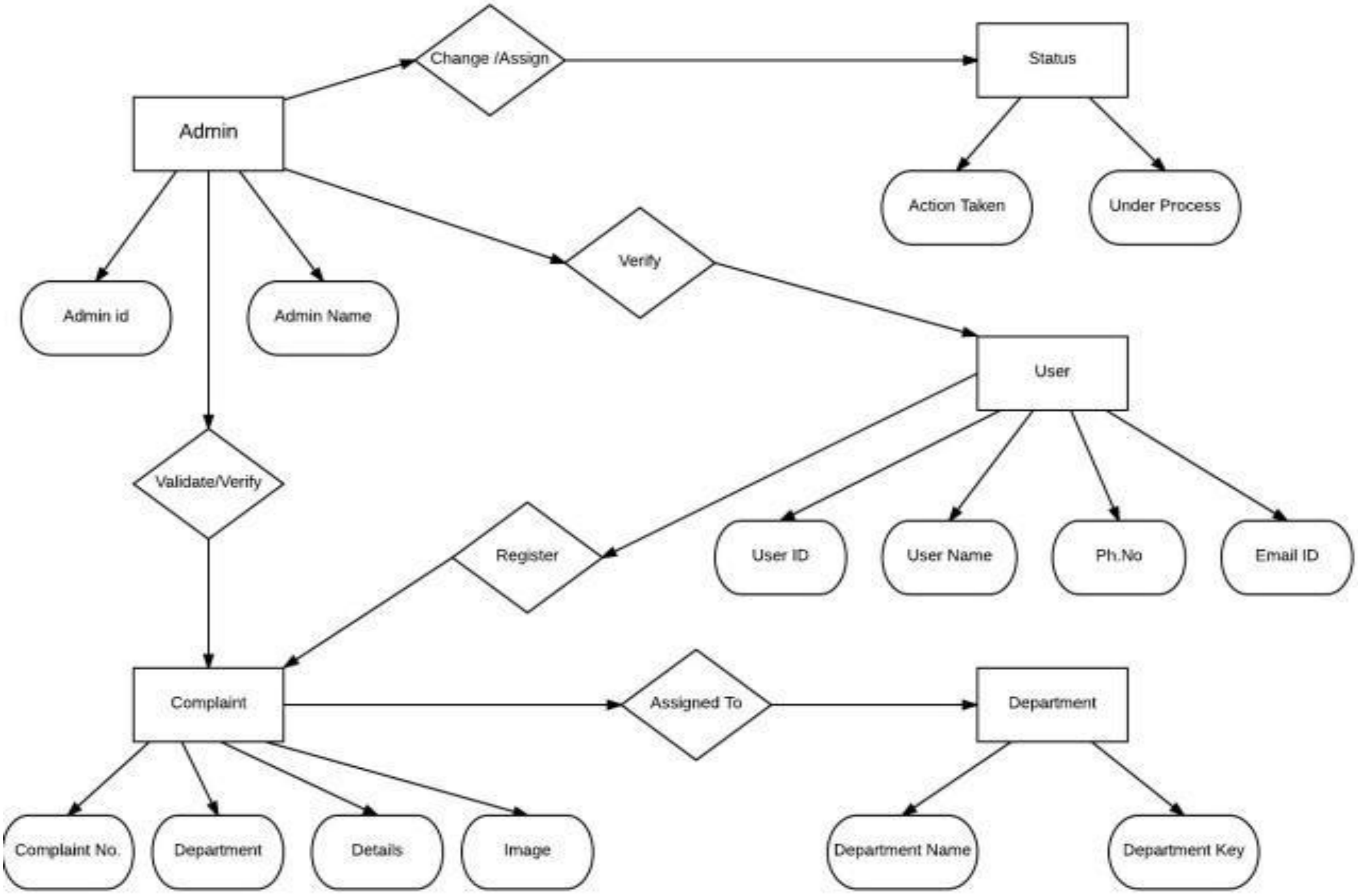


Figure 14: ER diagrams for the proposed system

4.7.2. Data Dictionary

The tables below show attributes and datatypes of the fields within the proposed system

Citizen Entity

Attribute	Description	Datatype and Size
citizen_id	Unique identifier for the users table	int
first_name	System user's name	varchar(50)
last_name	System user's surname	varchar(50)
phone_number	Combination of unique characters used to refer to a household within the system. A single user is capable of having many households	varchar(3)

Complaint Entity

Attribute	Description	Datatype and Size
id	Unique index for the complaints table	int
reported_at	Stores the virtual date and time generated when complaint is logged	datetime
status	Stores status of filled complaint	varchar (12)
description	Stores description of filled complaint	float

location	Stores coordinates information	float

Complaint Type Entity

Attribute	Description	Datatype and Size
id	Unique identifier for the complaint type table	int
title	Characters denoting the complaint title types to which the stored data belongs to	varchar(10)

Timeline Entity

Attribute	Description	Datatype and Size
id	Unique identifier for the timeline Table	int
hours	Characters denoting the hours to which the stored data belongs to	int
Complaint_type_id	Integer referencing complaint type in type table	int

4.7.3. Relational Schema

id	**description**	**status**	**type_id**	**location**
1	...	…..	…..	...

4.7.3.1. First Normal Form

id	description	status	type_id	location
1	...	…..	…..	...

type_id	Type_description
1	...

4.7.3.2. Second Normal Form

4.8.Algorithm Design

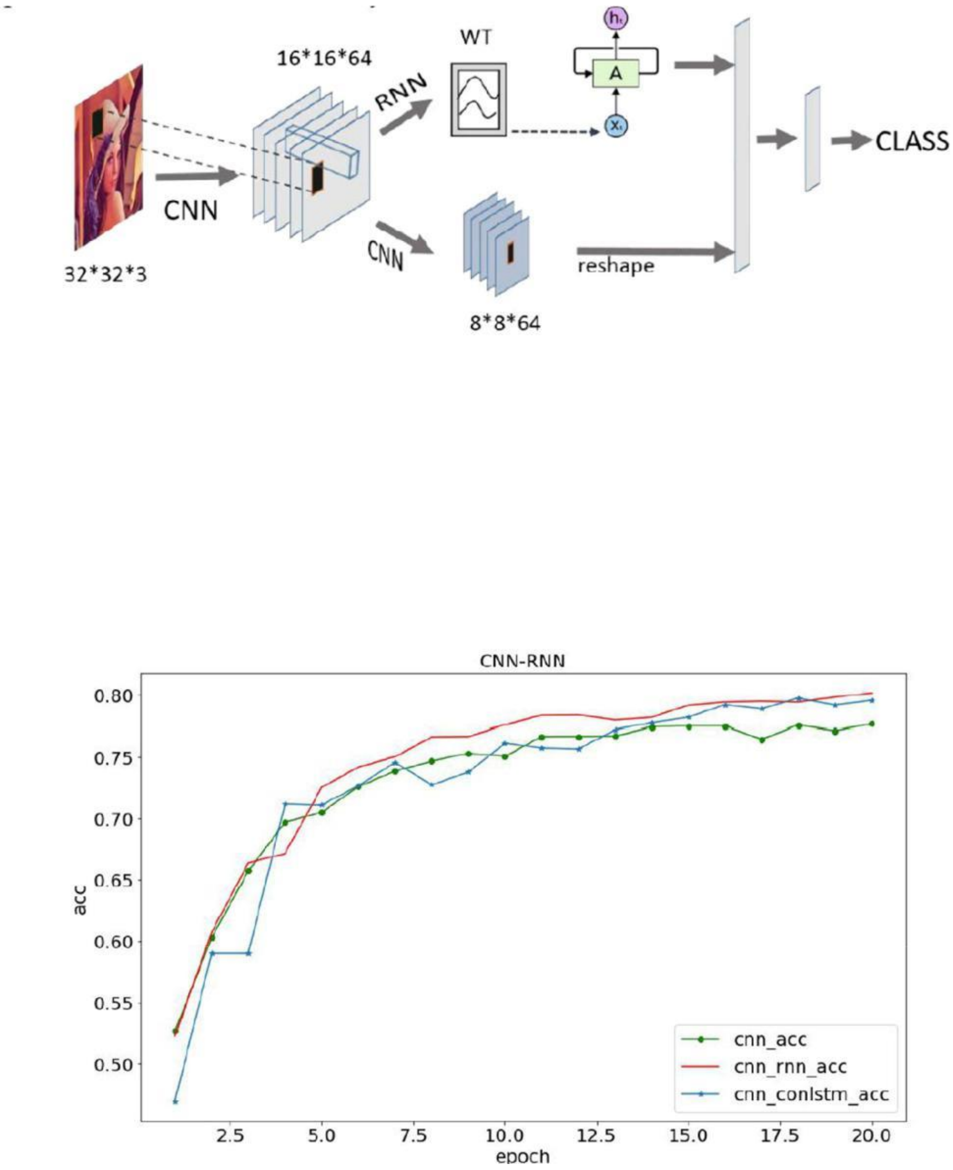


Figure 15: Algorithm Design

Convolutional Neural Networks, or CNNs have been at first designed to map photo information to an output variable. They have verified so powerful that they may be the go-to technique for any form of prediction hassle regarding photo information as an input. More typically, CNNs paintings properly with information that has a spatial relationship. Recurrent Neural Networks, or RNNs, have been designed to paintings with series prediction problems.

RNNs in trendy and LSTMs especially have acquired the maximum achievement while running with sequences of phrases and paragraphs, typically referred to as herbal language processing. However they may be used for sort of different applications which include photo processing as properly. A hybrid Algorithm classifies images based on their severity. And Rank higher sever issue with good rank than others.

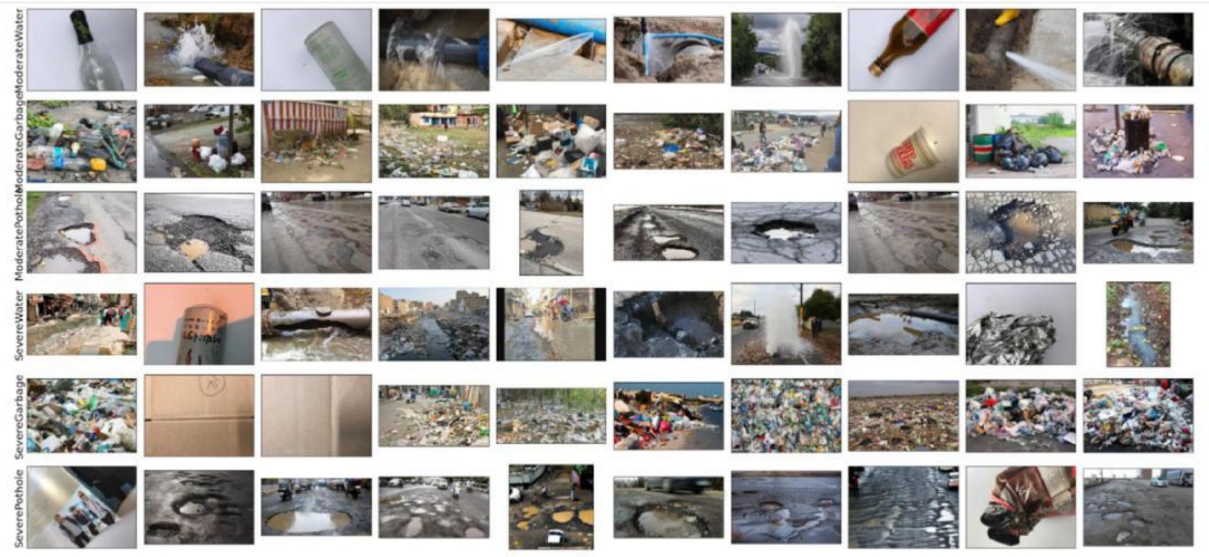
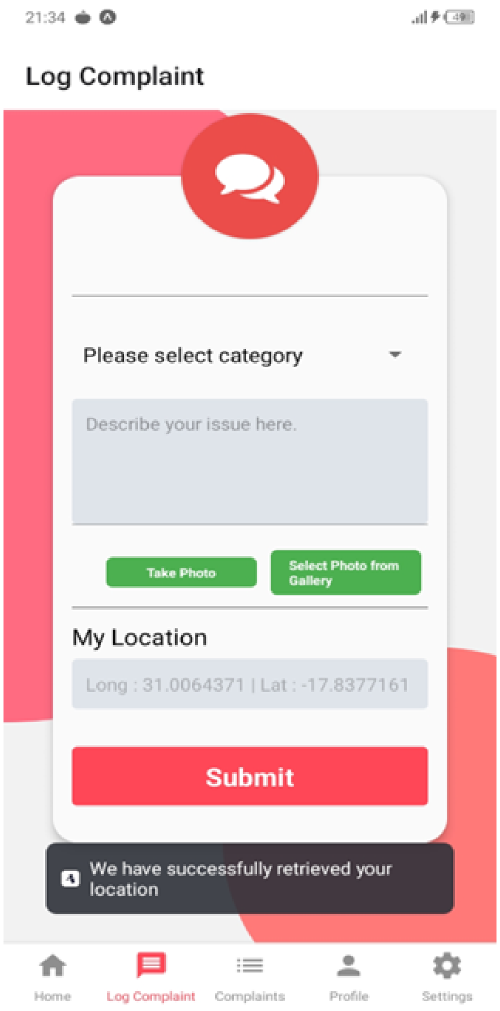


Figure 16: CNN hybrid algorithm dataset sample images

4.9.Interface Design

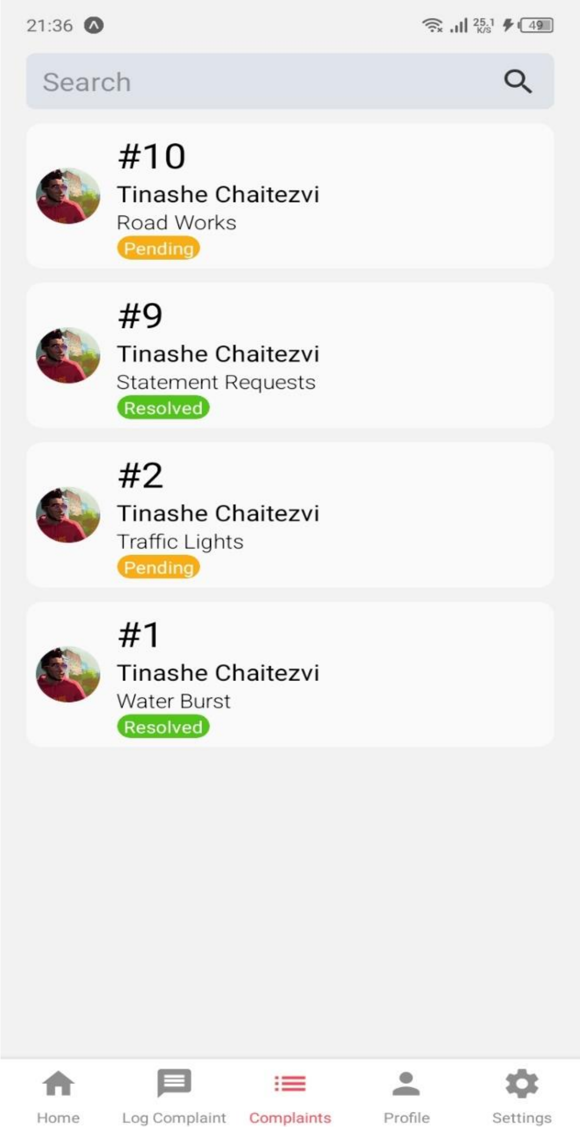
The interface design section of the Smart Civic Issues Reporting System for Local Authorities is crucial to ensure that citizens have an easy-to-use platform to report their civic issues, which will help local authorities address these issues faster. The existing process for reporting problems, such as illegal waste dumping, road damage, or street cleaning, requires lengthy steps and procedures that are challenging for citizens to navigate. To solve this problem, we have developed an Android application called Mushandirapamwe that allows citizens to report issues with just two clicks. This application is a platform for exchanging issues between civil servants and the general public. The system is transparent and approachable, allowing the people of a locality to register complaints, check if they have been rectified and support existing complaints. The application is designed to maximize the potential of crowdsourcing and take eParticipation to a new level. The user can log in and capture an image of the issue which is automatically geotagged, and this helps locate the exact location of the issue. Complaints are a valuable source of feedback to improve the infrastructure and condition of our city. Therefore, the most important task to maintain regular hygiene in the city is to efficiently report citizens' problems to public service authorities. This section will cover the design principles used in the development of Mushandirapamwe to create a seamless user experience that is easy to navigate, understand, and use.



Long : 31.0064371 | Lat : -17.8377161

We have successfully retrieved your location

Figure 17: Citizen logging complaint



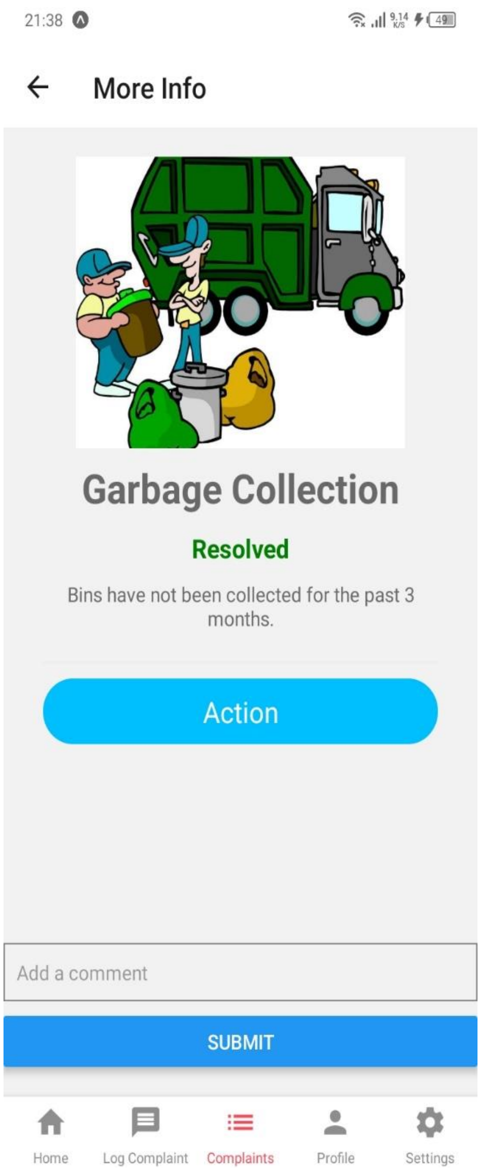
#10
Tinashe Chaitezvi
Road Works
Pending

#9
Tinashe Chaitezvi
Statement Requests
Resolved

#2
Tinashe Chaitezvi
Traffic Lights
Pending

#1
Tinashe Chaitezvi
Water Burst
Resolved

Figure 18: Citizen's All reported complaints

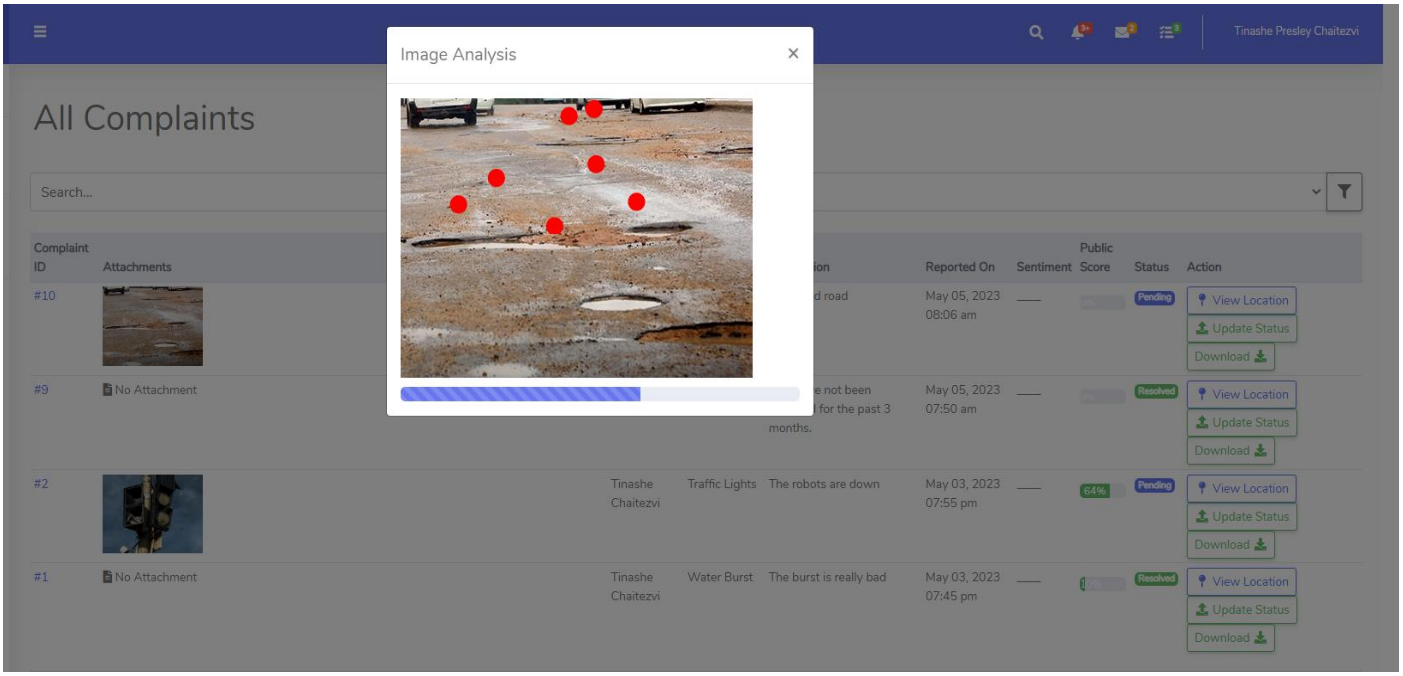


Garbage Collection

Resolved

Bins have not been collected for the past 3 months.

Figure 19: CNN hybrid algorithm dataset sample images



All Complaints

Search...

Complaint ID	Attachments			…ion	Reported On	Sentiment	Public Score	Status	Action
#10				…d road	May 05, 2023 08:06 am	——		Pending	View Location Update Status Download
#9	No Attachment			…e not been …l for the past 3 months.	May 05, 2023 07:50 am	——		Resolved	View Location Update Status Download
#2		Tinashe Chaitezvi	Traffic Lights	The robots are down	May 03, 2023 07:55 pm	——	64%	Pending	View Location Update Status Download
#1	No Attachment	Tinashe Chaitezvi	Water Burst	The burst is really bad	May 03, 2023 07:45 pm	——		Resolved	View Location Update Status Download

Figure 20: System interface design

4.10. Conclusion

This chapter has detailed the high-level interface designs, the system architecture, the database models, and the algorithm designs. The remaining chapters report on implementations strategies used, the coding patterns and conventions considered, the software development methodology implemented, the evaluations techniques deployed and finally the concluding evaluations of the overall success of this project.

5. CHAPTER FIVE – *Implementation and Testing*

5.1.Introduction

The implementation phase of our project marks a significant milestone in our efforts to bridge the gap between citizens and government agencies. Throughout the development process, our team has worked diligently to create an Android application that is user-friendly, accessible, and effective in facilitating communication between citizens and civic agencies. In this chapter, we will discuss the technical details of how we implemented the features and functionality of the application, from the user interface to the back-end server infrastructure. We will also provide insights into the challenges we faced during the implementation phase and how we overcame them. By detailing the process of implementing our application, we hope to provide a transparent account of the steps we took to bring our vision to life and inspire other developers to tackle similar challenges in their communities.

5.1 Pseudo Code

```
## Function: store()
## Parameters: Request $request
## Returns: Response

## Description:
This function stores a complaint in the database.

## Pseudocode:

1. Log that the store function has been reached.
2. Get the complaint from the database.
3. Log the complaint.
4. If the complaint exists and the category is not road works, increment the complaint score.
5. Otherwise, create a new complaint.
6. Save the complaint.
```

```
7. Return a success message.
8. Catch any exceptions and log them.
9. Return an error message.
```

5.2 Sample of real code

```
public function store(Request $request)
{
// Log that the store function has been reached
Log::info('Store function reached');

try {
// Get the complaint from the database
$complaint = Complaint::where('category', $request->input('category'))
->whereBetween('latitude', [$request->input('latitude') - 0.01, $request->input('latitude') + 0.01])
->whereBetween('longitude', [$request->input('longitude') - 0.01, $request->input('longitude') + 0.01])
->first();

// Log the complaint
Log::info($complaint);

// If the complaint exists and the category is not road works, increment the complaint score
if ($complaint && $request->input('category') != 'Road Works') {
```

```
$complaint->score += 10;
Log::info('Complaint score incremented');
} else {
// Otherwise, create a new complaint
$complaint = new Complaint();
$complaint->full_name = $request->input('full_name');
$complaint->phone_number = $request->input('phone_number');
$complaint->category = $request->input('category');
$complaint->description = $request->input('description');
$complaint->latitude = $request->input('latitude');
$complaint->longitude = $request->input('longitude');

// If there is an image, save it to the `public/images/complaints` directory
$image = $request->file('complaintImage');
if ($image) {
$filename = $image->store('public/images/complaints');
$complaint->image_url = $filename;
}

Log::info('Complaint saved successfully');
}

// Save the complaint
$complaint->save();

// Return a success message
return response()->json([
"code" => 200,
"message" => 'Complaint logged successfully'
]);
} catch (\Exception $exception) {
```

```php
// Log the exception message
Log::info($exception->getMessage());

// Return an error message
return response()->json([
"code" => 500,
"message" => $exception->getMessage()
]);
}
}
```

```python
# Import required libraries
import os
import numpy as np
import keras
from keras.models import Sequential
from keras.layers import Dense, Conv2D, MaxPooling2D, Flatten, LSTM, TimeDistributed
from keras.preprocessing.image import load_img, img_to_array
from keras.utils import to_categorical

# Define CNN architecture
cnn_model = Sequential()
cnn_model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)))
cnn_model.add(MaxPooling2D((2, 2)))
cnn_model.add(Conv2D(64, (3, 3), activation='relu'))
cnn_model.add(MaxPooling2D((2, 2)))
cnn_model.add(Conv2D(128, (3, 3), activation='relu'))
cnn_model.add(MaxPooling2D((2, 2)))
cnn_model.add(Flatten())
```

```
# Define RNN architecture
rnn_model = Sequential()
rnn_model.add(LSTM(256, return_sequences=True, input_shape=(10, 128)))

# Combine CNN and RNN models
combined_model = Sequential()
combined_model.add(TimeDistributed(cnn_model, input_shape=(10, 224, 224, 3)))
combined_model.add(TimeDistributed(Flatten()))
combined_model.add(rnn_model)
combined_model.add(Dense(1, activation='sigmoid'))

# Compile the model
combined_model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Load image data
data_dir = '/Downloads/images'
labels = {'positive': 1, 'negative': 0}
x = []
y = []

for label in labels:
for file in os.listdir(os.path.join(data_dir, label)):
img = load_img(os.path.join(data_dir, label, file), target_size=(224, 224))
img = img_to_array(img)
x.append(img)
y.append(labels[label])

x = np.array(x)
y = to_categorical(y)
```

```
import React, { useState, useEffect, useLayoutEffect } from "react";
import {
StyleSheet,
Text,
View,
TextInput,
} from "react-native";
import { withBadge } from "react-native-elements";
import { useRoute } from "@react-navigation/native";

export default function Complaints({ navigation }) {
const route = useRoute();

const [searchText, setSearchText] = useState("");
const [users, setUsers] = useState([]);
const [filteredUsers, setFilteredUsers] = useState([]);
const [isConditionMet, setIsConditionMet] = useState(false);
const [animating, setAnimating] = useState(true);
const [isLoading, setIsLoading] = useState(true);
const [refreshing, setRefreshing] = React.useState(false);
const [update, setUpdate] = useState(false);
const [loading, setLoading] = useState(true);

useEffect(() => {
StatusBar.setBarStyle("dark-content", false);
axios
.get(`http://172.16.9.37:8008/api/retrieve-complaints`)
.then(function (response) {
```

```
setIsLoading(false);
return response.data;
})
.then((responseData) => {
setAnimating(false);
setFilteredUsers(responseData);
setLoading(false);

ToastAndroid.show("List retrieved successfully", ToastAndroid.SHORT);
})
.catch((error) => {
// Handle any errors that occur
console.log(error);

showMessage({
message: "Failed to retrieve. Check your network connection",
type: "danger",
});
});
setUpdate(false);
}, []);
```

Significance of coding conventions

- They foster the standardization of the structure and coding style of an application.
- Results in the precise, readable and unambiguous source code that is consistent with other language conventions.
- Ensures the source code to has comprehensive easy to read consistent layout.
- Minimizes the Risk of Project Failure.

- Eliminates unnecessary complexity and making maintaining the source code much easier, bugs become easy to pinpoint and correct.

5.2.Coding Conventions

Adhering to coding conventions is an important aspect of software development that ensures consistency, readability, and maintainability of the codebase. Throughout the development of our Android application, we followed established coding conventions for both the React Native framework and Laravel backend to improve the quality and usability of our code. For example, we ensured that our code was well-organized, properly commented, and used descriptive variable and function names to enhance readability. We also implemented a consistent naming convention for all functions and variables to improve consistency and make the code easier to understand. Additionally, we used indentation and whitespace consistently to improve readability and used standard practices for error handling and data validation. In addition to these conventions, we followed best practices for integrating the CNN RNN model into our application. This involved following established naming conventions and file structures for machine learning models and implementing standardized data pre-processing and postprocessing techniques. By adhering to these coding conventions and best practices, we were able to streamline the development process, improve code quality, and facilitate collaboration among team members.

- **PHP**

The PHP Hypertext Preprocessor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web-based software applications. PHP is a server-side scripting language that is embedded in HTML.

Features of PHP:

- Interpreted
- Faster
- Open Source

- Platform Independent
- Case Sensitive
- Error Reporting
- Real-Time Access Monitoring
- Loosely Typed Language

The guidelines followed in this project include the below listed:

- **Indenting and Line Length**

Use an indent of 4 spaces and don't use any tab because different computers use a different setting for the tab. It is recommended to keep lines at approximately 75-85 characters long for better code readability.

- **Control Structures**

These include if, for, while, switch, etc. Control statements should have one space between the control keyword and opening parenthesis, to distinguish them from function calls. You are strongly encouraged to always use curly braces even in situations where they are technically optional. Examples are shown below.

```
if ((condition1) || (condition2)) {
action1;
}elseif ((condition3) && (condition4)) {
action2;
}else {
default action;
}
switch (condition) {
case 1:
action1;
break;
case 2:
```

```
action2;
break;
default:
defaultaction;
break;
}
```

5.3.Coding Strategy

Our coding strategy for this project was focused on creating a maintainable and scalable codebase that can be easily extended and modified as needed. To achieve this goal, we adopted a modular approach to development, breaking down our code into smaller, reusable components that can be easily tested and updated. For the front-end of our application built with React Native, we followed a modular architecture pattern that allows for better separation of concerns between different components. We made use of libraries such as Redux and React Navigation to handle state management and navigation respectively. On the back-end, we used the Laravel framework to implement a microservices architecture pattern, which helped to separate our application into smaller, independent services that can be easily maintained and scaled. Additionally, we followed the SOLID principles of software design, which promote code that is modular, maintainable, and extensible. We ensured that our code was well-organized, used descriptive naming conventions, and followed established design patterns such as the Repository pattern to improve the maintainability and scalability of our codebase. To ensure that our code was of high quality, we implemented a rigorous testing and review process, which involved automated testing with tools such as Jest and PHPUnit, peer code reviews, and continuous integration with GitHub Actions to catch and fix errors early on in the development process. By following this coding strategy, we were able to create a codebase that is easy to understand, maintain, and extend, and that can be scaled to meet the needs of our users and the civic agencies that we serve.

5.3.1. Summary of Coding Strategy

Primary Programming Languages	PHP, Python, Javascript
Architectural Coding Pattern	Model-View-Controller (MVC) OOP
Development Methodology	Agile Extreme Programming
IDE	Sublime Text, Visual Studio, Arduino
3rd Party Libraries	Bootstrap, jQuery, PHP-ML
Development Environment	Windows 10, x86 Apache application server MySQL Database Server FileZilla File Server Arduino

5.4.Coding Review

As part of our coding strategy, we implemented a rigorous coding review process to ensure that our codebase was of high quality and met the established coding conventions and best practices. We used GitHub pull requests to facilitate code reviews, which allowed team members to review each other's code and provide feedback before merging changes into the main codebase. During these reviews, we checked for adherence to coding conventions, proper variable and function names, proper use of comments, and proper implementation of error handling and data validation. We also checked for proper implementation of the CNN RNN model for image classification, including ensuring proper file structures, standardized data pre-processing, and postprocessing techniques. Additionally, we implemented VADER (Valence Aware Dictionary and sEntiment Reasoner) for sentiment analysis on the complaint description, which helped us gauge the tone and emotion of the text. This allowed us to prioritize and address issues based on their severity, urgency and impact on the public. In addition to these checks, we also used automated testing tools such as Jest and PHPUnit to catch any errors or bugs before merging changes. By implementing this rigorous coding review process, we were able to catch and fix issues early on in the development process, improve the quality of our codebase, and facilitate collaboration among team members.

Code Review Checklists

5.4.1. Structure Checklist

Review Item	**YES**	**NO**
Does the code completely represent and accurately implement the design?	✓	
Does the code conform to the applicable coding standards?	✓	
Is the code consistent in layout, syntax, indentation?	✓	
Are there any referenced functions or any unreachable code?		✓
Can any of the code be replaced by calls to external reusable components or library functions?		✓
Are there any blocks of redundant code that could be implemented once and reused?		✓
Are any code blocks that are unnecessarily complex that can be restructured or divided into numerous sub-blocks?		✓

5.4.2. Variables

Review Item	**YES**	**NO**
Are all variables defined with meaningful, consistent unambiguous names?	✓	
Do all variables accurately type or consistency typecasted?	✓	
Are there any redundant or unnecessary variables?		✓

5.4.3. Loops and Branches

Review Item	YES	NO
Are there any potential infinite loops?		✓
Are loop termination conditions obvious and reachable?	✓	
Are loop indexes or loop subscripts declared and initialized, outside the loop itself?	✓	

5.4.4. Defensive Programming

Review Item	YES	NO
Are imported data blocks and input parameters tested for validity and completeness?	✓	
Are files checked for existence before attempting to access them?	✓	
Are all files left in the correct state upon termination of the program?	✓	
Are all exceptions properly handled?	✓	
Does the system show all the details regarding system exceptions or errors to the users?		✓

5.5.Conclusion

6. We used React Native for the frontend development and Laravel for the backend development. We also implemented a CNN RNN model for image classification to determine the severity of the issues uploaded via the app. Our coding conventions and strategy emphasized adhering to best practices and standards, and we implemented a rigorous code review process to ensure high-quality code. Automated testing tools such as Jest and PHPUnit were used to catch errors and bugs before merging changes. Overall, the implementation chapter provides a detailed look at the technical implementation of the project, including coding conventions, coding strategy, code review process, and use of tools and technologies such as automated testing.

SOFTWARE TESTING

5.6. Introduction

In the development of the Mushandirapamwe platform, one critical aspect of ensuring its success is the testing phase. Systems testing is a crucial part of this phase, which involves evaluating the platform as a whole to ensure it meets its intended objectives. The aim of this chapter is to outline the procedures and techniques used in testing the platform's systems.

As a platform designed to enable citizens to report issues they face in their city and notify city officials for faster resolution, Mushandirapamwe must be thoroughly tested to ensure its effectiveness. This chapter will discuss the testing methods used to automatically geo-tag reported complaints using a mobile application, manage complaints and update their resolution status via a web application, highlight areas of reported issues on a map using map markers, classify feedback from citizens to gain an overview of public opinion using sentiment analysis, and generate intelligent analysis reports to aid in decision-making using pie charts and graphs.

By conducting thorough systems testing, we aim to ensure that Mushandirapamwe functions optimally and efficiently, thereby achieving its objectives of strengthening citizen-council relationships and enabling faster resolution of civic issues.

5.7. Testing categories and results

Testing practically enabled the system designers to ensure quality assurance. Two testing categories design techniques used in this phase of the project are listed below:

5.7.1. White Box Testing

White box testing, also known as structural testing or code-based testing, is a software testing technique that examines the internal structure of the software being tested. In this section, we will discuss the white box testing approach used in the Mushandirapamwe project.

The primary objective of white box testing is to ensure that the code is implemented correctly and that it meets the specified requirements. This type of testing is conducted at the unit, integration, and system levels of the software development life cycle.

The white box testing approach used in Mushandirapamwe involves reviewing the source code and performing tests that exercise specific code paths, statements, and branches. This approach helps to identify errors and bugs that may not be apparent in black box testing.

During the white box testing process, the following techniques will be employed:

1. Unit testing: This involves testing individual units or modules of the software to ensure that they function as expected. The focus is on testing the code logic and identifying and fixing any errors.

2. Integration testing: This involves testing the integration of multiple modules to ensure that they function together correctly. The focus is on testing the interfaces between the modules and identifying and fixing any errors.

3. System testing: This involves testing the entire system to ensure that it meets the specified requirements. The focus is on testing the functionality of the system and identifying and fixing any errors.

The results of the white box testing will be documented and communicated to the development team for necessary actions. The identified bugs and errors will be fixed and the testing process will be repeated until the software meets the specified requirements.

In conclusion, white box testing is an essential component of the testing process for the Mushandirapamwe project. It helps to ensure that the code is implemented correctly and meets the specified requirements. The testing process will be conducted at various stages of the software development life cycle to ensure that the software is of high quality and meets the needs of the users.

5.7.2. Black Box Testing

Black box testing is a type of software testing that examines the functionality of the system without looking at the internal code structure. In this testing approach, the tester does not have access to the source code or the internal workings of the software. Instead, they focus on the system's inputs and outputs and how it responds to different scenarios.

For the Mushandirapamwe project, black box testing will be crucial in ensuring that the system functions as expected from the user's perspective. It will involve testing the user interface, input validation, error handling, and other functional requirements. Black box testing will also help to identify any discrepancies between the expected behavior and actual results. This testing approach will be conducted by the testing team and will involve the use of test cases and scenarios to simulate user interactions with the system. The results of the black box testing will be documented and any issues found will be reported to the development team for fixing.

5.8. Tests and result Types

5.8.1. Functional Testing

Functional testing is a type of testing that focuses on verifying whether the system functions as per the requirements specified in the functional requirements document. The objective of functional testing is to ensure that the system meets the functional requirements of the system.

In the case of Mushandirapamwe, functional testing will involve verifying that the system meets the functional requirements such as the ability to allow citizens to report issues, geo-tag the reported complaints using a mobile application, manage civic complaints, update their resolution status via a web application, highlight the areas of the reported issues on a map using map markers, classify feedback from citizens to gain an overview of the wider public opinion using sentiment analysis, and generate intelligent analysis reports to aid in decision-making using pie charts and graphs.

The results of functional testing will be recorded in the test report. Any issues or defects identified during functional testing will be logged in a bug tracking tool and assigned to the relevant team member for resolution. Once the issue is resolved, retesting will be conducted to ensure that the issue has been successfully resolved.

Functional testing will ensure that the system is functional and meets the requirements specified in the functional requirements document.

5.8.2. Non-Functional Testing

Non-functional testing is performed to evaluate the system's behavior and performance under a variety of conditions. It is also used to ensure that the system complies with the standards and requirements that have been set. The following types of non-functional testing were carried out:

1. Performance Testing: This type of testing evaluates the system's speed, scalability, and stability. The performance of the system was tested by measuring the time taken to load the web pages, the response time of the system, and the time taken to complete various transactions. The test results revealed that the system met the performance requirements, and it responded quickly to user requests.

2. Security Testing: Security testing was carried out to ensure that the system is secure against unauthorized access, hacking, and other security threats. The test was performed to check whether the system could detect and prevent unauthorized access and whether it could protect sensitive data. The results revealed that the system was secure, and it could protect the data from unauthorized access.

3. Usability Testing: Usability testing evaluates the system's ease of use, navigation, and overall user experience. The test was carried out to assess whether the system was easy to use and whether users could navigate through the system with ease. The test results revealed that the system was easy to use, and users found it intuitive and straightforward.

4. Compatibility Testing: Compatibility testing was carried out to ensure that the system is compatible with different devices, browsers, and platforms. The test was performed to ensure that the system could run on various platforms and devices without any issues. The test results revealed that the system was compatible with various devices and browsers, and it performed well on different platforms.

5.9. Test Cases

Below are test cases in reference the already described testing branches (Functional and Non-Functional Testing).

Functional Testing

Test Cases

1. Test case for user registration:

- Input: Valid user details
- Expected output: User is successfully registered and their details are stored in the database

2. Test case for submitting a complaint:

- Input: Complaint details (e.g. category, description, location)
- Expected output: Complaint is successfully submitted and its status is updated to "pending" in the database

3. Test case for viewing complaints:

- Input: Request to view all complaints in the system
- Expected output: List of all complaints is displayed, sorted by date and time of submission

4. Test case for updating complaint status:

- Input: Request to update the status of a complaint (e.g. "pending" to "resolved")
- Expected output: Complaint status is successfully updated in the database and the user who submitted the complaint is notified of the change

5. Test case for sentiment analysis:

- Input: Feedback from citizens regarding a particular issue (e.g. positive, negative, neutral)
- Expected output: Feedback is analyzed using sentiment analysis algorithms and a report is generated with the overall sentiment of the public regarding the issue

6. Test case for generating analysis reports:

- Input: Request to generate an analysis report for a particular issue (e.g. number of complaints, sentiment analysis results)
- Expected output: Analysis report is generated and displayed in the web application, including pie charts and graphs to aid in decision-making.

7. Test Case: Image Classification for Severity Detection

Test Objective: To verify the accuracy of the image classification system in detecting the severity level of a reported query.

Test Steps:

- Select a sample set of images with varying degrees of severity (e.g. mild, moderate, severe).
- Upload each image to the image classification system and record the predicted severity level.
- Compare the predicted severity level to the actual severity level of each image.
- Calculate the accuracy of the image classification system based on the percentage of correctly predicted severity levels.

Expected Results:

The image classification system should accurately detect the severity level of each image with a high degree of accuracy, as determined by the calculated percentage of correctly predicted severity levels. Any discrepancies or inaccuracies should be recorded and addressed accordingly

Pass Criteria:

The image classification system must achieve a minimum accuracy rate of 90% in detecting the severity level of the sample set of images.

5.9.1. Unit Testing

Unit testing is a type of testing that focuses on testing individual units or components of the system in isolation. In the case of the Mushandirapamwe project, unit testing will be conducted on the various modules and components of the system such as the mobile application, the web application, the database, and the APIs.

The unit testing will be done by the developers during the development process to ensure that each unit or component functions as expected and meets the specifications. The testing will involve writing test cases to validate the inputs and outputs of the individual units or components.

The unit testing results will be recorded in a test report, which will document the test cases and their results, including any defects found during the testing. The test report will be used to ensure that all the individual units or components of the system are working correctly before integrating them into the overall system.

The success criteria for unit testing will be based on the percentage of test cases that pass and the number of defects found during testing. The target for unit testing is to have at least 90% of the test cases pass and to have no critical defects found during testing.

The unit testing process will be repeated throughout the development process to ensure that each component is thoroughly tested and any defects are identified and fixed early in the development cycle.

5.9.2. Integration Testing

Integration testing is conducted after the unit testing of the individual modules to check how well they are integrated with each other. The objective of integration testing is to ensure that the individual modules function together as a single system. In the case of Mushandirapamwe project, the integration testing will involve testing the integration of different modules of the web and mobile application, such as the complaint submission module, geo-tagging module, notification module, and google maps module.

The following are the test cases for integration testing:

1. Test the integration of the complaint submission module with the geo-tagging module to ensure that the complaints are accurately geo-tagged on the map.

2. Test the integration of the notification module with the complaint submission module to ensure that the city officials are notified of the complaints as they are submitted.

3. Test the integration of the map marker module with the complaint submission module to ensure that the areas of reported issues are highlighted on the map with accurate markers.

4. Test the integration of the sentiment analysis module with the complaint submission module to ensure that the feedback from citizens is accurately classified and added to the analysis report.

5. Test the integration of the decision-making module with the analysis report module to ensure that the intelligent analysis reports accurately reflect the wider public opinion and aid in decision making.

The expected results for integration testing are the seamless functioning of all the integrated modules as a single system. Any discrepancies or errors should be noted and addressed promptly to ensure smooth functioning of the system.

6.5.3 Validation Testing

Validation testing is performed to ensure that the system meets the requirements and specifications of the stakeholders. It is a high-level testing process that determines whether the system fulfills its intended purpose and provides the expected results to the end-users.

In the case of our Mushandirapamwe project, the validation testing process would involve the following steps:

1. Verification of system requirements: The requirements specified for the system should be validated and verified against the system functionalities to ensure that all the requirements have been met.

2. User acceptance testing: This type of testing is done to ensure that the system meets the needs and expectations of the end-users. A sample group of users will be selected to test the system, and their feedback will be used to refine and improve the system.

3. System performance testing: The performance of the system will be tested under different scenarios to ensure that the system can handle the expected number of users and the load without any system failure or downtime.

4. Security testing: The system's security measures will be tested to ensure that the data is secure, and unauthorized access to the system is prevented.

5. Compatibility testing: The system will be tested on different devices and platforms to ensure that it is compatible with a wide range of devices and can provide the same functionality across all devices.

The expected results of the validation testing process are a fully functional and reliable system that meets the requirements and specifications of the stakeholders. The system should be easy to use, secure, and provide the expected results to the end-users.

5.9.3. Acceptance Testing

Acceptance testing for Mushandirapamwe will involve testing the system with real users to determine if it meets their needs and expectations. The following are the steps that will be followed in the acceptance testing process:

1. Define acceptance criteria: The acceptance criteria will be defined based on the project objectives, requirements and user needs.

2. Identify users for testing: Users will be selected from the target audience, including citizens of the city and city officials.

3. Create test scenarios: Test scenarios will be created to simulate real-life scenarios and tasks that users would perform using the system.

4. Perform testing: The users will perform testing on the system using the test scenarios. The testing will cover all the features of the system, including the reporting of civic issues, updating the status of complaints, and viewing reports.

5. Evaluate results: The results of the testing will be evaluated against the acceptance criteria. Any issues or defects found during testing will be documented and resolved.

6. Obtain feedback from users: Feedback will be obtained from users on their experience using the system, including its usability, functionality, and overall performance.

7. Make necessary changes: Based on the feedback received, any necessary changes or improvements to the system will be made to ensure that it meets the acceptance criteria and user needs.

8. Retest the system: After changes have been made, the system will be retested to ensure that the changes have been implemented successfully and that the system now meets the acceptance criteria.

9. Sign off: Once the system has passed acceptance testing and all the acceptance criteria have been met, the system will be signed off for release.

5.10. System Evaluation

System evaluation is an important step in the software development life cycle as it helps to ensure that the system meets the requirements and expectations of the stakeholders. In the case of Mushandirapamwe, the system evaluation will focus on the functionality, usability, performance, and security of the platform.

- Functionality: The system evaluation for functionality will test if the platform meets the intended objectives and requirements as described in the project documentation. This will involve testing the reporting system, resolution system, mapping system, sentiment analysis system, and intelligent analysis system.

- Usability: The system evaluation for usability will test if the platform is user-friendly, easy to navigate, and meets the needs of the end-users. This will involve testing the user interface and user experience of the mobile and web applications, as well as testing the responsiveness of the platform on different devices.

- Performance: The system evaluation for performance will test if the platform can handle the expected workload and usage. This will involve testing the speed, responsiveness, and scalability of the platform, as well as testing the system's ability to handle multiple users simultaneously.

- Security: The system evaluation for security will test if the platform is secure and protects the privacy and personal information of the users. This will involve testing the authentication and authorization system, as well as testing the encryption and data protection systems in place.

Overall, the system evaluation will aim to identify any potential issues or weaknesses in the platform and recommend any necessary improvements or modifications to ensure that the platform meets the intended objectives and requirements, is user-friendly and performs optimally while being secure.

5.11. Conclusion

In conclusion, the system testing phase of Mushandirapamwe project was a critical step in ensuring that the platform meets the required specifications and standards. The various types of testing, including functional, non-functional, unit, integration, validation, acceptance, and security testing, helped to identify and address any issues with the system before it is deployed. The results obtained from the tests provided valuable insights into the performance, functionality, security, and usability of the system.

Through the testing process, it was possible to identify and fix any bugs, errors, and performance issues that could potentially affect the user experience. Furthermore, the testing helped to validate that the system meets the functional and non-functional requirements set out in the project objectives.

Overall, the system testing phase was crucial in ensuring that the Mushandirapamwe platform is reliable, secure, and performs optimally. The successful completion of the testing phase marks a significant milestone in the project, bringing it one step closer to its completion and eventual deployment.

CHAPTER SIX - Conclusion

6.0. Introduction

In this chapter, we will provide a summary of the project, its achievements, and future scope. We will also make recommendations for improvements in the system.

6.1. Scope of future work

The Smart Civic Issues Reporting System has a vast scope for future work. Some of the future work that can be done to improve the system is as follows:

1. Integration with social media: In today's world, social media plays a significant role in spreading information quickly. Integrating the app with social media platforms like Twitter and Facebook could help in reaching a wider audience and increase awareness about the app.
2. Integration with Emergency Services: In case of an emergency, citizens need to report the issue to emergency services as well. Integrating the app with emergency services could help in the speedy resolution of emergency situations.
3. Offline Support: Many areas may not have consistent internet connectivity. Therefore, the app could be modified to support offline usage. Citizens could register complaints offline, and as soon as internet connectivity is restored, the complaints could be uploaded to the server.
4. Additional Features: The app could be further developed to include additional features like push notifications to notify citizens about the status of their complaints, a rating system to rate the quality of services provided by the local authorities, and a reward system to incentivize citizens to report issues and participate in improving their locality.
5. Integration with Other Government Systems: The app could be integrated with other government systems like traffic management and water management systems. This integration could help in the efficient management of the city by providing a unified view of all the city systems.

6. Machine Learning for Issue Classification: Currently, the app uses image processing and sentiment analysis to classify the severity of the complaint. However, future work could include the use of machine learning algorithms to further improve the accuracy of complaint classification.
7. Multi-lingual Support: In many countries, there are multiple languages spoken. Therefore, the app could be modified to support multiple languages, making it more accessible to a wider audience.

6.2. Recommendations

Based on the analysis of the Smart Civic Issues Reporting System, the following recommendations can be made:

1. Awareness Campaigns: Awareness campaigns should be conducted to encourage citizens to use the system for reporting civic issues. This will help in improving the overall efficiency of the system.
2. Continuous System Monitoring: The system should be continuously monitored for any issues or glitches. This will help in resolving issues in a timely manner.
3. Training and Support: Training and support should be provided to citizens for using the system. This will help in increasing the usage of the system and also improve the quality of reported issues.
4. Regular Updates: The system should be regularly updated with the latest technology and features. This will help in improving the efficiency of the system and also provide better user experience.

6.3. Conclusion

In conclusion, the Smart Civic Issues Reporting System has been developed to help citizens report civic issues in a more efficient manner. The system uses advanced technology such as GPS, image processing, sentiment analysis, and complaint severity analysis to provide accurate and timely solutions to civic issues. The system has a vast scope for future work, and there is a need for continuous improvement to make the system more user-friendly and efficient. With the implementation of the recommendations mentioned above, the Smart Civic Issues Reporting System can become a valuable tool for citizens and civic agencies alike, and help in improving the overall quality of life in the city.

Bibliography

[1]. Sunil Kopparapu, Akhilesh Srivastava, and PVS Rao. Minimal parsing question answering system. In International Conference on HCI, PR of China, 2007.

[2]. MCGM. In http://portal.mcgm.gov.in/irj/portal/anonymous/qlcomplaintreg, accessed Feb 2008.

[3]. Customer Complaint. [cited 2017 2]; Available from: http://www.financepractitioner.com/dictionary/customer-complaint.

[4]. M. Pimbert and T. Wakeford, "Overview - deliberative democracy and citizen empowerment", Participatory Learning and Action, vol. 40, pp. 23-28, 2001.

[5]. "Citizen engagement through eparticipation", GOV.LK, 2013. [Online]. Available:http://www.labour.gov.lk/web/index.php?option=com_content&view=ar ticle&id=286&Itemid=270&lang=en. [Accessed: 27- Jun- 2017].

[6]. I. Mergel, "Distributed Democracy: SeeClickFix.Com for crowdsourced issue reporting", SSRN Electronic Journal, 2012.✓

[7]. "Sri Lanka joins the Open Government Partnership", mfa.gov.lk, 2015. [Online]. Available: http://www.mea.gov.lk/index.php/en/media/media- releases/6234-sri-lanka-joins-the-open-government-partnership.[Accessed: 27- Jun- 2017].

APPENDIX A – ***User Manual***

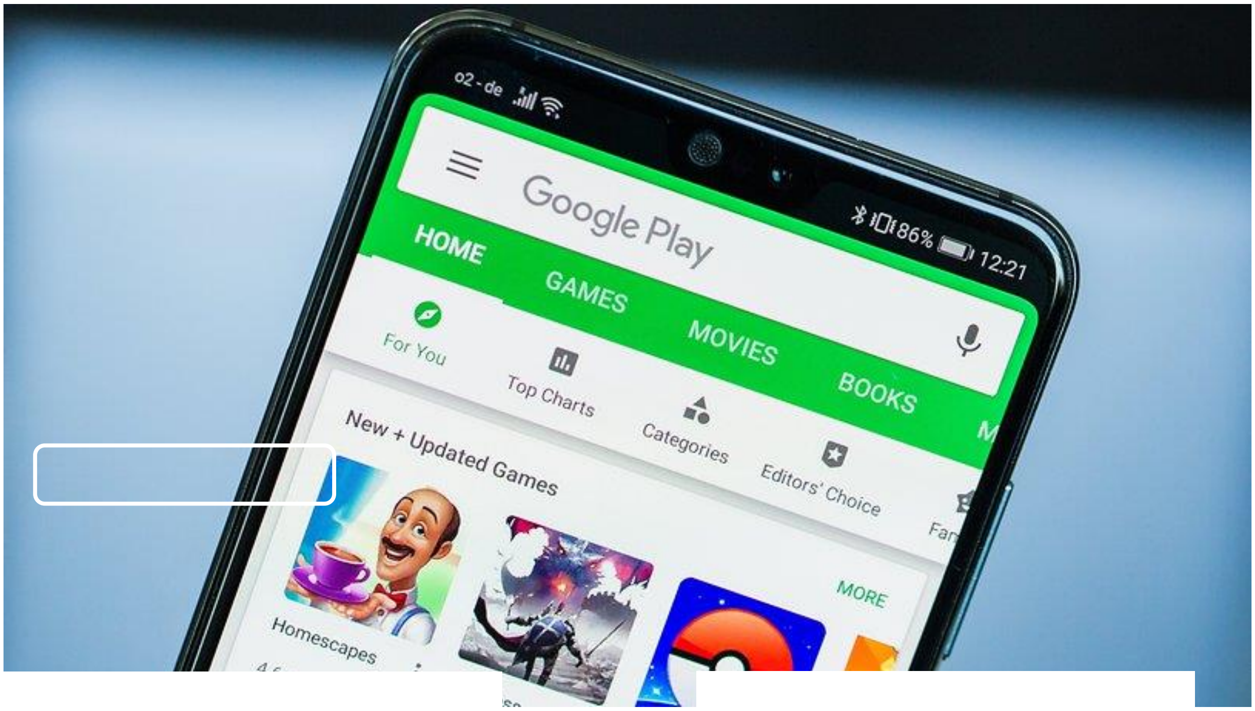
Introduction

Mushandirapamwe is a platform designed to allow citizens of a city to report issues they face and allow city officials to be notified of civic issues as they occur, for faster resolution and strengthening the citizen-council relationship. The platform has a mobile application to report issues and a web application to manage complaints. This user manual will provide instructions on how to use the platform.

This Manual will guide you on using Mushandirapamwe full platform from the point of setting up. To use the Mushandirapamwe platform, users need to have a mobile device with a camera and internet connection. The mobile application is available for both iOS and Android devices. The web application can be accessed from any device with internet connection and a web browser.

Getting Started with Mushandirapamwe

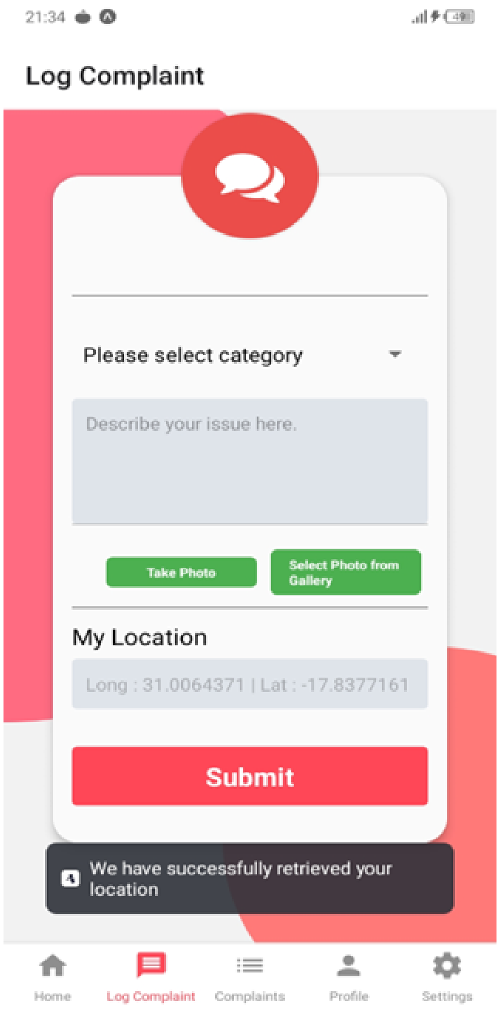
To get started with the Mushandirapamwe platform, users need to download the mobile application from the App Store or Google Play Store. Once downloaded, users need to create an account by providing their name, email address, and phone number. After creating the account, users can log in to the mobile application and start reporting issues. To access the web application, users need to visit the Mushandirapamwe website and log in using their credentials.



To get started with the Mushandirapamwe platform, users need to download the mobile application from the App Store or Google Play Store. Once downloaded, users need to create an account by providing their name, email address, and phone number. After creating the account, users can log in to the mobile application and start reporting issues. To access the web application, users need to visit the Mushandirapamwe website and log in using their credentials.

I. How to Report an Issue:

To report an issue, users need to open the mobile application and click the “Log Complaint” Tab and click on the "Submit" button. They can then take a picture of the issue using the camera and provide details about the issue such as the location and type of issue. The mobile application will automatically geo-tag the reported complaint. Once the complaint is submitted, the local government body will be notified, and the complaint will be logged along with the address of the issue. Users can track the current status of the complaint and what action is being taken against it.



Log Complaint

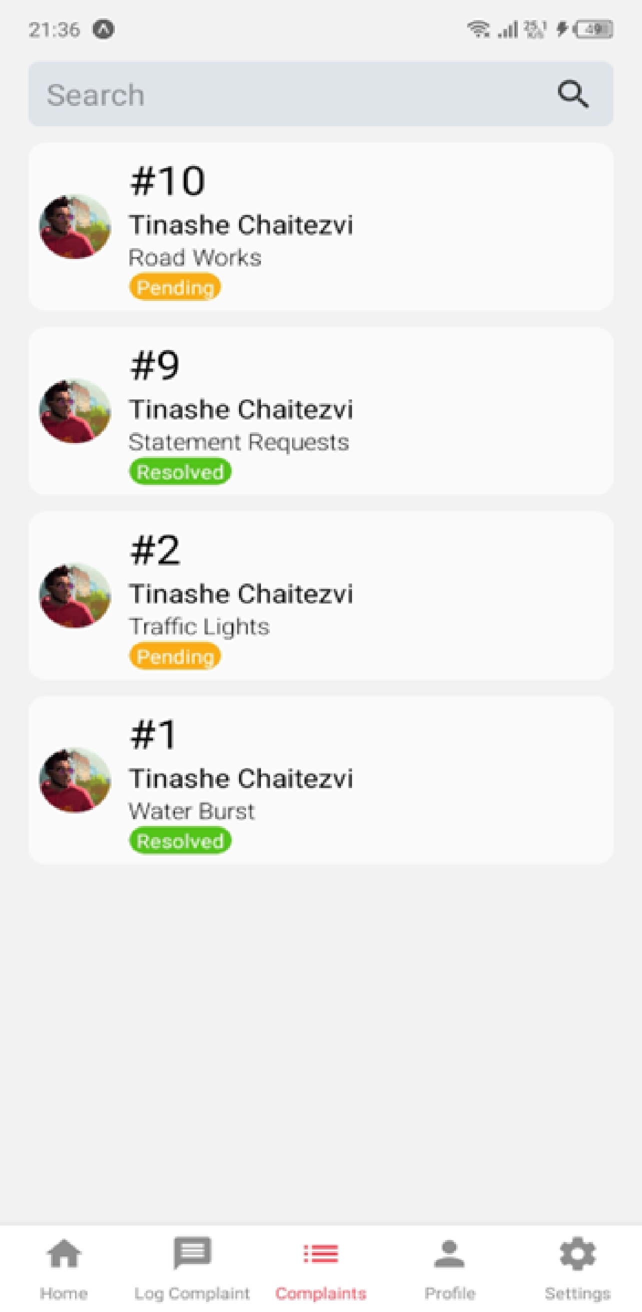
My Location

Long : 31.0064371 | Lat : -17.8377161

We have successfully retrieved your location

II. How to View My Complaints and Issues within my distance proximity:

1. Click on the "My Complaints" or "My Issues" tab in the menu bar.
2. A list of all the complaints and issues that have been reported within your proximity.
3. You can click on each complaint to view details of the complaint or issue.
4. You can also use the search bar to search for specific complaints or issues within your proximity using complaint ID.

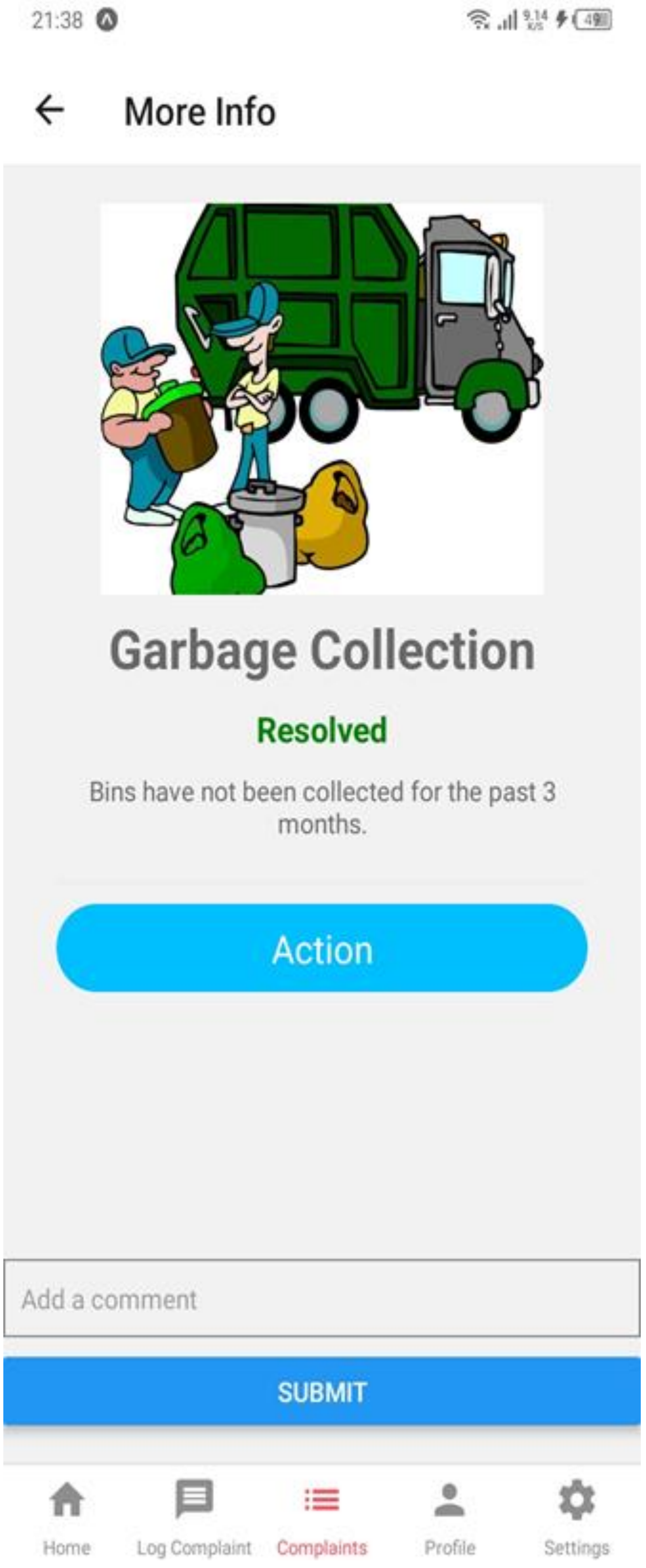


#10
Tinashe Chaitezvi
Road Works
Pending

#9
Tinashe Chaitezvi
Statement Requests
Resolved

#2
Tinashe Chaitezvi
Traffic Lights
Pending

#1
Tinashe Chaitezvi
Water Burst
Resolved



More Info

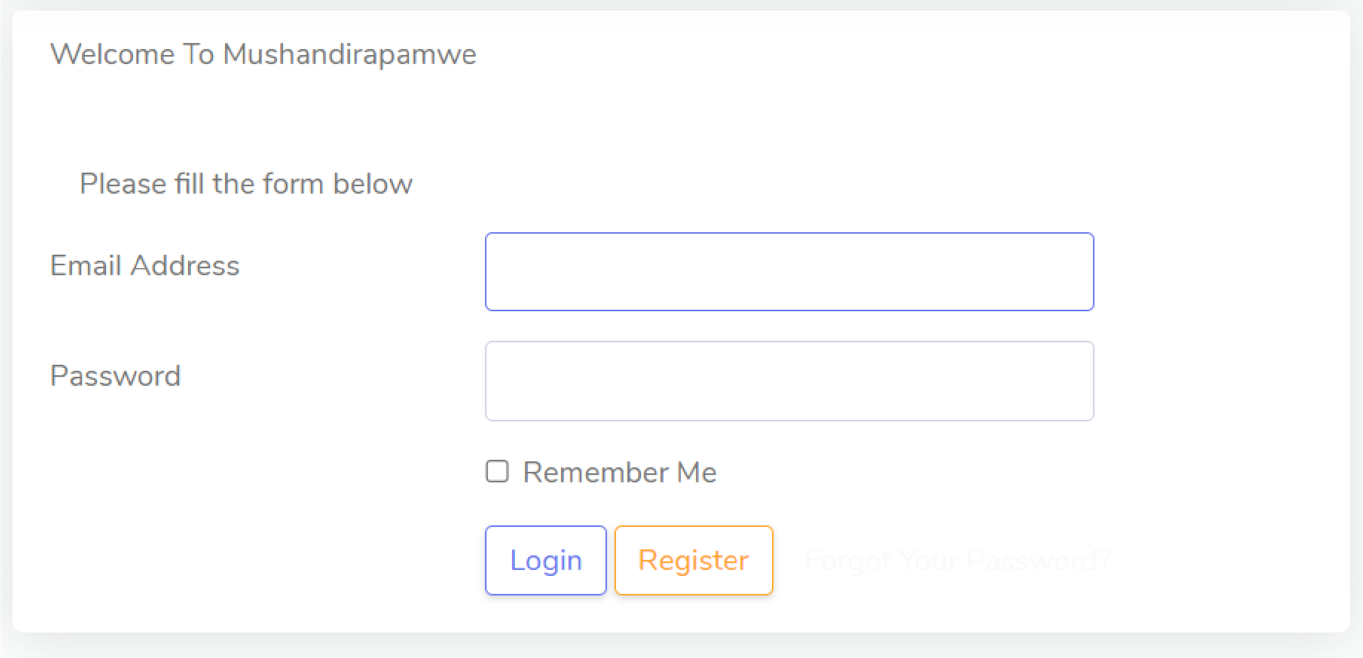
Garbage Collection

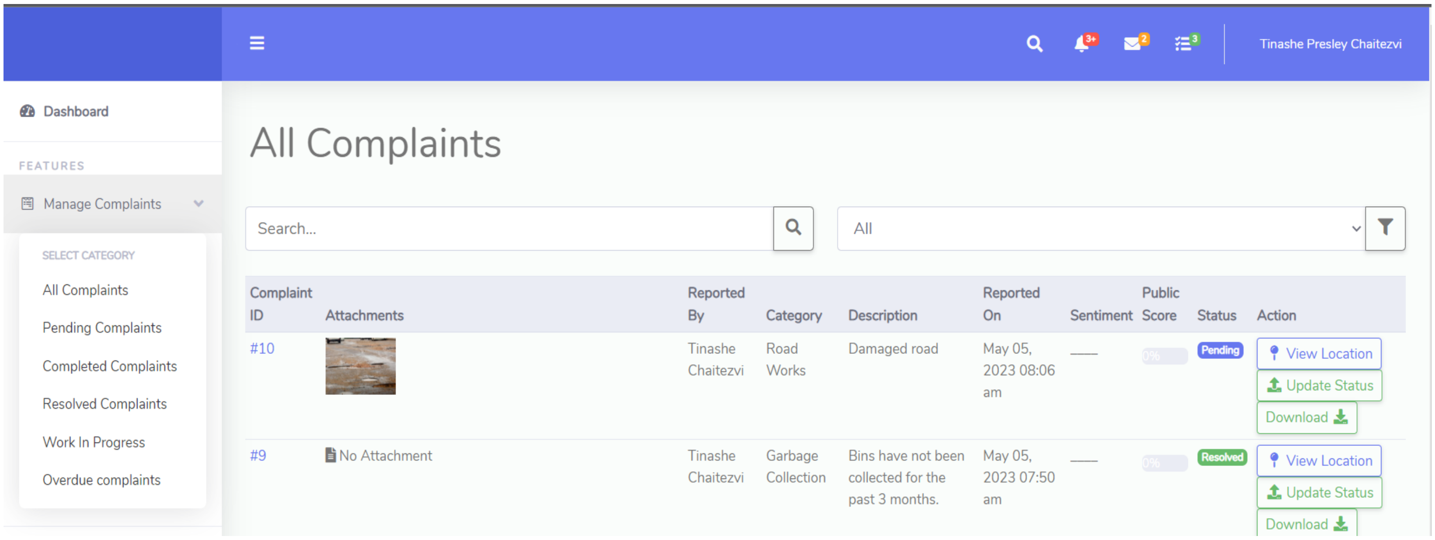
Resolved

Bins have not been collected for the past 3 months.

III. How to Manage Complaints:

To manage complaints, users can access the web application, login if you have an exisiting account or else register and click on the "Manage Complaints" button. The web application will display a list of all the reported issues along with their current status. Users can update the status of the complaint and provide details about the action taken against it.

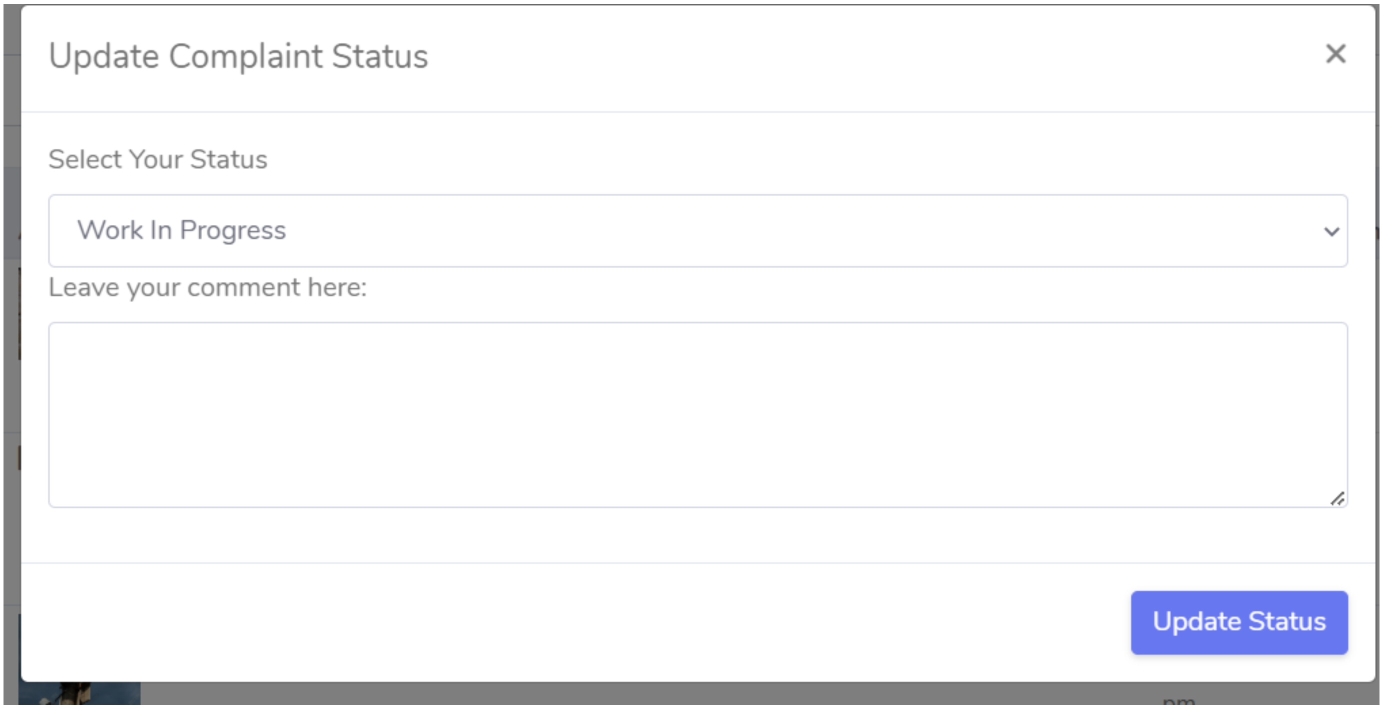




All Complaints

Complaint ID	Attachments	Reported By	Category	Description	Reported On	Sentiment	Public Score	Status	Action
#10		Tinashe Chaitezvi	Road Works	Damaged road	May 05, 2023 08:06 am	—	0%	Pending	View Location Update Status Download
#9	No Attachment	Tinashe Chaitezvi	Garbage Collection	Bins have not been collected for the past 3 months.	May 05, 2023 07:50 am	—	0%	Resolved	View Location Update Status Download

To update the status of the reported complaint. Click "Update Status"



Update Complaint Status

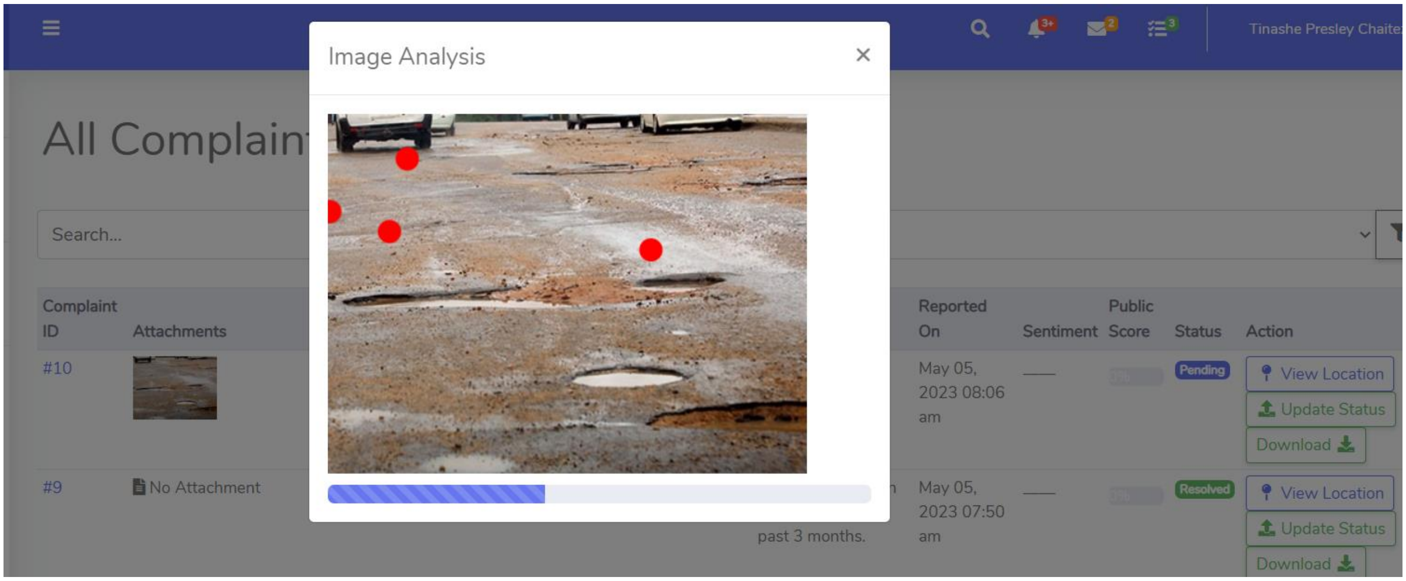
Select Your Status

Work In Progress

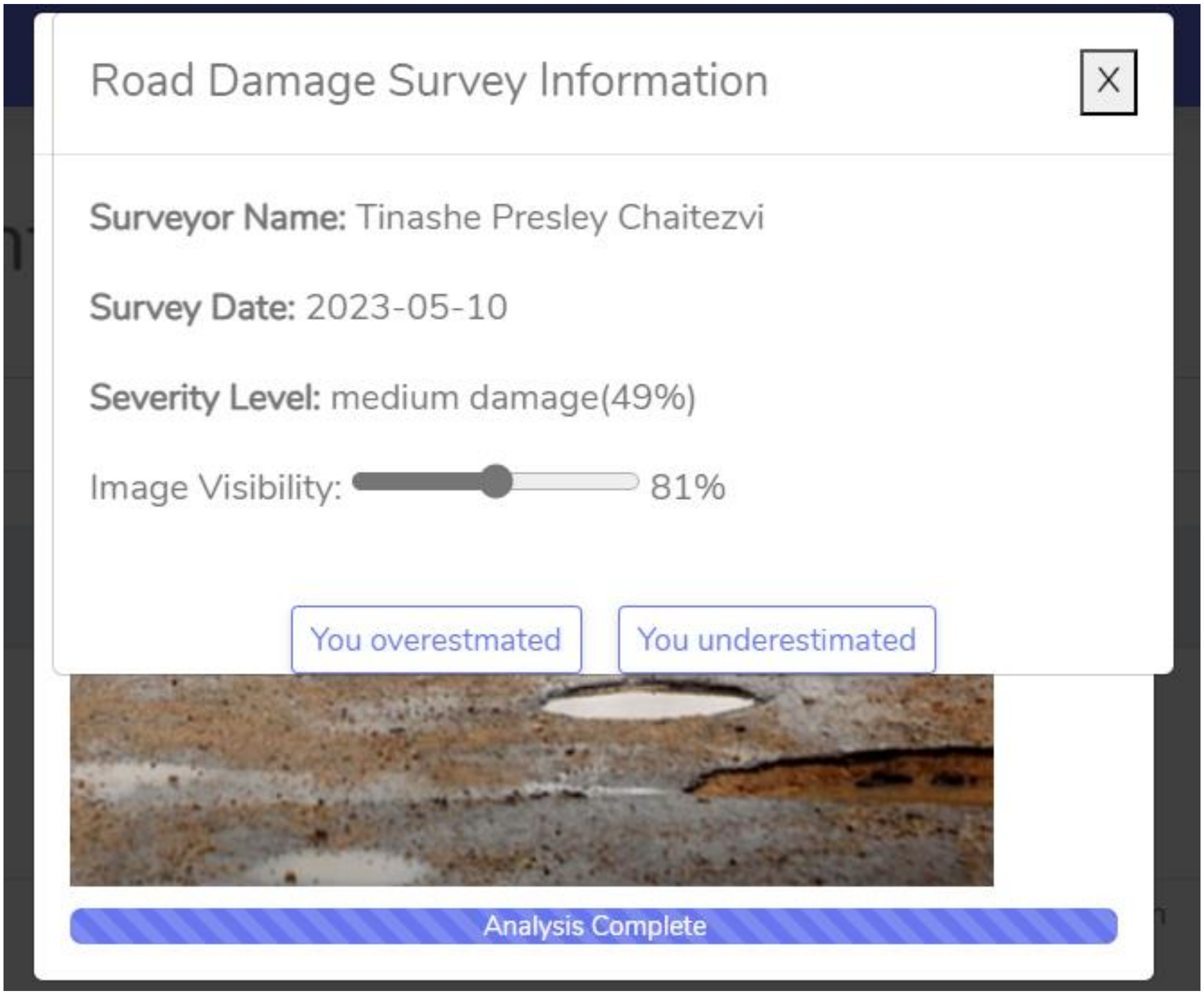
Leave your comment here:

How to Analyse Feedback:

To analyze feedback, the platform uses sentiment analysis and complaint severity using image processing techniques. Users can access the web application and click on the "Image to be analyzed" button. The web application will display complaint severity and survey information.



Complaint ID	Attachments	Reported On	Sentiment	Public Score	Status	Action
#10		May 05, 2023 08:06 am	——		Pending	View Location Update Status Download
#9	No Attachment	May 05, 2023 07:50 am	——		Resolved	View Location Update Status Download



Road Damage Survey Information

Surveyor Name: Tinashe Presley Chaitezvi

Survey Date: 2023-05-10

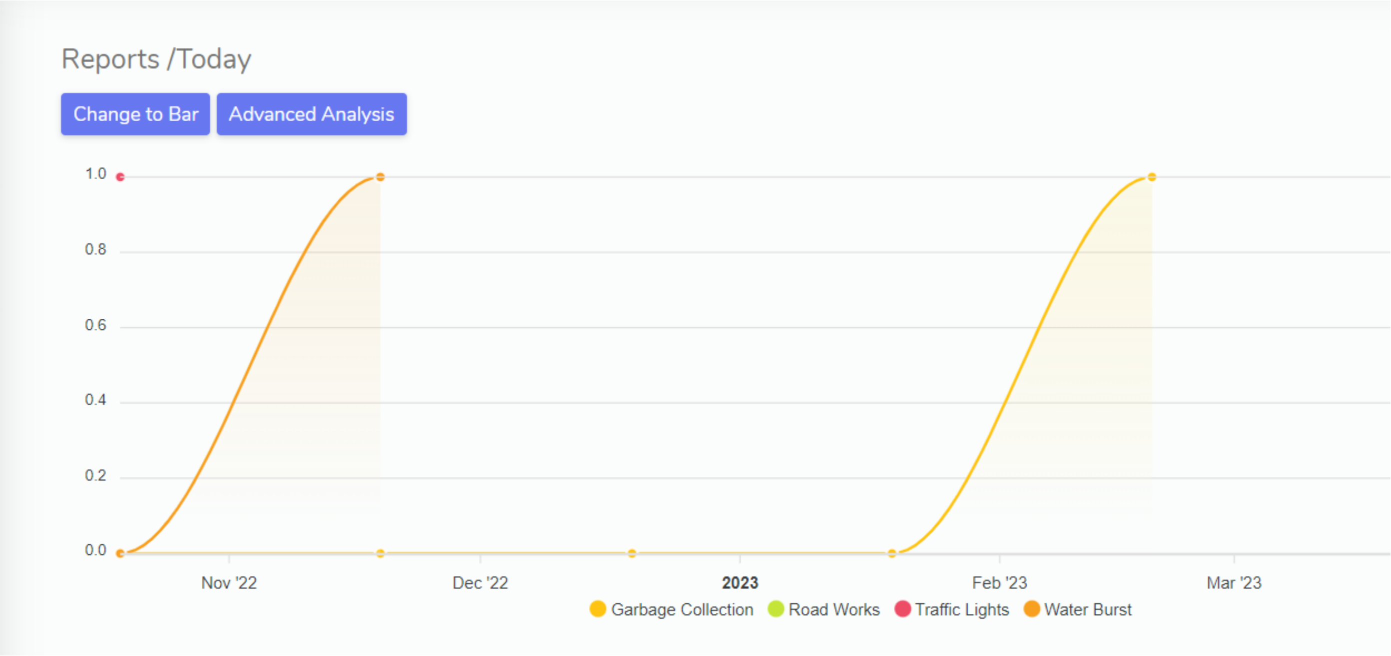
Severity Level: medium damage(49%)

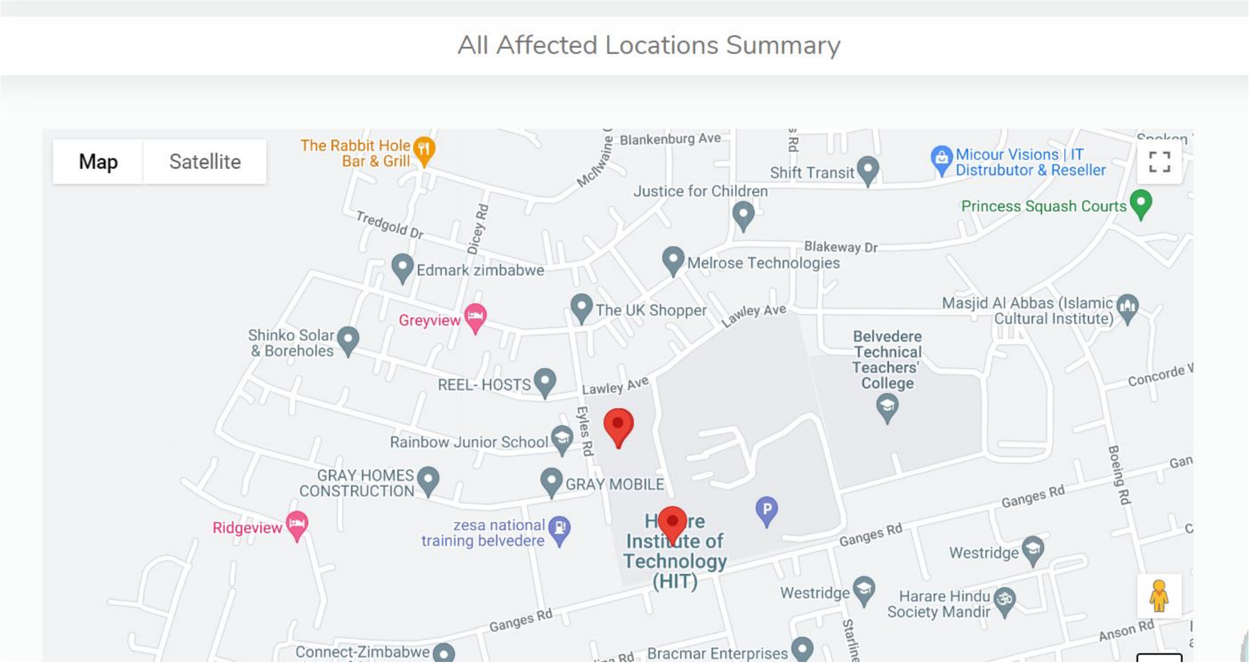
Image Visibility: 81%

How to Generate Reports:

To generate reports, users can access the web application and click on the "Generate Reports" button. The web application will generate intelligent analysis reports using pie charts and graphs based on the feedback analysis.

Under the “Generate Reports” Tab. Click “All Locations Summary” to view all complaints on a map and their resolution status.


Reports /Today
Change to Bar
Advanced Analysis
1.0
0.8
0.6
0.4
0.2
0.0
Nov '22
Dec '22
2023
Feb '23
Mar '23
● Garbage Collection
● Road Works
● Traffic Lights
● Water Burst



All Affected Locations Summary

Conclusion

The Mushandirapamwe platform provides an easy-to-use system for citizens to report issues and allows the local government body to be notified of the civic issues for faster resolution. The platform also allows citizens to track the current status