



GUJARAT TECHNOLOGICAL UNIVERSITY

Chandkheda, Ahmedabad
Affiliated

G H Patel College of Engineering and Technology

Department of Computer Engineering (07)

A.Y. 2016-17(EVEN), SEMESTER 6

SUBJECT CODE: 2160701

SUBJECT NAME: SOFTWARE ENGINEERING (SE)

Software Requirements Specification

Active City Administrations

-: **Made by:** -

Jemish Paghadar (140110107030)
Deep Patel (140110107037)

-: **Guided by:** -

Prof. Khyati Mehta

-: INDEX: -

1.0 Introduction

- 1.1. Purpose
- 1.2. Scope of Project
- 1.3. Glossary
- 1.4. References
- 1.5. Overview of Document

2.0 Overall Description

- 2.1 System Environment
- 2.2 Functional Requirements Specification
 - 2.2.1 Citizens Use Case
 - 2.2.3 NGO's Use Case
 - 2.2.3 Municipal authorities Use Case
 - 2.2.4 Administrators Use Case
- 2.3 User Characteristics
- 2.4 Non-functional requirements

3.0 Processing Model: The Waterfall Model

- 3.1 Description:
- 3.2 Phases of waterfall model
 - 3.2.1 Communication
 - 3.2.2 Planning
 - 3.2.3 Modeling
 - 3.2.4 Construction
 - 3.2.5 Deployment

4.0 Various Diagrams

- 4.1 Data Flow Diagram(DFD)
 - 4.1.1 Level 0 DFD
 - 4.1.2 Level 1 DFD
 - 4.1.3 Level 2 DFD
- 4.2 Entity-Relationship Diagram (E-R Diagram)
- 4.3 Class Diagram

5.0 Data Dictionary

6.0 Unit Testing

- 6.1 Black-Box Testing
- 6.2 White-Box Testing

2.0. Introduction

2.1. Purpose

The purpose of this document is to present a detailed description of the Active City Administration. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

2.2. Scope of Project

This software system will be a City Administration for a local citizen of the city. This system will be designed to maximize the interaction between local citizens and municipality. By maximizing transparency between both sides, the system will meet public's needs while remaining easy to understand and use. More specifically, this system is designed to Time bound delivery of essential services to the citizens of the district. The software will facilitate communication between general public and government via E-Mail. Preformatted reply forms are used in complaint page through the system to provide a uniform review process. The system also contains a relational database containing a list of various users.

2.3. Glossary

<u>Term</u>	<u>Definition</u>
Dashboard	Dashboards often provide at-a-glance views of user activity relevant to a particular complaint or log-activity.
User	A. Citizens of city. B. Local municipal authorities. C. Administrators. D. NGOs.
Activity	Complaint, search nearest to me, view latest govt schemes,etc..
Reviewer	Person who receives Complaints, check Complaints (review), and makes final judgments for Complaints.
Field	A cell within a form.
User's Complaints Database	The existing member's Complaints database.
Reader	Anyone visiting the site to read articles.
Review	A written recommendation about the appropriateness of a Complaints to Municipality; may include suggestions for improvement.
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.

2.4. References

- IEEE. IEEE STD 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
- <https://anand.gujarat.gov.in/municipality>
- <http://www.mcgm.gov.in/>
- <https://www.kmcgov.in>

1.5. Overview of Document

In the next, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification. Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product. The document describe the same software product in its entirety.

The information captured using this document is intended to provide us with information about your specification and the requirements for your Website. To enable us to gain as thorough an understanding of what you require this document contains questions about your overall development along with detailed questions regarding what you want from your website and the information (content) that is to be displayed on it.

3.0. Overall Description

2.1 System Environment

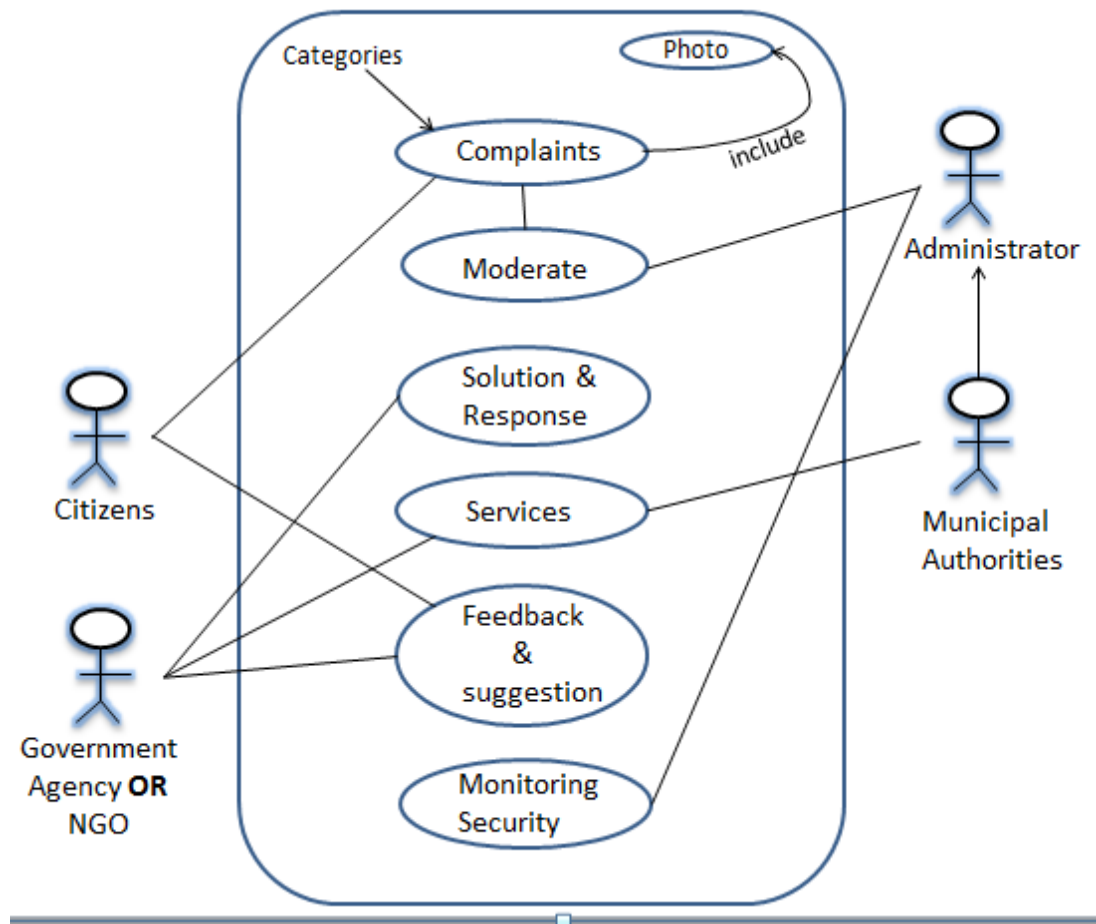


Figure 1 - System Environment

The Active City Administration has four active actors and one cooperating system.

The citizens and NGOs access the website and do such activity through the Internet. Any user communication with the system is through email. The Administrators and Local municipal authorities can access the entire system directly.

The division of Active City Administration mainly into six component parts - :

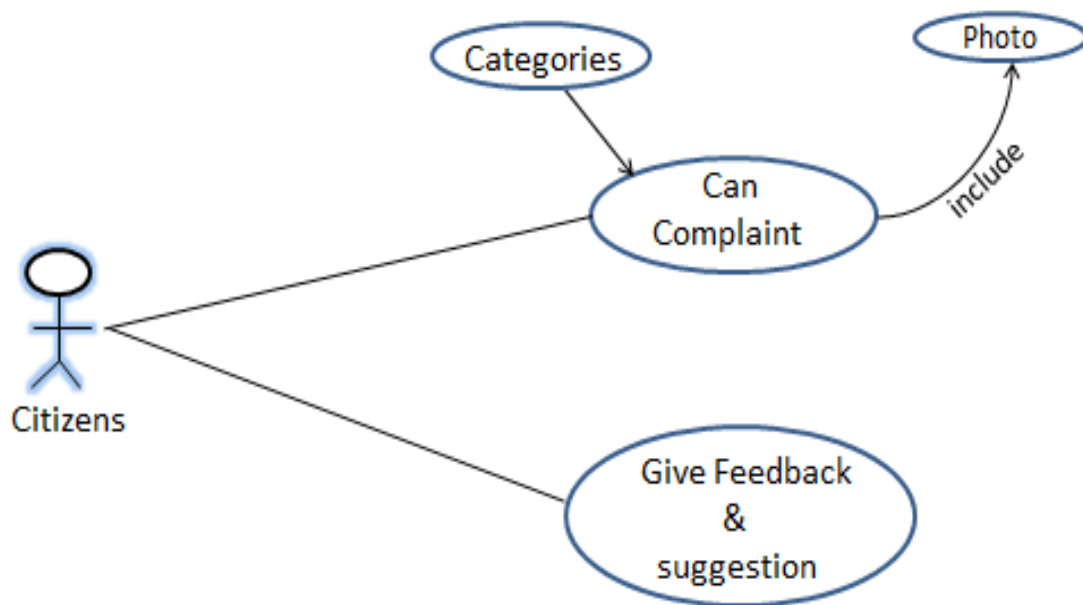
- 1) Complaints Box
- 2) Moderator portal
- 3) Feedback & suggestion
- 4) Monitoring Security
- 5) Solution & Response
- 6) Services.

2.2 Functional Requirements Specification

This section outlines the use cases for each of the active Users separately.

2.2.1 Citizens Use Case

Diagram:



Description:

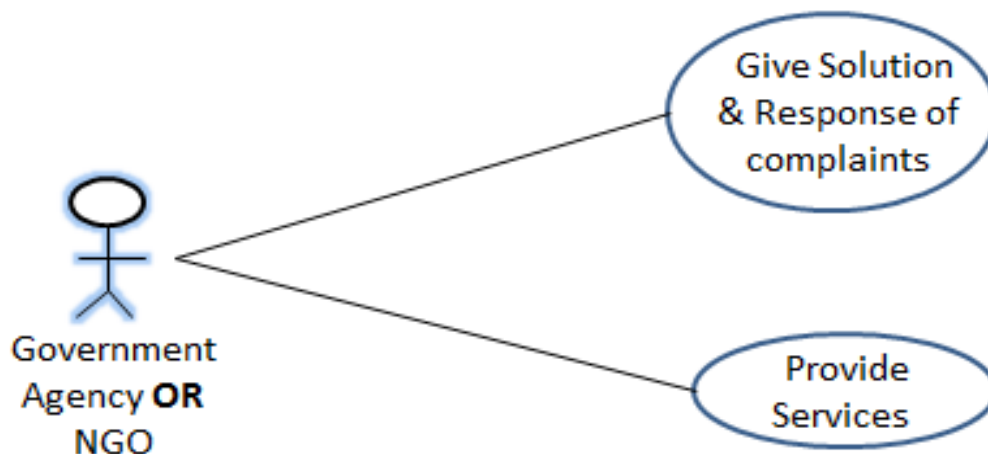
The Citizens accesses their account and see the day-by-day municipality's activity. One can submit valid complaint categories wise and give feedback and suggestion.

Functional Requirements:

1. Citizens should be able to create new account, login to their existing accounts.
2. Authenticated Citizens should be able to issue complaints, check complaint status, submit feedback, and browse through other complaints and their feedback.
3. Authenticated Citizens should be able to create suggestions/petitions.
4. Citizens can see status of his/her complaint.
5. Citizens can also see general complaint, which they are facing day by day life. e.g. road-service

2.2.2 NGO's Use Case

Diagram:



Description:

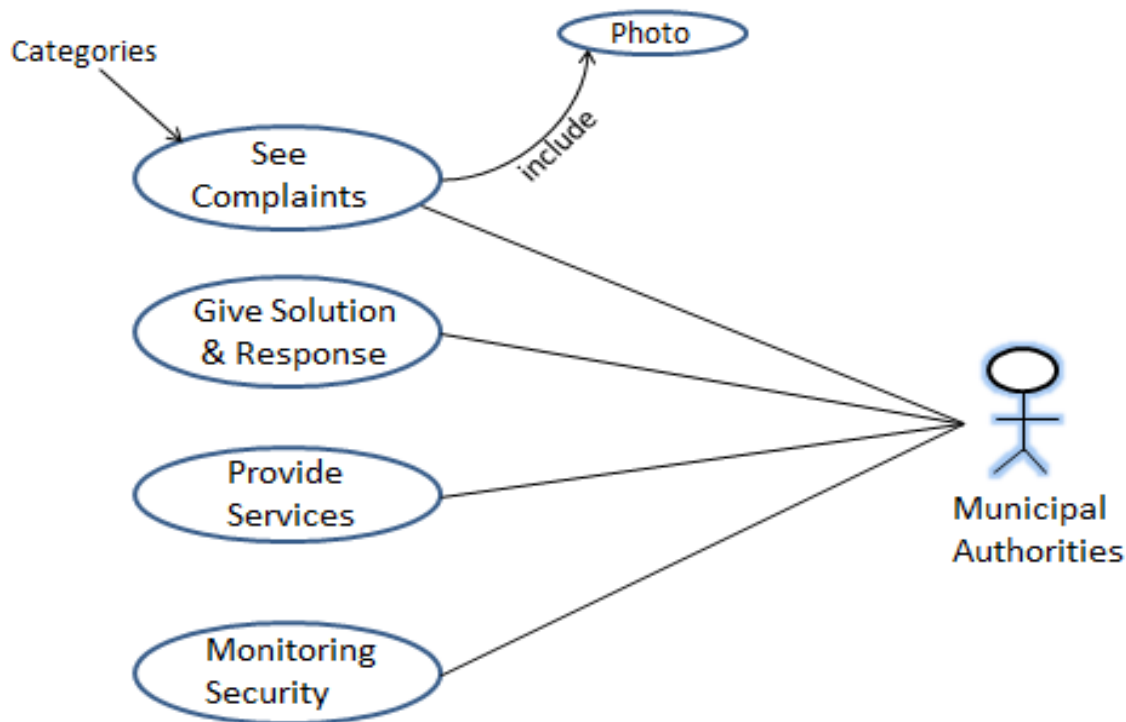
A **non-governmental organization (NGO)** is a not-for-profit organization that is independent from states and international governmental organizations. They are usually funded by donations but some avoid formal funding altogether and are run primarily by volunteers. NGO's can publicize their social causes on the site.

Functional Requirements:

1. NGO should be able to create new account, login to their existing accounts.
2. Authenticated NGO should be able to issue complaints which are not solved by municipalities, check complaint status, submit feedback, browse through other complaints and their feedback.
3. NGO can see status of complaint.
4. NGOs can facilitate communication upward from people to the government and downward from the government to the people.
5. Communication upward involves informing government about what local people are thinking, doing and feeling while communication downward involves informing local people about what the government is planning and doing.

2.2.3 Municipal authorities Use Case

Diagram:



Description:

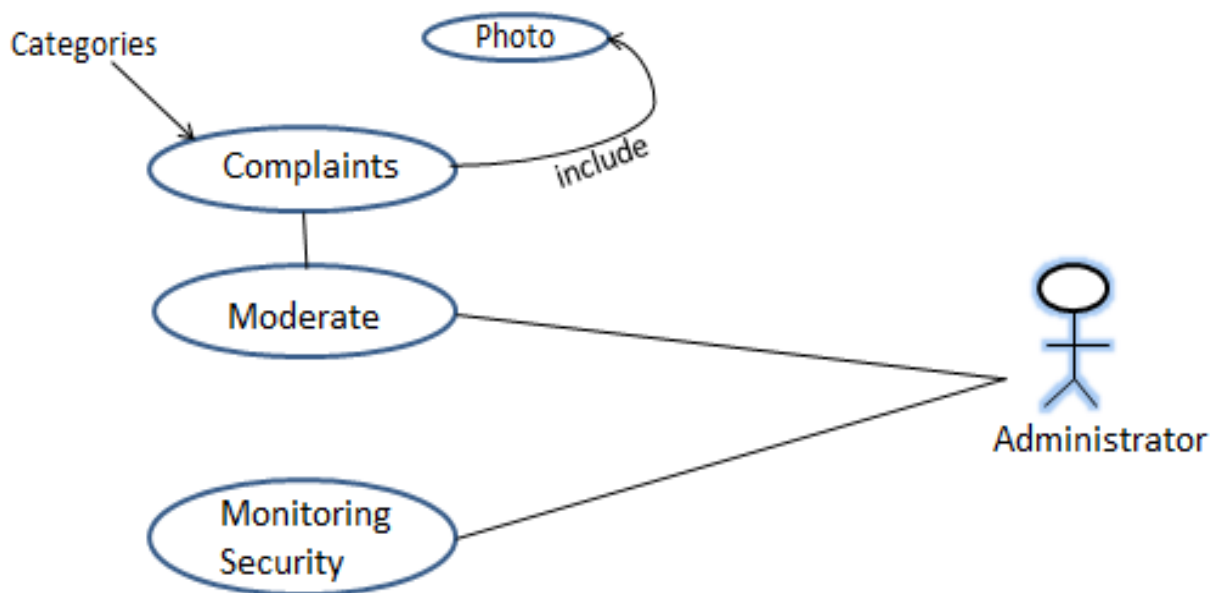
A Municipal Corporation, City Corporation, Mahanagar Palika or **Mahanagar Nigam** is a local government in India that administers urban areas with a population of more than one million. The growing population and urbanization in various cities of India were in need of a local governing body that can work for providing necessary community services like health care, educational institution, housing, transport etc. by collecting property tax and fixed grant from the State Government.

Functional Requirements:

1. Municipal authorities can login to their accounts as created by administrator.
2. Authorities can access all the complaints, suggestions from users.
3. Invoke proper activity in response to valid complaints, or redirect inappropriate complaints to the administrator.
4. Give response to complaints

2.2.4 Administrators Use Case

Diagram:



Description:

Administrators are who filters the complaints. They check every complaints, see that if any miscellaneous language are used or not? If used then it sends back to the complainer and ask to correct language. In addition, if not then it sends to the according department of the municipality. Administrators are plays important role in city management. In sort, they manages all complaints, which are made by Citizens.

Functional Requirements:

1. Create, and monitor accounts of authorities.
2. Filter the content reported as inappropriate and handle threats.
3. Handle complaints about improper response (threats) by municipal authorities.

2.3 User Characteristics

The Citizens are expected to be Internet literate and be able to use services. The main screen of the Website will have the dashboard of complaints, complaint panel, user acc. Info. Etc.

The Citizens and Administrators are expected to be Internet literate and to be able to use email with attachments.

Administrators are who filters the complaints. They check every complaint, see that if any miscellaneous language is used or not? If used, then it sends back to the complainer and ask to correct language. In addition, if not then it sends to the according department of the municipality.

A Municipal Corporation is providing necessary community services like health care, educational institution, housing, transport etc. by collecting property tax and fixed grant from the State Government.

NGOs are usually funded by donations but some avoid formal funding altogether and are run primarily by volunteers. NGO's can publicize their social causes on the site.

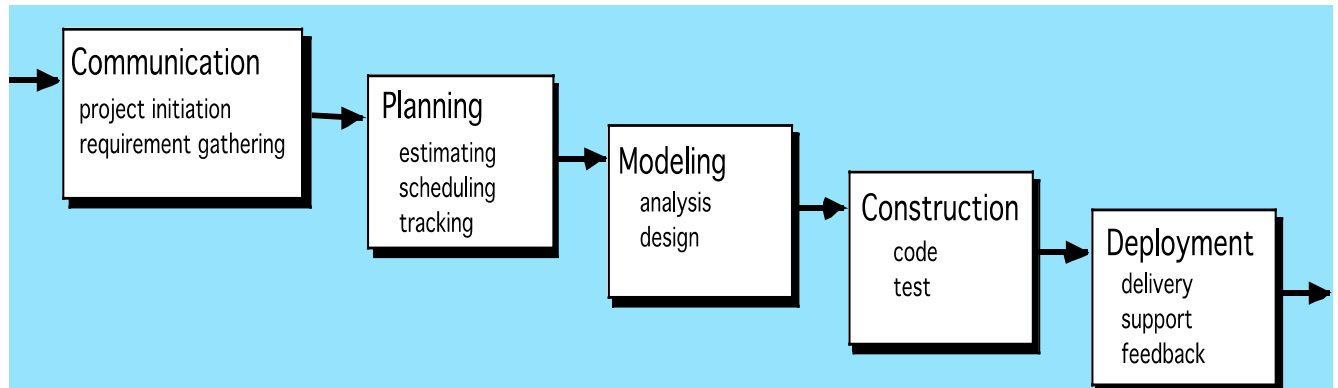
2.4 Non-functional requirements

1. Secure access of confidential data (user's details).
2. 24 X 7 availability.
3. Flexible service based architecture will be highly desirable for future extension
4. Simplicity of interface.
5. Website is highly customizable and flexible enough to easily deploy. That means, if we want to use this site for any other municipal authority, it would take least amount of time and efforts.

3.0 Processing Model

The Waterfall Model

Diagram:



Description:

The Waterfall model is a sequential design process, used in software development process, in which process is seen as flowing steadily downwards through the phases of conception , initiation , analysis , design , construction ,testing,production and maintenance.

It was first process model to be introduced and referred as a linear-sequential life cycle model. It is very simple and understand to use.

Waterfall approach was the first SDLC model to be used widely in software engineering to ensure success of the project. In waterfall model, the whole process of software development is divided into separate phases.

Phases of waterfall model:

1. Communication:

All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc. Here in this phase developer needs to communicate with the user to ask their specification and requirements.

Here some requirement specification of the Active city management:

- Secure access of confidential data (user's details).
- Flexible service based architecture will be highly desirable for future extension
- Simplicity of interface.
- Website is highly customizable and flexible enough to easily deploy. That means, if we want to use this site for any other municipal authority, it would take least amount of time and efforts.

2. Planning:

The requirement specification from first phase are studied in this phase and system design is prepared.

Planning of the Active city management:

- To create a user-friendly online interface for citizens to communicate with administrative body and, reduce the distance and time barrier between citizens and administration.

- To create an online platform where people can share ideas, invoke discussions, issue complaints, and create suggestion/petitions for improvement of city administration.
- To encourage the citizens to actively participate in city administration to bring transparency and flexibility in system.
- Users of the System
 - 1) Citizens of city.
 - 2) Local municipal authorities.
 - 3) Administrators.
 - 4) NGOs

3. Modeling:

There are some issues, which come up in the client environment. To fix those issues we need to analysis the modules of our product. If some module cannot satisfy customers satisfaction, then again those modules are send to planning phase and delivered again with some extra features.

Modeling in the Active city management:

If user has some question, regarding to some issue on the website then one extra module added called FAQs. Here user can give satisfied answer of the question. So to add this module first we analysis the requirement of the user.

4. Construction:

With inputs from system design, the system is first developed in small program called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as unit testing.

Construction in the Active city management:

Here suppose we construct the interface of the citizens the we first need to develop small units of that interface. E.g.

1. Users should be able to create new account, login to their existing accounts.
2. Authenticated users should be able to issue complaints, check complaint status, submit feedback, and browse through other complaints and their feedback.
3. Authenticated users should be able to create suggestions/petitions.

We need to develop this entire module and later combine these modules to develop citizens account interface.

5. Deployment:

Once the functional and non-functional testing is done, the product is deployed in the customer environment and giving feedback called Deployment.

Product should support in customer environment. Maintenance is done to deliver these changes in the customer environment.

Deployment in the Active city management:

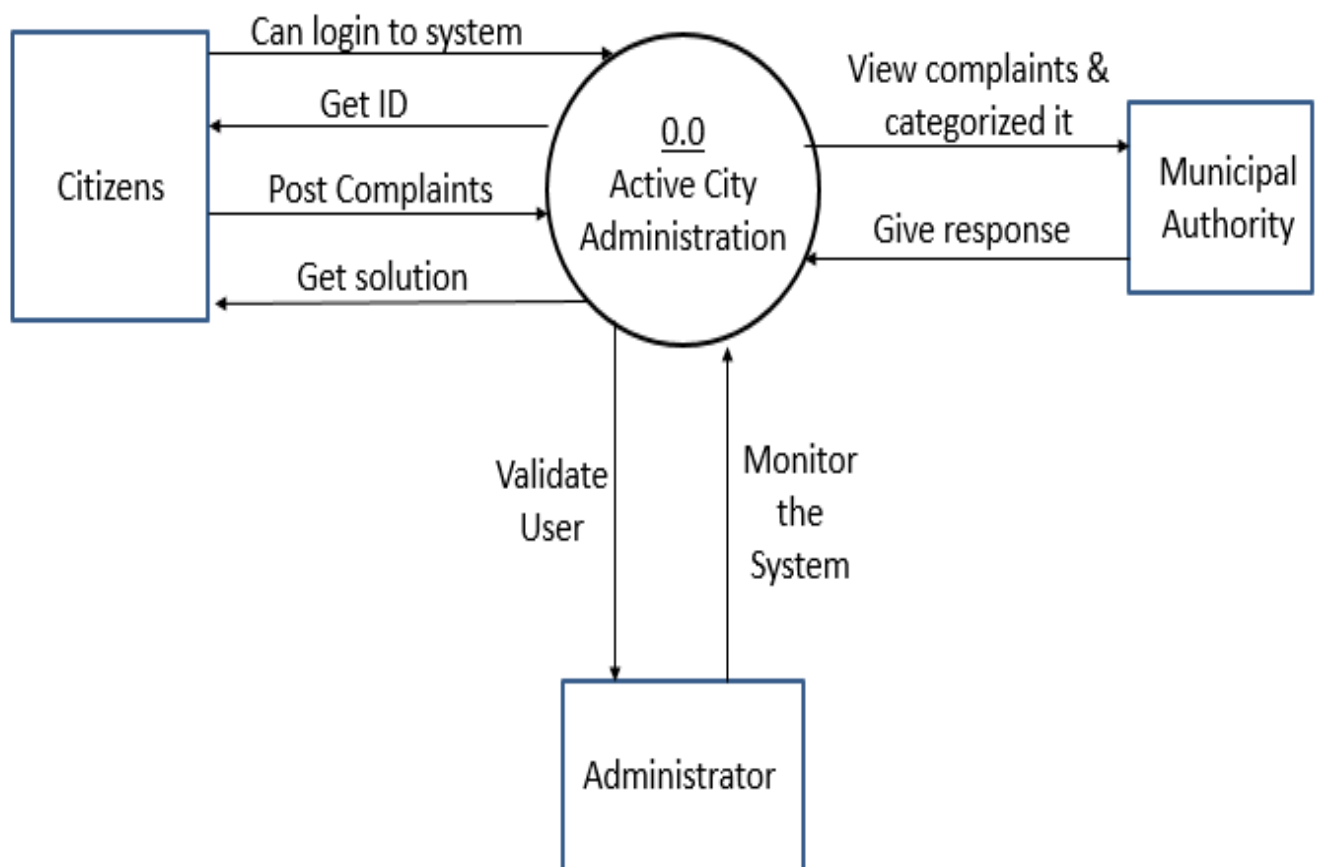
After the product is deployed to the municipality it is necessary that public and various sector of the Gov. are uses it and give feedback. The Maintenance is also necessary by managing database of the various users.

Model/Feature	Waterfall	Spiral	Incremental/Iterative
Specification of All the Requirements in the beginning	Yes	Not all and Frequently Changed	Not all and Frequently Changed
Long term project	Inappropriate	Appropriate	Appropriate
Complex Project	Inappropriate	Appropriate	Appropriate
Frequently Changed Requirements	Inappropriate	Appropriate	Appropriate
Cost	Not costly	Costly	Costly
Cost estimation	Easy to estimate	Difficult	Difficult
flexibility	Not	Less flexible	Flexible
Simplicity	Simple	Intermediate	Intermediate
Supporting high risk projects	Inappropriate	Appropriate	Appropriate
Guarantee of Success	Less	High	High
Customer Involvement	Low	Low, After Each Iteration	High, After Each Iteration
Testing	Late	At the end of each phase	After every iteration
Maintenance	Least maintainable	Yes	Maintainable
Ease of Implementation	Easy	Complex	Easy

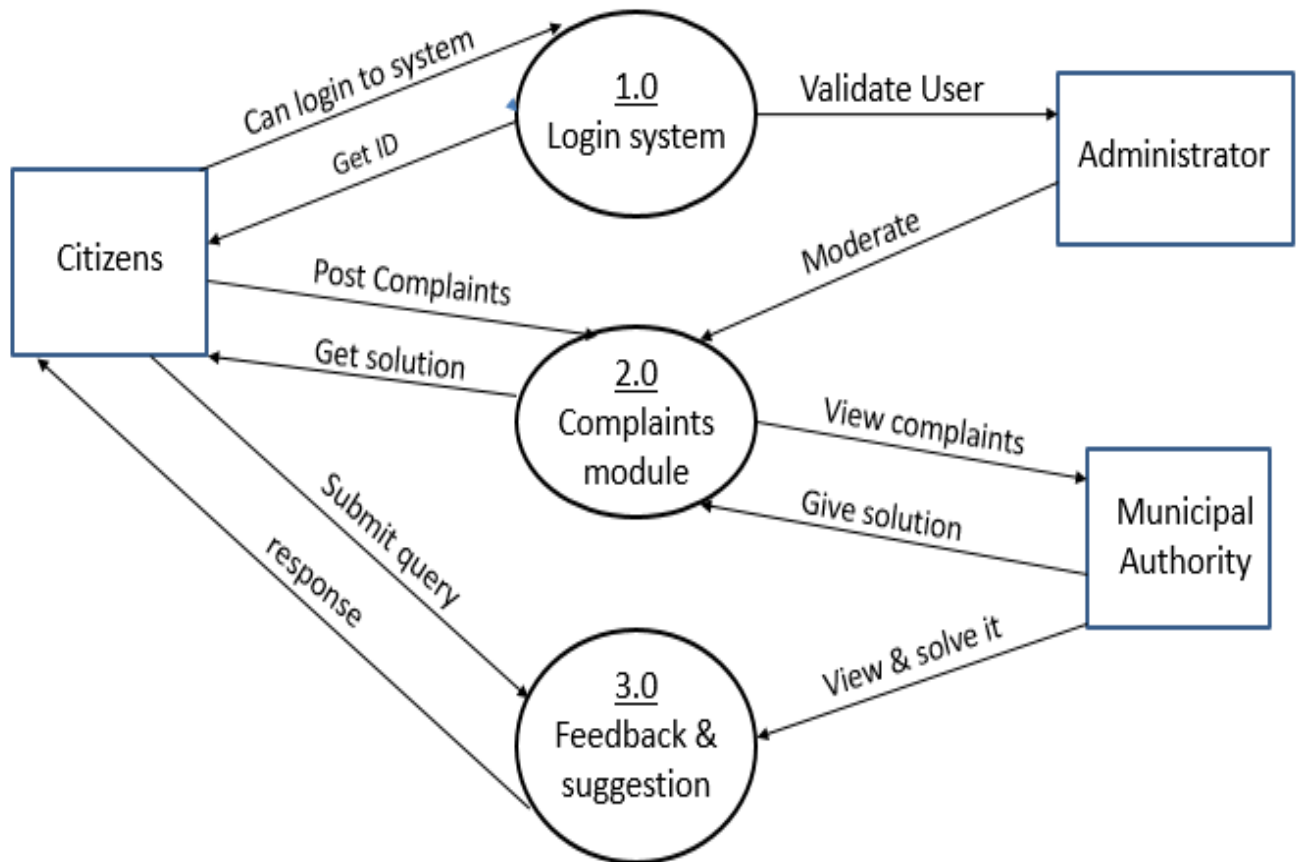
4.0 Various Diagrams

❖ Data Flow Diagram(DFD)

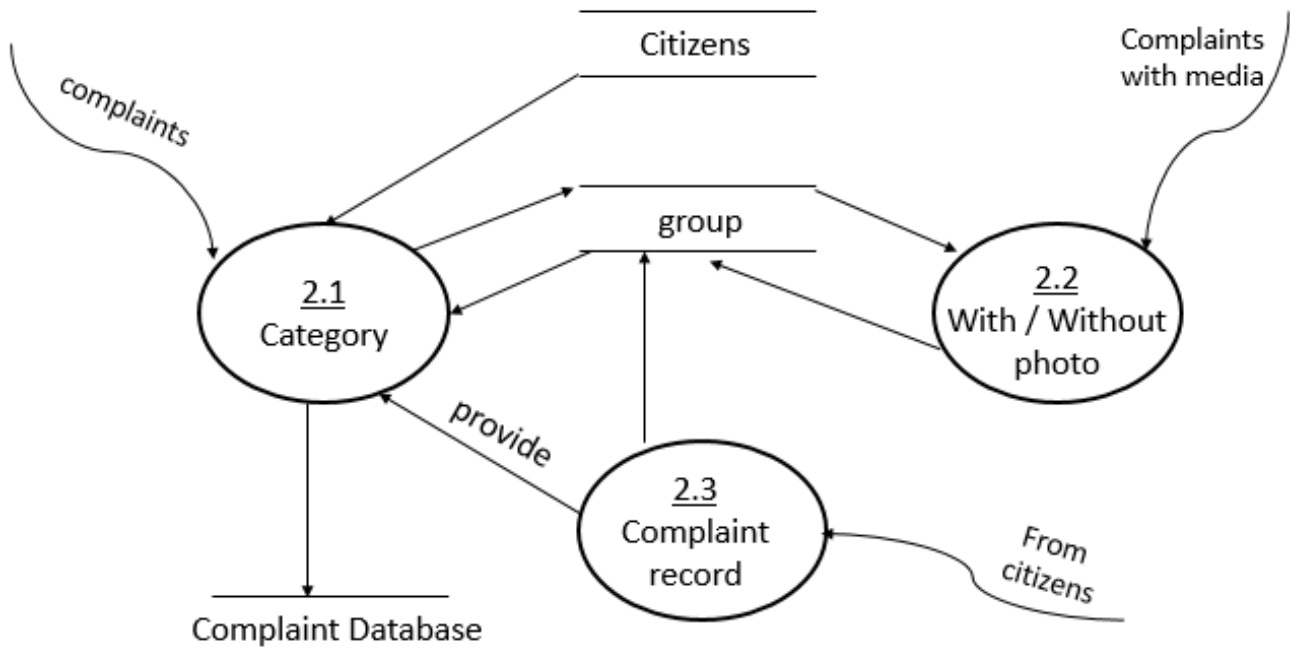
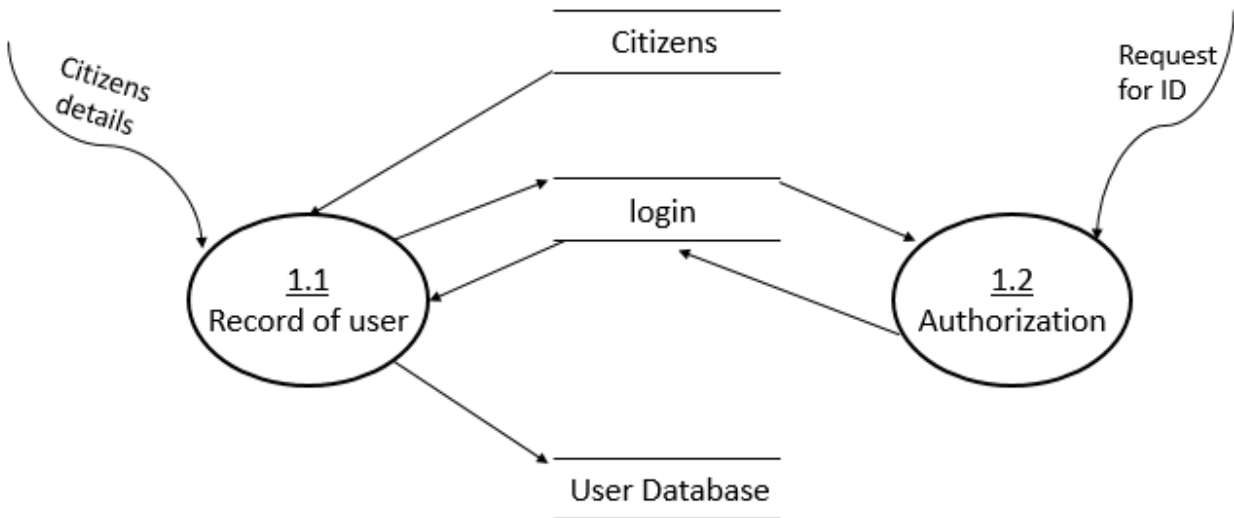
➤ Level 0 DFD

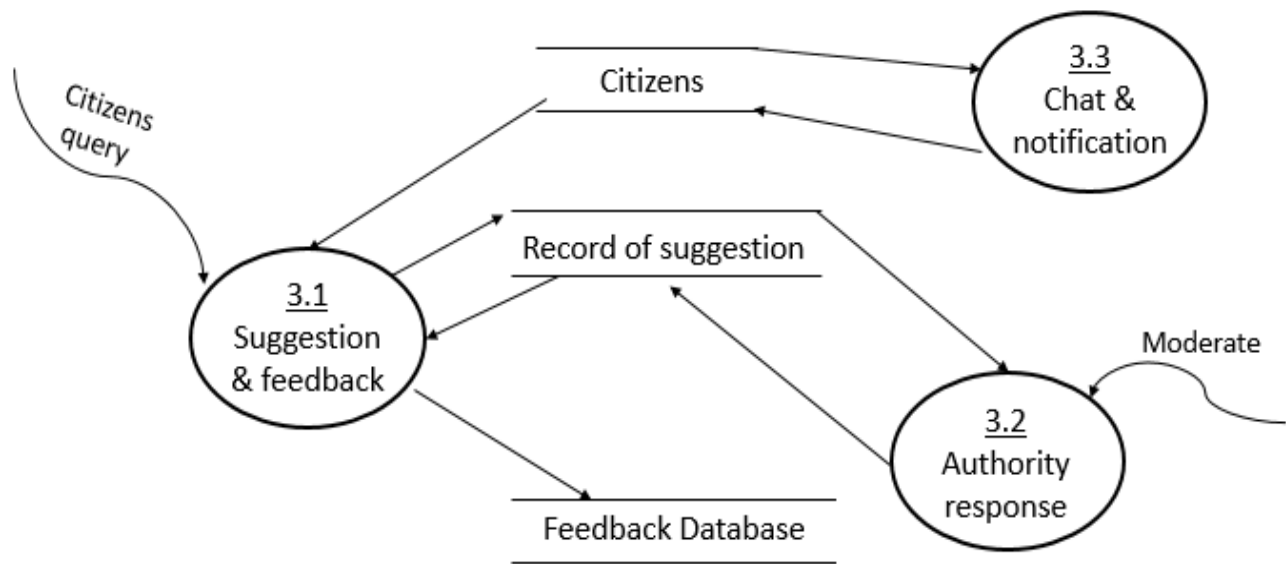


➤ Level 1 DFD

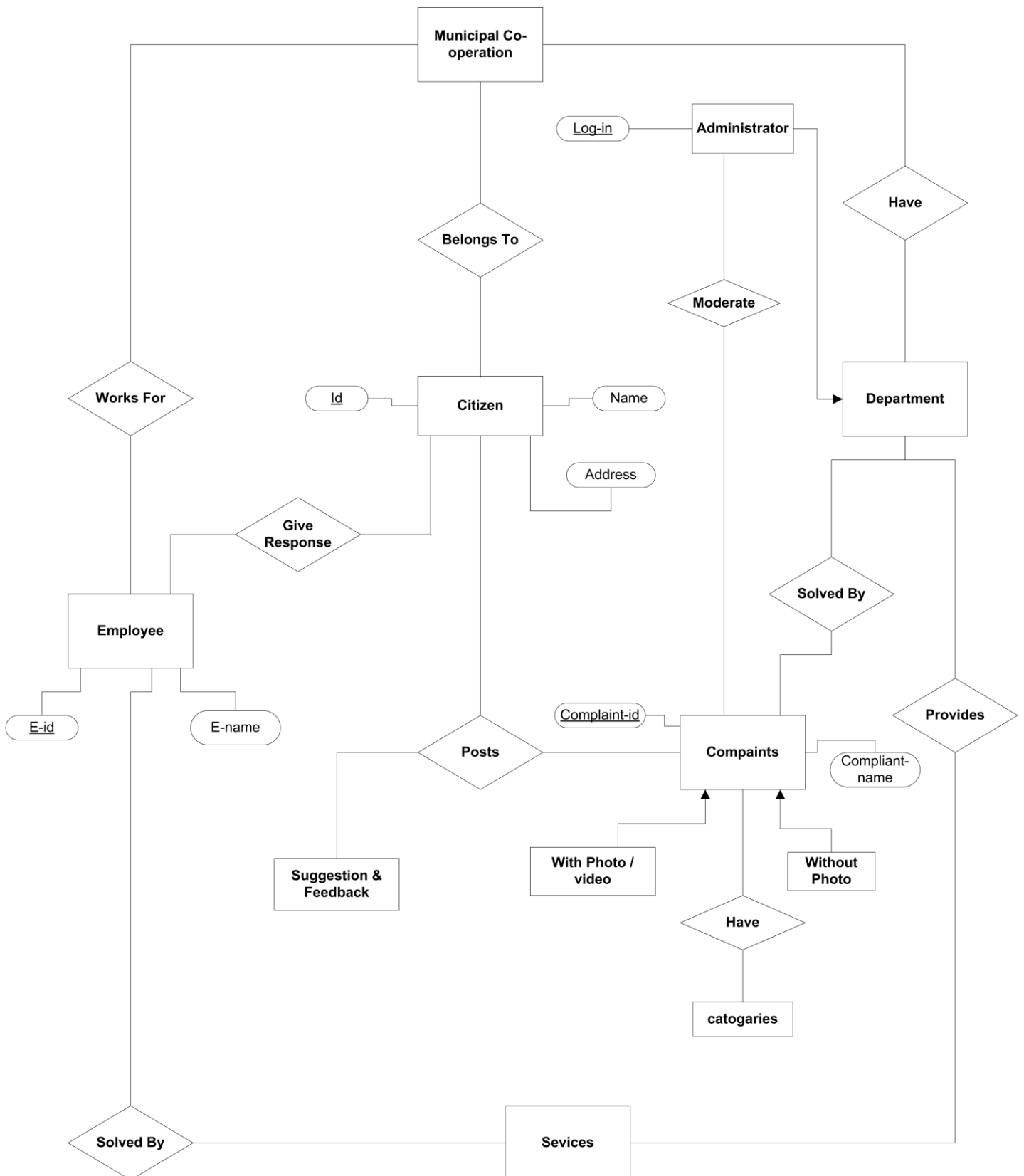


➤ Level 2 DFD

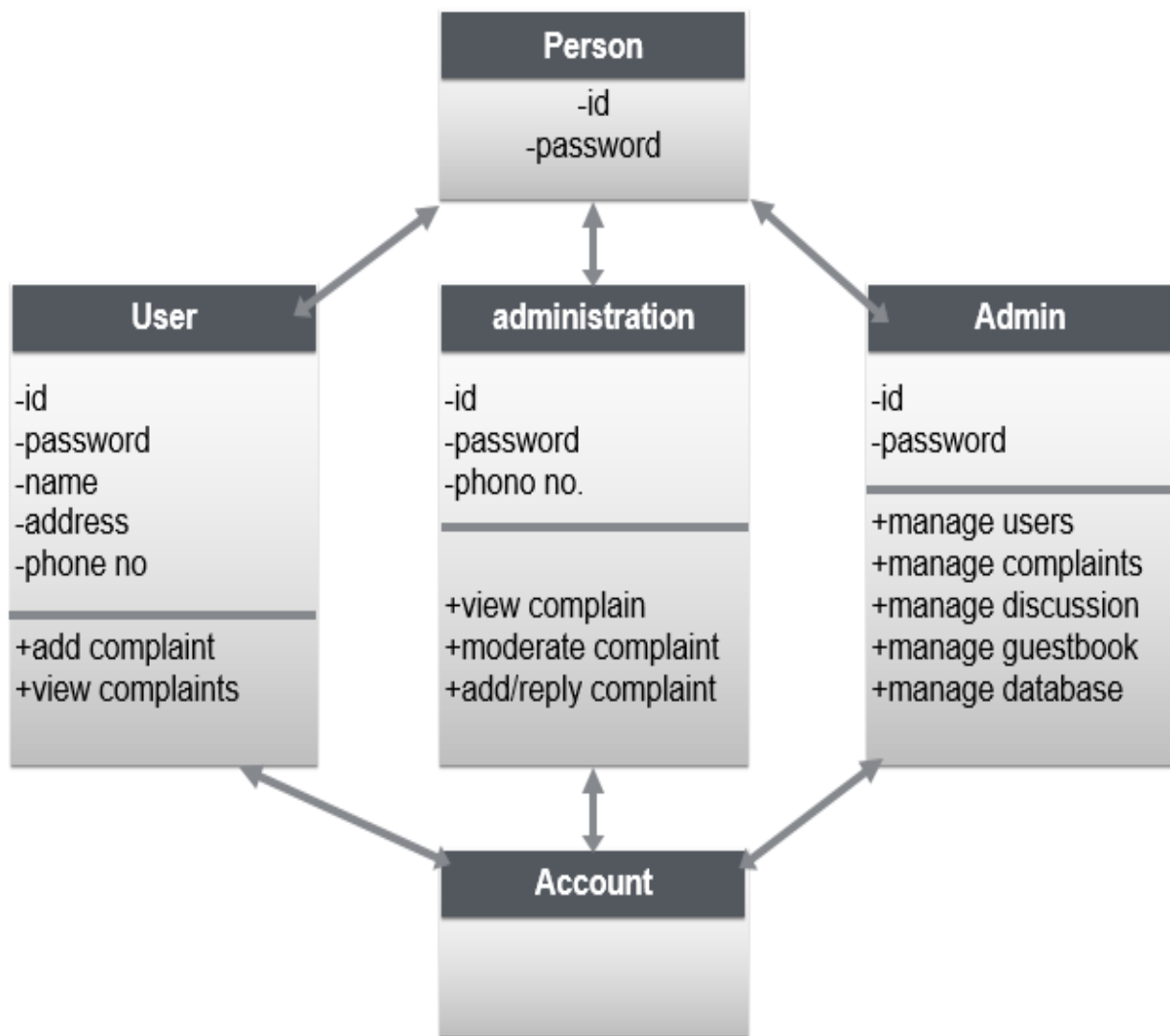




❖ Entity-Relationship Diagram (E-R Diagram)



❖ Class Diagram



5.0 Data Dictionary

➤ Table list

- citizen
- admin
- m_authority
- NGO
- complaints
- feedback
- pdf
- w_new

➤ citizen

Entity	Constraint	Data type(Size)	Description
u_id	PRIMARY KEY	Number(12)	User id(Adhar)
u_name	NOT NULL	Varchar(50)	User name
u_email	NOT NULL	Varchar(50)	User email
u_password	NOT NULL	Varchar(50)	User password
u_address		Varchar(100)	User address
u_phn		Number(10)	User contact

➤ admin

Entity	Constraint	Data type(Size)	Description
username		Varchar(50)	Admin name
password		Varchar(50)	Admin password
m_id	FOREIGN KEY from m_authority	Number(12)	Municipal authorities' id

➤ m_authority

Entity	Constraint	Data type(Size)	Description
m_name		Varchar(50)	Authority name
m_password		Varchar(50)	Authority password
m_id	PRIMARY KEY	Number(12)	Authority id

➤ NGO

Entity	Constraint	Data type(Size)	Description
n_id	PRIMARY KEY	Number(12)	NGO id
n_name	NOT NULL	Varchar(50)	NGO name
n_email	NOT NULL	Varchar(50)	NGO email
n_password	NOT NULL	Varchar(50)	password
n_address		Varchar(100)	NGO address
n_phn		Number(10)	NGO contact

➤ Complaints

Entity	Constraint	Data type(Size)	Description
c_id	PRIMARY KEY AUTO INCREMENT	Number(12)	Complaint id
c_name	NOT NULL	Varchar(50)	Complaint name
c_description	NOT NULL	Varchar(200)	Complaint description
c_category	NOT NULL	Varchar(50)	Category of Complaint
c_media		BLOB	By media(photo)
c_area		Varchar(50)	Area wise Complaint

➤ **feedback**

Entity	Constraint	Data type(Size)	Description
u_id	FOREIGN KEY from citizen	Number(12)	Citizen id
f_content	NOT NULL	Varchar(200)	Feedback description

➤ **pdf**

Entity	Constraint	Data type(Size)	Description
p_id	PRIMARY KEY	Number(12)	pdf file id
p_content	NOT NULL	BLOB	pdf content

➤ **w_new**

Entity	Constraint	Data type(Size)	Description
w_id	PRIMARY KEY	Number(12)	News id
p_id	FOREIGN KEY from pdf	Number(12)	pdf id
news		Varchar(200)	News content
circular		Varchar(200)	Circular content
notification		Varchar(200)	Notification content
tender		Varchar(200)	Tender content

6.0 Unit Testing

Black-Box Testing

- Black Box means opaque object or box where we can't see the internal structure.
- In Black box testing, we only concentrate on Input and output.
- Black-box test design is usually described as focusing on testing functional requirements. So we normally test the functionality of software without going deep in to its code and structure.
- Techniques that is used in Black Box testing
 1. Boundary-value analysis
 2. Error guessing
 3. Race conditions
 4. Cause-effect graphing
 5. Syntax testing
 6. State transition testing
 7. Graph matrix
 8. Equivalence partitioning
- Best example of Black box testing is Search on Google. User just enters keywords and get the expected results in turn. End user don't worry about what is behind this screen that is employed to fetch these results.
- Technical background is not the necessity of Black-Box Tester.

White-Box Testing

- White-Box testing is also known as clear box testing, glass box testing, transparent box testing and structural testing.
- White-Box testing deals with the internal structure and the internal working rather than only functionality.
- For White-Box testing, programming background is must because this helps in creating test cases for white-box testing.
- White-box testing is applied for Unit testing, integration testing and sometime in it is also used for system testing.
- These are few techniques that are used in white box testing
 1. Code Coverage
 2. Segment coverage: This is done to ensure that all statement or each line of code has been executed
 3. Compound condition coverage and loop coverage: In this we test all the conditions and all the branching and loops in code
 4. Data Flow testing: we test all the intermediate steps, in this we test how sequential steps behave
 5. Path Testing: In this we test all the path that is defined in code