



Palestine Technical University – Khadoori (PTUK)

Engineering Collage

Computer System Engineering (CSE)

Project : Municipalities Complaints Management

Supervised by: Dr.Nael Salman.

Team Members:

Ahmad Herzalla 202210435.

Yamen Sabbah 202210428.

Mostafa Mosleh 202210064.

Mohammed allam Abu-hmedat 202210324.

Municipalities Complaints Management

Table Content

Abstract.....	2
1. Introduction.....	2
1.1 Objective	3
1.2 Document Convention	3
1.3 Intended Audience and Reading Suggestions	4
1.4 Project Scope	5
References	6
2. REQUIREMENTS DEFINITION.....	7
2.1 Stakeholders.....	7
2.2 Methodology	7
2.3 Process Model	7
2.4 User Requirements (General)	8
2.4.1 Admin Functional Requirements	8
2.4.2 Citizen (User) Functional Requirements	8
2.4.3 Employee Functional Requirements.....	8
2.5 Non-Functional System Requirements	8
3 SYSTEM ARCHITECTURE.....	9
4. SYSTEM MODELS	10
4.1 Use Case Diagram.....	10
4.1.1 ADMIN USE CASE Diagram	10
4.1.2 Citizen (User) USE CASE Diagram	11
4.1.3 Employee USE CASE Diagram	12
4.2 Classes Diagram	13
4.3 Activity Diagram	14
4.4 Sequence Diagram.....	15
5 System building requirements	16
5.1 Hardware Requirements	16
5.2 Software Requirements	16
5.3 Development Tools.....	17
5.4 Stages of Implementation	17
5.5 Additional Considerations	18
5.6 Implementation and update	18

Municipalities Complaints Management

Abstract

This document presents the Software Requirements Specification (SRS) for the “Online Complaint Submission System for Municipalities in Palestine.” Designed as a comprehensive digital platform, the system enables citizens to submit, track, and manage complaints directed to their local municipal authorities. By incorporating dual interfaces—one for citizens and one for municipal administrators—the platform streamlines complaint processing, increases transparency, and promotes accountability in public service delivery. This SRS serves as the foundational reference for the entire software development lifecycle, detailing the functional and non-functional requirements, technical conventions, project scope, and stakeholder expectations.

1. Introduction

The purpose of this SRS document is to clearly and comprehensively define the requirements for the “Online Complaint Submission System for Municipalities in Palestine.” This document has been developed by a team of up to four software engineering students, following IEEE-standard practices to ensure consistency, clarity, and traceability. It is intended for use by software developers, municipal administrators, end users, and project stakeholders. The aim is to create a unified understanding of the system’s objectives and functionalities that will drive the design, development, and maintenance phases.

Public service delivery in many Palestinian municipalities is currently hampered by manual and outdated complaint management processes. These legacy systems often result in delayed responses and limited transparency. The proposed online system addresses these challenges by digitizing the entire complaint submission process. In doing so, the system seeks to:

- Reduce administrative processing times.
- Enhance transparency between citizens and local government.
- Facilitate better resource allocation by municipal administrators.
- Build trust through a more accountable and responsive service delivery model.

This document is organized into several key sections: objectives, document conventions, intended audience and reading suggestions, project scope, and references. Each section is developed to ensure that both technical teams and non-technical stakeholders gain a complete understanding of the project’s goals and implementation strategy.

Municipalities Complaints Management

1.1 Objective

The primary objective of the “Online Complaint Submission System for Municipalities in Palestine” is to develop a secure, user-friendly, and scalable online platform that modernizes the way citizens interact with their municipal authorities. The system is designed with the following goals:

- **For Citizens:**
 - Enable citizens to register, log in, and submit comprehensive complaints regarding municipal issues such as infrastructure, public services, or community safety.
 - Allow the inclusion of detailed descriptions, geolocation data, and supporting documents (e.g., photographs, PDFs) to substantiate each complaint.
 - Provide a real-time tracking mechanism for monitoring the progress and status of submitted complaints, with automated notifications for updates or resolutions.
- **For Municipal Administrators:**
 - Offer a dedicated, secure administrative dashboard for efficiently managing incoming complaints.
 - Enable functionalities to review, categorize, and prioritize complaints based on urgency, location, or nature of the issue.
 - Facilitate direct communication with citizens for clarifications or feedback and support the assignment of cases to specific departments.
 - Generate analytical reports and performance metrics that help in evaluating the effectiveness of municipal responses and overall system efficiency.

Through these objectives, the system intends to significantly reduce the administrative burden, improve service delivery, and foster a culture of transparency and accountability within Palestinian municipalities.

1.2 Document Convention

This SRS document employs standardized conventions to ensure that all information is presented in a clear, consistent, and accessible manner. The key conventions include:

- **Structured Sectioning and Numbering:**

The document is organized into numbered sections and subsections (e.g., 1, 1.1, 1.2) for logical progression and easy reference. This structure facilitates quick navigation and clarity.
- **Defined Terminology:**

All technical and domain-specific terms are defined upon first use. For example:

 - **Database:** Refers to the organized repository for electronic data. In this project, it will securely store user profiles, complaint details, attachments, and administrative logs.

Municipalities Complaints Management

- **Relation:** Describes the logical connections between data elements across tables. The system design will incorporate Entity-Relationship Diagrams (ERDs) to illustrate these connections (e.g., one-to-many or many-to-many relationships) and ensure data integrity.
- **Enhanced Formatting and Visual Aids:**
Important information is highlighted using bullet points, numbered lists, tables, and diagrams. ER diagrams will be used to represent the database schema and relationships visually. Bold text is used for critical concepts, while italics emphasize key terms.
- **Citing External Standards and References:**
The document adheres to external standards, such as IEEE Std 830-1998, and relevant case studies are referenced in the final section. This practice ensures that the SRS conforms to industry best practices and provides verifiable information.
- **Version Control and Document Maintenance:**
This SRS will be maintained in a version-controlled repository (e.g., Git) to track revisions and contributions. Version control is essential for ensuring that every team member works from the latest document version and for maintaining historical records of changes.

These conventions are critical for maintaining consistency, especially when documenting complex technical details like the system's database design and data relationships.

1.3 Intended Audience and Reading Suggestions

This SRS document is intended for a broad range of stakeholders, including:

- **Software Development Team:**
Developers, designers, and testers who require detailed functional and non-functional requirements to implement and validate the platform. They will use the technical specifications and ER diagrams to guide system development.
- **Municipal Administrators:**
Local government officials who will use the administrative dashboard to manage and resolve citizen complaints. This document helps ensure that the system meets operational requirements and facilitates effective service delivery.
- **End Users (Citizens):**
While citizens are non-technical, they will directly interact with the user interface for submitting complaints. Based on this document, user manuals and guides will be developed to simplify the user experience.
- **Project Managers and Stakeholders:**
Individuals responsible for overseeing project progress, budgeting, and alignment with organizational goals. This document provides them with a high-level overview of the project scope and implementation strategy.
- **Quality Assurance and Maintenance Teams:**
These teams will use the document as a baseline to develop test cases, set up maintenance protocols, and ensure long-term system reliability.

Municipalities Complaints Management

Reading Suggestions:

- **For Technical Personnel:** Concentrate on Sections 1.1 and 1.2 for a deep understanding of the system's objectives and technical documentation conventions.
- **For Operational Staff:** Section 1.3 offers valuable insights into the system's functionalities and its impact on daily workflows.
- **For General Stakeholders:** A thorough reading of the entire document is recommended to appreciate the comprehensive scope and benefits of the project.

1.4 Project Scope

The “Online Complaint Submission System for Municipalities in Palestine” is designed to deliver a comprehensive solution that modernizes and enhances complaint management. The project scope includes the following major components:

- **Citizen Interface:**
 - **User Registration and Login:** A secure portal for citizens to create accounts, log in, and manage their profiles.
 - **Complaint Submission Form:** A detailed form capturing personal data, descriptions of issues, geographical location, and the specific municipality involved. The form supports file attachments to include evidence.
 - **Complaint Tracking:** A real-time tracking system that allows users to monitor the status of their complaints, view historical submissions, and receive automated notifications regarding updates or resolutions.
- **Administrative Dashboard:**
 - **Secure Access:** A dedicated portal for municipal administrators to securely log in and access complaint data.
 - **Complaint Management:** Tools for categorizing, prioritizing, and assigning complaints. This includes filtering options based on urgency, issue type, and location.
 - **Status Updates and Communication:** Functionality for updating the progress of complaints and facilitating direct communication with citizens.
 - **Reporting and Analytics:** Features to generate detailed reports and performance metrics, providing insights into complaint trends, resolution times, and departmental efficiency.
- **Data Management and Security:**
 - **Database Design:** A robust database system will store all relevant data, including user profiles, complaint records, and administrative logs. The database design will incorporate clearly defined relationships (illustrated via ER diagrams) to ensure data consistency.
 - **Data Protection:** The system will adhere to local data protection regulations to maintain the confidentiality and integrity of personal information.
 - **Backup and Recovery:** Strategies will be in place to ensure data backup and recovery in the event of system failures.

Municipalities Complaints Management

- **Future Enhancements:**

- **Scalability:** The system will be designed to accommodate additional features, such as mobile app integration, multi-language support, and advanced real-time analytics.
- **Integration Capabilities:** Provisions for integrating the system with other municipal management systems and public service platforms.
- **Automated Support:** Future modules may include automated notifications, live chat support, and AI-driven analytics to further streamline complaint handling.

The scope defined herein ensures that the project delivers a robust and user-centric solution that significantly enhances the efficiency, transparency, and quality of public service delivery in Palestinian municipalities.

References

- IEEE Standard for Software Requirements Specifications (IEEE Std 830-1998).
- Scholarly articles and literature on E-Government Systems and Digital Public Services.
- Case studies on the implementation of online complaint management systems in public institutions.

Municipalities Complaints Management

2. REQUIREMENTS DEFINITION

2.1 Stakeholders

- **Citizens (Users):** Register, select municipality, submit/track/cancel tickets (complaints or service requests), attach documents/photos, and receive notifications.
- **Complaint-Handling Employees:** Review, assign, respond, escalate, and close tickets; trigger backend service workflows; access knowledge base and audit trails.
- **System Administrator (Admin):** Manage users/roles, configure services/SLAs/branding, set up notifications, monitor health, export data, enforce security policies.
- **Municipality IT/Management:** View dashboards and analytics on ticket volumes, response times, SLA compliance, and service performance.

2.2 Methodology

Our workflow:

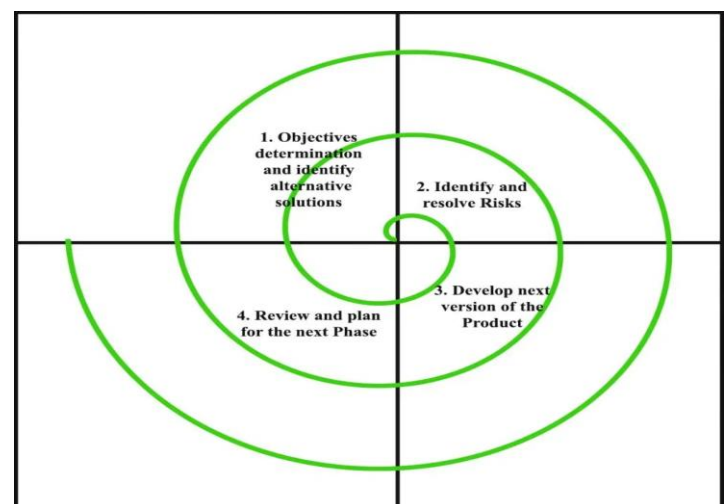
1. **Stakeholder Identification** → clarify who uses or benefits from the system.
2. **Scope & Objectives** → define core functions (ticketing, tracking, notifications, reporting).
3. **Requirements Elicitation** → ask targeted questions to uncover non-functional needs (security, scalability, usability).
4. **Process Model Selection** → choose an iterative model that guides design, development, and risk management (see 2.3).
5. **Technical Planning** → specify database, hosting, programming languages/frameworks.
6. **Detailed Requirements Definition** → write functional requirements per role and non-functional system requirements to support implementation and testing.

2.3 Process Model

We use the **Spiral Model**—each cycle includes:

1. **Planning & Requirements**
2. **Risk Analysis & Prototyping**
3. **Engineering (Design → Build → Test)**
4. **Evaluation & Stakeholder Review**

This ensures ongoing risk assessment (security, performance, integration) and progressive refinement, avoiding late surprises common in strict Waterfall or V-Model approaches.

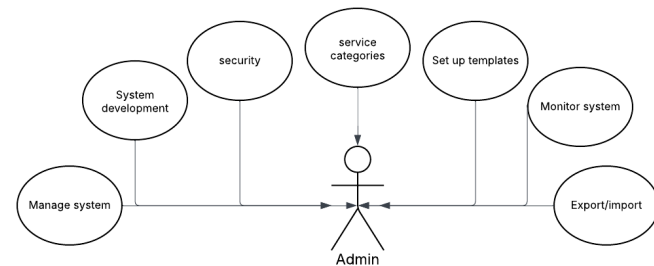


Municipalities Complaints Management

2.4 User Requirements (General)

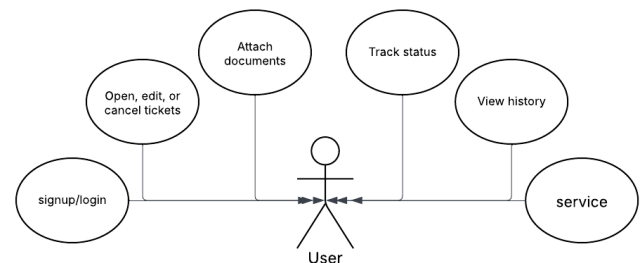
2.4.1 Admin Functional Requirements

- Manage users, roles, and permissions
- Configure service categories, SLAs, and municipal branding
- Set up notification templates/channels
- Monitor system health and view audit logs
- Export/import data and generate reports
- Enforce security policies (2FA, password rules)
- Manage the system and updated



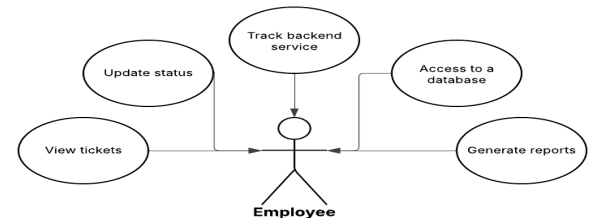
2.4.2 Citizen (User) Functional Requirements

- Secure signup/login and municipality selection
- Open, edit, or cancel tickets (complaints/service requests)
- Attach documents/photos and geotag locations
- Track status and receive notifications
- View history and provide feedback
- Submitting a municipal service request (extending water and electricity lines, etc.)



2.4.3 Employee Functional Requirements

- View and assign tickets in a dashboard
- Update status, add comments, and escalate as needed
- Trigger and monitor backend service workflows
- Generate performance reports and monitor SLAs
- Access to a database of municipal reports



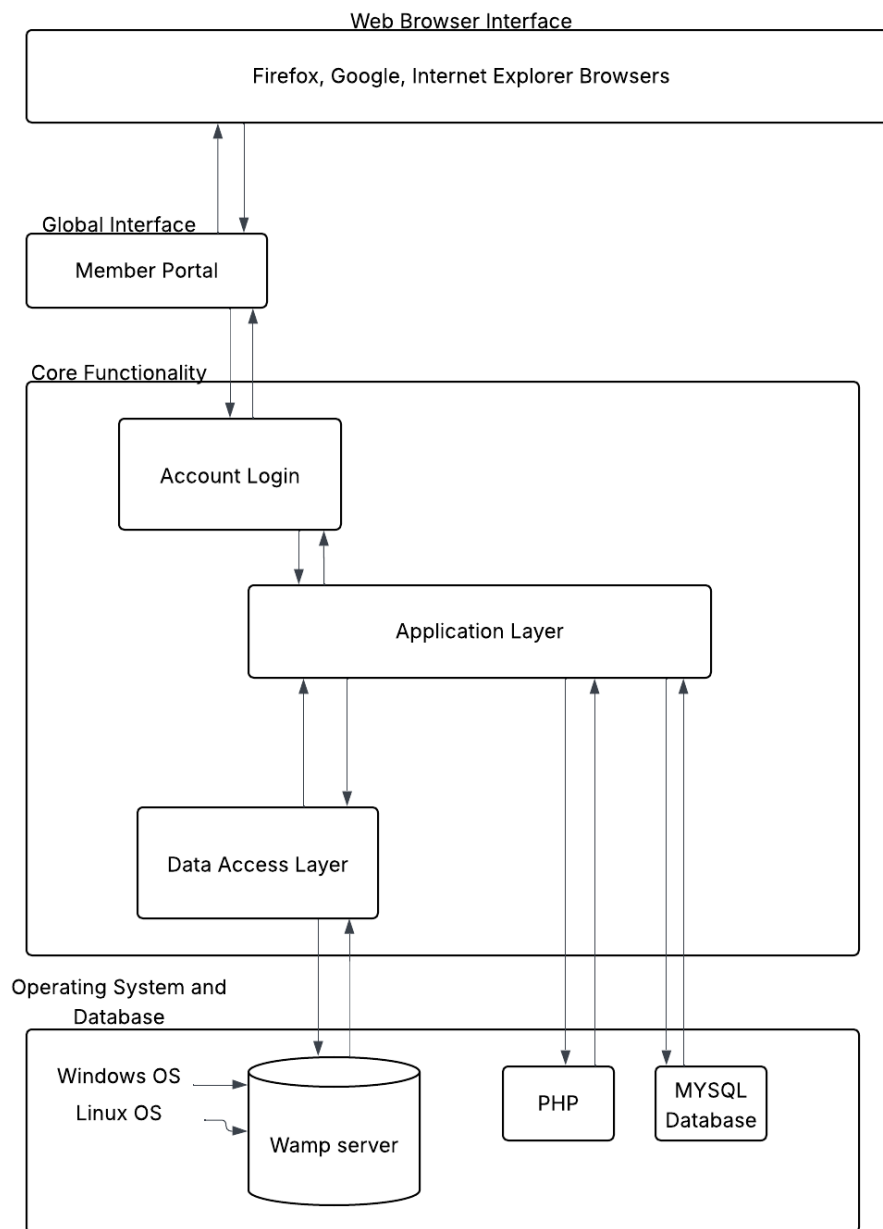
2.5 Non-Functional System Requirements

- **Scalability:** Support growing user base and municipalities without performance loss.
- **Security:** End-to-end encryption of messages/attachments; OWASP best practices; role-based access control.
- **Usability:** Responsive, intuitive UI on desktop and mobile; simple search and filtering.
- **Reliability & Availability:** $\geq 99.5\%$ uptime; automated backups and disaster recovery.
- **Analytics & Reporting:** Dashboards for open/in-progress/closed tickets; trends (average resolution time, tickets per category).

Municipalities Complaints Management

3 SYSTEM ARCHITECTURE

Organises the system into a set of layers (or abstract machines) each of which provide a set of services, it is support the incremental development of sub-systems in different layers. When a layer interface changes, only the adjacent layer is affected. As shown in figure

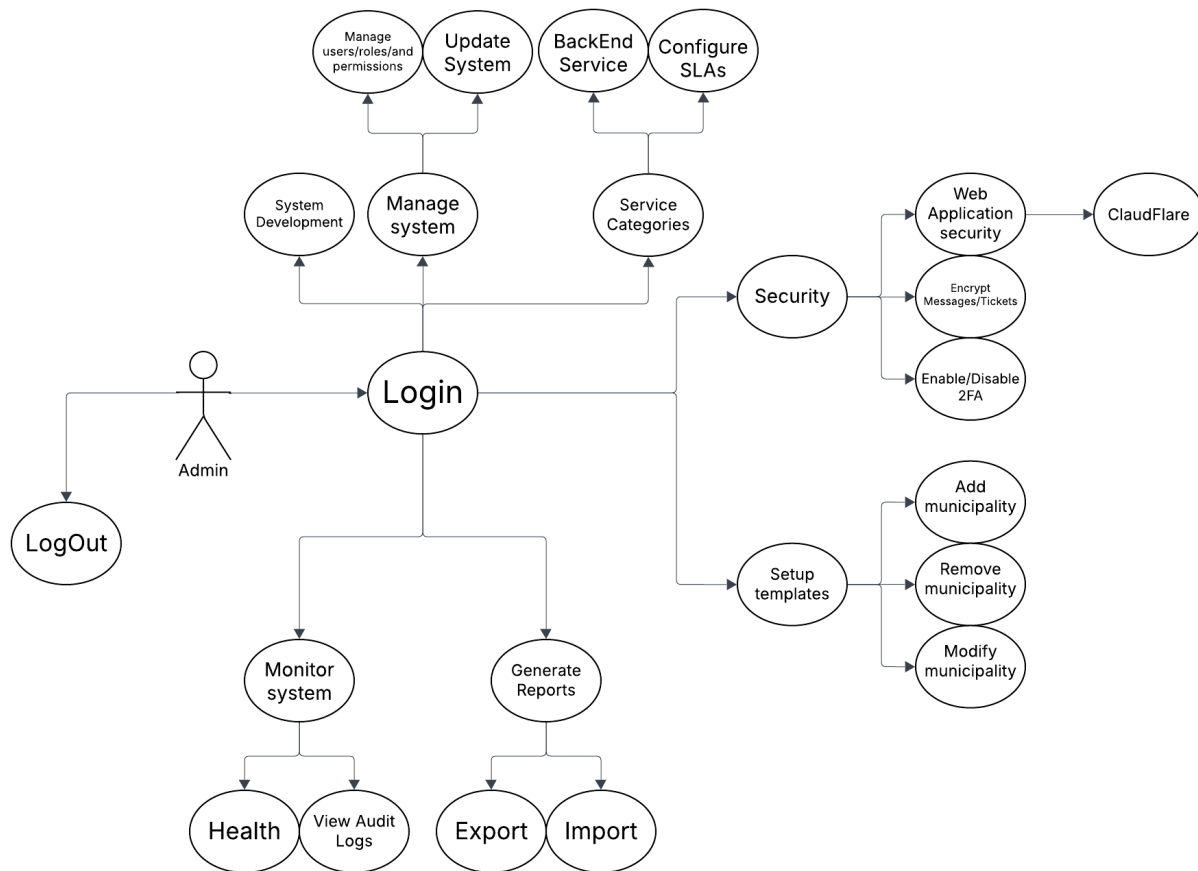


4. SYSTEM MODELS

4.1 Use Case Diagram

4.1.1 ADMIN USE CASE Diagram

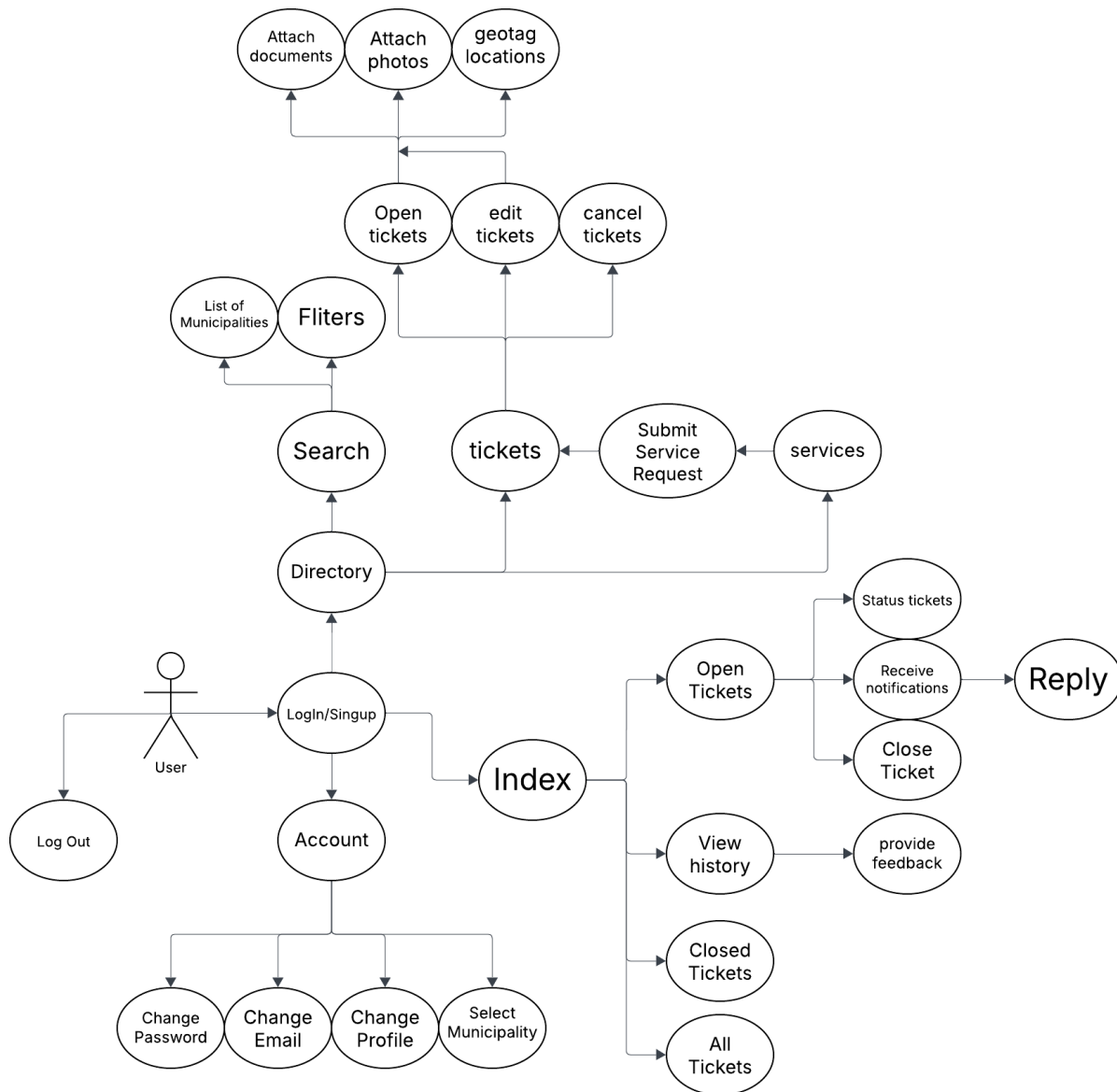
This drawing below shows ADMIN USE CASE Diagram After analyzing the stakeholders' requirements in Section 2.4.1 Admin Functional Requirement



Municipalities Complaints Management

4.1.2 Citizen (User) USE CASE Diagram

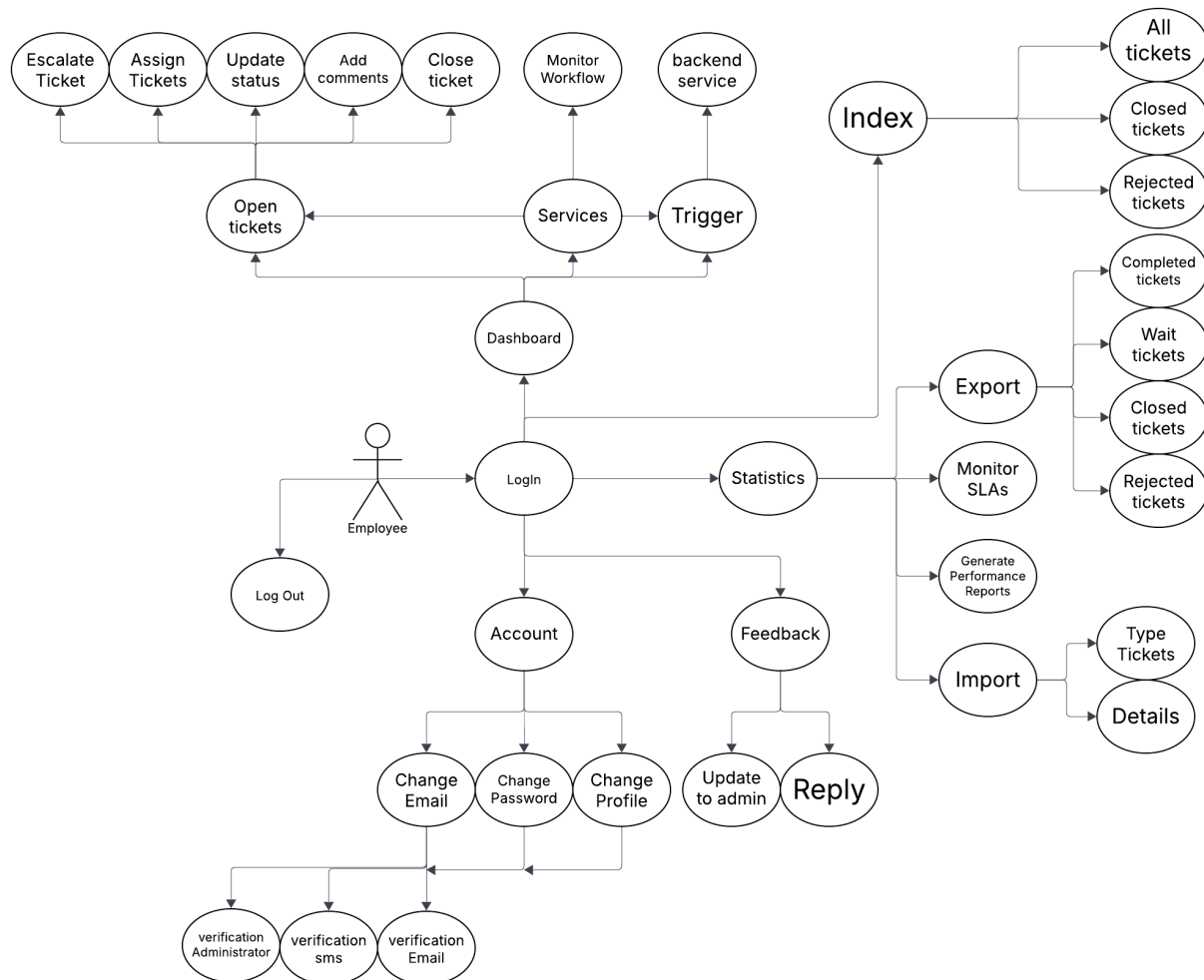
This drawing below shows Citizen (User) USE CASE Diagram After analyzing the stakeholders' requirements in Section 2.4.2 Citizen (User) Functional Requirements



Municipalities Complaints Management

4.1.3 Employee USE CASE Diagram

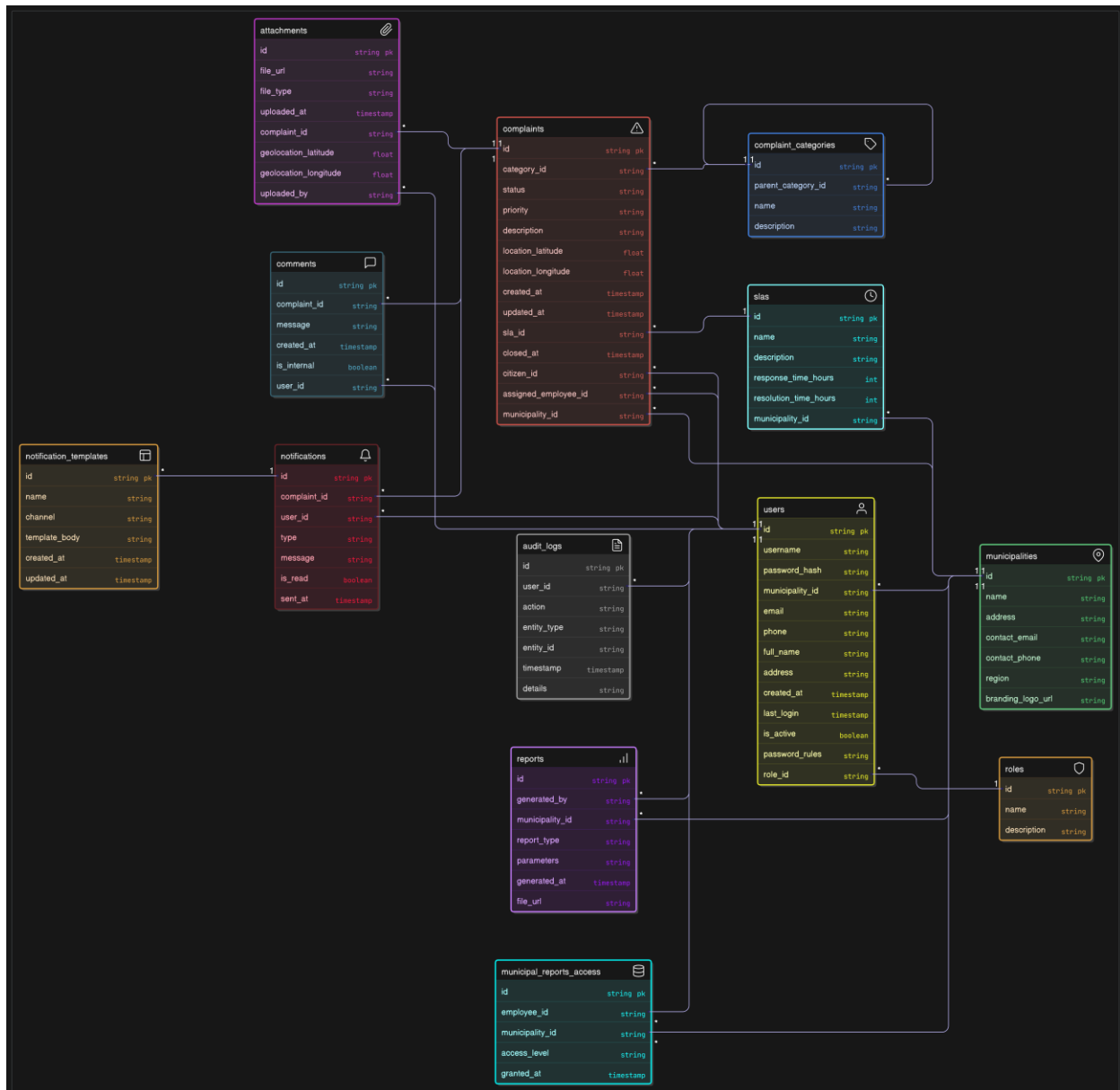
This drawing below shows Employee USE CASE Diagram After analyzing the stakeholders' requirements in Section 2.4.3 Employee Functional Requirements



Municipalities Complaints Management

4.2 Classes Diagram

This diagram shows the classes and relationships that will be implemented in the database to meet the needs of the system.

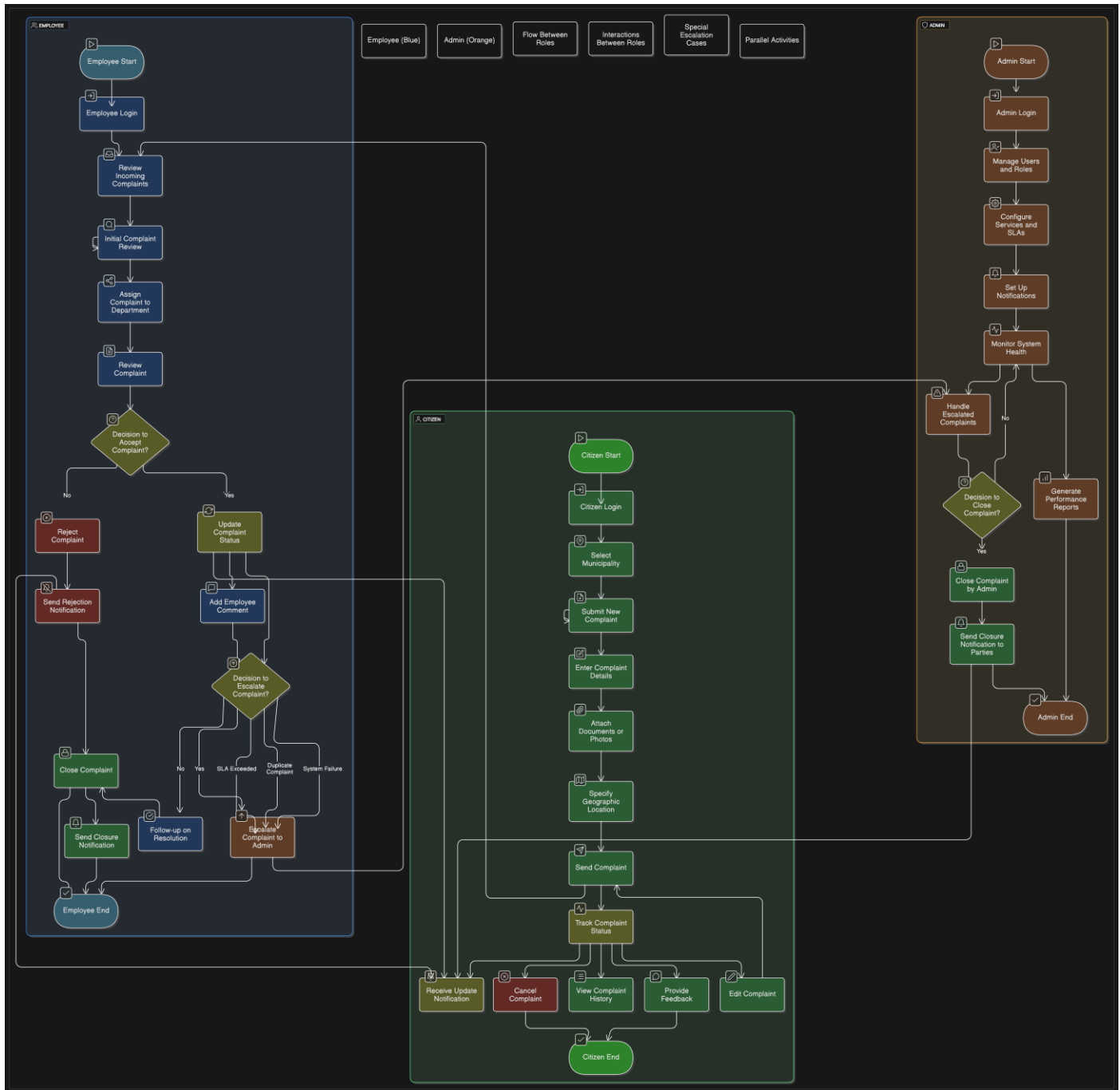


Municipalities Complaints Management

4.3 Activity Diagram

This drawing shows Activity Diagram for all Users Roles.

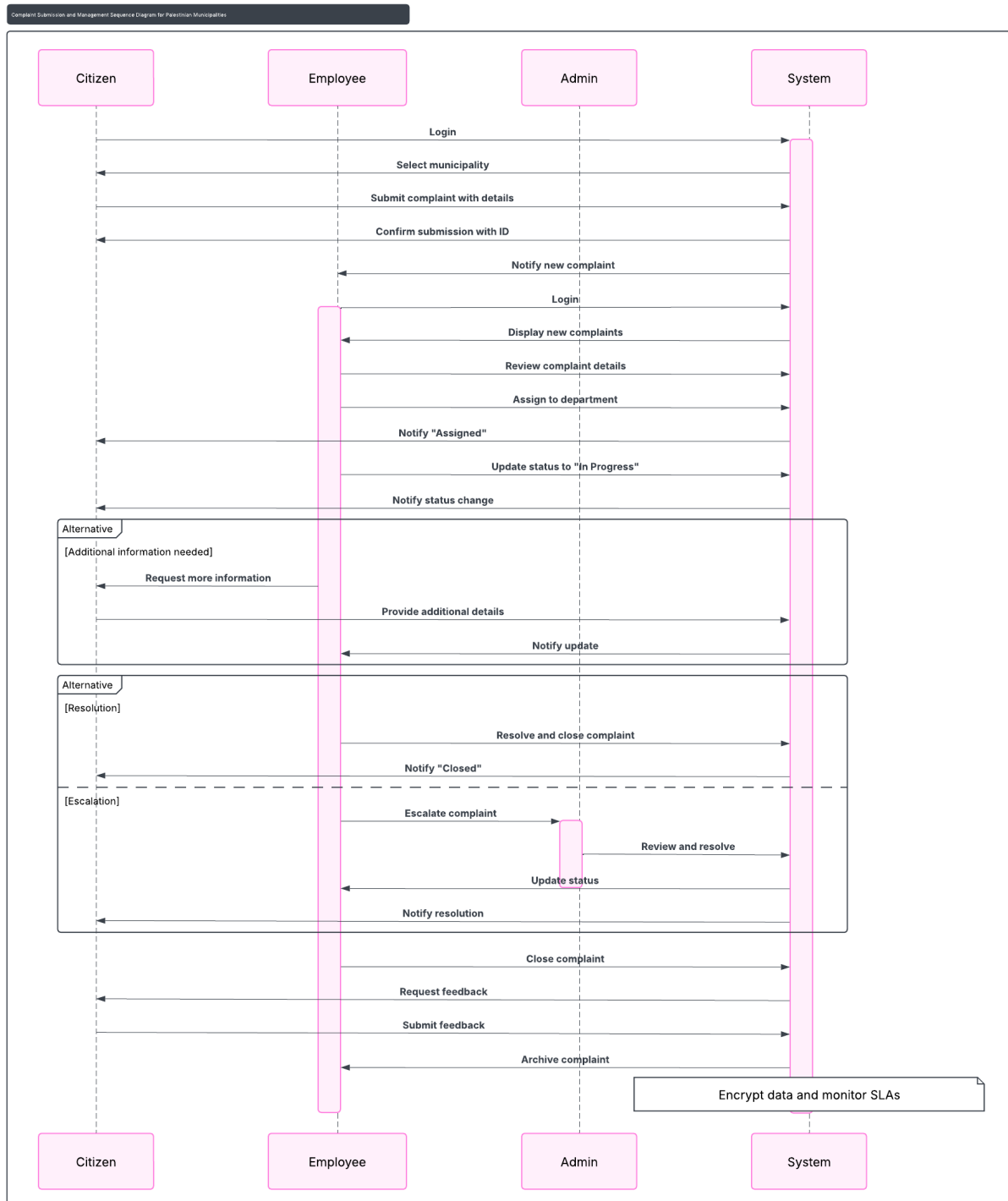
Activity Diagram : it's a visual way to understand and document how things work in a system or process, step by step. It helps us see who does what, what decisions are made, and what can happen simultaneously.



Municipalities Complaints Management

4.4 Sequence Diagram

This diagram shows the Sequence Diagram for Admin, Citizen and Employee. Sequence Diagram helps you visualize how different parts of a system communicate with each other in a specific scenario and in what order those communications happen. It's great for understanding the dynamic behavior of a system and the flow of information between its components.



Municipalities Complaints Management

5 System building requirements

To successfully develop the “Online Complaint Submission System for Municipalities in Palestine,” a comprehensive set of resources and a structured implementation process are required. This chapter outlines the hardware, software, and development tools necessary to build the system, as well as the stages of implementation. The requirements ensure the system is secure, scalable, and user-friendly, aligning with the project’s objectives as defined in the Software Requirements Specification (SRS).

5.1 Hardware Requirements

The system will be hosted on a web server and requires robust hardware to support its operations, including user traffic, data storage, and processing. The following hardware components are essential:

- **Server:** A dedicated or virtual server with sufficient processing power (e.g., multi-core CPU), memory (e.g., 16GB RAM or higher), and storage (e.g., SSD with at least 100GB) to host the website and manage the database. The server must handle concurrent user requests and large data volumes, accommodating the expected growth in users and municipalities.
- **Backup Server or Cloud Storage:** A secondary server or cloud-based solution (e.g., AWS S3 or Google Cloud Storage) is required for regular backups of user data, complaints, attachments, and system logs to ensure data redundancy and disaster recovery.
- **Network Infrastructure:** Reliable and secure network connectivity, including firewalls, routers, and load balancers, is necessary to ensure consistent access to the system and protect against unauthorized access or network failures.

5.2 Software Requirements

The software stack includes technologies for hosting, data management, development, and security. The following components are required:

- **Web Server Software:** A web server such as [Apache or Nginx](#) to host the website and manage HTTP requests efficiently.
- **Database Management System:** A relational database system like [MySQL](#) to store user profiles, complaint records, attachments, and administrative logs. The database must support secure storage, efficient querying, and scalability.
- **Programming Languages and Frameworks:**
 - **Back-end:** [PHP](#) will be used for server-side scripting to implement business logic and database interactions.
 - **Front-end:** [HTML5](#), [CSS3](#), and [JavaScript](#) will create a responsive and intuitive user interface. JavaScript frameworks such as React or Vue.js may be utilized to improve interactivity and user experience.
- **Security Tools:** To protect sensitive data and ensure compliance with security standards, the system will include:
 - [HTTPS/SSL](#) for end-to-end encryption of data transmission.

Municipalities Complaints Management

- [Web Application security \(Cloudflare of WAF\)](#) to prevent common vulnerabilities (e.g., SQL injection, cross-site scripting) based on OWASP guidelines.
- [Role-based access control \(RBAC\)](#) to manage permissions for citizens, employees, and administrators.

5.3 Development Tools

The development process requires tools to support coding, collaboration, testing, and project management. The following tools are essential:

- **Integrated Development Environment (IDE):** Developers will use IDEs such as [Visual Studio Code](#) for coding, debugging, and testing.
- **Version Control System:** [Git](#), hosted on platforms like [GitHub](#), will manage code versions and facilitate team collaboration.
- **Project Management and Collaboration Tools:** Tools like [Jira](#), [Trello](#) will organize tasks, track progress, and ensure effective communication among team members.
- **Testing Tools:** Tools for unit testing (e.g., [PHPUnit](#)), integration testing, and user acceptance testing (UAT) will validate the system's functionality, security, and performance.

5.4 Stages of Implementation

The system will be developed and deployed in structured stages to ensure quality and alignment with the project's requirements. The key stages are:

1. **Planning and Design:**
 - Define the system architecture, including technology stack and frameworks.
 - Design the database schema using Entity-Relationship Diagrams (ERDs) to ensure data integrity and efficient relationships (e.g., users, complaints, municipalities) It is explained in section 4.2 Classes Diagram .
 - Create wireframes and mockups for user interfaces (citizen portal and administrative dashboard).
2. **Development:**
 - Implement back-end logic using PHP for features like user authentication, complaint submission, tracking, and reporting.
 - Develop front-end interfaces with HTML, CSS, and JavaScript for a responsive and user-friendly experience.
 - Integrate the database with the application to securely store and retrieve data.
3. **Testing:**
 - Conduct unit testing to verify individual components (e.g., login functions, complaint submission).
 - Perform integration testing to ensure modules (e.g., user registration, notification system) work together seamlessly.

Municipalities Complaints Management

- Execute user acceptance testing (UAT) with stakeholders to confirm the system meets functional and non-functional requirements.
- 4. **Deployment:**
 - Set up the production environment, including the web server, database, and security configurations (e.g., SSL certificates).
 - Deploy the website and verify all components are correctly configured.
 - Conduct load testing to ensure performance under expected traffic conditions.
- 5. **Maintenance:**
 - Monitor system performance and user feedback after deployment.
 - Apply updates and security patches to maintain system integrity.
 - Provide ongoing support to address issues or bugs reported by users.

5.5 Additional Considerations

To meet the project's non-functional requirements, the following considerations are addressed:

- **Scalability:** The system must support a growing number of users and municipalities. This may involve load balancing, database optimization, or scalable cloud hosting.
- **Security:** Robust measures, including data encryption, secure authentication (e.g., two-factor authentication), and regular security audits, will protect sensitive user data and complaints.
- **Usability:** The interface must be intuitive and accessible on desktop and mobile devices, ensuring a seamless experience for all users.
- **Reliability:** The system should achieve at least 99.5% uptime, supported by automated backups and a disaster recovery plan.

5.6 Implementation and update

This section is dedicated to the system preparation and development phase, which will be carried out through our dedicated GitHub page. For more information, visit the following link:

<https://github.com/Ahmad-Herzalla0/Municipalities-Complaints-Management.git>