**Local Issues Classification and Resolution System**

**Submitted By**

**Tariq Hussain**

**04072013036**

**Supervised By**

**Dr. Ayyaz Hussain**

**Department of Computer Science**

**Quaid-I-Azam University,**

**Islamabad.**

**Abstract**

The **Local Issues Classification and Resolution System** addresses the challenge of ensuring local issues highlighted in national newspapers are efficiently communicated to relevant departments. This project focuses on building a web-based system that automatically extracts and classifies local issues based on their content, then assigns them to the appropriate stakeholders for resolution. By tracking the status of issues, users can see whether appropriate action is taken in a timely manner. The system provides a structured approach to managing local issues and involves focal persons for each department, including a super focal person to oversee all departments.

Table of Contents

[**Chapter 1** 6](#_Toc179472661)

[1.1 Introduction 6](#_Toc179472662)

[1.1.1 Project Overview: 6](#_Toc179472663)

[**1.2 Project Organization** 6](#_Toc179472664)

[1.2.1 Software Process Model: 6](#_Toc179472665)

[1.2.2 Roles and Responsibilities: 6](#_Toc179472666)

[1.2.3 Tools and Technology: 7](#_Toc179472667)

[**1.3 Project Management Plan** 7](#_Toc179472668)

[**Chapter 2** 10](#_Toc179472669)

[**2.1 Introduction** 10](#_Toc179472670)

[2.1.1 Product Overview: 10](#_Toc179472671)

[**2.2 Specific Requirements** 11](#_Toc179472672)

[2.2.1 External Interface Requirements 11](#_Toc179472673)

[**2.3 Software Product Features** 11](#_Toc179472674)

[2.3.1 Admin Features 11](#_Toc179472675)

[2.3.2 User Features 11](#_Toc179472676)

[2.3.3 Machine Learning Features 11](#_Toc179472677)

[**2.4 Software System Attributes** 12](#_Toc179472678)

[2.4.1 Reliability 12](#_Toc179472679)

[2.4.2 Availability 12](#_Toc179472680)

[2.4.3 Security 12](#_Toc179472681)

[**2.5 List of Use Cases:** 12](#_Toc179472682)

[**2.6 Use Case Diagram:** 13](#_Toc179472683)

[**2.7 Use Case Details:** 14](#_Toc179472684)

[1. Report Issue 14](#_Toc179472685)

[2. Track Status 14](#_Toc179472686)

[3. Process Issue 15](#_Toc179472687)

[4. Escalate Issue 15](#_Toc179472688)

[5. Sign Up 16](#_Toc179472689)

[6. Login 16](#_Toc179472690)

[**2.8 System Sequence Diagram:** 17](#_Toc179472691)

[2.8.1 UC1: Report Issue 17](#_Toc179472692)

[2.8.2 UC2: Track Status 18](#_Toc179472693)

[2.8.3 UC3: Process Issue 18](#_Toc179472694)

[2.8.4 UC4: Escalate issue 19](#_Toc179472695)

[2.8.5 UC5: Sign Up 20](#_Toc179472696)

[2.8.6 UC6: Login 21](#_Toc179472697)

[**2.9 Sequence Diagram:** 21](#_Toc179472698)

[2.9.1 UC1: Report issue 22](#_Toc179472699)

[2.9.2 UC2: Track Status 22](#_Toc179472700)

[2.9.3 UC3: Process issue 23](#_Toc179472701)

[2.9.4 UC4: Escalate issue 23](#_Toc179472702)

[2.9.5 UC5: Sign Up 24](#_Toc179472703)

[2.9.6 UC6: Login 24](#_Toc179472704)

[**Chapter 3** 25](#_Toc179472705)

[**Software Design Description** 25](#_Toc179472706)

[**3.1 Introduction** 25](#_Toc179472707)

[**3.2 System Architecture Design** 25](#_Toc179472708)

[**3.3 Architectural Diagram** 26](#_Toc179472709)

[**3.4 Domain Model** 27](#_Toc179472710)

[**3.5 Class Diagram:** 27](#_Toc179472711)

[**3.6 Activity Diagram:** 28](#_Toc179472712)

[3.6.1 UC1: Report Issue 29](#_Toc179472713)

[3.6.2 UC2: Track Status 30](#_Toc179472714)

[3.6.3 UC3: Process Issue 31](#_Toc179472715)

[3.6.4 UC4: Escalate Issue 32](#_Toc179472716)

[3.6.5 UC5: Sign Up 33](#_Toc179472717)

[3.6.6 UC6: Login 34](#_Toc179472718)

# **Chapter 1**

# 1.1 Introduction

## 1.1.1 Project Overview:

**Problem Statement:**

Local issues often highlighted in national newspapers are not communicated effectively to relevant departments, causing delays in resolution. Manually tracking these issues can be time-consuming and inefficient.

**Proposed Solution:**

Our solution is a web-based system that automatically extracts and classifies issues from newspaper articles and assigns them to appropriate departments based on the type of issue. The system also tracks the resolution process and allows users to submit new issues. Departments have focal persons responsible for specific issues, and a super focal person oversees all issues across departments.

**Scope of Work:**

The project involves the creation of a web application capable of extracting issues from online newspaper articles, classifying them, assigning them to departments, tracking their progress, and allowing user submissions. It will support the roles of focal persons and super focal persons who manage issue handling and resolution.

# **1.2 Project Organization**

## 1.2.1 Software Process Model:

We are using the **Agile software development methodology**, which allows iterative development and continuous improvement of the system.

## 1.2.2 Roles and Responsibilities:

* **Focal Person**: Manages issues within their department.
* **Super Focal Person**: Oversees all departments and monitors overall issue management.
* **System**: Automatically extracts, classifies, and assigns issues.
* **Newspaper Source**: Provides the data (articles) from which issues are extracted.

## 1.2.3 Tools and Technology:

* **Backend**: Python
* **Frontend**: HTML, CSS, JavaScript, Bootstrap
* **Framework**: Django or Flask (for the web app), Natural Language Processing (NLP) tools for issue extraction.
* **Database**: MySQL for storing issues and user data.
* **Deployment**: it will be deployed on Heroku or on any cloud platform.

# **1.3 Project Management Plan**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Task** | **Duration** | **Start Date** | **Finish Date** | **Predecessor** | **Resources** |
| 1 | **Chapter 1: Introduction** |  |  |  |  |  |
| 2 | Project Overview (Problem Statement, Solution) | 5 days | 04/03/24 | 08/03/24 |  | Tariq Hussain, MS Word, Project Supervisor |
| 3 | Scope of Work | 5 days | 09/03/24 | 13/03/24 | 2 | Tariq Hussain, MS Word |
| 4 | Roles and Responsibilities | 5 days | 14/03/24 | 18/03/24 | 3 | Tariq Hussain, MS Word |
| 5 | **Chapter 2: Software Requirements Specifications (SRS)** |  |  |  |  |  |
| 6 | Functional and Non-Functional Requirements | 5 days | 19/03/24 | 23/03/24 | 4 | Tariq Hussain, MS Word |
| 7 | Product Features (User Features, Admin Features, Machine Learning Features) | 6 days | 24/03/24 | 29/03/24 | 6 | Tariq Hussain, MS Word |
| 8 | List of Use Cases and Use Case Diagram | 6 days | 30/03/24 | 04/04/24 | 7 | Tariq Hussain, Draw.io, MS Word |
| 9 | **Chapter 3: Software Design Description** |  |  |  |  |  |
| 10 | System Architecture Design (MVC) | 8 days | 05/04/24 | 12/04/24 | 8 | Tariq Hussain, Draw.io, MS Word |
| 11 | Domain Model Diagram | 6 days | 13/04/24 | 18/04/24 | 10 | Tariq Hussain, Draw.io, MS Word |
| 12 | Class Diagram | 6 days | 19/04/24 | 24/04/24 | 11 | Tariq Hussain, Draw.io, MS Word |
| 13 | Activity Diagram (For each use case: Report Issue, Track Status, Process Issue, Escalate Issue, Sign Up, Login) | 10 days | 25/04/24 | 04/05/24 | 12 | Tariq Hussain, Draw.io, MS Word |
| 14 | Sequence Diagrams (For each use case) | 8 days | 05/05/24 | 12/05/24 | 13 | Tariq Hussain, Draw.io, MS Word |
| 15 | **Chapter 4: Software Implementation (Documentation)** |  |  |  |  |  |
| 16 | Implementation Overview (Tech stack, core components, etc.) | 6 days | 13/05/24 | 18/05/24 | 14 | Tariq Hussain, MS Word, Documentation |
| 17 | **Chapter 5: Software Testing** |  |  |  |  |  |
| 18 | Test Plan (Testing Tools, Environment, Test Cases) | 8 days | 19/05/24 | 26/05/24 | 16 | Tariq Hussain, MS Word, Supervisor |
| 19 | Test Cases (UC1 to UC6) | 6 days | 27/05/24 | 01/06/24 | 18 | Tariq Hussain, MS Word, Supervisor |
| 20 | **Chapter 6: Future Work and Conclusion** |  |  |  |  |  |
| 21 | Write Future Work and Conclusion Chapter | 6 days | 02/06/24 | 07/06/24 | 19 | Tariq Hussain, MS Word |

**Table 1.3.1: Project Management Plan Table**

# **Chapter 2**

**Software Requirements Specifications**

# **2.1 Introduction**

This chapter describes the functional and non-functional requirements of the Local Issues Classification and Resolution System. It includes the system’s user interface, communication protocols, and system behavior.

## 2.1.1 Product Overview:

The system is designed to automate the extraction and classification of local issues from national newspapers, facilitate communication with departments, and track issue resolutions.

**Stakeholders:**

* **Focal Persons**: Manage and resolve issues assigned to their department.
* **Super Focal Person**: Oversees the resolution of issues across all departments.
* **System**: Performs issue extraction, classification, assignment, and tracking.
* **User**: Submits issues and tracks the status of their submitted issues.

**Major Functions:**

1. **Issue Extraction**: Automatically extract local issues from newspaper content.
2. **Issue Classification**: Categorize issues based on their type (e.g., infrastructure, healthcare, education).
3. **Assign Issues**: Automatically assign issues to the relevant department.
4. **Track Issue Resolution**: Monitor the resolution progress of issues.
5. **Submit Issue**: Allow users to submit issues manually for classification and assignment.

**Major Inputs and Outputs:**

* **Inputs**: Newspaper articles, user-submitted issues.
* **Outputs**: Classified issues, status updates, email sent to stakeholders.

# **2.2 Specific Requirements**

## 2.2.1 External Interface Requirements

* **User Interface**: Simple and intuitive interfaces designed using Bootstrap for responsiveness.
* **Software Interface**: Accessible through any modern web browser with minimal hardware requirements.
* **Communication Protocols**: HTTP/HTTPS for secure communication between users and the server.

# **2.3 Software Product Features**

## 2.3.1 Admin Features

* **User Management**: The admin (super focal person) can manage focal persons and assign roles.
* **System Configuration**: Configure extraction and classification parameters.
* **Issue Management**: The Focal Person and System can track all issues and escalate them if necessary.

## 2.3.2 User Features

* **Submit Issues**: Users can submit issues through a simple form.
* **Track Status**: Users can track the status of issues using a unique issue ID.
* **Feedback**: Users receive updates on issue status

## 2.3.3 Machine Learning Features

* **Data Extraction:** Extract articles from newspapers using BeautifulSoup.
* **Summary Generation:** Use pre-trained models like BART or T5 to summarize the extracted articles.
* **Embedding:** Convert textual data into numerical representations using pre-trained models like BERT or Word2Vec.
* **Optimal k Determination:** Use methods like the Elbow Method or a fixed number of clusters.
* **Clustering:** Implement K-Means clustering for categorizing the issues.
* **Frequent Word Extraction:** Extract 6-7 frequent words from each cluster using techniques like TF-IDF.
* **Text Preprocessing:** Process text using techniques like lemmatization with spaCy or NLTK before embedding and clustering.

# **2.4 Software System Attributes**

## 2.4.1 Reliability

The system must accurately extract and classify issues with high reliability to reduce false positives or negatives.

## 2.4.2 Availability

The system should be available 24/7 with minimal downtime for maintenance and updates.

## 2.4.3 Security

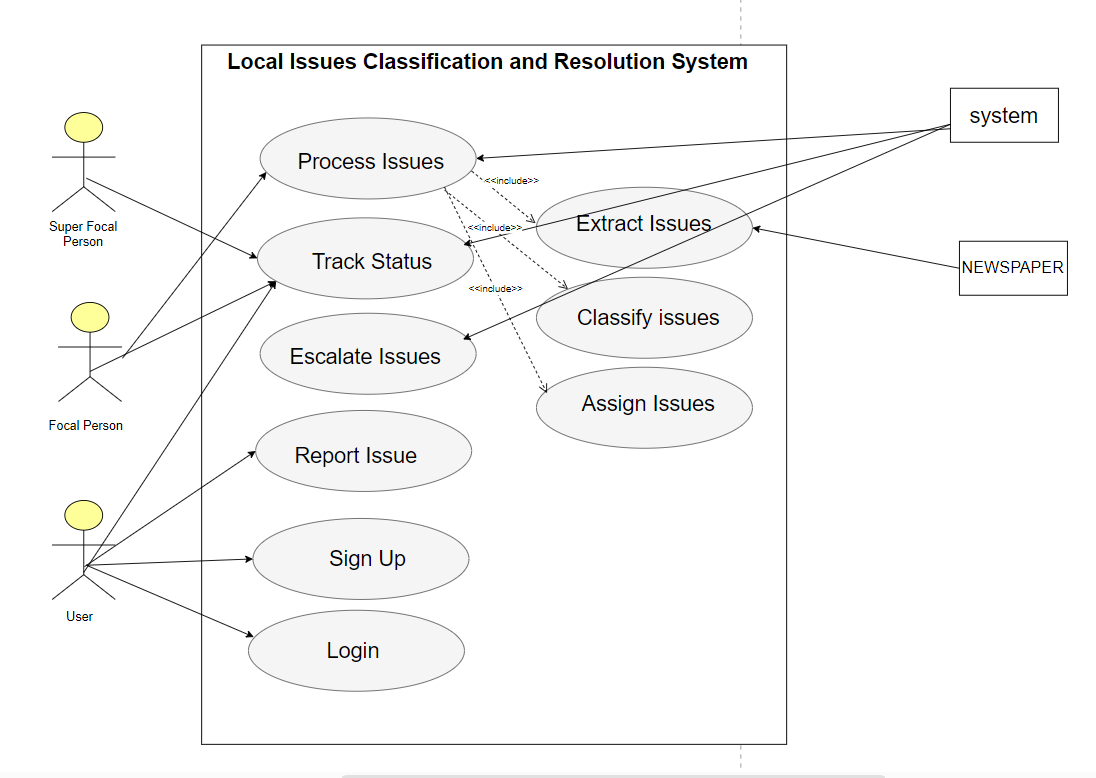
The system should employ standard security practices such as encryption to protect user data and system logs.

# **2.5 List of Use Cases:**

1. **Process Issues**: Extract, classify, and assign issues.
2. **Track Status**: Allow users and the system to track the status of issues.
3. **Escalate Issues**: Allow super focal persons to escalate unresolved issues.
4. **Submit Issue**: Users submit issues manually through the system.
5. **Sign Up:** The user must first sign up to create an account to access the system.
6. **Sign In:** The user needs to sign in with their credentials to access the system's features

# **2.6 Use Case Diagram:**

Figure 2.6.1 shows the interactions between users (actors) and the system, highlighting system functionalities and user roles in a concise, visual format.



**Figure 2.6.1: Use Case Diagram**

# **2.7 Use Case Details:**

Below the details of all the use cases are given.

## Report Issue

|  |  |
| --- | --- |
| **Name** | **UC1: Report Issue** |
| **Primary Actor** | User |
| **Pre-Conditions** | The user is logged into the system. |
| **Post-Conditions** | The issue will be submitted successfully and assigned a unique ID for tracking. |
| **Main Scenario** | 1. User selects the "Report Issue" option.  2. The system displays a form to describe the issue.  3. The user writes the issue details.  4. The system validates the input.  5. The system submits the issue and generates a unique issue ID. |
| **Alternative Scenario** | 1. The system fails during submission.  a) The system rolls back changes and prompts the user to try again.  2. Required details are missing.  a) The system asks the user to provide complete information. |

**Table 2.7.1: Report Issue**

## Track Status

|  |  |
| --- | --- |
| **Name** | **UC2: Track Status** |
| **Primary Actor** | User |
| **Pre-Conditions** | User has submitted an issue and has the unique ID. |
| **Post-Conditions** | The user can view the current status of the issue. |
| **Main Scenario** | 1. User selects the "Track Status" option.  2. The system prompts the user to enter the unique issue ID.  3. User enters the ID.  4. The system retrieves and displays the issue status (e.g., in progress, resolved). |
| **Alternative Scenario** | 1. Invalid or missing ID.  a) The system prompts the user to provide a valid issue ID.  2. System failure.  a) The system notifies the user and requests to try again later. |

**Table 2.7.2: Track Status**

## Process Issue

|  |  |
| --- | --- |
| **Name** | **UC3: Process Issues** |
| **Primary Actor** | System |
| **Secondary Actor** | Newspaper, Focal Person |
| **Pre-Conditions** | New issues have been extracted from newspapers or submitted by users. |
| **Post-Conditions** | The issues are classified and assigned to focal persons or departments. |
| **Main Scenario** | 1. The system extracts issues from newspapers and user submissions. 2. The system classifies the issues based on type and priority. 3. The system assigns the issues to the relevant focal persons. |
| **Alternative Scenario** | 1. Classification failure. a) The system retries classification or flags the issue for manual review. |

**Table 2.7.3: Process Issue**

## Escalate Issue

|  |  |
| --- | --- |
| **Name** | **UC4: Escalate Issues** |
| **Primary Actor** | Super focal Person |
| **Secondary Actor** | System |
| **Pre-Conditions** | Issue is unresolved within a set timeframe. |
| **Post-Conditions** | Issue is escalated to the super focal person. |
| **Main Scenario** | 1. System identifies unresolved issues beyond the set timeframe. 2. System escalates the issue to the super focal person. 3. The super focal person is notified and takes action on the escalated issue |
| **Alternative Scenario** | 1. Escalation request rejected. a) The focal person retries or revises the escalation request. |

**Table 2.7.4: Escalate Issue**

## Sign Up

|  |  |
| --- | --- |
| **Name** | **UC5: Sign Up** |
| **Primary Actor** | User |
| **Pre-Conditions** | The web home page is opened. |
| **Post-Conditions** | User is registered successfully. |
| **Main Scenario** | 1. User selects the "Sign Up" option. 2. The system prompts the user to enter required details (e.g., name, email, password). 3. User submits the registration form. 4. The system validates the details and creates the user account. 5. User is registered successfully, and a confirmation message is displayed. |
| **Alternative Scenario** | 1. The system detects incomplete or incorrect details. a) The system prompts the user to correct or complete the information. 2. System failure. a) The system rolls back changes and notifies the user of the failure. |

**Table 2.7.5: Sign Up**

## Login

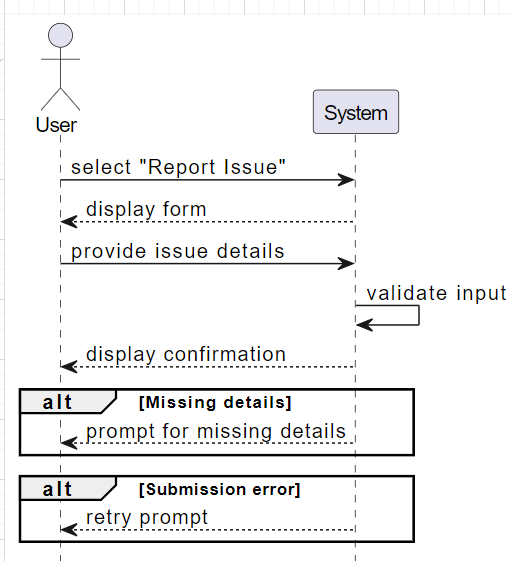
|  |  |
| --- | --- |
| **Name** | **UC6: Login** |
| **Primary Actor** | User |
| **Pre-Conditions** | The user is registered and on the login page. |
| **Post-Conditions** | User is successfully logged in to the system. |
| **Main Scenario** | 1. User selects the "Login" option.  2. The system prompts the user to enter a valid username and password.   1. User submits login credentials. 2. The system validates the credentials. 3. User is logged in successfully, and the dashboard is displayed |
| **Alternative Scenario** | 1. Invalid username or password.  a) The system prompts the user to enter correct credentials.   1. System failure. a) The system notifies the user of the failure and suggests retrying later. |

**Table 2.7.6: Sign In**

# **2.8 System Sequence Diagram:**

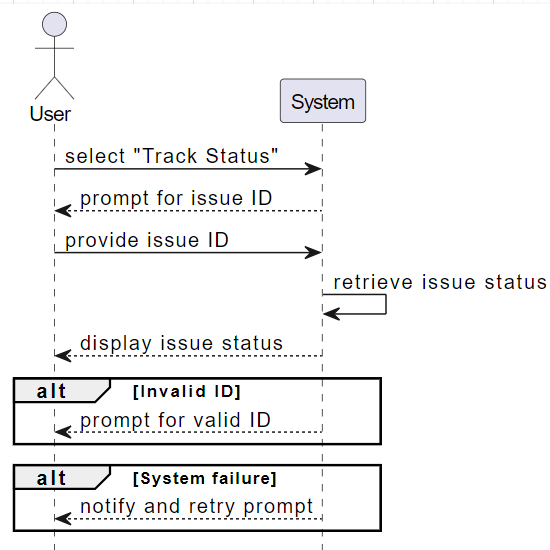
A system sequence diagram illustrates the interactions between external actors and the system, showing the sequence of messages exchanged to achieve specific functionality.

## 2.8.1 UC1: Report Issue



**Figure 2.8.1: Report Issue**

## 2.8.2 UC2: Track Status



**Figure 2.8.2: Track Status**

## 2.8.3 UC3: Process Issue

A diagram of a system

Description automatically generated

**Figure 2.8.3: Process Issue**

## 2.8.4 UC4: Escalate issue

A screenshot of a computer screen

Description automatically generated

**Figure 2.8.4: Escalate issue**

## 2.8.5 UC5: Sign Up

A diagram of a system

Description automatically generated

**Figure 2.8.5: Sign Up**

## 2.8.6 UC6: Login

A diagram of a system

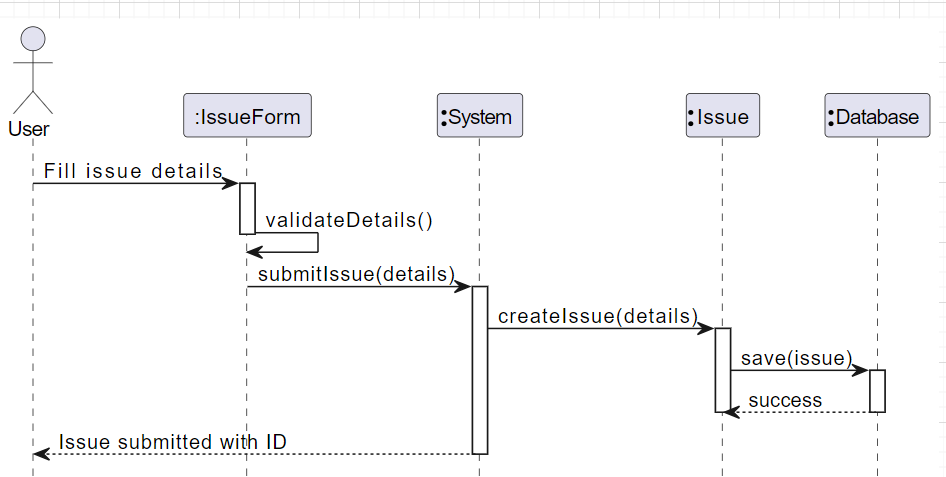
Description automatically generated

**Figure 2.8.6: Sign In**

# **2.9 Sequence Diagram:**

A sequence diagram shows the flow of interactions between objects or actors over time, detailing the order of messages exchanged to achieve a specific process within the system.

## 2.9.1 UC1: Report issue



**Figure 2.9.1: Report Issue**

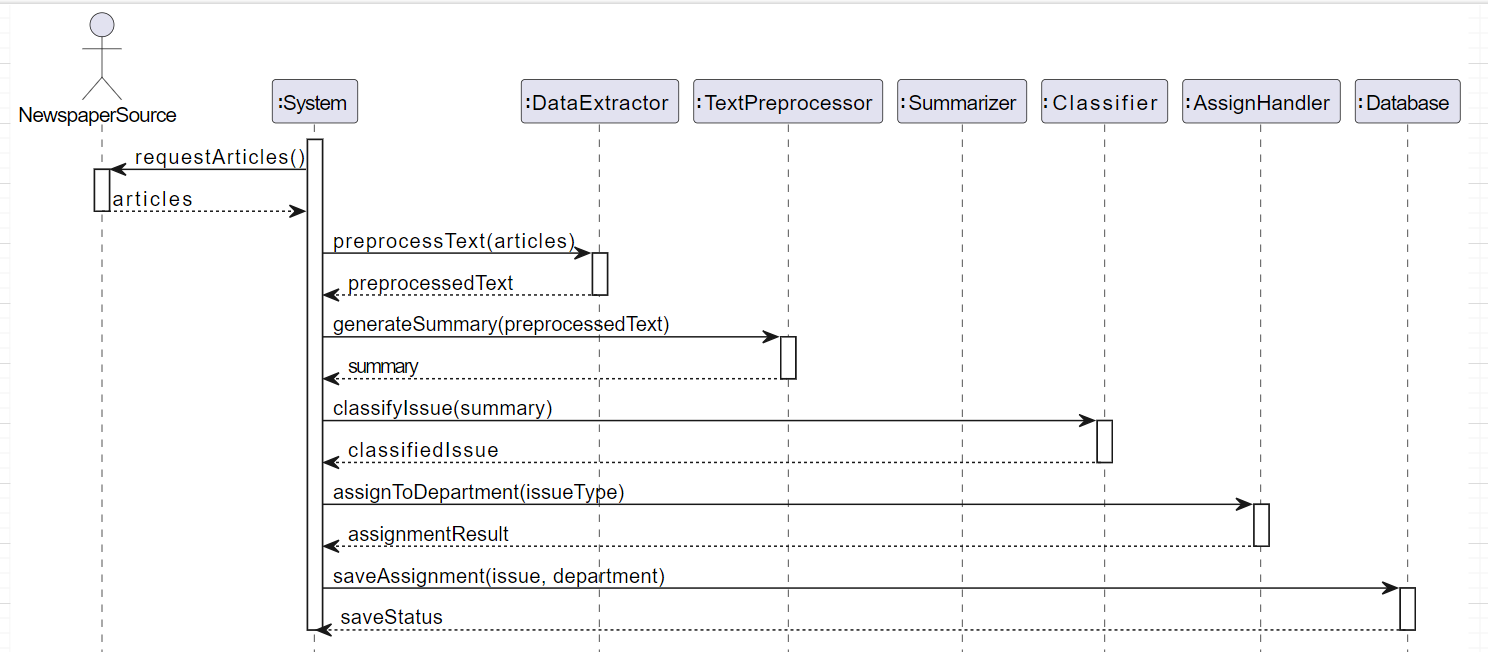
## 2.9.2 UC2: Track Status

A diagram of a system

Description automatically generated

**Figure 2.9.2: Track Status**

## 2.9.3 UC3: Process issue



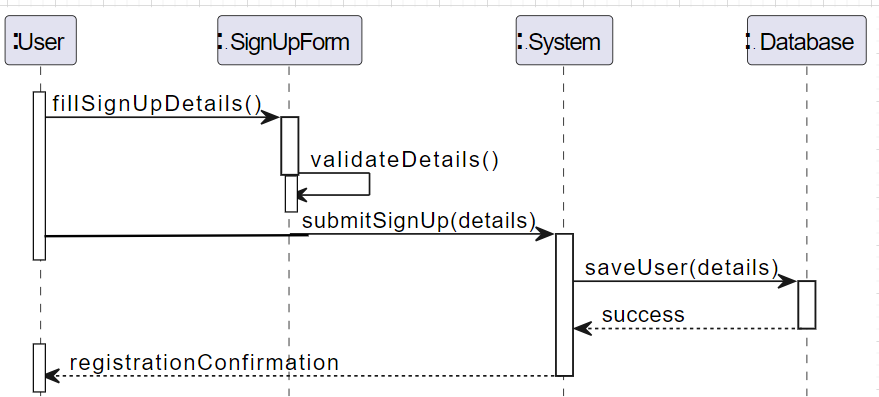
**Figure 2.9.3: Process issue**

## 2.9.4 UC4: Escalate issue

A screenshot of a computer

Description automatically generateds  
**Figure 2.9.2: Escalate issue**

## 2.9.5 UC5: Sign Up



**Figure 2.9.5: Sign Up**

## 2.9.6 UC6: Login

A diagram of a system

Description automatically generated

**Figure 2.9.6: Login**

# **Chapter 3**

# **Software Design Description**

# **3.1 Introduction**

This chapter discusses the design of the Local Issues Classification and Resolution System. It covers the system's architecture design, how it functions, and alternative design approaches. Additionally, it describes the system's interface, providing architectural diagrams, domain models, and activity diagrams to clarify user interactions.

**Design Overview**

System design defines the architecture, components, modules, interfaces, and data to meet system requirements. It applies systems theory to develop the product.

**Requirements Traceability Matrix**

The Requirements Traceability Matrix (RTM) ensures all system requirements are tested and validated throughout the development process.

# **3.2 System Architecture Design**

The architectural design of the system focuses on its structure, behavior, and how different views of the system are represented.

**Chosen System Architecture: Model-View-Controller (MVC)**

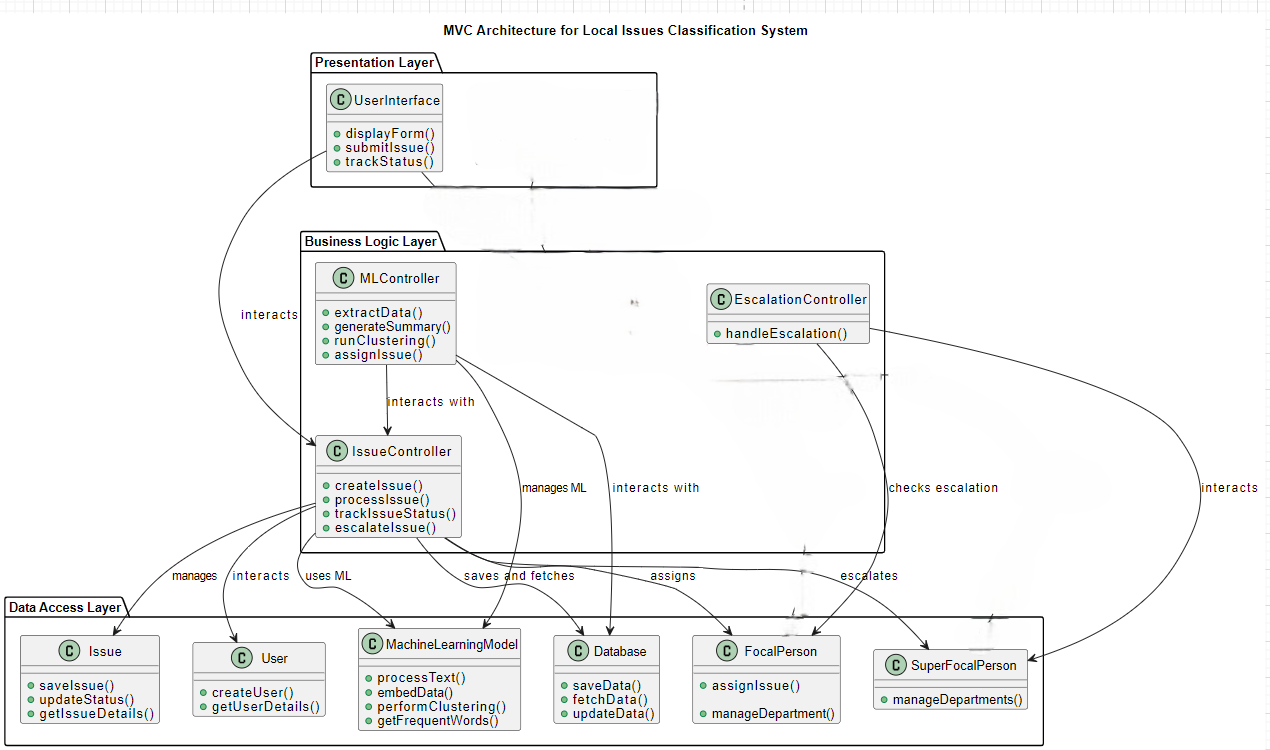
* **Model**: Maintains the data such as issues and user information.
* **View**: Displays data to users, such as issue status and submission forms.
* **Controller**: Handles the interaction between the model and the view, processing user input and updating the model accordingly.

**Discussion of Alternative Designs**

* **Model View Presenter (MVP)**: MVP pushes all presentation logic to the presenter, making the controller more complex.
* **Model View ViewModel (MVVM)**: This approach abstracts the view’s state, but MVC is more suited for the web-based nature of this project.

# **3.3 Architectural Diagram**

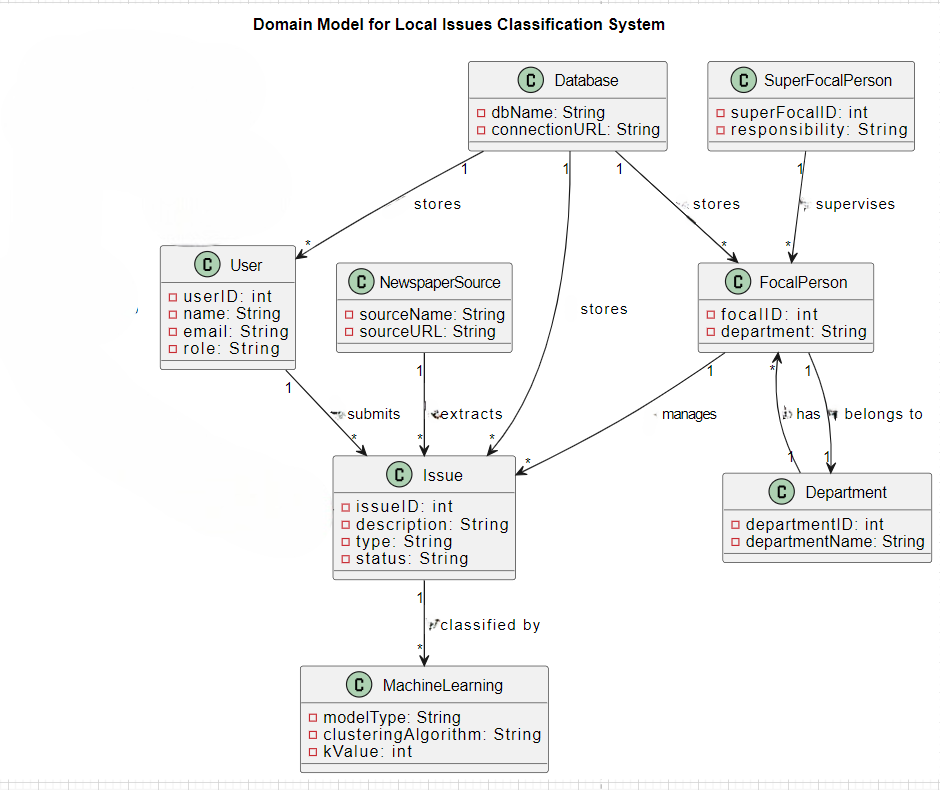
The figure 3.3.1 provides an overview of the system’s structure, including modules for issue extraction, classification, clustering, and tracking. These modules work together to extract data, classify it using machine learning, assign issues to relevant departments, and track their resolution.



**Figure 3.3.1: MVC Diagram**

# **3.4 Domain Model**

A figure 3.4.1 represents the key concepts, objects, and relationships within the system, offering a high-level view of the system’s structure and how different entities interact.



**Figure 3.4.1: Domain Model**

# **3.5 Class Diagram:**

The figure 3.5.1 defines the system’s structure by depicting classes, their attributes, methods, and the relationships between them, showing how they interact within the system.

A computer screen shot of a diagram

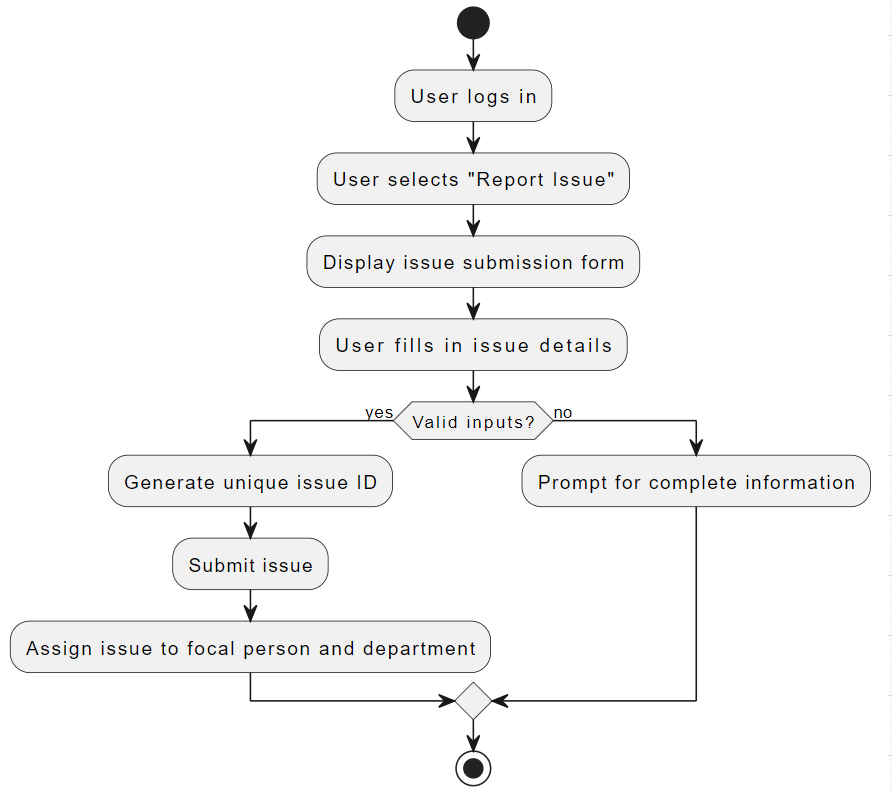
Description automatically generated

**Figure 3.5.1: Class Diagram**

# **3.6 Activity Diagram:**

An activity diagram visually maps out the flow of control and activities in a process, illustrating the sequence of actions and decision points required to complete a specific task.

## 3.6.1 UC1: Report Issue



**Figure 3.6.1: Report Issue**

## 3.6.2 UC2: Track Status

A diagram of a system

Description automatically generated

**Figure 3.6.2: Track Status**

## 3.6.3 UC3: Process Issue

A diagram of a system

Description automatically generated

**Figure 3.6.3: Process Issue**

## 3.6.4 UC4: Escalate Issue

A diagram of a system

Description automatically generated

**Figure 3.6.4: Escalate Issue**

## 3.6.5 UC5: Sign Up

A diagram of a system

Description automatically generated

**Figure 3.6.5: Sign Up**

## 3.6.6 UC6: Login

A diagram of a user login

Description automatically generated

**Figure 3.6.6: Login**