



**POLITECNICO**  
**MILANO 1863**

# REQUIREMENTS ANALYSIS AND SPECIFICATION DOCUMENT

***SE4G – SOFTWARE ENGINEERING FOR GEOINFORMATICS***  
*AMIEVA, AUSTONI, SONDOQAH*

## Content

Overview	2
Project Goals and Scope	2
Goals	2
Scope	3
Domain Analysis	4
Domain Entities	4
Domain analysis table	4
Use Cases	9
Regular Cases	9
Actors	9
Use cases concerning users	9
Member registration	9
Citizen user registration:	9
Specialized user registration:	10
Member login	11
Submitting a maintenance request	11
Updating of maintenance requests	11
Citizens personal requests visualization	12
Citizen visualization of updated requests	12
Filtering and visualization of graphs (Dashboard)	12
View Map	13
Exceptional Use Case	13
Requirements	14
Technical Requirements	14
Functional Requirements	14
Domain assumptions	14

## Overview

This document has as a main objective give a technical explanation about the requirements and specifications detected by the team in terms of the development of a new web APP “RODI -Milano Road Network Monitor”.

The mentioned app will detect anomalies in the state of the urban roads of Milano by giving the opportunity to the citizens to contribute when any of them detects a problem in the urban network.

At the same time the authorities in charge of the maintenance of the road network will acquire those anomalies reported and process the priorities of intervention of them. Finally, the platform will provide feedback to the citizens about the status of their requests and will show analysis performed with respect to this information to the “specialized users” in order to facilitate the detection of the main problems along the road network.

## Project Goals and Scope

### Goals

The project aims to develop a web app which could attend the following issues of the urban road network of Milano:

- Improve the level of communication between authorities in charge of road maintenance and citizens
  - Citizens may upload requests of maintenance once they detect an anomaly in the urban road
  - Authorities (Specialized users) should answer to citizen users about the status of the requests uploaded by each one of them
- Measure and display information about the Key Performance Indicators [KPIs] of the distresses<sup>1</sup> requests of maintenance on the urban road network of Milano
  - Clusters of requests: These clusters should organize the requests in groups depending the size of the distresses and the level of risk perceived by the users who upload those requests of maintenance.
  - Requests upload by date

---

<sup>1</sup> **Clarification:** The distresses are the anomalies detected by the users in the road network (potholes, block crack, longitudinal)

- Requests per user
- Quantity of distresses updated depending the material
- Quantity of distresses updated depending the size of the distress
- Quantity of distresses updated depending the kind of the distress
- Quantity of distresses updated depending the level of risk perceived by the users
- Generate a collaborative and public map of distresses in the urban road network of Milano.

## Scope

The application will be a “collaborative” management system which receives as inputs the requests of maintenance of citizens and transmits them to the authorities (who have a special user role in the platform: *Specialized users*).

This system will also show and monitor KPIs about the requests received by the citizen users to summarize the information for the specialized users.

Finally, the web app will offer a map to visualize all the distresses uploaded by citizen users in Milano and will show a layer of the population in the different sectors of Milano to provide contextual information which allows, to everyone who inspect that map (It is not restricted only for users of the platform), the possibility to understand the *approximate* demand that would suffer the road network in the different areas. The demand of the road network would be an important insight in terms of the road users affected by each distress.

### **Important Clarification:**

The information displayed in the web map and in the graphs presented in the dashboard section of the application should not be updated in real time. It is because they will be based on html templates without a dynamically rendering updating.

***Although the rendering is not updated dynamically, the information in which it is based is dynamically updated in the backend.***

It is considered that a future requirement for the application should consider in its scope a dynamic rendering using as base the retrieving information configuration implemented in this stage of development.

# Domain Analysis

## Domain Entities

- Users
  - Citizens
  - Specialized: Local Authorities (in charge of road maintenance)
- Street's anomalies database

## Domain analysis table

*Table 1: Domain Analysis*

Phenomena	Location	Controlled by
Data is collected	Shared	World
The user opens the web explorer	World	World
The user searches in the web explorer the url of the web platform	Shared	World
A citizen user goes to “Register” section to make his/her registration in the system	Shared	World
The user registers in the app by completing username, password and e-mail in a displayed form	Shared	World
If the username is shorter than 5 digits, an error message is displayed to the user	Shared	Machine
If the password is shorter than 5 digits, an error message is displayed to the user	Shared	Machine
If the mail does not contain “@” and “.” characters, an error message is displayed to the user	Shared	Machine
Passwords are encrypted by the system before being stored	Machine	Machine

User's credentials are registered in the Database server of the app	Machine	Machine
The specialized users are required to complete their registration by filling the related form in the section “Specialized user request”	Shared	World
The system informs the official moderator ( <b>Super user</b> of the platform, called “JAM”) by Email about the request of a new specialized user	Shared	Machine
The super user opens the email and analyzes the request of a new authority user.	World	World
If the super user considers that the request is correct (previous phenomenon), then he/she logs in with his/her credentials in the platform and select the option “Special Registration”	Shared	World
Finally, the super user completes the registration form with the data received by mail about the new specialized user request	Shared	World
The system queries the DB to be updated	Machine	Machine
Citizen Users sign in	Shared	World
Username and password are compared to already existing one in the app database	Machine	Machine
If login fails, an error message is displayed to the user	Shared	Machine
The citizen user selects the desired section in the web page depending on what he/she wants to do. Possible options:	Shared	World

1-Visualize his/her requests, 2-Visualize the map of request of Milano, 3- Upload a new request		
The system redirects to the right section of the web page	Machine	Machine
The citizen detects a distress in the urban road network	World	World
The system provides a link to the Epicollect5 project to the citizen	Shared	Machine
The system redirects the request to the Epicollect5 project form	Machine	Machine
The citizen completes in Epicollect5's forms the details about the distress detected	Shared	World
The request is included in the ep5 project	Machine	Machine
The platform gets the information by a get request from the ep5 project and stores it in the db	Machine	Machine
The citizen would visualize his/her new request if he/she goes to "My requests" section	Shared	World
For the specialized users, the system displays the all the uploaded requests in the section "Status".	Shared	Machine
The specialized users visualize the requests in "Status section" and plan the intervention in the real world.	Shared	World

Considering all the maintenance requests with status “On Going”, the specialized users analyze the information of each one, prioritize the required interventions, and plan the date to perform them	World	World
After the intervention, the distress is solved in the road network of Milano	World	World
Once the intervention of a maintenance request is solved, the specialized users should change the status of it from “On Going” to “Complete” by clicking the button “CHANGE” in “Status” section	Shared	World
The change in the request status is updated in the db	Machine	Machine
The citizen user access to the section “My requests” to control the status of his/her requests	Shared	World
The status of the correspondent request that had been modified by the specialized user will appear updated with the status “Complete”	Machine	Machine
A logged in specialized user selects the section “Dashboard” in the web page	Shared	World
A get request is performed by the app to ep5 project	Machine	Machine
The retrieved information is updated in the db of the app	Machine	Machine



The system queries the db and performs graphs to summarize information about different KPI's	Machine	Machine
The system renders to the <b>specialized</b> user the "Dashboard" section with the correspondent graphs contained in it	Machine	Machine
A user ( <b>not necessarily register in the app</b> ) goes to the section "Map"	Shared	World
The app performs a get request to ep5 project to obtain the information of the requests uploaded in the related project (MRNM project)	Machine	Machine
The requests are rendered and displayed in a web map which identifies the kind of distress related with each request of maintenance, and It also shows a layer with the population of each area of Milano	Machine	Machine
The user analyzes the information and would interact with the related information displayed in the screen:	Shared	World

*Activating or disactivating layers, augmenting the zoom or reducing it, pressing in the requests to understand the kind of distress of each one of them*

# Use Cases

## Regular Cases

### Actors

1. Non-registered users: Users who can use the website without having credentials to sign in, but **cannot see the status of their request made in Ep5**. They can see the map of distresses in Milano road urban network.
2. Registered users:
  - 2.1. *Citizen Users*: Users who can upload requests of maintenance in the ep5 project and access to the section “My requests” to check the status of the requests uploaded by themselves. They also could see the map with the distresses of Milano road urban network.
  - 2.2. *Specialized Users*: They have access to more sections: Can visualize summarized data in the dashboard section, update the status of the requests uploaded previously in ep5 project by citizen users and also have access to the map section.
  - 2.3. *Super User*: This is the moderator of the application which can register specialized users and he/she is who has access to the backend of the system to implement “Hard” modifications on the information in case they would be required.

## Use cases concerning users

### Member registration

#### *Citizen user registration:*

- Use case name: RegisterCitizenMember
- Actors: Users who want to be registered as a citizen user.
- Entry Condition: User opens the web page.
- Flow:
  - The user selects the “Register” option.
  - A window pops-up asking the user about the his/her username, his email, and his password.
  - The user fills the form and confirms the process.

- In case the user does not respect the format required for each of the required fields, the application will display a popup indicating the wrong input (example: mail without “@” or “.” characters).
- The registered user password is encrypted before being stored in the software’s database.
- The user credentials are stored in the database.
- Exit Condition: The page redirects the user to the log in page.

### *Specialized user registration:*

- Use case name: RegisterSpecializedMember
- Actors: Specialized users who work in the Ministry of Infrastructure and Transport of Milano and are responsible to plan the interventions of maintenance on the road network.
- Entry Condition: User opens the web page.
- Flow:
  - The user selects the “Specialized user request” option.
  - A window pops-up asking the user about the following fields: Name, Surname, Role, Work Department, Phone Number
  - The user fills the form and confirms the process.
  - A screen is displayed to the user with a message “Your request has been sent successfully!”.
  - Then the Super User receives a mail with the information filled by the person who wants to register as “specialized user”.
  - If the Super User considers that the registration is required because the request is correct, he will log in in the platform and will choose the option “Special registration”
  - Once in the “Special registration” section, the super user should complete the following fields in the displayed form (Information which belongs to the person who request for the registration): Username, Password and Mail.
  - After the registration is confirmed, the Super User is redirected to the homepage.
  - The registered user password is encrypted before being stored in the software’s database.
- Exit Condition: The software stores the input information in the database.

## Member login

- Use case name: LoginMember
- Actors: Registered members (Citizen, Specialized, Super Users).
- Entry Condition: A member opens the web page.
- Flow:
  - The Member selects the “login” option.
  - The software opens a window asking the Member for username and password.
  - The Member submits his/her username and password
  - The software checks if the member’s credentials are stored in the database.
  - If the credentials are right, the software grants the Member access to the properties of the web page according the kind of user.
  - The software returns an error message if the user provides wrong credentials.
- Exit Condition: The member is successfully logged in.

## Submitting a maintenance request

- Use case: UploadingMaintenance
- Actors: Citizen Members
- Entry condition: User open the related section on web page
- Flow:
  - The user chooses the section of the web page where is displayed the link to epicollect (Logo of Epicollect)
  - The user completes the entry using Epicollect5 web app.
- Exit condition: The entry is stored in the ep5 project

## Updating of maintenance requests

- Use case: RequestStatusModification
- Actors: Specialized users
- Entry condition: A specialized user selects goes to “Status” section and press on “Change” button to modify as “Complete” the status of a request of maintenance.
- Flow:
  - A specialized user selects a request which status is “On Going” after planning its resolution in the real world and press the button “Change”.
  - The system updates the status of the related request as “Complete”

- Exit condition: The system display the requests of maintenance with the status “Complete” in the “Status” section.

#### Citizens personal requests visualization

- Use case: MyRequestList
- Actors: Citizen users
- Entry condition: A citizen user access to the “My Requests” section
- Flow:
  - A citizen user access to “My Requests” section and check the requests displayed
  - The page renders the correspondent requests related with the user
- Exit condition: The citizen visualize the information about the requests that has uploaded previously.

#### Citizen visualization of updated requests

- Use case: VisualizationUpdatedRequests
- Actors: Citizen users
- Entry condition: A request is updated by a specialized user
- Flow:
  - A specialized user updates the status of a request
  - The citizen user who had uploaded the related request access can visualize in the section “My requests” the status updated as “Complete”.
- Exit condition: The citizen visualize the information updated and understand the resolution of the related request.

#### Filtering and visualization of graphs (Dashboard)

- Use case name: VisualizeData
- Actors: Specialized users.
- Entry Condition: A specialized user access to the “Dashboard” section.
- Flow:
  - The user chooses the Dashboard section from the main page of the website.
  - The page displays a summary of the main KPI that summarize the most important information about the distresses on the road network updated by users.
  - The user could filter the data by interacting over the graphs.
- Exit Condition: The software modify the visualization according to the filters applied by the user.

## View Map

- Use case name: ViewMap
- Actors: Members and unregistered users.
- Entry condition: A user selects the “Map” section in the web app.
- Flow:
  - The user selects the section “Map” of the web app
  - The app renders a map with two layers of geographical information:
    - Distresses of Milano
    - Population of the different areas of Milano (proxy of the demand over the road network)
  - The user could choose the base map and the layers to visualize by interaction with the web map
  - The web app renders the layers and the base map chosen by the user
- Exit Condition: The user visualize the rendered map and understand the information displayed on it.

## Exceptional Use Case

- Use case name: RequestMaintenanceModification
- Actors: Super User.
- Entry Condition: Specialized users detects an anomaly in the status of a maintenance request.
- Flow:
  - The specialized user informs the admin about the anomaly in the status of a maintenance request (example: Mistake in marking as complete a request).
  - The specialized member informs details about the anomaly detected to the super user.
  - The super user checks the data received by the specialized user.
  - The super user applies the modifications required in the DB of the web app in case they are correct.
- Exit Condition: The super user informs about the modification once it is done.

# Requirements

## Technical Requirements

- The system should be developed in python.
- The system should be used on the web.
- The system shall be created using **MRNM** Epicollect5 dataset.
- The system should offer customized visualization of the data:
  - The system will use OpenStreetMap as a default base-map and offer other options of basemaps
  - The geographic data will appear as a on the top of the basemap where the data point location appears in a specific style and the polygon layer of population have a scale of colors depending the values on them.

## Functional Requirements

- The system should allow the user to visualize all the on the map. This would be accessible for any kind of user, even for those non-registered in the app.
- The users should be divided into two categories, the first category is for those who use the website for operational use (specialized users), the second one is those who use the system as citizen users for non-operational uses. There is also the particular case of the Super User who is the admin of the platform.
- The system should allow citizen users to register using a username, a password, email address.
- The super user should register the specialized users by completing the related form for that purpose, considering the information received in his/her e-mail address (if he/she considers that the information is correct).
- The system should update the citizen user once the maintenance has been completed and informed by the specialized users.
- The system should update the section “Status” (which contains all the requests) once a new request is added by the citizens.
- The system should update the section “Status” once a request has been marked as “Complete” by a specialized user.

## Domain assumptions

- The collected data (distresses) by users are real entities in the field.

- The web app is monitored and authorized by the Ministry of Infrastructure and Transport of Milano.