POLITECNICO
MILANO 1863

Software Engineering for Geoinformatics

(A.Y. 2024)

Requirement Analysis and Specification Document

Landslide Hazard Exposure

AUPE RiskMonitor
RISK RADAR

Authors:

Mohammad Ammar Mughees

Mohmmad Umayr Romshoo

Sayed Erfan Eshghelahi

Praveenkumar Saminathan

Table of contents	Page No:

List of Tables	Page No

List of Figures	Page No

1. Introduction

1.1 Purpose

The primary objective of our project is to enhance the visualization and analysis of the landslides hazard exposure for specific census groups buildings across different administrative levels in Italy. The primary objective of our project is to enhance the visualization and analysis of landslides hazard exposure for specific census groups buildings) across different administrative levels in Italy. While Italian public authorities diligently collect and process hazard-related datasets to generate hazard maps, the effective presentation of this information is crucial for decision-makers in civil protection and insurance sectors. Our project aims to bridge this gap by providing an interactive platform that visualizes and elaborates on the exposure of census groups to flood hazards at specified administrative levels. By leveraging appropriate administrative units, such as national, regional, or provincial levels, our web application will facilitate a comprehensive understanding of hazard exposure, thereby empowering decision-makers to make informed choices regarding risk mitigation and disaster management strategies.

This purpose statement outlines the core objectives of our project, emphasizing the importance of effective data presentation for decision-makers in disaster management and insurance sectors. It highlights the need to visualize hazard exposure for specific census groups across different administrative levels in Italy, ultimately aiming to contribute to improved risk assessment and mitigation efforts.

1.2 Scope

The scope of this project is to develop a comprehensive client-server application that facilitates users in querying and visualizing hazard and risk indicator data retrieved from the Italian Institute for Environmental Protection and Research (ISPRA) IdroGEO API. The focus will be on utilizing data related to PIR (Hazards and Risk Indicators) for analysis and visualization.

The chosen province of Italy for flood hazard analysis will be determined based on the availability and relevance of data provided by the ISPRA IdroGEO API stored in group AUPE's database . The system will provide users with intuitive tools and functionalities to explore and understand hazard and risk indicators, ultimately supporting informed decision-making processes related to disaster management and risk mitigation strategies.

This scope delineates the key components and functionalities of the client-server application, aiming to provide users with a robust platform for querying, processing, and visualizing hazard and risk indicator data sourced from the ISPRA IdroGEO API.

---

1. IFFI Project - Inventory of Landslide Phenomena in Italy
2. IdroGEO API
3. European Space Agency - Prone Landslide areas in Italy

1.3 Short Overview

In a short view, the client-server application will provide users with a range of functionalities to query, process, and visualize hazard and risk indicator data sourced from the ISPRA IdroGEO API. Key features include:

Visualization Tools:

- An interactive map centered on the chosen province, allowing users to visualize hazard and risk indicator data spatially.
- Classical data view presented in a table format, providing users with detailed information about the selected data.

Processing Functionalities:

- Provision of basic statistics through charts related to landslide hazard indicators, allowing users to gain insights into overall landslide risk at different levels with different type within the chosen province.

1.4 Acronyms and other definitions

RASD - Requirements Analysis and Specification Document
JSON - JavaScript Object Notation
REST API - It is an Application Programming Interface that satisfies the constraints of REST architecture and allows interaction with RESTful web services.
Python- High-level programming language we are going to use for building our software. It will be the basis of every functional tool used in our web application.
API - Application Programming Interface

Table 1: Acronyms and Definitions

1. Landslide (who.int)
2. Landslide (who. Italy)

2. Application domain and phenomena description

2.1. Users:

These entities are responsible for assessing and managing natural disaster risks, including landslides.

2.2 Dataset Description

This dataset is a Collaborative Web Mapping Application Based on REST API Services and Open Data on Landslides and Floods in Italy.

The idea works on development of the IdroGEO web platform, which allows navigation, social sharing, and download of data, maps, reports related to the Italian Landslide Inventory (IFFI), national hazard maps, and risk indicators for landslides and floods in Italy. It is a tool for communication, dissemination of information, and to support decision-making in risk mitigation policies, land use planning, infrastructure design, prioritization of mitigation measures, and environmental impact assessment.

The key aspects of this web platform are:

1. Data model and editing workflow for the IFFI landslide inventory.
2. User interface design focusing on usability, accessibility across devices, and interactive mapping tools.
3. System architecture based on open standards, open libraries, RESTful APIs, and a progressive web application approach.
4. Methodology for evaluating the use and usability of the platform.

Table 2: Aspects of Web Platform

And, the challenges faced during the development, such as usability, interoperability, transparency, and improving the updating process of the IFFI inventory. It discusses the solutions adopted, including progressive web applications, APIs, open standards, and a collaborative landslide inventory management system via RESTful APIs.

1. [Welcome to Flask — Flask Documentation (2.2.x) (palletsprojects.com)]
2. [Restful API - Red Hat]

2.3 Phenomena Analysis for Landslide Hazard System

Description:

The following table outlines the key phenomena considered in the design and implementation of the landslide hazard system. Each phenomenon plays a crucial role in assessing, predicting, and mitigating landslide risks. While these phenomena may not be directly visualized on the interactive map interface, they serve as critical inputs and outputs managed within the system architecture.

Phenomena Table:

NO	Phenomenon	Category (W/S/M)	Control
1	Rainfall intensity	World	Sensors (World)
2	Soil moisture content	World	Sensors (World)
3	Slope steepness	World	Sensors (World)
4	Geological structure	World	Sensors (World)
5	Historical landslide data	Shared	Database
6	Vegetation cover	World	Sensors (World)
7	Land use changes	Shared	Database
8	Weather conditions	World	Sensors (World)
9	Groundwater level	World	Sensors (World)
10	Topographic elevation	World	Sensors (World)
11	Satellite imagery	World	Sensors (World)
12	User input (e.g., location)	Shared	User interface
13	Landslide susceptibility models	Machine	Algorithm
14	Emergency response plans	Machine	Algorithm
15	Seismic activity	World	Sensors (World)

Table 3: Phenomenon

Explanation:

- **Phenomenon**: Identifies specific factors or data elements relevant to landslide hazard assessment.
- **Category (W/S/M)**: Classifies each phenomenon into World (physical aspects), Shared (data accessible across the system), or Machine (algorithms and processes).
- **Control**: Specifies which part of the system manages or utilizes each phenomenon.
- **Description**: Provides a detailed overview of each phenomenon's role within the context of landslide hazard assessment, emphasizing its importance in system functionality.

The second column of the table lists various phenomena that are related to weather applications. Here's a brief description of each phenomenon:

- **Rainfall Intensity**: Measurement of rainfall intensity affecting soil saturation and landslide risk.
- **Soil Moisture Content**: Amount of water in soil influencing stability and landslide susceptibility.
- **Slope Steepness**: Degree of terrain inclination impacting landslide potential.
- **Geological Structure**: Composition and stability of geological formations affecting landslide risk.
- **Historical Landslide Data**: Records of past landslides used for risk assessment and planning.
- **Vegetation Cover**: Impact of vegetation on slope stability and water absorption.
- **Land use Changes**: Changes in land use patterns affecting landslide susceptibility.
- **Weather Conditions**: Current atmospheric factors influencing ground stability.
- **Groundwater Level**: Depth of underground water impacting soil moisture and landslide risk.
- **Topographic Elevation**: Height above sea level affecting slope stability.
- **Satellite Imagery**: Visual data of terrain conditions captured from satellites.
- **User Input**: User-provided location information for personalized risk assessment.
- **Landslide Susceptibility Models**: Models predicting landslide risk based on input data.
- **Emergency Response Plans**: Plans or algorithms for emergency response to landslide events.
- **Seismic Activity**: Earthquake occurrence influencing ground stability.

2.4 Domain Assumptions

The following assumptions hold with respect to the intended target users, application's software/hardware environment, data, and boundary conditions:

- Users have a basic understanding of geological hazards, including landslide types, triggers, and mitigation strategies.
- Users are aware of common terms and measurements related to landslide susceptibility and risk assessment.
- Users have consistent access to the internet while using the application.
- Users possess basic familiarity with Geographic Information Systems (GIS) functionalities and interaction schemas relevant to viewing maps, layers, and spatial data.
- The device used by the user has an internet browser software installed to access the application (a stable and updated release of any of the main web browsers).
- External raw data sources are continuously maintained and updated.

3. Use case

3.1 Use Cases:

View Landslide Inventory Information

Actor: all users

U1: View Landslide Inventory Information

Flow of Events:

- The user opens the AUPE platform.
- The user navigates to the Italian Landslide Inventory (IFFI) section.
- The user zooms and pans the map to their area of interest.
- The user hovers over or clicks on a specific landslide feature on the map.
- The platform displays detailed information about the selected landslide, including location, type, date, damage, causes, and any attached multimedia files.

Exit Condition: The user can view comprehensive landslide information for their selected area.

Exception Handling: If no landslides are present in the selected area, the platform will display a message indicating no data is available.

View Choropleth Map

Actor: User

U2: View Choropleth Map

Flow of Events:

- The user opens the Landslide Risk Dashboard.
- The user selects the desired risk type from the dropdown menu (e.g., Landslide Surface Area).
- The user selects the desired risk level from the dropdown menu (e.g., Very High).
- The application updates the choropleth map to visualize the selected risk type and level.

Exit Condition: The user can view the choropleth map depicting landslide risk based on the selected criteria.

Exception Handling: If data fetching fails, display an error message prompting the user to retry or contact support.

Adjust Map Opacity

Actor: User

U3: Adjust Map Opacity

Flow of Events:

- The user interacts with the opacity slider on the dashboard.
- As the user adjusts the slider, the opacity of the choropleth map changes in real-time.

Exit Condition: The user can effectively view underlying layers or details on the map based on the adjusted opacity.

Exception Handling: Ensure the application handles minimum and maximum opacity settings smoothly within the slider range (0 to 1).

Change Base Tile Layer

U4: Change Base Tile Layer

Flow of Events:

- The user selects a different base tile layer from the dropdown menu (e.g., Esri Topo).
- The application updates the choropleth map to display the selected base tile layer.

Exit Condition: The user can explore different geographical views and styles based on the newly selected base tile layer.

Exception Handling: If base tile layer options fail to load, default to a suitable alternative with appropriate messaging.

View Detailed Information in Popup

Actor: User

U5: View Detailed Information in Popup

Flow of Events:

- The user hovers over or clicks on a specific region displayed on the choropleth map.
- A popup appears, showing detailed information such as region name, risk type, risk level, and additional data.
- The user can interact with the popup to view more information or close it.

Exit Condition: The user can access comprehensive details about the selected region directly on the map.

Exception Handling: Ensure popups and tooltips are responsive and correctly displayed across different devices and screen sizes.

Compare Data with Bar Chart

Actor: User

U6: Compare Data with Bar Chart

Flow of Events:

- The user selects a risk type (e.g., Population) and risk level (e.g., Medium) from the drop-down menus.
- The choropleth map updates to show the selected data, depicting landslide risk across regions.
- The bar chart updates to visually compare the selected risk type and level across different regions.

Exit Condition: The user can interpret data trends and comparisons between the choropleth map and bar chart representations.

Exception Handling: Ensure smooth updates and synchronization between the choropleth map and bar chart without delays.

Sort Landslide Inventory Data

Actor: User

U7: Sort Landslide Inventory Data

Flow of Events:

- User interacts with the "Sort Order" dropdown to select the desired sorting order ('Default Order', 'Highest to Lowest', 'Lowest to Highest').
- The application fetches the landslide inventory data from the Flask API.
- Based on the selected risk type and risk level, the application sorts the data accordingly.
- The sorted data is displayed on the bar chart, updating the visualization to reflect the sorted order.

Exit Condition: User views the landslide inventory data sorted according to selected criteria.

Exception Handling:

If data fetching fails due to server issues or connectivity problems, the application notifies the user with an error message.

Visualize Data with Charts

Actor: User

U8: Visualize Data with Charts

Flow of Events:

- The user navigates to the Statistics tab.
- The user selects a data type (e.g., Surface Landslides or Population Landslides) from the dropdown.
- Optionally, the user selects a risk level (e.g., Medium LS risk) from the dropdown.
- The user selects multiple provinces to analyze from the dropdown.
- The user selects 'Bar Chart' and/or 'Pie Chart' from the visualization options.
- The charts are updated to visually represent the selected risk type and level across the chosen provinces.

Exit Condition: The user can interpret data trends and comparisons between provinces using the selected charts.

Exception Handling: Ensure smooth updates and synchronization between the dropdown selections and charts without delays.

View Landslide Risk on Map

Actor: User

U9: View Landslide Risk on Map

Flow of Events:

- The user navigates to the Map tab.
- The user selects a data type (e.g., Surface Landslides or Population Landslides) from the data type dropdown.
- The user selects multiple provinces to view on the map from the province dropdown.
- The map updates to show polygons representing the selected provinces with appropriate risk data as popups.

Exit Condition: The user can visually identify landslide risk areas on an interactive map.

Exception Handling: Ensure smooth map rendering and accurate display of polygons and popups without delays.

3.2 Use Case Diagram

The following use case diagram depicts a high-level overview of the relationship between use cases,

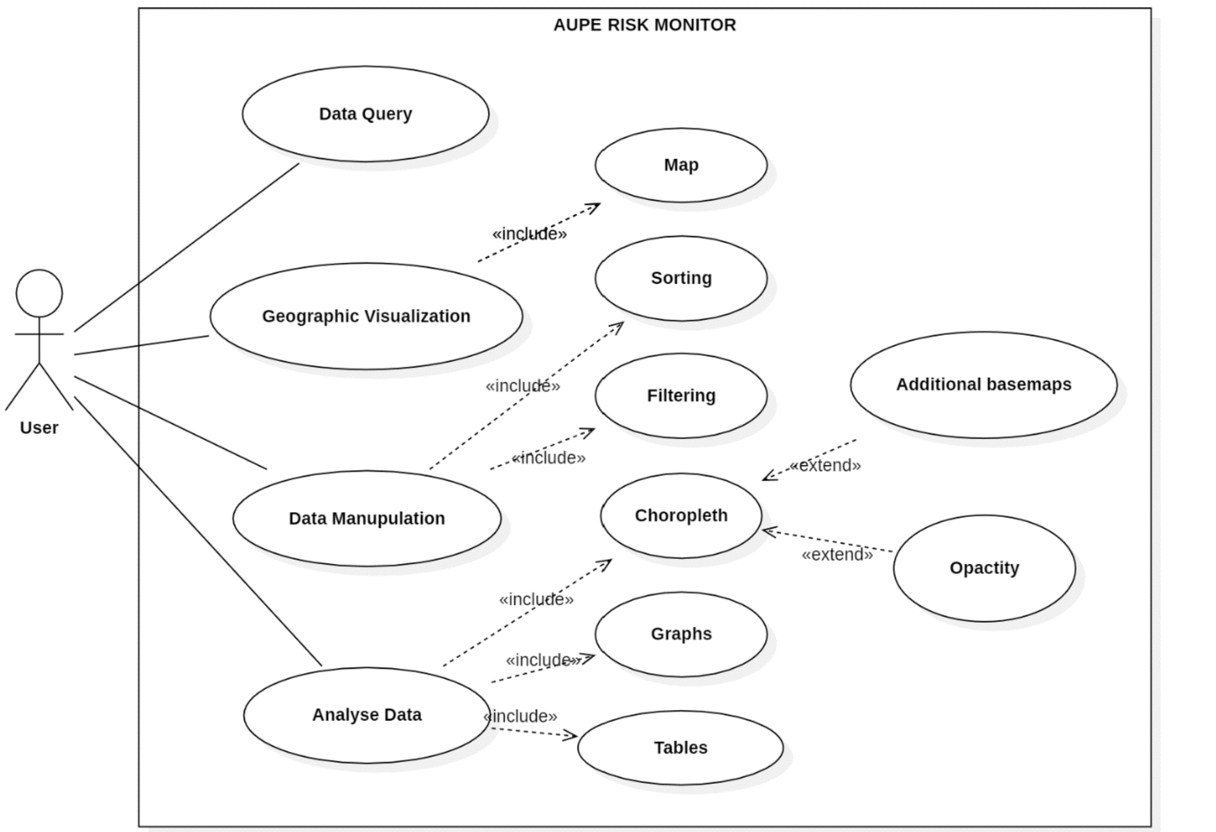

Figure 1: Use case diagram

- [Use Case Diagram - RASD]
- [Use Case Diagram - UML]

3.3 User Stories

- As a user I want to get data of landslide risk at certain a location.
- As a user I want to see the statistics of the location I want.
- As a user I want to see the data in a meaningful way through charts.
- As a system administrator I can provide web server for data cleaning and preprocessing before sending the results to the users.
- As a user, I want to view a map overlay depicting landslide risk zones, so that I can visually assess the areas of potential concern.
- As a user, I want to search for specific geographic locations on the map, so that I can access information about landslide risk in those areas.
- As a system administrator, I want to conduct periodic software updates and maintenance tasks to ensure the application remains up-to-date and functional.

4. Requirements

4.1. Functional Requirements:

- The system should allow the user to query the data.
- The system should be able to visualize the data according to some predefined visualization options implemented by the administrator through map-based views and interactive graphs.
- System should be able to provide custom viewing options to the user when the user filters some parameters.
- The system should be able to provide interactive tables to the user where the user can have a choice to select from the widget.
- The system should be able to allow the user to view the data after the user defined all the sorting and/or filters.
- System should be able to add additional basemaps and layers on the map-based views.

4.2. Nonfunctional requirements:

- The system should be able to query the data (risk) that the user wants that is within the domain of the database.
- The system should be able to retrieve the data from the database based on the user queries.
- Webserver should be able to do some data cleaning and preprocessing before returning the results to the user.

4.3. Technical Requirements

- The system shall use PostgreSQL as a database management system.
- The system shall use Flask as the web-server framework.
- The system shall use Pandas for data manipulation and analysis while GeoPandas for handling geographic data processing.
- The system shall use Jupyter Notebook for building interactive dashboard.
- Visualization libraries such as Matplotlib, Plotly and Folium shall be used for generating visualizations.
- The system shall integrate data from the Italian Institute for Environmental Protection and Research (ISPRA) IdroGEO API.

- The dashboard shall integrate data with leaflet for map-based views.

References

My Way: European Smart Mobility Resource Manager

Use Cases and Requirements for Standardizing Web Maps

Requirements Engineering for Web Applications – A Comparative Study