**Design Document and Test Plan**

***Emilia-Romagna region*** *exposure to potential risk and hazard to* ***flooding*** *and its impact on* ***households***

Sarmiento Ospina, Nataly Alejandra

Saud-Miño, Claudia Isabela

Wang, Xinmeng

Sayegh, John Cullen

**Deliverable:** DD

**Title:** Design Document and Test Plan

**Authors:** Sarmiento Ospina, Nataly Alejandra, Saud-Miño, Claudia Isabela, Wang, Xinmeng, Sayegh, John Cullen

**Version:** 1.0

**Date:** 31.05.2024

**Download page:**

**Table of Contents**

[**1. Introduction 4**](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.qypbk8vzzt4l)

[1.1. Purpose 4](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.t0yrmsfq414s)

[1.2. Scope 5](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.w1jp4isviqze)

[1.3. Definitions, Acronyms, and Abbreviations 5](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.nncq78vnpehx)

[1.4. References 6](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.tdfk03mtzoak) [7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.kiarmxelp935)

[**2. System Overview 7**](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.gmnn53utxl5c)

[2.1. Project Description 7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.y0xcec6z2vmz)

[2.2. Key Features 7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.h2u33deb3no7)

[2.3. Stakeholders 7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.k11tspj63mbv)

[2.5. Constraints 8](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.ihh8ogvdu2lk)

**3**[**. Architectural Design 1**](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.bg500fbm1r2o)**3**

3[.1. Overview 7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.y0xcec6z2vmz)

3.[2. System Components 7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.h2u33deb3no7)

3.3. Data Flow [7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.k11tspj63mbv)

**4**[**. Software Structure 1**](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.bg500fbm1r2o)**3**

4[.1. Presentation Tier 7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.y0xcec6z2vmz)

4.[2. Application Tier 7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.h2u33deb3no7)

4.3. Data Tier [7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.k11tspj63mbv)

**5**[**. User Interface Design 1**](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.bg500fbm1r2o)**3**

5[.1. Navigation Tools 7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.y0xcec6z2vmz)

5.2. Data Visualization [7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.k11tspj63mbv)

5.3. Analysis Tools [7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.k11tspj63mbv)

**6**[**. Use Cases 1**](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.bg500fbm1r2o)**3**

6[.1. Use Cases Diagram 7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.y0xcec6z2vmz)

6.2. Detailed Use Cases [7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.k11tspj63mbv)

**7**[**. Integration, Implementation, and Test Plan 1**](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.bg500fbm1r2o)**3**

7[.1. Implementation Plan 7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.y0xcec6z2vmz)

7.2. Integration Strategy [7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.k11tspj63mbv)

7.3. Testing Plan [7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.k11tspj63mbv)

**8**[**. Organization 1**](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.bg500fbm1r2o)**3**

8.[1. Team Roles and Responsibilities 7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.y0xcec6z2vmz)

8.2. Project Timeline [7](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.k11tspj63mbv)

**10**[**. References 1**](https://docs.google.com/document/d/1SLCSG0aqxZWa-uWXuzRZ5OGjhtbT3mlEQ26WJzmntVU/edit#heading=h.bg500fbm1r2o)**3**

### **1. Introduction**

The Design Document is an important document in the implementation of the project, as it ensures quality-control and sets the expectations for the final end-use product. The following pages contain a step-by-step process explaining the design and development of the project, starting with high-level aims and goals of the application, followed by a detailed analysis of the project database, its software structure, UI design, use cases and stakeholders, and finishing with implementation and team contributions. The application’s purpose and scope described below are a summary of the Requirements Analysis Specification Document (RASD), already developed earlier in the project timeline to delineate the overall aim of the group’s work.

#### **1.1 Purpose**

The purpose of this document is to outline the design of a software system for analyzing flooding in high-risk areas for vulnerable populations in the Emilia-Romagna region of Italy. Given that climate-change related flooding events are increasingly common across the world and threaten the lives of millions of people each year, the topic is relevant for several use categories and scenarios both in Italy and abroad. Within the Italian context, the region of Emilia-Romagna was chosen as the use case for several reasons; the region’s flat topography, its relative proximity to the Po River valley which winds its way throughout the region, and its high density cities and towns with vulnerable populations render it vulnerable to excessive flooding events. Future flooding has the potential to threaten the lives of thousands of people and cause billions of euros in economic damage across Emilia-Romagna. For these reasons, it is a useful exercise to develop a software that can easily highlight and identify areas where potential flood threat levels, in relation to vulnerable populations, are at an elevated risk within the region. It is hoped that this project could be expanded to other Italian regions for a large, composite view of flood risk across the country. This data and the method it is accessed and communicated is important for future considerations regarding regional urban planning, flood mitigation schemes, sustainable agricultural practices, and natural biodiversity in the Emilia-Romagna region.

#### **1.2 Scope**

This project aims to develop a web-based application that provides real-time flood risk analysis, data visualization, and alerts to assist in disaster management and planning. The ability to visualize the flood mapping data—which is already provided by Idrogeo, an open source data platform for hydrological risk–will allow organizations and individuals to better understand flood risk in their immediate vicinity in a clear and effective manner. The application allows users to easily query flood risk data related to a specific geospatial coordinate and visualize it using GIS software. In doing so, it is hoped this tool empowers users to combat “information asymmetry”, that is, the imbalance of power and transactions between individuals and public and private entities, so that citizens and local groups are more aware of population vulnerability and flood risks.

1. Flood Risk Analysis
   * Historical Data Analysis: Utilize historical flood data to identify patterns and predict future flood events.
   * Real-Time Monitoring: Integrate real-time weather data to monitor current conditions and provide up-to-date risk assessments.
2. Data Visualization
   * Interactive Maps: Create detailed and interactive maps that display flood-prone areas, current water levels, and potential future risks.
   * Heat Maps and Risk Zones: Highlight high-risk zones and areas most vulnerable to flooding, focusing on regions with significant vulnerable populations.
3. User Interaction and Customization
   * User Profiles: Enable users to create profiles where they can save their settings and preferred locations..
   * Query and Analysis Tools: Provide tools for users to perform custom queries and analyses, generating reports and insights tailored to their needs.
4. Stakeholder Integration
   * Local Government: Offer functionalities to support local authorities in planning and decision-making processes regarding flood management and emergency response.
   * Community Engagement: Engage local communities by providing accessible information and tools to help them understand flood risks and prepare accordingly.
5. Technological and Design Considerations
   * Scalability: Design the system to handle large volumes of data and a growing number of users, ensuring performance remains robust under heavy load.
   * Usability: Focus on creating an intuitive and user-friendly interface that is accessible to a wide range of users, including those with limited technical expertise.

By addressing these objectives, the project aims to provide a valuable tool for analyzing and managing flood risks in the Emilia-Romagna region, ultimately contributing to improved safety and preparedness for vulnerable populations.

#### **1.3 Definitions, Acronyms, and Abbreviations**

* GIS: Geographic Information System
* DBMS: Database Management System
* The app: the application that processes and visualizes the data
* JSON: JavaScript Object Notation
* gpd: GeoPandas abbreviation
* Matplotlib.pyplot: plotting function for Python
* Bokeh: Vizualization plug-in for Python
* PostgreSQL: Open-source relational database management system
* Flask: A web framework tool for Python
* API: Application Programming Interface
* ISPRA: Superior Institute of Environmental Protection and Research of Italy
* RASD: Requirements Analysis and Specification Document

#### **1.4 References**

### **2. System Overview**

#### **2.1 Project Description**

The project involves creating a system to collect, store, analyze, and visualize flood data for the Emilia-Romagna region. It will focus on areas most at risk and vulnerable populations. The application will leverage historical data to map areas of low, medium, and high flood risk, while overlaying demographic information related to population density and vulnerability (people aged 65 and older) who are at high risk of death and mortality in the event of a major flood event.

#### **2.2 Key Features**

* Real-time data integration: The system will integrate IdroGEO’s API data so that it can be easily updated when new flood data is released.
* Interactive maps: The user will be able to select a geospatial coordinate anywhere in Emilia-Romagna, which will provide up-to-date analysis on flood risk.
* Risk analysis tools: A risk analysis score, which overlays vulnerable population information taken from census data and flood risk level data taken from IdroGEO, will provide a specific and tailored risk assessment.

#### **2.3 Stakeholders**

* Local Government: Responsible for regional planning, disaster management, and policy-making, requiring detailed flood risk analyses and reliable data to make informed decisions.
* Emergency Response Teams: Need accurate and timely information to effectively coordinate and execute flood response and mitigation efforts.
* Residents of High-Risk Areas: Require accessible information about flood risks and alerts to take preventive measures and ensure safety.
* Environmental Researchers: Benefit from access to detailed flood data and analysis tools for research and development purposes.
* Non-Governmental Organizations (NGOs): Focus on community awareness and disaster preparedness, utilizing the system to educate and assist vulnerable populations.

#### **2.4 Constraints**

* Data Availability: The system’s effectiveness depends on the availability and accuracy of real-time and historical flood data.
* Internet Access: Users need reliable internet access to utilize the web-based application and receive real-time alerts.
* Resource Accuracy: Census and flood risk assessment data can only be traced to the comune-level (Italy’s equivalent lowest level of administrative organization), which means the analysis will not provide a granular, specific analysis of an area or plot smaller than the municipal level.
* Compliance with Regulations: The system must comply with local, national, and international regulations regarding data privacy and security.

By outlining these aspects, the System Overview provides a comprehensive understanding of the project's goals, features, stakeholders, constraints, and assumptions, setting a clear direction for the development and implementation of the flood risk analysis system.

### **3. Architecture Design**

#### 3**.1 Overview**

This software is developed using Flask, dash and postgreSQL by separating the front-end and back-end and realizing that the results of the user query on the user-interface show the corresponding Emilia Romagna Region's flood hazard class on household and culture heritage.

The project will use Flask as the backend development framework, and employ Python’s library dashboard libraries for frontend development.The database will be designed according to the attributes of retrieved data from IdroGEO, and established through PostgreSQL, data exchange and storage will be completed in Json format.

#### 3**.2 System components**

1.Database: Storing data and providing a data access interface. The database component will use PostgreSQL, which is an open-source relational database management system (RDBMS) that maintains extensibility (allowing data customization) and utilizes the standard language SQL (Structured Query Language) of RDBMS. In this way, it will be easy to update, retrieve, and add/remove data quickly from multiple sources. PostgreSQL’s extension, PostGIS, is useful for transforming data into geospatial polygons and points, which is necessary for the final map-based product offered to the end user.

2.Flask: Flask is used for handling data requests and responses, and calling database query operations. Flask is an easy and simple to use Python framework that can build out web applications and works well with SQL-based databases such as PostgreSQL.

3.Dashboard: Providing interactive user interface, displaying data and performing data analysis.

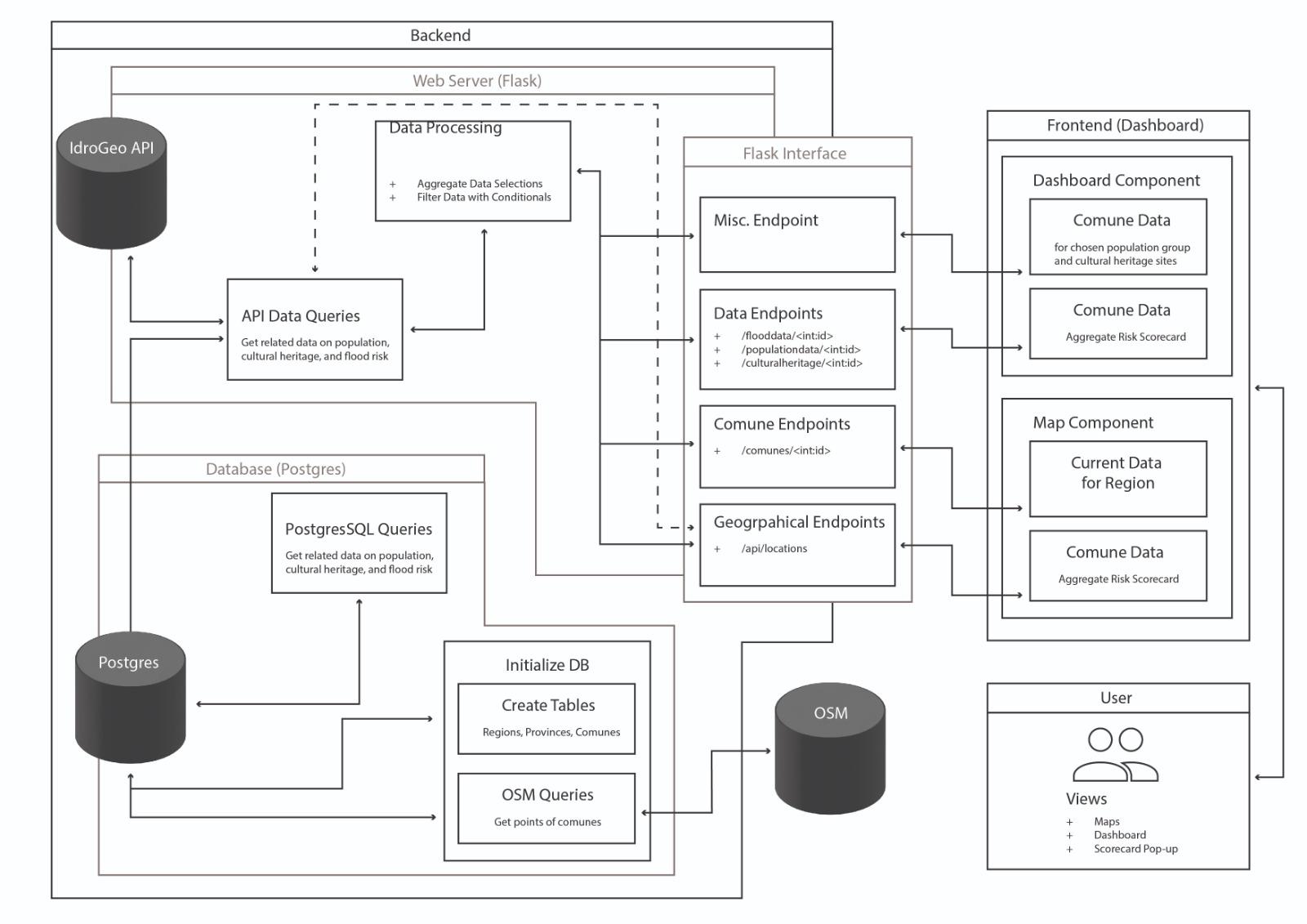


Figure 1:System components

#### 3**.3 Data flow**

The general idea of data process flow: when a user initiates a data request, the Flask application receives the request and routes it to the corresponding view function based on the routing rules. The view function handles the request, retrieves the required data from the database, and then passes the data to the frontend Dash application for display.

\*chart

### **4. software structure**

#### 4.1 Presentation tier

The presentation tier is responsible for rendering and presenting interactive user interfaces.This project will use Dash for creating the interactive interface.

The UI of the dashboard will consist of several key components:

1.Home Page: Displays an overview of the flood index data for Emilia-Romagna, provide

controls for querying and filtering.

3.Interactive Map: A central feature, displaying flood index data on a map using Geopandas and OpenStreetMap (OSM) as the base map.

4.Data Visualization: Charts and graphs showing historical trends, statistical analyses, and predictive models related to flood data.

5.Filter and Search Options: Users can filter data by selecting specific areas within the region Emilia-Romagna.

The presentation tier communicates with the application tier (Flask) through REST APIs. Data requests from the frontend are sent to the Flask backend, which retrieves and processes the data before sending it back to the frontend for display.

#### 4.2 Application tier

The application layer built with Flask is the bridge between the front end (presentation layer) and the database (data layer). It handles business logic, data processing, and API endpoints for the front end.

The Flask application exposes several REST API endpoints:

1.api/pir/regioni/8

Parameter: id of the interested region, in this case the id of Emilia Romagna.

Return value:flood index, historical flood data, population ,etc.

2. pir/comuni/{id}

Parameter: id of the interested comuni, in this case the id of comuni inside Emilia Romagna.

Return value:flood index, historical flood data, population ,etc.

The data processing sections will mainly consists of the following functions:

1.Data fetching:Fetchs flood index and related data of Emilia-Romagna to establish the local database.

2.Data Aggregation: Combines and processes raw data to generate summary statistics and insights.

#### 4.3 Data tier

The data tier consists of the PostgreSQL database, which stores all the relevant data for the application. This includes raw flood index data,historical records,historical heritage data, population data.

The database schema is designed to store and query flood index along with other related information.The key tables and their structures are:

\*Tables

### **5.Test plan**

The test plan of this system will mainly focus on the Flask backend and its interaction with the database.The primary focus is to ensure the API endpoints in the software system, data is actually stored and retrieved from the local database,and ensure the system is reliable under different circumstances.

The test strategy will include unit tests and end to end tests.The unit tests will be used to test the components of Flask,particularly its functions.The end to end tests will test the entire workflow of this system, from API requests to database operations and responses.