Software Engineering for Geoinformatics 2020

8th of June, 2020

Requirements Analysis and Specification Document

Web-Based Geospatial Data Analysis Application on Environmental Noise Pollution Survey in Pune, India

Version: 2.0

Authors:

R. Cedeno, D. Cumming, H. Mahmoudi, A. Zacchera

Contents

1. Introduction

Purpose

The purpose of our project is **to inform and to involve communities potentially affected by environmental noise pollution (ENP)** by means of a web application for desktop. Noise pollution is a **public health issue** according to the World Health Organization (WHO) and other leading research bodies:

“Noise is an important public health issue. It has negative impacts on human health and well-being and is a growing concern.” – WHO Regional Office for Europe, Environmental Noise Guidelines for the European Region, 2018.

“*Chronic exposure to environmental noise has significant impacts on physical and mental health and well-being.”* – European Environment Agency, Environmental noise in Europe, 2020.

Indeed, large scale epidemiological studies have shown a correlation between the population’s exposure to environmental noise and adverse health effects. Noise pollution has **negative impacts on human physical and mental health and well-being** as stated below:

“Exposure to noise can lead to auditory and non-auditory effects on health. Through direct injury to the auditory system, noise leads to auditory effects such as hearing loss and tinnitus. Noise is also a nonspecific stressor that has been shown to have an adverse effect on human health, especially following long-term exposure. These effects are the result of psychological and physiological distress, as well as a disturbance of the organism’s homeostasis and increasing allostatic load.” - WHO Regional Office for Europe, Environmental Noise Guidelines for the European Region, 2018.

Moreover, as found by the European Environment Agency, anthropogenic **noise pollution affects also a wide range of terrestrial and aquatic species** that inhabit the ecosystem, increasing mortality risk and emigration:

“Anthropogenic noise affects a wide variety of terrestrial and marine wildlife species causing a range of physiological and behavioral responses. These can reduce reproductive success and increase mortality and emigration, resulting in lower population densities.” – European Environment Agency, Environmental noise in Europe, 2020.

Scope

The scope of this project is to design and implement a **web-based application for desktop** that allows users **to access, retrieve, analyze and visualize, by means of an interactive mapping tool, the available ENP** data of a certain local area. On the website, the user will also be able to find general information regarding the issue of noise pollution, data description and instructions on how to contribute to the data collection since this project aims not only to inform people but also to involve local communities.

The dataset of ENP measurements that will be used in this project comes from Epicollect5 and contains data collected in an specific area of the municipality of Pune, India.

Short overview

As mentioned in the scope, the web-based application will allow users to access, retrieve, analyze and visualize, by means of an interactive mapping tool, the available ENP data of a certain local area. On the website the user will also find general information regarding the issue of noise pollution, data description and instructions on how to contribute to data collection.

With respect to the dynamic contents of the website the application will allow the user to visualize ENP data through an interactive map and to obtain statistical data description and analysis tools, for example filtering noise measurements by location, time of the day or any data feature.

Acronyms and definitions

Name	Definition
Epicollect5	Epicollect5 is a mobile and web application for free and easy data collection. It provides both the web and mobile application for the generation of forms and freely hosted project websites for data collection.
WSGI	Web server gateway interface (WSGI) is a simple calling convention for web servers to forward requests to web applications or frameworks written in the Python programming language.
ENP	Environmental noise pollution (ENP) is any unwanted or harmful outdoor sound created by human activity, such as noise emitted by different means of transport – e.g. road traffic, rail traffic, air traffic – and industrial activities.
DBMS	Database Management Service (DBMS) is a software that interacts with end users, applications and the database itself to capture and analyze stored data.
Web Application	Web application, or web app, is a client–server computer program that the client (including the user interface and client-side logic) runs in a web browser.
REST	Representational state transfer (REST) is a software architectural style that defines a set of constraints to be used for creating Web services. Web services that conform to the REST architectural style provide interoperability between computer systems on the Internet.
API	Application programming interface (API) is a computing interface to a software component or a system, that defines how other components or systems can use it.
GitHub	GitHub provides hosting for software development version control and source code management using Git.
WGS	World Geodetic System (WGS) is a standard for use in cartography, geodesy and satellite navigation. WGS 84 is the latest revision of this standard, established and maintained by the National Geospatial Intelligence Agency since 1984.

UTM	Universal Transverse Mercator (UTM) is a conformal map projection system for assigning coordinates to locations on the surface of the Earth.
Static	Static refers to any page or content of the website that is written in HTML and CSS languages and that does not change as a result of the interaction with the user.
Dynamic	Dynamic refers to any page or content of the website that is written in Python language and that changes as a result of the interaction with the user.
Dashboard	A dashboard is a type of graphical user interface which often provides at-a-glance views of maps or key performance indicators relevant to our project
CRUD	Operations to be performed in an SQL database. These transactions describe the overall interaction with the system and they are composed by: Create, Read, Update and Delete.
PostgreSQL	Free and open-source relational database management system with SQL compliance. It will be used in the development of this project.

2. Application domain and phenomena description

System

The system is a Web Application, which will acquire a client-server architecture composed by:

1. The client will be a web browser which will display the output interface of our system, with which the user will be interacting. This client will be able to visualize static data composed of description elements of the project and of the components that build-up the application.
2. The web browser will interact with dynamic elements. These elements will make requests to the Web Server, providing the content and in response, a part of the web browser will display this information. The user will be able to interact with the information in real time, in order to visualize it in different ways.
3. The server support will be provided by GitHub that can be accessed by any person with no extra cost and high support and functionality. The disadvantage of using this system is that it does not provide an independent URL.
4. The DBMS will be supported by PostgreSQL which will interact with the infrastructure provided by EpiCollect5 (accessed by the system REST API). The information will be updated, uploaded and requested by the server in real time. This will guarantee that the database is always available without depending on the EpiCollect5 system to perform CRUD operations.

Users

Given the goal to inform and involve communities potentially affected by environmental noise pollution, the user will be provided with tools, maps and other functions to explore and retrieve information about collected measurements. The information displayed could be different in aspects in consideration also of the level of knowledge of the user, for example the outcome could be useful metrics of punctual locations or an interactive map that shows noise pollution from interested areas.

From a profile point of view, we assume that users interested in our web application can be classified in three main groups based on the level of knowledge and domain of users:

1. Users with no a-priori knowledge about the ENP problem who would like to learn about this topic and to discover the level of ENP around where they live. We provide these users with a clear and intuitive web interface where they can access resources and information. This interface will include interactive maps with users being able to select the layers to be rendered.
2. Contributing users who already have some familiarity with the ENP domain and wish not only to visualize results but also to participate in data acquisition. The website will include a part

dedicated to this class of users where they can find instructions on how to contribute to data collection by means of Epicollect5.

3. Users belonging to local government or city planning committees who want to go deep on ENP data by professionally analyzing it in order to perform health studies, risk assessments or action plans that aim to reduce ENP.

Dataset

The dataset consists of 971 georeferenced measurement points of environmental noise pollution. The collected data refer to a certain area in the municipality of Pune, India. The main features of the dataset are summarized below:

1. Point position: expressed both in WGS84 geodetic coordinate reference system (latitude and longitude in decimal degrees) and UTC zone 43N projected coordinate reference system (east and north in meters).
2. Position accuracy: measurement error in GPS positioning. In this database point accuracy ranges between 1 m and 65 m, revealing that positioning was likely obtained through GPS receivers embedded in mass market devices, such as smartphones.
3. Date and time of the measurement.
4. Land use: Classification of the urban zone. Land use can be Commercial, Residential, Recreational, etc.
5. Noise Pollution instrumental measurement in dB.
6. Picture: Photograph of the surroundings in the moment of the measurement.

Product functions

The website will have the following functions

1. Mapping Environmental Noise Pollution Data: data layers (measurement data points, heat map, dot map, etc.) are displayed on top of basemaps (e.g. OSM, Bing). These maps will be dynamic objects with which the user might interact, for example selecting the visible layers, as opposed to static maps rendered as bitmap raster images.
2. Measurement Data Points Exploration: In the map visualization, each measurement data point can be clicked by the user, so the details about that specific measurement are displayed (e.g. time, land use classification, ENP value, picture). In addition, the raw collected data will be downloadable by the user.

3. Data Analysis Tools: The mapping tools will be contained in a dashboard where the user will be able to retrieve customized plots and descriptive statistics of the data. The geospatial data can be mapped through different techniques (e.g. heat map, contours, dot map, etc) over user-defined filtered data (e.g. filters by time, coordinates, ENP value). The plots and analytics tools also can be based on filtered data.
4. Discovering How to Contribute: The website will have a section where the user can get a general overview about the project objective and scope, additional information and references about the ENP problem as well as the details about how to contribute to the project data set through the Epicollect5 application.

Software

The software to be used by our web application will be provided by the following interfaces:

1. HTML code: this part of the software will contain the main display and structure of the web application. This will work as the backbone of the application sustaining the rest of the software implementation
2. CSS: Will provide tall the templates and styling of the application
3. Python code: the Python code will serve the web application as the dynamic and content manager which will be interacting with the DBMS, performing operations between components and will be in charge of managing the interactive maps that are the main source of display for web applications.

Operations

For the website to work in a correct and efficient manner, several operations need to be completed, in particular:

1. The user will perform operations inside the web application that will trigger internal processes to request information to the server and web services.
2. Web client - web server: The operations involved in this section will contain requests made to the web server with JSON, HTML, Python codes to provide the information to be displayed to the user.
3. WSGI: Web Server Gateway Interface will include all the operations between the web server and the python code that will be managing the JSON files and Databases that contain the information relevant to the user. This interface may also manage the user registration to the web application.

Hardware

The hardware supporting the website application will be provided by cloud services which will be available at the time of publication. This hardware will guarantee that the traffic of users can be supported and provides the requested information.

Since this web application will be hosted in a free and open source system, the traffic will be limited to a low number of simultaneous users and operations between the dynamic content and the needs of the users.

Domain assumptions

In this section we provide a list of assumptions and dependencies:

1. The storage of the data will take place in Epicollect5 host servers.
2. Epicollect5 will continue to provide access to the public database via REST API
3. The quality of data will depend on the accuracy of data collected by Epicollect5 users.

Relevant phenomena

In this section we briefly discuss the phenomena that take place in the “real world”, in the “machine” and those that are shared among the two. The purpose of describing these phenomena is to guide our analysis of use cases and requirements in the following sections.

World phenomena

The main phenomenon taking place exclusively in the real world is the recording of the levels of ENP by collectors. This phenomenon occurs any time the user wants to contribute to the survey, in buzzling streets of Pune or in any other place where he or she might be.

The users also might upload its measurements to the EpiCollect5 platform, which includes the positioning of each point by means of a GPS device.

Another world phenomenon, is the need to retrieve and analyze ENP data by the user. This can be done either through the user-interactive tools that our application is going to offer or by downloading the raw data and doing an external implementation or using another software.

Shared Phenomena

In this section, the Phenomena shared between the world and the machine are described. This type of Phenomena encompasses the interactive **user-controlled** interface whose components are described:

1. Map: Interactive geographic information tool.
2. Visualization options: Options to visualize spatial data (heat maps, contour maps, dot maps, hot spot maps, etc)
3. Layer Manager: List of layers to set visibility or hiding.
4. Filter Manager: Options to customize subset of data from which operations are done. From this setting, the database query is generated.
5. Data Analysis Tool: Manager of client-side data analysis operations. The following operations are available:
    a. Descriptive statistics query (mean, standard deviation, quantiles, etc.).
    b. Data exploratory plotter: It allows the client to create different types of plots (scatter plots, histograms, box plots, etc.) on the filtered data subset.
    c. Style manager: It allows the client to define different styles for the generated plots.

3. User cases and requirements

User cases

In order to explain how the requirement fulfilment are mapped in the software functionalities considering possible exceptions, this section is going to address an explanation about the actions taken by the software and the user in a list of cases that are useful to explain the requirements.

In this section we describe what is going on from server-side and client-side when the user cases happen by specifying the different actions that take place in these situations.

UC1: Discover about noise pollution and how to contribute in data collection

Providing a platform to learn about the topic of noise pollution and how to contribute to the survey is one of the purposes of our project. This user case describes the actions taken by the user and the system in order to fulfil this requirement:

1. The user enters the main website.
2. The system provides access to static pages containing information about the noise pollution problem and hyperlinks to external resources (e.g WHO website, etc.) as well as instructions on how the user can contribute to the data collection in the Epicollect5 platform.
3. The user can click on hyperlinks to navigate inside the website (e.g. back to homepage, go to the map, about us section, etc) or to access external websites or resources (e.g. Epicollect5, WHO, etc.).

UC2: Data points visualization

Through the interactive map, users can obtain the details about each measurement points, as in this user case is described:

1. The user enters the user interface for interactive mapping.
2. The whole available raw data is retrieved from the Epicollect5 host server, cleaned up (remove strings from numerical values and setting as missing the invalid ones) and placed in the DBMS.
3. From the DBMS, a Geodataframe with the whole available data (checked and cleaned up) is generated by the Application Server.

4. The Geodataframe is input in the Mapping Tool, which adds it to the Interactive Map Figure (created with Bokeh Library).
5. The user places the cursor on a certain measurement point (its geographical position in the map).
6. The pop-up function (tabular dataframe) is activated. Therefore, once the cursor is placed in the point, the specific details about that point (measurement date & time, land use classification, ENP value, etc) and the corresponding picture are displayed as a pop-up.

   6.1. By using hovertool from Bokeh we are able to generate tabular tooltips for data points.

   6.2. This function enables us to retrieve specific data from column data sources like land use, measurement date and time, images and etc for every point that is shown in the map.

   6.3. It is also possible to custom HTML templates for tooltip. These HTML templates customize the tabular appearance for better displaying information and images.

7. The user is able to do the same with each available measurement point.

Exceptions:

1. There is a possibility that two different users measure one specific point and enter data in the Epicollect5, hence we have two measurements from one specific point.

UC3: Map data with a custom visualization

The Interactive map is one of the capabilities which is provided on the web-application in the aim of visualizing spatial data. This user case describes how a user can obtain a customized visualization of spatial data:

1. The user enters the user interface for interactive mapping.
2. The whole available raw data (which has already been retrieved from the Epicollect5 host server, cleaned up and placed in the DBMS) is obtained by the Data retrieval and pre-processing function.

3. As explained in UC2, the whole available data points from the DBMS are visible in the interactive map by default (with a certain default basemap).
4. The user wants to consider the data just within a specific area, in a certain time window and with a specific land use classification, so he/she does the corresponding settings in the friendly user-interface for filtering to keep just the needed data subset.
5. The settings request is taken by the Filter Manager, which performs the logical operations in order to obtain the indexes of the needed data subset.
6. The indexes of the filtered data points are delivered to the Mapping Tool, which use these to obtain only the needed data subset in order to add it to the Map Figure. The page is rendered again with the updated Map (with the filtered data points).
7. The user clicks the Visualization Options Menu and selects a certain style to plot the spatial data (heat maps, contour maps, dot maps, hot spot maps, etc). The user request is sent to the Data Analysis Tools.
8. The indexes of the filtered data points are also delivered to the Data Analysis Tools, which perform the logical operations to generate the required data to obtain the visualization with the data filtering.
9. The visualization data is input in the Mapping Tool, which computes and adds the custom visualization to the Figure. As before, the page is rendered again with the updated Map.

Exceptions:

1. The user is looking for a certain data point that he has added to EpiCollect5 database but still it doesn't appear in our web application; e.i. the DBMS data has not been updated already.
2. The basemap (loaded from a server such as Open Street Map) is currently not available, thus the web application only loads data points without a basemap.
3. Newly data that has been uploaded to the EpiCollect5 database and as a consequence has been updated to the DBMS has an incorrect format and cannot be displayed on the map.

UC4: Data analysis of the noise pollution

The user will be able to discover information about Environmental Noise Pollution also by means of interactive analytical tools (e.g. histograms and other charts). These tools are meant to be interactive in the sense that the user can choose which data to visualize, for example filtering data by specific land use or time of the day of the surveying. This use case describes the main analytical tools which will be provided by the web application:

1. The user enters the user interface for interactive analytical tools;
2. The whole available raw data which has already been retrieved from the Epicollect5 host server, cleaned up and placed in the DBMS, is obtained by the Data retrieval and pre-processing function.
3. As explained in UC2, the whole available data points from the DBMS are visible in the interactive map by default (with a certain default basemap).
4. Different operations and filters are available for personalizing these charts, for example the user can filter the data by land use or by the time of day when the noise pollution measure was taken. In this case the use clicks in the Filter Manager to display just the subset of needed data;
5. The settings request is taken by the Filter Manager, which performs the logical operations in order to obtain the indexes of the needed data subset;
6. The filtering information is input in the Data Analysis Tools.
7. Then, the user clicks in the Data Analysis service and selects the option to get Descriptive statistics.
8. The software calls the statistics function which computes and returns the general statistics of each data subset feature.
9. The statistics data is passed to the Template Engine. The page is rendered with the requested Statistic display.

Exceptions:

1. The user applies a filter (e.g. land use = residential) for which there is no data to be displayed in the plot; in this case the plot will result empty.

UC5: Repeatedly visualization-filtering requesting

Now we consider a case similar to UC4, but supposing that the user requests a sequence of data analysis requests (geospatial chart visualization or statistics over filtered data) and filtering updates. If the user does many statistics-filtering requests it is not a problem, since the statistics display will be simply updated and overwritten.

Given that each geospatial chart visualization requests are going to be added to the client-side map, this is an interesting user case. If we consider a scenario in which for each filtering request, each map layer is updated with the new filtering, then it would be very computationally expensive, especially when the number of layers is high. For handling this problem, the previously generated Geospatial Visualizations will not be updated with each filtering request.

The actions carried out by the user and the software in this case are outlined:

1. When the Mapping Tool is initialized, the only default layer is the measurement data points (the basemap is also visible but not as a user-controlled layer).
2. The user can do a filtering request by setting up the filter manager controls. The Filtering Function will obtain the required data point indexes (As Boolean Series) and send these to the Data Analysis and Mapping Tools.
3. The Mapping Tool will update the Bokeh Figure by modifying the Data Points layer according to the new filtering. The page is rendered again with the updated figure by the Template Engine.
4. The user can see the updated Data Points layer. Now he/she does a visualization request by specifying which type and submitting the layer name.
5. The Data Analysis Tools will receive the visualization request and the layer name. It will generate the figure data (taking into account the new filtering!) and send it to the Mapping Tool.
6. The Mapping Tool will update the Bokeh Figure by adding the new visualization layer according to the new filtering. The page is rendered again with the updated figure by the Template Engine.
7. Steps 2-6 can be repeated many times as illustrated in Figure 4.

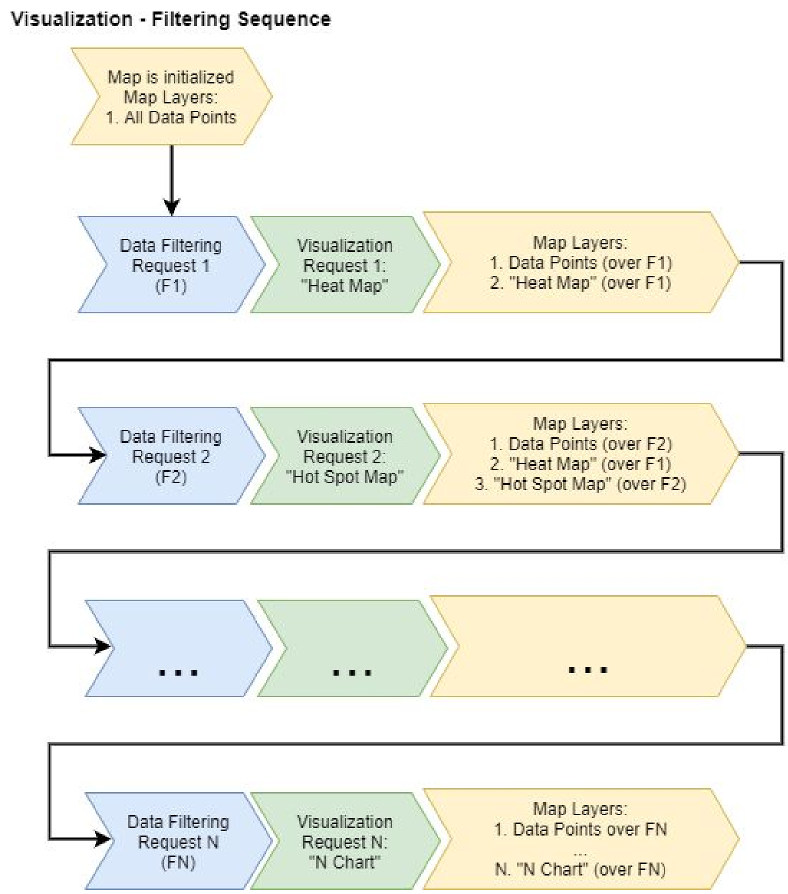

Figure 4: Flowchart which summarizes how the map is updated during a filtering-visualization request sequence.

UC6: Someone added invalid data to the Epicollect5 project

Now we consider a scenario in which invalid data is uploaded in the Epicollect5 platform. If this type of exception would not be taken into account, it could lead to failures in the software because each component expects an specific format for input data. In order to avoid failures, the data must be checked during the preprocessing and invalid data points are discarded. When a certain data point is discarded means that it is not considered for any purpose (visualization, statistics, filtering, etc.)

General user interface requirements

The user will interact with the application by a website. The website will be composed of HTML web pages, with hyperlinking to help the user navigation, and it will be optimized for desktop devices. The language of the website will be English.

Functional requirements

Base on the above user cases, the interactive user application must fulfil the following high priority functional requirements:

1. It must have an interactive map where layers of the data (measurement data points, heat map, dot map, etc.) and basemaps (e.g. OSM, Bing) can be selectable by the user in order to be visible.
2. Each measurement data point can be clicked by the user, so the details about that specific measurement are displayed (e.g. time, land use classification, ENP value, picture)
3. The raw measurement data table must be displayable in the website as well as available to be downloaded by the user in a set of formats (e.g. csv, shape, etc.).
4. The filters of the data can be personalized by the user. He/she can select a window of time, a specific spatial area or a filter based on a categorical or numerical data feature.
5. The user must be able to obtain descriptive statistics on the data as well as personalized plots using the available filters described previously.

The web-application may also implement the following functional requirements:

6. The spatial filter can be personalized by the user by means of a custom polygon user-drawing.
7. The user is able to find the GitHub repository in the page of the project.
8. The user is able to find additional references about the ENP problem.
9. The user can send feedback and report bugs to the development team.

Moreover, with reference to UC1, the web-application shall have a static section containing general information about noise pollution and instructions on how to contribute to data collection. This section will include hyperlinks for the users to navigate within the website or to access outside resources.

Performance requirements

The database storage in our project is synchronized with the one on Epicollect5 so the changes that any user might do via Epicollect5 platform will be also reflected on the web-app.

1. When the user enters the interactive application, the map and all the management services with their default settings must be ready to be used within 5 seconds.
2. Once the user has set up the filtering of the data (Filter Manager), the system must answer in no more than 5 seconds. This means that after this time the storage database must have provided the queried data subset and the filtered data must be available to be plotted.

Non functional requirements

In this section the Non-Functional requirements will be described. These requirements define how a system is supposed to be. These requirements will be divided into two sections: the execution qualities and the evolution qualities.

1. Execution qualities:
    a. In terms of security, the website must not put on risk the user while browsing or downloading the information, it is important that it is not possible for an external attacker to use the website in a malicious manner, modifying the information and affecting the user's integrity.
    b. Regarding usability, the user may be able to download data from the website. The system shall always provide a safe environment which ensures that the downloaded content is virus and malware free.
    c. The web application shall be a safe environment regarding data veracity, it must always display the real information uploaded to the EpiCollect5 interface without any malicious modification that can attempt against the data integrity.

2. Evolution qualities: during the development and deployment, the web application must be subject to several quality checks that will ensure that the requirements and final product aimed quality are met.
    a. Testability: due to the importance of having the least amount of bugs in our final product, our aim is to have a high testability; this refers to the fact that each part of the project will be tested as well as each of the components. Since the website will contain several layers and components as building blocks we will be able to test each of the parts as individual components.

b. Maintainability: As established in the testability specification, the project will be divided into several layers and components, due to this the maintainability will be high the following are the components divided into groups and their maintainability description.

    i. Server: the server system shall be highly maintainable because it will be hosted by an external service provider that will be in charge of maintaining all of the Hardware that supports the website. The small amount of users is also important to consider this as a highly maintainable project

    ii. Static website: will require periodical maintenance with a low frequency to keep the static information up to date. This will not be time consuming and it may be done in a simple and fast way.

    iii. Dynamic website: this component of the system is considered to have a high maintainability given that it will be fully automated with the Epicollect Rest API, alongside with the Python modules. The most demanding component inside the dynamic website in terms of maintainability is the database management, since the data cleaning may be needed to change depending on the updates on the Epicollect5 database or API.

c. Scalability: the system shall be scalable, with the ability of extending to countries or areas that currently do not have service (outside the area that today contains information. The modular structure of the product will permit us to build more components around the website and simplify the scalability.

Effort

R. Cedeno: 12 hours

D. Cumming: 12 hours

H. Mahmoudi: 12 hours

A. Zacchera: 12 hours