



Software Engineering for Geoinformatics

## Requirements Analysis and Specification Document

# Web-Based Geospatial Data Analysis Application on North Rupununi Wetland Guyana, South America

### Authors:

Ali B. Mohmmmed  
Hussamalden M. Shereif  
Mazin A. Saeed  
Nizar O. Fadi Elseed  
Nshwan A. Khairalla

**Version 2**

April 19, 2021

## Table of Contents:

<b>Table of Contents.....</b>	<b>2</b>
<b>Introduction .....</b>	<b>3</b>
General overview:.....	3
Goal:.....	4
Scope:.....	4
<b>Domain analysis.....</b>	<b>5</b>
Users .....	<b>Error! Bookmark not defined.</b>
<b>Phenomena .....</b>	<b>5</b>
Relevant Phenomena:.....	5
World Phenomena:.....	5
Machine Phenomena:.....	5
Shared Phenomena:.....	6
<b>Use Cases .....</b>	<b>7</b>
UC1: Registration to the website .....	7
UC 2: login.....	7
UC 3: Visualization .....	8
UC 4: Data Analysis .....	8
UC 5: Enter Comment .....	9
UC 8: Add data by members to the database.....	10
<b>Requirements and Assumptions .....</b>	<b>12</b>
Requirements:.....	12
Technical Requirements.....	12
Non-Functional Requirements:.....	12
Functional Requirements:.....	12
Domain Assumptions: .....	13
<b>Efforts: .....</b>	<b>14</b>
<b>References:.....</b>	<b>15</b>

## Introduction

### **General overview:**

An ecosystem is a geographic area where plants, animals, and other organisms, as well as weather and landscapes, work together to form a bubble of life.

A wetland is a distinct ecosystem that is flooded by water, either permanently or seasonally, where oxygen-free processes prevail. The primary factor that distinguishes wetlands from other landforms or water bodies is the characteristic vegetation of aquatic plants, adapted to the unique hydric soil.

Wetlands are lands transitional between terrestrial and aquatic systems where the water table is usually at or near the surface, or the land is covered by shallow water. . . ., Wetlands must have one or more of the following three attributes:

(at least periodically, the land supports predominantly hydrophytes; the substrate is predominantly undrained hydric soil; and the substrate is saturated with water or covered by shallow water at some time during the growing season of each year).

The different vegetation types in a wetland can be divided up into emergent, floating, and submerged plants.

The North Rupununi Wetlands support a lot of vegetation type that differs from Grass, Shrubs to trees.

The data we are using is located North of Rupununi river in Guyana and collected from 18th of September 2020 to 13th of April 2021.

**Goal:**

The aim of the project is to explain the characteristics of North Rupununi wetlands and its valuable “services” for society and the environment, by providing a useful visualization and analysis tools in a desktop web-based application interface, which is a result of the classification of North Rupununi into subareas regarding to vegetation type, vegetation index, grass density, moisture content, watershed, streams.

**Scope:**

this project is concerning about providing information to users (Specialist/researchers) about wetland areas and to let them have access to sample datasets of our interest (**North Rupununi Wetlands**), contribute on data collection and data validation, and to make custom visualization on these data and get used of some analysis tools offered by a **desktop web-based application**.

The web-application will offer maps of dataset with the ability of the manipulation of the data and to add comment to database. also, the web-application will allow user to access and find information about coordinate regarding to dataset analysis to be displayed as (Bar Chart, Pie Chart, Histogram/ Land use, land Cover maps/ Coordinates, Contour).

## Domain analysis

### **Members:**

The system allows member to visualize data in map and tables form.

The member is an expert user who uses the web-application concerning analyzing the geographical aspects, add new data that collected by him and add comment if it is needed.

Beside visualizing data, the system offers the member adding new data to the dataset and commenting.

## Phenomena

### **Relevant Phenomena:**

In this section we discuss the phenomena's that the machine cannot observe "**World Phenomena**", phenomena located entirely in the machine "**Machine Phenomena**" and phenomena's that shared among the two.

### **World Phenomena:**

1. Collecting data that users can add to database.

### **Machine Phenomena:**

1. Storing added data by the user to the database.
2. Storing the registration data into database.
3. Storing the comment to the database.

### Shared Phenomena:

1. Opening the web-application/ **controlled by world.**
2. The application asks members for username and password / **controlled by machine.**
3. The member enters the username and password/ **controlled by the world.**
4. The software retrieves the data from the data base and visualize it to the users/members/ **controlled by the machine.**
5. The member adds his collected data to the software/ **controlled by the world.**
6. The member adds comment to the software/ **controlled by the world.**
7. Map's visualization (Base maps with coordinate, attribute tables)/ **controlled by the world.**
8. Visualize analysis (watershed map, streams maps, contours maps- Bar Chart, Pie Chart)/ **controlled by the world.**

## Use Cases

### **UC1: Registration to the website**

Actor: User

Entry Condition: The User opens the website.

Flow:

1. The user selects to create your account.
2. The website opens the registration window.
3. The User enters his username, password repeat password and email.
4. The software checks if the username and email in the database.

Exit Condition:

The software saves the username and password to the database.

Exception:

The software return with "Username exists" if username exists.

The software displayed an "ERROR" message if one of the username or password not entered.

### **UC 2: login.**

Actors: Member.

Entry Condition: The Member opens the web-application.

Flow:

1. The software ask member to enter his username and password.
2. The member submits his username and password.
3. The software check if the username and password are within the database.

Exit Condition:

The software retrieves the home page.

Exception:

If the username or the password are not within the database, the software shows an error message to the member.

### **UC 3: Visualization**

Actors: members.

Entry Conditions:

The Member opens the web application:

- a) The member enters his username and password.

Flow:

1. the web application displays base map for the member.
2. the software offers to the Member information about points on map that contain (coordinates, grass height, etc).

Exit Condition:

The member logout and the user close the website.

### **UC 4: Data Analysis**

Actor: Member.

Entry Condition:

The member enters his username and password.

Flow:

1. The member enters the analysis icon.
2. The software retrieves the data from database.
3. The software provide member with many analysis choices [Statistics, Thematic maps, Topographic maps].
4. The member chooses a method.



5. The software asks the member to choose an option (Bar Chart, Pie Chart, Histogram/ Land use, land Cover maps/ Coordinates, Contour).
6. The member makes a choice.
7. The software displays the analysis information of members selection.

Exist Condition:

The software returns the desired data.

Exceptions:

When the member selects a method with no data then the system returns with an empty page.

### **UC 5: Enter Comment**

The website display data that has been collected from multiple technicians and members of the website.

The member can add comment to validate the data being visualized by the website.

Actor: Member.

Enter condition: The member enters the website using credentials username and password.

Flow:

1. The member enters the website.
2. The member chooses to login.
3. The member enters his username and password.
4. The member selects attribute table.
5. The member pushes the add comment button.
6. The member adds his comment regarding the data he sees on the website.

Exit Condition:

- The software saves the comment to the database.

Constrains:

- The person who can add comment must be a member of the website.

### **UC6: Edit comment:**

Entry condition:

The member login to the website.

Actor: member.

Flow:

1. The member pushes the edit button.
2. The web application allows the member to edit the comment.
3. The member pushes save button.

Exit condition:

The edit to the comment has been saved by the web application.

Constrains:

The title cannot be empty or else there will be an error message.

### **UC7: Delete comment:**

Entry condition:

The member login to the web application

Actor: member

Flow:

1. the member pushes the edit button.
2. the member pushes the delete button.
3. the system deletes the comment from the database.

Exit condition:

The member logs out of the session.

### **UC 8: Add data by members to the database.**

Actors: member

Entry condition:

The member opens the home page.

Flow:

1. the member selects “add data “icon.
2. the software will ask the member to fill:
  - a. Date: optional
  - b. ID: mandatory
  - c. Time: time of the observation: mandatory
  - d. Technician name: mandatory
  - e. Coordinate: mandatory
  - f. Other data information (habitat type, vegetation type, height grass, water source, ...)
3. the member enters his collected data.

Exit condition:

The software stores the input data into database.

Exception:

1. The software does not store the added data if one mandatory column at least is missed, and the software replay an “error” message.
2. The software will not add data if it is not in the right format.

### **UC9: Logout:**

Entry condition:

The member logs in

Actor: member

Flow:

1. The member pushes the logout button
2. The web application confirms if the member wants to logout.
3. The member confirms the logout
4. The system closes the web applications session.

## Requirements and Assumptions

### Requirements:

#### Technical Requirements

1. The dataset is provided by Epicollect5.
2. The software should be coded in python language.
3. The system should be available as web application that can be used throughout the web browser.

#### Non-Functional Requirements:

1. The system should be available for 24/7.
2. The system should return a feedback within a 5 second window.

#### Functional Requirements:

1. The system shall allow members to login in with username and password before opening the web page
2. The system will offer different layers of data with base map to members and visualizing the coordinate by clicking on the point
3. The system shall allow members to interact with analysis graphs such as bar chart by selecting the which section of data (height grass – vegetation type)
4. The software will offer to members to add their opinion of the area and data evaluation by entering comments and displays to other members
5. The system shall allow members to add data to be stored on database in case the members collect all required information.
6. The system will offer maps display as images such as contour maps ... to members so he/she can know the topography of the area.
7. The system shall describe the purpose of software and what is the information that could be found on the web application
8. The software shall allow any to client to register and become a member

## **Domain Assumptions:**

- Area is disregarded if and only if it is not a wetland.
- the quality of the data will depend on the accuracy of data collected by Epicollect5 technicians.

## Efforts:

<b>Ali Badr Eldin Ali Mohammed</b>	12 hours
<b>Nizar Osman Yousif Fadl Elseed</b>	12 hours
<b>Nshwan Ali Abdalla Khairalla</b>	12 hours
<b>Mazin Ali Hassan Saeed</b>	12 hours
<b>Hussamalden Mohammed Shereif</b>	12 hours

## References:

- Epicollect5 website

<https://five.epicollect.net/project/detect-nr-groundtruth2020>

- Wetland's initiative website

<https://www.wetlands-initiative.org/what-is-a-wetland>