23/06/2020

A picture containing food

Description automatically generated

RASD Document

AIMS Web Application

Tree Inventory

Ver.3

BY :

* SOGORE Abdoul Kassim
* MUCUCETE Firmino Manuel
* INDUSEKHAR Ravilla

Table of content

[**I.** **Project scope and goals** 2](#_Toc44465424)

[**II.** **Domain analysis** 3](#_Toc44465425)

[**III.** **World and Machine Phenomena Scheme** 4](#_Toc44465426)

[**IV.** **Use cases** 6](#_Toc44465427)

[**1.** **Registration** 6](#_Toc44465428)

[**2.** **Login** 7](#_Toc44465429)

[**3.** **Log-out** 8](#_Toc44465430)

[**4.** **Post data** 8](#_Toc44465431)

[**5.** **Leave Comments** 9](#_Toc44465432)

[**6.** **Modify comments** 10](#_Toc44465433)

[**7.** **Delete/Remove comments** 10](#_Toc44465434)

[**8.** **View Data** 11](#_Toc44465435)

[**9.** **Download Data** 12](#_Toc44465436)

[**V.** **Requirements:** 13](#_Toc44465437)

[**VI.** **Effort Spent:** 14](#_Toc44465438)

# **Project scope and goals**

A **tree inventory** is the gathering of accurate information on the health and diversity of the community forest. How many street trees are there? What kind? In what condition are they? You cannot manage the community forest effectively unless you know its condition. Tree inventories are an essential tool of good management.

During this project which is done by a group of Geoinformatics Engineering Students of Politecnico di Milano, after acquisition of data of tree inventory by the "Nature Museum Teens" available on [Epicollect website](https://five.epicollect.net/project/asm-su19-trees/data), these data with 181 entries need to be treated in order to understand, and use them. So, we are required to design a web application which follows these **goals**:

* Processing and exposing on the web the data using some original manipulation strategy, by leveraging both the geographic content (map-based view) as well as attributes (interactive exploratory graphs)
* Allowing users to extract custom views of the data, leave comments, and eventually discover how to contribute to the data collection.

# **Domain analysis**

"AIMS" web application could be used by every farmer to see the organization used by other farmers who succeeded.

On the other side, Municipalities could use it to perform better in terms of urban management and know better about the city’s vegetation health.

Our software will not interact with other software of hardware the whole manipulations will be done online by the users.

This web application can be used for surveying, monitoring or as an exchange information platform. To preserve the relevance of the web site each user will be force to sign in so that his identity will be controlled by anybody. As web master we will be allowed to remove any comments, data of other users as we want. The web site, the representation of raw data will entirely be performed by using python so many libraries are needed such as shapely, pandas/geopandas for the representation of data; psycopg2 which is the python SQL version. The web site will be implemented by using Flask.

# **World and Machine Phenomena Scheme**

WORLD

MACHINE

Sign up, Login,

Upload, Download

Delete, Comments

Trees Location

Trees Address

Name of the

Trees

Database queries

Visualization of data by Map View

Vegetation

Condition of the Tree’s

**World Phenomena:**

1. **Vegetation:**

The Plantation of the trees are done by the Municipal Department of the District.

1. **Condition of the Trees:**

This refers to the, how healthy are the trees, their diameter and physical condition. It also taken care and provided by a team from the Municipal department of the district.

**Shared Phenomena:**

1. SignUp:

Process each new user will do by filling the fields with his data.

1. Login:

The process user will do to go to their home page after they signed-up (If user is new) or the process user will do to go to their Home Page (If user already registered)

1. Upload, Download, Delete, Comments:

Processes user can do after they signed up and Logged in to our Web App

1. Trees Location:

Exact position of the trees in a specific reference system

1. Trees Address: The address of the trees is shown
2. Name of the Tree: The name of the tree is shown.

**Machine Phenomena:**

1. Data Base Queries:

To Retrieve the data from the database (User Data, Trees Data)

1. Visualization of Data by Map View:

The data of the trees is shown on an OSM base Map with an Interactive View.

# **Use cases**

In the following section we are going to discuss different general scenarios of the use of the system

**The following use case is related only to a non-registered user**

## **Registration**

* **Actors:**

This UC concerns every new user of our website, who will register through an email address as username and a password. As an administrator we will have a special password.

* **Entry condition:**

The user is not registered and is new to the system.

* **Flow of the Process:**

1. User goes to registration page
2. System opens the registration phase and establish connection with database
3. User inserts required data
4. User finishes with the operation

* **Exit condition:**

The system returns the result of execution of the operation

* **Exceptions:**

1. User Enters Wrong Data such as his Email Address
2. The Failure of the system during the process

**The following use cases are related only to a registered user**

## **Login**

* **Actors:**

This UC concerns every user has already registered. They should enter their email address as username and their passwords.

* **Entry condition:**

The User should be Already Registered

* **Flow of the Process:**

1. The User Clicks “Login” Button
2. The System Open the Login Page
3. The user Inserts his Credentials
4. Then the user clicks Confirm to login
5. The System verifies the entered data

* **Exit condition:**

1. The system after confirming the data opens the user’s private area of the Web application.

* **Exceptions:**

1. The User Inserts Wrong Credentials or the System Fails

## **Log-out**

* **Actors:** This UC concerns every user already logged in.
* **Entry condition:** 
  + The user is logged in so he/she will click on a log out button, then he/she will be logged out his/her account.
* **Flow of the Process:**

1. The User clicks the Logout Option
2. The system then confirms the logout

* **Exit condition:**

1. The system returns to the Web App main page where the user can Log in Again.

* **Exception:**

1. Failure of the system

## **Post data**

* **Actors:** This UC concerns every user logged in users who wants to post something.
* **Entry condition:**

1. The user should be logged in to post the Data.

* **Flow of the Process:**

1. User Logs in
2. Users click on Post data button
3. System opens the page where the user can upload/ add data
4. User clicks confirm option
5. System uploads Data

* **Exit condition:**

1. System confirms the Data has been uploaded

* **Exceptions:**

1. User Not Logged in
2. The user did not upload the data in correct format
3. The system Failure

## **Leave Comments**

* **Actor:** This UC concerns every logged in user.
* **Entry condition:**

1. The User is Registered and Logged in

* **Flow of the Process:**

1. The user Logs in
2. User visualizes the data
3. User clicks on the “Add Comment” button
4. User writes the Comments
5. Users submit the comments
6. System uploads the comments

* **Exit condition:**

1. The system confirms the user that the operation has been successful.

* **Exceptions:**

1. The user not Logged in
2. The Comment field is Empty
3. System Failure during the upload process

* **Special requirement:**

The comments should not be Rude or Un-Polite or else the system administrator will remove the comments if there are any in comments in such a way

## **Modify comments**

* **Actor :** This UC concerns every logged in users.
* **Entry condition:**

1. User is Registered and Logged in
2. User Has Already posted a comment

* **Flow of the Process:**

1. User Logs in
2. User views the data
3. User selects the comment and clicks the “Modify Comment” button
4. The user makes the changes to the selected comment
5. User submits the modified comment
6. System uploads and shows the new modified comment in the place of previous one.

* **Exit Condition:**

1. System returns the result of the operation

* **Exceptions:**

1. The user not logged in
2. The comment inserted is empty
3. System failure

## **Delete/Remove comments**

* **Actor:**

This UC concerns every logged in users

* **Entry Condition:**

1. User is Registered and Logged in
2. User has already Posted a Comment

* **Flow of the Process:**

1. User Clicks the Remove Comment Option
2. User Selects the Comment he wants to remove
3. User clicks the Confirm Button
4. The system confirms the End of the operation

* **Exit Condition:**

The system confirms that the Comment has been successfully Removed/Deleted

* **Exception:**

1. User is Not Logged in
2. There are no comments posted
3. System failure

* **Special Requirements:**

The Posted Comments should not be rude or hurting in any way or else the comments will be removed automatically by the system administrator

## **View Data**

* **Actor :** This UC concerns every logged in user who wants to see the charts, graphs, and maps etc. .
* **Entry condition:**

1. The user should be Logged in

* **Flow of the Process:**

1. The User Log in
2. The user clicks on the View Data
3. The user then clicks on the type of view he wants to visualize the data, such as, Chart, graph, Map views
4. The user clicks the specific view Button and confirms
5. The system confirms the View option selected

* **Exit condition:**

1. The system shows the data in the selected View format that has been selected by the user.

* **Exception:**

1. The user is not Logged in
2. No data is available to show in particular format
3. System Failure

* **Special Requirements:**

1. The user should be able to have the knowledge about how to understand the graphs and chart data view

## **Download Data**

* **Actor :** This UC concerns every user logged in users who want to download data of a chart.
* **Entry condition:**

1. The user should be Logged in

* **Flow of the Process:**

1. User Clicks on the “Download Data”
2. User selects the type of data he wants to download.
3. User confirms the selected option
4. The system finishes the process

* **Exit Condition:**

1. The system confirms the download process by downloading the data to the user computer or mobile

* **Special requirement:**

1. The user should be able to know how to view the downloaded data on his computer or mobile device

# **Requirements:**

We can classify the requirements into three categories

1. Technical requirements
2. Functional requirements
3. Non-Functional requirements

**Technical requirements:**

The system should be implemented in:

1. Python
2. HTML, CSS, JS
3. Interface should be Easy
4. Language should be English
5. Database to store the information (PostgreSQL)

**Functional requirements:**

1. The system should allow multiple users to access the services at same time.
2. The system should allow registered users to enter their home page.
3. The system should allow registered users to upload the data, view data, and be able to use other available services.
4. The system shall allow non-registered users to see the about of Web app.

**Non-Functional Requirements:**

1. The Web App should be available 24h/7
2. The data should be updated in the real time when the user uploads data or deletes his data.
3. The web app should be mobile phone compatible

# **Effort Spent:**

* Sogore Abdoul Kassim: Web application implementation, RASD, DD (Comments on RASD and DD)
* Mucucete Firmino Manuel: Web application implementation, RASD, DD (Comments on RASD and DD)
* Indusekhar Ravilla: RASD and DD implementation, web application (Web App Design View comments)