07/05/220

A picture containing food

Description automatically generated

RASD Document

AIMS Web Application

Tree Inventory

Ver.2

BY :

* SOGORE Abdoul Kassim
* MUCUCETE Firmino Manuel
* ATAEIGHALEHGHASEMI Ahmad
* INDUSEKHAR Ravilla

Table of content

[I. Project scope and goals 2](#_Toc39712802)

[II. Domain analysis 3](#_Toc39712803)

[III. Relevant phenomena 3](#_Toc39712804)

[IV. Use cases 4](#_Toc39712805)

[1. welcome and explanation message 4](#_Toc39712806)

[2. Registration 4](#_Toc39712807)

[3. Login 5](#_Toc39712808)

[4. Log-out 6](#_Toc39712809)

[5. Load some data 6](#_Toc39712810)

[6. Remove some data 7](#_Toc39712811)

[7. Select data and how to show them (chart, map…) 8](#_Toc39712812)

[8. Leave and modify some comments 8](#_Toc39712813)

[9. Download chart 9](#_Toc39712814)

[10. Choose the kind of chart 9](#_Toc39712815)

[V. Requirements and domain assumptions 10](#_Toc39712816)

# **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 through the use of 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 farmers in order 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 cities 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. In order 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 library 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.

# **Relevant phenomena**

The software will totally be dependent on the data from the web site Epicollect. Indeed the quality of the data are published on the website the quality are the charts produced.

R-REQUIRMENTS

WORLD

MACHINE

Density of vegetation

Different types of trees

Distribution of trees

How healthy are the trees

Database queries

Classification of trees

Base Map

Projection system

# **Use cases**

## **Welcome and a short description**

* **Actor**

A message will explain the purpose of the website, how the upload data and perform graphs. It will appear even for guest users of the website.

* **Entry condition**

This use case will appear on the home page of the web site, it’s particularly important for new users of the web site because as administrator we should communicate with the users.

* **Flow of events**

A formal message from administrator is shown on the home page of website, the message could only be modified by the administrator.

* **Exit condition**

This message will appear only on the home page.

* **Exceptions**

There are no real exceptions, it’s only a « print » of a string.

* **Special requirements**

In practice, this use case will be particularly easy to implemented.

## **Registration**

* **Actor**

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 has no account so he/she will click on the register button, then he/she will redirected on a specific registration web page.

* **Exit condition**

After registration the user will normally be redirected on another web page with respect to his/her request.

* **Exceptions**

Only new users are allowed to register, the old ones will receive an error message saying « This email address has registered before, please enter a new email address!».

* **Special requirement**

Users must have a functional email address to register, a confirmation mail will be sent.

## **Login**

* **Actor**

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

* **Entry condition**

Each user has an account so he/she will click on a login button, then he/she will be redirected on his/her dashboard.

* **Exit condition**

After login, the user will normally be redirected on his/her dashboard to post or delete or edit something.

* **Exceptions**

Only registered users are allowed to log in, the new ones will receive an error message saying « This username does not exist!, please register at first ».

* **Special requirement**

Users must enter the correct password, In case the user name doesn’t match with the password, user receives an error message saying « Incorrect password, try again! ».

## **Log-out**

* **Actor**

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.

* **Exit condition**

After logging out, the user will normally be redirected on home web page.

* **Exceptions**

Only logged in users are allowed to log out, the log out button is only visible for them.

* **Special requirement**

Users must have already been logged in, after some inactivity time all users are automatically logged out.

## **Post some data**

* **Actor**

This UC concerns every user logged in users who wants to post something.

* **Entry condition**

For the logged in user , there is a post button on the dashboard which will allow the user to access to a specific web page.

* **Exit condition**

After posting data, the web page will be refreshed, the new posted data will be visible on the web page.

* **Exceptions**

Only logged in users are allowed to post, an error message saying « Session timeout, please log in again!» will appear if they are not.

* **Special requirement**

Users must have already been logged in, the user can interrupt the updating of the data.

## **Remove some data**

* **Actor**

This UC concerns every logged in user.

* **Entry condition**

For the logged in user, who has posted once at least there is a remove button on the dashboard which will allow the user to remove selected data.

* **Exit condition**

After removing the selected content, the dashboard will be refreshed and the removed data will no longer be visible on the web page.

* **Exceptions**

Only logged in users are allowed to remove, an error message saying « Session timeout, please log in again!» will appear if they are not.

* **Special requirement**

The remove button is just available for the users who has already posted something before. So, posting something is an initial requirement to remove it.

## **Select data and how to show them (chart, graph, map etc. )**

* **Actor**

This UC concerns every logged in user who wants to see the charts, graphs and maps etc. .

* **Entry condition**

The user is logged in, there is select and show button in the browser which will allow the user choose some data and how to represent them.

* **Exit condition**

After representing data the web page will still show the last represented thing.

* **Exceptions**

Only logged in users are allowed to select, an error message is printed if they are not.

* **Special requirement**

Users must have already been logged in, the user must select some data before representing them.

## **Leave and modify some comments**

* **Actor**

This UC concerns every logged in users who want to comment on a post or modify a comment.

* **Entry condition**

The user is logged in, there is a comment section for each post.

* **Exit condition**

After submitting a comment, the web page will be refreshed.

* **Exceptions**

Only logged in users are allowed to leave or modify a comment, an error message is printed if they are not.

* **Special requirement**

The administrator can remove a comment, he will have priority. Normal users can only modify their own comments.

## **Download chart**

* **Actor**

This UC concerns every user logged in users who want to download data of a chart.

* **Entry condition**

The logged in user, can download the shown charts on every html web page.

* **Exit condition**

User remains on the last web page.

* **Exceptions**

Only logged in users are allowed to download, an error message is printed if they are not

* **Special requirement**

No real special requirement is needed.

## **Choose type of the charts**

* **Actor**

This UC concerns every logged in users who want to change type of the charts.

* **Entry condition**

The user will able to choose type of the chart after selecting some data.

* **Exit condition**

After choosing type of the chart, data will be represented on a new web page.

* **Exceptions**

Only logged in users are allowed to change type of the charts, an error message is printed if they are not. Some data could be represent only in specific types of charts.

* **Special requirement**

The selected data should be able to be shown in form of the selected type of chart.

# Requirements and domain assumptions

* every logged in user should have permission to add or remove some data and leave or modify comments.
* Every new user must be able to create a new account.
* The user's id must be public.
* The users shall be able to draw a lot of basic figures.
* Every manipulation must be done through the web site.
* Few users must be able to use the web site at the same time.