RASD Document():

{

CLEAN THE WORLD;

(Web application)

}

By():

{

Ahmad Ataeighalehghasemi

},

Supervisors():

{

Prof. Di Nitto,

Daniele Oxoli

}

What is Clean The World?

Clean the world is a project done by Ahmad Ataeighalehghasemi, MSc student of Politecnico di Milano. This project is a web app which gather data about litters in our cities from users and Organizations, NGOs and individuals could use these data to remove the litters and make cities cleaner.

Project Goals and Scopes

According to Wikipedia,
"Litter consists of waste products that have been discarded incorrectly, without consent, at an unsuitable location. Litter can also be used as a verb; to litter means to drop and leave objects, often man-made, such as aluminum cans, paper cups, fast food wrappers, cardboard boxes or plastic bottles on the ground, and leave them there indefinitely or for other people to dispose of as opposed to disposing of them correctly."

Litter can have a detrimental impact on humans and the environment in different ways; effect on humans and wildlife are the most important ones.

Now, "Clean The World" is here to decrease these impacts as much as it's possible by means of a public contribution!

Main goals of the project are:

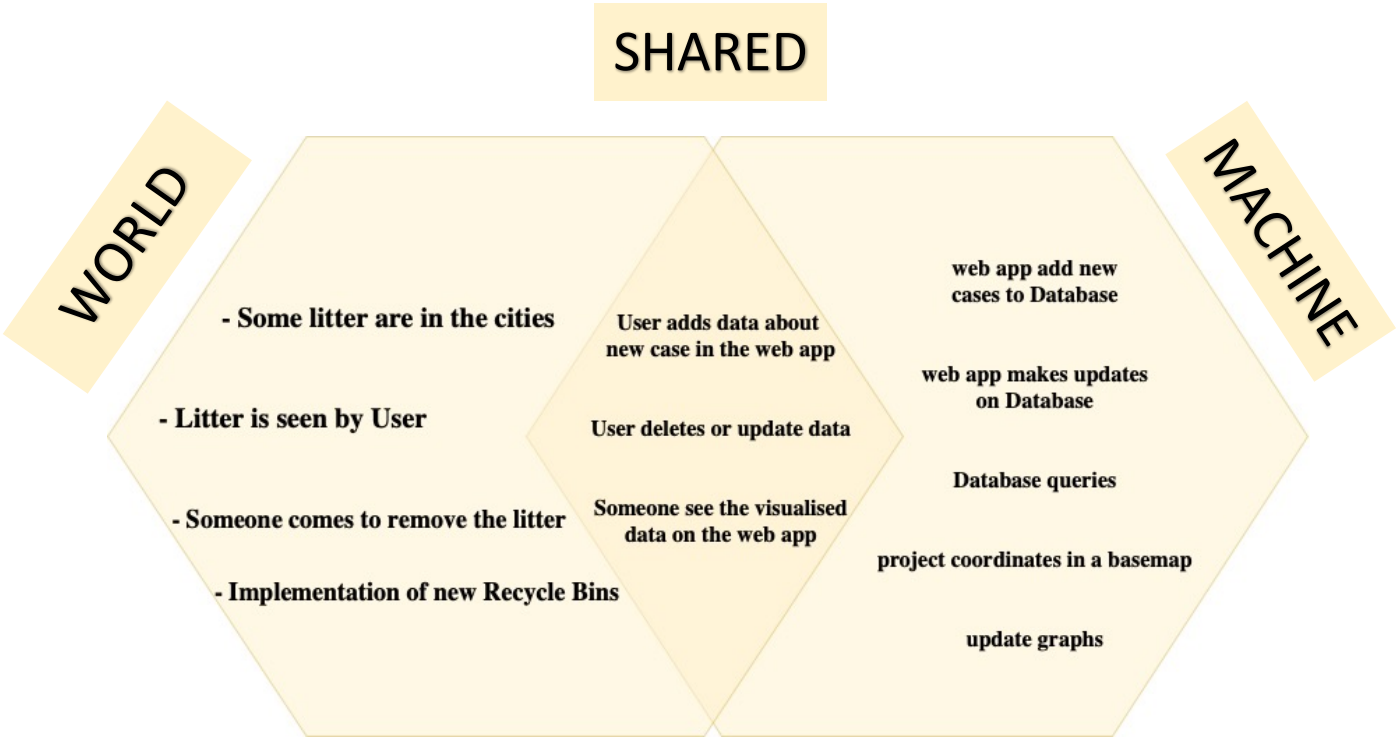
- Collect data(Coordinates, kind of litter and etc.) about litters in the cities
- Visualize the data collected in the Database on the web app
- Let everyone to see the location of litters to remove them

Domain Analysis

"Clean the World" web app can be used by everyone who is just connected to the internet. Individuals can reach the HomePage of the web app by entering web app URL in their browser.
If someone wishes to contribute with the web app, he/she should SignUp using a unique username and password. After Signing Up, it is possible to SIgnIn and Add new cases. It is mandatory to enter Latitude, Longitude and kind of the litter for each case.
On the other hand; different organizations, governments, NGOs and environmental friendly individuals could easily access to these data and help each other to remove them or put some bins in the most vulnerable areas.
Each Signed In user can update and delete just cases which he/she has added before. The representation of raw data will entirely be performed by using python so many library are needed such as shapely, pandas, geopandas, psycopg2 and Flask. There are some pages which uses JavaScript.

Relevant Phenomena

In this project, as other software project there are different relevant phenomena. Some of them are located entirely in the “WORLD”, some in the “MACHINE” and some are “SHARED” between machine and world.



Use Cases

1. Home Page preview

Actors:
Web app, Users

Entry condition:
True. Always home page and description of the web app is shown for every user.

Flow of events:
A user enters the web app URL in the browser and reaches home page. There is a short description of the web app and also some contextual data about litter cases.
Users are allowed to see the graph and other visualizations.
Users can SignUp and SignIn to add new cases.
Users may visit About and Contact pages.

Exit condition:
According to action taken by user, SignUp page, SignIn page, About page, Contact page and visualized data pages can be reached.

Exceptions:
There is no exception for this use case.

Special requirements:
There is no specific requirement for this use case.

2. SignUp to the web app

Actors:
Web app, Users

Entry condition:
User clicks the SignUp button and the dedicated page appears.

Flow of events:
After clicking on the SignUp button on the Home page, SignUp page appears.
Users should enter a unique username and password to SignUp.
After filling the fields, there is a green SignUp button to submit the registration.
A new user added to Database.

Exit condition:
SignIn page appears for new signed up user.

Exceptions:
In case the entered username by user is already signed up by another user, system shows an error to try another username.

Special requirements:
Each username should be unique.
Both username and password should be entered.

3. SignIn to the web app

Actors:
Web app, Users

Entry condition:
User clicks the SignIn button and the dedicated page appears.
Also after signing up, SignIn page appears for new users.

Flow of events:
After clicking on the SignIn button on the Home page or signing up by new user, SignIn page appears.
Users should enter his/her username and password to SignIn.
After filling the fields, there is a green SignIn button to sign in into the web app.
When user signs in, he/she could add new case or modify cases which has entered before.
Changes made by signed in user, update the Database.
User can log out after adding or modifying data

Exit condition:
User to redirects Home page with option to Add new case.

Exceptions:
Only Signed up users can Sign in to the web app.
User just can modify cases entered by him/herself.

Special requirements:
Signing up before signing in is required.
Each user should remember his/her username and password.

4. Add new case

Actors:
Web app, Users

Entry condition:
A signed in user is on the Home page and wants to Add new case(s)

Flow of events:
For signed in users, there is a button on the Home page to Add new case.
Users by pressing the Add button, redirect to a new page.
User should enter Latitude, Longitude and Kind of seen litter.
After filling the required fields, user press Add button.
A new case is added to Database and contextual information of new added case appears on the Home page.
User can remove or update data of each case which has added before.

Exit condition:
Home page appears with new added case by the user with remove and update options.

Exceptions:
Only signed in users can Add new case(s).

Special requirements:
Sign in before add new case is required.
All the fields should be filled by user.
Users just are able to delete or modify cases entered by themselves.

5. Modify added cases

Actors:
Web app, Users

Entry condition:
A signed in user is on the Home page and wants to update some data about cases that he/she has entered before or wants to delete the case completely.

Flow of events:
For signed in users, there is a button for each case added by themselves.
Users by pressing the update button, redirect to a new page which allows user to change some data about the case.
When user presses the remove button, a message appears and asks user if he/she is sure to remove the submitted case.
In case of update, after changing the required fields, user press Update button to make changes.
New updated data make changes in Database and contextual information of the case updates on the Home page.
In case of remove, after pressing yes button of the alert message, the case removes from Database and Home page as well.

Exit condition:
Home page appears with updated data of the modified case or without the removed case.

Exceptions:
Users just are able update or remove cases entered by themselves.

Special requirements:
Sign in before modifying is required.
In case of update, all the fields should be filled by user.

6. Logout from account

Actors:
Web app, Users

Entry condition:
A signed in user is on the Home page and wants exit the web app or wants to prevent undesired changes.

Flow of events:
For signed in users, there is possibility to log out if they want to leave the web app or don’t want to make more changes.
User clicks on the Logout button on the Home page.
A new Home page appears with SignUp and SignIn possibilities.

Exit condition:
New Home page appears with SignUp and SignIn possibilities.

Exceptions:
Just signed in users could logout.

Special requirements:
Sign in before log out is required.

7. Exploring visualized data

Actors:
Web app, Users

Entry condition:
A user on the home page has two options, watch bar graph of quantity of each kind of litter or see the litter locations on the map.

Flow of events:
Each user could press Graph button or Map button to explore the visualized data.
After clicking each button, user redirects to dedicated page.
Locations of the litters and also Number of each kind of litter is available by means of bar graph and map.

Exit condition:
User has different choice, Home page, About page, Contact page, SignUp and SignIn pages.

Exceptions:
Signed in users couldn’t see SignUp and SignIn buttons but Add button.

Special requirements:
No specific requirement is needed.

Requirements and domain assumptions

- Every user who just visits the web app should be able to see contextual data, graphs and visualized data.
- Every user who wants to add new cases, should sign in.
- Every user who wants to sign in, should sign up to the web app using his/her unique username and password.
- Every signed in user should be able to add new case and also update or delete the previous cases added by him/herself.
- Every user who wants to delete a case, should be altered by a message.
- All the required fields should be filled by signed in users to commit the changes in the Database.
- Log out is not mandatory to commit the changes.