SW Engineering CSC648/848 Summer 2019

"The Better City"

Team: 04

Team Members:

Maithri Chullikana House: Team Leader   Email: mchullikanahouse@mail.sfsu.edu
Soham Harshadbhai Prajapati: Front-End Leader
Patreck Benjamin Raymore: Back-End Leader
Bahar Moattar: GitHub Master, Document Editor
Ziping Huang: Back-End Developer
Aashutosh Bajgain: Back-End Developer
Fengze You: Front-End Developer
Zeyuan Cai: Fron-End Developer

Milestone: 1
Date: 07/01/2019

Revision: Final Draft (1.5)

1, Executive summary:

We believe everyone is needed in our attempt to preserve our mother nature. Our world is getting to a place that environmental issues are becoming a big part of everyone's daily life, if it's not already. We are all concerned about the future of our planet if these issues continue and never get fixed or resolved. Our application helps and inspires people to take action on behalf of our shared public parks/environment. Together, our passion for nature can make a difference.

On the way of having a better future for our city and its nature, we made the "TheBetterCity" application that provides a web-based service for allowing people to post and view environmental problems in their neighborhoods. It's also a great tool for the environmental services companies to monitor and assign work for these issues and get them fixed.
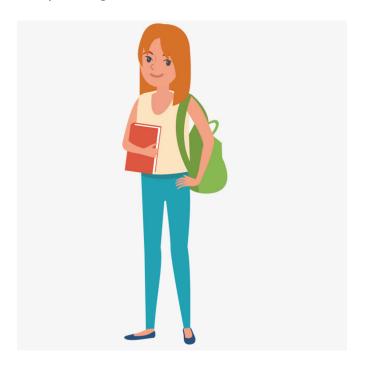
This site will allow people to see environmental concerns, report them, and follow their status in the parks of their interest. Our Users can post these reports in our website, prioritize them, follow up with them, make comments, and rate their concerns based on severity. These reports raised by people will help the city, and the environmental managers who have access to the website, to check on the reports and manage the workflow by assigning the status to the posted issues.

This website makes the reporting environmental issues to the right authority much easier. We provide easy access to the user interface. Users can search and filter environmental issues in our application which can't be done in any other websites.

We at "TheBetterCity" are a team of 8 computer science students at the SanFrancisco State University. We are all working together to get this project done and do our job to make a difference and a better future for our nature.

2. Personas and main Use Cases:

Emily: College student and a member of an environmental club

About Emily:

- Emily is 20 years old college student and has a part time job.
- She would volunteer for environmental related organizations.
- She likes to go to national parks and hike.
- She likes taking pictures of nature and post them on the internet.
- She loves nature and passionate about the safety of the environment.
- She would report environmental issues if she had seen any.
- Knowledgeable with technology and the internet.
- Impatient and hates slow loading websites.

Goals and Scenario:

- Emily went hiking in a national park and suddenly saw the roads were not safe to travel which needed to be fixed.
- She wants to find a way to report this to the specialist and get the problem fixed and make this place safer for people to hike.
- She thinks maybe she can take a picture of this issue and finds a way on the internet to report it to the workers and specialists in this area.

Jessica: Mother of 2 children and a teacher in an elementary school

About Jessica:

- Jessica is 38 years old.
- Very busy taking care of her kids and her job as a teacher.
- She loves to take her kids and her students to the parks and explore nature.
- She is so environmentally conscious and reports any issues she finds.
- She pays attention to education and always try to teach her students about protecting the environment.
- She is also a volunteer to be responsible for different kinds of environmental protection.
- She has basic technology and internet skills.
- Impatient and hates slow loading websites.

Goals and Scenario:

- Jessica has decided to take her students to the park for a field trip and has to make sure the park is safe and it's without any environmental issues.
- If she finds any problems with the park, she wants a convenient way to report the issue and have it resolved as soon as possible.

David: Environmental worker for a big company

About David:

- David is 51 Years old, married, and father of 3 children.
- He works for a company involved with monitoring and fixing environmental issues.
- He loves his job and he is very busy with it.
- He browses the internet all the time and looking for what people have been reported about these issues.
- He supervises a team of workers and they try to fix the issues in the location based on the reports.
- He has proficient skills with technology and the internet.
- He is so patient with web applications since there is so much reports for him to handle.

Goals and Scenario:

- David is currently at work and have to monitor environmental related issues that are reported by people.
- He has to prioritize which environmental issues has to be fixed first and set the status to show that they are currently working on it.
- He would then organize a team of workers to fix these issues.

Main Use Cases:

Emily wants to go hiking today. She checks for the nearest park in our website for any serious environmental issues. She goes to the park and she sees that the trash is overflowing. She goes to our website and searches the park by the name, and then she will try to post the issue with a photo from the situation and a small description. Emily chooses the appropriate function from our website and writes about the issue. Then she is prompted to sign up before she can submit her write up.

Jessica took her two children to the seaside for a holiday. There were so many people at the beach. It was a very busy day, but they had so much fun. After a while, she found that there were many plastics and trash left by tourists on the beach such as drinking cans, empty bottle of waters, and discarded swimming equipment, which polluted the beach and the water, makes it so ugly. Later when she gets home, she decided to find an application and reports this environmental problem. So she browsed the internet, found our website the best source to do this, and submitted the report to follow up with it until it gets fixed.

David is a senior manager of our '*TheBetterCity*' company. He goes to our portal and login as a system administrator with his credentials to see if there's any issue reported. Once David logs in, he will be able to see all the reported issues including what Emily and Jessica reported yesterday. He can also filter these reports based on their status. For example, currently he wants to check only on newly reported issues, so he filters the list by the "Open" status. David will also have an ability to assign issues to respective department if it requires immediate attention. When the problem is resolved by the team members, he will change the status and mark it as "Solved".

3. List of Main Data Items and Entities:

- Users
  - Name
  - email
  - Password
  - Id
  - Photo
  - Issues

The user's entity will be used to store the information for all individuals who logged into the site and wish to post an Issue. Upon registration each user must input a Name, valid email, and password. However, the user is not required to upload a photograph. The issues data item is used to store all the past issues that the user had reported.

- Admins (i.e. City Manager/worker)
  - Name
  - Email
  - city
  - Password
  - Id
  - Photo
  - Issues

The Admin entity is used to store all users of the application who have the role of Admin. The admin has a similar registration process as a regular user. However, after registering they must be approved by another admin. Although the issues data field is named similar to those of regular users, it serves a different function. This data item is used to store the ID of all the issues that the admin is tracking. And it's only these issues that the admin can change the status of.

- Issues
  - Photo(s)
  - Location
  - Status (e.g. open/in progress/solved)
  - Description
  - Reporter (i.e. user name)
  - Timestamp

The Issues entity will be used to store each issue reported by a user. Each issue must have at least one photo but can also have multiple (i.e. up to five). The location for each issue can be inputted manually or gathered via geo-location. The status of the issue can be toggled between three states: open, In-progress, and solved. However, the state of the status can only be altered by a user with the role "admin". Furthermore, each issue must have a description in order to be submitted. The time stamp will be used to keep track of how long each issue has been open.

4. Initial list of functional requirements:

Unregistered Users

1. Unregistered users shall be able to register an account.
2. Unregistered users shall be able to view postings/reports.
3. Unregistered users shall be able to browse through the issues.
4. Unregistered users shall be able to see the current status of any reported issue.
5. Unregistered users shall be able to choose the park or environment they want to see the report for.
6. Unregistered users shall be able to see the ratings of a report.

Registered Users

7. + Unregistered users' functions
8. Registered users shall be able to log into website
9. Registered users shall be able to search reported issues.
10. Registered users shall be able to report environmental issues.
11. Registered users shall be able to edit the reported issues.
12. Registered users shall be able to post comments on the reports.
13. Registered users shall be able to filter issues by categories, ratings, and status.
14. Registered users shall be able to rate reports based on their priority.
15. Registered users shall be able to geotag.
16. Registered users shall be able to upload a photo to their report to show more details.
17. Registered users shall be able to upload a video related to their reports.

Admins

18. Admin shall be able to publish a report.
19. Admin shall be able to change the status of a report.
20. Admin shall be able to log in, by using the organization email and password.
21. Admin shall be able to assign tasks to their staff to get the issues resolved.
22. Admin shall be able to manage the user database. e.g Admin shall be able to remove duplicate or suspicious accounts.
23. Admin shall be able to edit/delete reported issues. e.g Admin shall be able to delete duplicate or irrelevant reports.

5. List of non-functional requirements:

- The application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team, but all the tools and servers have to be approved by class CTO).
- The application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers
- Selected application functions must render well on mobile devices
- Data shall be stored in the team's chosen database technology on the team's deployment server.
- No more than 50 concurrent users shall be accessing the application at any time
- Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.
- The language used shall be English.
- The application shall be very easy to use and intuitive.
- Google analytics shall be added
- No email clients shall be allowed
- Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated.
- Site security: basic best practices shall be applied (as covered in the class)
- Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
- The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Summer 2019. For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application).

6. Competitive Analysis:

Feature:	Competitor A (The Wilderness Society):	Competitor B(IVAN Online):	Competitor C(EPA US Environmental Protection Agency):	Our Product:
User Report Issue(text, pictures, video)	+	+	++	++
User Filter Issue	-	-	-	+
Rate Issue based on Priority	-	-	-	+
Geotag where the issue happened	+	-	-	++
• + feature exists , ++ superior , - did not exist				

There are 3 main competitors on the market for our product (A: The Wilderness Society, B: IVAN Online and C: EPA US Environmental Protection Agency). Based on the key features that we have listed on the table, none of them are better compared to our product. For this product , the most important feature would be the Issue report function. A, B competitors do have the feature, but they cannot upload any images or video. C competitors have the feature to upload image and video, but before they do so, there will be a long login, verification process. Our product on the other hand give users easy access to report an issue with text, image and video. The next features which are filter issue and rate issue based on priority, for these two features, none of these competitors have these features. Our product does take these two features serious, our team felt being able to rate issue based on priority is very important feature and it allows the admin or user to know the priority of each issue and be able to take action as soon as possible. Geotag was any key feature for our product, Competitor B and C did not have that feature. Compare the geotag feature between A and our product, the advantage that we have would be the convenience of putting Geotag. Competitor A allows users to insert the location by adding text. Our product would allow users to get their reported issue location by using their GPS in their phone. Each competitor has their own specialty but based on the features that each of them get, our team believes that our product has advantages on the market.

7. High-level system architecture and technologies used:

Frameworks
- Express.js
- Handlebars
- Bootstrap

Database
- MySQL

APIs
- Google Maps Platform: Geolocation API

Tools
- Postman
- MySQL Workbench
- Nodemon
- Git & Github

Supported browsers
- Google Chrome
- Firefox
- Internet explorer

Deployment Platform
- AWS EC2
- Ubuntu 18.04

8. Team:

Maithri Chullikana House: Team leader
Patreck Benjamin Raymore: Back-end leader
Soham Harshadbhai Prajapati: Front-end leader
Bahar Moattar: GitHub master, document editor
Ziping Huang: Back-end developer
Aashutosh Bajgain: Back-end developer
Fengze You: Front-end developer
Zeyuan Cai: Front-end developer

9. Checklist:

•Team found a time slot to meet outside of the class: DONE
•Github master chosen :DONE
•Team decided and agreed together on using the listed SW tools and deployment server: DONE
•Team ready and able to use the chosen back and front-end frameworks and those who need to learn are working on learning and practicing: ON TRACK
•Team lead ensured that all team members read the final M1 and agree/understand it before submission: DONE
•Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents, etc.): DONE