SW ENGINEERING CSC 648/848 SUMMER 2019 MILESTONE 1

Team 07
Erick Velez (Team Lead, evelez1@mail.sfsu.edu)
Vincent Santos (Backend Lead)
Kevin Paredes (Frontend Lead)
Hector Aguirre (Github Master)
Jackie Shan
Jimmy Kwan

Date	*Revision History*	*Comments*
7/1/2019	*1.0.0*	*Initial Draft*

July 1, 2019

Table of Contents

1. Executive Summary

The public's personal time is important. However they spend it, they would like for it to be perfect. For example, they might spend it by taking a walk at their local park. It can be easily ruined by an unwanted blemish on the scenery. How can someone avoid going to a park that might be filled with trash or flooded from bad plumbing? If the issue happened during an outing, then how and where will a concerned citizen, eager to inform the public on an environmental incident, post and talk about the issue? With information so readily available on the Internet, it seems like this is almost necessary for the average person.

The public would, of course, hold the city government responsible to fix any problems seen. Without a central system for citizens to report such issues, then the citizens might call city hall. Where could a city manager view and organize all issues reported by concerned citizens? If they are called in, then they would probably end up as paper reports on a wall, waiting to be looked at and easy to lose. If a system existed where concerned citizens could report an issue in a public place, then the city would not have to worry about organizing, locating, or labeling any issues that are reported. Instead, they can focus on deploying the necessary resources to fix the problem in a timely fashion.

[NAME] will be developed to address all of these questions, concerns, and benefits. [NAME] will allow the public to view environmental issues that might have occurred in their favorite parks by simply navigating to a website. Once there, they will be able to search their favorite park, zip code, or simple scroll across a map to locate any environmental issues out of curiosity or for their own well-being. Concerned citizens will be allowed to register and post, with a picture and description, about an environmental issue that they come across on their time out. City managers will be able to log-in to view and manage the status of all submitted issues. They will be able to change the status from open to in-progress to finished with access to the history of all issues.

[NAME] will be developed by a small, eager startup team of 6 students. Young and ambitious, they will be sure to deliver a quality product.

2. Personas and Use Cases

Persona: Sofia

- She is a mother
- She loves nature and bonding time with her children
- Environmentally Conscious: Likes to take kids to environmentally safe parks; will report.
- Has basic www skills.

Goals and scenario:

- Decided to go to a park. Wants to quickly check (from home) if park is safe for her kids.
- If she finds a problem in the park she will report it.

Persona: Mary

- She is an environmental activist
- She loves nature and visits multiple parks in her free time.
- Environmentally Conscious: Cares about environmental issues and animal life preservation, will report issues.
- Basic www skills.

Goals and scenario:

- Decided to go to a park and then report the waste management issues.
- She is really concerned about environmental issues and will do allocate all her time to report her concerns.

Persona: David

- He is a professional photographer
- He loves nature and visits multiple parks.
- Environmentally Conscious: Cares about environmental issues and animal life preservation, will report issues.
- Basic www skills.

Goals and scenario:

- Decided to go to a park and then report the waste management issues.
- He is really concerned about environmental issues and will do allocate all his time to report concerns.

Persona: Bill

- He is in charge of environmental issues post
- Environmentally Conscious: Cares about environmental issues being solved and creating revenue for his company.
- Advance www skills.

Goals and scenario:

- Performs daily management on tasks related to the company website.
- Will update posts as soon as issues are solved

Use Case: Mother and Children

Sofia is a mother of two children. She takes her children out to parks to explore nature. She currently uses the internet to find the safest parks available for her children. Therefore, she will be able to open the website and type the name of a park that she would like to visit in the homage. Then the search engine shall open the park information and its details about environmental issues. Sofia will see that the park that she is currently attempting to visit has no major environmental issues, and then she can decide if the park is currently safe to take her children to.

Use Case: Environmental Activist

Mary is an environmental activist, she enjoys visiting parks every weekend, she loves nature and is currently visiting multiple parks around her area. One day, she notices that the park she visited has a waste management issue. Mary finds the website and then she enters the name of the park that she visited. Then, the website will display the details and posts about the park. She decides to create a post for people to be aware of the needs of the park. However, she needs to register to create a post. She decides to register, she needs to enter the information needed to complete the registration and now she will be able to create a post.

Use Case: Photographer

David is a young active nature photographer. He loves nature and is currently visiting multiple parks to capture the best pictures that he could find. Therefore, it is important that the parks he visits are currently well maintained and have no environmental issues. One day, David notices that the park he visited needs some work related to waste management. Therefore, he decides to request maintenance services for the park. David finds the website and then he enters the name of the park that he visited. Then, the page will open a list of comments and details about the park. He has the option Log in or Sign up to create a post. He already has an account, so decides to log in. Then he will be able to create a post for people to be aware of the needs of the park and he can also read previous posts from other users. He decides that the problem has been shared and hopes that someone will take actions of this problem.

Use Case: Administrator

Bill is the website administrator for the environmental website. His current role is to manage the post created by the users. To manage the post, he will use the website as a tool to update the post created by the users. He will be able to log in and have an admin mode interface that allows him to update the list of post created by date, place and status. After contacting the entities that are responsible for the environmental solving issues he will be able to change the

status and create an update post for the users to be informed that an action has been completed about the environmental issue that was posted.

3. List of Main Data Items and Entities

Entities:

- Users
    - Unregistered User
    - Registered User
    - Administrator
- Issues
    - Issue Categories (fire, trash, spill, etc.) with implicit priority

These two entities make the bulk of the application and are the key components in our database

Data Items:

- Registered User Information
    - Name
    - Email
    - Privileges
    - Reference to Open/Ongoing Issues
- Issue Information
    - Location
    - Description
    - Pictures

The app will have a map of the parks and their corresponding Issues. However, the Google Maps API will abstract most of the data items associated with this interface.

4. Initial List of Functional Requirements

1. Users shall be able to view any issue that has been posted.
2. Users shall be able to search using specific locations.
3. Users shall be able to search using zip codes.
4. Users shall be able to search using a visual map.
5. Users shall be able to access any previously completed issues.
6. Users shall be able to register which requires their name and email, making them a Registered user.
7. Users shall be able to log in with valid email and password.
8. Registered user information shall be stored in the database.
9. Registered users shall have the same permissions as unregistered users and more.
10. Registered users shall be able to post issues.
11. Registered users shall be able to edit the issues they created.
12. Issues shall have an optional image attached to the post.
13. Issues shall have multiple statuses for each post (open, in progress, completed).
14. Resolved issues shall be removed from the map.
15. Resolved issues shall be removed from active issues.
16. Resolved issues shall be moved to the completed issues page.
17. Administrators shall have the same permissions as registered users and more.
18. Administrators shall be able to edit the status of an issue.
19. Administrators shall be able to delete issues.
20. Deleted issues shall be inaccessible to any users.

5. List of Non-functional Requirements

- Server Host: Google Compute Engine f1-micro 0.2vCPU 0.6 GB RAM
- Operating System: Ubuntu 18.04 LTS
- Database: MySQL 8.0.16
- Web Server: Apache 2.4.39
- Server-Side Language: JavaScript (Node.js 10.16.0 LTS runtime)
- Application shall be optimized for standard desktop/laptop
- Data shall be stored in the team's chosen database technology on the team's deployment server
- The language used shall be English
- Application shall be very easy to use and intuitive.
- Google Analytics shall be added
- No e-mail clients shall be allowed
- Pay functionality, if any, shall not be implemented or simulated
- Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users
- No more than 50 concurrent users shall be accessing the application at any time
- Browsers
    - Chrome
        - Windows version 75.0.3770.100
        - macOS version 75.0.3770.100
    - Firefox
        - Standard release version 67.0.4
- The website will display *"SFSU Software Engineering Project CSC 648-848, Summer 2019. For Demonstration Only"* at the top of every page.
- Application shall also be able to render in any mobile device such as android and apple users
- For site security, best practices will be applied

6. Competitive Analysis

Feature	echo.epa.gov	calepa.ca.gov	dec.ny.gov	Our future product
Text Search	+	+	+	+
Browse completed issues	+	-	-	+
Posts can contain image	+	+	-	+
Locations of issues presented on a map	-	-	-	++
Submitted post can be edited	-	-	-	+

+ feature exists
++ superior feature
- feature does not exist

The planned advantages of our product will be its superiority in ease of use compared to our competition. It will allow users to browse and search ongoing and past issues in multiple ways. Our product will have two distinguishing features that our competitions do not have. One feature is having a map display all ongoing issues in a particular region. The other unique feature that our product will have is the option to allow our users to edit previous posts that they've made. Our product will be easily navigable and more user friendly compared to our competitions.

7. High-Level Systems Architecture and Technologies Used

1. Google Cloud Platform Compute Engine (Ubuntu VM)
2. Supported browsers: Mozilla Firefox and Google Chrome
3. Node.js
4. MySQL
5. Express.js (Node.js web framework)
6. Bootstrap (frontend library)
7. jQuery (frontend library)
8. Google Maps API
9. Google Places API

8. Team

a. Erick - Team Lead
b. Vinny - Backend Lead
c. Kevin - Frontend Lead
d. Jimmy - Backend team member
e. Jack - Frontend team member
f. Hector - Backend team member and Github Master

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**