**SW Engineering CSC648/848 Section 02 Spring 2018**

**Software Engineering Term Project:**

**Cleansweep Application**

Team Number 11

Brady Helkenn bhelkenn@gmail.com

Jason Guan jasonaung@ymail.com

Dylan Abrames abrames.dylan@gmail.com

Avi Mukherjee aviraj.mukh@gmail.com

Rodolfo Salgado rsalgado@mail.sfsu.edu

Frank Hood hood@mail.sfsu.edu

Milestone 1

2-28-2018

|  |  |
| --- | --- |
| History Table | |
| Document Created | 2-28-2018 |
| Final Draft Submitted | 3-1-2018 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1. Executive Summary

Improve your city parks, clear your roadways, and allow your kids to breathe cleaner air. All this can be accomplished with Cleansweep. Our small team has designed a cross platform application allowing you to report and manage a wide range of environmental issues across your city or even region. Cleansweep is set apart from the rest of the competition by our key functionalities such as ease of use across devices, intuitive interfaces, and reliability.

Our software serves all members of your community, for anyone with access to the internet can report an environmental issue to Cleansweep thereby allowing the city manager to see real time data on critical issues. Unclog water drains, clean up waste, and much more, all before it starts to impact those in your community. Our inbound marketing strategy is to reach out to your community members via social networks, email, and posters located in key areas throughout your community.

Our small team of 6 has over 30 hours of combined experience in developing and deploying software. We care about our community and are dedicated to make it better which is why we are set to make Cleansweep a reality.

2. Use Cases

1. Guest User - Jill

Before traveling to a local park, Jill checks the Cleansweep website to see if any environmental issues exist nearby. On the website she sees a map interface and search functionality. She uses filters to quickly find issues based on location, type, status, or time since reported. She opens an issue report to view **issue location**, **issue type**, and **issue photo** submitted by the Registered User.

2. Registered User - Jack

Jack becomes aware of illegally dumped cleaning chemicals. He goes to the website and is prompted to describe the issue, provide a location, and given the option to include a photo. He is required to log in or register before the report is posted. He bookmarks the report and later logs in to immediately see the report’s status.

3. Admin - Bob

Bob logs in using the website administrator account. He removes content flagged as undesired and deactivates the account of a repeat offender. He is not allowed to edit posted content, only remove.

4. City User - Rachel

Rachel logs in using a city account. She views the new reports and marks each as **received**, **duplicate**, or **undesired**. She updates the system with information from the city by changing the status of reports from received to in progress and from in progress to complete.

3. Data Definition

Admin - privilege to ban user’s IP if certain users violate forum rules

- To delete irrelevant posts or offensive posts

-Change users’ credentials

-And everything that a regular user can do

Unregistered User - View certain posts but not all

- Can’t post, can’t comment

- Can’t store any info to the server

Registered User - View all posts

- Can post new post

- Can comment under posts

-Can store their personal info to the server

Forum post - Shows locations of the event (if location service was enabled)

- Show current status (in progress/resolved/intact)

-Show contents

-Show related users’ comments

Report status: - indicates the current status of the submitted report

-show past reports as well

-city manager has the privilege to change the status of a report

User Registration Data(URD): -contains uploaded user’s information(name, DOB, etc.)

-URD also shows reports that are related to a specific user

4. **FUNCTIONAL REQUIREMENTS**

1. All users shall be able to search for Reports by zip code.
2. All users shall be able to view Reports per a zip code sourced from Google maps.
3. All users shall be able to post images when submitting an environmental incident.
4. All users shall be able to see the status of submitted Reports.
5. Unregistered Users shall be allowed to fill out a Report, but must register to submit the report.
6. Registered Users shall be able to log in using a username and password.
7. Registered Users shall be able to reset their password from the sign-in screen.
8. Registered Users shall be able to save up to five Reports to be displayed while logged in.
9. City Users shall be able to view all Reports submitted by users.
10. City Users and Admins shall be able to view user information attached to submitted Reports.
11. City Users shall be able to adjust the Status of a submitted Report to received, duplicate or undesired depending on the content of the submitted Reported.
12. Admins shall be able to delete submitted Reports.
13. Admins shall not be able to edit submitted Reports.
14. Admins shall be able to deactivate Registered Users’ accounts.

5. **NON-FUNCTIONAL REQUIREMENTS**

1. 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 tools and servers have to be approved by class CTO).
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome.
3. Application shall have responsive UI code so it can be adequately rendered on mobile devices but no mobile native app is to be developed
4. Data shall be stored in the team’s chosen database technology on the team’s deployment server.
5. Application shall be media rich (at minimum contain images and maps)
6. No more than 50 concurrent users shall be accessing the application at any time
7. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
8. The language used shall be English.
9. Application shall be very easy to use and intuitive.
10. Google analytics shall be added
11. No e-mail clients shall be allowed
12. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated.
13. Site security: basic best practices shall be applied (as covered in the class)
14. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
15. The website shall prominently display the following exact text on all pages *"SFSU Software Engineering Project, Spring 2018. For Demonstration Only”* at the top of the WWW page. (Important so as to not confuse this with a real application).

6. Competitive analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | CalEPA | Broward | EPA.gov | Cleansweep |
| Good Instructions | ++ | - | + | ++ |
| Picture Upload | + | + | - | + |
| Maps Integration | + | - | - | + |
| User Profile | - | - | - | + |
| Nice-looking UI | ++ | - | - | + |

Across the market are numerous websites for reporting environmental issues which range in types of features and quality of presentation. Cleansweep, Team 11’s website, will improve on many issues present in other websites. The option to upload a picture is a rarity across the current websites and Cleansweep will have this option from version 1.0. With other websites which only allow an address input, it is difficult to narrow down where an incident occurred, but Cleansweep makes use of map input to record the exact location. The user profile on Cleansweep will allow a user to flag reports and easily check back on those reports’ statuses. Matching the CalEPA website is the goal for user features, ease of use, and a nice-looking UI.

7. High-Level System Architecture

1. Node.js/JavaScript - SSL

2. Express - Framework of Node.js

3. ProgreSQL - DB

4. Latest version of Chrome and Firefox

5. Heroku - Server

8. Team

Brady Helkenn

Position: Team Lead

Oversees general coordination of team primarily through direct contact with the Front-End and Back-End Leads and by ensuring any struggling teammates are helped out.

Jason Guan

Position: Back-End Lead

Maintains a strong understanding of the software stack and coordinates the Back-End development of new functionality.

Dylan Abrames

Position: Front-End Lead

Maintains a strong understanding of the software stack and coordinates the Front-End development of visual design and process of data validation.

Avi Mukherjee

Position: Front-end Teammate, Documentation Assistant

Creates, updates, and maintains code to support the strong visual design and preform data validation in coordination with the Front-end Lead. Provides assistance with documentation as needed.

Rodolfo Salgado

Position: Back-end Teammate

Creates, updates, and maintains code to provide functionality, stability, and security in coordination with the Back-end Lead. Helps train other teammates on code as needed.

Frank Hood

Position: Documentation

Creates, updates, and refines documentation for the team. Assists elsewhere as needed.

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 and working on it:

**DONE**

• Team lead ensured that all team members read the final M1 and agree/understand it before submission:

**DONE**