

SW Engineering CSC648/848 Section 02 Spring 2018

**Software Engineering Term Project:
Cleansweep Application**

Team Number 11 – Local

Brady Helkenn
Jason Guan
Dylan Abrames
Avi Mukherjee
Rodolfo Salgado
Frank Hood

The team is local.

Brady Helkenn bhelkenn@gmail.com

Milestone 2

3-18-2018

History Table	
Submitted for review	3-19-2018
Revised after feedback	

DATA DEFINITIONS

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

- To delete irrelevant posts or offensive posts
- Change users' credentials
- And everything that a regular user can do

Registration – user provide certain info to become part of the community

Username: user picked username

Password: user picked password

Email: User email for notices and password recovery

Security question: For password recovery

Phone number: For password recovery as well as receiving user-followed-issue status

Unregistered User (without user privilege) - View certain posts but not all

- Can't post, can't comment
- Can't store any info to the server

Registered User (with user privilege) - View all posts

- Can post new post
- Can comment under posts
- Can store their personal info to the server

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




Issue - reported issues uploaded by users


- Title: Issue related title like – not enough trash cans in a certain park
- Description: details about this issue, use can include as many details or nothing at all to describe a certain issue. 
- Location: City, State, Longitude and Latitude of the park.
- Status: Current status of this issue, is it being reviewed, is it being worked on, is it uncertain (fake report), or is it resolved.
- Author: User who submitted this issue
- Date: Reported date
- Image: Image of the park.

FUNCTIONAL REQUIREMENTS

Priority 1

1. All users shall be able to search for Reports by zip code.
2. All users shall be able to search for Reports by Park name.
3. All users shall be able to post images  on submitting an environmental incident.
4. All users shall be able to see the status of submitted Reports.
5. All users shall be able to view a park's location through Google Maps.
6. Unregistered Users shall be allowed to fill out a Report, but must register to submit the report.
7. Registered Users shall be able to log in using a username and password.
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.

Priority 2

15. All users should be able to select  park by viewing the park's location through Google Maps
16. Registered Users shall be able to reset their password from the sign-in screen.
17. Registered Users shall be able to place a marker on a Google Map to designate the location of an environmental incident

Priority 3

18. All users shall be able to view Reports per a zip code sourced from Google maps.
19. All users shall be able to locate parks through markers placed through Google Maps
20. Registered Users shall be able to see markers corresponding to environmental incidents placed by other users.
21. City Users shall be able to contact Registered Users directly when looking at a submitted environmental incident.

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).

UI MOCKUPS and STORYBOARDS

Use Case 1:

Logo

ContactSign UpLogin

CSC 648-Team11

This website is for demonstration purposes only.

Purpose/Mission Statement

Search

Search By ▼

AboutReport

Recent Issues

Issue Thumbnail	Issue Category Park (location) Issue Date
Issue Thumbnail	Issue Category Park (location) Issue Date
Issue Thumbnail	Issue Category Park (location) Issue Date

Logo

CSC 648

Contact

Sign Up

Login

Search

Search By

About

Report



Back to
Results

Name of Park



Google Map of Park

Current Issues in <Park Name>

Issue Thumbnail	Issue category Date submitted Issue status
--------------------	--

Issue Thumbnail	Issue category Date submitted Issue status
--------------------	--

Logo

CSC 648

Contact

Sign Up

Login

Search

Sort By

About

Report



Back to
Results

Title of Issue

Date submitted

Park

Category

Current Status

User submitted description

Image
of
Issue

Use Case 2:

The wireframe depicts a web application interface for reporting an issue. At the top, a header section contains a 'Logo' placeholder, the text 'CSC648-Team11', and three navigation links: 'Contact', 'Sign Up', and 'Login'. Below the header, a 'mission Statement' is displayed. A search bar with the placeholder text 'Search' and a 'Search By' dropdown menu is positioned next to it. A secondary navigation bar includes links for 'About' and 'Report'. The main content area is titled 'Report an Issue' with a back arrow icon to its left. The form consists of several input fields: 'Location', 'Category' (which includes a yellow speech bubble icon), and a larger 'Description' field. Below these is an 'Upload Img' section with a 'Browse' button. At the bottom of the form are two buttons: 'Submit' and 'Cancel'.

Logo

CSC648-Team11

Contact Sign Up Login

mission Statement

Search Search By

About Report

Report an Issue

Location

Category

Description

Upload Img Browse

Submit Cancel

Logo

CSC 648-Team11


Contact Logout

Search

Search By

About Report

Welcome to your dashboard <username>!

Here are your submitted  reports:

Issue Thumbnail	Issue category Park location Issue status Date
--------------------	---

Issue Thumbnail	Issue category Park location Issue status
--------------------	---

Issue Thumbnail	Issue category Park location Issue status
--------------------	---

Use Case 3:

Logo

CSC648

ContactLogout

Search

Search By

AboutReport

Issue Removal

Name	Status	Category	Location	Date	
~~~~~	Resolved				Remove
~~~~~	Resolved				Remove
~~~~~	Resolved				Remove
~~~~~	etc.				Remove
-					Remove
					Remove

Use Case 4:

Logo

CSC648-


ContactLogout

Search

Search By ▼

AboutReport

Issue Management



Name ▲	Status ▼	Category ▼	Location ▼	Submission Date ▼
~~~~~				
~~~~~				
~~~~~				
~~~~~				
~~~~~				

↙

can change  
status of issue



## HIGH LEVEL SYSTEM ARCHITECTURE

Our app is built upon the MVC architectural pattern so that the app is divided into three parts: Model, View and Controller. Our model part consists of 3 different collections, issues, parks and users. Every member of Model already has their schema initialized for future utilization. Furthermore, we also added a database section to seek a particular item and return it. Next, we have a controller section which processes what is coming into our app (user submitted content), and what is going out from our app (search result, posts), then tell these contents to go the View section for user to see. The Controller section also manipulates the Model section so that the Controller decides what to acquire from the Model section and what not to acquire (unwanted search result, like the user search for dog but Model will return dogs and dog poops, so the Controller will filter out those poop pics). Finally, the View section have all these different processed data and make them look beautiful so that users can see them as well as understand them.

## SOFTWARE STACK USED

1. Node.js/JavaScript - SSL
2. Express - Framework of Node.js
3. ProgreSQL - DB
4. Latest version of Chrome and Firefox
5. Heroku – Server


## DATABASE ORGANIZATION

<i>User</i>	FN	LN	Email	Username	Password	Title	
<i>Park</i>	Name	Zip	Description	City	State	Longitude	Latitude
<i>Issue</i>	Title	Park Name	Issue Type	Description	Status	Author	Image



Media Storage: All media will be kept in file system. Videos will be kept in the file system along with other media files, when user uploads a video, it will be storage in a sub folder named user video inside the main media folder.

Search Algorithm: User inputs a string, then our API handlebars will parse it and send it to our JavaScript codes, then our codes will assign a variable name to the user input (like var locationName = req.body.name) and pass this variable to a search function which provides by Sequelize (orm for nodejs for SQL server) something like .findOne(locationName) and whatever this function returns, it will be displayed on the search page whether if it is found or not. We are going to implement the %like function as well for typo-search. The searching algorithm will search string, numbers, you can type whatever you want in there and if such thing does exist, user will get their result.

 own API: Google Map is added to better identify the location of a park.



## CONTENT for VERTICAL PROTOTYPE



Title: Broken Sprinkler

Park Name: Golden Gate Park

Issue Type: Other

Description: You're wasting water fix it

Status: In Progress

Author: Hailey Arriola



Title: Cocaine in Restroom

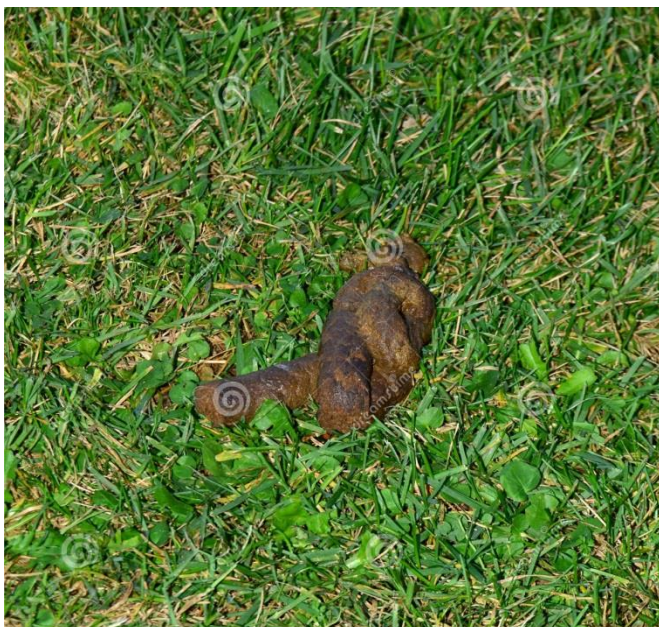
Park Name: Pine Lake

Issue Type: Dry Spill

Description: Mysterious white powder in bathroom. Probably cocaine. Local kids do all drugs. Tell cops.

Status: Received

Author: Brendan Smith



Title: Bad dog

Park Name: Glen Canyon

Issue Type: Biohazard

Description: Fix this

Status: Received

Author: Lee Tolosa



Title: Bad dog

Park Name: Glen Canyon

Issue Type: Biohazard

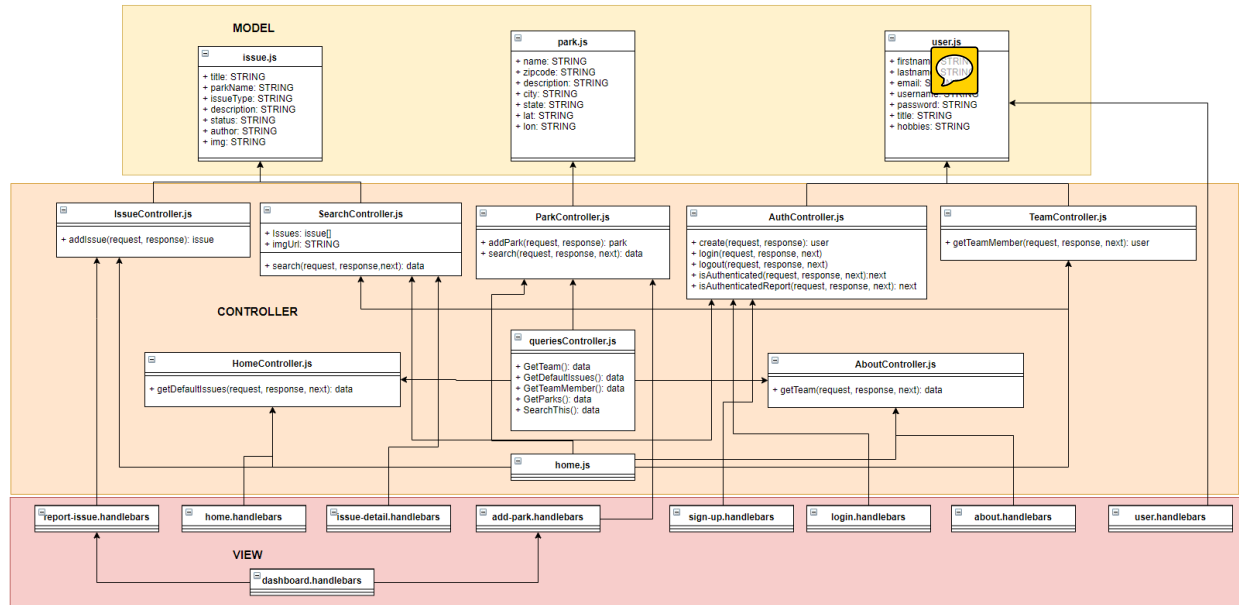
Description: Fix this

Status: Received

Author: Lee Tolosa




# HIGH LEVEL UML DIAGRAMS



## KEY RISKS

### Skills:

Team is unsure how to design UML.

**Steps Planned:** Avi and Jason are  collaborating on UML rather than just Jason.

Team is largely unfamiliar with Handlebars.

**Steps Planned:** Planning better documentation for files, and Rodolfo is training Jason on the back-end to better understand it. Dylan has also been familiarizing himself more and can help Avi in turn on the Front End.

Frank is from a tech unfriendly environment and team is unable to get the local server infrastructure working on his computer.


**Steps Planned:** Frank is mostly focusing on documentation and QA.

### Scheduling:

Brady has a very tight schedule with work, personal issues, and other classes, and is concerned about having sufficient time to continue guiding team progress towards Milestone deliverables safely.

**Steps Planned:** Team has implemented a project management app called Asana to help manage tasks and assignments, with respective due dates, to better organize.


In general, we are somewhat concerned about having sufficient time for addressing QA.

**Steps Planned:** Frank has been assigned overall responsibility for now to run QA tests on live site every few days and put in feedback accordingly. 

### Technical:

Back-end has concerns over how to implement database setup correctly and cleanly.

**Steps Planned:** Dylan will help out Jason on back-end for the moment to help resolve this hurdle.

 Front-end has concerns over how to properly secure code.

**Steps Planned:** Will restrict access to database behind DB credentials, while keeping good validation of input from website fields.