

# SW Engineering CSC 648/848 Section 02 Spring 2018

Team 09

**Tony Chen (tchen1992@gmail.com)**

Greg Strom

Siri Motamarry

Jerry Zhang

Patrick Wong

Alonzo Contreas

Milestone 2

## Data Definitions:

Unregistered users - Shall be able to search and view current posts and issues. **(1)**

Registered users - Shall be able to register, create, and edit posts. **(1)**

Admin - Shall be able to monitor/block posts. Shall not be able to edit other users posts. **(1)**

City Manager - Shall be able to close tickets as well as allocate work to specific parks. **(1)**

Search - Users shall be able to find specific park/location **(2)**

List – A page that lists out the parks. **(1)**

Case status - Shows whether park/location is clean or not. **(1)**

Incident - Problem/issue at park/location **(1)**

Register - User shall be able to register to allow them to post, upload pictures and edit posts. **(1)**

Creates - User shall be able to create a post. **(1)**

Edits - User shall be able to edit their posts. **(2)**

ID - a unique number given by the system to identify users and incidents. **(3)**

Username - a unique name the user chooses to post under. **(2)**

Password - a key that must be typed in to allow access to users. **(1)**

Incident Report - this includes date, description, photo, address, location, and priority of the report. **(1)**

## Mockups

USE  
(logo)

Login/ Register

Search by zipcode, park name, city

Tabs: About List Contact

Description about USE website

or

most recent issues.

USE

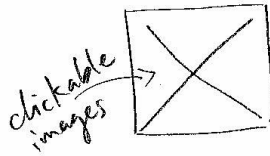
(Search results page)

Login / Register

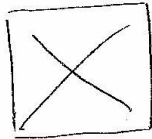
Search by zipcode, parkname

Search results

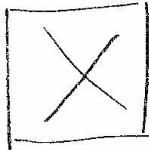
1, 2, 3, ...



Description about park.  
info.



\_\_\_\_\_



\_\_\_\_\_

1, 2, 3, ...

USE

(about the park)

Login/Register

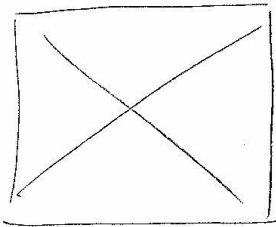
Search

Return to search

Park Name

Status :

Report Issue



Description of  
the park

Issues

Open / In progress

1) \_\_\_\_\_

2) \_\_\_\_\_

3) \_\_\_\_\_

Pg. 1, 2, 3, ...

USE

(Report Issue page)

Logout / Welcome  
user

Search

## Report Issue

Park

Address

Issue 

Title (drop down menu)
Comments *

Photo

\* required

Cancel

Save

USE

(Login/Register page)

Login/Register

Login

Register

☐ Agree to terms

USE

(Registered User - Homepage)

Logout | Welcome  
User

Search

My Account

List of Issues posted

Park name : Title : Status

Park name : Title : Status



USE

(Case manager Homepage)

Welcome  
manager

Search

Open	In Progress	Most Recent	Closed
Park name : Issue : Status : Assigned To			

USE

(Case Manager Assignment Page)

Welcome  
Manager

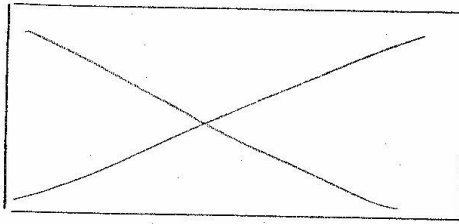
Search

Park name

Issue

Text Autofilled

Submitted  
Photo



Assign to

Text

Comments

Text (Optional)

Status

Drop down menu

Save

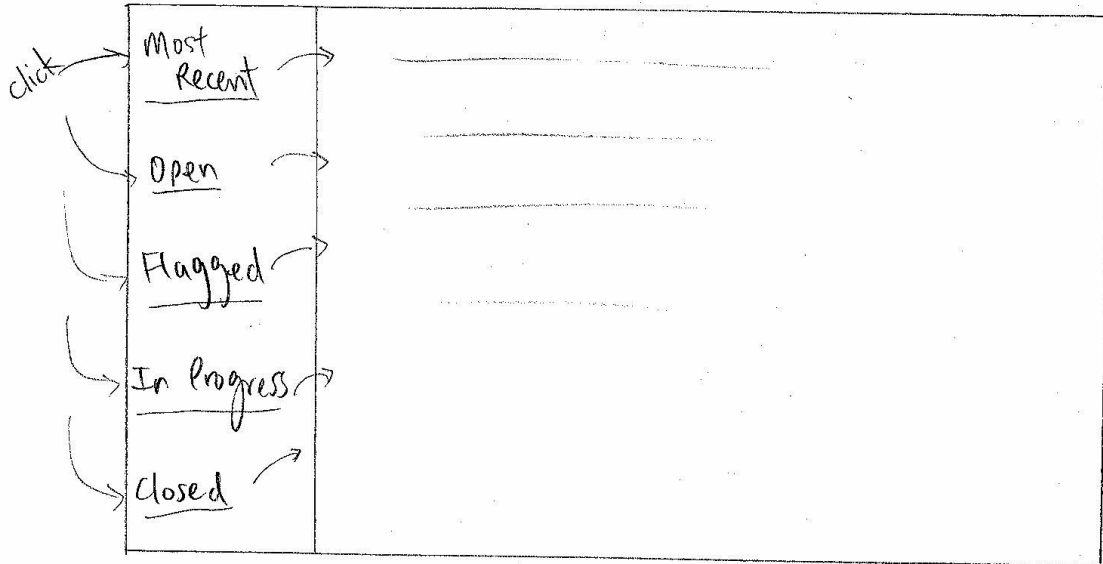
Cancel

USE

(Admin Homepage)

Logout / Welcome Admin

Search



USE

(Admin Delete Issue)

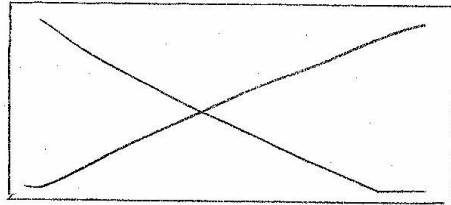
Logout | <sup>Welcome</sup>  
Admin

Search

Park name

Issue

Submitted  
Photo



Delete

Cancel

## High Level Architecture

### *Model:*

We use a MySQL database that is split into two main tables, users and incidents. Users create/assign incidents. The data for our webpages as well as everything that is not an input from the user is stored and pulled from GitHub. Apache and AWS are used to display the data from GitHub on the users' monitor. The database is able to process queries and compute and display the correct results.

### *View:*

The user will be able to view the webpage on the 3 latest versions of Chrome and Firefox. Our application uses dynamic page validation which will check the dimensions of the window that is open and displaying the page and change the resolution relative to that.

### *Controller:*

The controller will include printing statements, storing values, and getting commands as well as setting them.

## Database Organization

The database is comprised of two tables; a users table and an incidents table. The users table includes the following:

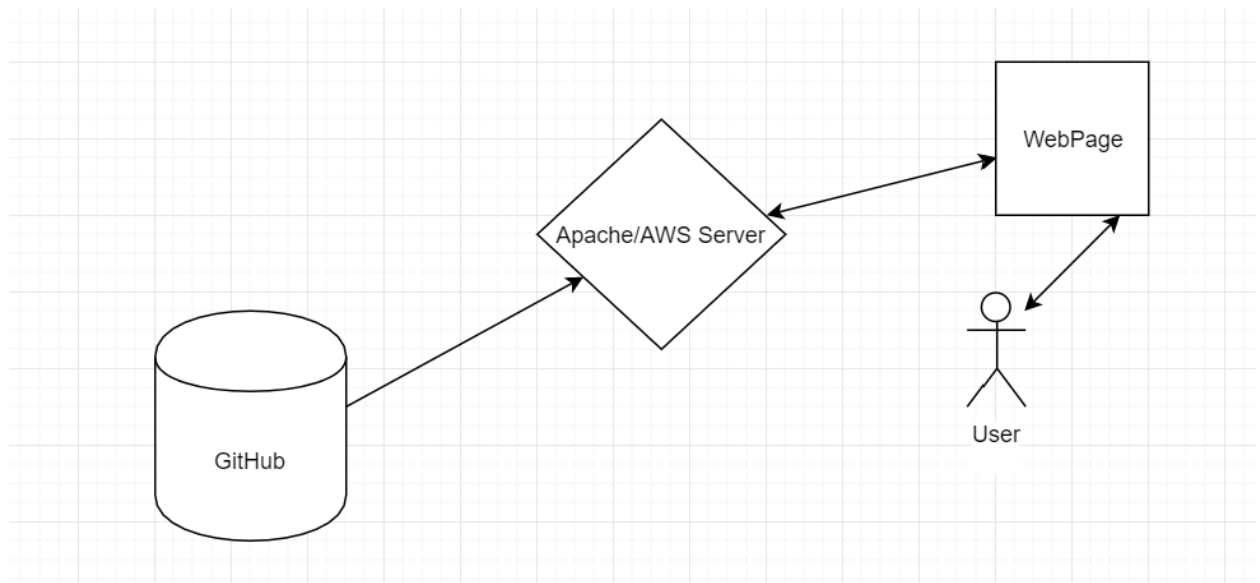
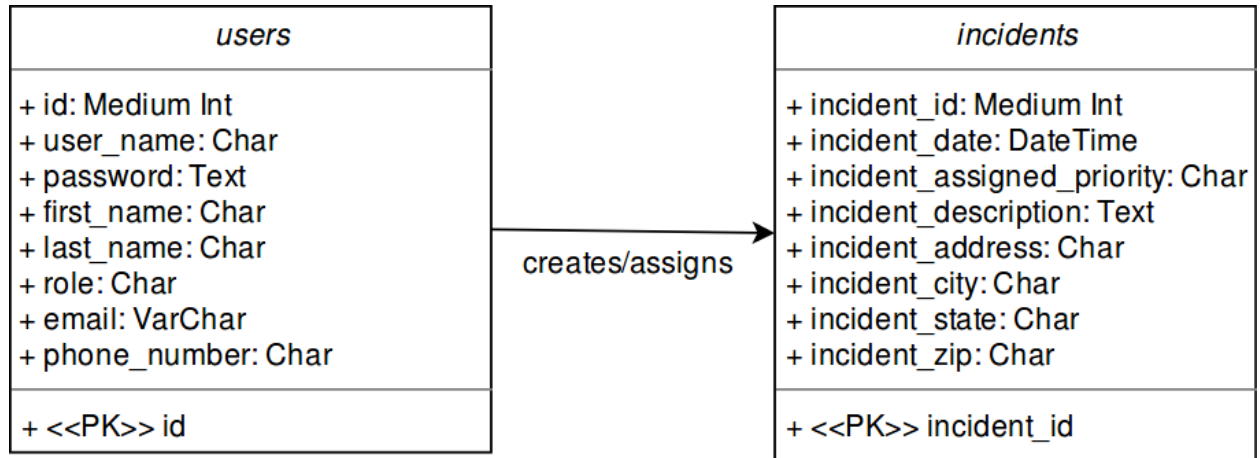
- User\_id
- User\_name
- User\_password
- User\_first\_name
- User\_last\_name
- User\_role
- User\_email
- User\_phone\_number

The incidents table includes the following:

- Incident\_id
- Incident\_date
- Incident\_assigned\_priority
- Incident\_description
- Incident\_location
- Incident\_address
- Incident\_city
- Incident\_state
- Incident\_zip
- Incident\_latitude
- Incident\_longitude
- Incident\_status
- Incident\_status
- Incident\_photo

The first table is when a user registers for a new account and must enter those data fields. The second table is created by the users in the first table when they report an incident. **Media storage is in BLOBs. All photos used are BLOBs.** Although it is still not final at this stage as there are many trials to be run, we will preliminarily use the SQL command %like for searches. Google Maps API will be used to locate parks and will be integrated in our webpages.

## High Level UML Diagrams



## Current Risks to the Team

- skills risks (do you have the right skills), I believe that this is not a major risk since everyone on the team is a fast learner and the growth on the team has been great so far.
- schedule risks (can you make it given what you committed and the resources), The team has been able to meet every time when needed to complete tasks, so I believe that this will not be a problem unless it is during either a break or exams weeks.
- technical risks (any technical unknowns to solve), The current problem we are facing is making dynamic content work on our webpage. This will only take a couple weekends to solve and is not a problem.
- teamwork risks (any issues related to teamwork); The team is working great together.
- legal/content risks (can you obtain content/SW you need legally with proper licensing, copyright). Since we will be using clip art and photos of public parks, there will be no copyrights we need to worry about.