

# SW Engineering CSC648/848

## Section 01 Spring 2018 Team 3

# EnviRepair

## Milestone 2

3/18/2018

**Gary Straub** [gstraub@sfsu.edu](mailto:gstraub@sfsu.edu)

**Gayoung Kim**

**Gerren Penaloza**

**Justin Abarquez**

**Lily Linh Lan**

**Uzair Inamdar**

Submission	3/18/18
Revised and Frozen	

## Contents

1. Data Definitions .....	2
2. Functional Requirements.....	3
Priority 1.....	3
Priority 2.....	3
Priority 3.....	4
3. UI Mockups .....	5
Registration / Login.....	5
Home/Dash .....	6
User Profile / Edit Profile .....	7
Contact.....	8
Report Submission .....	8
Detailed View .....	9
Unregistered User Registers Then Browses.....	10
Unregistered User Submits an Issue Then Leaves .....	11
Registered User Checks Profile History to checkup on post.....	12
4. High Level Architecture, Database Organization .....	13
Model.....	13
View .....	13
Controller .....	13
Database Schema.....	13
Logical Schema for database .....	14
Media Storage.....	14
Search/filter architecture.....	14
6. High Level UML Diagrams .....	15
7. Identify actual key risks for your project at this time .....	16

## 1. Data Definitions

- a.** Environmental Issue – This is an issue in a user's community that poses some kind of danger or discomfort to the user and those around them, this issue is normally caused by the environment or human negligence. Attributes: Location, type of issue, category, status, comments and photos.
  - i.** Category: The category table will be responsible for storing the specific type of environmental issue (ie. pollution, waste, toxicants, etc.).  
Issues: The issues table is responsible for storing all of the information related to the issue that a user reports. There is a foreign key pointing to the users table as well as the category table so that we will be able to record which user posted the issue as well as which the category the issue belongs to. The issues table records all of the detailed information about the issue at hand such as the location of the issue (street address, city, state, zip code), the status of the issue (resolved, in progress, not resolved), the number of times the issue has been reported, the severity of the issue ranked from 1 to 5 (a rank 1 issue could be as simple as a fallen tree at a park whereas a rank 5 issue could be a nuclear waste spill in a highly populated area), and a short description of the issue provided by the user that is provided at the time of the report.
- b.** Flag – Users are allowed to flag content as inappropriate, after a post receives a number of flags, it is a target for moderation.
- c.** Unregistered User: May post and browse environmental issues
- d.** Registered User: Same privilege set as Unregistered User, but may also: Flag posts deemed inappropriate, access to personal info, posting history, can edit a post made by them only if the city manager has not changed the status of the post.  
User Data: The users table stores the user information which includes their username, password (hashed), email, and their privilege (the details for this are stored in milestone 1).
- e.** City Manager: Same Privilege Set as Registered User, plus can edit status of postings from in-progress to complete, may also post relevant information to status changes.
- f.** Administrator – Same Privilege set as Registered User, also given moderation capabilities, can hide, unhide and delete posts that are inappropriate, may also delete users but not edit posts.

## 2. Functional Requirements

### Priority 1

- 1.1) Unregistered Users shall be able to report an environmental issue
- 1.2) Unregistered Users shall be able to be prompted with potential duplicate issues as to not report the same issue twice (Not AI search)
- 2.1) Unregistered Users shall be able to browse current environmental issues based on a variety of search parameters (Location, severity, and type)
- 3.1) Unregistered Users shall be able to browse environmental issues in the form of a map
- 15.1) Unregistered Users shall be able to register and become a Registered User
- 4.1) Registered Users shall be able to have the same access and functionality as an Unregistered User upon logging in.
- 5.1) Registered Users shall be able to search functionality to return results closest to my location which I provided upon registering

### Priority 2

- 6.1) Registered users I would like to browse issues I have submitted (such as a history page)
- 6.2) Registered users when visiting my history page, I would like to see the most relevant content first (Pending issues first, then solved, organized by date)
- 6.3) Registered users when visiting my history, I would like to have some filtering options to better view older posts.
- 6.4) Registered users I would like my history page to reflect the difference between posted reports and comments.
- 7.1) Registered Users I would like to edit a post I made with relevant information I thought of later.
- 8.1) Registered users, when I submit an issue and prompted with the same issue, I would like to comment on the issue rather than create a duplicate.
- 8.2) Registered users, when I submit an issue and prompted with similar issues, I would like to continue submitting my issue because the issue I am submitting is unique.
- 9.1) Registered users I would like to flag posts I think are inappropriate.
- 10) Registered users I would like to change my account information
- 16.1) City managers I would like environmental issues to be frozen once I have assigned a workflow (User should not be able to edit posts after city manager freezes the post)
- 11) City Managers I would like to view current issues in my assigned district
- 12) City Managers shall be able to post solution generated by department, marking an issue solved or in progress, with the addition to comments pertaining to the change in status
- 15.2) City Managers I would like to mark posts as duplicates making my workflow easier
- 13.1) Administrators I would like to be notified once there have been sufficient flags on a post
- 13.2) Administrators I would like post to be automatically hidden if they receive a number of flags
- 14) Administrators I would like to hide or unhide posts based on the premise of inappropriate content.

- 14.2) Administrators I would like to delete users or posts due to inappropriate content
- 15.1) Administrators I would like to mark environmental issues as duplicates

### Priority 3

- 17.0) Unregistered Users shall be able to browse posted environmental issues.
- 17.1) Registered users I would like to view the status of my posted environmental issues when I log in
- 17.2) Registered users I would like there to be a visual difference between the difference progress levels of the environmental issues I have posted
- 18) Registered users I would like to follow certain posts which I find interesting, which will be displayed on my dash.
- 11.2) City Managers, when viewing issues in my district I would like them to be grouped by their different stages in workflow
- 11.3) City Managers, when viewing issues in my district I would like the option of grouping them by severity or other criteria
- 18) City managers I want to update severity or priority of an incident.
- 19) City managers, I want automatic escalations for incidents that have not been addressed within a time period.
- 13.3) Administrators I would like my dash to display post most relevant for my attention such as post that have been hidden [first then by flag count](#)

### 3. UI Mockups

#### Registration / Login

Registration

Home Issues Contact About Login Register

Name

E-mail

We won't spam. Promise!

Password

Confirm Password

Submit

Login

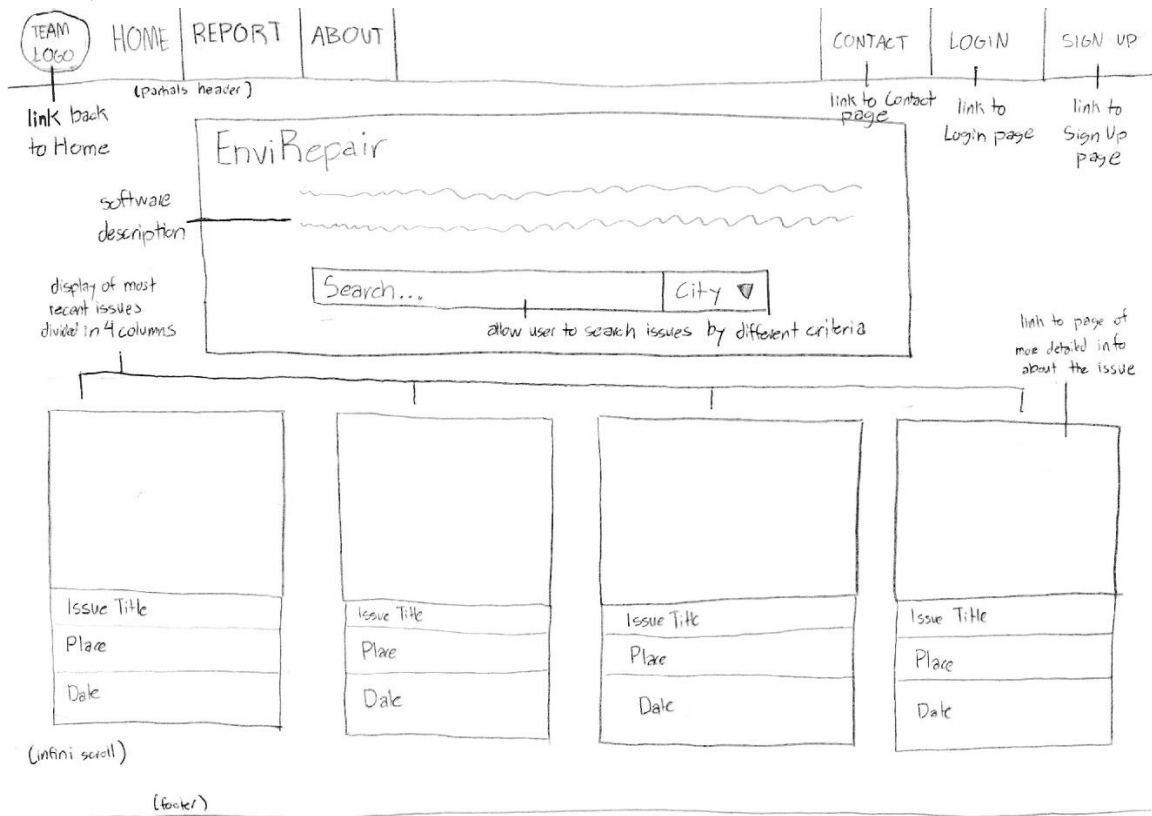
Home Issues Contact About Login Register

Username

password

Submit

## Home/Dash



## User Profile / Edit Profile

EnviRepair HOME | REPORT | ABOUT

Notifications <sup>issues status changed</sup> <sup>new comments made</sup> Hello, Lily

### My profile

#### My personal info

First name : Lily

Last name : L

Username : Lily123

Email : Lily@gmail.com

Phone : 123-456-7890

Change password

Current

New

Re-type new

#### My ENVI reports

Date ▼	Title	Location	Status
03-01-18	Oil Spill	Golden Gate	Pending

### Recent activities

If I click on reports

- Created a report : "Oil spill" Feb 1 Pending
- Created a report : "Glass broken" Jan 10 Solved

link to issue detail's page

If I click on comments

- Commented to Peter's report : "Illegal dumping" Feb 3 Pending

If I click on flags

- Flagged to Mary's report : "\_\_\_\_\_ " Jan 3 Pending



## Contact

Team Logo

HOME REPORT ABOUT CONTACT LOGIN SIGN UP

CONTACT

Open the new tap for the company website

image

Kevin  
Environmental World Company  
800) 205 - 1234  
kevin @ google . com

image

image

Name  
Company  
phone number  
e-mail

image

image

Name  
Company  
phone number  
e-mail

image

Name  
Company  
phone number  
e-mail

## Report Submission

EnvRepair Cares

### REPORT AN ISSUE

Issue:  — text field for ~~name~~ issue title

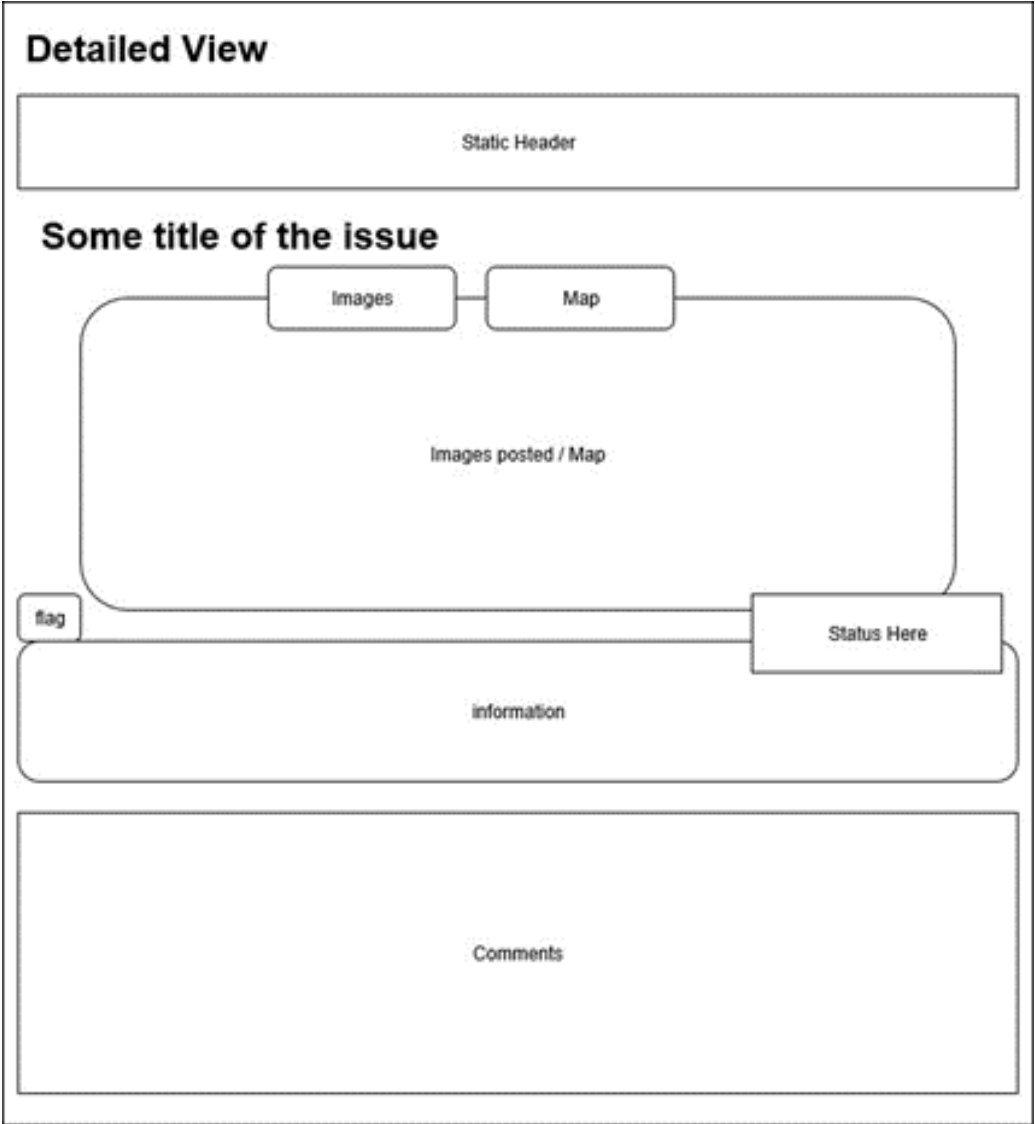
Description:  — larger textfield for description

Image:    — allow user to browse computer for image to upload

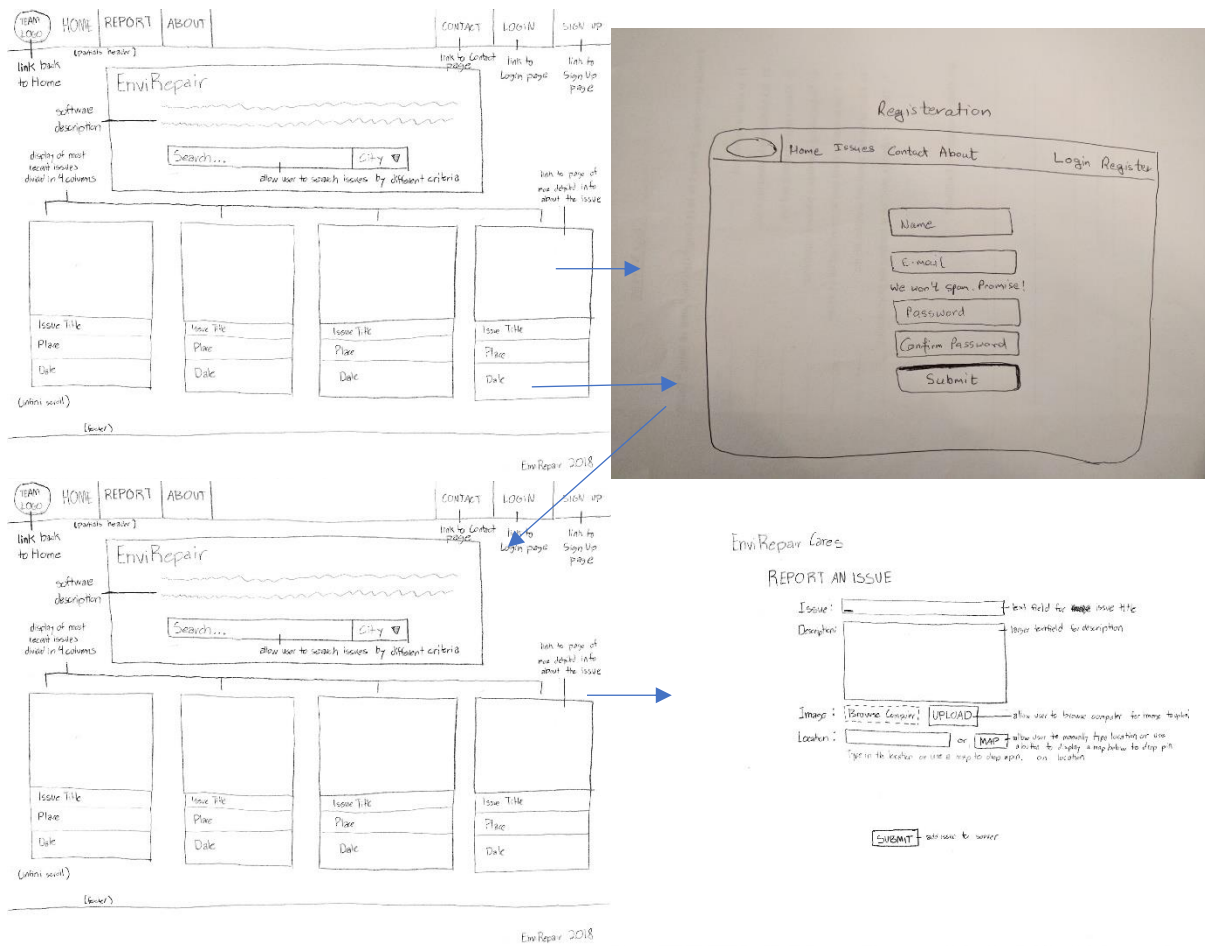
Location:  or  — allow user to manually type location or use a button to display a map below to drop pin  
Type in the location or use a map to drop a pin on location

— add issue to server

Detailed View

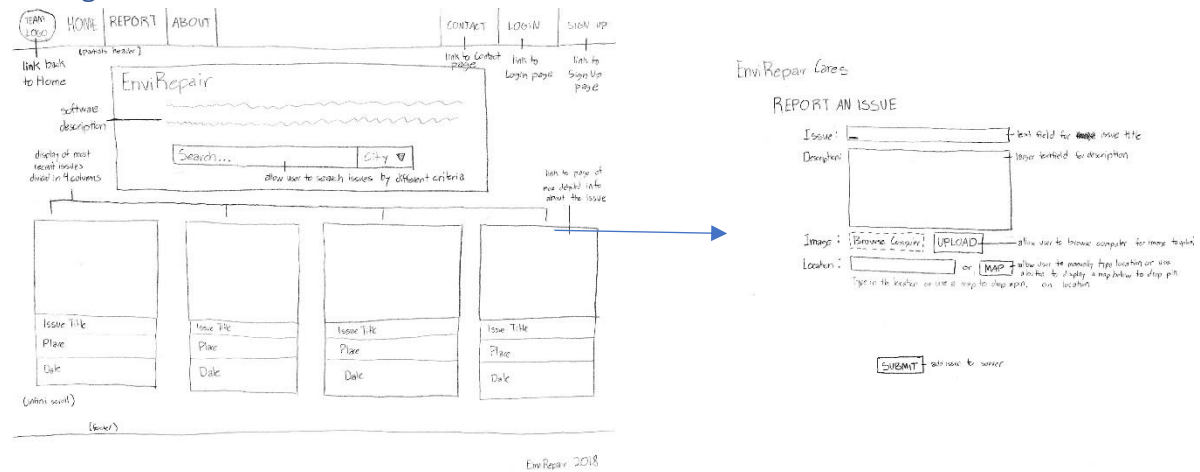


## Unregistered User Registers Then Browses



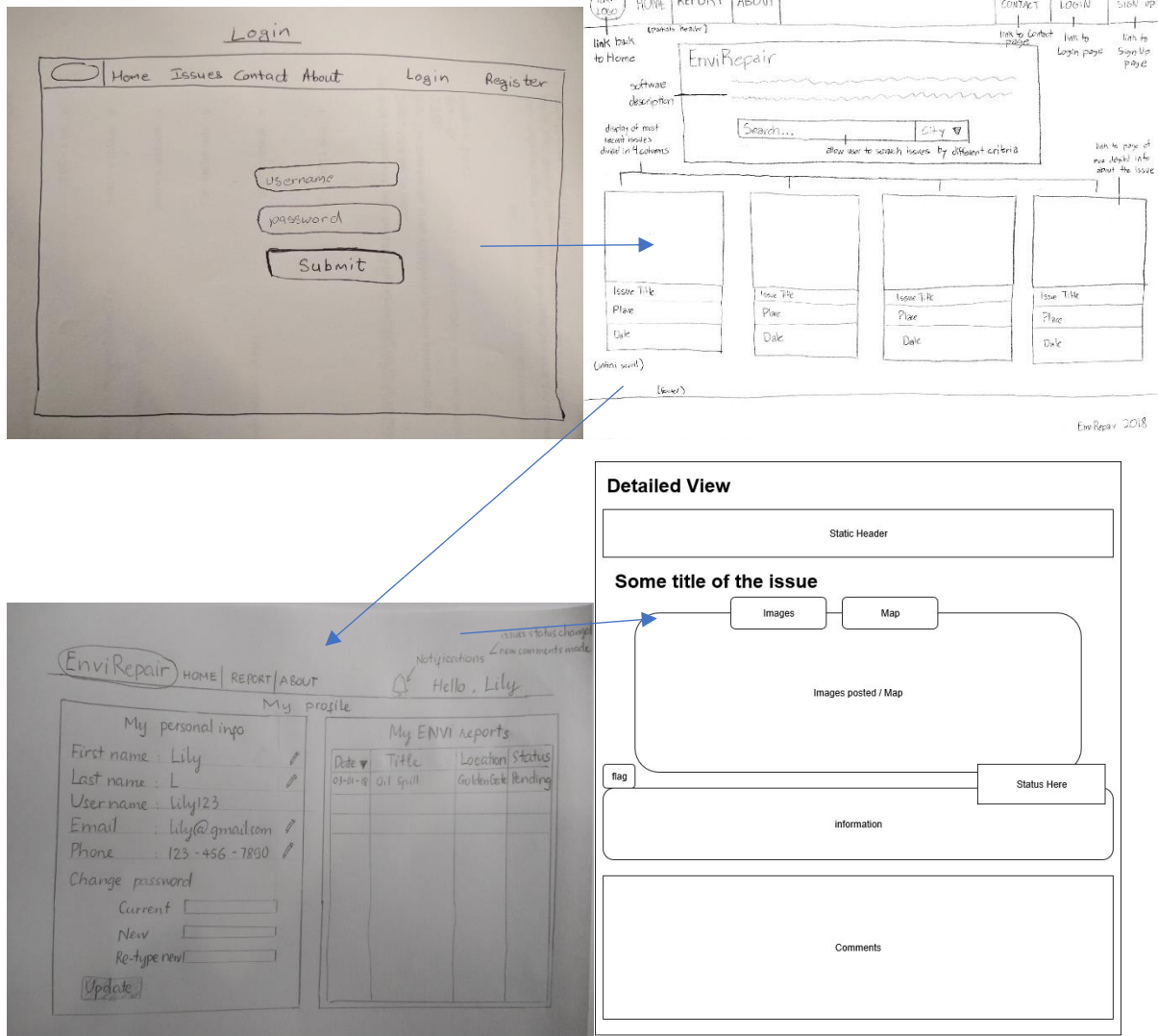
Unregistered users should be able to register, this is a part of the core functionality, please note that currently (Version 1) unregistered users will not be able to import the reports they submit as an unregistered user

## Unregistered User Submits an Issue Then Leaves



An unregistered user notices something and wants to report it, they are not interested in registering or checking up later. UI should allow someone to quickly report or view issues without having to go through an intense navigation

## Registered User Checks Profile History to checkup on post



A registered user logs in, checks out their history and then selects a report to look at in detail. Ideally Logging in should be fluid, basic/recent history may be incorporated into the dash, certain iterations of this story will skip over the profile view.

## 4. High Level Architecture, Database Organization

### Model

Our postgres database in conjunction with our node app serve as our model. Our database will store all relevant information from our website. This information includes user data as well as any issues submitted by registered and non-registered users. There will be functions that are triggered by the controller that allow us to read/modify any data in our database which will then update the view accordingly.

### View

The content in our views are rendered from the data in our model. The user-submitted issues will be posted onto our website and represented as bootstrap cards. The cards will contain a picture, a description, a location, a category, and any relevant user information linked to the post. Our site will also contain a dashboard for user information that displays individual user information that is readily available for modification. The dashboard also contains a user's history of reported issues. Lastly, our site will have a contact page that provides users with the information necessary to contact admins, as well as city managers with any concerns that they may have.

### Controller

Our controller will be a combination of keyboard inputs as well as button clicks that trigger different events. There will be buttons allowing users to view log in, registration, and submit issue forms. Hyperlink clicks will allow users to navigate other parts of the website. Users will use their keyboard to fill out the forms as well as button clicks to submit the information to be stored onto the model. On the user dashboard, there will be options for the user to click and edit/update their user information.

### Database Schema

**Users:** The users table stores the user information which includes their username, password (hashed), email, and their privilege (the details for this are stored in milestone 1).

**Category:** The category table will be responsible for storing the specific type of environmental issue (ie. pollution, waste, toxicants, etc.).

**Issues:** The issues table is responsible for storing all of the information related to the issue that a user reports. There is a foreign key pointing to the users table as well as the category table so that we will be able to record which user posted the issue as well as which the category the issue belongs to. The issues table records all of the detailed information about the issue at hand such as the location of the issue (street address, city, state, zip code), the status of the issue (resolved, in

progress, not resolved), the number of times the issue has been reported, and a isFlagged boolean that other users can mark if they find that an issue on the website is inappropriate/irrelevant.

### Logical Schema for database

Users: { user\_id: integer, username: String, password: String, email: String, privilege: integer }

Categories: { category\_id: integer, type: String }

Issues: { issue\_id: integer, category\_id: integer, user\_id: integer, username: String, title: String, description: String, resolved: String, streetAddress: String, city: String, state: String, zipcode: String, numberOfOccurrences: integer, imagePath: String, isFlagged: boolean, createdAt: Date, updatedAt: Date }

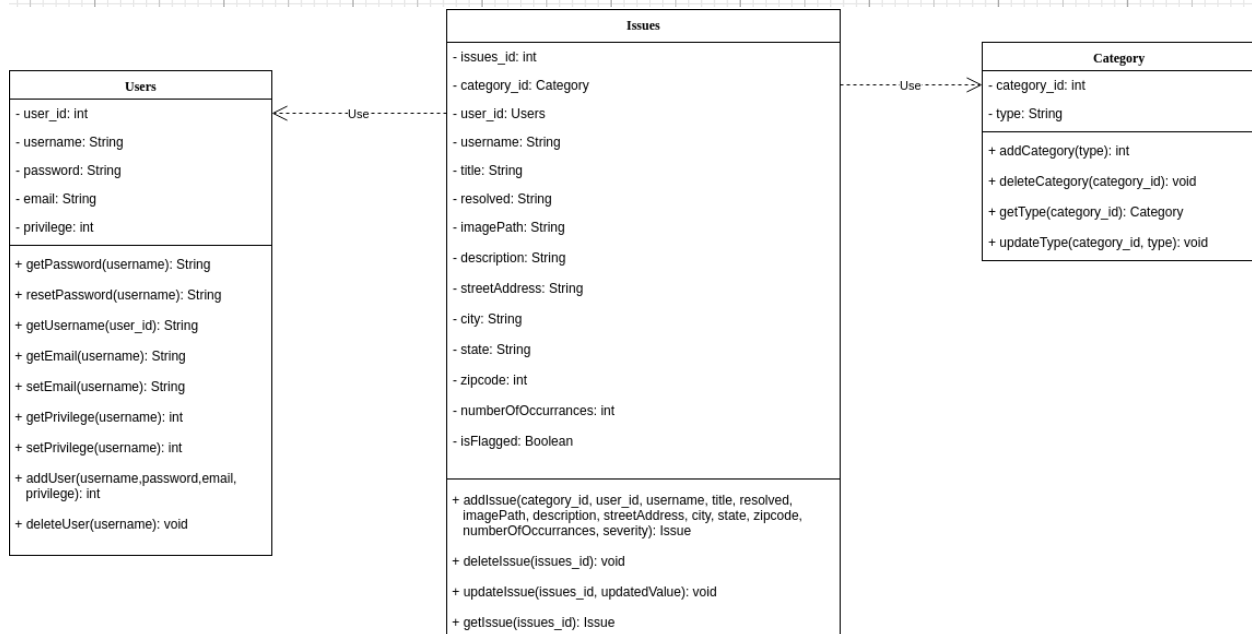
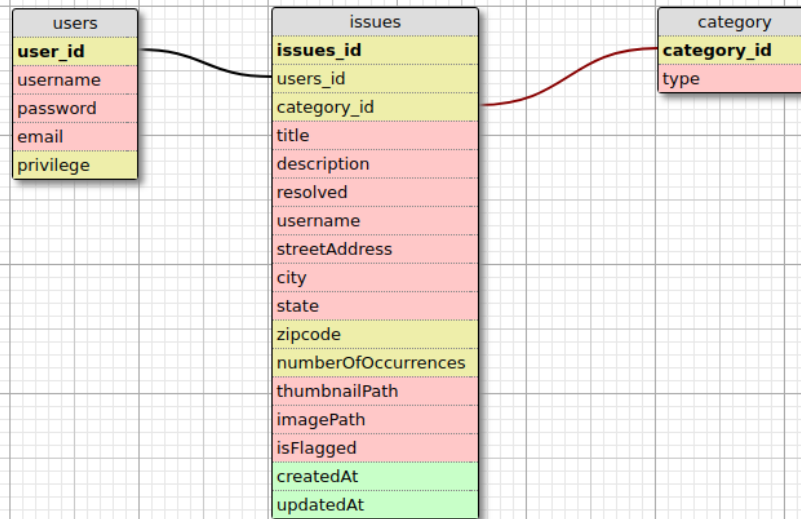
### Media Storage

We are choosing to use the file system to store images submitted by users. The relative path to each of these image files will be stored in our database. When issues are being rendered to the view, the path to the correct image to display will be read from the database. Bootstrap offers a class that will make each image appear as a thumbnail which will be using to display our issues in a more compact way when displaying search results.

### Search/filter architecture

For now, we implemented our fuzzy search using SQL and %like. We were advised to allow the user to search by city and zip code for simplicity. Our search operates similar to how a basic fuzzy search works in the sense that the user can type in full cities/zip codes or substrings of cities and zip codes and the database will try to match the input to any pre-existing record in the database. By default, we are going to order the search results from most recent to least recent by using the created At column in the issues table that stores the date of when the issues were created.

## 6. High Level UML Diagrams





## 7. Identify actual key risks for your project at this time

- 1) Experience Risk – Team has never worked with a cloud service such as Google Cloud Compute

Solution: Read up on the documentation offered online. The only real risk right now is migrating the database schema over to the database instance on the cloud, but the last resort if we cannot figure out how to get sequelize to work with it will be to just create the schema for the database manually using SQL commands.

- 2) Experience Risk – Front End team has limited experience with templating engines such as EJS

Solution: EJS has been around for some time thus there is a good amount of documentation as well as YouTube tutorials the team can use to learn it. Uzair is familiar with EJS and is helping facilitate the learning process.

- 3) Scheduling Risk: The only time the entire team is able to meet is on Monday.

Solution: We are documenting any newly written code and updating the README in our GitHub repo to help those who cannot meet during the week understand the new code.

We are also ensuring that the front-end lead meets with one of the back-end leads at least once during the week. Scrum meetings and being communicative over slack should be enough to keep everyone in the loop.