SW Engineering CSC 648/848
Section 01 Team 07
Eco Hazards
Milestone 2

Spring 2018

Sean Sutherland(ssutherl@mail.sfsu.edu), Lance Larsen, Corey Humeston,
Mark Soriano, Girish Tiwale, Ali Alavi, Amelie Cameron

Revision	Date
1.0	03/18/2018
1.1	

1. Data Definitions V2

Types of users & their privileges

Anonymous User

Unregistered user browsing the site

- View (active) environment reports
- Submit environmental report validated by personal information
- Register

Registered User

User who has created an account

- Anonymous User plus:
- Login
- Post environmental report without having to re-provide information
- Follow up on problem (see status, provide more info, etc)
- Confirm resolved
- Update profile

Env dept

Nontechnical employee of the SF Environmental department

- Registered User plus:
- View requests
- Assign/update status
- Reply to reporter

Admin

Technical user responsible for site moderation.

- Registered User plus:
- Create env manager account
- Approve/reject reports
- Flag users/suspend account

Types of stored data and contents:

Hazard Report
Report on a single environmental issue submitted by a user

- Date
- Location
- Type of issue (chemical spill, etc)
- Reported by (if user was registered)
- Image(s)
- Hazard Status

  Location
  A park, beach, or other public area which the user can select when submitting a report
  - Name
  - Coordinates
  - Type

  Hazard Status
  Whether a hazard has been addressed by the environmental department.
  - Awaiting assignment, assigned, in progress, hazard removed
  - Entity assign to, if any
  - Removed hazard reports considered inactive

User Account

- Username
- Email
- Password
- Account type (User, Env Dept, Admin)
- Phone number
- Hazard Posts
- Hazard Post count

2. Functional Requirements V2

Priority 1

1. Users shall be able to post information about environmental hazards in their area.
2. User shall be able to add all relevant information when uploading a report, including images, excluding status and who it is assigned to
3. Authentication to post hazard reports shall be provided by either Login or by user providing identifying information (Name/email/phone number)
4. Users shall be able to view posted (active) hazard reports and associated information.
5. Users shall be able to search for reports by zip code or location

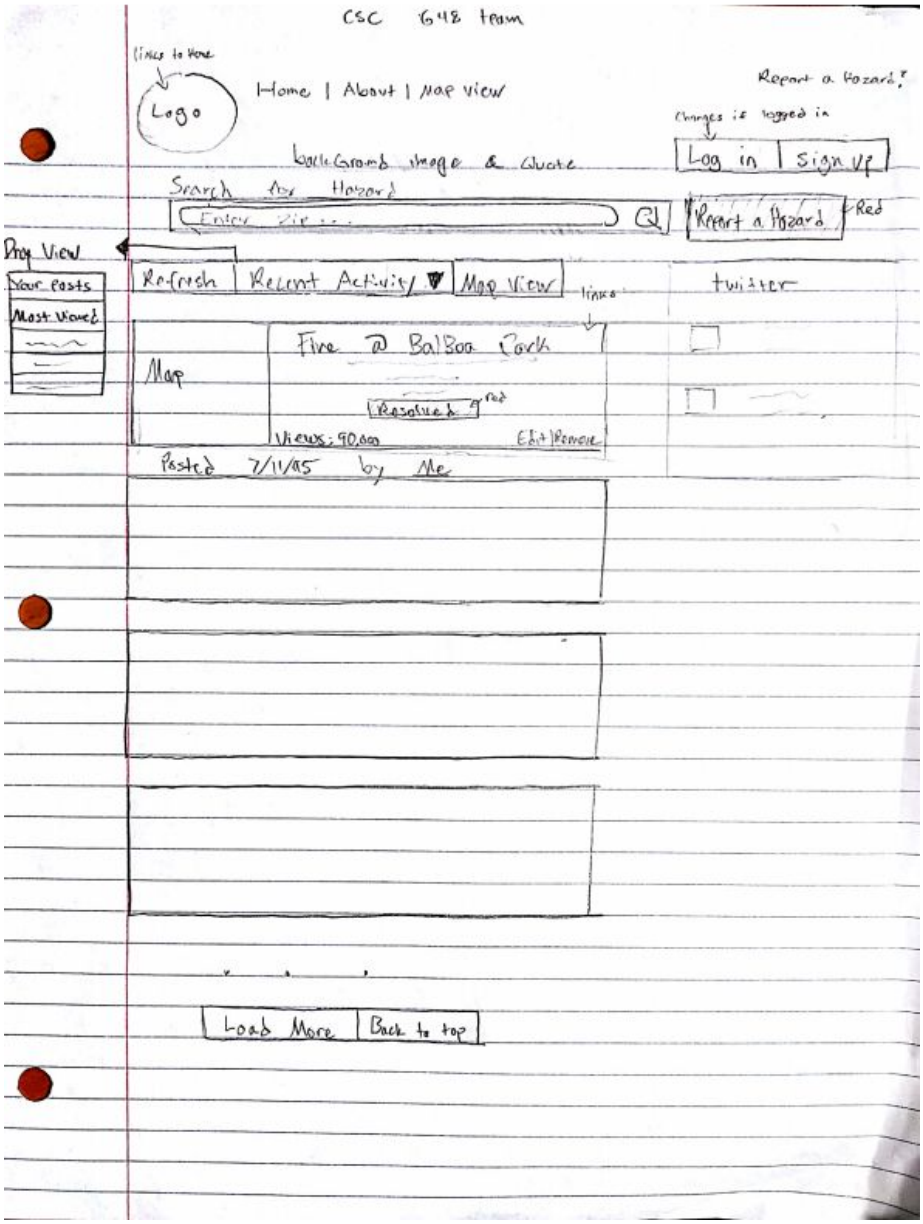
Priority 2

6. Anonymous Users shall be able to register and create an account.
7. Registered Users shall be able to log in
8. Administrators shall have the privileges to remove inappropriate post that do not meet the website guidelines.
9. Administrators shall have the privilege to ban users who break the guidelines.

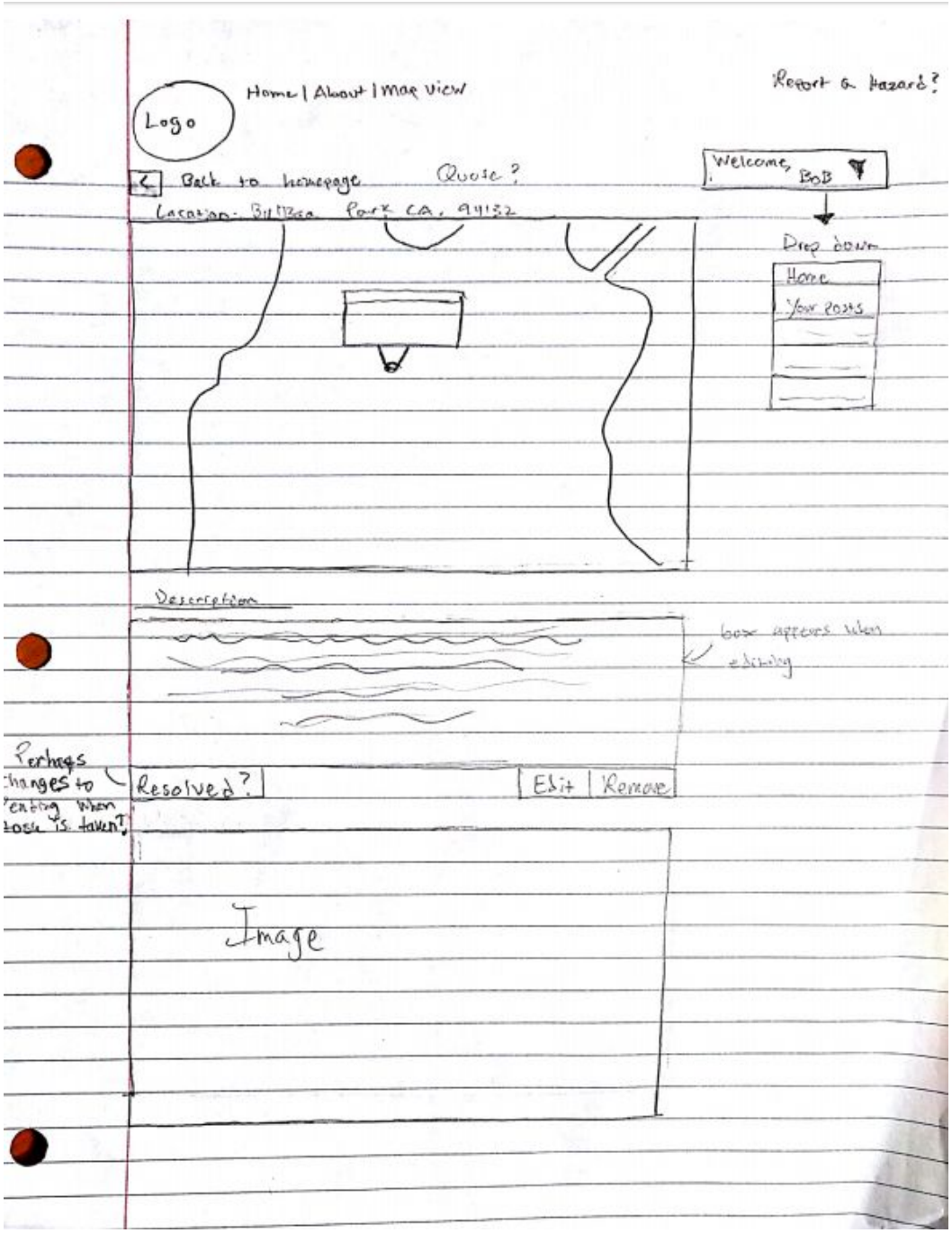
Priority 3

10. Env Dept shall be able to view list of active reports and see who they are assigned to
11. Env Dept shall be able to sort active reports by their status and who they are assigned to
12. Env Dept shall be able to update data of hazard reports including status and who they are assigned to

3. UI Mockups and Storyboards (high level only)

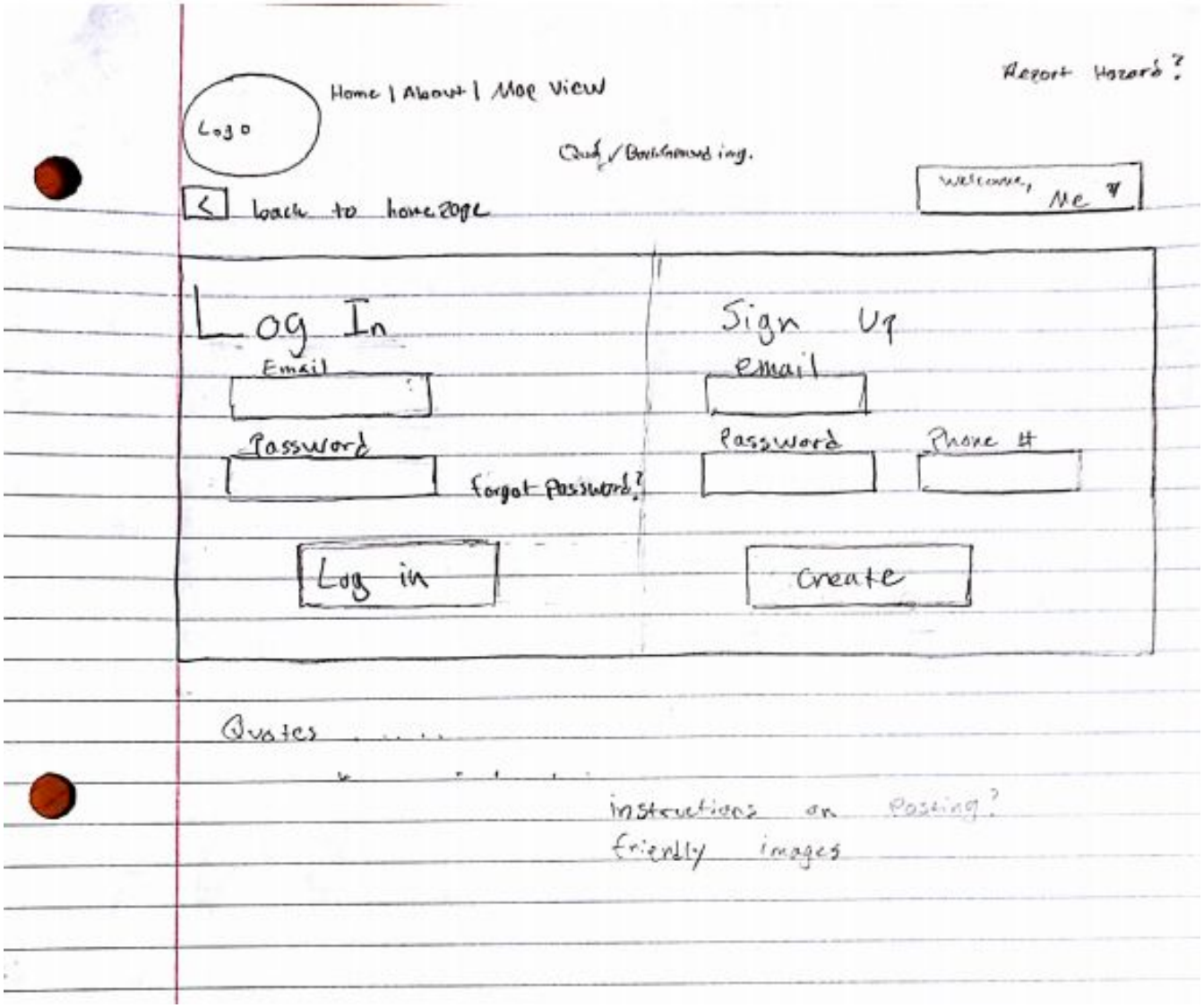


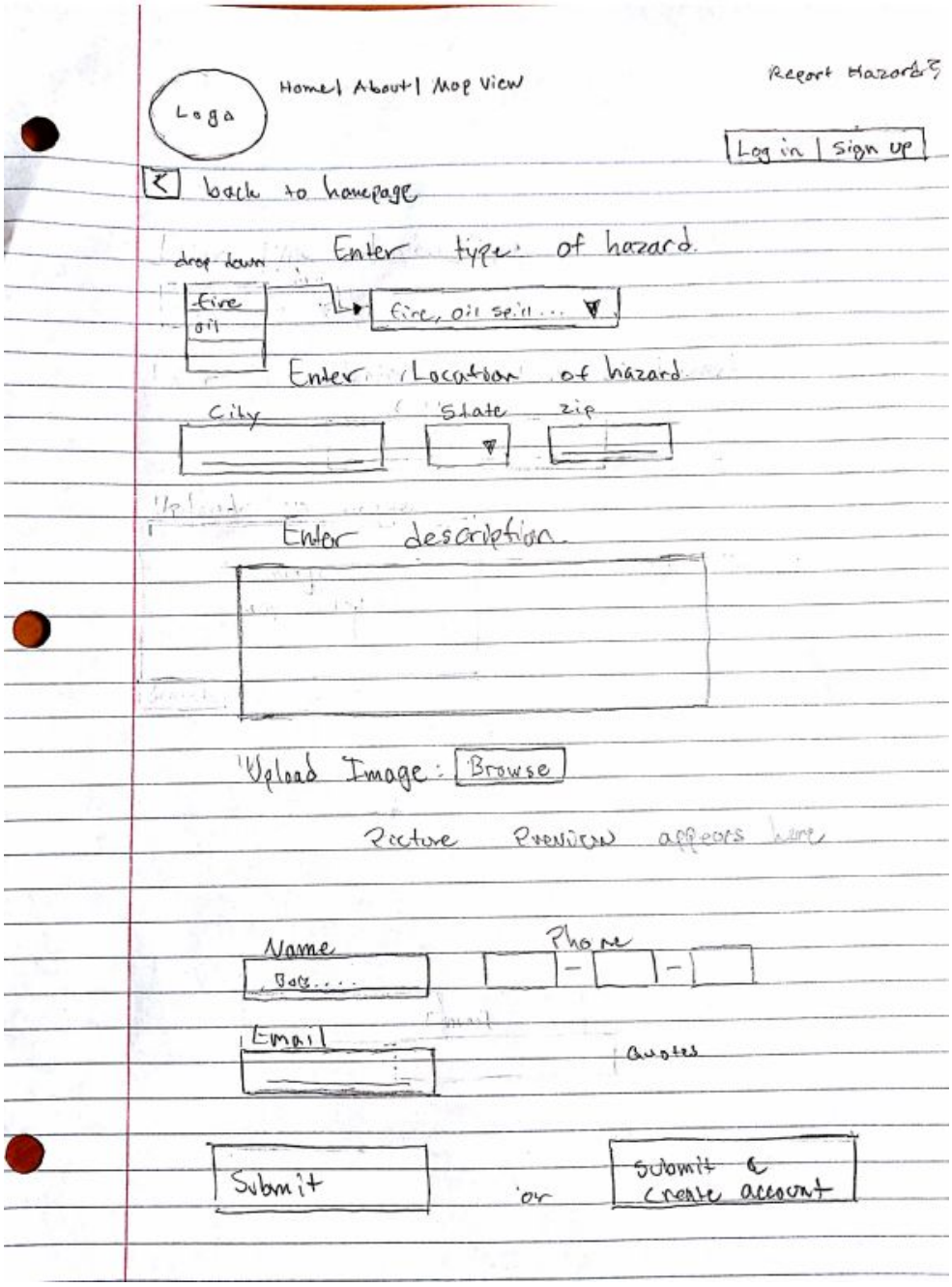
Homepage View - this will be the main page of our website



Registered User View - when a user has logged in, this main screen shall be view of the user screen.

Login/Signup - this view is transitioned by tapping on the login or signup button on homepage





Post Hazard Report View - this screen will be viewed when a user has tapped on “report a hazard” button on home page view.

4. High level Architecture, Database Organization

Development tools:

The application shall be developed using: Linux, Apache, MySQL and Python.
The main framework that shall be used in the development of this application is the Python Django framework.
Technologies that shall be used in the development of this application are JQuery, Bootstrap and HTML/CSS.

Version control:

The code for this application shall be stored on our project repository on Github.
Team members shall collaborate together on Github, in person, and on Asana.

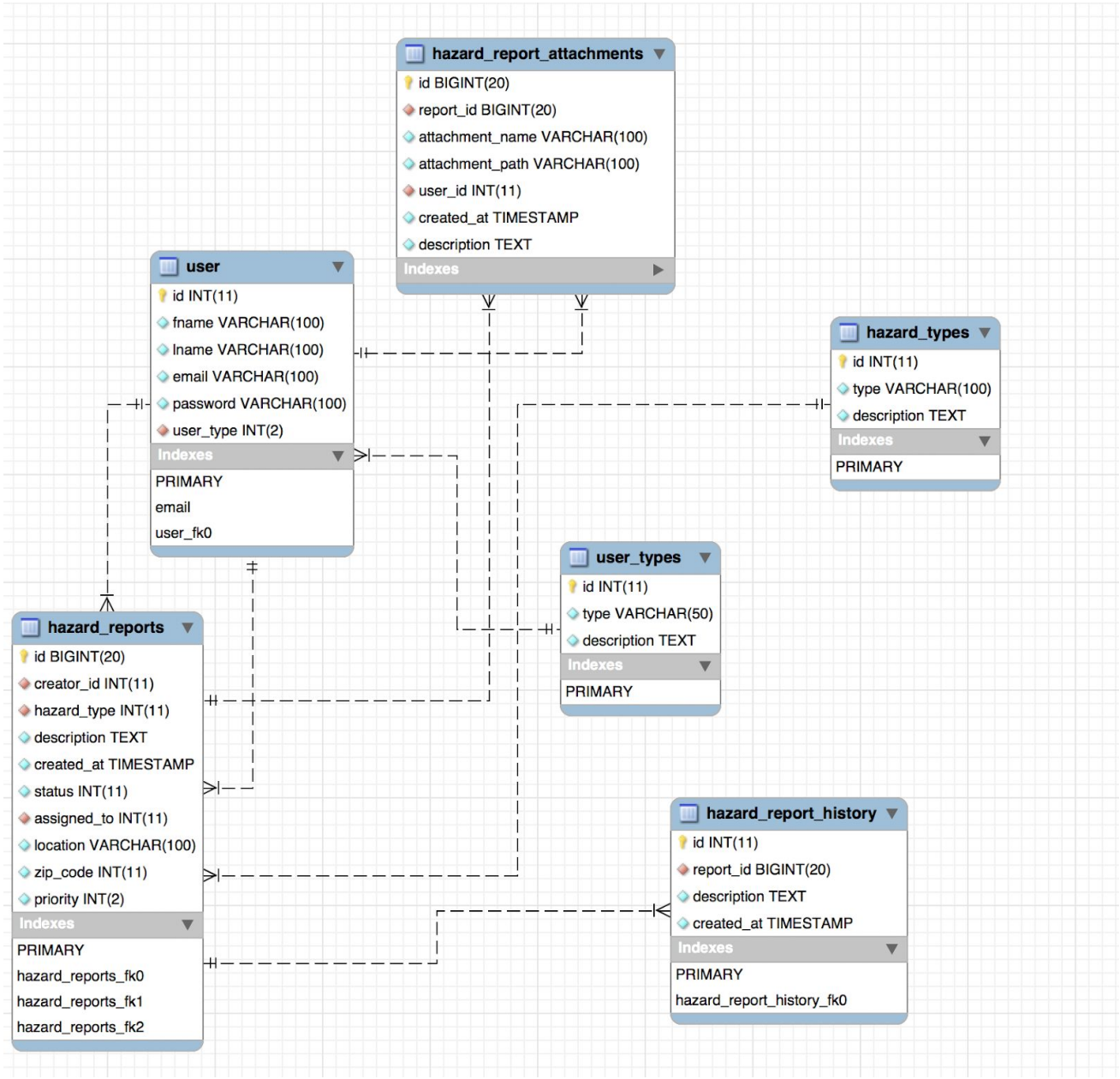
APIs:

Our application “Eco Hazards” shall use of APIs which are Google Maps and Google Analytics.
This application shall support these browsers: Chrome, Safari, and Firefox.

Admin User Interface:

The admin interface for the application shall be managed through MySQL Workbench, Tools and IDEs.
Our framework Django has a built in admin system which can be use to remove, edit, and add necessary information into the website.
The tools and systems that shall be used for the application are terminal/command line.
Team members wanting to use an IDE shall do so.

DB Organization (Database Schema ClearDB MySQL)



Media Storage

Our application will store image paths in our Database under a class, hazardAttachments(). The Hazard Attachments class will be able to determine the size of the image, the name, and any other related variables. The database could also store video/audio in a designated class.

Search/Filter Architecture and Implementation:

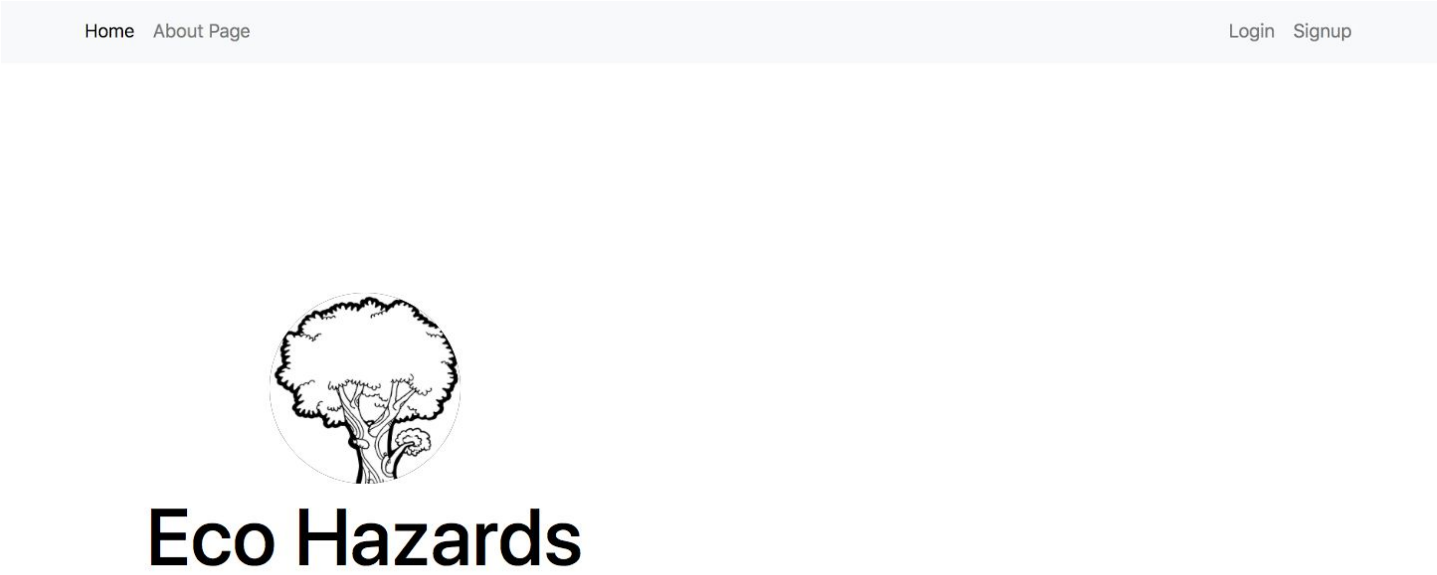
Database terms such as zip code will be use to determine the search for hazards report of a user. The user shall be able to search their community hazards report by inputting their own zip code in the following search bar on the homepage.

How it will be coded and organized in the DB:

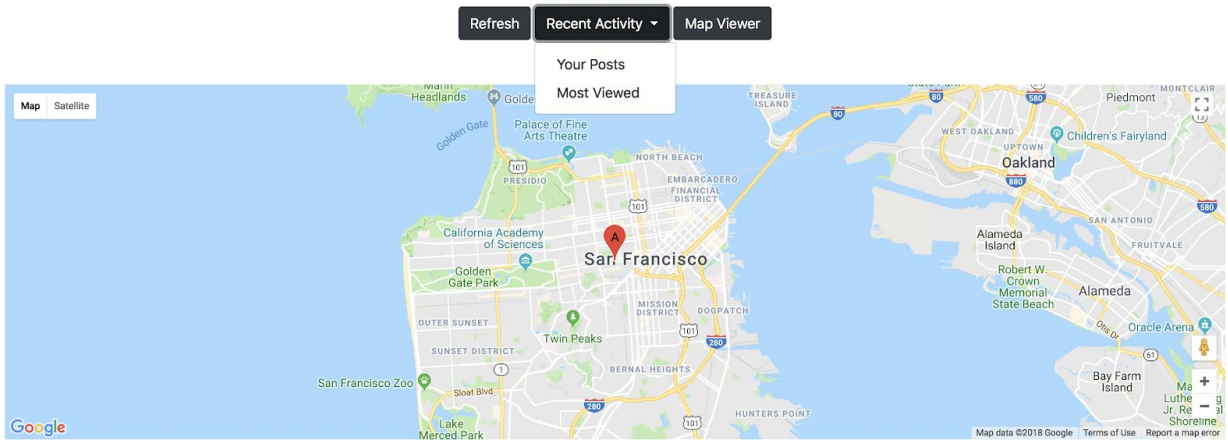
We will be using MySQL Workbench to code our Database. The database will be passed to our server and used for markers, user information, and server related permission. The database will have a huge roll in our Google Map implementation and allow us to pinpoint the users environmental hazard.

5. Content for vertical prototype (text, data, images…)

VERTICAL PROTOTYPE CONTENT



This is our current navbar on our main home page of the website. There will be changes to it in the future. For now it is a placeholder or a prototype on our website.



We have implemented a Google Map (given from Google Map API). The map will allow users to click on their destination (where there is an environmental issue). The information will be stored in our database. Above the Google Maps are a few buttons to navigate through the map. The refresh button will allow the user to refresh the map and the activity dropdown button will have functionality for different posts. The map viewer will contain hazard report locations (pinpointed locations.

We have implemented our way to search different post by inputting the zip code location of a hazard report. Once the the user click on the search button. It will render out the hazard report containing the zipcode matching the input zip code.

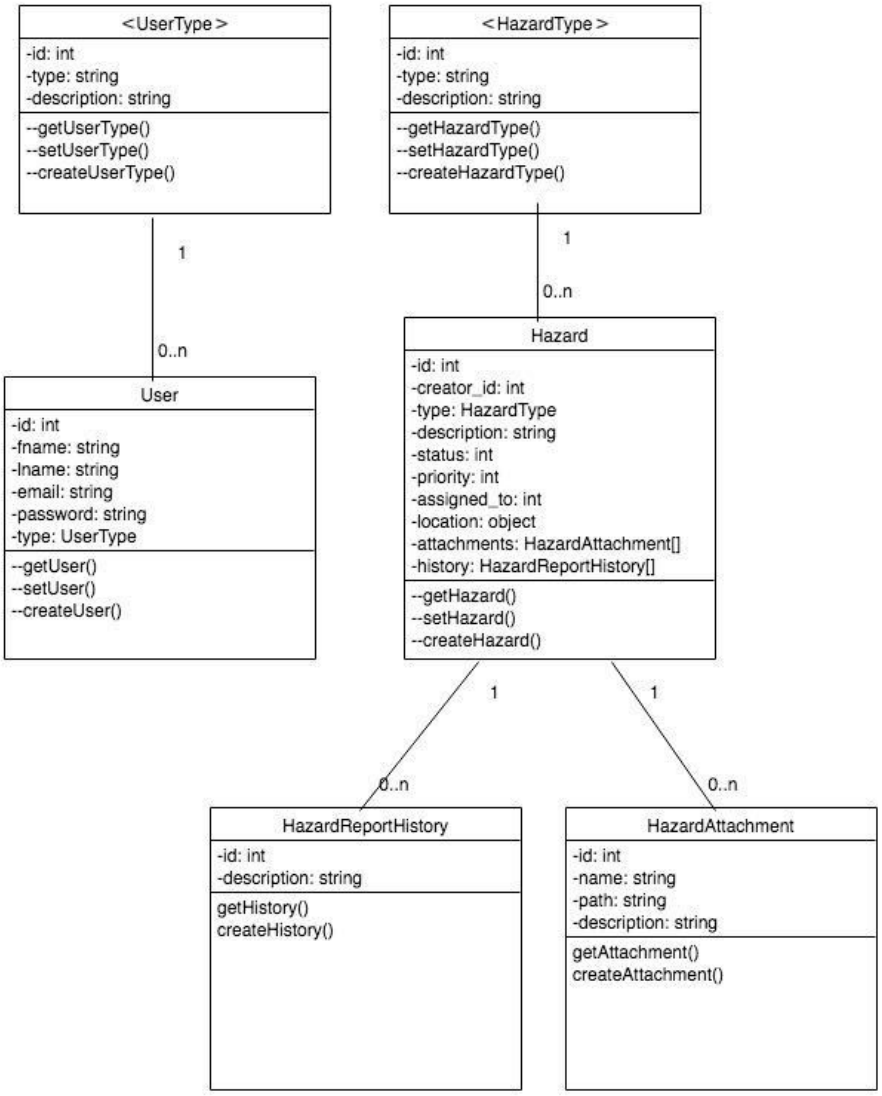
The red button “report a hazard” will be a button used to post a hazard report. This will transition to a view that will prompt registered users to talk in detail about their community hazard.

We also implemented the functionality of Tweeting on our application. A live tweet status from our very own SF Environment will be on the home screen for all users to see.

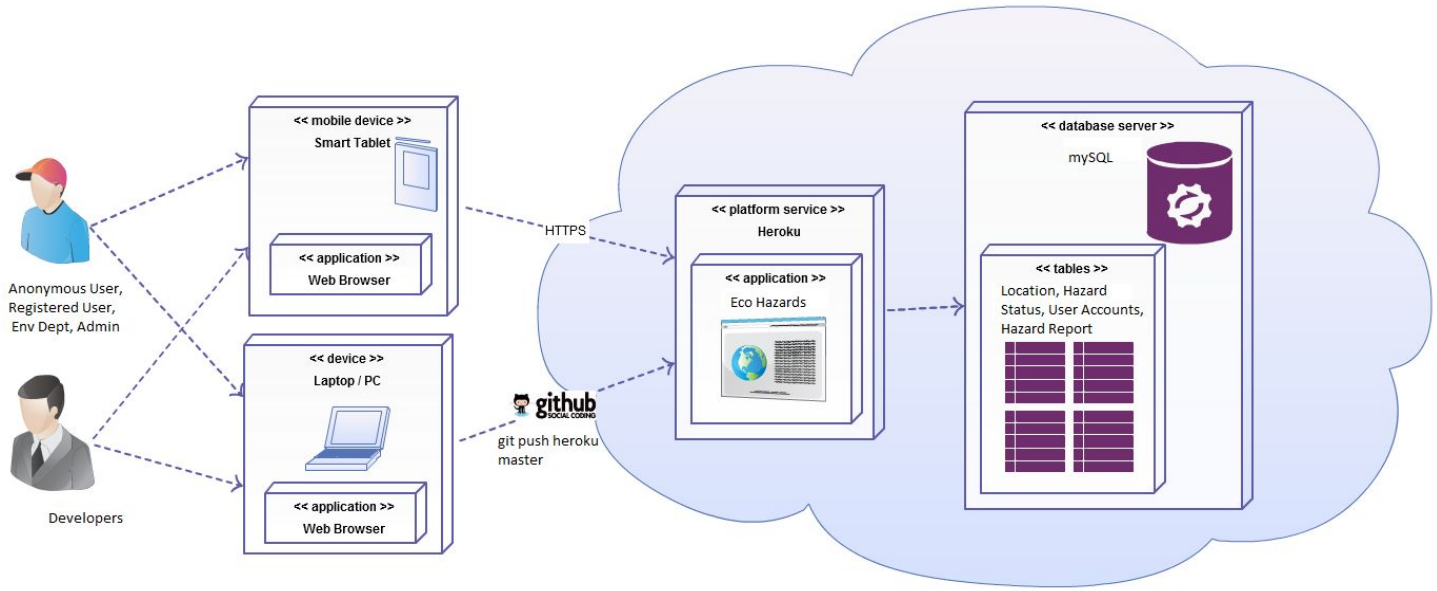
The Postings by Me and Recent Postings are located just underneath the Google Map. The recent postings will be implemented as the hazard reports that are posted at the current time of day. Hazard reports contains dates that will be use to determine recent postings.
Postings by Me will be implemented as a table for the users current postings. The Posting by Me will keep track of the users posts and also give status updates when available.

6. High Level UML Diagrams

a) High-level UML class diagrams



b) UML Component and deployment diagrams



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

- ***skills* risks**

Most of the team have little experience with django framework. The front-end and back-end team are capable of learning the skills needed to finish. Problems that cannot be solved between coordinating with other team members are solved with Google.

- ***schedule* risks**

The schedule of each team member vary. The meetings are solved mostly on slack. We also use Asana to communicate with team members on the tasks they will be completing each week.

- ***technical* risks** (any technical unknowns to solve),

Our team needs more experience on server side language (Python). Other risks include how the backend and frontend teams communicate. How the implementation of database and other

- ***teamwork* risks** (any issues related to teamwork),

A risk involved could be two team members working on the same functionality for the application, thus wasting time. This is why we use Asana to coordinate our efforts.

- ***legal/content* risks** (can you obtain content/SW you need legally with proper licensing, copyright).

Google Maps is included from the API. You are given a certain code to use specifically for “school work”. We also have our disclaimer on top of the application to make sure no one can mark our team for copyright.