Environmental Issue Reporter

SW Engineering CSC648/848 Section 01 Spring 2018

Team 05 - Super Heroes In Training

Team is local

James Quintero - jamesaquint@gmail.com

Stanley Liu

Danielle Nunez

Tumar Temirova

Alex Hernandez

Jianhao Zhong

Milestone 2

16 March 2018

Revision Number	Name	Date
1.0	Initial Design Documentation	3/16/18

1. Data Definitions V2

Avatar

A small image used to depict a user profile in posts and comments that may be a preset provided by the website or a custom uploaded image. The uploaded image will have a minimum width and height of 300px, and maximum width and height of 5000px. The image will be downsized to reduce file size. Supported image format: png, jpg, bmp. Max file size is 500KB

Ban

A status applied to a user that will limit their ability to use features of the website that registered users normally have. This status is applied to users that violate the site rules or show abusive behavior.

Bio

A section of the user profile that displays text input by the user that describes them.

Captcha

Used to deter bots from accessing site features. It is a small puzzle in which the user is asked to verify that they are human by the task of transcribing an image of words into text.

City Official

Special account for a city official or a representative of a city official. Can make official announcements.

Comment

Text submitted by users to provide extra information or their opinions on posts. Limited to 300 characters, and text only.

Filter search

An advanced search that will query results limited by keywords, date, original poster, or location. Results will be displayed on the search page, and will be limited to 30-50 results per page. Users can refine their search on the page, or browse more results by clicking through the pages.

Message

Private text sent from registered user to another registered user. Limited to text only, and 1000 chars.

Post

A environmental issue submitted to the site by a registered user. It will provide a location of the issue, images provided by the user, and a description. Images will be limited in size with a minimum width and height of 500px, and maximum width and height of 5000px. The images will be downsized to reduce file size and therefore page load speed. Supported image format: png, jpg, bmp. Max file size is 10MB each, which will be downsized to be around 500KB-1MB. Description will have a maximum character limit of 300 characters.

Profile Page for each user that displays user information, posts, comments, bio, and other information.

Registered user A user with an account, not necessarily logged in.

Report Users notify the site of inappropriate posts by choosing to report.

Subscribe Users can subscribe to posts or areas, allowing them to receive in-site notifications or emails of new posts.

2. Functional Requirements V2

Priority 1 (must have):

Unregistered users:

1.01 Users shall be able to register
1.08 Users shall be able to browse posts
1.10 Users shall be able to filter search by issue type - tag
1.13 Users shall be able to filter search by keyword in post - limited to area/timeframe

Registered users:

1.08 Users shall be able to browse posts
1.10 Users shall be able to filter search by issue type - tag
1.13 Users shall be able to filter search by keyword in post - limited to area/timeframe
2.02 Users shall be able to post about environmental issues
2.03 Users shall be able to include images and location in posts
2.07 Users shall have a profile page

City Officials:

1.08 Users shall be able to browse posts
1.10 Users shall be able to filter search by issue type - tag
1.13 Users shall be able to filter search by keyword in post - limited to area/timeframe
2.02 Users shall be able to post about environmental issues
2.03 Users shall be able to include images and location in posts
2.07 Users shall have a profile page
3.05 Users shall be able to post official statements that stand out from regular posts
3.06 Users shall be given a city official account by a site admin

Admins:

1.08 Users shall be able to browse posts
1.10 Users shall be able to filter search by issue type - tag
1.13 Users shall be able to filter search by keyword in post - limited to area/timeframe

Priority 2 (desired):

Unregistered users:

1.02 Users shall be able to sign up using email, FB, or Google
1.04 Users shall be required to create a name and a username
1.05 Users shall be required list the general area they live in the profile
1.06 Users shall be required to agree to age restriction of 13+
1.09 Users shall be able to filter search by area
1.11 Users shall be able to filter search by date - chronologically, or specific date

Registered users:

1.05 Users shall be required list the general area they live in the profile
1.09 Users shall be able to filter search by area
1.11 Users shall be able to filter search by date - chronologically, or specific date
2.04 Users shall be able to comment on posts
2.09 Users shall be able to create avatars or use one of the preset avatars
2.11 Users shall be required to specify the location in their post
2.12 Users shall be required to list tags that relate to the environment issue - limited
2.13 Users shall be to able to upload pictures, max-limit of 4 pictures
2.14 Users are required to write a description, max-limit of 300 chars
2.17 Users shall be able to flag posts
2.20 Users shall be limited in messaging text only
2.28 Users shall receive a strike if their post is reported and removed
2.29 Users shall be banned if they receive 3 or more strikes

City Officials:

1.09 Users shall be able to filter search by area
1.11 Users shall be able to filter search by date - chronologically, or specific date
2.04 Users shall be able to comment on posts
2.09 Users shall be able to create avatars or use one of the preset avatars
2.20 Users shall be limited in messaging text only
3.02 Users shall be able to link official environmental report website
3.04 Users shall have extra information on a unique profile

Admin:

1.09 Users shall be able to filter search by area
1.11 Users shall be able to filter search by date - chronologically, or specific date
4.01 Admin shall provide City Official accounts for city officials

Priority 3 (opportunistic):

Unregistered users:

1.03 Users, who sign up using email login, shall be required to use complex password
1.07 Users shall be required to agree to TOS
1.12 Users shall be able to filter search by person
1.14 Users shall be able to filter search by pages
1.15 Users shall be able to browse a heatmap of issues

Registered users:

1.12 Users shall be able to filter search by person
1.14 Users shall be able to filter search by pages
1.15 Users shall be able to browse a heatmap of issues
2.05 Users shall be limited to 300 characters in comments
2.08 Users shall be able to create a bio, max-limit of 500 chars.
2.10 Users shall be able to plug in their social media: FB, Twitter, Instagram & Linkedin
2.15 Users shall be able to disable comments
2.18 Users shall go through captcha to send a message
2.19 Users shall be able to limit messages received to an area
2.21 Users shall be able to report messages they receive
2.22 Users shall not be allowed to message City Officials
2.23 Users shall be able to subscribe to comments, but only if their own post
2.24 Users shall be able receive email updates on new comments
2.25 Users shall be able to receive notification on site on new comments
2.26 Users shall be able to subscribe to areas by area code, city, county
2.27 Users shall be able to subscribe to any new post or weekly/daily updates

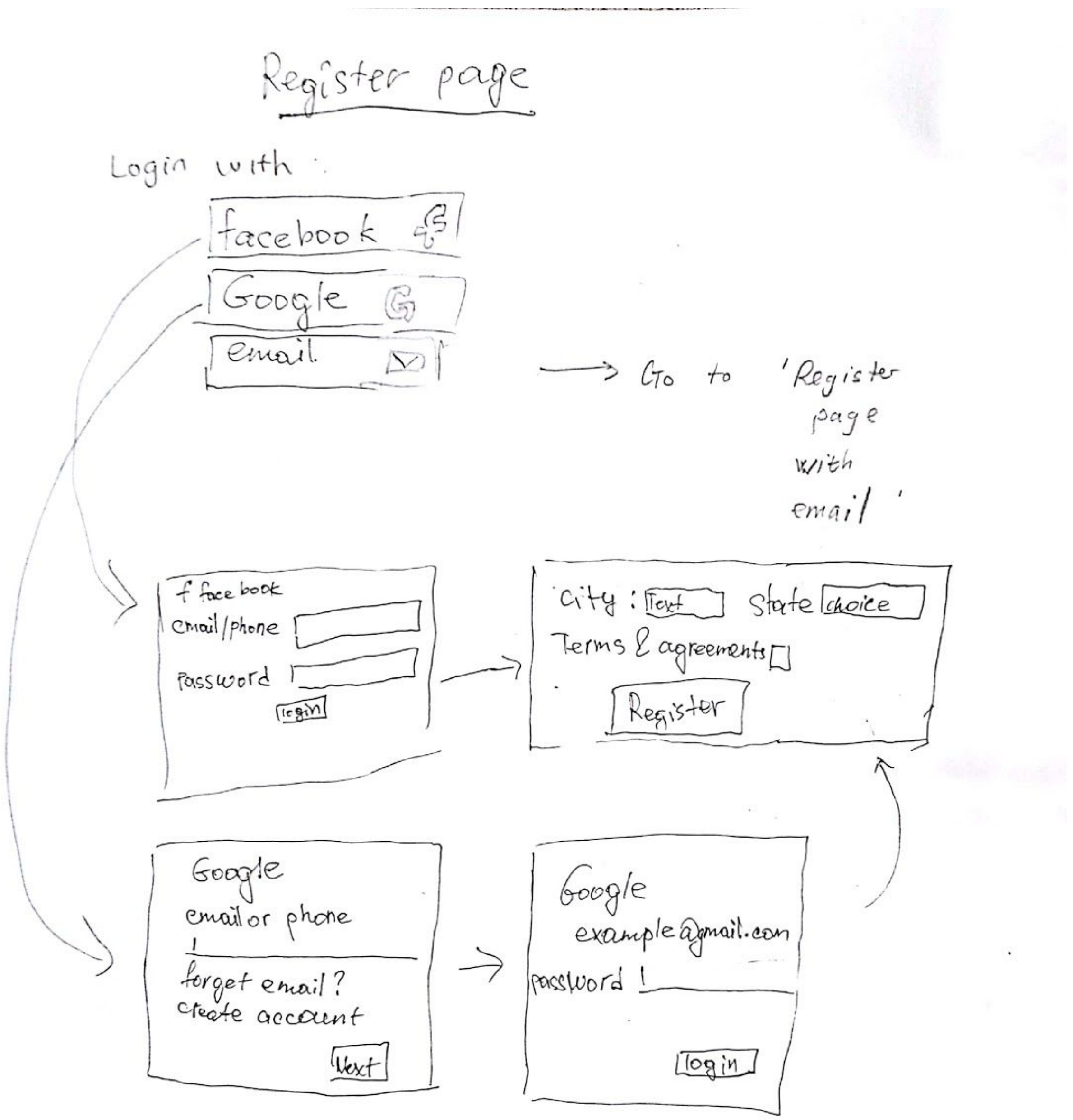
City Officials:

1.12 Users shall be able to filter search by person
1.14 Users shall be able to filter search by pages
1.15 Users shall be able to browse a heatmap of issues
2.05 Users shall be limited to 300 characters in comments
2.08 Users shall be able to create a bio, max-limit of 500 chars.
2.10 Users shall be able to plug in their social media: FB, Twitter, Instagram & Linkedin
2.15 Users shall be able to disable comments
2.19 Users shall be able to limit messages received to an area
2.21 Users shall be able to report messages they receive
2.23 Users shall be able to subscribe to comments, but only if their own post
2.24 Users shall be able receive email updates on new comments
2.25 Users shall be able to receive notification on site on new comments
2.27 Users shall be able to subscribe to any new post or weekly/daily updates

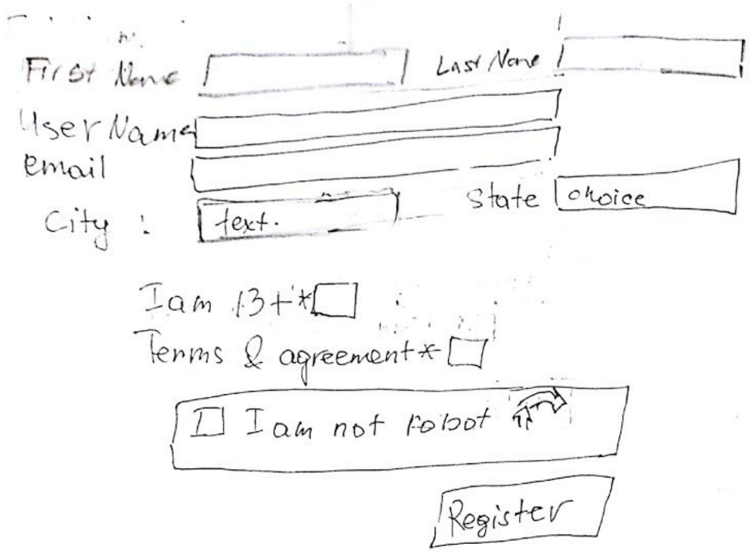
Admin:

1.12 Users shall be able to filter search by person
1.14 Users shall be able to filter search by pages
1.15 Users shall be able to browse a heatmap of issues
4.02 Admin shall ban users who’s bio contains inappropriate text
4.03 Admin shall ban users who post inappropriate posts
4.04 Admin shall ban users who post inappropriate comments

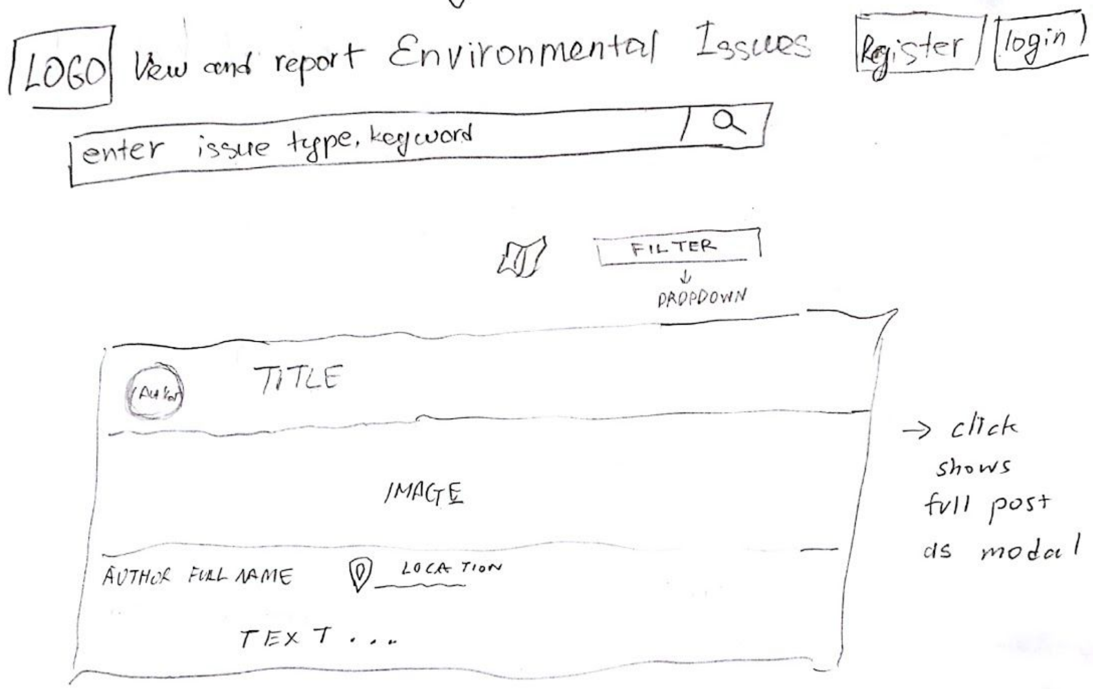
3. UI Mockups and Storyboards (high level only)



Register page with email

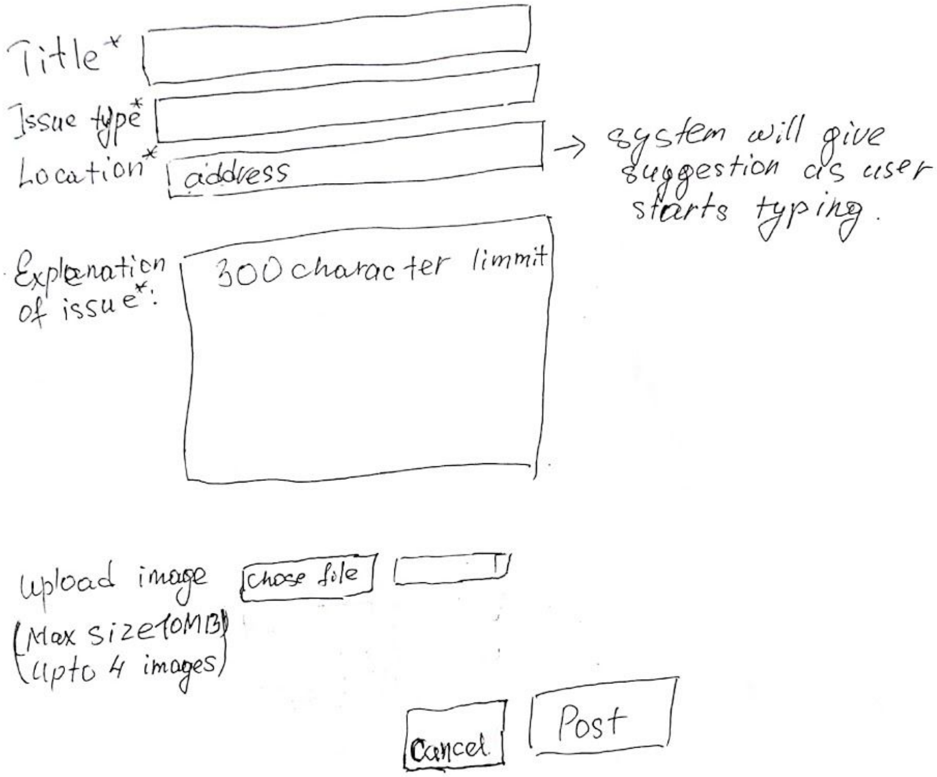


Home Page



→ click shows full post as modal

Make a post page



Title*

Issue type*

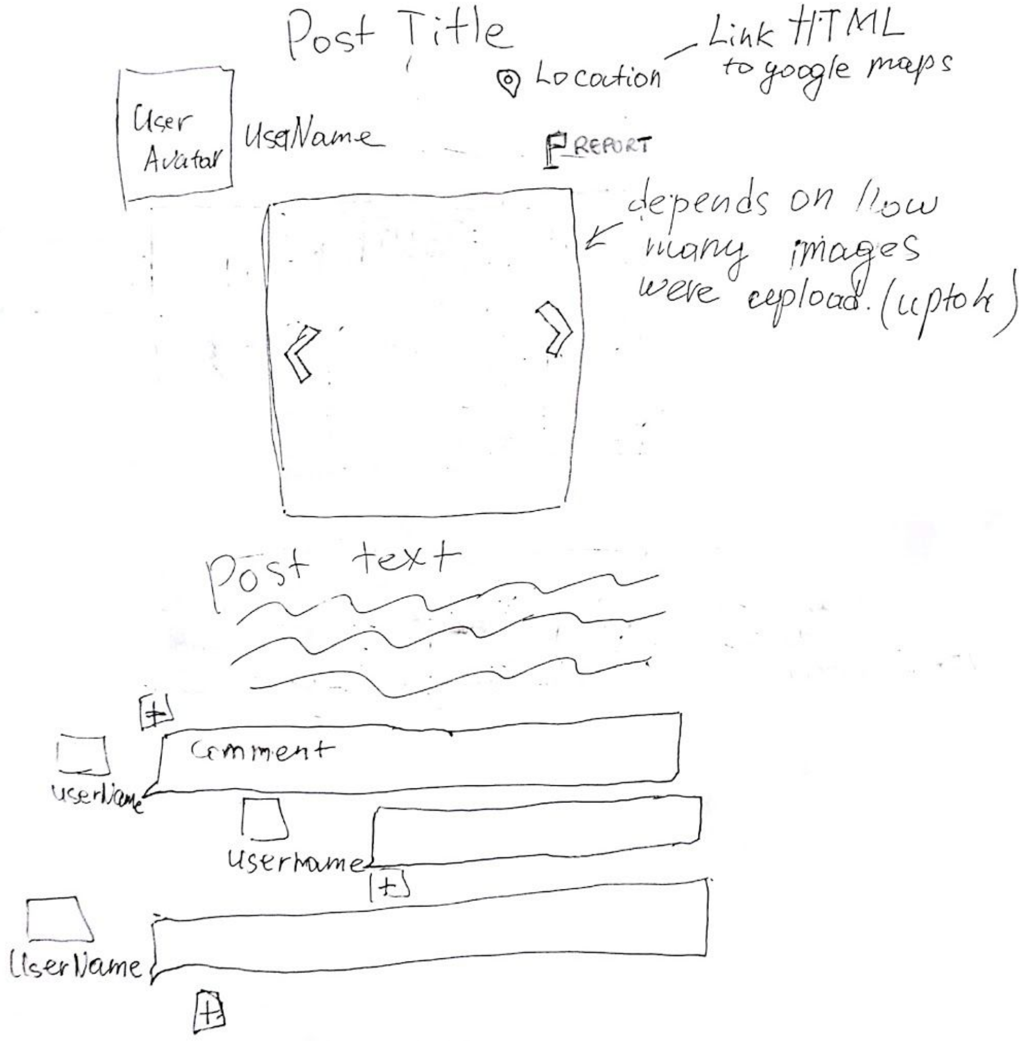
Location* address

→ system will give suggestion as user starts typing.

Explanation of issue*: 300 character limmit

Upload image (Max size 10MB) (upto 4 images)

Post pop page



Link HTML to google maps

depends on how many images were upload. (upto k)

MAP

SOME TEXT ABOUT THIS PAGE

WHEN USER
CLICKS ISSUE
POST DATA
SHOWS HERE

HEATMAP
ISSUES
MAP

4. High level Architecture, Database Organization

High Level Architecture:

MVC framework

Model:

- Python Class for each database table, where it solely deals with that table
- One main class will facilitate with queries that require multiple tables
- User class
    - EX: User class logs in user, and registers user
- UserData class
    - EX: User data class retrieves and updates user data
- Posts class
- Comments class
- Messages class
- Locations class

View:

- Front-end framework that the user sees.
- AJAX requests sends GET or POST requests to Controller.
- Javascript/jQuery gathers data to send to Controller
- Python receives data from Controller, and displays it in HTML.

Controller:

- View.py in Django uses the model classes in models.py to retrieve and send data.
- GET request handled by using model classes to retrieve data and returning it to View.
- POST requests handled by using model classes to save data, no information needed to return to View.

DB Organization:

Tables:

Users:

user_id - INT, Primary Key, Auto Increment
username - VARCHAR(255)
email - VARCHAR(255)
password - VARCHAR(255)
type - INT

strikes - INT

user_data

user_id - INT, Primary Key
first_name - VARCHAR(255)
last_name - VARCHAR(255)
biography - TEXT

Posts:

post_id - INT, Primary Key, Auto Increment
user_id - INT
username - VARCHAR(255)
location_lon_lat - VARCHAR(255)
issue_types - VARCHAR(255)
description - TEXT
timestamp - DATETIME
comments_user_ids - TEXT
comments - TEXT
comments_timestamp - TEXT
reports - TEXT

Issue_types:

Issue_id - INT, Primary Key
Issue_name - VARCHAR(255)

Messages:

user_id1 - INT
user_id2 - INT
messages - TEXT
Dates - DATETIME

For messages, a user can go to messages.html, and see the people they've ever messaged. Once they click on a certain user, their messages with that user pops up on the same page. For speed, we shouldn't have to load all messages for all users when the user first goes to the messages page. We should have a field in user_data or something that just has a list of user_ids that the user has messages with. When the user clicks on a user to view the messages, we then retrieve the messages that are between current user and clicked-on user. So in Messages table, we'll have two columns for user_ids, and the "current" user can be in either user_id1 or user_id2 with the other user being in the other column. Our SQL query will search for current user's user_id in user_id1 or user_id2, and recipient's user_id in user_id1 or user_id2.

Media Storage:

Post images will not be stored in the table as blobs, since it's better to store them in S3 buckets for available and scalability. There's no need to store path names in the table, since our directory structure in S3 will follow a format like “./Posts/images/[user_id]/[post_id]/image1.png”. This will allow us to not be limited by the number of images we allow, and for us to change the directory structure without modifying image path values in the database.

Search/filter architecture and implementation:

- Keyword:

  ```
  SELECT * FROM Posts WHERE Posts.description LIKE ”%keyword%”
  ```

  Backend will sort results by the number of instances of keyword in returned post descriptions
- User (username):

  ```
  SELECT * FROM Posts WHERE Posts.username=username
  ```

  Only return posts from the exact user. Otherwise a user can search for username “e” and have returned millions of posts, putting a strain on the server.
- Issue Type:

  ```
  SELECT * FROM Posts WHERE Posts.issue_type LIKE ”%issue_type%”
  ```

  Issue_type will be selected from an approved list of issue types on the frontend and backend.
- Area (location):

  ```
  SELECT * FROM Posts WHERE Posts.location=“location”)
  ```

  Will try and use lon/lat for locations. When a user wants to search by location, we have them find the location on a google map, and google maps will give us the proper lon/lat of that location.
- Date:

  ```
  SELECT * FROM Posts ORDER BY Posts.date ASC LIMIT X
  ```

  Will limit the number of returned posts so that not all millions of posts will be returned. The limit will be determined by the search page number the user is on and the number of results that are displayed per search page.

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

Mockup Posts:
Post 1:

Location:
Olompali State Historic Park
8900 Redwood Blvd, Novato, CA 94945

Image links:

- https://www.marinfirehistory.org/uploads/4/6/1/8/46186139/5315059_orig.jpg
- https://www.marinfirehistory.org/uploads/4/6/1/8/46186139/mcfd-nicasio-barn-fire-nicasio-valley-road-mcfd-photo-july-19-2017-copy_orig.jpg

Description:
Wildfires
The fires was the most destructive; it had already destroyed more than 1,000 buildings and homes.

Post 2:

Location:
Lake Henshaw
California 92070

Image links:

- https://www.lotus-cortina.com/images/sandiego/henshaw.jpg

Description:
Drought
The drought reduced Lake Henshaw to its lowest level ever; its capacity just under 500,000 acre-feet.

Post 3:

Location:
Downtown Los Angeles, CA

Image links:
http://i.dailymail.co.uk/i/pix/2017/08/11/01/4326634C00000578-4780306-image-a-3_1502411548692.jpg

Description:
Extreme Heat
The area has been a heat island; the temperatures hover 15 degree F above the average high temperature for two weeks.

Post 4:

Location:
Lake Merced Park
Skyline Blvd & Harding Rd, San Francisco, CA 94132

Image links:
http://www.latimes.com/resizer/yP_OXOQuENeTjq2At5V3AZMZJzM=/1400x0/arc-anglerfish-arc2-prod-tronc.s3.amazonaws.com/public/7JYTCVBYUNC3RLXYESV3OZH4TE.jpg

Description:
flooding
Lake Merced's water level has increased; it has climbed 10 feet in just two days.

Post 5:

Location:
San Francisco, CA

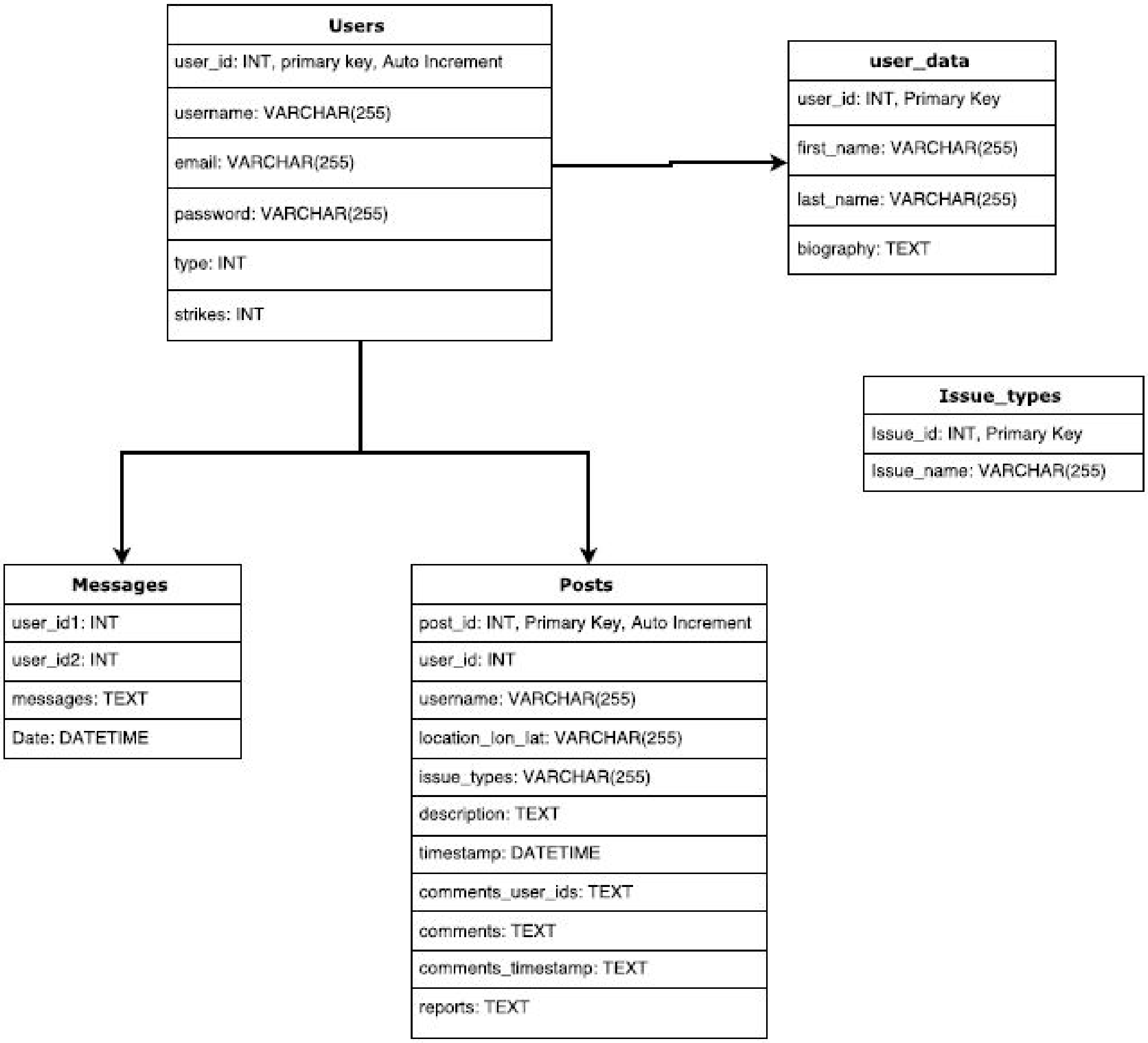
Image links:
https://cdn.cnn.com/cnnnext/dam/assets/140501131224-01-polluted-cities-0501-restricted-horizontal-large-gallery.jpg

Description:
Air pollution
The Air Quality Index of 404 for small particulate matter was 255; children and adults should avoid all outdoor exertion.

6. High Level UML Diagrams

High-level UML class diagrams

.



Users

user_id: INT, primary key, Auto Increment

username: VARCHAR(255)

email: VARCHAR(255)

password: VARCHAR(255)

type: INT

strikes: INT

user_data

user_id: INT, Primary Key

first_name: VARCHAR(255)

last_name: VARCHAR(255)

biography: TEXT

Issue_types

Issue_id: INT, Primary Key

Issue_name: VARCHAR(255)

Messages

user_id1: INT

user_id2: INT

messages: TEXT

Date: DATETIME

Posts

post_id: INT, Primary Key, Auto Increment

user_id: INT

username: VARCHAR(255)

location_lon_lat: VARCHAR(255)

issue_types: VARCHAR(255)

description: TEXT

timestamp: DATETIME

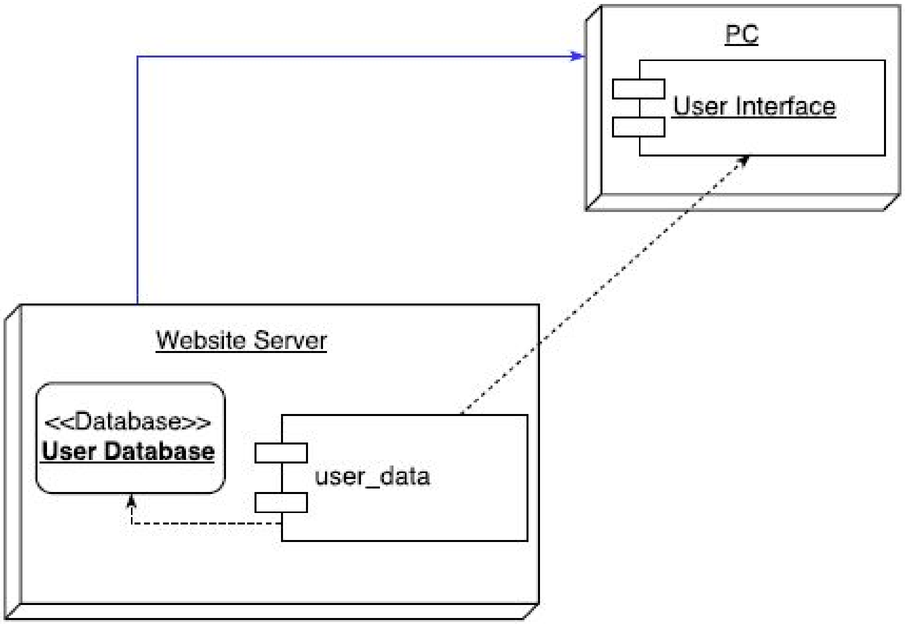
comments_user_ids: TEXT

comments: TEXT

comments_timestamp: TEXT

reports: TEXT

UML Component and deployment diagrams



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

Skills risks: Our team doesn’t have anyone experienced with Django, and there are only a few members who have experience in Python. These risks will be resolved by the members who don’t know python, learning python through online tutorials. Those who already know python will learn Django through online tutorials, then share documentations and instructions to those who are learning python.

Schedule risks: These is no risk to our team not delivering on each milestone on time.

Technical risks: dealing with user sessions in django, google maps API, and data upload/download from AWS S3 are unknowns. These will be solved by following example programs that implement them, and online tutorials.