**Data Definition v2**

Admin - privilege to ban user’s IP if certain users violate forum 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

Category: Entities Category

* Issues’ Category

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

Environmental 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.

**High Level system Architecture**

We have design our application with the Model-View-Controller (MVC) architecture in mind. This will allow our code to be well organized, and therefore easy to maintain. Our Model is where our application’s data objects are stored. We have 3 model objects; issues, parks and users each with unique attributes associated with each object. The View is the information presented to the user (client) and how the user interacts with the applications across multiple interfaces.

We use Handlebars.js as a template engine to generate our HTML and Bootstrap’s framework for quick prototyping and development. The Controller is where the logic and communication between the Model and View take place. With event listeners we trigger queries to PUT / POST / GET from the database and relay that information to the View

**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

**DB Organization**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Category\_id | Category\_user\_FN | Category\_LN | Email | Username | Post\_Title |
| 0 | Matther | Emory | Memory12@mail.com | Memory1 | Clean the dog poop! |
| 1 | Ravenna | Ocean | [ravenocean@mail.com](mailto:ravenocean@mail.com) | Rocean2 | This is nasty |
| 2 | Dory | Tahnee | [dorytahhh@mail.com](mailto:dorytahhh@mail.com) | Dtahnee3 | Beautiful |
| 3 | Elwyn | Dallas | [edallas@mail.com](mailto:edallas@mail.com) | Edalla56 | Dog walk |
| 4 | Kent | Harpers | [Kh24@mail.com](mailto:Kh24@mail.com) | Kh24 | Good place for cooking meth |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Category\_id | Category\_park\_name | zip | Location | Coordinate | Description |
| 0 | Golden Gate Park | 94122 | SF,CA,US | 37N 122W | Nice |
| 1 | Mission Dolores Park | 94114 | SF,CA,US | 37N 122W | Crowded |
| 2 | Twin Peaks | 94114 | SF,CA,US | 37N 122W | High Elevation |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Category\_id | Category\_issue | Park\_name | Issue\_type | Image | Author |
| 0 | Dog Poop | Golden Gate Park | Sanitary | “dog poop image” | Matther |
| 1 | Discussion | Dolores Park | Weather | “Dolores park image” | Dory |
| 2 | Miscellaneous | Twin Peak | Entertainment | “Cooking meth image” | Kent |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Status\_id | Category\_Issue\_name | Category\_Status |
| 0 | Dog Poop | Processed |
| 1 | Weather | Active |
| 2 | Cooking Meth | Removed |

Media Storage: All medias 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.

Personal API: An api to track how many parks are there in SF