

# **SW Engineering CSC648/848 Section 01 Spring 2018**

## **PARSE**

Pollution and Accident Reporting for a Safer Environment

### **Milestone 2**

March 18, 2018

### **Team 04**

Rodrigo Bell (rbell594@gmail.com)

Charlie Tuttle

Rohan Patel

Dion Matthew Lagos

Kamran Khadivi-Dimbali

Michael Schwiebs

Zhenru Huang

#### *Revisions*

March 18, 2018	Initial Draft
----------------	---------------

# Table of Contents

<b>Data Definitions . . . . .</b>	<b>5</b>
<b>Functional Requirements . . . . .</b>	<b>6</b>
<b>UI Mockups and Storyboards . . . . .</b>	<b>7</b>
<b>High-Level Architecture . . . . .</b>	<b>8</b>
<b>Database Organization . . . . .</b>	<b>9</b>
<b>UML Diagrams . . . . .</b>	<b>10</b>
<b>Project Risk Assessment . . . . .</b>	<b>11</b>

# Data Definitions

**Unregistered user:** Able to view environmental issue posts. No login required.

**Registered User:** Same abilities as unregistered user AND is able to post about environmental problems in a specific neighborhood. Account creation requires email, username and password. Must login to access personal posts, saved posts.

**Administrator:** Same ability as registered user AND accept and delete environmental issues. Ensures the proper user and content policies are followed.

**Development Team:** Same ability as registered user AND can update status on a post. Also has access Priority Issue Tool. The development team will assign and manage site administrators.

## **Environmental Issues Posts (Visible to all):**

- Images of issue
- Maps/location/address
- Description of problem
- Option for user to save a post

## **Priority Issue Tool (Development Team use only):**

- Analytics for service company to evaluate priority of issues

# **Functional Requirements**

## **Priority 1 – MUST HAVE**

### **Unregistered Users (guests):**

1. Application shall display location of posted environmental issues on Google Maps.
2. Application shall provide a search functionality filtering by keywords.
3. Application shall allow guests to register for a User account.
4. Application shall display new and pending environmental incident listings.

### **Registered Users (in addition to unregistered user functions):**

5. Users shall be required to register with full name, email, and password.
6. Users shall be able to provide a photo and a description of the environmental issue.
7. Users shall have permission to create postings of environmental issues to the database.

### **Administrator:**

9. Administrators shall have permission and power to remove postings.
10. Administrators shall have the permission and power to remove any type of account or listing.

### **Development Team:**

11. Development team shall have access to a dashboard to choose which issues to resolve.
12. Listings can only be resolved by Service Company accounts or Administrators.

## **Priority 2 – MEDIUM IMPORTANCE**

### **Unregistered Users (guests):**

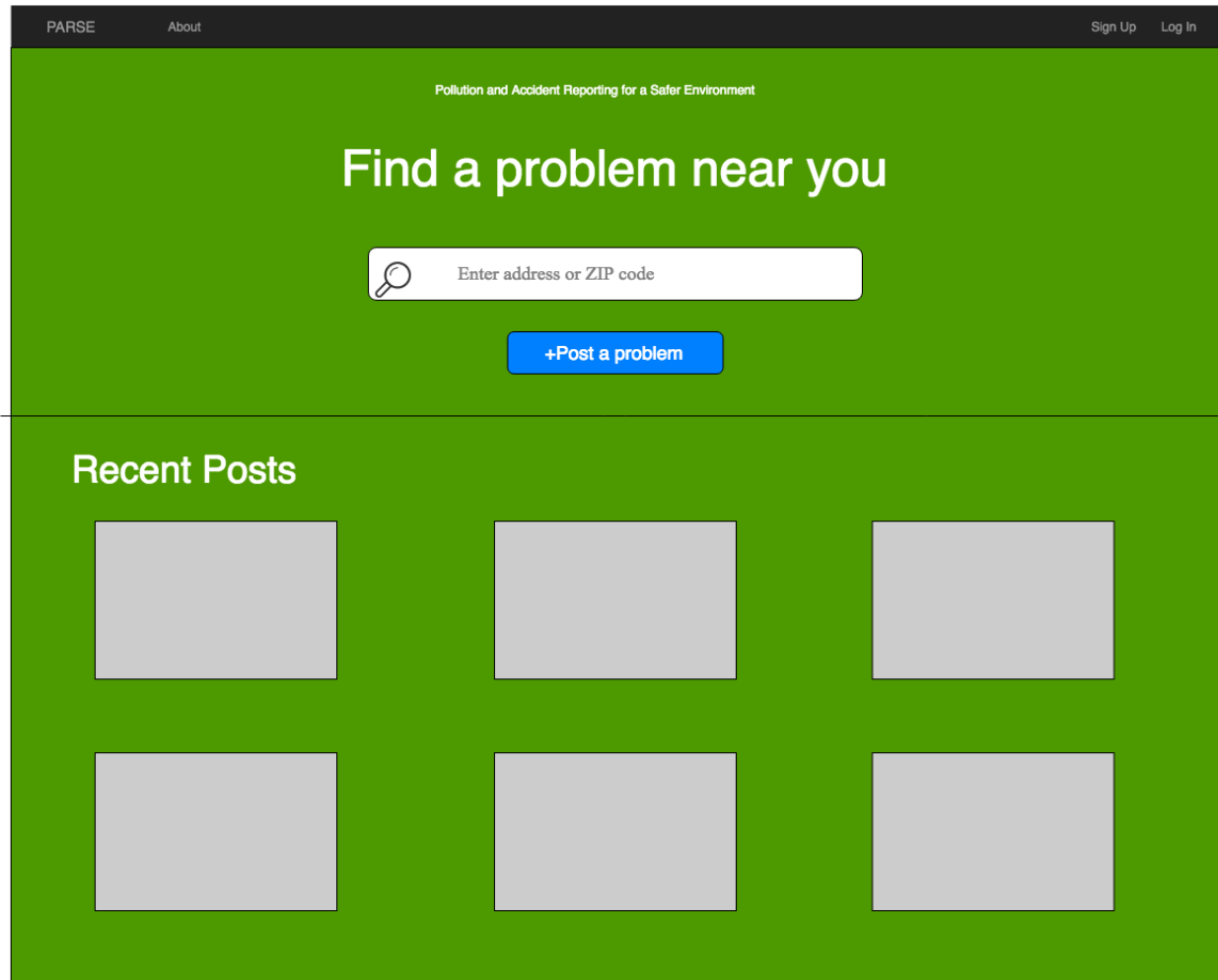
1. Application shall provide a search functionality filtering by address, zipcode, or category.

### **Registered Users (in addition to unregistered user functions):**

2. Users shall have a log of reported, pending, and resolved issues saved to their account.
3. Users shall be able to make bookmarks of environmental issues.

# UI Mockups and Storyboards

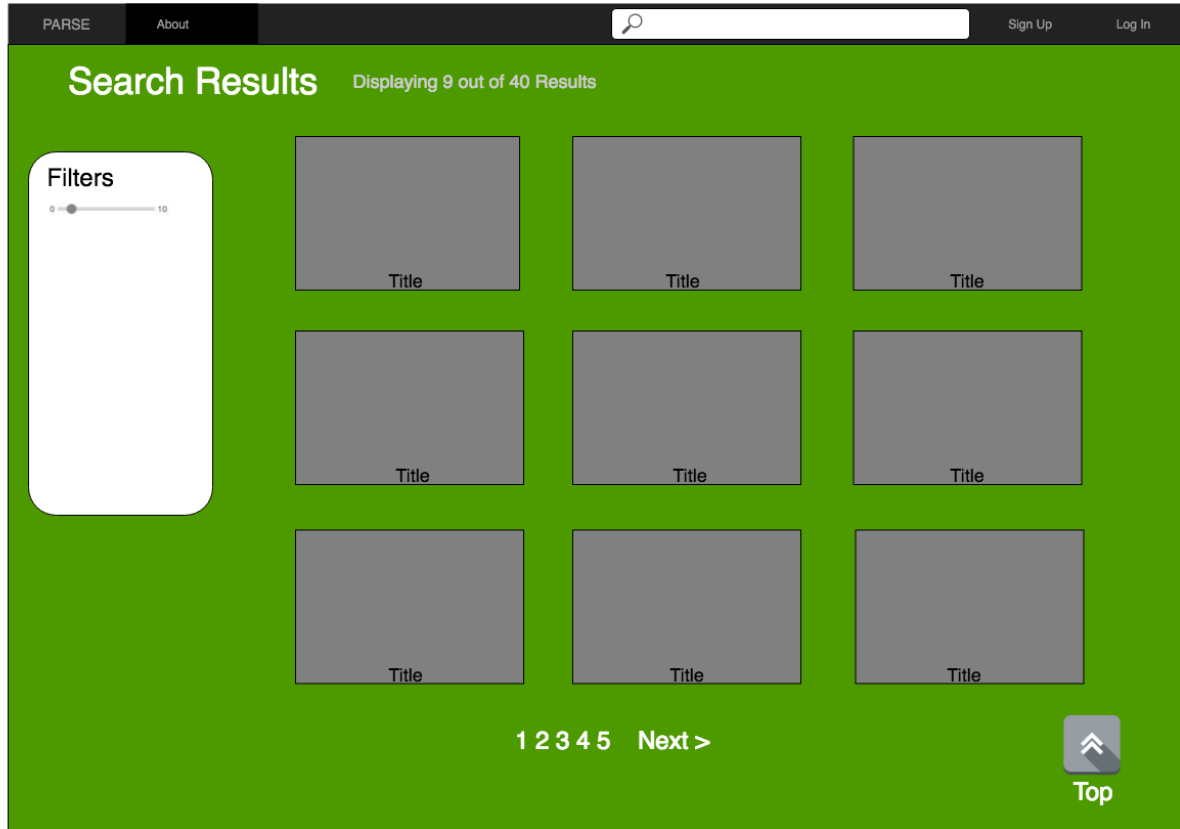
## Landing Page



## Use Cases

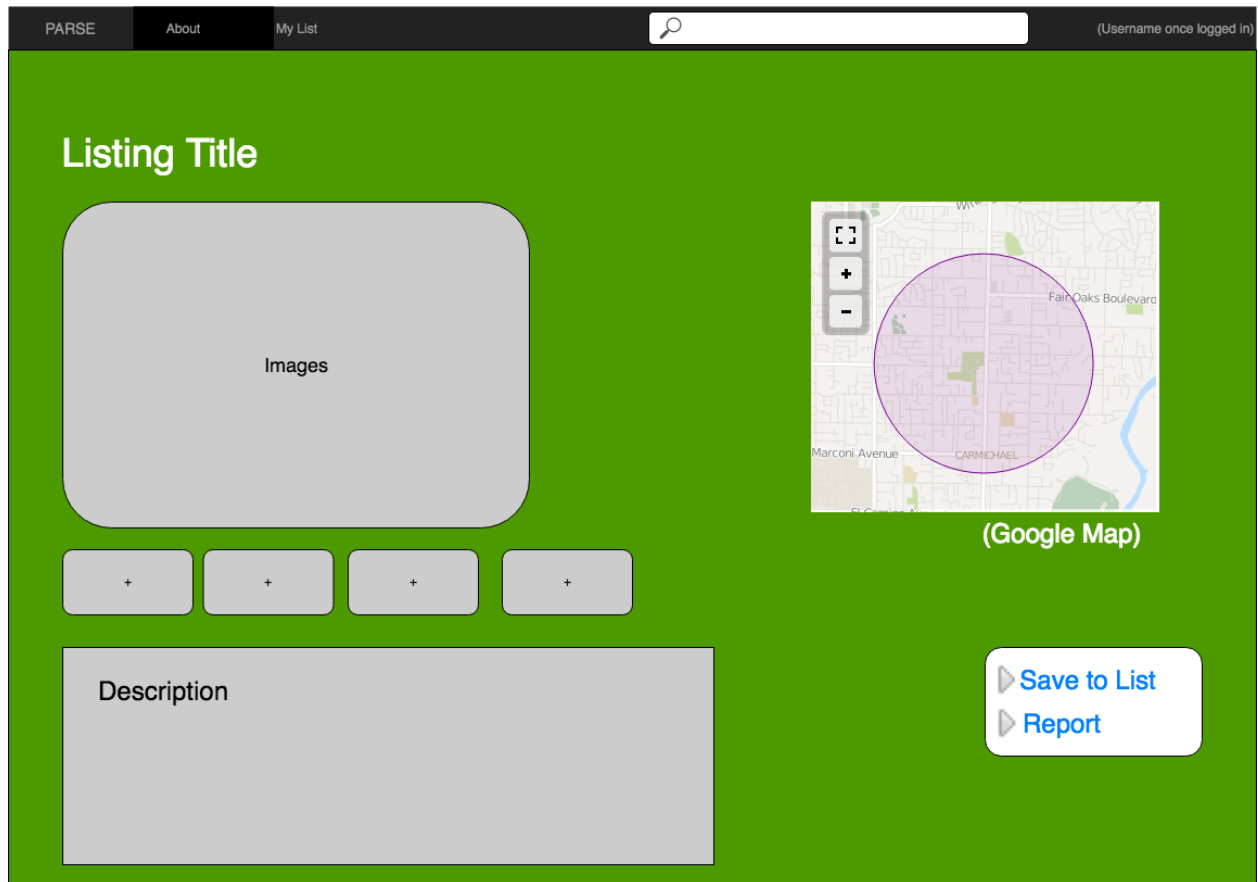
- **Unregistered User:** Able to search/view posts, sign up in the top right corner.
- **Registered User:** Able to sign in at top right corner, post/search/view a post.
- **Service Company:** Same abilities as registered user.

## Search Results Page



**All users:** This page shows the listings according to the search. You can also refine the search by using the filter tool, with specifications coming in the future. The user can go to the next page by clicking the “Next>” button or clicking the page number. Search bar is included if the user wants to do a different search.

# Individual Post Page




**Unregistered User:** Can only view the post.

**Registered User and Service Company:** Able to save to their personal list and report the post if it does not abide to the terms of agreement.

# Login Page

PARSE

About



Sign Up

Log In

Pollution and Accident Reporting for a Safer Environment

login

username

password

☐ remember

forgot your password? click here

new user? create new account

**All Users:** Registered users enter their username and password.



# Sign Up Page

PARSE About  Sign Up Log In

Pollution and Accident Reporting for a Safer Environment

**Sign Up**  
Please fill out the form below to create a new account.

**Full Name**

**Username**

**Email Address**

**Password** **Repeat Password**

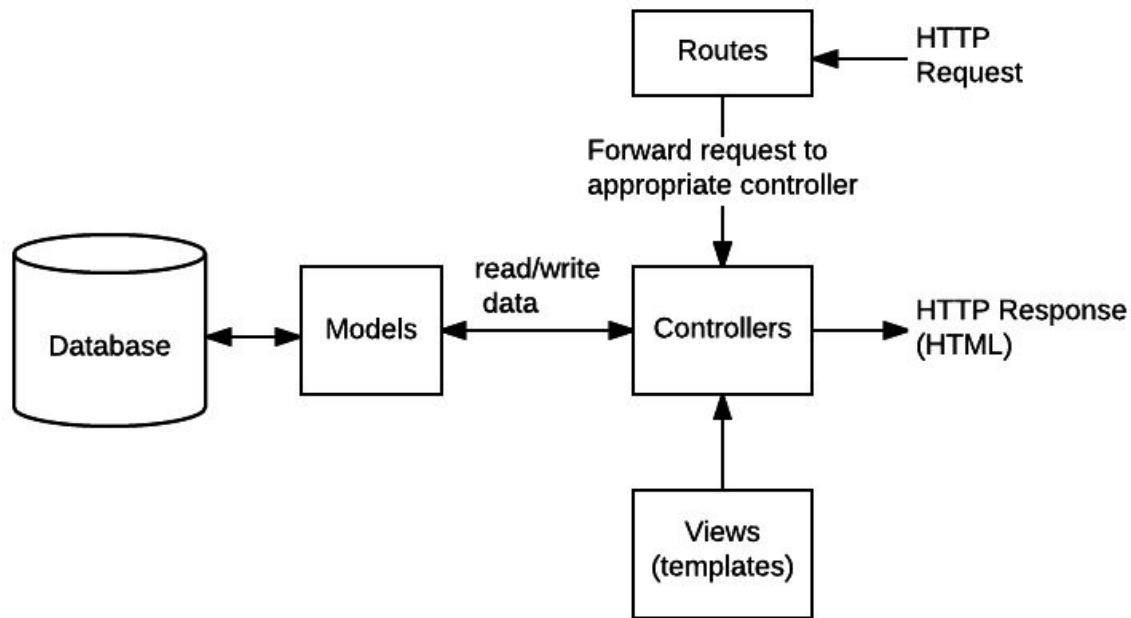
☐ Check box if you are a city service company administrator

[Create an account](#)

**All users:** Sign up by filling the following form. If you are from the service company, you check the box as you finish creating your account.

## High-Level Architecture

For this project, we will use the general MVC pattern which is under Express framework in Node JS.



Source: [https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\\_Nodejs/routes](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/routes)

The diagram shows the MVC pattern we are going to use for our project. First user will send a HTTP request to our Controller. And with this request, the Controller will request the Model to call the Database to get the data that the user request. And then the Model will send the data back to our Controller. Finally, our Controller will use these response data and render them into our View templates and send it back as HTTP or HTML response.

# Database Organization

## Schema

category	issue	user
-id(PK)	-id(PK)	-user_id(PK)
-name	-title	-name
	-category(FK)	-email
	-description	-phone
	-zipcode	-password
	-image	-isAdmin
	-isApproved	
	-isResolved	
	-user_id(FK)	
	-latitude	
	-longitude	

## Image System

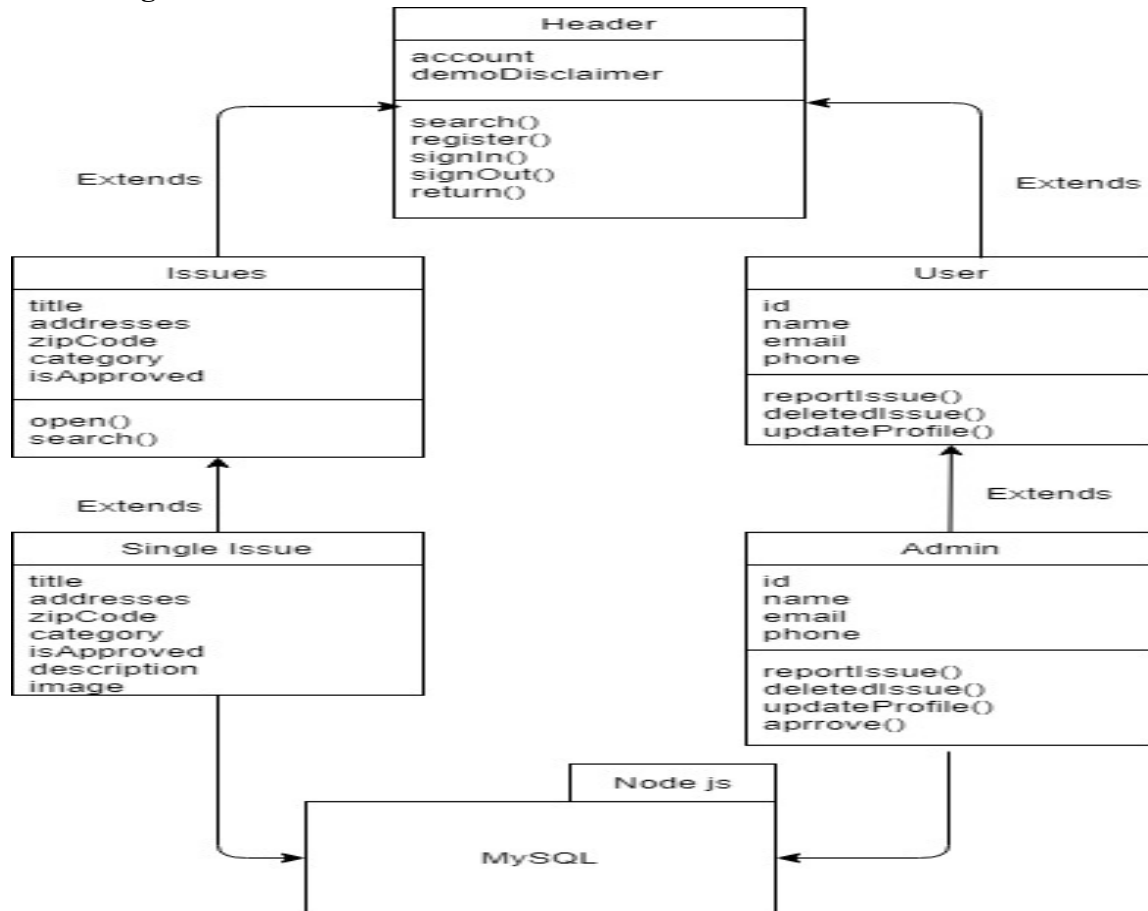
For images, we will use the file system to store it. We will have a path for images source. Once the users try to add images, the images will all be uploaded to our cloud under this path. And our database system will store the path and name of these image files. And once we try to use these images, we can easily pull up their path from our database and get it from our server.

## Search Architecture

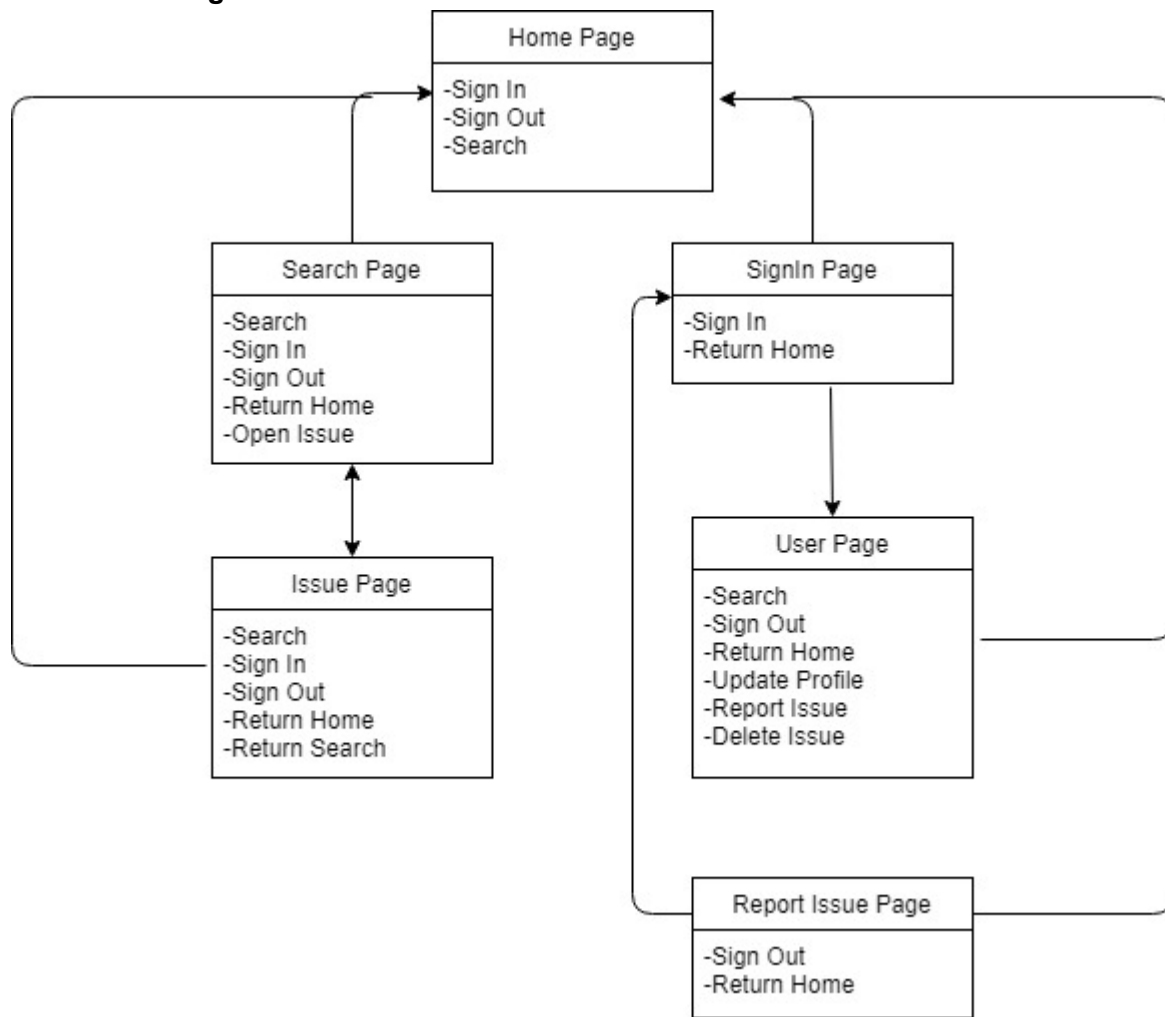
For search system, we will use key word search. Once the title or description contains the key word which is typed in by the users, we will return a list with all that results. And we are also going to have a dropdown menu for categories, so users can filter their results with different category. And also the users can type in the zip code and get the results of nearby.

# UML Diagrams

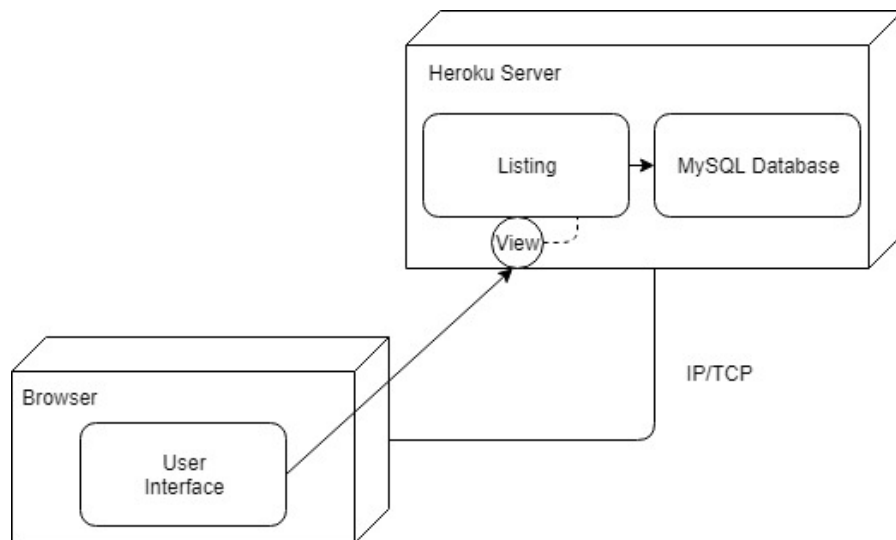
## Class Diagram



## View Flow Diagram



## Component and Development Diagram



# Project Risk Assessment

Upon review of the organization of the PARSE web application and its development team, the following risks have been identified:

- *Skills risks*: the majority of the team is unfamiliar with the given development stack (node.js/express) for the project, and may need accommodate time to gather and learn from various resources.
- *Schedule risks*: as the development team is comprised of senior level undergraduate students, scheduling conflicts may emerge due to individual projects or jobs.
- *Legal/content risks*: this project requires the use of media (uploaded images) which may potentially infringe upon the rights of its owners.

In order to resolve the identified risks, SCRUM-style meetings will be held between the team lead and the rest of the team weekly either online or in person to discuss any issues with the application development. At scheduled extended meetings, we will periodically review the coding procedures that need to be done, and research the methods of accomplishing them within the chosen framework and development stack. Code reviews will be conducted within the smaller teams to resolve any coding issues. The back-end team will work together to document a vertical prototype and educate the rest of the team.

During the weekly SCRUM-style meetings, we discuss availabilities for the week and plan for any scheduling conflicts that may arise with any of the members. To ensure deadlines are met, tasks are assigned with soft deadlines depending on the availabilities discussed. An extended weekly meeting is typically scheduled to discuss issues in depth at these meetings. We will keep the scope of the project to a minimal P1 set as development continues on.

For demo purposes, our project will use images of properties. In order to avoid any legal issues, all images must be reviewed by team leads before pushed to the live project site. The sources will be checked for the distributive rights of the images - only commercial-free (and no attribution required) images will be approved for demo use.