# SW ENGINEERING CSC 648/848 SUMMER 2018

# FAWBYR PROJECT

## TEAM 1

## Global

Flavy Tonfack ftonfack@mail.sfsu.edu

Ben Clark

Razmik Hakobyan

Andy Lai

Wenjun Zhu

Yusen Ma

## Milestone 2

**07/08/2018**

**Date submitted for review:** 07/08/2018

# 1. DATA DEFINITIONS

1. **User**
    - user_id (PK)
    - e-mail
    - password
    - type
2. **Image**
    - Image_id (PK)
    - user_id (FK: References User)
    - Category_id (FK: References Category)
    - image_name
    - thumbnail
    - med_res
    - high_res
    - description
    - approved
3. **Category**
    - category_id
    - category_name
4. **Comment**
    - comment_id (FK: References Image)

- user_id (FK: Rferences User)
- comment

**5. Rating**

- rating_id(FK: References Image)
- user_id (FK: References User)
- rating

# 2. FUNCTIONAL REQUIREMENTS

### 1. Search Bar

1.1 Users shall be able to search keywords with specific categories to limit the images presented to the user.

1.2 The search bar shall be at the same location on each page and keep track of what the user had searched to have consistent UI.

1.3 The search bar shall be usable for all users in order to search images for viewing.

1.4 The search bar shall save most recent query after clicking submit in order to keep track of what the user had searched.

### 1. Registering for Website

1.5 Users shall be able to sign up for an account on our website in order to keep track of their activity on the website and to interact with other users.

1.6 Users' email and password shall be used for authentication and shall give them access to major functionalities on the website.

### 1. Uploading Images

1.6 Users shall be able to upload images that is connected to their username only if they have properly signed up with our website.

1.7 The uploading image form shall also allow users to set descriptions to their images.

**1. Downloading Images**

1.8 Users shall be able to download images with a single button on an image post

1.9 Users shall be able to download images without a protected watermark if they have properly signed up with our website.

**1. Admin Ability**

1.10 Admin shall be able to take down any posts/pictures from public use/viewing, but still saved onto database.

1.11 Approve incoming images for appropriateness and proper usage of the website

1.12 Admin shall be able to view rejected/approved posts

**2. Like/Commenting on Images**

2.1 Users shall be able to comment on specific images to create a social networking platform.

2.2 There shall also be a like system for users to show their appreciation to specific photo posts.

**3. Share on Social Media Platforms**

3.1 Users shall be able to send a short URL to share their image post on social media platforms such as Facebook that shall be formatted as a preview link.

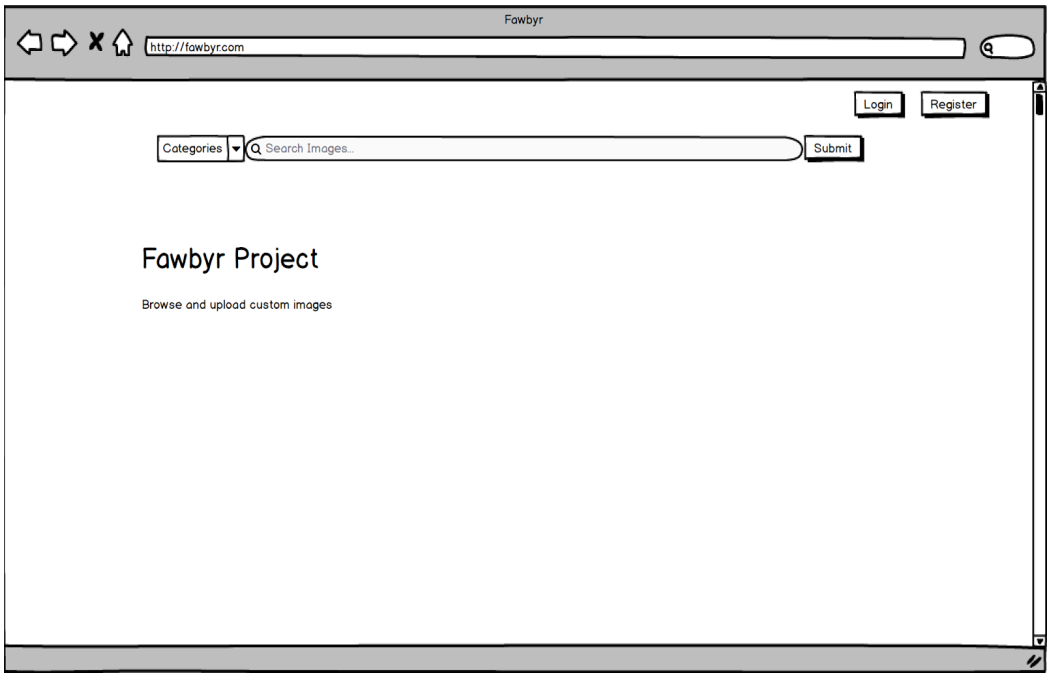
**3. Download history for users**

3.2 Users shall be able to view their past downloads

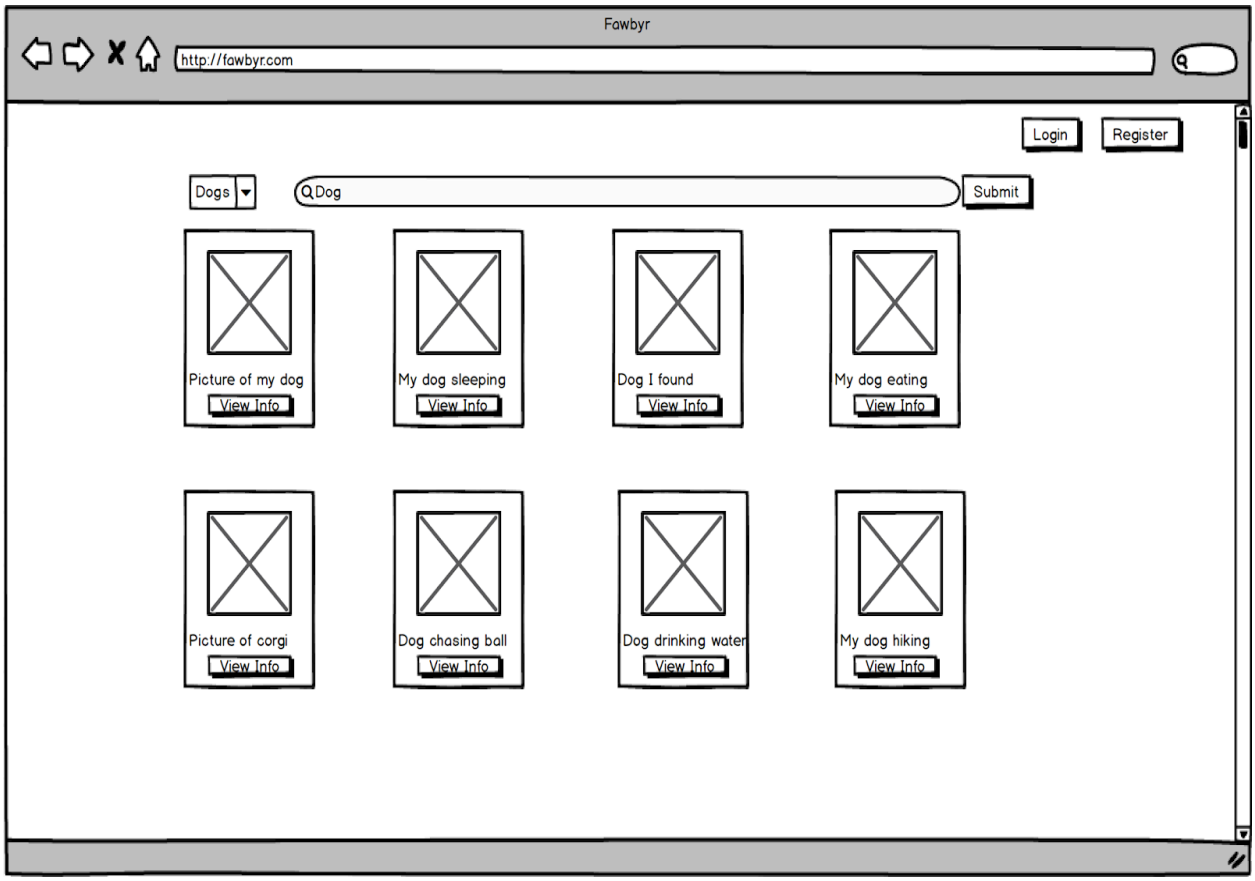
# 3. UI MOCKUPS AND STORYBOARDS

## Downloading Images

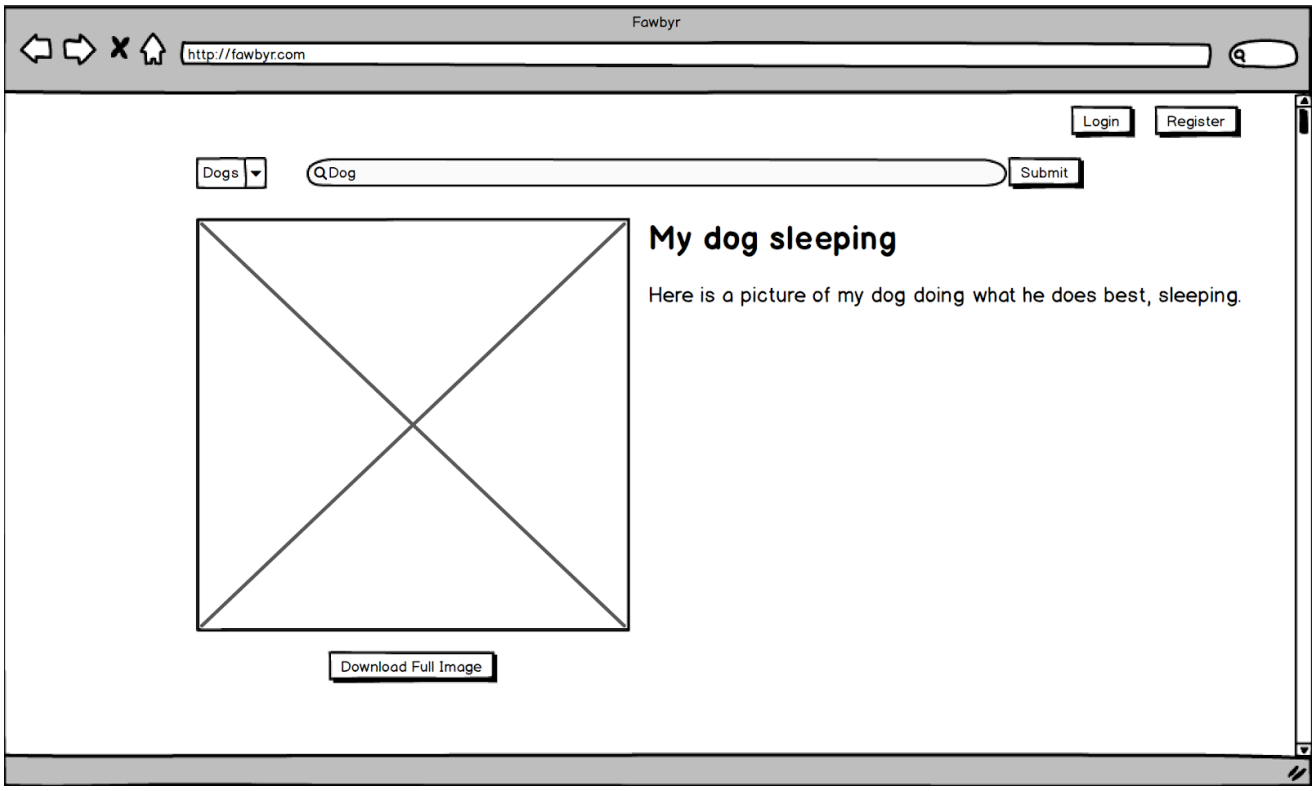
Landing page with search bar supplemented with categories dropdown



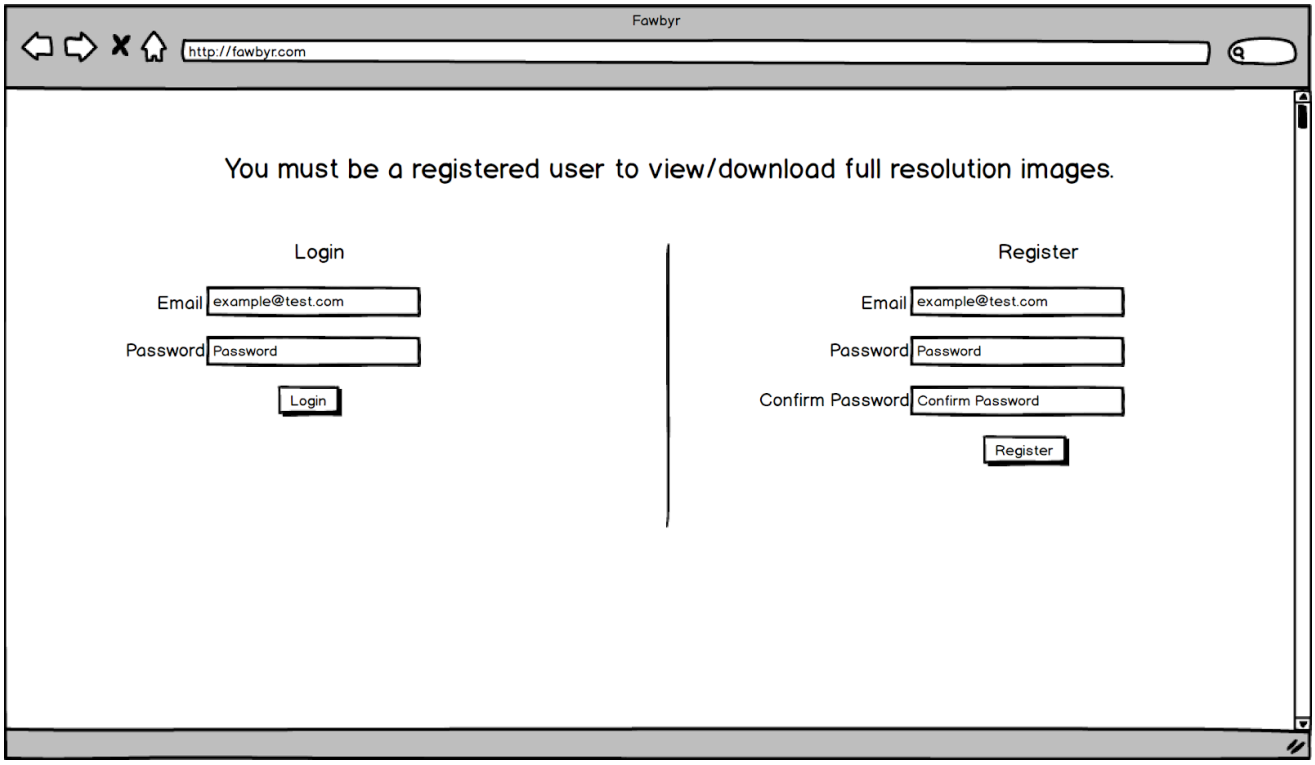
Search results page with card-style postings of images including a button to view more information on that image post.



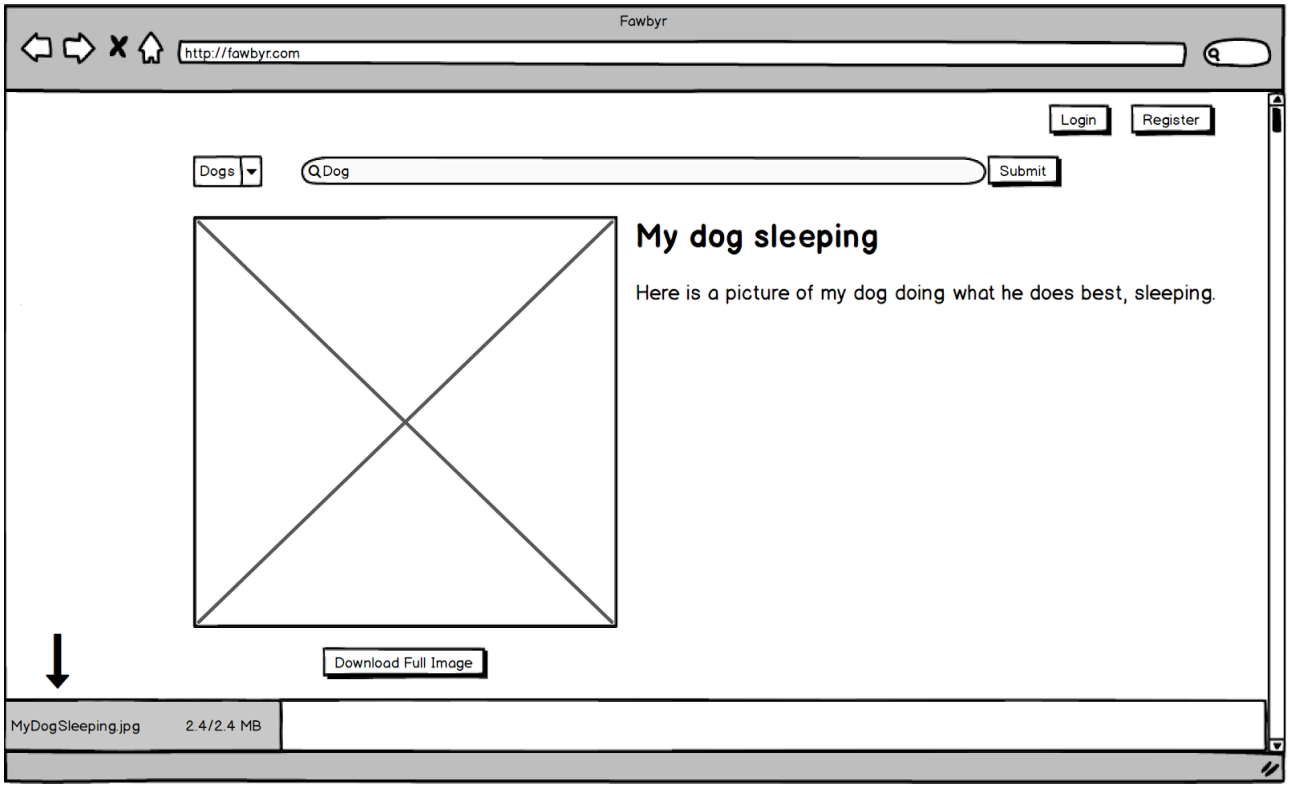
More information/download page



Unauthorized user clicking download button

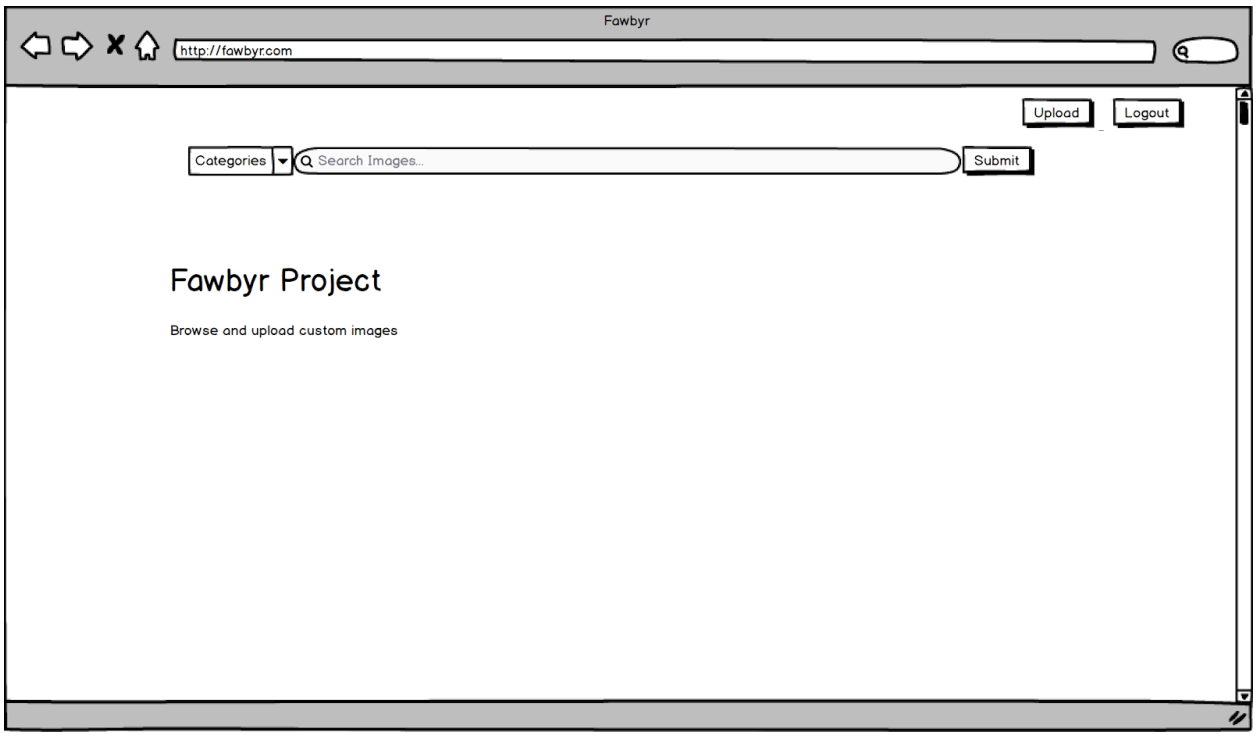


After download click

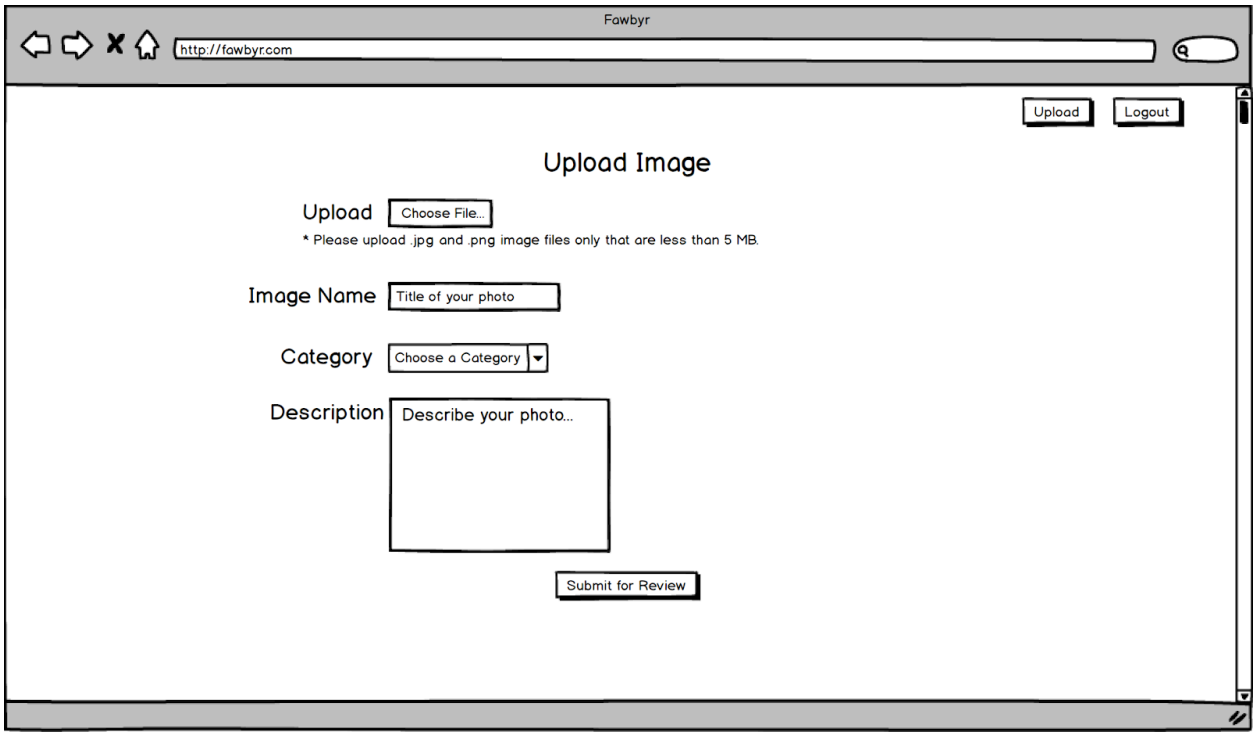


## Uploading Images

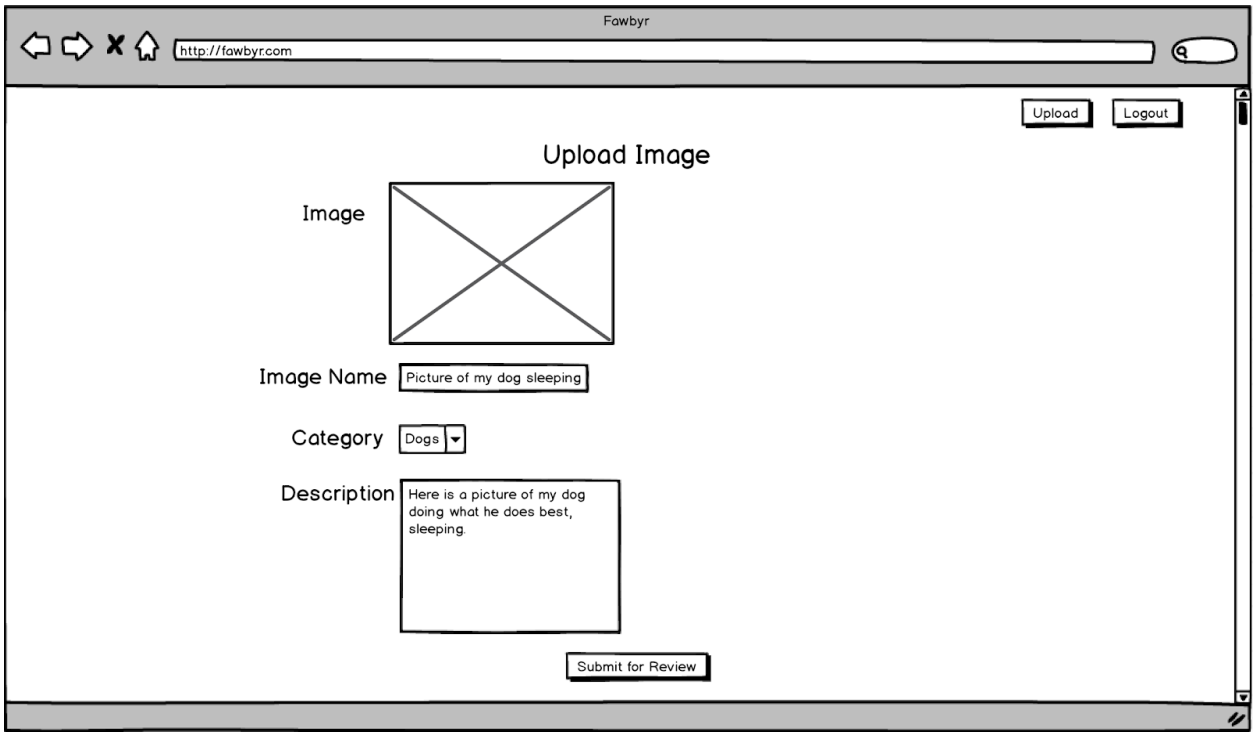
Landing page for authorized user with upload/logout buttons on footer



# Uploading form page

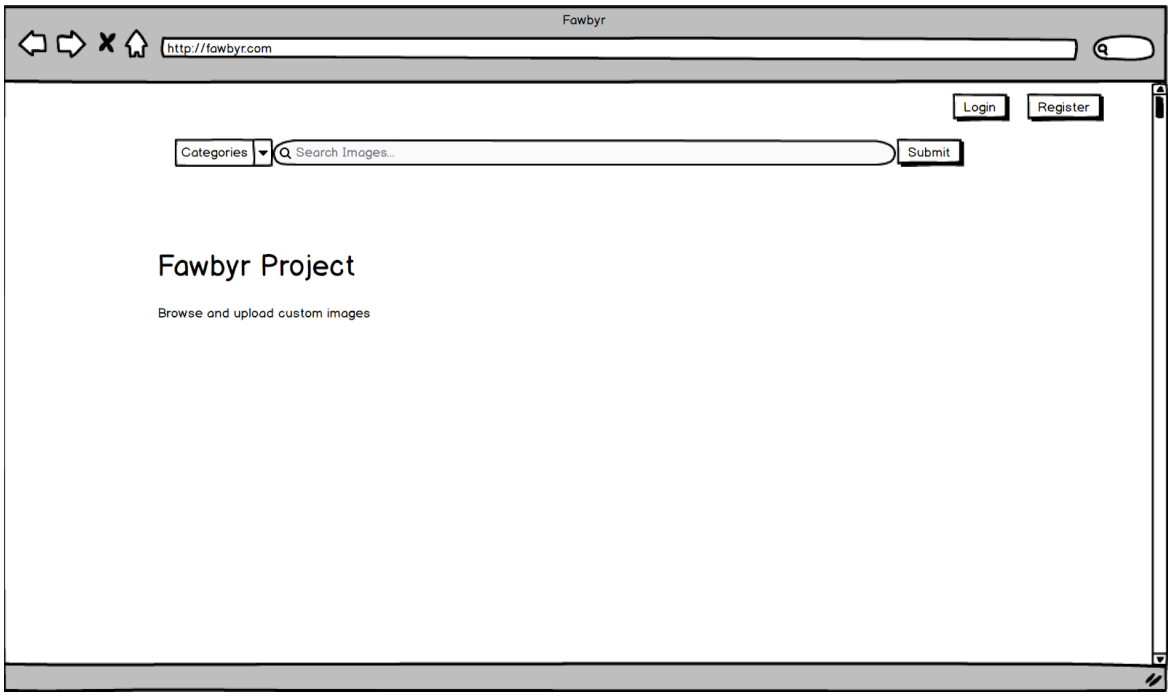


# Uploading form filled out

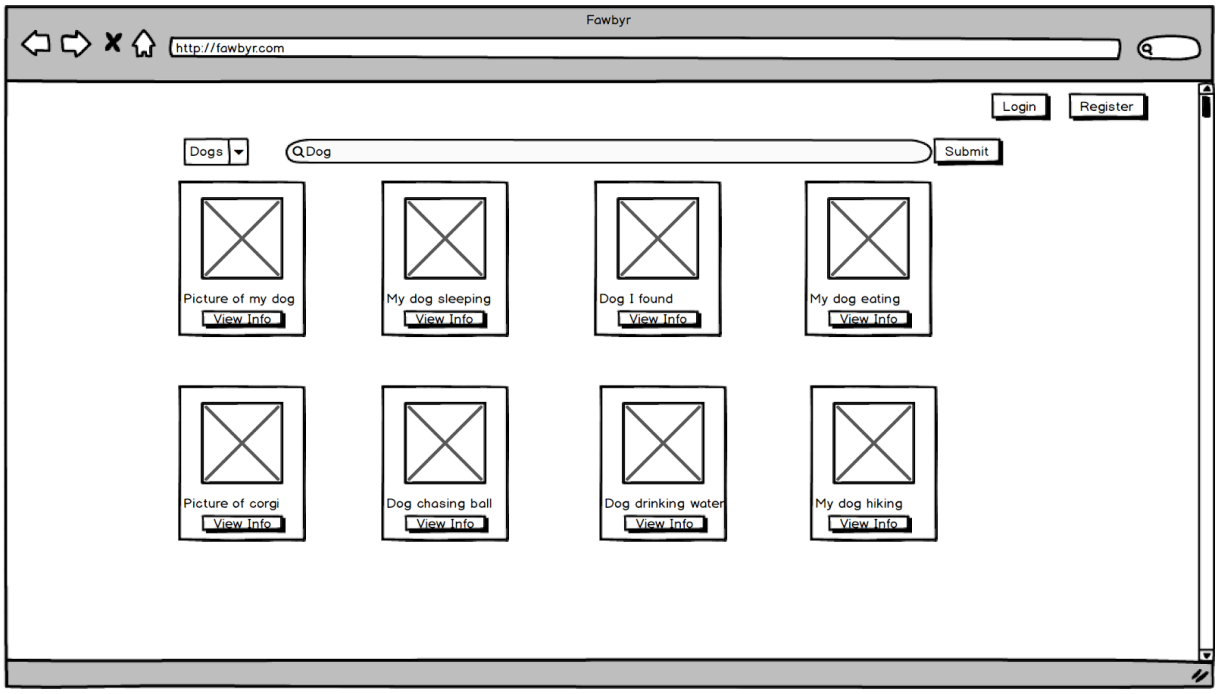


## Viewing/searching images

Landing page with our consistently located search bar area

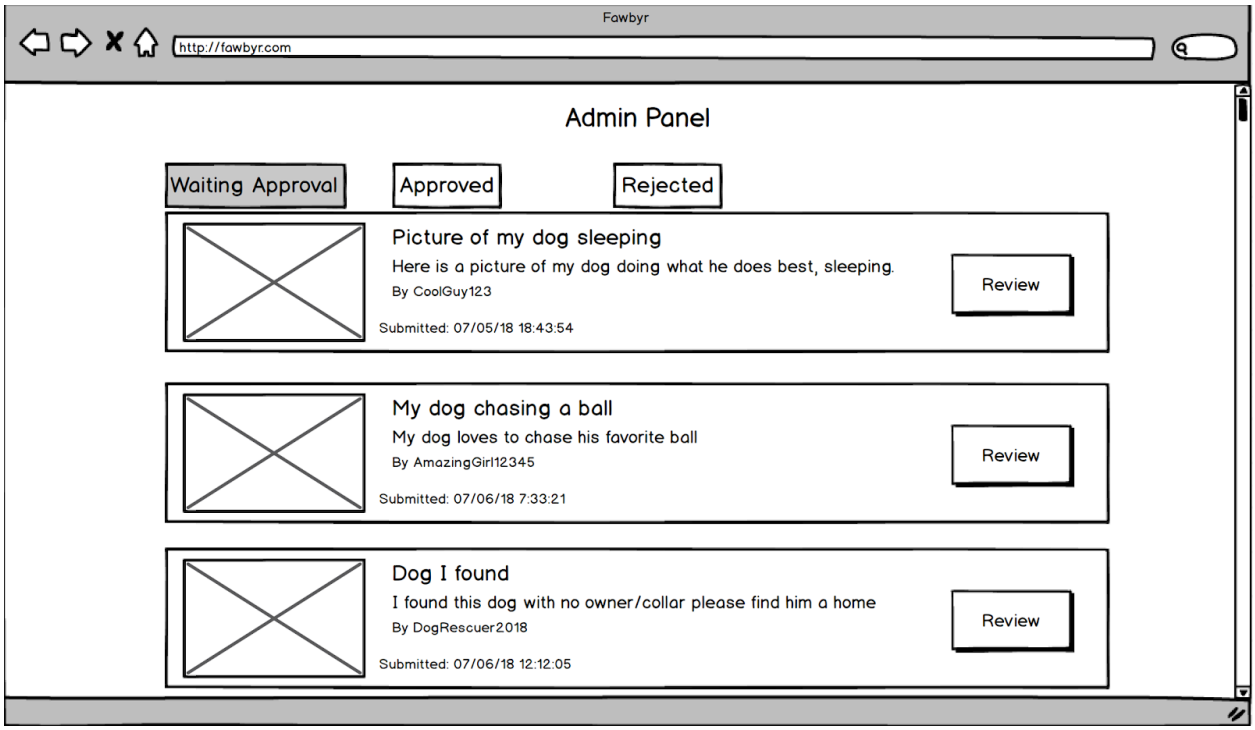


Search results page of dogs for viewing, registration not needed

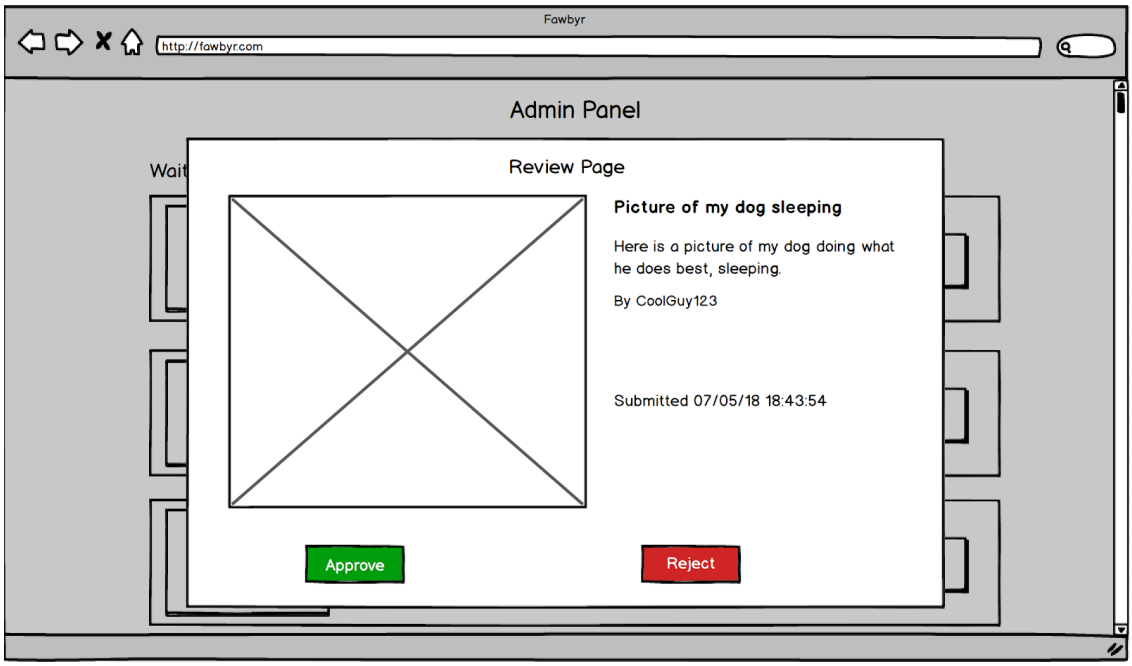


## Admin user

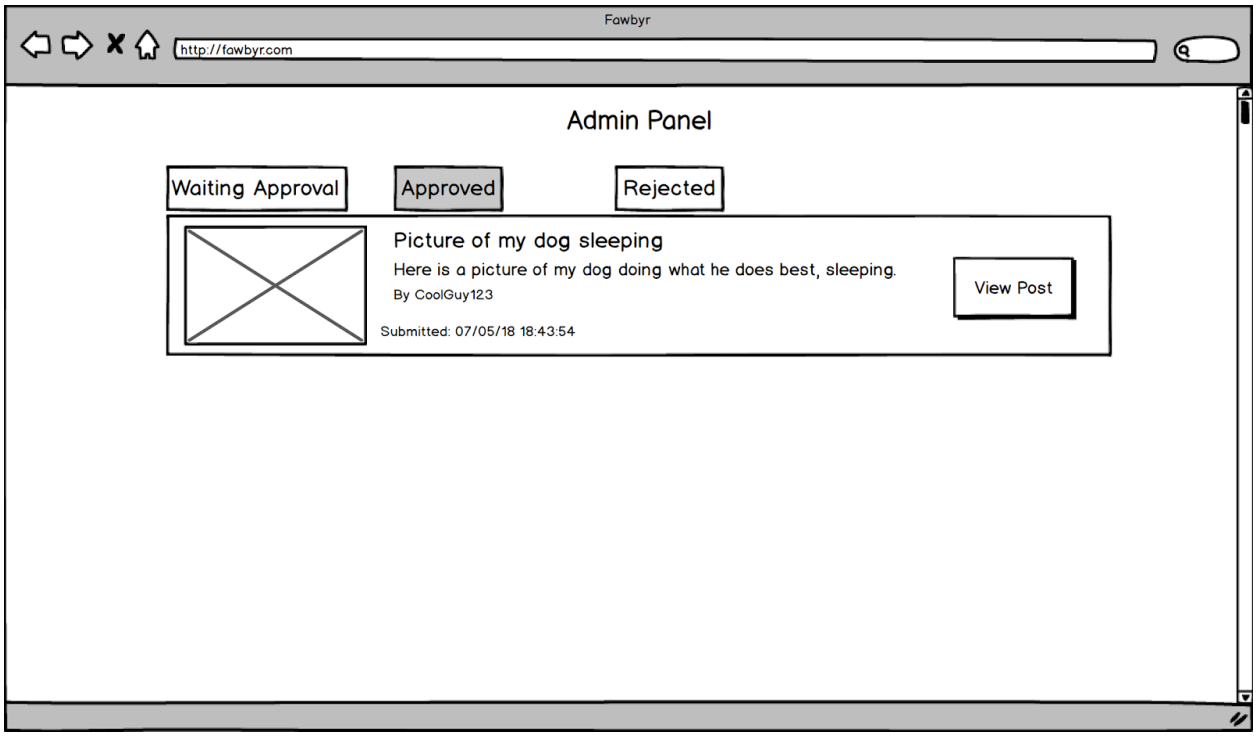
Admin panel for waiting approval, approved, and rejected posts



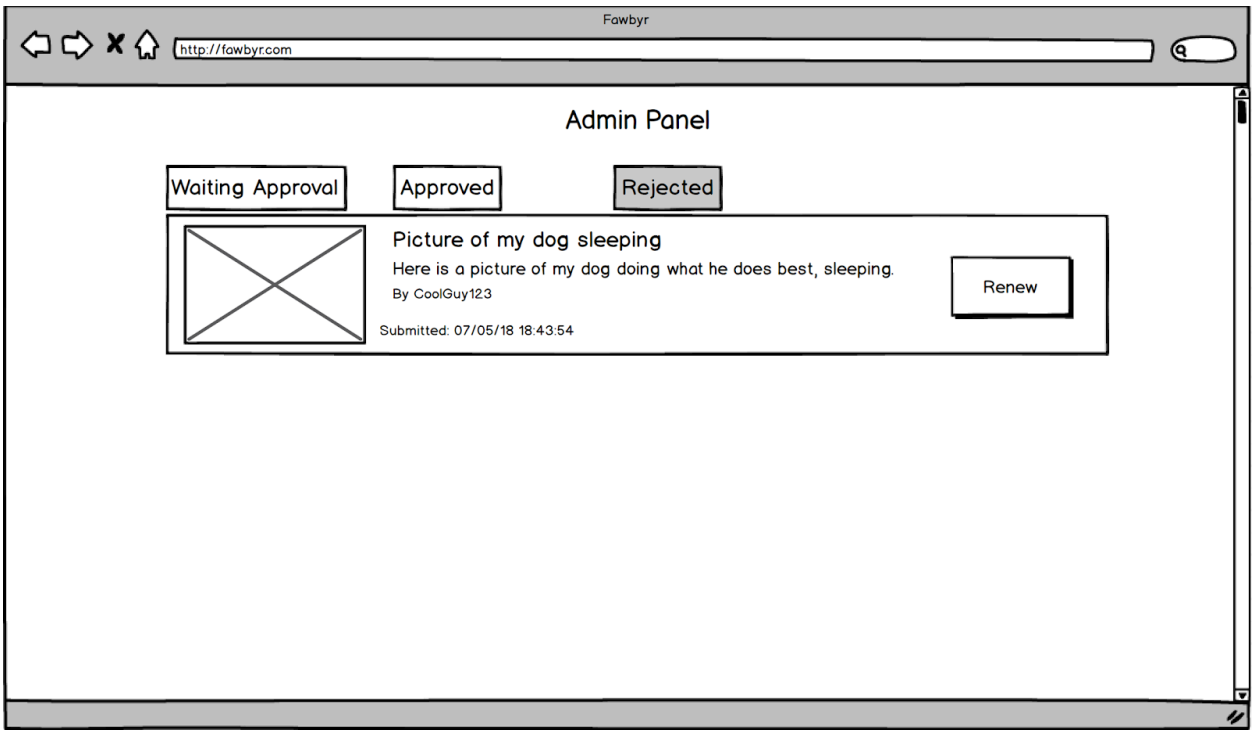
Review modal to display approve/rejection actions



Approved tab to view approved posts



Rejected tab to view rejected posts and renew posts in order to bring them back for reviewing



# 4. HIGH LEVEL ARCHITECTURE, DATABASE ORGANIZATION

**DB Organization:**

Note* All tables shall have an INTEGER id

1. **User**
    - user_name: The *user_name* shall be used as a display name on the website as well as the name displayed for author of images
    - e-mail: The *e-mail* of a user shall be used as their login name
    - password: basic method of authentication, *passwords* shall be stored as a Hash string in the database
    - type: There shall be three main *types* of users
        - Guest
        - Registered
        - Admin
2. **Image**

Images shall be upload by Users and shall be referenced by their user_id

Images must be categorized by their User. Referenced by category_id

- image_name: the name of the image given by the user. The *image_name* shall be used as one of the search keys.

- thumbnail: An uploaded image shall be converted into a *thumbnail* and stored in the database as a BLOB. *Thumbnails* shall be the images first seen in a search.
- med_res: An uploaded image shall be converted into a *med_res* image and stored in the database as a BLOB. *Med_res* images shall be the images seen by non-registered guest users.
- High_res: The originally uploaded image is the *high_res* image and shall be the one that can be downloaded by registered users. Stored as a BLOB in the database.
- description: Users shall be prompted to give a *description* for their image. Descriptions shall be used as one of the search keys
- approved: *approved* shall be stored as a boolean value in the database and is false by default. *Approved* shall only become true after review from an admin where only then shall that image be viewable on the website.

3. **Category**
    - category_name: There shall be categories predefined that users must choose to categorize their images. Categories shall be used as one of the search keys.
4. **Comment**

- comment: Other registered users have the option to put *comments* on images.

5. **Rating**
    - rating: Other registered users have the option to give *ratings* to images. *Ratings* will be stored as integers and shall be incremented or decremented based on what other registered users rate.

**Media storage:** Our database shall only accept images with the following MIME types:

- image/jpeg
- image/png

There shall be a 5 MB size limit on images. Any oversize image or incorrect types of files shall be rejected. The image shall be converted to bytecode and saved on in the database as a BLOB.

**Search/filter architecture and implementation**

The DB terms that shall be used for searching shall be the following:

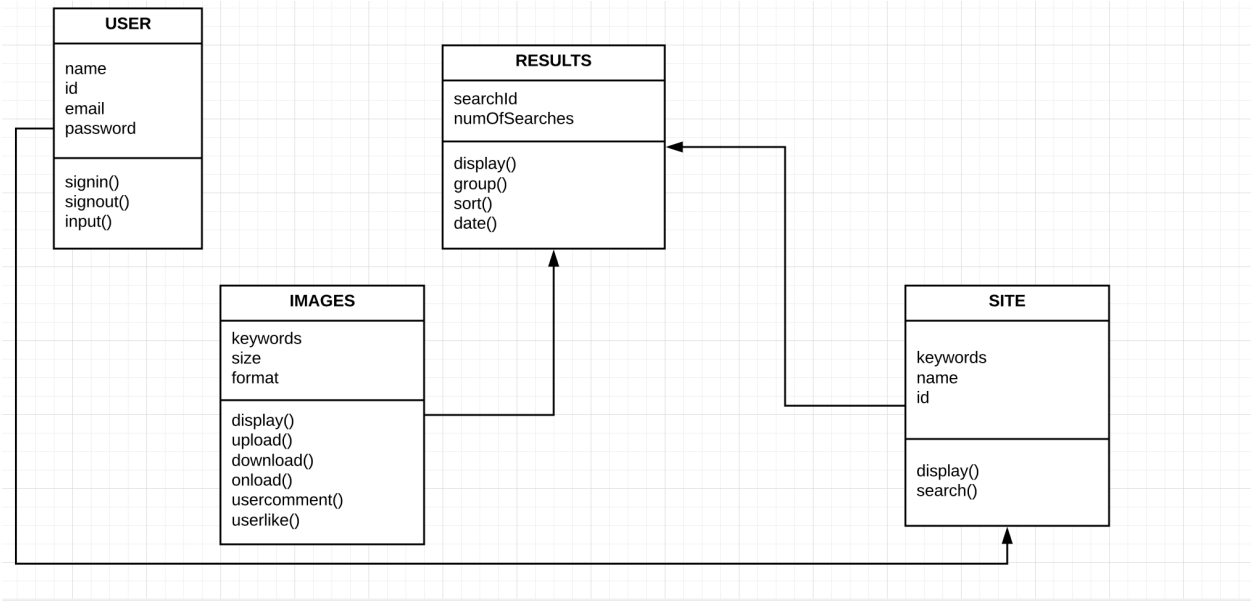
- image_name
- description
- category_name

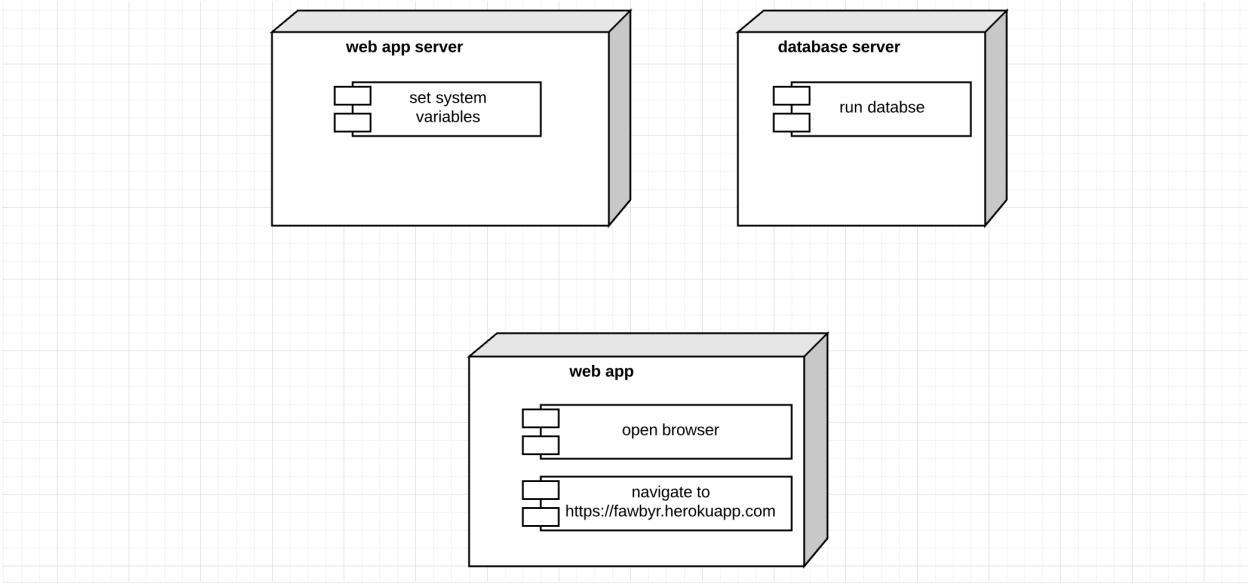
Some sample SQL for search:

SELECT * FROM image WHERE image_name iLIKE ‘%{user input}%’
SELECT * FROM image WHERE category_id = {user input}

A simple algorithm shall be implemented where if no search results are found, then take every word separated by spaces in the input and attempt to use iLIKE search on every single word.
If that fails, then simply display random images for the user to browse.

# 5. HIGH LEVEL UML DIAGRAMS



# 6. ACTUAL KEY RISKS

| Risk Type | Risk | Solution |
|---|---|---|
| **Technical Risk** | *Learning and using Git* – As we have team members at various levels of competency with regards to Git, we need to ensure we take precautions to avoid any loss of productivity due to any mistakes with the version control system. | We've made modifications to our repo to enforce the following rules: a) code can only be added to master by opening a pull request, b) any pull request must have the approval of at least one other team member prior to merging, c) any pull request that touches code with an assigned owner must have approval of the owner prior to merging, and d) the master branch is protected against any forced updates or delete actions. |
| **Teamwork Risk** | *Team desynchronization and loss of cohesion* - As the team is working remotely there is a risk each person may not know exactly what other team members are working on, causing task overlap and wasted cycles. | By using Slack for team communication, we can ensure all team members are kept informed of ongoing issues relating to the project and have a common point of contact. By using Trello for task management, we can ensure that anyone can quickly see which tasks other team |

| | | |
|---|---|---|
| | | members are currently working on. |
| **Skills Risk** | *Propagating engineering knowledge to all team members* – As each person is creating unique pieces of the app it can be difficult to generate tribal knowledge about the application's entire code base. | Require all code submissions to made by pull request and approved by at least one team member to encourage cross pollination of knowledge during the review process. Pull requests must also contain a reasonable amount of comments to ensure other team members can review the code at a later date to see how a given feature was implemented. |
| **Scheduling Risk** | *Assessing workload* - As we don't have an abundance of time we need to be realistic about the number of features we can complete before the end of the semester. | We currently have features broken down into 3 main priority categories ranging from essential (p1), nice to have (p2), and stretch goals (p3). We will focus solely on the p1 category before allowing any scope creep into categories p2 and p3 to occur. To aide in task prioritization will add markers to Trello cards to act as a visual indication to the importance of a given task. |

| | | |
|---|---|---|
| | | |