

# SW Engineering CSC648/848 Summer 2018

## **InfinitelImage**

### **Team 6:**

Zac Henney

Paul Ancajima

Andre Leslie

Teodora Caneva

Abdullah Amir

Joe Phabmixay

Rohit Nair

Team 6 is local

Team Lead: Zac Henney - [zhenney@mail.sfsu.edu](mailto:zhenney@mail.sfsu.edu)

Milestone 2

July 8, 2018

First Submission

## 1. Data Definitions V2

**Registered Users:** shall be able to view and download photos that are uploaded by photographers.

*Username:* user's identifier.

*Password:* user's password

*Friends:* group of other registered users that a particular user will have an ease of access to their content.

*Cart:* a selection of photos that the registered user will be able to add/remove photos to before downloading.

**Non-Registered Users:** shall only be able to view photos, must register in order to download any photos.

**Administrative Users:** shall be able to approve all photos before they go live on the site, as well as has access to view all content and is able to remove content as they see fit.

*Username:* user's identifier.

*Password:* user's password

*Pending:* Photos that haven't gone live on the site waiting for approval.

**Photographer:** user that is able upload photos to website.

*Album:* Collection of photos organized into a single group by Photographer

*Size:* amount of photos in album

*Date:* date of album upload

**Images:** accessible to be viewed by all users, and downloadable by registered users.

*Thumbnail:* smaller version of image.

*Date:* date of the photos upload.

*Location:* where the photo was taken

*Tags:* keywords added to the image in order for search

*Photographer:* name of photographer who took photo for searching purposes

*Description:* text description of photo.

*Licensing:* license and copyright information for photo.

**User Count:** amount of users registered to the site.

## **2. Functional Requirements V2**

**Home Page** - The homepage will be filled with a search engine and most popular photos for the non registered users. And in term of registered users, the home page will be filled with recent friends' uploaded photos, notifications, and also a search engine.

**Upload** - The ability to upload photos by posting it to a specific category so visitors can search for it and see it easily. That will include the ability to add tags associated with the post, and a title.

**Download** - The ability to download photos from the website. The user has to be registered and logged in in this case.

**Register/Login** - any user will be able to search and browse through the photos. However, in order to either upload or download photos, the user has to register by signing up and then log in to the website.

**Category** - Having categories will make the posts organized and easier to access through the web engine. The registered users will be able to categorize their photos if needed.

**Admin** - An administrator account who approves/disapproves user content(images and tags) that is uploaded. Notifies users of time taken to complete this request. Cannot alter tags once they are posted.

**Search** - The website will have the ability to search via the search engine to find photos and a specific registered user in order to find the photos he uploaded.

**Tags** - When registered users upload a photo they can post tags along with it describing relevant keywords to the photo.

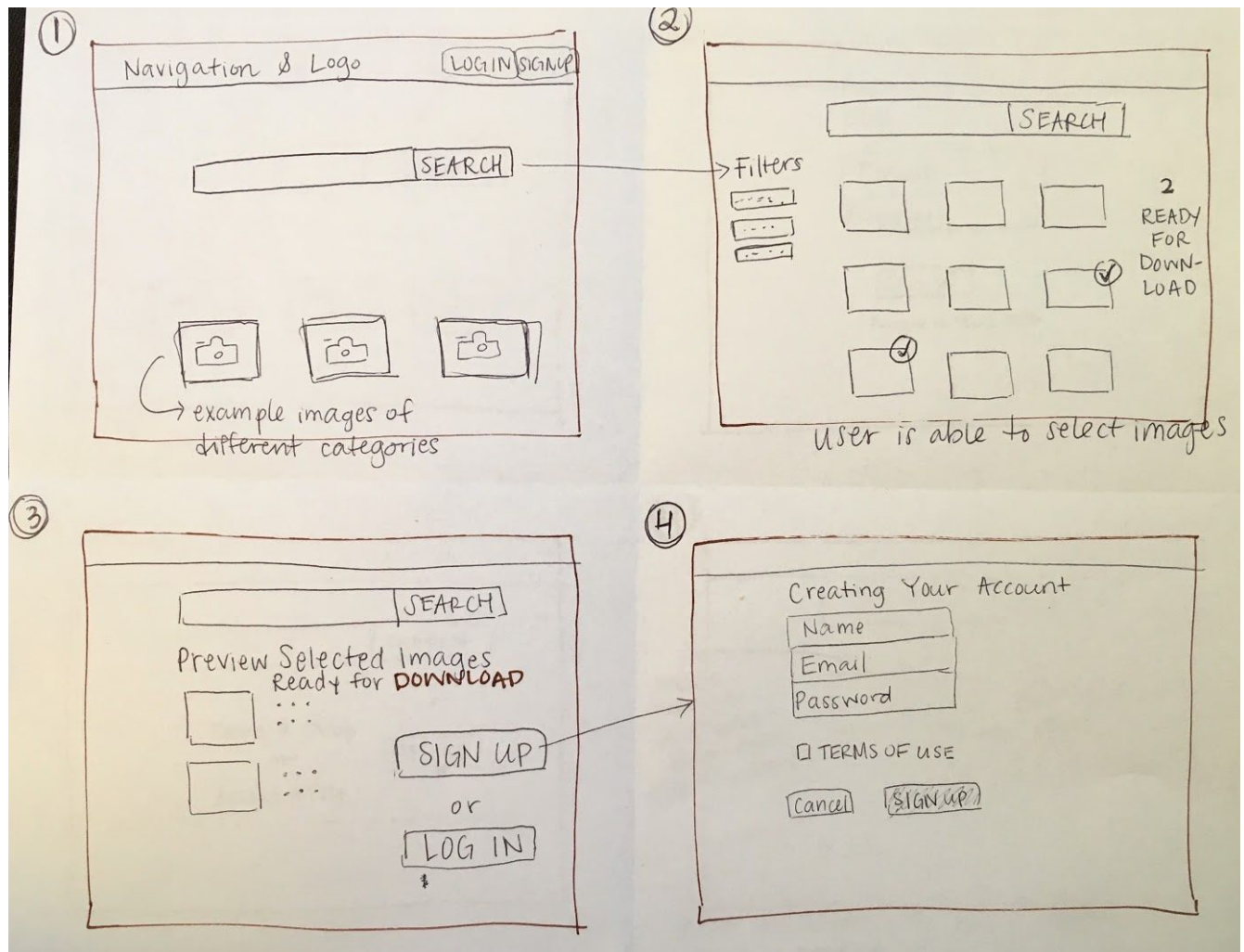
**Watermark** - In order to preview an image full-size and require users to log in, display watermark as an overlay.

**Filters** - To help users search for a more relevant results. Filters can be the upload date, category, user, and size of the photo.

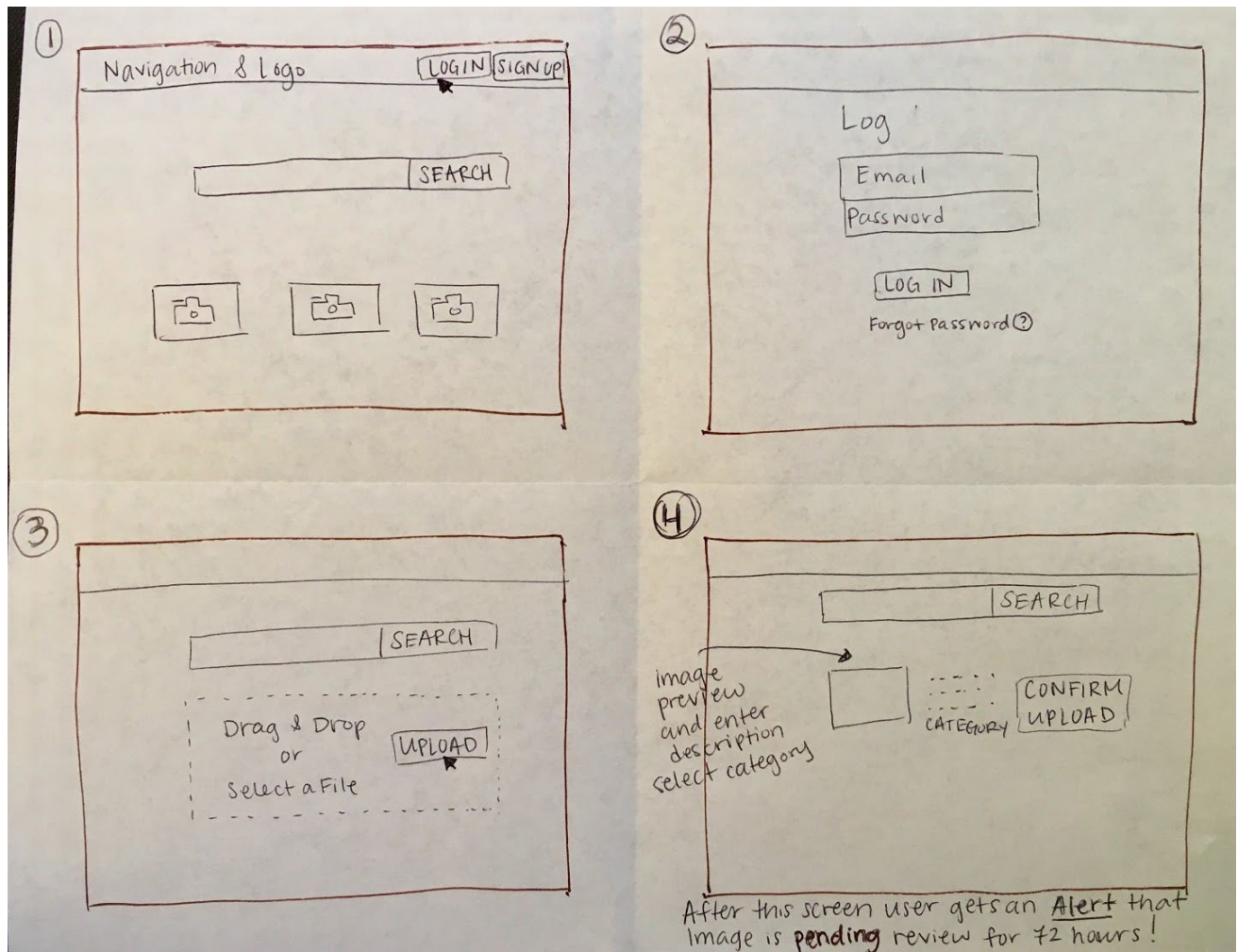
**Friends** - Once registered and logged in, one can add friends via their username so he can get notified whenever a friend uploads a new photo and see his uploaded photo.

### 3. UI Mockups & Storyboards (high level)

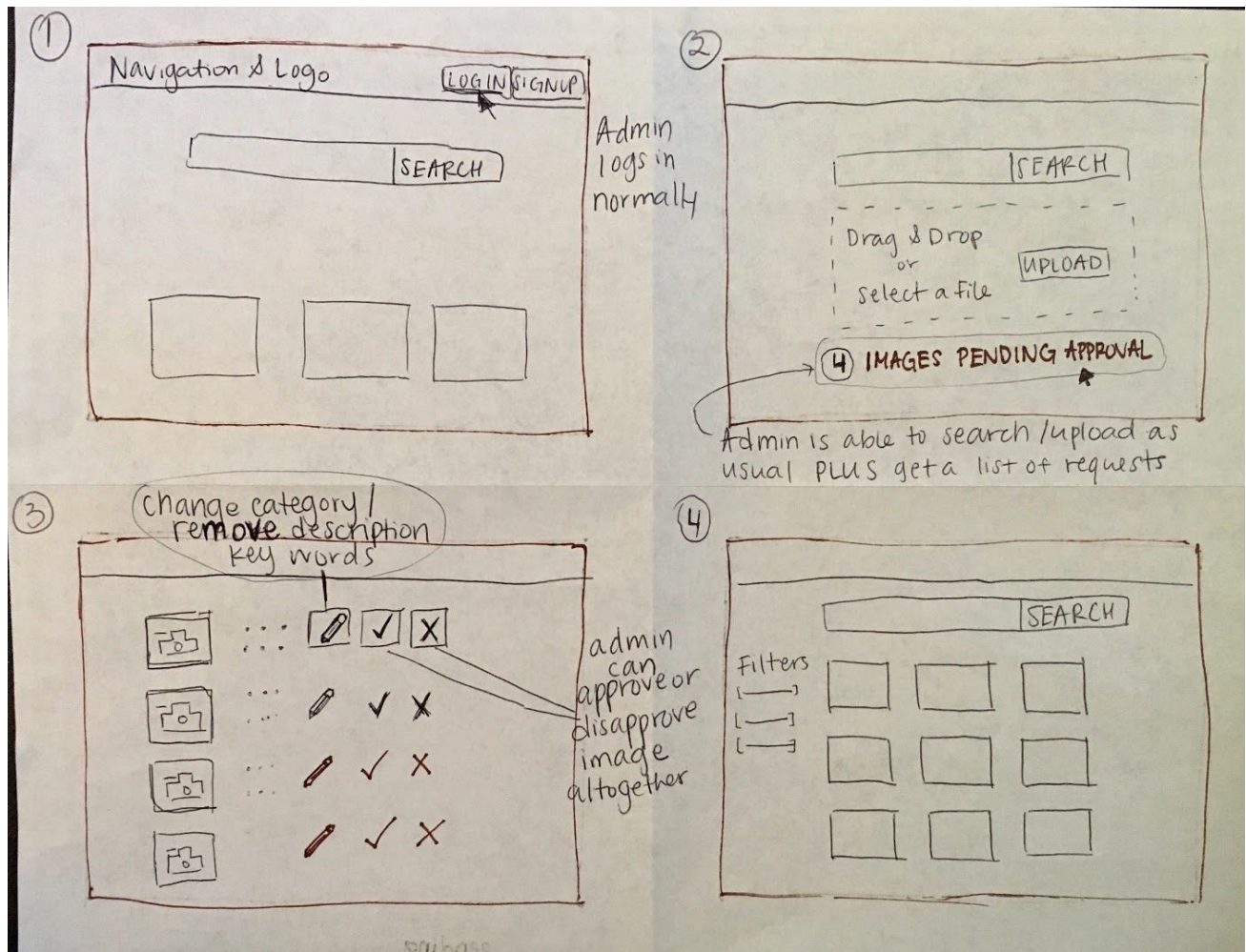
Use case: John is searching for quality images to download, using lazy registration. He is able to preview images he wants and is prompted with Sign up/Log in buttons.



Use case: Fred the photographer's main concern is uploading his images. (Assuming he already has an account) He proceeds to Log In screen and then is able to Search or Upload. Once he is ready to Upload he is able to write in a description and select a category. After, he is alerted that images are pending admin review.



Use case: Admin is able to log in. He has an additional panel that shows him user requests for photos to be uploaded to the site. He can remove certain description words if they are not appropriate or change the category the image should be sorted under. He can approve or disapprove (delete) the photo.

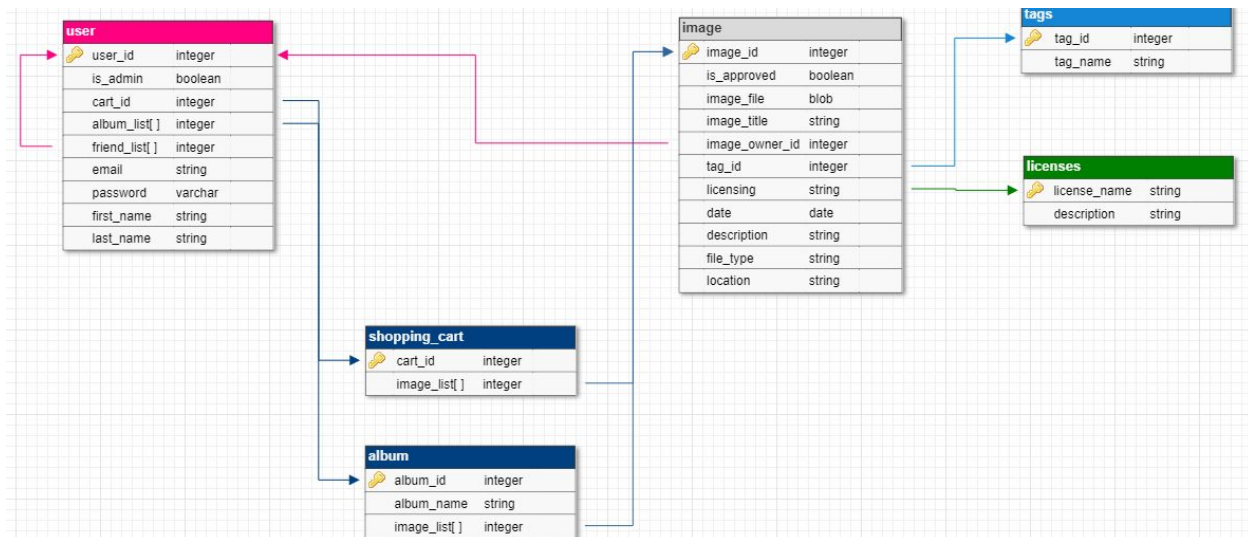




## 4. High Level Architecture, Database Organization

### Database Organization

The database is organized in MySQL based on the following Schema. Tags and Licenses are pre-defined inside the database, which are selected from when an image is posted.



### Media Storage

**BLOBs(Binary Large Objects):** Image media files will be stored as raw files in the database.

**Image Format:** Images must be .jpg, .png

**Thumbnails and Enlarged Image:** Thumbnails shall be created in the database during the time the file is first uploaded, and stored with the blob.

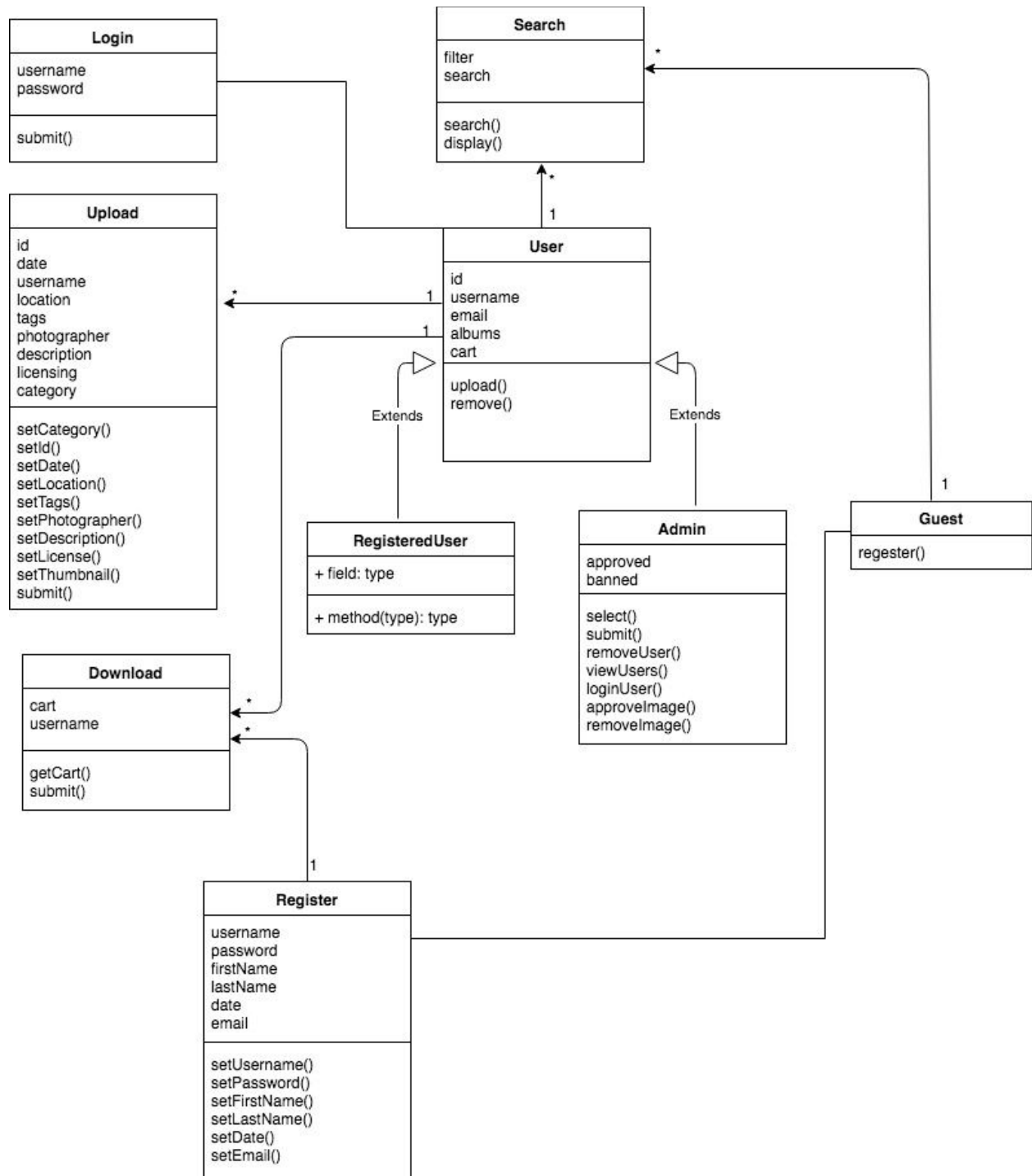
### Search/Filter Architecture and Implementation

**Tags:** All images are assigned a tag, categorizing them among similarly tagged items (e.g. Nature, Cars, Food). The home page shall have a drop-down menu where a user may search among a list of predetermined tags.

**%Like:** MySQL has a built-in search architecture, %LIKE, which we shall implement alongside a main category browser.

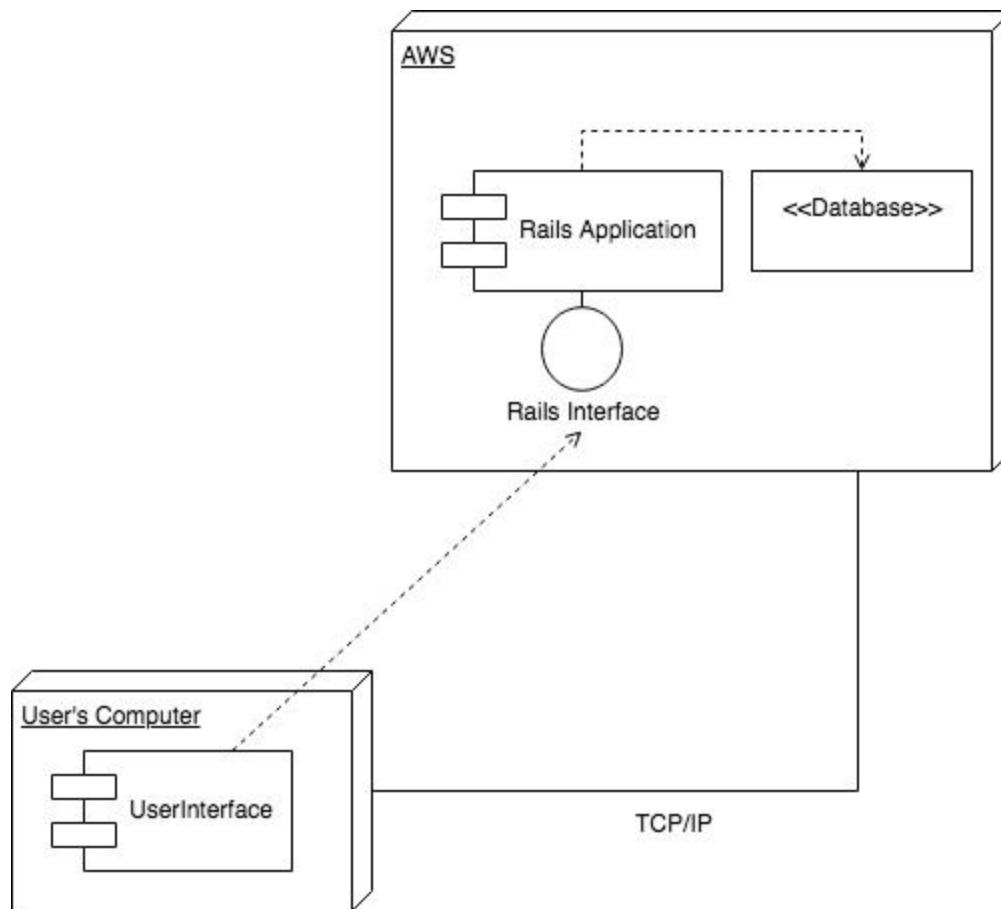
## 5. High Level UML Diagrams

### Class Diagram





## Deployment Diagram



## **6. Key Risks**

### **Skills Risk**

Our team as a whole are not experts in either Ruby, Rails, or MySQL. With this in mind everyone has set out to learn about each technology as quickly as possible. Also, team members have been following and completing various tutorials online to assist in the learning process.