SW Engineering CSC648/848 Spring 2019

Gator Housing

Team 12

Andrew St. Germain (Team Lead)

Sagar Pandya (Backend Lead)

Steven Apicella (Frontend Lead)

David Adams (Document Master)

Sunny Wong (Github Master)

Ganzolboo Ayurzana (Frontend member)

Peter Lin (Frontend member)

Milestone 2

March 22, 2019

1. Data Definitions v2

1. **Users**: Anyone who uses Gator Housing is a User
    a. **Unregistered users**: Users that are only allowed to browse the website, but are not able to send messages or get contact information
    b. **Registered users**: Users that are registered have the permissions of unregistered users but also the following:
        i. **Tenants**: Registered Tenants can message landlords, favorite postings
        ii. **Landlords**: Landlords can put up postings of their unit for other users to see. They can also edit their posts.
    c. **Administrators**: Can review and delete posts before they go live. They are unable to edit posts.
2. **Registration Form**: Required for users to register. Contains name and email, and optionally address, phone.
3. **Postings**: Postings are posted by Landlords and are reviewed by an administrator before it is posted onto the website. A posting consists of the Name/Title, Posting ID, Category, Price, Images(s), Location, Status and Description about the item.
4. **Messages**: The in-house form of communication between tenants and landlords. Once the tenant sees a posting they like, the tenant has to contact the landlord first through the message system. Both the tenants and landlords can check for new messages.
5. **Landlord Dashboard**: A dashboard for the landlords after they log onto the Gator Housing website. From here the landlord can post a new posting, check on the status of existing postings, edit existing postings, remove existing postings and check and reply to their messages.

6. **Tenant Dashboard**: A dashboard for the Tenants. After logging into the website the tenant can view their messages and send new messages to landlords.

7. **Administrator Dashboard**: A dashboard for the administrator. After logging onto the website, the administrator has access to special privileges. These privileges include approving postings to be posted onto the website, removing postings from the website that are deemed inappropriate or ones that don't follow the website guideline and removing suspicious users.
8. **User Record**: The user record is created when an unregistered user becomes a registered user. The user record contains information relevant to that specific user including username, user ID, user type, permissions, password, etc.

2. Functional Requirements v2

Priority 1:

1. **Unregistered Users**
    1. Unregistered users shall be able to browse apartments.
    2. Unregistered users shall be able to search for apartments.
    3. Unregistered users shall be able to view the apartment location.
    4. Unregistered users shall be able to view how far the apartment is away from campus.
    5. Unregistered users shall be able to view the price of the apartment.
    6. Unregistered users shall be able to become a registered user.
    7. Unregistered users shall not be able to contact the landlord.

8. Unregistered users shall be prompted to be logged in when using registered user services.

2. Registered Users

1. Registered users shall be able to browse apartments.
2. Registered users shall be able to search for apartments.
3. Registered users shall be able to view the apartment location.
4. Registered users shall be able to view how far the apartment is away from campus.
5. Registered users shall be able to view the price of the apartment.
6. Registered users shall be able to post information about apartments.
7. Registered users shall be able to edit their own posts.
8. Registered users shall be able to contact the landlord.
9. Registered users shall have access to their dashboard.
10. Registered users shall be able to see messages received.
11. Registered users shall be able to send messages.

3. Administrators

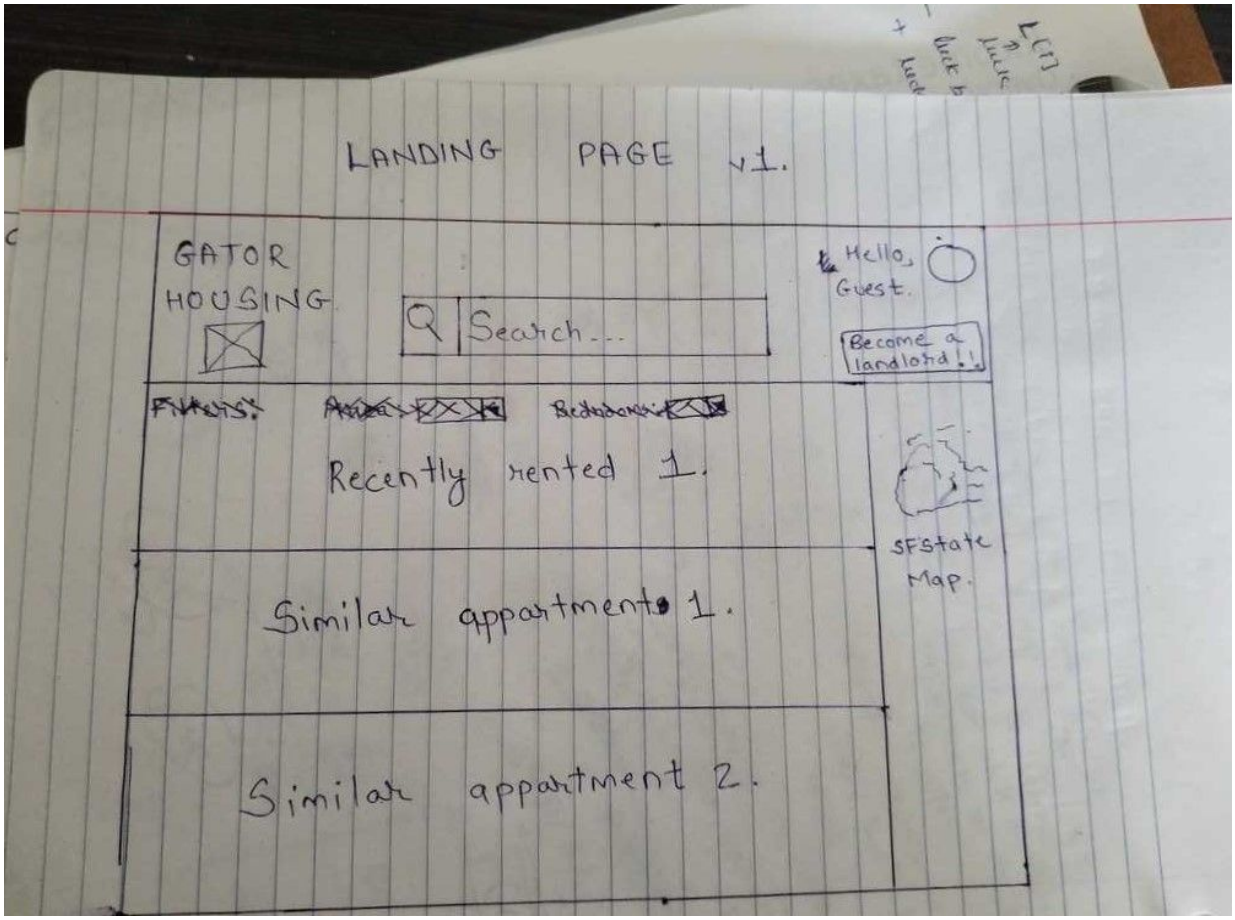
1. Administrators shall review posted apartments before they go live.
2. Administrators shall be able to remove inappropriate posts.
3. Administrators shall be unable to edit posted information.
4. Administrators shall be able view their dashboard with pending posts.
5. Administrators shall be able to browse apartments.
6. Administrators shall be able to search for apartments.

Priority 2:

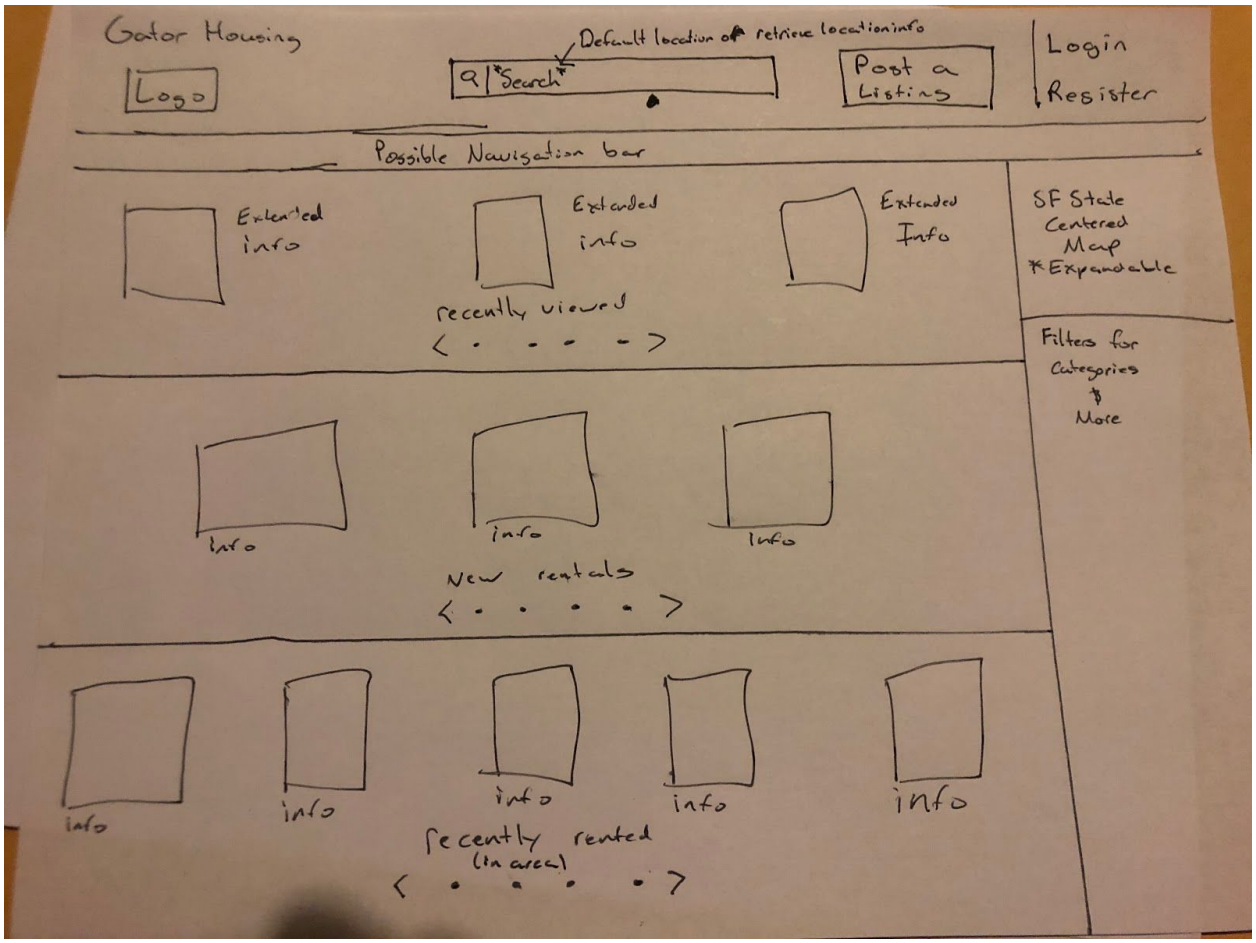
1. **Registered Users**
    1. Registered users shall be able to contact the administrators.
2. **Administrators**
    1. Administrators shall be able to send messages to registered users.
    2. Administrators shall notify user if their post has been removed.

3. UI Mockups

Version 1 Mockup-



Version 2 Mockup-



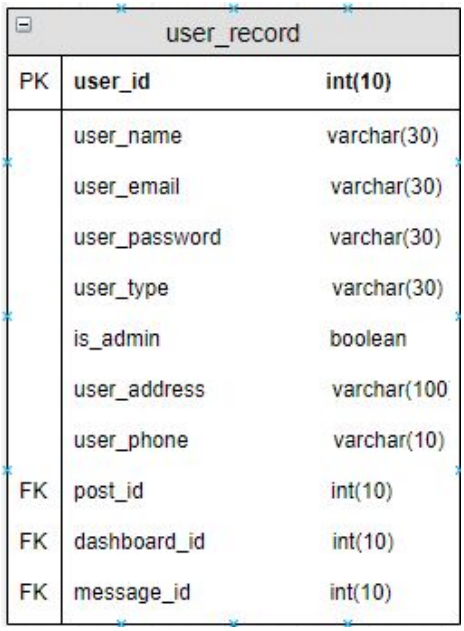
Still in the process of finalizing a design.

4. High Level Architecture

a. Database Organization

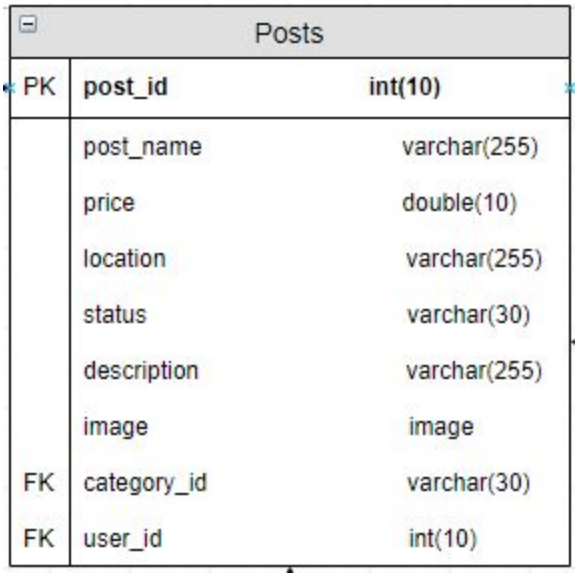
Note: PK stands for primary key. FK stands for foreign key.

1. User Record:



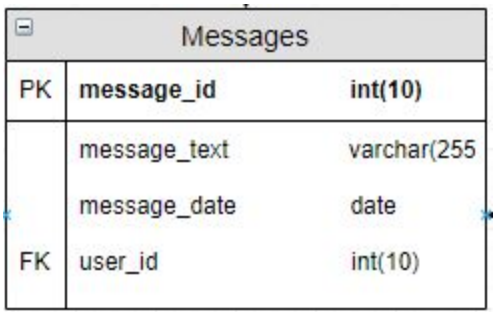
The user record will keep track of all the information that is required or optional when an unregistered user decides to register. We will be able to keep track of their posts and assign a dashboard to them as well.

2. Posts:



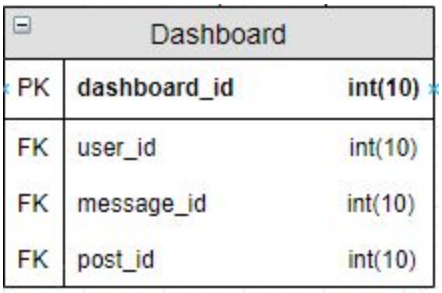
This table requires that all users who may want to post to fill out each of these fields to be able to complete the post. All posts will be linked to an ID to identify who posted the apartment information.

3. Messages



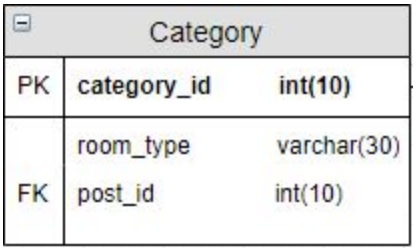
The messages table is used to keep track of the messages sent and received by registered users.

4. Dashboard



Depending on the user ID, a dashboard will be assigned. This will change depending on whether a user is a tenant, landlord, or an administrator.

5. Category



Posts will be categorized as either some type of house, apartment, or room. This will enable registered users to be able to sort places by category as well.

b. Media Storage

We will be using files to store our images for the website.

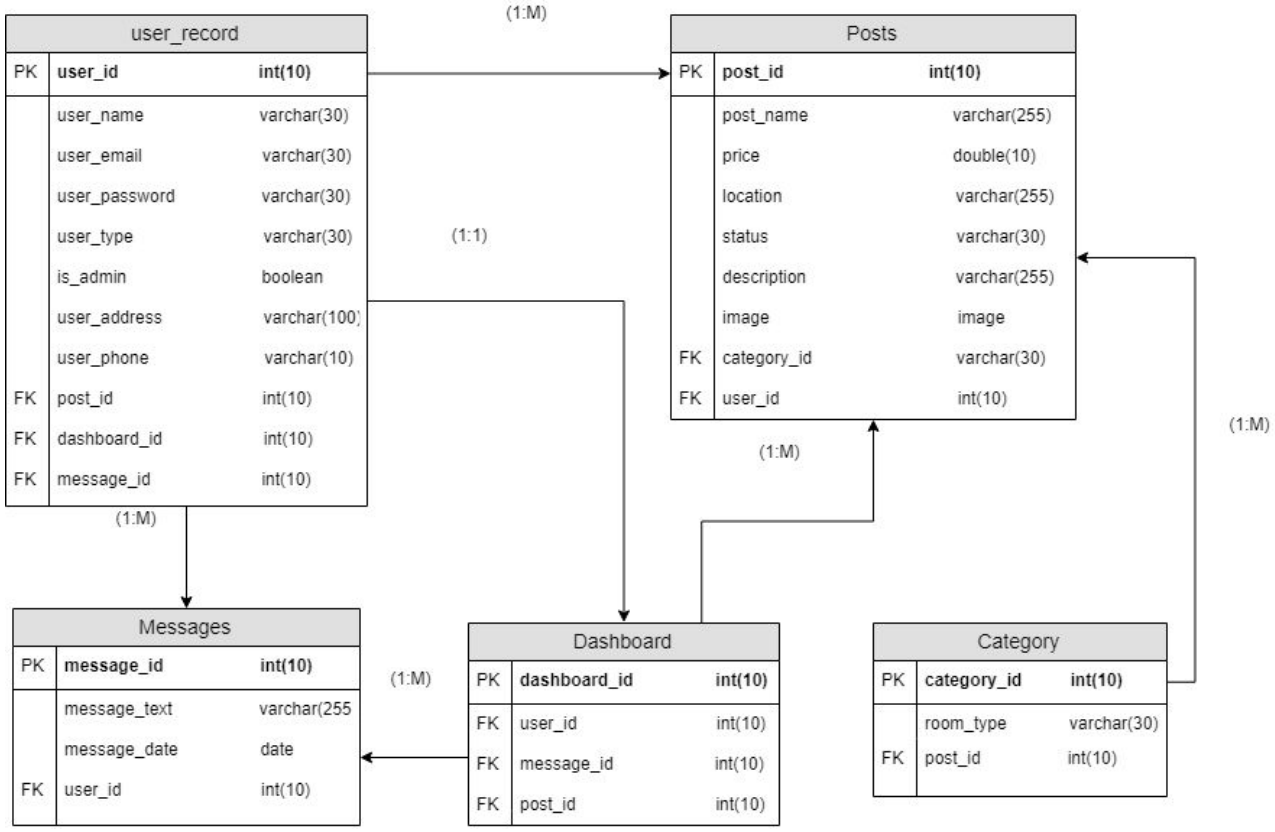
c. Search/Filter Implementation

There will be a small drop down menu near the search bar to filter posts by category such as a house, apartment or room, which will send a query to the Category table to our database and return each type. The end user shall be able to search using keywords to find an apartment, such as Daly City as an example, to find a place by location by using the %LIKE Mysql query.

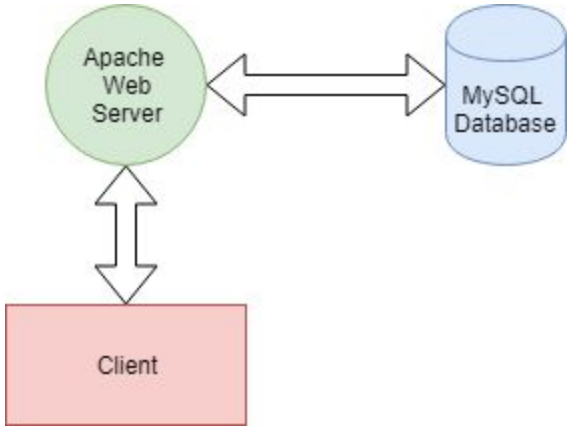
d. Possible API usage

6. Diagrams

UML:



Deployment:



7. Key Risks

a. Skills Risks

Our back end team is inexperienced with working with Node, but will try to work to the best of their abilities while still learning Node independently on the side.

b. Schedule Risks

A few members have a tight schedule to work around. Coordinating within teams may prove more difficult to get things done in a timely manner. Trello will still be useful to solve this along with help by other teammates in case unexpected events occur.

c. Technical Risks

Not everyone in our team is familiar with GitHub usage. However, a member uploaded an easy to read and follow documentation with pertaining to git push/pull/merge and branching. Whenever a member is unsure of how to use Git, they can refer to the documentation for a smoother process. Still, if any merge errors occur it will be handled by the Git master to resolve issues before pushing to prevent further errors.

8. Project Management

We will be using Trello to keep track of what each side is doing. We will assign cards for specific features for front end members and back end members to work on in a timely manner. Task distribution will be assigned as equally as possible and also based on the strengths and weaknesses of front/back end team members. Each end will work on their parts independently from each other, merging pieces together slowly as features are being finished. This will allow our team to steadily build the product until it is complete.