SW-Engineering CSC 648 Spring 2019
Gator Room

Milestone 2

Date of Submission

3/25/2019

Date of Revision

--

Team 103

Nico Graves - Jakhongir Khusanov - Michael McDonald Tran - Michael Nelson

Feona Garcia - David Hernandez - Carlos Velasco

Data Definitions V2

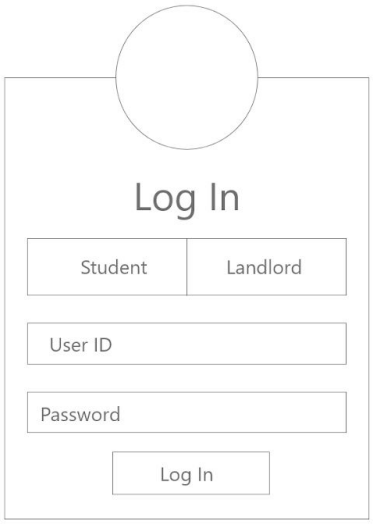
1. **Landlord**
    a. Landlords are identified by their landlord ID (their username in the database),and the landlord can post on the website and in their profile: their email address and other student's emails that are looking for a roommate, their phone number and student's phone numbers, apartment listing addresses, and student's ages must be recorded.
    b. Once a student has filled out the application provided by the landlord then the landlord will be notified about the applicant and see if that student covers the criteria to be approved for the apartment listing.
2. **Student**
    a. Students are identified by their username(which will be their SFSU email). SFSU students can customize their personal information in their profile with their email addresses, phone number, and their age.
    b. Every student can register for a listing and they can specify how long their intended stay will be and what filters they prefer.
    c. They can also "favorite" a listing to save it on a watch list for future registration/perusal.
3. **Listing**
    a. For each listing, the address, type of service, essential details, and price must be recorded. Each listing is managed by a landlord, and the service identifies a listing uniquely from other listings on GatorRoom.
    b. If a listing is deleted, then the site will no longer keep track of it any longer. There will be a status code that will display if the listing is available or off the market.
    c. Essential details for the listing will be number of bedrooms, bathrooms , kitchens, garage, and square footage.
    d. A listing is categorized into services which will be for example shared bedrooms, whole apartments, private bedrooms, short term/long term housing, etc.
    e. There will be additional filters for each listing which can be: dog friendly, smoke free, laundry onsite, furnished.

Functional Requirements V2

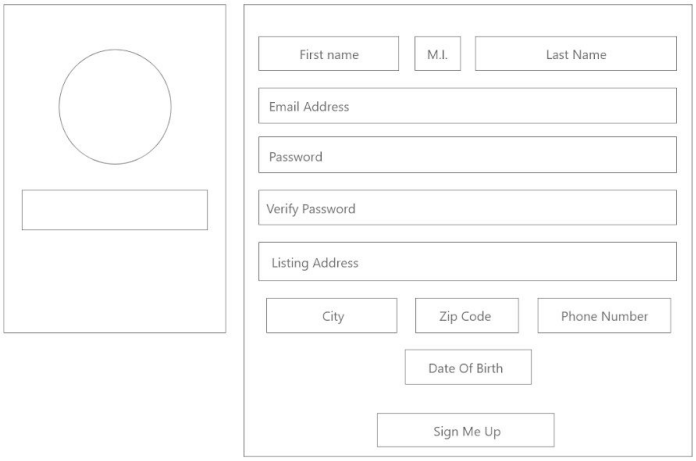
1. Students can browse the student portal and view listed houses.
2. Landlords can upload houses through the landlord portal to the house database.
3. Users can set filters to their profile to affect the houses listed for them.
    a. Smoking Habits
    b. Pets
    c. Budget
    d. Age of roomates
    e. Gender of roomates
    f. Location of houses
    g. Disabilities
    h. Amenities
        i. Gym
        ii. Office
        iii. Study/Public area
        iv. Onsite Laundry
    i. Parking
4. Landlords can set specific rules for different houses they list.
    a. Smoking
    b. Pets
    c. Visitors Allowed
5. Students can view a map of the location of the listed houses.
6. Students can ask their landlords for permission to become master tenants.
7. Landlords can approve master tenants.
8. Master tenants can list the house they live in with vacancies just like landlords.

UI Mockups and Storyboards

Gater Room



Gater Room Landlord



Gater Room

Hi, User

Find the right place for you

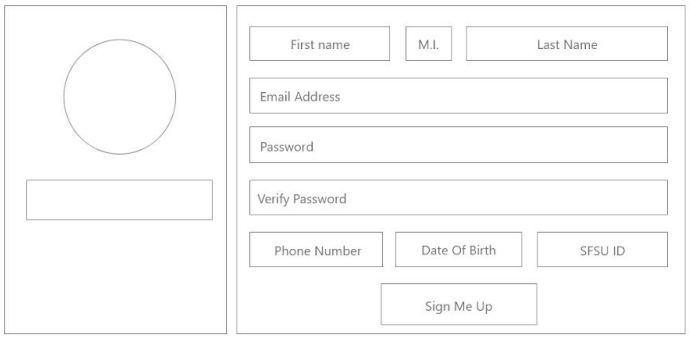
Search for housing options...

Search

Gater Room

Hi, User

Gater Room Student

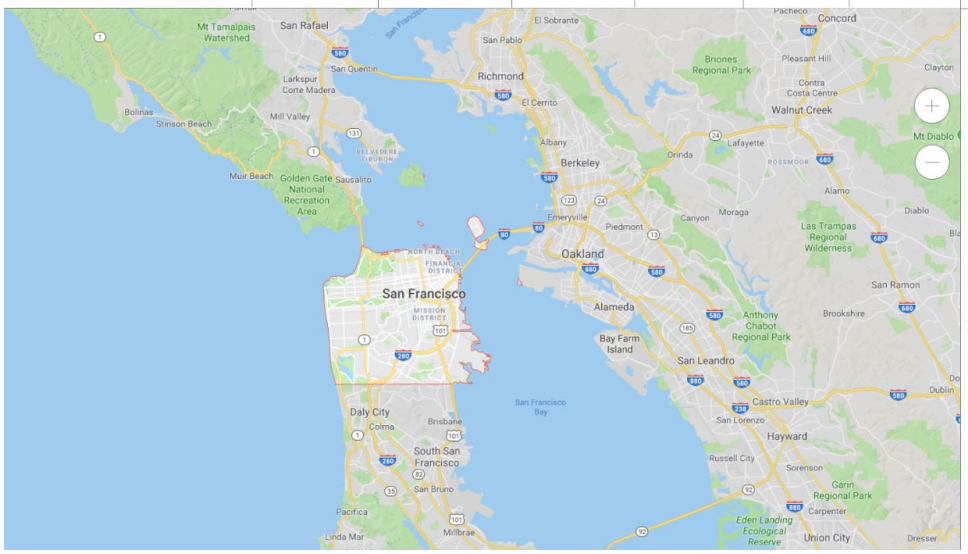


Gater Room

Hi, User

Search....

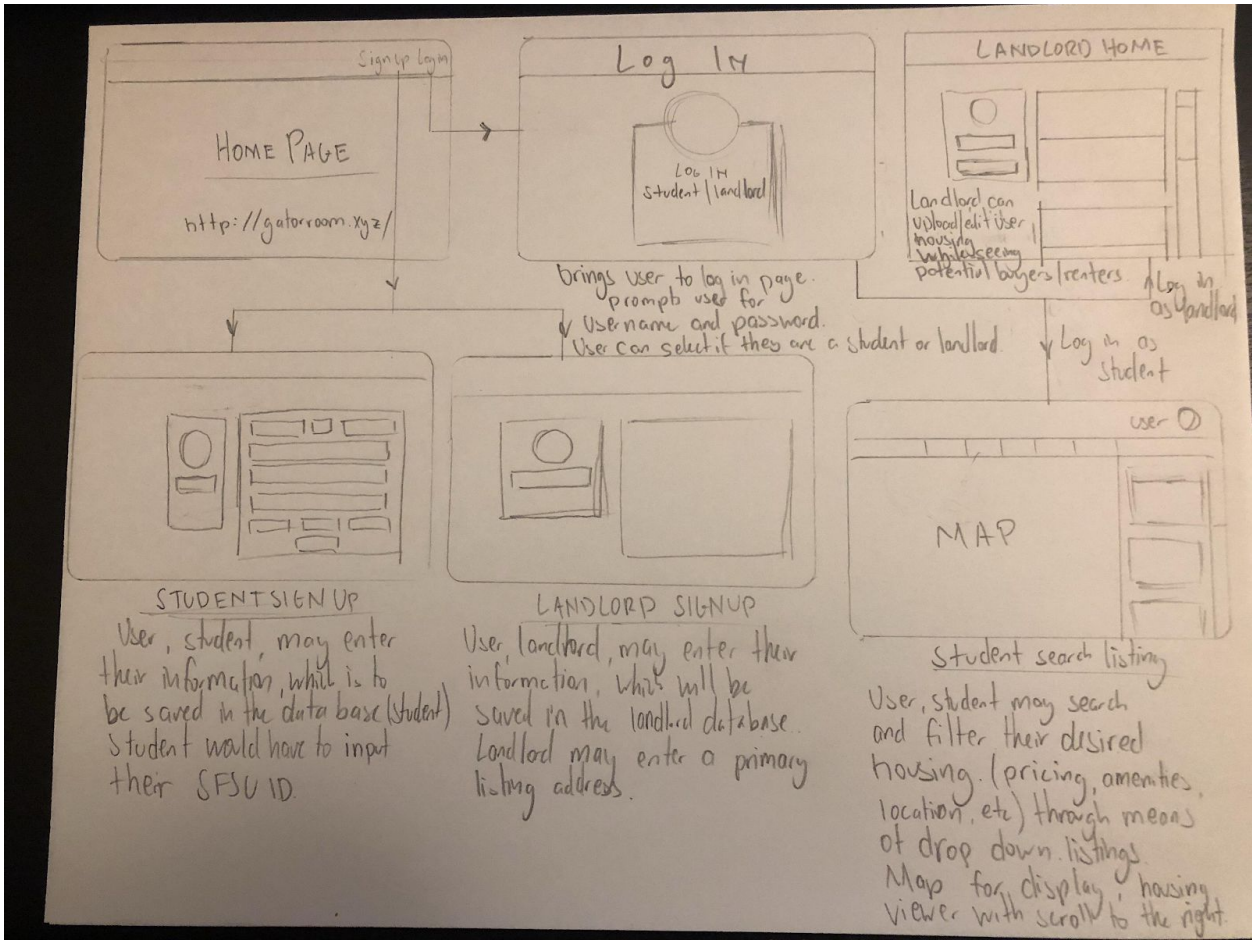
Listing Price v | Rooming v | Amenities v | Location v | Services v | More v



South San Francisco

Daly City

San Francisco



High level Architecture, Database Organization

Listing Table - Holds House entities.

Listing

- Listing ID
- Address
- City
- Postal Code
- Picture
- Smoking Filter
- Pet Filter
- Parking Filter
- Laundry Filter
- Num Bedrooms
- Num Bathrooms
- Num Kitchens
- Price (amount)

Student Table - Holds student entities

Student

- Student ID
- SFSU Email
- First Name

- Last Name
- Phone
- Username
- Password
- Picture

Landlord Table

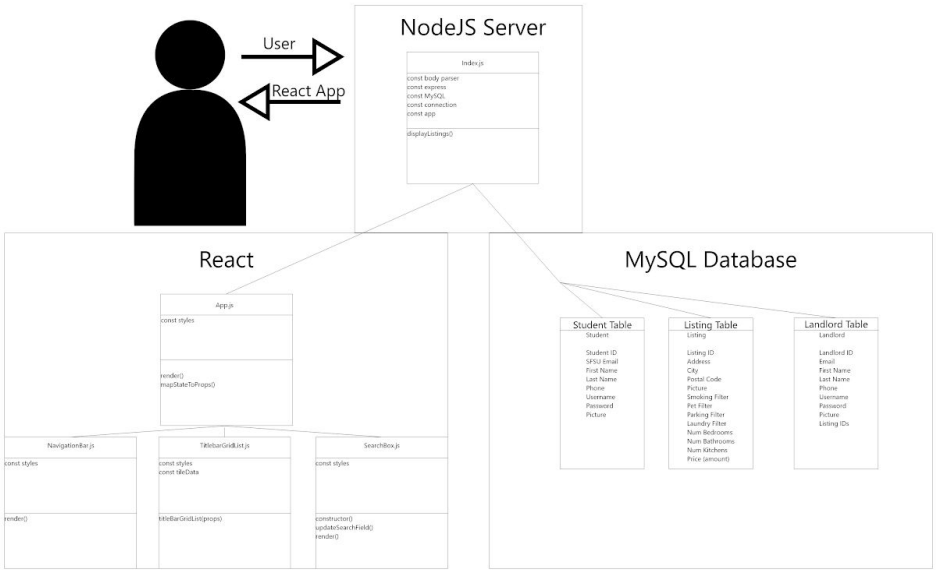
Landlord

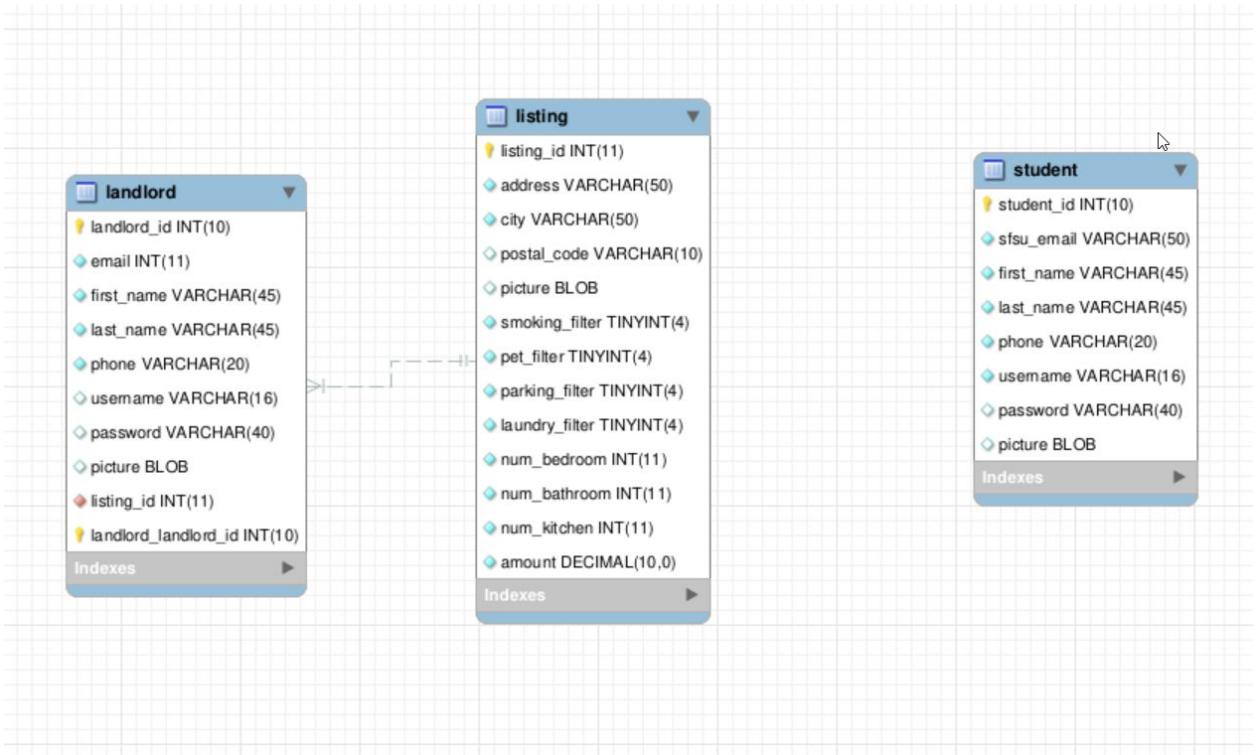
- Landlord ID
- Email
- First Name
- Last Name
- Phone
- Username
- Password
- Picture
- Listing IDs

Media Storage : Data Blobs

Search/Filter Architecture And Implementation:

High Level UML Diagrams





UML - Easier to Read Version Available Here
DATABASE - Easier to Read Version Available Here

Key Risks

Risk: Schedule Risks are the main concern for our project at the moment. Every member has 3 other classes that compete for time with this project.
Resolution: Communication. Our outside class meeting are held regularly and attended by all members and we use Discord heavily to stay in contact with each other. We attempt to make the most of the time committed by each member.

Risk: Technical Risks. There will be unknown technical challenges to be overcome involving three major technologies (MySQL, Node, React) interacting with one another.
Resolution: Development Branches and experimenting on local machines. Keeping separate branches for testing and being able to try things out in a vm on your machine help in understanding the core technologies. When problems arise we can try solutions locally and bring our results to Discord.

Project management

The initial planning of our approach to milestone 2 was to meet and first decide what exactly our vertical software stack was going to look like. We decided on a simple react application that would display several listing database objects (listings) with a simple search bar feature. Due to the fact that we all have several other upper division classes to worry about we worked in small spurts up till about the Saturday before the deadline where a couple members who had the time and comfort with the system made the final push for the deadline. We used a task tracking board to put small jobs up and made great use of our discord channel to post tutorials and notify everyone of updates on the project. For this deadline, most members of the team never even touched the codebase, which is a problem that needs to be addressed between now and the start of work for the next milestone. Rather than try and get everyone up to speed in the small amount of time we had for this milestone, we rather had mainly Jahon and Michael (hats off to them) handle a large amount of the setup and work for the vertical prototype. We appreciate their efforts greatly, and hope to have the work more evenly distributed for the next milestone.