



Final Project

for SW Engineering

Class CSC 648-848 Spring 2019

Team 9 Milestone 5

GatorRooms.com

URL

<http://ec2-54-241-132-56.us-west-1.compute.amazonaws.com/>

Inez Wibowo
Team Lead & Document
Editor
iwibowo@mail.sfsu.edu

Aliaksei Siarheyeu
Front End & Back End Lead
Github Master

Marcus Wong
Front End & Back End Lead

Romeel Chaudhari
Back End Developer

Ismael San Juan
Front End Developer

Hang Li
Front End Developer

Jia Nan Mai
Front End Developer

1. Product Summary

The rental website GatorRooms.com is a website for San Francisco State University students. SFSU students will find it easier to find rentals on a website designed specifically with features important to students. GatorRooms therefore stands out by prioritizing sorting by price and by distance to campus. The website is also lightweight (support ~50 users at a time) and it's easy for students to find rentals on the go, i.e. on mobile. Registered users/landlords on GatorRooms.com can also post their available rentals easily and will be able to post only after the post has been verified by GatorRooms.com admin.

We hope students and registered users alike find GatorRooms a website indispensable and spread the availability of the site through word-of-mouth, through Slack channels, and posters across campus.

All major committed functions this team will deliver (P1):

User - Guest User

- ✓ User shall be able to browse the list.
- ✓ User shall be able to sign up and register.
- ✓ User shall be able to sort the list ascending, descending date of posting and monthly rent price.
- ✓ User shall be able to click display the property's larger version of the image.
- ✓ User shall be able to review the listings quickly whether they are on mobile or desktop.
- User shall be able to sort based on distance to the campus. Filter available
- ✓ User shall be able to review the About page and read team's profile on mobile and desktop easily.

User - Registered User

- ✓ Users shall be able to login to their accounts.
- ✓ Users shall be able to post after the post have been approved by admin.
- ✓ Users / Landlords shall be able to publish multimedia like images so that students can view it.
- ✓ Users/ Renters shall be able to send a message from their dashboard to the landlord/listing owner.
- ✓ Users / Landlord shall have a dashboard allowing them to review messages from user and contact the student.

User - Admin

- ✓ Admins shall be able to login as admin.
- ✓ Admins shall be able to review, remove, approve listings before they go live.
- ✓ Admins shall be able to reject but not modify the listings from customers.

Locally it is <http://localhost:3000>

Amazon Web Services URL is : <http://ec2-54-241-132-56.us-west-1.compute.amazonaws.com/>

2. Milestone Documents

Attached



SW Engineering CSC648/848

Spring 2019

GatorRooms.com

Team 9 Milestone 1

Product Requirements Document

3/8/2019

Inez Wibowo

Team Lead & Document
Editor

iwibowo@mail.sfsu.edu

Aliaksei Siarheyeu

Front End & Back End Lead
Github Master

Marcus Wong

Front End & Back End Lead

Romeel Chaudhari

Back End Developer

Ismael San Juan

Front End Developer

Hang Li

Front End Developer

Jia Nan Mai

Front End Developer

Versions

History Table

Version	Date	Notes
1.0	3/6/2019	Initial draft for Milestone 1 submission.
1.1	3/7/2019	Incorporated feedback received on 3/6 for M1 submission.
1.2	3/8/2019	Incorporated feedback received on 3/ 8 for M1 submission and freeze.

1. Executive Summary

The rental website GatorRooms.com was created with the desire to build a website that is useful and built specifically for San Francisco State University students. We believe that SFSU students have unique rental preferences and their rental experiences can be better served by a website that is specifically designed for them, which may include features such as providing distances to the campus, or to study spots around the neighborhood, etc. The website is meant to be in general more attractive and relevant to those SFSU students compared to more general purpose sites.

The rental website features sorting based on price, search functionality, rental terms (temporary, long term, etc), means of transportation to school (bike, walk, bus), distance to study areas (i.e. coffee shops, libraries, wifi spots, tea shops, printing shops like Kinkos/FedEx, etc.) and general usability features that would make it easy for students to review rentals on the go.

The general user experience on GatorRooms.com is comparable to other rental websites and aligned to the average expectations of any users of rental website today. The rental website should require very little to no learning experience as it is also built to be intuitive, simple, and fast for student users as well as landlord users.

All users can search, sort, view, and click on any of the listings on the site. Once a user registers, they are able to access higher level of features, such as posting a listing, viewing communications, bookmarking a listing, and adding/modifying alerts.

GatorRooms.com is a website designed by SFSU senior students who have had years of experience searching for rentals. We believe that a website designed by, and made specifically for, students that is fast, highlights the right features, and easy to use on mobile and desktop will prove to be valuable and competitive to our target audience compared to a general public site. Our site is designed to support ~50 users concurrently and we are confident that this niche website will serve our students and landlords for a

long time.

2. Personae and Use Cases

Personae

Actor - Student



Students are members of the university. Some require renting because they do not have family housing available and some prefer to live closer to school. Some students require sharing rent cost and others do not. Presumably, the average modern day student is relatively familiar with technology and can follow/operate simple to moderately complicated user interface design. In spite of their savvy technical skills, we want students to enjoy a simple experience. They will be interested in seeing a website that can provide distance to school, sorting by price, and a simple experience. We will also make sure to pay attention to website security, so that students will have a safe user experience.

Students are under pressure and would appreciate a website that makes it easy for them to act quickly on a rental apartment deal. Many students use lingo and simplified English, they do not usually communicate with advanced formal English. All San Francisco State University students speak English. Most students understand how to rent an apartment, that is, they understand that certain procedural steps are required to achieving an apartment lease such as contact with the landlord.

Actor - Landlord



Landlords can work independently or represent a realty group. Landlords rely on email to do their business online. They are highly communicative with their clients. Some Landlords will receive large amounts of emails while others are not very comfortable with technology. Some Landlords may be comfortable with emails that are available to the public because their clientele is the public. However, some may have privacy concerns because the protection of their data is a priority.

We expect Landlords who visit our site to have some comfort with computers but a respectively low level of technological skill. Landlords are hard working, busy people, and maybe juggling many tasks, therefore Landlords would highly desire a simplified user interface that provides the convenience when only limited amount of time is available and there is a sense of high urgency. Additionally, being able to post and read messages with very low effort, will satisfy a typical Landlord and hopefully would enjoy interacting with a rental website. Many Landlords have particular rules and preferences that are in accordance to their lease contracts which directly affect students. This direct effect between users is very important because it can cause our student clients to be discouraged from using the website. Because we will allow Landlords to have very specific details regarding their ads, we will respond by providing students the ability to filter out all the posts that do not meet their preferred living criteria.

Actor - Admin



Admins are detail focused observers of computer and network systems. Admins have a moderate to high technological skill level. Admins are not expected to have programming skills but may have programming knowledge. Admins understand how to make database queries and use databases. Admins are trusted

agents who are employed by the company to manage site content and administrate the website's user behavior therefore admins will do things like filter our website data. Admins will not be using automation scripts work will be done manually and carefully so that the quality of the listings they manage encourages confidence and trust. Admins have good communication skills and are able to communicate to both clients and with the developers. Admins should be presented with easy to use and understand administration tool sets, so that we can help support fast paced demands and heavy workloads.

Use Case

2.1. Student: Mary - A rental website with alert features.

Mary, an SFSU student, needs a rental apartment. She finds GatorRooms.com and enjoys the simplicity of the site since she can use the site **without registering**. She **sorts by lowest price** and sets an **alert** to be emailed when a listing is available in her price range. She can check listing details on the go, while she is commuting on the bus, or walking, or having lunch with her friends. She can feel secure that she does not need to provide her email address to the landlord to contact them, as the **email addresses are masked**. She can use the same website without having to go to another mail app. She **logs in** and **reads** messages from the landlord, or she can **post a reply** message. She can easily, and securely find a rental apartment without taking too much time out of her studies.

2.2. Student: Jon - A student seeking a student roommate using lifestyle features.

Jon is an SFSU student who is renting an apartment close to SFSU but his current roommate is soon moving out and he needs to find a new roommate. He has a good relationship with the landlord and the landlord trusts him to find a roommate to share the total costs of the two bedroom apartment. Jon finds GatorRooms, and sees it is directed specifically to SFSU students, whom he feels he can relate easily with and may have the same lifestyle. He can **categorize his room availability** under shorter rental terms on GatorRooms.com because the landlord only wants month to month and he also includes the **distance between the rental, to the campus**, or between **popular study spots** by walking, biking, or driving, as he knows students will find helpful.

2.3. Landlord: Jill - A realtor experienced with renting needs fast performance and flexibility.

Jill is a busy realtor and she is always in her car, driving to the next open house, or on her laptop in a coffee shop entertaining prospective clients. However, she realizes that at the same time, she needs to create a more stable source of income and had bought a house within walking distance to SFSU. She finds GatorRooms.com and finds the website **easy to access on her laptop** and **can create a post between meetings**. She is a realtor so she likes that she **can use her business email address as a contact** and does not need a masked email address. She prefers to keep it all in her business email account because she highly monitors that one inbox. She is a skilled marketer and knows how to make her listing attractive and finds 3 freshman renters that she can be sure will provide her with rental income until graduation.

2.4. Landlord: Steve - A landlord inexperienced with technology, likes suggestions and simplicity.

Steve is newly retired and he has some time on his hands. He likes to stay active but finds that his retirement income is not giving him enough income living in the bay area. He has an apartment he bought in the Sunset that he can put on the market but he is not familiar with listing rentals to the public. He finds GatorRooms.com and is more comfortable because the site **looks simple** and **easy to use**, and it targets a small set of audience, SFSU students. He is able to view other listings and see examples of how others have posted without having to register. Once he is ready, he finds the posting process easy, as the pages **prompts him to register**, and **provides tips on how to make the listings attractive**, and he becomes more confident that the posting can be attractive. He likes that his email address is masked, and he wants students to contact him through the site.

2.5. Admin: David - An inexperienced admin leverages convenience tools on the dashboard.

David is a new website administrator who just graduated from SFSU. He is excited to work on GatorRooms.com because he is familiar with the target market, and believes the work level is just right. GatorRooms can only hire one developer and his automation experience is limited. He still believes he can do a great job. Using the **Admin dashboard** on GatorRooms, he can review, approve, and reject, several listings **all on one page, in bulk**, without having to click on each link, and he is not required to edit the postings himself. Considering GatorRooms.com has a small database, David can go to his dashboard and tackle most of the posting requests coming in during the 8-hour work day work level and does a great job. Additionally, as a plus, he can effectively **troubleshoot** logging in issues, **review and approve all postings** on the website **before they go live, on the same dashboard page**.

3. Data Definitions and Entities

Listing The advertisement posting of the apartment by the landlord. Includes the date it was posted, image, number of rooms, address, distance, and price.

- **Image** An image should be a proper size and resolution, and should be only of an apartment.

- **Address** A residential valid address that follows a defined format.

- **Price** A price of the listing per month in US dollars.

- **Distance** A distance in miles to SF State campuses.
 - **Type** A category of listings available, for example a room for rent in a house, in an apartment, the whole house/apartment
- Guest User** A user that can browse website and see the listings and contact Landlords.
- Registered User** The same abilities as a guest user, but their information will be stored.
- Admin User** A user that will verify the listings submitted by the Landlord users.
- Landlord User** A user that will have an ability to create the listings and send them for review.
- User Email** Email address of the user that will needed to be verified to validate this a legitimate address.
- User Phone Number** Phone number to connect to the user for admin purposes.
- Message** A communication unit that will be used by Users. Includes content, date, sender and recipient.
- Favorite** A rental listing can be added to the set of saved info of user's database.
- Registration record** A set of information obtained during registration such as first name, last name, email address.

4. Functional Requirements

Guest User (Not registered)

1. User shall be able to browse the list and details of the listing best on desktop.
2. User shall be able to sort the list ascending, descending date of posting, monthly rent price, distance to the campus, distance to popular study spots, and filter amenities (i.e., garage, community amenities, heating/cooling and more types of filters).
3. User shall be able to review a listing and find commuting directions and time estimate
4. User shall be able to review the About page and read team's profile on mobile and desktop easily.
5. User shall be able to review the listings quickly whether they are on mobile or desktop.
6. User shall be able to view GatorRooms.com's social media platforms
7. User shall be able to fill out a "Refer a friend" form for their friends, families and relatives.
8. User shall be able to print their listings in a printer-friendly format from the site.
9. User shall be able to click a different way of viewing the listing and display the property's larger version of the image.
10. User shall be able to have access to a built-in rental calculator to help students avoid committing to a property they cannot afford and keep rental expectation in check.
11. User shall be able to view the Frequently Asked Questions(FAQ) page.
12. User shall be able to access the forum page where they can contact other students or landlords for assistance.

Registered User

13. Users / Landlord shall have a dashboard allowing them to review messages from user and contact the student to reply.

14. Users shall be able to post message to their dashboard to contact the landlord/listing owner.
15. Users shall be able to bookmark or save their favorite listings to their shopping cart.
16. Users shall be able to login to access favorited listings and respond to messages.
17. Users shall be able to create alerts and be emailed when the listing fitting their criteria is posted.
18. Users / Landlords shall be able to publish multimedia like images, or a videotape(virtual tour) of their properties so that students shall be able to view it.

Admins

19. Admins shall be able to have a dashboard allowing them to review and reject posting in bulk.
20. Admins shall be able to review and approve listings before they go live.
21. Admins shall be able to delete but not modify the listings from customers.
22. Admins shall be able to help users troubleshoot issues with registration such as logging in, form issues, resetting password, reconciling old/new accounts, etc.

5. Non Functional Requirements

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO).
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers
3. Selected application functions must render well on mobile devices
4. Data shall be stored in the team's chosen database technology on the team's deployment server.
5. No more than 50 concurrent users shall be accessing the application at any time
6. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
7. The language used shall be English.
8. Application shall be very easy to use and intuitive.
9. Google analytics shall be added.
10. No e-mail clients shall be allowed.
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated.
12. Site security: basic best practices shall be applied (as covered in the class).
13. Before posted live, all content (e.g. apartment listings and images) must be approved by site administrator.
14. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development.
15. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2019. For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application).

5. Competitive Analysis

The goal of our website is that it strikes a balance between being comparable with competitors and especially attractive to SFSU students.

	Zillow.com	Homes.com	Craigslist.com	GatorRooms.com
Search Bar	++	++	+	+
Guest Browse	+	+	+	+
Sort by price	+	+	+	+
Login	+	+	+	+
User Friendly	+	+	+	+
Saved Searches	++	+	+	+
Favorites	-	+	-	+
Sort by walking, biking, driving, muni time estimate to school	-	-	-	+
Simple and fast posting process	-	-	++	++

GatorRooms.com vs Competitors comparison

Gatorrooms compares favorably against the competition that currently is available. GatorRooms stand out by being the only website that is not lacking in any of the features we deem necessary because we cater to a specific target audience, which are SFSU students. Going category by category we can see that Zillow.com and Homes.com have an upper hand when it comes to their search bar. All four sites have the capability to browse homes and they all do it about the same. All four sites have a good sort by price option as well as an easy login process. After trying out all websites it was determined that all 4 websites have a friendly user interface. Where GatorRooms.com stands out is by being the only website to provide the SFSU students with a walking, biking, driving and muni time estimate to school. Additionally, GatorRooms posting experience will be a simple and fast process that would require very little to no experience in posting.

6. System Architecture and Technologies

Team 9 stack:

- Server Host: Amazon EC2, Instance size 8 gB (1 CPU and 1GB RAM)

- Operating System: Amazon Linux 2 AMI (HVM), SSD Volume Type
- Database: MySQL 5.5.62
- Web Server: Amazon Web Services
- Server-Side Language: JavaScript (1.7[ECMAScript2018])
- Additional Technologies:
 - UI Framework : React.js Material UI
 - Backend Framework: Node.js Express, NodeJS 11.10.1
 - Team Communication: Slack, Trello, Zoom
- Browsers we support:
 - Google Chrome: Google 74.0, 72.0.3626
 - Safari: Safari 12.03, 12.02

7. Team

1.	Romeel Chaudhari	Back End Developer	
2.	Hang Li	Front End Developer	
3.	Jia Nan Mai	Front End Developer	
4.	Ismael San Juan	Front End Developer	
5.	Aliaksei Siarheyeu	Front End & Back End Lead	Github Master
6.	Marcus Wong	Front End & Back End Lead	
7.	Inez Wibowo	Team Lead	Document Editor

8. Checklist

Team found a time slot to meet outside of the class	✓
Github master chosen	✓
Team decided and agreed together on using the listed SW tools and deployment server	✓
Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing	✓

Team lead ensured that all team members read the final M1 and agree/understand it before submission	✓
Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)	✓



SW Engineering CSC648/848

Spring 2019

GatorRooms.com

Team 9 Milestone 2

Product Requirements Document

Inez Wibowo

Team Lead & Document
Editor

iwibowo@mail.sfsu.edu

Aliaksei Siarheyeu

Front End & Back End Lead
Github Master

Marcus Wong

Front End & Back End Lead

Romeel Chaudhari

Back End Developer

Ismael San Juan

Front End Developer

Hang Li

Front End Developer

Jia Nan Mai

Front End Developer

Versions

History Table

Version	Date	Notes
1.0	3/22/19	First version for submittal.
2.0	4/1/19	Second version. Revised document to incorporate feedback from professor.

1. Data Definitions and Entities

Listing -	<p>The advertisement post of the apartment by the landlord. It includes the date it was posted, images, number of rooms, address, distance, and price of the apartments.</p> <p>This ‘Listing’ class will include the ‘description’ method.</p>
- Image -	<p>An image of the apartment within acceptable size and resolution.</p> <p>Image formats supported:- jpeg, png, bmp, gif, tiff, dicom.</p> <p>A maximum image size of 7 MB per picture.</p> <p>No video.</p> <p>The listing will include ‘image’ field.</p>
- Address -	<p>A valid residential address that follows a defined format.</p> <p>A valid address contains the following:</p> <ul style="list-style-type: none">All capital letters, No commas or periodsA valid and deliverable addressIncluded company name, apartment number, or suite number if applicableValid acronyms and abbreviationsA valid ZIP CODE <p>The example below shows a valid address:</p> <p>JACK SMITH COMPANY LTD 123 N MAIN ST STE 45 SF CA 94132</p> <p>This ‘Listing’ class will include the ‘address’ method.</p> <p>The ‘Address’ class will include street address, city, state, zipcode.</p>
- Price -	<p>This ‘Listing’ class will include:</p> <p>A price (amount method) of the listing per month in <u>US dollars</u>. Filters will be available.</p> <p>Amount type (amountType method) is also available for listings showing per month, per semester, per year.</p>

- Distance -	A distance in <u>miles</u> to SF State campuses. Filters will be available.
- Type -	A category of listings available, for example a room for rent in a house, in an apartment, the whole house/apartment.
User-	
-Guest User -	A user that can browse website and see the listings and contact Landlords. The user can start to post a listing, can start a message to to landlord, but they will not be able to save or continue unless they become a registered user. Their information will not be stored.
-Registered User-	A registered user has privileges guest user has, and additional privileges, which include ability to post a listing, contact another registered user, and have an account in which this information is stored. For example, landlord will be able to post listing, renter can contact landlord, and their registration information will be stored. Information will be stored are, at a minimum : email address, password, first and last name.
-Admin User -	A user who will be able to approve listing before they are published and reject the posting without editing it. Data will be flagged “is_admin”.
-User Email -	Email address of the user that will needed to be verified to validate this is a legitimate address. Validation will include that the address was entered, then an “@” symbol, and then something else followed @.
User Phone Number-	Phone numbers to connect to the user for admin purposes. These are the valid formats. (123) 456-7890 (123)456-7890 123-456-7890 123.456.7890 1234567890
Message -	A communication unit that will be used by Users. Includes content(text), date, sender's and recipient's name. Text field will have ability to include one standard text version only.

	Message class will include fields from, to, message, date, status “is_read”, and the listing ID it is referring to.
Favorite -	A rental listing can be added to the set of saved info of user’s database. A “heart” heart-shaped icon can be clicked to indicate to the user that the listing will be saved.
Registration record -	A set of information obtained during registration such as first name, last name, email address

2. Functional Requirements

User - Guest User

First Priority (P1)

User - Guest User

- P1-Guest-1. User shall be able to browse the list.
- P1-Guest-2. User shall be able to sign up and register.
- P1-Guest-3. User shall be able to sort the list ascending, descending date of posting and monthly rent price.
- P1-Guest-4. User shall be able to click display the property's larger version of the image.
- P1-Guest-5. User shall be able to review the listings quickly whether they are on mobile or desktop.
- P1-Guest-6. User shall be able to sort based on distance to the campus.
- P1-Guest-7. User shall be able to review the About page and read team's profile on mobile and desktop easily.

User - Registered User

- P1-Reg-1. Users shall be able to login to their accounts.
- P1-Reg-2. Users shall be able to post after the post have been approved by admin.
- P1-Reg-3. Users / Landlords shall be able to publish multimedia like images so that students can view it.
- P1-Reg-4. Users/ Renters shall be able to send a message from their dashboard to the landlord/listing owner.
- P1-Reg-5. Users / Landlord shall have a dashboard allowing them to review messages from user and contact the student.

User - Admin

- P1-Admin-1. Admins shall be able to login as admin.
- P1-Admin-2. Admins shall be able to review, remove, approve listings before they go live.
- P1-Admin-3. Admins shall be able to reject but not modify the listings from customers.

Second Priority

User - Guest User

- P2-Guest-1. User shall be able to click on a different way of viewing the listing, i.e. list, thumbnails, etc.
- P2-Guest-2. User shall be able to filter based on distance to popular study spots, and filter amenities (i.e., garage, community amenities, heating/cooling and more types of filters).
- P2-Guest-3. User shall be able to print their listings in a printer-friendly format from the site.
- P2-Guest-4. User shall be able to view the Frequently Asked Questions(FAQ) page.

P2-Guest-5. User shall be able to review a listing and find commuting directions and time estimate.

User - Registered User

- P2-Reg-1. Users / Landlords shall be able to publish multimedia like images so that students can view it.
- P2-Reg-2. Users shall be able to send message from their dashboard to the landlord/listing owner.
- P2-Reg-3. Users / Landlord shall have a dashboard allowing them to review messages from user and contact the student.

User - Admin

- P2-Admin-1. Admins shall be able to have a dashboard allowing them to review and reject posting in bulk.

Last Priority

User - Guest User

- P3-Guest-1. User shall be able to print their listings in a printer-friendly format from the site.
- P3-Guest-2. User shall be able to have access to a built-in rental calculator to help students avoid committing to a property they cannot afford and keep rental expectation in check.
- P3-Guest-3. User shall be able to access the forum page where they can contact other students or landlords for assistance.
- P3-Guest-4. User shall be able to fill out a “Refer a friend” form for their friends, families and relatives.
- P3-Guest-5. User shall be able to view GatorRooms.com’s social media platforms.

User - Registered User

- P3-Reg-1. Users shall have a dashboard viewing status of their postings, i.e. under review, approved.
- P3-Reg-2. Users/Landlords shall be able to publish videos (virtual tour) of their properties so that students can view it.
- P3-Reg-3. Users shall be able to bookmark or save their favorite listings.
- P3-Reg-4. Users shall be able to create alerts and be emailed when the listing fitting their criteria is posted.

User - Admin

- P3-Admin-1. Admins shall be able to help users troubleshoot issues with registration such as logging in, form issues, resetting password, reconciling old/new accounts, etc.

3. UI Mockups and Storyboards

Admin Dashboard - Reviewing, Approving (Jia Nan)

Registered User/Landlord Dashboard - Write Posting, Submitting for admin review (Ish)

Guest User Dashboard - Reviewing communication from landlord (Inez)

Guest User - Browsing, Sorting (Hang)

3.1. Use Case: Guest User Mary

Mary, an SFSU student, needs a rental apartment. She finds GatorRooms.com and enjoys the simplicity of the site since she can use the site **without registering**. She **logs in** and **reads** messages from the landlord, or she can **post a reply** message. She can easily, and securely find a rental apartment without taking too much time out of her studies.

Mary sees website page with listings

The screenshot shows a web-based application for finding apartments. At the top left, there's a sidebar with a 'Rental Type' section containing 'Apartment' and 'House' options. Next to it is a search bar with a magnifying glass icon and the text 'Daly city, Ca'. To the right of the search bar are navigation links: 'Home', 'Post', 'About', and 'Login/Reg'. Below these links is a horizontal menu with 'Sort by' (set to 'Distance'), 'Price', and 'sqft' buttons. The main content area is titled 'Listings, Daly City House for rent' and displays four apartment listings. Each listing includes a small thumbnail image, the address, price, square footage, distance to SFSU, and a 'Save' button. The first listing is for a house at 272, Alameda Blvd, Daly City, CA 94***, 3beds/2ba, \$3000/mo, 1200 sq ft, 1.9 mi, with a 'Save' button. The second listing is for a house at 333, 2nd Blvd, Daly City, CA 94***, 2beds/2ba, \$2000/mo, 1000 sq ft, 2.1 mi, with a 'Save' button. The third listing is for a house at 555, Palisades Blvd, Daly City, CA 94***, 1bed/1ba, \$1500/mo, 800 sq ft, 3.0 mi, with a 'Save' button. The fourth listing is for a house at 556, Kent Blvd, Daly City, CA 94***, 3beds/3ba, \$3600/mo, 1500 sq ft, 3.5 mi, with a 'Save' button. At the bottom of the listing table, there are navigation arrows: '< 1 2 3 .. 13 >'.

Address	Price \$	sqft	Distance to SFSU mi	Action
272, alameda blvd, Daly City, CA 94*** 3beds/2ba	\$3000/mo	1200 sq ft	1.9	<input type="button" value="Save"/>
333, 2nd blvd, Daly City, CA 94*** 2beds/2ba	\$2000/mo	1000 sq ft	2.1	<input type="button" value="Save"/>
555, palisades blvd, Daly City, CA 94*** 1beds/1ba	\$1500/mo	800 sq ft	3.0	<input type="button" value="Save"/>
556, kent blvd, Daly City, CA 94*** 3beds/3ba	\$3600/mo	1500 sq ft	3.5	<input type="button" value="Save"/>

Mary clicks on one of the listings and it pops up as a new child window, so she can go back to the browser where the listings are displayed.

1461 Duck Creek Road, Daly City, CA 94015
1 bed 1 bath 800 sqft \$1500/mo
[Map View](#)

- Great for city commuters.
- Close to all amenities and SFSU.
- Designated parking spot. Lots of street Parkings.
- Very open deck to enjoy Sunset daily.

[Favorite](#)
[Contact Landlord](#)

Guest User Mary's Storyboard



Mary sees website page with listings, and selects one of the listings.

Mary clicks on one of the listings and visits the listing page. Clicks “Contact Landlord” button and is asked to Login/Register first.

Mary is asked to Login/Register Page before submitting her message to the landlord/poster.

3.2. Use Case: Registered User/Landlord Jill

Jill is a busy realtor and she is always in her car, ...finds GatorRooms.com and finds the website **easy to access on her laptop** and **can create a post between meetings**. She is a realtor so she likes that she **can**

use her business email address as a contact and does not need a masked email address.

Jill creates her posting on New Listing page

The screenshot shows a web-based application for listing rooms for rent. At the top, there is a search bar with the placeholder "Daly city, Ca" and a magnifying glass icon. Below the search bar is a navigation menu with links: Home, Post, About, Message, and Login/Reg. On the left side, there is a sidebar with filters for "Rooms for rent": "Description only", "Near SF/SU", "Posted this week", and "Date Posted Newest first". There are also filters for "Price" (Low to High, High to Low) and "Type" (Apartment, House). The main content area is titled "New Listing". It contains four mandatory fields with asterisks: "Price*", "Bed*", "Bath*", and "Image*". The "Image*" field is currently empty and displays a large red X. Below these fields is a note: "*Asterisk marks mandatory fields." A large text area labeled "Description" is provided for the listing details. At the bottom right, there are two buttons: "Cancel" and "Submit". To the right of the "Submit" button, a note states: "All postings are reviewed and posted upon approval."

Daly city, Ca

Home Post About Message Login/Reg

Rooms for rent

- Description only
- Near SF/SU
- Posted this week
- Date Posted Newest first

Price

- Low to High
- High to Low

Type

- Apartment
- House

New Listing

Price* :

Bed* :

Bath* :

Image*:

*Asterisk marks mandatory fields.

Description

Cancel Submit

All postings are reviewed and posted upon approval.

After logging in, Jill can view her list of postings and its status

The screenshot shows the homepage of gatorrooms.com. At the top, there is a search bar with the placeholder "Search" and a magnifying glass icon. To the right of the search bar are two blue buttons: "Home" and "New Listing". Below the search bar, the text "Welcome to Your Hub" is displayed in purple. To the left of this text is a placeholder image of a person's head and shoulders.

Your Properties

- 22 Arballo Drive
- 600 Stockton

About

Cool lenient, renter friendly landlord that is looking for a great group of individuals to take care of my properties.

Messages

2 Unread Messages

22 Arballo Drive

Active Message Remove

Price: \$1,500
Bed : 2
Bath : 1

Price: \$1,500
 Bed : 2
 Bath : 1

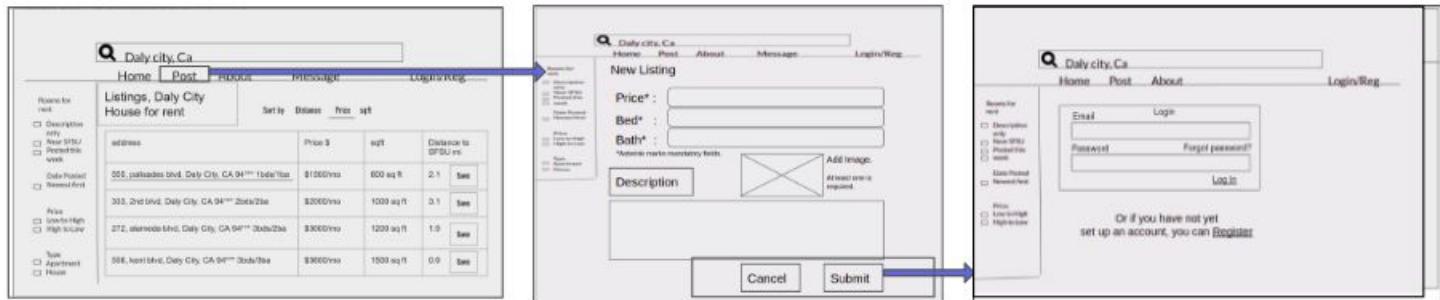
600 Stockton

Request to Make Active Message Remove

Price: \$1,750
Bed : 1
Bath : 1

Price: \$1,750
 Bed : 1
 Bath : 1

Registered User/Landlord Jill's Storyboard

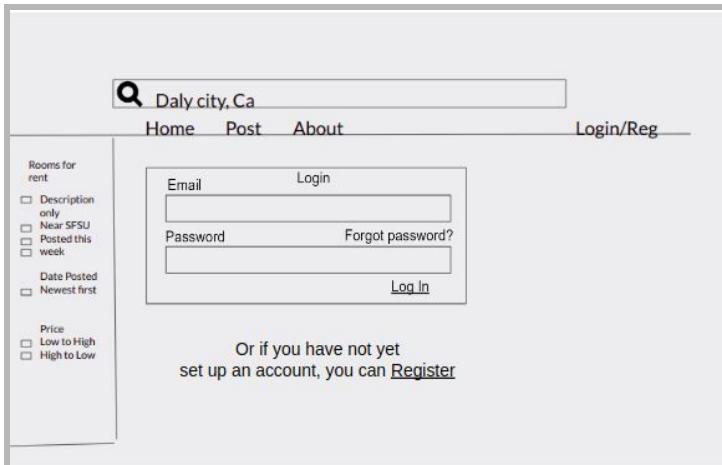
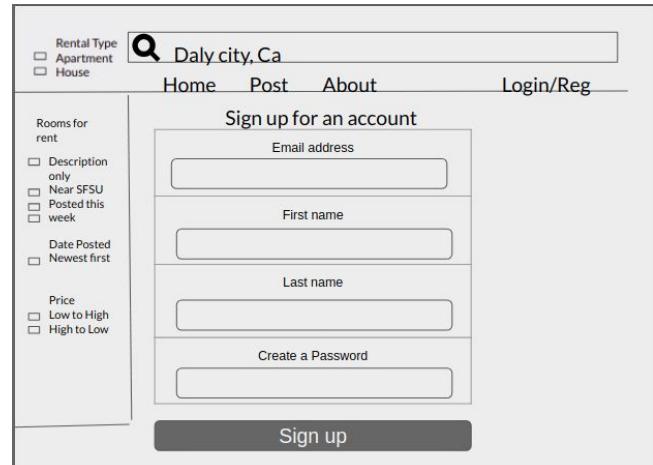


Jill sees website page with listings and clicks on “Post”.	Jill fills out her listing details and clicks “Cancel” or “Submit”.	Jill is asked to Login/Register Page before submitting her listing to the admin for review.
--	---	---

3.3. Use Case: Registered User/Landlord Jon’s Login Experience

Jon represents all users who want to login/register. They are looking to complete the login/registration process painlessly with minimum clicks and forms to fill out.

Jon wants to register and a form to register is presented to sign up for an account.

 <p>Daly.city.Ca</p> <p>Home Post About Login/Reg</p> <p>Rooms for rent</p> <ul style="list-style-type: none"> <input type="checkbox"/> Description only <input type="checkbox"/> Near SFSU <input type="checkbox"/> Posted this week <input type="checkbox"/> Date Posted <input type="checkbox"/> Newest first <p>Price</p> <ul style="list-style-type: none"> <input type="checkbox"/> Low to High <input type="checkbox"/> High to Low <p>Email <input type="text"/> Login Password <input type="password"/> Forgot password? <input type="button" value="Log In"/></p> <p>Or if you have not yet set up an account, you can Register</p>	 <p>Daly.city.Ca</p> <p>Home Post About Login/Reg</p> <p>Rental Type</p> <ul style="list-style-type: none"> <input type="checkbox"/> Apartment <input type="checkbox"/> House <p>Rooms for rent</p> <ul style="list-style-type: none"> <input type="checkbox"/> Description only <input type="checkbox"/> Near SFSU <input type="checkbox"/> Posted this week <input type="checkbox"/> Date Posted <input type="checkbox"/> Newest first <p>Price</p> <ul style="list-style-type: none"> <input type="checkbox"/> Low to High <input type="checkbox"/> High to Low <p>Sign up for an account</p> <p>Email address <input type="text"/> First name <input type="text"/> Last name <input type="text"/> Create a Password <input type="password"/></p> <p><input type="button" value="Sign up"/></p>
---	--

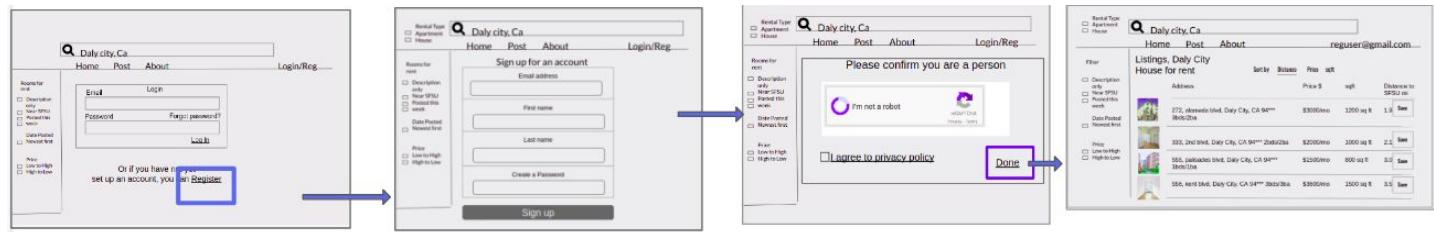
Jon confirms he is a real person during creation of new account

 <p>Daly.city.Ca</p> <p>Home Post About Login/Reg</p> <p>Please confirm you are a person</p> <p>I'm not a robot <input type="checkbox"/></p> <p><small>reCAPTCHA Privacy - Terms</small></p> <p><input type="checkbox"/> I agree to privacy policy <input type="button" value="Done"/></p>

Following login, he lands on GatorRooms.com homepage with his email address displayed at top-right corner.

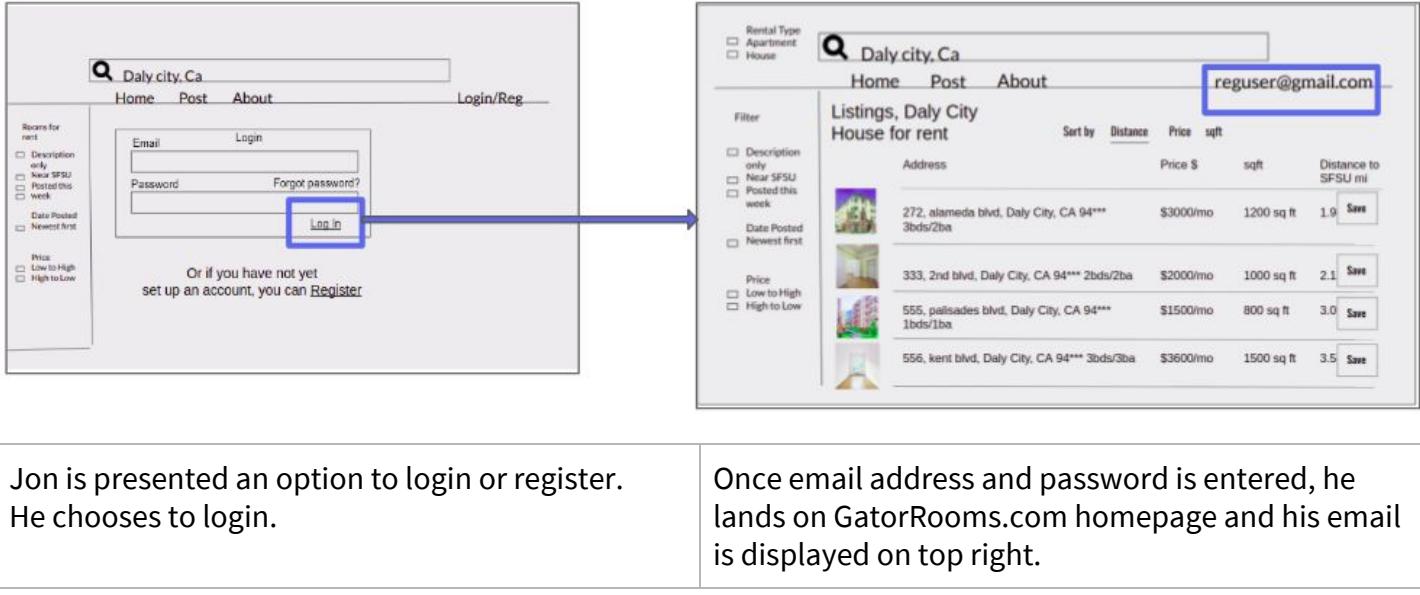
Address	Price \$	sqft	Distance to SFSU mi	Action
272, alameda blvd, Daly City, CA 94*** 3bds/2ba	\$3000/mo	1200 sq ft	1.9	<input type="button" value="Save"/>
333, 2nd blvd, Daly City, CA 94*** 2bds/2ba	\$2000/mo	1000 sq ft	2.1	<input type="button" value="Save"/>
555, palisades blvd, Daly City, CA 94*** 1bds/1ba	\$1500/mo	800 sq ft	3.0	<input type="button" value="Save"/>
556, kent blvd, Daly City, CA 94*** 3bds/3ba	\$3600/mo	1500 sq ft	3.5	<input type="button" value="Save"/>

Storyboard - Jon's registration flow



Jon is presented an option to login or register. He clicks Register link.	Jon fills out the form with simple info: email, first, last name, password.	For his security, he confirms that he is not a robot, and agrees to Privacy terms	Once complete, he lands on GatorRooms.com homepage and his email is displayed on top right.
---	---	---	---

Storyboard - Jon's login as existing user flow



3.4. Use Case: Admin Dashboard David

David is a new website administrator... using the **Admin dashboard** on GatorRooms, he can review, approve, and reject, several listings **all on one page, in bulk**. He can also **review and approve all postings** on the website **before they go live, on the same dashboard page**.

David's dashboard where he can see the status of listings he has reviewed: "x" for Reject, "✓" for Approved and "N/A" is shown for pending. He can filter by status, Approved, Rejected, or Pending.

Rental Type

Apartment
 House

Daly city, Ca

Home
Post
About
adminuser@gatorrooms.com

Filter

- Description only
- Near SFSU
- Posted this week

- Date Posted
- Newest first

- Price
- Low to High
- High to Low

Listings, Daly City

House for rent

Filter by
Approved
Rejected
Pending

Address	Price \$	sqft	Status	
	272, alameda blvd, Daly City, CA 94*** 3bds/2ba	\$3000/mo	1200 sq ft	<input checked="" type="checkbox"/>
	333, 2nd blvd, Daly City, CA 94*** 2bds/2ba	\$2000/mo	1000 sq ft	<input checked="" type="checkbox"/>
	555, palisades blvd, Daly City, CA 94*** 1bds/1ba	\$1500/mo	800 sq ft	<input checked="" type="checkbox"/>
	556, kent blvd, Daly City, CA 94*** 3bds/3ba	\$3600/mo	1500 sq ft	N/A

David's dashboard and the listings he can edit, to either approve or reject.

Rental Type
 Apartment
 House

 Daly city, Ca

Home Post About adminuser@gatorrooms.com

Filter

Description only
 Near SFSU
 Posted this week

Date Posted
 Newest first

Price
 Low to High
 High to Low

Listings, Daly City
House for rent

Address	Price \$	Approve	Reject
272, alameda blvd, Daly City, CA 94*** 3bds/2ba	\$3000/mo	<input checked="" type="checkbox"/>	<input type="checkbox"/>
333, 2nd blvd, Daly City, CA 94*** 2bds/2ba	\$2000/mo	<input type="checkbox"/>	<input checked="" type="checkbox"/>
555, palisades blvd, Daly City, CA 94*** 1bds/1ba	\$1500/mo	<input checked="" type="checkbox"/>	<input type="checkbox"/>
556, kent blvd, Daly City, CA 94*** 3bds/3ba	\$3600/mo	<input type="checkbox"/>	<input type="checkbox"/>

3.5. Use Case: Renter Nancy & Diana Landlord Messaging Dashboard

Nancy can use the same website without having to go to another mail app. She **logs in** and **reads** messages from the landlord, or she can **post a reply** message. She can easily, and securely find a rental apartment without taking too much time out of her studies.

Diana does not need a masked email address. She prefers to keep it all in her business email account because she highly monitors that one inbox but likes the **convenience of the dashboard** while she is checking in with **all of her listings/renters at once**.

After Nancy the renter logs in, she can view the messages from the landlords she contacted.

The diagram illustrates Nancy's interaction with the rental listing website. On the left, the search results page for "Daly City, Ca" shows a list of four rental properties. Nancy clicks on her email address, "reguser@gmail.com", which is highlighted in a red box. A blue arrow points from this click to the right side of the diagram.

Left Panel (Search Results):

- Rental Type:** Apartment, House
- Search Bar:** Daly.city.Ca
- Navigation:** Home, Post, About
- Email Address:** reguser@gmail.com
- Filter Options:**
 - Description only
 - Near SFSU
 - Posted this week
 - Date Posted
 - Newest first
 - Price
 - Low to High
 - High to Low
- Results:** Listings for Daly City House for rent. One listing is selected: 272, alameda Blvd, Daly City, CA 94018, 3beds/2ba, \$3000/mo, 1200 sq ft, 1.9 miles from SFSU.

Right Panel (Messages Dashboard):

- Rental Type:** Apartment, House
- Search Bar:** Search description or zip code
- Navigation:** Home, Post, About
- Email Address:** reguser@gatorrooms.com
- Filter Options:**
 - Description only
 - Near SFSU
 - Posted this week
 - Date Posted
 - Newest first
 - Price
 - Low to High
 - High to Low
- Messages:** 3 messages
 - From: Diana 07/01/18 Great to see... Daly City
 - From: George 02/03/18 Thank you for ... San Francisco
 - From: James 03/05/18 Hello there, I ... San Jose

Nancy clicks on her email address at the top line to access her account.

She sees a list of messages dashboard between her and other landlords she has contacted.

After Diana the landlord logs in, she can view the messages from Renters or from Admin.

The diagram shows Diana viewing messages for a specific listing. She clicks on a message from a renter, which is highlighted in a red box. A blue arrow points from this click to the right side of the diagram.

Left Panel (Messages for Listing):

- Rental Type:** Apartment, House
- Search Bar:** Daly.city.Ca
- Navigation:** Home, Post, About
- Email Address:** reguser@gatorrooms.com
- Filter Options:**
 - Description only
 - Near SFSU
 - Posted this week
 - Date Posted
 - Newest first
 - Price
 - Low to High
 - High to Low
- Messages:** 3 messages
 - From: Diana 07/01/18 Great to see... Renter
 - From: Admin 02/03/18 Thank you for ... Approved
 - From: Admin 03/05/18 Hello there, I ... Rejected

Right Panel (Message Thread for Listing):

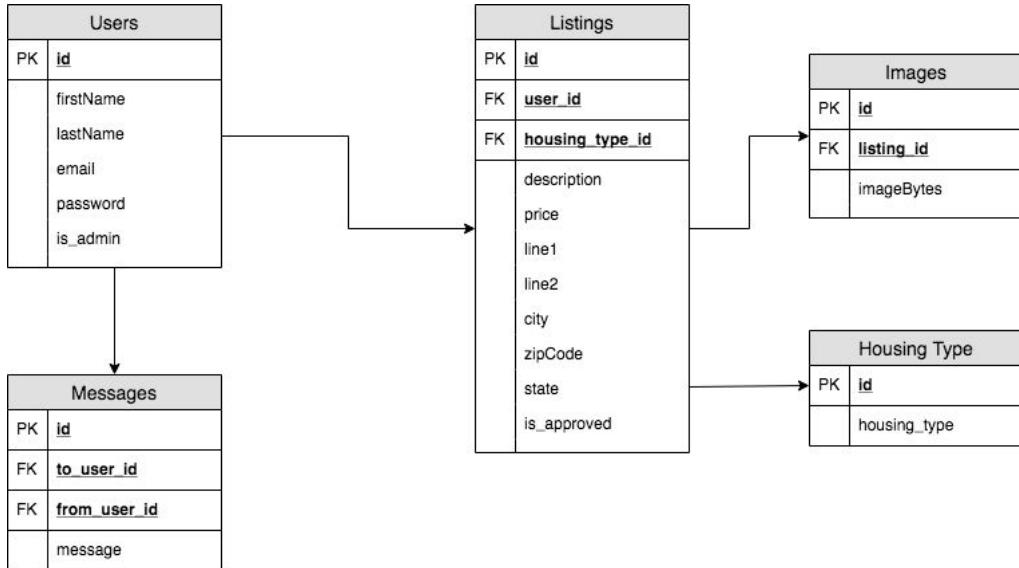
- Rental Type:** Apartment, House
- Search Bar:** Daly.city.Ca
- Navigation:** Home, Post, About, Login/Reg
- Listing Details:** 1461 Duck Creek Road, Daly City, CA 94015, 1 bed 1 bath 800 sqft \$1500/mo, Map View
- Messages:**
 - 7/15/18 From Nancy: I'm interested in the room. Here's my details, link here. Contact me.
 - 7/17/18 From Diana: Great to talk with you. You got it. Will send details through gmail.
 - 7/19/18 From Nancy: Awesome.

Diana clicks on the message from the Renter.

She sees a historical view of the messages between her and the renter, Nancy, tied to the listing they are discussing.

4. High Level Architecture, Database Organization

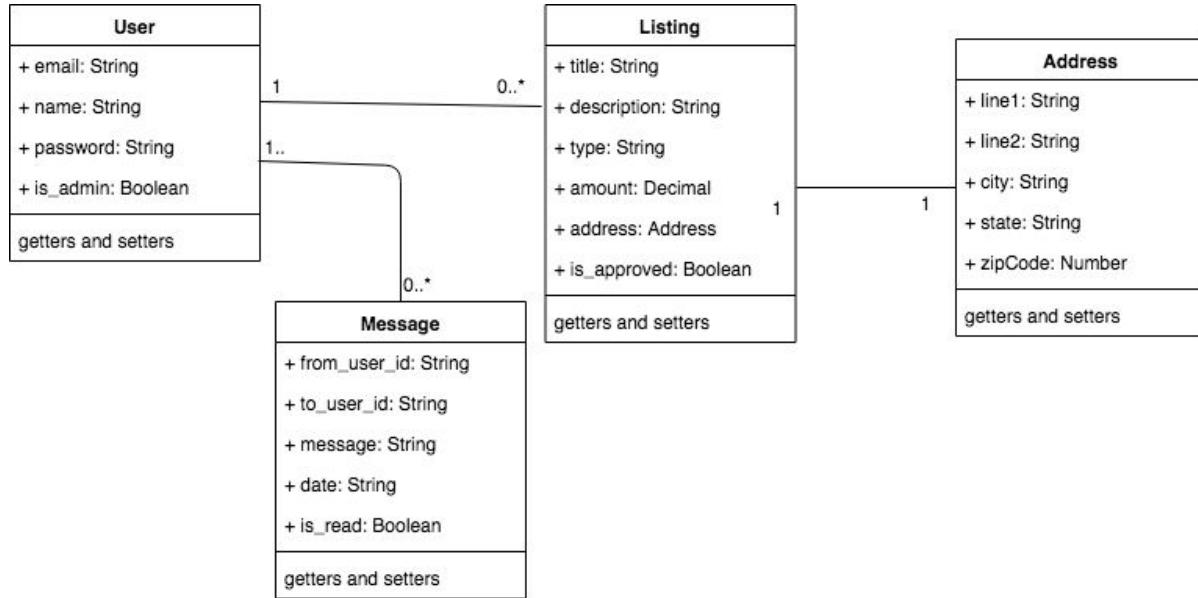
- Database Organization
 - There will be several tables that will store the users, chats, messages, and listing information.



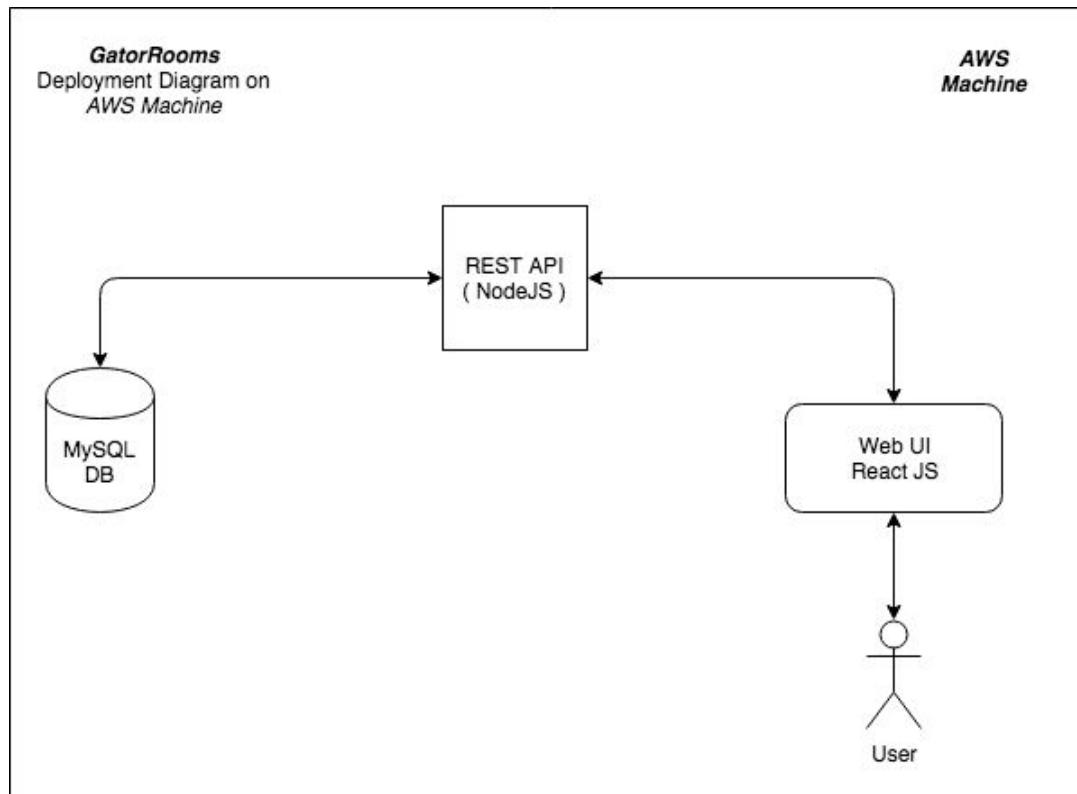
- Media Organization
 - The listing images will be stored in the database using BLOB (Binary Large Object) in the high resolution format, the client will resize the image based on its needs.
- Search / Filter architecture and implementation
 - A client will be sending the keywords entered by the user and their type (All, Zip code, Name Address, City, Name, and Description) based on the type a SQL query will be picked and the results will be returned if there is a match in the Database, “DB”.
 - For ‘All’ type, all the query will be executed. For name, city, description and address, SQL %like will be used, if no response than no listings will returned.
- APIs
 - Application will be divided into two parts, front-end and back-end.
 - Back-End will expose REST-API that will be used for communication between client and back-end application
 - App will be using Google Maps API to show locations on the maps and Google Analytics
- Listing Sorting
 - DB will be used to sort the data and return it to the client.
 - Listing will default and sorted by date it was posted, following approval by admin.
- Tools/Frameworks
 - No Changes

5. High Level UML Diagrams (Alex)

Class Diagram



Deployment Diagram



6. Risks (Inez & Anyone else)

1. Skills risks

- Team is ramping up on learning React. In order to efficiently work through the pages in the later milestones, the team will have to continue self-study every week, and follow along even though tasks of learning React, are not assigned.
- We will be reviewing React tutorials, sharing documentation/tutorial with each other to help us get practice on creating pages as personal training.

2. Schedule risks

- Team has one 30-minute time slot and majority of time works remote. We have used Slack and Google Drive for M1. We are starting to use Trello heavily to assign tasks that lead up to bigger tasks/to-dos in Trello

3. Technical risks

- We are starting to think about the pages for the prototype and how the data will be structured. Need to test and ensure the page displays the correct list of data, as organized the data tables.
- Keep design simple (e.g. no user roles, no char) and P1 list to a minimum

4. Teamwork risks

- Need to touch base with the team periodically to make sure the level of work is manageable and equally shared across individual team members.

5. Legal/Content risks

- We are going to ensure we use royalty free images, and fictional listing, therefore we won't need to create legal/content risk.

7. Project Management

1. After class:
 - a. We discuss any changes since the last meeting/milestone.
 - b. We discuss dividing up tasks, with due dates, and record them using a tool like Trello.
 - c. Answer any questions from the previous milestone.
2. Between class / team meetings:
 - a. We ask/answer questions on Slack, assign tasks on Trello, and set up Zoom meetings whenever necessary.
 - b. Team lead set up templates and docs on Google Docs & Google Slides so that everyone can collaborate on one tool.
 - c. We create tasks in Trello with our names as the Title, assign to ourselves or teammate, and provide due date when applicable
3. Before class:
 - a. We meet for 30 min to whiteboard, troubleshoot, answer questions
 - b. Walk through any questions that require more collaboration
 - c. Set up next steps/to-dos



Final Project

for SW Engineering

Class CSC 648-848 Spring 2019

Team 9 Milestone 3

GatorRooms.com

Inez Wibowo

Team Lead & Document
Editor

iwibowo@mail.sfsu.edu

Aliaksei Siarheyeu

Front End & Back End Lead
Github Master

Marcus Wong

Front End & Back End
Lead

Romeel Chaudhari
Back End Developer

Ismael San Juan
Front End Developer

Hang Li
Front End Developer

Jia Nan Mai
Front End Developer

1. Summary of Feedback

For Milestone 3 we presented our website for Product Review and received feedback to improve the way features are presented on the site so that it is easier for any user, without having to log in, to enjoy most of the site. Usability issues such as appearances of buttons, icons, and general visibility, at level 1 of the site, meaning the level for any user, without having to log in, was good to surface at the time and gave our team a list of tasks to fix. Additionally, there was also feedback regarding commenting on code, adding more descriptive comments on github, that would overall improve the sustainability of the site for other engineers managing the site.

2. Our Plan

Our plan following the feedback was to continue to build out P1 of the website, while making fixes to the changes that were highlighted during the UI review session. The team had built several pages by the time of the Product Review feedback and we incorporated feedback given while still balancing continued work on P1 items.

3. Fixes and Build

a. Fixes for Feedback from Product Review session completed and those at risk

Below is the list of fixes that were completed. For the final demo, there are items on this list that were not on our P1 list, such as Search button on nav at all times, or changing icon has different images, and posting as level one. We, were unable to deliver filtering by distance, 1 item that was listed on our P1.

Status	Page/Feedback Type	Feedback/Edits
Done	Github	Don't use feature/team member name as branch name, but just the feature as branch name
At Risk	GLOBAL comments	Search button to be on the nav at all times. The search must be always there(visible) through every page.
Done	GLOBAL comments	About Tab should be visible at all times
Done	Search (including search field validation)	In the search field, we enter zip code , address, or city like zillow (there's no way to validate zip), but search functionality is finished
Done	Search (including search field validation)	40 character field validation (he will test 0, 1, 25, 40 and 41 characters)
Done	Search Details and maps (if applicable)	Needs to open in a separate page, not occupy the same page (listing details with map)
Done	UI responsiveness (resize the browser)	Done
Done	Performance (e.g. display of results list)	Good

Done	Messaging/contact seller/user (if applicable)	When clicked the apartments need to open in a separate window, not occupy the same window
Done	Search Details and maps (if applicable)	Listings/Details page should have Maps link and Contact Landlord link
At Risk	GLOBAL comments	Instead of "i" icon we can have a logo
Done	Google Analytics	Add google analytics
Done	Google Analytics	Create screenshot of Analytics dashboard
Done	Github	Comments on Github can be more specific, ex. made this code change on this file, so that blabla
Done	Coding style and documentation	Coding Style added to the Github repo
Done	Coding style and documentation	Documentation, exists for tools, setting up database, react tutorial , and aws
Done	Screenshot of Trello	Screenshot of Trello
Done	GLOBAL comments	Add a welcome message. All pages should have "Welcome to GatorRooms.com. A place where SFSU students rent apartments" Each page must have a title
Done	Dashboards (user, admin)	Admin dashboards need to Approve , Reject, Delete a posting and Delete a User.
At Risk	GLOBAL comments	Have post regardless of logging in or not . Posting button should be visible on nav at all times and anyone start posting without logging in
At Risk	GLOBAL comments	When they hit submit on the posting form at that time they will be prompted to login
Done	Search results	Looks good, need apartment images, make sure they are royalty free
Partial	Filtering	Add Distance to SFSU , and Price filters (we have price filter)
At Risk	Filtering	Icon with the text to replace image sorting feature on the right of search field, instead of icons
Done	Messaging/contact seller/user (if applicable)	Messaging page should be clear which apartments are being discussed
Done	Messaging/contact seller/user (if applicable)	Show messages with dates
Done	Data Upload	Use the Listing Form
At Risk	Data Upload	Posting should at level one but not at level four.
Done	Listing Form	Create new listing form needs to have housing type dropdowns higher
Done	Listing Form	Upload image button needs to be bigger
Done	Listing Form	Form width overall needs to be more narrow
Done	Listing Form	Only one image to be uploaded
Done	Listing Form	Submit button should be farther to the right
At risk	Listing Form	Put a red asterisk or flag fields somehow to be clear what is mandatory , see mock ups

Done	Listing Form	Next to the submit button have a text that says “Postings may take up to 24 hours for publishing”
Done	Screenshot of DB tables	Presented at Product Review
Done	GLOBAL comments	Don't hide Login and Register, expand the buttons
Done	Login/Registration	Password needs to be encrypted in the database, not clear text
Done	Login/Registration	When user enters password, it should be masked, not visible to the user
Done	Login/Registration	Add captcha (Just added a static image with the code)
Done	GLOBAL comments	Smaller buttons(maybe?) Make buttons look like buttons

b.P1 items we delivered and one at risk

All major committed functions this team will deliver (P1):

User - Guest User

- ✓ User shall be able to browse the list.
- ✓ User shall be able to sign up and register.
- ✓ User shall be able to sort the list ascending, descending date of posting and monthly rent price.
- ✓ User shall be able to click display the property's larger version of the image.
- ✓ User shall be able to review the listings quickly whether they are on mobile or desktop.
- ! User shall be able to sort based on distance to the campus.
- ✓ User shall be able to review the About page and read team's profile on mobile and desktop easily.

User - Registered User

- ✓ Users shall be able to login to their accounts.
- ✓ Users shall be able to post after the post have been approved by admin.
- ✓ Users / Landlords shall be able to publish multimedia like images so that students can view it.
- ✓ Users/ Renters shall be able to send a message from their dashboard to the landlord/listing owner.
- ✓ Users / Landlord shall have a dashboard allowing them to review messages from user and contact the student.

User - Admin

1. Admins shall be able to login as admin.
2. Admins shall be able to review, remove, approve listings before they go live.
3. Admins shall be able to reject but not modify the listings from customers.

5. Original feedback from UI session

1. UI and functionality feedback (P1 functions only)

Home page

GLOBAL comments

Have post regardless of logging in or not . Posting button should be visible on nav at all times and anyone start posting without logging in

When they submit the posting at that time they will be prompted to login

Search button to be on the nav at all times. The search must be always there(visible) through every page. Instead of "i" icon we can have a logo

Don't hide Login and Register, expand the buttons

Add a welcome message. All pages should have "Welcome to GatorRooms.com. A place where SFSU students rent apartments" Each page must have a title

About Tab should be visible at all times

Smaller buttons(maybe?) Make buttons look like buttons

Search (including search field validation)

In the search field, we enter zip code , address, or city like zillow

Search results

Looks good, need apartment images, royalty free

Filtering

Add Distance to SFSU , and Price filters

Icon with the text to replace image sorting feature on the right of search field, instead of icons

Search Details and maps (if applicable)

Listings/Details page should have Maps link and Contact Landlord link

Needs to open in a separate page, not occupy the same page

Messaging/contact seller/user (if applicable)

Messaging page should be clear which apartments are being discussed

When clicked the apartments need to open in a separate window, not occupy the same window

Show messages with dates

Data Upload

Use the Listing Form

Posting should be at level one but not at level four.

Listing Form

Create new listing form needs to have housing type dropdowns higher

Upload image button needs to be bigger

Form width overall needs to be more narrow
Only one image to be uploaded
Submit button should be farther to the right
Put a red asterisk or flag fields somehow to be clear what is mandatory , see mock ups
Next to the submit button have a text that says “Postings may take up to 24 hours for publishing”

Dashboards (user, admin)
Admin dashboards need to Approve , Reject, Delete a posting and Delete a User.

Login/Registration
Password needs to be encrypted in the database, not clear text
When user enters password, it should be masked, not visible to the user

UI responsiveness (resize the browser)
Done

Performance (e.g. display of results list)
Good

Github
Don't use feature/team member name as branch name, but just the feature as branch name
Comments on Github can be more specific, ex. made this code change on this file, so that blabla

2. Brief review of coding, github, database etc.

Be ready to show examples of code so coding style and documentation can be checked. Same with github (file organization e.g. MVC, comments on code posting etc.)

Github
Don't use feature/user name, but just the feature
Comments on Github can be more specific, ex. made this code change on this file, so that blabla
Coding Style added to the Github repo
Documentation, exists for tools, setting up database, react tutorial , and aws

Also be ready to show DB tables and data in them using the tool of your choice (e.g. whatever your site admin would use). Stored PW will be checked for encryption.

Presented at Product Review

3. Project status – be ready to verbally explain status of the issues below

a) *Teamwork:* is the team working out, any related issues

Working good here , all team members are contributing. Those who have less Javascript/ReactJS familiarity will reach out for help and help with other aspects of the project.

- b) *Risks*: all actual (not hypothetical) risks (schedule, technical, skills etc.) should be identified and either resolved or plans made how to resolve them asap.

AWS port 80 not working. Need to troubleshoot.

- c) *Coding practices*: by this time you should have decided which coding style to use and actually use it, so be ready to report on this. Actual code will be verified

Coding style posted on Github

- d) *Usage of proper SE code management practices*: be ready to comment on how you manage code submissions, reviews, build and test etc. This will be verified by looking at actual code and server logs

We have multiple Leads as Github pull request reviewers to manage overall project development and code reviews.

- e) *How did you address site security and safe coding practices*: PW management, use of secure protocols, checking for valid inputs etc.

We have implemented form validation for the new listing submission form , listing and image admin approval will later be integrated to provide security. We also plan to implement session tokens per logged in user to manage user access and authentication.

- f) *Digital content* (e.g. images, video) please provide information on status and availability, chosen formats, resolution etc.

Images are in the SQL database

- g) Other

4. List of P1 features committed for delivery

First Priority (P1)

User - Guest User

P1-Guest-1. User shall be able to browse the list.

P1-Guest-2. User shall be able to sign up and register.

P1-Guest-3. User shall be able to sort the list ascending, descending date of posting and monthly rent price.

P1-Guest-4. User shall be able to click display the property's larger version of the image.

P1-Guest-5. User shall be able to review the listings quickly whether they are on mobile or

desktop.

P1-Guest-6. User shall be able to sort based on distance to the campus.

P1-Guest-7. User shall be able to review the About page and read team's profile on mobile and desktop easily.

User - Registered User

P1-Reg-1. Users shall be able to login to their accounts.

P1-Reg-2. Users shall be able to post after the post have been approved by admin.

P1-Reg-3. Users / Landlords shall be able to publish multimedia like images so that students can view it.

P1-Reg-4. Users/ Renters shall be able to send a message from their dashboard to the landlord/listing owner.

P1-Reg-5. Users / Landlord shall have a dashboard allowing them to review messages from user and contact the student.

User - Admin

P1-Admin-1. Admins shall be able to login as admin.

P1-Admin-2. Admins shall be able to review, remove, approve listings before they go live.

P1-Admin-3. Admins shall be able to reject but not modify the listings from customers.



SW Engineering CSC648/848

Spring 2019

GatorRooms.com

Team 9 Milestone 4

**QA and Usability Testing and
Final Commitment for Product Features**

Inez Wibowo

Team Lead & Document
Editor

iwibowo@mail.sfsu.edu

Aliaksei Siarheyeu

Front End & Back End Lead
Github Master

Marcus Wong

Front End & Back End Lead

Romeel Chaudhari
Back End Developer

Ismael San Juan
Front End Developer

Hang Li
Front End Developer

Jia Nan Mai
Front End Developer

Versions

History Table

Version	Date	Notes
ver 1.1	5/8/19	First submission Milestone 4
ver 1.2	5/19/19	Revised version after integrating prof feedback

1. Product Summary

The rental website GatorRooms.com is a website for San Francisco State University students. SFSU students will find it easier to find rentals on a website designed specifically with features important to students. GatorRooms therefore stands out by prioritizing sorting by price and by distance to campus. The website is also lightweight (support ~50 users at a time) and it's easy for students to find rentals on the go, i.e. on mobile. Registered users/landlords on GatorRooms.com can also post their available rentals easily and will be able to post only after the post has been verified by GatorRooms.com admin.

We hope students and registered users alike find GatorRooms a website indispensable and spread the availability of the site through word-of-mouth, through Slack channels, and posters across campus.

All major committed functions this team will deliver (P1):

User - Guest User

- User shall be able to browse the list.
- User shall be able to sign up and register.
- User shall be able to sort the list ascending, descending date of posting and monthly rent price.
- User shall be able to click display the property's larger version of the image.
- User shall be able to review the listings quickly whether they are on mobile or desktop.
- User shall be able to sort based on distance to the campus.
- User shall be able to review the About page and read team's profile on mobile and desktop easily.

User - Registered User

- Users shall be able to login to their accounts.
- Users shall be able to post after the post have been approved by admin.
- Users / Landlords shall be able to publish multimedia like images so that students can view it.
- Users/ Renters shall be able to send a message from their dashboard to the landlord/listing owner.
- Users / Landlord shall have a dashboard allowing them to review messages from user and contact the student.

User - Admin

- Admins shall be able to login as admin.
- Admins shall be able to review, remove, approve listings before they go live.
- Admins shall be able to reject but not modify the listings from customers.

URL is a temporary AWS instance, which will be provided on demo day, locally it is <http://localhost:3000>

2. Usability Test Plan

We are going to test the Posting function for usability (2.5 page max)

1. Test Objectives (about half a page)

We will test the Post function. When a landlord needs to start a new listing posting is one of the most important function for this rental web application. It offers the landlord a tool to share their property to students who are looking for a place to rent. On the posting page, there are 11 fields that allow landlords to enter detailed information about their property, such as Title, Price, Address, City, State, Zip, Description, and dropdown boxes to further categorize Housing Type, Bedrooms, Bathrooms. Upload buttons allow landlord to upload images of their property and at end of the page, there are two buttons, reset and submit. Reset button allows users to delete all fields that is inputted, submit button allows users to submit everything that is ready to post. We will check these tasks and record results of the test to see whether the software is easy to use. We will give the user a task and he/she will navigate the pages. We'll collect input and how user feels after doing this task by using a Likert scale survey.

2. Test Description (total up to 1 page)

a. System setup :

A user needs to launch a web browser on a laptop. The evaluator will be close by. At home of the tester.

b. Starting point:

The starting point of post function is the homepage.

c. Intended Users:

An older person, looking to post an apartment listing.

d. URL of the system to be tested:

<http://localhost:3000/home>

e. What is to be measured (User Satisfaction evaluation Likert Tests):

Methodology:

- First: test that available functions allow completion of main tasks/use cases
- Measure task completion: % of people who completed the defined task in defined time
- Count errors per task

What will be measured:

- We will measure the effectiveness, specifically: We'll test Post function and ensure it can be successfully completed by the user.
- We will measure the efficiency, we'll test that the posting task can be completed quickly and easily.

The following is a result of testing tasks for Effectiveness and Efficiency

Test/Use Case	% Completed <Effectiveness>	Errors <Effectiveness>	Average time spent/Clicks <Efficiency>	Comments
Find the post button	100	No	1s	submit button is located at bottom
Start a post	100	no	1min	works fine
Submit post	100	no	5s	works fine
Fill out the form	100	No	20s/1 click	works easily
Upload an image	100	No	3s	it is pretty easy

- Satisfaction: Testing satisfaction using Likert scale with simple questions from Qualtrics survey https://sfsu.co1.qualtrics.com/jfe/form/SV_bBOhvNIWzqqorsN

Below are qualitative results from Qualtrics Survey small sample size n=3, 73% satisfaction

Questions <Satisfaction>	Rating (1-5)
Overall, how satisfied were you with GatorRooms?	3
Based on your experience with GatorRooms, how likely are you to purchase their products or services again?	3
Based on your experience with GatorRooms, would you recommend their products or services to a friend or family member?	4
Average	3.6 (73%)

3. Usability Task description for Testers

These are the tasks the testers will conduct during testing:

- a. Tester will provide user a list of actions as shown above, i.e. "Find the post button", "Start a post", etc.

- b. Tester will record the Effectiveness, Efficiency, and Satisfaction metrics and record them in a table like the above
- c. Tester will deliberately leave it to the user to complete the task to be able to measure the intuitiveness of the UI
- d. Goal of the tester is to expose usability flaws and document them

4. Questionnaire , provided here are 3 general Likert scale question format to gain insight into user satisfaction

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Post function is easy to find.					
I have all the fields required to create a high quality listing.					
Posting is intuitive and easy to use.					
I would recommend GatorRooms to a friend.					
Comments:	<Free-form field>				

3. QA Test Plan

1. Test Objectives

Check if the software is according to specs.

We will give the user a task and he/she will navigate the pages.

We'll collect input on whether or not the experience is as written in the specs, PASS or FAIL

Make sure there aren't any bugs

Perform these test on Chrome and Safari:

Test Homepage search

Test Login

Test registration of new Listing Post

2. HW and SW setup (and the URLs)ng to post from home page

<https://localhost:3000/home>

<https://localhost:3000/new>

<https://localhost:3000/login>

Perform Tests on Safari and chrome installed in different systems, such as windows, mac, test data, test database, front end running environment

3. Features to be tested

Posting flow:

- a. Navigation to posting a listing
- b. Creating a listing
- c. Filling out the form
- d. Submitting the form for approval
- e. Review the listing as a registered user

4. Risk of data loss due to improper process implementation, failed system or some external events risk

5. QA Test Plan (include 3 test cases and results of testing them on your app about 1 page, use table from class. Suggested format for QA Test Plan Table: table columns are: test #; test title; test description; test input; expected correct output; test results (PASS or FAIL for each tested browser).

Test Number	Test Title (Unit/Smoke/	Test Description	Test Input	Expected Correct	Test Results (PASS/FAIL)
-------------	----------------------------	---------------------	------------	---------------------	-----------------------------

	Regression)			Output	
1	Unit	Testing to find posting on Homepage	Enter in search field zipcode “10000”	GET 3 results all have “10000” in zip field	PASS(chrome) PASS(safari)
2	unit	Search for listing post in home page	Enter in search field “listing”	Get 1 result with “Listing1” in the description	PASS(chrome) PASS(safari)
3	Smoke	Login User	Try login in with invalid credentials like “...” for username and password entered in the inputs	Get error username and password are invalid	PASS(chrome) PASS(safari)
4	Regression	Adding new listing	Add a new posting “new apartment” to database	Data “new apartment” inserted	PASS(chrome) PASS(safari)

4. Code Review (Alex & Marcus)

4a. Coding style we have chosen:

Our project follows the basic folder structure for any nodeJS web app.

<https://docs.npmjs.com/misc/coding-style.html>

Coding style

https://github.com/CSC-648-SFSU/csc648-sp19-team09/blob/master/Coding_Style.txt

Additionally to coding style and documentation, we have two main folders which divide the server side code and the client side code

A. Client side

Folder Structure

All React components are divided into their own folders which contain related styles or other components related

We have one front end api folder that contains an api for all the frontend endpoint calls. For our simple, api we categorize requests by their action type for example all user requests are contained in a user.actions.js file.

B. Server side

Folder Structure

Server side folder structure is very basic and simplified.

We are using an ORM to model our SQL database tables so we contain all our model definitions in the models folder.

We have one express router per model which handles all HTTP requests for the model. All these routes are contained in one routes folder.

Naming

Models files are named with lowercase and dashes substituting spaces

Model variables are named following Java's class naming convention, that is they are using UpperCamelCase. lowerCamelCase is used for all other variables.

C. Code Clean up

Code clean up is done periodically and typically before or during pull requests. We utilize the

Javascript ES6 syntax, for example to reduce code amount.

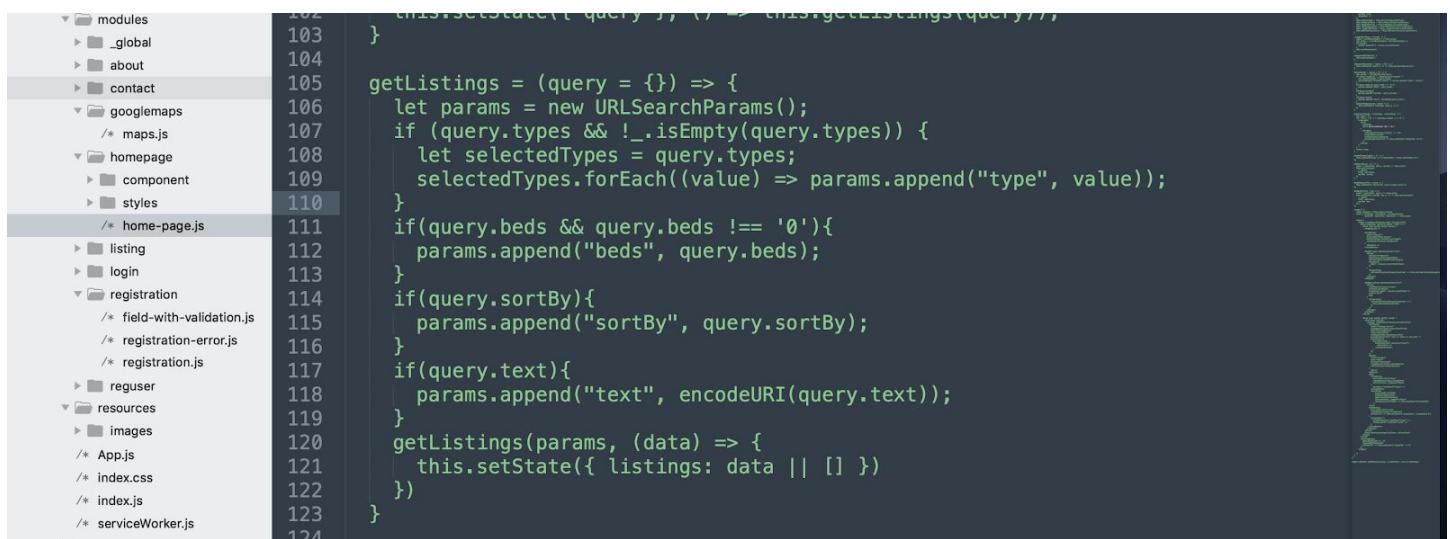
https://www.w3schools.com/js/js_es6.asp

4b. Here is an example of the code under review : Github and Code Review

Overview

- Code review is managed through github pull requests.
 - Team members are expected to submit working tested code before making a pull request.
 - Since all new code is integrated with pull request, we have the ability to do code review on all code.
 - Conflicting code will also be resolved during pull request.
 - All reviews are reviewed by our main lead software engineer Alex.
 - As part of review, the code reviewers Alex and Marcus also check for header and in-line comments to make sure they are descriptive.
-
- Code Review by Marcus

Reviewing Front End HomePage component



```
101     this.setState({ query }, () => this.getListings(query));
102 }
103
104
105 getListing = (query = {}) => {
106   let params = new URLSearchParams();
107   if (query.types && !_.isEmpty(query.types)) {
108     let selectedTypes = query.types;
109     selectedTypes.forEach((value) => params.append("type", value));
110   }
111   if(query.beds && query.beds !== '0'){
112     params.append("beds", query.beds);
113   }
114   if(query.sortBy){
115     params.append("sortBy", query.sortBy);
116   }
117   if(query.text){
118     params.append("text", encodeURI(query.text));
119   }
120   getListing(params, (data) => {
121     this.setState({ listings: data || [] });
122   })
123 }
124 }
```

The HomePage class is getting too large, and it will get larger if we have more searching mechanisms. But to clean up this code a bit and make our project more maintainable, we could implement an OOP design pattern to reduce some of the responsibility of the HomePage class.

For example in the code snippet above, we could use the Strategy Design Pattern to build Query classes. Therefore building the queries would not be the HomePage's responsibility.

- Code Review by Alex Reviewing Listing/Posting functionality

Review below includes pull request/code review requested by Marcus to Alex, through Github.

Listings/post #24

Edit

Merged alexsergeev merged 9 commits into master from listings/post 19 days ago

Conversation 2 Commits 9 Checks 0 Files changed 9 +315 -126

GandalfGrey123 commented 21 days ago

post new listing form with form validation done.

GandalfGrey123 and others added some commits on Apr 8

- Update package.json 1c695e5
- Delete 20190323013838-fix-keys.js 1aa1201
- make drawer responsive, clean up grid, added search textfield a9d7067
- Merge branch 'marcus-test' of https://github.com/CSC-648-SFSU/csc648-... 2160164
- ...
_sp19-team09 into marcus-test
- Added more improvements to Home Page Grids aaifd79
- Update listing-card.js d0e31c6
- added image previewer and editor 5b4e070
- Update Home page to include a switch between the column and list ... 44aaafa5
- ...
_view
- finished validation 18e065f

GandalfGrey123 requested a review from alexsergeev 21 days ago

GandalfGrey123 closed this 21 days ago

GandalfGrey123 reopened this 21 days ago

GandalfGrey123 commented 21 days ago

I think this is ready to merge, I can't tell if the grid changes you added are in here.

alexsergeev approved these changes 19 days ago

alexsergeev left a comment

Changes look good! Think can merge it.

alexsergeev merged commit f8a701a into master 19 days ago

Reviewers alexsergeev ✓

Assignees No one—assign yourself

Labels None yet

Projects None yet

Milestone No milestone

Notifications

Unsubscribe You're receiving notifications because you're watching this repository.

2 participants

Lock conversation

5. Self-check on best practices for Security (½ page Romeel)

Some user data assets we are protecting are:-

1. User Information

The user information includes email-id and the password used during the login and registration is encrypted in the database. In this way, the user information cannot be obtained in a malicious manner.

2. Passwords

To protect users identity, we are encrypting masking the password entry. We further encrypt the password in the database as described below.

3. Image validation

Whether the image of a listing is declined/approved/pending. The approval of the image will depend on the admin. It is same as admin approval of the listing. If the image is approved it can be posted. If declined or pending it will be held back with the listing.

4. Encryption of Passwords (PW) in the Database

Passwords in the database are not clear text. We are hashing the combination of the username and password to encapsulate the password.

5. Data validation

Data validation – In our application, we are validating the search bar input based on the city, zip, state. We are also encrypting user Login and registration information. Thetextfield data validation is also done during the login and registration process.

6. Self-check Adherence to original Non-functional specs (Jia nan)

6.1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO).	DONE
6.2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers	DONE
6.3. Selected application functions must render well on mobile devices	DONE
6.4. Data shall be stored in the team's chosen database technology on the team's deployment server.	DONE
6.5. No more than 50 concurrent users shall be accessing the application at any time	DONE
6.6. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.	DONE
6.7 The language used shall be English.	DONE
6.8. Application shall be very easy to use and intuitive.	DONE
6.9. Google analytics shall be added.	DONE
6.10. No e-mail clients shall be allowed.	DONE
6.11. Pay functionality, if any (e.g. paying for goods and services) shall not be	Not Applicable

implemented nor simulated.	
6.12. Site security: basic best practices shall be applied (as covered in the class).	DONE
6.13. Before posted live, all content (e.g. apartment listings and images) must be approved by site administrator	DONE
6.14. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development.	DONE
6.15. The website shall <u>prominently</u> display the following <u>exact</u> text on all pages <i>"SFSU Software Engineering Project CSC 648-848, Spring 2019. For Demonstration Only"</i> at the top of the WWW page. (Important so as to not confuse this with a real application).	DONE

3. Screen shots of actual final product as shown in the demo, one per page

Main Homepage

Displaying all apartments

The screenshot shows a web browser window for the 'SFSU - CSC 648 Team #9 Project' at localhost:3000. The page title is 'Welcome to GaterRooms'. On the left, there are two filter sections: 'Housing Types' (with 'All' checked) and 'Beds' (with '0+' checked). The main content area displays three apartment listings with images and details:

- Frodos Apartment**: May 20. This is a 2 bedroom apartment for rent, pets are allowed.
- Townhome in Rohan**: May 20. Beautiful Town house with horse stables in the front.
- Shire cottage**: May 20. Nice cozy cottage in the shire , nice view of the village.

A fourth listing is partially visible below the others.

Main Homepage - Housing Type

Displaying Housing Type filters for Apartment and Room

The screenshot shows the main homepage for GaterRooms. On the left, there is a sidebar with filtering options for 'Housing Types' (All, Apartment, Room), 'Beds' (0+, 1+, 2+), and 'Listing Image'. The 'Room' option is selected. The main content area displays two listing cards: 'Frodos Apartment' and 'Dead marsh rock studio'. Both cards show small thumbnail images, the listing name, a date (May 20), a brief description, and a note about pets.

Main Homepage - Registered User Menu

Displaying Profile, My Listings, New Listings, Log out.

The screenshot shows the main homepage for GaterRooms with a user menu open on the right side. The menu includes options for 'Profile', 'My Listings', 'New Listing', and 'Log Out'. The sidebar on the left is identical to the previous screenshot, showing housing type and bed filters. The main content area displays three listing cards: 'Frodos Apartment', 'Townhome in Rohan', and 'Shire cottage'. Each card includes a thumbnail image, the listing name, a date (May 20), a brief description, and a note about pets or features like horse stables.

Main Homepage - Search filter

Displaying Newest, Cheapest, Bedrooms, and SFSU Distance.

The screenshot shows the GaterRooms homepage. On the left, there are two vertical filter panels. The first panel, 'Housing Types', has three options: 'All' (selected with a checked checkbox), 'Apartment' (unchecked), and 'Room' (unchecked). The second panel, 'Beds', has three radio button options: '0+' (selected with a red outline), '1+' (unchecked), and '2+' (unchecked). In the center, there's a search bar with placeholder text 'Enter city or state or zip code'. To the right of the search bar is a dropdown menu with four options: 'Newest' (selected with a hand cursor icon), 'Cheapest', 'Bedrooms', and 'SFSU Distance'. Below the search bar are four listing cards. The first card, 'Frodos Apartment', shows an exterior photo of a building and a brief description: 'May 20 This is a 2 bedroom apartment for rent , pets are allowed'. The second card, 'Townhome in Rohan', shows an interior photo of a living room and a brief description: 'May 20 Beautiful Town house with horse stables in the front'. The third card, 'Shire cottage', shows an exterior photo of a house and a brief description: 'May 20 Nice cozy cottage in the shire , nice view of the village'. The fourth card, 'Dead marsh rock studio', shows an exterior photo of a modern building and a brief description: 'Dead marsh rock studio'. A blue button labeled 'NEW LISTING +' is located at the top right of the main content area.

Login Page

Displaying Login and Masked Password input field
“Remember me” checkbox, “Forgot Password?” link
And, register link “Don’t have an account?”

The screenshot shows the login page. At the top, there's a blue header bar with a help icon and the text 'SFSU - CSC 648 Team #9 Project'. Below the header is a horizontal ellipsis (...). The main form area has several input fields: 'Email' with value 'frodo@yahoo.com', 'Password' with a masked value '...', and a 'Remember Me' checkbox which is unchecked. Below the form are two links: 'Forgot password ?' and 'Don't have an account?'. At the bottom is a large blue 'LOG IN' button.

Registration Page

Displaying password masked

Captcha

Checkbox to indicate admin user

SFSU - CSC 648 Team #9 Project

First Name
Gollum

Last Name
Smeagol

Email
gollum@yahoo.com

Create a password
••••

Enter password again
••••

Enter Image Code
••••

6 ZKWC

Admin

REGISTER

Registration->Login Page - Account successfully created

Success message

Take user to Login Page

SFSU - CSC 648 Team #9 Project

✓ Account successfully created. Please login.

Email*

Password*

Remember Me

[Forgot password ?](#)

[Don't have an account?](#)

LOG IN

Profile Page

Displaying password masked

Displaying other pages available for the user, My Listings, Messages, and Admin if the registered user is an Admin

Screenshot of the Profile Page showing a form for updating user information. The page has a blue header with tabs: PROFILE (which is active), MY LISTINGS, MESSAGES, and ADMIN. Below the header is a section titled "Profile Info". The form includes fields for AvatarUrl (with value https://cdn1.iconfinder.com/data/icons/user), First Name (Frodo), Last Name (Baggins), Email (frodo@yahoo.com), and Password (*****). A blue "UPDATE" button is at the bottom.

AvatarUrl	https://cdn1.iconfinder.com/data/icons/user
First Name	Frodo
Last Name	Baggins
Email	frodo@yahoo.com
Password	*****

UPDATE

Profile Page - Updated

The user can choose to change their profile

Screenshot of the Profile Page showing a success message: "profile has been updated!" in a white box. The rest of the page is dimmed, indicating it's a modal or a partially active screen. The "Close" button is visible in the bottom right corner of the message box.

profile has been updated!

Close

Details Page

The user can view the listing

Contact Landlord button

And a map to check the distance to SFSU

The screenshot shows a listing for a three-story apartment building at 2115 14th Ave, San Francisco CA 94122. The listing includes a price of \$4000, 2 bedrooms, and 1 bathroom. It was posted on May 20. A blue "CONTACT LANDLORD" button is visible. Below the listing is a Google map of the surrounding neighborhood, including the T District, Inner Parkside, Forest Hill, and West Portal. The map shows the location of the apartment building and nearby landmarks like the Golden Gate Heights and the 16th Avenue Tiled Steps.

Details Page - Contact Landlord button

Contacting landlord module

The screenshot shows the same listing as above, but the "CONTACT LANDLORD" button is now open, displaying a modal window titled "Contact Landlord". The modal contains a text input field with the placeholder "Please enter a message to send to landlord. The message history is going to appear in your profile inbox." and a message box containing "Hello this is Frodo I would like to rent this apartment!". At the bottom of the modal are "CANCEL" and "SEND MESSAGE" buttons. The background of the page is dimmed, and the map from the previous screenshot is partially visible behind the modal.

Messaging Page - Contacting Landlord

Contacting landlord via message page on the site

The screenshot shows a messaging interface. At the top, there are tabs for PROFILE, MY LISTINGS, MESSAGES (which is underlined), and ADMIN. Below the tabs, a sidebar on the left lists "Direct Messages" and shows a message from "gandalf@yahoo.com" about a listing titled "Frodos Apartment". The main area displays a conversation between "gandalf@yahoo.com" and "frodo@yahoo.com". The messages are:

- gandalf@yahoo.com sent - May 20th 2019, 9:56:54 am: Hey Frodo! this is Gandalf , nice apartment but mine is better
- frodo@yahoo.com sent - May 20th 2019, 9:57:44 am: hello Gandalf, nice try but well see who gets the last laugh!

At the bottom, there is a text input field labeled "Type message" and a blue "SEND" button.

Listing Info Page - Create listings

The user can create a new listing

User is notified that it will 24 hours

Can provide categorization : Housing type, Bedrooms, Bathrooms

Can upload multiple images at once

The screenshot shows a form for creating a new listing. At the top, there are tabs for PROFILE, MY LISTINGS (which is underlined), and MESSAGES. The main form area is titled "Create New Listing" and contains the following fields:

- Title: A text input field with a placeholder "Title".
- Price: A text input field with a placeholder "Price".
- Address: A text input field with a placeholder "Address".
- City: A text input field with a placeholder "City".
- State: A text input field with a placeholder "State".
- Zip: A text input field with a placeholder "Zip".
- Description: A large text input field with a placeholder "Description".

Housing Type
Townhome

Bed rooms
3

Bathrooms
3

+ UPLOAD IMAGES

SUBMIT

Listings pay take up to 24 hours to process

Admin Dashboard - Approve/Reject

Provide ability to approve or reject the listing

Listing Image

PROFILE MY LISTINGS MESSAGES ADMIN

Admin Types

- All
- Approved
- Not Approved

Admin	
	Frodos Apartment May 20 This is a 2 bedroom apartment for rent , pets are allowed <small>Approved</small> APPROVE REJECT
	Townhome in Rohan May 20 Beautiful Town house with horse stables in the front <small>Approved</small> APPROVE REJECT
	Shire cottage May 20 Nice cozy cottage in the shire , nice view of the village <small>Approved</small> APPROVE REJECT
	Dead marsh rock studio May 20 Enjoy nice cool and cold atmospheres? Don't want to be bothered by people? Like ghostly towns? we have a nice little cozy rock hut right in the middle of the

Admin Dashboard - Approving a listing

Status changes on the listing “Eye of morder”

- Approved
- Not Approved



Frodos Apartment

May 20
This is a 2 bedroom apartment for rent , pets are allowed

Approved
APPROVE REJECT



Townhome in Rohan

May 20
Beautiful Town house with horse stables in the front

Approved

APPROVE REJECT



Shire cottage

May 20
Nice cozy cottage in the shire , nice view of the village

Approved
APPROVE REJECT



Dead marsh rock studio

May 20
Enjoy nice cool and cold atmospheres? Don't want to be bothered by people? Like ghostly towns? we have a nice little cozy rock hut right in the middle of

Approved
APPROVE REJECT



Eye of morder

May 20
Live in the tower of sauron

Approved

APPROVE REJECT

Admin Dashboard - Newly approved listing

New listing “Eye of morder” approved and shows in the results

- All
- Apartment
- Room
- Beds
- 0+
- 1+
- 2+

SEARCH

≡

NEW LISTING +

Frodos Apartment

May 20
This is a 2 bedroom apartment for rent , pets are allowed

Townhome in Rohan

May 20
Beautiful Town house with horse stables in the front

Shire cottage

May 20
Nice cozy cottage in the shire , nice view of the village

Dead marsh rock studio

May 20
Enjoy nice cool and cold atmospheres? Don't want to be bothered by people? Like ghostly towns? we have a nice little cozy

Eye of morder

May 20
Live in the tower of sauron

4. Screen shots of key DB tables (1-2 pages)

User Table

Displaying 3 registered users

There are 7 data fields / columns that the User table holds.

The screenshot shows the MySQL Workbench interface with the User table selected. The table has 7 columns: id, email, avatarUrl, password, firstName, lastName, isAdmin, and sessionToken. The data for three users is displayed:

id	email	avatarUrl	password	firstName	lastName	isAdmin	sessionToken
1	frodo@yahoo.com	https://cdn1.iconfinder.com/data/icons/user-pict...	\$2b\$10\$olq2Kg1x/NN4KZFWxE67ePSNvcW8...	Frodo	Baggins	1	dy77nvu96swtwa
2	gandalf@yahoo.com	https://cdn1.iconfinder.com/data/icons/user-pict...	\$2b\$10\$bLsLFBFHz64ChUV14rHoOBaHd7Nf1...	Gandalf	Grey	1	1659efw8q8hnqsk
3	samwise@yahoo.com	https://cdn1.iconfinder.com/data/icons/user-pict...	\$2b\$10\$pbdk7cCij3raWtAkgwfJOdly/kjvdfnds9...	samwise	gamgee	0	43gb7suv24rlb6
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Housing Types Table

Displaying static data table for available housing types

The screenshot shows the MySQL Workbench interface with the HousingTypes table selected. The table has 2 columns: id and type. The data for five housing types is displayed:

id	type
1	APARTMENT
2	ROOM
3	HOUSE
4	STUDIO
5	TOWNHOME
	NULL

Listing Posts Table

Displaying all listing posts created by the registered users

The data fields / columns hold information about the listing as well as the user and housing type it belongs to

Result Grid 1 (Listings):

id	title	description	price	line1	line2	city	state	zipCode	bedrooms	bathrooms	isApproved	distance	datePosted
1	Frodos Apartment	This is a 2 bedroom apartment for rent , pets ar...	4000	2115 14th Ave		San Francisco	CA	94122	2	1	1	NULL	2019-05-20 16:47:07
2	Townhouse in Rohan	Beautiful Town house with horse stables in the f...	3000	219 Diamond Peak Rd		Diamond Peak	SouthLand	123213	2	2	1	NULL	2019-05-20 16:49:44
3	Shire cottage	Nice cozy cottage in the shire , nice view of the...	300	12 bywater rd		middle earth	the Shire	0	1	1	1	NULL	2019-05-20 16:52:50
4	Dead marsh rock studio	Enjoy nice cool and cold atmospheres? Don't w...	666	dead marshes		wetlands	middle earth	666	1	1	1	NULL	2019-05-20 16:55:34
HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

Result Grid 2 (Housing Types):

city	state	zipCode	bedrooms	bathrooms	isApproved	distance	datePosted	HousingTypeId	UserId
San Francisco	CA	94122	2	1	1	NULL	2019-05-20 16:47:07	1	1
Diamond Peak	SouthLand	123213	2	2	1	NULL	2019-05-20 16:49:44	5	1
middle earth	the Shire	0	1	1	1	NULL	2019-05-20 16:52:50	4	2
wetlands	middle earth	666	1	1	1	NULL	2019-05-20 16:55:34	2	2
HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

Chats and Messages

Displaying all messages that users have made, and the chats they belong to

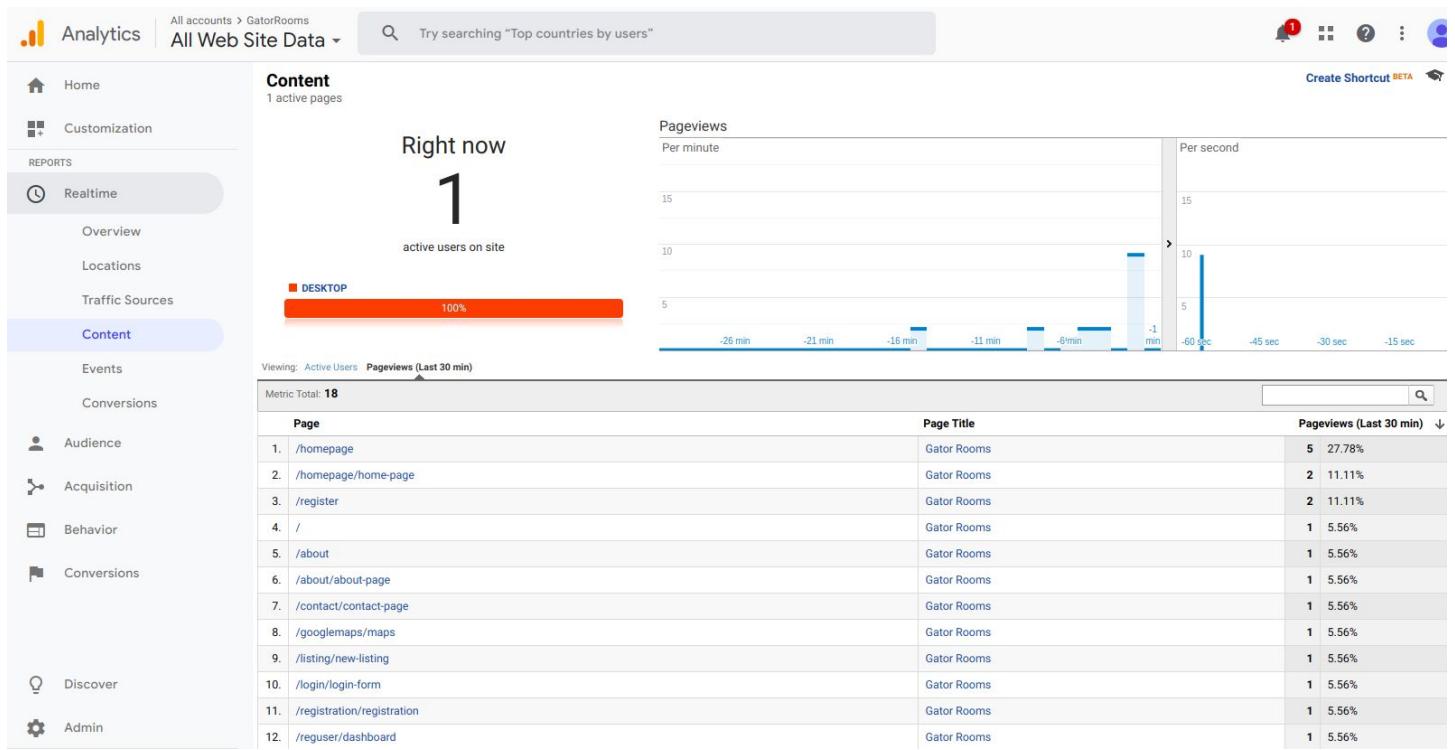
Result Grid 1 (Chats):

id	userEmail	message	dateSent	ChatId	UserId
1	gandalf@yahoo.com	Hey Frodo! this is Gandalf , nice apartment but...	2019-05-20 16:56:54	1	2
2	frodo@yahoo.com	hello Gandalf, nice try but well see who gets the...	2019-05-20 16:57:44	1	1
3	samwise@yahoo.com	hey my name is Samwise Gamgee , I'm looking...	2019-05-20 16:58:54	2	3
4	samwise@yahoo.com	most people say I am very brave	2019-05-20 16:59:09	2	3
5	gandalf@yahoo.com	hey sam thanks for contacting me, I would love t...	2019-05-20 16:59:33	2	2
HULL	HULL	HULL	HULL	HULL	HULL

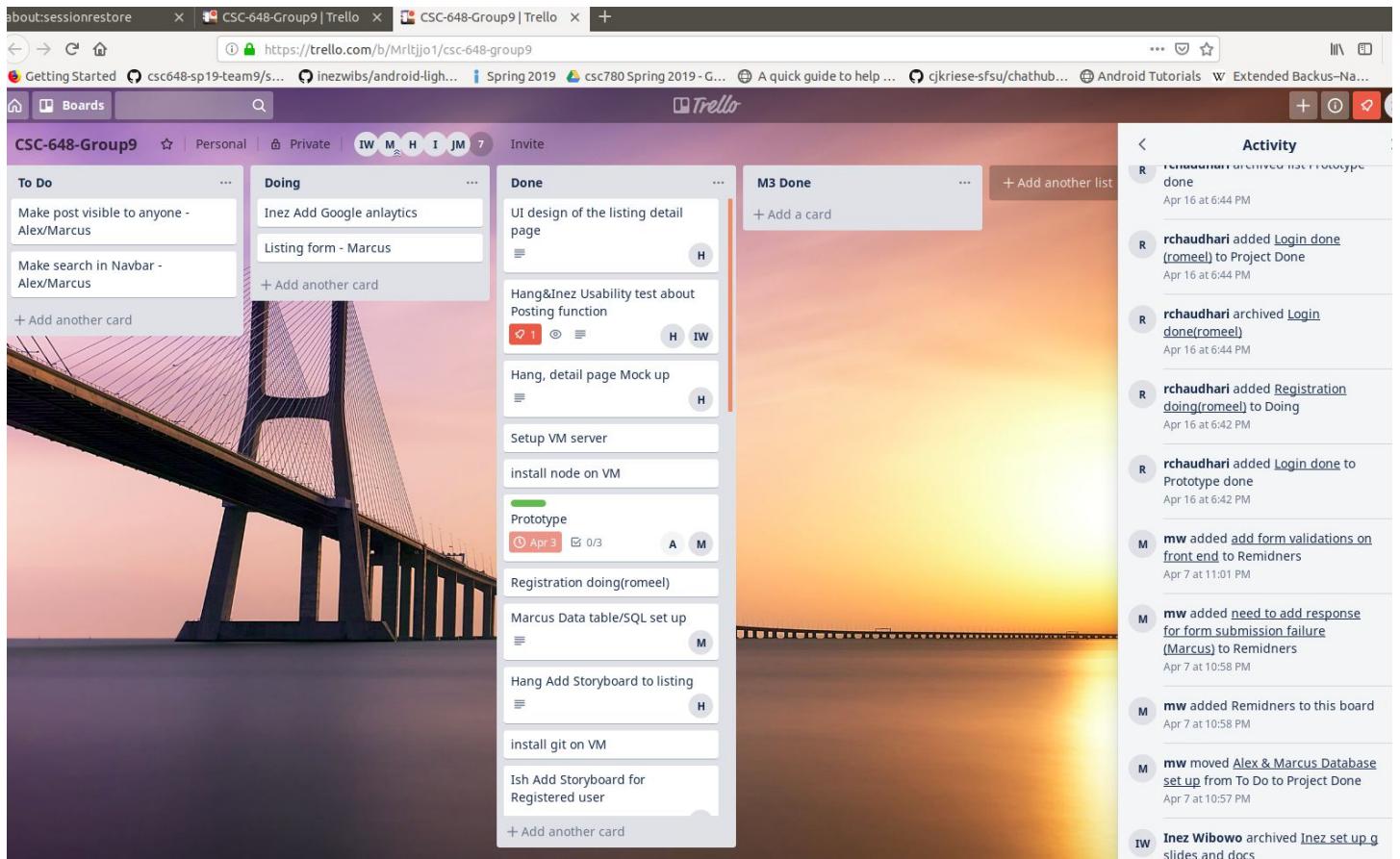
Result Grid 2 (Messages):

id	dateSent	ListingPostId	lesseeChatFk	landLordChatFk
1	2019-05-20 16:56:54	1	2	1
2	2019-05-20 16:58:54	3	3	2
HULL	HULL	HULL	HULL	HULL

5. Google analytics plot for your WWW site (1 page)



6. Screen shot(s) of your task management system (like Trello) showing a snapshot of your project management



7. Emailed Team Member Contributions

Attached

8. Post analysis – lessons learned (one page or so)

Main challenges

Knowledge risk: At the start of this project, we already noticed and highlighted a knowledge risk, which is that most of the people on the team do not have experience with ReactJs, and have little experience with JavaScript, the language and framework we've chosen. We chose this because we believed that it would give us the ability to create a nice, responsive, website on desktop and mobile, in the time constraints.

Scheduling risk: Additionally, we also noticed a scheduling risk. To find a time when everybody can be available to meet was noted as a challenge. More than half of the team have a full-time job, and full-time student, and those who don't have heavy class loads.

Advantage

Motivation: On the other hand, all team members are graduating soon, or are graduating this semester, so we have accumulated knowledge from previous classes and are motivated to perform and finish strong in this class.

Teamwork: And additionally, in terms of team work, I believe this team has done a great job. Each team member independently asks whether there is anything additional they can contribute to the team, after their work is complete. Also, some team members who have more skill in the language and framework do often compensate for others by helping others resolve coding questions with their pages, including package installation, resolving bugs, providing training materials on github, reviewing code, and editing final code.

Learning

The two main challenges mentioned above, skill & scheduling, continued to be challenges throughout the project.

How we closed the knowledge gap: The team trained offline using online resources shared on Slack channel and reached out to tech leads. The tech leads also structured the code in such a way that is easy for the team to follow a pattern, and create training documentation. Even with these efforts, there were still some features we were not able to deliver that is on our P1, which is the sorting by distance. The filter itself is available. There are also other cosmetic UI changes on the site, such as swapping info icon for custom logo, which we did not have a chance to complete. However the majority of changes were completed. We learned that if the team had more distribution of skill, maybe we could deliver all P1, the feedback from Product Review, and P2. The learning here is to limit scope or change the language & framework chosen.

How we closed the scheduling gap: We used team collaboration tools like Skype, Google Spreadsheet and Google Docs, and Trello. Tech leads also used movies, to show the flow of particular use case. The team tried our best to make schedules work using checkpoints, and standing meeting before and after the class. We also made ourselves online daily, over the weekends, and late nights. Given the busy schedule we learned to give our teammates notification of any issues a few days early, so that we can either descope the deliverable, or help each other.

Re: Contribution to the project

Inez Valentine Wibowo <iwibowo@mail.sfsu.edu>

Mon 5/20/2019 7:56 AM

To: Ismael San Juan <ishy30@mail.sfsu.edu>; Aliaksei Siarheyeu <asiarheyeu@mail.sfsu.edu>; Hang Li <hli11@mail.sfsu.edu>; M Wong <MW6354@mail.sfsu.edu>; Romeel Rajoo Chaudhari <rchaudhari@mail.sfsu.edu>; Jianan Mai <jianan@mail.sfsu.edu>;

Thank you all!

On 5/20/19 3:13 AM, Ismael San Juan wrote:

Hi team, this is Ismael and I was a front end developer.

a. project contribution

- gave useful input throughout process
- helped make mockups for reguser dashboard
- was assigned reguser dashboard branch and completed it
- Did QA test plan for our project
- Completed assigned portion of documentation for milestones
- Submitted my assigned work and followed team lead and front and back end leads

b. Commits

I made about 5-10 commits

From: Aliaksei Siarheyeu

Sent: Sunday, May 19, 2019 2:52:05 PM

To: Hang Li; Inez Valentine Wibowo; Ismael San Juan; M Wong; Romeel Rajoo Chaudhari; Jianan Mai

Subject: Re: Contribution to the project

Hi Team,

Role: *Front and Back-end lead.*

Project Contribution:

- **Worked on home page (both back-end and front-end)**
- **Helped with authentication, profile, and admin dashboards. (both back-end and front-end)**
- **Helped with project documentation**
- **Did code reviews for the team members**

Commits:

33 commits, # of lines changed 4,400.

Thanks,
Aliaksei

From: Hang Li
Sent: Sunday, May 19, 2019 12:56 PM
To: Inez Valentine Wibowo; Ismael San Juan; M Wong; Romeel Rajoo Chaudhari; Jianan Mai; Aliaksei Siarheyeu
Subject: Re: Contribution to the project

Hi, this is Hang and I am the front end developer.

a. project contribution

- Helped the beginning development by creating UI mockups and storyboard.
- Helped with the UI design of the front-end listing page and detail page.
- Helped with making a usability test plan, did usability test on posting feature.
- Had a branch to work on, completed portions of documentation for milestone.

b. I made about 10-13 commits

On May 18, 2019, at 3:18 PM, Inez Valentine Wibowo <iwibowo@mail.sfsu.edu> wrote:

Hi
guys, sending my response to team member contributions to start it off:

a.

Contribution to team project & teamwork

- Documentation: set up Google docs for team to contribute, edit , and submit revised and final version of the doc in github and email for submissions.

- Collaborator & Organizer: listen & discuss questions/feedback/input to upcoming/previous tasks, modify deliverables and/or roles as needed, encourage team members to reach out to others with relevant knowledge, communicate tasks and clear instructions to the team, set up checkpoints to manage on-time deliverables, highlight issues to CEO/CTO, keep the team focused on milestone priorities

- Development: write code for Google maps, Google analytics following direction from back end and front end lead

- Motivate team to feel proud about their contributions and deliver best work

b.
number of submissions: 15 commits

8)Team

member contributions

This
is a collection of e-mails that each team member sends

to ALL of their respective team members (so
that all team members can see all the mail on this) outlining:

- a.
His/her
contributions to team project and teamwork (technical and any other) in no more
than half a page – point format is OK.
- b.
Number of
submissions he/she made to github team dev. branch

<https://github.com/CSC-648-SFSU/csc648-sp19-team09/graphs/contributors>

Length

of e-mail is ¾ page max. This is kind of “peer review” – this makes sure you are
comfortable with what you have done since all other team members will see it. Team leads
and front-end leads: mention your role too and explain hat you have done (be brief)

NOTE:

**You must submit copies of raw (original) e-mails sent from each team member to all
other team members
showing
e-mail address lines (sender and recipients)**

Re: Contribution to the project

Jianan Mai

Sun 5/19/2019 12:11 PM

To: M Wong <MW6354@mail.sfsu.edu>; Inez Valentine Wibowo <iwibowo@mail.sfsu.edu>; Ismael San Juan <ishy30@mail.sfsu.edu>; Romeel Rajoo Chaudhari <rchaudhari@mail.sfsu.edu>; Aliaksei Siarheyeu <asiarheyeu@mail.sfsu.edu>; Hang Li <hli11@mail.sfsu.edu>;

My name is Jia Nan and I'm a front end developer for Team 9.

a) Project Contribution

- Worked on the admin dashboard interface with MaterialUI.
- Created non-functional specs for landlords, registered/unregistered users and admins.
- Self-checked the adherence to original non-functional specs.
- I was actively doing the required duties as mentioned above and always followed up with my team lead and front end lead for any changes or concerns that has to do with my duties.

b) Number of Submissions

- On master branch : Two commits.
- On admin/dashboard branch : Nine commits.
- On front end branch : Two commits.

From: M Wong

Sent: Sunday, May 19, 2019 1:01 AM

To: Inez Valentine Wibowo; Ismael San Juan; Romeel Rajoo Chaudhari; Jianan Mai; Aliaksei Siarheyeu; Hang Li

Subject: Re: Contribution to the project

My name is Marcus, I was a backend and frontend coder for this project.

a) Project Contribution

- contact / chat page, new listing form and a few other small features
- did GitHub pull request reviews
- collaborated on project design documentations

b) Number of submissions

- Github commits 140
- Doc submissions 2

From: Inez Valentine Wibowo

Sent: Saturday, May 18, 2019 3:18:12 PM

To: Ismael San Juan; M Wong; Romeel Rajoo Chaudhari; Jianan Mai; Aliaksei Siarheyeu; Hang Li; Inez Valentine Wibowo

Subject: Contribution to the project

Hi guys, sending my response to team member contributions to start it off:

a. Contribution to team project & teamwork

- Documentation: set up Google docs for team to contribute, edit , and submit revised and final version of the doc in github and email for submissions.
- Collaborator & Organizer: listen & discuss questions/feedback/input to upcoming/previous tasks, modify deliverables and/or roles as needed, encourage team members to reach out to others with relevant knowledge, communicate tasks and clear instructions to the team, set up checkpoints to manage on-time deliverables, highlight issues to CEO/CTO, keep the team focused on milestone priorities
- Development: write code for Google maps, Google analytics following direction from back end and front end lead
- Motivate team to feel proud about their contributions and deliver best work

b. number of submissions: 15 commits

8)Team member contributions

This is a collection of e-mails that each team member sends to ALL of their respective team members (so that all team members can see all the mail on this) outlining:

a.

His/her contributions to team project and teamwork (technical and any other) in no more than half a page – point format is OK.

b.

Number of submissions he/she made to github team dev. branch

<https://github.com/CSC-648-SFSU/csc648-sp19-team09/graphs/contributors>

Length of e-mail is ¾ page max. This is kind of “peer review”

– this makes sure you are comfortable with what you have done since all other team members will see it. Team leads and front-end leads: mention your role too and explain hat you have done (be brief)

NOTE: You must submit copies of raw (original) e-mails sent from each team member to all other team members showing e-mail address lines (sender and recipients).

Re: Contribution to the project

Romeel Rajoo Chaudhari

Sat 5/18/2019 3:31 PM

To: Inez Valentine Wibowo <iwibowo@mail.sfsu.edu>; Ismael San Juan <ishy30@mail.sfsu.edu>; M Wong <MW6354@mail.sfsu.edu>; Jianan Mai <jianan@mail.sfsu.edu>; Aliaksei Siarheyeu <asiarheyeu@mail.sfsu.edu>; Hang Li <hli11@mail.sfsu.edu>;

Hi, this is Romeel and I am writing my contribution to the team project

- Worked on making decisions in the group
- Helped with some technical issues at the environmental setup stage
- Made technical contribution like made the login and registration pages.
- Had multiple branches to work on, so made about 10- 15 commits.
- Submitted my assigned work and always followed the group members and team lead.

Thank You
Romeel

From: Inez Valentine Wibowo <iwibowo@mail.sfsu.edu>

Date: Saturday, May 18, 2019 at 3:18 PM

To: Ismael San Juan <ishy30@mail.sfsu.edu>, M Wong <MW6354@mail.sfsu.edu>, Romeel Rajoo Chaudhari <rchaudhari@mail.sfsu.edu>, Jianan Mai <jianan@mail.sfsu.edu>, Aliaksei Siarheyeu <asiarheyeu@mail.sfsu.edu>, Hang Li <hli11@mail.sfsu.edu>, Inez Valentine Wibowo <iwibowo@mail.sfsu.edu>

Subject: Contribution to the project

Hi guys, sending my response to team member contributions to start it off:

a. Contribution to team project & teamwork

- Documentation: set up Google docs for team to contribute, edit , and submit revised and final version of the doc in github and email for submissions.
- Collaborator & Organizer: listen & discuss questions/feedback/input to upcoming/previous tasks, modify deliverables and/or roles as needed, encourage team members to reach out to others with relevant knowledge, communicate tasks and clear instructions to the team, set up checkpoints to manage on-time deliverables, highlight issues to CEO/CTO, keep the team focused on milestone priorities
- Development: write code for Google maps, Google analytics following direction from back end and front end lead
- Motivate team to feel proud about their contributions and deliver best work

b. number of submissions: 15 commits

8) Team member contributions

This is a collection of e-mails that each team member sends to ALL of their respective team members (so that all team members can see all the mail on this) outlining:

- a.
- b. His/her contributions
- c. to team project and teamwork (technical and any other) in no more than half a page – point format is OK.
- d.
- e.
- f.
- g. Number of submissions
- h. he/she made to github team dev. branch
- i.

<https://github.com/CSC-648-SFSU/csc648-sp19-team09/graphs/contributors>

Length of e-mail is $\frac{3}{4}$ page max. This is kind of “peer review” – this makes sure you are comfortable with what you have done since all other team members will see it. Team leads and front-end leads: mention your role too and explain what you have done (be brief)

NOTE: You must submit copies of raw (original) e-mails sent from each team member to all other team members showing e-mail address lines (sender and recipients)

8. Post analysis – lessons learned (one page or so)

Main challenges

Knowledge risk: At the start of this project, we already noticed and highlighted a knowledge risk, which is that most of the people on the team do not have experience with ReactJs, and have little experience with JavaScript, the language and framework we've chosen. We chose this because we believed that it would give us the ability to create a nice, responsive, website on desktop and mobile, in the time constraints.

Scheduling risk: Additionally, we also noticed a scheduling risk. To find a time when everybody can be available to meet was noted as a challenge. More than half of the team have a full-time job, and full-time student, and those who don't have heavy class loads.

Advantage

Motivation: On the other hand, all team members are graduating soon, or are graduating this semester, so we have accumulated knowledge from previous classes and are motivated to perform and finish strong in this class.

Teamwork: And additionally, in terms of team work, I believe this team has done a great job. Each team member independently asks whether there is anything additional they can contribute to the team, after their work is complete. Also, some team members who have more skill in the language and framework do often compensate for others by helping others resolve coding questions with their pages, including package installation, resolving bugs, providing training materials on github, reviewing code, and editing final code.

Learning

The two main challenges mentioned above, skill & scheduling, continued to be challenges throughout the project.

How we closed the knowledge gap: The team trained offline using online resources shared on Slack channel and reached out to tech leads. The tech leads also structured the code in such a way that is easy for the team to follow a pattern, and create training documentation. Even with these efforts, there were still some features we were not able to deliver that is on our P1, which is the sorting by distance. The filter itself is available. There are also other cosmetic UI changes on the site, such as swapping info icon for custom logo, which we did not have a chance to complete. However the majority of changes were completed. We learned that if the team had more distribution of skill, maybe we could deliver all P1, the feedback from Product Review, and P2. The learning here is to limit scope or change the language & framework chosen.

How we closed the scheduling gap: We used team collaboration tools like Skype, Google Spreadsheet and Google Docs, and Trello. Tech leads also used movies, to show the flow of particular use case. The team tried our best to make schedules work using checkpoints, and standing meeting before and after the class. We also made ourselves online daily, over the weekends, and late nights. Given the busy schedule we learned to give our teammates notification of any issues a few days early, so that we can either descope the deliverable, or help each other.