Software Engineering CSC 648/848 Section 02 Spring 2019

GATORRENT

Milestone 2

Team 08

March 22th, 2019

Local

Cheryl Fong
cfong8@mail.sfsu.edu

Jerry Wong

Chirag Agarwal

Harold Pedroso

Johanna Hadgu

Sharan Lao

Ermias Haile

REVISION TABLE		
Date	**Version Number**	**Description**
3/20/2019	1.0	Date submitted for review
3/21/2019	1.1	Revised after Feedback

FROZEN

TABLE OF CONTENTS

1. DATA DEFINITIONS V2

User/Users A user of the site. Could be registered or non-registered. If registered, will be stated as “Registered user(s)” otherwise “Non-registered user(s)”.

Registered User A user who has signed up on the website, i.e. the user’s username, password, and email address will be contained in the Users Table. This user has full access to all user functionalities, such as sending messages to other users and posting listings.

Non-registered User A user of the website who did not sign up, i.e. this user does not have an entry on the Users Table. This user is only allowed to browse listings on the website.

Admin/Admins Website staff who are site usage maintainers, i.e. responsible for handling the appropriateness of user activities by disapproving or deactivating users. This user has full access to all user functionalities.

Student A user of the site, specifically looking/browsing for listings.

Landlord A user of the site, specifically looking to rent out their property**.** Has the intention of creating listings.

Filter The ability/functionality to filter results from the Listings Table.

Listing An entry on the Listings Table, which includes information on a posted property such as rent-type, its description, rent-type size in square feet, the rent price per month, distance from campus, if pets are allowed, if the rent-type is a shared or not shared occupancy and postcode.

rent-type Refers to the type of space that is intended to be rented out (e.g. house, room, apartment, etc).

Registration An entry added to the Users Table containing user data during the registration process, such as their username, password, and email address.

Drop-pins Refers to an entry on the Markers Table, as a visual marker 📍 on Google Maps

Terms and Conditions The website staff is not responsible for interactions between users outside the realms of the website.

2. FUNCTIONAL REQUIREMENTS V2

2.1 Priority 1

Unregistered Users:

2.1.1 Shall be able to see/browse and filter all **listing** results on the website.

2.1.2 Shall have the option to register themselves as a **user** of the website.

2.1.3 Shall have the option to use the message function to inform **admins** about site related issues, such as inappropriate images or untrustworthy listings via the contact button on the side footer.

2.1.4 Shall be able to **filter listing** results by lowest or highest price, price range, postcode, distance from campus, rent-type, rent-type price, if pets are allowed, and if the rent-type is shared.

Registered Users:

2.1.5 Shall have access to all the functionalities as a **non-registered user**.

2.1.6 Shall be able to see/browse all **listing** results on the website, including the description of **listing** results.

2.1.7 Shall accept all **terms and conditions** during the registration process.

2.1.8 Shall be able to publish a **listing** for rent, i.e. this process creates a **listing** entry with the required data fields as stated in the **Listings Table**.

2.1.9 Shall be able to communicate with other **users** via the website message function.

2.1.10 Shall be able to check for messages related to posted listings.

Administration:

2.1.11 Shall have access to all the functionalities of a **registered user**.

2.1.12 Shall approve or disprove **listing** results before they go live on the site.

2.1.13 Shall view website **activity log** and act accordingly to unusual website behaviour.

2.1.14 Shall deactivate **users** for their misuse of the website.

2.2 Priority 2

Registered Users:

2.2.1 Shall have the option to view nearby transit methods via the **Google Transit API**.

2.3 Priority 3

Registered Users

2.3.1 Shall have the option to view their **username**.

2.3.2 Shall have the option to view and change their **email address**.

2.3.3 Shall have the option to view and change their **user** preferences.

2.3.4 Shall receive notifications and the option to turn it off for new messages sent to them.

2.3.5 Shall have the option to receive notifications for newly added **listing** entries that match their price range.

2.3.6 Shall have the option to provide their price range preference on their **user** profile.

2.3.7 Shall have the option to save and unsave desired **listing** results.

2.3.8 Shall have the option to view all saved **listing** results.

2.3.9 Shall have the option to return to a draft version of their **listing**.

2.3.10 Shall have the option to label a **listing** as currently occupied, under maintenance and vacant/open.

2.3.11 Shall have the option to share a **listing** on social media i.e. on **Facebook** and **Twitter** via their respective **API's**.

2.3.12 Shall have the option to view directions to any location they prefer, with SFSU as a quick selection option, from a **listing** result's location address as a starting point.

2.3.13 Shall have the option to rate any other user, if and only if the two **registered users** (the one rating and the one being rated) have at least sent a message to each other via the message function.

2.3.14 Shall have the option to view the number of **registered users** who have their preferred price preference equal to the **listing** price that this registered user has created.

2.3.15 Shall be able to see available **listing** results as **drop-pins** of locations on the Google Maps, via the **Google Maps API**.

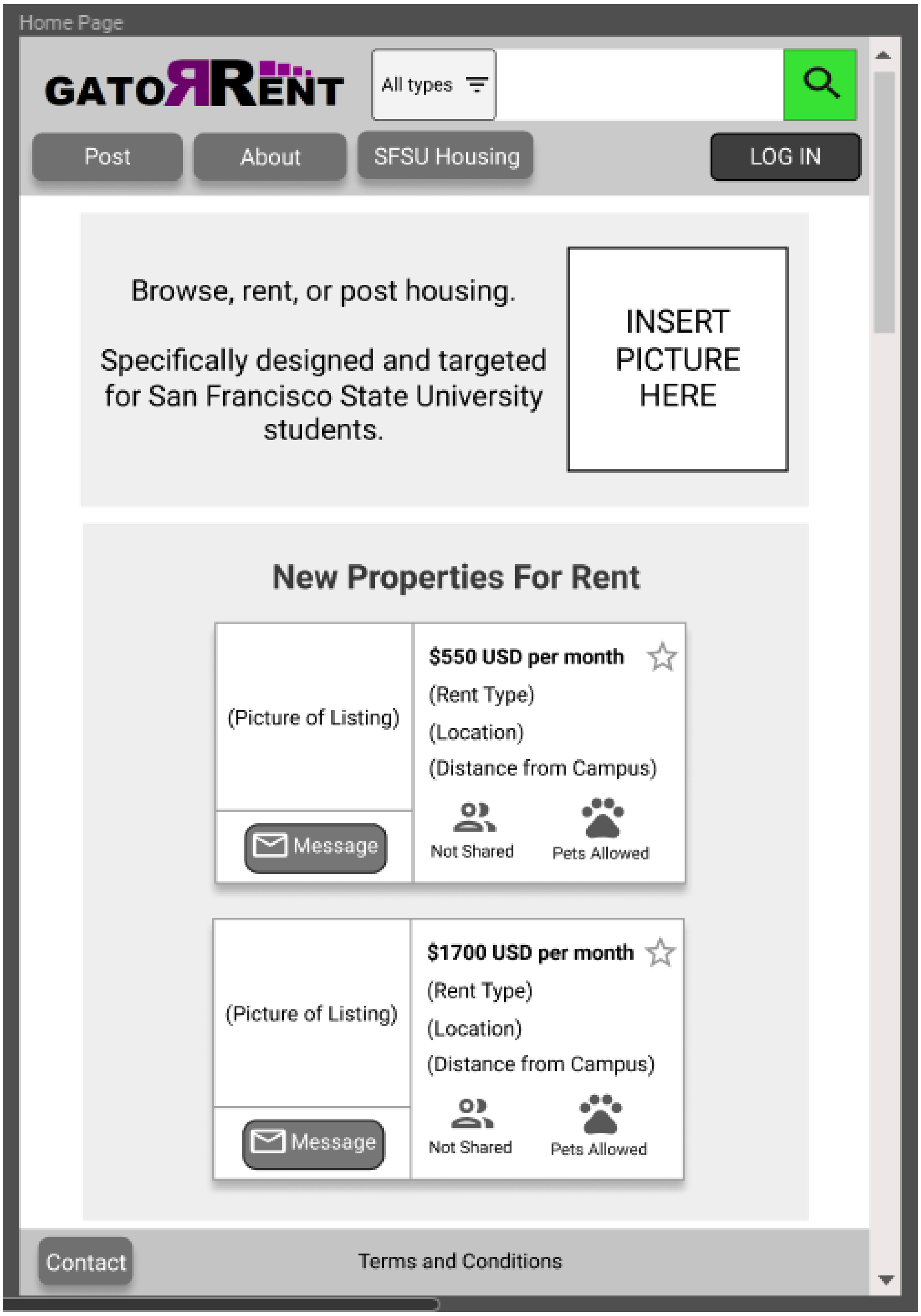
2.3.16 Shall be able to edit or modify their send **messages.**

2.3.17 Shall be able to receive read receipts when their message has been seen by the receiver (isRead function).

3. UI MOCKUPS AND STORYBOARDS

Link to Interactive Figma UI Mockups

3.1. UI Mockups



Home Page

GATORRENT

All types

Post About SFSU Housing LOG IN

Browse, rent, or post housing.

Specifically designed and targeted for San Francisco State University students.

INSERT PICTURE HERE

New Properties For Rent

(Picture of Listing)

Message

$550 USD per month

(Rent Type)

(Location)

(Distance from Campus)

Not Shared Pets Allowed

(Picture of Listing)

Message

$1700 USD per month

(Rent Type)

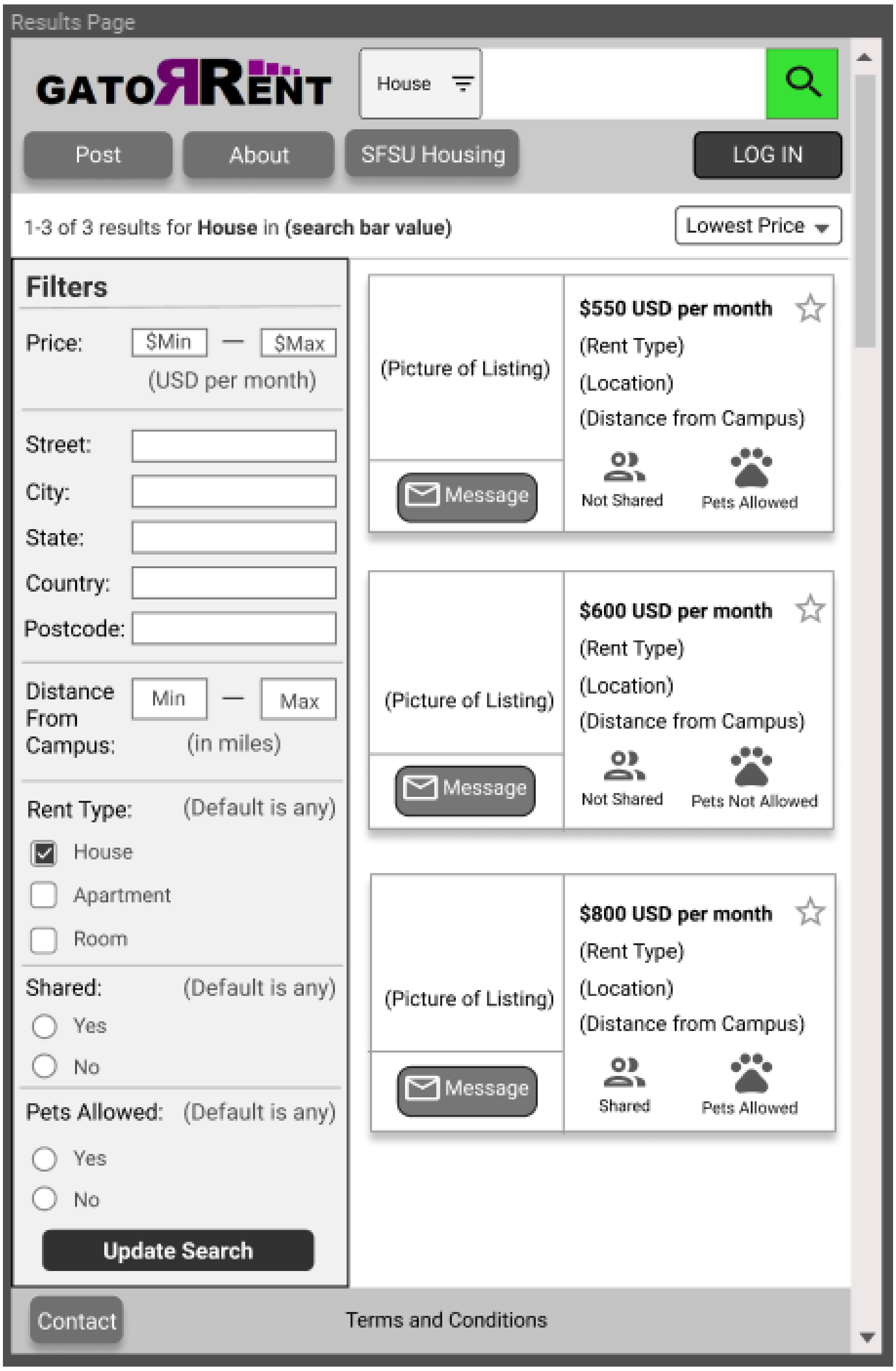
(Location)

(Distance from Campus)

Not Shared Pets Allowed

Contact Terms and Conditions

Home Page



House

1-3 of 3 results for **House** in **(search bar value)**

Price: $Min — $Max (USD per month)

Street:

City:

State:

Country:

Postcode:

Distance From Campus: Min — Max (in miles)

Rent Type: (Default is any)

☑ House

□ Apartment

□ Room

Shared: (Default is any)

○ Yes

○ No

Pets Allowed: (Default is any)

○ Yes

○ No

(Picture of Listing)

$550 USD per month ☆
(Rent Type)
(Location)
(Distance from Campus)
Not Shared
Pets Allowed

Message

(Picture of Listing)

$600 USD per month ☆
(Rent Type)
(Location)
(Distance from Campus)
Not Shared
Pets Not Allowed

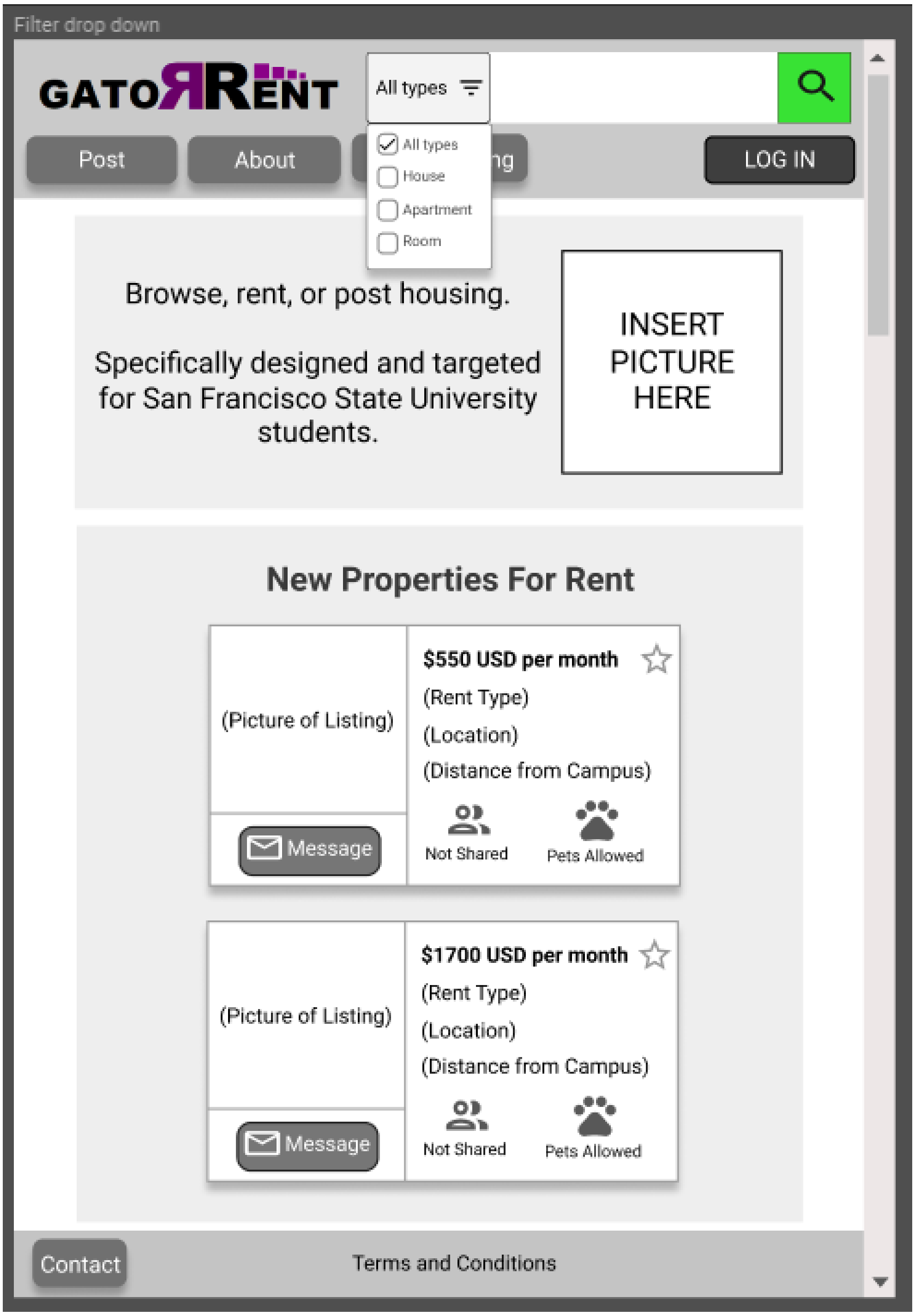
Message

(Picture of Listing)

$800 USD per month ☆
(Rent Type)
(Location)
(Distance from Campus)
Shared
Pets Allowed

Message

Browsing Page



Browse, rent, or post housing.

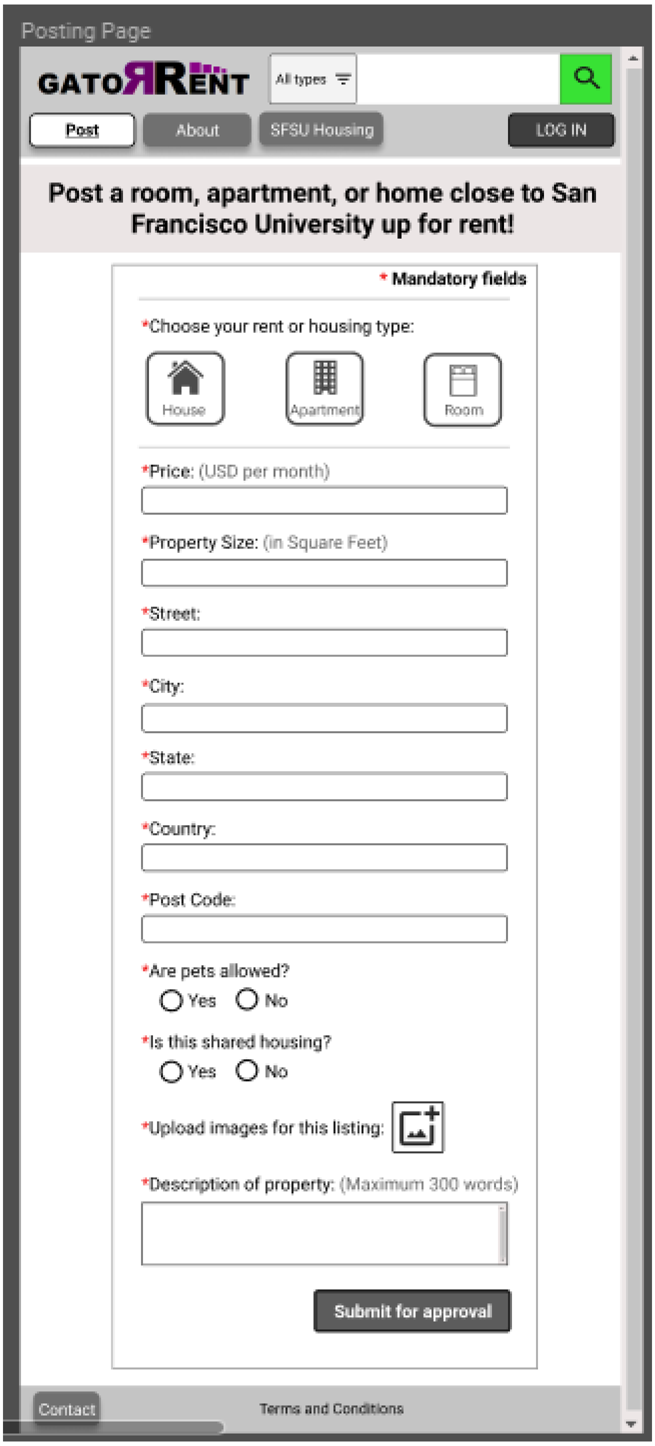
Specifically designed and targeted for San Francisco State University students.

New Properties For Rent

$550 USD per month
(Rent Type)
(Location)
(Distance from Campus)

$1700 USD per month
(Rent Type)
(Location)
(Distance from Campus)

Filter



Post a room, apartment, or home close to San Francisco University up for rent!

* Mandatory fields

*Choose your rent or housing type:

*Price: (USD per month)

*Property Size: (in Square Feet)

*Street:

*City:

*State:

*Country:

*Post Code:

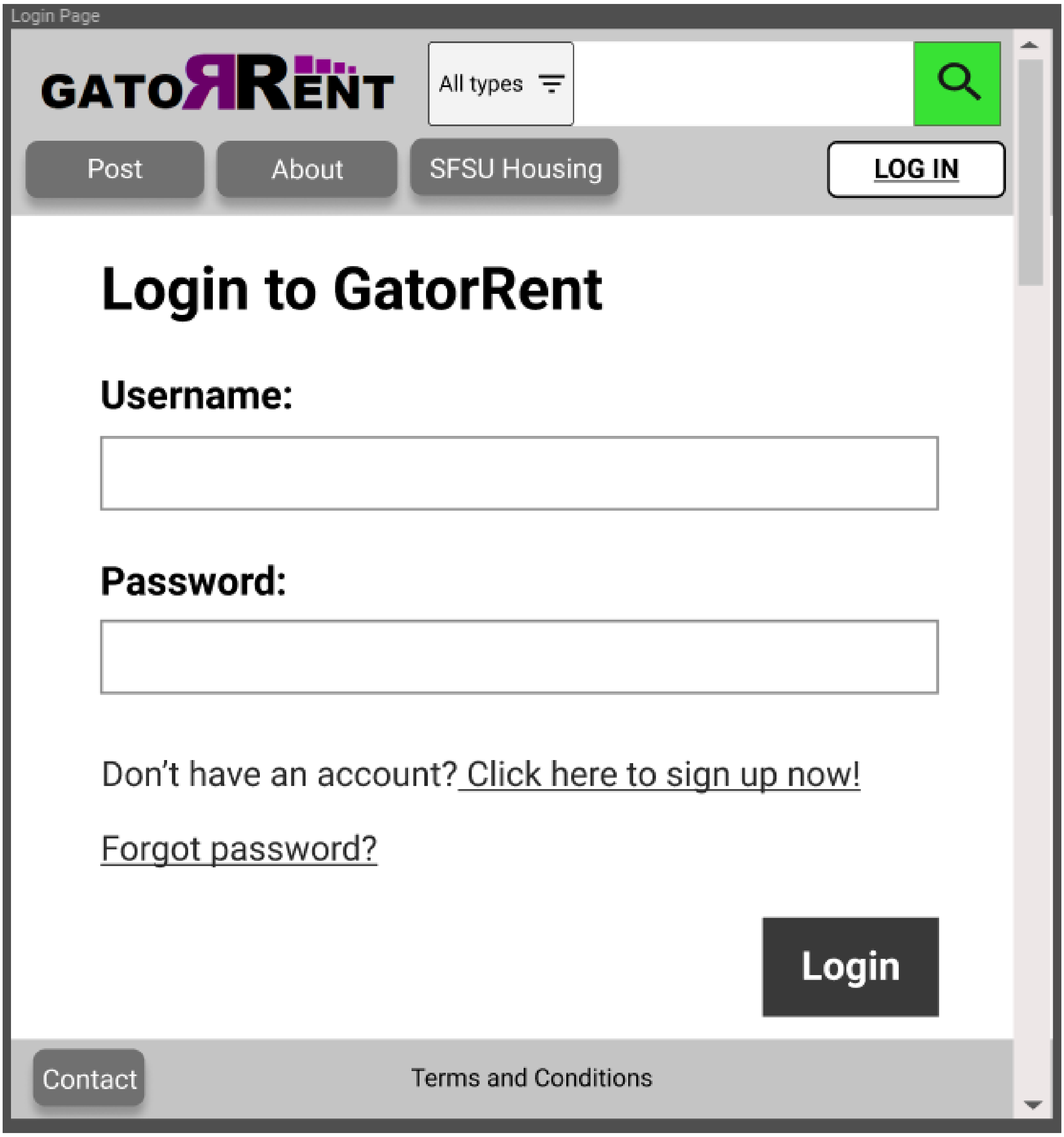
*Are pets allowed?
○ Yes ○ No

*Is this shared housing?
○ Yes ○ No

*Upload images for this listing:

*Description of property: (Maximum 300 words)

Posting a Listing



Login to GatorRent

Username:

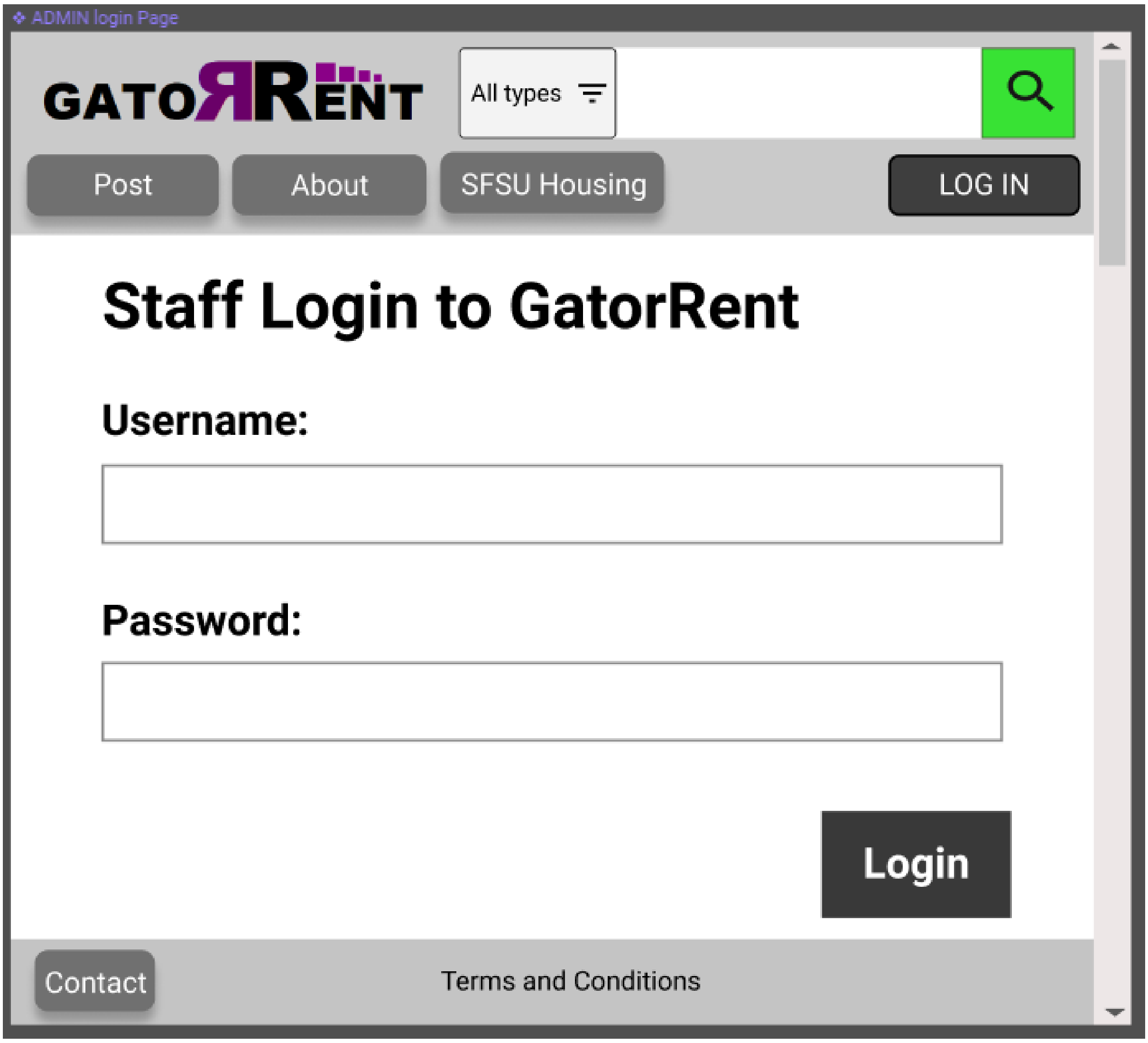
Password:

Don’t have an account? Click here to sign up now!

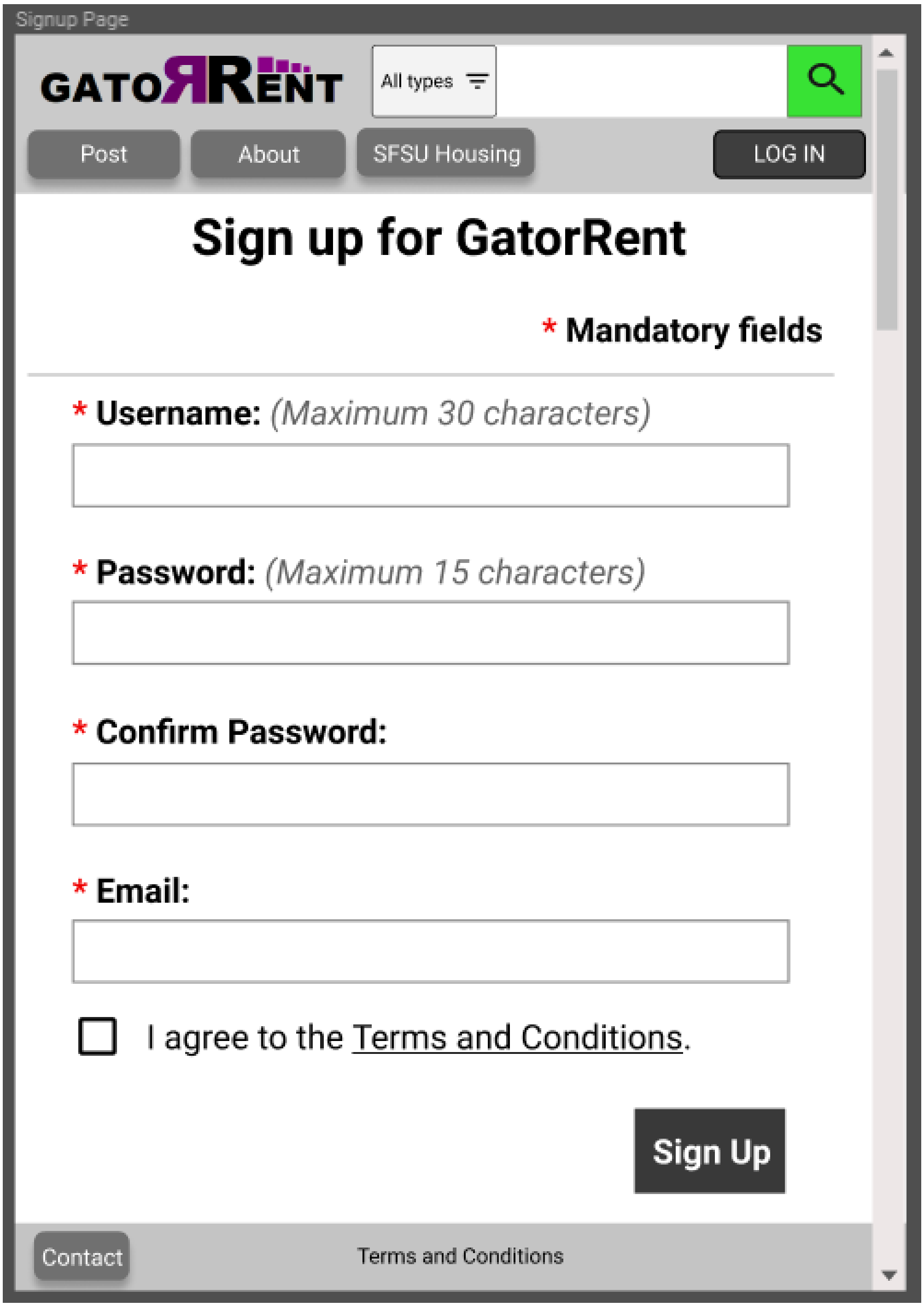
Forgot password?

Login

Login Page



Admin Login Page



Sign up for GatorRent

* Mandatory fields

* Username: *(Maximum 30 characters)*

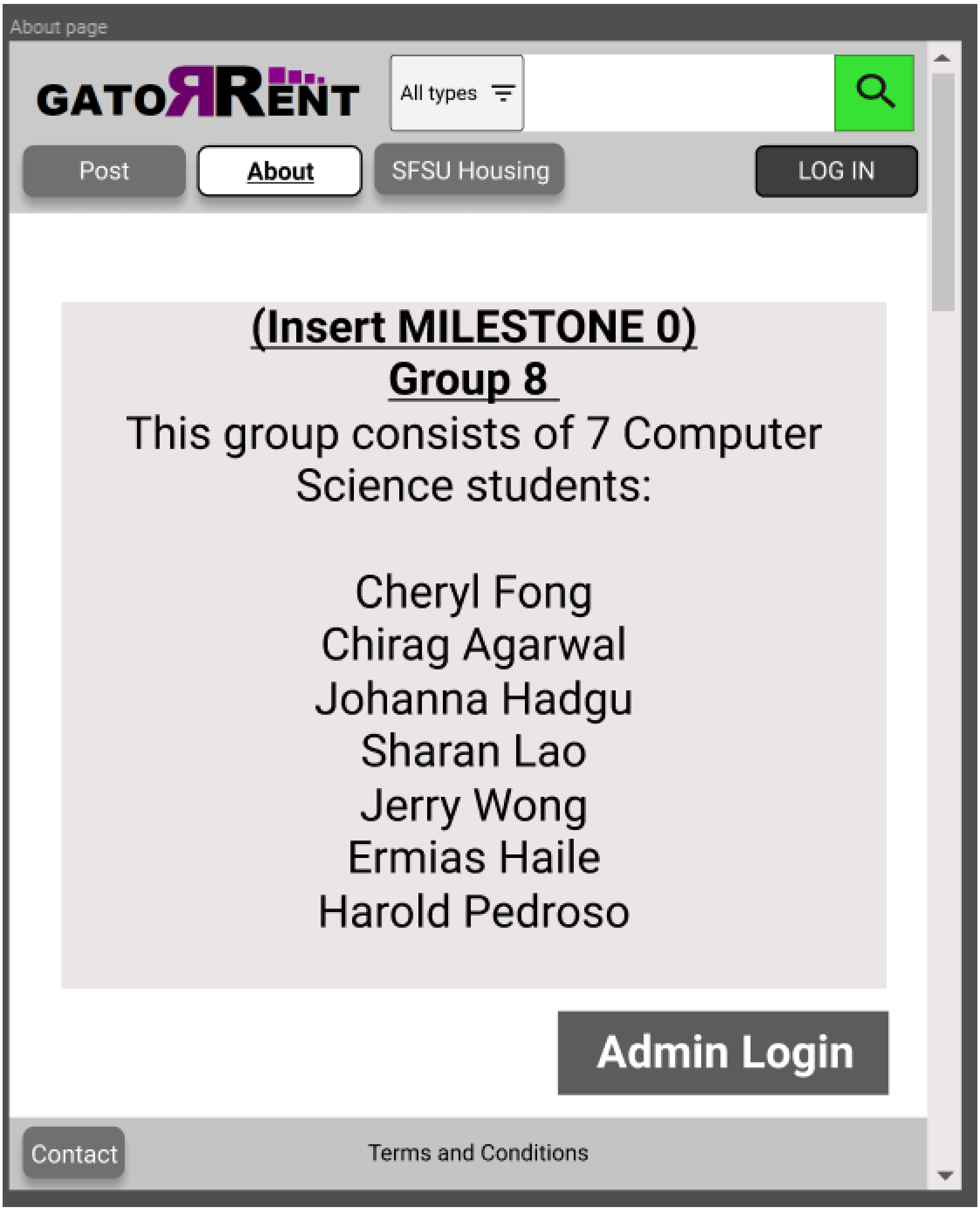
* Password: *(Maximum 15 characters)*

* Confirm Password:

* Email:

□ I agree to the Terms and Conditions.

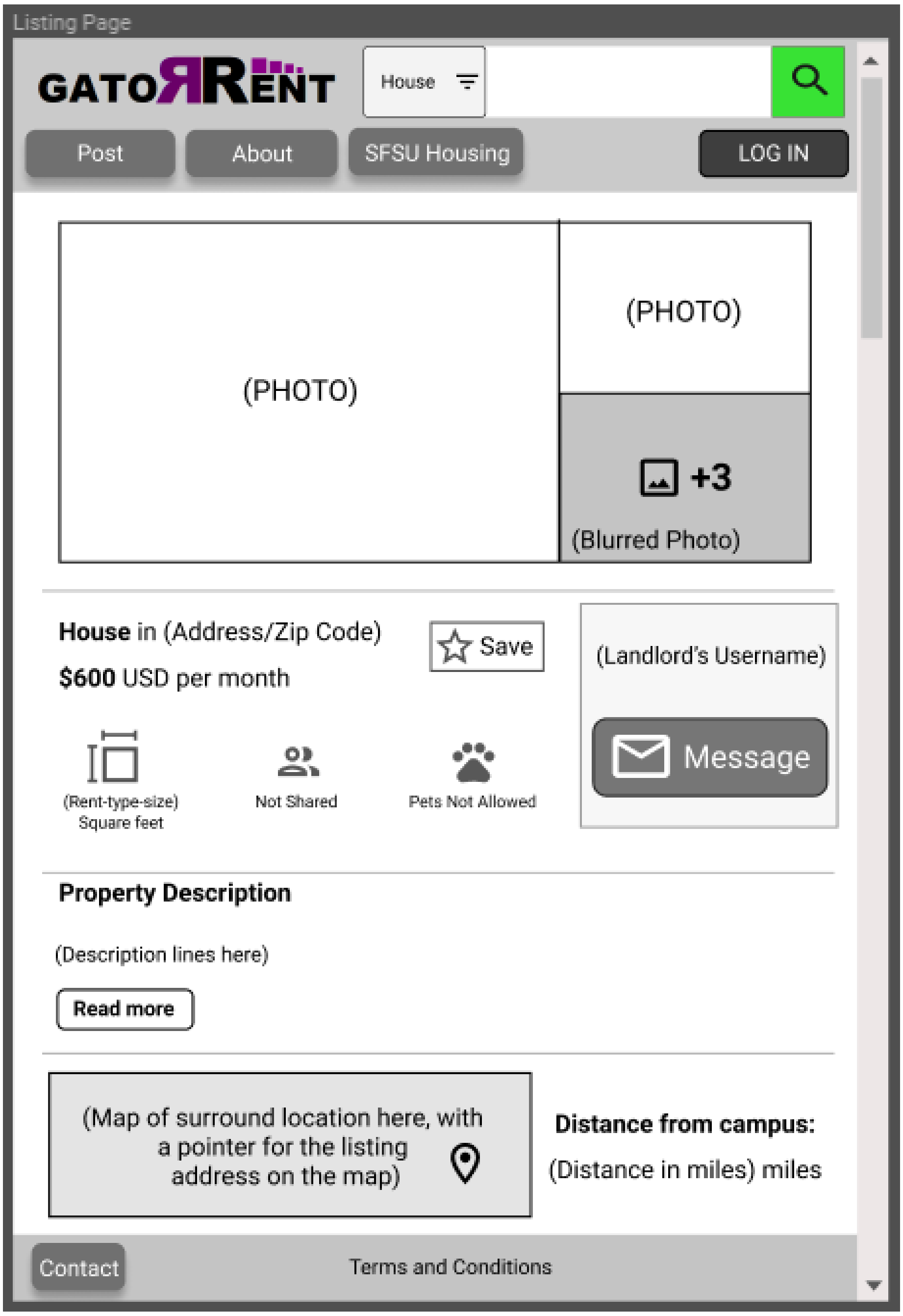
Registration Page



(Insert MILESTONE 0)
Group 8
This group consists of 7 Computer Science students:

Cheryl Fong
Chirag Agarwal
Johanna Hadgu
Sharan Lao
Jerry Wong
Ermias Haile
Harold Pedroso

About Page



Listing Page

House in (Address/Zip Code)

$600 USD per month

Property Description

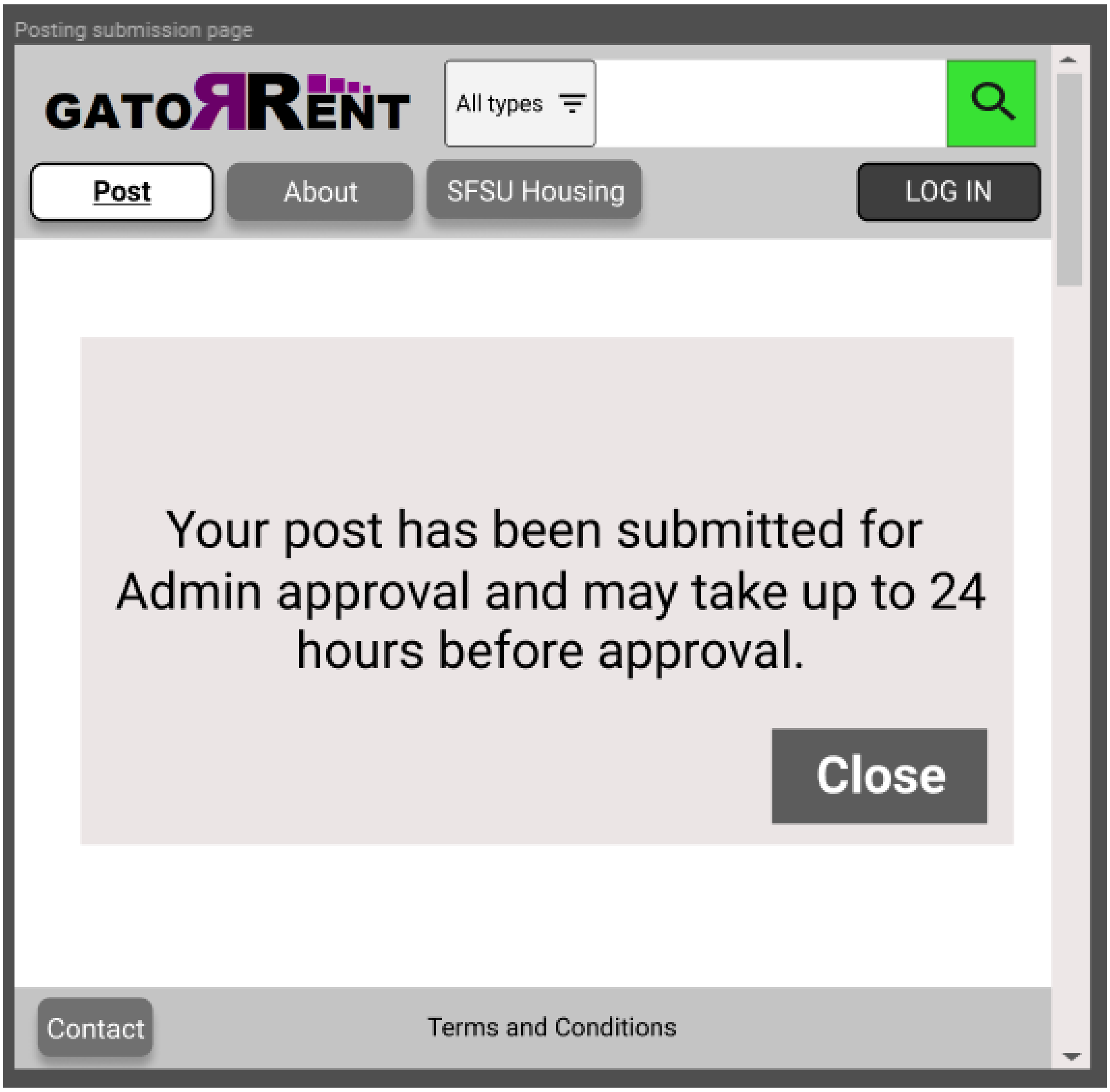
(Description lines here)

(Map of surround location here, with a pointer for the listing address on the map)

Distance from campus:

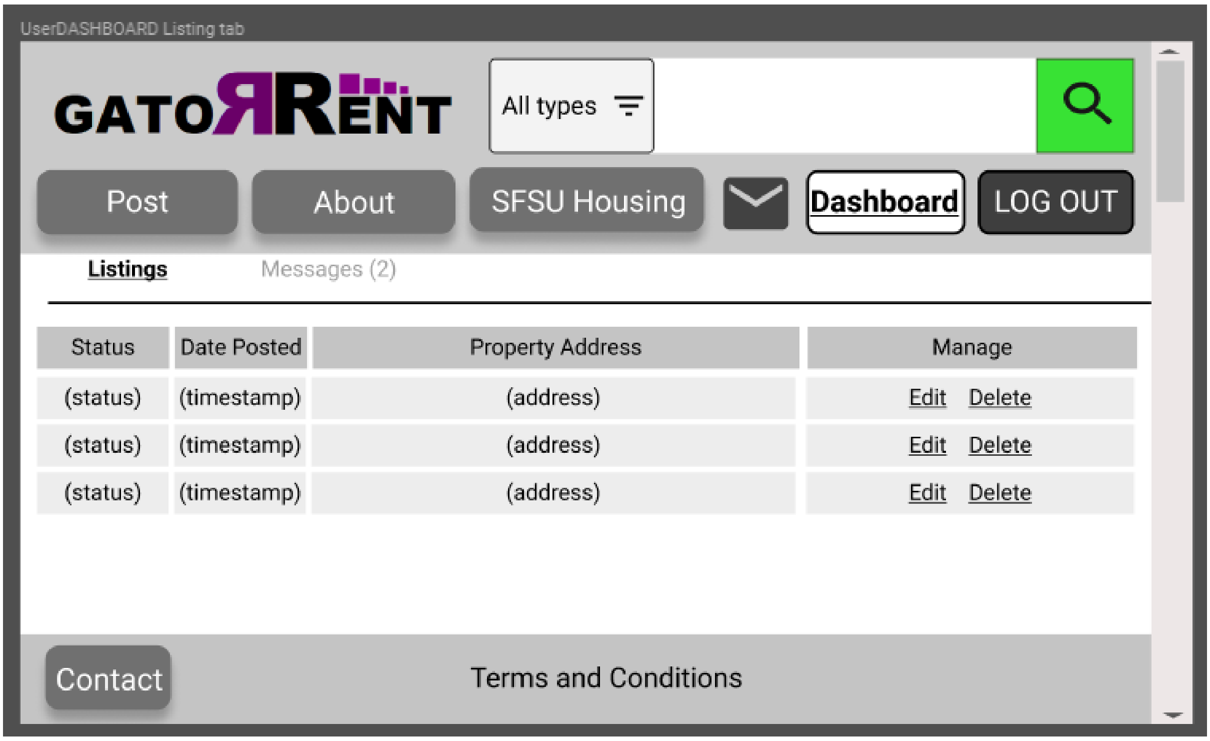
(Distance in miles) miles

Selecting a Specific Listing



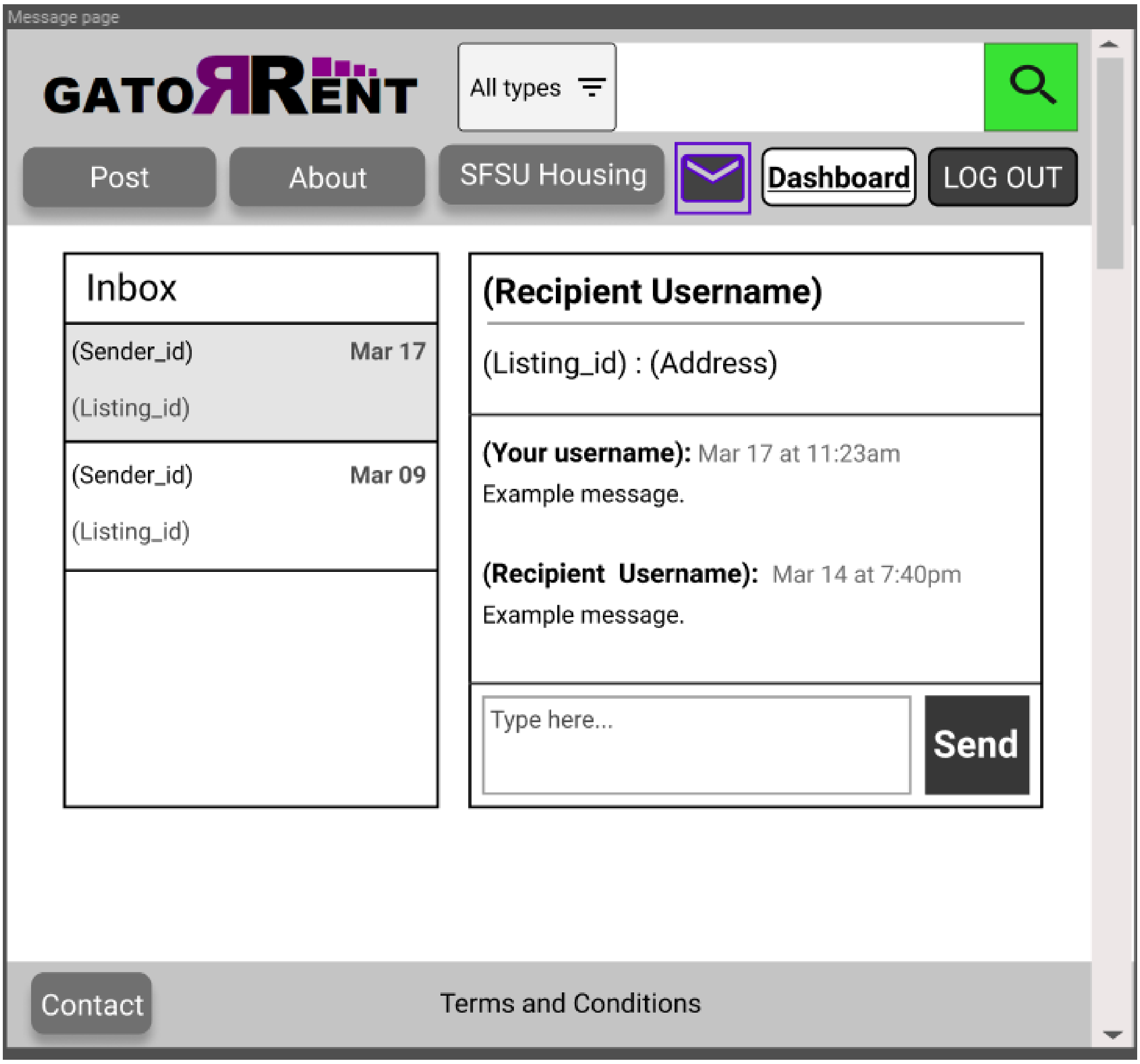
Your post has been submitted for Admin approval and may take up to 24 hours before approval.

After Submitting a Post



Status	Date Posted	Property Address	Manage
(status)	(timestamp)	(address)	Edit Delete
(status)	(timestamp)	(address)	Edit Delete
(status)	(timestamp)	(address)	Edit Delete

User Dashboard



Inbox

(Sender_id) Mar 17
(Listing_id)

(Sender_id) Mar 09
(Listing_id)

(Recipient Username)

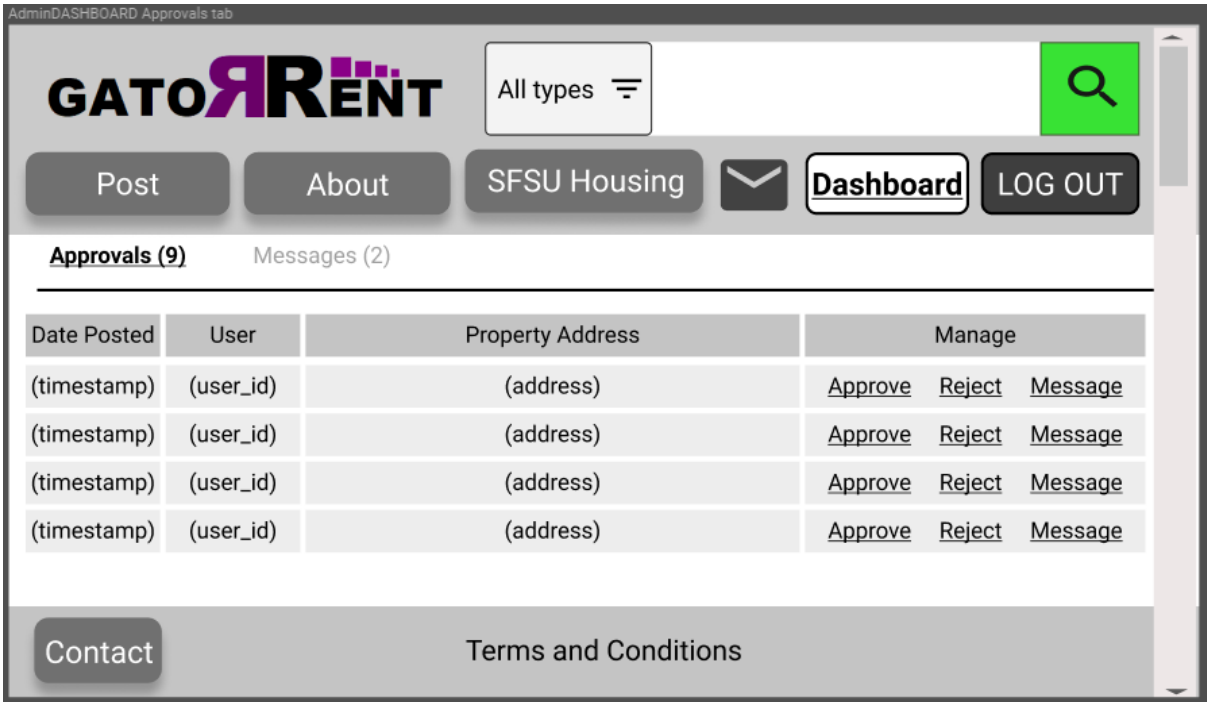
(Listing_id) : (Address)

(Your username): Mar 17 at 11:23am
Example message.

(Recipient Username): Mar 14 at 7:40pm
Example message.

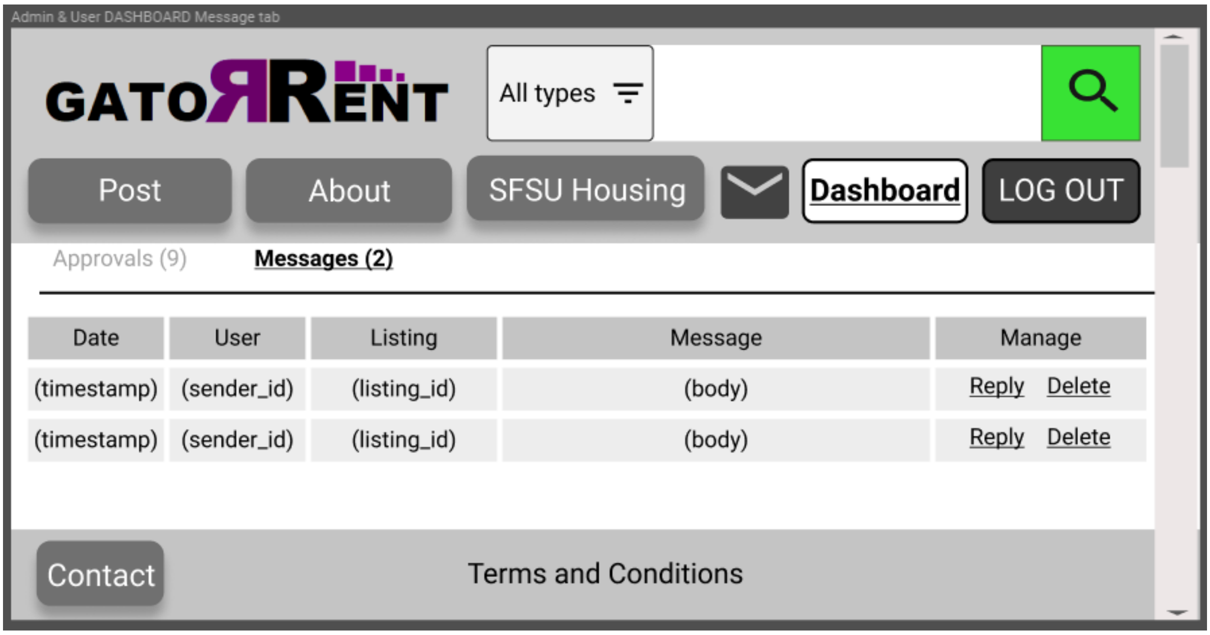
Type here...

User Messages



Date Posted	User	Property Address	Manage
(timestamp)	(user_id)	(address)	Approve Reject Message
(timestamp)	(user_id)	(address)	Approve Reject Message
(timestamp)	(user_id)	(address)	Approve Reject Message
(timestamp)	(user_id)	(address)	Approve Reject Message

Admin Dashboard (Listing Management)



Admin & User DASHBOARD Message tab

Date	User	Listing	Message	Manage
(timestamp)	(sender_id)	(listing_id)	(body)	Reply Delete
(timestamp)	(sender_id)	(listing_id)	(body)	Reply Delete

Admin Dashboard (Messages)

3.2 STORYBOARDS

3.2.1 Unregistered User (Kelly)

As a **non-registered user**, Kelly is presented with a variety of apartments **listings** available for rent and is able to **browse** through them. Kelly wants to search for apartments that are in close proximity to public transportation, campus and also within her price range. She **filters** her results by **zip code**, **distance,** and **price** and is presented with several apartment listings that are according to her specifications. Now that she found an acceptable apartment, she has solved her problem with GatorRent.

3.2.2 Registered User (Andrew)

Upon opening the website, Andrew notices that several **listings** are far beyond his price range and are not convenient enough to support his commute to work and school. Therefore, he filters his results according to the **distance** from local transit and **price.** Additionally, GatorRents allows him to accommodate for his cat allergies by **filtering** out renters that allow **pets** to reside in the premises. After finding one apartment that satisfies his conditions, Andrew creates an account and becomes a **registered user,** where he is required to accept the **terms and conditions** of the site. Once this is done, he is able to message the **landlord** and secure an apartment with the help of GatorRent.

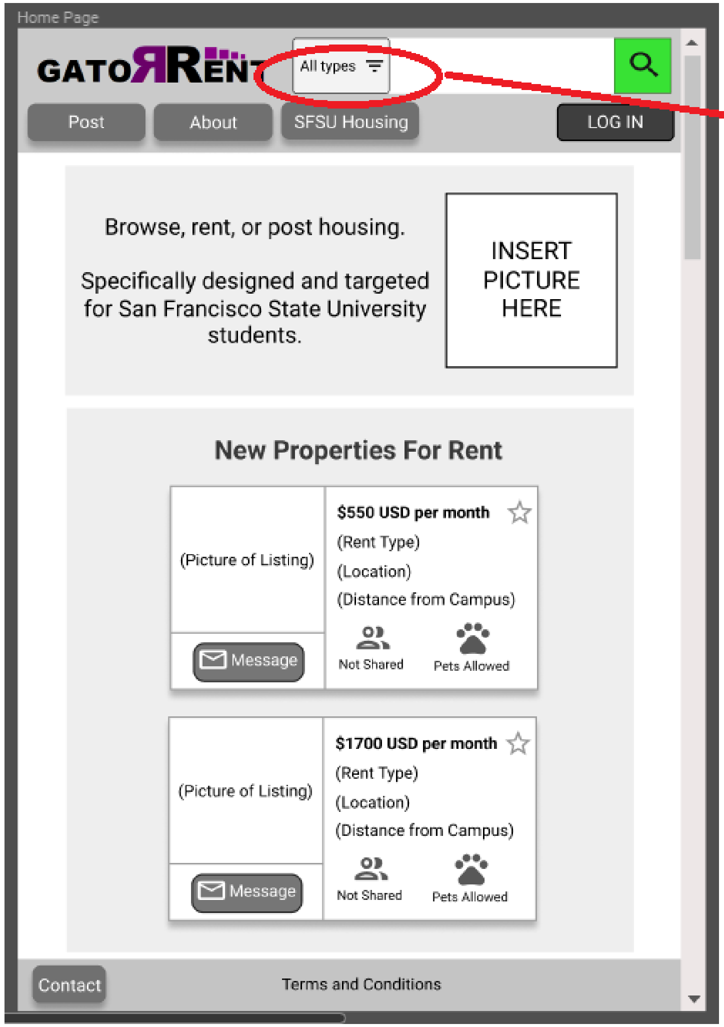
3.2.3 Landlord (Max)

Having logged into GatorRent, Max is able to find options that allow him to post his apartment as a potential place to rent. Once he provides a description of the place, pricing, and images of the **property**, Max is prompted to register or log in before completing the post. Then, Max submits his post for **admin** approval.

3.2.4 Admin (Jordan)

After logging in to the GatorRent dashboard, Jordan is greeted with a well-organized data list to include **listings** in need of approval and user-abuse reports. As part of her job, during a session of listing evaluation, Jordan comes across a **listing** that did not include photos of the **property**. Jordan denies the listing, contacting the lister and specifying that he'll need to modify his **listing** before it becomes live on the site. After approving several **listings**, she also notices that another **landlord** has been reported by 3 registered users. Having recalled that this specific **landlord** had already received a warning several months prior, Jordan immediately terminates the account.

1. **Use Case #1**
   a. Browse, search filter (filters: zip code, DISTANCE, PRICE, allowing pets, no), select, create account (username, password, email) accept Terms and Conditions), message landlord.



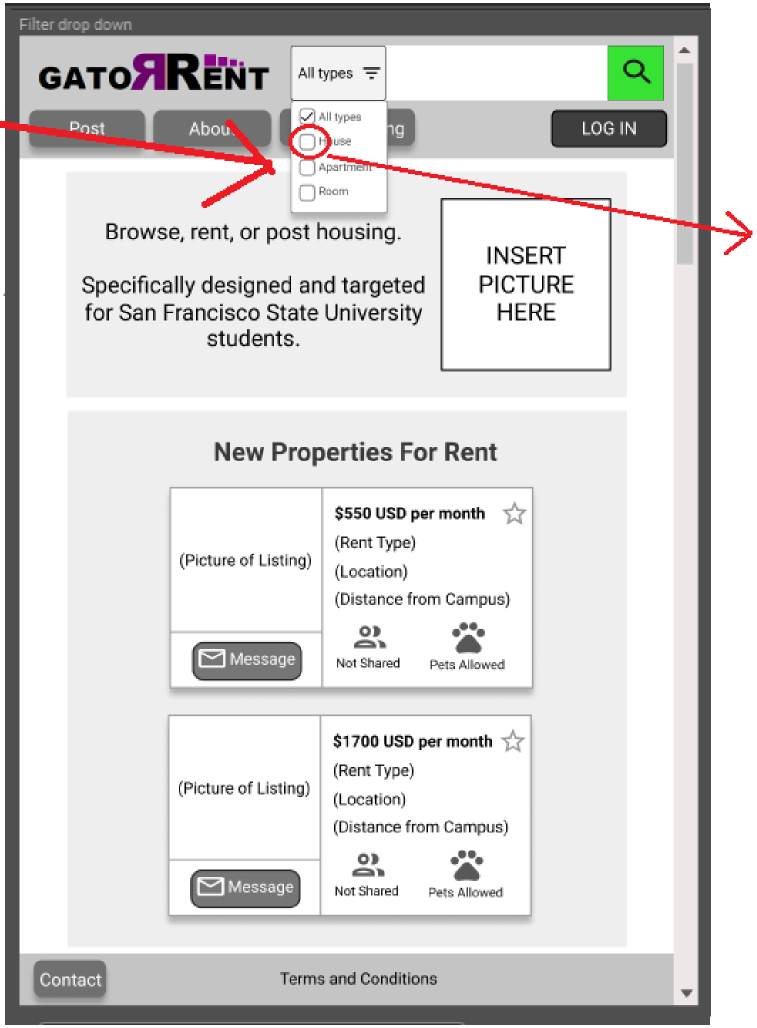
Browse, rent, or post housing.

Specifically designed and targeted for San Francisco State University students.

New Properties For Rent

$550 USD per month
(Rent Type)
(Location)
(Distance from Campus)

$1700 USD per month
(Rent Type)
(Location)
(Distance from Campus)



Browse, rent, or post housing.

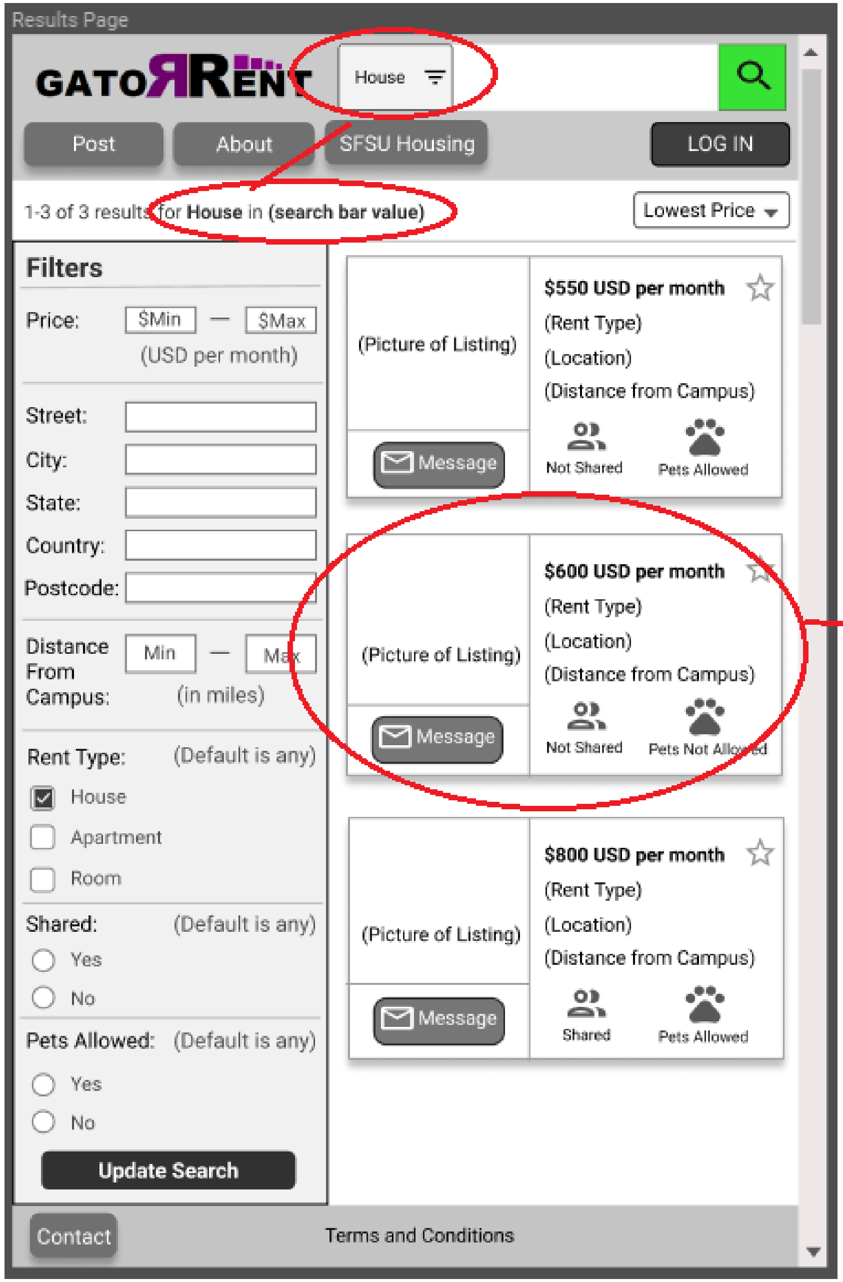
Specifically designed and targeted for San Francisco State University students.

New Properties For Rent

$550 USD per month
(Rent Type)
(Location)
(Distance from Campus)

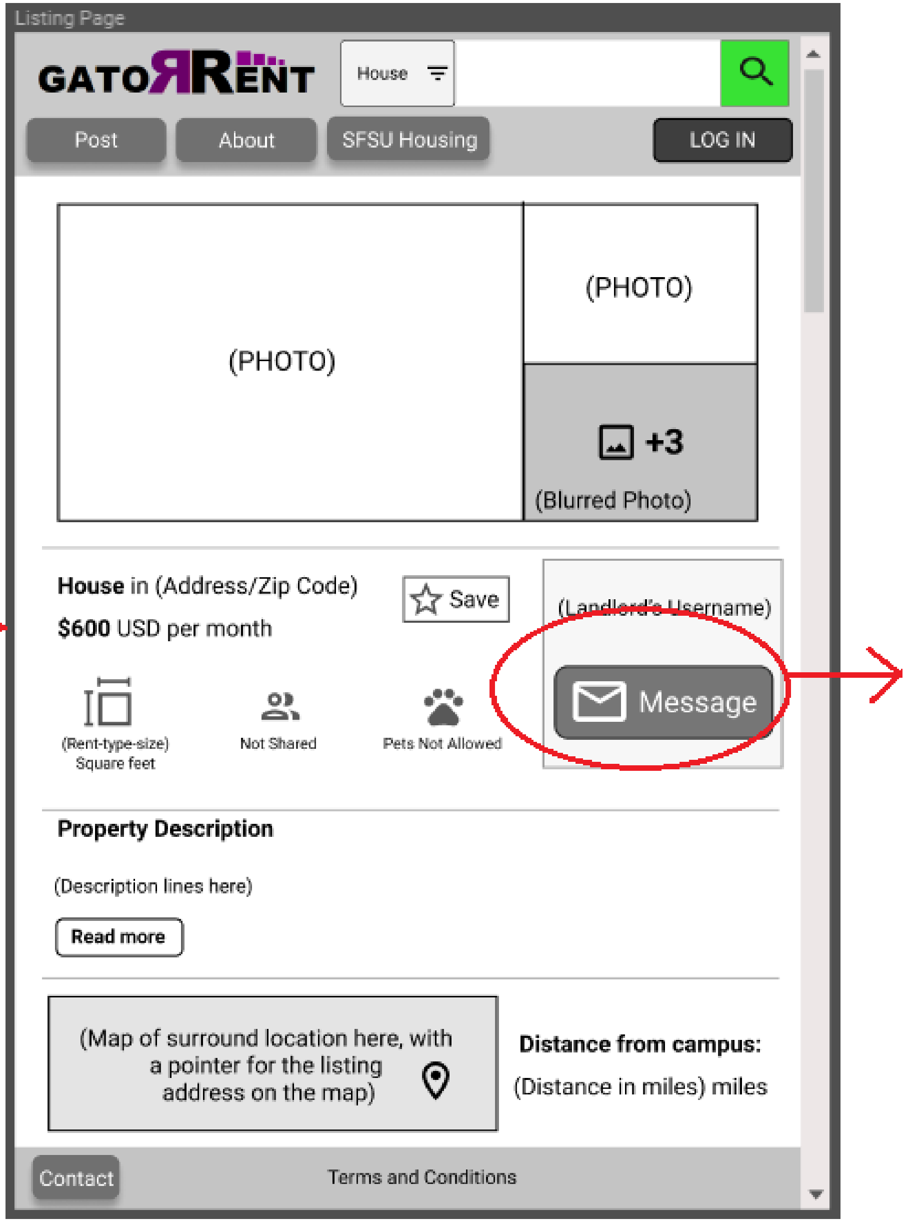
$1700 USD per month
(Rent Type)
(Location)
(Distance from Campus)

Results Page



1-3 of 3 results for **House** in **(search bar value)**

Listing Page



House in (Address/Zip Code)

$600 USD per month

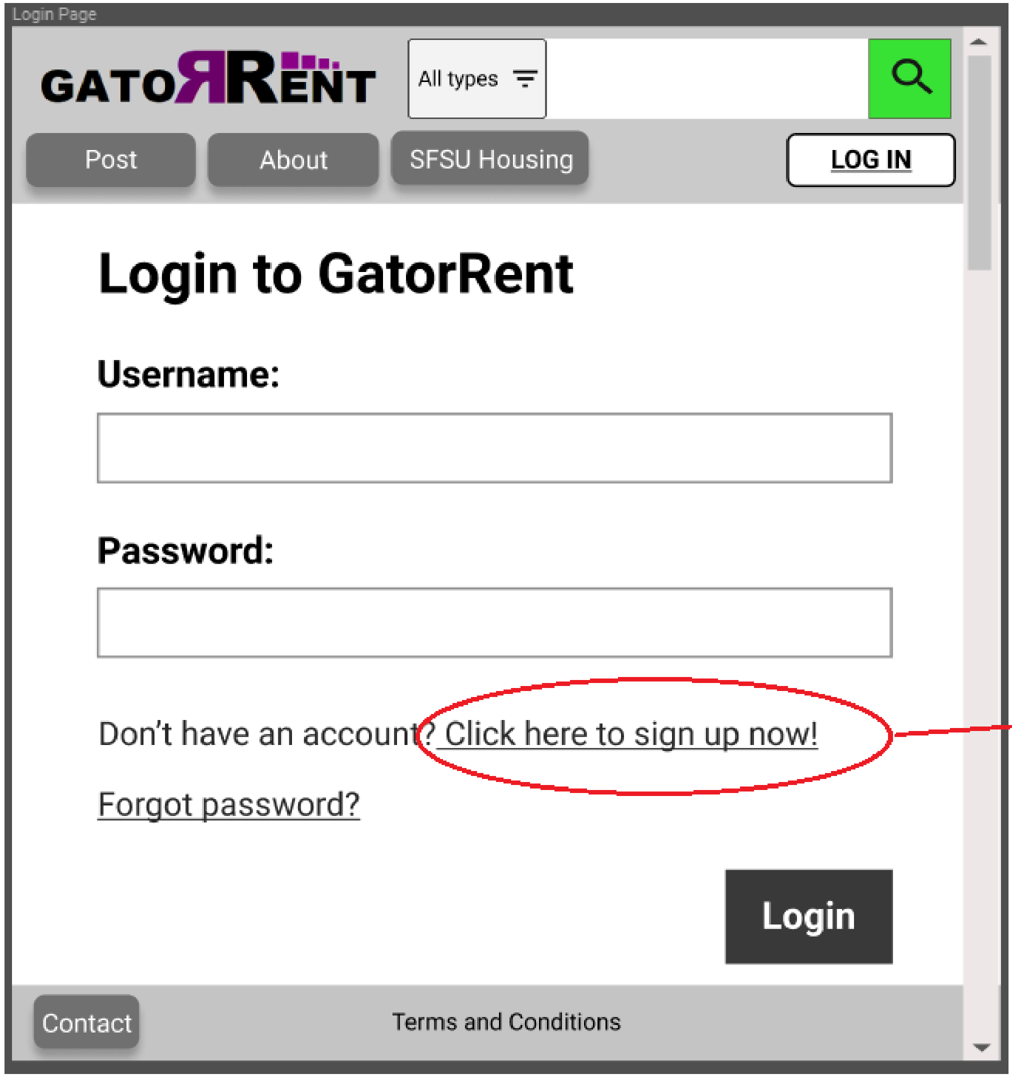
Property Description

(Description lines here)

(Map of surround location here, with a pointer for the listing address on the map)

Distance from campus:

(Distance in miles) miles



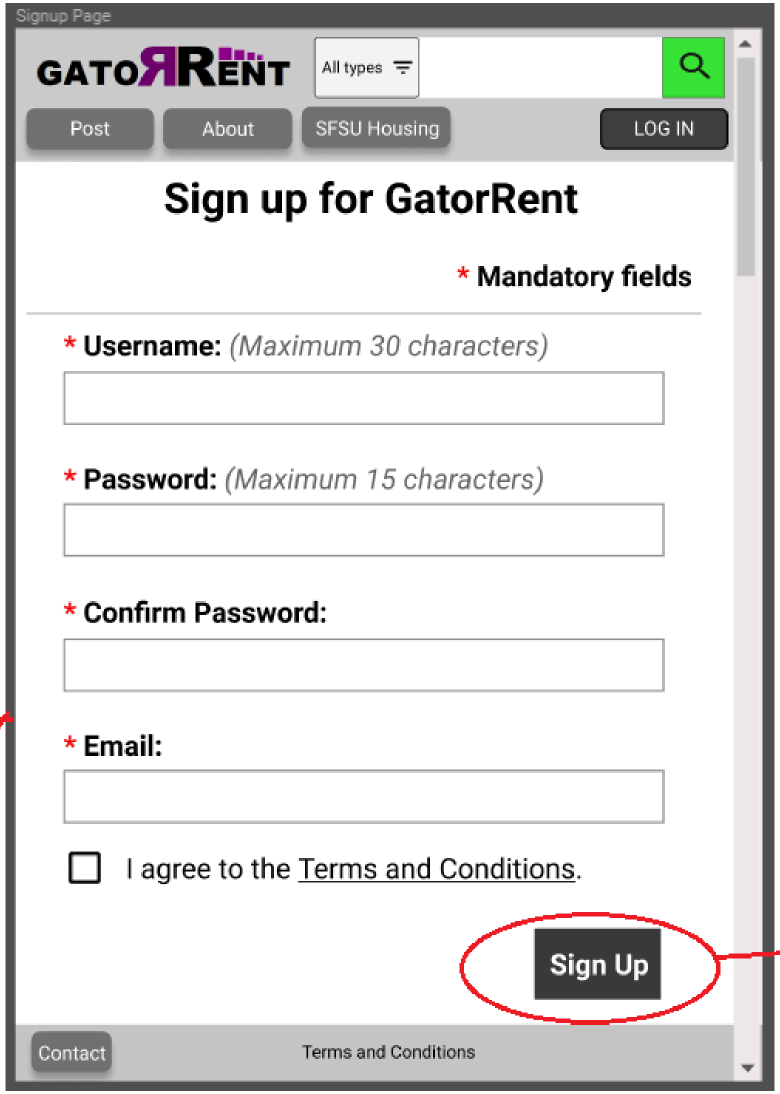
Login to GatorRent

Username:

Password:

Don’t have an account? Click here to sign up now!

Forgot password?



Sign up for GatorRent

* Mandatory fields

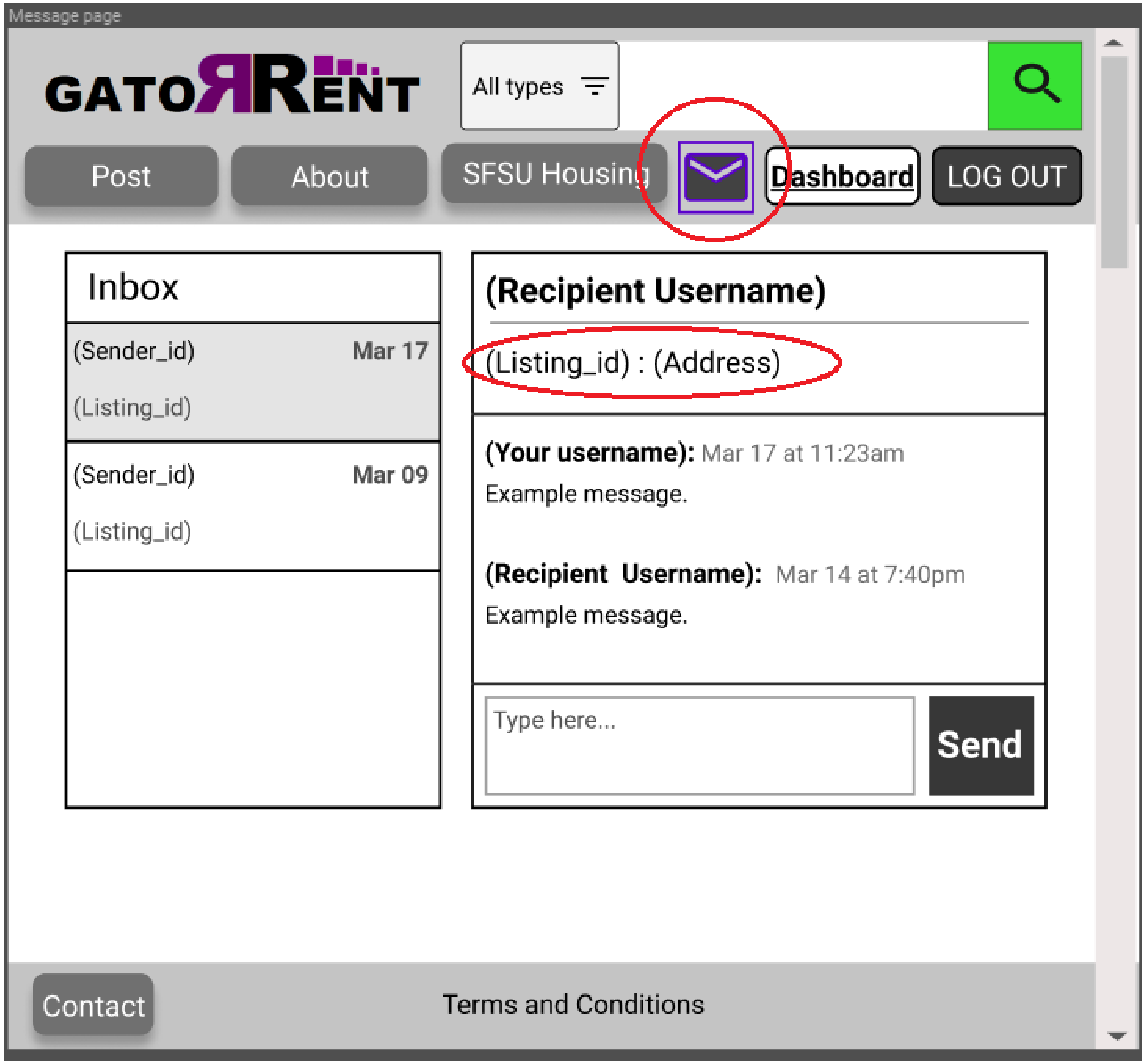
* Username: *(Maximum 30 characters)*

* Password: *(Maximum 15 characters)*

* Confirm Password:

* Email:

□ I agree to the Terms and Conditions.



Inbox

(Sender_id) Mar 17
(Listing_id)

(Sender_id) Mar 09
(Listing_id)

(Recipient Username)

(Listing_id) : (Address)

(Your username): Mar 17 at 11:23am
Example message.

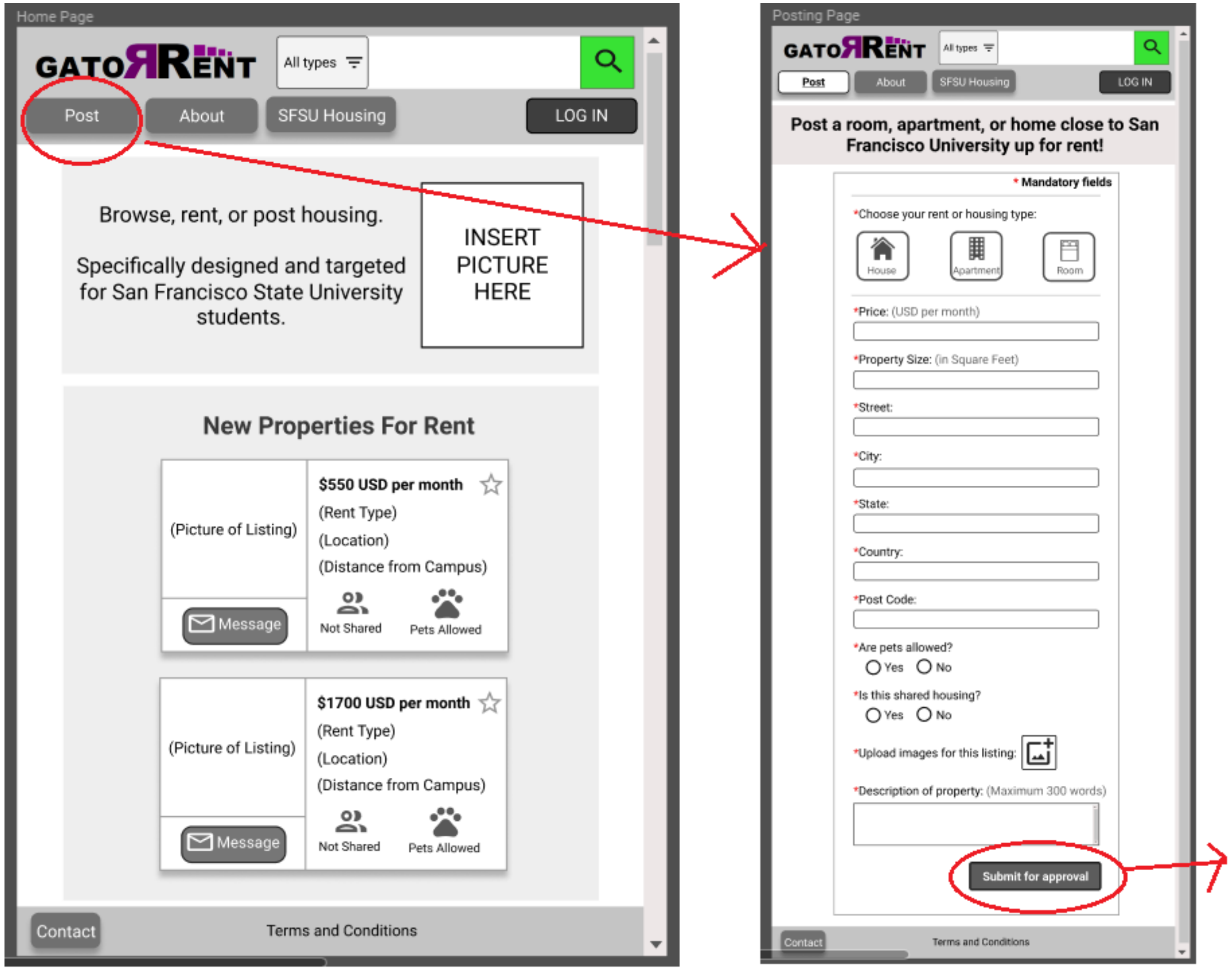
(Recipient Username): Mar 14 at 7:40pm
Example message.

Type here...

Send

2. **Use Case #2**

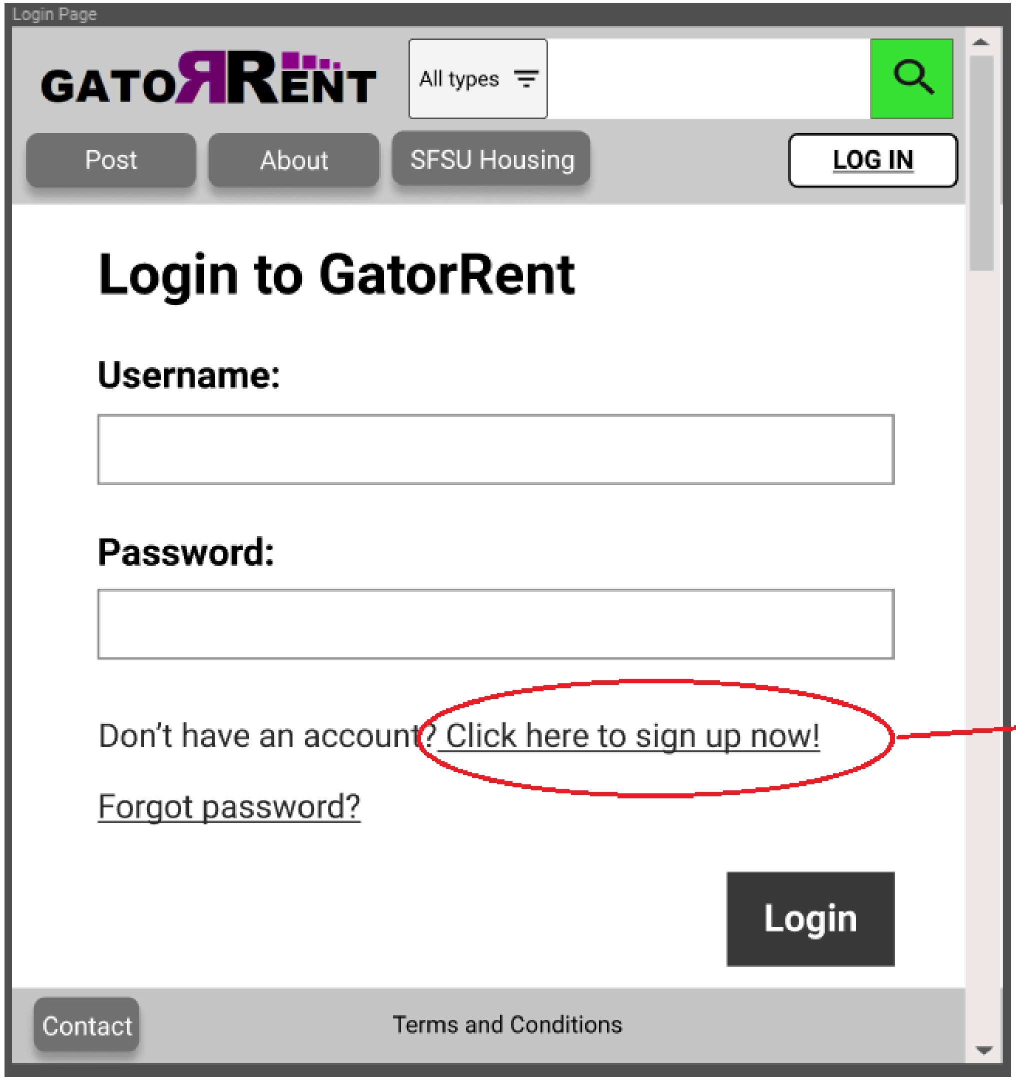
   a. Post place for rent (rent-type:house/apartment/room, rent-type-size?insquarefeet, price:permonth, distance:fromcampus, postcode, pets?, shared? description, images), prompted to log in, submit post for admin approval.



Browse, rent, or post housing.

Specifically designed and targeted for San Francisco State University students.

Post a room, apartment, or home close to San Francisco University up for rent!



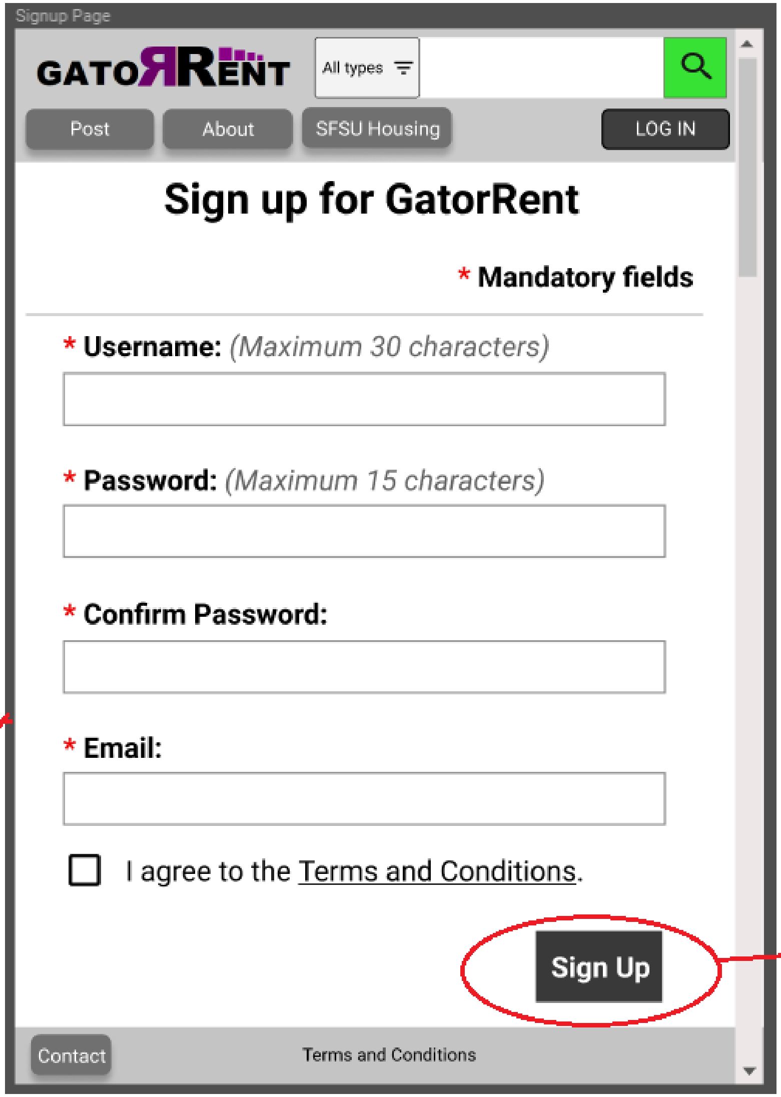
Login to GatorRent

Username:

Password:

Don't have an account? Click here to sign up now!

Forgot password?



Sign up for GatorRent

* **Mandatory fields**

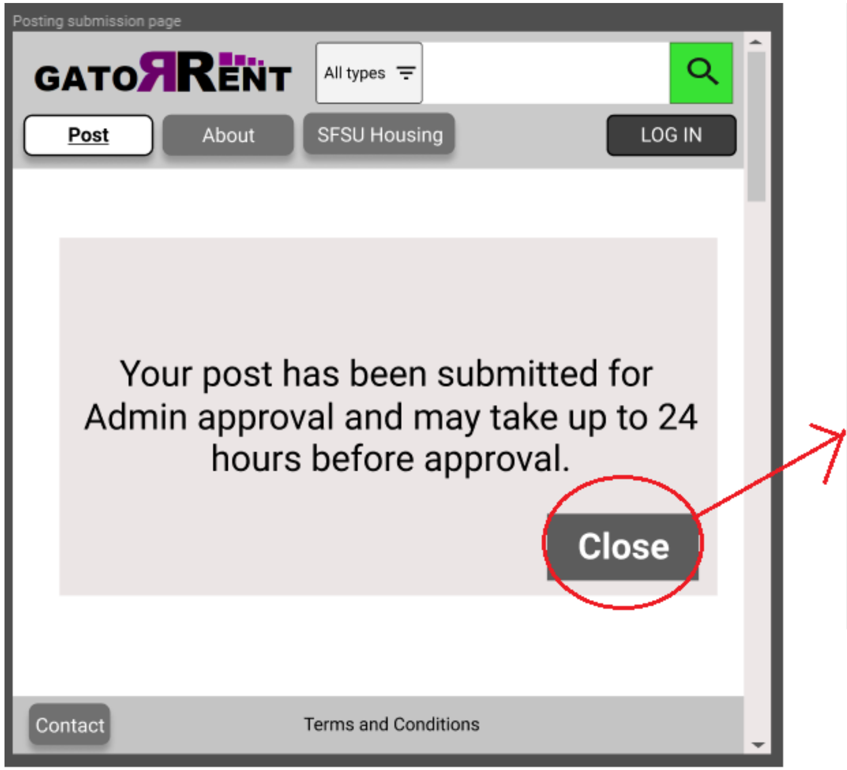
* **Username:** *(Maximum 30 characters)*

* **Password:** *(Maximum 15 characters)*

* **Confirm Password:**

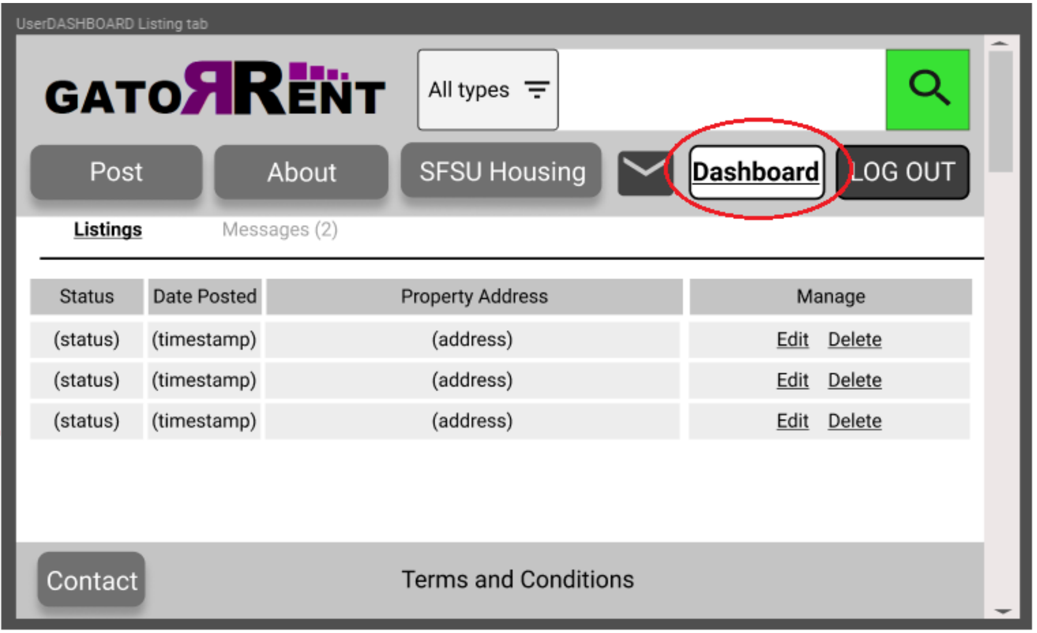
* **Email:**

□ I agree to the Terms and Conditions.



Posting submission page

Your post has been submitted for Admin approval and may take up to 24 hours before approval.

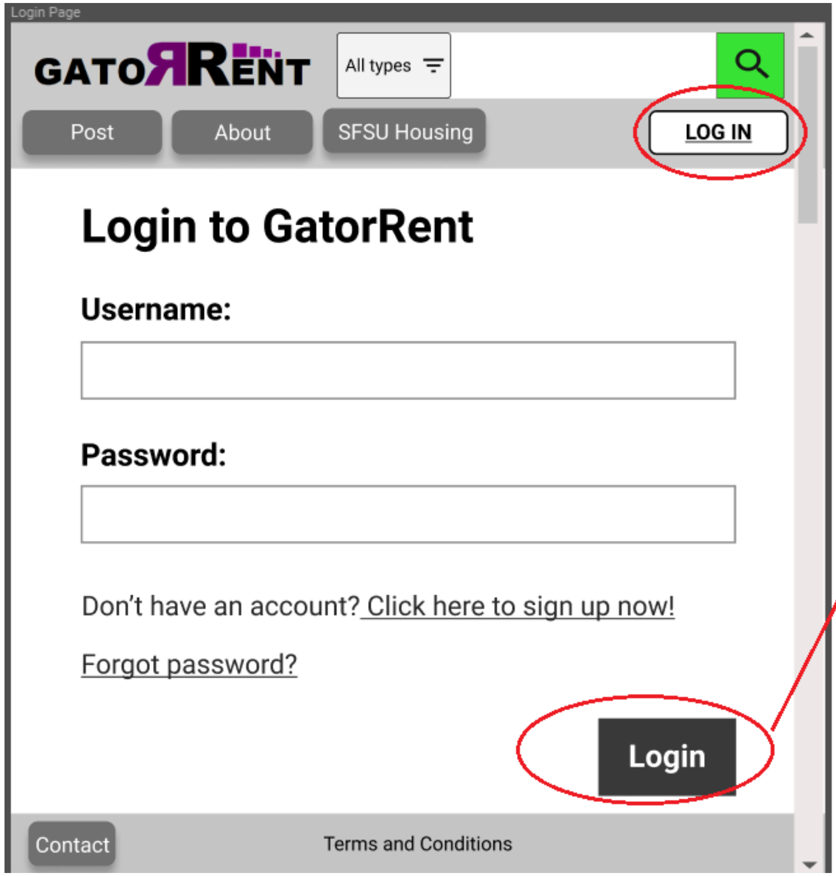


UserDASHBOARD Listing tab

Status	Date Posted	Property Address	Manage
(status)	(timestamp)	(address)	Edit Delete
(status)	(timestamp)	(address)	Edit Delete
(status)	(timestamp)	(address)	Edit Delete

3. **Use Case #3**

   a. User login dashboard and messaging



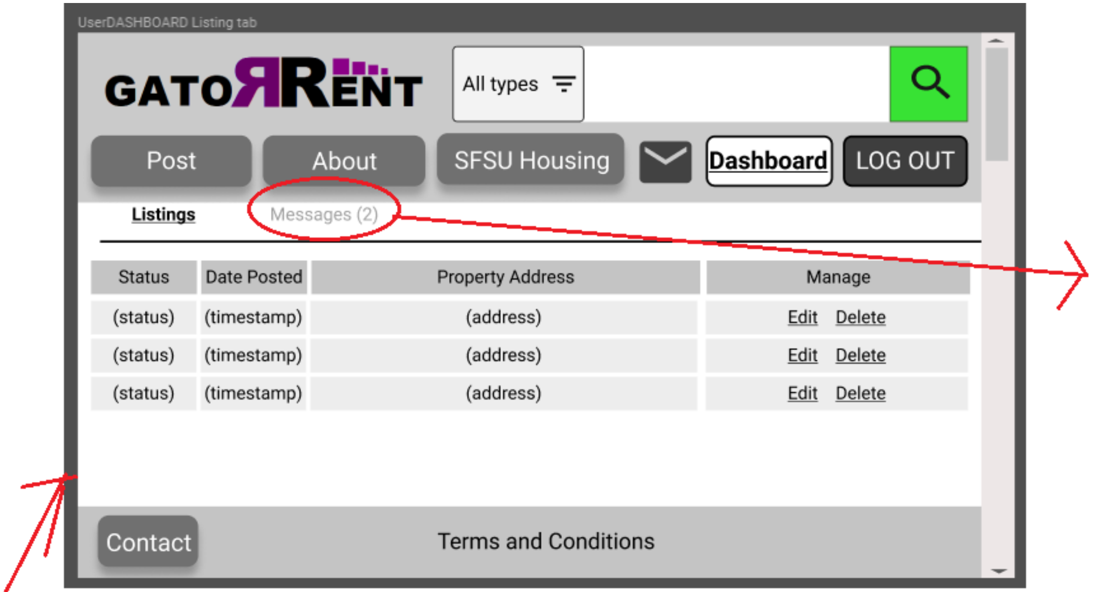
Login to GatorRent

Username:

Password:

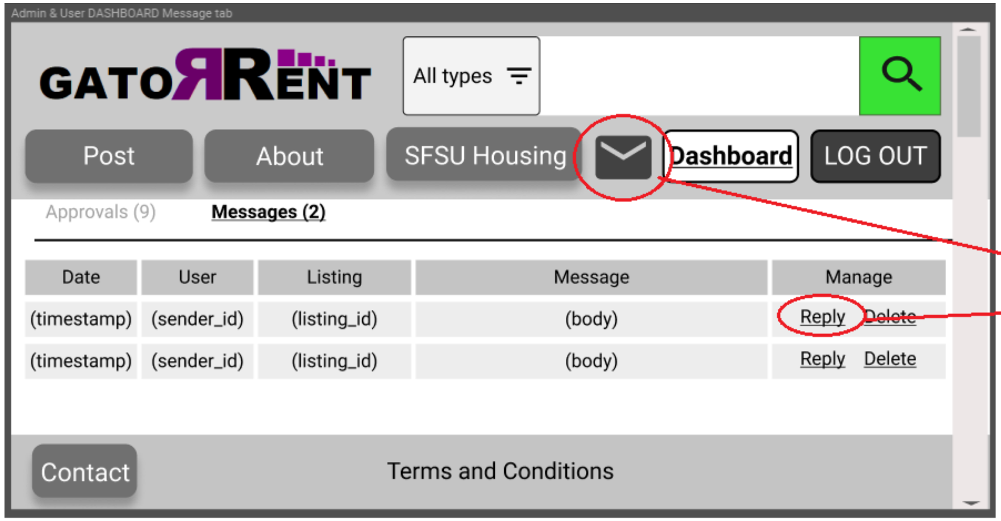
Don’t have an account? Click here to sign up now!

Forgot password?



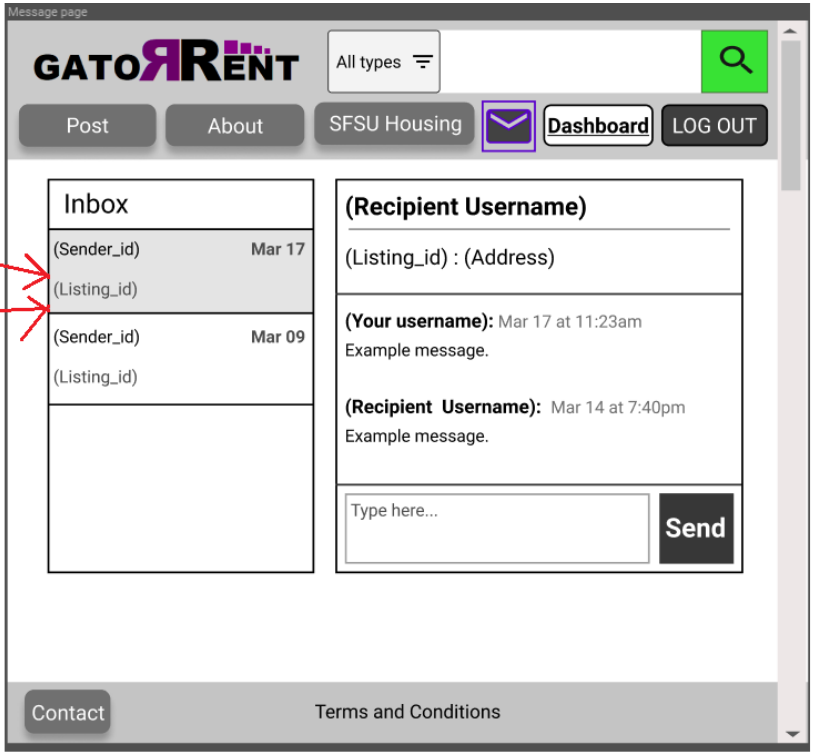
Status	Date Posted	Property Address	Manage
(status)	(timestamp)	(address)	Edit Delete
(status)	(timestamp)	(address)	Edit Delete
(status)	(timestamp)	(address)	Edit Delete

Admin & User DASHBOARD Message tab



Date	User	Listing	Message	Manage
(timestamp)	(sender_id)	(listing_id)	(body)	Reply Delete
(timestamp)	(sender_id)	(listing_id)	(body)	Reply Delete

Message page



(Sender_id) (Listing_id)	Mar 17
(Sender_id) (Listing_id)	Mar 09

(Recipient Username)

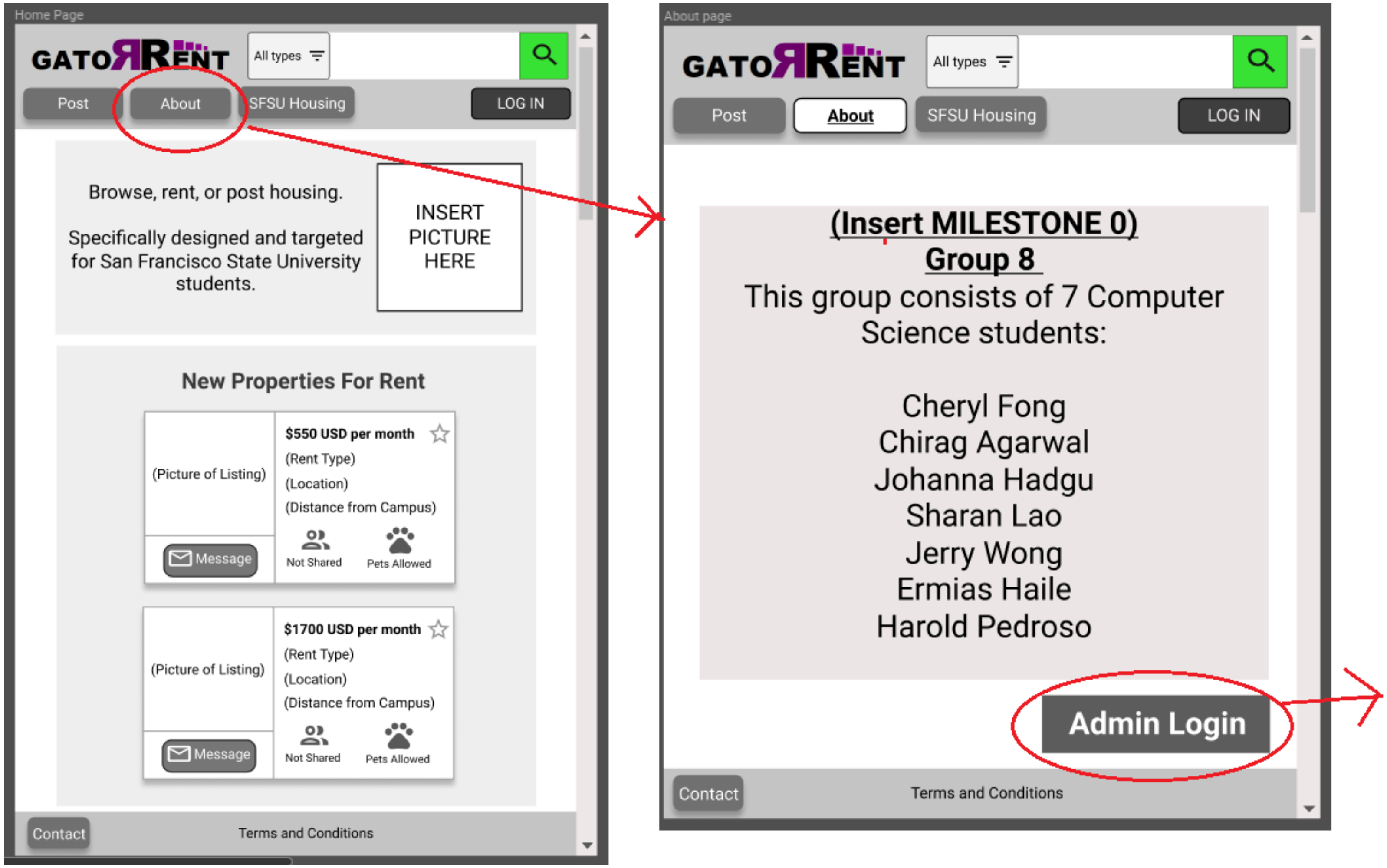
(Listing_id) : (Address)

(Your username): Mar 17 at 11:23am
Example message.

(Recipient Username): Mar 14 at 7:40pm
Example message.

4. **Use Case #4**

   a. Admin logging in, gatorrent dashboard layout, show data list that includes listings that need approval + user-abuse reports



Browse, rent, or post housing.

Specifically designed and targeted for San Francisco State University students.

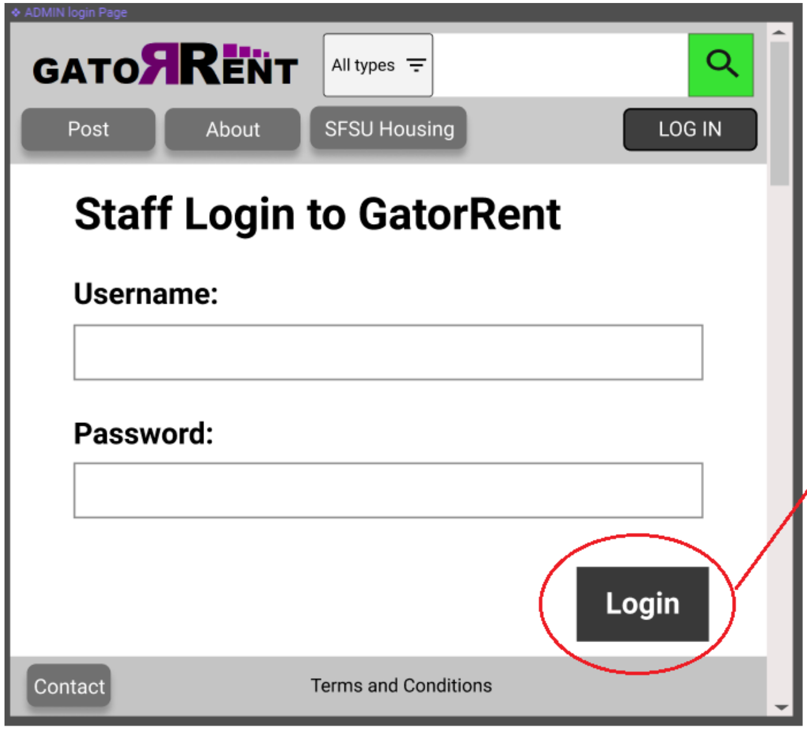
New Properties For Rent

$550 USD per month
(Rent Type)
(Location)
(Distance from Campus)

$1700 USD per month
(Rent Type)
(Location)
(Distance from Campus)

(Insert MILESTONE 0)
Group 8
This group consists of 7 Computer Science students:

Cheryl Fong
Chirag Agarwal
Johanna Hadgu
Sharan Lao
Jerry Wong
Ermias Haile
Harold Pedroso

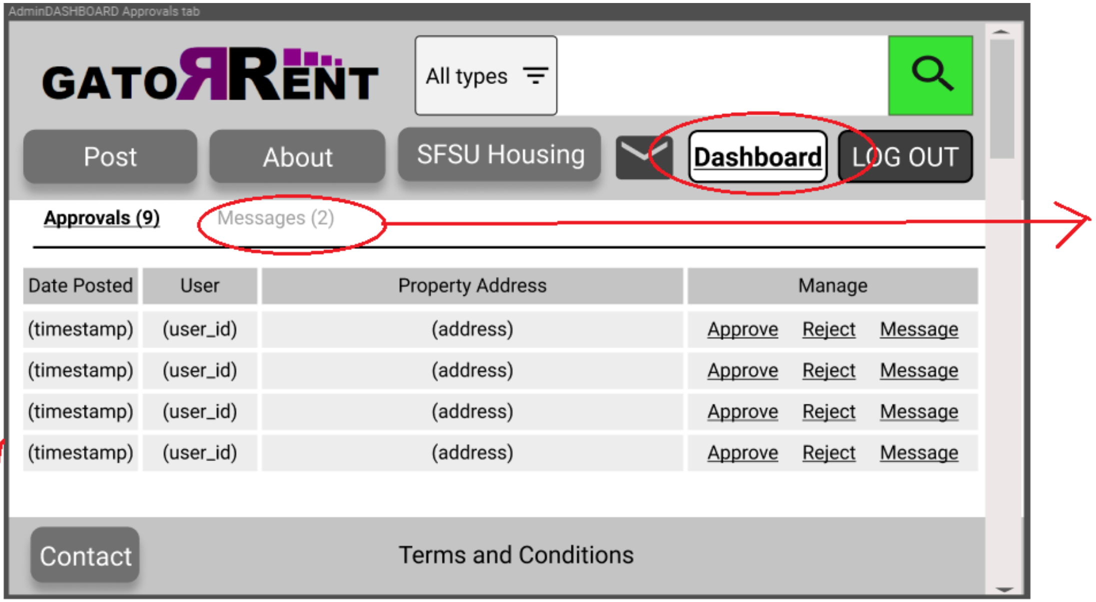


ADMIN login Page

Staff Login to GatorRent

Username:

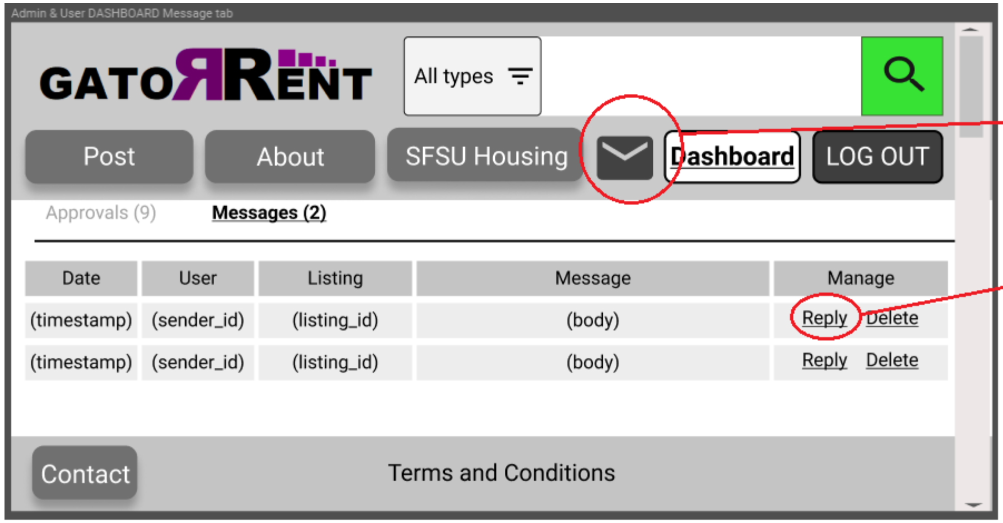
Password:



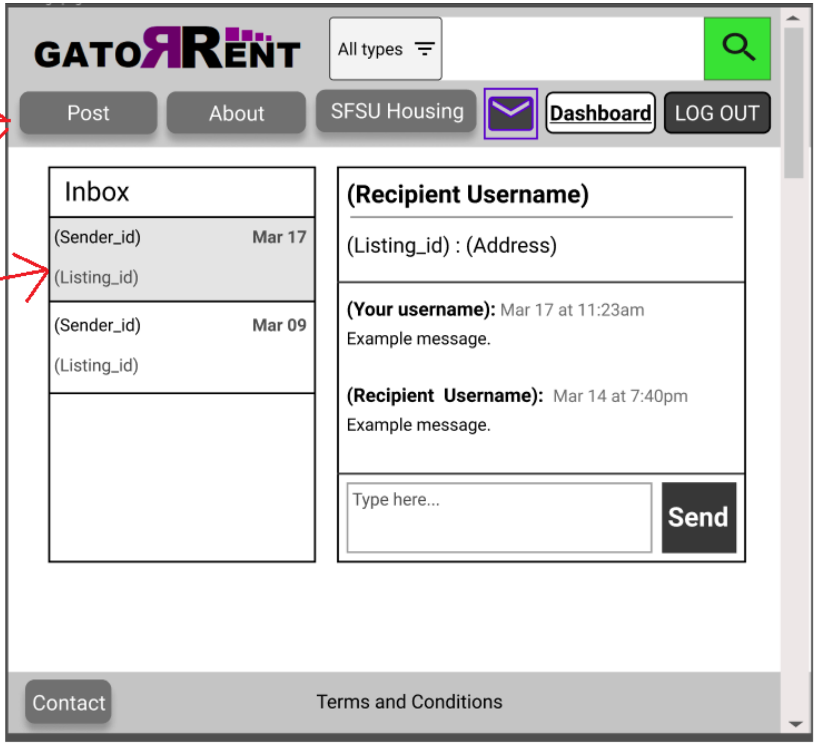
AdminDASHBOARD Approvals tab

Approvals (9) Messages (2)

Date Posted	User	Property Address	Manage
(timestamp)	(user_id)	(address)	Approve Reject Message
(timestamp)	(user_id)	(address)	Approve Reject Message
(timestamp)	(user_id)	(address)	Approve Reject Message
(timestamp)	(user_id)	(address)	Approve Reject Message



Date	User	Listing	Message	Manage
(timestamp)	(sender_id)	(listing_id)	(body)	Reply Delete
(timestamp)	(sender_id)	(listing_id)	(body)	Reply Delete



Inbox

(Sender_id) (Listing_id)	Mar 17
(Sender_id) (Listing_id)	Mar 09

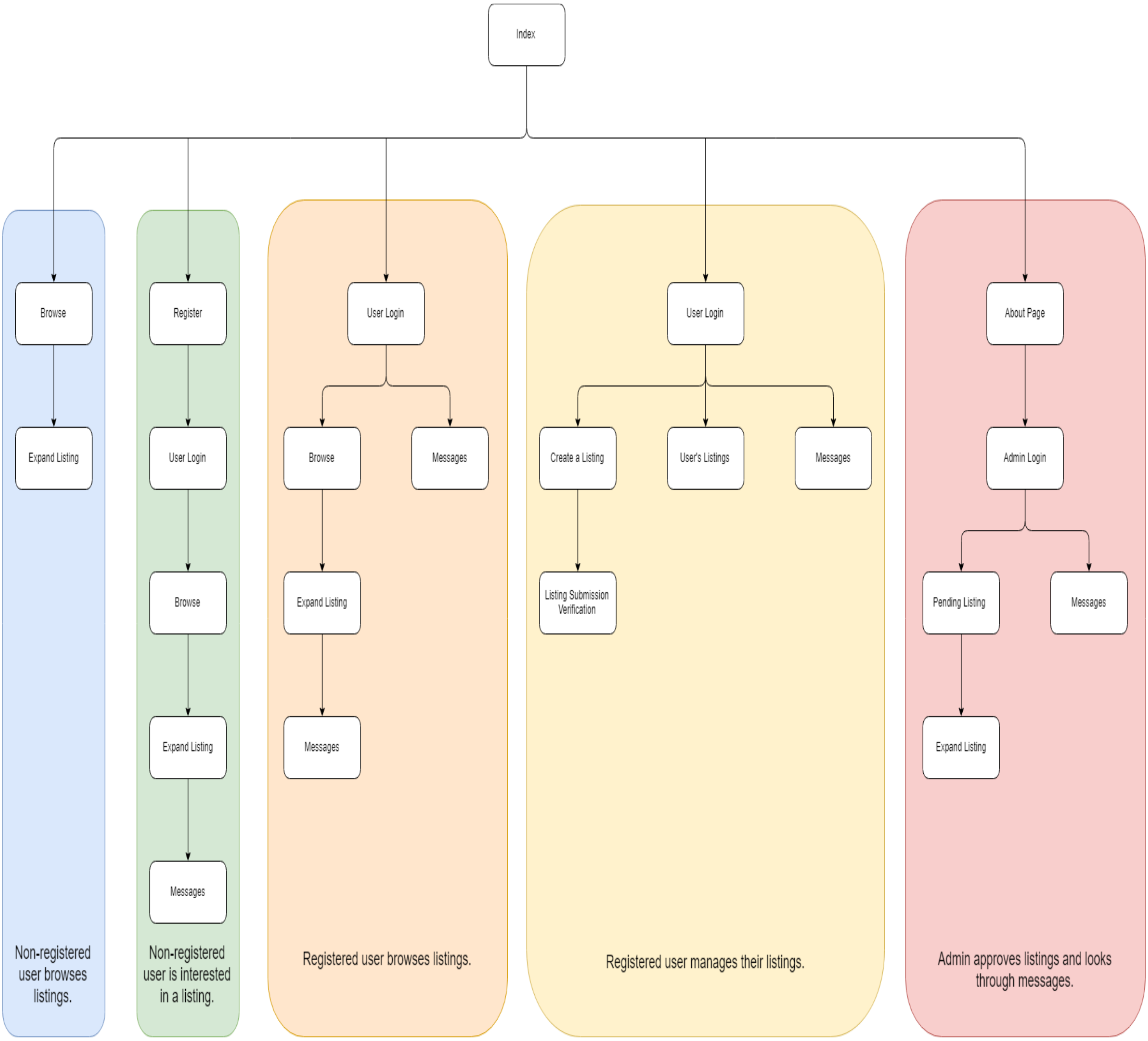
(Recipient Username)

(Listing_id) : (Address)

(Your username): Mar 17 at 11:23am
Example message.

(Recipient Username): Mar 14 at 7:40pm
Example message.

3.3 Use Case Flow Chart


Index
Browse
Expand Listing
Non-registered user browses listings.
Register
User Login
Browse
Expand Listing
Messages
Non-registered user is interested in a listing.
User Login
Browse
Messages
Expand Listing
Messages
Registered user browses listings.
User Login
Create a Listing
User's Listings
Messages
Listing Submission Verification
Registered user manages their listings.
About Page
Admin Login
Pending Listing
Messages
Expand Listing
Admin approves listings and looks through messages.

Expand Image for Use Case Flow Chart

4. HIGH LEVEL ARCHITECTURE, DATABASE ORGANIZATION

4.1 Database Organization (continued on 4.6):

4.1.1 We have a MySQL database which stores all the information in the form of different table entries. There are various tables such as a user table consisting user information, listing table consisting posting details and several others tables explained in the schema.

4.1.2 Database Schema:

4.1.2.1 The database contains six tables:

i. **User:** This table will contain information on all registered users, such as their **id**, **username**, **password**, **email**, and **user_type**. The password will be encrypted. Each entry creation is marked with a timestamp.

ii. **Listing:** This table will contain information of all listings such as a listing's **id**, **user_id** (foreign key to use table), its **description**, **rent-type_id** (foreign key to rent-type table), **price** (monthly rent amount), **distance_campus** (distance away from campus). Additional, columns that specify if the listing allows pets, and if its a shared property is recorded under the i**sPet** and **isShared** columns respectively. **Status** column specifies if the listing is still a draft, if the listing is currently open (unoccupied), under maintenance, or occupied and thumbnail. This table also contains the listing information, such as the **street**, **city**, **state**, **country**, and the coordinates of the location i.e. **lat** for latitude and **long** for longitude. Each entry creation and updates are marked with a timestamp.

iii. **Rent-type:** This table serves as a point for reference for various **type**(s) of rentable properties to the **Listing** table such as room, apartment and house (where rent-type_id is the foreign key to listing table).

iv. **Images**: This table will contain the image information in the form of the path of that file.

v. **Message:** This table will contain all the information pertaining the messages sent users of the website. Each message entry will contain the **body** of the message, the **sender_id** (id of the user who sent the message), and **listing_id** (id of the listing entry that this message is inquiring about). Each entry creation is marked with a timestamp.

[continued on Section 4.6]

4.2 Media Storage:

4.2.1 Images will be stored on the file system where the Images table where each entry will contain paths pointing to a folder of images belonging to a Listing entry. Images can be in the format, png, gif, jpeg or jpg.

4.3 Search/Filter Architecture and Implementation:

4.3.1 Search: The search will be implemented by using the LIKE statement combined with the '%' wildcard or pattern matching on strings. MySQL uses B-trees to queries the specified terms.

4.3.2 Implementation: For example, users can filter listing results to show results that matches the street field that they input, and where pets are allowed: SELECT * FROM listing WHERE MATCH (street) AGAINST ('keyword1) AND listing.isPet=1

4.4 API Endpoints:

4.4.1 Get all listing results, newest to oldest e.g. ip-address/listings/

4.4.2 Get listing results based on user query e.g. ip-address/listings?<query-string>

4.4.3 Get all listings that need approval e.g. ip-address/listings-to-approve/

4.5 New SW Tools/Frameworks:

4.5.1 As per discussion with Professor Anthony Souza, we are allowed to use an S3 Bucket to store images on AWS. However, we have decided as a team that we will use the Images table instead as described on item 4.1.2.1.iv above.

4.5.2 We will also use Google Map and Google Analytics APIs.

4.6 Visual Representation of Database Organization

Gatorrent-db – This database will contain the following tables displayed below:

USER		
Column	Type	Attributes and Constraints
id	int	PRIMARY KEY IDENTITY(1,1) NOT NULL
username(unique)	varchar(30)	UNIQUE NOT NULL
password	varchar(15)	NOT NULL
email	varchar(40)	NOT NULL
user_type	tinyint	
timestamp	timestamp	DEFAULT CURRENT_TIMESTAMP NOT NULL

LISTING		
Column	Type	Attributes and Constraints
id	int	PRIMARY KEY IDENTITY(1,1) NOT NULL
user_id	int	FOREIGN KEY(user_id) REFERENCES users(id)
description	text	
rent-type_id	int	FOREIGN KEY(rent-type_id) REFERENCES rent-type(id)
price	double	NOT NULL
distance_campus	float	
isPet	boolean	NOT NULL
isShared	boolean	NOT NULL
status	varchar	
street	varchar	
city	varchar	
state	varchar	
country	varchar	
postcode	int	
lat	float(10.6)	NOT NULL
long	float(10.6)	NOT NULL

isApproved	boolean	DEFAULT 0
image_id	int	FOREIGN KEY (image_id) REFERENCES images(id) NOT NULL
thumbnail_path	varchar	NOT NULL
timestamp	timestamp	DEFAULT CURRENT_TIMESTAMP NOT NULL

RENT-TYPE		
Column	Type	Attributes and Constraints
id	int	PRIMARY KEY IDENTITY(1,1) NOT NULL
type	varchar(40)	UNIQUE NOT NULL

IMAGES		
Column	Type	Attributes and Constraints
id	int	PRIMARY KEY IDENTITY(1,1) NOT NULL
image_path	varchar	NOT NULL

Message		
Column	Type	Attributes and Constraints
id	int	PRIMARY KEY IDENTITY(1,1) NOT NULL
body	text	

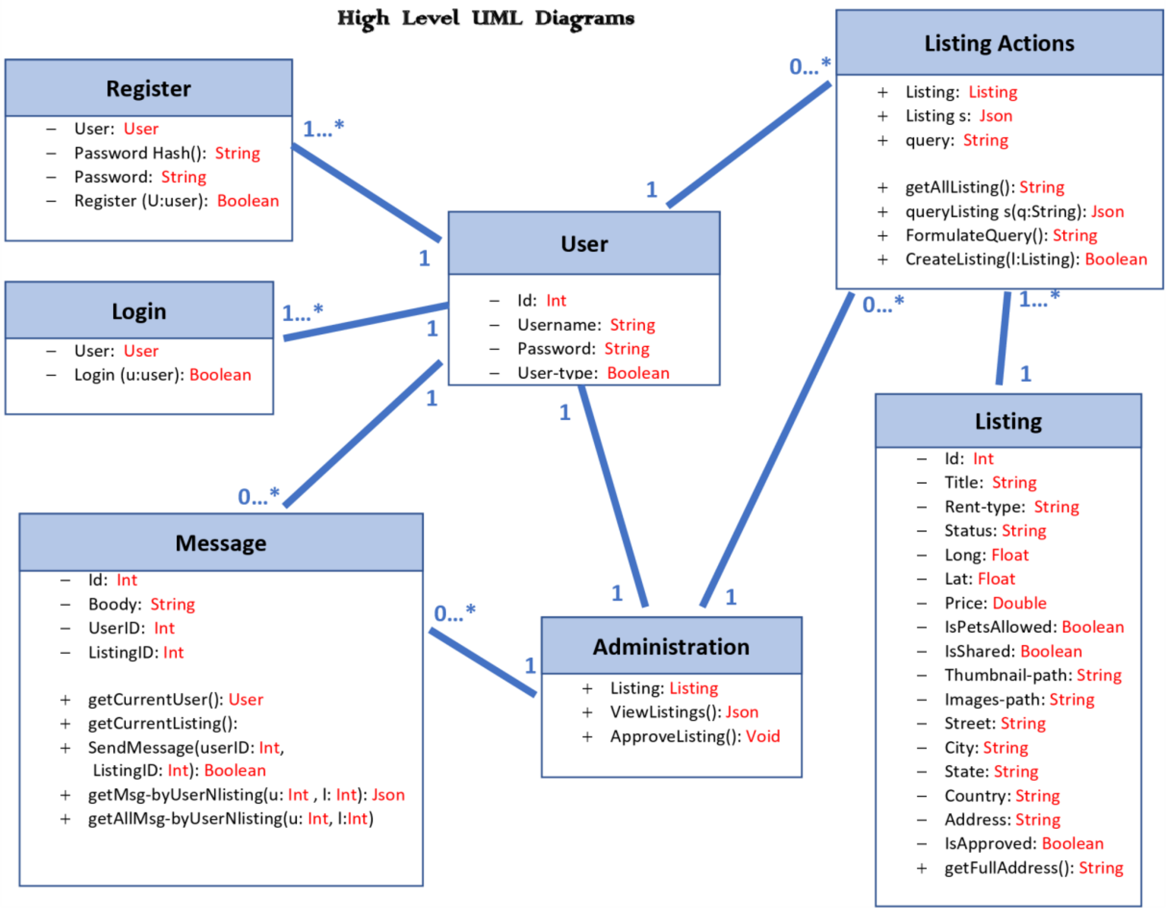
sender_id	int	FOREIGN KEY (sender_id REFERENCES user(id) NOT NULL
listing_id	int	FOREIGN KEY (listing_id REFERENCES listing(id) NOT NULL
timestamp	timestamp	DEFAULT CURRENT_TIMESTAMP NOT NULL

4.6.1 Website Meta-Data:

user-session — To keep track of the user’s state when using the website
user-favourites — To keep a list of URLs of the user’s saved listings
user-preference — To keep track of a user’s preferences
activity-log — A log of entries added or removed from databases

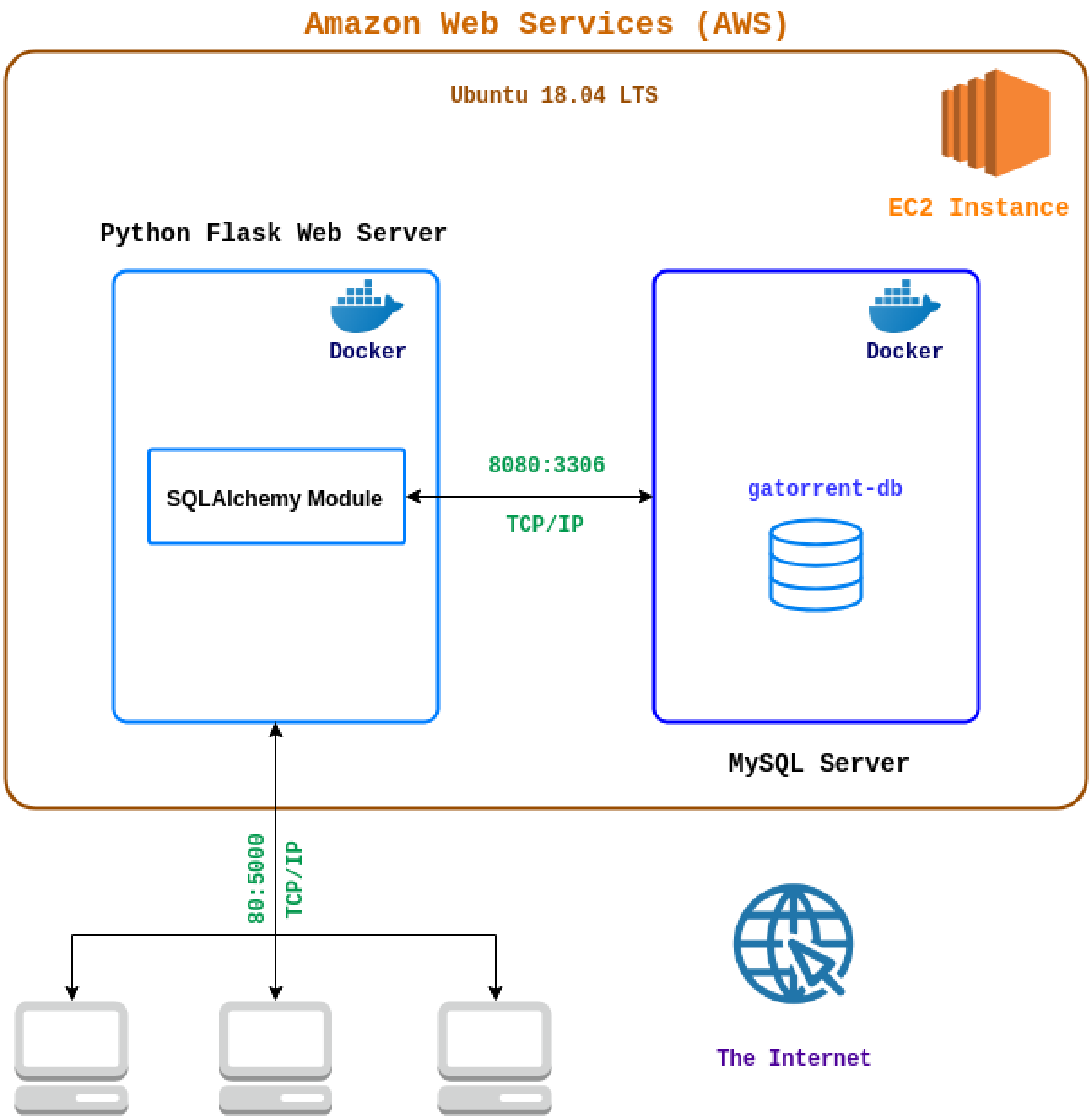
5. HIGH LEVEL UML DIAGRAMS

5.1. High Level UML Class Diagram:



Expand image for High Level UML Class Diagram

5.2. UML Component and Deployment Diagram:



6. ACTUAL KEY RISKS

6.1 Skill Risks

As a group, we have varying skills. Yet a majority of the group are first being exposed to using Git as a command line utility for version control and GitHub in this course. Therefore, one of the key issues that have arisen is ensuring that everyone is comfortably adjusting to using Git and GitHub. Some are still confused with the concept of branches, pull requests, and pushing the correct content to the right branches. In order to resolve this, our team leader has provided excellently well-documented instructions to the group repository that clearly outlines the correct commands necessary for execution, as well as the reason why we are doing it.

Moreover, familiarity with using Python and the web microframework Flask is part of our team's skills risk. For example, the usage of built in template tags and filters, various python libraries e.g. boto, manipulating and retrieving information in the form of JSON from the database.

One way to resolve our skills risks is to spend a sufficient amount of our own individual time to research the concepts we are struggling with, whether it's watching youtube videos online or finding online textbooks. If the confusion still persists, it's also important to ask team members for assistance.

6.2 Schedule Risks

The team has collectively agreed to meet on Thursday mornings. However, we have experienced several issues with meeting on other days due to conflicting schedules. We found the best possible solutions are to first, prioritize the time allocated in class for working on the group project while being respectful of everyone's time. Second, to make strides in being a collaboratively distributed, yet effective team. If neither of these are possible, we plan to utilize online services such as Skype, Google Hangouts, or Zoom to meet.

6.3 Technical Risks

One of the key technical risks our team faces is understanding the working components of the development environment particularly on Docker usage.

Docker was chosen as part of the development environment and as well as the deployment foundation because it is one of the leading components in modern web microservices and for software build/test environments. Milestone 0 was a great opportunity to introduce the idea of containerization and learn about industry web technologies. It was a refresher to some and a brand-new topic to others. All in all, the team got to learn about a hot industry trend and potentially add a new skill to their bag of tricks to show off to their future employers.

To help ease the transition in using Docker, installation instructions were given to the team and conducted together as a team activity. There are parts of the installation that individual team members did not understand, and understanding what is going on when the development environment is launched (i.e. understanding log outputs) may come across as difficult. The type of commands to use and tweaking the environment to the preference of the individual can seem very unfamiliar at this point.

To address these risks, detailed documentation has been placed on the team's GitHub repository for quick reference. Additionally, there's an abundance of online resources to take advantage of and as well as a welcoming Docker user community to reach out to for help via Docker's GitHub and Slack (and even on IRC). Overall, with patience and practice, using Docker will eventually become second nature.

Another technical risk that the team faces is the usage of Amazon Web Services (AWS). AWS offers an easy to use interface, however, it will take time to get use to and understand the technical terms used by AWS.

In order to reduce the severity of these risks, it is up to the team members to put time into reading, research and be willing to ask for help.

6.4 Teamwork Risks

The prominent risks that we face as a team are inconsistent usage of our dedicated group Slack channel and GitHub pull requests and issues. One way to mitigate these risks is to have all team members install the Slack app on their mobile devices to get notifications instantly and check our team's GitHub repository regularly. The best solution is to tackle the problem that arises as soon as possible.

Additionally, it's key that we address the individual(s) who aren't putting the time into their tasks and/or group activities in a respectful manner. Asking questions and offering help whenever and wherever need is highly encouraged among team members. The goal is to maintain a level of open-mindedness, transparent communication and dedication to ensure productivity and to mitigate confusion and miscommunication.

6.5 Legal/Content Risks

Any legal or content issue that may arise is the presence of users that may be attempting to use this website for malicious purposes. For example, users may post fake listings and bait users into renting a fake property offline. Our message feature will allow users to contact the site administrators about any suspicious listings. Messages containing such reports will arrive at the admin dashboard. It will be up to the site administrators to dispute such claims and take appropriate action such as taking a listing off the browsing results and/or ban a user by placing the user’s email on a blacklist.

Additionally, any acts committed by registered users who violate the terms and conditions agreement will be acted upon accordingly by the site administrators.

7. PROJECT MANAGEMENT

In order to promote efficient workflow that is both timely and productive, we have collectively decided to use the GitHub Projects Board as an alternative to Trello. It is comprised of issues, pull requests, and notes that are organized according to the user. It will allow us to keep track of tasks and bugs as well as enhancements to our project. For this specific project, the Projects Board will allow us to assign certain individuals with specific tasks and deadlines. It will also enable us to organize our tasks on a higher level. Once an individual has been assigned a task, they will receive an email notification, alerting them of the task description, deadline, and people working on this issue. Although our team has been using slack as a means of communication, the Projects Board provides a clear and concise outline for the group to reference throughout the semester.

All in all, for Milestone 2 and future tasks, our Projects Board will allow us to reduce the amount of confusion on what needs to be done by whom. Every assigned task, whether closed or opened will remain on GitHub, allowing us to keep a log of who did what. Additionally, we will be able to add labels to help organize our issues as well as include any dependencies amongst current or future tasks. This will be helpful as our tasks get increasingly greater in number as the days progress and promote efficient workflow.