SW Engineering CSC648/848 Section 01 Spring 2019
Milestone 2
SAHARA LLC.
Oasis
Team 07

Team Members:

Ratna Lama:	Team Leader	*rlama7@mail.sfsu.edu*
Ade Adetayo:	Front End Lead	
Andrew Sarmiento:	Back End Lead	
Adam Tremarche:	Github Master	
Hadia Andar	UI / UX Designer	
Shuyuan Deng	Database Master	

3/21/19

Version History:

3/21/19: document submitted for review

Content and structure for Milestone 2 document for review:

1. **Data Definitions Ver. 2**
2. **Functional Requirements Ver. 2**
3. **UI Mockups and Storyboards**
4. **High-Level Architecture, Database Organization**
5. **High-Level UML Diagrams**
6. **Identify Key Risks**
7. **Project Management**

1. Data Definitions Ver. 2:

1.1 Unregistered_User:
The user has not registered. The Unregistered_User can only browse the web pages.

1.2 Registered_User:
The Registered_User can post and edit rental information, upload images, send and receive a message from other Registered_User.

1.3 Administrator:
The user can review and manage the posts on Oasis. The Administrator can approve and delete the posts, but unable to edit the posts.

1.4 User_Dashboard:
The Registered_User can view and edit their profile, saved favorite and posted properties list, and messages on the dashboard.

1.4.1 **User_Profile**:
We shall collect basic user profile such and first name, last name, and email from the registered users. Registered_User can update the information as they wanted.

1.4.2 **Favorite**:
The list can store the registered user's favorite properties links. The user can keep tracking and comparing different properties.

1.4.3 **Post_Property**:
The list stores the properties that post by the Registered_User, and shows the status of pending, approved, available or unavailable.

1.4.4 **Messages**:
The list can store the information of messages between the registered users. Only the associated senders and receivers can read the messages.

1.5 Admin_Dashboard:
The Administrator can view, delete, and approve the pending post properties. The Messages function is also applied.

1.6 Login:
For registered user and administrator to log in.

1.7 Logout:
For login registered user and administrator to log out.

1.8 Register:
For an unregistered user to register as the registered user. The registration process shall allow the unregistered user to create an account by creating the username, password and filling out the valid first name, last name, and email.

1.8.1 **User_Name**:
A name created by the user.

1.8.2 **User_Password**:
The password created by the user.

1.8.3 **User_Firstname**:
User’s first name.

1.8.4 **User_Lastname**:
User’s last name.

1.8.5 **User_email**:
User’s email.

1.9 **Property**:

The list can store the information of properties listed on Oasis. The property list is further subdivided as follows:

1.9.1 **Property_Type**:

The type of property, such as Room, Apartment, and House.

1.9.2 **Property_Image**:

The image archive can store the image of properties on Oasis. The image can be loaded as necessary.

1.9.3 **Property_Price**:
The price of rent.

1.9.4 **Property_Description**:
The description of the posted property. (Optional)

1.9.5 **Property_Address**:
The Address of the posted property location.

1.9.6 **Property_ID**:
The assigned ID number of the posted property.

1.9.7 **Property_Date**:
The beginning posted date of the property.

1.9.8 **Property_Room**:
The number of rooms in the property.

1.9.9 **Property_Bathroom**:
The number of bathrooms in the property.

1.9.10 **Property_Size**:

The size of the property.

1.9.11 **Property_Status**:
The availability of the Property.

1.10 Property_Filter:

The filter shall allow all users to sort and show out the properties as they desired.

1.10.1 **Property_Type**:
Filter by the category of Property_Type, such as Room, Apartment, and House.

1.10.2 **Property_Price**:
Filter by the category of Property_Price.

1.10.3 **Property_Room**:
Filter by the category of Property_Room number.

1.10.4 **Property_Bathroom**:
Filter by the category of Property_Bathroom number.

1.10.5 **Property_Distance**:
Filter by the category of the distance away from CCSF. (If API available)

1.11 Message:

A messaging system that enables communication among Registered_Users and Administrator to communicate.

1.11.1 **Sender**:
A person who sends the message.

1.11.2 **Receiver**:
A person who receives the message.

1.11.3 **Content**:
A message that is communicated between sender and receiver.

1.11.4 **Timestamp**:
The time when the message is sent.

1.11.5 **Property_ID**:
The property_ID regards to the message.

1.11.6 **Reply**:
The message sent in response to the sender by the receiver.

1.12 Search:
All users are able to find desired properties to show out by entering the associated information.

2. Functional Requirements Ver. 2:

Priority 1:

2.1 Unregistered Users:

2.1.1 Unregistered users shall be able to browse the items on the website without being registered.

2.1.2 Unregistered users shall be able to search the apartments according to proximity to SFSU.

2.1.3 Unregistered users shall be able to search the apartments by using the text search bar.

2.1.4 Category tags shall be available as a drop-down menu.

2.1.5 Unregistered users shall be asked for registration only when they decide to contact the seller or post an item.

2.1.6 Unregistered users shall be able to sign up by creating username and password and providing email first name, last name, and address.

2.1.7 Unregistered users shall be required to accept terms and condition upon signing up.

2.2 Registered Users:

2.2.1 Registered users shall have all the privileges of Unregistered users.

2.2.2 Registered users shall be able to sign in by providing their username or email and password

2.2.3 Registered users shall be able to contact sellers and post apartment listings.

2.2.4 Registered users shall be able to access the dashboard with the list of items they have posted.

2.2.5 Registered users shall have the privilege to edit the items they have posted.

2.2.6 Registered users shall be able to send messages to sellers through the website.

2.2.7 Registered users shall be able to receive messages regarding any postings from inquiring buyers.

2.3 Administrator:

2.3.1 Admin shall have all the privileges of Registered users plus the following privileges.

2.3.2 Admin shall be required to censor and delete inappropriate items.

2.3.3 Admin shall not be able to edit items posted by the Registered users.

2.3.4 Admin shall be able to delete Registered users who post inappropriate items.

2.3.5 Admin shall be required to approve legal items before they are shown on the website.

Priority 2:

2.4 Unregistered Users:

2.4.1 Unregistered users shall be able to sort the list of items by price, in both ascending and descending orders.

2.5 Registered Users:

2.5.1 Registered users shall be able to delete posts

2.5.2 Registered users shall be able to delete messages

2.5.3 Registered Users shall be able to reset password

Priority 3:

2.6 Registered Users

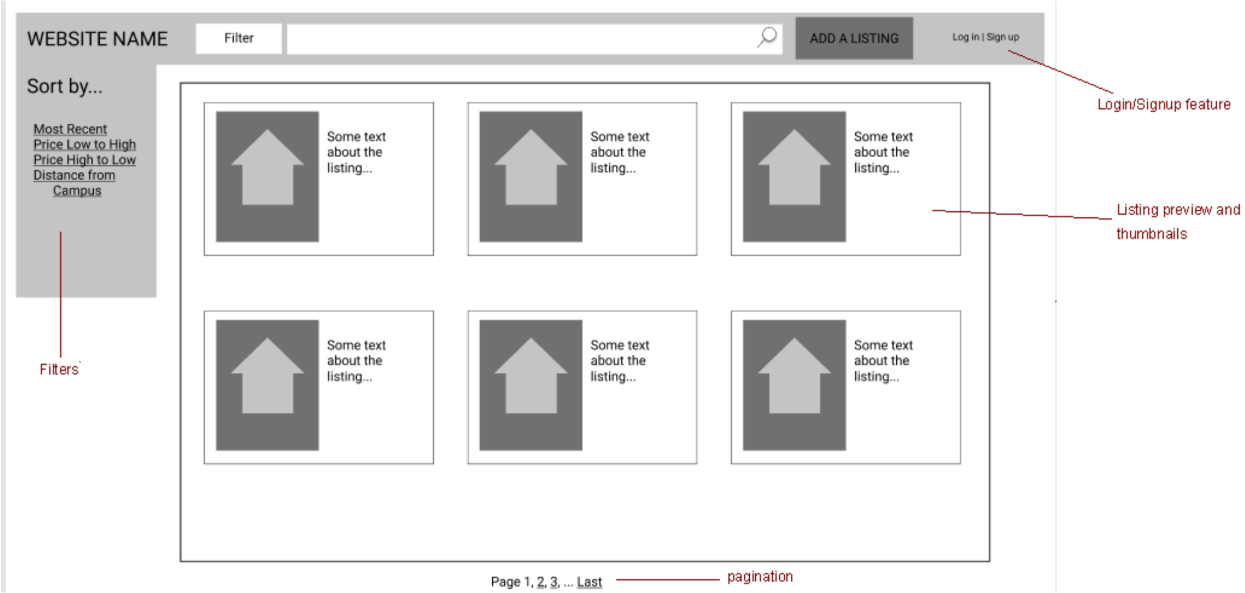
2.6.1 Registered users shall have access to their recently viewed listings

2.6.2 Registered users shall be able to favorite/save listings

3. UI Mockups and Storyboard:

3.1 Use Case – Sarah: Renter

Our renter is browsing the site, looking for an apartment to rent. She searches up rooms and filter through them by price. She sees all the postings related to her search, with a picture of the apartment, the title, and price. She clicks on the description to see the full view of the apartment and information. She gets interested and messages the lister for more information. She then gets taken to the page of the messages between her and the seller.


WEBSITE NAME
Filter
ADD A LISTING
Log in | Sign up
Sort by...
Most Recent
Price Low to High
Price High to Low
Distance from Campus
Some text about the listing...
Some text about the listing...
Some text about the listing...
Some text about the listing...
Some text about the listing...
Some text about the listing...
Page 1, 2, 3, ... Last

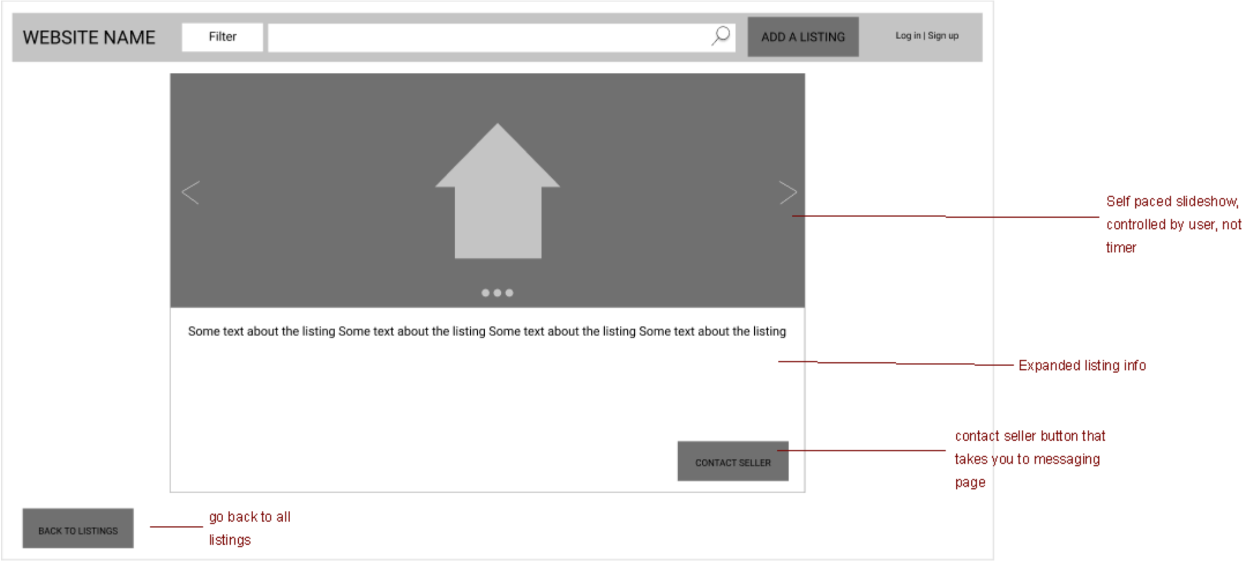
Login/Signup feature

Listing preview and thumbnails

Filters

pagination

Home Page


WEBSITE NAME
Filter
ADD A LISTING
Log in | Sign up
Some text about the listing Some text about the listing Some text about the listing Some text about the listing
CONTACT SELLER
BACK TO LISTINGS

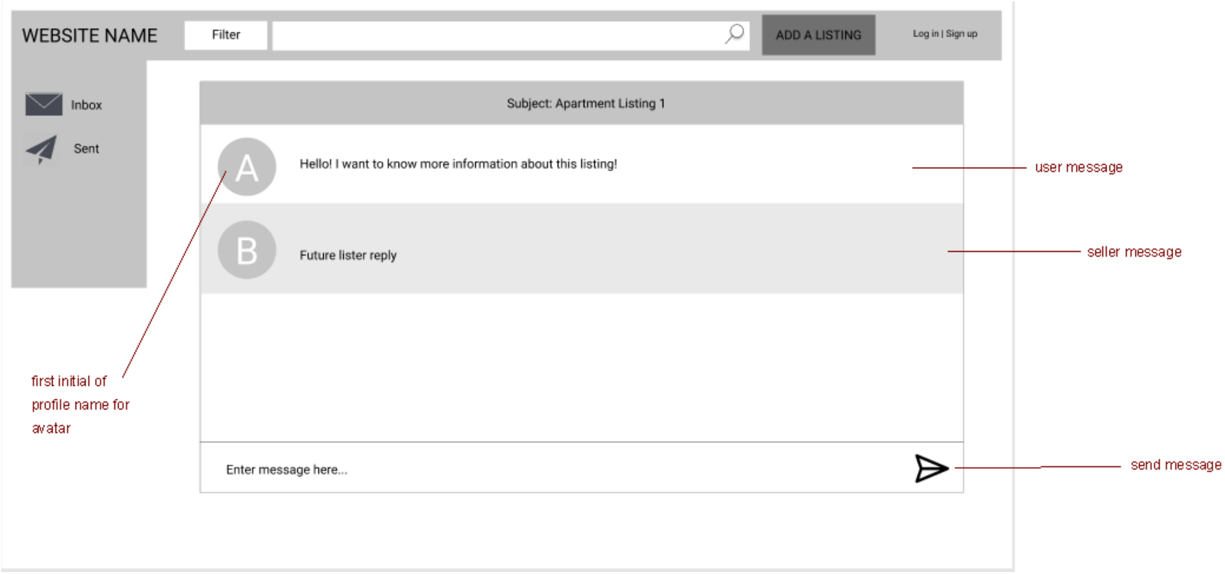
Self paced slideshow, controlled by user, not timer

Expanded listing info

contact seller button that takes you to messaging page

go back to all listings

Listing Description Page



Hello! I want to know more information about this listing!

Future lister reply

user message

seller message

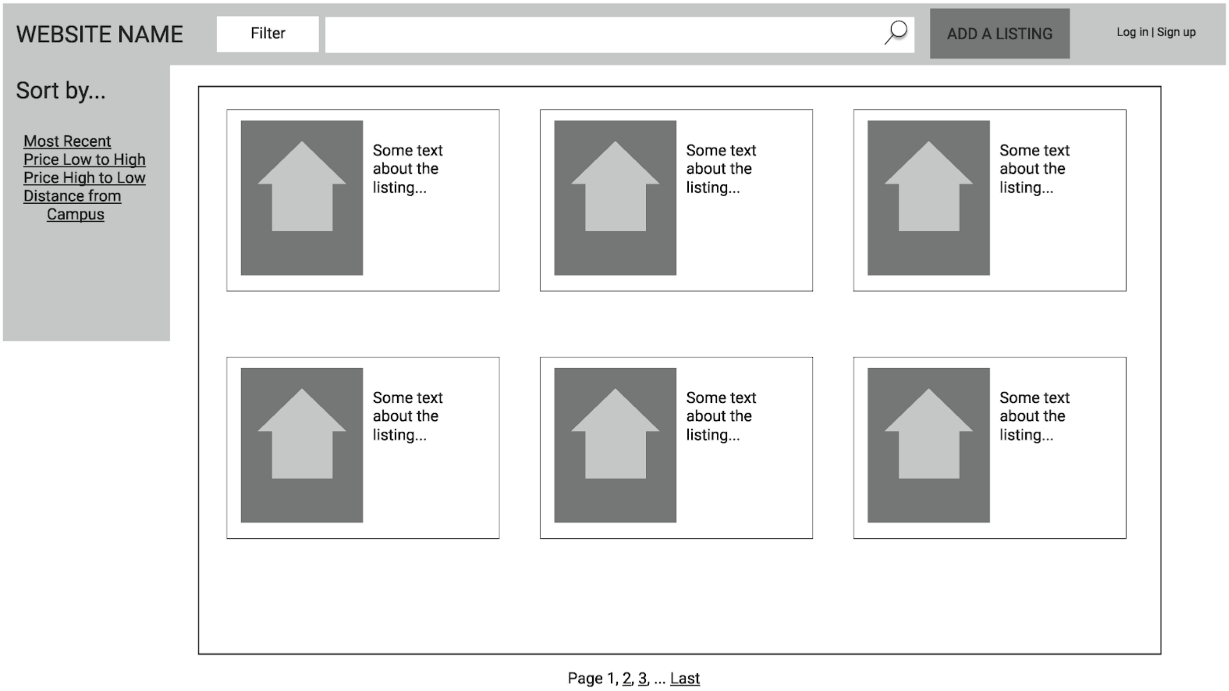
send message

first initial of profile name for avatar

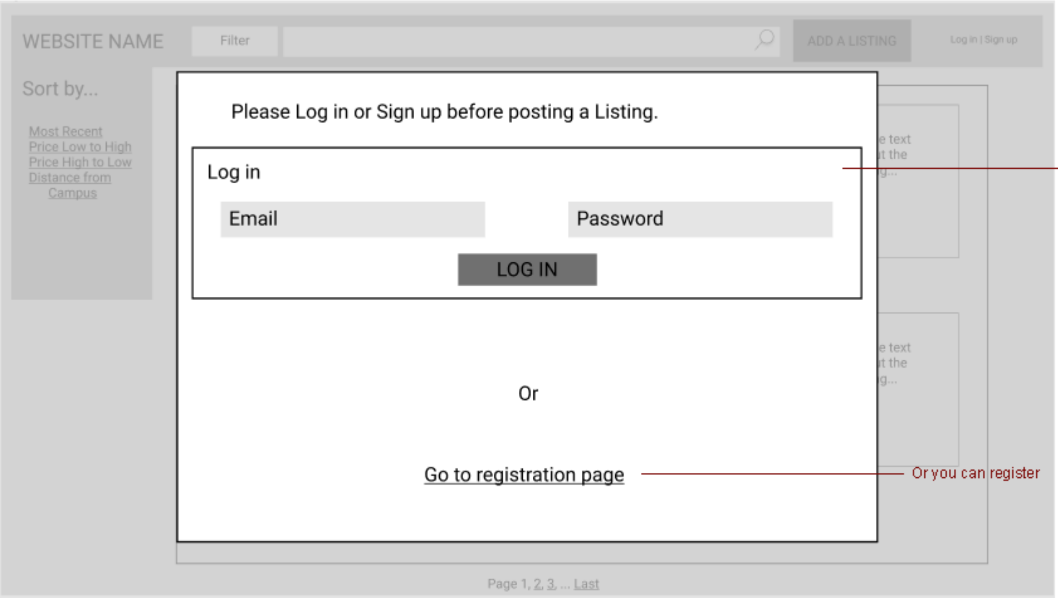
Message Page

3.2 Use Case – Robert: Lister

Our lister goes on the website but forgets to log in. He looks for an upload/post button to post his listing. He finds it and clicks on it. Next, he is prompted to log in. He logs in, then sees the listing form with clear prompts on what to do. He follows the prompts, uploads his photos and writes a description and price, and posts it with a simple click. He is taken to his dashboard where he sees a message above his post that says his post is under review before becoming public.


WEBSITE NAME
Filter
ADD A LISTING
Log in | Sign up
Sort by...
Most Recent
Price Low to High
Price High to Low
Distance from Campus
Some text about the listing...
Some text about the listing...
Some text about the listing...
Some text about the listing...
Some text about the listing...
Some text about the listing...
Page 1, 2, 3, ... Last

Home Page

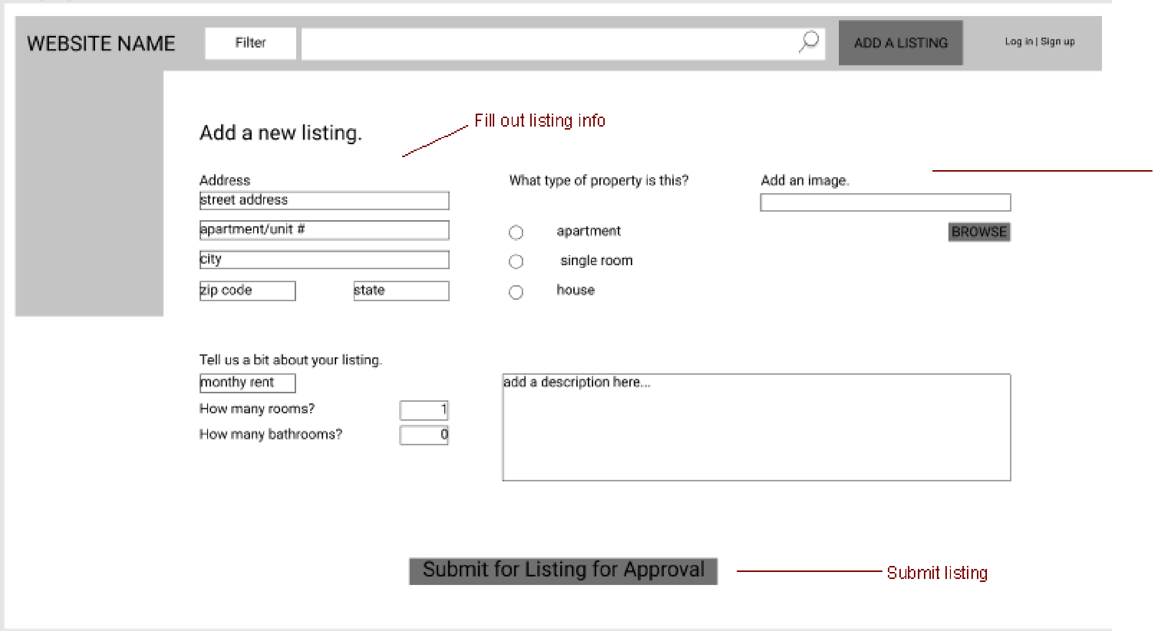


Please Log in or Sign up before posting a Listing.

Input username and password

Or you can register

Log in/Sign up Page



Fill out listing info

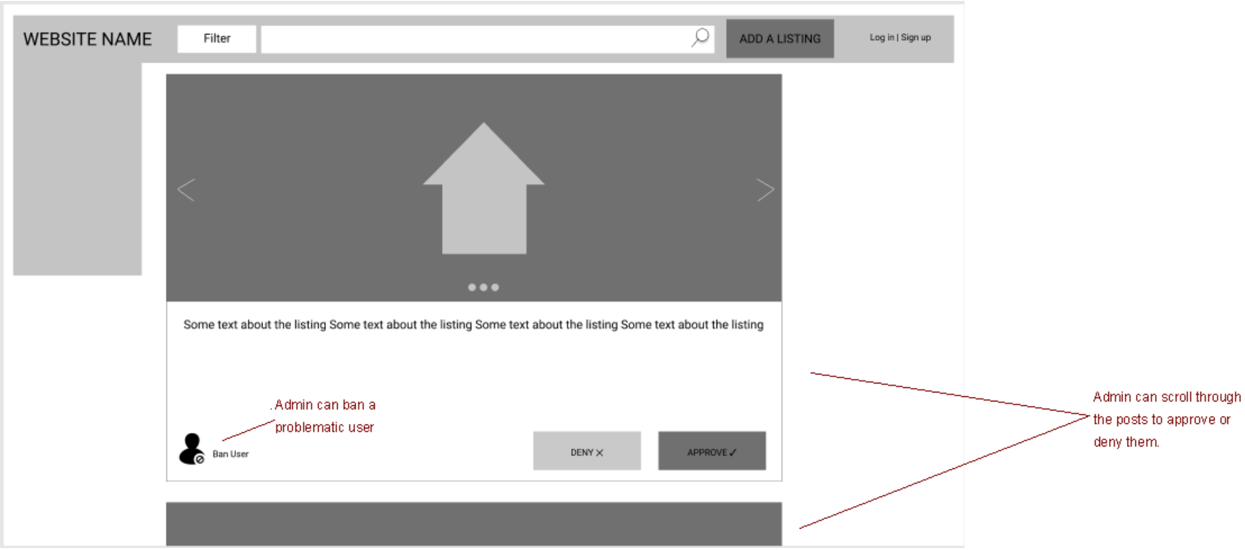
User can upload an image

Submit listing

Add new listing Page

3.3 Use Case – Tammy: Admin

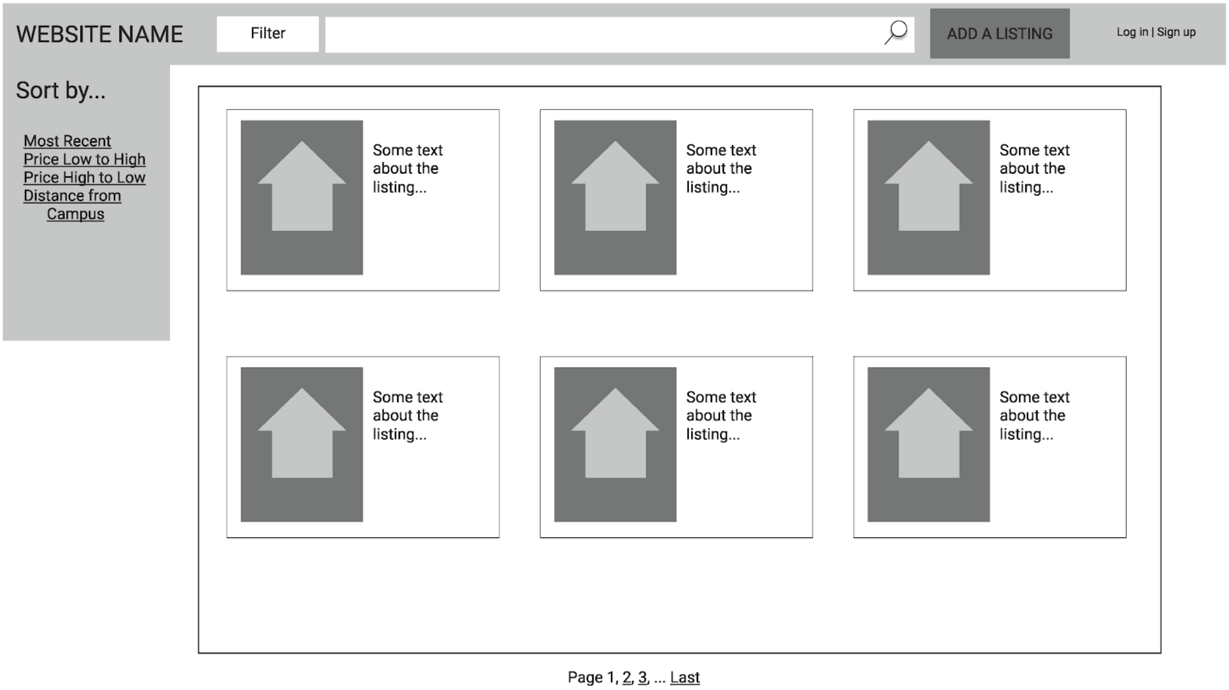
Our administer can shop and administer at the same time. She can check her dashboard for new posts that need to be approved. She sees the unapproved posts and reads through them to approve or deny it. She sees 5 posts that have spam with inappropriate images and deletes them. She then goes on to ban the user who made the posts.


WEBSITE NAME
Filter
ADD A LISTING
Log in | Sign up
Some text about the listing Some text about the listing Some text about the listing Some text about the listing
Admin can ban a problematic user
Ban User
DENY ✕
APPROVE ✓
Admin can scroll through the posts to approve or deny them.

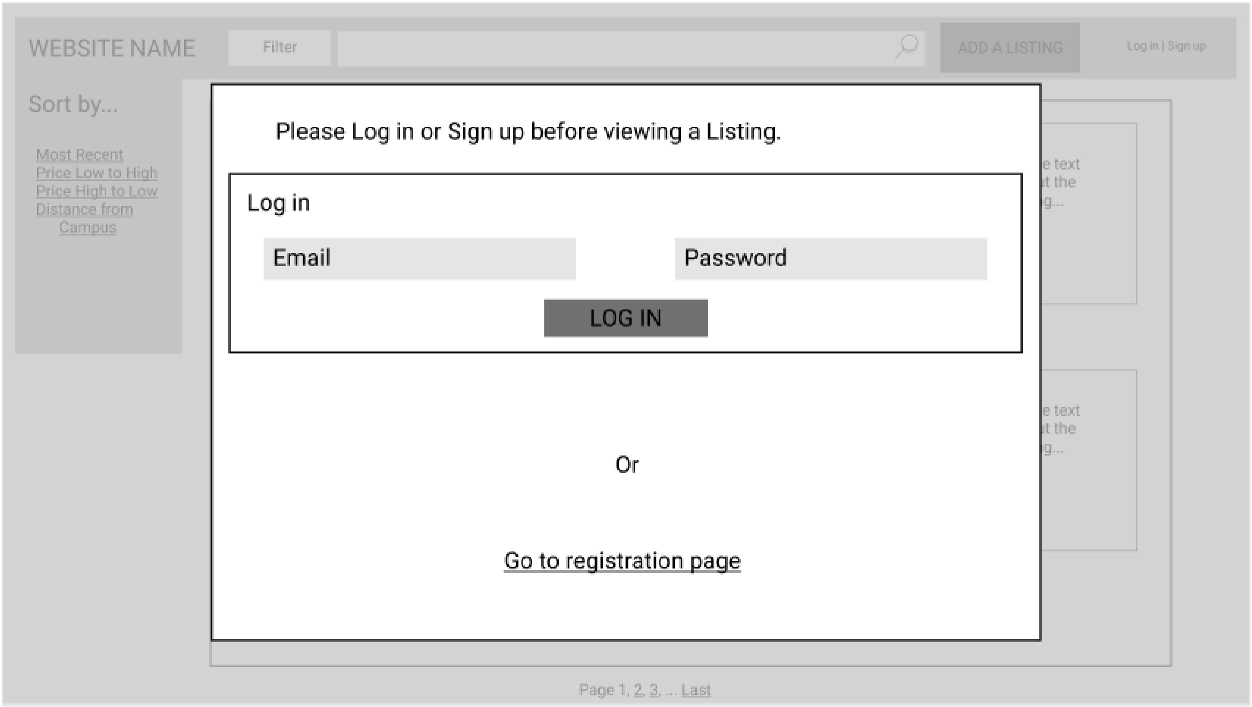
Dashboard Page

3.4 Use Case – Adam: Guest

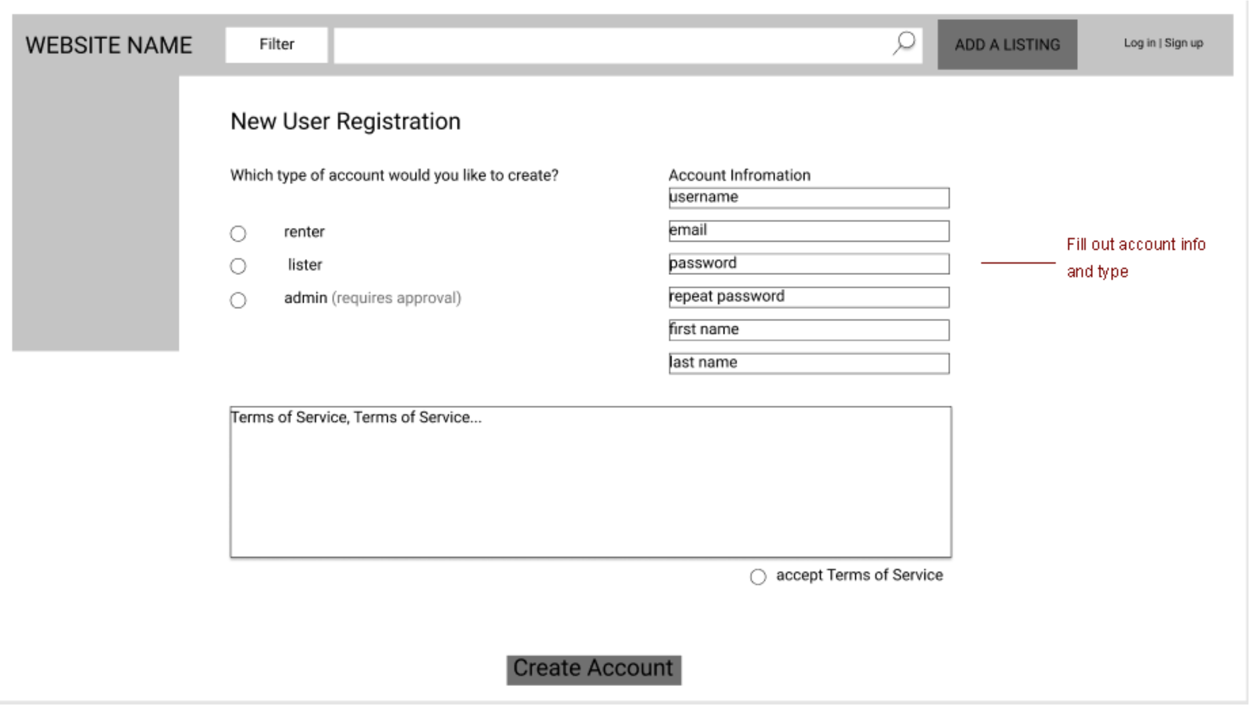
Our guest finds our website by searching online for apartments. He wants to check out the site. He goes on the site and scrolls through the listings. A post catches his eye because of how low the price is. When he wants to click on it to view it, he is prompted to register or login in order to continue. Adam gets a bit annoyed but really wants to see the post, so he clicks “register” on the login page and quickly fills out the short online form. He was taken back to the now unlocked description page where he can view more details on the listing and direct message the seller


WEBSITE NAME
Filter
ADD A LISTING
Log in | Sign up
Sort by...
Most Recent
Price Low to High
Price High to Low
Distance from Campus
Some text about the listing...
Some text about the listing...
Some text about the listing...
Some text about the listing...
Some text about the listing...
Some text about the listing...
Page 1, 2, 3, ... Last

Home Page


Please Log in or Sign up before viewing a Listing.

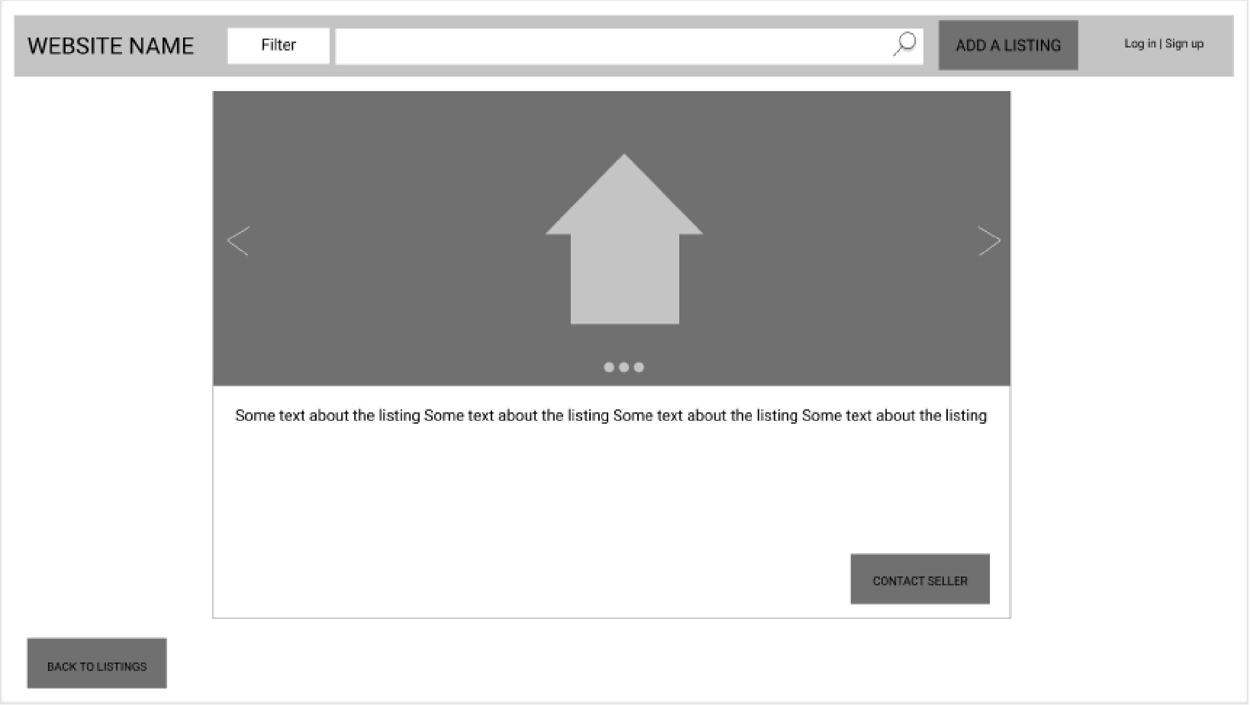
Log in/Sign up Page



Which type of account would you like to create?

Terms of Service, Terms of Service...

Registration Page



Some text about the listing Some text about the listing Some text about the listing Some text about the listing

List description Page

4. High Level Architecture, Database Organization:

4.1 DB Organization

4.1.1 User: first_name, last_name, password, email, id, privilege, user_name. (primary key: id; foreign key: user_name)

4.1.2 Property_List: property_id, landlord_id, room, bathroom, size, type, address, price, park_space, status. (primary key: property_id; foreign key: landlord_id).

4.1.3 Message: sender, receiver, content, time, message_id, property_id. (primary key: message_id; foreign key: property_id, sender, receiver)

4.2 Media Storage

4.2.1 Images files will be kept in the file system.

4.3 Search/Filter architecture and implementation

4.3.1 Setup website and create tables in AWS database.

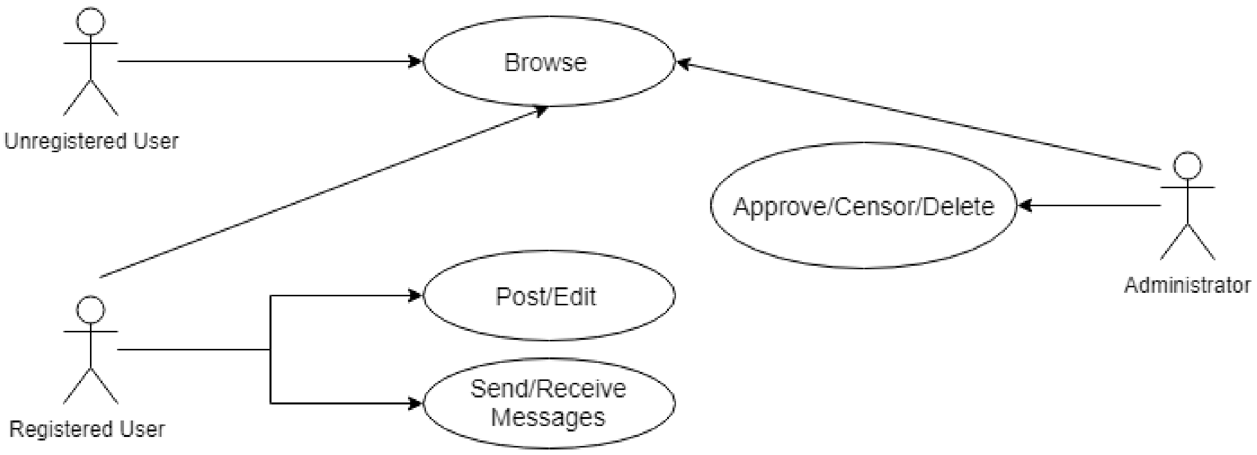
4.3.2 Establish a connection between MySQL database in AWS and Node.js website.

4.3.3 Collect keywords by filter click or search bar entry from the website and create validly MySQL query to the database.

4.3.5 Show the Search/ Filter result from the database on the webpage.

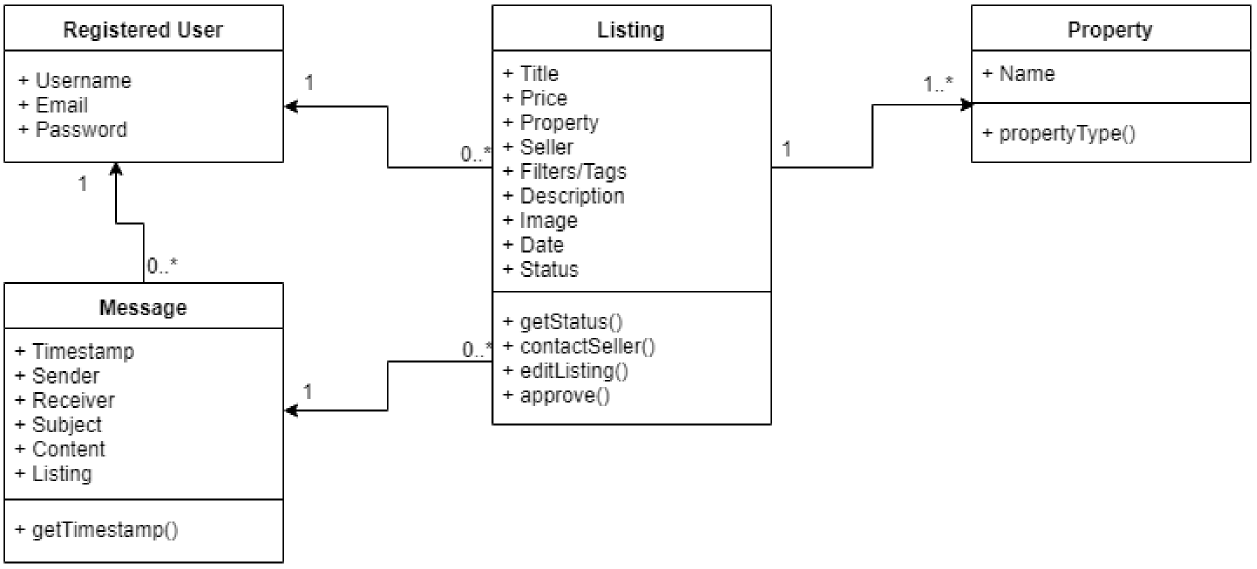
5. High-Level UML Diagrams:

Use Case Diagram



This high-level use-case UML diagram describes what a system does from the standpoint of an external observer (Unregistered user can only browse, the registered user can browse, post/edit listing, and send/receive messages, etc.). The emphasis is on what a system does rather than how. This particular diagram communicates the vision of our school renting/leasing system easily to our non-technical crowd (CEO, marketing, sales, investors).

Class Diagram



6. Identify Key Risks:

6.1 Skills risks

Each team member has different skill sets. We fear that gap in skill among team members could drag the entire team from achieving milestones as prescribed. In order to mitigate the skill, risk the team shall implement open communication among team members. In addition, each team member shall seek the necessary resources to bridge the skill gap.

6.2 Schedule risks

Each team members have different schedules. We fear we might not be able to do sit down meeting on a regular basis to discuss ideas and projects. In order to mitigate the schedule conflict, the team started making using of project management tool Trello and the team communication tool Discord from the very beginning of the class.

6.3 Technical risks

The backend team has limited exposure to the backend technologies. The backend lead and the team lead shall coordinate to learn the necessary technology at any cost.

6.4 Teamwork risks

Every individual brings unique traits to the team. In the event that conflict may arise between team members, it may derail the entire team from achieving the milestone. In order to mitigate the conflict between team members, the team leader shall encourage each and every member to communicate openly. Each team member shall put the need of the team above any individual’s need.

6.5 Legal/content risks

We do not have a legal advisor in the team. Therefore, even while adhering to the best practices to minimize intellectual property (IP) infringements, we may unintentionally infringe on intellectual property rights and copyrights. In order to mitigate legal risks, the team shall apply contents/images that are under creative commons license.

6.6 Financial risks

The project is currently being hosted on a free-tier AWS EC2 instance. However, the free-tier quota might be surpassed and the team might be charged extra fees. To mitigate the financial risk the team lead shall monitor the free-tier AWS account frequently.

7. Project Management:

Using project management tool Trello team leader shall divide the task to each team member. Each task shall be subdivided under different board names as follows:

7.1 To Do BucketList Board:

All the tasks to be completed before the milestone due date shall be listed in the To Do BucketlList board. Each team member is assigned the respective task to be completed before the due date. Team lead shall enforce checkpoint date prior to the milestone due date.

7.2 In Progress Board:

All the tasks that the team is attempting to accomplish shall be posted in the In Progress Board.

7.3 Completed Board:

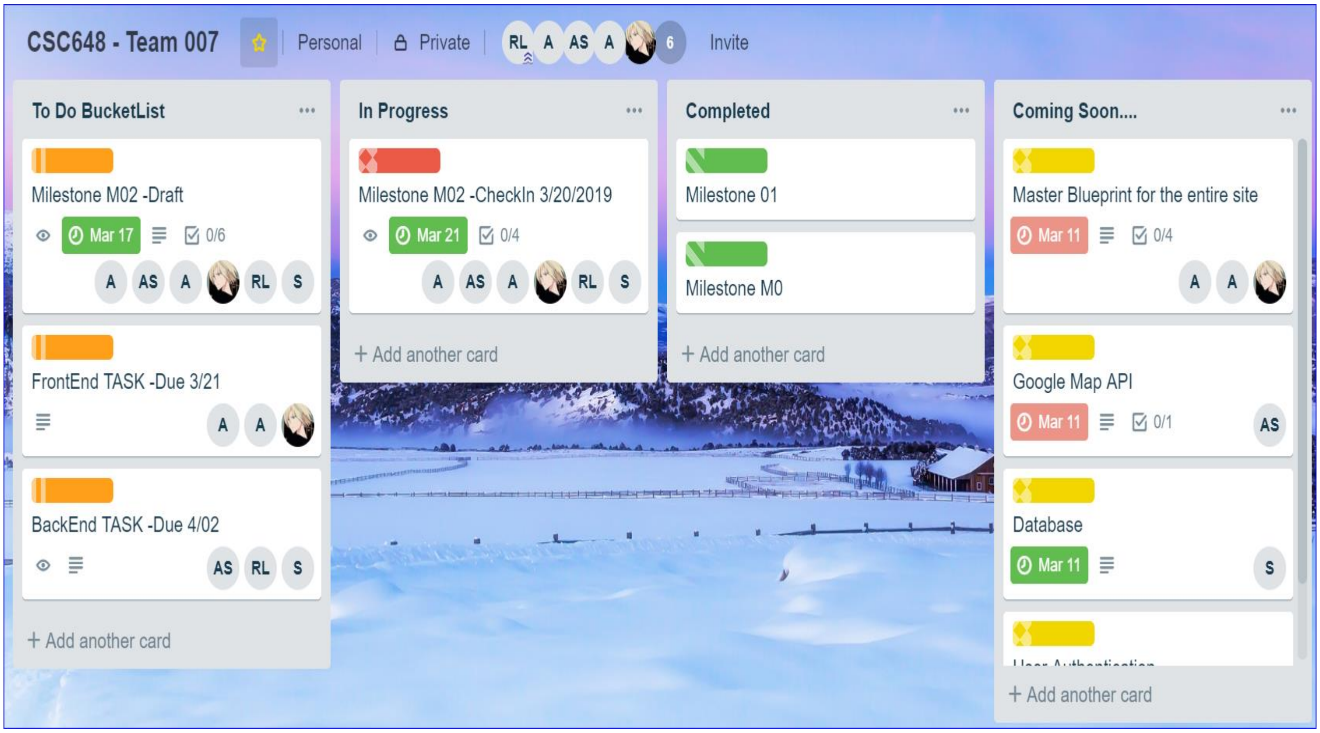
Any completed milestone shall be listed in the Completed Board.

7.4 Coming Soon Board:

Coming Soon board is a pro-active approach to prepare the team in anticipation of the upcoming milestone. This board lists top priority matter that team must address for the upcoming milestone. This way the team shall be prepared for the upcoming milestone which is to be solved in the coming week.

Conclusion:

Trello makes it easy for the team to identify two key goals - each member’s task objectives and expected due date. Moreover, checkpoint enforcement prior to the due date will prompt each member to be proactive and contribute towards project goal in a timely manner.



To Do BucketList

- Milestone M02 -Draft — Mar 17 — 0/6 — A AS A RL S
- FrontEnd TASK -Due 3/21 — A A
- BackEnd TASK -Due 4/02 — AS RL S

In Progress

- Milestone M02 -CheckIn 3/20/2019 — Mar 21 — 0/4 — A AS A RL S

Completed

- Milestone 01
- Milestone M0

Coming Soon....

- Master Blueprint for the entire site — Mar 11 — 0/4 — A A
- Google Map API — Mar 11 — 0/1 — AS
- Database — Mar 11 — S
- User Authentication

Team 007’s Trello Dashboard