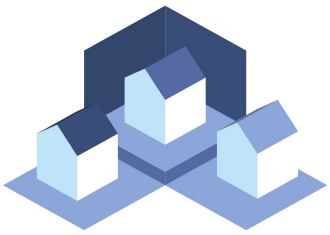


lental

VRK Solutions Inc.

Design Document

University of British Columbia - Okanagan

COSC 499 - Summer 2019

June 5th, 2019

Contacts:

Jae Ung Kim (khyy37@hotmail.com)

Stephen Okanlawon (stephen.t.okanlawon@gmail.com)

Jan Reisler (janreisler@gmail.com)

1. Project Description

Our website, Lental, will allow users to post ads/listings for their suites and homes and search for ads/listings to rent. The idea is to have a simple design that is very user friendly and easy to find homes/suites to rent. It will have many filtering options to find a specific listing, a map to show listing locations and distance, allow for posting ads/listings, logging into an account and save listings to view later, and even a search agent that automatically emails listings to users with specified filters.

2. Technical Specifications

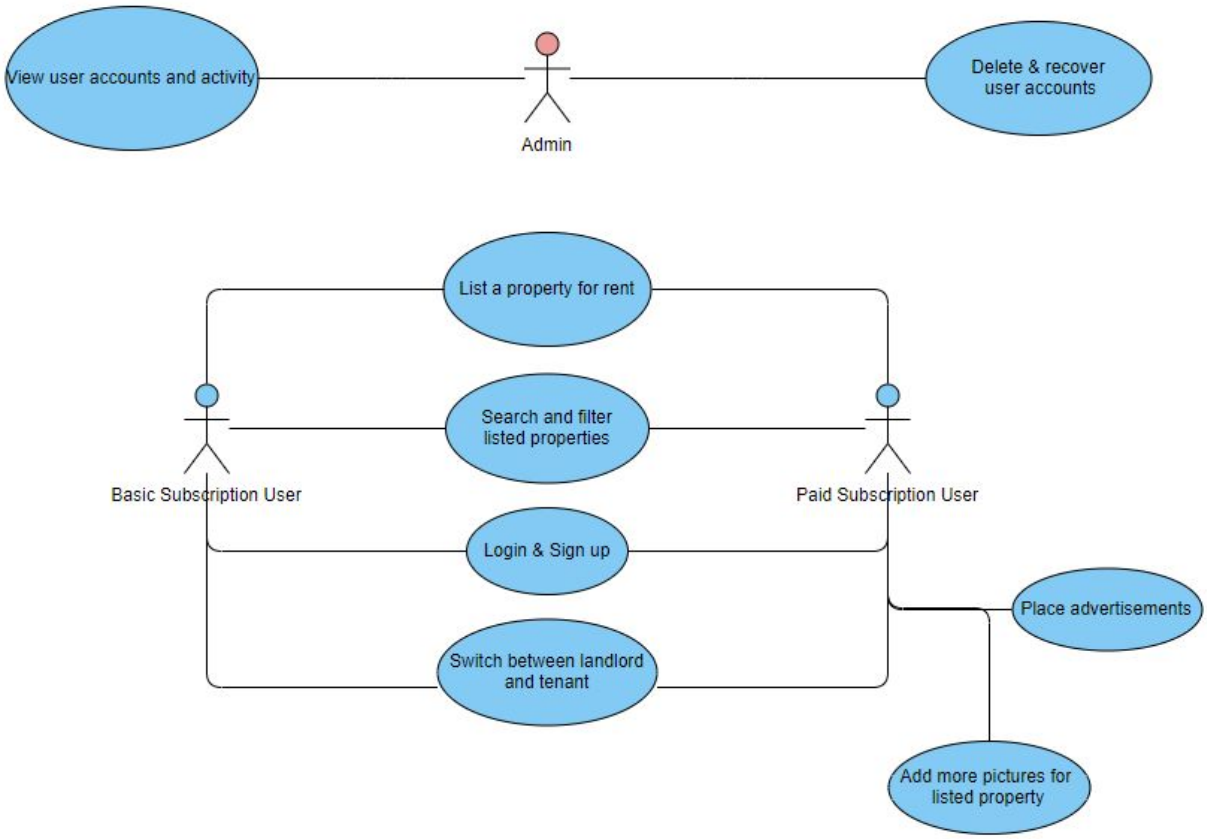
- HTML, CSS, and JavaScript/JQuery will be used for the front end and bootstrap will be used as the framework
- PHP will be used as the backend with no framework
- PHPUnit will be used to test the PHP and Selenium will be used to test the website in a more automated way
- Atom will be used as the main IDE, GitHub for version control, and Laragon for local server/database hosting for offline development
- MySQL will be used for the database
- Google maps or Bing maps will be used for maps view

3. User Groups & Usage Scenarios

There will be three main user groups for this website namely:

a. Admin: The admin will be granted access to the back end of the database so they will be able to view, update, delete and recover accounts and account information from the website so that the website can be constantly monitored in case of spam, inappropriate posts or errors by the customer.
b. Basic (free) user: The basic/free user will be granted access to listing properties and viewing properties but some of the features will be limited
c. Paid user: The paid user is similar to the basic user except, the limitations are lifted and give the user more access to the website's features.

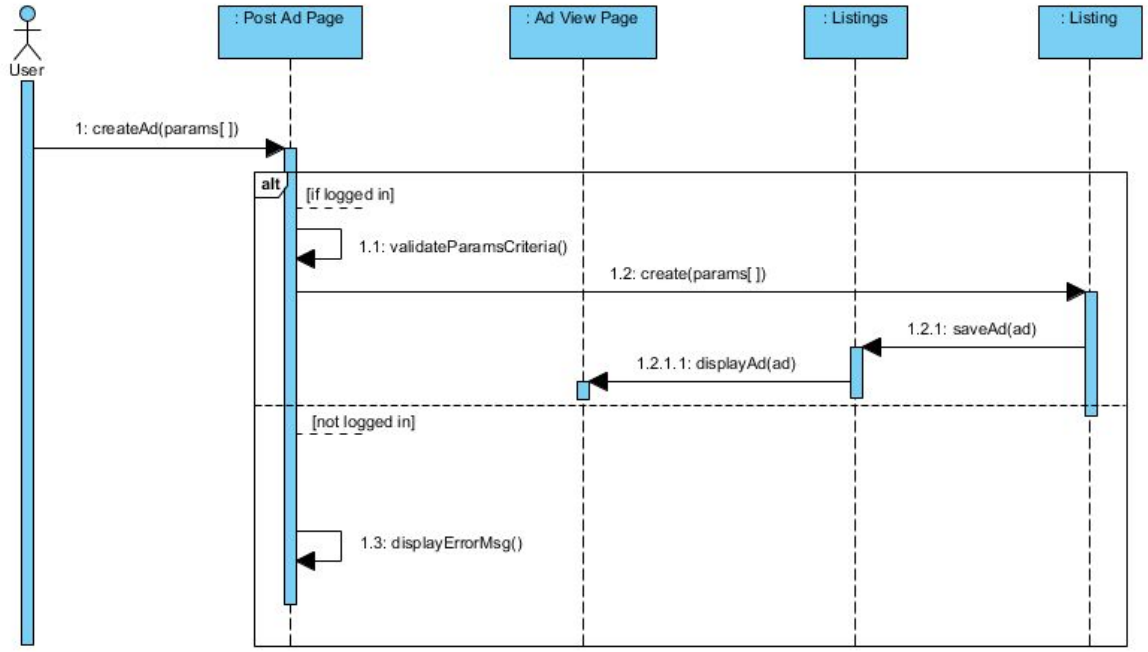
Below is a visual representation of some usage scenarios for the aforementioned user groups:



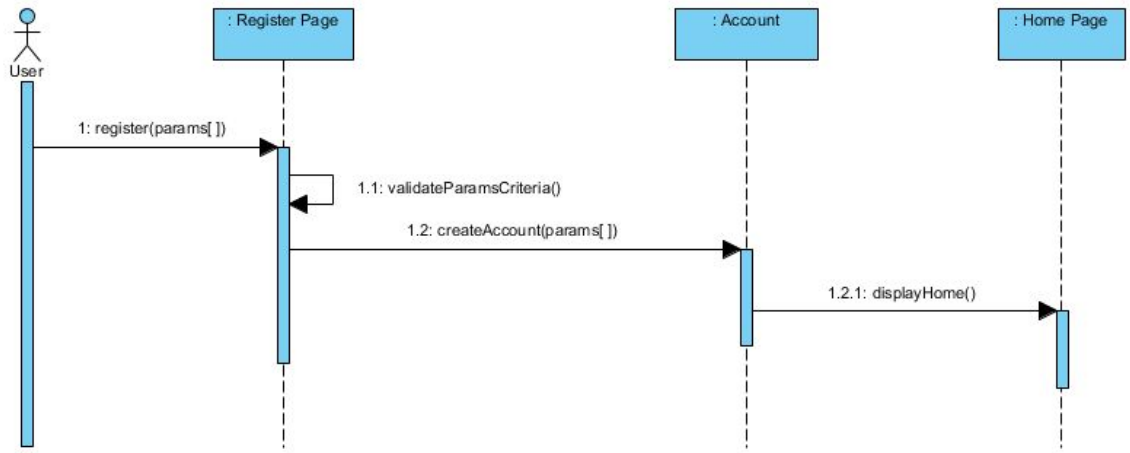
4. System Architecture Diagrams and Models

a. Sequence Diagrams: UML Sequence Diagrams to detail how actions are carried out based on user interaction

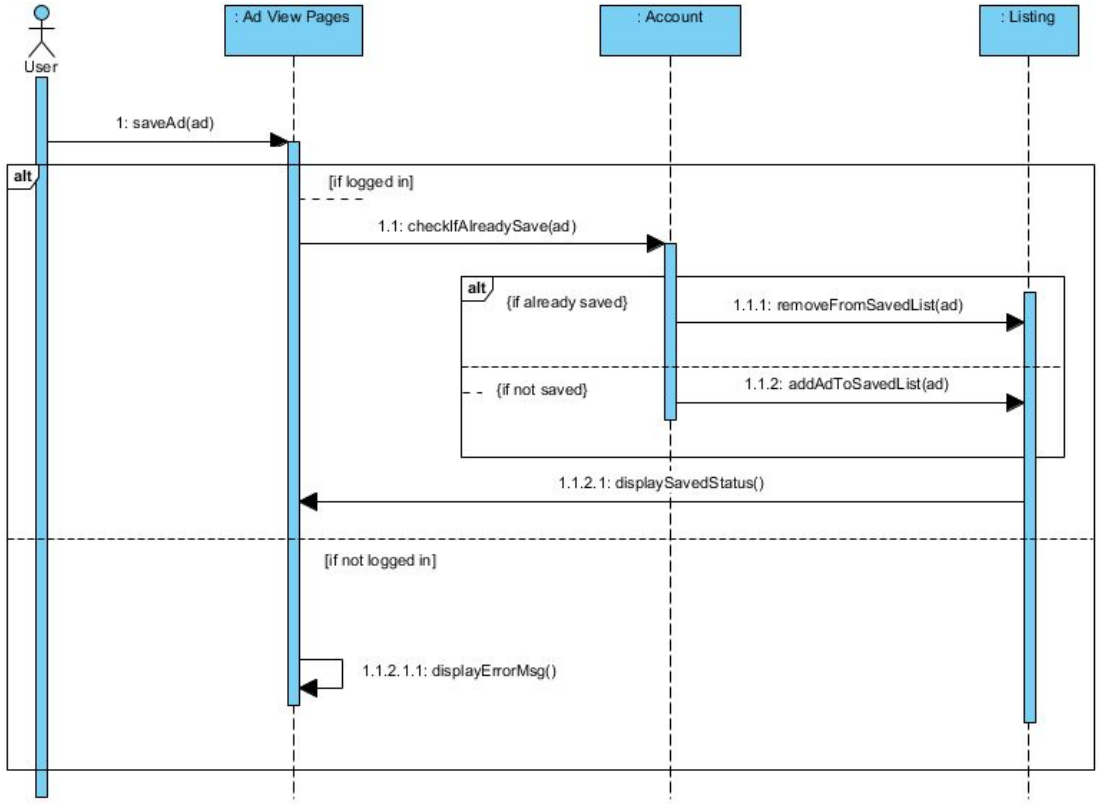
The Post Ad sequence diagram shows the operations involved when posting an ad/listing. This sequence diagram also follows a very similar structure to editing an ad/listing.



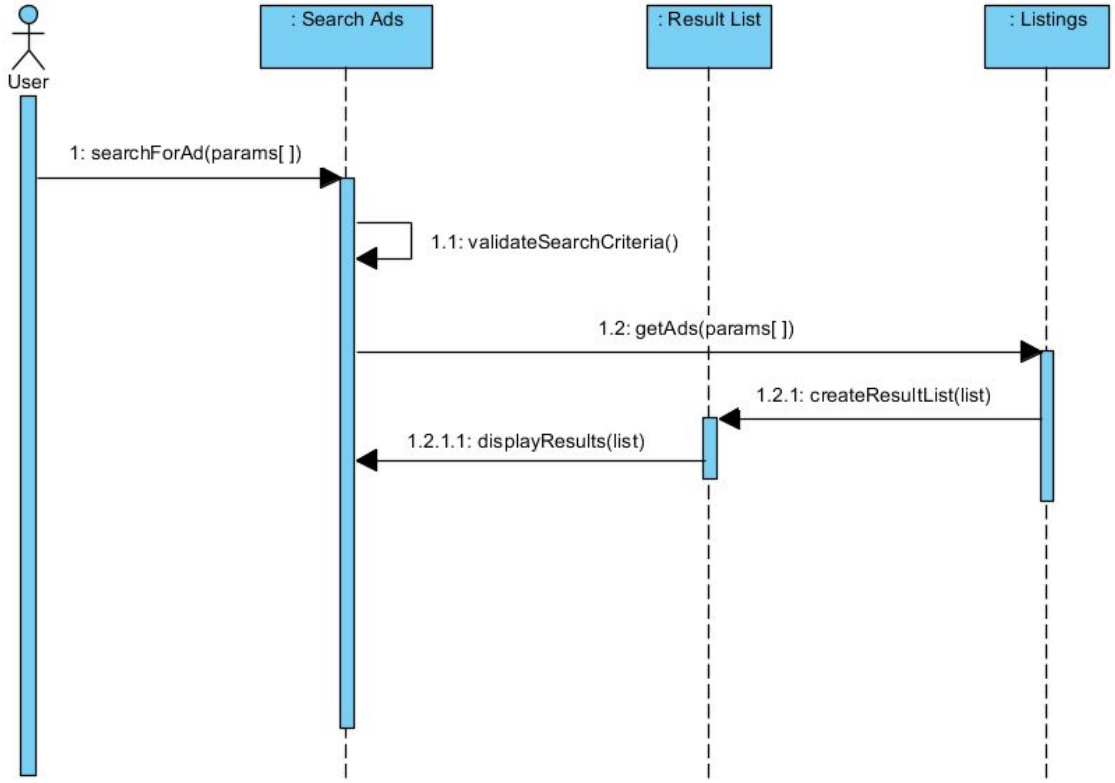
The Register sequence diagram shows the operations involved in registering an account.



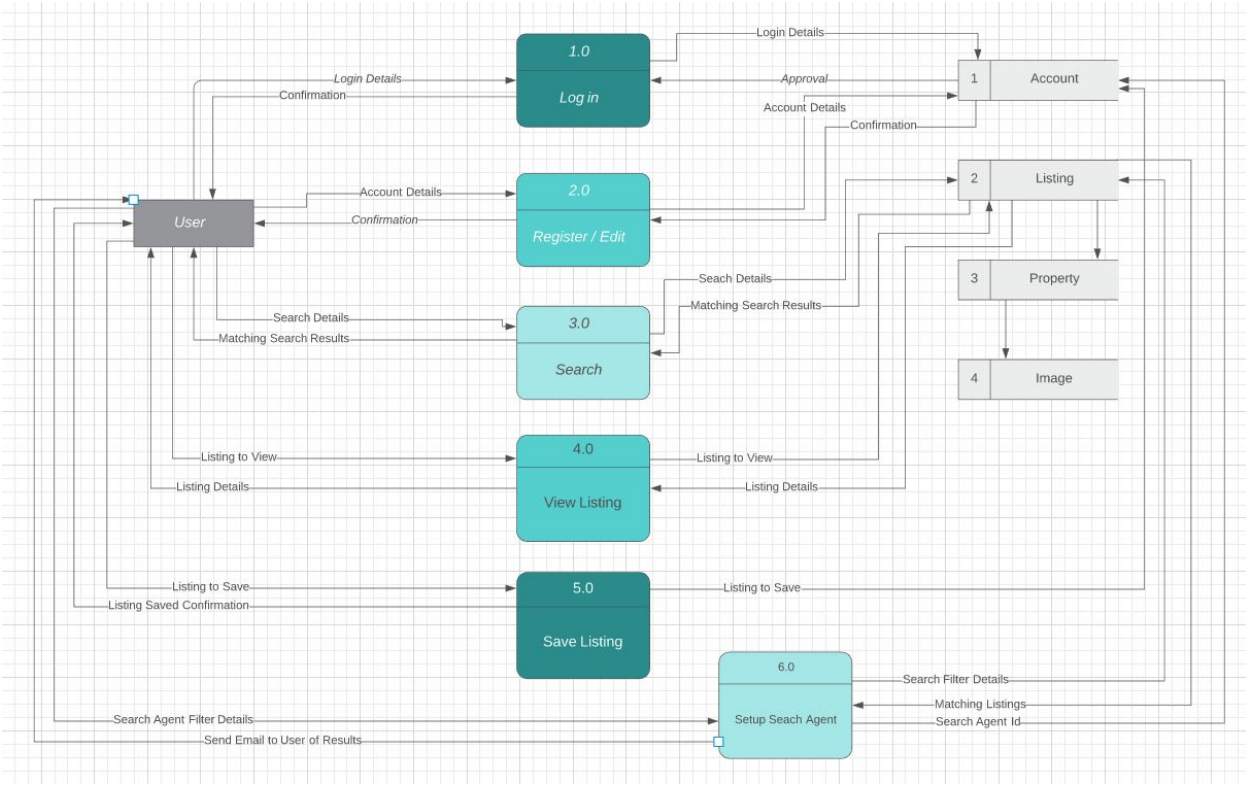
The Save Ad sequence diagram shows the operations involved in saving a ad/listing to be viewed later.



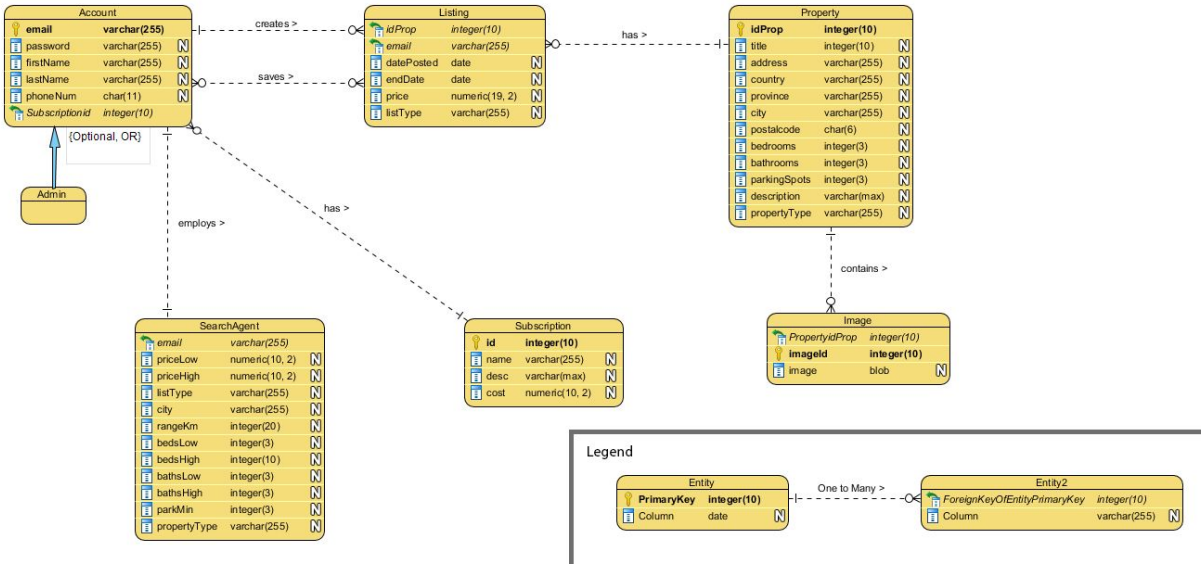
The Search For Ads sequence diagram shows the operations involved in searching for and filtering ads/listings.



b. Data-Flow Diagram: Data-Flow Diagram to show how data is processed and created in the system



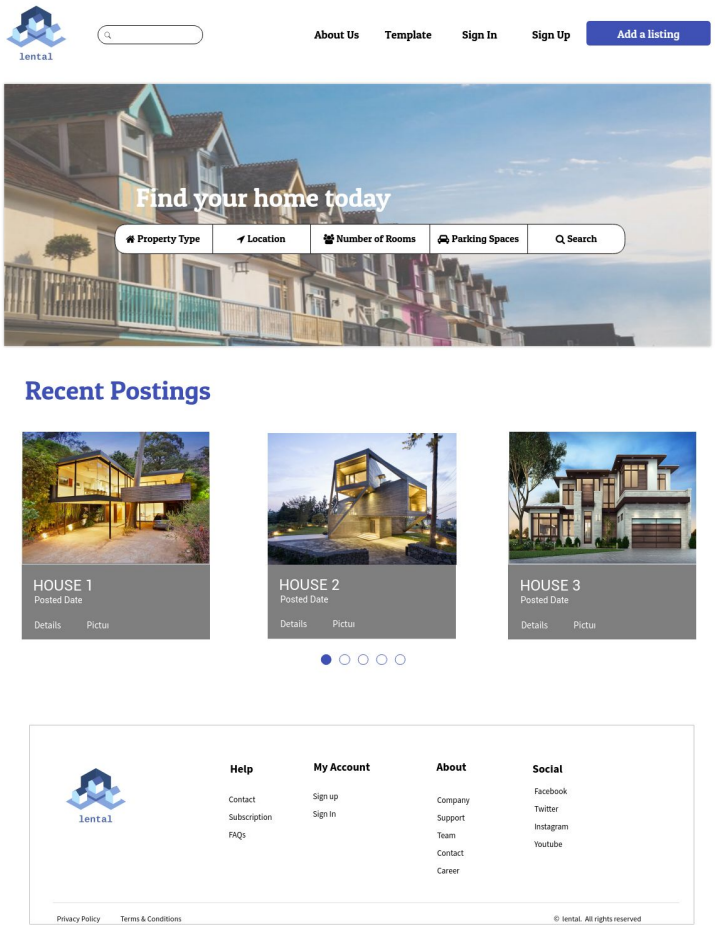
c. ER Diagram: ER Diagram (Entity Relationship Diagram) to show how entities relate to each other in the database



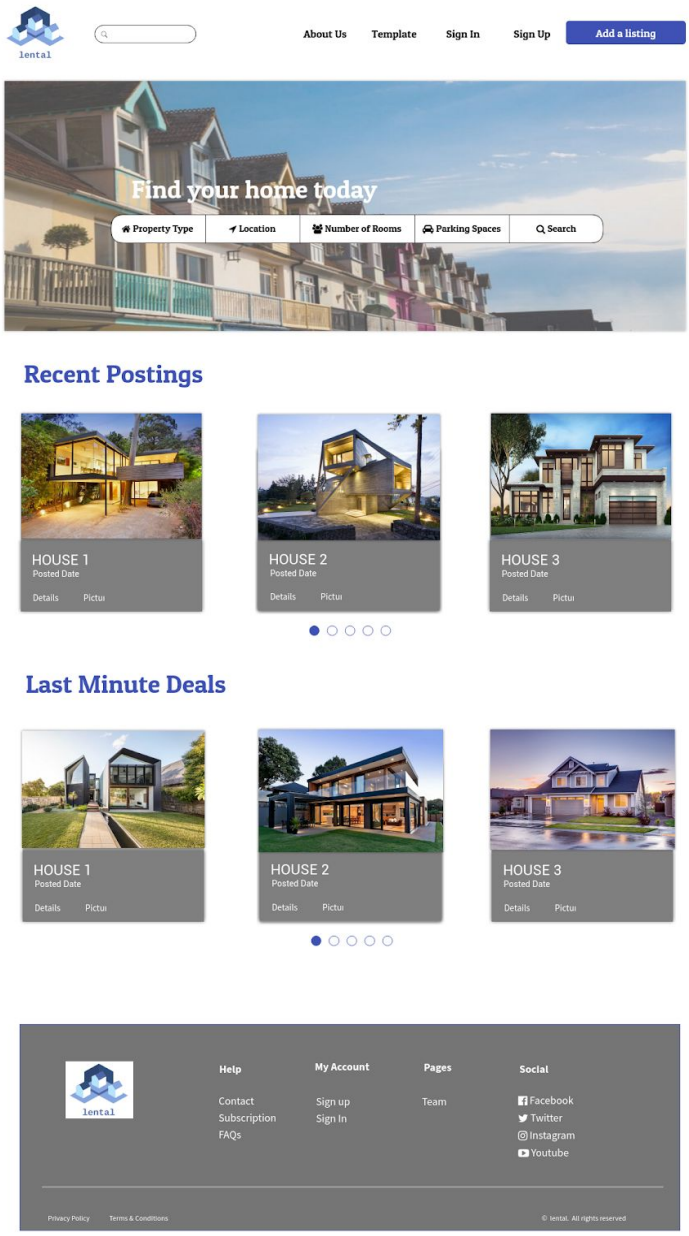
5. UI Mockups

a. Front Page

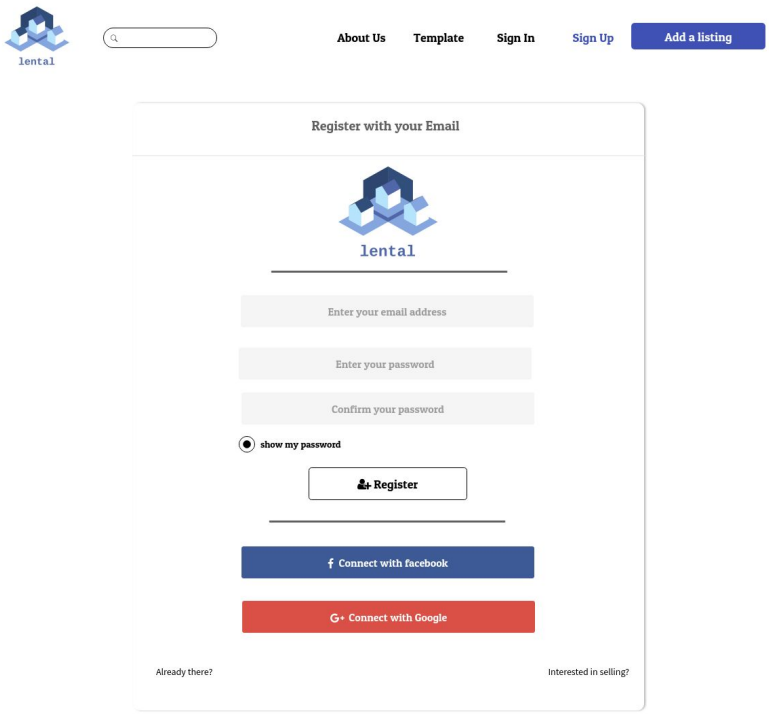
i. Initial



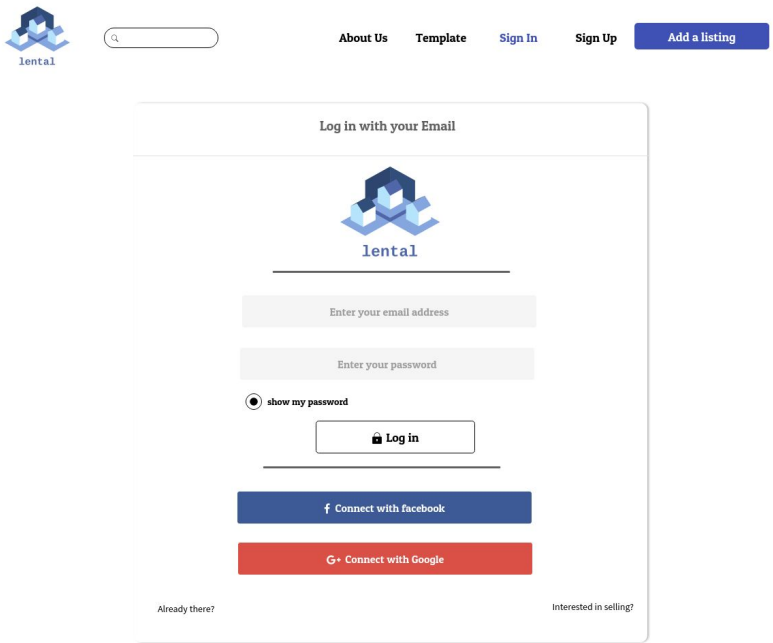
ii. Adjusted


lental
About Us
Template
Sign In
Sign Up
Add a listing
Find your home today
Property Type
Location
Number of Rooms
Parking Spaces
Search
Recent Postings
HOUSE 1
Posted Date
Details
HOUSE 2
Posted Date
Details
HOUSE 3
Posted Date
Details
Last Minute Deals
HOUSE 1
Posted Date
Details
HOUSE 2
Posted Date
Details
HOUSE 3
Posted Date
Details
lental
Help
Contact
Subscription
FAQs
My Account
Sign up
Sign In
Pages
Team
Social
Facebook
Twitter
Instagram
Youtube
Privacy Policy
Terms & Conditions

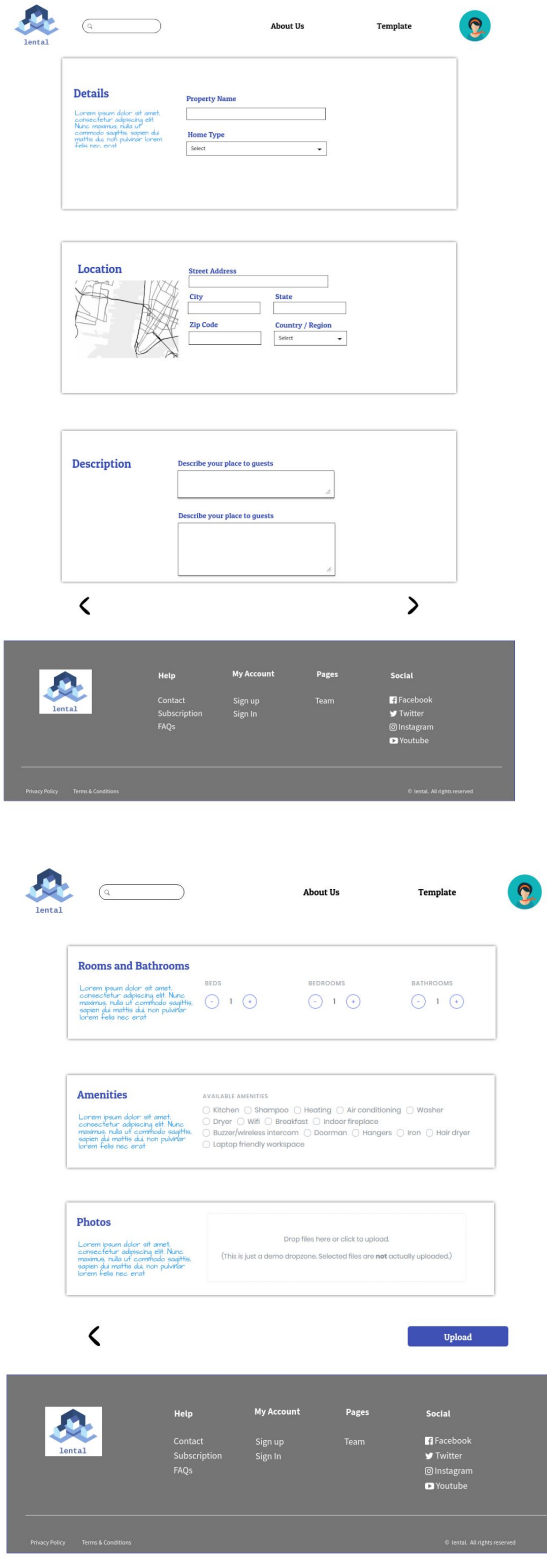
b. Sign Up



c. Sign In



d. Adding Listings



6. Test Plans

a. Functionality Testing

i. Check all the links in web pages
   1. Test the outgoing links from all pages to the specific domain under test.
   2. Test links jumping on the same pages
   3. Test to check if there are any orphan pages
   4. Check for broken links

ii. Check forms on all pages
   1. Check validations on each fields.
   2. Check how forms respond to SQL injection, wrong inputs, and etc.

b. Component Testing

i. Validate HTML, CSS, and JavaScript
   1. In order to test for syntax errors for HTML & CSS, we will run our markups through W3C validator.
   2. Unit testing will be conducted for each JavaScript functions.

ii. Browser Compatibility
   1. We will check if our website work with different types of browsers.

c. Integration Testing

i. Check how all different features & pages work together
   1. Before merging into master branch, we will run our codes through the CI tool (TravisCI) to test whether our components behave as expected.
   2. We will combine the individual tested components one by one and test incrementally.

d. User Testing

i. User walk-through of website (mobile & browser)
   1. We will have a random person go through the website attempting to complete a list of tasks such as “create an account and sign in.”
   2. Once the user does that, we can ask him / her to “rent a home in Kelowna” or “buy a home in Kelowna”