Design Document

sublett$_r$

8 September 2017

Team 9

Josh Newlin, Michael Vieck, Jeremy Lehman, and Joel Van Auken

Purpose

Currently, there are many individuals at Purdue searching for subleases. Many times, people will put their advertisement up on Facebook or an apartment selling website only to have no one look at the post. A person may even be undersold by their own website, giving preference over renting long term than subleasing. This creates many unnoticed opportunities that individuals in Lafayette and West Lafayette miss out on. Our aim is to streamline this process and make it much easier for an individual to lease and find an apartment.

Functional

1. As a user, I would like to search through sublease listings in my area.
2. As a user, I would like to see/share images of the property.
3. As a lister, I would like to advertise/post my residence for a sublease.
4. As a user, I would like to message property owners.
5. As a user, I would like to review properties after staying.
6. As a user, I would like to sort and filter posts on different criteria; distance, rating, cost.
7. As a user, I would like to be able to save subleases I am interested in
8. As a user, I would like to favorite subleases.
9. As a user, I would like to be able to sort properties by location.
10. As a lister, I would like to be able to add tags to properties (furnished, scenic etc.)
11. As a lister, I would like to be able to edit posts of properties
12. As a user, I would like to be able to tell if a property I am interested already is in the process of being reserved

13. As a lister, if my property is scraped from an external website, I would like to be notified if my apartment is being displayed (time permitting)
14. As a user, I would like to see posts from Facebook pages. (Purdue Housing Rent/Sublease)
15. As a user, I would like to see posts from other apartment sites such as BoilerApartments, ApartmentGuide, etc
16. As a user, I would like to see apartment complexes and wholesale apartment lessors that have subleases available
17. As a lister, I would like to be able to have profile with details such as location, contact info, and profile picture
18. As a lister, I would like to post open houses of apartments
19. As a user, I would like a map of all locations and be able to highlight a specific area instead of searching.
20. As a user, I would like to enable a heatmap on the apartment map with the highest subleased or most desirable apartments highlighted.
21. As a user, I would like to see how many roommates, if any, are going to be in the apartment i'm renting/subletting and some anonymous information about them.
22. As a lister, I would like to request a membership or a flat fee that gives me priority in search results over other subletters when users of the site search for apartments like mine. (Time Permitting)
23. As a lister, I would like to view statistics on the web traffic of my listing and if those users are visiting any similar listings. (Time Permitting)
24. As a user, I would like to have messages stored so I can re-read them later.

Non-Functional

1. As a developer, I would like to scrape local sublet listings from Facebook, etc.
2. As a developer, I would like the server to handle up to ~200 concurrent users without decrease in latency.
3. As a developer, I would like to have automated deployment and be alerted if the website goes down for any reason. (Docker, Jenkins, TeamCity, etc)
4. As a user, I would like to have a clean UI experience that is not cluttered with too many features or elements on the landing page.
5. As a user, I would like the website to have animations.
6. As a user, I would like my website flow to be straightforward and never be confused about site navigation.

7. As a developer, I would like any sensitive data, such as chat logs or personal information, to be encrypted on our server using the best techniques possible. (HTTPS, 512-Bit Encryption, etc)
8. As a developer, I would like to have a fast and automatic recovery if the site goes down for any reason.
9. As a user, I would like the website UI to be mobile ready (Time Permitting).
10. As a user, I would like the website to have a 2 second or less loading time for all pages.

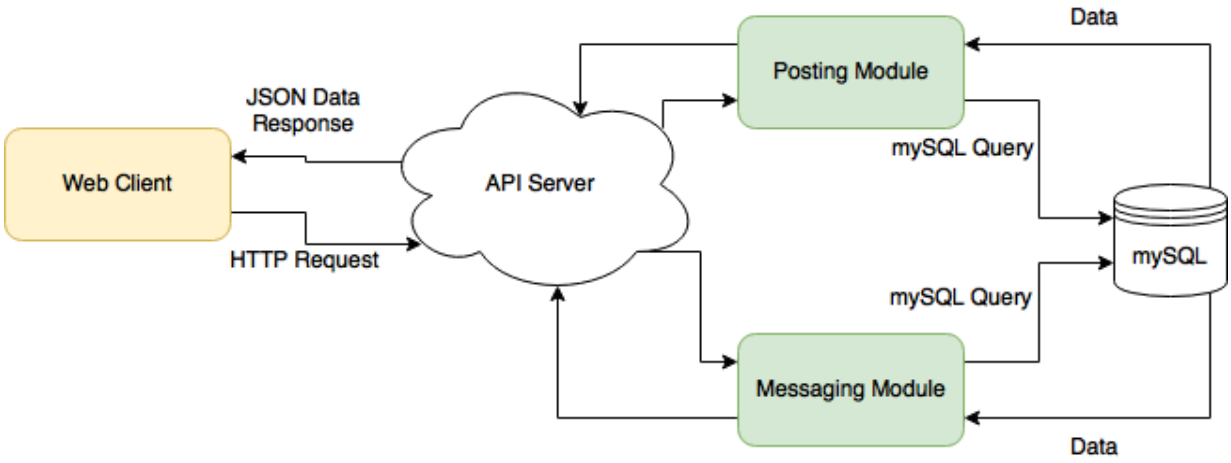
Design Outline

Our project provides a service to find or sell living spaces, primarily sublets. Our implementation will be a web application using Angular javascript to produce the frontend. The backend will be primarily in .NET Core, while using python for scraping Facebook or local apartment subleasing sites.

Components

A. Client
   a. Angular2 web application that will make HTTP requests using JSON to interface with the web server
B. Backend
   a. Scraping Service
      i. This service will aggregate all data that will be scraped from other websites. These sites would include things like Facebook Housing Rent/Sublease, BoilerApartments, etc.
   b. RESTful API
      i. This will be where the client application will request resource data from our different databases. All client requests should be routed through this API.
   c. Messaging Service
      i. This is the service that will handle in app communication between users. It will be able to receive, retrieve, and send messages
   d. Database
      i. A single resource to hold user, house, and messaging data.

Architecture Diagram (High Level)



Design Issues

Functional Issues

Issue: Should users be able to post and comment on subleases anonymously?

- **Option 1**: Allow users to browse anonymously.
- Option 2: Force users to sign in to view posts to be able to collect statistics on who is visiting the post.

  Decision: Users should be able to browse anonymously since we do not want to turn people away from the site. Even though we as developers might not be able to track every piece of information about that user, we can still track how many people are visiting this post.

Issue: Should buyers be allowed to make posts?

- Option 1: Buyers should be allowed to post to allow them to find a sublease quicker.
- **Option 2**: Buyers should not be allowed to post because it would cause sellers to compete for publicity.

  Decision: We decided that buyers should have exclusive rights to make posts, otherwise it would cause the page to be cluttered. For example, there is a high likelihood of having many more buyers than sellers, so the ratio of selling posts to buying posts would be very low.

Issue: What are the limits on addresses?

- Option 1: Sellers should be able to post any address
- **Option 2**: Sellers should only be able to post addresses in the greater-Lafayette area.

  Decision: We're limiting the scope to the greater-Lafayette area. This is to restrict the user base to not allow for sublets from different locations populating our results.

Issue: What anonymous information should we allow sellers to include with their post?

- **Option 1**: A user would be able to view age, gender, college major (if any), and personal preferences.
- Option 2: A user would be able to view age and gender.

Decision: We are leaving out personal preferences since this might reveal too much about the current tenants. Since there are various laws about protecting personal privacy, it would be a grey area to keep personal preferences. Age and gender would be the most amount of information we can show without compromising the tenant's privacy.

Issue: Should the search have a map?

- **Option 1**: A google maps view with pins on the locations of search results
- Option 2: No map

Decision: Maps are useful when looking for somewhere to live. People using the application might consider that a useful tool to figure out distances to campus or other locations around the greater-Lafayette area.

Issue: How should the posts be displayed?

- Option 1: Posts are listed based on what is getting the most attention, or what is popular in a shorter timespan.
- **Option 2**: A grid containing images and single line descriptions on the left, and a map on the right showing the locations of available sublets.

Decision: A grid with a map is more appealing to the average user and easier to use. Using a grid relieves us of the risk of sublets not being seen due to a short window of time to give it publicity. Going with the grid allows us to display sublets that tend to be more desired. The map also allows us to give a visual representation of the amount of sublets and where they might be located. If we were to go with the chronological or popularity list, a sublet that has been open for a month may be buried in newer postings.

Issue: How should users communicate?

- **Option 1**: In app messaging
- Option 2: Via a form of contact info from a user’s profile

Decision: We could force a user to message another user by using contact info from their profile. However, creating an in app messaging system will allow users to have a more seamless experience, as well as reduce the amount of personal information made available to other users.

Non-Functional Issues

Issue: What type of architecture should our application backend use?

- **Option 1**: Service Oriented Architecture (SOA)
- Option 2: Monolithic Architecture

  Decision: We believe that for our application it makes the most sense to use a SOA instead of a monolithic backend. We believe this is the best option since it will allow us to split the different type of site scraping we are using, as well as our messaging service. If we were to use a monolithic architecture, these resources would all be contained in one server, which could cause scalability issues. Additionally, it will help us separate areas of concern so different teammates can work on different services concurrently without conflict from other team members.

Issue: How should messaging be implemented?

- Option 1: Utilize a Software as a Service (SaaS)
- **Option 2**: Create a chat client in-house

  Decision: Creating a chat client will allow us to have more control over the service as well as less cost.

Issue: Which continuous integration should be used for site deployment?

- **Option 1**: Docker
- Option 2: Jenkins
- Option 3: TeamCity
- Option 4: Kubernetes
- Option 5: Circle CI
- Option 6: Travis CI

  Decision:

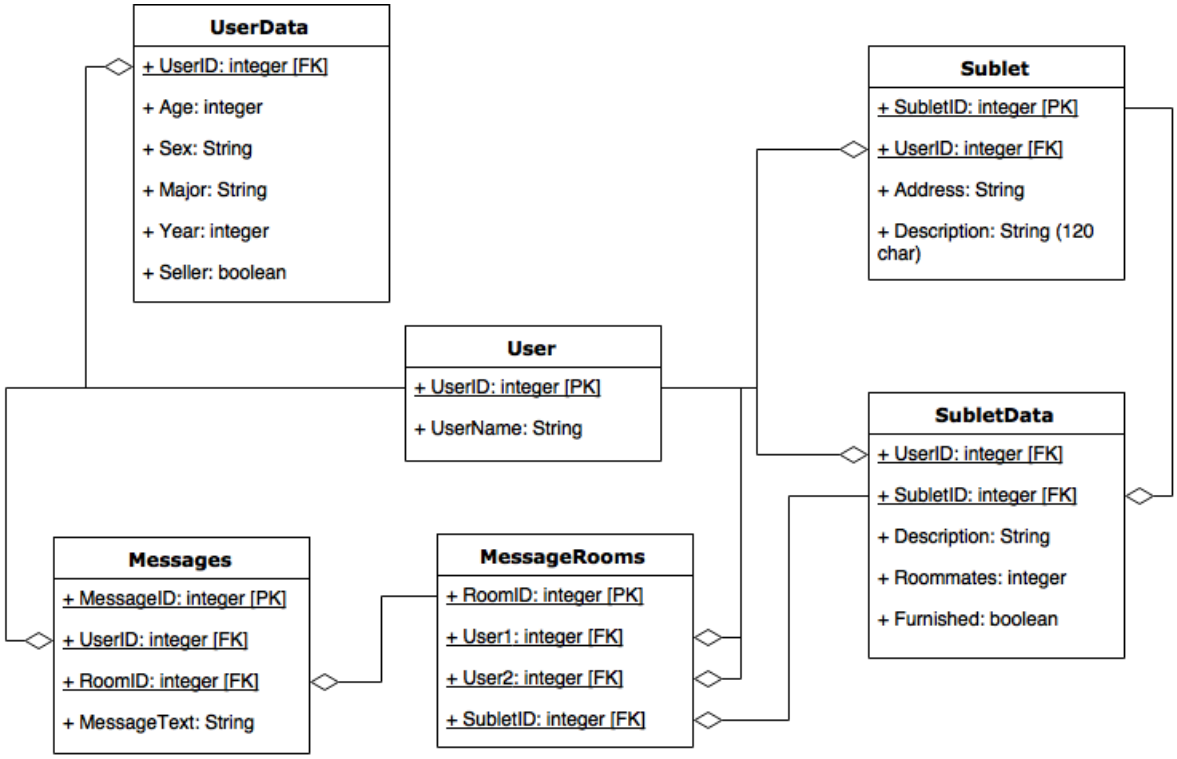
Issue: Should we follow test driven development?

- **Option 1**: Yes
- Option 2: probably

  Decision: We will be following the TDD methodology for the backend mostly. In the front end, we will mostly conduct functional testing for UI based code.

Design Details

Database Schema



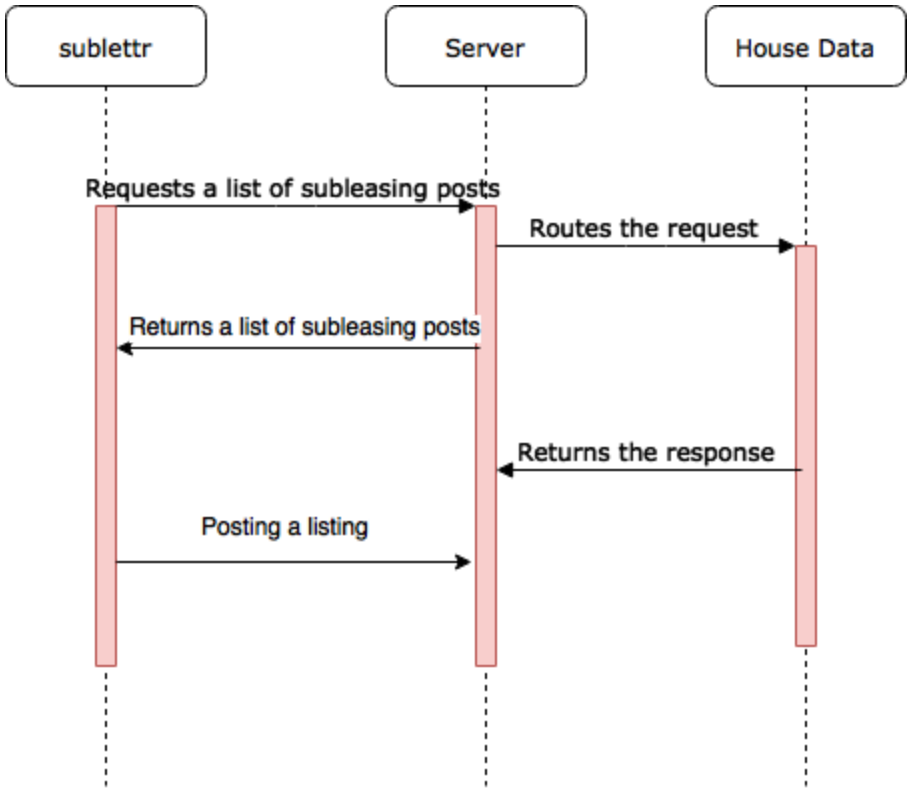
Database Table Descriptions

1. User
    a. Users are saved under this table.
    b. Determines user name.
2. User Data
    a. Data for the user table is stored here.
    b. Determines if the user is a seller or not.
3. Messages
    a. Holds messages that users can use in messaging rooms.
    b. Determines what text is sent in the messaging room.
4. Message Rooms
    a. Holds two users for a chat room.
    b. Tells which sublet the users are messaging about.
5. Sublet Data
    a. Holds information about sublets.
    b. Created by sellers and viewed by buyers.

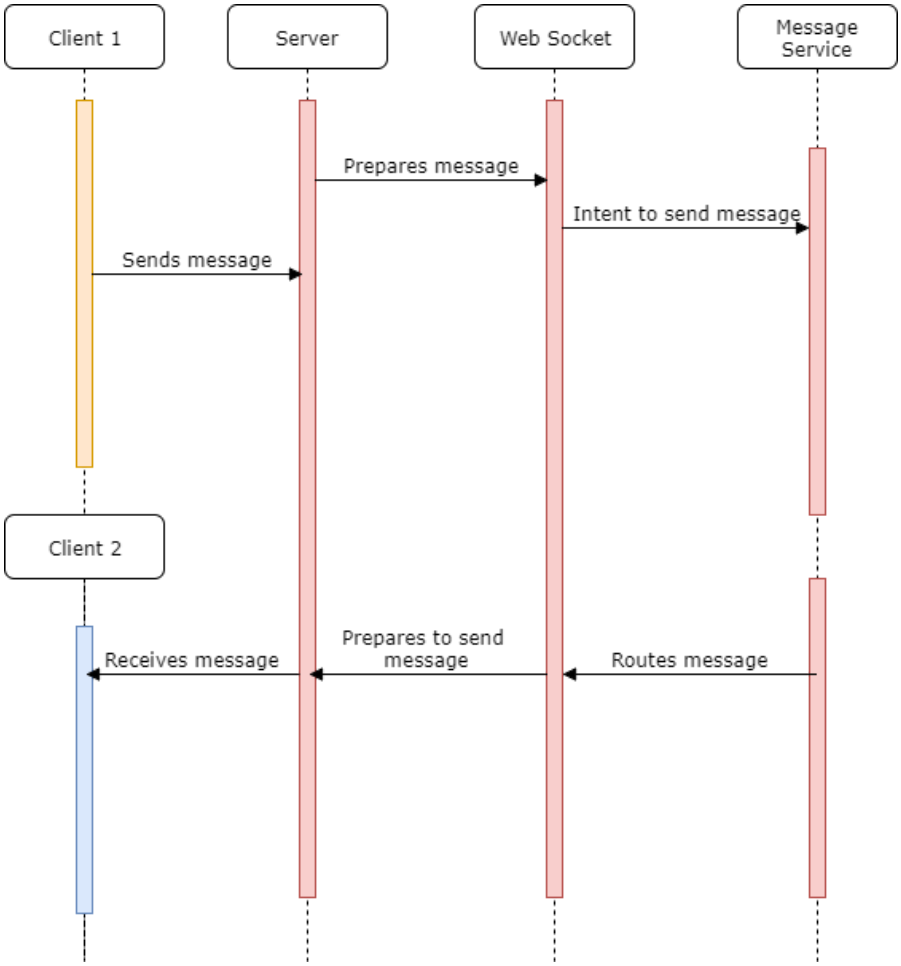
6. Sublet
    a. Table which holds the actual sublet id and basic information that's not held in the sublet data, such as address and descriptions.
    b. Created by sellers.

Sequence Diagrams

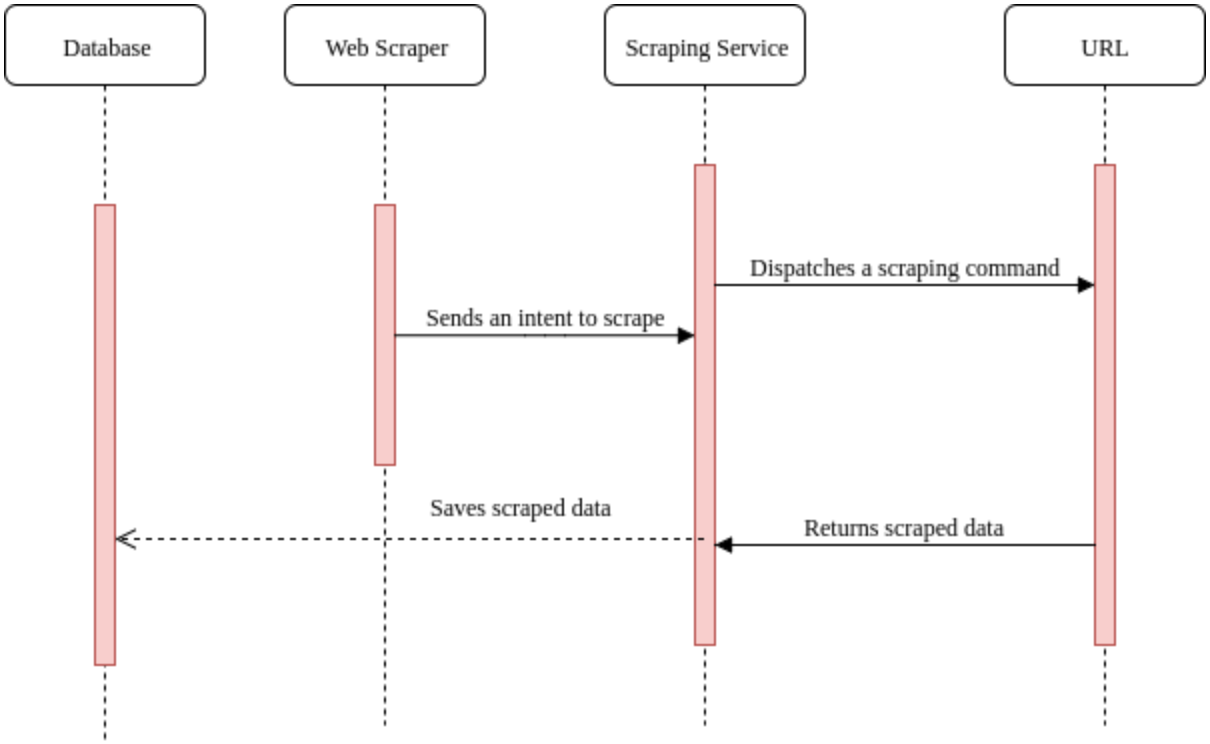
- **Sequence of events when the user posts a listing**



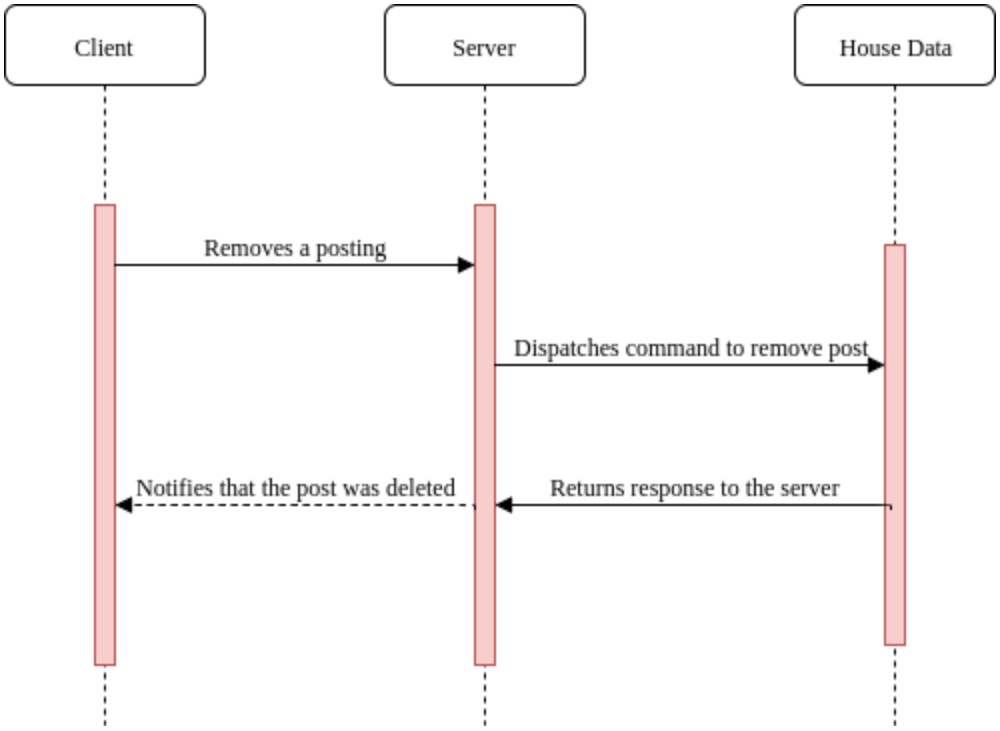
- **Sequence of events when a user messages a seller**



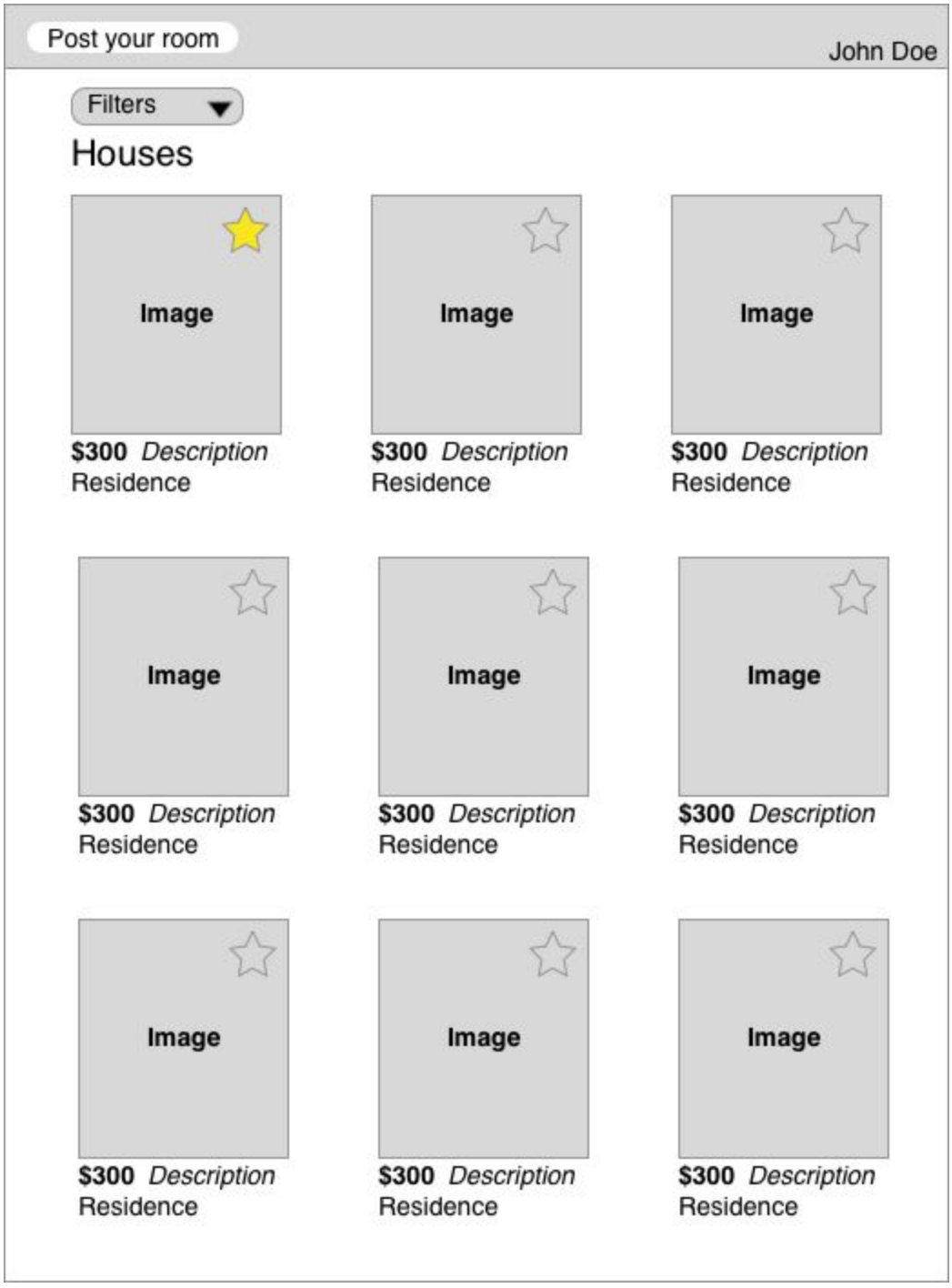
- **Sequence of events when the server scrapes for information**



- **Sequence of events when the user removes a listing**



UI Mockup


Post your room
John Doe
Filters
Houses
Image
$300 Description
Residence