Student Housing Solution (SHS) Report 1 (Group 8)

-Ansh Gambhir, Rishi Shah, Kyle Tran, Srinivasniranjan Nukala, Michael Giannella, John Yager, Rishab Ravikumar, Sneh Shah, Ketu Patel, Sahil Patel

https://mgiannella.github.io/StudentHousingSolution-Landing/

Submitted 2/23/20

Individual Contribution Breakdown

Everyone contributed equally to this report (10% each).

Parts	Rishi	Kyle	Sneh	Ketu	Sahil	Michael	Rishab	John	Srinivas	Ansh
Project Management	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
Sec 1: Problem Statement	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
Sec 2: System Reqs.	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
Sec. 3: Func Reqs. Spec.	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
Sec. 4: UI Specs	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
Sec. 5: Domain Analysis	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
Sec. 6: Plan of Work	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%

Table of Contents

Customer Problem Statement

Problem Statement

Tenant Searching for Housing Problem Statement:

As a student at Rutgers University in New Brunswick, the majority of my peers, particularly upperclassmen, do not live in on-campus housing. Many commute from home, but a large number choose to live in rented houses and apartments in the New Brunswick area ("off-campus"). For students, this move allows for greater flexibility and autonomy: housing can go up to 10 bedrooms, are equipped with kitchens and bathrooms, and do not fall under the formal authority of Rutgers University Residence Life. The move off campus symbolizes a step further towards adulthood, but the process of finding and securing a residence is confusing, outdated, and unreliable. The current solution is using a third party, Rutgers-sanctioned website, "places4students". As essentially the only website and source of off-campus listings, places4students plays a huge role in connecting thousands of potential student tenants with landlords. Unfortunately, the site is outdated and confusing: navigating the site and search pages is cumbersome, the search feature is broken and ineffective and implementing the filters and map features, and there is no standardized way to contact landlords to set up a viewing. As a student searching for off-campus housing, an already overwhelming process is made worse by the ineffective tools at my disposal. I should be able to search housing options by number of beds/bedrooms and bathrooms, distance from campus, and other important factors. The website should be easy to navigate and effectively show me the results of my search on a map. Current solutions do not provide any sort of notification system, so I have to manually check the site every day for new listings that may match my criteria. Additionally, many listings are outdated, have no pictures, and are unreliable. When I do find a listing I am interested in, the methods for contact landlords are varied; some provide emails, others texts, others don't respond. A standard way to schedule tours of houses and apartments would be a huge step forward for prospective tenants to make sure we are informed consumers. As the main gateway to off-campus living, places4students does not adequately serve the Rutgers University student population.

Landlord Seeking Tenants Customer Problem Statement:

As a landlord owning multiple properties in New Brunswick, New Jersey, there is really only one option for advertising my houses for students to rent. This current implementation, Places4Students, feels outdated and does not make listing my properties any easier.

First of all, there is no automation and I need to manually schedule appointments through email or phone in order to find a time that works best for both myself and the prospective tenants. This process can take anywhere between a couple days to weeks because the communication goes back and forth many times. This usually causes a lot of confusion between the students and myself. I believe having a website where my listing can be seen and having a calendar or scheduling program built into the listing would be an extremely convenient feature to have. I would also like the scheduling feature to be customizable where I will be able to specify what times I am available to give out tours to students and remind the students through email or text one day or a couple hours before the actual tour. This would save me a lot of time because I have a busy life balancing my day job and landlord duties.

The current website does not let me manage multiple properties in an efficient manner. I need a platform that allows me to manage everything from one landing page. Here, I can see the page views as well as whether or not students are actually interested in my property. This latter feature would especially help me because it would allow me to go ahead and schedule house tours and other such meetings with potential tenants. On my properties page, I should also be able to see the reviews left on the properties by past tenants so that I can improve upon them. This will significantly help me in renting out the house in the future.

Another feature I would like is the ability to provide important information about the amenities of the house such as the number of bathrooms and bedrooms, the max occupancy limit, parking spots, etc. I would also like to be able to post pictures and videos of the house. The videos will especially help prospective tenants get a better feel of what to expect out of the

property. This would help save a lot of time on both my end and the tenants because the tenant would be able to research all this information before deciding whether or not they actually want to schedule an official house tour.

Landlord Has Tenant Problem Statement:

As a landlord in New Brunswick who has multiple properties and multiple tenants, collecting of all the payments becomes difficult. Tenants split the rent between their roommates and prefer to send checks separately. However, with all the different properties I own, keeping track of all the checks and making sure I receive them on time is very inconvenient. Therefore, having a website where all the tenants can pay rent would really help me stay organized and help me make sure I am receiving all payments in a timely manner. This would also help me fine the tenants who are always late with their payments.

Additionally, having a feature where I would be able to look at and manage all my properties would be really helpful. This would allow me to check on the status of all my properties, making sure they are up to date and maintained properly. If there are any issues with the property, it would be a lot easier to see the issues on the website than having to communicate with the tenants back and forth until the issue is fixed.

Tenant Has Housing Customer Problem Statement:

As a tenant, I find that traditional methods of communication between myself and my landlord are ineffective because I am a student that has a busy school schedule. Thus, finding an appropriate time to meet with my landlord is very difficult. The website ‘Places4Students’ only provides me minimal contact information, such as their phone number and email, however, I would like to know more about their weekly schedule in order to easily schedule an appointment with my landlord that which I know will work for both of us.

I would feel it is beneficial for both of us if we were able to handle all of the payment processes before our meetings in order to make our meetings more productive. Currently, I

would only be able to pay my rent and other expenses through mail or in person, which tends to take quite some time to be processed. I believe it would be better if the payment process for my rent, utilities, and security deposits could all be automated and all handled online through a site. Places4Students does not allow me to do this action, which inevitably makes the entire process of handling payments more difficult.

Currently, scheduling a maintenance request requires me to call the landlord, who may not always pick up, but even if they do there is a possibility that they may not remember the conversation or the scheduled date for the request. I would appreciate it if I was able to document the meeting online so that it is set in stone for the landlord and me to be constantly reminded of. However, if the maintenance requests are not able to solve the issue I am having in my house, I would feel obligated to write a review of the property to let future tenants know about the issue. It would be very beneficial if there was some sort of online rating system so that I can let others know about the quality of the property.

Additionally, another thing that I would like to access online would be the housing agreement between myself and the landlord. That way, if I want to check on any rules and regulations I must follow for my house, I can easily view a pdf document of my housing agreement on a site. Rather than having to access a picture of it sent by my landlord, which is sometimes blurry in some parts, the online document can be viewed at any time, since it would always be accessible to me.

If all of these features were accessible in one easy to use and friendly site, it would make my life a lot easier because I would not have to worry about dealing with my landlord on top of my current student responsibilities.

Glossary of Terms

Landlord Seeking Tenants - User who owns property(s) and is seeking to find residents for that property(s).

Prospective Tenant- User who is seeking to rent out a property owned by a landlord.

Landlord Has Tenant - User who owns property(s) and has tenants occupying his property(s).

Current Tenant - User who is currently renting out a living space owned by a landlord.

Property- Place of residence that tenants live on.

Rent - Monthly payments that tenants have to make to landlords.

System Requirements

User Stories

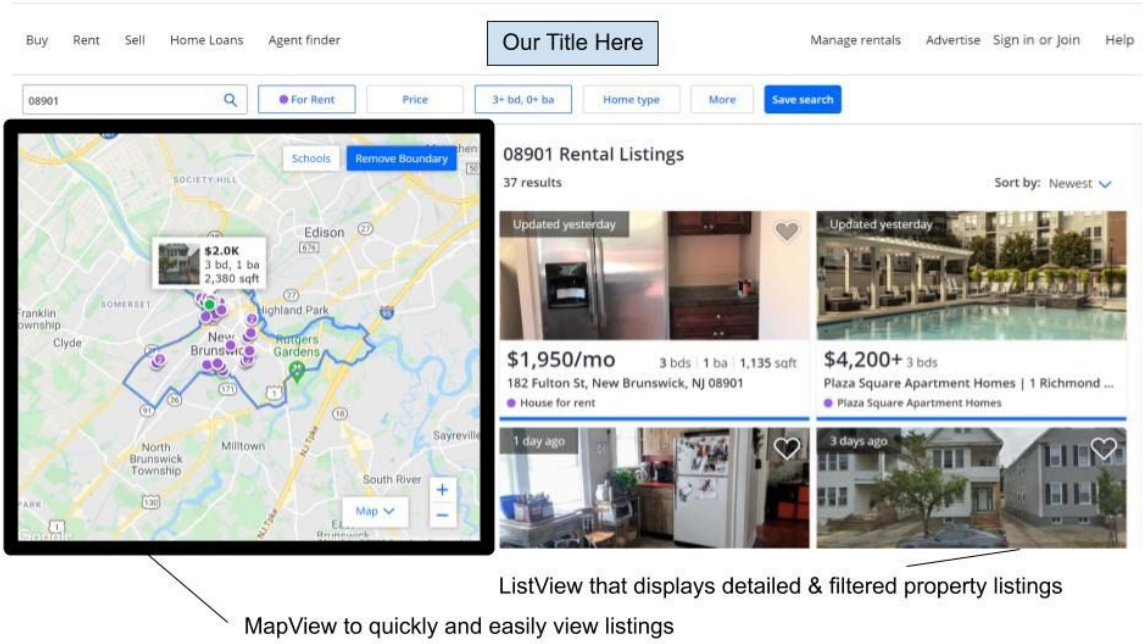
Requirement Number	User Story	Size (out of 10 points)
REQ-1	As a prospective tenant, I will be able to view the results of my search as a list, so that I can quickly scroll through the potential properties.	3
REQ-2	As a landlord, I should be able to receive rent and utility payments online.	8
REQ-3	As a tenant, I should be able to have a safe, quick, and automated way to pay my rent/utilities/deposits.	8
REQ-4	As a landlord, I should be able to manage all my properties online from one page.	3
REQ-5	As a tenant, I should be able to leave a rating on the property that I am renting.	3
REQ-6	As a landlord, I should be able to see analytics for my properties such as pageviews and interested students.	5
REQ-7	As a landlord, I should be able to upload pictures and video tours to my listing.	4
REQ-8	As a prospective tenant, I should be able to search seamlessly, easily selecting and deselecting filters without having to refresh the page each time so that I can have an uninterrupted user experience.	6
REQ-9	As a prospective tenant, I will be able to view pictures and details on properties including amount of bedrooms, bathrooms, maximum occupancy, price, ratings, reviews, and amenities so that I have more details about the property.	4

REQ-10	As a prospective tenant, I will be able to view property information in a location based map view so that I can quickly and efficiently view/select properties that are in my ideal location.	7
REQ-11	As a landlord, I should be able to post information about the house such as amenities, and price structure (rent, utilities, etc.).	4
REQ-12	As a landlord, I should be able to schedule any meetings through an online system.	7
REQ-13	As a tenant, I want to have quick and easy access to a copy of my housing agreement.	3
REQ-14	As a tenant, I should be able to easily schedule a meeting with my landlord through the website.	7
REQ-15	As a prospective tenant, I will be able to sort the order of properties shown by distance from campus, price, and date listed so that I can view and find properties easier.	3
REQ-16	As a tenant, I should be able to easily file a maintenance request if something in the house is broken.	5
REQ-17	As a system, I will notify prospective tenants about new properties that fit their filters every week by email to ensure that they have seen all the new listings.	6
REQ-18	As a prospective tenant, I will be able to subscribe/unsubscribe to a weekly digest containing new properties that fit my filters, so that I can make sure I have seen all the new listings.	3
REQ-19	As a prospective tenant, I will be able to compare up to three properties at a time so that I can look at a side-by-side comparison of them including price, and other key pieces of information.	5
REQ-20	As a prospective tenant, I will be able to hover on property location pins so that I can view abbreviated property details to make quick decisions on each property.	4

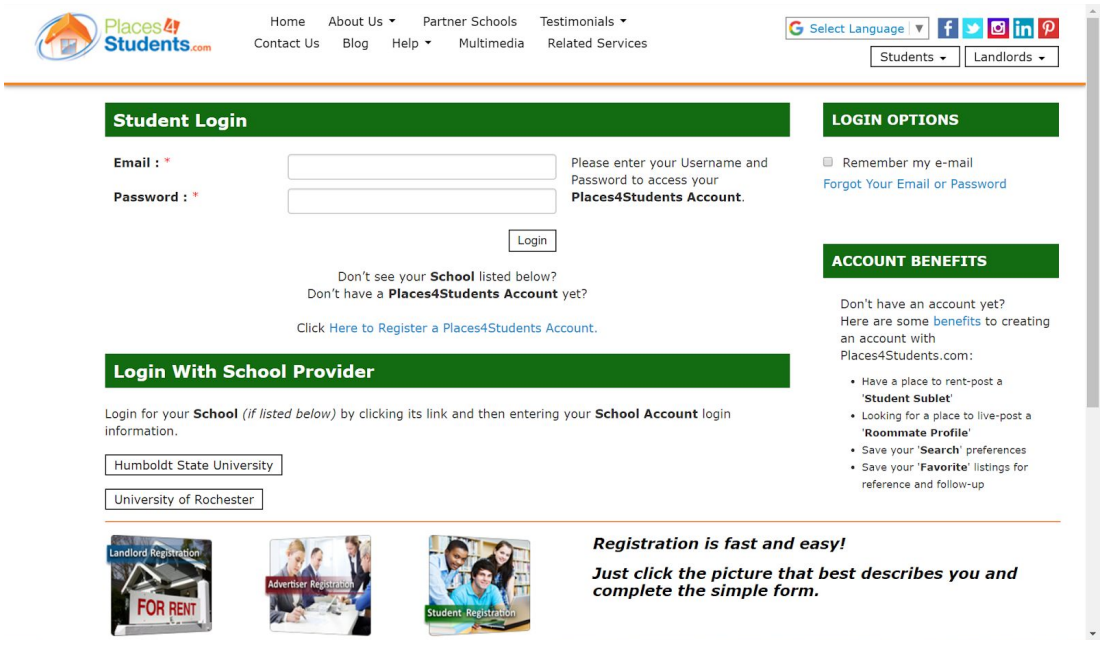
REQ-21	As any type of user, I expect every page of the entire website to contain login and registration buttons, list of features, links to Contact, Help, FAQ, and About Us page, so that I can access all the information I need to.	3
REQ-22	As a tenant/landlord, I have a registration page, so that I can create a new account, and a login page to log in to that account.	3
REQ-23	After logging in as a landlord, I should be taken to a landing page with important information to me so that I can view account information, listing management, listing analytics, notifications regarding listings or scheduled tours.	7
REQ-24	As a landlord, I should have access to a form for adding a listing to the website so that I know what information to include and it is straightforward and uniform.	4
REQ-25	As a tenant, I should be taken to a landing page that lists favorited listings, featured available listings, and account information after I've logged in so that I can easily view my information.	5

User Interface Requirements

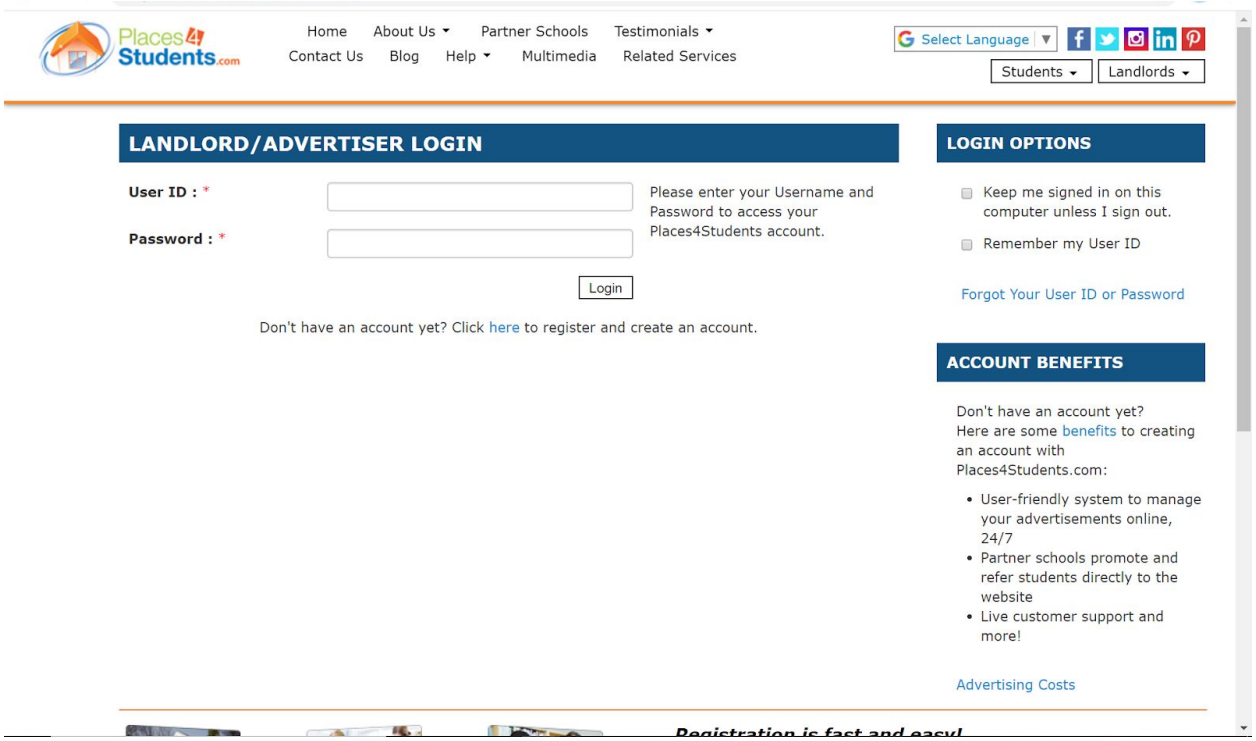
Front page of site, with MapView and listings:



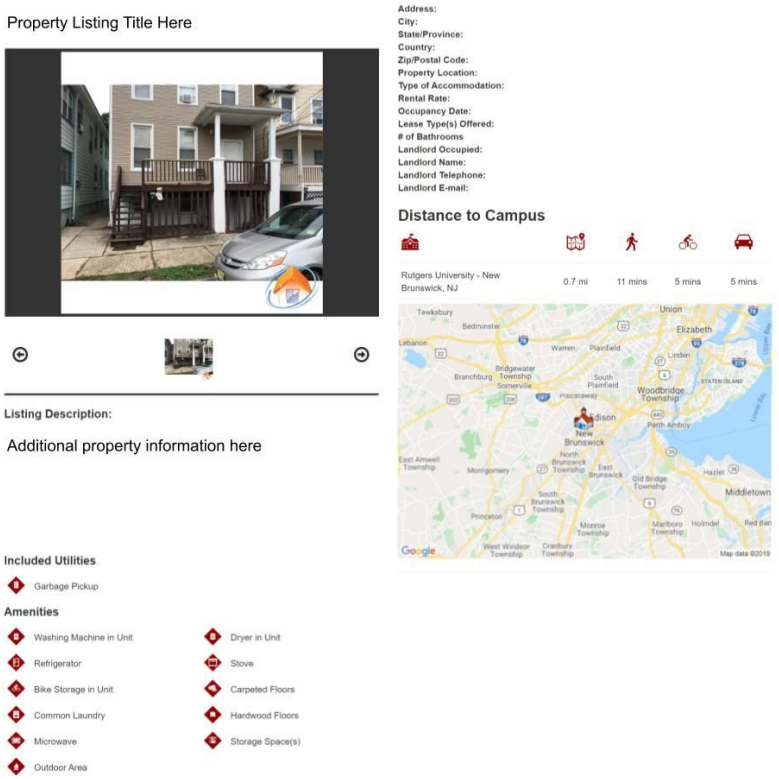
Users logging/registering into the site as a Tenant:



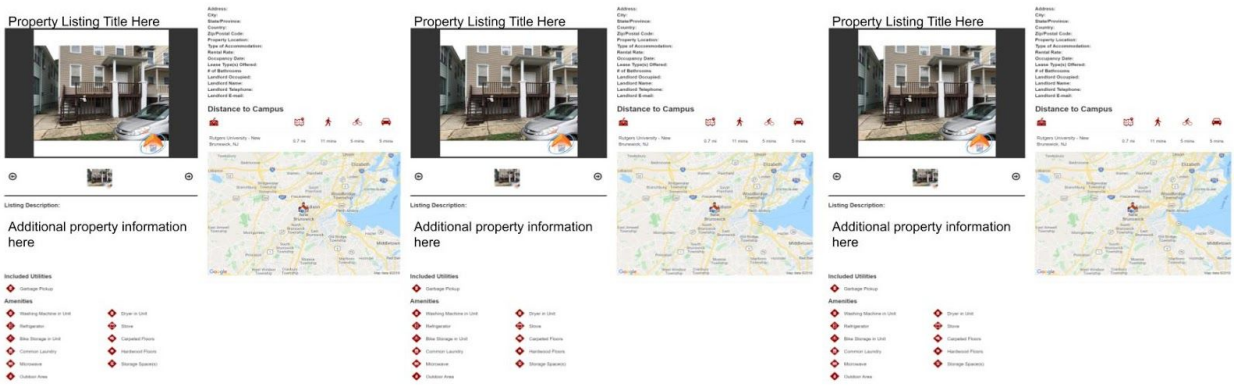
Users logging/registering into the site as a Landlord:



Individual Property Listing Page:



Side by side comparison:



Functional Requirements Specification

Stakeholders

- Landlords looking for tenants
- Landlords who have tenants
- Tenants who don't have housing
- Tenants who have housing
- Students' families
- Rutgers University

Actors and Goals

Actor	Type	Goals
Landlord Looking for Tenants	Initiating	Can create property listings, view page analytics, see schedule
Landlord with Tenants	Initiating	Manage property listings, receive rent payment from current tenants, manage maintenance requests, upload and view lease agreement
Website	Participating	Display information to the user
Prospective Tenant	Initiating	Can view property listings, schedule viewings, and create groups of tenants to lease a property
Current Tenant	Initiating	Submit maintenance requests, schedule meetings, and send landlord rent payment
Database	Participating	Store information regarding users, properties, reviews, etc
Controller	Participating	Facilitate connections between database and website

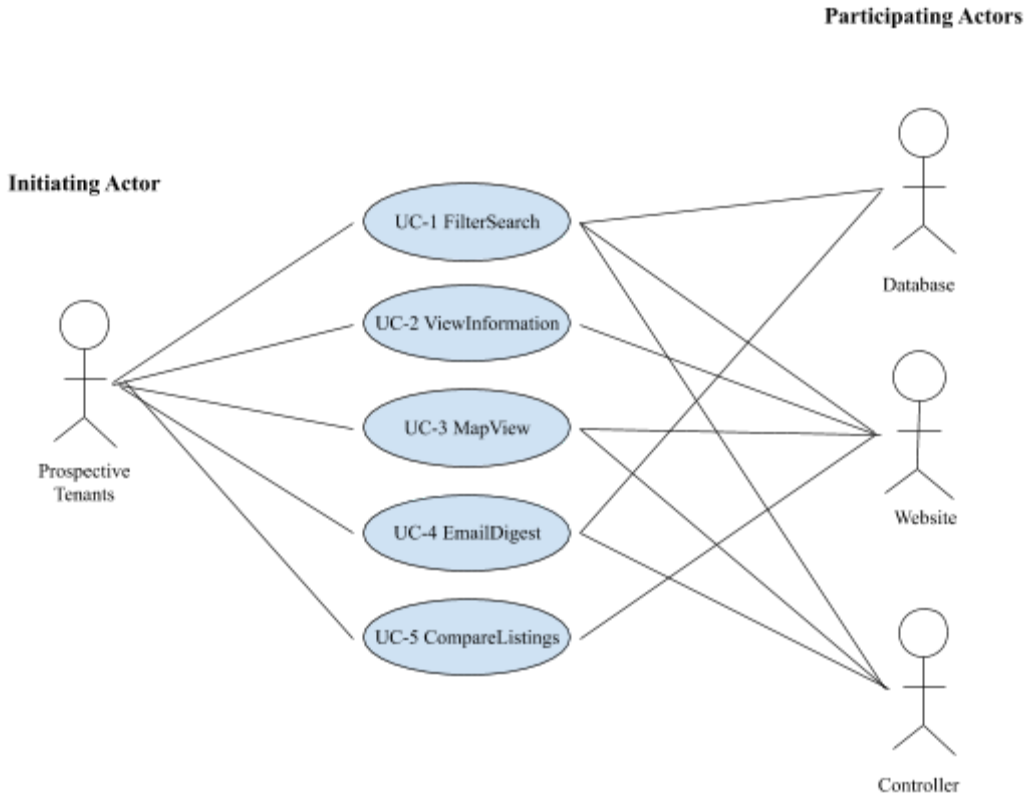
Use Cases

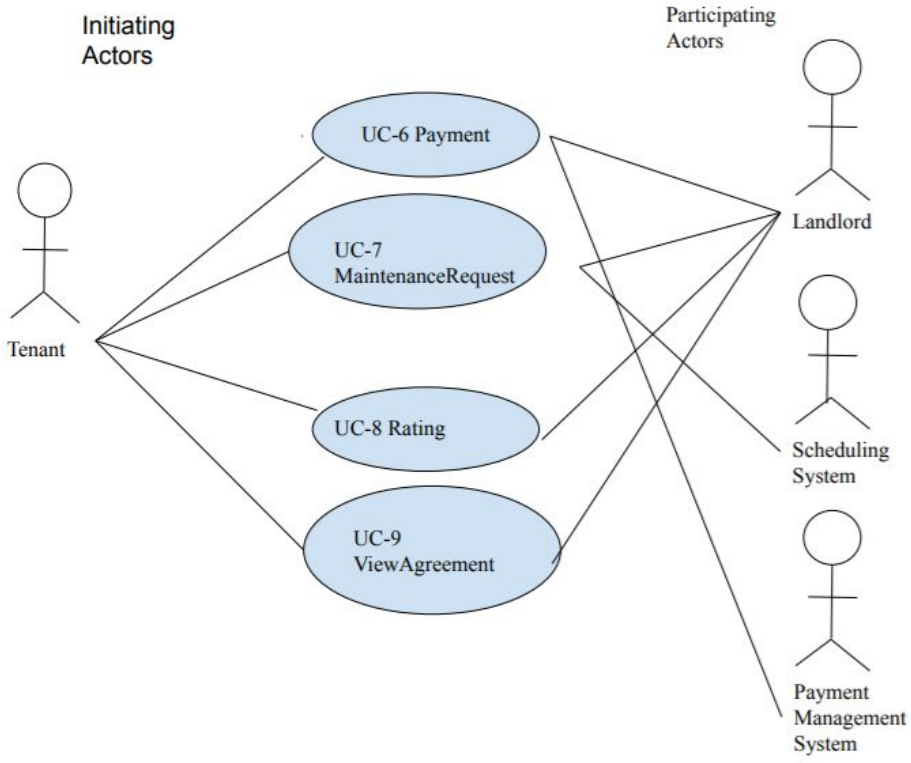
Casual Description

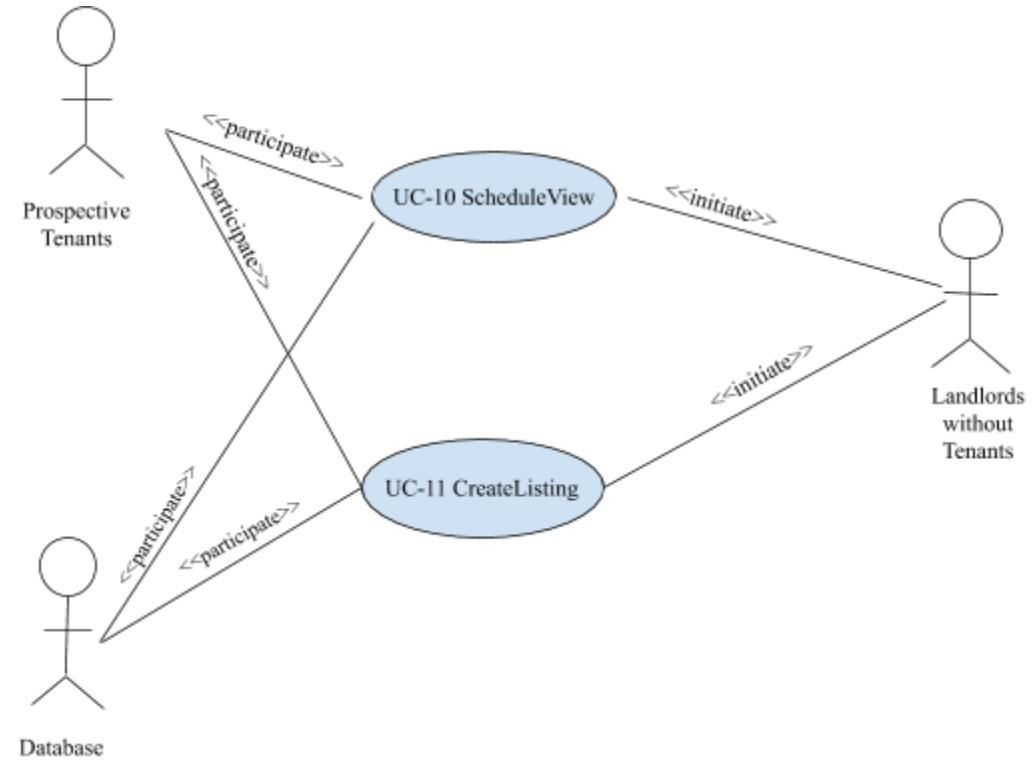
Actor	Actor Goal	Use Case Name
Prospective Tenant	Search for properties using filters.	SearchProperties UC-1
Prospective Tenant	View information on properties, including description, location, amenities, and reviews.	ViewInformation UC-2
Prospective Tenant	View search results on map that auto-populates with results as the tenant moves the map	MapView UC-3
Prospective Tenant	Receive weekly notifications regarding new property listings based on a set of filters	EmailDigest UC-4
Prospective Tenant	Compare property listings by price, amongst other descriptors.	CompareListings UC-5
Current Tenant, Landlord	To allow for a tenant to make safe, quick, and automated online payments of rent/utilities/security deposits to their landlord.	Payment UC-6
Current Tenant	Tenant should also be able to file a maintenance request if something in the property needs to be fixed.	MaintenanceRequest UC-7
Current Tenant	Tenant should be able to leave a rating on the property that they are renting.	Rating UC-8
Current Tenant, Landlord	Tenant should be able to view a digital version of housing agreement that landlord uploads online, and is able to view any updates that are made to the agreement.	ViewAgreement UC-9
Landlord Looking for Tenants	Be able to put in available time slots for prospective tenants to choose from and see any upcoming meetings with tenants	ScheduleView UC-10
Landlord Looking for	Create a new page for a listing with all amenities, photos, videos, location, and any other details	CreateListing UC-11

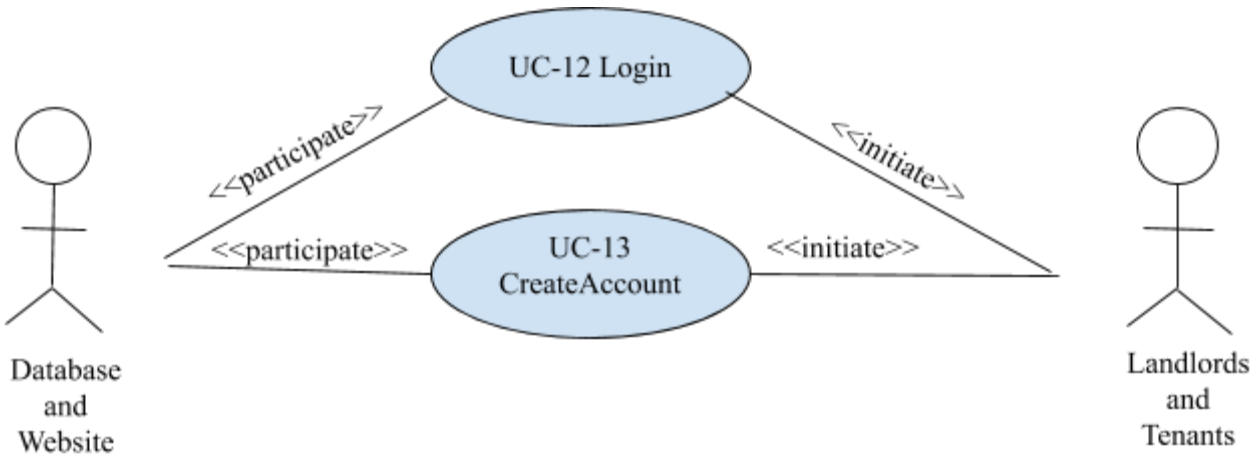
Tenants	regarding the property that he/she is planning to rent out.	
Landlords, Tenants	Input username and password to take you to home page	Login UC-12
Landlords, Tenants	Create account for landlord/tenant when they use the website for the first time; input personal information such as name, phone number, email, etc.	Create Account UC-13

Use Case Diagrams




Initiating
Actors
Participating
Actors
UC-6 Payment
UC-7
MaintenanceRequest
UC-8 Rating
UC-9
ViewAgreement
Tenant
Landlord
Scheduling
System
Payment
Management
System


Prospective Tenants
<<participate>>
<<participate>>
UC-10 ScheduleView
<<initiate>>
Landlords without Tenants
<<initiate>>
<<participate>>
<<participate>>
UC-11 CreateListing
Database


UC-12 Login
<<participate>>
<<initiate>>
<<participate>>
UC-13 CreateAccount
<<initiate>>
Database and Website
Landlords and Tenants

Traceability Matrix

		Use Cases												
Requirements	**Point Weight**	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10	UC-11	UC-12	UC-13
REQ-1	3	x				x								
REQ-2	8						x							
REQ-3	8						x							
REQ-4	3											x		
REQ-5	3								x					
REQ-6	5		x						x			x		
REQ-7	4		x									x		
REQ-8	6	x		x	x									
REQ-9	4	x	x			x			x					
REQ-10	7	x		x										
REQ-11	4		x			x						x		
REQ-12	7									x	x			
REQ-13	3							x		x				
REQ-14	7							x			x			
REQ-15	3	x												
REQ-16	5							x						
REQ-17	6				x									
REQ-18	3				x									
REQ-19	5					x								

REQ-20	4			x										
REQ-21	3	x	x	x		x	x	x	x	x	x	x	x	x
REQ-22	3												x	x
REQ-23	7											x	x	
REQ-24	4											x		
REQ-25	5												x	
Weight Total	120	26	20	20	15	19	19	18	15	13	17	30	18	6

Fully Dressed Descriptions

Use Case UC-1: SearchProperties

Related Requirements: REQ-1, REQ-8, REQ-9, REQ-10, REQ-15, REQ-21

Initiating Actor: Prospective Tenant looking for Housing

Actor's Goal: Search for properties using filters

Participating Actors: Website, Controller, Database

Preconditions: Prospective Tenant has an account on the website.

Postconditions: Prospective Tenants can find properties by searching using filters that they are seeking. Prospective Tenants can also sort their search results by a multitude of factors.

Flow of Events for Main Success Scenario:

1. Prospective Tenants click on the search link in the navigation bar.
2. They select filters based on property listing details (price, # of bedrooms, etc.)
3. Listings are retrieved from the database and displayed to the Prospective tenant in a user-friendly way narrowed down by the specified filters.

Flow of Events for Extensions (Alternate Scenarios):

1. Property listings do not contain enough information about the property, or there aren't any properties fulfilling the user's filters
2. Search feature retrieves properties that best fit the user's specified filters, and then shows the listings that partially fit the user's requirements
3. Users are asked to expand their filters to view more properties, or if there are no properties that fit their filters.
4. Users can select a sort factor (price: high to low, price: low to high, etc.), to help them find what they are looking for faster.

Use Case UC-7: MaintenanceRequest

Related Requirements: REQ-14, REQ-16, REQ-19

Initiating Actor: Tenant

Actor's Goal: Inform the landlord of any maintenance needed for the property.

Participating Actor: Landlord

Preconditions: Landlord has tenants for properties.

Postconditions: Properties are maintained and the system is notified.

Flow of Events for Main Success Scenario:

1. Tenant logs a maintenance request on the system and the system notifies the landlord.
2. Landlord handles the request.
3. Landlord updates the maintenance request on the system.

Flow of Events for Extensions (Alternate Scenarios):

1. After Step 3, if the Tenant is not satisfied with the service, the Tenant can update the request on the system.
2. The landlord is notified again, and deals with the request accordingly.

Use Case UC-10: ScheduleView

Related Requirements: REQ-12, REQ-14, REQ-21

Initiating Actor: Landlords looking for tenants

Actor's Goal: Be able to put in available time slots to meet for housing tours so that prospective tenants can choose from them

Participating Actors: Prospective Tenants, Database

Preconditions: Landlord has an account on the website, as well as properties that are available to be rented. Tenant has an account on the website and is searching for properties.

Postconditions: Prospective tenants and landlords who need these tenants can pick available times where they are free to meet for housing tours. They both can see if they have any upcoming meetings.

Flow of Events for Main Success Scenario:

1. Landlords login and go to their schedule page.
2. Landlords choose times when they are available to meet with potential tenants.
3. Potential tenants select which time slot they want to visit the house.
4. Landlords and tenants can view these selected times and refer back to it in the future so they know when their meeting is.

Flow of Events for Extensions (Alternate Scenarios):

1. Landlords and tenants cannot agree on time.
2. They contact each other using email or phone numbers and agree on separate meeting time.

Use Case UC-11: CreateListing

Related Requirements: REQ-4, REQ-6, REQ-7, REQ-11, REQ-21, REQ-23, REQ-24

Initiating Actor: Landlord Looking for Tenants

Actor's Goal: Create a new page for a listing with all amenities, photos, videos, location, and any other details regarding the property that he/she is planning to rent out.

Participating Actors: Prospective Tenants, Database

Preconditions: Landlord has an account on the website.

Postconditions: Landlords can view information about all their properties (plus maintenance requests and more) from their home page. Tenants can view information about listing on their end as well.

Flow of Events for Main Success Scenario:

1. Landlords login and go to the homepage.
2. They create a new listing by inputting address, pictures, amenities, and other such features.
3. Listing is added to landlord's home page (and the database) and can be seen along with all the landlord's information and other properties.

Flow of Events for Extensions (Alternate Scenarios):

1. Landlords do not post enough information about the property.
2. Tenants contact respective landlords using phone number or email address provided online.
3. Landlord provides the needed information or changes their online posting to satisfy the customer's needs.

Use Case UC-6: Payment

Related Requirements: REQ-2, REQ-3, REQ-21

Initiating Actor: Tenant that has housing

Actor's Goal: To make automated safe and quick online payments for rent, utilities, and security deposits.

Participating Actors: Landlord, Payment review/process system

Preconditions: Tenant is already logged into the system and is renting a property.

Postconditions: The payment was successfully received by the landlord from the tenant.

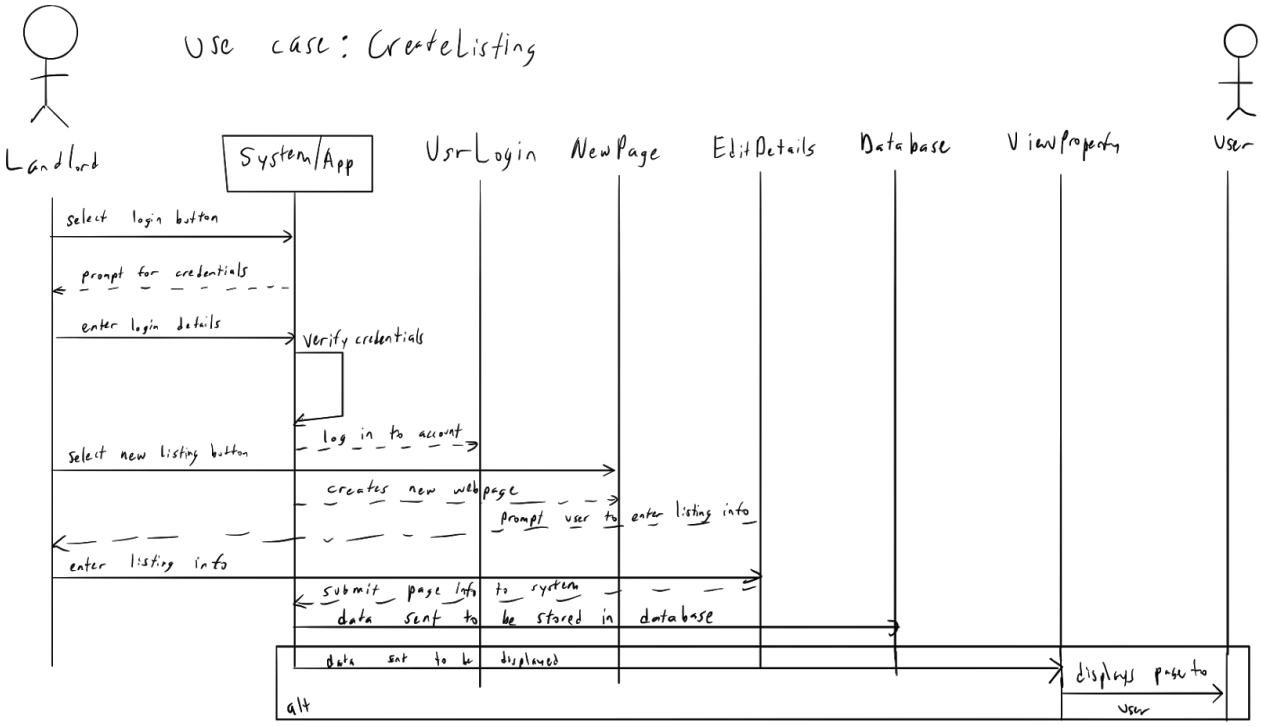
Flow of Events for Main Success Scenario:

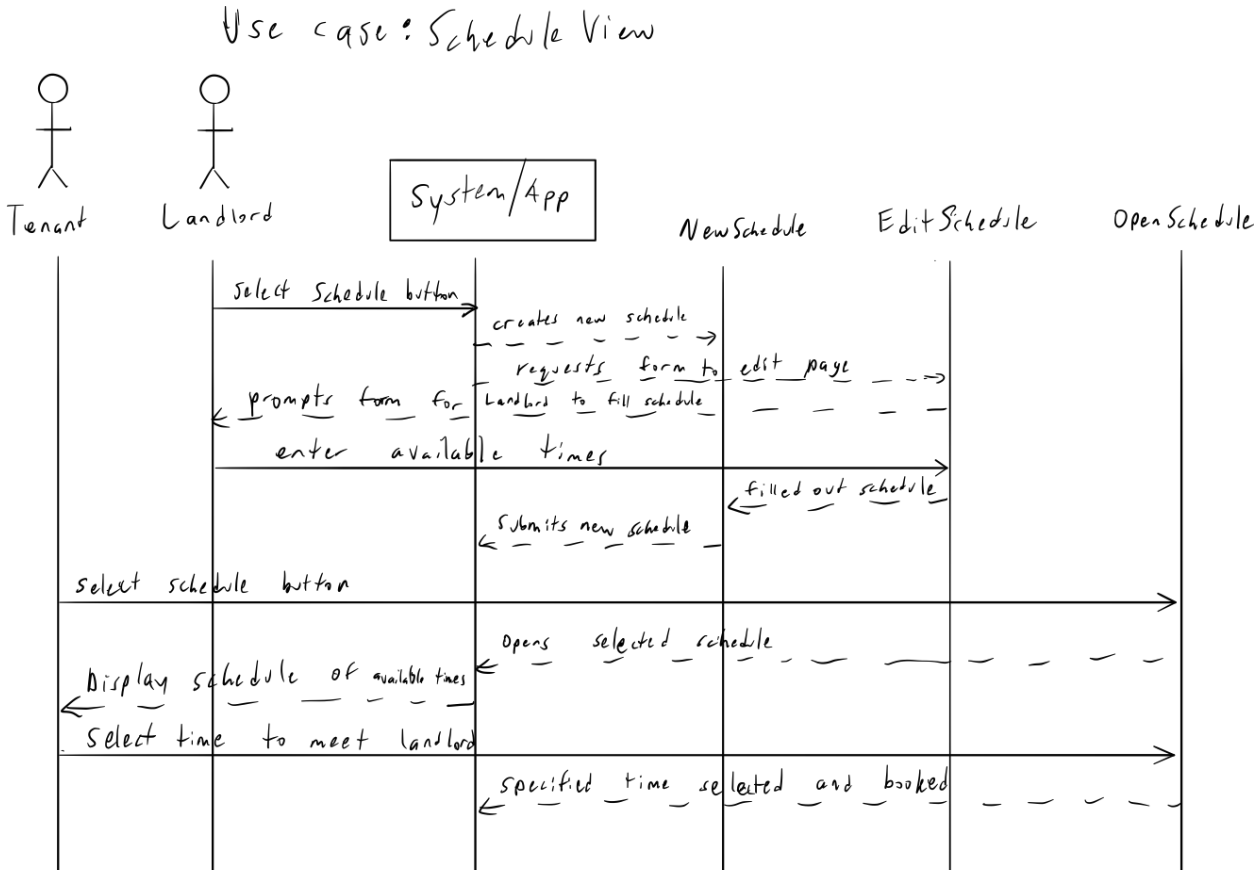
1. Tenant logs into the system and checks to see their pending payments.
2. Tenant enters payment information for the necessary balance.
3. Payment is processed and reviewed, then sent to the landlord.
4. Landlord receives payment, and notifies the system that the payment has been received
5. System notifies the tenant that the payment was received by the landlord

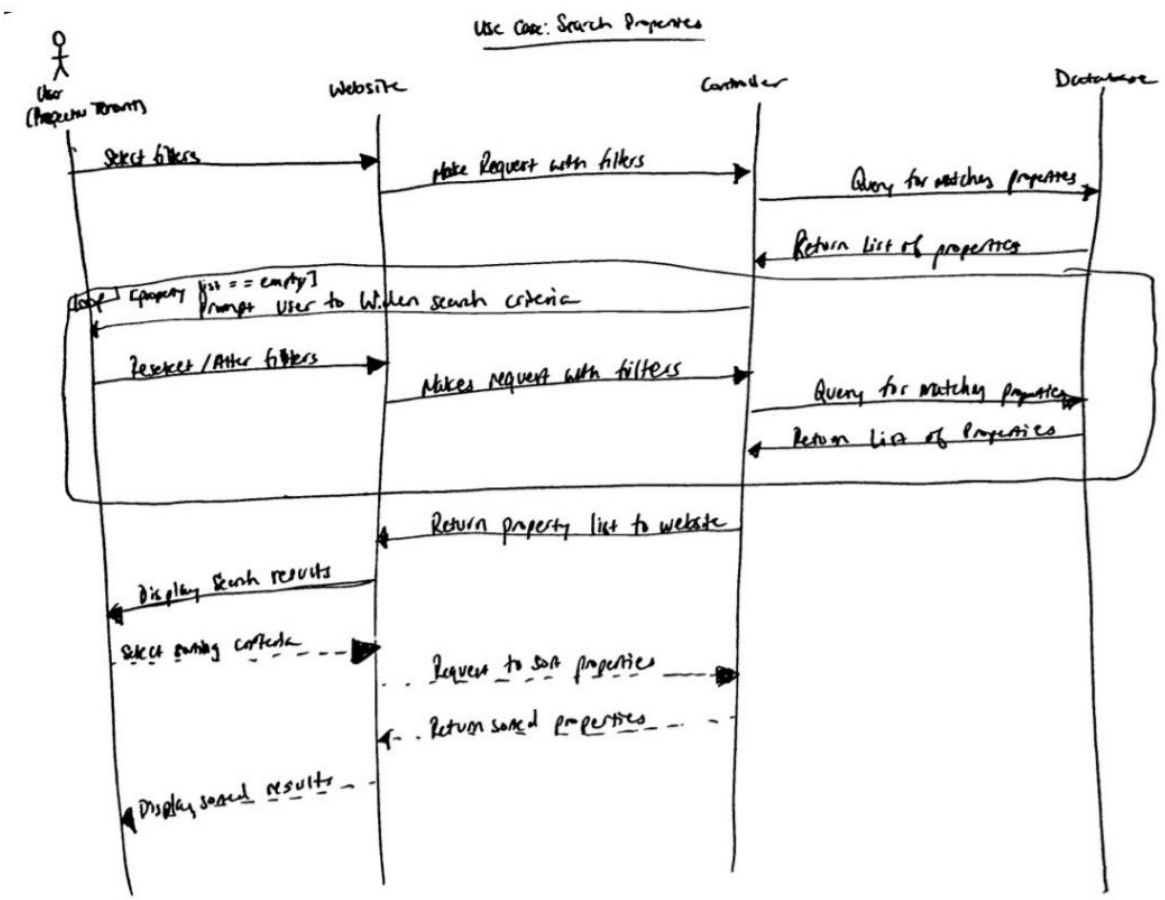
Flow of Events for Extensions (Alternate Scenarios):

1. A. Tenant enters wrong payment into the system, a value that is higher than the required payment
2. A. The tenant will have another chance to review payment, and file any mistakes within the system.

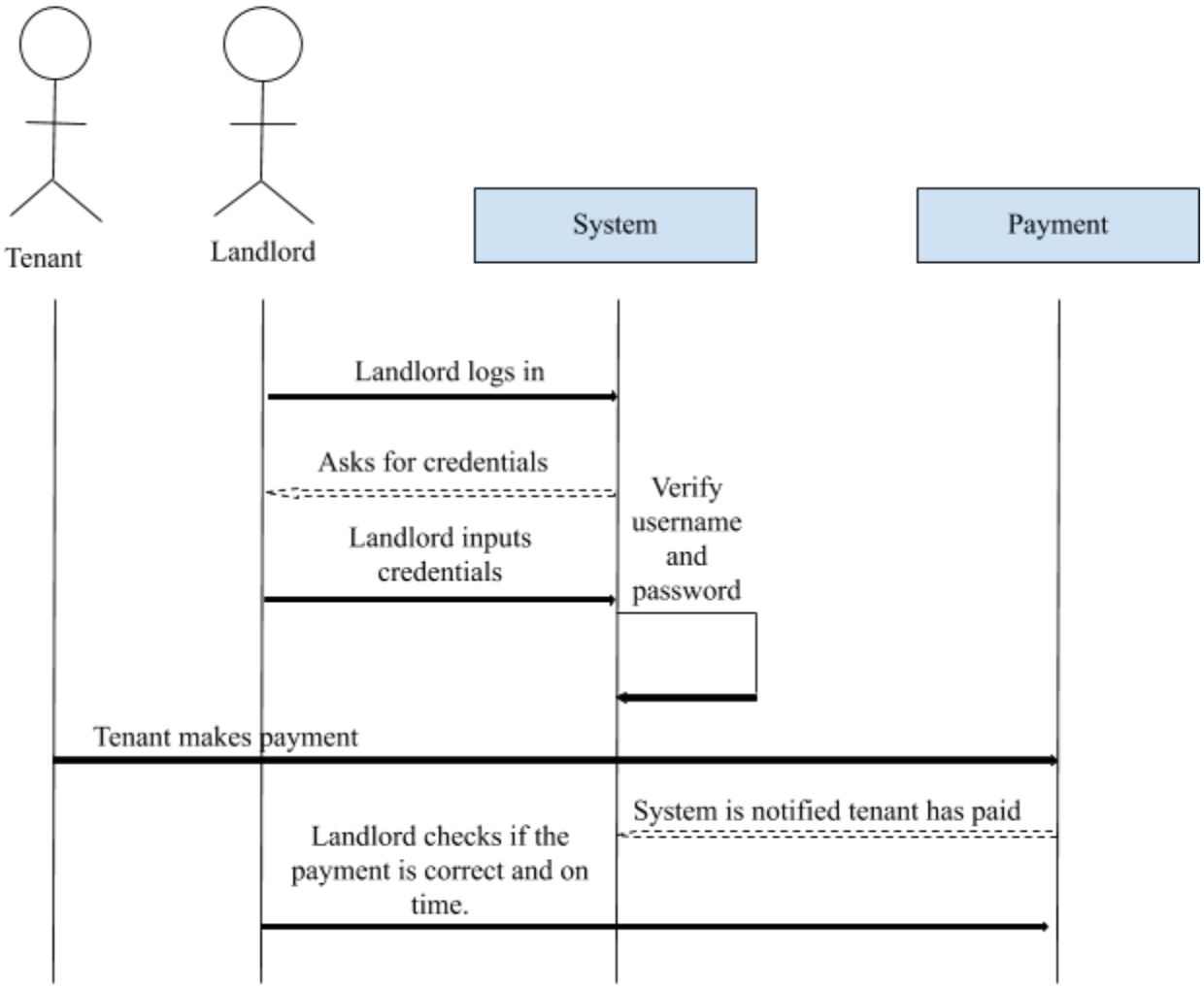
System Sequence Diagrams



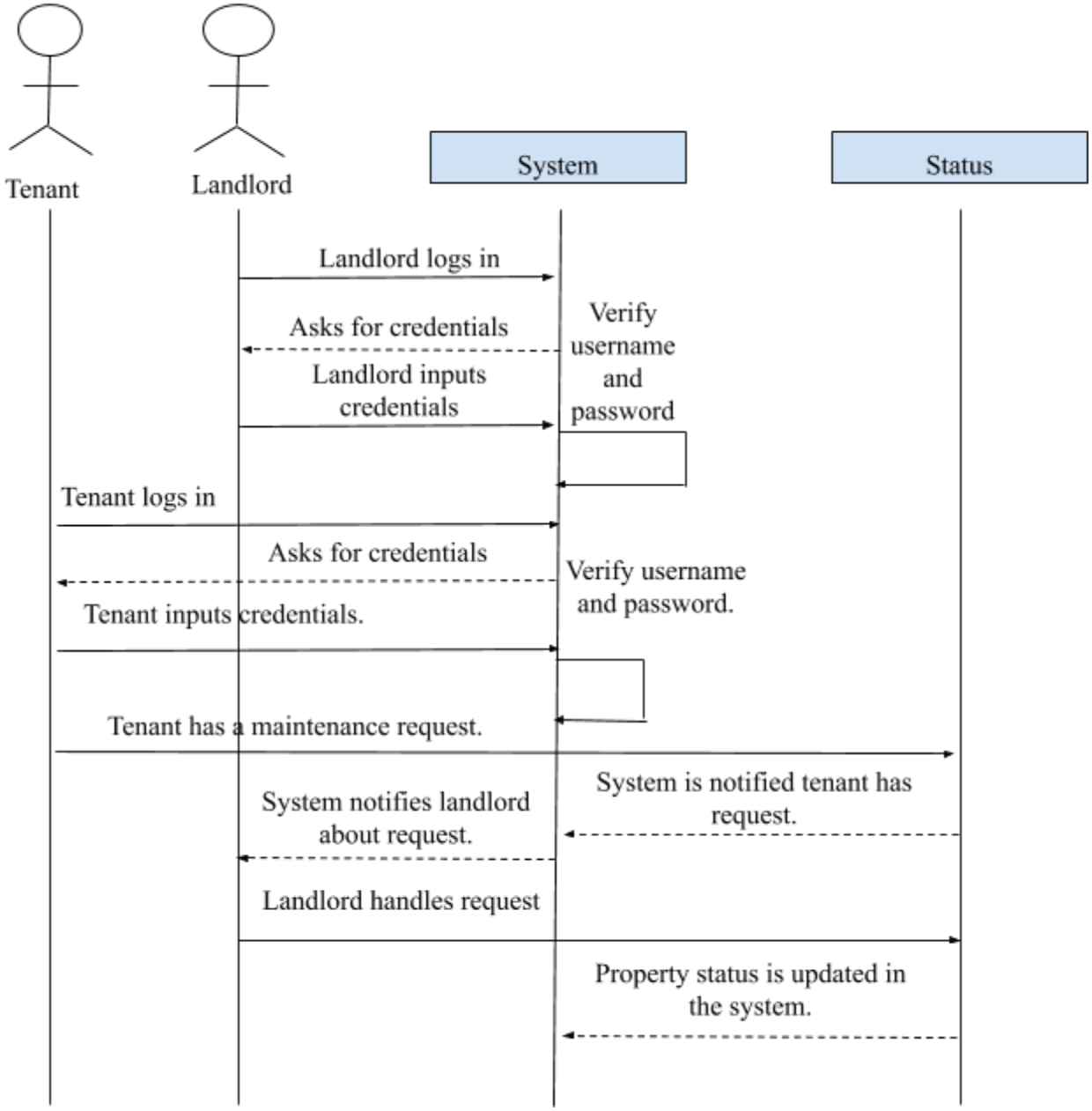



Use Case: Search Properties
User
Website
Controller
Database
Select filters
Make Request with filters
Query for matching properties
Return list of properties
[property list == empty]
Prompt User to Widen search criteria
Reselect / Alter filters
Makes request with filters
Query for matching properties
Return list of properties
Return property list to website
Display search results
Select sorting criteria
Request to sort properties
Return sorted properties
Display sorted results

Use Case:Payment



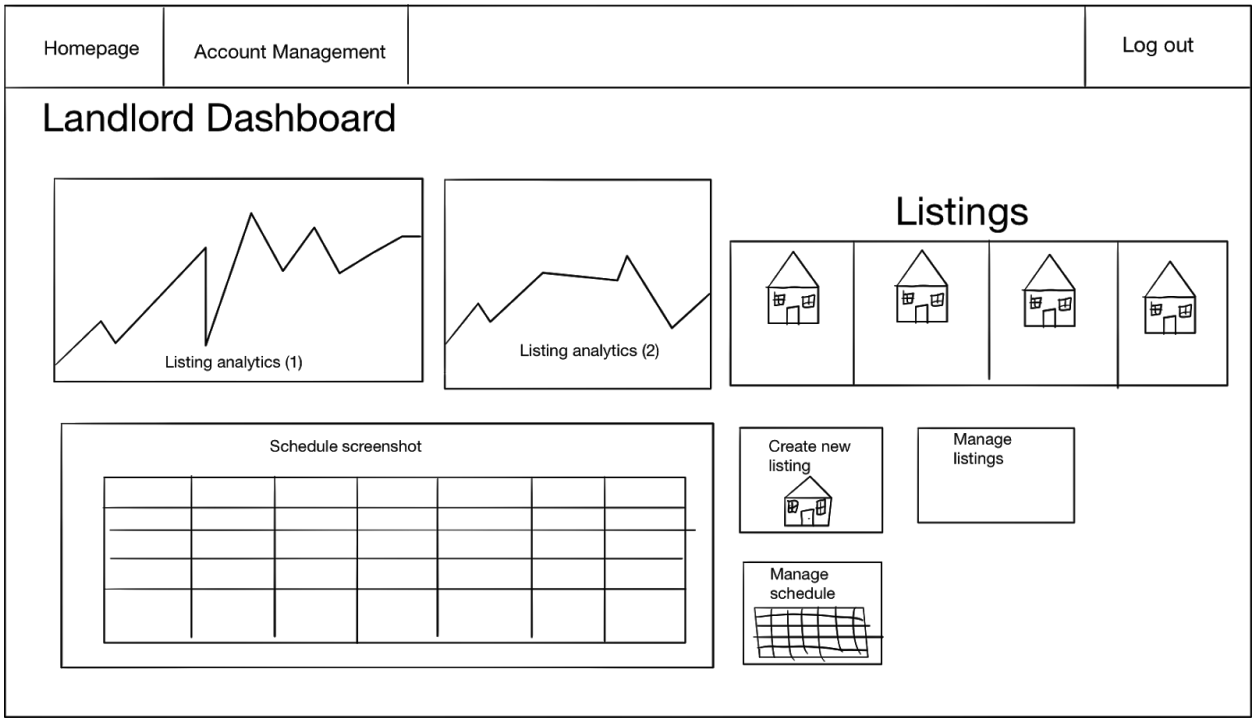
Use Case: MaintenanceRequest



User Interface Specification

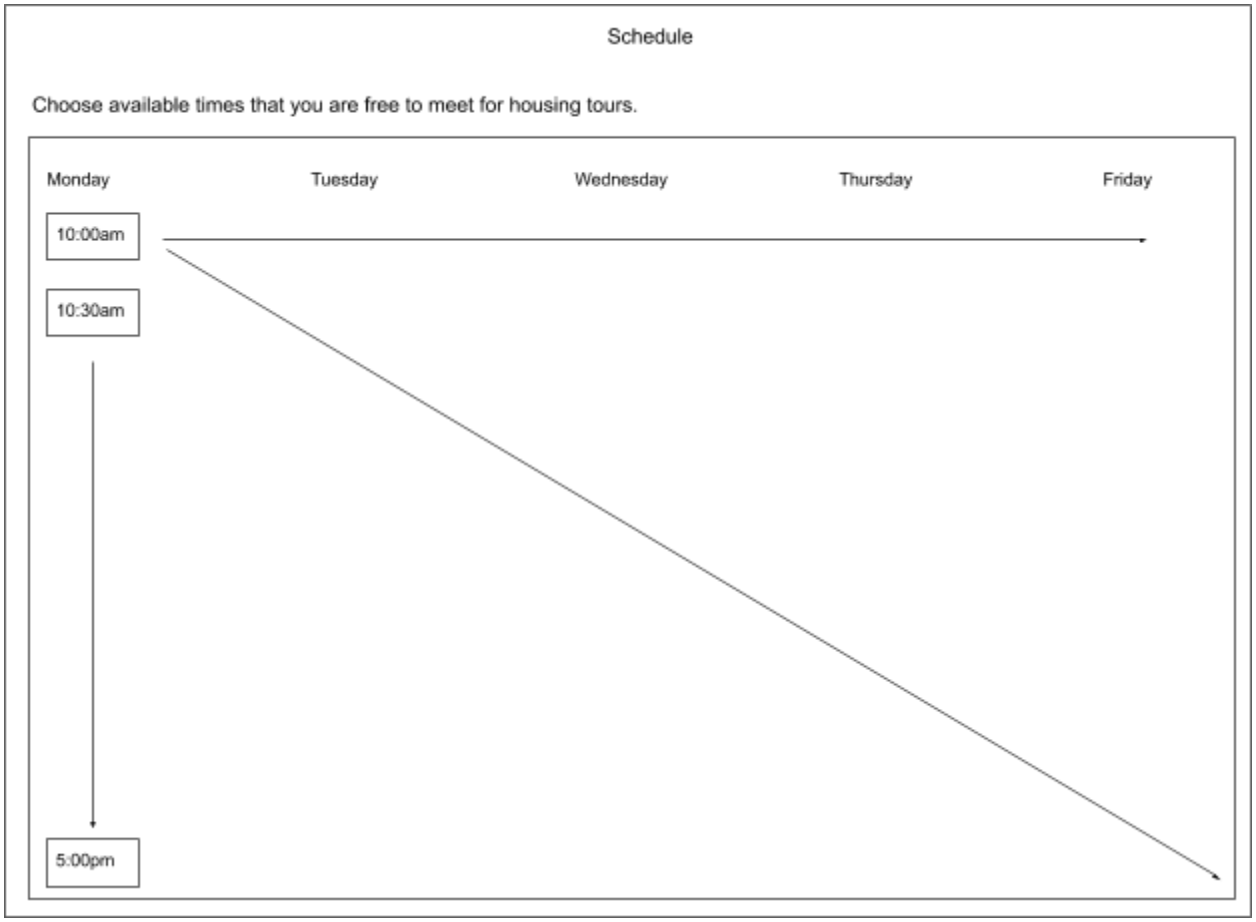
Preliminary Design & User Effort Estimation

Landlords looking for tenants:



- Landlord will need to click on the login button on the homepage.
- Once login credentials are entered and submitted, the landlord is taken to his/her dashboard where a summary of all the functions and information can be viewed.
- To manage account credentials and account information, the user must click on the account management button on the top menu bar.
    - This will take the user to the account management page where they can edit their user info or change account information.
- To manage listings information and details, the user must click on the “Manage listings” button.
    - From there, the user is taken to a page where they can select a listing they want to edit and adjust the information accordingly.
- To create a new listing, the “Create new listing” button must be selected.

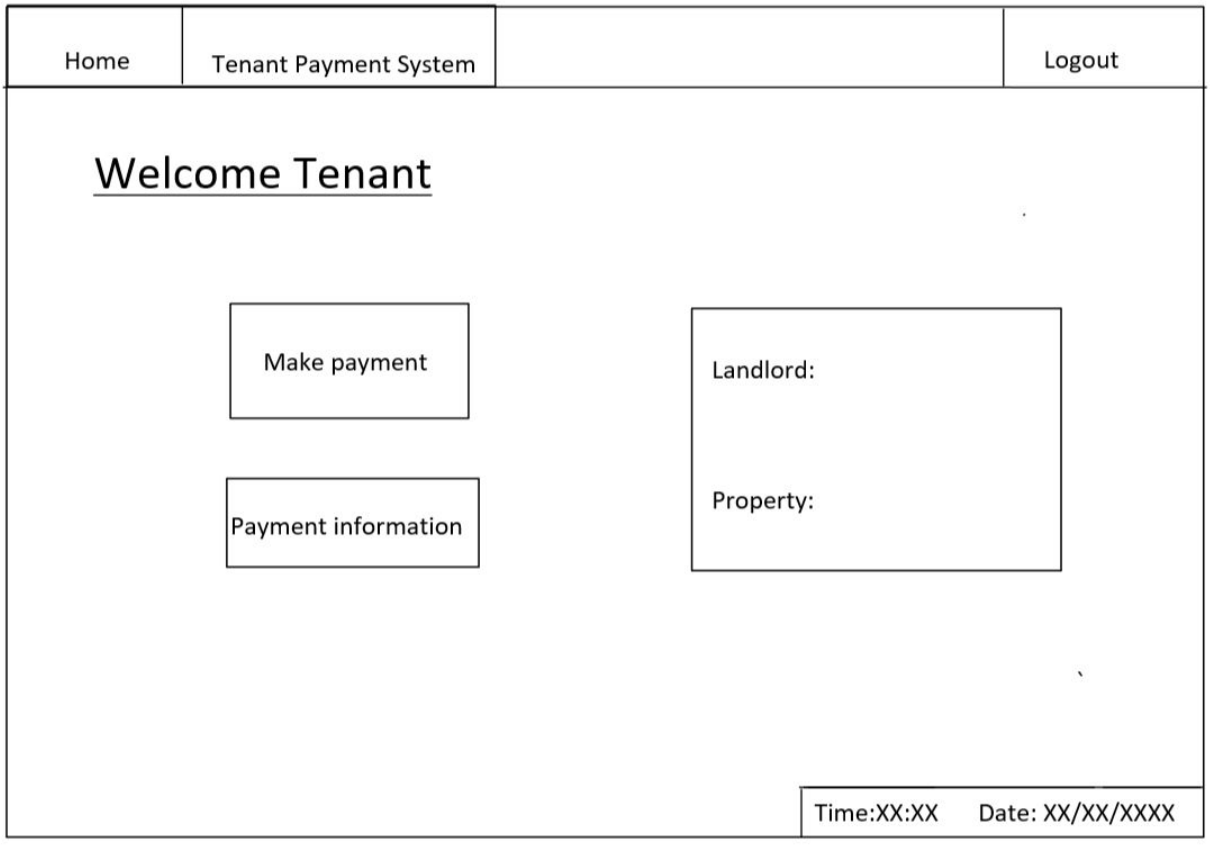
    - The user is then taken to a form where they enter all the new information for a property that they own.
    - The user then submits this form and then they are taken to the new webpage showing their newly created listing.
- **NAVIGATION:** 1 mouse click
    - Click on Login Button after entering username and password
- **DATA ENTRY:** 1 mouse click, 3 Keystrokes
    - Once in username box, enter username.
    - Click “Tab” to move to password box.
    - Enter password.



- To manage their schedule for meetings with potential tenants, the landlord must click on the “Manage schedule button.”

- The user is taken to a page where they can manage the meeting times scheduled for their properties.
- They will be able to adjust which times when they are available.
- Tenants can then see these available times and choose from them accordingly.

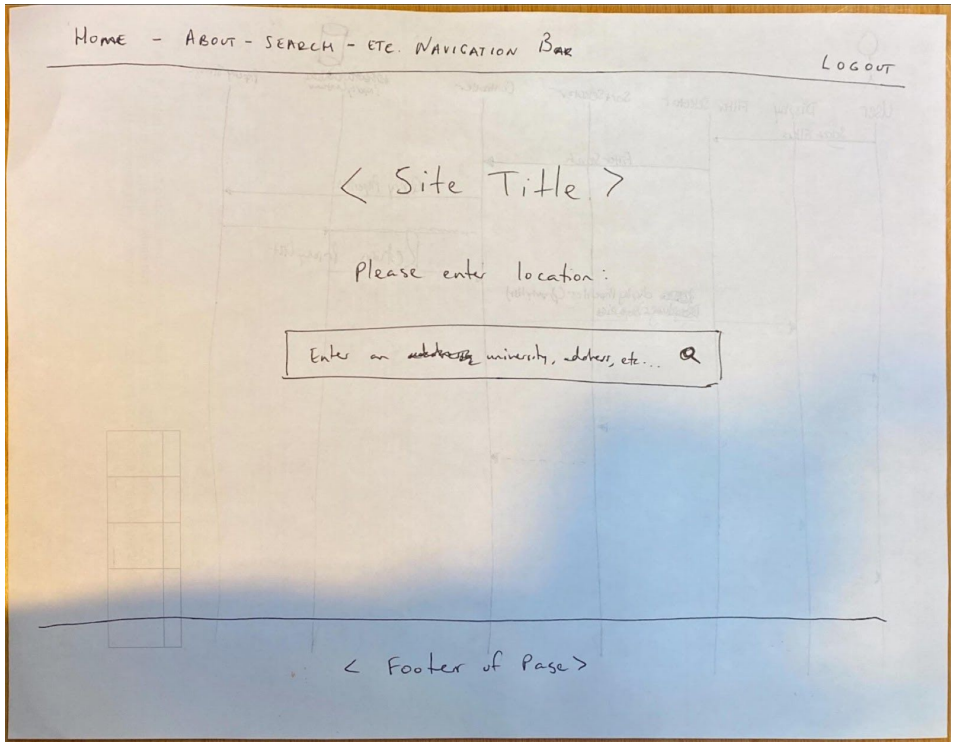
- **NAVIGATION:** 1 mouse click for tenant and landlord
  - Click on Manage Schedule button.
- **DATA ENTRY:** "N" mouse clicks for landlord, 1 mouse click for tenant
  - The number of mouse clicks for the landlord depends upon how many free time slots the landlord has.
  - There's only one mouse click for the tenant because they're choosing their agreed upon time for the housing tour.

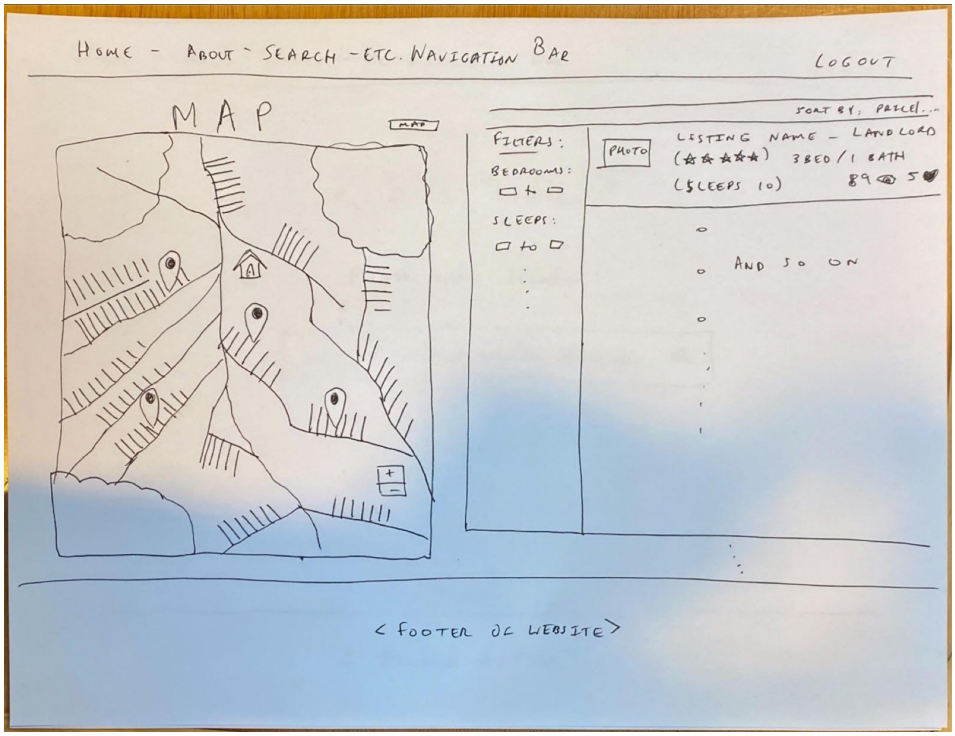


Tenants who have housing:

- When a tenant logs in and clicks on the payment system tab, they will see this screen that will have different options in the forms of buttons, which will link them to another secured page.
    - The button named *Make Payment* will transfer the user to a page where they have the option to input the amount they would want to pay to their respective landlord
    - The button named *Payment Information* will transfer the user to a page where the user can decide their preferred payment, and place any relevant information.
- The landlord and property information are listed as text on the right side, to display the corresponding information such as who is the tenant's landlord and what property they live on.
- Once the payment has been processed and reviewed and successfully received by the landlord, the system will notify the tenant that the payment was successfully sent.
- **Navigation:** total 2 mouse clicks
    - Click Payment Information
    - Click Make Payment
- **Data Entry:** 1 mouse click, 1 keystroke
    - Click Payment Information
    - Enter Payment Information

Prospective Tenants:

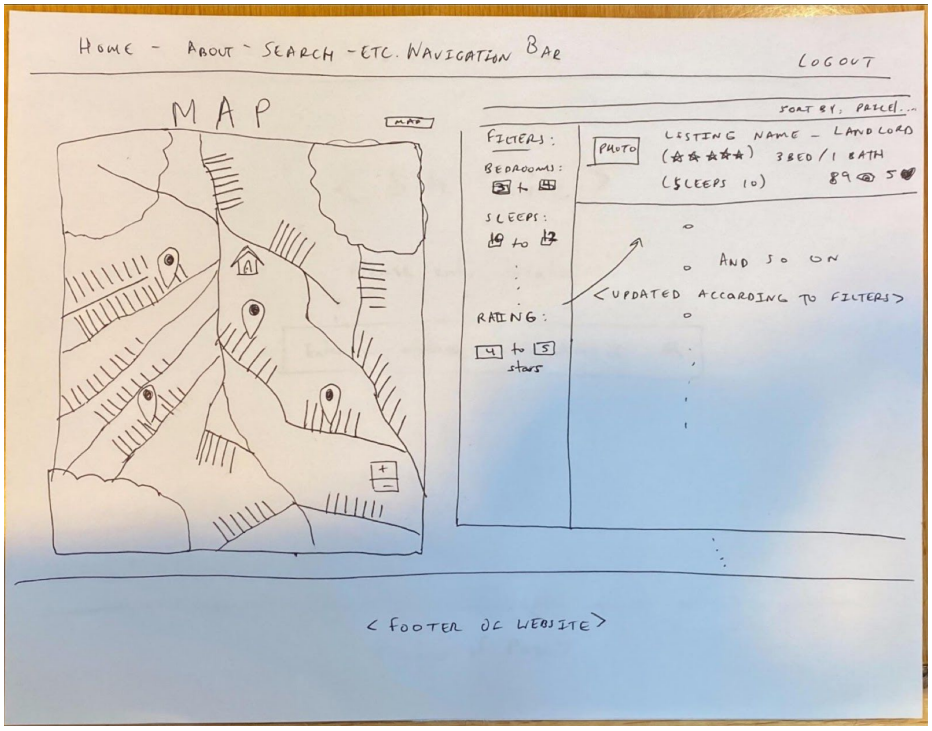




- Upon clicking the search link in the navigation bar, the user will see a page resembling this. There will be filters on the left side, and the filtered listings will appear on the right.

There is also a map view on the left side of the screen, where the users can see location markers for each of the properties.

- The user can sort by views, rating, distance from campus, etc. using the dropdown in the top right of the box in the middle.
- An example of a filled out filter area is as shown below:

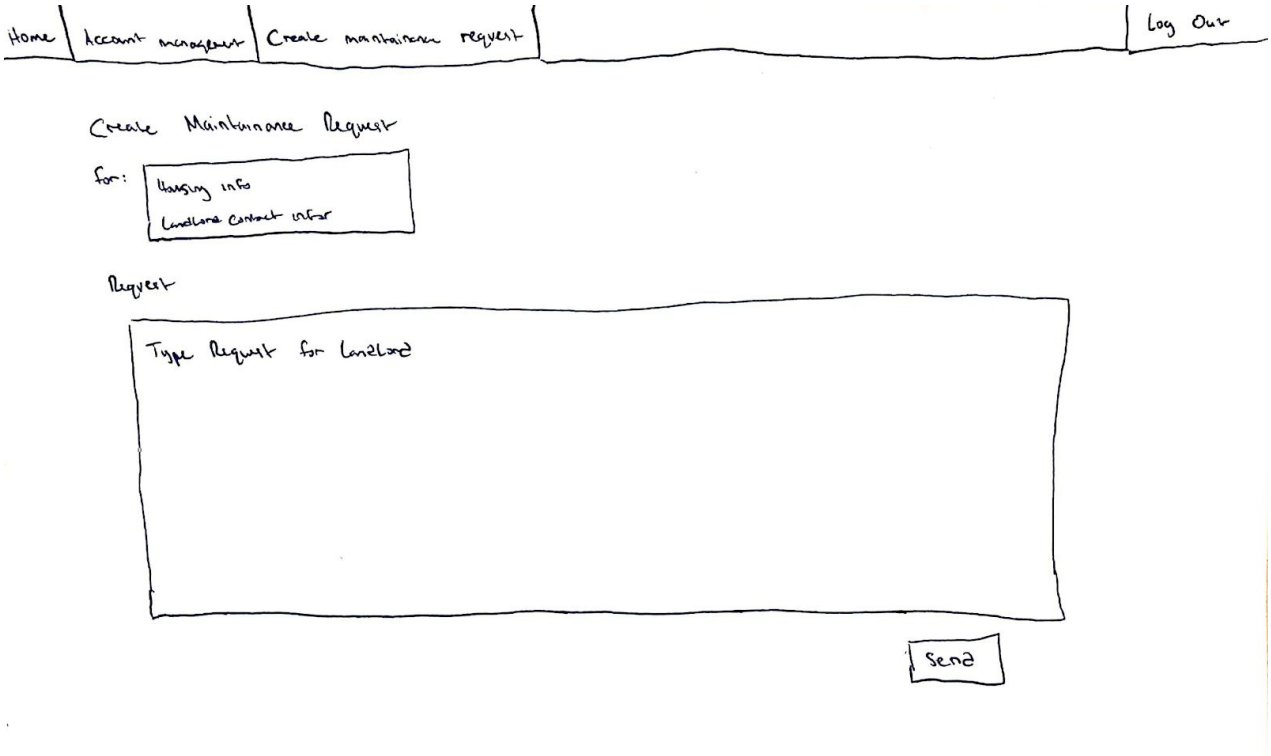


- Once the user inputs information, the listings are filtered, and are shown. The only time this would not be the case is if there are no properties that fit the user's requirements. In this case, the website would show a notification to the user asking them to widen their search criteria.
- User Scenarios:
    - Prospective Tenant looking for a house without selecting any filters:
        - This would require the most manual time, as there is no filtering, so the user would have to manually search through each listing. Depending on what they're looking for, and how lucky they get, it could take anywhere from minutes to hours. The number of mouse clicks/keyboard strokes would be low, but the amount of scrolling would be high. It is a

counter-intuitive approach as the website can handle most of the manual searching through filters.

    - Prospective Tenant looking for a house while selecting filters:
        - This would require as many mouse clicks and keyboard strokes as the user would like to declare filters. The more declared filters, the more refined results and the less amount of time used to search through properties. There is a tradeoff point between selecting filters vs searching through properties, as selecting filters would front-load effort and user interface time, and less filters would mean that the user would spend more time manually searching, and depending on the amount of properties, the time spent will increase.
- No matter which scenario, to get to this page, the user has to login, which is 4 clicks, and click on search. Then they are at the search page.
- **Navigation:**
    - First Scenario:
        - Little to no clicks/keystrokes
        - A lot of scrolling and reading
    - Second Scenario:
        - Less time scrolling and reading due to data entry overhead.
- **Data Entry:**
    - First Scenario:
        - No data entry
    - Second Scenario:
        - Depending on the amount of filters selected, there can be a lot of data entry. No more than 20 clicks and 40 or so keystrokes.

Maintenance Request:



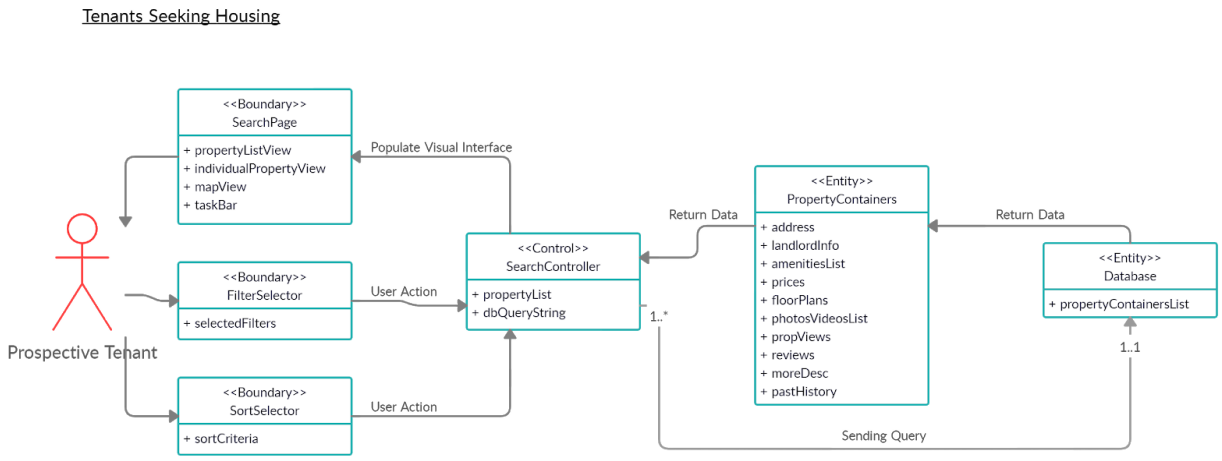
- Once the User's Information is verified, they are able to follow a link that leads them to the maintenance request page.
- Once here, Tenant's are able to create a request for the landlord to let them know a repair is needed, and the property status is changed
- On the flip side, the Landlord will receive this message and a notification, and they will be able to read the request upon log in.
- Upon fixing the issue, the Landlord will let the system know the request has been handled and the property status will once again change to normal.

- **Navigation**:
    - For Tenant: 2 Clicks
        - 1 click to log in
        - 1 click to access Maintenance request page
    - For Landlord: 2 Clicks
        - 1 click to log in
        - 1 click to access messages
- **Data Entry**
    - For Tenant:
        - Typing a descriptive passage of the issue that needs to be handled by the landlord
    - For Landlord:
        - Marking request as completed

Domain Analysis

UC-1: SearchProperties

Domain Model



Concept Definitions

Responsibility Description	Type	Concept Name
Rs1: Requesting Property Container from Property Storage	D	SearchController
Rs2: Sends list of PropertyContainers to display	D	SearchController
Rs3: Display property information	D	SearchPage
Rs4: Selects filters to search by	D	FilterSelector
Rs5: Contains which filters were selected to search by	K	FilterCollector
Rs6: Selects how the results will be sorted	D	SortSelector
Rs7: Contains a picture and information	K	PropertyContainer
Rs8: Returns a list of Property Containers	D	PropertyStorage

Association Definitions

Concept Pair	Association Description	Association name
SearchController->SearchPage	Populates Display (User Interface) with results from PropertyContainer query.	Populate Visual Interface
FilterSelector -> SearchController	Prospective Tenant selects various filters from the website which get formed into a database request	User Action
SortSelector -> SearchController	Prospective Tenant selects one sorting order (price low to high, # beds, etc.) from the website which get formed into a database request	User Action
SearchController <- PropertyContainer	List of PropertyContainer objects are returned to the Controller following the Property Container Query. Controller then populates the UI wih this data.	Return Data
PropertyContainer <- Database	PropertyStorage (Database) returns the results of the query as a list of PropertyContainer objects.	Return Data
SearchController -> Database	Controller sends query to the database for retrieval of PropertyContainer objects. Query is formed based on selected Filters or Sort option.	Sending Query

Attribute Definitions

Concept	Attributes	Attribute Description
SearchPage	propertyListView	UI component that shows the user an updated list of properties based on their search filters/sort preference.
	individualPropertyView	UI component that shows pictures and detailed information regarding a selected property
	mapView	UI component that shows location pins of properties within the user's search filters
	taskBar	UI component containing login and registration buttons, list of features, links to Contact, Help, FAQ,

		and About Us page, so that I can access all the information I need to
FilterSelector	selectedFilters	UI interactor component to select the # of bedrooms, bathrooms, distance from campus, etc
SortSelector	sortCriteria	UI interactor component to select sort by distance from campus, price, or date listed
SearchController	propertyList	Has list of all the property containers returned by property storage to be sent to the display
	dbQueryString	Formatted query string including selected filters and/or search criteria
PropertyContainer	address	Street Name, Town, Zip Code, etc.
	landlordInfo	Information about person who owns the house
	amenitiesList	AC/Heating, Parking Spots, etc.
	prices	Rent, utilities, security deposit
	floorPlans	Layout of house and number of bedrooms, bathrooms, etc.
	photosVideosList	What the house looks like on the inside
	propViews	Number of people that have viewed the property listing online
	reviews	Past tenants feelings about house and landlord
	moreDesc	Anything extra the landlord wants to add
	pastHistory	Previous dates for renovation, when the house was last rented, etc.
Database	propertyContainersList	PropertyContainer objects storing information regarding each listing, for easy retrieval to controller

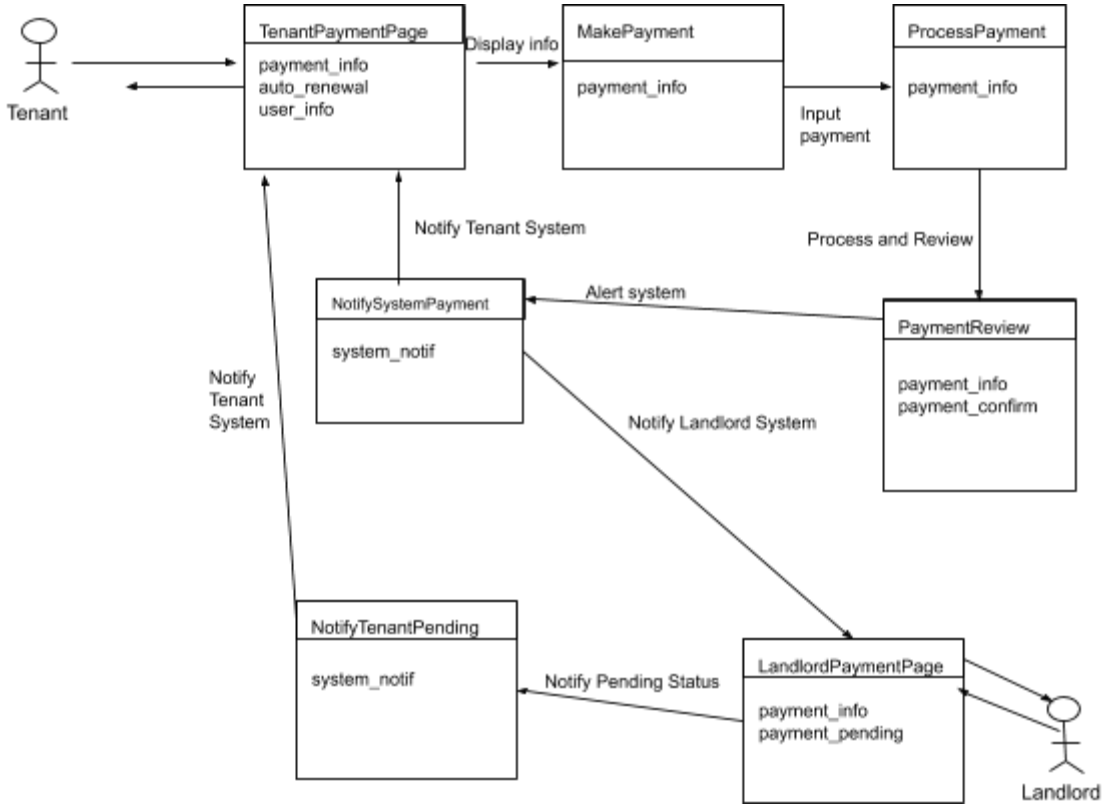
System Operation Contract

Operation	SearchProperties
Use Case	UC-1

Preconditions:	• The user has opened the search page • The user has provided any filters they wish to apply
Postconditions:	• The user can see all the properties for their filters • The user can sort the properties by multiple factors(price,etc.)

UC-6: Payment

Domain Model



Concept Definitions

Responsibility Description	Type	Concept Name
Rs1. Tenant has the capability of making automated secure payments through the website by entering in their payment information.	D	MakePayment

Rs2. The system should process the payments after it is sent out by the user.	D	ProcessPayment
Rs3. When the payment has been processed, the system will give an opportunity to review.	K	PaymentReview
Rs4. System will be given a signal that payment has been processed.	D	NotifySystemPayment
Rs5. System will notify the tenant that payment is due. The details are given by the landlord, and approved by the tenant.	D	NotifyTenantPending
Rs6. Displays all necessary information in order for the Tenant to enter their payment information in and make payments.	K	TenantPaymentPage
Rs7. Displays all necessary information in order for the Landlord to view any payments made by their tenants.	K	LandlordPaymentPage
Rs8. Stores payment info for future automated payments.	K	Database

Association Definitions

Concept Pair	Association Description	Association Name
LandlordPage ⇔ PaymentHandler	Monthly rent paid shows up on the landlord's profile.	Transaction
PaymentPage ⇔ MakePayment	Payment page displays necessary information for Tenant to enter information and make payments.	display info
MakePayment ⇔ ProcessPayment	Tenant enters their payment information.	input payment
ProcessPayment⇔ PaymentReview	Once the payment has been processed, the system will provide an opportunity for the user to review the payment before confirming once more.	process and review

PaymentReview ⇔ NotifySystemPayment	Once the payment has been reviewed, it is then sent and the system is notified.	alert system
NotifyTenantPending⇔ TenantPaymentPage	Sends signal to notify Tenant that payment is due, and page will notify the Tenant.	send signal
NotifyLandlordPayment ⇔ LandlordPaymentPage	Sends signal to notify Landlord that payment has been successfully sent by the tenant, and displays notification on the Landlord Page.	send signal
NotifySystemPayment ⇔ LandlordPaymentPage	Notifies landlord that the payment has been delivered.	notify landlord system
LandlordPaymentPage ⇔ NotifyTenantPending	Landlord page sends a pending payment status to tenant.	notify pending status
NotifyTenantPending ⇔ TenantPaymentPage	Pending payment status is sent to the tenant's payment page.	notify tenant system
NotifySystemPayment ⇔ TenantPaymentPage	When payment has been processed, the system will alert the tenant on the status of the payment.	notify tenant system

Attribute Definitions

Concept	Attributes	Attribute Description
MakePayment	Payment Information	User enters in their own payment information used to make payments
ProcessPayment	Payment Information	Used for the tenant's benefit to easily process payments
PaymentReview	Payment Information	Allows a second review of the information placed by the tenant

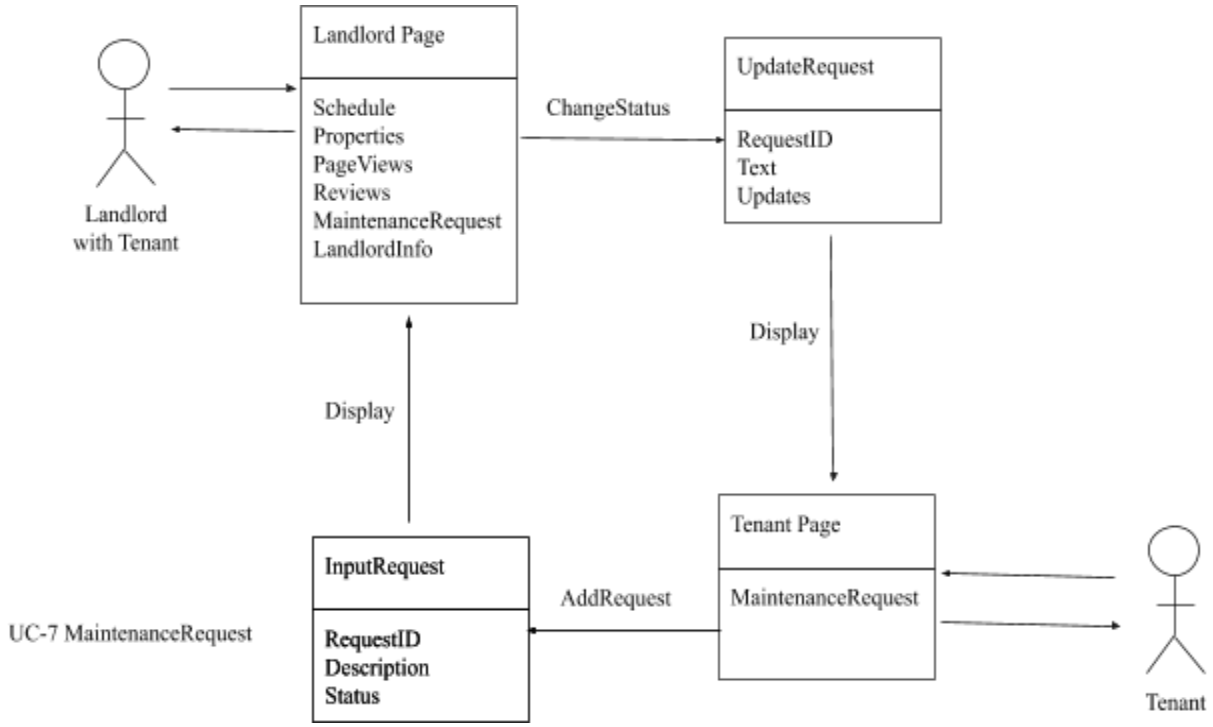
PaymentPage	Payment Information	Allows a second review of the information placed by the tenant.
	Confirm Payment	Option for tenant to select to confirm payment before it is sent out.
NotifyTenantPending	System Notification	Notification that tenant receives that payment still needs to be sent.
TenantPaymentPage	Payment Information	System will display the amount that is due.
	Automatic Renewal	Option for user to automatically renew their payment next time the payment is due.
	User Info	Allows user to enter in their payment information and store it in the system.
LandlordPaymentPage	Payment Information	System will display the amount that is due.
	Payment Pending	Shows how much the tenant owes towards the landlord.

System Operation Contracts

Operation	Making and Receiving Payments
Use Case Number	UC-6
Preconditions	● Tenant or landlord is already logged into the system. ● Tenant is currently renting a property from a landlord.
Postconditions	● The landlord successfully receives the payment that was sent by the tenant.

UC-7: MaintenanceRequest

Domain Model



Concept Definitions

Responsibility Description	Type	Concept Name
Rs1. Home page for landlord's important tools and info	K	LandlordPage
Rs2. To be able to change the status of a tenant's maintenance request	D	UpdateRequest
Rs3. Home page for tenant with property	K	TenantPage
Rs4. To be able to put in a maintenance request.	D	InputRequest

Association Definitions

Concept Pair	Association Description	Association Name
TenantPage ⇔ InputRequest	The tenant's maintenance request shows up on the	AddRequest

	landlord's profile for the landlord to update.	
LandlordPage ⇔ UpdateRequest	The tenant's maintenance request shows up on the landlord's profile for the landlord to update.	ChangeStatus

Attribute Definitions

Concept	Attributes	Attribute Description
LandlordPage	Properties	List of landlord's properties
	MaintenanceRequests	Current tenant's maintenance requests for the property
	LandlordInfo	Landlord's name and contact information
TenantPage	MaintenanceRequest	Tenant can request for maintenance
InputRequest	RequestID	Identifies the request
	Description	Tenant's explanation of what is wrong with the property
	Status	Shows current status of the request
UpdateRequest	RequestID	Identifies the request
	Text	Description of maintenance request
	Updates	Updated status of the maintenance request.

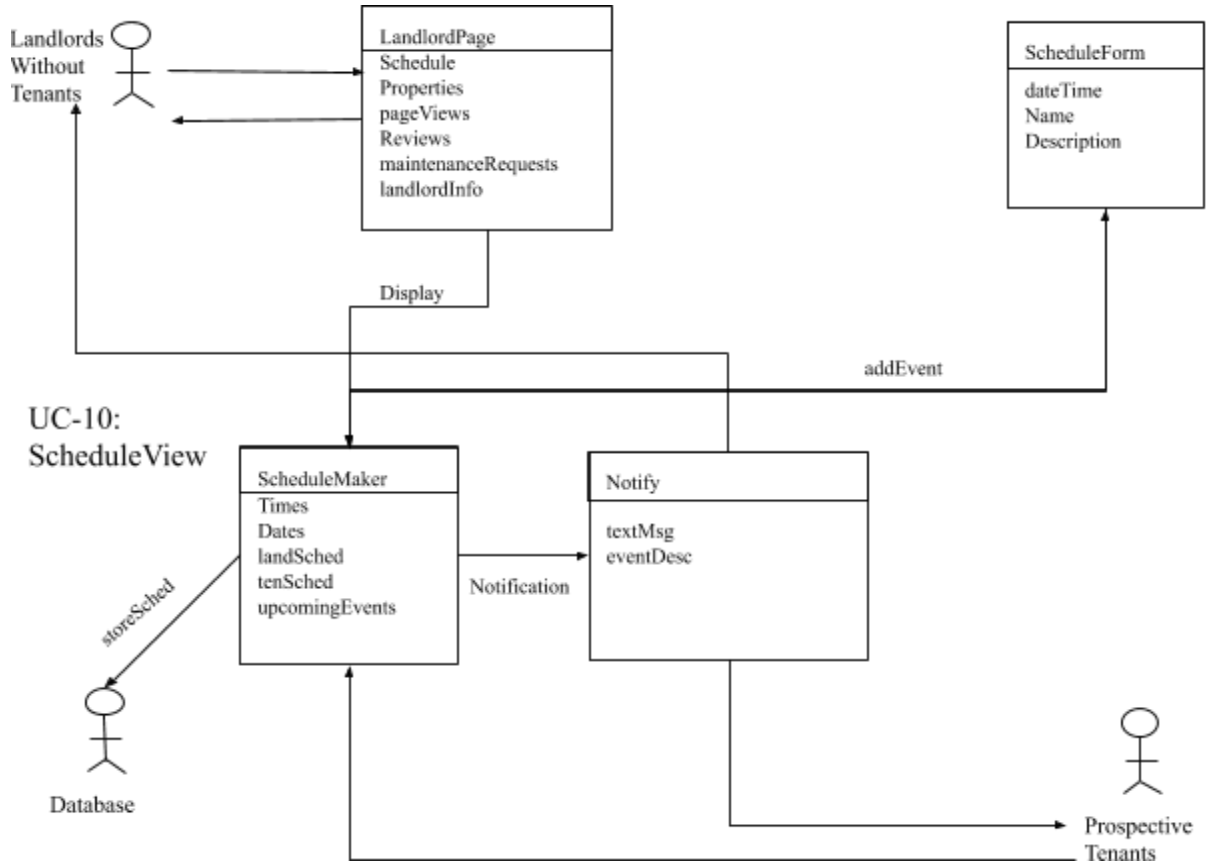
System Operation Contracts

Operation	Handling Maintenance Requests
Use Case Number	UC-7

Preconditions	● Landlord currently has tenants for the property. ● Property is in need of some maintenance. ● User (tenant and landlord) is logged in.
Postconditions	● Maintenance requests are physically dealt with. ● The status of the maintenance request is updated on the website.

UC-10: ScheduleView

Domain Model



Concept Definitions

Responsibility Description	Type	Concept Name
Rs1. Screen with times for landlord to pick times for meetings	K	ScheduleMaker
Rs2. Form to put in information about meeting	K	ScheduleForm
Rs3. System will notify the landlord and tenant when they have an upcoming meeting.	D	Notify
Rs4. Schedule is stored and saved.	K	Database

Association Definitions

Concept Pair	Association Description	Association Name
ScheduleMaker⇔ ScheduleForm	When the user is trying to schedule a meeting, he will fill out a form for the description of the meeting.	AddEvent
ScheduleMaker⇔ Notify	When a landlord creates an event and the date or time for that event is coming up, they will receive a notification.	Notification
LandlordPage⇔ ScheduleMaker	Landlord's schedule is shown on profile.	Display
ScheduleMaker ⇔ Database	Once user makes his or her schedule, it is stored in the database	StoreSched

Attribute Definitions

Concept	Attributes	Attribute Description
LandlordPage	Schedule	Landlord's tool to plan and see upcoming meetings
	Properties	List of landlord's properties
	pageViews	The number of people that have viewed their properties
	Reviews	Property and landlord reviews left by past tenants
	maintenanceRequests	Current tenant's maintenance requests for the property
	landlordInfo	Landlord's name and contact information
	Times	Time slots (buttons) for available times on each day
	Dates	Calendar dates

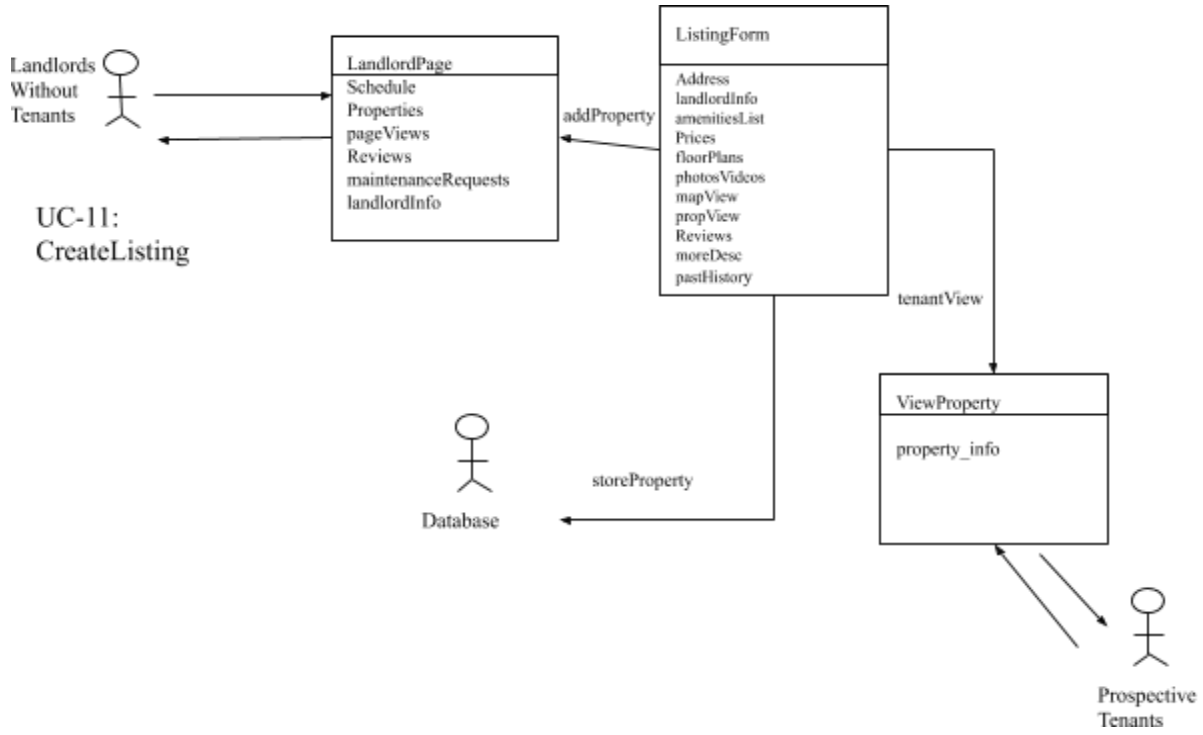
ScheduleMaker		(month/date/year)
	landSched	Landlord's side of schedule to pick available times.
	tenSched	Once landlord picks available time, tenant has to choose a final time from the selections.
	upcomingEvents	List of all scheduled events and dates/times for them
ScheduleForm	dateTime	Calendar Date plus time
	Name	What the meeting is for
	Description	Address of the property and any additional information
Notify	textMsg	Means by which message is sent and message itself
	eventDesc	All the details of the meeting

System Operation Contract

Operation	Viewing and Making Schedule
Use Case Number	UC-10
Preconditions	● User is logged in ● Listing page is created ● Tenant is interested in property ● Tenant actually schedules meeting once landlord puts in times.
Postconditions	● Landlords and tenants receive notifications when the appointment draws near.

UC-11: CreateListing

Domain Model



Concept Definitions

Responsibility Description	Type	Concept Name
Rs1. Form with important info to fill in by the landlord when creating a new property.	K	ListingForm
Rs2. List of properties for each landlord with analytics and reviews	K	LandlordPage
Rs3. Listing is stored and saved.	K	Database
Rs4. Displays property information when tenant views listing.	D	ViewProperty

Association Definitions

Concept Pair	Association Description	Association Name
LandlordPage⇔ ListingForm	After landlord fills out his	AddProperty

	page, property is shown on landlord's home page.	
ListingForm⇔ ViewProperty	After listing form is submitted, it can be viewed and searched for by tenant.	TenantView
ListingForm⇔ Database	After listing form is submitted, it is stored in database.	storeProperty

Attribute Definitions

Concept	Attributes	Attribute Description
ListingForm	Address	Street Name, Town, Zip Code, etc.
	landlordInfo	Information about person who owns the house
	amenitiesList	AC/Heating, Parking Spots, etc.
	prices	Rent, utilities, security deposit
	floorPlans	Layout of house and number of bedrooms, bathrooms, etc.
	photosVideos	What the house looks like on the inside
	mapView	Google Maps view of house
	propViews	Number of people that have viewed the property listing online
	reviews	Past tenants feelings about house and landlord
	moreDesc	Anything extra the landlord wants to add
	pastHistory	Previous dates for renovation,

		when the house was last rented, etc.
ViewProperty	propertyInfo	All information about the property

System Operation Contract

Operation	Creating and Adding a Listing
Use Case Number	UC-11
Preconditions	● Landlord has logged in. ● Landlord has legitimate property to put on the website
Postconditions	● Listing is added to the landlord's property list on his home page. ● Prospective tenants can view this property and its descriptions from the website.

Traceability Matrix

Use Cases	Point Weight	ScheduleMaker	ScheduleForm	Notify	Database	ListingForm	LandlordPage	ViewProperty	UpdateRequest	TenantPage	InputRequest	MakePayment	ProcessPayment	PaymentReview	NotifySystemPayment	NotifyTenantPending	TenantPaymentPage	LandlordPaymentPage	SearchPage	SearchController	FilterSelector	SortSelector	PropertyContainer
UC-1	26				x														x	x	x	x	x
UC-6	19				x							x	x	x	x	x	x	x					
UC-7	18				x		x		x	x	x												
UC-10	17	x	x	x	x		x																
UC-11	30				x	x	x	x															

Mathematical Model

The website will not incorporate any mathematical models.

Project Size Estimation

Actor classification

Use Case(s)	Actor	Description	Complexity	Weigh t
UC-1, UC-10, UC-11	Prospective Tenant	User interacting with system through GUI	Complex	3
UC-1	Website	Interacting through defined API as front-end	Simple	1
UC-1	Controller	System that interacts through protocol and manages other systems	Average	2
UC-1, UC-10, UC-11	Database	Another system interacting through protocol	Average	2
UC-6, UC-7	Tenant	User interacting with system through GUI	Complex	3
UC-6, UC-7	Landlord with tenants	User interacting with system through GUI	Complex	3
UC-10, UC-11	Landlord Seeking Tenants	User interacting with system through GUI	Complex	3
UC-6	Payment process system	System interacts through protocol	Average	2

Unadjusted Actor Weight = (1 x Simple) + (3 x Average) + (4 x Complex) = (1 x 1) + (3 x 2) + (4 x 3) = 1+6+12 = 19.

UAW = 19

Use Case Classification

Use Case	Description	Category	Weigh t
UC-1 SearchProperties	Moderate interface. 3 participating actors.3	Average	10

	participating actors. 3 steps for success scenario. Domain Model includes 6 concepts.		
UC-6 Payment	Complex interface. 2 participating actors. 5 steps for success scenario. Domain Model includes 7 concepts.	Average	10
UC-7 MaintenanceRequest	Moderate interface. 1 participating actor. 3 steps for success scenario. Domain model includes 4 concepts	Simple	5
UC-10 ScheduleView	Moderate interface. 2 participating actors. 4 steps for success scenario. Domain model includes 4 concepts	Average	10
UC-11 CreateListing	Simple interface. 2 participating actors. 3 steps for success scenario. Domain model includes 3 concepts.	Simple	5

Unadjusted Use Case Weight = 2xSimple + 3xAverage = 2x5 + 3x10 = 10+30 = 40.

UUCW = 40

Unadjusted Use Case Points = UAW + UUCW = 19+40 = 59.

UUCP = 59

Technical Complex Factors

Technical Factor	Description	Weigh t	Perceived Complexity	Calculated Factor (Weight x Perceived Complexity)
T1	Distributed between front-end on one machine, controller on another, and database on another	2	3	2x3 = 6
T2	Response time important for SearchingProperties (UC-1)	1	3	1x3 = 3
T3	End user expects efficiency, but no exceptional demands	1	3	1x3
T4	Internal processing is relatively simple	1	1	1x1 = 1

T5	Reusability helpful within system for cutting down on redundancies	1	1	1x1= 0
T6	No installation	0.5	0	0.5x0 = 0
T7	Ease of use is very important	0.5	5	0.5x5 = 2.5
T8	Portability concerns regarding scalability and easy migration	2	2	2x2 = 4
T9	Important to be easy to change as more features and filters will be added based on user requests	1	4	1x4 = 4
T10	Concurrent use is essential so that different tenants, prospective tenants, and landlords can all be on website at once without affecting each other	1	5	1x5 = 5
T11	Security on payments and maintenance requests is essential (UC-6, UC-7)	1	4	1x4 = 4
T12	No direct access for third parties	1	0	1x0 = 0
T13	No unique training needs	1	0	1x0 = 0
			Technical Factor Total =	32.5

Technical Complexity Factor = C1 + C2*TFT = .6+ .01(32.5) = .6+.325

TCF = .925

Here, we will assume the Environmental Complexity Factor is 1.

Thus, the Use Case Points = UUCP x TCF x ECF

UUCP = 59

TCF = .925

ECF = 1

UCP = 59x.925x1

USE CASE POINTS = 54.575

Plan of Work

We have decided to split this project by assigning different components of the application to different subgroups, rather than splitting the project by technology used (ex. There won't be one group responsible for the entire backend/frontend etc.). With this approach, different functionalities of the app can still be fully completed even if one particular subgroup is unable to deliver. There are four main 'user statuses' present in this project, namely: 'Landlord seeking tenants', 'Landlords that have tenants', 'Prospective Tenants', and 'Tenant that have Housing'.

We have started to complete the SQL database setup and normalization, as this will be required by all components of the application. In addition, we will work on general standardization of the API, so that there are no duplications of effort across the product. This will be what is part of our central infrastructure, and then the work on the separate "stacks" described above can be commenced.

As far as our plan of work, each "stack" or "user status" will be responsible for creating multiple layers within said group. There will be a front-end component, which consists of user experience design, as well as client-side scripting, along with HTML/CSS to put everything together. Additionally, there will be a middleware component, which has multiple layers. Each section will have to be able to communicate with other "stacks", so as to avoid duplicating work, and to keep a clean codebase while minimizing redundancy. There will be an internal API, which can be accessed by other parts of the application, as well as an internal methods, which include the routing for the specific actions specific to that "user status". The APIs contain all of the business logic for the different "user statuses", and there will be API-to-API communication to minimize work, and to maintain efficiency. This middleware will be implemented in Java, using the Spring web framework to handle routing and database connections.

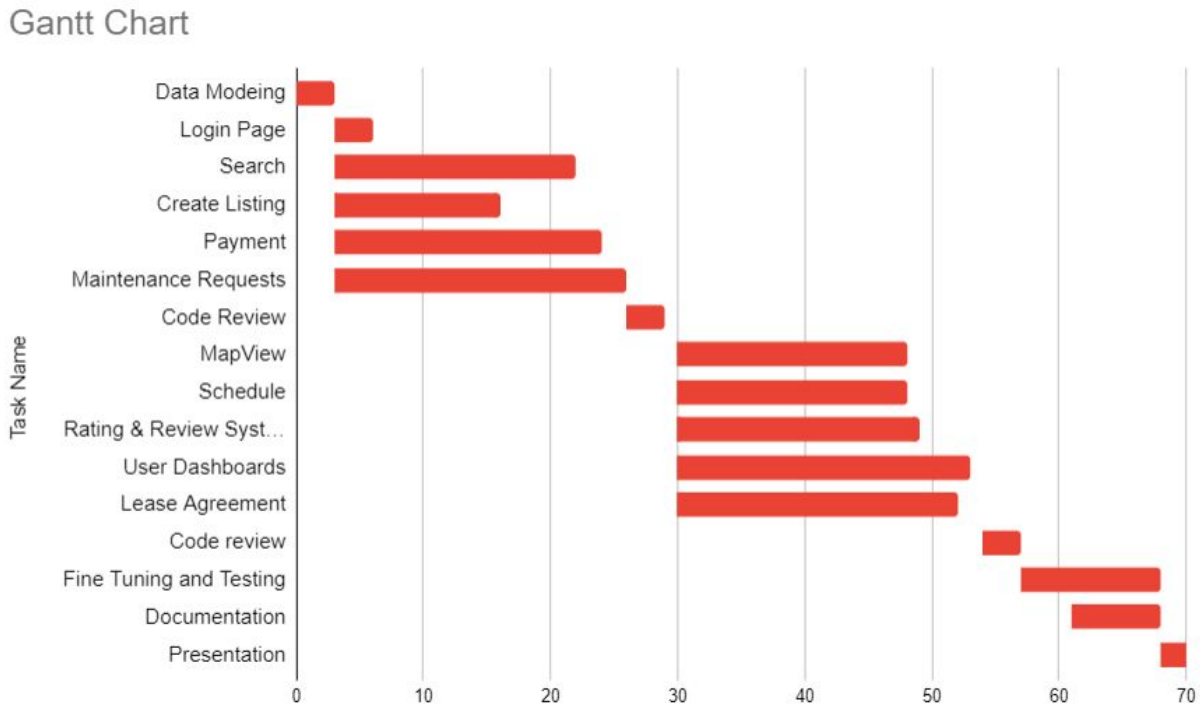
These individual pieces will connect to our central database with a standardized schema that supports all parts of the application. We will design this central infrastructure as a whole team in

order to eliminate any confusion, and so that everyone has a say in data storage and retrieval, as that will be the backbone of our application.

The following is our Gantt Chart. The sub-teams are organized as follows:

Acronym	Team	Members
PT	Prospective Tenants	Michael Giannella Rishab Ravikumar John Yager
CT	Tenants with Housing	Srinivasniranjan Nukala Ansh Gambhir
LWT	Landlords with Tenants	Sneh Shah Ketu Patel Sahil Patel
LNT	Landlords Seeking Tenants	Kyle Tran Rishi Shah

	Task	Start	End	Start on Day	End on Day	Duration	Team
Up to First Demo							
	Data Modeling	2/23/2020	2/26/2020	0	3	3	Everyone
	Login Page	2/26/2020	2/29/2020	3	6	3	Everyone
	Search	2/26/2020	3/16/2020	3	22	19	PT
	Create Listing	2/26/2020	3/10/2020	3	16	13	LNT
	Payment	2/26/2020	3/18/2020	3	24	21	CT
	Maintenance Requests	2/26/2020	3/20/2020	3	26	23	LWT
	Code Review	3/20/2020	3/23/2020	26	29	3	Everyone
Up to Second Demo							
	MapView	3/24/2020	4/11/2020	30	48	18	PT
	Schedule	3/24/2020	4/11/2020	30	48	18	LNT
	Rating & Review System	3/24/2020	4/12/2020	30	49	19	PT
	User Dashboards	3/24/2020	4/16/2020	30	53	23	Everyone
	Lease Agreement	3/24/2020	4/15/2020	30	52	22	LWT
	Code review	4/17/2020	4/20/2020	54	57	3	Everyone
Up to Final Submission							
	Fine Tuning and Testing	4/20/2020	5/1/2020	57	68	11	Everyone
	Documentation	4/24/2020	5/1/2020	61	68	7	Everyone
	Presentation	5/1/2020	5/7/2020	68	74	6	Everyone



Project Management

In order to keep ourselves productive and on top of our deadlines, there are several tools we use to help. We all use Messenger to be able to contact each other quickly and efficiently with any problems and/or questions we have for each other. Additionally, we meet at least one a week to go over any deadlines we have coming up, check in with the progress we have made, and to distribute the work so that everyone knows what they are responsible for. We have also created a folder in Google Drive that has all the relevant documents we need. This folder also helps us work together on assignments at the same time and also check in on work that others have done. As far as code organization, we plan to use GitHub for version control and source code management. We will use the built in GitHub Projects feature to track features and progress on those features in a scrum management style. The day to day programming will be done amongst the sub-teams we have created, each responsible for their own “stack” of the application.

References

Website Title: Zillow
URL: https://zillow.com

Website Title: Places4Students
URL: https://www.places4students.com/Index

Website Title: Creating Spring Boot and React Java Full Stack Application with Maven
URL: https://www.springboottutorial.com/spring-boot-react-full-stack-crud-maven-application

Website Title: Use React and Spring Boot to Build a Simple CRUD App
URL: https://developer.okta.com/blog/2018/07/19/simple-crud-react-and-spring-boot

Website Title: Software Engineering – Domain Modelling
URL:
https://computersciencesource.wordpress.com/2009/11/26/year-2-software-engineering-domain-modelling-2/

Website Title: What is Sequence Diagram?
URL:
https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/