National College of Ireland

BSc in Computing

2017/2018

Andre MacNamara

X14380181

Andre.macnamara@student.ncirl.ie

Rentable

A portal to streamline the process of securing and managing tenancies.

Technical Report

Supervisor: Dr. Paul Hayes

Declaration Cover Sheet for Project Submission

SECTION 1

Name:	**Andre MacNamara**
Student ID:	**X14380181**

Supervisor: Dr Paul Hayes

SECTION 2 Confirmation of Authorship

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: *Andre MacNamara*

Date: 8th May 2018

Executive Summary

Rentable is a web application that is developed for the rental market in Ireland, targeting both Landlords and Tenants. At the core of the application, it allows tenants to search for properties available to rent, and landlords to post properties for rent.

Tenants can increase the efficiency of their property search by creating different watchlists and adding properties to the current active list by clicking an icon on the image of the property. Tenants can then go back through various watchlists to compare properties they are interested in and refine their choices.

The search is a standard filtered search. Users can narrow down their search by specifying the desired attributed such as county, town, the minimum and max rent, bedrooms and bathrooms. The search results only show the matched properties (if any).

Currently in Ireland, there is a huge shortage in properties available to rent. The landlord will be inundated with applications. A possible solution to this is the tenant search which is available to landlords in this application. Tenants will set their property preferences, and the landlord can select one of their active adverts and do a search for tenants whose preferences match or are like the property specs. This streamlines the tenant search for landlords (and tenants), as the landlord will only get relevant tenant results.

A crucial element in this application is the ability to start a tenancy. This is essentially like a friend's system that you will see in social media sites. Only the landlord can initiate the tenancy request, and once registered the tenancy can be managed.

The management tools include a expense tracker for properties. Landlords can log an expense associated with one of their properties by entering some details such as title, cost, category, and date. Once logged, this data is used to generate various chart. Each property has its own unique chart, and then each landlord has a chart with the aggregated data from all properties. Benefits of this would be, a permanent record of expenses, without the worry of keeping track of receipts, and landlords can spot areas where they can potentially reduce expenses, and as a result make more money.

Other features in this application include a private messaging system, an expense claimer for tenants so they can recoup outlay on their rental property, a feedback system, where tenants can air grievances, and inform the landlord on how the tenancy is going.

Introduction to the project

Problem

Many landlords today in Ireland are accidental landlord because of the boom and bust the county has gone through over the last decade. Whether this is because of the inability the afford the repayments, or a couple who bought an apartment many years ago, are stuck in negative equity and can't sell.

The process of renting out a property in Ireland can be a painful experience for many who have no desire to be a landlord and is an industry many are dying to get out of, once it makes financial sense. If the process can be made easier, many people would assumedly jump on it.

On the flip side of the industry, it's currently very hard for tenants also. There's a lack of supply out there, and many people are in properties that does not suit their needs. Whether the property is too big, or too small. People need properties that better suit there needs, and the ability to find people who are offering them properties. Moving to a smaller property could reduce bills and help save money. Moving the a bigger, more suitable property could increase a family's quality of life.

At the current time in Ireland, there isn't any all in one application for landlords and tenants to advertise rental property, search for rental property, and to manage the tenancy and any issues that may arise during the term of the tenancy. Landlords may have to manually keep track of receipts and create/maintain spreadsheets to track expenses for their tax return every year. If a tenant has an issue they will have to use external applications such as email, or text which the landlord may never follow up on.

The objective of this project is to create a software product is to create an application that helps landlords find the most suitable tenant, and the tenant find the most suitable property. The aim is to improve the experience for both landlords and tenancy, when looking to start the tenancy, and managing the tenancy.

Solution

The app is going to provide several tools to improve communication between the landlord and tenant in various ways.

The first is a private messenger. This is essentially like email, but it's only available within the site. As you can't message anyone on the by typing in their name, you must search for a tenant, or landlord, and click the message button on their profile. The messenger can be used to talk about a potential tenancy and keep a written record of all communications. Message are sorted into different message sections (inbox for received message, and a sent box for sent messages).

Next is a feedback system. When a tenancy is in place, the tenant can send feedback to a landlord. This is done via a survey. The property address, tenant name, and landlord name are pre-populated, and the questions are provided with various checkbox options below. There is a comment box for additional information also. The landlord will be provided with a visual overview of all the aggregated data from all feedback reports received from their tenant. Currently landlords can only have one tenant, so the data will be from the one landlord.

Landlords have an expense tracker where they have a separate logger for each active add on the site. The property doesn't need to be involved in a tenancy, as it might not rent for a few months, but it will still be costing them money. Each expense will be used to provide a visual representation in the form of a pie chart, and on the expense home page, an aggregated chart from all property expenses will be provided. A visual representation of data gives a quick overview of where money is being spent, and landlords they can potentially spot areas where money can be saved. The text-based log allows the landlord to have a permanent record of expenses so if they happen to accidentally delete an excel sheet, or loose some receipts, they still have the figures and details miscarry for there tax return when the time comes.

Tenants can submit a request to their landlord to claim back expenses they may have had to spend on the property. The tenant will open a form enter in the details (title, description, cost, date) and submit the claim. On the landlord's end, they will have the option to either accepted the claim request or reject the claim request. The decision will then appear on the tenant's end. If there's any issues here, they can issue the message system to communicate.

Among the communication tools provided there are various search features that have been briefly touched on in the summary and will be discussed further on in this report.

Technologies

Laravel

Laravel is the backend technology being used for this project. Laravel is a PHP framework. I chose it due to my experience in the past with standard vanilla PHP. Using Laravel makes its easier to write clean maintainable code. Laravel provides some great basic features which can be built upon such a authentication.

Eloquent ORM

Eloquent is Laravel's implementation of active record. It's an object relational mapper, which eliminates the need to write complex SQL queries, and data is easily retrieved. This increases the speed of development. An example of a query with be **$Properties = Properties::all();**

This query would get all proprieties in the database and store it in a collection. The process here would be to pass the variable to a view, and loop over the collection to get all the properties. This is an example of how powerful it can be, and the reduction in code needed vs a raw SQL statement.

Blade

Blade is the templating language used in Laravel. It's essentially the template for your views. It's a compiled php files, as such allows you to write PHP code into your views where necessary. Following on from the above example, we pass the properties variable in our view from are controller like so

```
return view('/views/properties'. Compact($properties'));
```

In your view you would call the variable and display the data in it's collection like so.

```
@foreach ($properties as $property)

<h1>{{$property->address</h1>

@endforeach
```

Bootstrap 4

Bootstrap is a library used to design the user interface of website quickly. Bootstrap 4 is the main styling used in the app, as it comes with a lot of pre-created elements such as navigations, cards, and tables to hold and display data. Most of the data is display using bootstrap cards, and tables and it makes it easy to create decent looking apps. It also makes the app responsive.

SCSS

SCSS is the default styling language used with Laravel. It's very similar to CSS, and as it came as default, I decided to use it for my customer styles. SCSS is the newer version of sass. The benefits of sass are – CSS variables, the ability to use functions, the ability to join different files are compile time.

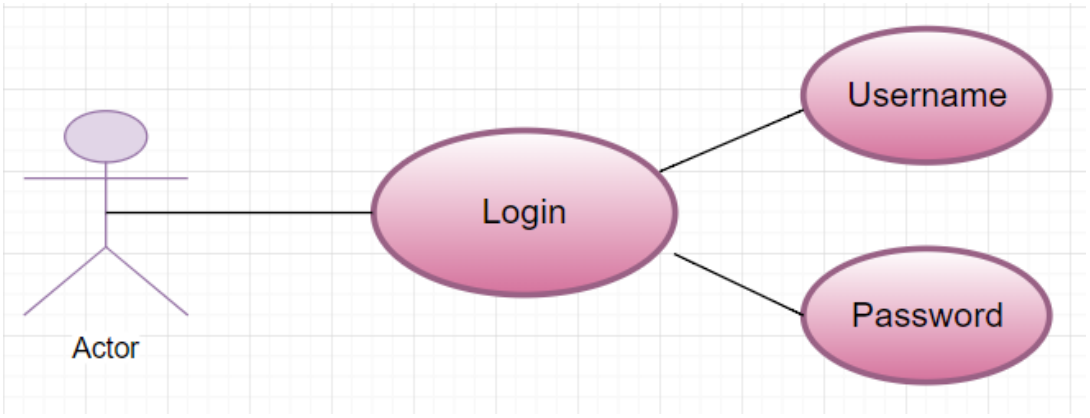
JavaScript

JavaScript is used for the typehead used in various sections in the site using the town variable. As there are over 1000 towns available to choose from in this application, it wouldn't make sense to have a dropdown menu. Users can start typing a town, and any matches will to the keywords will appear. This create a better user example. JavaScript, and in some instances, jQuery is used for animations, dropdown menus, and other user interface effects.

System

Functional-Requirements

Requirement 1 <Login>



Description

- Users can login to their accounts.

Precondition

- The user must have an existing account.

Activation

- The user navigates to the log in page.
- The user clicks a link that only logged in users can access and are redirected to login page.

Main Flow

- The login page renders the login form.
- The user enters email and password.
- The credentials are verified against the database.
- The use is logged in.

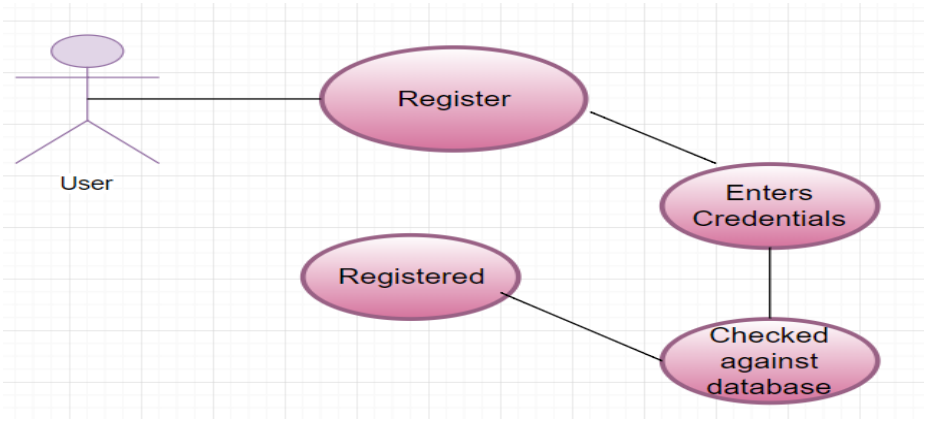
Alternative Flow

- The user enters email and password.
- The details are incorrect.
- The user is prompted to sign in again.

Post Condition

- The user is logged into the site.

Requirement 2 <Create Accounts>



Description

- User can create the account.

Precondition

- The user must not have an existing account.

Activation

- The user navigates to the registration page.

Main Flow

- The register page renders the registration form.
- The user enters name, email, user-type (Landlord or Tenant) and password.
- The credentials are verified against the database.
- The user is registered

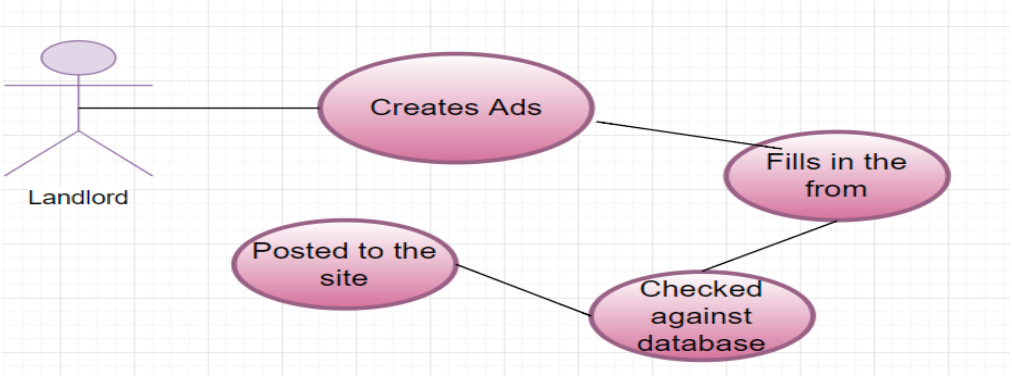
Alternative Flow

- The user already has an account.
- The user is prompted to log in to the existing account.
- The user logs in.

Post Condition

- The user is registered and logged into the site.

Requirement 3 <Post Property Advertisement>



Description

- A landlord can post a property advertisement.

Precondition

- The user be a landlord.
- The user must be signed in.

Activation

- The user navigates to the property advert create page.

Main Flow

- The create page renders the advert form.
- The user enters all the required property information.
- The data is validated.
- The advert is posted the database.
- The user is navigated to the property show page.

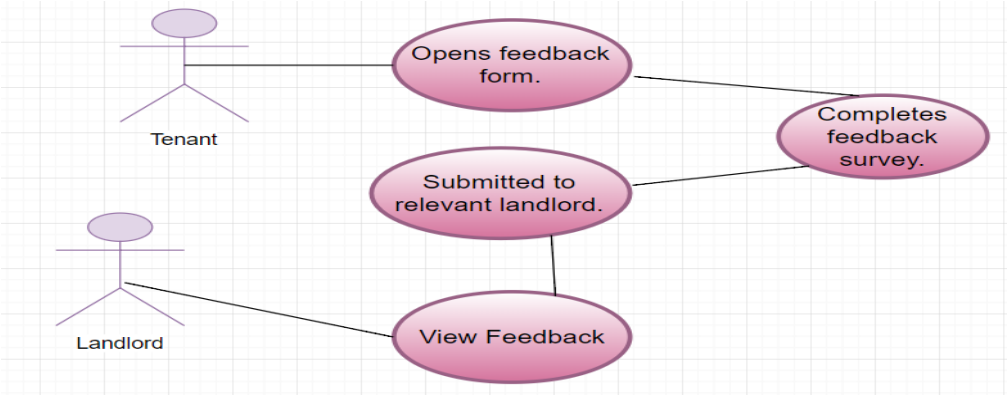
Alternative Flow

- The advert is submitted with missing data.
- The landlord is prompted to fix the issues.
- The advert is posted.

Post Condition

- The landlord has a advert posted.

Requirement 4 <Feedback>



Description

- A tenant can submit a feedback survey to landlords, for them to view.

Precondition

- The tenant must be in a tenancy with the landlord.

Activation

- The tenant navigates to the feedback centre and clicks the submit feedback button.

Main Flow

- The tenant arrives on the feedback page.
- The tenant answers the questions on the survey.
- The tenant submits the form when complete.
- The tenant can review the feedback from feedback centre.

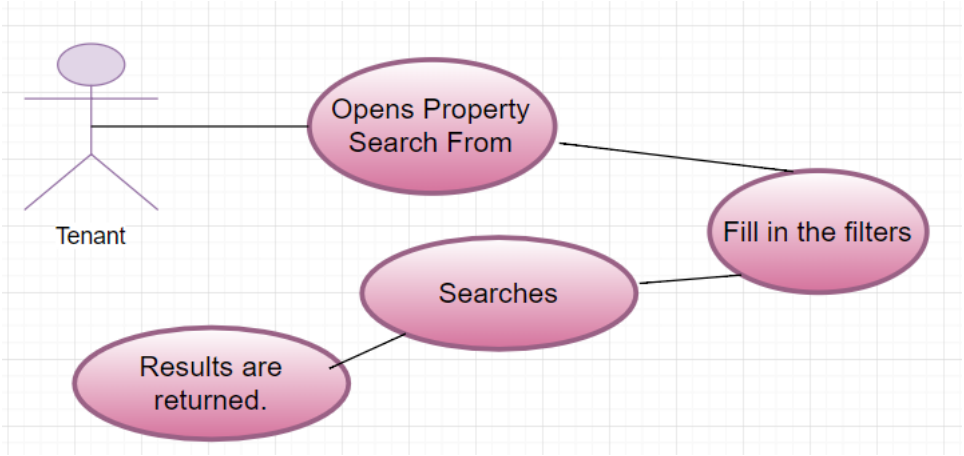
Alternative Flow

- The tenant has submitted the feedback.
- The landlord can view the feedback.
- The landlord reads the feedback.
- The landlord is property with a visualisation of the results.
- The landlord deals with issues records as necessary.

Post Condition

- The tenant has submitted a feedback survey for the landlord to read.

Requirement 5 <Property Filtered Search>



Description

- A tenant can search for properties based on specified filters.

Precondition

- There must be properties in the database, or no results will be returned.

Activation

- The tenant clicks the search button, and the form is rendered.

Main Flow

- The tenant arrives on the filtered search form.
- The tenant specifies the properties for the desired search. (County, town, price ect).
- The tenant presses search when they're ready.
- The corresponding properties are return to the user on the results page.

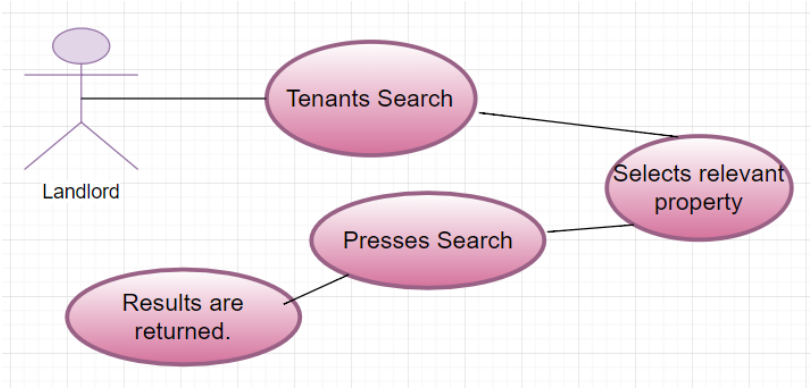
Alternative Flow

- The tenant fills out the search form.
- The tenant presses search.
- There are no results available based on the user's properties.
- The user is promoted to search again to find properties.

Post Condition

- The tenant has a list of properties to view and search through.

Requirement 6 <Tenant Search>



Description

- A landlord can search for a tenant based on a property matching to a tenant preference.

Precondition

- The landlord must have a property advert.

Activation

- The landlord arrives the tenant search form, selects a property and presses the search.

Main Flow

- The landlord navigates to the tenant search page.
- The landlord selects a property to search for tenants for.
- The landlord presses search.
- The results are returned.
- Any tenant whose preferences are equal to, or like the property specs are returned.

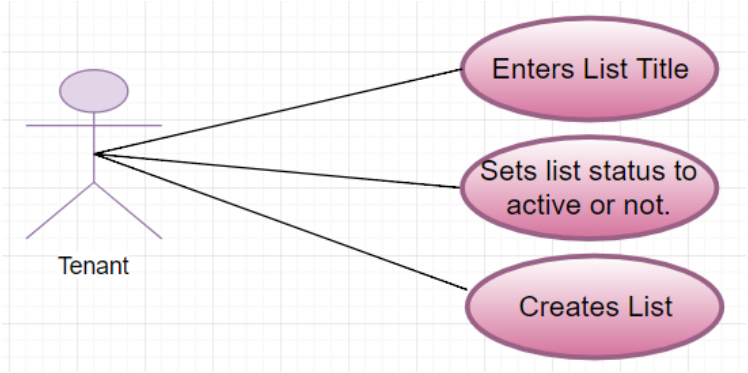
Alternative Flow

- The landlord selects a property to search for tenants for.
- The landlord presses search.
- There are no results.
- The landlord tried a different property or tried again later.

Post Condition

- The landlord has a list of properties to view and search through.

Requirement 7 <Create Property Watchlist>



Description

- A tenant can create a watchlist for they can add property adverts for review later.

Precondition

- The tenant must be signed in to their account.

Activation

- The tenants navigate to the main list page and clicks the create button.

Main Flow

- The tenant clicks create.
- The tenant fills in the form.
- The tenant gives the list a name.
- The tenant sets a status to active or not.
- The tenant submits the form.

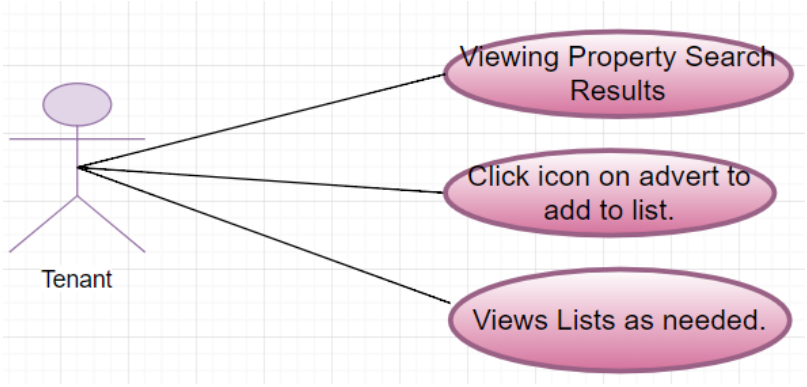
Alternative Flow

- The tenant has many lists.
- The tenant wants to add property to a certain list.
- The tenant clicks to a watchlist.
- The tenant sets that list to active, where properties will be saved to.

Post Condition

- The tenant has list(s) where properties can be added to view later.

Requirement 8 <Add property to watchlist >



Description

- A tenant can add a property they come across in the search results to the current active watchlist.

Precondition

- The tenant must be signed in.
- The tenant must have at least one list created an active.

Activation

- This activates when a tenant clicks the save icon the advert image.

Main Flow

- The tenant has performed a property search.
- The tenant finds a property they like and is an option.
- The tenant clicks a save button that appears on the advert add.
- The advert is saved to the active watchlist.

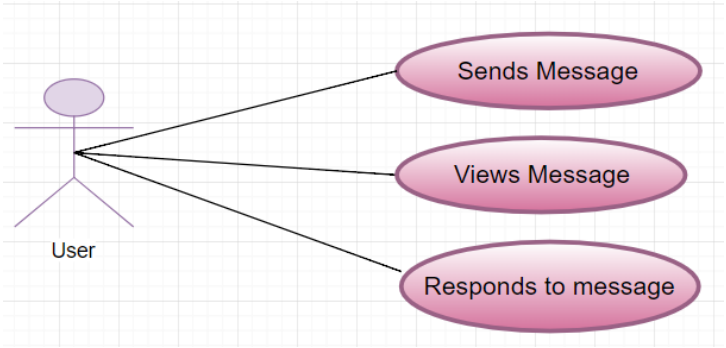
Alternative Flow

- The property is currently in the active watchlist.
- The property has a marked icon, to let the user know it's already in the list.
- The user knows that this property is already selected.
- The user picks a different property to the add to the list.

Post Condition

- The tenant adverts in their various lists.

Requirement 9 <Messenger>



Description

- The users can send a message to another user, views a message, and responds to a message.

Precondition

- The users must be signed in.
- The users can only initiate a message by clicking sign in on another user's profile.

Activation

- This activates a user clicks send message on another user's profile page.

Main Flow

- The user is currently on another user's profile.
- The user clicks on the send message button.
- The user is presented with a message box, prepopulate with a read only input field with the name of the recipient user.
- The user enters a message title.
- The user enters the message.
- The user clicks send.

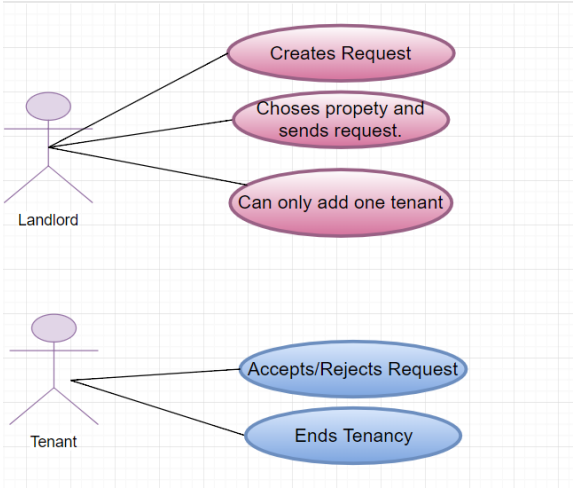
Alternative Flow

- The user is in their inbox.
- The user has received a new message.
- The user can see the send and the title on the home page.
- The user clicks into the message to read the message.
- The user can reply to the message if necessary.

Post Condition

- The user has received a new message or sent a new message.

Requirement 10 <Start a tenancy>



Description

- A landlord can start a tenancy with a tenant. This can be used to associate different tools between landlords and tenants.

Precondition

- The landlord must have an active property.
- The landlord/tenant must also not be in another tenancy.

Activation

- This starts when a landlord is on a tenant page and provides to start a tenancy.

Main Flow

- The landlord clicks the start tenancy button.
- The landlord fills out the form where they choose the property to use in the tenancy.
- The request is sent to the tenant.
- The landlord cannot send another request.

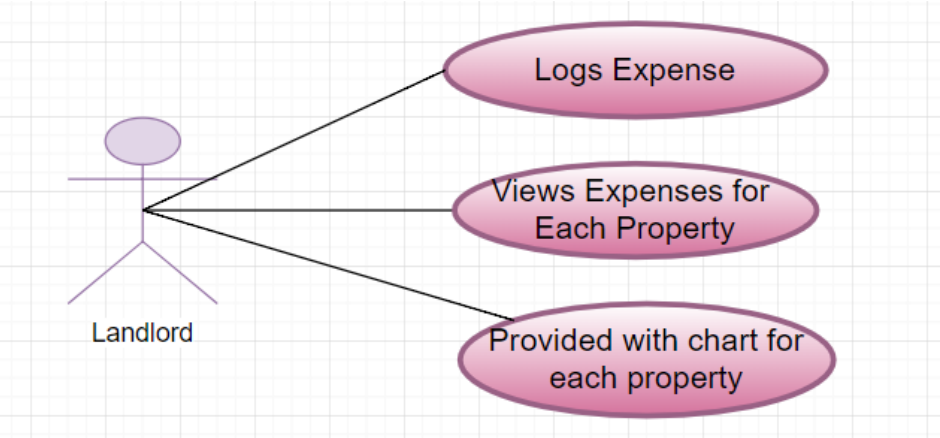
Alternative Flow

- The landlord has send a request.
- The tenant is on their profile.
- The tenant is provided with buttons to accept/reject it.
- The tenant accepts. The tenancy is displayed on landlord's and tenants profile.
- The tenant can then end the tenancy.

Post Condition

- The landlord/tenant are in a tenancy together.

Requirement 11 <Landlord Expense Logger>



Description

- The landlord can log expenses for each active property they have. This is good for keeping track of expenses and figuring out how to reduce costs.

Precondition

- The landlord must be signed in.
- The landlord must have a least 1 active account.

Activation

- This starts when the landlord navigates to the expenses page, selects the appropriate property, and logs the expenses associated with that property.

Main Flow

- The landlords click the property and is brought to that properties expense page.
- The landlord clicks the ‘log expense’ button, and they are brought to the log page.
- The log page renders a form which allow the landlord to enter the expense details such as title, category, cost and date.
- The landlord is redirected to the property expense page where all the expenses are show, and a chart breaking down the cost to category for that property.

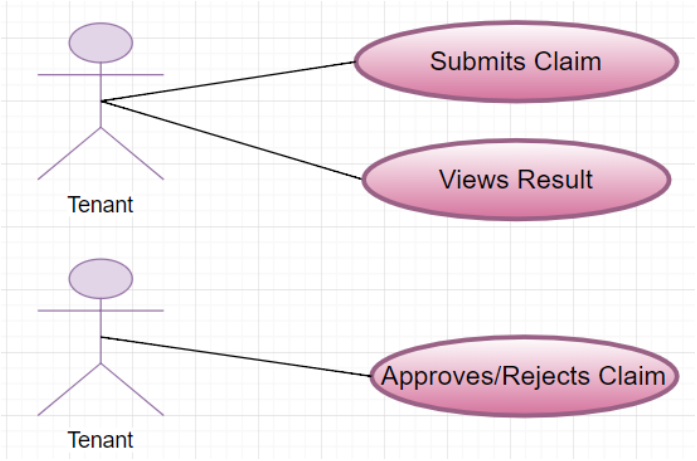
Alternative Flow

- The landlord has logged expenses for different properties.
- The landlord navigates to the expenses logger main page.
- A chart consists of the expenses from all the landlord’s properties is generated.

Post Condition

- The landlord has a list of expenses, and an overview of the expenses for each property, and all properties.

Requirement 12 <Tenant expense claimers>



Description

- The tenant can submit a claim for an expense to be reimbursed by the landlord.

Precondition

- The tenant must currently be in a tenancy.
- The tenant must be signed in.

Activation

- This starts when a tenant navigates to the claim page and clicks submit a claim.

Main Flow

- The tenant navigates to the claims page and clicks the submit claim button.
- The tenant is presented with a form to log expenses.
- The tenant fills in the form, with attributes such date, cost, description.
- The tenant has an option to include a link to a pdf/image of a receipt.
- The tenant submits the claim.
- The status is “Not Reviewed” by default.

Alternative Flow

- The landlord navigates their side of the claim page.
- The landlord can see a brief overview of all expense claims made.
- The landlord clicks into the individual expense to see more details.
- The landlord can have decided is this is a valid claim or not.
- The landlord can then update the status of the claim to approve or rejected.
- The tenant will this message then on there side of the section.

Alternative Flow

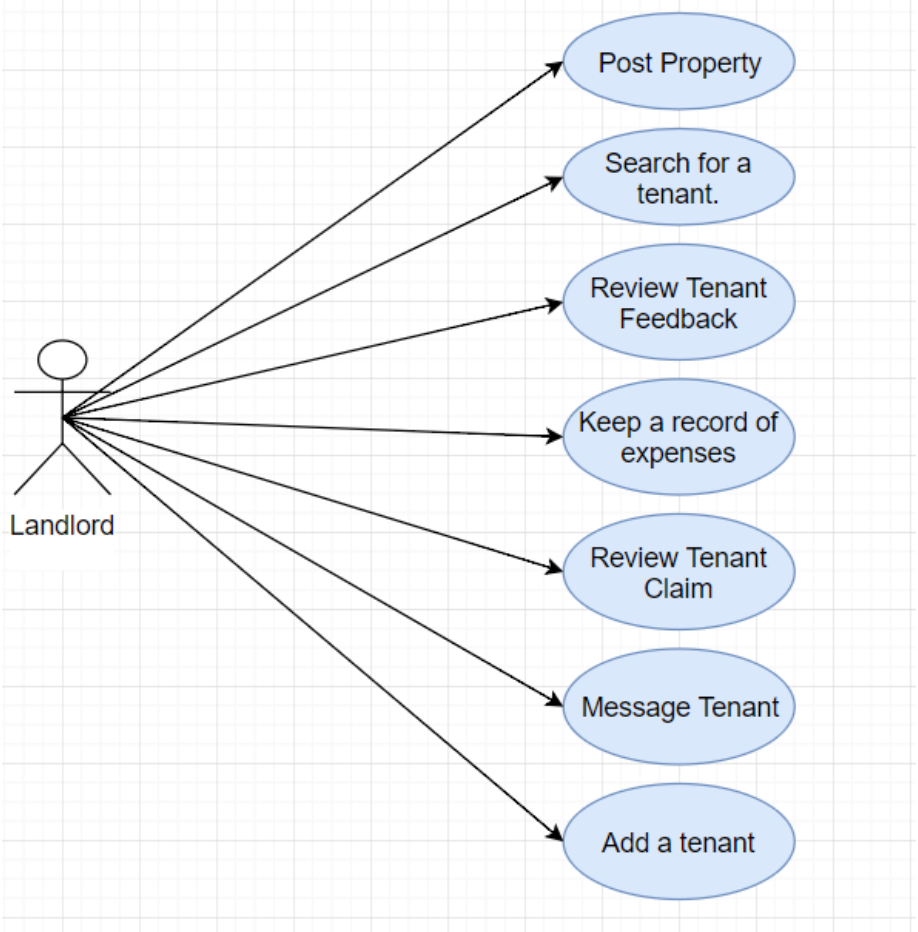
- The tenant sees the result of the claim.
- The tenant feels it's an unfair result.
- The tenant can use the message centre to further explain why they feel they should be reimbursed for the expense.
- The landlord can consider all of this, and update the status to be accepted, or leave the status as rejected.

Post Condition

- The tenant has the submitted expense claim either accepted or rejected by their landlord.

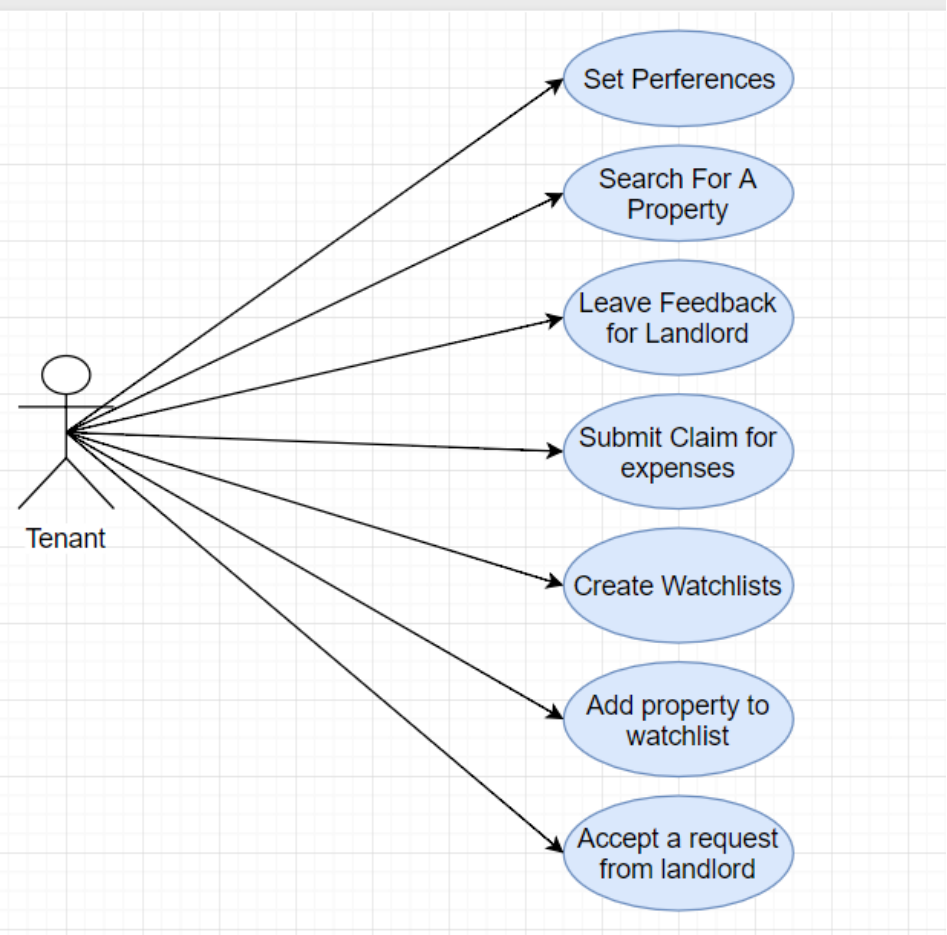
Use Case Diagrams

Landlord Use Case



This is the landlord use case and all the actions they can perform.

Tenant Use Case



This is the tenant use case and all the actions they can perform.

Non-Functional Requirements

Data requirements

County and Town Data

The application needs to have county and town data. These are stored in tables and are called when the users need to post an add, set preferences, and search for a property. The town data was scraped using R, from a technique learned in a previous module. As there were over 1300, it would have been inefficient to type them all. The counties were manually inputted. The data is stored in spreadsheets then imported to the database through phpMyAdmin.

Property Data

The app will need property data to work. The property data will be in the form of adverts. The adverts will be provided by tenants, and in return the landlords will be provided with accesses to the tenant database, and various useful tenant management tools. These tools will increase efficiency and, in some respects, allow them to identify how to reduce costs, and how to improve the tenants stay for the duration of the tenancy.

Tenant Data

The app will need tenant data. This will be provided by tenants when they sign up and fill in their profile. This will be used in the search when Landlords search for tenants. The data is provided in exchange for access to the property database, and access to landlords.

Tables

There are 10 main tables a user can interact with that hold data in the database. There are another 9 tables that hold data but can't be update by the user.

- ***Expense Claims***
  - Holds all the information related to a claim submitted by the tenant, as well as the status of the claim. (Has it been approved or not).
- ***Feedback***
  - Holds all the surveys submitted by the tenant to the landlord.
- ***Messages***
  - Hold all the messages sent by the various users. Inbox and sent box messages are determined by which user sent the message.
- ***Property Adverts***
  - Holds all the adverts and various related info submitted by the landlord.
- ***Property Expenses***
  - Holds all expenses logged by landlords to each property they own.
- ***Tenancy***
  - Hold the tenant. Landlord and the property associated in a tenancy when the add action is undertaken.
- ***Tenant Preferences***
  - Holds the tenant's preferences. Attributes include desired town, county and rent. This is used by landlords to find tenants in the tenant's search.

- ***Users***
    - This holds the user registration that. This also holds the user type (landlord/tenant), which is used to determine what user is show what content and determines what actions they can undertake in the system.
- ***Watched Properties***
    - This holds all watched properties, that are stored in the watched lists created by the user.
- ***Watchlists***
    - This holds all the watched list create by various users.

Performance/Response time requirement

The site should be developed to a high standard in relation to performance, and response time. Users to find the site easy to use. When a request is made, either by querying the database for information, or sending data to the database, there should be no lag. It should be a few seconds at most.

Load time should be a maximum of 5 seconds. Load time longer than on averages causes a user to leave the site. Improvement in speed can be achievement by caching data when applicable, so data doesn't constantly have to be requested from the database.

Availability requirement

The system should be available to any user with a device that can connect to the internet. As it's a web app, it's not system dependent, as it would be if I chose to develop an Android, or IOS app.

Recover requirement

The system should be able to recover when necessary. Recovery is the ability to revert to a previous version of the project. The key to achieving a successful recovery, and minimize the loss of user data, and source code is to regular back everything up. In the event of a disaster, the site and all associated data should be relatively easy to get back up and running with minimal data loss.

Security requirement

The application should be secure. To protect user accounts, a strong password policy will be enforced, passwords will also be encrypted, and not stored in plain text. The database where the data is stored will also need to be secured. As a result, research will need to be carried out to ensure the chosen host is reputable and will no allow data to be compromised. A strong password will also be used, to eliminate the change of third parties getting access to the database.

Maintainability requirement

This application will need to be maintained. The purpose of maintain is to keep performance high and prevent security vulnerabilities. This is to be achieved by using the latest version of technologies being used in development, also not using deprecated methods in the code. Maintainability is related to the quality of the code. If code

becomes scrambled and written inefficiently its increase the cost to deliver it to the user and develop the application.

Accessibility requirement

The application should be simple, and easy to navigate. There should be a focus on user experience i.e. the usability of the site. The ease of use should be incorporated into all aspects of the site. From the sign in, to moving around from page to page. The application should be responsive, so it's useable on all devices.

Error Checking requirement

This is very important, especially in terms of the log in system. The system must recognise when data must be of specific format, and only accept that info in the specified field. A user must not be able to type the email in the password field, and password in the email field, and be able to sign in successfully.

User Requirement

User Stories

Landlord – John Smith

John is a landlord who has a property to rent. He is struggling to find a tenant to needs the property he has to offer. His property is in 1 bed apartment in Dublin. The tenant that will take it is a single person/couple. He decides to post the advert on Rentable, to see if he can find a suitable tenant.

John posts the add up on the site, including all the details about the property. He then decides to start his search for a tenant. He navigates to the "Find a tenant" button and discovers that it can he can select his property and find a tenant that has requirements for a property like the one he has. After clicking the search button, he is provided with a list of potential tenants who could be interested.

John clicks through on a few tenant's profiles, and decides to use the chat button, to send the tenant a message and discuss a possible tenancy. The messages may yield results and John can go to that tenant's profile, click the "Add Tenant" button, fill out the details in the form and send the tenancy request to the tenant.

Once they're in the tenancy, John doesn't have to be finished using the site. He has several tools available to him to help him manage the tenancy. John can keep track of his expenses for the property. He goes to the property expense log page. He can select a property and is brought to a unique expense page for that property. He can choose to add an expense and fill in the details. The expense is then report back in the form of a pie chart, where John can see where he might be spending a bit too much. He can also from a list of all expenses, if he wants a more details breakdown of expenses, as well as clicking into an individual expense. John will also be provided with a report on all expenses from all properties.

John can receive feedback from the tenant in the form of a survey. Here John will be able to review feedback send by the tenant. Here the tenant can review the tenancy, and the results will be shown only to the tenant who showed it, and the landlord it was received by. The feedback asks several questions should has an overall rating, or airiness of rent. This allows both tenant and landlord to resolve any underlying issues and achieve are harmonious tenancy. The landlord is provided with an aggregated chart, of feedback from all tenancies, to help identify common issues.

John can review expense claim submitted by their tenants. Tenants may have to cover a cost out of their own pocket and look to be reimbursed by the landlord. John can review these claims, and either accept or reject them. Any issues can be sorted out by using the message facility.

Tenant – Amanda Lyon

Amanda is a tenant looking for a property to rent. It's a very competitive market out there, and it's hard to find exactly what she is after. She signs up to rentable and decided to try out the features that are offered on the site.

She sees can set her preferences for a property she likes. She goes the preferences page and fills in the required form. She sets the ideal county, the minimum bedrooms, bathrooms, and the max rent she ideally would like to pay. Once her details are set, and he has set her status to active, she can set the status to active, and she will be available to appear in the tenant search used by landlords to find property.

Amanda can create watchlists to assist in her hunt for a property. She can have multiple lists, that can cover many different property types if she so chooses. For example, she can have a list entitled "Apartments within a 5-minute walk of work", "Houses within a 30-minute commute of work".

Amanda can use the filtered property search to find a property. The search returns a refined list of property from the database. Amanda enters in the county, town, min and max of bedrooms, bathrooms, and rent, and presses the search button to retrieve results. When going through the results there will be a check icon on the property image. She will be able to click the icon and add that property to her watchlist. If Amanda clicks through on a property she will see the add in the entirety. She can read all the details and find the landlord who is advertising it. If she is interest she can go onto the landlord's profile, and message the landlord to discuss the possibility of her renting it.

If herself and the landlord are happy, the landlord can send a tenancy request to Amanda, which she can accept or reject. Once they are in a tenancy Amanda has some nice communication tools to help manage the tenancy.

Amanda can use the message centre to talk the landlord. The message centre is designed like a minimised email service. She has an inbox to display messages received, and a sent box to review message she sent. Having this removes the need for both parties to disclose personal contact information such as email addresses, and phone numbers.

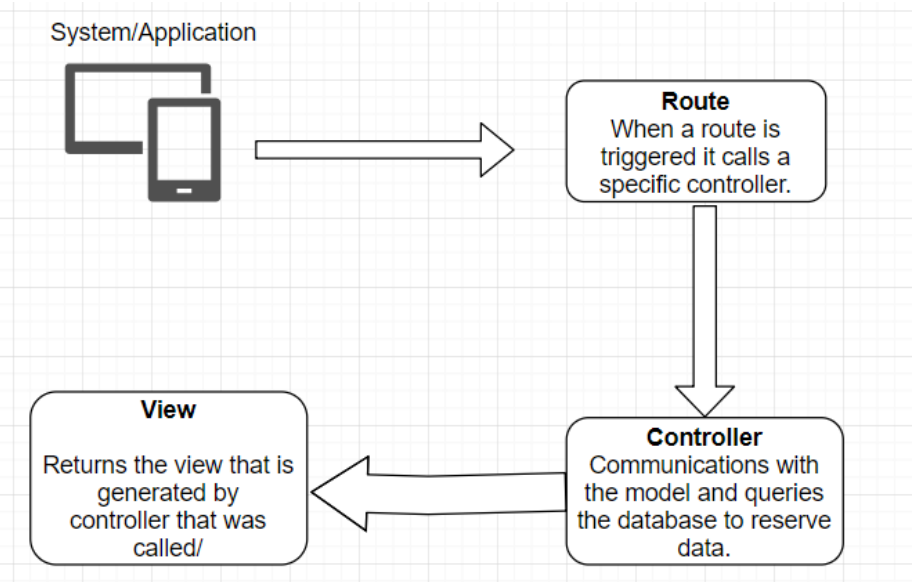
Amanda can review the landlord as often as she wants. The reviews are called feedback and are carried out in the form of a survey. The tenant, and landlords name and property address are pre-populated. The tenant is the asked a series of questions, with an option to leave a comment at the end. This is then sent to landlord.

Amanda can submit a claim to be reimbursed for expenses she paid for property out of her own pocket. Amanda fills out a claim form and can include a receipt to help with the claim. When she submits it, the status is 'Not Reviewed by default. When the landlord reviews it, they can set the status to either approved or rejected. The tenant will then be informed.

These tools can make the tenancy easier for Amanda and her landlord as there is a constant record of any issues, it's easier to find a property for her as she has to different options, and her feedback provides the landlord with useful insights.

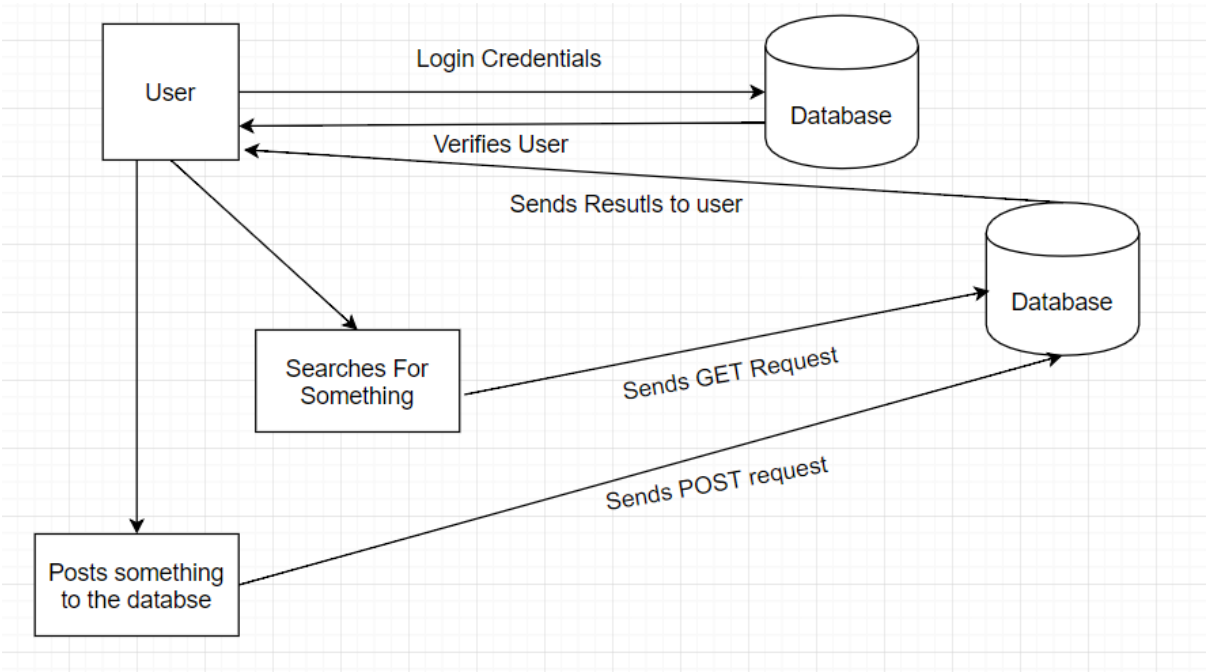
Architecture Design

The architecture for the Rentable application is a normal web app. It consists of 3 tiers. The presentation layer, the business layer, and the data layer. The presentation layer is shown to the user. This is how they interact the web service. Actions are performed using HTTP requests. To preforms requests the user must be logged into the system. As shown in the diagram below the user must be verified. There are a number of actions that users can take in this application, but it's not necessary to show them all, as they follow the same architecture.



Basic use of how the architecture works.

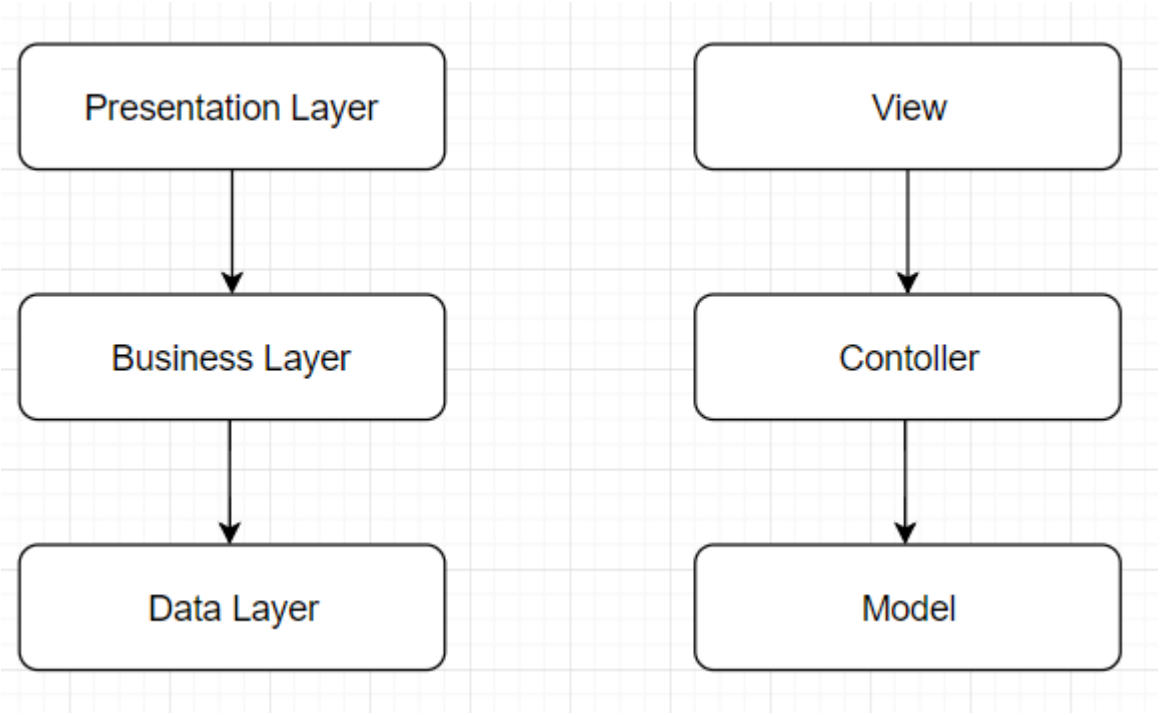
The user navigates to a page, such as the search property page. They click the search button, and this triggers a get request. The request takes the query, runs it again the database, and returns any result back to the user. The request logic is stored in the business layer, while the databases is the third layer. Layer 3.



System Diagram.

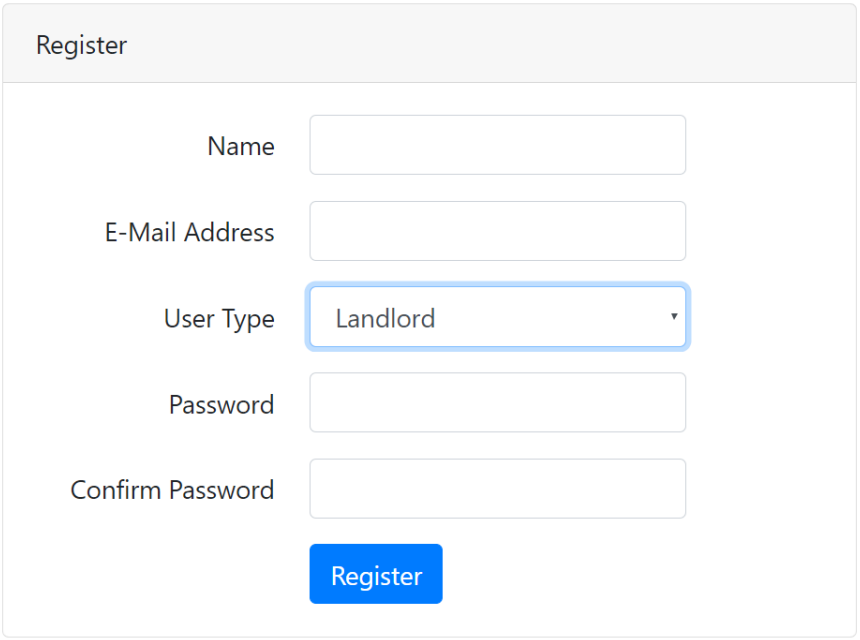
As discussed above the application is broken down into 3 tiers. The Presentation layer, the business layer, and the data layer. The application also follows the MVC pattern. MVC is model views controller. The views represent the presentation layers. The controller's representation the business layer, and the models represent the data layer. There's also a database which links with the models.

The views show the users the relevant buttons. A click on the button triggers a route which links to a controller in the application. The controller completes what ever action it's been programmed to carry out, by communicating with the model/database. The corresponding result is passed to the view, where it's rendered to the user.



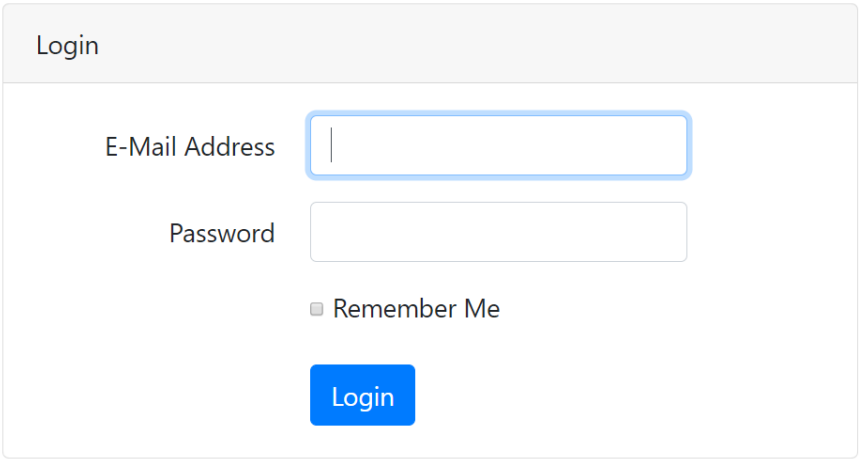
GUI

Registration



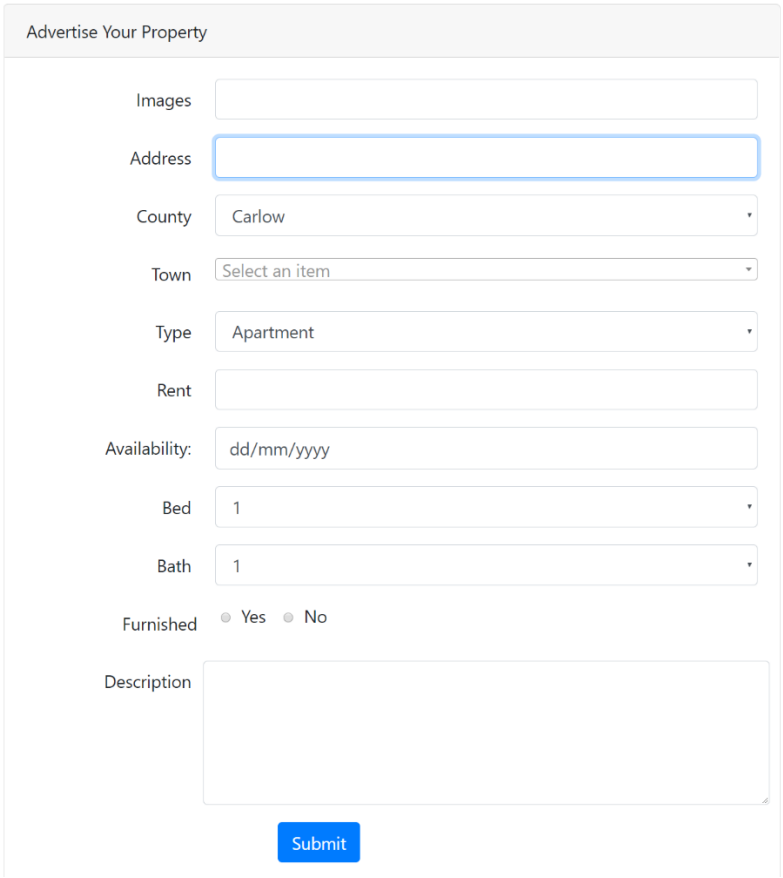
This is the registration page. It's a normal registration and it allows users to sign up and select their user type.

Login



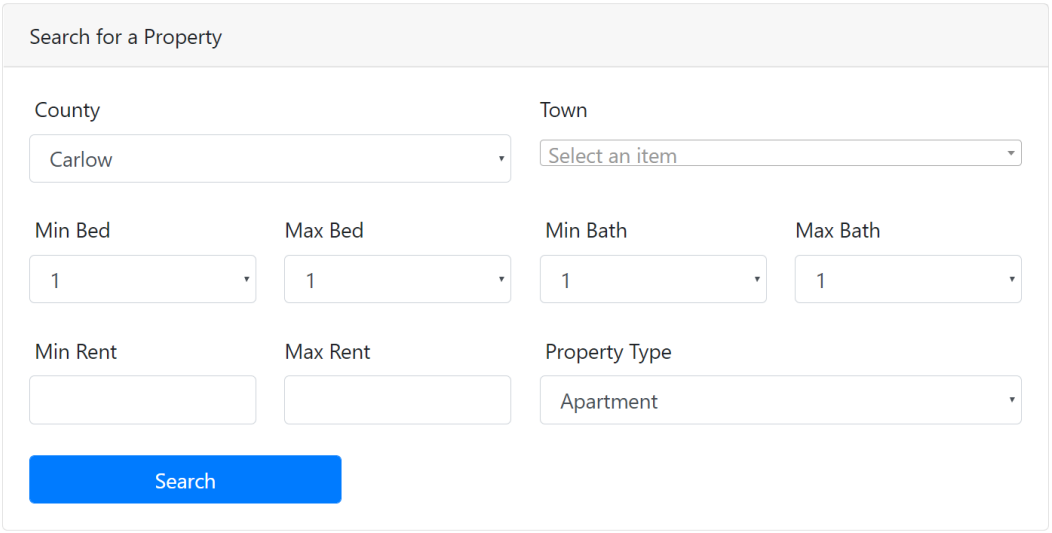
This is the login. Users must be signed in to do anything on the site.

Property Advert



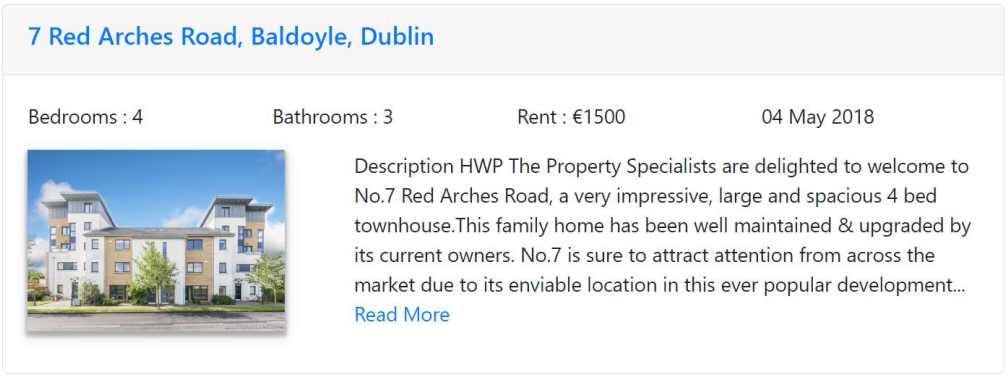
This is the page to advertise your property. Only landlords can access this page. Here they can fill in the form, and once it passes the validation, the advert will go through.

Property Search



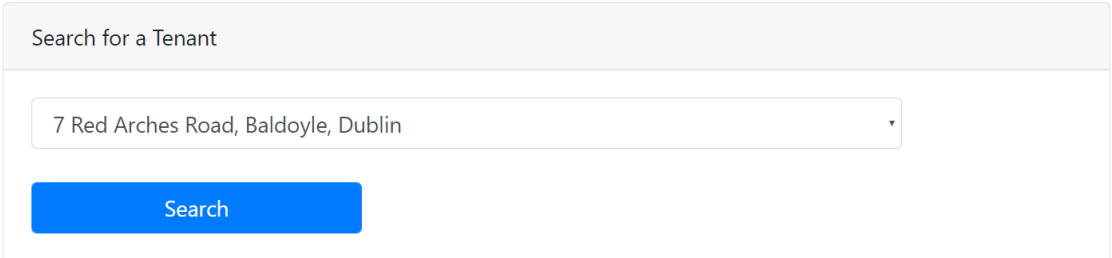
This is the property search. Users can enter in filters to narrow down the properties they want. When the press search, the values from the code will be used in a query which will return the relevant properties.

Property Results



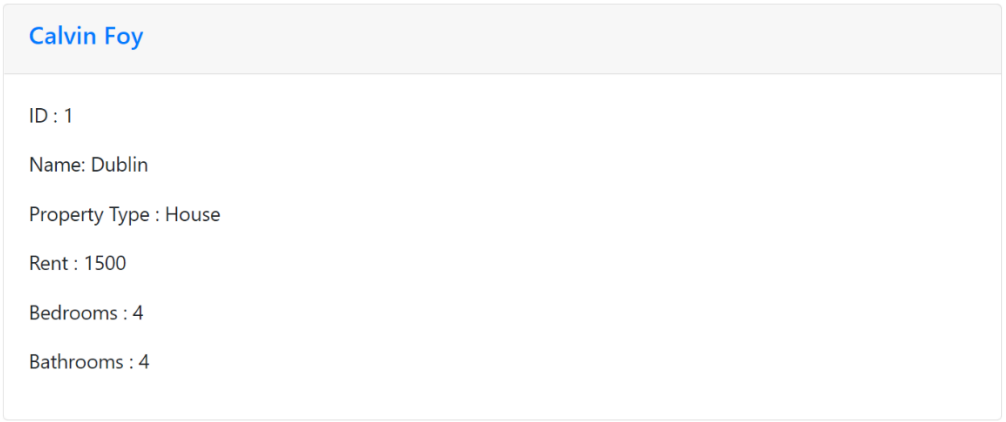
This is an example of the property results. This is only showing one result, but if there is many they will appear stacked in rows. If the user has a watchlist created, they will be provided but a button to add this property to the watchlist when they hover over the image. The user can click on the address to view more information.

Tenant Search



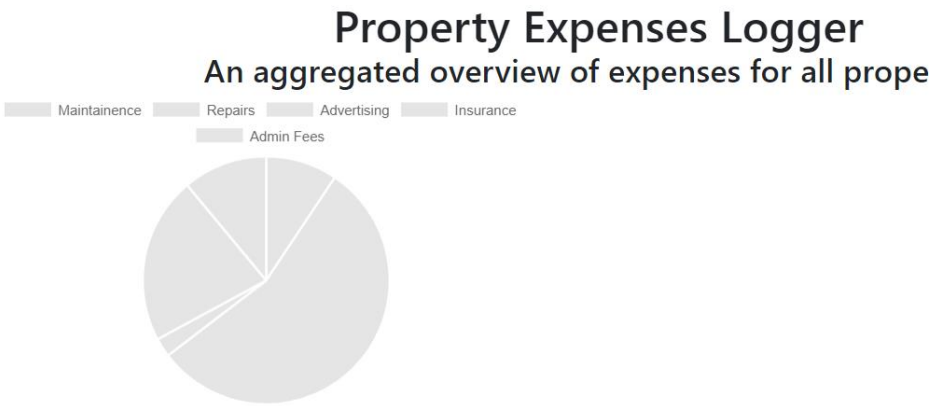
This is the tenant search page. It’s made straight forward from a user point of view. The landlord can select the property they want tenants for, and press search. The results will consist of tenants who have set preferences, have a status of active, and have similar requirements to the property specs.

Tenant Results



This is the results page for tenants. The tenants name links to their profile where they can be messaged etc. It also shows the exact specs they are looking for.

Expense Logger

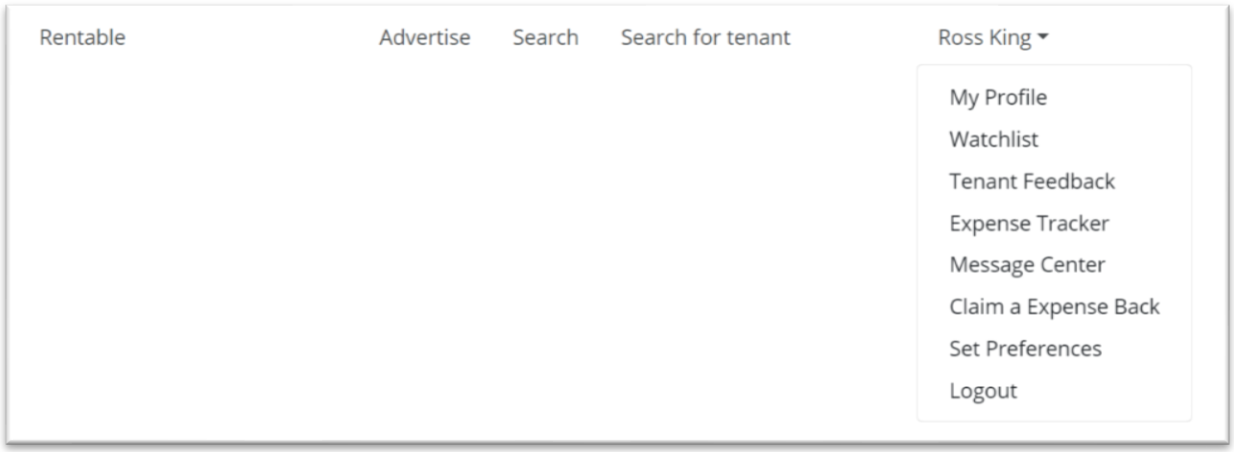


7 Red Arches Road, Baldoyle, Dublin

29 Myrtle House, Baldoyle, Dublin

This is the overview page for the expense logger. It lists all the landlord's properties. The chart choses aggregate costs from all the expense from all properties by category and total cost. The use can click through one a property and log expenses for that induvial property. The individual property will have its own chart.

Navigation



There are a lot of features in this application, as such they will not all be shown here, and this will be saved for presentation. This is a picture of the navigation which shows the majority of page available in the site. The pages all have sub features.

Implementation

Technical Approach

It was decided to build this application as a web app. Being on the web means more people have access to it as apposed to a mobile app. The app will be responsive, so mobile users will still have a good user experience, and if required the app can be built using react native to cater for both major mobile device platforms.

Another reason for choosing a web application, is down to the specs of the development machines accessible. Android studio does not run well on the machine that was used for development. It's a resource hungry program, and while it's brilliant, it would not have been a seamless development experience. The device is windows, which rules out using XCode to develop an iOS app.

The ide used to develop the application was initially atom. This was fine for the first few months, but as the application grew, atom became slow and buggy. After some research, it was discovered that a this is a common problem with using atom for big projects. Because of this a decision was made to switch to Microsoft VS Code. It's a lightweight text editor but runs big projects very well. VS code was a flawless experience.

The main language used in the development process was PHP. But vanilla PHP wasn't used. A framework called Laravel was used. Laravel makes development slightly easier and provides some beautiful tools out of the box. The first tool is known as blade. It's a templating language for the views.

The next tool is eloquent. Eloquent is an object relational mapper reduces in most cases the need to use SQL statement. Eloquent uses query builders which queries the database using very little code. This keeps the code easy to read and maintainable.

This application utilizes charts a lot for reporting. Charts are used to show landlords an overview of feedback results, an overview of expenses for individual properties, and an overview of expenses for all the properties.

The library used for the charts was called Charts.js. Charts is a html5 based JavaScript library that utilizes the canvas element. As it's JavaScript, JSON objects had to be used. The process of doing this was querying the databases, and instead of returning the data as a collection, the toJson method was used.

```
public function AggregatedPropertyOverview(){
    $user = Auth::user()->id;
    $result = DB::table('property_expenses')
                ->where('user_id', $user)
                ->get();
    return response()->json($result);
}
```

The above snipper choses a query that gets all expenses for all the user's properties. The data is retrieved as a collection, and its then converted from a collection to json data.

```
<div class="row">
  <div class="col-md-6">
    <div class="panel-body">
      <canvas id="canvas" height="150" width="300"></canvas>
    </div>
  </div>
</div>

<script>
  var url = "{{url('/aggregatedpropertyoverview')}}";
  var Cost = new Array();
  var Category = new Array();
  $(document).ready(function(){
    $.get(url, function(response){
      response.forEach(function(data){
        Cost.push(data.cost);
        Category.push(data.category);
      });

      var ctx = document.getElementById("canvas").getContext('2d');
      var myPieChart = new Chart(ctx,{
        type: 'pie',

        data :
        {
          datasets: [{
            data: Cost
          }],
        labels:  Category
        }
      });
    });
  });
</script>
```

In the view, the URL that controls the above method is called. This gets the data and uses it for the charts. The various data is passed to the chart data attribute and the chart is rendered. In this case it's a pie chart showing the cost labelled by category.

Testing

Unit Testing

Unit testing was tested to test some features in the application. Laravel comes preinstalled with PHPUnit and the tests can be easily run by using the command vendor/bin/phpunit. This runs all the tests in the tests folder. Of course, you can also choose to run one test by entering the directory after the command.

The features that were tested were login, registration, property advert creation, and tenant preferences submission.

Login Test

```
public function testLogin()
{
    $this->visit('/login')
         ->type('ross.king@gmail.com','email')
         ->type('password','password')
         ->press('Login')
         ->see(Auth::user()->name);
}
```

Login Test

This test ensures that the login works. It takes in the email address, and a password of a real user when they're on the login page and ensures that the signed in user is the user that credentials. If the user logs in, the test will successfully pass.

Register Test

```
public function testRegister()
{
    $this->visit('/register')
    ->type('Andre MacNamara', 'name')
    ->type('AndreMacNamara@test.com', 'email')
    ->type('Landlord', 'userType')
    ->type('password', 'password')
    ->press('Register')
    ->seePageIs('/register');
}
```

Register Test

This test checks that a user can register. It accepts all the user credentials and test it against the database. If all the attributes succeed in registering a user, the test will pass.

Tenant Preferences Test

```
public function testTenantPerference()
{

    $this->be(User::find(1));

    $this->visit('/account/preferance/create')
    ->type('Dublin', 'county')
    ->type('House', 'type')
    ->type('1000', 'rent')
    ->type('2', 'bedrooms')
    ->type('4', 'bathrooms')
    ->press('submit');
}
```

Tenant Preference Test

This is the tenant preference test. The tenant preference is a feature in the app that allows a user to submit preferences for a property they would like. This data is then used to allow a landlord to search for tenants.

This test the user can submit a preference form. It first gets a valid user. In this case the user being used is user 1. It then tests the route for creating a preference and checks all the input fields presses submit and if the preference is successful, the test will pass. If anything is missing, the test will fail as the validation rules won't let the data go through with missing fields.

Property Advert Test

```php
public function testPropertyAdvert()
{

    //Authenticates User id:1
    $this->be(User::find(1));

    $this->visit('/property/advertise/create')
    ->type('45 Red Arches Road', 'address')
    ->type('Dublin', 'county')
    ->type('House', 'type')
    ->type('1000', 'rent')
    ->type('11/05/2018', 'date')
    ->type('3', 'bedrooms')
    ->type('5', 'bathrooms')
    ->select('Yes', 'furnished')
    ->type('Test Description', 'description')
    ->press('Submit')
    ->seePageIs('/property/advertise/create');
}
```

Property Advert Test

This is the test for property advert submissions. The property advert submission is the create an add feature. This tests that adverts can go though successfully. It test all the attributes and if valid the test with pass.

As you can see from the below image we ran the phpunit command to test all the tests. This ensured that are tests were valid. You can see that the tests were ok, and the keyword OK is at the bottom. In various editors this line may be highlighted green to emphasis that the test has passed, but this is fine.

```
PS C:\Users\andre\Dropbox\College\Project\Rentable> vendor/bin/phpunit
PHPUnit 7.1.5 by Sebastian Bergmann and contributors.

....                                                                4 / 4 (100%)

Time: 5.39 seconds, Memory: 14.00MB

OK (4 tests, 17 assertions)
```

The ensure the accuracy of the tests, test that should fail were tested. In this case we changed the name of a button that one of the forms has. As there was no button with that new name, the test should fail.

Example of a failed test

```
1) Tests\Feature\MessageTest::testTenantPerference
InvalidArgumentException: Could not find a form that has submit button [sbmit].

C:\Users\andre\Dropbox\College\Project\Rentable\vendor\laravel\browser-kit-testing\src\Concerns\InteractsWithPages.php:634
C:\Users\andre\Dropbox\College\Project\Rentable\vendor\laravel\browser-kit-testing\src\Concerns\InteractsWithPages.php:614
C:\Users\andre\Dropbox\College\Project\Rentable\vendor\laravel\browser-kit-testing\src\Concerns\InteractsWithPages.php:594
C:\Users\andre\Dropbox\College\Project\Rentable\vendor\laravel\browser-kit-testing\src\Concerns\InteractsWithPages.php:581
C:\Users\andre\Dropbox\College\Project\Rentable\tests\Feature\TenantPerferenceTest.php:29

ERRORS!
Tests: 4, Assertions: 15, Errors: 1.
PS C:\Users\andre\Dropbox\College\Project\Rentable>
```

As you can see. The test could find the button name sbmit. That is right as the name of the button is submit. This verifies that the tests carried out were accurate and that they past. The failure shows us that simple mistakes can make a test fail and ensures that tests should be written carefully to pass.

Iterative Testing

Testing is carried out during the development process of the application form a user perspective. As new features are developed, they are put to the test. It is not enough to assume they will work as intended. Each feature is used, and different browsers, and system conditions are used to ensure that the system does work as intended.

For example, when the search property feature was being implemented, it was built up and tested incrementally. As each filter was added it they were tested on their own, as well as being tested together. So, when there was one filter it was tested, and when there were 6 create, it was tested. Different properties were entered into the database also to ensure that the aloes entered returned the corresponding properties. So, if it was specified that a min 3 bed property is wanted, 1 and 2 beds were not showing up.

This testing is crucial to the development as it needs to be ensured that each feature works, and just break other parts of the system, before moving on to the next phase.

User Acceptance Testing

As the application was being developed, other students were sought to test out the features. This was a quick process usually and no notes were taken. It was a think aloud test. Users used the application and spoke what thought the system should do, and this information was considered and improved on if necessary. Usually the system worked as intended. But the test was helpful to verify that the application worked as well as it can possibly can.

Trunk Test

The trunk test is a test that analyses the site navigability. By modifying/upgrading the site to pass the trunk test, you will be creating a user experience that is comfortable to users, and the site will more likely be used for the intended purpose it was developed for.

For the purpose of this test I asked three classmates to use the site and allow me record the results.

The questions asked were based on what was taught to use in a previous module. Notes should be taken on the amount of time taken to answer, user findings and any differ enations.

User 1

Question	Answer	Time Taken	Correct/Incorrect	Notes
What type of site is this?	This seems to be a property related website.	6 Seconds.	Correct.	
What page am I on?	A page to advertise a property.	15 Seconds.	Correct.	Wasn't initially sure what the page was.
What are the major sections/functionality on this page?	Some information about what to do, and a form.	10 seconds	Correct.	
How can you navigate off this page	Using the navigation menu, or the home button.	5 seconds	Correct.	
Can you navigate the home page?	By clicking the logo/page title.	5 seconds	Correct.	

User 2

Question	Answer	Time Taken	Correct/Incorrect	Notes
What type of site is this?	A rental site.	3 Seconds	Correct.	
What page am I on?	A page to send a message to a user.	5 Seconds.	Correct.	
What are the major sections/functionality on this page?	Sending a message.	5 Seconds.	Correct.	
How can you navigate off this page	Using the navigation menu, or the home button.	7 seconds	Correct.	
Can you navigate the home page?	Clicking the logo in the header.	3seconds	Correct.	

User 3

Question	Answer	Time Taken	Correct/Incorrect	Notes
What type of site is this?	A property rental site.	10 Seconds.	Correct.	
What page am I on?	A registration page.	3 Seconds.	Correct.	
What are the major sections/functionality on this page?	Allows a user to sign up and pick if they're a tenant/landlord.	10 seconds	Correct.	
How can you navigate off this page	Use the home button.	5 seconds	Correct.	
Can you navigate the home page?	By clicking the logo/page title.	5 seconds	Correct.	

Conclusions and future work

This was a very enjoyable project. Working on an app that real people could actually use is brilliant, and building a app from scratch on your own where you have free reign is a wonderful experience. Using Laravel was a blessing, and I'm defiantly going to look for a job using this framework. It was applause t work with.

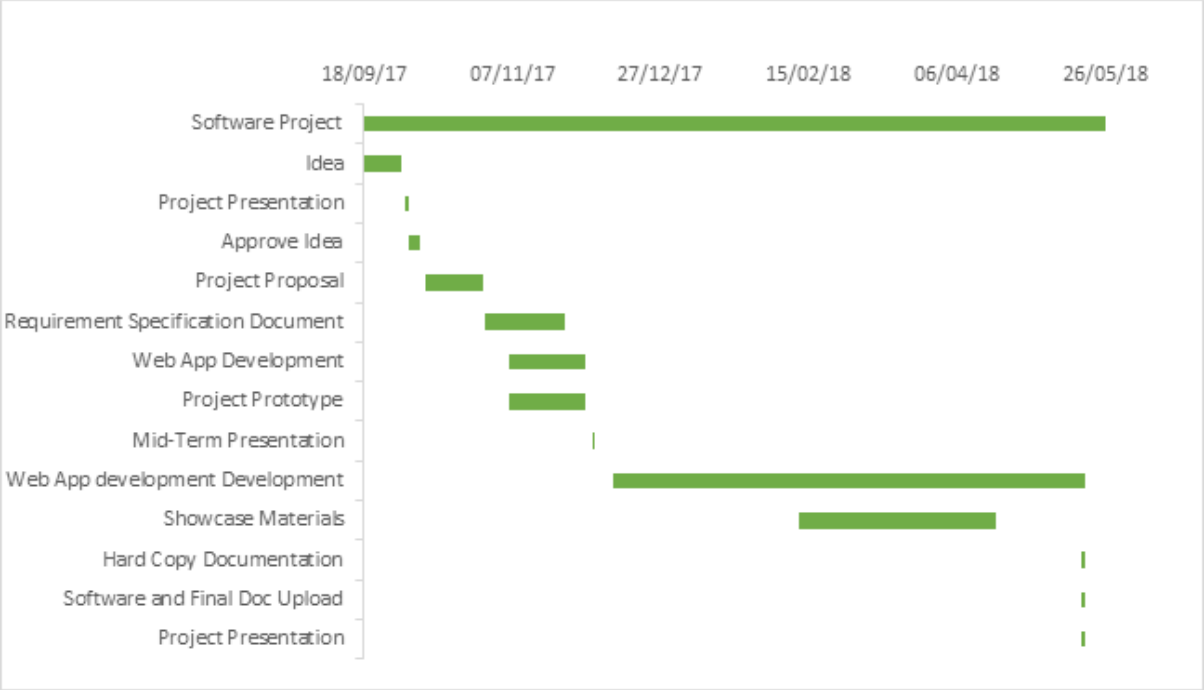
I plan to continue working on this project in the future. I have a lot of ideas that I would like to add, and the better it gets the better of a project/talking point I have during interviews. This app can be used as a playground to test new technologies. For example, If I learn CSS grid, or react, I can delete bootstrap out of the app and implement the technologies mentioned. This app is a base to build off when it comes to trying out new tools and technologies.

Regarding other features I would like to implement in the future are

- Ability for landlord to pay the expensive once it's approved.
- Pay rent.
- More reporting – Stats
- Lease creation.
- Accounting – The expenses is a good start, but getting income in, and creating a tax return statement would be huge. If the landlord could then directly pat revenue this would circumvent the need to pay an accountant.
- Handymen search – If a landlord needs a plumber, or another service, they can find them in one site.

Appendix

Project Plan



Monthly Journals

September 2017

18th September 2017

Today was my first day back in college for my final year. I was thrilled to be back, as I can't wait to get stuck into this semester's course load. Today I only had one class which was Software Project. The class was an hour long and the intro of the project was covered, such as the objectives, milestone, marking scheme.

After class, all I had scheduled for the rest of the day was independent study. I took this opportunity to get back into the gym, I spent an hour or so there. Afterwards i made some progress with Node.js by continuing through "The Web Developer Bootcamp" course on Udemy. I finished off the intermediate express section by going through POST requests. I have spent the summer working through this course, and I plan to write my project in node.

Project wise, I have a general idea. A web app for landlords to search for tenants, and vet their chosen tenants by reviewing from historical tenancies. There will be many other features, but this i my starting point. I hope it's approved, property I an area I have a keen interest in, so I believe I will enjoy the process of developing this application.

22nd September 2017

Today ended up being a very annoying day in college. Up until 3 o'clock all was fine. At 3 my Data Application Development lab was on. There was quite a big amount of work in the lab if I wanted to get done. I couldn't sign into Citrix. I tried multiple computers, different instances, but I could not get in. I left class early as it was pointless just sitting there.

23rd September 2017

I had plans to get all my assignments done this weekend, and progress with my node course. But as it goes, i underestimated how long all my tasks would take. I started with the Data App Lab I couldn't do yesterday due to technical issues. There was a lot of work in it, as well as thinking about how to solve the problems, I spent an awful amount of time researching about R as this was my first proper encounter. I get through most of it, but I didn't get it finished. I stop at about 6 as I couldn't keep working.

24th September 2017

I finished my R lab and started my Intro to AI research. It is worth 4% of the module. Not loads but I'm chasing a 1:1 year so I need all the marks I can get. I spent a couple of hours researching and started the assignment. I finished early enough so I could make some progress learning Node, watch the football and just generally relax.

25th September 2017

I had software project at 9. Here Eammon discussed the project pitch and it really turned out to be a simple affair. He also stated we should start on a project plan. Even though I have a good idea about how to get through the project in my head. It might not be so good if I put it on paper.

26th September 2017

One class today. Strategic Management at 3 until 5. It wasn't exactly full, and I think if it wasn't for the fact I never missed a day in college ever, I wouldn't have gone in. It would have been a shame to ruin this record in the final year. It's probably a good thing, as I will attend every strategic management class on Tuesday just to keep this attendance record.

27th September 2017

Awful start to the morning with the weather. To avoid the rain, I got a lift to the train station, and got a late train. I arrived into college at 9:15 for web services. As there were few people in (due to the weather?) the class didn't start until more people showed up, so I just uploaded my AI report.

29th September 2017

The end of the week, and I have a day full of labs. I didn't go to one. I spent all day in the Library with Sean getting out Web Services report done. We were there from 10 until 6:30, and we finished it, all our research, citations and the 3500-word report were done in one day. Maybe we need to manage our time a bit better to avoid skipping classes.

October 2017

3rd October 2017

Today I had my Project pitch. My idea is a few words is a web app to try match the best landlords/tenants with each other, so they have an easy tenancy.

I presented in front of Lisa, Frances, and Michael. I have the 3 of them in the past as lecturers. They are nice, so I knew I had nothing to worry about. They like the Idea on a technical perspective. But they had an issue with possible Data Protection issues. They didn't like the idea of all user's past activity on the site. But that's the point of the site, i argued. I wasn't forcing taking anyone's data forcibly. They understood, but suggested I use the idea for a different aspect. I have a big interest in property, so the Landlord/Tenant aspect was used to keep my interest in this.

The idea was rejected. I'll talk to Eamon tomorrow.

4th October 2017

After being freaked with getting rejected, I spoke to Eamon. He told be the data protection concerns don't matter and I can go ahead and do it. He even checked his sheet and noted that my idea was accepted. Weird considering, I assumed it was rejected as they all said no. They must have reconsidered when I left the pitch. I'm thrilled now that I can do this.

13th October 2017

I spent the last couple of days working on my AI chess game. It's hard to get it 100%. There's so many conditions to get right. I got most of it working. I spent the day in the library attempting to finish it off. It was due at 5, I lost interest at 4 and just uploaded it. Realistically I wasn't going to get the final part done in under an hour.

16th October 2017

Due to Storm Ophelia, all schools and colleges were closed on Monday. So i had a day "off". I spent the day at home doing work.

19th October 2017

I had my AI presentation with Arghir today. It was fine, except I forgot to restrict one side of the board to prevent pieces from leaving. I will lose some marks but nothing to major.

21st October 2017

I spent today doing my multimedia CA1. It's due tomorrow, and I wanted to focus on strategic management tomorrow.

22nd October 2017

I did part 1 of my Strategic Management CA today. The pestle analysis. I spent a good chunk of the today doing research, and much less on writing. But I got part 1 done.

23rd October 2017

I did part 2 of my strategic management CA done today. The tows analysis. It was easier and quicker than part 1, as the pestle fed into the tows.
Eamon said today, the proposal isn't due on Friday. He just gave us that deadline, so we get it done. This give me some leeway this week as i need to focus on Data Applications CA. I can finish the proposal on the weekend.

24th/25th October 2017

I spent these days working on the proposal and studying R.

26th October 2017

I met with my supervisor today Paul Hayes. I spoke with him about my project idea. He wasn't keen on it when I started talking, but after explaining everything he seemed to think it was good. He gave me ideas such as adding in Geolocation to provide directions from current location to find the house, and to find local amenities. He said this will boost my marks.

27th October 2017

Today was my last day of term. I had an R CA in data application development. I spent all day in the library studying, as it was worth 25%, and I needed to do well. The CA was ok. I felt I did decent enough, as i studied what he said and targeted the questions I knew. The feeling was not same around the class, as people came out saying "worst exam ever". Hopefully I don't think I did better than I did.

28th October 2017

Today I came in to the library to finish off my project proposal. I got most of it done, and I'll finish it off on Sunday and get it uploaded.

November

6th November 2017

Today was my first day back after reading week. I have to say I felt having a week of no classes/assignments really helped. I spent the week trying to understand Android, and the topics from Web Services. I understand what is going on now, and this will help greatly going into the second half of the semester.

9th November 2017

I had assignments to work on after class today. I finally got my part of the web services CA done. I spent all week trying to get JSON data to load from an external API, and I have it working now. This will take the pressure off for tomorrow.

10th November 2017

Web API and AI CAs were due today. I had my section of API done, Sean had the logic of his done, and it worked as a basic Java application. All we had to do was implement a web service for that. We got it working within 10 minutes. The rest of the day was spent in between classes, and doing the AI CA.

10/11th November 2017

I spent the weekend working through a Data Application Tutorial in python, starting my requirements and working through the node.JS course to get started on my project.

20th November 2017

I felt that today was a very productive day. Not in the sense that I was progressing with assignments, but I spent the day going through some of my Udemy course on NodeJS and Mongo db. This section was focused on creating relations between tables, and I understood it immediately. I think my prototype will be easy enough to make, as I won't have to spend hours researching, this course covered loads.

23rd November 2017

I had my meeting with Paul Hayes today. It was only 10 minutes or so, but it was well worth it. He was happy with my requirements document and didn't see much that could be changed. He did address a concern he had with the project. The rating section of the app can only be done once a tenancy end. Paul didn't think this would draw users to the site and said I might lose marks in terms of innovation as even with using mock data, to avoid data protection issues, it would be hard to argue why someone should use this site.
He gave me a great suggestion.
I should flip the application around, by having the main feature of the site being a feedback section. Every 6 months or so, a tenant can fill out a survey about how the tenancy is going and this data is presented to the landlord. As this is private between the landlord and tenant there would be no data protection concerns, and it's a more innovative project. Paul said that my documentation looked very good, so hopefully that's a good sign.

30th November 2017

I uploaded my midpoint presentation today. Such a relief to get it done and out of the way. I can spice up my prototype for the presentation now on the 4th December.

December.

4th December 2017

I had my midpoint presentation today. I feel like it went ok. Not great. I had to defend my idea? I got some information regarding if it will draw users, so I'm going to pivot it to a property management app, as well as a search app. Simon Caton suggested changing the app to commercial, but I don't feel it will make a difference as the functionality will be the same. It'

15th December 2017

Today is a busy day. I have to demo my mobile app to Dominic at 2pm. My web services project was due tonight, and I need to go home a record a video demo. The midpoint results were also out today. I'm very disappointed. I only got 50%. I was optimistic after Paul said my requirements looked good, but obviously my project let me down.
This is probably my last entry of the year.

16 - 20th December 2017

I just spent this time working on my data applications report. Thank god that was extended. I'm looking forward to taking a week off and recovering them such an intense semester.

27th - 28th December 2017

I spent time doing some more of my node course, but as I'm using it, I'm not enjoying it, and there doesn't seem to be many jobs in Node in Dublin now. It might be risky but I'm going to looking into using either PHP or C#. They seem to be big job wise. Python was an option, but my experience in Data applications relay put me off.

January

2nd - 12th January

This period was spent studying for exams. I had 2 Strategic Management and Artificial Intelligence. I feel Strategic went well, but AI was a disaster. I'll say I'll scrape a pass.
My last exam was on the 12th, and I have until the 26th off.

13th - 19th January

I spent this time studying a course a bought on Laravel. I have decided to change my technology from node to Laravel. Simply for the reason there are more jobs going in Laravel. I finished the course, which covered the basics and recreated the user authentication, upload form, user association with a property and landlord/tenant differentiation.

I have got my self back to the same position I was in for the midpoint, and even more. I have user authentication, user roles, and the ability to allow landlords post a property. Laravel Is such a beautiful language, eloquent and blade make the experience cleaner. I'm glad I made the switch.

19th - 31st January

Back in college. I got more work done this week. I got users photos previewing before they upload. I'm working on allowing upload multiple photos, but this is a struggle.

I have decided not to allow users upload photos and allow them to paste in a link to a property. This link will then be rendered in a image tag. This will do for the project as it will allow me to move on to other features. I also don't have to worry about storing files.

February 2018

January 22nd

I forgot to mention in January entry. I had a supervisor meeting with Paul on January 22nd. Here we discussed the midpoint results. I was really disappointed as the marks didn't reflect the work put in. We were trying to think of other features to expand on the application and Paul told me to email him next week to organise another meeting.

February 1st

I emailed Paul yesterday to organise another meeting. He hasn't replied to me, so I'm to add my features in and discuss next time we meet. The new features are a payment system (tenants can pay rent), a Matching system. Tenants automatically presented with properties that meet criteria. An expense system for landlords. Landlords can enter an expense and a category. This will display a pie chart breakdown.

February 12th - 18th

I spent this time doing some project worked. I started research on my filtered search to see how I get on with it. I have attempted to upload multiple photos, but there isn't much in terms of tutorials and one I did find has an error. I found the same solution everywhere, but it wasn't working in my project. I'll have to find a different method.

February 19th

I started my showcase write up in advance of it starting next week. Eamon said it looks good so hopefully I'm approved straightaway next week.

February 26th

My profile was approved. I have it saved into the system. Helen approved it almost immediately. I just need my picture taken next week to finalise it.

March

March 2nd

I had my photo approved my showcase profile. That is now everything submitted and hopefully I get the full 3%.

March 9th

Today was spent working on the Usability design project. I prefer using independent study to work on my project, but this had to be done today.

March 10 - 11th

This weekend I spent on my project. I get the property forum editing and updating working on the Saturday. On the Sunday I got the filtered search UI, and backend working 100%. Users can specify a county, a town, min, max bedrooms, min, max bathrooms, min, max rent, property type and only properties matching that criteria will be shown. I thought this would take me longer than 1 day, so I'm ahead of schedule. I just need to add some validation to this now, but it works.

March 12th – 31st.
I have done no project work now since the last post. The time off before the project will be viral.