Simon Fraser University

Realtorest - Iteration 3

Requirements and Specification Document

Project Group 6

CMPT 276: Introduction to Software Engineering

Kevin Cheng

Nam Nguyen

Drishty Dhiman

Haramrit Toor

Malaika Qureshi

Requirements and Specification Document

Project Name:	Realtorest	
Client Contact:	Aman Dhiman	250-588-0739
Project Due Dates:	Pre-production	February 25, 2024
	Iteration 1	March 12, 2024
	Iteration 2	March 26, 2024
	Iteration 3	April 9, 2024
Documentation:	Project Proposal	February 25, 2024
	Iteration 1	March 12, 2024
	Iteration 2	March 26, 2024
	Iteration 3	April 9, 2024

Project Abstract

Realtorest is a real estate web application that is created for Aman, a realtor located in British Columbia. The application displays a list of properties that can be sorted based on price, location and room features, and the available map feature can assist users to explore properties in different locations. From a general users' perspective, users can create an account, manage their basic information, control their property choices with features such as 'Save as Favorite', and connect with Aman to discuss or set up an appointment. The application is specifically tailored for our client Aman, who is the middleman that helps property sellers and buyers connect with each other. This application focuses on helping Aman upload and display information of properties that are being sold and help potential buyers who are interested in those properties to make the purchase.

Client

Aman is a realtor in British Columbia who does not have a personal website for his real estate business, and due to the competitiveness in the industry, a personal website can benefit Aman a lot by helping him connect with potential clients and widening his business scope. Hence, the web application Realtorest is created to help Aman connect to property buyers and sellers. The description will outline two types of users that will use this software: Aman, the project's client, who will be the admin of the website. And the general customers, who are home buyers, real estate investors, renters, and real estate professionals. Detailed below are the high-level descriptions of the features that Realtorest has to offer. Aman can be contacted through his phone number: 250-588-0739.

Competitive Analysis

There are many personalized websites tailored for realtors similar to Aman, hence it is very competitive for the project to have its own distinctive features that stand out from the rest of the other websites. However, Realtorest is different from many other realtor websites in which it does not have any advertisements, because the project is focused on the client, rather than profit like the others. In addition, under UI perspective, most of the realtors websites that the team encountered are usually very hard to navigate and not user friendly, since there are many website features and interactions that all appear at once. Hence, to avoid this problem, Realtorest will focus on the customer interaction experience by creating a minimalist layout that can help the user to navigate with ease, while also providing sufficient features.

User Stories

Introduction: Realtorest is a website that helps the client, who is a realtor, connects with his potential customers; hence the actors involved will be the administrator, registered users, and unregistered users. The administrator will be Aman, our client who can add new listings, manage existing listings, and contact interested customers. Registered users can view listings, contact Aman for more information, and bookmark their favorite properties. Unregistered users have the ability to browse the listing and contact Aman for more information.

Site Navigation

// Add additional stuff from iteration 3

Actor: Alice is an unregistered user visiting the website for the first time. Alice wants to look around and see what information is provided by the website.

Precondition:

- Alice lands on the home page, where she can navigate to different pages of Realtorest using the navigation bar on the top of the page.

Action:

- Clicking on “Realtorest” will redirect Alice to the homepage.
- Clicking on “About Us” will cause a drop-down tab to appear from the top containing Aman’s name, title, phone number, clickable social media links, and a brief introduction of Aman.
- Clicking on the “Listing” link on the top nav bar or the “Explore Our Listing” link in the center of the home page will redirect Alice to the property listings page.

Acceptance Criterias:

- The home page needs to be able to load while the website is up and running.
- When the user clicks on a button, they need to be either redirected to the corresponding page, or the relevant information needs to be displayed.

Tests:

- The server can be hosted up and running.

    - RealtorestApplicationTests.java > contextLoads
- The home page can be displayed.
    - ControllersTests.java > testShowHomePage

Iteration: 1

Story Points: 1

Property Listings

Actor: Joan is a new user, who is using Realtorest for the first time and just wants to see what the website has to offer.

Preconditions:

- Joan clicks on the “Listings” or “Explore Our Listings” and gets redirected to the listing page.

Actions:

- Joan can see all available properties on the listing page, such that only 9 properties will be displayed at a time. She can use the pagination bar below the list to navigate to see more properties.

Acceptance Criterias:

- The listing page must only show 9 properties at a time and navigating through the pagination will display different properties.

Test:

- The property listing page can be displayed.
    - ControllersTests.java > testShowPropertyListingsPage

Iteration: 1

Story Points: 2

Property Listings - Viewing Details

Action: Jeremy is on the property listing page, and he wants to explore some potential properties that he will be invested in the future.

Preconditions:

- He is interested in seeing more information about a property near his workplace and decides to click on the figcaption under the property’s image carousel.

Actions:

- A modal window will popup and display a bigger images carousel of the property. The carousel will automatically change the picture every few seconds when Jeremy is not hovering his cursor onto the picture.
- A detailed description of the property and the realtor’s information will also be displayed under the carousel.

Acceptance Criterias:

- A popup modal window will be displayed after the user clicks on the property figcaption.
- The modal window will display the correct information of the property, and it should automatically change the carousel’s picture every few seconds if the user is not hovering their cursor on the picture.
- Clicking outside of the modal window will elegantly close the window.

Iteration: 2
Story Points: 3

Property Listings - Filtering and Sorting

Actors: Emily, a young professional on a budget, looking for an affordable home with 3 bedrooms.
Preconditions:

- Emily uses the search filter on the listing page to filter out properties with 3 bedrooms and uses the bar to sort the property price from low to high, and she then clicks on the "Apply Filters" button.

Actions:

- The page presents the property that has 3 bedrooms.

Acceptance Criterias:

- The page displays the property that matches with the user's filter criterias. In other words, none applicable properties to the filter search will be hidden.

Iteration: 2
Story Points: 2

Account - Registration

Actor: Bob is an unregistered user wanting to register for an account.
Precondition:

- From the home page, Bob clicks the Sign In button on the top right, then clicks "Register Here" to visit the registration page.
- Bob then fills out the registration form, and he clicks the "Register" button.

Action:

- The system will validate the information Bob has entered. If any of those conditions fails, the system will prevent the form from being submitted.
- The user has left one of the fields empty.
- The contents of the two password fields do not match.
- The password is not strong enough (minimum of 8 characters, and include uppercase, lowercase, numbers, and special characters).
- After all validations are passed, Bob's account will be created in the database, with his information stored.
- Bob will then be redirected to the home page.

Acceptance Criterias:

- Bob needs to be able to visit the registration page.
- The information Bob has entered needs to be validated.
- If the form validation has failed:
    - No account should be created in the database.
    - The user should not be redirected to the home page.
    - There should be visual prompts telling the user what went wrong.
- If the validation has passed, the account needs to be created in the database.
- The user then needs to be redirected back to the home page.

Tests:

- The register page can be displayed.
  - ControllersTests.java > testShowRegisterPage
- The user can register after entering all the correct information.
  - ControllersTests.java > testUserRegister
- The user can be redirected to the home page after a successful registration.
  - ControllersTests.java > testUserRegister

Iteration: 1

Story Points: 2

Account - Login and Logout

Actors: Shane is a registered user who wants to sign into his account.

Preconditions:

- From the home page, Shane clicks the Sign In button on the top right, which leads him to the login page.
- Shane already has an account on the website, and his account information is being stored in the database.
- Shane enters his credentials and clicks the “Sign In” button.
- After his session, Shane wants to sign out, so he clicks on the “Sign Out” button.

Actions:

- The system will validate the form:
  - If a field is empty, Shane will be prompted to fill in what is missing.
  - If the “Email” field is not in a correct format (e.g. username is being used), Shane will be prompted to fix this.
- Then, it will search the database with the email and password combination entered.
  - If no matching account can be found, Shane will not be able to login and will remain on the same page.
- If the entered email and password combination is found, the login information will be stored in Shane’s session cookie.
- Then, Shane will be redirected to the home page.
- After clicking on the “Sign Out” button:
  - Shane’s session will be invalidated, and he will be redirected to the home page.

Acceptance Criterias:

- Shane needs to be able to visit the registration page.
- The information Shane has entered needs to be validated.
- If invalid credentials are entered:
- Shane needs to not be able to login and no session information should be stored.
- Shane needs to stay on the same login page.
- If the correct credentials are entered, Shane needs to be logged in, and his browser now needs to contain the correct session information.
- Shane then needs to be redirected back to the home page, and can see “Sign Out” on the top right replacing the “Sign In” button.

- After signing out, Shane must be redirected to the home page. With his session ended, the “Sign In” button will reappear on the navigation bar.

Tests:

- The login page can be displayed.
  - ControllersTests.java > testShowLoginPage
- The user can be redirected to the home page after logging out.
  - ControllersTests.java > testUserLogout

Iteration: 1

Story Points: 2

Account - Changing User Settings

// Including subscribing to mailing list

Iteration: 3

Story Points: 2

Account - Password Reset

Iteration: 3

Story Points: 3

Favorites - Add Property to Favorites

Actors: Sarah is a user looking for suitable properties to move into with her children.

Preconditions:

- She is browsing the website for potential homes, and she clicks “Add to Favorite” to bookmark some properties to her favorites list for easy access and comparison later.

Actions:

- If she is already a registered user and signed in, the bookmark button next to “Add to Favorites” will have the color yellow, and the property is added to the joined database that stores the user's unique favorite list. She can view her favorite properties displayed in the favorite page.
- If Sarah is not signed in, she is redirected to the login page, where she has the option to either login (if she already has an account) or register for a new account.

Acceptance Criterias:

- If the user is signed in:
  - The bookmark button should have the color yellow after the user clicked on it.
  - The bookmarked property will be saved in the joined database of the user with its correct information.
  - The properties the user has bookmarked on the listings page are displayed on their favorites page.
- If the user is not signed in:
  - The user will be redirected to the sign in page.
  - The attempted bookmarked property will not be added into the user’s joined database.
- Each users’ favorite list should be independent from each other.

Tests:

- The user can add property listings to their favorites.
    - ControllersTests.java > testAddToFavorites

Iteration: 2

Story Points: 2

Favorites - Remove Property from Favorites

Actors: Peter is a signed in user looking through his favorite properties.

Preconditions:

- He clicks on the highlighted “Add to Favorites” button in the property modal.

Actions:

- The bookmark button next to “Add to Favorites” will no longer have the color yellow.
- The property is removed from the favorites page for the user.

Acceptance Criterias:

- The bookmark button should be gray after the user clicks on it.
- The bookmarked property will be removed from the database.
- The favorite list will be unique between registered users.
- The properties the user had favorited on the listings page will now no longer be displayed on the favorites page.

Tests:

- The user can remove property listings to their favorites.
    - ControllersTests.java > testRemoveFromFavorites

Iteration: 2

Story Points: 1

Favorites - View Favorites

Actors: Sarah is a registered user who has a few properties in her favorite list. She wants to see the list now.

Preconditions:

- Sarah is registered and has a user account.
- Sarah has some properties already added on her favorite list.
- Sarah is able to navigate to the favorites page.

Actions:

- When Sarah visits the favorites page, if she is logged in, a list of properties will be displayed to her, based on her list of favorites.
- If she is not logged in, she will be redirected to the login page and need to login.

Acceptance Criterias:

- If the user is signed in:
    - The favorites page should be able to be accessed.
    - The user should be able to see a list of their favorites.
    - The user should be able to see details of a favorite entry by clicking on it.
- If the user is not signed in:
    - The favorites page should not be accessible.

    - The user will be redirected to the sign in page.
- Each users’ favorite list should be independent from each other.

Tests:

- The users list of favorites can be obtained.
    - ControllersTests.java > testGetFavorites

Iteration: 2

Story Points: 1

Admin - Login and Logout

Actors: John, an admin user of the site, wants to login to the site as an admin. After he is done with everything he will then logout.

Preconditions:

- From the homepage, John clicks the “Sign In” button on the top right. Then, he clicks the “Admin? Login Here” button, which redirects him to the admin login page.
- After filling in the form, he clicks the “Sign In” button.
- After completing what he needs to do, John then clicks the “Admin Logout” button to logout.

Actions:

- Validate the form to check if all fields are filled by searching the admin database to find the admin account by email and password entered.
- If an admin account is found:
    - Store login information in John’s session cookie, and John is redirected to a protected admin page.
- If the credentials do not match any in the database:
    - John will remain on the admin sign in page.
- When John clicks the “Admin Logout” button to logout, the session will be destroyed and he will be redirected to the home page.

Acceptance Criterias:

- The admin user needs to be able to visit the registration page.
- The information the admin user has entered needs to be validated.
- If invalid credentials are entered:
    - The admin user needs to not be able to login and no session information should be stored.
    - The admin user needs to stay on the same login page.
- If the correct credentials are entered, the admin user needs to be logged in, and his browser now needs to contain the correct session information.
- When John clicks the “Admin Logout” button to logout, the session will be destroyed and he will log out.

Tests:

- The admin login page can be displayed.
    - ControllersTests.java > testShowAdminLoginPage
- If the admin user is logged in, the protected admin page can be displayed instead.
    - ControllersTests.java > testShowAdminLoginPageLoggedIn

Iteration: 2
Story Points: 2

Admin - View User List

Iteration: 3
Story Points: 1

Admin - View Property List

Iteration: 3
Story Points: 1

Admin - Create New Property Entry

Iteration: 3
Story Points: 1

Admin - Edit Details of Property Entry

// Not yet implemented
Iteration: 3
Story Points: 0

Admin - Delete Property Entry

Iteration: 3
Story Points: 1

Admin - Edit Images of Property Entry

// Including adding and deleting images
Iteration: 3
Story Points: 1

Admin - Send Mail to Mailing List

Iteration: 3
Story Points: 3

Mortgage Calculator

Iteration: 3
Story Points: 1

Inactive Popup

Iteration: 3
Story Points: 1

Velocity Report

Iteration 1

In iteration 1, feature wise, we have completed setting up the website, the property listings page, and the functionality to register, login, and logout for regular users. This adds up to 7 story points in total, which is equal to 3.5 points per week.

Iteration 2

In iteration 2, we have completed displaying the detailed view of property listings, the functionality to filter properties by attributes, the admin user system, the favorites system. This adds up to 10 story points in total, which is equal to 5 points per week. We have also implemented the unit test system, and the properties database to store property information.

Generally speaking, we are getting more things done in iteration 2, compared to the previous iteration. The difference in amount of work done is likely even bigger than the story points calculation entails. Since some work is improving the features done in iteration 1, like adding a check to password security, or actually implementing the table to store property data in our database. We also worked on getting unit tests implemented for both features in iteration 1 and 2.

Iteration 3

In iteration 3, we haven't done anything because the funding went dry and the project is scrapped!

Dividing Work

In iteration 1, a problem we had was dividing the work into “frontend” and “backend”. This turned out to be causing a lot of troubles, since the person who worked on the “backend” worked on creating endpoints for GET and POST requests, but there isn't any UI to actually test if those endpoints work.

Meanwhile, the person working on “frontend” needed those endpoints to actually read and write data from the database, and without them, the UI is just talking to imaginary endpoints, and cannot be tested at all. As a result, the last few days of iteration 1 is very heavily loaded, because a large amount of time is being spent on connecting the frontend with the backend, which often means doing pretty big rewrites of parts that need the UI to interact with the database. In iteration 2, we have shifted towards assigning work to each person that is more “feature” based. When allowing each person to work on a new “feature” that includes both part of UI and database interactions, work being done can be tested by the person working on it to make sure it works, before being merged into the main branch. This way, the main branch always has a copy of the website that is working.

User Interface Requirements

Home Page

// Add details on featured properties and explore listing button
</>

As described in the user stories, the first page the user sees when visiting the site should be the home page. The nav bar on top of the screen has some buttons on the right that can be clicked, including the information of the website owner, the link to jump to property listing view, and login or logout button.

Navigation Bar

For the user page, on the top left of the nav bar, the website logo is displayed. This is also a button that you can click to go back to the home page. The top right corner displays the login button, but if the user is already logged in, it will display the logout button instead.

When clicking the “About Us” button on the nav bar, a detailed view will drop down from the top, while dimming everything else on the website. The view will contain information about the client, who is the website owner. This includes the client’s photo, their contact information, a short self introduction, and links to their social media pages.

On the top right, there are also the “Listing”, “Mortgage Calculator”, and “Favorites” buttons, which lead the user to the corresponding pages.

For the admin pages, the navigation bar includes a logout option located at the top-right corner, allowing the admin to securely exit their session. After logging out, it takes the admin to the home page. On the top right, there are also the “Manage Listing” and “View Users” buttons, which leads to the corresponding pages.

Property Listing Page

// Add info on property details, and adding properties to favorites
</properties>
After clicking on either “Listing” or “Explore Our Listing”, the user will be directed to the property listing page that will display different properties with information such as price, location, and number of baths and bedrooms.

The pagination, located at the bottom of the property listing, allows users to navigate through different pages of properties. This way, the user will not have to scroll forever; instead, the user can easily jump to different sections and find what they are looking for faster.

The property listing page has a filter and sorting bar on top, letting users narrow down their search for an ideal property. The bar prompts users to refine their search through options like specifying a city, sorting properties by price, and selecting the desired number of bedrooms and bathrooms. There's also a spot to search for a house by its name. Once the user picks what they want, just hit the "Apply Filters" button to the filtered results.

Account - Login Page

</login>

Visiting the login page, the user can see a form where they can enter the email and password associated with their account to login. There is also a link prompting the user to register an account if they do not have one. Clicking it will redirect them to the register page.

Below the login form, there is a button for leading the user to the admin login portal. There is also a "Forgot Password" button that leads the user to the corresponding page.

Account - Register Page

</register>
On the register page, there is a form, which includes the username, email, password, and a password confirmation slot. After entering those details, the user can click the register button to create the account. After so, the user will be redirected back to the home page.

Account - Forgot Password Page

</forgotpassword>

Account - Password Reset Page

// The page users see when they click on the email they received
</resetpassword>

Account - User Settings Page

</settings>

Favorites Page

</favorites>
Properties that the user has favorited are displayed on the page if they are logged in. A user who is not logged in cannot access the favorites page and is redirected to the login page. If a user wants to remove a property from their favorites they have to unhighlight the bookmark on the listing and the property is then removed from the user's favorites page.

Admin - Home Page

// Add details of home page, including buttons
</admin>
Admin page is designed by keeping the security in mind, it requires admin login with email and password to access the protected admin page. Once authenticated, the admin is directed to a protected admin area, ensuring that sensitive operations are kept secure.

Admin - Login Page

</admin/login>

Admin - Manage Property Listings Page

// Include image editing

</admin/properties>
</admin/images>

Admin - User List and Mailing List Page

</admin/users>

Mortgage Calculator Page

</mortgage>

Inactive Popup

User Interface Mockups

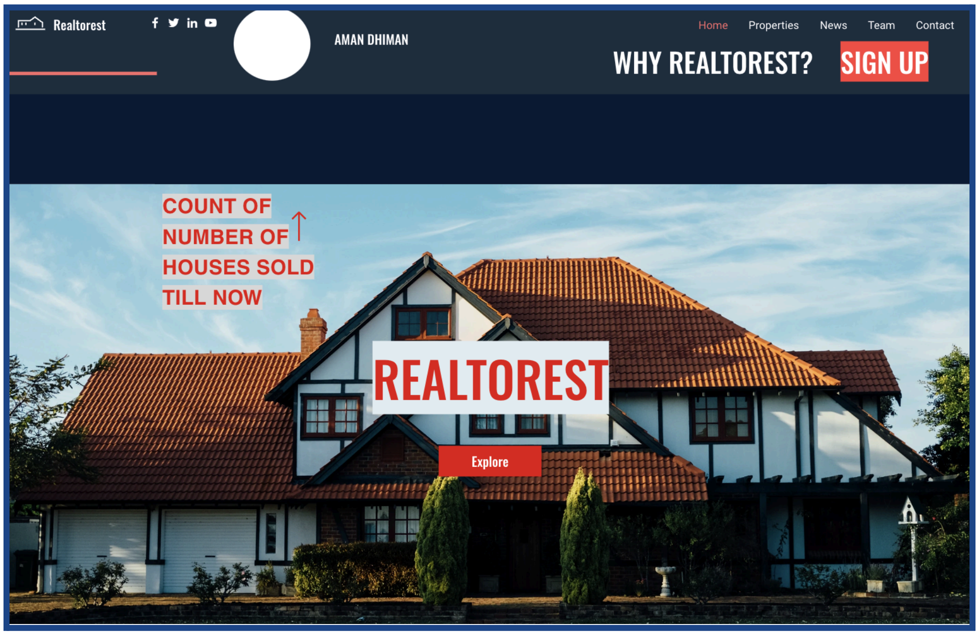
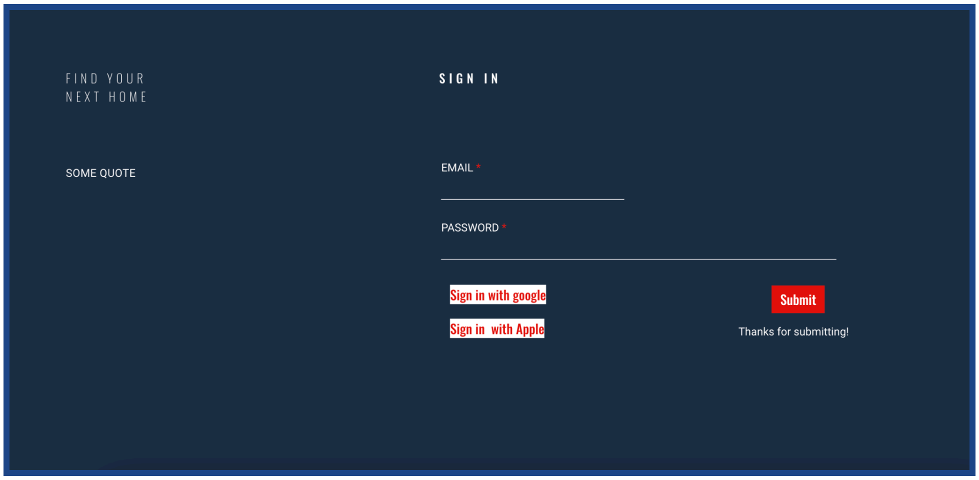


Figure 1. Home Page



Thanks for submitting!

Figure 2. Login Page

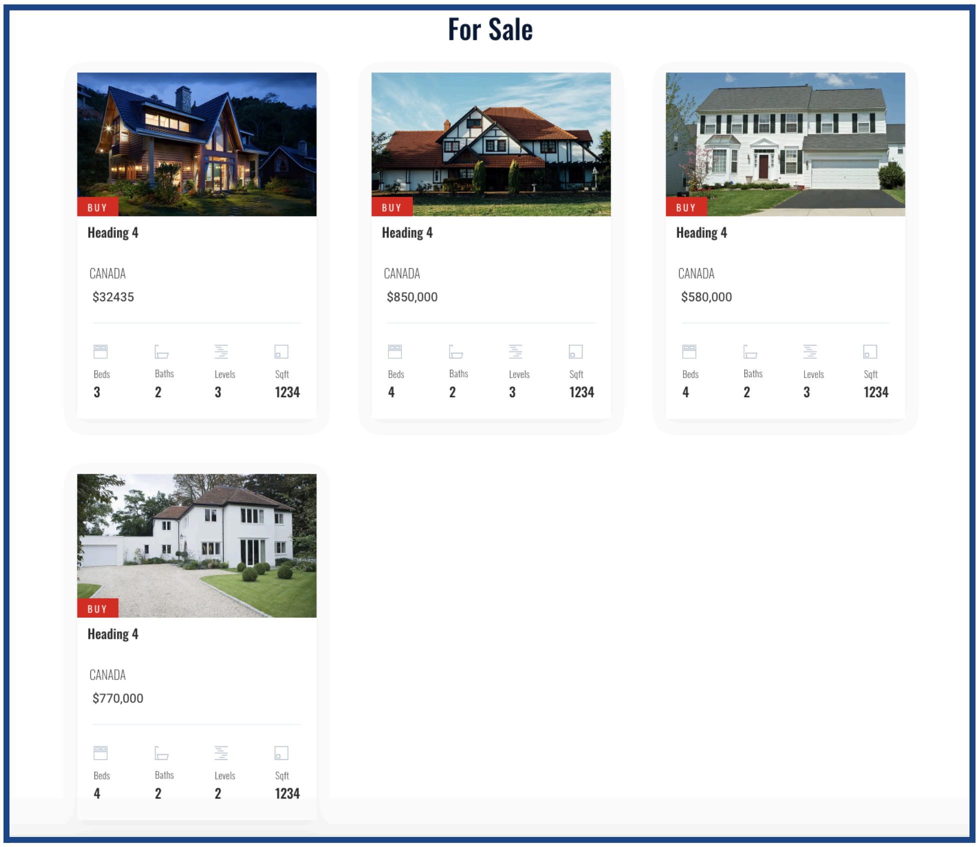


Figure 3. Properties Listing View