IT5007 Software Engineering on Application Architecture

Course Project Report

Property Rental Platform

Li Yueling (A0254056H/ e0954457@u.nus.edu)

Liao Yueh Fan (A0275718R / e1127372@u.nus.edu)

Jiang Xindou(A0268277M/ e1101576@u.nus.edu)

Contents

1. Project Background

The project is to develop a property rental platform to address the evolving needs of the housing rental market in Singapore, ensuring its relevance and effectiveness not only in the present but also in the foreseeable future.

Location significantly influences the choices of potential tenants, and yet, many existing property rental applications fail to provide an intuitive and efficient geographical location search feature.

A Map-based Property Search is introduced to complement the above situation. By integrating a map-based search feature into our rental platform, we intend to enhance the user experience, providing a visual and interactive tool for users to explore properties in their preferred locations.

At the same time, it's essential to consider safeguarding our unique solution from potential copycats. To protect our innovation, we plan to implement comprehensive terms of service and user agreements on our platform. These agreements will incorporate clauses that explicitly prohibit users from copying, reverse engineering, or using our platform for unauthorized purposes. Additionally, we recognize the importance of consulting with intellectual property attorneys who specialize in technology and software. They will offer invaluable guidance on the most suitable legal protections for our project, including patent filings, copyright registrations, and trademark applications.

2. System Design

1.1 Use-Cases

1.1.1 Use Case 1: Tenant Authentication

Description

The entrance of the tenant Log In and Sign Up. Using Email to check whether the user is already registered.

Actors:

Unregistered tenants, registered tenants

Flow:

1. In the "Tenant Welcome" mode, tenants input their email addresses for authentication.
2. The system validates the email address, ensuring it is a valid email format.
3. Tenants submit the form.
4. The system checks if the entered email is associated with an existing tenant account.
5. If the email is found, the system stores their password for login check and transitions to "Tenant Login" mode.
6. If the email is not found, the system transitions to "Register" mode for the tenant to create a new account.

7. Tenants can click "Agent" button to switch to the "Agent Welcome" mode.

1.1.2 Use Case 2: Agent Authentication

Description

The entrance of agent Log In. Using Email to check whether the user is already registered.

Actors:

Registered agents

Flow:

1. In the "Agent Welcome" mode, agents input their email addresses for authentication.
2. The system validates the email address, ensuring it is a valid email format.
3. Agents submit the form.
4. The system checks if the entered email is associated with an existing agent account.
5. If the email is found, the system stores their password for login check and transitions to "Agent Login" mode.
6. If the email is not found, the system alerts the agent to contact staff for registration.
7. Agents can click "Tenant" button to switch to the "Tenant Welcome" mode.

1.1.3 Use Case 3: Tenant Login

Description

Tenants can log in using their email address and password.

Actors:

Registered tenants

Flow:

1. In the "Tenant Login" mode, tenants input their password.
2. The system validates the password, ensuring it is not empty.
3. Tenants submit the form.
4. The system checks if the entered password matches the stored password for the email.
5. If the password is correct, the system logs the tenant in and navigates to the main application page.
6. If the password is incorrect, an alert is displayed.

1.1.4 Use Case 4: Agent Login

Description

Agents can log in using their email address and password.

Actors:

Registered agents

Flow:

1. In the "Agent Login" mode, agents input their password.
2. The system validates the password, ensuring it is not empty.
3. Agents submit the form.
4. The system checks if the entered password matches the stored password for the email.
5. If the password is correct, the system logs the agent in and navigates to the main application page.
6. If the password is incorrect, an alert is displayed.

1.1.5 Use Case 5: Tenant Sign Up

Description

Tenants can register for a new account using their email address and password.

Actors:

Unregistered tenants

Flow:

1. In the "Register" mode, tenants input a username, a password and confirm it.
2. The system validates username input, ensuring the password length is not shorter than 4 and not longer than 10.
3. The system validates both passwords, ensuring the password length is not shorter than 4 and not longer than 20.
4. The system ensures that the passwords match.
5. Tenants submit the form.
6. The system collects the form data and prepares it for an API call.
7. The system sends tenant data to the server to add a new tenant account to the tenant collection with a username, email, password, and newly assigned ID.
8. The system logs the tenant in and navigates to the main application page.

1.1.6 Use Case 6: User Log Out

Description

Users can log out of the system.

Actors:

Logged in tenants/ agents

Flow:

1. Tenants or Agents can select the “Log Out” option from the dropdown menu, whose entrance is the account username button.
2. The user is logged out and returned to the main application page.

1.1.7 Use Case 7: Navigating to Different User Service Pages

Description

After the user log in, their account username will displayed instead of the log-in button. The account button also provides the entrance of the private links, which are “Favorites”, “History” and “My Profile” for tenants, as well as “Add Property”, “My Posts” and “My Profile” for agents. Users can navigate to different pages within the application by clicking on various options available within the account dropdown.

Actors:

Logged in tenants/ agents

Flow:

1. The system checks the user's login status.
2. If the user is authenticated, the system displays an account dropdown button.
3. If the user has not logged in yet, the system displays a "Log in/ Sign up" button.
4. Logged users can interact with the account dropdown to access features, while unlogged users can click "Log in/ Sign up" to navigate to the login page.
5. Users open the account dropdown.
6. They select from the available options based on their user type.
7. The system navigates the user to the selected page, providing access to features such as viewing profile, favorites, history, all posts, or posting new content.

1.1.8 Use Case 8: Displaying Tenant Favorites List

Description

Tenants can view a list of housing data that they clicked “Like” before.

Actors:

Logged in tenants

Flow:

1. Tenants can enter this page by clicking the “Favorites” option in the dropdown menu.
2. The system fetches the tenant’s favorites list from the back-end database, in tenants collection.
3. Tenants can view all available records on the page.
4. The rest of the page control is the same as map-based house searching.

1.1.9 Use Case 9: Displaying Favorites in Table

Description

Tenants can view a tabular representation of housing data that they clicked “Like” before. This function is designed to allow tenants to better compare different properties and help them to make a decision.

Actors:

Logged in tenants

Flow:

1. Open the toggle switch on the default “favorites” page to enter the tabular comparison mode.
2. The system fetches the tenant’s favorites list from the back-end database, in tenants collection.
3. The data is rendered in a tabular format.
4. Tenants can view all available records on the page.
5. Tenants can click the “heart” button to change Like/ Dislike status. The property that the tenant has liked will automatically added to the history list.
6. After refreshing the component, the system will pull the latest data and display the latest list of favorites.
7. The system displays pagination controls, including the previous and next buttons. Users can click the previous and next buttons to navigate between pages of data.
8. Tenants can also specify a page number to jump to a specific page.
9. Tenants can adjust the number of items displayed per page.
10. The system collects the updated favorites list and history list on change of Like/ Dislike status. Then the system prepares them for an API call.
11. The system sends the new favorites list and history list to the server to update the tenant collection.

1.1.10 Use Case 10: Displaying Viewed History in Table

Description

Tenants can view a list of housing data that they checked for details before.

Actors:

Logged in tenants

Flow:

1. Tenants can enter this page by clicking the “History” option in the dropdown menu.
2. The system fetches the tenant’s viewed history list from the back-end database, in tenants collection.

3. Tenants can view all available records on the page.
4. Tenants can click the “heart” button to change Like/ Dislike status. The property that the tenant has liked will automatically added to the history list.
5. After component refreshment, the newest favorites list will show on the page.
6. The rest of the page control is the same as Use Case 9: Displaying Favorites in Table.
7. The system collects the updated favorites list and history list on change of Like/ Dislike status. Then the system prepares them for an API call.
8. The system sends the new favorites list and history list to the server to update the tenant collection.

1.1.11 Use Case 11: Add Property

Description

Agents can post a new property, by submitting a new housing listing.

Actors:

Logged in agents

Flow:

1. Agents can enter this page by clicking the “Add Property” option in the dropdown menu.
2. Agents input details in the form fields, including the type, bedroom count, bathroom count, price, area, address, and postal code.
3. The system validates the input, ensuring that the data meets the specified requirements, error message will display if the input field does not meet the validation criteria.
4. Users submit the form.
5. The system collects the form data and prepares it for an API call.
6. The system sends the new listing data, including the agent's id, to the server to add the listing to the database.
7. After successful submission to the back-end, system will navigate to the Use Case 12: Displaying All Posts in Table Page.

1.1.12 Use Case 12: Displaying All Posts in Table

Description

Agents can view a list of housing data that they posted before.

Actors:

Logged in agents

Flow:

1. Agents can enter this page by clicking the “My Posts” option in the dropdown menu.
2. The system fetches the agent’s posted list from the back-end database.
3. Agents can view all available records on the page.
4. Agents can find the entrance of property edition and deletion on this page.
5. The rest of the page control is the same as Use Case 9: Displaying Favorites in Table.
6. After edition or deletion, the system will pull the latest data and display the latest list of properties.

1.1.13 Use Case 13: Edit Property Data

Description

Agents can edit the details of the property.

Actors:

Logged in agents

Flow:

1. Agents can enter this page by clicking the “My Posts” option in the dropdown menu.
2. In the penultimate column of each row, namely each property, there is an edit icon.
3. Agents can check the details by clicking the edit icon. Before agents click “Edit” button, the input fields are read-only.
4. After agents click “Edit” button, the input fields change into editable mode.
5. Validation is performed on all input fields as in the registration section.
6. Agents can cancel the edition by clicking on “Cancel”.
7. Users can confirm the changes by clicking on “Update Data”.
8. The system collects the form data and prepares it for an API call.
9. The system sends the updated data to the server to update the corresponding information of the property in `properties` collection according to id.

1.1.14 Use Case 14: Delete Property

Description

Agents can delete properties they are responsible for.

Actors:

Logged in agents

Flow:

1. Agents can enter this page by clicking the “My Posts” option in the dropdown menu.
2. In the last column of each row, namely each property, there is an delete icon.
3. Agents can click the delete icon to delete specific property.
4. A pop-up modal will show up to double check if agents really want to delete that property.
5. Agents can cancel it by clicking on “Cancel”.
6. Agents can confirm the delete by clicking on “Delete”.
7. The system collects property id and prepares it for an API call.
8. The system sends the id to the server to delete the property in `properties` collection according to id.

1.1.15 Use Case 15: Show and Edit User Profile

Description

Agents and Tenants can check and edit their profile.

Actors:

Logged in tenants/ agents

Flow:

1. Logged in users can enter this page by clicking the “My Profile” option in the dropdown menu.
2. Users check the profile information in the page. Before user click “Edit” button, the input fields are read-only.
3. After users click “Edit” button, the input fields change into editable mode.
4. Validation is performed on all input fields as in the registration section.
5. Users can cancel the edition by clicking on “Cancel”.
6. Users can confirm the changes by clicking on “Update Profile”.
7. The system collects the form data and prepares it for an API call.
8. The system sends the updated profile to the server to update the corresponding information of the agent/ tenant in `agents`/ `tenants` collection according to id.

9. After successful submission to back-end, the system will pull the latest data and display on the page.

1.1.16 Use Case 16: De-registration

Description

Tenants can delete their account.

Actors:

Logged in tenants

Flow:

1. Logged in tenants can access this feature by clicking the “My Profile” option in the dropdown menu..
2. Tenants can click “Delete Account” button to de-register.
3. A pop-up modal will show up to double check if tenants really want to delete their account.
4. Tenants can cancel the de-registration by clicking on “Cancel”.
5. Users can confirm the de-registration by clicking on “Delete”.
6. The system collects tenant id and prepares it for an API call.
7. The system sends the id to the server to delete the tenant in tenants collection according to id.

1.1.17 Use Case 17: Map-Based Search: Show All Properties

Description

The filter button “All Properties” on the left side can show users all properties list.

Actors:

All users

Flow:

1. Users can click on the filter button “All Properties” on the left side.
2. The left part of the page shows all properties' brief information list.
3. The map shows all properties’ markers.

1.1.18 Use Case 18: Map-Based Search: Show Favorite Properties

Description

The filter button “Favorite Properties” on the left side can show users their favorite properties list.

Actors:

Logged in tenants

Flow:

1. Users can click on the filter button “Favorite Properties” on the left side.
2. The left part of the page shows all their favorite properties' brief information list.
3. The map shows all their favorite properties’ markers in pink color.

1.1.19 Use Case 19: Map-Based Search: Show Viewed Properties

Description

The filter button “Viewed History” on the left side can show users their view history list.

Actors:

Logged in tenants

Flow:

1. Users can click on the filter button “Viewed History” on the left side.

2. The left part of the page shows all the viewed history properties' brief information list.

3. The map shows all the viewed history properties' markers.

1.1.20 Use Case 20: Map-Based Search: Like the Property

Description

Users can add property to their favorite properties list.

Actors:

Logged in tenants

Flow:

1. Users can click on the like button shown in the house's brief or detailed information.
2. The system will add this property to users' favorite list.

1.1.21 Use Case 21: Map-Based Search: Share the Property

Description

Users can share the property's information on social media platforms.

Actors:

All users

Flow:

1. Users can click on the share button shown in the house's brief or detailed information.
2. The popout window redirects users to Facebook post with a "SingaporeHouseRental" hashtag. (This is a local project, thus the url is a dummy one)

1.1.22 Use Case 22: Map-Based Search: View Detail Information

Description

Users can view detailed information about the properties.

Actors:

All users

Flow:

1. Users can click on the "Show Detail Information" button shown in the house's brief information.
2. The left part of the website displays detailed information about the house.

1.1.23 Use Case 23: Map-Based Search: Show Property on the Map

Description

Users can view the exact place of the housing list on the map.

Actors:

All users

Flow:

1. Users can click on the "Show on Map" button shown in the house's brief information.
2. The map points out the location of the house by using orange markers on the map.

1.1.24 Use Case 24: Map-Based Search: Mouseover the Marker

Description

Users can get an overview of the property intuitively.

Actors:

All users

Flow:

1. Users can put their mouse on the markers that are shown on the map.
2. There is a popup window displaying the title, house type, price, bedroom number, and bathroom number of the house.

1.1.25 Use Case 25: Map-Based Search: Click the Marker

Description

Users can get detailed information about the property intuitively.

Actors:

All users

Flow:

1. Users can click on the markers that are shown on the map.
2. The left part of the website displays all the detailed information about the house.

1.1.26 Use Case 26: Landing Page: Search Property by Address

Description

Users can search properties by address key words.

Actors:

All users

Flow:

1. User can input key words for the address interested.
2. Click the button 'Search', all related properties will be displayed below.

1.1.27 User Case 27: Landing Page: Search Property by Type

Description

Users can search properties by property type such as 'HDB' or 'Condo'.

Actors:

All users

Flow:

1. User can select the type of property interested.
2. On change of the selection field, all properties of the selected type will be displayed below.

1.1.28 User Case 28: Landing Page: Search Property by Room Number

Description

Users can get search properties by number of bedrooms interested.

Actors:

All users

Flow:

1. User can select the number of bedrooms they look for in a property.
2. On change of the selection field, all properties with the selected bedrooms number will be displayed below.

1.1.29 User Case 29: Landing Page: Search Property with Maximum Price

Description

Users can get search properties by limiting the maximum price.

Actors:

All users

Flow:

1. User can input the maximum price for properties they search for.
2. Click on the button right next to the input field, all properties with price smaller than or equal to the maximum price be displayed below.

1.2 Database Design

1.2.1 Properties Collection

- id: Primary key of the housing dataset, unique identifier.
- display_address: The address of the property that agent would like to display.
- street_address: The actual address of the property.
- price: Price of the property.
- type: Type of the property (e.g., HDB, Condo).
- bedrooms: Number of bedrooms in the property.
- bathrooms: Number of bathrooms in the property.
- area: Total area of the property.
- postal_code: Postal code of the property.
- manager_id: The id of the agent who is responsible for this property
- lat: The latitude of the house for rendering marker on google map
- lng: The longitude of the house for rendering marker on google map

1.2.2 Tenants Collection

- id: Primary key of the user dataset, unique identifier.
- name: The user name.
- email: The email address of the user.
- password: The password of the user.
- favorites: The list that stores ids of the properties that the user clicked the “Like” button.
- history: The list that stores ids of the properties that the user clicked to view details.

1.2.3 Agents Collection

- id: Primary key of the user dataset, unique identifier.
- name: The user name.
- email: The email address of the user.
- password: The password of the user.
- properties: The list that stores ids of the properties that the agent responsible for.

1.2.4 Authentication Data (Private, only used in authentication code, not stored)

- id: The unique identifier of the user.
- email: The email address of the user.
- name: The user name.
- asTenant: A Boolean flag indicating whether the user is a tenant.
- isAuthenticated: A Boolean flag indicating the user’s authentication status.

1.3 User Interface Design

1.3.1 Log In/ Sign Up Welcome Interface

Actors:

Unregistered tenants, registered tenants and registered agents

Description:

- The entrance of Log In and Sign Up for tenant users, and Log In for existing agent users.
- An input field to receive their email to identify whether they are registered or not.
- After the “Continue” button is clicked, error messages are displayed if there are input field validation issues.
- They can click “Agent” or “Tenant” to switch between tenant mode and agent mode.
- The sketch figure of the interface design is shown as below.

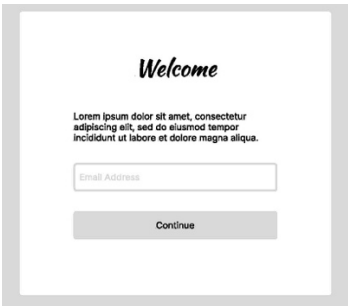

Figure 1 Log In/ Sign Up Welcome Interface

1.3.2 Log In Interface

Actors:

Registered tenants and registered agents

Description:

- An input field to receive the password input to match with the database.
- After the “Log In” button is clicked, error messages are displayed if there are input field validation issues.
- Users can navigate back to the email input stage.
- Users will be navigated to the homepage after successful login.
- The sketch figure of the interface design is shown as below.

Figure 2 Log In Interface

1.3.3 Register Interface

Actors:

Unregistered tenants.

Description:

- Tenants can register for a new account by setting a password.
- Three input fields to receive the username, password and confirm the password.
- After clicking the "Register and Login" button, if there is an input field validation problem (including password length, empty password field, and input field match), an error message will be displayed.
- Users can navigate back to the email input stage.
- Users will be navigated to the homepage after successful registration.
- The sketch figure of the interface design is shown as below.

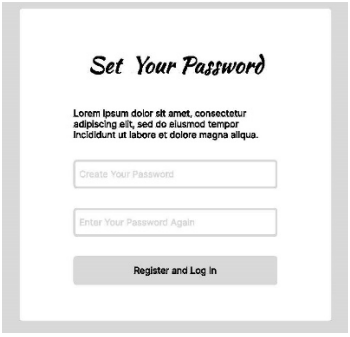

Figure 3 Register Interface

1.3.4 Show and Edit Profile Interface

Actors:

Logged in tenants/ agents.

Description:

- Users check the profile information in the page. Before user click “Edit” button, the input fields are read-only.
- After users click “Edit” button, the input fields change into editable mode.
- Validation is performed on all input fields as in the registration section.
- Users can cancel the edition by clicking on “Cancel”.
- Users can confirm the changes by clicking on “Update Profile”.
- The sketch figure of the interface design is shown as below.

Figure 4 Edit Profile Interface (right: agent profile, middle: tenant profile, left: edition)

1.3.5 De-registration Interface

Actors:

Logged in tenants.

Description:

- Tenants can click “Delete Account” button to de-register.

- A pop-up modal will show up to double check if tenants really want to delete their account.
- Tenants can cancel the de-registration by clicking on “Cancel”.
- Users can confirm the de-registration by clicking on “Delete”.
- The sketch figure of the interface design is shown as below.

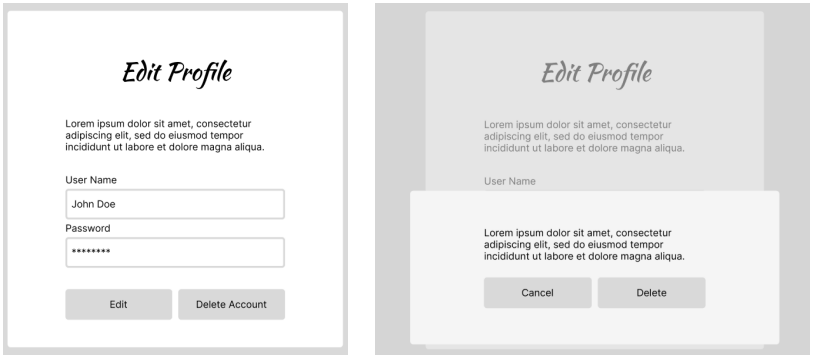

Figure 5 Delete Account Interface

1.3.6 Navigation Dropdown

Actors:

Logged in tenants and agents

Description:

- For authenticated users, the account button displays an account dropdown.
- For unauthenticated users, the account button displays a "Log In" button, which directs users to the login page.
- The account dropdown appears when an authenticated user clicks the account button. The dropdown displays the user's username as a button.
- The dropdown menu offers different options based on the user's type. For tenants, options include "My Favorites" and "History". For agents, options include "My Posts" and "Add Property".
- The “My Profile” and "Log Out" option are available for both user types within the dropdown menu.

1.3.7 Favorites Display Interface

Actors:

Logged in tenants

Description:

- Logged-in tenants can enter this page by clicking the corresponding option in the account dropdown menu.
- A toggle switch is designed to switch between default listing mode and tabular comparison mode.
- The housing data is presented in a way that is similar to the map-based house searching page.
- The sketch figure of the interface design is shown as below.

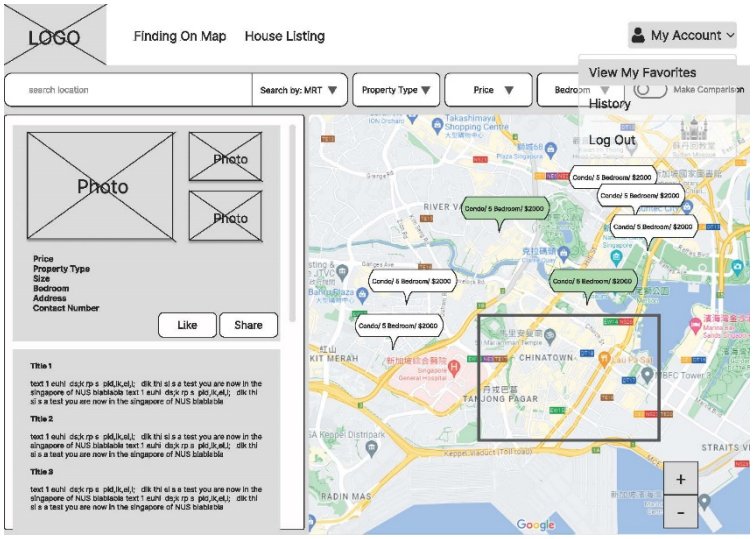

Figure 6 Favorites Display Interface

1.3.8 Favorites Comparison Interface

Actors:

Logged in tenants

Description:

- Logged-in tenants can enter this page by clicking the corresponding option in the account dropdown menu.
- A toggle switch is designed to switch between default listing mode and tabular comparison mode.
- The housing data is presented in a tabular format.
- Users can navigate between pages of data using pagination controls:
  > "Previous" and "Next" buttons allow users to move to the previous or next page.
  > A page number input field lets users jump to a specific page.
  > A dropdown allows users to change the number of items displayed per page.
- Users can change the like/ dislike status by clicking “heart” icon.
- The sketch figure of the interface design is shown as below.

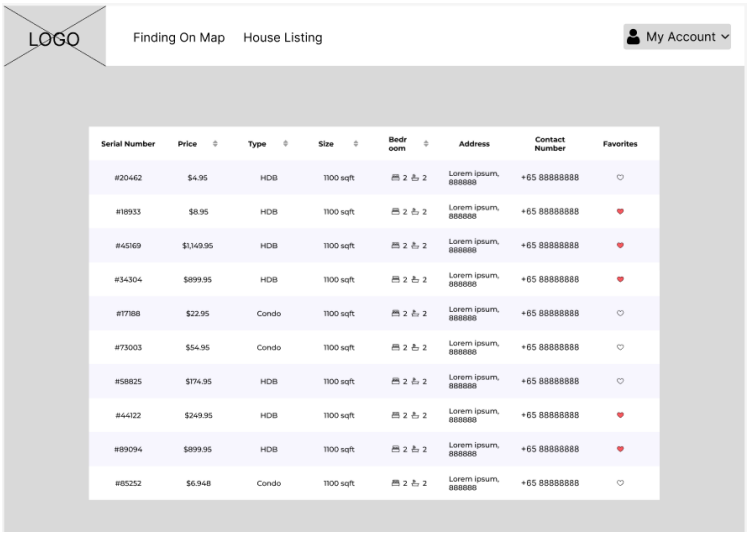

Figure 7 Favorites Comparison Interface

1.3.9 View History Interface

Actors:

Logged in tenants

Description:

- Logged-in tenants can enter this page by clicking the corresponding option in the account dropdown menu.
- The housing data is presented in a way that is similar to the Favorites Comparison Interface.
- Users can navigate between pages of data using pagination controls:
  > "Previous" and "Next" buttons allow users to move to the previous or next page.
  > A page number input field lets users jump to a specific page.
  > A dropdown allows users to change the number of items displayed per page.
- Users can change the like/ dislike status by clicking “heart” icon.
- The sketch figure of the interface design is shown as below.

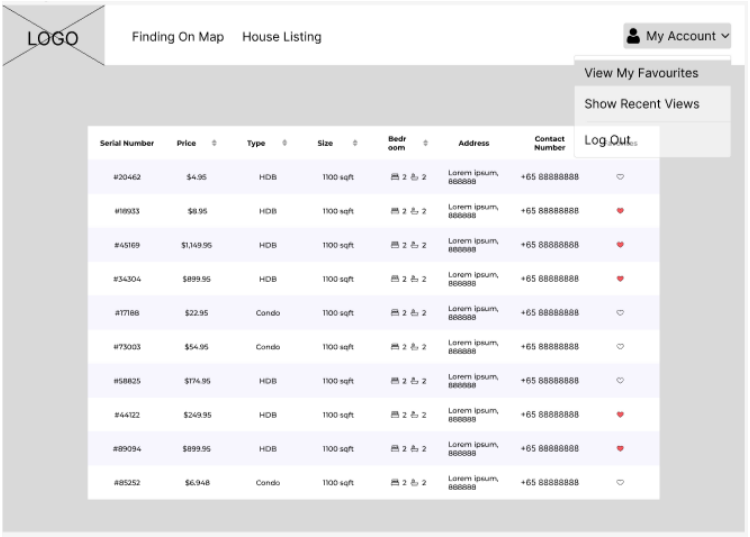

Figure 8 View History Interface

1.3.10 My Posts Interface

Actors:

Logged in agents

Description:

- Logged-in agents can enter this page by clicking the corresponding option in the account dropdown menu.
- The housing data is presented in a way that is similar to the Favorites Comparison Interface.
- Agents can edit or delete property by clicking on the corresponding icon.
- Users can navigate between pages of data using pagination controls:
  > "Previous" and "Next" buttons allow users to move to the previous or next page.
  > A page number input field lets users jump to a specific page.
  > A dropdown allows users to change the number of items displayed per page.
- Users can change the like/ dislike status by clicking “heart” icon.
- The sketch figure of the interface design is shown as below.

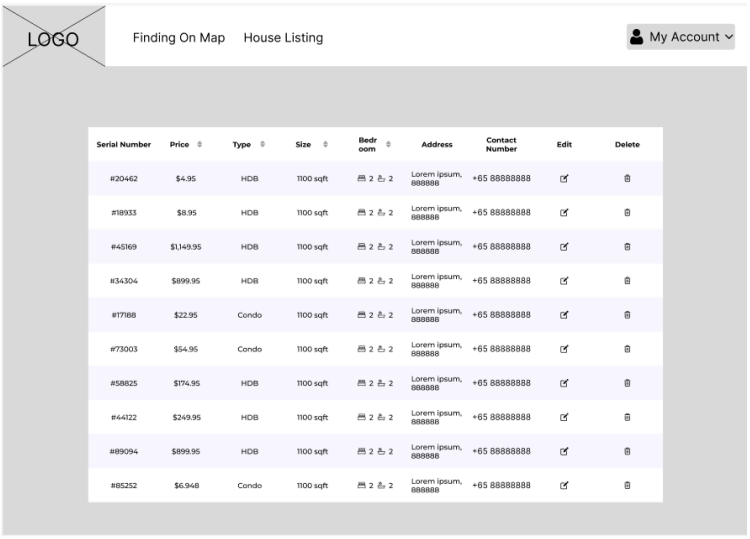

Figure 9 My Posts Interface

1.3.11 Delete Property Interface

Actors:

Logged in agents

Description:

- In the last column of each row, namely each property, there is a delete icon.
- Agents can click the delete icon to delete specific property.
- A pop-up modal will show up to double check if agents really want to delete that property.
- Agents can cancel it by clicking on “Cancel”.
- Agents can confirm the delete by clicking on “Delete”.
- The sketch figure of the interface design is shown as below.

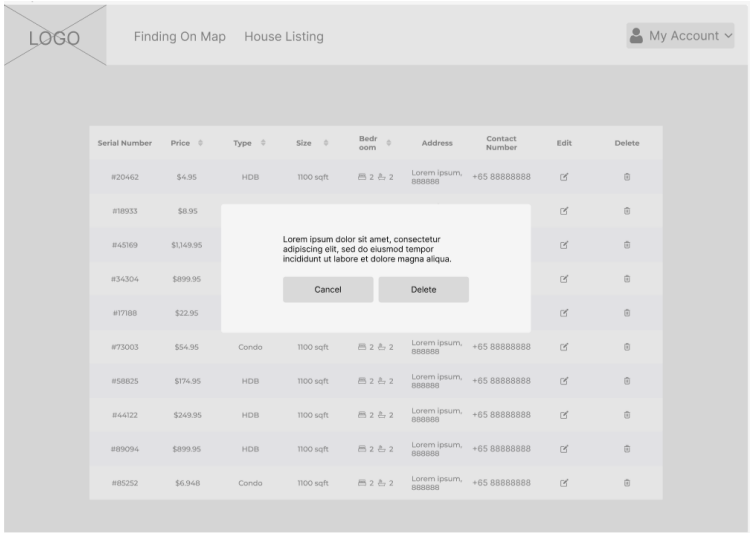

Figure 10 Delete Property Interface

1.3.12 Show and Edit Property Interface

Actors:

Logged in agents

Description:

- In the penultimate column of each row, namely each property, there is an edit icon.

- Agents can check the details by clicking the edit icon. Before agents click “Edit” button, the input fields are read-only.
- After agents click “Edit” button, the input fields change into editable mode.
- Validation is performed on all input fields as in the registration section.
- Agents can cancel the edition by clicking on “Cancel”.
- Users can confirm the changes by clicking on “Update Data”.
- The sketch figure of the interface design is shown as below.

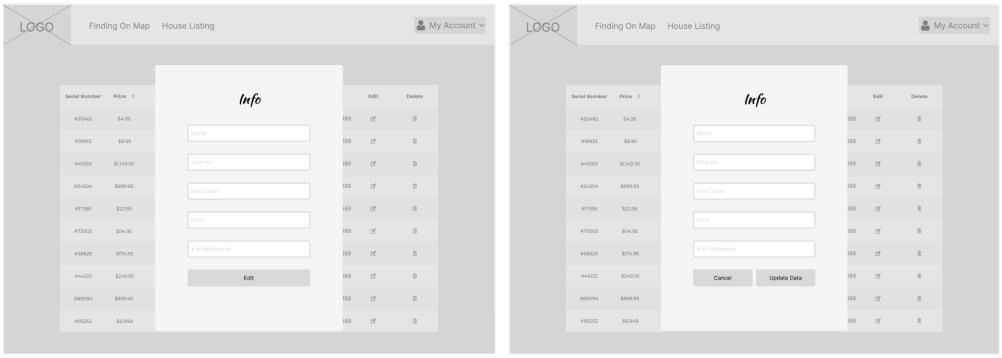

Figure 11 Show and Edit Property Interface

1.3.13 Add Property Interface

Actors:

Logged in agents

Description:

- Logged-in agents can enter this page by clicking the corresponding option in the account dropdown menu.
- Agents can input various details related to the housing listing.
- Placeholders offer guidance on the expected data format.
- A dropdown menu is provided for selecting the type of property. Users can choose from "HDB" or "Condo."
- Error messages are displayed below each input field if the user's input doesn't meet the specified validation criteria.
- A "Submit" button allows users to submit the form after inputting the required details.
- The sketch figure of the interface design is shown in Figure.

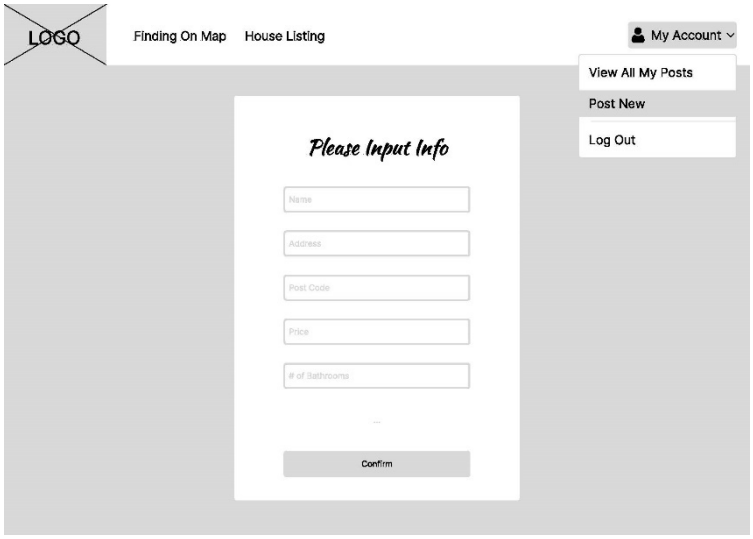

Figure 12 Add Property Interface

1.3.14 Map-Based Search: House Brief Information

Actors:

All users

Description:

- There are three filter buttons including all properties, my favorite, and viewed history on the left side of the page.
- The middle part of the page displays the properties' brief information
- There is one photo, title, house type, price, size, and bedroom number in the properties' brief information.
- There are four buttons including like, share, show on map, and show detail information in the properties' brief information.
- There is a map showing on the right side of the page containing rental house markers.
- The sketch figure of the interface design is shown in Figure 29.

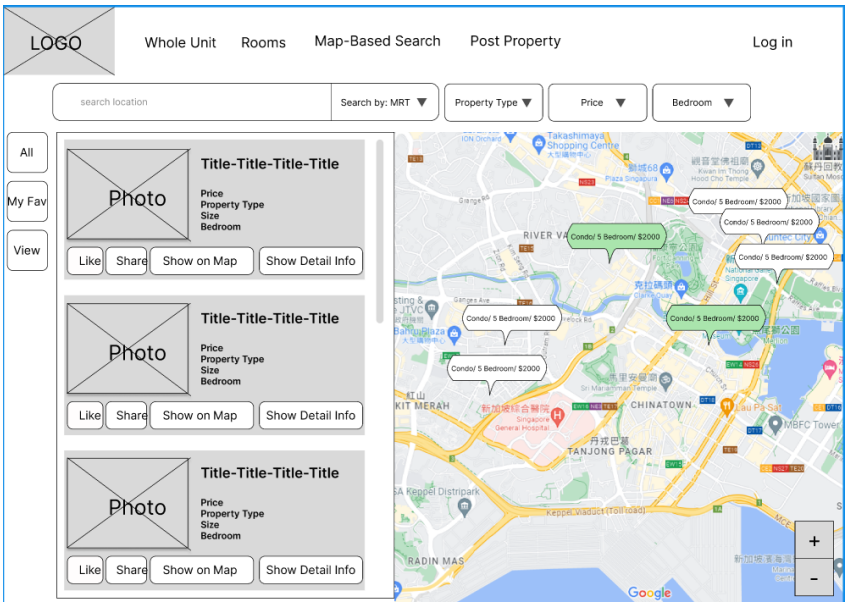

Figure 29 Map-Based Search: House Brief Information

1.3.15 Map-Based Search: House Detail Information

Actors:

All users

Description:

- There are three filter buttons including all properties, my favorite, and viewed history on the left side of the page.
  - The middle part of the page displays the properties' detailed information
  - There are photos, title, house type, price, size, bedroom number, address, contact number, descriptions in the properties' detailed information.
  - There are two buttons including like and share in the properties' detailed information.
  - There is a map showing on the right side of the page containing rental house markers.
  - The sketch figure of the interface design is shown in Figure 30.

1.3.16 Landing Page

Actors:

All users

Description:

- There are three components on this page.
  - Header: As a warm welcome for users and state the intention of the website.
  - Sample Property Display: Display sample properties.
  - Property Display and Search: Display the listing of properties for rent and allow users to search properties of their interest by address, property type, bedroom number and maximum price. (The demo below is for interim, for final design please refer to the final project)

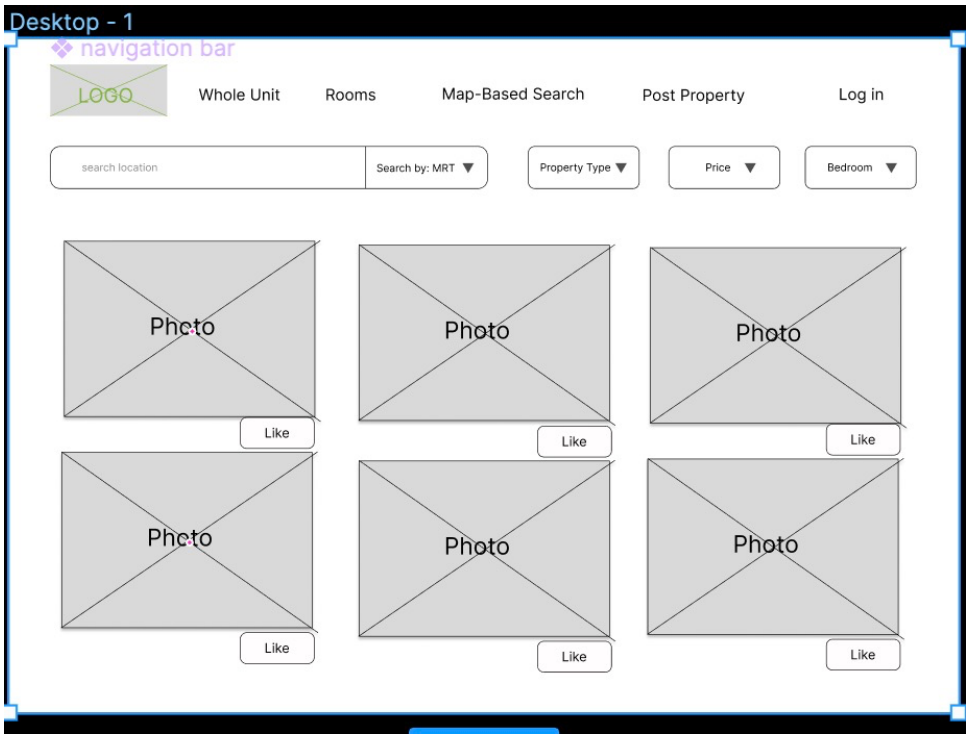

Figure 30 Landing Page