**Use Cases**

**for**

FlatFinderSG

**Version 1.0 approved**

**Prepared by Wei Hao and Jabez**

**SC2006 - One.zero**

**5/2/2023**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Jabez Ng Yong Xin | 05/02/2023 | First Draft of Use Case Descriptions | 0.0 |
| Gay Wei Hao | 06/02/2023 | Added Use Case Descriptions | 1.0 |
| Jabez Ng Yong Xin | 07/02/2023 | Updated Use Case Descriptions | 1.2 |
| Jabez Ng Yong Xin | 09/02/2023 | Refine Use Case Descriptions | 1.3 |
| Jabez Ng Yong Xin | 19/03/2023 | Edit Use Case Description | 1.4 |
| Gay Wei Hao | 29/03/2023 | Edit Use Case Description | 1.5 |

# **Guidance for Use Case Template**

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

# **Use Case Identification**

## **Use Case ID**

Give each use case a unique numeric identifier, in the hierarchical form: X.Y. Related use cases can be grouped in the hierarchy. Functional requirements can be traced back to a labeled use case.

## **Use Case Name**

State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun. Some examples:

* View part number information.
* Manually mark hypertext source and establish a link to the target.
* Place an order for a CD with the updated software version.

## **Use Case History**

### **Created By**

Supply the name of the person who initially documented this use case.

### **Date Created**

Enter the date on which the use case was initially documented.

### **Last Updated By**

Supply the name of the person who performed the most recent update to the use case description.

### **Date Last Updated**

Enter the date on which the use case was most recently updated.

# **Use Case Definition**

## **Actor**

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor(s) that will be performing this use case.

## **Description**

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

## **Preconditions**

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. User’s identity has been authenticated.
2. User’s computer has sufficient free memory available to launch task.

## **Postconditions**

Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. Document contains only valid SGML tags.
2. Price of item in database has been updated with new value.

## **Priority**

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

## **Frequency of Use**

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

## **Flow of Events**

Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, “How do I <accomplish the task stated in the use case name>?” This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system.

## **Alternative Flows**

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative course, and describe any differences in the sequence of steps that take place. Number each alternative course using the Use Case ID as a prefix, followed by “AC” to indicate “Alternative Course”. Example: X.Y.AC.1.

## **Exceptions**

Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. Number each exception using the Use Case ID as a prefix, followed by “EX” to indicate “Exception”. Example: X.Y.EX.1.

## **Includes**

List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.

## **Special Requirements**

Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.

## **Assumptions**

List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.

## **Notes and Issues**

List any additional comments about this use case or any remaining open issues or TBDs (To Be Determineds) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

**Use Case** Descriptions

| Use Case ID: | ACC1 | | |
| --- | --- | --- | --- |
| Use Case Name: | Register | | |
| Created By: | Wei Hao | Last Updated By: | Glendon |
| Date Created: | 6/2/2023 | Date Last Updated: | 14/4/2023 |

| Actor: | User |
| --- | --- |
| Description: | Creates a new account for actors |
| Preconditions: | 1. The app starts up   and   1. The user clicks “Register” on the welcome page |
| Postconditions: | 1. The user clicks “Register” on the register page   and   1. The user’s details are considered valid by the system   or   1. The user exits the app |
| Priority: | High |
| Frequency of Use: | Low. New users to the app have to register (once) while existing users do not need to register |
| Flow of Events: | 1. The system requests the details:Email Address, Password, Display Name and Age 2. The system validates the requested details 3. The system stores details in the database 4. The system logs the actor in |
| Alternative Flows: | ACC1.AF.1 If the Email Address is already registered   1. The system displays “Email Address is already registered” 2. The system returns to Step 1 when the actor clicks the “Register” button   ACC1.AF.2 If the Email Address provided is not valid   1. The system displays “Email Address is not valid” 2. The system returns to Step 1 when the actor clicks the “Register” button   ACC1.AF.3 If the new password does not meet the requirement which is a minimum length of 8 characters containing at least a capital letter and a number   1. The system displays “Password must have at least 8 characters, with at least 1 capital letter and 1 number!” 2. The system returns to Step 1 when the actor clicks the “Register” button   ACC.AF.4 If the username is already taken   1. The system displays “Username already exists” 2. The system returns to Step 1 when the actor clicks the “Register” button |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | User is connected to the Internet |
| Notes and Issues: | NIL |

| Use Case ID: | ACC2 | | |
| --- | --- | --- | --- |
| Use Case Name: | Login | | |
| Created By: | Wei Hao | Last Updated By: | Glendon |
| Date Created: | 6/2/2023 | Date Last Updated: | 14/4/2023 |

| Actor: | User |
| --- | --- |
| Description: | Signs users into account to access features |
| Preconditions: | 1. The app starts up   and   1. The user has to have an existing account registered in the system |
| Postconditions: | 1. The user clicks “Login”   and   1. The user has entered the valid email address and password associated with the account / successful validation by the system   or   1. The user exits the app |
| Priority: | High |
| Frequency of Use: | Low. Typically a user logs in once and remains signed in. However, in the case where a user signs out, a subsequent login is necessary. |
| Flow of Events: | 1. The system displays a welcome page and awaits the user to click on ‘Get started!’ button 2. The system requests the details: email address and password 3. The system validates the details requested: 4. The email address belongs to a registered account 5. The password matches the registered account’s password 6. The system logs the user in 7. The system will display the Homepage |
| Alternative Flows: | ACC.AF.1 If Email Address or Password is incorrect   1. The system displays the message“Invalid Email Address or Password” 2. The system will return to step 2 |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | User is connected to the Internet |
| Notes and Issues: | NIL |

| Use Case ID: | PF1 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Profile | | |
| Created By: | Wei Hao | Last Updated By: | Jabez Ng |
| Date Created: | 6/2/2023 | Date Last Updated: | 9/2/2023 |

| Actor: | User |
| --- | --- |
| Description: | Allows users to view their profile page |
| Preconditions: | 1. The user selected “Profile” in the navigation bar |
| Postconditions: | 1. The user selects “Edit Profile”   or   1. The user selects “Change Password”   or   1. The user selects “My Properties”   or   1. The user selects a navigation bar option   or   1. The user selects “Log Out”   or   1. The user exits the app |
| Priority: | High |
| Frequency of Use: | 0-4 times per login |
| Flow of Events: | 1. System displays Profile 2. If the user selects any of the navigation bar options, the user will be redirected to the selected page 3. On the profile page, the user's profile image, username, and email will be displayed 4. The system lets the user select between multiple options: 5. Edit Profile 6. Change Password 7. My Properties |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | User is connected to the Internet |
| Notes and Issues: | NIL |

| Use Case ID: | PF3 | | |
| --- | --- | --- | --- |
| Use Case Name: | Manage My Properties | | |
| Created By: | Wei Hao | Last Updated By: | Jabez Ng |
| Date Created: | 6/2/2023 | Date Last Updated: | 9/2/2023 |

| Actor: | User |
| --- | --- |
| Description: | Used to manage properties listed |
| Preconditions: | 1. The user selects “Manage My Properties” on Profile page |
| Postconditions: | 1. The user selects “Edit Listing”   or   1. The user selects “Remove Listing”   or   1. The user selects the page return button (top left corner)   or   1. The user exits the app |
| Priority: | High |
| Frequency of Use: | Medium. For sellers on the application, changes may be made frequently regarding price especially. |
| Flow of Events: | 1. The system displays the list of properties listed by the user 2. The user can choose one of the properties listed 3. The user can then select one of two options for the selected property listing: 4. Edit listing 5. Delete listing |
| Alternative Flows: | NIL |
| Exceptions: | PF3.EX.1 If there are no property listings   1. The system displays “No property listings yet!” |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | 1. User is connected to the Internet 2. User has at least one property listed lest the exception case |
| Notes and Issues: | NIL |

| Use Case ID: | MP1 | | |
| --- | --- | --- | --- |
| Use Case Name: | Edit Listing | | |
| Created By: | Wei Hao | Last Updated By: | Glendon |
| Date Created: | 6/2/2023 | Date Last Updated: | 14/4/2023 |

| Actor: | User |
| --- | --- |
| Description: | Allows users to edit details about property listed |
| Preconditions: | 1. The user has at least one property listing   and   1. The user chooses a property listing   and   1. The user selects “Edit” |
| Postconditions: | 1. Details are updated based on the edit made by the user   or   1. The user exits the app |
| Priority: | High |
| Frequency of Use: | Medium. For sellers on the application, changes may be made frequently regarding price especially. |
| Flow of Events: | 1. System prompts the user to edit details: Property Name, Description, Price (per square feet), Address, Image, Number of bedrooms, Number of bathrooms, Dimensions (square feet), Leftover lease, and Neighbourhood 2. The user will then select “Save Changes” upon completion of the edit and be prompted with two buttons “Confirm Changes” and “Cancel” 3. Seller selects “Confirm Changes” 4. The system will update the listing based on the details provided on the edit |
| Alternative Flows: | MP1.AF.1 User selects “Cancel”   1. The system returns to step 1 |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | 1. User is connected to the Internet 2. User has at least one property listed lest the exception case |
| Notes and Issues: | NIL |

| Use Case ID: | MP2 | | |
| --- | --- | --- | --- |
| Use Case Name: | Remove Listing | | |
| Created By: | Wei Hao | Last Updated By: | Glendon |
| Date Created: | 6/2/2023 | Date Last Updated: | 14/4/2023 |

| Actor: | User |
| --- | --- |
| Description: | Allows users to remove their property listing |
| Preconditions: | 1. The user has at least one property listing   and   1. The user chooses a property listing   and   1. The user selects “Delete” |
| Postconditions: | 1. The listing data in the database is removed and cleared   or   1. The user exits the app |
| Priority: | High |
| Frequency of Use: | Medium. Removal of a property listing is typically done by a seller who either decides to not sell the property or has sold the property to a buyer. |
| Flow of Events: | 1. The user will select “Del” and be prompted with two buttons “Confirm Remove” and “Cancel” 2. Seller selects “Confirm Remove” 3. The system will remove the listing from the database |
| Alternative Flows: | MP2.AF.1 User selects “Cancel”   1. The system returns to step 1 |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | 1. User is connected to the Internet 2. User has at least one property listed lest the exception case |
| Notes and Issues: | NIL |

| Use Case ID: | PF2 | | |
| --- | --- | --- | --- |
| Use Case Name: | Edit Profile | | |
| Created By: | Wei Hao | Last Updated By: | Glendon |
| Date Created: | 6/2/2023 | Date Last Updated: | 14/4/2023 |

| Actor: | User |
| --- | --- |
| Description: | Allows users to enter/edit their personal details (Profile photo, Display Name, Email Address, Phone Number, Personal address location, Single/Couple/Family, First Time Applicants or Not, Household Income Ceiling, Address of Parents, Age, Citizenship, Ownership in other properties) |
| Preconditions: | 1. The user selects “Edit Profile” in the “Profile” page |
| Postconditions: | 1. The user selects “Save Changes”   or   1. The user selects the page return button (top left corner)   or   1. The user exits the app |
| Priority: | High |
| Frequency of Use: | 0-1 times per login |
| Flow of Events: | 1. The user selects the “Edit Profile” button 2. The system displays a list of details for the user to fill in (Profile photo, Display Name, Email Address, Phone Number, Personal address location, Single/Couple/Family, First Time Applicants or Not, Household Income, Address of Parents, Age, Citizenship, Ownership in other properties) 3. Upon completion, the user selects “Save Changes” 4. The data entered by the user will be saved to the user's account in the database by the system |
| Alternative Flows: | PF2.AF.1 User selects “Cancel”   1. The system returns to step 1   PF2.AF.2 If the Email Address is already registered   1. The system displays “Email Address is already registered” 2. The system returns to Step 1 when the actor clicks the “Save changes” button   PF2.AF.3 If the Email Address provided is not valid   1. The system displays “Email Address is not valid” 2. The system returns to Step 1 when the actor clicks the “Save changes” button   PF2.AF.4 If the username is already taken   1. The system displays “Username already exists” 2. The system returns to Step 1 when the actor clicks the “Save changes” button   PF2.AF.5 If the Phone Number is already taken   1. The system displays “Phone Number is registered with another account” 2. The system returns to Step 1 when the actor clicks the “Save changes” button   PF2.AF.6 If the Phone Number is not valid   1. The system displays “Phone Number is not valid” 2. The system returns to Step 1 when the actor clicks the “Register” button   PF2.AF.7 The user leaves at least one required field empty   1. User selects “Save changes” 2. The system will display a notification saying “Not all required fields have been filled up!” |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | User is connected to the Internet |
| Notes and Issues: | NIL |

| Use Case ID: | PF3 | | |
| --- | --- | --- | --- |
| Use Case Name: | Change Password | | |
| Created By: | Wei Hao | Last Updated By: | Wei Hao |
| Date Created: | 6/2/2023 | Date Last Updated: | 29/3/2023 |

| Actor: | User |
| --- | --- |
| Description: | Allows Users to change password |
| Preconditions: | 1. The user selects “Profile”   and   1. The user selects “Change Password” |
| Postconditions: | 1. The user selects “Save Changes” for the system to validate the data   or   1. The user selects the page return button (top left corner)   or   1. The user exits the app |
| Priority: | High |
| Frequency of Use: | Medium. For security reasons, users should regularly change their passwords (Once every 3 months) |
| Flow of Events: | 1. The system requests the user for their email 2. The user selects the “Change Password” Button 3. The system sends a change password link to the users email address 4. The system prompts the user to log out 5. User changes password from the link that was sent to the users email address |
| Alternative Flows: | PF3.AF.1 If the email field is empty   1. The system displays “Please Enter your email” 2. The system returns to Step 1 and the user is required to enter their required password again   PF3.AF.4 The user does not fill in any password and selects “Save Changes”   1. The system displays “New Password is empty!” |
| Exceptions: | PF3.EX.1 User does not change password   1. The user selects the page return button (top left corner) |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | User is connected to the Internet |
| Notes and Issues: | NIL |

| Use Case ID: | SE1 | | |
| --- | --- | --- | --- |
| Use Case Name: | Send Email | | |
| Created By: | Jabez Ng | Last Updated By: | Jabez Ng |
| Date Created: | 19/03/2023 | Date Last Updated: | 19/03/2023 |

| Actor: | User |
| --- | --- |
| Description: | Allows the user to send an email to the property owner to establish a form of communication |
| Preconditions: | 1. The user selects “Email” in the individual listing page |
| Postconditions: | 1. The user presses the send button to send the input message via email to the receiving party   or   1. The user exits the app |
| Priority: | High |
| Frequency of Use: | High. Communication between buyers and sellers is a key aspect of FlatFinderSG, hence it is vital to allow buyers to establish a form of communication with the seller |
| Flow of Events: | 1. System displays a text field to allow the user to input a message that he/she intends to send to the seller 2. The user presses the send button to send an email containing the message input to the seller |
| Alternative Flows: | NIL |
| Exceptions: | SE1.EX.1 No email associated with the individual listing   1. The system displays the error “No email associated with seller!” |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | User is connected to the Internet |
| Notes and Issues: | NIL |

| Use Case ID: | VL1 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Likes | | |
| Created By: | Wei Hao | Last Updated By: | Jabez Ng |
| Date Created: | 6/2/2023 | Date Last Updated: | 9/2/2023 |

| Actor: | User |
| --- | --- |
| Description: | Allows the user to view their list of “liked” property listings |
| Preconditions: | 1. User selects “Likes” in the navigation bar |
| Postconditions: | 1. User returns back to Homepage   or   1. User exits the app |
| Priority: | Low |
| Frequency of Use: | Medium. The user may frequent the likes page to avoid the hassle of using the search bar. |
| Flow of Events: | 1. System displays a list of property listings that the user has liked 2. The user is able to remove a listing from likes by selecting the heart icon button labeled “Remove from likes” 3. The user is able to select a property listing to view the details of the selected listing |
| Alternative Flows: | NIL |
| Exceptions: | VL.EX.1 No properties listed   1. If there are no liked properties listed, the system displays the message “No liked properties” 2. System returns to Homepage when user clicks “Home” button on navigation bar |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | User is connected to the Internet |
| Notes and Issues: | NIL |

| Use Case ID: | AL1 | | |
| --- | --- | --- | --- |
| Use Case Name: | Add Listing | | |
| Created By: | Wei Hao | Last Updated By: | Glendon |
| Date Created: | 6/2/2023 | Date Last Updated: | 14/4/2023 |

| Actor: | User |
| --- | --- |
| Description: | Allows users to upload a property listing |
| Preconditions: | 1. The user selects the “+” button in the navigation bar which signifies the “Add Listing” function |
| Postconditions: | 1. The user submits the data for the listing to be added   or   1. The user exits the app |
| Priority: | High |
| Frequency of Use: | Medium. Depending on the nature of the user, the number of properties that a user has to sell is generally low (unless the user is a property agent) |
| Flow of Events: | 1. System prompts the user to input the details: Description, Price (per square feet), Address, Image, Number of bedrooms, Number of bathrooms, Dimensions (square feet), Leftover lease, Neighbourhood, Postal Code and Property Name 2. The user will be provided the feature such that on postal code input, the address fields will be input automatically 3. The user will select “Submit” and be prompted with two buttons to “Confirm Submit” and “Cancel” 4. The user selects “Confirm Submit” 5. The system will upload this new listing to the database upon data validation and user will be redirected back to HomePage |
| Alternative Flows: | AL1.AF.1 Description is left empty   1. The system displays “Description Cannot Be Empty!” 2. The system returns to Step 1   AL1.AF.2 Price is left empty   1. The system displays “Price Cannot Be Empty!” 2. The system returns to Step 1   AL1.AF.3 Address is left empty   1. The system displays “Address Cannot Be Empty!” 2. The system returns to Step 1   AL1.AF.4 No Images are uploaded   1. The system displays “Please Upload At Least One Photo!” 2. The system returns to Step 1   AL1.AF.5 Number of bedrooms and bathrooms are left empty   1. The system displays “Number of Bedrooms Cannot Be Empty!” and “Number of Bathrooms Cannot Be Empty!” respectively. 2. The system returns to Step 1   AL1.AF.6 Dimensions is left empty   1. The system displays “Dimensions Cannot Be Empty!” 2. The system returns to Step 1   AL1.AF.7 Leftover Lease is left empty   1. The system displays “Leftover Lease Cannot Be Empty!” 2. The system returns to Step 1   AL1.AF.8 Neighbourhood is left empty   1. The system displays “Neighbourhood Cannot Be Empty!” 2. The system returns to Step 1   AL1.AF.9 Address added is not in Singapore   1. The System displays “Address must be in Singapore” 2. The system returns to Step 1   AL1.AF10 Property name is left empty   1. The system displays “Property Name Cannot Be Empty!” 2. The system returns to Step 1 |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | User is connected to the Internet |
| Notes and Issues: | NIL |

| Use Case ID: | BP1 | | |
| --- | --- | --- | --- |
| Use Case Name: | View List Of Properties | | |
| Created By: | Wei Hao | Last Updated By: | Jabez Ng |
| Date Created: | 6/2/2023 | Date Last Updated: | 9/2/2023 |

| Actor: | User |
| --- | --- |
| Description: | Allows the user to properties listed |
| Preconditions: | 1. The user has a registered account and has logged in |
| Postconditions: | 1. The user selects an individual listing   or   1. The user selects one of the multiple buttons on the navigation bar   or   1. The user exits the app |
| Priority: | High |
| Frequency of Use: | High. Users with the intention to find and purchase properties will be browsing through property listings frequently. |
| Flow of Events: | 1. The system displays a list of properties on the HomePage 2. The system lets the user browse through existing property listings whose data have been fetched using the Property API as well as property listings added by users of the app 3. Users are able to select the “Add to likes” button to save a particular listing to their “Likes” page 4. Users are able to select the individual property listing to view its details |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | 1. There is at least one property listed 2. User is connected to the Internet 3. Property listed is in Singapore |
| Notes and Issues: | NIL |

| Use Case ID: | AL1 | | |
| --- | --- | --- | --- |
| Use Case Name: | Add To Likes | | |
| Created By: | Wei Hao | Last Updated By: | Glendon |
| Date Created: | 6/2/2023 | Date Last Updated: | 14/4/2023 |

| Actor: | User |
| --- | --- |
| Description: | Allows the user to save properties to a list of liked properties |
| Preconditions: | 1. The user has logged in 2. The user is browsing through properties that have not been added to likes   or   1. The user is viewing a particular individual property and it has not added that listing to likes |
| Postconditions: | 1. The property listing selected is added by the selection of the button “Add to likes”   or   1. The user exits the app |
| Priority: | Low |
| Frequency of Use: | High. Users will browse through many listings on the homepage and add listings that they are interested in, to their likes page. |
| Flow of Events: | 1. User clicks on “Add to likes” button 2. The system adds the current property to a list of liked properties 3. The system displays a message “Successfully added to likes” |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | 1. User is connected to the Internet 2. There is at least one property listed 3. Property listed is in Singapore |
| Notes and Issues: | “Unliking” or removing a listing from the likes page can be performed by selecting the heart icon button which will be labeled “Remove from likes”. However, this can only be done if the listing is in the “Likes” page. |

| Use Case ID: | SP1 | | |
| --- | --- | --- | --- |
| Use Case Name: | Search Properties | | |
| Created By: | Wei Hao | Last Updated By: | Jabez Ng |
| Date Created: | 6/2/2023 | Date Last Updated: | 15/3/2023 |

| Actor: | User |
| --- | --- |
| Description: | Allows the user to search for properties listed based on address |
| Preconditions: | 1. The user has logged in 2. The user has selected the search bar |
| Postconditions: | 1. The system has obtained the property listings whose address matches the input given to the search bar 2. The user begins browsing through the filtered property listings   or   1. The user exits the app   or   1. The user selects the property listing |
| Priority: | High |
| Frequency of Use: | High. The search bar provides a more convenient alternative to manually browsing through listings on the homepage. |
| Flow of Events: | 1. The user enters an address location of interest into the search bar input 2. The user is able to apply filters (by the number of rooms) if needed 3. The system filters the property listings in the database based on the search bar address input and / or the filter input. The search is done in real-time and checks if the text input is in the address string. 4. A list of properties matching the user’s inputs is displayed 5. The user is able to browse and select an individual listing of their choice |
| Alternative Flows: | SP1.AF.1 If there are no listings that meet the user's requirements   1. An empty screen will be displayed indicating that no property has an address that contains the text input. 2. System returns to step 3 |
| Exceptions: | NIL |
| Includes: | View List of Properties |
| Special Requirements: | NIL |
| Assumptions: | 1. User is connected to the Internet 2. There is at least one property listed 3. Properties listed are in Singapore |
| Notes and Issues: | NIL |

| Use Case ID: | SP2 | | |
| --- | --- | --- | --- |
| Use Case Name: | Apply Filters | | |
| Created By: | Jabez Ng | Last Updated By: | Jabez Ng |
| Date Created: | 9/2/2023 | Date Last Updated: | 15/3/2023 |

| Actor: | User |
| --- | --- |
| Description: | Allows the user to filter their search input by the number of rooms |
| Preconditions: | 1. The user is at the search page 2. The user needs to select a choice chip to filter the search results based on the number of bedrooms |
| Postconditions: | 1. The system has filtered out the property listings based on the filters added   or   1. The user exits the app |
| Priority: | High |
| Frequency of Use: | High. A filter function is quintessential in optimizing the results shown to the users |
| Flow of Events: | 1. The system will display 4 choice chips of filters by the number of rooms (2-room, 3-room, 4-room, 5-room). 2. The user selects the desired filter option based on their requirements 3. The system subsequently filters the listview of property listings by the number of rooms based on the choice chip option selected    1. The selected filter option will remain as such until the user makes a change 4. Pressing the choice chip will deselect it if it is selected |
| Alternative Flows: | SP2.AF.1 If there are no listings that meet the user's requirements   1. The system displays an empty listview indicating that there are not listings that match the filter input 2. System returns to step 3 |
| Exceptions: | NIL |
| Includes: | View List of Properties |
| Special Requirements: | NIL |
| Assumptions: | 1. User is connected to the Internet 2. There is at least one property listed 3. Properties listed are in Singapore |
| Notes and Issues: | NIL |

| Use Case ID: | PP1 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Individual Property | | |
| Created By: | Wei Hao | Last Updated By: | Wei Hao |
| Date Created: | 6/2/2023 | Date Last Updated: | 25/2/2023 |

| Actor: | User |
| --- | --- |
| Description: | Allows the user to view more details about the selected property |
| Preconditions: | 1. The user has logged in 2. The user is browsing through property listings 3. The user selects an individual property |
| Postconditions: | 1. The user selects the page return button (top left corner)   or   1. The user selects “Compare”   or   1. The user exits the app |
| Priority: | High |
| Frequency of Use: | High. Viewing an individual listing to be able to utilize the innovative feature the Grant Calculator |
| Flow of Events: | 1. Pictures of the house, details of the house, price of the house, and address are displayed for the user. These data will be fetched from the application database. 2. The status of the listing will be displayed as well (“Available!”, “Sold!”) 3. If the “Compare” button is selected, the user will be directed to a new page where the user can compare the price of the current listing with the past prices of similar flats 4. If the “Add to likes” button is selected, the selected property listing will be added to “Likes” page 5. If the “Google Maps” view is selected, the user will be able to view the property location on Google Maps 6. The user will be able to utilize the Grant Calculator feature to view eligible grants 7. The user will also be able to send an email to the seller by pressing the “Email” button |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | 1. User is connected to the Internet 2. Property listed is in Singapore |
| Notes and Issues: | NIL |

| Use Case ID: | PP2 | | |
| --- | --- | --- | --- |
| Use Case Name: | Grant Calculator | | |
| Created By: | Wei Hao | Last Updated By: | Jabez Ng |
| Date Created: | 6/2/2023 | Date Last Updated: | 7/2/2023 |

| Actor: | User |
| --- | --- |
| Description: | Allows the user to determine what sort of grant they are eligible for |
| Preconditions: | 1. The user selects an individual property listing 2. The user completed filling in all details required for profile |
| Postconditions: | 1. The user exits the “Utilise Grant Calculator” feature   or   1. The user exits the app |
| Priority: | High |
| Frequency of Use: | 1-8 times per login |
| Flow of Events: | 1. The system will examine the details of the house listed and the user's details to determine the grants the user is eligible for 2. The system will display the grant amount the user is eligible for 3. The system will display the type of grants the user is eligible for 4. The system will display the price before the grant is applied and the price after the grant is applied |
| Alternative Flows: | PP2.SF.1 Missing Profile Details   1. The system will display a button notification on top of the Grant Calculator UI section displaying “Edit Profile” 2. Users will have to fill in the required details for their profile to allow the system to calculate the grants that they qualify for 3. The system will overlay the Grant Calculator section |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | User is connected to the Internet |
| Notes and Issues: | NIL |

| Use Case ID: | PP3 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Address Location | | |
| Created By: | Wei Hao | Last Updated By: | Jabez Ng |
| Date Created: | 6/2/2023 | Date Last Updated: | 9/2/2023 |

| Actor: | User, GoogleMaps API |
| --- | --- |
| Description: | Allows the user to view the location of the listed property on GoogleMaps |
| Preconditions: | 1. The user selects an individual property listing 2. The user selects the Google Maps display |
| Postconditions: | 1. The user exits the Google Maps view in the app   or   1. The user exits the app |
| Priority: | High |
| Frequency of Use: | Medium. Having such a feature where the user can have a better view and understanding of the property’s exact location is crucial |
| Flow of Events: | 1. The system integrates with the GoogleMaps API 2. The system displays the location of the property based on the location address data of that listing 3. The user is able to zoom in and out and adjust the orientation of the window |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | 1. User is connected to the Internet 2. There is at least one property listed |
| Notes and Issues: | NIL |

| Use Case ID: | PP4 | | |
| --- | --- | --- | --- |
| Use Case Name: | Compare Properties Feature | | |
| Created By: | Jabez Ng | Last Updated By: | Wei Hao |
| Date Created: | 9/2/2023 | Date Last Updated: | 9/2/2023 |

| Actor: | User, Properties API |
| --- | --- |
| Description: | Allows the user to select a specific type of property (Location and Number of rooms) before displaying the average price and a listview of that property type based on past resale listing data after 2016 from the API, for comparison. |
| Preconditions: | 1. The user selects an individual property listing 2. The user selects the “Compare” button |
| Postconditions: | 1. The user selects the page return button (top left corner)   or   1. The user exits the app |
| Priority: | High. |
| Frequency of Use: | High. This is a unique feature of the application which aids the user in understanding the property market based on past trends. This will allow the user to make a wiser and more informed decision in purchasing the property. |
| Flow of Events: | 1. The system will retrieve all properties’ prices based on the location and number of rooms for the past 3 years from API 2. The system will calculate the average price of the listings matching the property’s location and bedroom number 3. The system will display the number of rooms and price of the individual listing along with the average price of the resale listings and the list view of those resale listings 4. In the listview, details regarding the number of rooms, address and individual price of the resale listing will be displayed |
| Alternative Flows: | NIL |
| Exceptions: | PP4.EX.1 No listings matches the individual property listing   1. System will display the average price to be null and there will be no listings displayed in the list view 2. The system returns to Step 1 when the actor clicks the “Continue” button |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | User is connected to the Internet |
| Notes and Issues: | NIL |