**Software Requirements Specification**

**for**

FlatFinderSG

**Version 1.0 approved**

**Prepared by**

BRYAN TAN SHAO XUAN

FAN YUPEI

GAY WEI HAO

GOH XIAN HAO GLENDON

JABEZ NG YONG XIN

**Team One.Zero**

**1 Apr 2023**

**Table of Contents**

**1. Introduction**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 2

1.5 References 2

**2. Overall Description**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 3

2.4 Operating Environment 4

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 5

**3. External Interface Requirements**

3.1 User Interfaces 6

3.2 Hardware Interfaces 26

3.3 Software Interfaces 26

3.4 Communications Interfaces 27

**4. System Features**

4.1 Register - ACC1 28

4.2 Login - ACC2 29

4.3 View Profile - PF1 31

4.4 Manage My Properties - PF3 32

4.5 Edit Listing - MP1 33

4.6 Remove Listing - MP2 34

4.7 Edit Profile - PF2 35

4.8 Change Password - PF3 36

4.9 Send Email - SE1 37

4.10 View Likes - VL1 38

4.11 Add Listing - AL1 38

4.12 View List of Properties - BP1 40

4.13 Add To Likes - AL1 41

4.14 Search Properties - SP1 42

4.15 Apply Filters - SP2 43

4.16 View Individual Properties - PP1 44

4.17 Grant Calculator - PP2 45

4.18 View Address Location - PP3 46

4.19 Compare Properties Feature - PP4 47

**5. Other Nonfunctional Requirements**

5.1 Performance Requirements 48

5.2 Safety Requirements 49

5.3 Security Requirements 49

5.4 Software Quality Attributes 49

5.5 Business Rules 50

**6. Other Requirements**

6.1 Reuse Objectives50

6.2 MVC Framework 51

**Appendix A: Glossary** 51

**Appendix B: Analysis Models** 54

**Appendix C: To Be Determined List** 55

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

# **Introduction**

## **Purpose**

This document specifies the software requirements of the mobile application FlatFinderSG version 1.0, a cross-platform mobile application. This document will cover the main functionalities of the application, the various user interfaces, as well as the constraints.

FlatFinderSG is a user-friendly mobile application that seeks to streamline the process of buying and selling home properties by providing a range of features for users to gain insights on HDB grants and resale housing pricing.

## **Document Conventions**

The document makes use of the following terms:

| **Term** | **Definition** |
| --- | --- |
| User | Potential house sellers or buyers who want to search for an ideal housing with specific search requirements to find the housing with specifications that best suit them. |
| Device | Mobile Phone (iOS, Android) |
| Input | Actions by Users - e.g Search Keywords, Filters Applied |
| Output | Displays on the screen after input - e.g Search Results of List of Properties, Map of Property Location |
| Property Listing | A housing property that has been listed on the application |
| Resale Prices | Singapore’s Resale Housing Price data |

## **Intended Audience and Reading Suggestions**

This document is intended for developers, project managers, and users, to fully understand how the FlatFinderSG mobile application works. All stakeholders are encouraged to read the documentation in a sequential order to fully understand the architecture of the FlatFinderSG application.

## **Product Scope**

From research, despite the high property prices, 74% of Singaporeans are looking to buy a new home. Moreover, 1 in 2 Singaporeans plan to buy a new home in the next 2 years. Furthermore, in light of the soaring BTO prices, the demand for resale housing has been increasing.

Thus, our group believes in streamlining the process of buying and selling housing property in Singapore. Buyers will be able to easily find housing properties that suit their requirements while sellers can list their housing properties with just a simple tap of a button. Therefore, our application is ideal for potential home buyers/sellers in Singapore who want to buy/sell housing properties with ease. Our application is extremely relevant and useful for Singaporean homeowners/buyers.

## **References**

*Demand For Housing*. (2021, August 25). Singapore Business Review. <https://sbr.com.sg/residential-property/in-focus-news/1-in-2-singaporeans-plan-buy-new-home-in-next-2-years>

*Demand For Resale Housing.* (2022, December 22). PropertyLimBrothers. <https://www.propertylimbrothers.com/year-in-review-2022-rise-of-million-dollar-resale-hdb-flats/>

# **Overall Description**

## **Product Perspective**

FlatFinderSG is an improved version of current applications that allow the buying and selling of housing properties in Singapore. By providing a one-stop e-commerce platform, individuals seeking to buy or sell housing properties can easily add a listing to the platform or search for an ideal listing that they desire, thereby improving the overall process of buying and selling property listings. By leveraging a grant calculator feature and a compare price with resale flats feature, buyers will be able to easily gain access to crucial data to aid them in making a more informed decision. This will help tremendously in improving the overall user experience.

## **Product Functions**

1. FlatFinderSG allows users to access and view all property listings on the application conveniently.
2. FlatFinderSG allows users to search for property listings by their address location.
3. FlatFinderSG allows users to filter the displayed property listings by the number of bedrooms.
4. FlatFinderSG allows users to view the relevant details of the property listing such as the price, address, number of bedrooms, number of bathrooms, dimensions, and get-locations of the property.
5. FlatFinderSG allows users to add a property listing to the platform.
6. FlatFinderSG allows users to compare the price of the property listing with the prices of resale properties in the same neighborhood and with the same number of rooms
7. FlatFinderSG allows users to view the average price of all resale properties in the same neighborhood and with the same number of rooms
8. FlatFinderSG allows users to view the number of grants and the grant types that they are eligible for as well as the property price after the grant deduction

## **User Classes and Characteristics**

FlatFinderSG aims to improve individual’s home searching experience by providing a one-stop e-commerce platform with information about property listings on sale, its price, its address location, its number of bedrooms, number of bathrooms, its dimensions etc. There are mainly 2 groups of users with different purposes on the application.

| **User Class** | **Description** |
| --- | --- |
| Individuals who are looking for their ideal property listing | This user class consists of users who want to find an ideal housing property based on possibly the location of the property, the number of bedrooms that the property has, and most importantly, the price of the housing property. All these factors are crucial in the decision-making process of a buyer. Moreover, they must be able to compare the property price with the prices of other resale listings in Singapore to gauge if the price is ideal. Furthermore, being able to take into account any possible grants that they are eligible for will also aid the decision-making process tremendously. |
| Individuals who intend to list their property for sale | This user class consists of users who want to sell their housing property and wish to be able to list their property without much hassle. Viewing other housing properties on the application allows the seller to list their property at a price that they presume will be competitive. |

## **Operating Environment**

**Frontend Development**

* Dart: Flutter
  + Flutter Version: 3.7.8
  + Dart Version: 2.19.5
  + Architecture: x64

**Backend Development**

* Firebase (NoSQL Database)
  + Cloud Firestore
  + Analytics: Firebase Analytics
  + Monitoring: Firebase Performance Monitoring
  + Test: Firebase Test Lab

**External API**

* Google Maps API
* Data.Gov Resale Flat Prices API
* Geocoding API
* EmailJS API

**Operating Platform**

* Development Platform: IOS XCode Simulator and Android Studio
* For Android Studio: The minimum Android SDK version is 28, with the SDK version 32 recommended for full functionality
* Operating System Platform: IOS and Android

## **Design and Implementation Constraints**

One design limitation we face is the reliance on current data with regards to the available grants which was built into the grant calculator feature. Due to the constant changes in grant schemes and policies in Singapore, our users will be limited to the current grant data that we have built into our application.

## **User Documentation**

List of User Documentation Components:

1. Demo Video to bring the User through how to use each aspect of the application.
2. User Manual (under User Interfaces in Section 3.1) to list detailed steps and preconditions with illustrated examples to assist the user in using the application, including instructions for troubleshooting problems that crop up when using the application, such as how to deal with different errors.

## **Assumptions and Dependencies**

The main underlying assumption is that users using the application have an active internet connection and enabled location services.

Google Maps is also utilized as a third-party application, and therefore we require this service to be installed and working in the users’ device for the application to be used accurately

The application relies on external API such as Data.gov Resale Flat Prices API, thus we require the service to be working for the application to be used accurately.

# **External Interface Requirements**

## **User Interfaces**

| **3.1.1: Login Screen** |
| --- |
| Figure 3.1.1.1 and 3.1.1.2: Welcome Page and Login Screen |
| 1. Welcome page must be the first page users see when entering the application.    1. Users shall click on ‘Get started!’ to navigate to the Login Screen (Figure 3.1.1.2) 2. Users shall key in their registered email and password to login to their account in the respective text fields for email and password as shown in figure 3.1.1.2.    1. Texts keyed into the password field shall be censored.    2. Login page shall have a feedback that indicates erroneous login details. 3. Login page shall have an option to allow users to register an account which will allow users to navigate to the sign up page (Figure 3.1.3.1) |

| **3.1.2: Home Page** |
| --- |
| Figure 3.1.2.1: Home Page |
| 1. Home Page shall include a welcome message, showing “Welcome, \*email\*”    1. Email is the email that the user registered with 2. Users shall be able to scroll vertically to browse through listings    1. Listings shall display:       1. Property Name       2. Property Address       3. Price       4. Number of bedroom and bathroom    2. Listings shall have the option “Add to likes” which will include the property in the Likes page (Figure 3.1.4.1) 3. Users shall be able to click on listings to navigate to individual listing pages (Figure 3.1.11.1) 4. Home page shall include a search bar    1. Users must be able to enter address of interest into search bar input    2. Users must be able to apply filters if needed       1. Only one filter shall be applied at any one time    3. Search bar is responsible for filtering property listing in the database based on the input text       1. Search option must check if the text input is in the address string in real time    4. List of properties matching the user input description must be displayed    5. Users shall be able to browse filtered properties by scrolling vertically 5. Home page shall include a navigation bar at the bottom which includes options to:    1. Return to the home page which shall navigate users to the home page (Figure 3.1.2.1)    2. Add listings which shall navigate users to the add property page (Figure 3.1.5.1)    3. View profile which shall navigate users to the profile page (Figure 3.1.6.1) 6. The house icon that is labeled “Home” shall be highlighted to indicate that users are on the Home page |

| **3.1.3: Sign Up Page** |
| --- |
| Figure 3.1.3.1 and 3.1.3.2: Sign Up Page 1 and Sign Up Page 2 |
| 1. Users shall be able to register an account by filling up the fields below:    1. Email address    2. Password       1. Password field should have the option to uncensor the text entered 2. Users shall click on ‘continue’ to navigate to Sign Up Page 2 (Figure 3.1.3.2) 3. Users shall be able to register an account by filling up the fields below:    1. Display Name    2. Age 4. Users shall be able to register for an account by clicking on the register button at the bottom of the page    1. Register button shall only register if all the above field mentioned in point 1 and 3 above is filled up       1. Failure to fill up any of the fields shall result in generation of error message    2. Register button shall navigate to the Home page (Figure 3.1.2.1) 5. Sign up page shall include an option to login at the bottom of the page to allow existing users to key in their log in details in the login page.    1. Login button will navigate users back to the login page (Figure 3.1.1.1) |

| **3.1.4: Likes Page** |
| --- |
| Figure 3.1.4.1: Liked page |
| 1. Liked page shall allow users to scroll vertically to browse liked properties 2. Users shall be able to remove a listing from likes by selecting the heart icon button labeled “Remove from likes” 3. Users shall be able to select a property listing to view the details of the selected listing    1. Clicking on the listing will navigate users to the Individual Listing page of the selected listing (Figure 3.1.11.1) 4. Liked page shall include a navigation bar at the bottom of the page as per the home page in 3.1.2    1. The heart icon that is labeled “Likes” shall be highlighted to indicate that users are on the Likes page |

| **3.1.5: Add Property Page** |
| --- |
| Figure 3.1.5.1: Add property page |
| 1. Users shall be navigated to the first page of add property page (Figure 3.1.5.1) when they click on the ‘+’ button on the navigation bar 2. ‘+’ button on the navigation bar shall be highlighted red to indicate that users are currently on the add property page 3. Users shall be able to scroll vertically on the add property page 4. Users shall be able to upload a photo from their device gallery when they click on the add photo field 5. Keying in the postal code shall fill up the Block and Street fields automatically 6. Users must fill in all fields in the add property page    1. Failure to fill in any of the field shall lead to a prompt displaying “\*field\* Cannot Be Empty!”       1. Field will be replaced by the name of the empty field. 7. Users can press the add property button at the bottom of the page to submit their listing    1. User can only press the add property button upon filling up all the fields in both the pages    2. Upon pressing add property, users shall be prompted with two buttons, “Confirm Submit” and “Cancel”       1. “Cancel” button shall navigate close the prompt       2. “Confirm Submit” button shall add the new listing to the My Properties page (Figure 3.1.7.1)       3. “Confirm Submit” button shall add the new listing to Home Page (Figure 3.1.2.1) |

| **3.1.6: Profile Page** |
| --- |
| 3.1.6.1: Profile Page |
| 1. “Profile” button in the navigation bar below shall be highlighted red to indicate that users are currently on the profile page 2. Profile page shall display:    1. User profile photo    2. Username of the user 3. Profile page shall have an Edit Profile option    1. Edit profile option will navigate users to Edit Profile page (Figure 3.1.8.1) 4. Profile page shall have a Change Password option    1. Change password option will navigate users to Change Password Page (Figure 3.1.9.1) 5. Profile page shall have a My Properties option    1. My Properties option will navigate users to My Properties Page (Figure 3.1.7.1) 6. Profile page shall have a log out option    1. Log out option shall navigate users to login page (Figure 3.1.1.1) |

| **3.1.7: My Properties Page** |
| --- |
| Figure 3.1.7.1: My Properties Page |
| 1. Users shall be able to scroll vertically on My Properties page 2. If users do not have any currently listed properties, My Properties page will display “No property listings yet!” 3. My Properties page shall display a list of properties listed by the user    1. Listings shall display:       1. Property Name       2. Price of the property       3. Number of bedroom       4. Number of bathroom       5. Property Address    2. Listings shall have the option to delete listing       1. Upon clicking on the ‘Delete’ button , users will be prompted with two options, ‘Confirm Delete’ and ‘Cancel’          1. ‘Confirm Delete’ button will remove the listing from My Properties page and Home Page          2. ‘Cancel’ button will close the prompt    3. Listings shall have option to edit listing       1. Upon clicking on the ‘Edit’ button, users will navigate to the Edit Listing Page (Figure 3.1.11.1) |

| **3.1.8: Edit Profile Page** |
| --- |
| Figure 3.1.8.1: Edit Profile Page |
| 1. Edit Profile page shall include an arrow at the top left corner of the page to allow users to navigate back to the Profile page (Figure 3.1.6.1) 2. Users shall be able to scroll vertically on the Edit Profile page 3. Users shall be able to choose the fields they want to edit. Users can edit:    1. Username    2. Profile photo    3. Email Address    4. Phone Number    5. Residential Address    6. Whether they are first time applicants       1. A dropdown bar shall display available options    7. Household Income    8. Marital Status       1. A dropdown bar shall display available options    9. Age    10. Citizenship 4. Users shall be able to save changes by pressing the ‘Save Changes’ button at the bottom of the page |

| **3.1.9: Change Password Page** |
| --- |
| Figure 3.1.9.1: Change Password Page |
| 1. Change Password page shall include an arrow at the top left corner of the page to allow users to navigate back to the Profile page (Figure 3.1.6.1) 2. Users are required to key in their email in the ‘Email’ field 3. Users shall press the ‘Send Reset Link’ to change their password    1. ‘Send Reset link’ button can only be pressed when the “Email’ field contains a valid email address    2. Users shall change password from the link sent to the email address keyed in |

| **3.1.10: Edit Property Page** |
| --- |
| 3.1.10.1: Edit Property Page |
| 1. Edit Property page (Figure 3.1.10.1) shall include an arrow at the top left corner of the page to allow users to navigate back to the My Property page (Figure 3.1.7.1) 2. Edit Property page (Figure 3.1.10.1) shall allow users to scroll vertically 3. Users shall be able to edit in Edit Property Page (Figure 3.1.10.1):    1. Property Address    2. Property Image    3. Description    4. Dimensions    5. Neighbourhood    6. Leftover Lease    7. Number of bedroom    8. Price    9. Property Name 4. Users shall be able to save changes at the bottom of Edit Property Page by clicking on the ‘Save Changes’ button (Figure 3.1.10.1) |

| **3.1.11: Individual Listing** |
| --- |
| Figure 3.1.11.1: Individual Listing |
| 1. Individual Listing page (Figure 3.1.11.1) shall include an arrow at the top left corner of the page to allow users to navigate back to the Home page (Figure 3.1.2.1) 2. Users shall be able to scroll vertically on the Individual Listing page 3. Individual Listing page shall have a text box at the bottom of the page that shows the price of the house    1. Textbox shall remain at that position as users scroll vertically 4. Individual Listing page shall display attributes of the listed property. This includes:    1. Address    2. Neighbourhood    3. Availability    4. Number of bedroom    5. Number of bathroom    6. Dimensions    7. Remaining lease    8. Description 5. Individual listing shall display an interactive map that shows the location of the property    1. Users shall be able to move the map in all directions    2. Users shall be able to zoom in and out of the map    3. Users shall be able to open the map in Google Map 6. Individual Listing page shall display the grant calculator at the bottom of the page.    1. Grant calculator shall include the names of grants awarded and the amount awarded based on user’s personal information       1. If users have not filled up their personal information in their profile, grant calculator will prompt users to update the required information          1. There will be an ‘Edit Profile’ button which navigates users to the Edit Profile page (Figure 3.1.8.1) |

| **3.1.12: Compare Page** |
| --- |
| Figure 3.1.12.1: Compare Page |
| 1. Compare page (Figure 3.1.12.1) shall include an arrow at the top left corner of the page to allow users to navigate back to the Individual Listing page (Figure 3.1.10.1) 2. Compare page shall retrieve all properties’ prices based on the location and number of rooms for the past 3 years 3. Compare page shall calculate the average price of the listings matching the property’s location and bedroom number 4. Compare page shall 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    1. List view of the resale listings shall include the neighborhood, number of rooms and the price of the housing |

| **3.1.13: Send Email Page** |
| --- |
| Figure 3.1.13.1: Send Email page |
| 1. Send email page shall display a text field to allow the user to input a message that he/she intends to send to the seller 2. Users shall press the send button to send an email containing the message input to the seller    1. Send email page shall display a message saying “No email associated with seller!” if there is no email associated with the individual listing |

## **Hardware Interfaces**

The application does not require any hardware to run on.

## **Software Interfaces**

**Firebase: Backend as a Service (BaaS)**

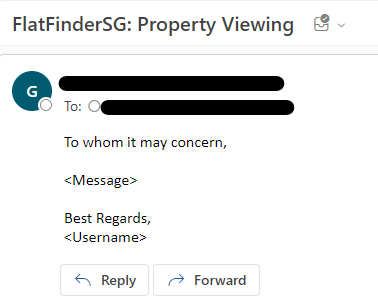
Firebase is a Backend-as-a-Service (BaaS). It provides developers with a variety of tools and services to help them develop quality applications, grow their user base, and gain profit. It is built on Google’s infrastructure. Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents.

**Data.Gov Resale Flat Prices**

The live resale flat prices enables us to obtain real-time updates on the prices of resale flats in various neighborhoods in Singapore to provide users with up-to-date information when utilizing the compare feature in the application.

## **Communications Interfaces**

**3.4.1. Email Feature**

****

Upon clicking on the “Send” button on the Send Email page, an Email will be sent to the Email address associated with the individual listing. The contents of the email will be as per message typed in the Send Email page. Email will be titled FlatFinderSG: Property Viewing and signed off with the username of the sender.

# **System Features**

**4.1: Register - ACC1**

| 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 |

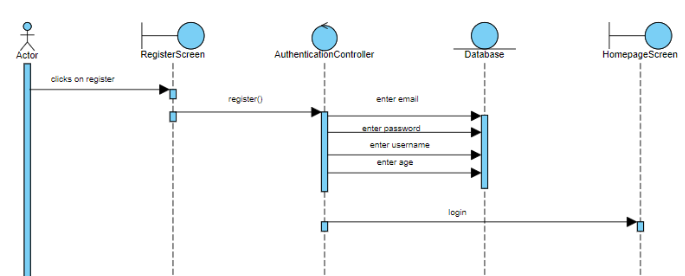
****

Figure 4.1.1: Register sequence diagram

**4.2: Login - ACC2**

| 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 |

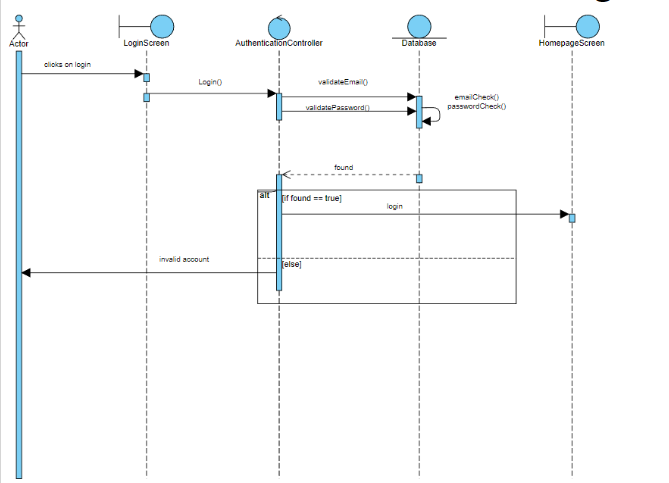


Figure 4.2.1: Login sequence diagram

**4.3: View Profile - PF1**

| 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 |

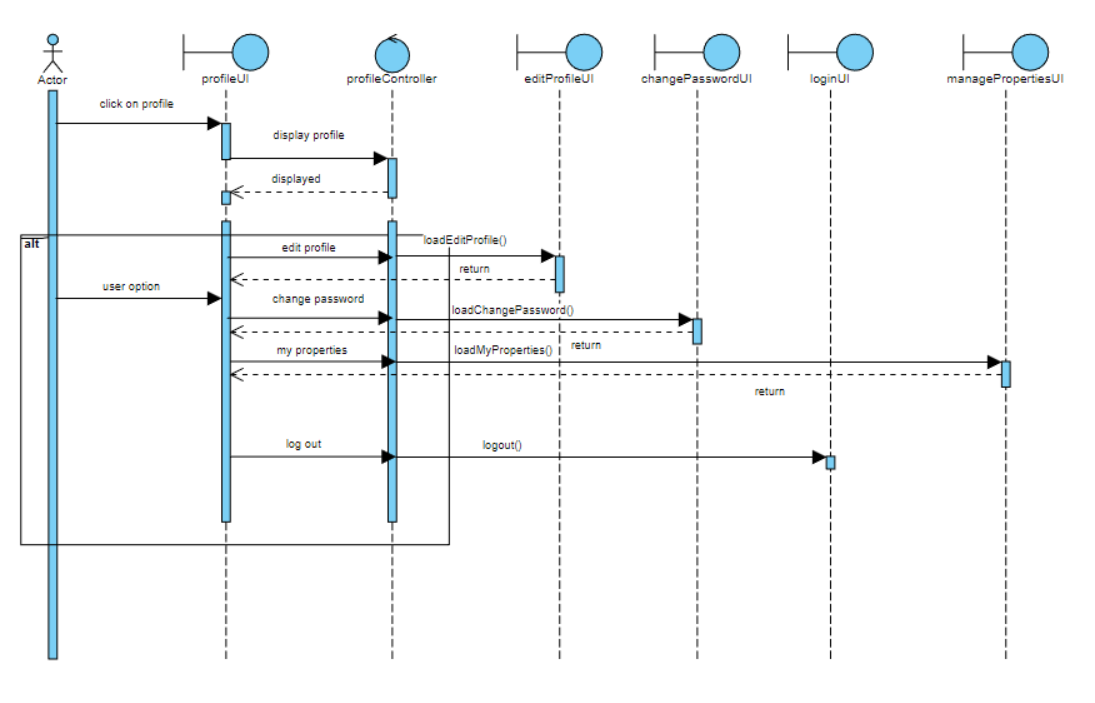


Figure 4.3.1: View Profile sequence diagram

4.4: Manage My Properties - PF3

| 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 |

4.5: Edit Listing - MP1

| 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 |

**4.6: Remove Listing - MP2**

| 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 |

**4.7: Edit Profile - PF2**

| 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 |

**4.8: Change Password - PF3**

| 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 |

**4.9: Send Email - SE1**

| Use Case ID: | SE1 | | |
| --- | --- | --- | --- |
| Use Case Name: | Send Email | | |
| Created By: | Jabez Ng | Last Updated By: | Bryan |
| Date Created: | 19/03/2023 | Date Last Updated: | 16/04/2023 |

| 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 |

**4.10: View Likes - VL1**

| 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 |

**4.11: Add Listing - AL1**

| Use Case ID: | AL1 | | |
| --- | --- | --- | --- |
| Use Case Name: | Add Listing | | |
| Created By: | Wei Hao | Last Updated By: | Bryan |
| Date Created: | 6/2/2023 | Date Last Updated: | 16/4/2023 |

| 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. The system prompt user to upload property picture using “Add photo” button 2. The system prompts the user to input the fields Property Name, Postal Code, Block, Street, Level, Unit, Neighbourhood, Bedrooms, Lease, dimensions, price, description 3. The user can choose to fill in postal code and search, system will automatically fill in Block and Street fields 4. The user click on “Add Property” 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 Property Name is left empty   1. The system displays “Please enter property name” 2. The system returns to Step 1   AL1.AF.2 Postal Code is left empty   1. The system displays “Please enter an address” 2. The system returns to Step 1   AL1.AF.3 Block is left empty   1. The system displays “Please enter an address” 2. The system returns to Step 1   AL1.AF.4 No Images are uploaded   1. The system displays “Please Upload Photo!” 2. The system returns to Step 1   AL1.AF.5 Number of bedrooms are left empty   1. The system displays “Please enter an number of bedroom” 2. The system returns to Step 1   AL1.AF.6 Dimensions is left empty   1. The system displays “Please enter dimensions” 2. The system returns to Step 1   AL1.AF.7 Lease is left empty   1. The system displays “Please enter lease” 2. The system returns to Step 1   AL1.AF.8 Neighbourhood is left empty   1. The system displays “Please enter neighbourhood” 2. The system returns to Step 1   AL1.AF.9 Street is left empty   1. The System displays “Please enter street” 2. The system returns to Step 1   AL1.AF10 Property name is left empty   1. The system displays “Please enter property name” 2. The system returns to Step 1   AL1.AF11 price is left empty   1. The system displays “Please enter price!” 2. The system returns to Step 1   AL1.AF12 Description is left empty   1. The system displays “Please enter description” 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 |

**4.12: View List of Properties - BP1**

| 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 |

**4.13: Add To Likes - AL1**

| 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. |

**4.14: Search Properties - SP1**

| 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 |

**4.15: Apply Filters - SP2**

| 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 |

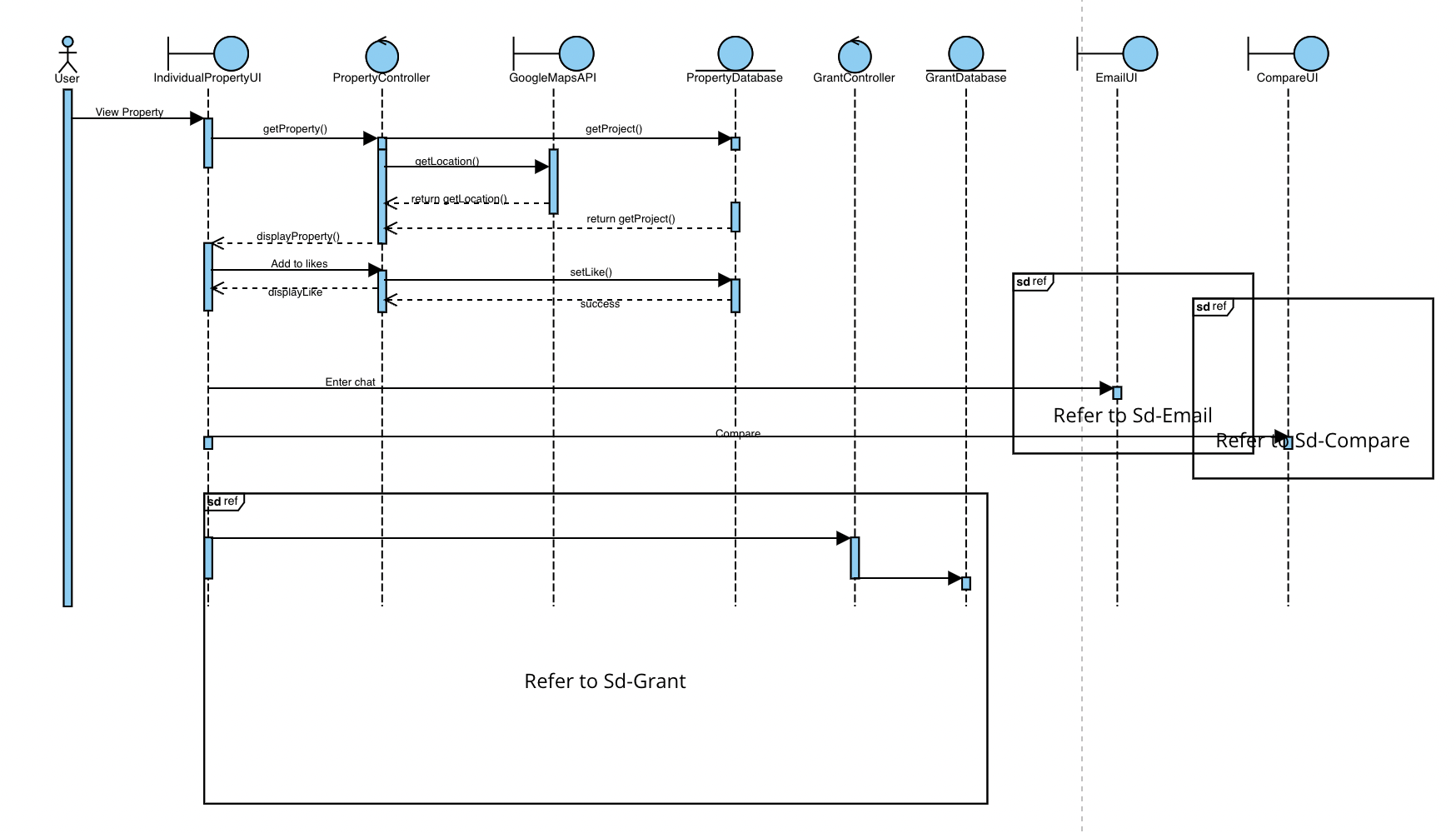
| Actor: | User |
| --- | --- |
| Description: | Allows the user to filter their search input by the number of rooms |
| Preconditions: | 1. The user is at the search page 2. The user needs to select a choice chip to filter the search results based on the number of bedrooms |
| Postconditions: | 1. The 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 |

**4.16: View Individual Property - PP1**

| Use Case ID: | PP1 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Individual Property | | |
| Created By: | Wei Hao | Last Updated By: | Bryan |
| Date Created: | 6/2/2023 | Date Last Updated: | 15/4/2023 |

| 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 selects “Edit Profile”   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. 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 3. If the “Google Maps” view is selected, the user will be able to view the property location on Google Maps 4. The user will be able to utilize the Grant Calculator feature to view eligible grants and awarded amount 5. The user will be able to edit profile information by pressing “Edit Profile” button 6. 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 |

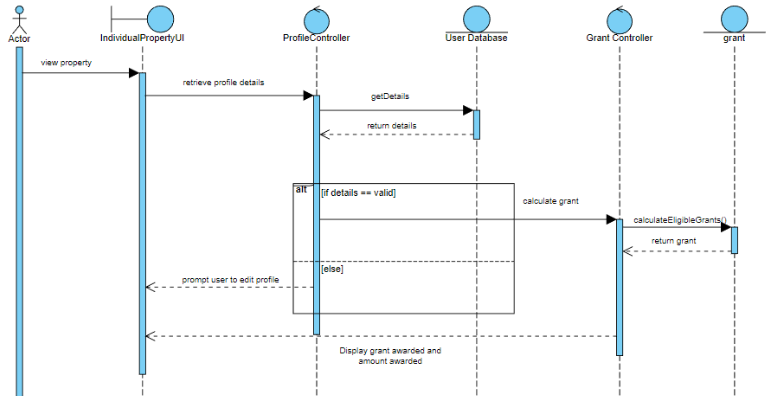
4.16.1: View Individual Property sequence diagram

****

**4.17: Grant Calculator - PP2**

| Use Case ID: | PP2 | | |
| --- | --- | --- | --- |
| Use Case Name: | Grant Calculator | | |
| Created By: | Wei Hao | Last Updated By: | Bryan |
| Date Created: | 6/2/2023 | Date Last Updated: | 14/4/2023 |

| Actor: | User |
| --- | --- |
| Description: | Allows the user to determine what 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 “Individual Property Page” 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 profile 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 “Not Applicable” in grant amount 2. The system will display “Not Applicable” in type of grants 3. The user have to click on “Edit Profile” button to fill in user profile details 4. The system will recalculate type of grant and grant amount 5. The system will display the type of grant 6. The system will display grant amount |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | User is connected to the Internet |
| Notes and Issues: | NIL |



4.17.1: Grant Calculator sequence diagram

4.18: View Address Location - PP3

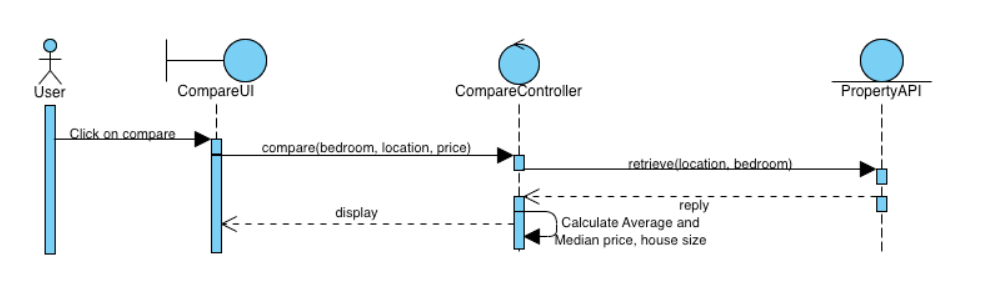
| Use Case ID: | PP3 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Address Location | | |
| Created By: | Wei Hao | Last Updated By: | Bryan |
| Date Created: | 6/2/2023 | Date Last Updated: | 14/4/2023 |

| 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 display pinned location of property address 2. The user is able to zoom in and out and adjust the orientation of the window 3. The user click on pinned location of property address to view on Google Maps |
| 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 |

4.19: Compare Properties Feature - PP4

| 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 |



4.19.1: Compare Properties Feature sequence diagram

# **Other Nonfunctional Requirements**

## **Performance Requirements**

1. 95% of users must be able to login within 10 seconds after pressing the login button.
2. Email should be sent out within 1 minute. This includes:
   1. Email sent from New Users page to verify new user
   2. Email sent from Change Password page to change password
   3. Email sent from Send Email page to contact seller

## **Safety Requirements**

1. Application must not ask for the personal information of users
2. Application must not store any transaction details, including but not limited to the address of recently sold houses and price of sale
3. Application must not store users location or track users location for any other purpose other than the aforementioned functionalities.

## **Security Requirements**

1. Password is to be masked with ‘.’
2. All email sent out from the application must be to the correct recipient
   1. For emails sent for the purpose of verification in the New Users page, emails must be sent to the same email in the text box
   2. For emails sent for the purpose of changing password in the Change Password page, emails must be sent to the same email in the text box
   3. For emails sent for the purpose of contacting sellers in the Send Email page, emails must be sent to the same email associated with the listing.

## **Software Quality Attributes**

1. **Reliability**

* 1. The application must function correctly and reliably under normal and exceptional conditions and should be able to recover from errors quickly

* 1. The application must be fully operational before launch

* 1. The application must not crash under normal and exceptional conditions

* 1. The application must undergo regular checks for bugs and fixes for bugs shall be carried out as soon as possible

* 1. The application must undergo regular reviews to upgrade and optimise codes to ensure efficiency

1. **Scalability**

* 1. The application shall be able to handle increasing data, user, and transaction without facing significant performance degradation

* 1. The application shall be able to integrate new functionalities without disrupting existing functions

* 1. The application shall allow for vertical and horizontal scaling. That is to increase the amount of database and storage capabilities for each database when required.

1. **Compatibility**

* 1. The application shall be able to work on different systems to carry out its functionalities correctly.
  2. The application shall be able to run on different platforms and operating systems and must be able to adapt to the platform to carry out its functionalities

1. **Efficiency**

* 1. The application is to have acceptable processing time and resource allocation, which will determine the system’s speed and responsiveness

* 1. The application shall have consistent, acceptable processing time when carrying out functionalities. This includes but is not limited to:
     1. Search functions
     2. Retrieving data from database
     3. Loading and displaying data on user interface

## **Business Rules**

1. All individuals shall be allowed to use the application for its stated function, which is to act as a marketplace for resale housing.

1. All sales are subjected to Singapore’s laws that govern real estate transactions.

1. The application and the organization behind the application do not borne any legal responsibility with regard to fraudulent activities. Users shall carry out their own due diligence and report to relevant authorities in such cases.

# **Other Requirements**

## Reuse Objectives

We utilize service-oriented systems by linking shared services within our application that are externally provided. When users want to view the location of the property listing, they will be brought to the Google Maps display on the application. Users can perform map navigation on the maps display as well as zoom-in and zoom-out functions.

* 1. **MVC Framework**

The application interface should follow the Model-View-Controller (MVC) model for rendering and modeling data objects. The interface must be able to connect to a database (Firebase in our case) to retrieve data such as user and property listing data. Source and destination formats for the data include JavaScript Object Notation (JSON).

**Appendix A: Glossary**

**Data Dictionary**

| **Term** | **Definition** |
| --- | --- |
| User | Users of the application with the intent to either purchase or sell properties |
| Login | The process whereby a User enters their Account’s username and password in the application to access its features, given that they have an existing Account |
| Register | The process whereby a User without an existing Account creates one by entering their desired Username, Email Address, Address, Phone Number and Password |
| Account | A combination of User details involving: Username, Email Address, Address, Phone Number, Password, Profile photo, Address location, Single/Couple/Family, First Time Applicants or Not, Household Income Ceiling, Address of Parents, Age, Citizenship, Ownership in other properties |
| Homepage | Refers to the page that users see right after they log into their account. Users will be able to access the search bar, browse through property listings and utilize the navigation bar |
| Address | Singaporean address location consisting of block, road, avenue etc. |
| Property Listing | A listing on the mobile application that provides information about a property for sale, including details like its Description, Price (psf), Address, Image, Number of bedrooms, Number of bathrooms, Dimensions (sqft), Leftover lease, and Neighbourhood |
| Grants | Refers to the grants available under CPF housing grants scheme or any resale grants offered by HDB under their website. |
| Loan Repayment | The process of making payments to repay a loan, in the case, of the purchase of a property |
| Income Ceiling | The maximum income limit set by the Singapore government for eligibility for certain housing grants |
| Browse | The process of searching through the properties listed on the homepage or the various chats that are listed in the chats page |
| Dimensions | The size of a property, measured in square feet |
| Leftover Lease | Refers to the amount of lease remaining on a property, measured in years |
| Map | Visual representation of the location of properties, displayed on Google Maps |
| Navigation Bar | The bar or menu on the mobile application that allows users to navigate to different parts of the application with ease (Home, My Likes, Add Listing, Profile) |

**User**

| **Attribute Name** | **Description** | **Required** | **Format** |
| --- | --- | --- | --- |
| *\_uid* | User id | Yes | String |
| *\_displayName* | User name | Yes | String |
| *\_emailAddress* | User email | Yes | String |
| *\_age* | User age | Yes | Int |
| *\_citizenship* | User citizenship | Yes | String |
| *\_martial* | User martial status | Yes | String |
| *\_applicationStatus* | User application status | Yes | String |
| *\_applicationType* | User application type | Yes | String |
| *\_averageMonthlyHousehold* | User average monthly household | Yes | Int |
| *\_firstTime* | User first-time status | Yes | Bool |

**Listing**

| **Attribute Name** | **Description** | **Required** | **Format** |
| --- | --- | --- | --- |
| *\_id* | Listing id | Yes | String |
| *\_address* | Address of listing | Yes | String |
| *\_description* | Description of listing | Yes | String |
| *\_dimension* | Dimensions of listing | Yes | Int |
| *\_lease* | Lease period of listing | Yes | Int |
| *\_neighbourhood* | Neighborhood of listing | Yes | String |
| *\_numOfBedroom* | Number of rooms | Yes | String |
| *\_numOfBathroom* | Number of bathrooms | Yes | String |
| *\_price* | Price of listing | Yes | Double |
| *\_propertyName* | Name of property | Yes | String |

**Grant**

| **Attribute Name** | **Description** | **Required** | **Format** |
| --- | --- | --- | --- |
| *\_type* | Type of grant provided | Yes | String |
| *\_firstApplicant* | Name of first applicant | Yes | String |
| *\_secondApplicant* | Name of second applicant | Yes | String |
| *\_incomeCeiling* | Income ceiling of applicants | Yes | Int |
| *\_firstApplicantCitizenship* | Citizenship of first applicant | Yes | String |
| *\_secondApplicantCitizenship* | Citizenship of second applicant | Yes | String |
| *\_age* | Age of applicant | Yes | Int |

**Resale**

| **Attribute Name** | **Description** | **Required** | **Format** |
| --- | --- | --- | --- |
| *\_town* | Town that resale flat belongs to | Yes | Town |
| *\_flatType* | Flat type of resale flat | Yes | FlatType |
| *\_fullCount* | Full count of the resale flat | Yes | String |
| *\_flatModel* | Resale flat model e.g Model A | Yes | FlatModel |
| *\_floorAreaSqm* | Floor area of resale flat | Yes | String |
| *\_streetName* | Street name of the resale flat | Yes | StreetName |
| *\_resalePrice* | Price of the resale flat | Yes | String |
| *\_rank* | Rank of the resale flat | Yes | Double |
| *\_month* | Month and year of the resale flat data | Yes | Month |
| *\_remainingLease* | Remaining lease of the resale flat | Yes | RemainingLease |
| *\_leaseCommenceDate* | Lease commencement date of the resale flat | Yes | String |
| *\_storeyRange* | Storey range of the resale flat | Yes | StoreyRange |
| *\_id* | Id number of the resale flat | Yes | Int |
| *\_block* | Block of the resale flat | Yes | String |

**Appendix B: Analysis Models**

**System Architecture Diagram**

(See System Architecture.png in System Architecture folder)

**Dialog Map**

(See Dialog Map.png in Dialog Map folder)

**Use Case Diagram**

(See Use Case Diagram.png in Use Case Model folder)

**Class Diagram**

(See Class Diagram.png in Class Diagram folder)

**Appendix C: To Be Determined List**

There are no TBD items in our final submission and all TBD items have been cleared.