Lab 1 Deliverables
Sweet Home Finder
SCSB T2

1. Target User

Our target users of this application are Singaporean residents who are searching for a suitable home location near specified amenities (e.g. MRT stations under construction) in both mature and developing estates.

2. Functional Requirements

1. The user must be able to register for an account in the system.

    1.1. The system must display text fields for the user to enter his information.

        1.1.1. A text field for the username must be included.

        1.1.2. A text field for the email address must be included.

        1.1.3. A text field for the password must be included.

    1.2. The system must display a checkbox stating, “I agree to the Terms of Use and Private Policy”.

    1.3. The user must fill in all the text fields and check the checkbox before clicking the “Sign Up” button.

    1.4. The system must verify the text fields filled in by the user before creating the account.

        1.4.1. The username must be unique for all users of the system.

        1.4.2. The email address used must be new within the system.

        1.4.3. The email address used must be valid in receiving emails.

            1.4.3.1. The system must send a One Time Password (OTP) to the user’s email address.

        1.4.4. The password used must be at least 8 characters in length, with 1 upper-case letter and 1 lower-case letter.

    1.5. The system must store the user’s information in the user database.

    1.6. The system must display a message if the registration is successful.

    1.7. The system must log the user into the system.

2. The user must be able to log into the system.

    2.1. The system must display text fields for the user to enter his information.

        2.1.1. A text field for the username must be included.

        2.1.2. A text field for the password must be included.

    2.2. The user must fill in all the text fields before clicking the “Login” button.

    2.3. The system must verify that the fields filled in by the user are within the user database.

        2.3.1. The system must be able to find the username inputted by the user.

        2.3.2. The system must verify that the password entered matches the password associated with the username.

    2.4. The system must log the user into the system.

3. The user must be able to view an interactive map of Singapore.

    3.1. The system must display a map of Singapore with the available properties highlighted with distinct markers.

    3.2. The user must be able to click on a property to view detailed information on it.

    3.3. The system must display information for each property displayed.

        3.3.1. The system must display the name of the property.

        3.3.2. The system must display the price of the property.

        3.3.3. The system must display the address of the property.

    3.4. The user must be able to zoom in and zoom out of the map.

    3.5. The user must be able to pan through the map.

4. The user must be able to search for properties based on keywords.

    4.1. The user must be able to click on a search bar.

        4.1.1. The user must be able to fill text in the search bar.

    4.2. The system must return a list of properties that are relevant to the keywords entered in the search bar.

    4.3. The system must display information for each property displayed.

4.3.1. The system must display the name of the property.

4.3.2. The system must display the price of the property.

4.3.3. The system must display the address of the property.

5. The user must be able to select a specific property to view more details of.

5.1. The user must be able to view nearby amenities to the selected property.

5.1.1. The system must display amenities within a 500-metre radius from the selected property.

5.1.1.1. The system must display a route from the selected property to a nearby amenity.

5.1.1.2. The system must display the time taken to travel from the selected property to a nearby amenity.

5.2. The user must be able to search for routes from the selected property to a location inputted by the user.

5.2.1. The system must display the time taken and the transportation used for the route.

5.2.2. The system must be able to save the inputted location as a saved location.

5.3. The user must be able to bookmark the selected property.

5.3.1. The system must store the selected property within a database.

3. Non-functional Requirements

Performance:

1. The system must be able to successfully add an account to the database within 15 seconds after successful registration.
2. The system must be able to retrieve and display the search results of the user within 20 seconds.

Usability:

1. The system must be able to display an FAQ message of the functions the user can use.

Security:

1. The system must use a hashing algorithm to encrypt the user's password within the user database.

Reliability:

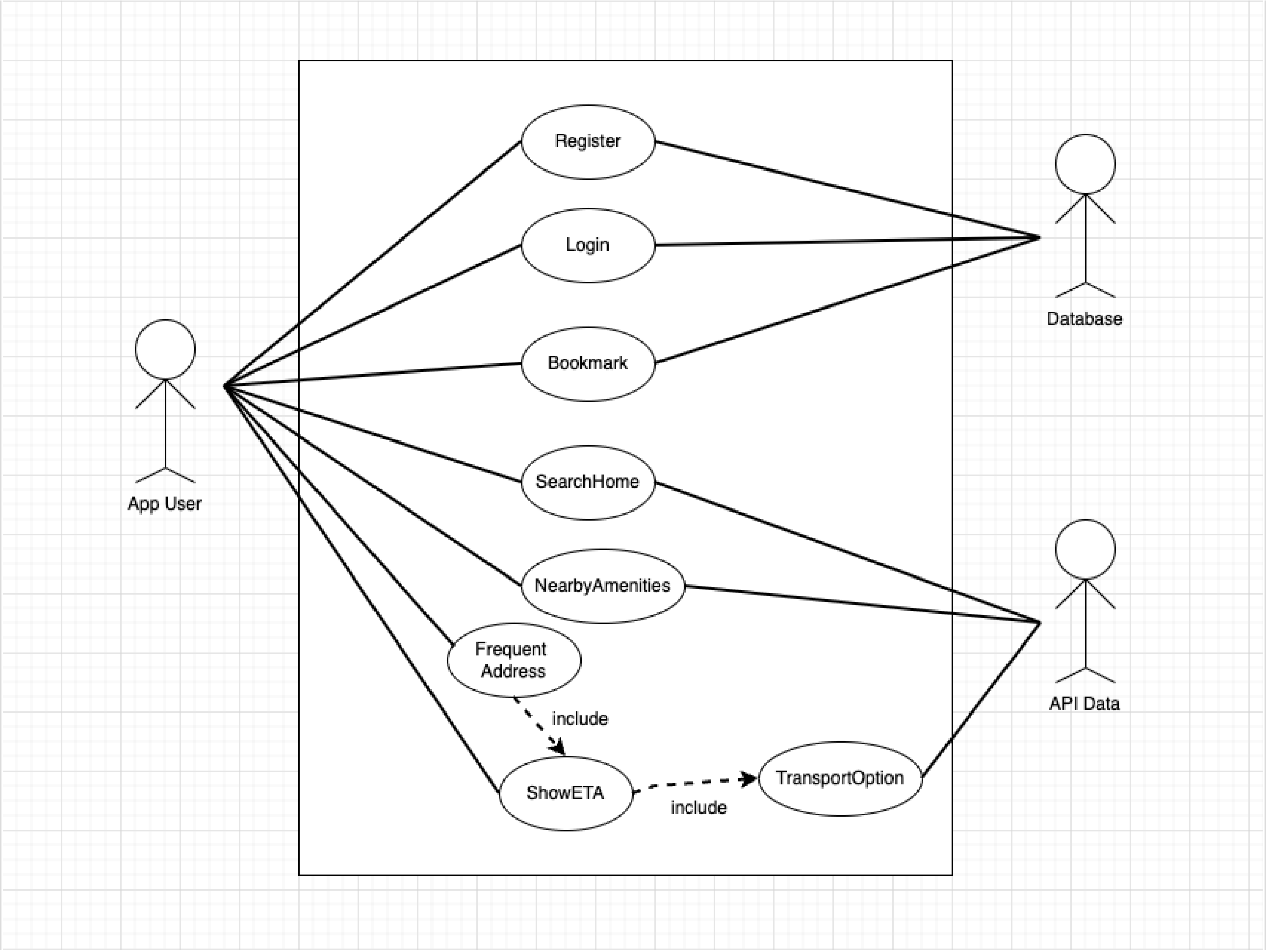
1. The system must not be down for more than 2 hours in 1 month.

4. Data dictionary

User	An individual who has created and logged into their account to use Sweet Home Finder. The user will then be granted permission to use the services provided by the application. Users will be able to view the properties listed and also bookmark them for future reference. Users will be able to view the distance and travel time from the current property to nearby amenities and their saved locations.
Bookmark	A list which contains all the users property that they have liked and saved for them to refer to in the future.
One-Time Password (OTP)	A 4 digit combination which is sent to the user registered email address to act as a second layer of protection. In the event of the user forgetting their account password, an OTP will be sent to them.
Username	A unique identifier created by the user for future log in.
Property	A resale property such as HDB flats, condos and apartments that are available for purchase.
Saved Location	A location that the user frequently travels to such as their parents home or their work. The user will be given the option to input their saved locations and see the ETA from the property they are viewing to these locations.
Estimated Time of Arrival (ETA)	The predicted time that it will take a user to one of their saved locations or nearby amenities.

5. Use case model

5.1. Use Case Diagram



5.2. Use Case Descriptions

Use Case ID:	001		
Use Case Name:	Register		
Created By:	Tan Wu Ji	**Last Updated By:**	Tan Wu Ji
Date Created:	11/2/2024	**Date Last Updated:**	11/2/2024

Actor:	App User, Database
Description:	App User can register for an account which will be stored in the Database
Preconditions:	1. Database must be up and online 2. App User must be connected to Internet
Postconditions:	App User has successfully registered an account with a unique username and password. OR App User is notified of the unsuccessful registration of an account
Priority:	
Frequency of Use:	
Flow of Events:	1. At the home page, App User clicks on “Sign up” and is redirected to the registration page. 2. App User inputs a valid email, a unique username, a password with at least 8 characters, 1 upper-case letter and 1 lower-case letter and the repeated password in the respective fields in the registration page. 3. App User checks the checkbox of “I agree to the Terms of Use and Private Policy” and clicks on ” “Sign Up” to register and create an account. 4. The system verifies the username is unique and the password satisfies the constraints. 5. App User inputs a One Time Password (OTP) which is sent to his/her email inbox for further verification. 6. The system stores App User’s information in the Database. 7. App User is notified that the registration is successful.

Alternative Flows:	AF-S2: App User left input field(s) blank. 1. When the App User clicks on “Sign Up”, the system displays the message “Please fill in all input fields to register for an account!” above the registration page. 2. The system returns to Step 2 and waits for the App User inputs. AF-S3: App User did not check the checkbox of “I agree to the Terms of Use and Privacy Policy”. 1. When the App User clicks on “Sign Up”, the system displays the message “Please tick the checkbox for acknowledging the Terms of Use and Privacy Policy!” above the registration page. 2. The system returns to Step 2 and waits for the App User inputs. AF-S4: App User inputs a taken username. 1. The system displays the message “Username has been taken. Please input a new username!” above the registration page. 2. The system returns to Step 2 and waits for the App User inputs. AF-S4: App User inputs a password that does not satisfy the given requirements. 1. The system displays the message “Password must contain at least 8 characters, 1 upper-case letter and 1 lower-case letter!!!” above the registration page. 2. The system returns to Step 2 and waits for the App User inputs. AF-S4: App User inputs mismatched passwords. 1. The system displays the message “Passwords do not match! Please check again!” above the registration page. 2. The system returns to Step 2 and waits for the App User inputs. AF-S5: App User inputs an incorrect OTP. 1. The system displays the message “Incorrect OTP! Please try again!” above the registration page. 2. The system returns to Step 5 and waits for the App User inputs.

Exceptions:	EX-1: App User did not receive the OTP in his/her email inbox. 1. The “Resend another OTP” button is available after 60 seconds. 2. App User clicks on the “Resend OTP” button to resend another OTP to his/her email inbox. 3. The system returns to Step 5 and waits for the App User inputs.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	002		
Use Case Name:	Login		
Created By:	Tan Wu Ji	**Last Updated By:**	Tan Wu Ji
Date Created:	11/2/2024	**Date Last Updated:**	11/2/2024

Actor:	App User, Database
Description:	App User can login to his/her account with the correct username and password which are stored in the Database.
Preconditions:	1. The Database must be up and online. 2. App User must be connected to the Internet. 3. App User has a registered account.
Postconditions:	App User has successfully logged into his/her account. OR App User is notified of the unsuccessful login to his/her account.
Priority:	
Frequency of Use:	
Flow of Events:	1. At the home page, App User clicks on “Log in” and is redirected to the login page. 2. App User inputs his/her username and password. 3. App User clicks on “LOGIN”. 4. The system verifies the username and password with the Database. 5. Once the information is verified, the App User is redirected to the home page.
Alternative Flows:	AF-S2: If the App User inputs an incorrect username or password. 1. When the App User clicks on “LOGIN”, the system displays the message “Invalid username and/or password!” above the registration page. 2. The system returns to Step 2 and waits for the App User inputs.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	003		
Use Case Name:	SearchHome		
Created By:	Tan Wu Ji	**Last Updated By:**	Tan Wu Ji
Date Created:	11/2/2024	**Date Last Updated:**	11/2/2024

Actor:	App User, API data
Description:	App User will be able to search the properties with keywords such as location or the property's name. The system will retrieve information from the API data and display the result for the App User.
Preconditions:	1. App User is connected to the Internet. 2. App User registered for an account with the Database. 3. App User has logged in to his/her account.
Postconditions:	App User obtained a list of searched properties based on the keywords. OR App User is unable to obtain a search result based on the keywords.
Priority:	
Frequency of Use:	
Flow of Events:	1. At the home page, the App User clicks on the search bar, types a keyword in the search bar and clicks on the "Search" icon/button. 2. The system retrieves a list of properties from the API data and searches for the relevant properties based on the keyword. 3. The system displays information of the relevant properties such as property name, price and address.
Alternative Flows:	AF-S1: App User inputs nothing and clicks on the search icon. 1. The system displays a list of 20 random properties. 2. The system returns to Step 1 and waits for the App User actions.
Exceptions:	EX-1: The keyword input by the App User does not match any searched items. 1. The system displays the message "No relevant properties are found!". 2. The system returns to Step 1 and waits for the App User actions.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	004		
Use Case Name:	Bookmark		
Created By:	Tan Wu Ji	**Last Updated By:**	Tan Wu Ji
Date Created:	11/2/2024	**Date Last Updated:**	11/2/2024

Actor:	App User, Database
Description:	App User can bookmark their interested properties. The system saves the bookmarked properties in the database. App User is able to quickly find the properties in a more convenient way.
Preconditions:	1. App User is connected to the Internet. 2. App User registered for an account with the Database. 3. App User has logged in to his/her account.
Postconditions:	App User has successfully added the properties into their bookmark and saved into the database. OR App User has successfully removed the properties into their bookmark and updated the database.
Priority:	
Frequency of Use:	
Flow of Events:	1. When the App User clicks the properties, the system directs App User to the detailed information page. 2. The "bookmark" icon located at the top right corner displayed in grey colour. 3. App User clicks the "bookmark" icon. 4. The system saves the properties into the App User's database. 5. The "bookmark" icon turns yellow, indicating that this property is successfully added into the bookmark database.
Alternative Flows:	AF-S2: "Bookmark" icon is in yellow colour 1. When the App User clicks the yellow colour bookmark, the system removes the properties from the database. 2. The system changes the "bookmark" icon to grey colour, indicating that the property is successfully removed from the database. 3. The system returns to Step 2 and waits for the App User actions. AF-S2: App User does not log into his/her account 1. The system displays the message "Please log into your account to bookmark this property". 2. The system directs the App User to the login page. 3. App User uses the LOGIN use case to log into his/her account. 4. The system returns to Step 2 and waits for the App User input.
Exceptions:	
Includes:	Login
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	005		
Use Case Name:	TransportOption		
Created By:	Tan Wu Ji	**Last Updated By:**	Tan Wu Ji
Date Created:	11/2/2024	**Date Last Updated:**	11/2/2024

Actor:	App User
Description:	System prompts App User to choose the transport options such as MRT, CAR, BUS.
Preconditions:	1. App User is connected to the Internet. 2. App User is in the detailed information page.
Postconditions:	App User successfully input the transportation option
Priority:	
Frequency of Use:	
Flow of Events:	1. The system prompts App User to select the transportation option. 2. App User chooses the corresponding option and clicks “Select”. 3. The system saves the App User’s choice.
Alternative Flows:	AF-S2: App User does not choose any option and clicks “Select” 1. The system displays the message “Please choose your transportation option”. 2. The system returns to Step 1 and waits for the App User input.
Exceptions:	EX-1: App User clicks “Cancel” button 1. The system returns to the detailed information page.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	006		
Use Case Name:	ShowETA		
Created By:	Tan Wu Ji	**Last Updated By:**	Tan Wu Ji
Date Created:	11/2/2024	**Date Last Updated:**	11/2/2024

Actor:	App User, API data
Description:	The system shows the Estimated Time of Arrival (ETA) based on the App User’s input of the transportation option.
Preconditions:	1. App User is connected to the Internet. 2. App User is in the detailed information page. 3. API data is up and connected.
Postconditions:	The system successfully showed the ETA and the corresponding transport option.
Priority:	
Frequency of Use:	
Flow of Events:	1. App User clicks “Show ETA”. 2. App User uses the TransportOption use case to select the transportation option. 3. The system retrieves the property’s address from the API data. 4. The system makes API request calls to calculate the ETA between the addresses. 5. The system displays the ETA.
Alternative Flows:	AF-S4: The system failed to calculate the ETA 1. The system displays the message “Calculating the ETA again…”. 2. The system returns to Step 3.
Exceptions:	EX-1: The system failed to calculate ETA in 30 seconds 1. The system uses the InputAddress use case to wait for App User to input the address again.
Includes:	TransportOption
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	007		
Use Case Name:	FrequentAddress		
Created By:	Boon Yi	**Last Updated By:**	Boon Yi
Date Created:	12/02/2024	**Date Last Updated:**	12/02/2024

Actor:	App User, Database
Description:	App User can input places they frequent. The system will show the ETA from their properties of interest.
Preconditions:	1. App User is connected to the internet 2. App User is in the detailed information page
Postconditions:	The system successfully showed the ETA from their property of interest and their destination.
Priority:	
Frequency of Use:	
Flow of Events:	1. App User click on the property of interest 2. System directs App User to the detailed information page 3. The "ShowETA" icon is located at the top right hand corner of the page 4. App User click on the "ShowETA" icon 5. System prompt the App User to input their destination 6. System then calculate ETA using ShowETA function 7. System display ETA calculated
Alternative Flows:	AF-S2: App User input invalid address 1. System display "Unable to find address" 2. System return to step 5 and wait for App user input
Exceptions:	EX-1: App User clicks "Cancel" button 2. The system returns to the detailed information page.
Includes:	ShowETA
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	008		
Use Case Name:	Show nearby amenities		
Created By:	Boon Yi	**Last Updated By:**	Boon Yi
Date Created:	12/02/2024	**Date Last Updated:**	12/02/2024

Actor:	App User, API data
Description:	The system will display nearby amenities from App User property of interest.
Preconditions:	1. App User is connected to the internet 2. App User is in the detailed information page
Postconditions:	The system successfully displays nearby amenities from App User property of interest.
Priority:	
Frequency of Use:	
Flow of Events:	1. App User click on the property of interest 2. System directs App User to the detailed information page 3. The "Amenities" icon is located at the top right hand corner of the page 4. App User click on the "Amenities" icon 5. System display nearby amenities from property of interest
Alternative Flows:	<u>AF-S2: No amenities nearby</u> 1. System display "Unable to find any amenities" 2. System return to step 2
Exceptions:	<u>EX-1: App User clicks "Cancel" button</u> 1. The system returns to the detailed information page.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

6. UI MockUp

The UI MockUp is shown in another document. Please refer to the UI MockUps pdf file.