SCC0006 - Software Engineering

Group Project Report (Use case description)

Table No.: C14

Name	**Matriculation number**
Chua Chong En	U2320759L
Wong Si Kei Wynette	U2210236B
Sanjana Shanmugasundaram	U2321261L
Murugan Rithika	U2323063E
Kumaresan Gayathri Devi	U2323162L
Chang Jing Xian, Max	U2221183G

Use case description

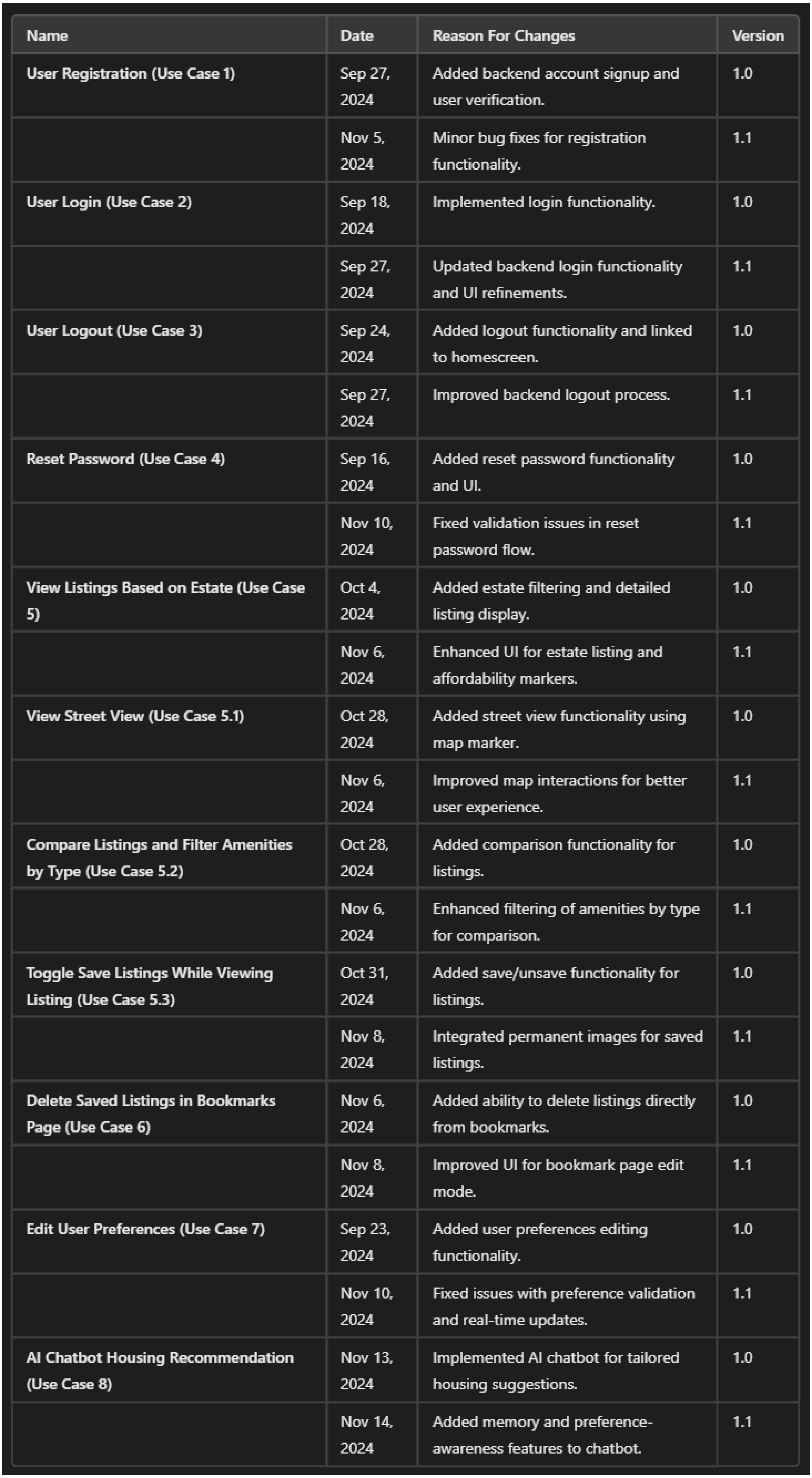
for

<Flatfinder>

Version 1.0 approved

Prepared by <Group 43>

<17 November 2024>

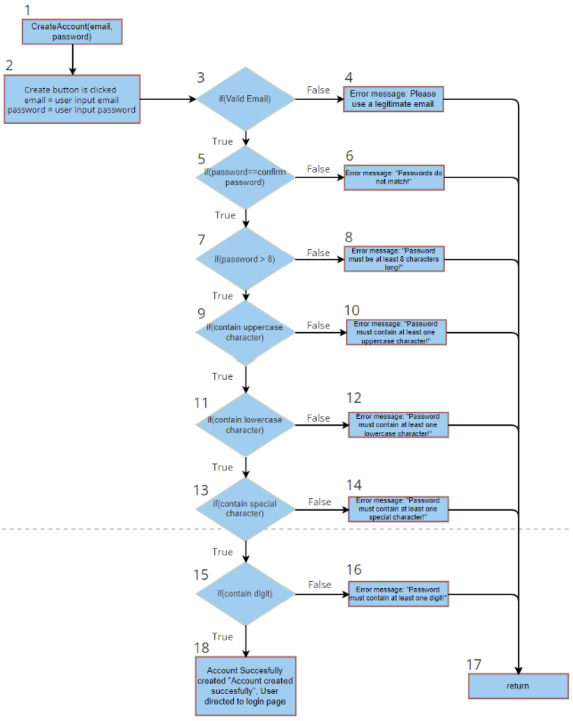


Name	Date	Reason For Changes	Version
User Registration (Use Case 1)	Sep 27, 2024	Added backend account signup and user verification.	1.0
	Nov 5, 2024	Minor bug fixes for registration functionality.	1.1
User Login (Use Case 2)	Sep 18, 2024	Implemented login functionality.	1.0
	Sep 27, 2024	Updated backend login functionality and UI refinements.	1.1
User Logout (Use Case 3)	Sep 24, 2024	Added logout functionality and linked to homescreen.	1.0
	Sep 27, 2024	Improved backend logout process.	1.1
Reset Password (Use Case 4)	Sep 16, 2024	Added reset password functionality and UI.	1.0
	Nov 10, 2024	Fixed validation issues in reset password flow.	1.1
View Listings Based on Estate (Use Case 5)	Oct 4, 2024	Added estate filtering and detailed listing display.	1.0
	Nov 6, 2024	Enhanced UI for estate listing and affordability markers.	1.1
View Street View (Use Case 5.1)	Oct 28, 2024	Added street view functionality using map marker.	1.0
	Nov 6, 2024	Improved map interactions for better user experience.	1.1
Compare Listings and Filter Amenities by Type (Use Case 5.2)	Oct 28, 2024	Added comparison functionality for listings.	1.0
	Nov 6, 2024	Enhanced filtering of amenities by type for comparison.	1.1
Toggle Save Listings While Viewing Listing (Use Case 5.3)	Oct 31, 2024	Added save/unsave functionality for listings.	1.0
	Nov 8, 2024	Integrated permanent images for saved listings.	1.1
Delete Saved Listings in Bookmarks Page (Use Case 6)	Nov 6, 2024	Added ability to delete listings directly from bookmarks.	1.0
	Nov 8, 2024	Improved UI for bookmark page edit mode.	1.1
Edit User Preferences (Use Case 7)	Sep 23, 2024	Added user preferences editing functionality.	1.0
	Nov 10, 2024	Fixed issues with preference validation and real-time updates.	1.1
AI Chatbot Housing Recommendation (Use Case 8)	Nov 13, 2024	Implemented AI chatbot for tailored housing suggestions.	1.0
	Nov 14, 2024	Added memory and preference-awareness features to chatbot.	1.1

Table of contents

Use Case ID:	1		
Use Case Name:	User Registration		
Created By:	<Sanjana>	Last Updated By:	<Sanjana>
Date Created:	<Sep 27, 2024>	Date Last Updated:	<Nov 5, 2024>

Actor:	Unregistered User
Description:	A new user creates an account on the platform by providing an email and password. The system validates the input data and assigns a unique User ID.
Preconditions:	The user is on the registration page.
Postconditions:	The user’s account is created, and the system generates a unique User ID.
Priority:	High
Frequency of Use:	Once

Flow of Events:	(Valid Email and Strong password and same confirmed password) ● The user navigates to the registration page. ● The system shows registration form ● The user enters a valid email, a strong password and the same confirmed password. ● The system validates the input data for valid email and for a strong password. ● The user is notified of successful registration and directs them to the home page.
Alternative Flows:	 1 CreateAccount(email, password) 2 Create button is clicked email = user input email password = user input password 3 if(Valid Email) False 4 Error message: Please use a legitimate email True 5 if(password==confirm password) False 6 Error message: "Passwords do not match!" True 7 if(password > 8) False 8 Error message: "Password must be at least 8 characters long!" True 9 if(contain uppercase character) False 10 Error message: "Password must contain at least one uppercase character!" True 11 if(contain lowercase character) False 12 Error message: "Password must contain at least one lowercase character!" True 13 if(contain special character) False 14 Error message: "Password must contain at least one special character!" True 15 if(contain digit) False 16 Error message: "Password must contain at least one digit!" True 18 Account Succesfully created "Account created succesfully", User directed to login page 17 return
Exceptions:	System fails to create an account due to network failure. ● System displays an error alert("Network error. Please check your internet connection.");
Includes:	None

Special Requirements:	-Passwords must meet a strong password policy (minimum 8 characters, at least one upper and lower case, contain at least 1 number, and contain at least 1 special characters). -Password and confirmed password must be the same. -The system must ensure that email addresses are unique.
Assumptions:	Users will follow password reset instructions correctly.
Notes and Issues:	TBD on potential 2-factor authentication for added security.

Use Case ID:	2		
Use Case Name:	User Login		
Created By:	<Sanjana>	Last Updated By:	<Sanjana>
Date Created:	<Sep 18, 2024>	Date Last Updated:	<Sep 27, 2024>

Actor:	Registered User
Description:	A registered user logs into the system by providing their email and password.
Preconditions:	The user has a valid account.
Postconditions:	The user is successfully logged in and taken to the homescreen.
Priority:	High
Frequency of Use:	Occasional, based on user activity
Flow of Events:	(User key in their correct email and password) ● The user navigates to the login page. ● The user enters their email and password. ● The system verifies the credentials. ● Login successful, user is direction to the homepage

Alternative Flows:	AF1 (User key in wrong email and correct password) : ● The user navigates to the login page. ● The user enters a wrong email and the correct password. ● The system verifies the credentials. ● Error message displays login failed, incorrect password or email. AF2 (User key in correct email and wrong password): ● The user navigates to the login page. ● The user enters their username and the wrong password. ● The system verifies the credentials. ● Error message displays login failed, incorrect password or email. AF3 (User key in wrong email and wrong password): ● The user navigates to the login page. ● The user enters their email and the wrong password. ● The system verifies the credentials. ● Error message displays login failed, incorrect password or email.
Exceptions:	System fails to login due to network failure. ● System displays an error message alert("Network error. Please check your internet connection.");
Includes:	None
Special Requirements:	None
Assumptions:	The user remembers their login credentials or has access to reset them. The system is operational and can process login requests.

Notes and Issues:	None

Use Case ID:	3		
Use Case Name:	User logout		
Created By:	<Sanjana>	Last Updated By:	<Sanjana>
Date Created:	<24 Sep, 2024>	Date Last Updated:	<27 Sep, 2024>

Actor:	Registered User
Description:	User logout of the application.
Preconditions:	User on the homepage
Postconditions:	User directed to the login screen when logout button is pressed
Priority:	High
Frequency of Use:	Occasionally
Flow of Events:	● The user presses logout on the home screen ● System displays successful logout and directs the user to the login page.
Alternative Flows:	None
Exceptions:	If the system fails to logout due to connectivity issues, error message displays logout unsuccesful. ● System displays an error

	alert("Network error. Please check your internet connection.");
Includes:	None
Special Requirements:	User on the home page to press logout
Assumptions:	Users can find the logout button
Notes and Issues:	None

Use Case ID:	4		
Use Case Name:	Reset password		
Created By:	<Wynette>	Last Updated By:	<Wynette>
Date Created:	<16 Sep, 2024>	Date Last Updated:	<10 Nov, 2024>

Actor:	Registered User
Description:	The user resets their password
Preconditions:	The user forgets password
Postconditions:	The user successfully resets their password
Priority:	Medium
Frequency of Use:	Low

Flow of Events:	● The user navigates to the reset password page. ● The user enters their email ● A 6 digit pin sent to their email to reset ● User retrieves 6 digit pin ● User keys in correct 6 digit pin ● System direct user to key in new password and confirms password ● User key in a strong password and same confirmed password. ● System displays successful password reset, directs user to login page.
Alternative Flows:	AF 1 (The user keys in a wrong pin): ● The user navigates to the reset password page. ● The user enters their email ● A 6 digit pin sent to their email to reset ● User retrieves 6 digit pin ● User keys in wrong 6 digit pin ● System displays error message “wrong pin” AF 2 (The user keys in a weak password): ● The user navigates to the reset password page. ● The user enters their email ● A 6 digit pin sent to their email to reset ● User retrieves 6 digit pin ● User keys in correct 6 digit pin ● System direct user to key in new password and confirms password ● User keys in weak password ● System displays, password reset failed AF 2 (The user keys in a strong password and not the same confirm password): ● The user navigates to the reset password page. ● The user enters their email ● A 6 digit pin sent to their email to reset ● User retrieves 6 digit pin ● User keys in correct 6 digit pin

	- System direct user to key in new password and confirms password - User keys in weak password - System displays, password does not match
Exceptions:	EX 1 (The user keys in a wrong email): - The user navigates to the reset password page - The user enters an email which is not theirs - A 6 digit pin sent to email that is not theirs - User unable to retrieve 6 digit pin from their email (not found) - User keys in correct email. EX 2 System fails to login due to network failure. - System displays an error alert("Network error. Please check your internet connection.");
Includes:	None
Special Requirements:	-Passwords must meet a strong password policy (minimum 8 characters, at least one upper and lower case, contain at least 1 number, and contain at least 1 special characters). -Password and confirmed password must be the same.
Assumptions:	User knows where to check the pin in their email.
Notes and Issues:	None

Use Case ID:	5		
Use Case Name:	View listings based on estate		
Created By:	<Sanjana>	Last Updated By:	<Wynette>

Date Created:	<Oct 4, 2024>	Date Last Updated:	<Nov 10, 2024>

Actor:	Registered User
Description:	The users can click on the markers on the map, each representing a resale housing listing. When clicked, it will display the info of the listing. (Eg, Location, size, hdb type, price, contact info of property agent and amenities around that area 1km radius)
Preconditions:	Users select an estate and click on one of the housing listing markers.
Postconditions:	The listing information is shows through a pop-up sidebar on the left
Priority:	High
Frequency of Use:	Occasionally
Flow of Events:	(If there is input for user preferences) ● The user navigates to the home screen ● The user selects a housing estate to filter from the drop down bar. ● The system computes the adjusted price using the user preferences ● The system filters the listings and colour coded the markers based on affordability on the map for that selected estate ● The user selects one of these markers and a pop-up slide bar of the listings details will be shown.
Alternative Flows:	AF1 (If there is no input for user preferences) ● The user navigates to the home screen

	- The user selects a housing estate of their choice from the drop down bar to filter. - The system tries to compute the adjusted price for filtering but fails to do so. - The system does not filter based on user preferences and sets all markers as red markers on the map for that selected estate.
Exceptions:	If there are any connectivity issues, that the backend is unable to retrieve the data from the resale housing api, the system will display an error message. - alert("Network error. Please check your internet connection.");
Includes:	None
Special Requirements:	None
Assumptions:	Users will pick an estate if not no markers will show.
Notes and Issues:	None

Use Case ID:	5.1
Use Case Name:	View street view

Created By:	<Wynette>	Last Updated By:	<Wynette>
Date Created:	<Oct 28, 2024>	Date Last Updated:	<Nov 6, 2024>

Actor:	Registered User
Description:	The user selects a housing estate and drags the yellow man marker to a desired location on the map to view a 3D panoramic street view of the selected location.
Preconditions:	Users are on the homepage
Postconditions:	User views the street view of estate of their choice
Priority:	Medium
Frequency of Use:	Occasionally
Flow of Events:	● The user navigates to the home screen. ● The user selects a housing estate of their choice from the drop down bar to filter. ● The user will drag the yellow man marker and place it on the street they want to view. ● The screen will display a 3D panoramic image view of the street location that the user has placed the marker.
Alternative Flows:	None
Exceptions:	If the system is unable to retrieve Google Maps or Street View due to connectivity issues: ● System displays an error

	alert("Network error. Please check your internet connection.");
Includes:	None
Special Requirements:	None
Assumptions:	The user will understand how to use google map.
Notes and Issues:	None

Use Case ID:	5.2		
Use Case Name:	Compare Listings and filter amenities by type		
Created By:	<Wynette>	Last Updated By:	<Wynette>
Date Created:	<Oct 28, 2024>	Date Last Updated:	<Nov 6, 2024>

Actor:	Registered User
Description:	The user selects two listings of their choice to compare based on their location, price, size, hdb type (is the same since the system is filtered based on hdb type) and amenities which can be further filtered by comparing the type of amenities. The choice of compared listings can be from different estates.
Preconditions:	The user is already viewing one listing and wants to compare against another listing.
Postconditions:	The user compares two listings and their listing information.
Priority:	Medium
Frequency of Use:	Occasionally
Flow of Events:	(Choose another listing to compare from a same estate, afterwards filter amenities according to a specific type eg, schools nearby) ● The user presses the compare listing button. ● The map will then zoom out allowing the user to choose the second listing from the same estate. ● The user selects on another listings of their choice ● The second listing details will be shown on a pop-up sidebar on the right screen. At this point the original

	listing is on the left and the second listing is on the right of the screen. ● The user filters amenities by type through the drop down bar at the nearby amenities section. By default all the amenities are shown. ● The screen will filter both listings of nearby amenities according to the type.
Alternative Flows:	AF 1 (Choose another listing to compare from a different estate) ● The user presses the compare listing button. ● The map will then zoom out allowing the user to choose the second listing. ● The user selects a different estate of their choice ● The system will filter the listings according to the adjusted price and colour code according to affordability on the map. ● The user selects on another listings of their choice ● The second listing details will be shown on a pop-up sidebar on the right screen. At this point the original listing is on the left and the second listing is on the right of the screen. ● The user filters amenities by type through the drop down bar at the nearby amenities section. By default all the amenities are shown. ● The screen will filter both listings of nearby amenities according to the type.
Exceptions:	If the system is unable to retrieve Google Maps or Street View due to connectivity issues: ● System displays an error alert("Network error. Please check your internet connection.");
Includes:	None
Special Requirements:	None
Assumptions:	The user has already keyed in their user preferences, as a result all the listings are filtered based on the affordability calculated

	from user preferences and also filtered based on the hdb type that the user wants. This means that if the users do not key in their preferences, the comparison of the listings will not be fair. (Eg, a 3 room against a executive)
Notes and Issues:	It is important to note that while viewing listings without entering user preferences is optional, it may limit the ability to compare listings as designed by the system. Users have the flexibility to explore different types of HDB flats, but for a more tailored experience and effective comparison, providing preferences is recommended. This ensures users can make more informed decisions based on their specific needs and criteria.

Use Case ID:	5.3		
Use Case Name:	Toggle save listings while viewing listing		
Created By:	<Chong En>	Last Updated By:	<Sanjana>
Date Created:	<Oct 31, 2024>	Date Last Updated:	<Oct 31, 2024>

Actor:	Registered User
Description:	The user can save and unsave listings by clicking the heart icon when viewing the listing.
Preconditions:	The user is already viewing at least one listing. The heart icon is not filled (indicating not saved)
Postconditions:	The heart icon is filled when the save is successful
Priority:	Medium
Frequency of Use:	Occasionally
Flow of Events:	● The user presses the heart icon when the heart icon is not filled (not saved). ● The system saved the listing to the database of that user. ● The system displays a message indicating a success in the saving of listing. ● The heart turns full red. (Listing saved)
Alternative Flows:	(User save listings but decides to change their mind)

	● The user presses the heart icon when the heart icon is not filled (not saved). ● The system saved the listing to the database of that user. ● The system displays a message indicating a success in the saving of listing. ● The heart turns full red. (Saved) ● The user presses the heart icon when the heart icon is filled (saved). ● The system removes the saved listing to the database of that user. ● The system displays a message indicating a success on removal of the saved listing. ● The heart turns full red. (not saved)
Exceptions:	The system fails to save it to the firestore which contains the user preferences info and their saved listings due to connectivity issues. ● System displays an error message alert("Network error. Please check your internet connection.");
Includes:	None
Special Requirements:	For saved listings -> Precondition (unsaved) For unsaved listings -> Precondition (saved)
Assumptions:	The user understands that unfilled heart is unsaved listings, filled heart means that the listing is saved.
Notes and Issues:	None

Use Case ID:	6		
Use Case Name:	Delete saved listings in bookmarks page		
Created By:	<Chong En>	Last Updated By:	<Sanjana>
Date Created:	<Oct 31, 2024>	Date Last Updated:	<Oct 31, 2024>

Actor:	Registered User
Description:	The user is uninterested in specific listings that they saved, they can easily go to their bookmarks page and delete instead of going to the listings on the map and unsaving it.
Preconditions:	The user is on the bookmarks page
Postconditions:	-User successfully removes the bookmark
Priority:	Medium
Frequency of Use:	Occasionally
Flow of Events:	● The user navigates to the bookmarks page. ● User press on edit ● Each of the listings would have a delete button which the user can delete individual listings at a time. ● User deletes listings ● System remove the listing at userdatabase (firestore) ● The listings will be removed from the screen upon deletion ● Users press the confirm button to finish the deletion process.

Alternative Flows:	None
Exceptions:	If the system fails to delete a listing due to a network or database error as the screen will be unable to retrieve listings. The screen will be blank and display an error message, unable to delete listings.
Includes:	None
Special Requirements:	Users have saved listings to delete.
Assumptions:	Users understand how to navigate to the bookmarks page and use the "Edit" and "Delete" functions.
Notes and Issues:	Consider adding a confirmation dialog before permanently deleting a listing to prevent accidental deletions. Ensure the system provides feedback (e.g., a success message or visual confirmation) after the deletion.

Use Case ID:	7		
Use Case Name:	Edit user preferences		
Created By:	<Chong En>	Last Updated By:	<Sanjana>
Date Created:	<Oct 31, 2024>	Date Last Updated:	<Nov 10, 2024>

Actor:	Registered User
Description:	The user edits their preferences (e.g., location, HDB type, salary, CPF OA balance, preferred price) to update the tailored property recommendations displayed on the map. The system validates the input and recalculates the recommendations in real-time.
Preconditions:	The user is logged in and on the edit user preferences page
Postconditions:	User preferences are successfully updated. The system recalculates and updates the tailored property recommendations in real-time. The updated preferences are saved in the database.
Priority:	High
Frequency of Use:	Occasionally

Flow of Events:	● The user navigates to the preferences page. ● The user modifies their preferences, such as location, HDB type, salary, CPF OA balance, or preferred price. ● The system validates the user input for completeness and whether it is within acceptable range. ● If the input is valid and complete the system calculates the new affordability range (adjusted price) using the updated preferences. ● System Action: The system updates the property recommendations on the map, with affordability colour-coded markers: ● Green marker for affordable listings. ● Yellow marker for moderately affordable listings. ● Red marker for unaffordable listings. ● User Action: The user confirms and saves the updated preferences.
Alternative Flows:	The system detects invalid or incomplete preferences. The system displays an error message prompting the user to enter valid input. The user corrects the input and proceeds with the update.
Exceptions:	If the system fails to save the updated preferences due to network or database errors, the system will give an error message, Unable to Update User preferences.
Includes:	None

Special Requirements:	Users should be able to see the updated recommendations in real-time after modifying their preferences. The system must ensure no duplicate or conflicting entries in preferences. Required Fields: 1. **Monthly Salary** ○ Must be provided. ○ Must be a positive number. ○ Must be a valid decimal (up to 2 decimal places). 2. **CPF OA Balance** ○ Must be provided. ○ Must be a positive number. ○ Must be a valid decimal (up to 2 decimal places). 3. **HDB Type** ○ Must be selected from the dropdown menu (e.g., 1-Room, 2-Room, etc.). 4. **Preferred Price** ○ Must be provided. ○ Must be at least $200,000. ○ Must be a valid decimal (up to 2 decimal places). 5. **Housing Loan Type** ○ Must be selected (e.g., HDB Housing Loan, Bank Loan).
Assumptions:	Users understand how to navigate the preferences page and adjust their inputs.
Notes and Issues:	Provide clear feedback (e.g., success message) once preferences are successfully updated. Consider implementing a tutorial or help guide for users unfamiliar with the preferences functionality. Ensure robust validation for all user inputs (e.g., valid salary ranges, existing CPF balances).

Use Case ID:	8		
Use Case Name:	AI Chatbot housing recommendation		
Created By:	<Sanjana>	Last Updated By:	<Sanjana>
Date Created:	<Nov 13, 2024>	Date Last Updated:	<Nov 14, 2024>

Actor:	Registered User
Description:	The user interacts with a chatbot on the "Edit Preferences" page to get personalised housing suggestions or ask generic housing-related questions. The chatbot utilises the user's existing preferences to tailor responses.
Preconditions:	The user is on the "Edit Preferences" page. The chatbot window is available and accessible. The user's preferences are already stored in the firestore database.
Postconditions:	The user successfully receives suggestions based on their saved preferences. The chatbot session persists while the user remains on the "Edit Preferences" page. The chatbot resets when the user navigates away from the page.
Priority:	Medium
Frequency of Use:	Occasionally

Flow of Events:	The user navigates to the "Edit Preferences" page. The chatbot appears as an empty chat window with prompts (e.g., "Ask me about housing suggestions!", "Need help finding affordable homes?"). The user initiates a conversation with the chatbot (e.g., "What are my housing options in Bedok?"). The chatbot retrieves the user's preferences (e.g.,salary, CPF OA balance, preferred, preferred location and other attributes not just user preferences) and provides tailored suggestions. The user continues asking questions, and the chatbot responds while considering the user's data.
Alternative Flows:	Alternative Flow 1: ● **Scenario**: The user minimises the chat window. ○ The chatbot remains accessible, retaining the chat history until the user leaves the "Edit Preferences" page. ○ When reopened, the user can continue the session seamlessly. Alternative Flow 2: ● **Scenario**: The user updates preferences during the chatbot session. ○ If the chatbot is capable of live updates, it recalculates suggestions in real-time based on the new preferences. Alternative Flow 3: ● **Scenario**: The user asks a generic question (e.g., "What’s the best area for families?"). ○ The chatbot responds with general insights while keeping the user's data in mind for privacy and relevance.

Exceptions:	If the chatbot cannot retrieve user preferences due to a system error, it notifies the user and provides generic suggestions. If due to connectivity issues or the ai chat box is done, an error message will inform the user the ai chatbot is currently unavailable please try again later.
Includes:	None
Special Requirements:	The chatbot should display intuitive prompts to guide the user. The chatbot must handle both generic and personalized queries. Chat history should persist while the user remains on the page.
Assumptions:	The user does not make simultaneous changes to preferences while chatting unless the chatbot supports live updates. The user understands the chatbot's role is to suggest housing options based on preferences but not to modify preferences directly.
Notes and Issues:	Consider adding a feature to allow users to export chatbot suggestions as a PDF or image. Potential for integrating with third-party APIs (e.g., for real-time housing availability). Ensure user data privacy by securing backend calls and not exposing sensitive information during conversations. Explore whether the chatbot can support live preference updates or provide a warning when preferences are modified mid-session.