

System Feature

Feature 1: Account Management

Description and Priority

This feature allows users to create and manage their accounts, including registration, login, logout, and profile updates. It ensures secure handling of user credentials and offers a smooth experience for managing profile details like salary, CPF balance, and housing loans.

Stimulus/Response Sequences

Stimulus: The user enters email address, and password on the registration page.

- **Response:** The system validates the details and creates a new account, assigning a unique User ID.

Stimulus: The user logs in using their username/email and password.

- **Response:** The system authenticates the credentials and grants access to the user's profile.

Stimulus: The user initiates the password reset process.

- **Response:** The system sends an email to the registered email address with instructions to reset the password.

Functional Requirements

1. User Registration

- 1.1. Users shall be able to create an account by providing an email address and password.

2. User Login and Logout

- 2.1. Users shall be able to log in using their registered credentials (email and password).
- 2.2. Users shall be able to securely log out from their account

3. Password Management

- 3.1. Users shall be able to reset their password using a secure process involving their registered email address.
- 3.2. Passwords must comply with the strong password policy (minimum 8 characters, mixed case, numbers, and special characters).

4. Profile Management

- 4.1. Users shall be able to update their profile information, including salary, CPF balance, and housing loan type.

5. Error Handling and Feedback

- 5.1. The system shall provide feedback for invalid login attempts or invalid registration details.

6. Account Lockout and Security

- 6.1. The system shall lock the user's account after 5 consecutive failed login attempts for 15 minutes and send a notification email.

Feature 2: Tailored Recommendations

Description and Priority

This feature allows users to input housing preferences and receive customized property recommendations based on their profile information.

Stimulus/Response Sequences

Stimulus: The user inputs preferences such as location, HDB type, and price range.

- **Response:** The system calculates an affordability range using the user's profile data and displays relevant property listings.

Stimulus: The user adjusts their financial or housing preferences.

- **Response:** The system updates the recommendations in real-time.

Functional Requirements

1. Calculate Affordability

- 1.1. The system shall calculate an affordability range based on the user's profile data.

2. Display Recommendations

- 2.1. The system shall display property recommendations based on the user's preferences and financial data.

3. Update Preferences

- 3.1. Users shall be able to modify their preferences and see real-time updates in recommendations.

4. Default Display

- 4.1. When no preferences are indicated, all available properties shall be displayed.

Feature 3: Listings DisplayFunctional Requirements

Description and Priority

This feature involves displaying property listings in a user-friendly manner, including sorting and colour-coding based on the user's affordability range.

Stimulus/Response Sequences

Stimulus: The user views search results based on their preferences.

- **Response:** The system displays the results using a colour-coded map indicating property affordability.

Stimulus: The user changes their preferred price range.

- **Response:** The colour-coding of property listings is updated in real-time.

Functional Requirements

1. Color-Coding

- 1.1. The system shall display property listings on a map using color-coding (green, orange, red) based on the user's affordability range.

2. Price Range Adjustment

- 2.1. Users shall be able to adjust their preferred price range, and the color-coded results shall update accordingly.

3. Property Details

- 3.1. The system shall allow users to view property details in a pop-up sidebar.

4. Property Comparison

- 4.1. Users shall be able to compare multiple properties side-by-side.

Feature 4: Property Management and Bookmarking

Description and Priority

This feature allows users to manage and bookmark their favourite properties for future reference, enabling them to save, sort, and compare listings.

Stimulus/Response Sequences

Stimulus: The user clicks on a property in the search results to view more details.

- **Response:** The system presents detailed property information in a sidebar.

Stimulus: The user clicks the heart symbol to save a property to their favourites.

- **Response:** The property is saved to the user's profile, and the heart symbol is highlighted.

Stimulus: The user unclicks the heart symbol on a saved property.

- **Response:** The property is removed from the user's favourites, and the heart symbol is unhighlighted.

Functional Requirements

1. View Property Details

- 1.1. Users shall be able to view detailed information about a property from the search results.

2. Sorting Options

- 2.1. The system shall allow users to sort property listings by criteria such as relevance, price, and date listed.

3. Save Property

- 3.1. The system shall offer users the ability to save properties by clicking the heart symbol.

4. Remove Saved Property

- 4.1. Users shall be able to remove properties from their saved list by clicking the heart symbol again.

5. Compare Properties

- 5.1. The system shall allow users to compare selected properties side-by-side by displaying multiple sidebars simultaneously.

Feature 5: Data Management

Description and Priority

This feature manages the retrieval, processing, and display of resale unit data from government databases such as HDB and URA, ensuring users access the most accurate and up-to-date listings.

Stimulus/Response Sequences

Stimulus: The system automatically triggers the refresh process every 1 hour

- **Response:** The system retrieves the latest resale unit listings from external government databases and updates the displayed data for users.

Functional Requirements

1. Data Retrieval

- 1.1. The system shall retrieve resale unit data from government databases, such as HDB and URA, to ensure accuracy and up-to-date information.

2. Data Refresh

- 2.1. The system shall update the resale unit listings only when the system administrator triggers the manual refresh process.

3. Connectivity Alerts

- 3.1. The system shall alert the administrator in case of connectivity issues with external databases.

4. Data Retrieval Retry

- 4.1. In case of a connectivity issue, the system shall retry data retrieval after the next scheduled refresh by the administrator.

Other Nonfunctional Requirements

Performance Requirements

1. Performance

This feature ensures that the system returns search results within 3 seconds when handling up to 1,000 concurrent users. It is crucial for providing a responsive user experience.

1.1. Stimulus/Response Sequences

Stimulus: The user performs a search query while the system is handling 1,000 concurrent users.

Response: The system returns the search results within 3 seconds.

1.2. The system shall be tested under a simulated load of 1,000 concurrent users performing searches.

1.3. The response time shall not exceed 3 seconds during these tests.

2. Scalability

The application must handle up to 10,000 concurrent users with no more than a 10% increase in response time compared to handling 1,000 users.

2.1. Stimulus/Response Sequences

Stimulus: The system is subjected to a load of 10,000 concurrent users performing searches.

Response: The system's response time increases by no more than 10% compared to the response time with 1,000 users.

2.2. Load testing shall be conducted with 10,000 concurrent users.

2.3. The response time for 10,000 users shall be compared with that of 1,000 users to ensure the increase is within 10%.

3. Intuitive Navigation

This feature ensures that users can access any feature within three clicks from the homepage, improving the ease of navigation.

4. Ease of Use

This feature ensures that 80% of first-time users can complete a flat search within 5 minutes without external assistance, supporting a smooth onboarding experience.

4.1. Stimulus/Response Sequences

Stimulus: A first-time user begins a flat search

Response: The system enables the user to complete the search within 5 minutes

4.2. Usability testing shall be conducted with 50 first-time users.

4.3. At least 40 users must complete a flat search within the 5-minute timeframe.

5. Supportability

This feature ensures that the application is fully functional on all major browsers and devices, providing a consistent experience across platforms.

5.1. Stimulus/Response Sequences

Stimulus: The user accesses the system using different browsers.

Response: The system functions properly, with all features accessible and responsive across platforms

5.2. Validation tests shall ensure that all core functionalities work on Chrome Safari and Edge.

5.3. The layout shall adapt to the different browsers with no horizontal scrolling required to access all functions of the system.

Safety Requirements

1. Security Requirements

1.1 Description and Priority

This feature ensures the security of user accounts through strong password policies, account lockout measures, and identity verification processes.

1.2 Stimulus/Response Sequences

Stimulus: The user creates a new password during account registration or password reset.

- **Response:** The system validates the password strength and prompts the user if the password does not meet the requirements.

Stimulus: The user attempts to log in and fails multiple times

- **Response:** The system locks the user's account after 5 failed login attempts and sends a notification email to reset the password using a pin.

2. Functional Requirements

2.1. Strong Password Policy

- 2.1.1. The system shall enforce a strong password policy requiring at least 8 characters, with a combination of uppercase letters, lowercase letters, numbers, and special characters.
- 2.1.2. The system shall require users to enter their password twice for confirmation during account creation and password changes.

2.2. Password Confirmation

- 2.2.1. The system shall require users to enter their password twice for confirmation during account creation and password changes.

2.3. Account Lockout

- 2.3.1. The system shall lock the user's account after 5 consecutive failed login attempts for a duration of 15 minutes.

2.4. Lockout Notification

- 2.4.1. The system shall notify the user via email when their account is locked and provide instructions for unlocking it.

2.5. Account Unlocking

- 2.5.1. The system shall allow users to unlock their accounts by verifying their identity through a secure verification code sent to their registered email address.

Data Dictionary

User and Account Management

Field Name	Data Type	Data Format	Description
Manage Account	Varied		Functionalities related to updating and maintaining user account details, including password, salary, CPF balance, and housing preferences.
CPF Balance	Number		The current amount in the user's CPF (Central Provident Fund) account, which can be used towards the purchase of a resale flat.
Housing Preferences	Varied		The user's desired HDB type, location, and price range for search purposes.
Housing Loan Type	String		The category of loan the user intends to take (e.g., bank loan, HDB loan).
Email	String		The user's email address, used for account management tasks such as password recovery.
Password	String		A secure string of characters used by users for authentication. It must meet the system's password policy (minimum 8 characters, and a

			combination of character types).
Profile Information	Varied		User's personal details, such as salary, CPF balance, and housing preferences, which can be updated from their account.
User ID	String	A string of 28 letters	A unique identifier assigned to each user upon account creation, used to distinguish different users in the system.

Property Listing and Management

Field Name	Data Type	Data Format	Description
Agency Name	String		The name of the real estate agency handling the flat.
Agency Website	String		The website of the real estate agency.
Availability Date	Date/Time	DD/MM/YYYY	The date the resale flat is available for purchase.
Contact Number	String	NNNNNNNN	The phone number of the landlord or property manager.
Customised Property Recommendations	Varied		A list of property listings that are generated based on the property listings, user information and housing preferences, using our formula.
HDB Location	String	Street number, Street name, Unit number, Postal	The specific estate where the resale flat is located.

		Code	
HDB Resale Flat	String		A previously owned flat that is for sale after the Minimum Occupation Period (MOP).
HDB Type	String		The size or type of the flat (e.g., 3-room, 4-room).
Landlord ID	String	A string of 28 letters	A unique identifier assigned to the landlord or property manager.
Listing Date	Date/Time	DD/MM/YYYY	The date when the resale flat was published online.
Postal Code	String	NNNNNN	The postal code of the resale flat.
Price	Number		The selling price of the HDB resale flat.
Price Range	Number		The range of prices inputted by the user.
Property ID	String		A unique identifier assigned to each property listing.
Saved Listings	Varied		Properties that the user has bookmarked for future reference.
Saved Searches	Varied		Search parameters saved for future reference.
Street Name	String		The name of the street on which the resale flat is located.
Street Number	String	NNN	The building number of the resale flat.

Property Listings and Management	String		The name of the real estate agency handling the flat.
----------------------------------	--------	--	---

Affordability

Field Name	Data Type	Data Format	Description
Affordability	Number		A computed metric to determine how much the user can afford for a resale flat.
Green Affordability Category	Boolean		Indicates flats within the user's preferred price range.
Orange Affordability Category	Boolean		Indicates flats slightly above the user's preferred price range.
Red Affordability Category	Boolean		Indicates flats outside the user's preferred price range.

Search and Filtering

Field Name	Data Type	Data Format	Description
Comparison	Function		Feature allowing users to view multiple properties side-by-side.
Search Criteria	Object		Filters and inputs provided by the user to find resale flats.
Search Functionality	Function		Allows users to input attributes like location and price for flat searches.
Search Results	Array		The list of resale flats displayed based on search criteria.

Sort Criteria	Varied		Criteria by which users can sort their search results (e.g., price, date listed).
---------------	--------	--	---

System and Performance

Field Name	Data Type	Data Format	Description
Browser Compatibility	Boolean		Ensures the application works across different browsers.
Data Refresh	Function		The process of updating property listings from external databases.
Device Compatibility	Boolean		Ensures functionality across devices (desktops, tablets, mobile).
Downtime	Number		The period when the system is unavailable.
External Databases	Array		Sources such as HDB and URA from which flat data is retrieved.
Last Data Refresh	Date/Time	NN:NN	The timestamp of the last data refresh initiated by the admin.
Response Time	Number		The time taken for the system to return search results.
Scalability	Number		The system's ability to handle increased load.
System Reboot	Function		The process of restarting the system.
System Uptime	Number		The percentage of time the system remains operational.

Zero-Downtime Updates	Boolean		Capability to implement system updates without downtime.
-----------------------	---------	--	--

Use Case Descriptions

Use Case ID:	1.1		
Use Case Name:	User Registration		
Created By:	<Author>	Last Updated By:	<Author>
Date Created:	<Date>	Date Last Updated:	<Date>

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:	<ul style="list-style-type: none"> • The user navigates to the registration page. • The user enters their email and password. • The system validates the input data (e.g., email format, password strength). • The system creates a new account and assigns a unique User ID. • The user is notified of successful registration.
Alternative Flows:	If validation fails, the system displays an error message, prompting the user to re-enter valid details.
Exceptions:	System fails to create an account due to network failure. <ul style="list-style-type: none"> • System displays an error and prompts user to retry later.
Includes:	None
Special Requirements:	Passwords must meet strong password policy (minimum 8 characters, mixed case, numbers, and special characters). 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:	1.2
Use Case Name:	User Login

Created By:	<Author>	Last Updated By:	<Author>
Date Created:	<Date>	Date Last Updated:	<Date>

Actor:	Registered User
Description:	A registered user logs into the system by providing their username and password.
Preconditions:	The user has a valid account.
Postconditions:	The user is successfully logged in and taken to the dashboard.
Priority:	High
Frequency of Use:	Occasional, based on user activity
Flow of Events:	<ul style="list-style-type: none"> • The user navigates to the login page. • The user enters their username and password. • The system verifies the credentials. • The user is logged in and redirected to the dashboard.
Alternative Flows:	<ul style="list-style-type: none"> • If the login credentials are incorrect, the system displays an error message and prompts the user to re-enter them. • If the user forgets their password, they can initiate the "Reset Password" flow.
Exceptions:	System fails to calculate affordability due to missing profile data. <ul style="list-style-type: none"> • System prompts user to complete profile.
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:	Users may attempt to log in too frequently with incorrect credentials, triggering account lockout mechanisms. The password reset flow must ensure security while being user-friendly.

Use Case ID:	1.3		
Use Case Name:	Update Profile		
Created By:	<Author>	Last Updated By:	<Author>
Date Created:	<Date>	Date Last Updated:	<Date>

Actor:	Registered User
Description:	The user updates their profile information, such as salary, CPF balance, and housing preferences.
Preconditions:	The user is logged in and on the profile page.
Postconditions:	The profile is updated, and the system recalculates the user's preferred price range.

Priority:	Medium
Frequency of Use:	Moderate
Flow of Events:	<ul style="list-style-type: none"> • The user navigates to the profile page. • The user updates their salary, CPF balance, and/or housing preferences. • The system validates the changes. <ul style="list-style-type: none"> • For example, it checks that the salary and CPF balance are positive numbers and the housing preferences are within the allowed options. • The system updates the profile in the database and recalculates the user's preferred price range based on the new data. • The system displays a confirmation message indicating that the profile has been successfully updated.
Alternative Flows:	<p>If the input is invalid (e.g., the CPF balance is negative):</p> <ul style="list-style-type: none"> • The system displays an error message explaining the issue. • The user is prompted to correct the invalid input and resubmit the changes.
Exceptions:	If the system is unable to save the profile updates due to network failure, the system prompts the user to try again later.
Includes:	None
Special Requirements:	Profile changes must be validated before being saved. The system must securely handle and store sensitive information (e.g., salary and CPF balance).
Assumptions:	The user understands the meaning of the fields they are updating.
Notes and Issues:	Future integration with additional financial tools (e.g., CPF calculators) could be beneficial.

Use Case ID:	1.4		
Use Case Name:	Search for Resale Flats		
Created By:	<Author>	Last Updated By:	<Author>
Date Created:	<Date>	Date Last Updated:	<Date>

Actor:	Registered User
Description:	The user searches for resale flats and inputs search criteria such as HDB location, price range, and/or HDB type.
Preconditions:	The user is logged in.
Postconditions:	The system displays a list of flats that are color-coded based on their affordability to the user.
Priority:	High
Frequency of Use:	Frequently

Flow of Events:	<ul style="list-style-type: none"> • The user navigates to the search page. • The user enters search criteria such as HDB location, price range, and HDB type. • The system retrieves relevant listings from the Resale Housing API. • The system calculates the affordability range according to the user preferences. • The system displays the search results, categorized by affordability (green, orange, red). • The user reviews the listings.
Alternative Flows:	If no listings match the search criteria, the system displays a message indicating that no results were found.
Exceptions:	<ul style="list-style-type: none"> • If the system is unable to retrieve listings due to API failure, it displays an error message.
Includes:	None
Special Requirements:	<p>The system must retrieve up-to-date data from the Resale Housing API.</p> <p>The results should be displayed with a color-coding system for affordability.</p>
Assumptions:	The user understands the meaning of the affordability color coding.
Notes and Issues:	Consider handling scenarios where the API response is delayed to avoid poor user experience.

Use Case ID:	1.5		
Use Case Name:	Save Resale Flat Listing		
Created By:	<Author>	Last Updated By:	<Author>
Date Created:	<Date>	Date Last Updated:	<Date>

Actor:	Registered User
Description:	The user saves a resale flat listing for future reference.
Preconditions:	The user is logged in and viewing a resale flat listing..
Postconditions:	The listing is added to the user's saved listings.
Priority:	Medium
Frequency of Use:	Occasionally
Flow of Events:	<ul style="list-style-type: none"> • The user selects a resale flat from the search results. • The user clicks the "Save" button (heart icon). • The system adds the listing to the user's saved listings. • The system confirms that the listing has been saved.
Alternative Flows:	If the listing is already saved, the system will not duplicate it and may notify the user.
Exceptions:	If the system fails to save the listing due to network failure, it prompts the user to try again later.
Includes:	None
Special Requirements:	Duplicate listings must not be saved.
Assumptions:	Users will occasionally want to review saved listings.

Notes and Issues:	Consider adding a maximum limit on the number of saved listings per user.
-------------------	---

Use Case ID:	1.6		
Use Case Name:	Remove Saved Listing		
Created By:	<Author>	Last Updated By:	<Author>
Date Created:	<Date>	Date Last Updated:	<Date>

Actor:	Registered User
Description:	The user removes a previously saved resale flat listing.
Preconditions:	The user is logged in and has previously saved listings.
Postconditions:	The selected listing is removed from the user's saved listings.
Priority:	Medium
Frequency of Use:	Occasionally
Flow of Events:	<ul style="list-style-type: none"> • The user navigates to their saved listings. • The user selects a listing to remove. • The user clicks the "Unsave" button (heart icon). • The system removes the listing from the saved list. • The system confirms that the listing has been removed.
Alternative Flows:	None
Exceptions:	If the system fails to remove the listing due to network failure, it prompts the user to try again later.
Includes:	None
Special Requirements:	The system must ensure that removed listings are fully deleted from the user's saved list without leaving remnants.
Assumptions:	The user intends to remove listings they no longer want to track.
Notes and Issues:	Consider adding a confirmation step before finalizing the removal of a saved listing.

Use Case ID:	1.7		
Use Case Name:	Compare Listings		
Created By:	<Author>	Last Updated By:	<Author>
Date Created:	<Date>	Date Last Updated:	<Date>

Actor:	Registered User
Description:	The user selects multiple resale flat listings to compare key details such as price, location, and floor area.
Preconditions:	<ul style="list-style-type: none"> • The user is logged in and has selected multiple listings to compare.
Postconditions:	User successfully creates an account and logs in.
Priority:	Medium
Frequency of Use:	Occasionally

Flow of Events:	<ul style="list-style-type: none"> • The user navigates to the search results or saved listings. • The user selects multiple listings to compare by checking a box or selecting a "Compare" button. • The user clicks the "Compare" button. • The system displays a comparison view, showing details such as price, location, floor area, and affordability for each listing. • The user reviews the comparison.
Alternative Flows:	If fewer than two listings are selected, the system displays a message prompting the user to select at least two listings for comparison.
Exceptions:	If there is a system error and the comparison cannot be displayed, the system prompts the user to try again later.
Includes:	None
Special Requirements:	<ul style="list-style-type: none"> • The comparison view must be user-friendly and display key details side by side. • The system should allow for comparison of at two listings at once
Assumptions:	The user will understand which fields are compared and how to interpret the comparison.
Notes and Issues:	<ul style="list-style-type: none"> • Consider adding a feature that allows users to export the comparison as a PDF or image. • Potential for integrating third-party tools to enhance the comparison experience (e.g., more data points, additional filters).

Use Case Diagram

