

**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

**SC2006 Software Engineering
Lab 3 Deliverables**

Dream Neighbourhood

Group Member	Matric Number
How En Hsin	U2421325J
Veda Ho Yong Qian	U2321709B
Jarrel Ng Tze Xuan	U2421691L
Isaac Teo Yi Jay	U2421749A
Sean Winston Susanto	U2423834G
Joel Ng Wei Heng	U2422193L

Table of Contents

Add Headings (Format > Paragraph styles) and they will appear in your table of contents.

Submissions:

Complete User Case Descriptions

Complete Class Diagrams

Complete Sequence Diagrams

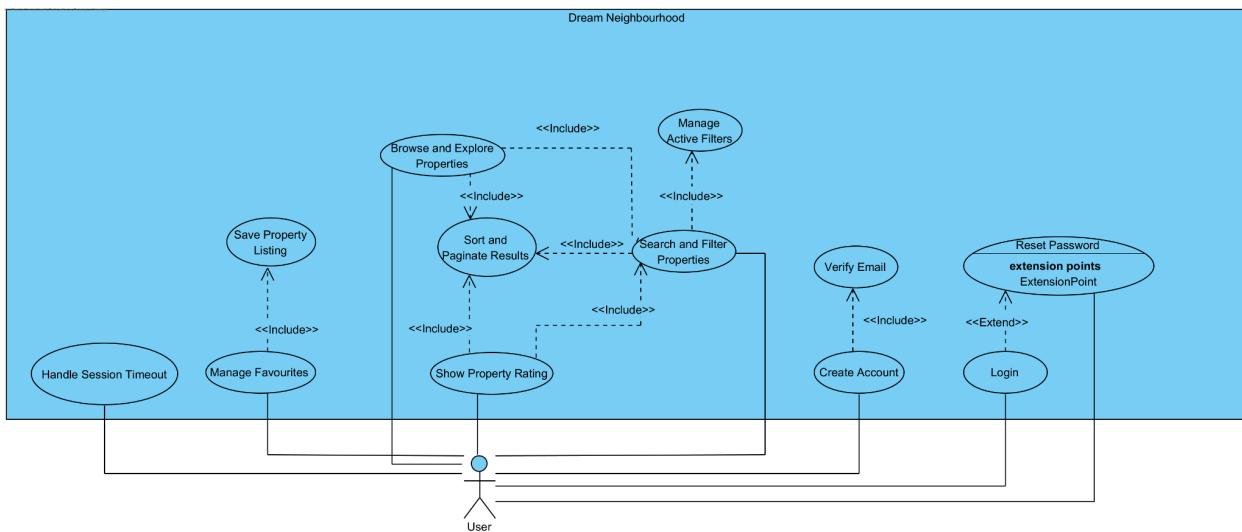
Complete Dialog Map

System Architecture

Application Skeleton

Use Case Model (Use Case Diagram and Descriptions)

A. Use Case Diagram



B. Use Case Descriptions

I. Search and Filter Properties

Use Case ID	01		
Use Case Name	Search and Filter Properties		
Created By	Jarrel Ng	Last Updated By	Isaac Teo
Date Created	1 September 2025	Last Date Updated	1 September 2025

Actor	User
Description	Allows Users to search for properties with filters
Preconditions	User is logged in Property listings database and map/geolocation service is available
Postconditions	System displays filtered and sorted property listings

Priority	High
Frequency of Use	High
Flow of Events	<ol style="list-style-type: none"> 1. User navigates to the search interface 2. User enters a budget range (minimum and maximum) 3. System validates the range ($\text{min} \leq \text{max}$, no negative values) 4. User selects property type(s) 5. User enters one or more key locations (postal code or place name) 6. System validates location input and calculates distance 7. User specifies the maximum acceptable distance for the key location 8. User applies Amenity filters (e.g., parks) 9. User specifies maximum acceptable distance for each amenity (default 4km) 10. User clicks the search button 11. System displays results in pages of 20 items 12. User can sort results by price, distance, or average rating 13. User can clear filters at any time
Alternative Flows	<ol style="list-style-type: none"> 1. User clears filters, system will reset and shows all listings 2. User modifies filters (e.g., budget, amenities), system dynamically updates the result 3. User removes specific key locations or amenities, system re-applies remaining filters
Exceptions	<ol style="list-style-type: none"> 1. Network or database failure, the system will display an error message. 2. No listings are found, and the system displays, "No properties found." 3. Invalid budget input (e.g., $\text{min} > \text{max}$), the system displays, "minimum amount cannot be more than the maximum" 4. Distance calculation fails due to missing map data; the system will exclude affected listings from results
Includes	Manage Active Filters Sort and Paginate Results
Special Requirements	Filters must update results dynamically while filters are modified

	<p>System must support up to 5 key locations and 10 amenities per search</p> <p>Response time for applying filters should not exceed 3 seconds</p>
Assumptions	<p>The property database is available</p> <p>Property listings contain valid postal codes and coordinates</p>
Notes and Issues	<p>UX challenge when filters conflict and eliminate all listings</p> <p>Handling duplicate or ambiguous amenity keywords (e.g., “school” vs “Primary school”)</p> <p>Performance may degrade when many filters are applied simultaneously</p>

A. Manage Active Filters

Use Case ID	01A		
Use Case Name	Manage Active Filters		
Created By	Jarrel Ng	Last Updated By	
Date Created	8 September 2025	Last Date Updated	

Actor	User
Description	User modifies, clears, or resets filters.
Preconditions	At least one filter applied.
Postconditions	Results have been updated to reflect the new filter state.
Priority	High
Frequency of Use	High
Flow of Events	<ol style="list-style-type: none"> 1. User views active filters below the search bar 2. User modifies or removes filters by clicking the minus icon on the filter and applying new filters 3. System updates results dynamically
Alternative Flows	<ol style="list-style-type: none"> 1. User clicks “Clear All” in the filter section, and the default results are shown

Exceptions	1. When the filter eliminates all listings, the system suggests resetting
Includes	
Special Requirements	Real-time feedback on active filters
Assumptions	Filters are combinable
Notes and Issues	UX when multiple filters overlap

B. Sort and Paginate Results

Use Case ID	01B		
Use Case Name	Sort and Paginate Results		
Created By	Jarrel Ng	Last Updated By	
Date Created	8 September 2025	Last Date Updated	

Actor	User
Description	User sorts search results and navigates between pages.
Preconditions	Filtered results available.
Postconditions	Listings shown according to chosen sort order.
Priority	Medium
Frequency of Use	High
Flow of Events	<ol style="list-style-type: none"> 1. User selects sort criteria (price, distance) 2. System reorders listings 3. System displays listings in pages of 20 4. User scrolls or clicks “Load More” 5. System loads next page
Alternative Flows	<ol style="list-style-type: none"> 1. When the results are fewer than 20 listings, the system will show all without pagination
Exceptions	<ol style="list-style-type: none"> 1. When there are no more results, the system will display “No more listings.”

Includes	Nil
Special Requirements	Pagination must not exceed 2s load time.
Assumptions	The listing dataset is large.
Notes and Issues	Infinite scroll vs. paged navigation UX.

II. Browse and Explore Property Listings

Use Case ID	02		
Use Case Name	Browse and Explore Property Listings		
Created By	Jarrel Ng	Last Updated By	Jarrel Ng
Date Created	1 September 2025	Last Date Updated	8 September 2025

Actor	User
Description	The user browses a list of property listings, selects one to view details in a split-view layout, and views a map showing only that property's location. When the user selects a different listing, the map updates to display the newly selected property.
Preconditions	Property search results are available Property database and map service are operational
Postconditions	The user views property details and the location of the selected property on the map The user may switch between different properties, and the map updates accordingly
Priority	High
Frequency of Use	High
Flow of Events	<ol style="list-style-type: none"> System displays a list of filter property listings (20 items per page) System displays property cards with name, price, location, postal code, and thumbnail.

	<ol style="list-style-type: none"> 3. User scrolls or paginates through the list 4. User selects a card 5. System shows split view: left = property info, right = interactive map. 6. User can zoom, pan, but only the selected property remains highlighted.
Alternative Flows	<ol style="list-style-type: none"> 1. Map fails to load, the system will display a static image or an error message 2. User clicks on the same property again, the map will remain unchanged, and the details will stay visible 3. User closes details view, the system will hide the map and return to list only view 4. When the user loads additional results, the user can select a new property to update the map
Exceptions	<ol style="list-style-type: none"> 1. Database failure, the system will display an error message. 2. When the map service fails, only the property details are shown 3. When property lacks location data, the system will display details with “location unavailable.”
Includes	Search and Filter Properties Sort and Paginate Results
Special Requirements	<p>Map integration: the property card must show essential details</p> <p>The map should load only for the selected property</p> <p>Switching between properties should update the map within 2 seconds</p> <p>The map must always highlight exactly one property (the current selection)</p>
Assumptions	Every property has valid geolocation data
Notes and Issues	None

A. Show Property Rating

Use Case ID	02A
Use Case Name	Show Property Rating

Created By	Isaac Teo	Last Updated By	
Date Created	9 September 2025	Last Date Updated	

Actor	User
Description	The user selects a few amenity types from the amenity filtering function and adjusts the priority of what they value most, and the system calculates a score for each property listing based on their weighted average distance to all the selected amenities.
Preconditions	Property search results are available Property database and map service are operational User has applied amenity search filter
Postconditions	Property listing cards will show the rating score
Priority	High
Frequency of Use	High
Flow of Events	<ol style="list-style-type: none"> After selecting amenity types to filter, user selects the option to show property rating The user can assign weights to the selected amenity types through a slider that they can increase or decrease. The system will calculate the rating score for property listings based on the weighted average of their distance from the closest amenity types selected. The closer the weighted average the higher the property listing score. While property ratings are shown, adjusting the selected amenity type filter or the weights will automatically adjust the rating scores shown to reflect the change. When viewing a selected property listing in page format, the rating score is shown.
Alternative Flows	Nil
Exceptions	<ol style="list-style-type: none"> The system fails to update rating scores after new changes are applied.
Includes	Search and Filter Properties

	Sort and Paginate Results
Special Requirements	Map integration: the map must display the rating score above the property listing The map should load only for the area visible on the screen. When the user moves or expands the map, the map must be able to update in real time to show the scores for the newly visible property listings.
Assumptions	Every property and amenity has valid geolocation data
Notes and Issues	None

III. Manage Favourites

Use Case ID	03		
Use Case Name	Manage Favourites		
Created By	Jarrel Ng	Last Updated By	Isaac Teo
Date Created	1 September 2025	Last Date Updated	7 September 2025

Actor	User
Description	Save and manage favourite properties
Preconditions	User has viewed property listings
Postconditions	Favourites saved across sessions
Priority	Medium
Frequency of Use	Medium
Flow of Events	<ol style="list-style-type: none"> 1. User saves the property to favourites by clicking the star icon, and the system adds it to the favourites list 2. User clicks on the icon to enter the favourites page, and the system redirects them to the favourites menu 3. Clicking on a property in the list brings up the

	<p>options to remove or view the property listing.</p> <ol style="list-style-type: none"> 4. If so desired, the user may click on “remove” to delete the property listing from their favourites list. The system will then remove the property from the menu 5. The deleted property will no longer be treated as a favourited property, and its icon indicator on the interactive map returns to its original icon.
Alternative Flows	<ol style="list-style-type: none"> 1. If the user instead chooses to view the selected property listing, then the system redirects them to the information page for that selected property listing. 2. Choosing to return to the previous page will redirect the user to the favourites page.
Exceptions	<ol style="list-style-type: none"> 1. Database failure, system will display error message.
Includes	Save Property listing
Special Requirements	None
Assumptions	None
Notes and Issues	None

A. Save Property Listing

Use Case ID	03A		
Use Case Name	Save Property Listing		
Created By	Isaac Teo	Last Updated By	Isaac Teo
Date Created	30 August 2025	Last Date Updated	7 September 2025

Actor	User
Description	The user selects a property listing and saves it under favourite properties
Preconditions	None

Postconditions	A new property listing is added to the list of favourite properties, and it gets a pin highlighted on the map
Priority	High
Frequency of Use	High
Flow of Events	<ol style="list-style-type: none"> 1. The user clicks on the property icon on the map, bringing up an informative pop-up about the listing. 2. The user selects the option to save it 3. The listing is added to the favourites list, and the listing gains a “favourited” pin
Alternative Flows	<ol style="list-style-type: none"> 1. The selected listing is already in the favourites list, and the option to save it is greyed out. 2. Clicking on the greyed out button triggers a small pop-up informing the user that this property listing has already been saved.
Exceptions	<ol style="list-style-type: none"> 1. The user attempts to save an already saved property listing, despite the button to do so being greyed out.
Includes	None
Special Requirements	None
Assumptions	The User has an existing Dream Neighbourhood Account.
Notes and Issues	None

IV. Account Creation

Use Case ID	04		
Use Case Name	Create Account		
Created By	Jarrel Ng	Last Updated By	
Date Created	1 September 2025	Last Date Updated	

Actor	User
-------	------

Description	Allows the User to register a new account in Dream Neighbourhood
Preconditions	User has accessed the registration page
Postconditions	User account is created and verified, and is able to log in
Priority	High
Frequency of Use	Medium
Flow of Events	<ol style="list-style-type: none"> 1. The User clicks “Sign Up.” 2. User inputs username, email, and password 3. System shall enforce password strength rules 4. System triggers Verify Email after clicking submit 5. User clicks verification link from their email address that he/she have entered during registration 6. System confirms account creation
Alternative Flows	<ol style="list-style-type: none"> 1. Email is already registered. The system will prompt the user to log in or use another email 2. The password is invalid, and the system will prompt for correction
Exceptions	<ol style="list-style-type: none"> 1. Email fails to deliver to User's email address, allow to resend the verification email
Includes	Verify Email
Special Requirements	Password strength enforcement and email verification are required
Assumptions	The User has a valid email address
Notes and Issues	Ensure email uniqueness and handle network failures during verification

A. Verify Email

Use Case ID	04A		
Use Case Name	Verify Email		
Created By	Jarrel Ng	Last Updated By	
Date Created	1 September 2025	Last Date Updated	

Actor	User
Description	Ensures User's email is valid before activating the account
Preconditions	User has submitted the registration form
Postconditions	User account is verified
Priority	High
Frequency of Use	Medium
Flow of Events	<ol style="list-style-type: none"> 1. The system sends a verification email with a link 2. User clicks the link 3. System validates the token and confirms verification
Alternative Flows	<ol style="list-style-type: none"> 1. Token has expired, System shall allow resend 2. Email is undelivered. System shall allow resend
Exceptions	<ol style="list-style-type: none"> 1. System fails to generate a token and shall log the error and notify the admin
Includes	None
Special Requirements	Email service must be reliable
Assumptions	User checks their email
Notes and Issues	None

V. Authentication

Use Case ID	05		
Use Case Name	Login		
Created By	Jarrel Ng	Last Updated By	Isaac Teo
Date Created	29 August 2025	Last Date Updated	7 September 2025

Actor	User
-------	------

Description	Allows the User to log into his/her Dream Neighbourhood account using his/her email and password.
Preconditions	None
Postconditions	User is logged into his/her Dream Neighbourhood application and is directed to the home page of the application
Priority	High
Frequency of Use	High
Flow of Events	<ol style="list-style-type: none"> 1. The User enters his/her email and password. The password is masked as dots, but the User can choose to unmask it by clicking the eye icon. 2. The User clicks the login button. 3. If the information entered is correct, the user is redirected to the home page of their Dream Neighbourhood account.
Alternative Flows	<ol style="list-style-type: none"> 1. The user enters incorrect information, in which the system prevents them from logging in. 2. The user then chooses to click the reset password option. 3. Redirects to the “Reset Password” use case
Exceptions	<ol style="list-style-type: none"> 2. If any of the required information is missing or invalid, an error message is displayed 3. If the email and password do not match when the User tries to log in in step 2, Dream Neighbourhood shall display “Email and password do not match” to the User
Includes	None
Special Requirements	The system needs to validate user input data
Assumptions	The User has an existing Dream Neighbourhood Account.
Notes and Issues	None

A. Reset Password

Use Case ID	05A
-------------	-----

Use Case Name	Reset Password		
Created By	Jarrel Ng	Last Updated By	
Date Created	1 September 2025	Last Date Updated	

Actor	User
Description	Allows the User to reset a forgotten password.
Preconditions	User has a registered email
Postconditions	User can log in with a new password
Priority	High
Frequency of Use	Low
Flow of Events	<ol style="list-style-type: none"> 1. User selects “Forgot Password” 2. User enters registered email 3. System sends reset link 4. User clicks the link and sets a new password 5. System confirms reset
Alternative Flows	<ol style="list-style-type: none"> 1. Email is not found, the system will prompt the user to try again
Exceptions	<ol style="list-style-type: none"> 1. Link has expired, the system shall allow resend
Includes	None
Special Requirements	None
Assumptions	None
Notes and Issues	None

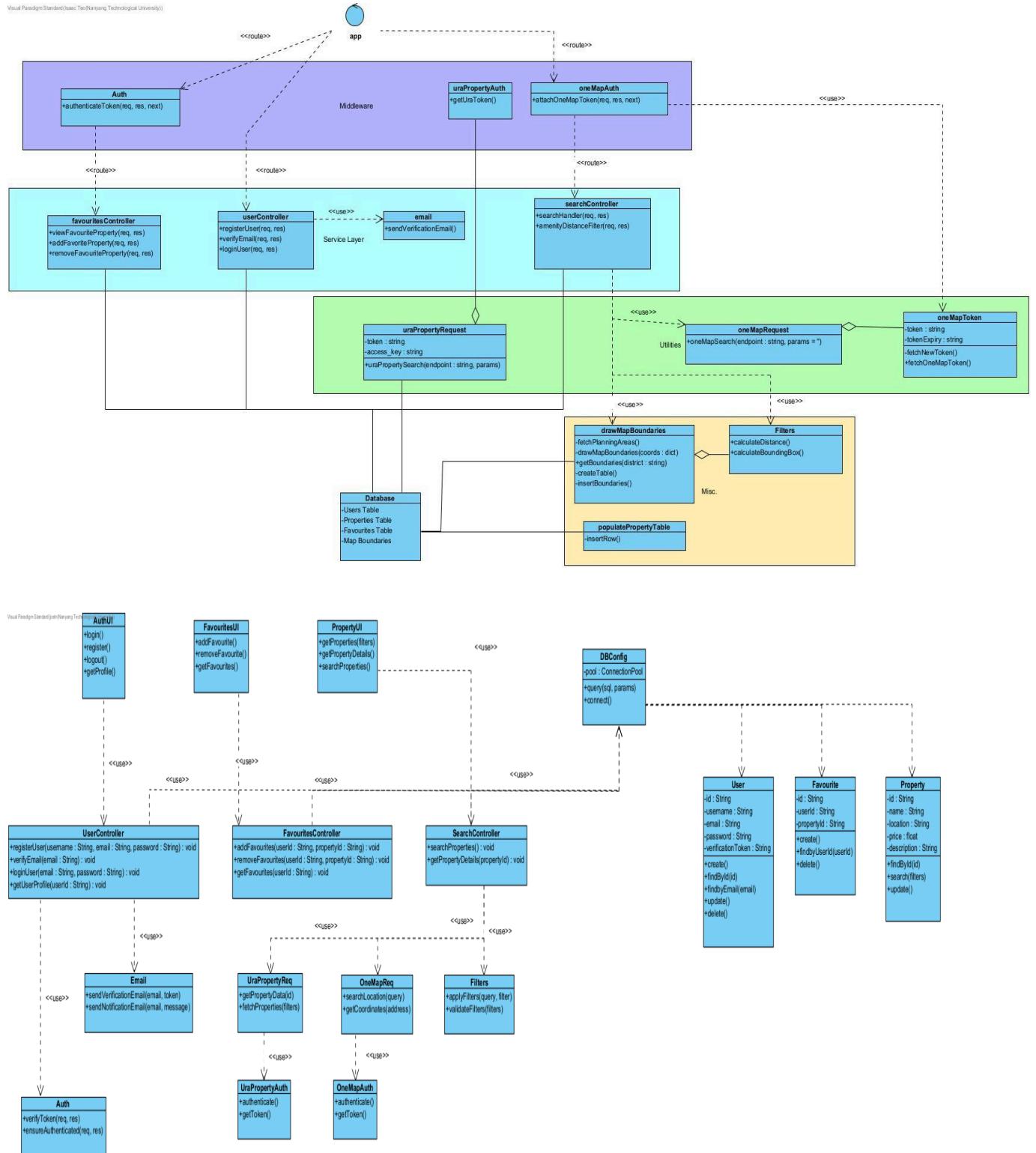
VI. Handle Session Timeout

Use Case ID	06A		
Use Case Name	Handle Session Timeout		
Created By	Jarrel Ng	Last Updated By	
Date Created	8 September 2025	Last Date Updated	

Actor	User
Description	System manages inactivity by prompting the user.
Preconditions	User logged in.
Postconditions	Session extended or ended.
Priority	Medium
Frequency of Use	Medium
Flow of Events	<ol style="list-style-type: none"> 1. User is inactive for a set time 2. System displays a timeout prompt, “The session has ended, do you want to extend?” 3. If user clicks the yes button, the session is extended 4. If user clicks the no button, the session ends
Alternative Flows	None
Exceptions	None
Includes	None
Special Requirements	Default timeout is 15 minutes
Assumptions	User's internet connection is stable
Notes and Issues	None

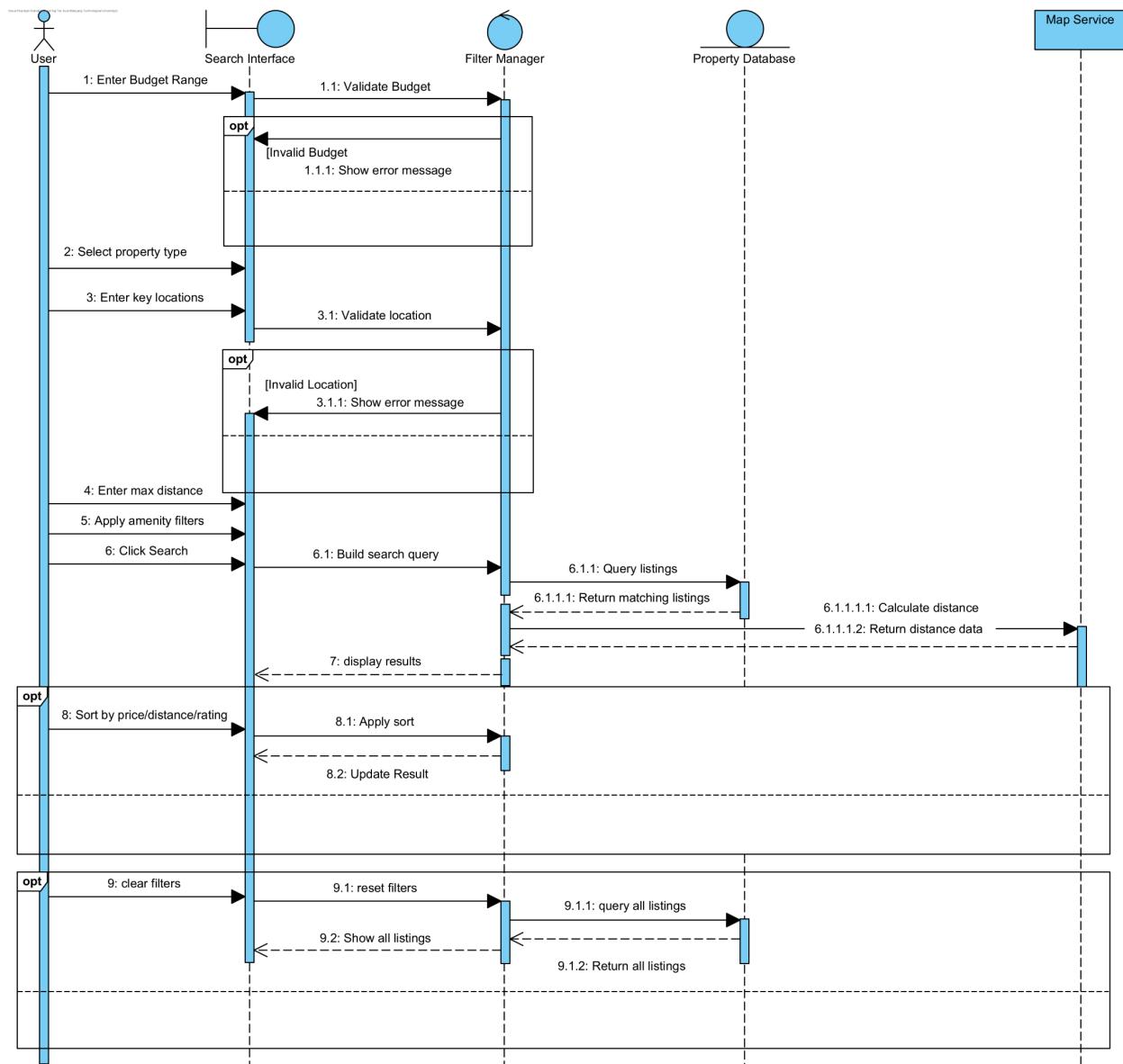
Class Diagram(s)

Backend Class diagram:

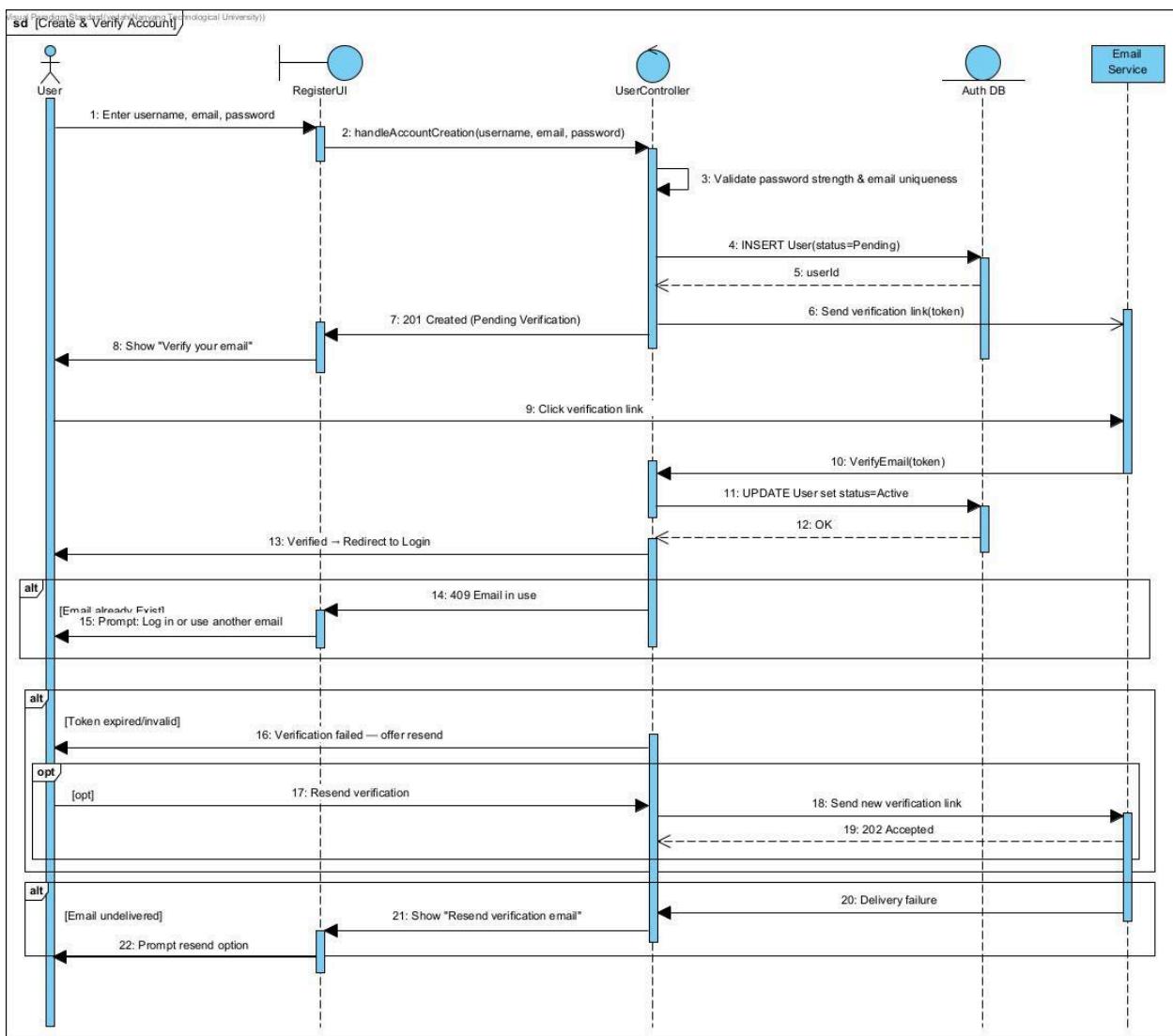


Sequence Diagram

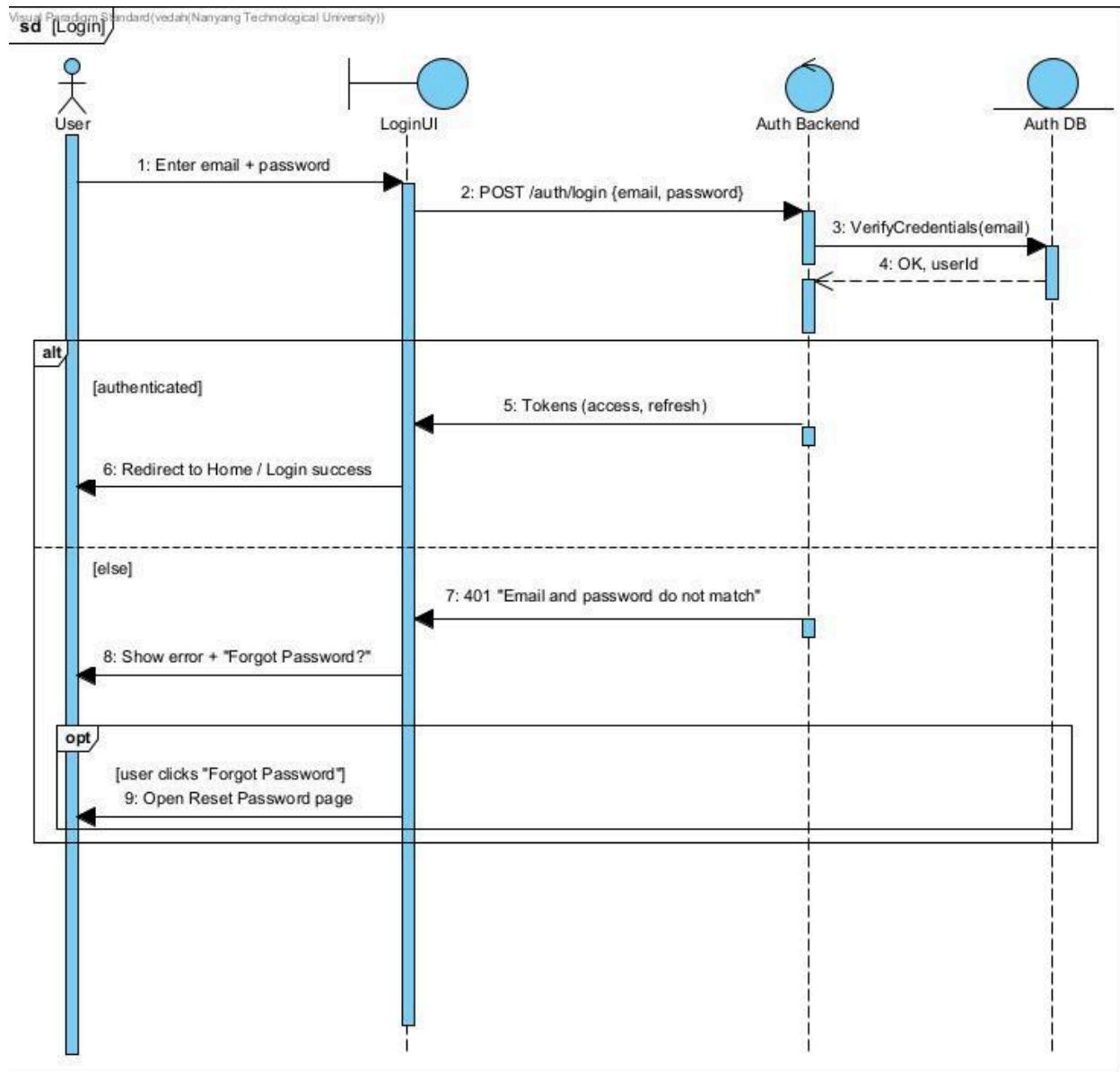
Search and filter properties:



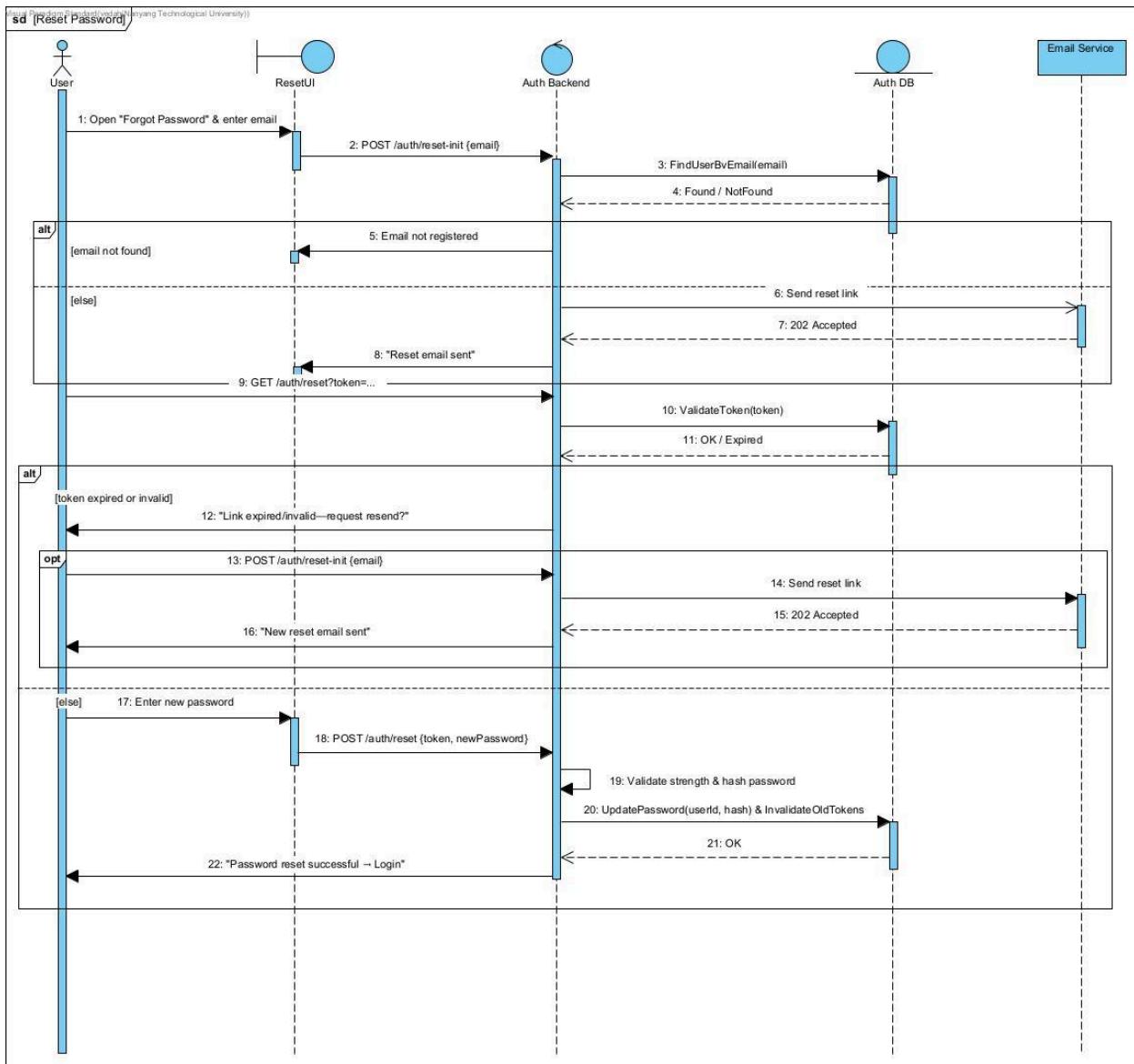
Create and Verify Account



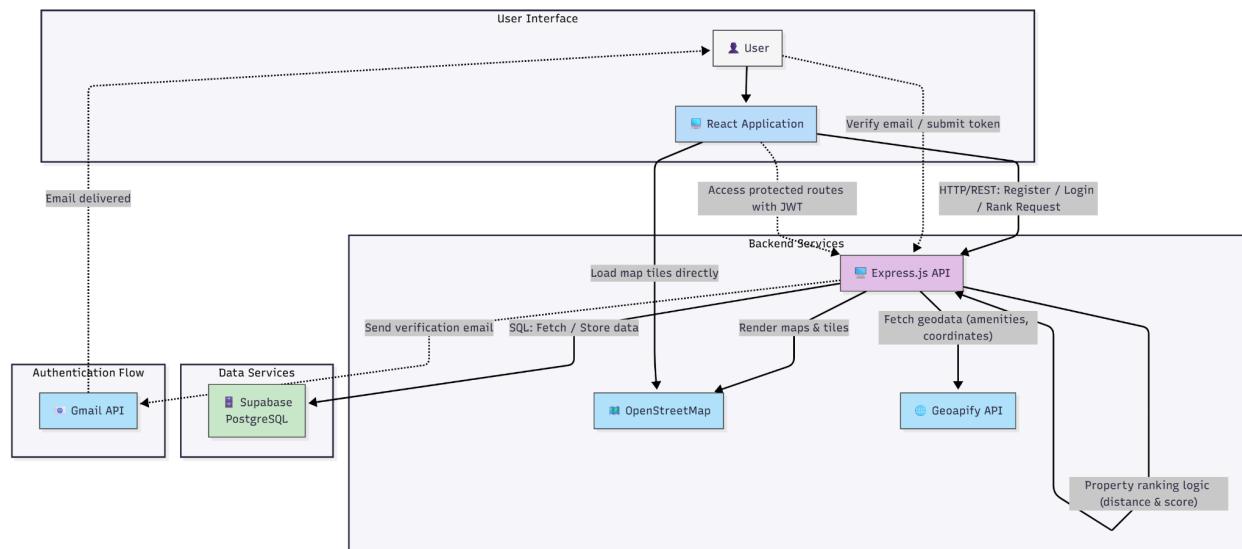
Login



Reset Password



System Architecture Diagram



Dialog Map

