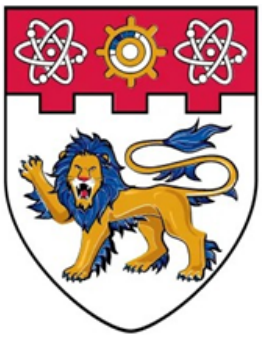


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SC2006 – Software Engineering

Lab 2 Deliverables

Lab Group	SCS6
Team	5
Members	Nguyen Le Tam (U2420673E)
	Mehul Modi (U2423434H)
	Nguyen Tran Chien (U2420243A)
	Anthea Toh (U2423691F)
	Chan Tzen Loong Max (U2422224B)

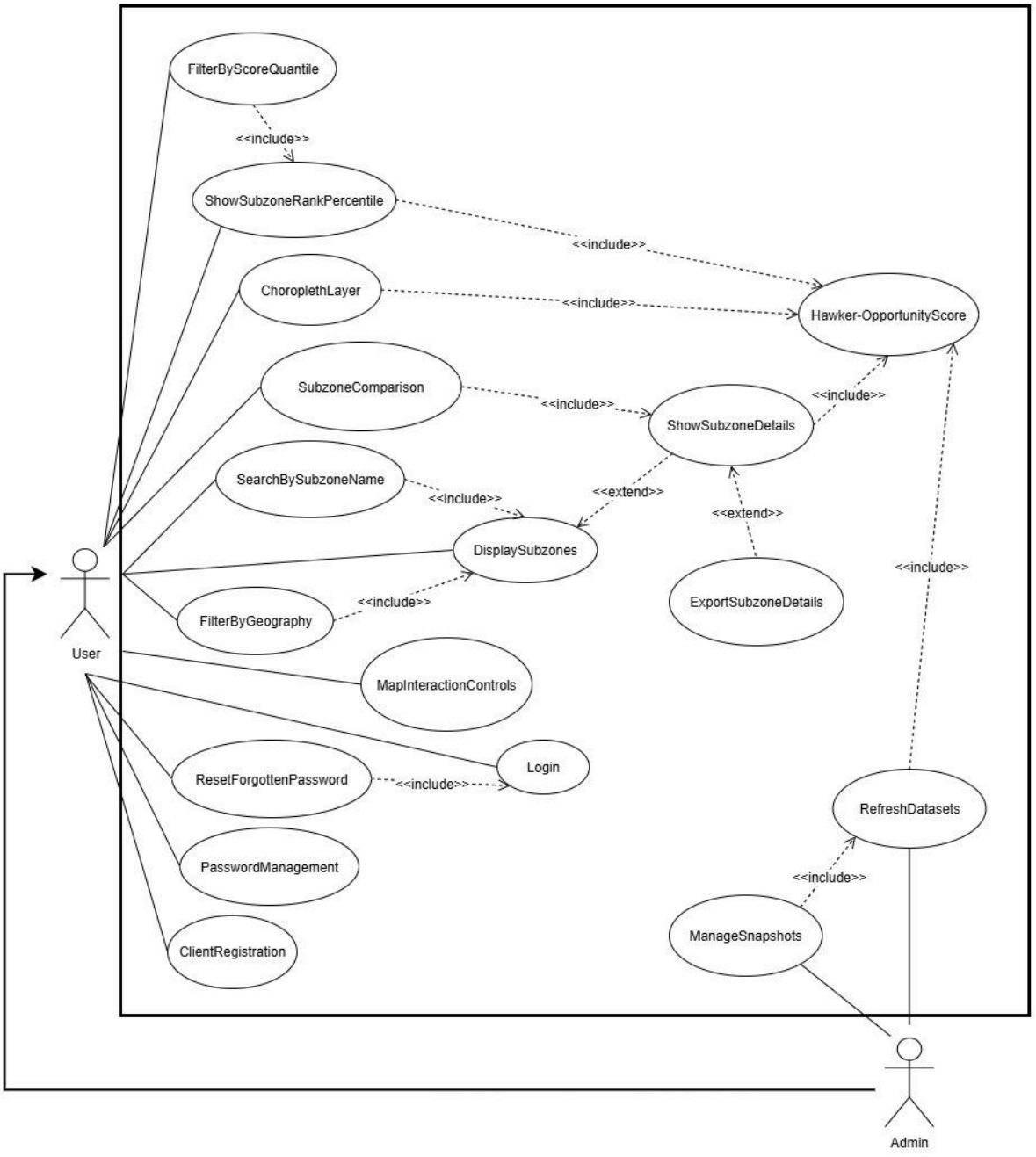
Table of Contents

1. Use Case diagram and Use Case description..... 3
A. Use Case Diagram..... 3
B. Use Case Descriptions..... 4
1. For Functional Requirement #1..... 4
1.1. DisplaySubzones..... 4
1.2. ChoroplethLayer..... 5
1.3. MapInteractionControls..... 6
2. For Functional Requirement #2..... 7
2.1. Hawker-OpportunityScore..... 7
2.2. ShowSubzoneRankPercentile..... 8
3. For Functional Requirement #3..... 9
3.1. FilterByGeography..... 9
3.2. FilterByScoreQuantile..... 10
3.3. SearchBySubzoneName..... 11
4. For Functional Requirement #4..... 12
4.1. ShowSubzoneDetails..... 12
4.2. SubzoneComparison..... 13
5. For Functional Requirement #5..... 14
5.1. RefreshDatasets (Admin)..... 14
5.2. ManageSnapshots (Admin)..... 15
5.3. ExportSubzoneDetails..... 16
6. For Functional Requirement #6..... 17
6.1. ClientRegistration..... 17
6.2. UserLogin..... 18
6.3. PasswordManagement..... 19
6.4. ResetForgottenPassword..... 20
2. Class Diagram of Entity Classes..... 21
3. Key Boundary Classes and Control Classes..... 21
4. Sequence Diagrams of Use Cases..... 23
I. For Use Cases Under 1. (Display map)..... 23
1.1 DisplaySubzones..... 23
1.2 Choropleth layer..... 23
1.3 MapInteractionControls..... 24
II. For Use Cases Under 2. (Display score and percentile)..... 26
2.1 Hawker-Opportunity Score..... 26
2.2 ShowSubzoneRankPercentile..... 26
III. For Use Cases Under 3. (Filtering and search)..... 28
3.1 FilterByGeography..... 28
3.2 FilterByScoreQuantile..... 29
3.3 SearchBySubzoneName..... 30
IV. For Functional Requirement #4..... 31
4.1 ShowSubzoneDetails..... 31

4.2 SubzoneComparison (Two subzones side-by-side).....32
V. For Functional Requirement #5.....33
5.1 RefreshDatasets (Admin).....33
5.2 ManageSnapshots (Admin).....34
5.3 ExportSubzoneDetails.....35
VI. For Functional Requirement #6.....36
6.1 ClientRegistration.....36
6.2 UserLogin.....37
6.3 PasswordManagement.....38
6.4 ResetForgottenPassword.....39
5. Initial Dialog Map.....40

1. Use Case diagram and Use Case description

A. Use Case Diagram


FilterByScoreQuantile
<<include>>
ShowSubzoneRankPercentile
<<include>>
ChoroplethLayer
<<include>>
Hawker-OpportunityScore
SubzoneComparison
<<include>>
ShowSubzoneDetails
<<include>>
SearchBySubzoneName
<<include>>
<<extend>>
<<extend>>
DisplaySubzones
ExportSubzoneDetails
<<include>>
FilterByGeography
<<include>>
User
MapInteractionControls
Login
ResetForgottenPassword
<<include>>
RefreshDatasets
PasswordManagement
<<include>>
ClientRegistration
ManageSnapshots
Admin

B. Use Case Descriptions

1. For Functional Requirement #1

1.1. DisplaySubzones

Use Case ID:	1.1		
Use Case Name:	DisplaySubzones		
Created By:	Nguyen Le Tam	Last Updated By:	Nguyen Le Tam
Date Created:	6th September 2025	Date Last Updated:	3th October 2025

Actor:	User
Description:	User views the Singapore map segmented by subzones. Each subzone appears as a polygon boundary on the map.
Preconditions:	The system has loaded URA subzone boundary datasets.
Postconditions:	Subzones are displayed on the map as polygons that can be clicked or hovered.
Priority:	High
Frequency of Use:	Every time a user accesses the system.
Flow of Events:	1. The user opens the application home screen. 2. System loads map base layer. 3. The system overlays polygons of subzones 4. The user sees the polygons drawn on the map.
Alternative Flows:	If the dataset is unavailable, the system shows “Unable to load subzones” and provides retry.
Exceptions:	Map rendering fails due to API or browser issues.
Includes:	None
Special Requirements:	The map must be zoomable and responsive.
Assumptions:	URA datasets are complete and up-to-date.s
Notes and Issues:	Polygons may overlap if datasets are misaligned.

1.2. ChoroplethLayer

Use Case ID:	1.2		
Use Case Name:	ChoroplethLayer		
Created By:	Nguyen Le Tam	Last Updated By:	Nguyen Le Tam
Date Created:	6th September 2025	Date Last Updated:	6th September 2025

Actor:	User
Description:	Users view each subzone shaded according to its Hawker-Opportunity Score.
Preconditions:	Score computation has been completed and stored in the system.
Postconditions:	Subzones are displayed with a colour representing their score.
Priority:	High
Frequency of Use:	Each session when the user interacts with the map.
Flow of Events:	1. The user accesses the home map. 2. The system retrieves scores for each subzone. 3. The system normalizes scores and maps them to a colour gradient. 4. The system applies shading to each polygon. A legend is displayed to explain colour ranges.
Alternative Flows:	If dataset is unavailable, system shows error “Unable to load boundaries” and provides retry
Exceptions:	Rendering fails due to browser or API error.
Includes:	Use case 2.1 Hawker-OpportunityScore
Special Requirements:	Must support zooming and responsiveness.
Assumptions:	The latest snapshot of scores is available.
Notes and Issues:	Polygons may overlap if datasets are misaligned.

1.3. MapInteractionControls

Use Case ID:	1.3		
Use Case Name:	MapInteractionControls		
Created By:	Nguyen Le Tam	Last Updated By:	Nguyen Le Tam
Date Created:	6th September 2025	Date Last Updated:	6th September 2025

Actor:	User
Description:	The user interacts with the Singapore map of subzones.
Preconditions:	The system has loaded subzone polygons and computed scores.
Postconditions:	Subzones can be zoomed, panned, hovered, or selected.
Priority:	High
Frequency of Use:	Every time a user explores the map.
Flow of Events:	1. User zooms in/out to adjust map scale. 2. User pans map to move to another area. 3. User hovers over a polygon.
Alternative Flows:	If map API fails, the user is restricted to default zoom level.
Exceptions:	None
Includes:	None
Special Requirements:	Must support mouse gestures.
Assumptions:	Browser/device supports modern mapping libraries.
Notes and Issues:	Performance may lag if too many polygons are displayed at once.

2. For Functional Requirement #2

2.1. Hawker-OpportunityScore

Use Case ID:	2.1		
Use Case Name:	Hawker-OpportunityScore		
Created By:	Nguyen Le Tam	Last Updated By:	Nguyen Le Tam
Date Created:	6th September 2025	Date Last Updated:	6th September 2025

Actor:	System (triggered indirectly by Admin refresh or initial load).
Description:	The system calculates the Hawker-Opportunity Score (H_i) for each subzone using kernel-smoothed demand, competing-adjusted supply, and transport accessibility.
Preconditions:	Population, hawker centres, MRT stations, and bus stop datasets are loaded and validated.
Postconditions:	Each subzone has stored values for Dem_i, Sup_i, Acc_i, and the final H_i score.
Priority:	High
Frequency of Use:	Whenever datasets are refreshed or recomputed.
Flow of Events:	1. The system retrieves resident counts and their centroid locations. 2. System computes smoothed demand (Dem_i) by convolving population with kernel $K\lambda_D$. 3. System computes supply (Sup_i) by convolving hawker centres with kernel $K\lambda_S$, adjusting each centre's contribution by the demand it already serves. 4. System computes accessibility (Acc_i) by convolving MRT and bus stops with their respective kernels $K\lambda_M$ and $K\lambda_B$, weighted by β_{MRT} and β_{BUS}. 5. System standardizes each component using robust z-scores. 6. System computes the final: $H_i = w_D . Z(Dem_i) - w_S . Z(Sup_i) + w_A . Z(Acc_i)$ 7. Scores are stored in the snapshot with metadata.
Alternative Flows:	A1: If capacity (C□) for a hawker centre is missing, system assumes C□ = 1 A2: If transport weights are not provided, system assumes β_{MRT} = β_{BUS} = 1
Exceptions:	Failure to load one dataset aborts computation; system logs error and retains previous snapshot.
Includes:	None
Special Requirements:	Kernel bandwidth (λ) must be configurable.
Assumptions:	Census and NEA datasets are up-to-date.
Notes and Issues:	Admin may adjust weights (w_D, w_S, w_A) before recomputation.

2.2. ShowSubzoneRankPercentile

Use Case ID:	2.3		
Use Case Name:	ShowSubzoneRankPercentile		
Created By:	Nguyen Le Tam	Last Updated By:	Nguyen Le Tam
Date Created:	6th September 2025	Date Last Updated:	6th September 2025

Actor:	User
Description:	User views how a selected subzone ranks relative to all other subzones in terms of Hawker-Opportunity Score.
Preconditions:	Scores for all subzones have been computed.
Postconditions:	Percentile rank is displayed in the tooltip and subzone details panel.
Priority:	Medium
Frequency of Use:	Occasionally, when comparing subzones or selecting one.
Flow of Events:	1. The user hovers or clicks on a subzone. 2. The system retrieves the percentile rank of the subzone. 3. The system displays “Top X%” or equivalent in the tooltip and details panel.
Alternative Flows:	If percentile cannot be computed (incomplete data), the system hides percentile and shows message “Rank not available”.
Exceptions:	Error in ranking algorithm leads to incorrect percentile displayed.
Includes:	Use case 2.1. Hawker-OpportunityScore
Special Requirements:	Percentiles should be recomputed automatically whenever dataset refresh occurs.
Assumptions:	Ranking is based on the latest snapshot of scores.
Notes and Issues:	Percentile presentation must be clear (e.g., Top 10%, Bottom 50%).

3. For Functional Requirement #3

3.1.FilterByGeography

Use Case ID:	3.1		
Use Case Name:	FilterByGeography		
Created By:	Nguyen Le Tam	Last Updated By:	Nguyen Le Tam
Date Created:	6th September 2025	Date Last Updated:	6th September 2025

Actor:	User
Description:	User filters the map to show only specific subzones within a region.
Preconditions:	The system has loaded all subzone polygons and computed scores.
Postconditions:	Only polygons matching the regions remain visible on the map.
Priority:	Medium
Frequency of Use:	Occasionally, when focusing on a specific region.
Flow of Events:	1. The user opens the filter panel. 2. The user selects a region from the dropdown list. The system highlights and displays only subzones within that region. 3. Optionally, the user selects a subzone for more fine-grained filtering. 4. System updates map view accordingly.
Alternative Flows:	If no region is selected, the system shows all subzones.
Exceptions:	Dropdown fails to load due to missing dataset.
Includes:	Use case 1.1. DisplaySubzones
Special Requirements:	Filter must be applied instantly without requiring page reload.
Assumptions:	Subzones list matches URA official dataset
Notes and Issues:	Must handle user deselection gracefully (restore all subzones).

3.2.FilterByScoreQuantile

Use Case ID:	3.2

Use Case Name:	FilterByScoreQuantile		
Created By:	Nguyen Le Tam	Last Updated By:	Nguyen Le Tam
Date Created:	6th September 2025	Date Last Updated:	6th September 2025

Actor:	User
Description:	User filters the map to show only subzones that fall within a selected score percentile range (e.g., Top 10%, Top 25%).
Preconditions:	Scores and percentile ranks have been computed for all subzones.
Postconditions:	The map updates to display only subzones within the chosen percentile threshold.
Priority:	Medium
Frequency of Use:	Often, when identifying high-potential areas.
Flow of Events:	1. The user opens the filter panel. 2. The user selects a quantile option (Top 10%, Top 25%, Top 50%). 3. The system retrieves a list of subzones that meet the criterion. 4. System updates map to show only those subzones. 5. Legend updates to reflect visible score range.
Alternative Flows:	If the percentile option “All” is chosen, the system restores the full map view.
Exceptions:	If percentile cannot be calculated (missing scores), the system shows the message “Filter unavailable”.
Includes:	Use case 2.2. ShowSubzoneRankPercentile
Special Requirements:	Updates should be dynamic (<1 second delay).
Assumptions:	All scores normalized before applying percentile thresholds.
Notes and Issues:	Must be consistent with percentile values shown in tooltips.

3.3. SearchBySubzoneName

Use Case ID:	3.3
Use Case Name:	SearchBySubzoneName

Created By:	Nguyen Le Tam	Last Updated By:	Nguyen Le Tam
Date Created:	6th September 2025	Date Last Updated:	6th September 2025

Actor:	User
Description:	The user searches for a subzone by entering its name in a search bar.
Preconditions:	The list of subzone names is available in the system.
Postconditions:	The map zooms to and highlights the selected subzone.
Priority:	High
Frequency of Use:	Frequently, when users know exactly which subzone they want to view.
Flow of Events:	1. The user types a subzone name into the search bar. 2. The system provides autocomplete suggestions as user types. 3. The user selects a suggested subzone from the dropdown. 4. The system zooms into and highlights the selected subzone polygon. 5. Tooltip appears with subzone name, score, and percentile rank.
Alternative Flows:	If no matches are found, the system displays “No subzone found”.
Exceptions:	Autocomplete fails due to missing dataset or system error.
Includes:	Use case 1.1. DisplaySubzones
Special Requirements:	Autocomplete must handle both full names (e.g., “Tampines”) and partial input (e.g., “Tam”).
Assumptions:	Subzone names are stored exactly as defined in URA datasets.
Notes and Issues:	Should handle both subzone and Subzone search in one field.

4. For Functional Requirement #4

4.1. ShowSubzoneDetails

Use Case ID:	4.1		
Use Case Name:	ShowSubzoneDetails		
Created By:	Nguyen Le Tam	Last Updated By:	Nguyen Le Tam

Date Created:	6th September 2025	Date Last Updated:	3th October 2025

Actor:	User
Description:	User selects a subzone and views demographics, hawker supply, accessibility, component values, and final Hawker-Opportunity Score in a details panel.
Preconditions:	Subzone polygons and IDs loaded; demographic, hawker, and transport datasets available; scores computed for all subzones.
Postconditions:	Details panel shows population totals and age groups, nearby hawker centres, nearby MRT/bus, component values (Dem, Sup, Acc), final score, and simple charts.
Priority:	High
Frequency of Use:	Often during exploration.
Flow of Events:	1. The user selects a subzone on the map or via search. 2. The system highlights the subzone and opens the details panel. 3. The system retrieves demographics and renders counts and age-group chart. 4. The system counts nearby hawker centres by radius and lists names and distances. 5. The system counts nearby MRT stations and bus stops by radius and lists names/codes and distances 6. The system loads Dem, Sup, Acc, and final score and renders small visuals (bars or badges).
Alternative Flows:	● Demographic data missing → show “Demographics unavailable.” ● Hawker dataset outdated → show “Data not refreshed—counts may be inaccurate.” ● MRT data missing → show “MRT data unavailable.” ● Bus data missing → show “Bus stop data unavailable.”
Exceptions:	Geospatial query or distance calculation fails → skip metric, log error, show placeholder.
Includes:	Use case 1.1 DisplaySubzones
Special Requirements:	Trays must remain visible and persistent during navigation.
Assumptions:	Users understand the limit of two subzones.
Notes and Issues:	Consider allowing more than two subzones in future versions.

4.2. SubzoneComparison

Use Case ID:	5.1		
Use Case Name:	SubzoneComparison		
Created By:	Nguyen Le Tam	Last Updated By:	Nguyen Le Tam
Date Created:	6th September 2025	Date Last Updated:	3th October 2025

Actor:	User
Description:	User views demand, supply, accessibility, and scores for two selected subzones displayed side-by-side.
Preconditions:	Subzone data, details, and scores are loaded.
Postconditions:	Comparison tray holds selected subzones and comparison view displays their metrics.
Priority:	High
Frequency of Use:	Occasionally, during evaluation of candidate sites.
Flow of Events:	1. User selects a subzone on the map or via search. 2. System highlights the subzone and shows “Add to Compare.” 3. User adds subzone to the comparison tray. User repeats with another subzone. 4. When two subzones are in tray, “Compare” becomes available. 5. User clicks “Compare.” 6. System opens comparison view with both subzones shown in parallel columns. 7. System renders radar chart and/or tables showing differences in demand, supply, accessibility, and final score.
Alternative Flows:	● If same subzone is added twice → system blocks duplicate entry. ● If more than two are added → system shows “Maximum two subzones allowed.” ● If only one subzone selected → tray remains but “Compare” button disabled. ● If data missing for one → system shows “Data not available” for that metric.
Exceptions:	If rendering fails, the system falls back to a tabular-only comparison.
Includes:	None
Special Requirements:	Charts must be responsive and exportable.
Assumptions:	Percentile ranks are recomputed at the same time as scores.
Notes and Issues:	Labels and colours must clearly distinguish subzones

5. For Functional Requirement #5

5.1. RefreshDatasets (Admin)

Use Case ID:	5.1		
Use Case Name:	RefreshDatasets		
Created By:	Nguyen Le Tam	Last Updated By:	Nguyen Le Tam
Date Created:	6th September 2025	Date Last Updated:	6th September 2025

Actor:	Admin
Description:	Admin triggers a refresh of official datasets (population, hawker centres, MRT/bus stops) and recomputes scores.
Preconditions:	Admin is logged in and authorized.
Postconditions:	A new dataset is loaded, scores are recomputed, and a snapshot is stored.
Priority:	High
Frequency of Use:	Occasionally, when datasets are updated (e.g., monthly or quarterly).
Flow of Events:	1. Admin logs into the system. 2. Admin navigates to the Admin Console. 3. Admin clicks “Refresh Data”. 4. The system fetches the latest official datasets. 5. The system recomputes scores for all subzones. 6. The system creates and saves a new snapshot with version notes and timestamp.
Alternative Flows:	A1: If a dataset cannot be retrieved, the system keeps the last valid version and logs an error. A2: If recomputation partially fails, system shows “Partial refresh completed – some scores unavailable”.
Exceptions:	Internet or API failure prevents fetching datasets.
Includes:	Use case 2.1. Hawker-OpportunityScore
Special Requirements:	Only Admin accounts may execute this function.
Assumptions:	Official datasets are accessible at data.gov.sg.
Notes and Issues:	Admin must verify refresh success via system logs.

5.2. ManageSnapshots (Admin)

Use Case ID:	5.2		
Use Case Name:	ManageSnapshots		
Created By:	Nguyen Le Tam	Last Updated By:	Nguyen Le Tam
Date Created:	6th September 2025	Date Last Updated:	6th September 2025

Actor:	Admin
Description:	Admin manages snapshots of score calculations, including viewing versions, rolling back, and verifying dataset notes.
Preconditions:	At least one snapshot has been created previously.
Postconditions:	Admin can view or restore specific snapshots.
Priority:	Medium
Frequency of Use:	Occasionally, when auditing or verifying calculations.
Flow of Events:	1. Admin logs into Admin Console. 2. Admin opens the “Snapshots” section. 3. The system displays a list of snapshots with timestamp, dataset versions, and notes. 4. Admin selects a snapshot to view or restore. 5. If restored, the system reverts scores to that snapshot’s values.
Alternative Flows:	If no snapshots are available, the system displays “No snapshots found”.
Exceptions:	Rollback fails due to a corrupted snapshot file.
Includes:	Use Case 6.1 RefreshDatasets
Special Requirements:	Snapshots must be archived for at least 30 days.
Assumptions:	Dataset versions are recorded correctly in metadata.
Notes and Issues:	Consider long-term archival beyond 30 days for traceability.

5.3. ExportSubzoneDetails

Use Case ID:	5.3		
Use Case Name:	ExportSubzoneDetails		
Created By:	Nguyen Le Tam	Last Updated By:	Nguyen Le Tam
Date Created:	6th September 2025	Date Last Updated:	3th October 2025

Actor:	User
Description:	User exports the current subzone details as a PDF or PNG file with dataset version and timestamp included.
Preconditions:	A subzone or map view has been loaded successfully.
Postconditions:	File is generated and downloaded to the user’s device.
Priority:	Medium
Frequency of Use:	Occasionally, for reporting or presentations.
Flow of Events:	1. The user clicks the “Export” button on the subzone details page. 2. The system collects current subzone state, filters applied, and visible details. 3. The system generates export content with subzone, legend, and sidebar details. 4. System appends metadata (timestamp, dataset version, project name). 5. The system generates files in chosen format (PDF/PNG). 6. The file is downloaded to the user's device.
Alternative Flows:	If export fails, the system displays “Export unsuccessful – please try again.”
Exceptions:	Browser blocks file download; system prompts user to allow it.
Includes:	None
Special Requirements:	Export must preserve readability (legends, labels, scale).
Assumptions:	The user device supports file downloads.
Notes and Issues:	Consider CSV export in future to allow raw data download.

6. For Functional Requirement #6

6.1. ClientRegistration

Use Case ID:	6.1		
Use Case Name:	ClientRegistration		
Created By:	Nguyen Le Tam	Last Updated By:	Nguyen Le Tam
Date Created:	6th September 2025	Date Last Updated:	3th October 2025

Actor:	Client
Description:	Client registers for an account to access system features.
Preconditions:	The client does not already have an account with the same email.
Postconditions:	A new client account is created with role = “Client.”
Priority:	High
Frequency of Use:	Once per client (account creation).
Flow of Events:	1. User action: Client navigates to the registration page. System response: Displays registration form. 2. User action: Client enters name, email, contact number, and password. System response: Validates input fields. 3. System action: Checks if email already exists. 4. System action: Hashes password and creates the account. 5. System response: Displays success message and confirms registration.
Alternative Flows:	If email already registered → system shows “Email already registered.”
Exceptions:	Database error → system shows “Registration failed, please try again.”
Includes:	None
Special Requirements:	Passwords must meet policy requirements (length, character diversity).
Assumptions:	The user provides valid contact details.
Notes and Issues:	The option to add a profile picture may be added later.

6.2. UserLogin

Use Case ID:	6.2		
Use Case Name:	Login		
Created By:	Nguyen Le Tam	Last Updated By:	Nguyen Le Tam
Date Created:	6th September 2025	Date Last Updated:	6th September 2025

Actor:	User
Description:	User logs into the system using email and password.
Preconditions:	A valid account exists.
Postconditions:	User is authenticated and redirected to the home map view.
Priority:	High
Frequency of Use:	Every time a session begins.
Flow of Events:	1. User action: User navigates to login page. System response: Displays login form. 2. User action: User enters email and password. System response: Masks password input. 3. System action: Validates credentials against stored data. 4. System response: If valid, system logs in user and redirects to map view.
Alternative Flows:	If credentials invalid → system shows “Email and password do not match.”
Exceptions:	Authentication server unavailable → system shows “Login unavailable.”
Includes:	None
Special Requirements:	The session must expire after 30 minutes of idle time.
Assumptions:	The user provides correct credentials.
Notes and Issues:	Consider adding 2FA for Admin accounts in future.

6.3. PasswordManagement

Use Case ID:	6.3		
Use Case Name:	PasswordManagement		
Created By:	Nguyen Le Tam	Last Updated By:	Nguyen Le Tam
Date Created:	6th September 2025	Date Last Updated:	6th September 2025

Actor:	User
Description:	Change an existing password while signed in.
Preconditions:	Valid authenticated session.
Postconditions:	Password is updated
Priority:	Medium
Frequency of Use:	Occasionally, when password reset is needed.
Flow of Events:	1. User action: User navigates to Settings → “Change Password.” System response: Displays change password form. 2. User action: User enters current password and new password. System action: Verifies current password. 3. System action: Validates new password against policy. 4. System response: Updates password, invalidates other sessions, and confirms change.
Alternative Flows:	Wrong current password → “Incorrect current password.” New password fails policy
Exceptions:	Identity service unavailable.
Includes:	None
Special Requirements:	Passwords must be hashed securely before storage.
Assumptions:	Stable network and session storage.
Notes and Issues:	Consider notifying users by email after change.

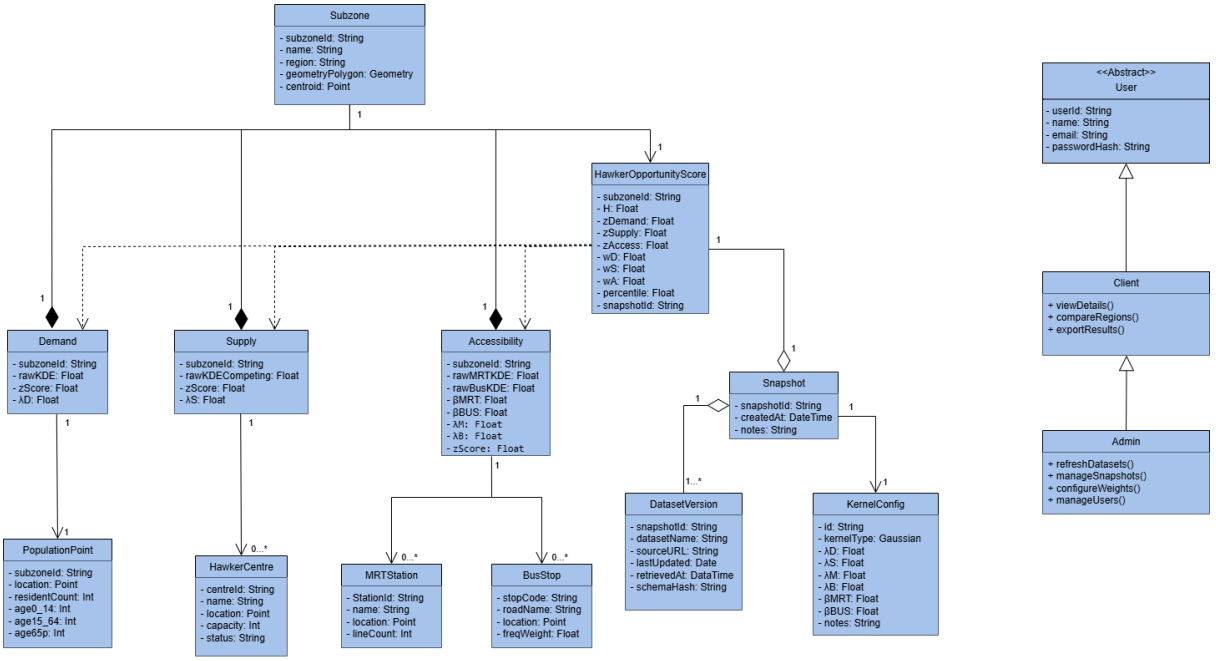
6.4. ResetForgottenPassword

Use Case ID:	6.4		
Use Case Name:	ResetForgottenPassword		
Created By:	Nguyen Le Tam	Last Updated By:	Nguyen Le Tam
Date Created:	3th October 2025	Date Last Updated:	3th October 2025

Actor:	User
Description:	User resets password when it is forgotten
Preconditions:	Account exists with a valid registered email
Postconditions:	Password is reset and user can log in with the new one
Priority:	Medium
Frequency of Use:	Occasionally, when the password is forgotten.
Flow of Events:	1. User action: User clicks “Forgot Password” on login page. System response: Prompts for registered email. 2. User action: User enters email. System action: Sends reset link with a one-time token valid for 15 minutes. 3. User action: User clicks reset link. System response: Displays reset password form. 4. User action: User enters new password. System action: Validates new password against policy, updates account, and confirms reset.
Alternative Flows:	If reset token expires → system shows “Link expired—request new reset.”
Exceptions:	Email server fails to send reset link → system shows “Unable to send reset email.”
Includes:	Use Case 7.2. UserLogin
Special Requirements:	Passwords must be hashed securely before storage.
Assumptions:	User has access to registered email inbox.
Notes and Issues:	Consider notifying users by email after change.

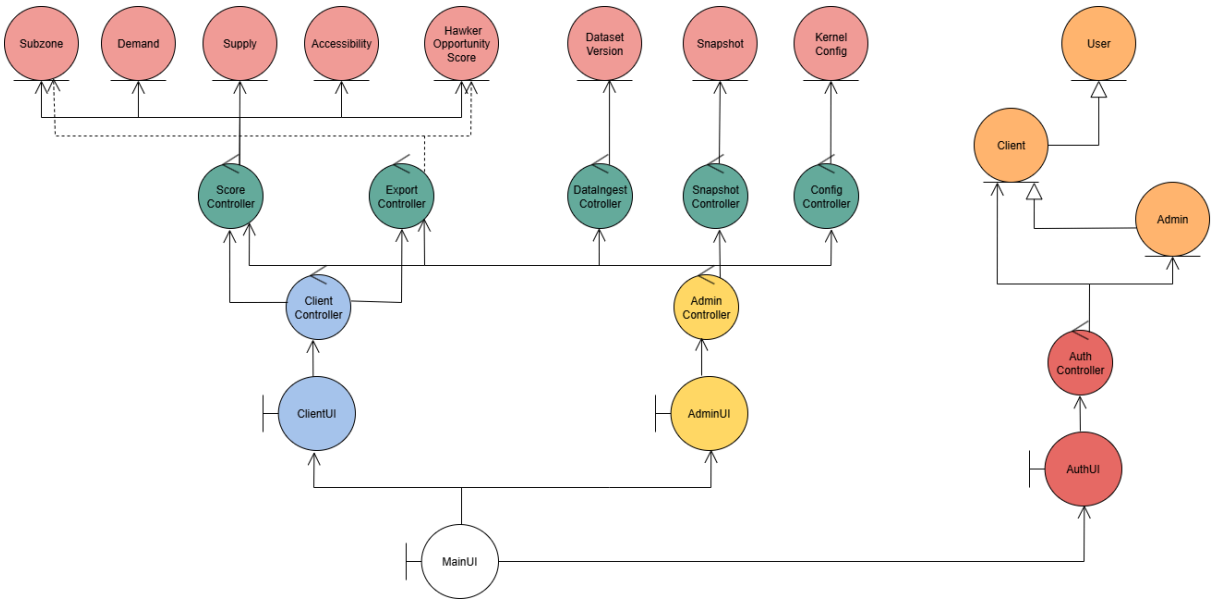
2. Class Diagram of Entity Classes

If the image is unclear, please refer to the pdf file that is uploaded together with this document.



3. Key Boundary Classes and Control Classes

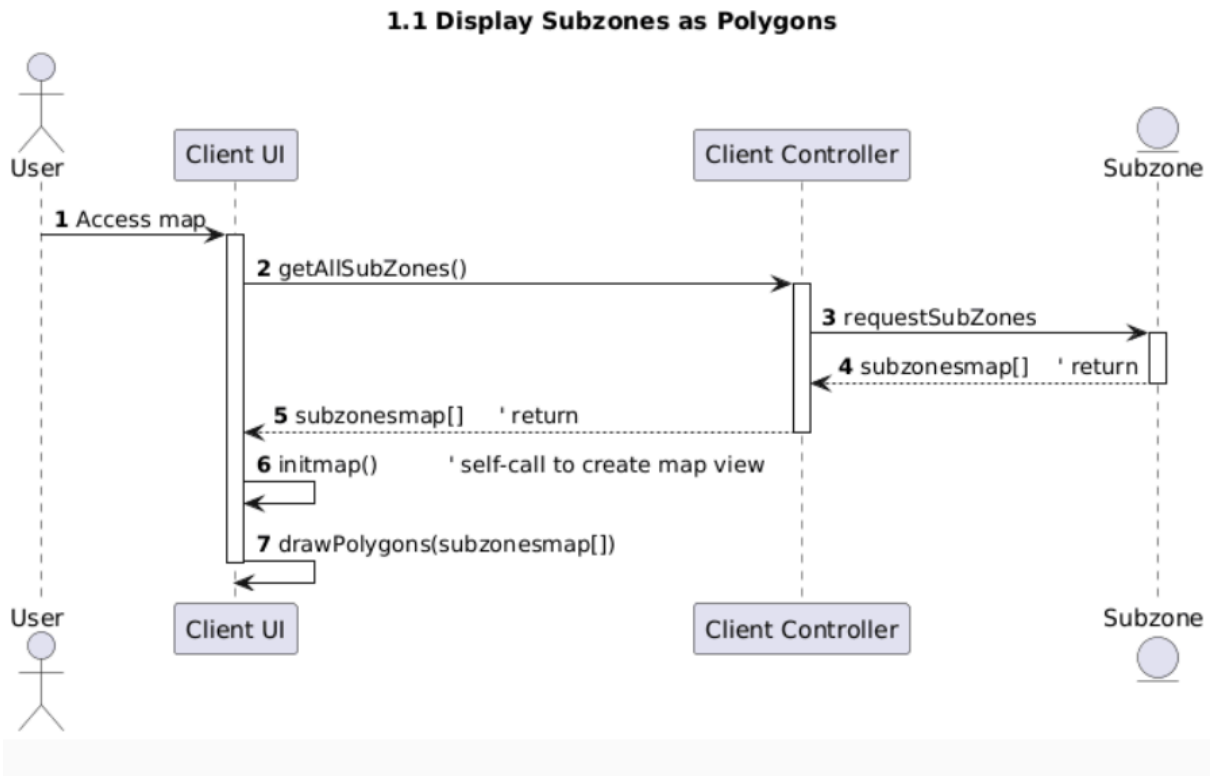
If the image is unclear, please refer to the pdf file that is uploaded together with this document.



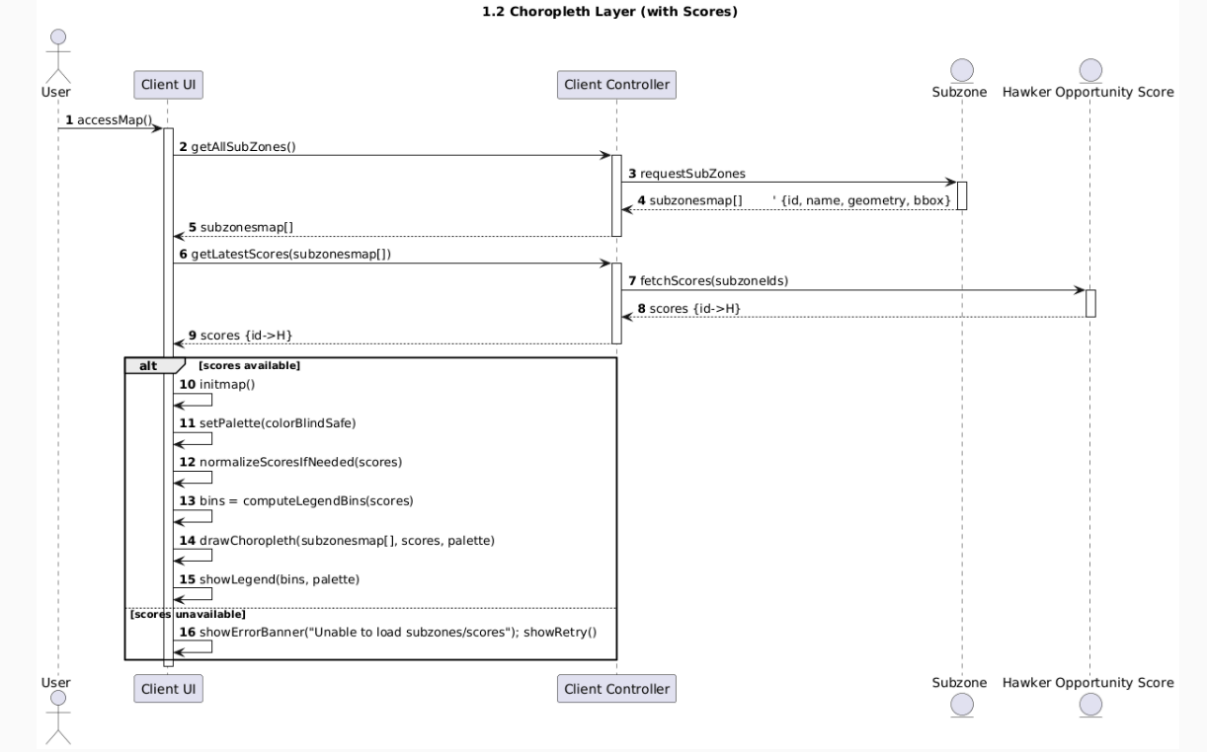
4. Sequence Diagrams of Use Cases

I. For Use Cases Under 1. (Display map)

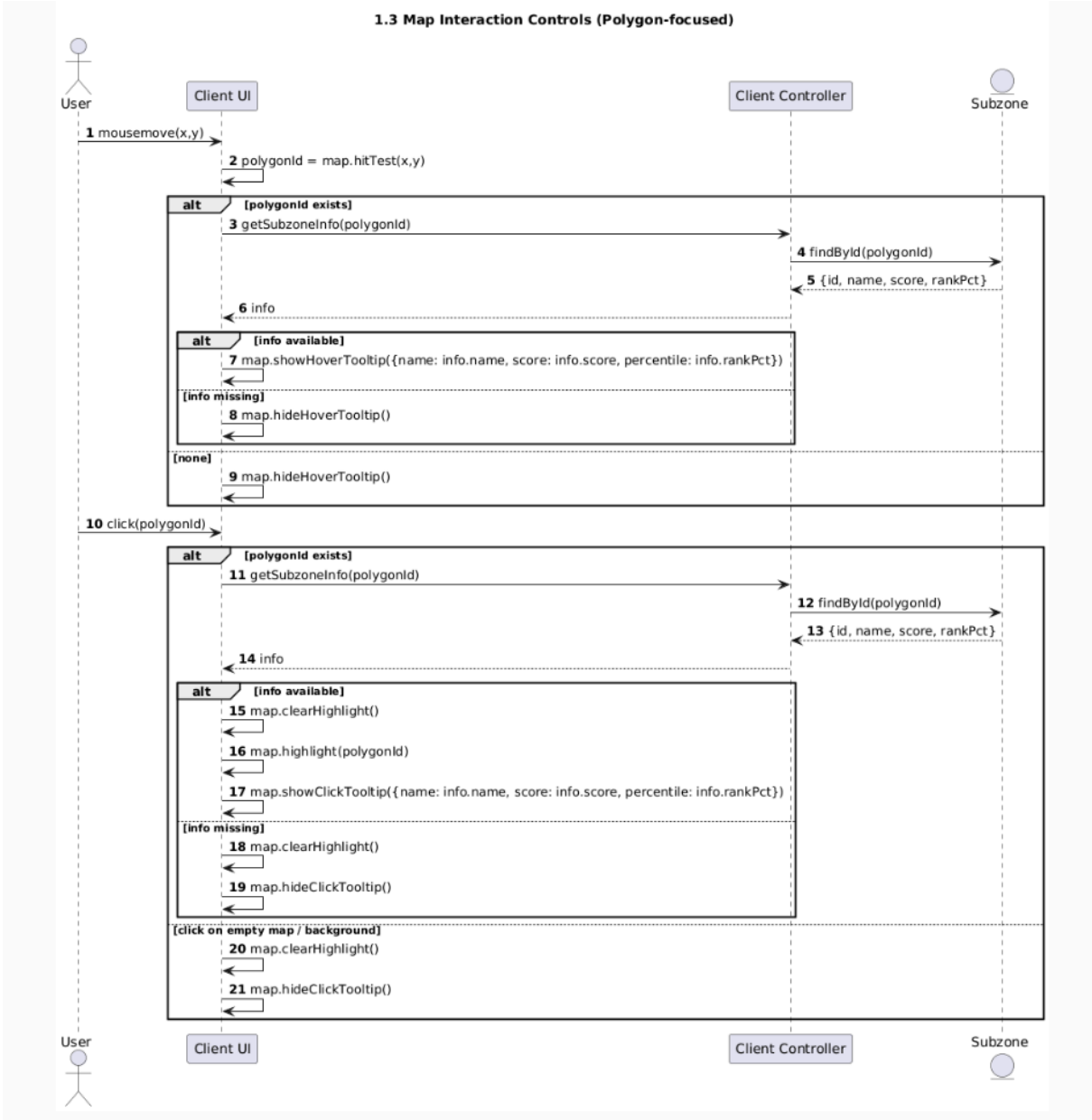
1.1 DisplaySubzones



1.2 Choropleth layer

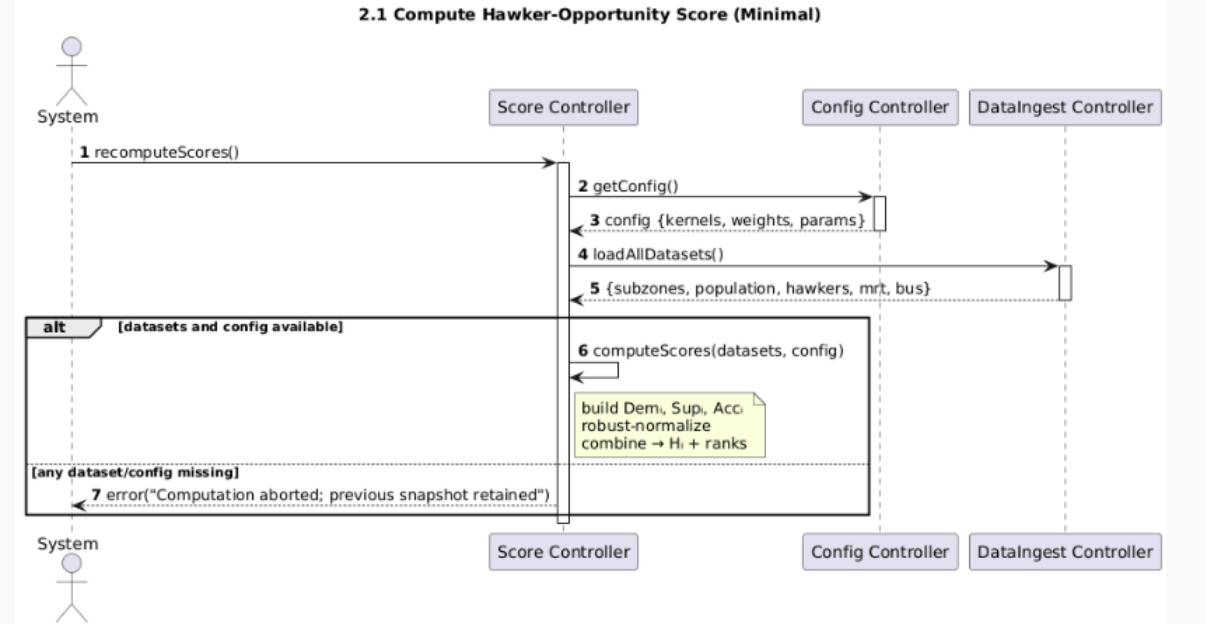


1.3 MapInteractionControls

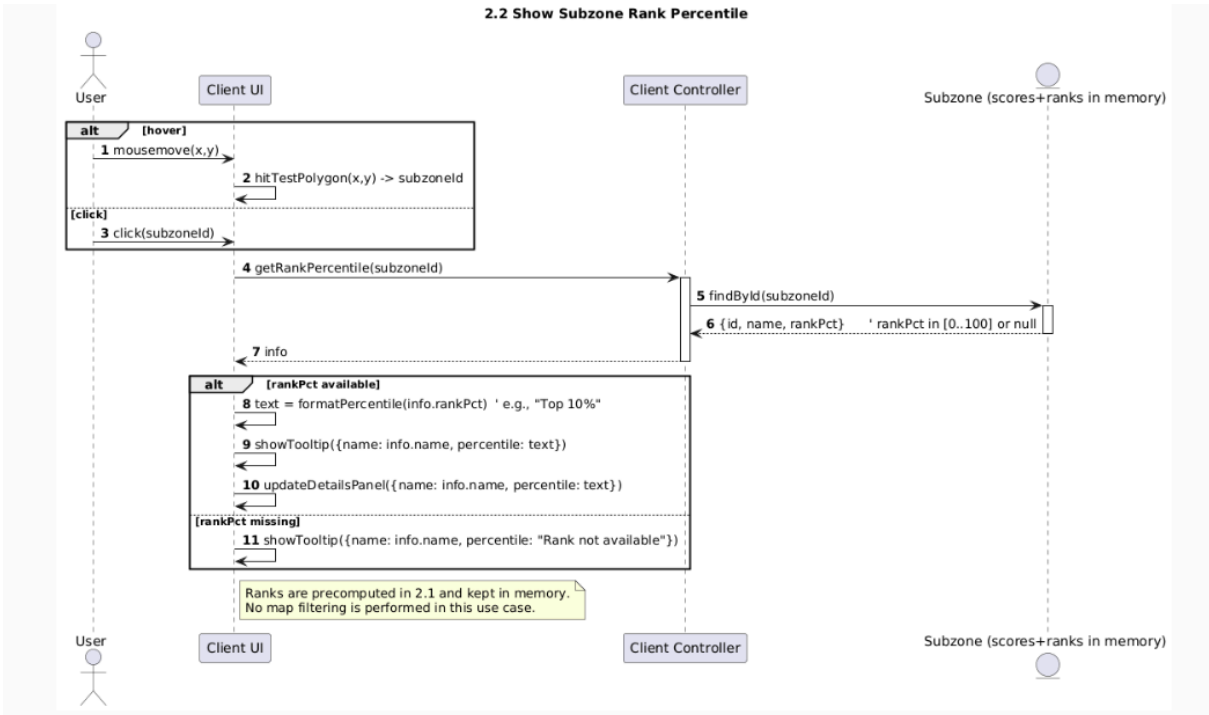

1.3 Map Interaction Controls (Polygon-focused)
User
Client UI
Client Controller
Subzone
1 mousemove(x,y)
2 polygonId = map.hitTest(x,y)
alt
[polygonId exists]
3 getSubzoneInfo(polygonId)
4 findById(polygonId)
5 {id, name, score, rankPct}
6 info
alt
[info available]
7 map.showHoverTooltip({name: info.name, score: info.score, percentile: info.rankPct})
[info missing]
8 map.hideHoverTooltip()
[none]
9 map.hideHoverTooltip()
10 click(polygonId)
alt
[polygonId exists]
11 getSubzoneInfo(polygonId)
12 findById(polygonId)
13 {id, name, score, rankPct}
14 info
alt
[info available]
15 map.clearHighlight()
16 map.highlight(polygonId)
17 map.showClickTooltip({name: info.name, score: info.score, percentile: info.rankPct})
[info missing]
18 map.clearHighlight()
19 map.hideClickTooltip()
[click on empty map / background]
20 map.clearHighlight()
21 map.hideClickTooltip()
User
Client UI
Client Controller
Subzone

II. For Use Cases Under 2. (Display score and percentile)

2.1 Hawker-Opportunity Score

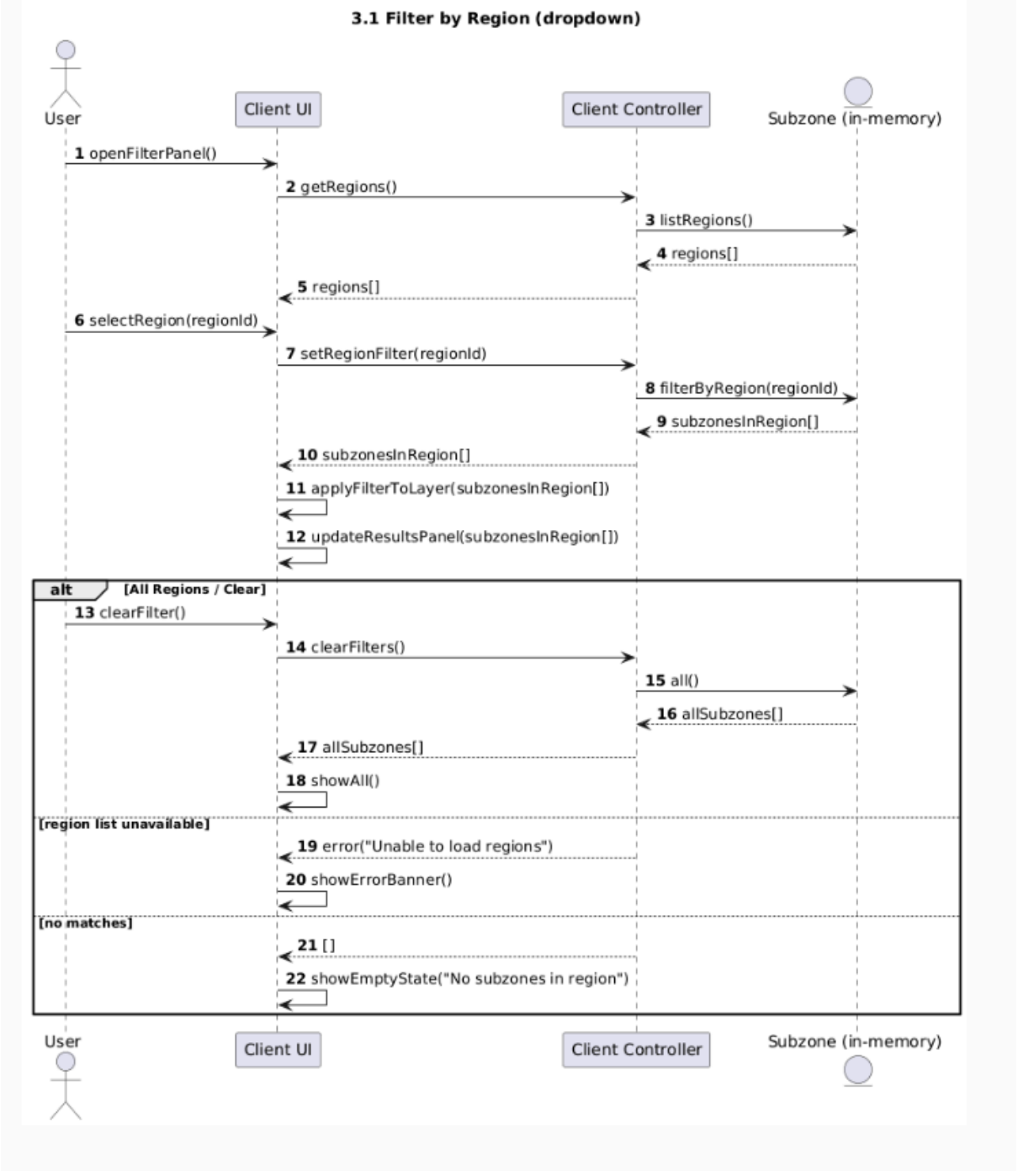


2.2 ShowSubzoneRankPercentile

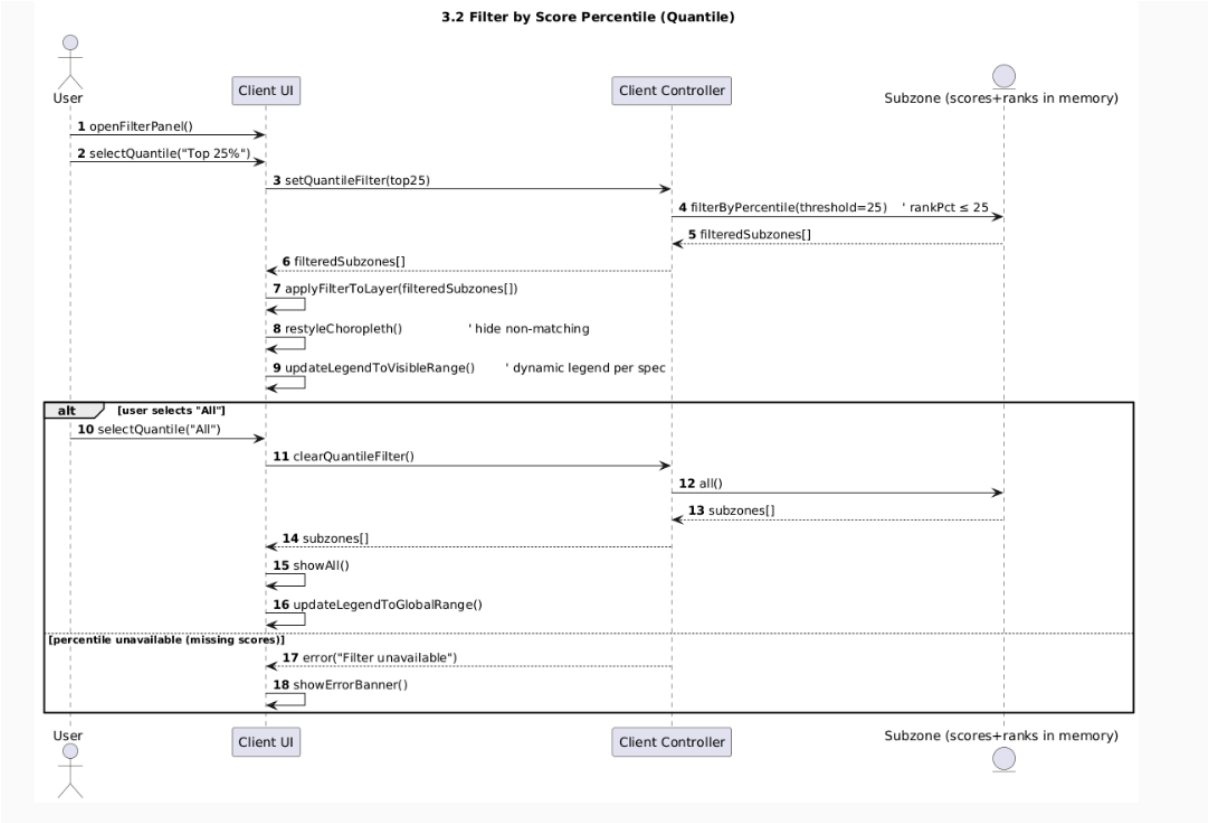

2.2 Show Subzone Rank Percentile
User
Client UI
Client Controller
Subzone (scores+ranks in memory)
alt [hover]
1 mousemove(x,y)
2 hitTestPolygon(x,y) -> subzoneId
[click]
3 click(subzoneId)
4 getRankPercentile(subzoneId)
5 findById(subzoneId)
6 {id, name, rankPct} ' rankPct in [0..100] or null
7 info
alt [rankPct available]
8 text = formatPercentile(info.rankPct) ' e.g., "Top 10%"
9 showTooltip({name: info.name, percentile: text})
10 updateDetailsPanel({name: info.name, percentile: text})
[rankPct missing]
11 showTooltip({name: info.name, percentile: "Rank not available"})
Ranks are precomputed in 2.1 and kept in memory.
No map filtering is performed in this use case.
User
Client UI
Client Controller
Subzone (scores+ranks in memory)

III. For Use Cases Under 3. (Filtering and search)

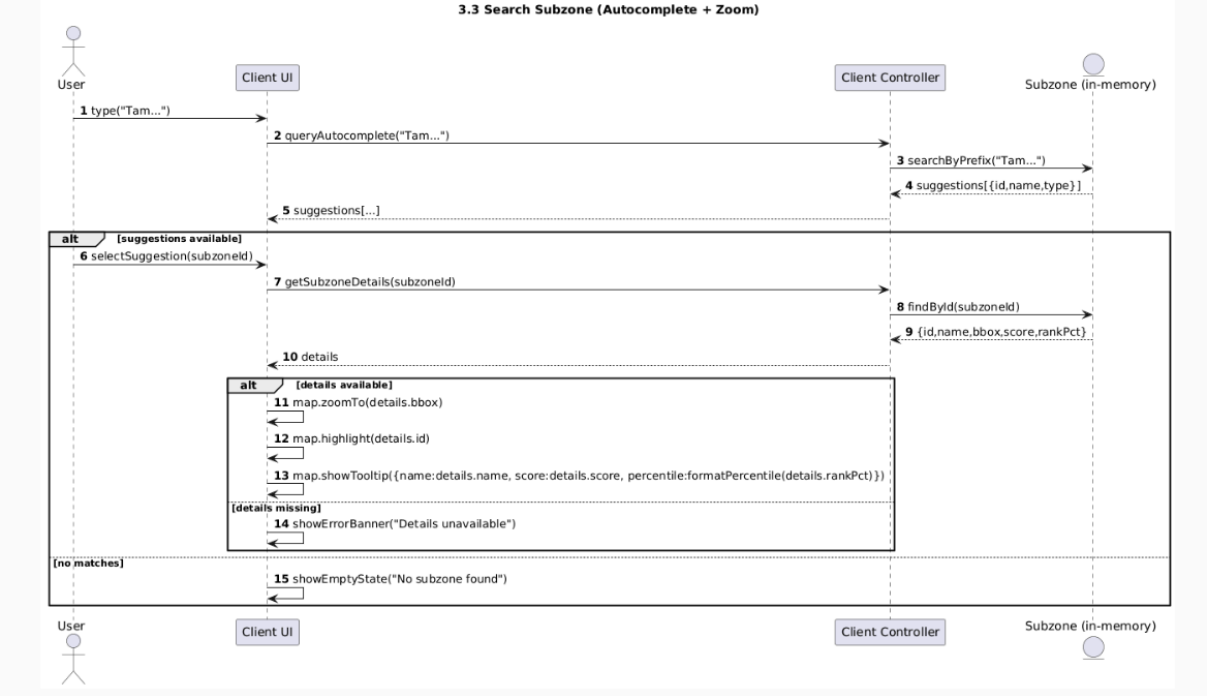
3.1 FilterByGeography



3.2 FilterByScoreQuantile

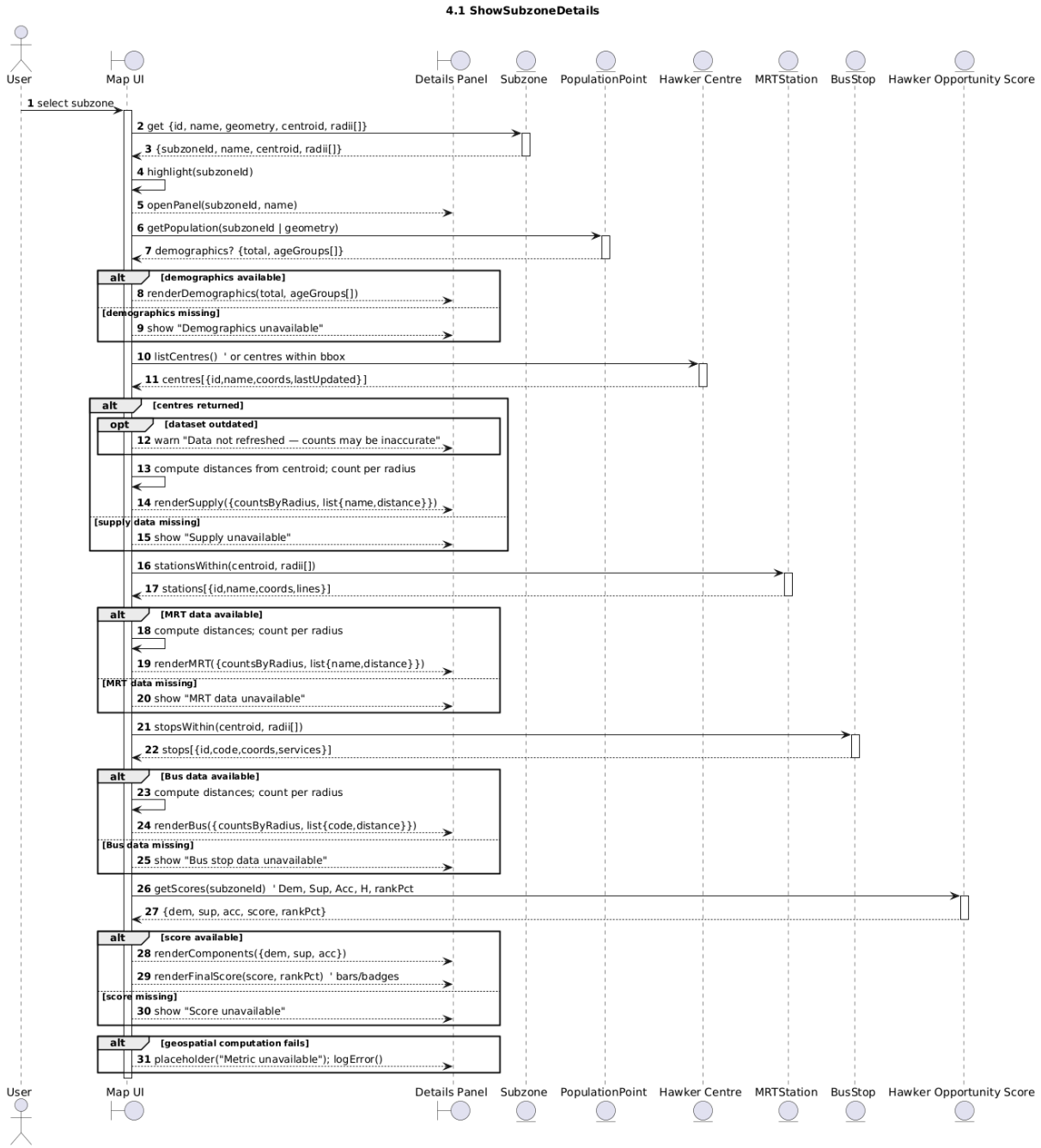


3.3 SearchBySubzoneName

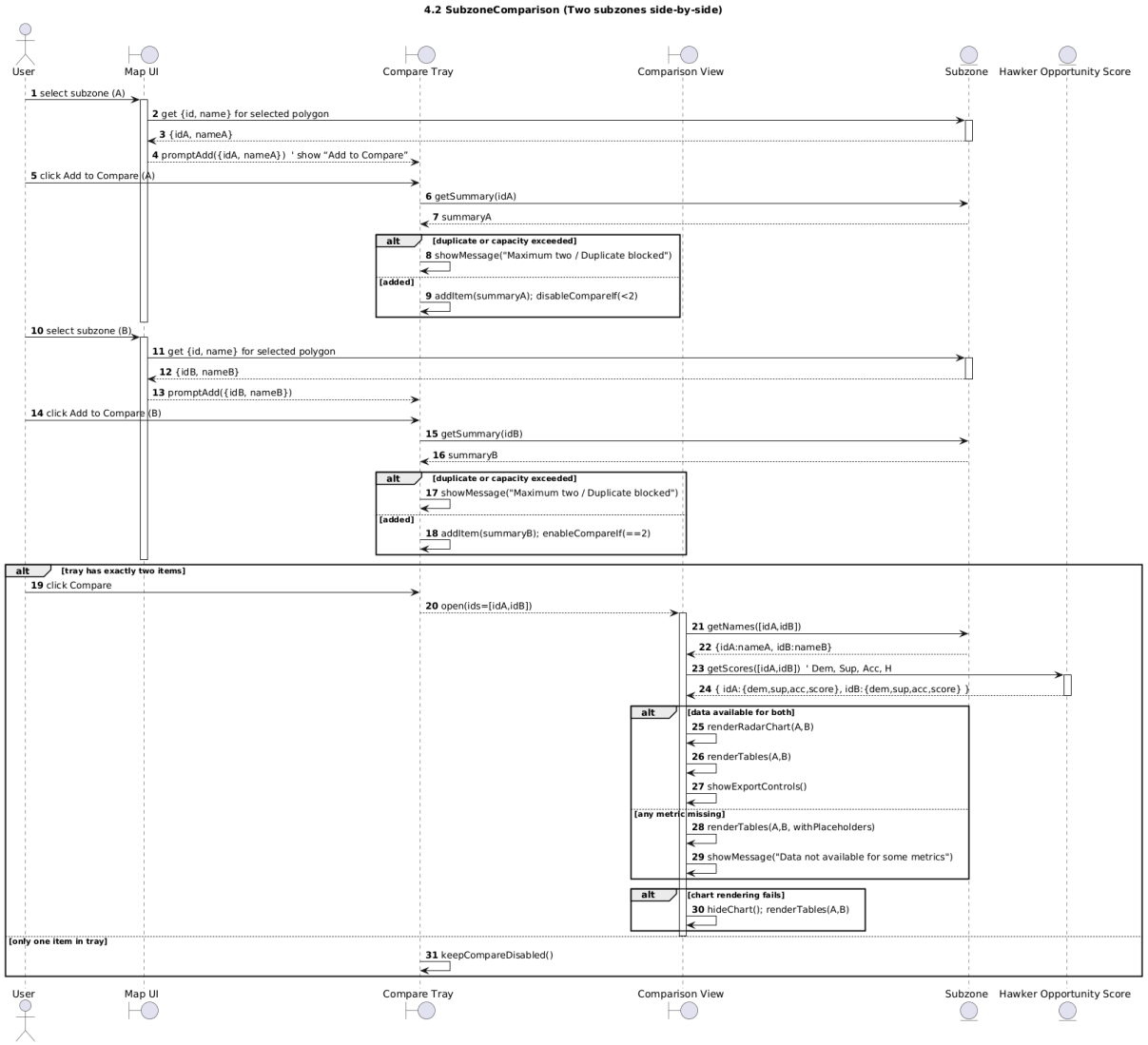


IV. For Functional Requirement #4.

4.1 ShowSubzoneDetails

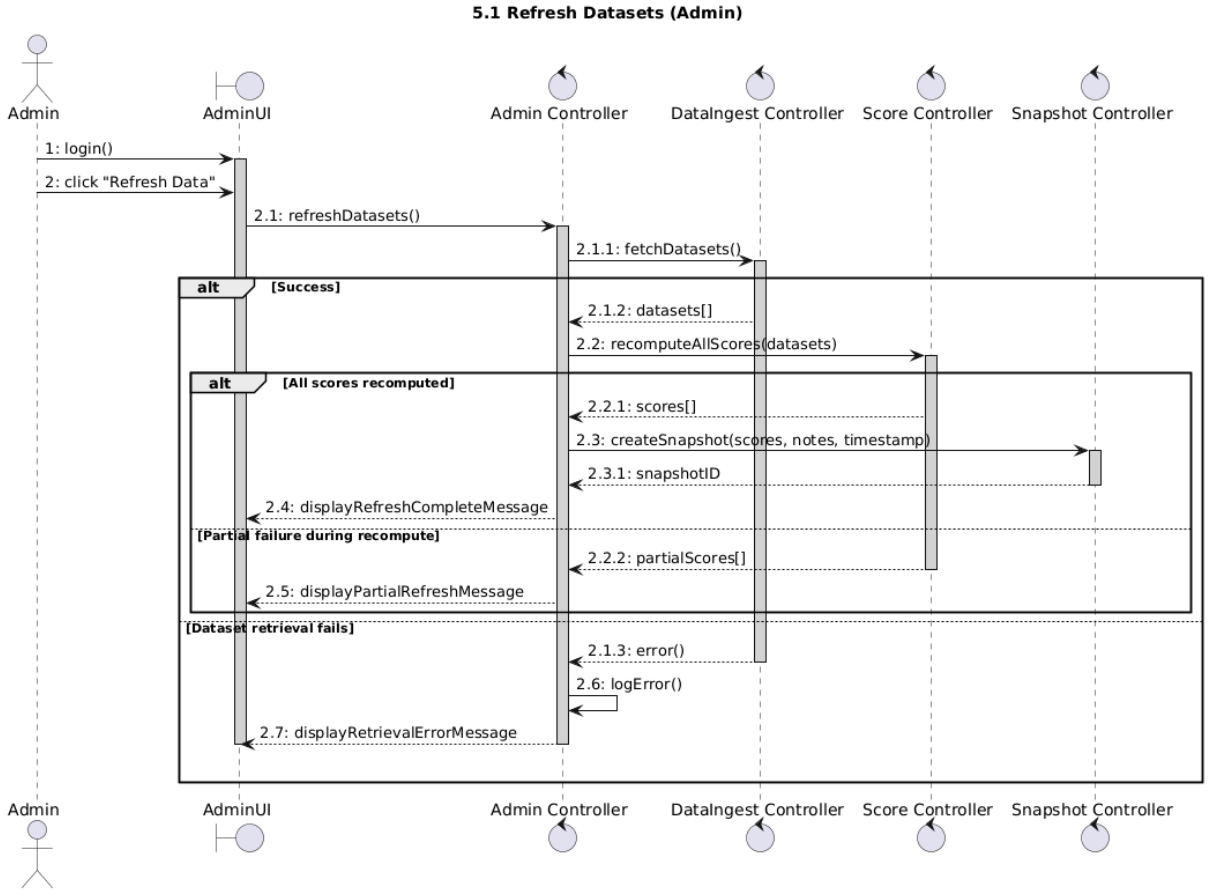


4.2 SubzoneComparison (Two subzones side-by-side)

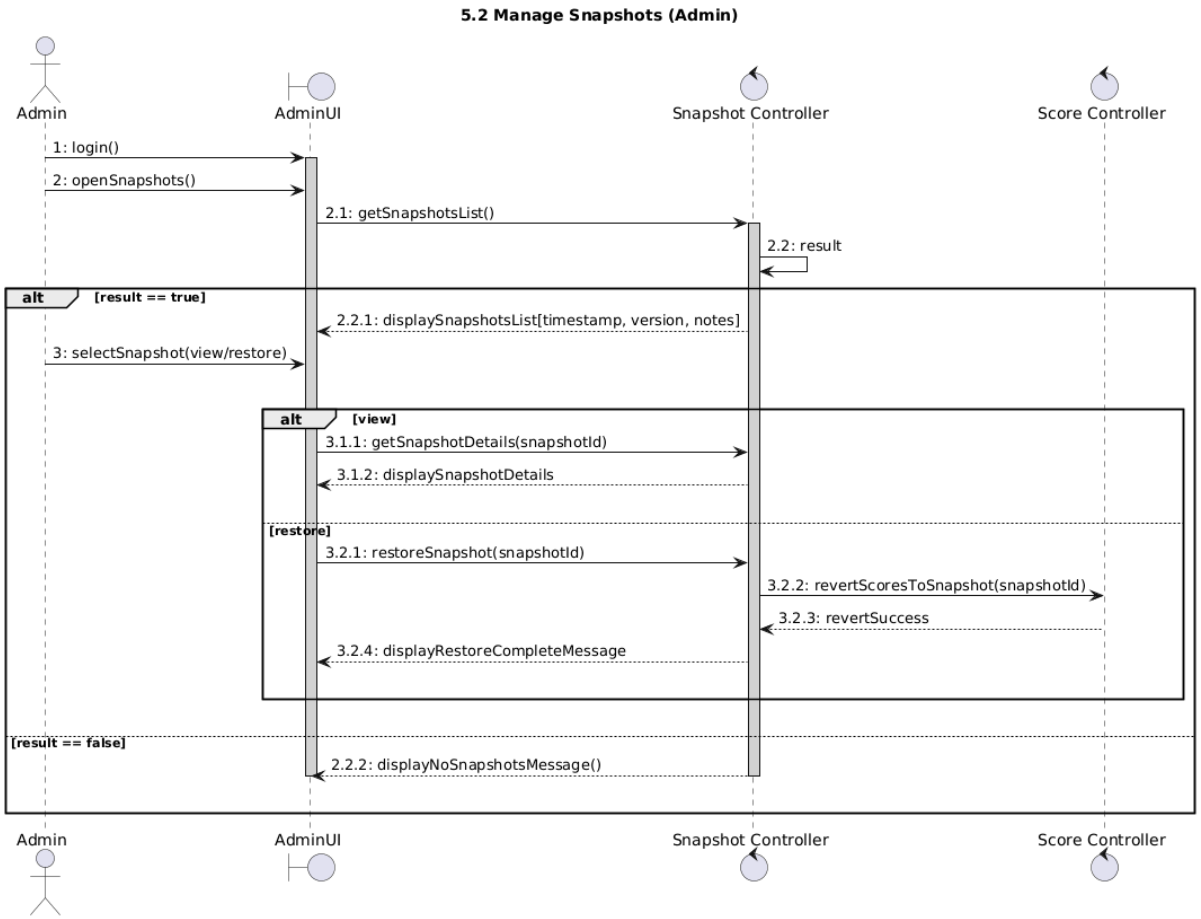


V. For Functional Requirement #5

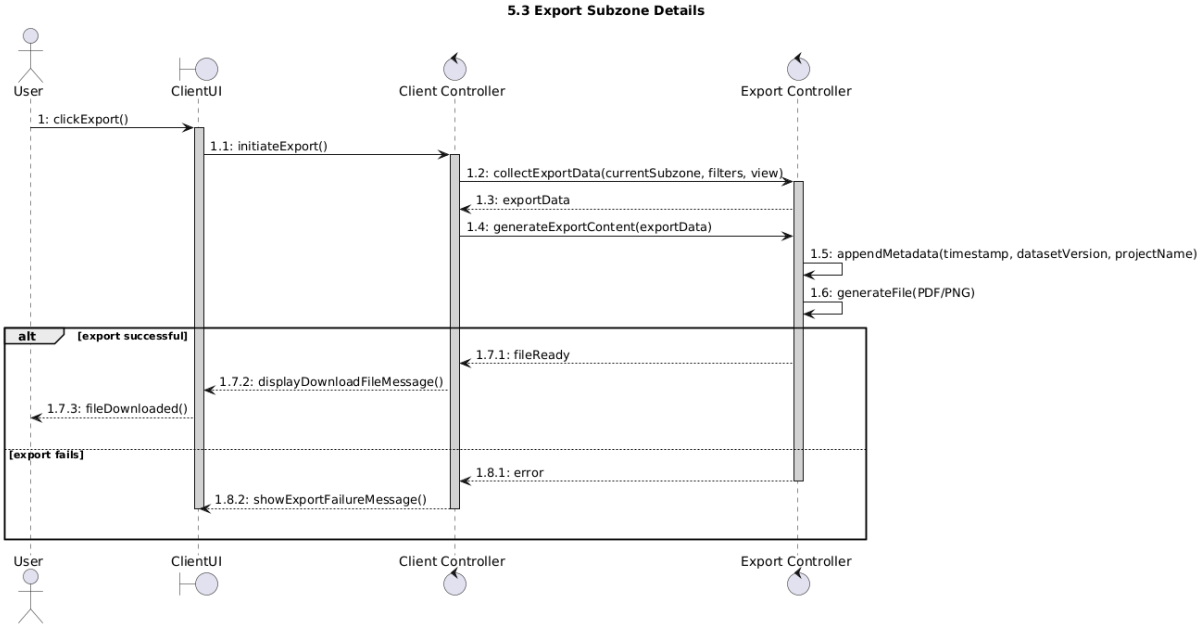
5.1 RefreshDatasets (Admin)



5.2 ManageSnapshots (Admin)

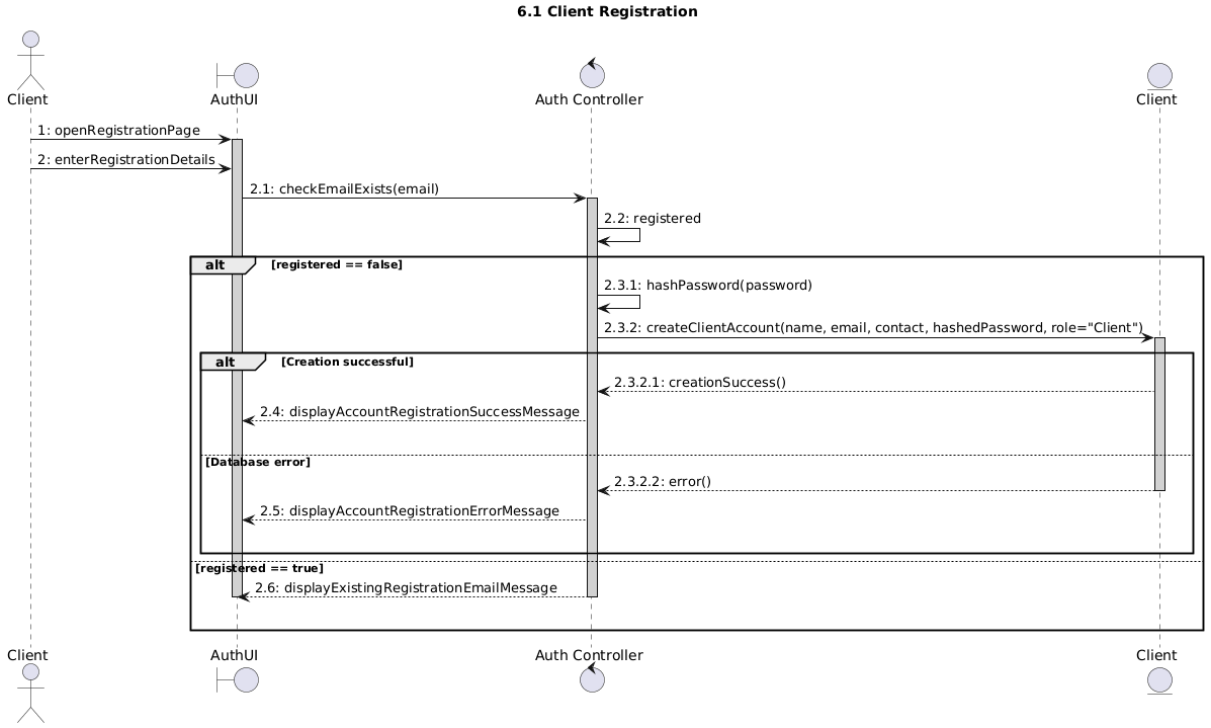


5.3 ExportSubzoneDetails

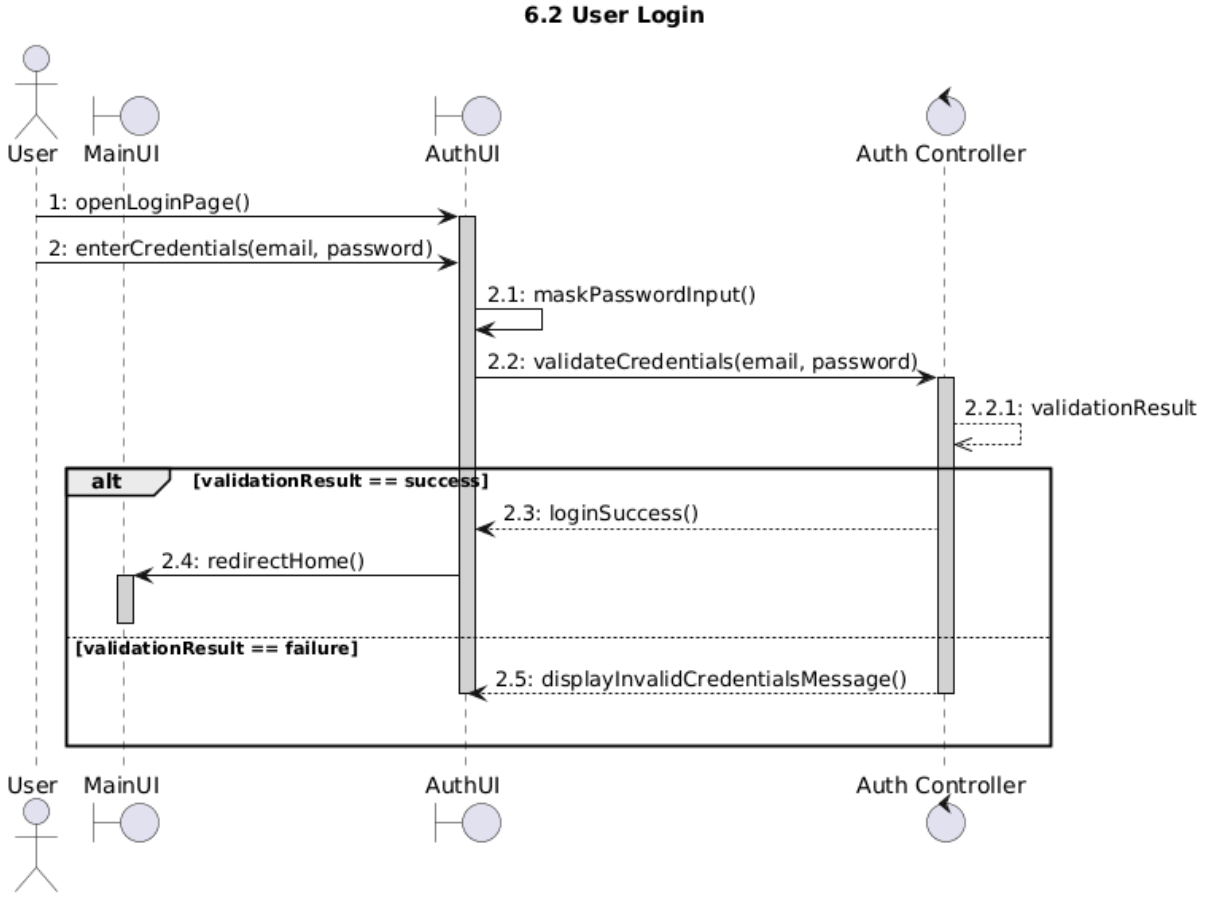


VI. For Functional Requirement #6

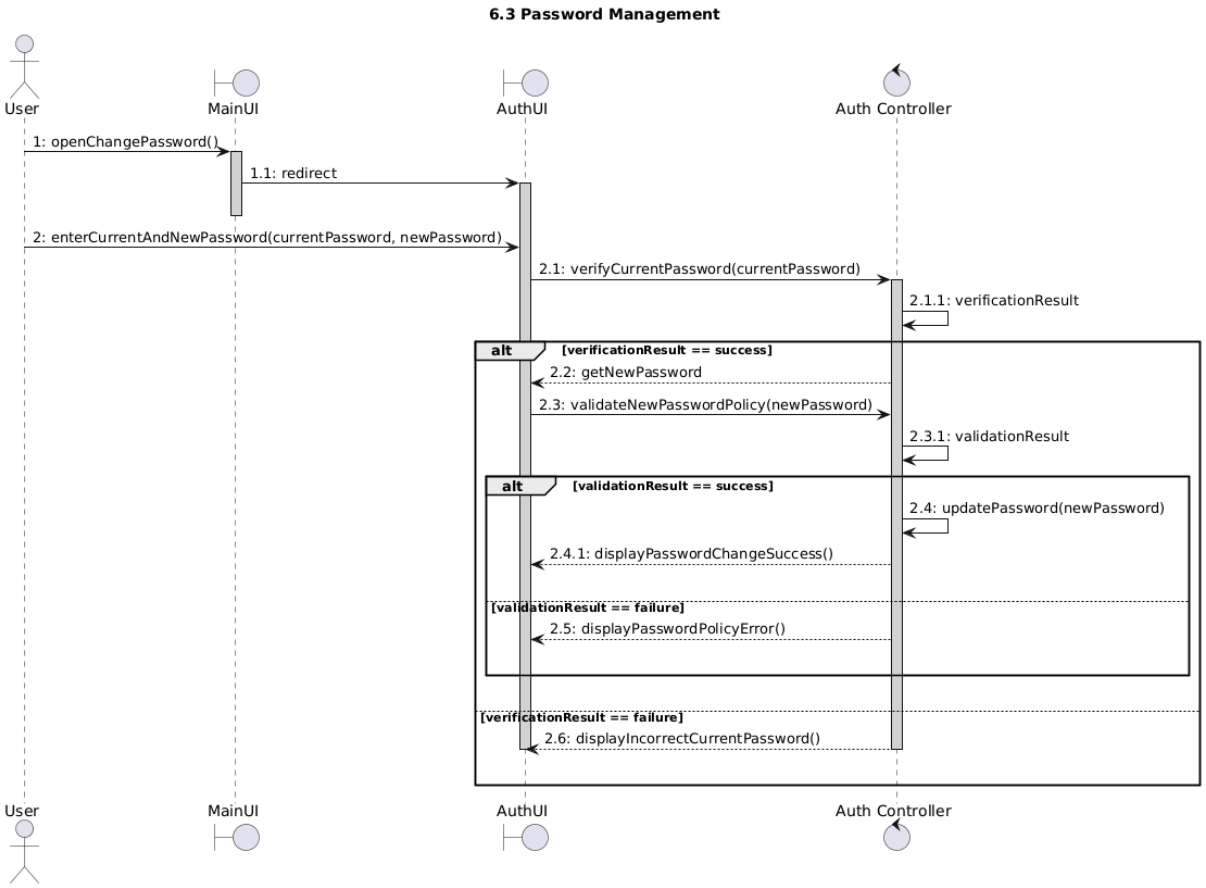
6.1 ClientRegistration



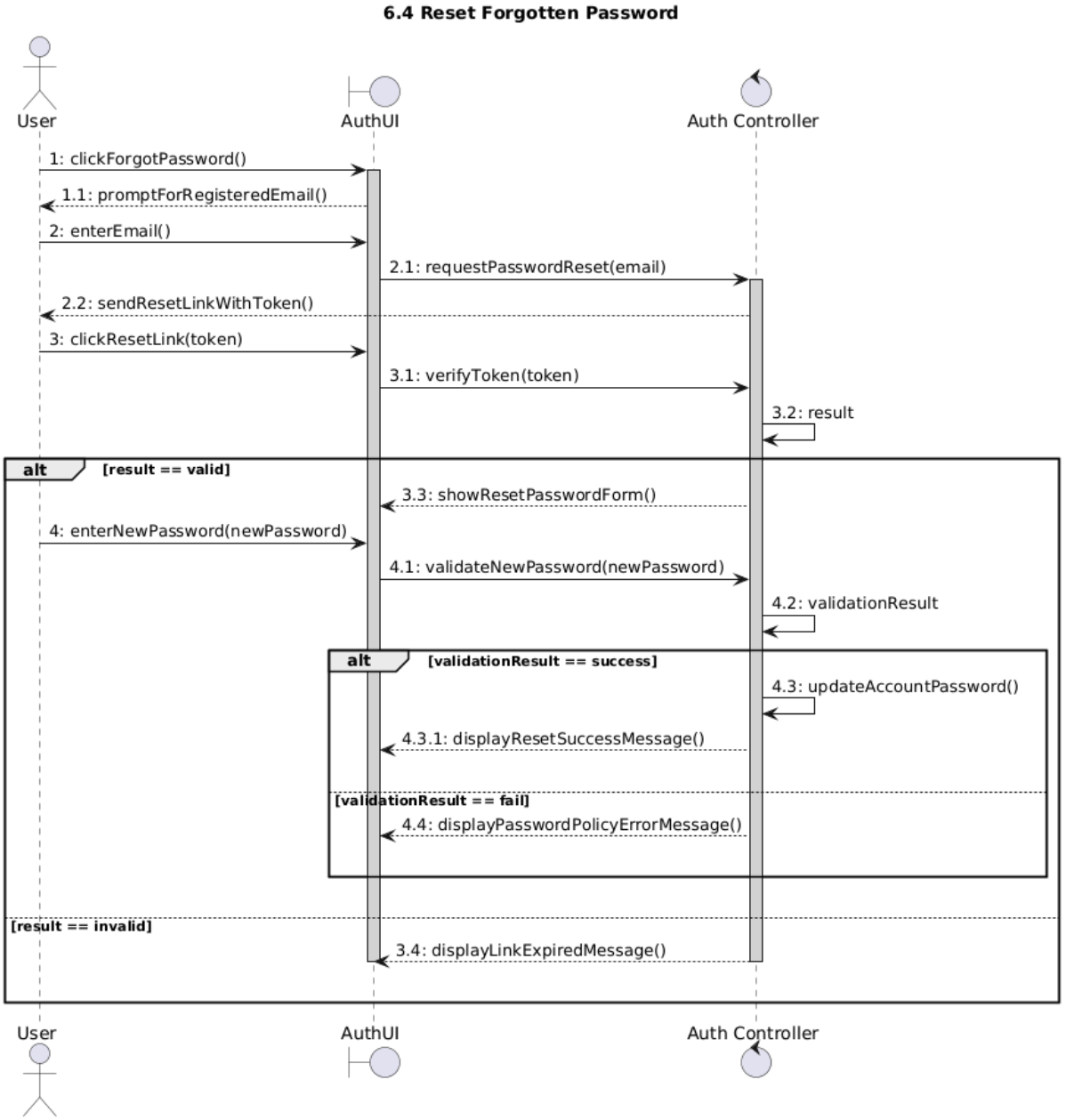
6.2 UserLogin

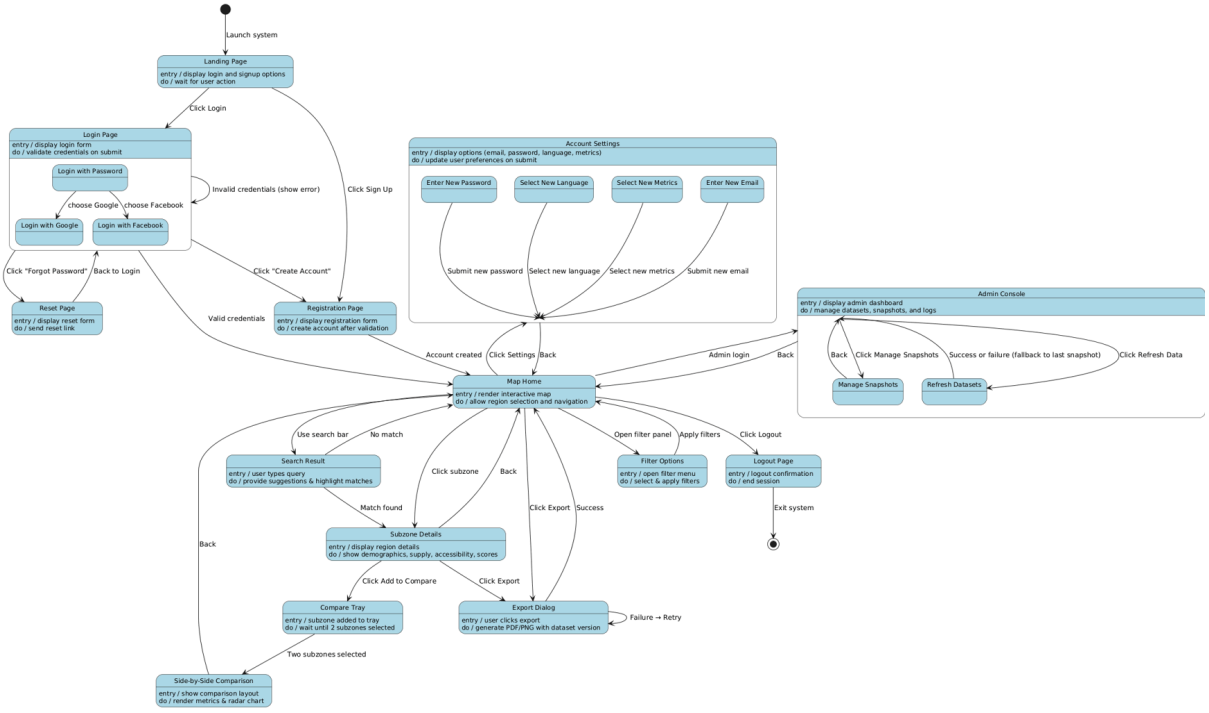


6.3 PasswordManagement



6.4 ResetForgottenPassword



5. Initial Dialog Map

If the image is unclear, please refer to the png file that is uploaded together with this document.