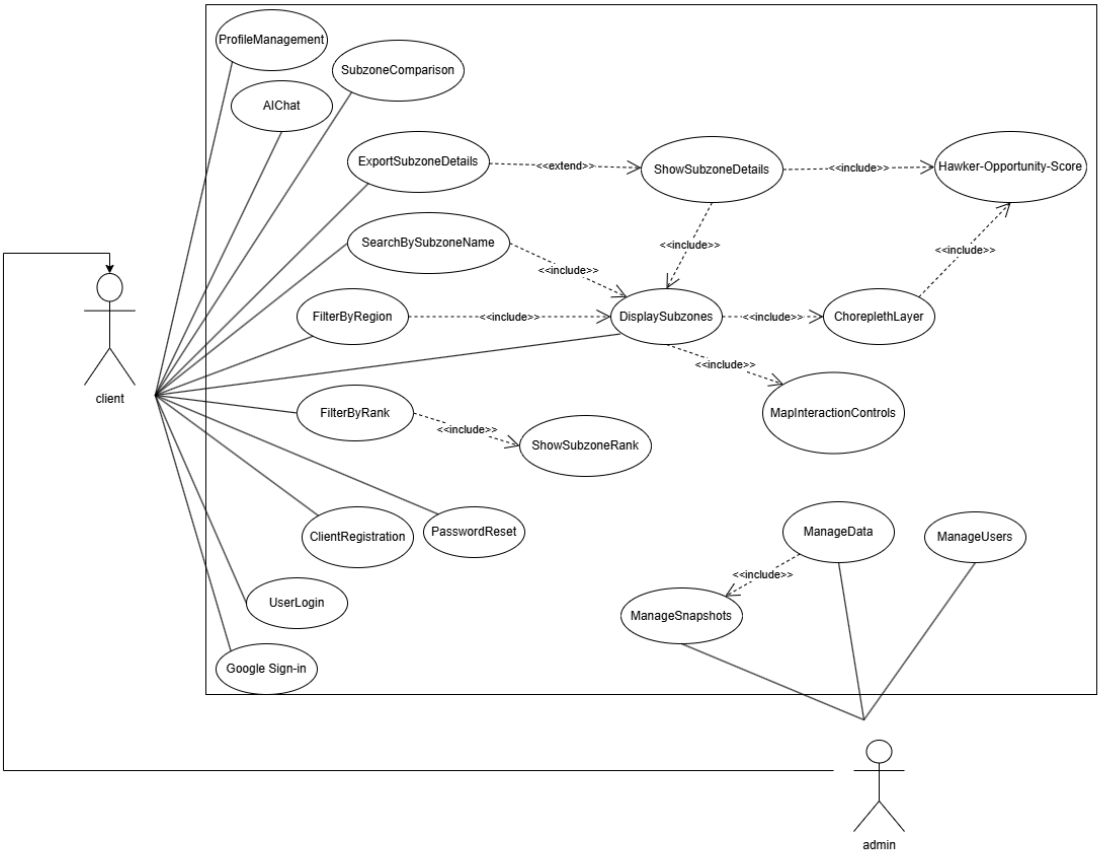
Use case Model and Use case Description

Table of contents

A. Use Case Diagram



B. Use Case Descriptions

1. For Functional Requirement #1 (Display Map)

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:	1.2 ChoreplethLayer and 1.3 MapInteractionControls
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:	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 (Calculating score and rank)

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. ShowSubzoneRank

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:	None
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, 20, All)

3. For Functional Requirement #3 (Filtering and Search)

3.1. FilterByRegion

Use Case ID:	3.1		
Use Case Name:	FilterByRegion		
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
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:	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. FilterByRank

Use Case ID:	3.2		
Use Case Name:	FilterByRank		
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 rank.
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:	2.2. ShowSubzoneRank
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:	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 (Details and Comparison)

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:	1.1 DisplaySubzones and 2.1 Hawker-OpportunityScore
Special Requirements:	Panel 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:	4.2		
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

4.3. ExportSubzoneDetails

Use Case ID:	3.3		
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 subzone details as an Excel file
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. 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:	4.1 ShowSubzoneDetails
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.

5. For Functional Requirement #5 (Admin functions)

5.1. ManageData

Use Case ID:	5.1		
Use Case Name:	ManageData		
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 uploads a FeatureCollection Geojson, system ingests it and applies this new data version.
Preconditions:	Admin is logged in and authorized.
Postconditions:	A new dataset is loade and a snapshot is stored.
Priority:	High
Frequency of Use:	Medium
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 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:	None
Special Requirements:	Only Admin accounts may execute this function.
Assumptions:	FeatureCollection contains expected properties used by scoring
Notes and Issues:	Admin must verify refresh success via system logs.

5.2. ManageSnapshots

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 views historical snapshots and restores any snapshot as the current dataset.
Preconditions:	Admin is logged in and authorized.
Postconditions:	Selected snapshot is marked current
Priority:	Hight
Frequency of Use:	Medium
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 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:	5.1 ManageData
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. ManageUsers

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

Actor:	Admin
Description:	Admin manages users: view list of users, create a new admin, and delete users.
Preconditions:	Admin is logged in and authorized.
Postconditions:	User list reflects created/deleted users
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	1. Admin logs into Admin Console 2. Admin opens User Management 3. System lists users and option to create new admin 4. Admin select a user and clicks Delete 5. System deletes or creates new admin and refresh list
Alternative Flows:	None
Exceptions:	Browser blocks file download; system prompts user to allow it.
Includes:	None
Special Requirements:	Do not expose sensitive fields
Assumptions:	Admin has permission to manage other admins.
Notes and Issues:	Consider soft-delete vs hard-delete based on compliance needs.

6. For Functional Requirement #6 (Authentication and Profile)

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 and creates user 3. System sends a verification email with a token to the user. 4. System shows success and prompts to verify email
Alternative Flows:	A1: If email is already registered → system shows “Email already registered.” A2: Weak password → show policy hints
Exceptions:	Network/Server error → show retry message
Includes:	None
Special Requirements:	Do not log plaintext passwords
Assumptions:	SMTP is configured
Notes and Issues:	Rate-limit registrations to prevent abuse

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:	User has a verified account
Postconditions:	User is authenticated and redirected to the main page.
Priority:	High
Frequency of Use:	Hight
Flow of Events:	1. User action: User navigates to the login page. System response: Displays login form. 2. User action: User enters email and password. System validates credentials and verification status 3. 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:	Access token short-lived
Assumptions:	The user provides correct credentials.
Notes and Issues:	Consider adding 2FA for Admin accounts in future.

6.3. GoogleSign-In

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

Actor:	Client
Description:	Authenticate using Google Identity Services;
Preconditions:	Client has Google account
Postconditions:	App tokens stored
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	1. Client clicks “Sign in with Google” in Login Page System action: Validates and issues app token 2. System action: System stores tokens 3. System response: redirect client to Main Page
Alternative Flows:	First-time Google user → account auto-provisioned
Exceptions:	Token verification fails → show error and allow fallback to password
Includes:	None
Special Requirements:	Verify nonce and aud/iss claims
Assumptions:	Correct Google Client ID configured
Notes and Issues:	Consider linking existing email accounts to Google

6.4. PasswordReset

Use Case ID:	6.4		
Use Case Name:	PasswordReset		
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:	Request a password reset email and set a new password using the reset token.
Preconditions:	User has an account with email
Postconditions:	Password is updated; old refresh tokens should be considered invalidated
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 the 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:	None
Special Requirements:	Passwords must be hashed securely before storage.
Assumptions:	SMTP operational
Notes and Issues:	Consider notifying users by email after change.

6.5. ProfileManagement

Use Case ID:	6.5		
Use Case Name:	ProfileManagement		
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:	View and update profile fields (display name, industry, phone, picture URL) and optionally change password.
Preconditions:	User has an account with email
Postconditions:	Profile updates persisted; password change (if any) enforced by policy
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	1. User action: opens Profile Page System action: loads current profile 2. User action: edits fields and submits changes System action: validates and saves changes 3. For password change: client submits current + new password System action: validates and rotates tokens if needed
Alternative Flows:	User cancels edits; no changes saved
Exceptions:	401/expired token → redirect to login page
Includes:	None
Special Requirements:	Do not expose sensitive fields; secure password change flow
Assumptions:	Stable network; backend reachable
Notes and Issues:	Consider optimistic UI for minor profile fields

7. For Functional Requirement #7 (AI Assistant)

7.1. AIChat

Use Case ID:	7.1		
Use Case Name:	AIChat		
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:	Interact with an AI assistant inside the map UI to ask questions about the platform, methodology, and subzone rankings. The system streams answers from a local LLM and grounds responses using real subzone data when relevant.
Preconditions:	User is logged in
Postconditions:	User sees an answer streamed into the chat window
Priority:	High
Frequency of Use:	Medium
Flow of Events:	1. User opens chatbox; system shows welcome. 2. User sends question (e.g., “Top 5 subzones”). 3. ChatController validates JWT, detects data query, injects subzone context if needed. 4. ChatService queries Ollama and streams a formatted response. 5. Controller streams to frontend; chatbox renders incrementally and finalizes on completion.
Alternative Flows:	● User action: closes the chatbox mid-stream. System action: stops rendering further chunks; no state corruption. ● User action: asks a general platform question (non-data). System action (ChatController): skips data injection; ChatService answers with general guidance.
Exceptions:	● 401 Unauthorized (expired/invalid JWT) → prompt user to log in again. ● LLM unavailable (Ollama not running or model missing) → show “AI service unavailable; please start Ollama.”
Includes:	None
Special Requirements:	None
Assumptions:	Ollama installed, running, and model pulled
Notes and Issues:	First model invocation may be slow while loading; subsequent calls are faster.