

**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

SC2006 – Software Engineering

Lab 1 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 Content

1. Documentation of functional and non-functional requirements.....	3
A. Functional Requirements.....	3
B. Non-Functional Requirements.....	6
2. Data dictionary.....	7
3. Initial Use Case Model, consisting of Use Case diagram and Use Case description..	9
A. Use Case Diagram.....	9
B. Use Case Descriptions.....	10
1. For Functional Requirement #1.....	10
1.1. DisplayRegions.....	10
1.2. ChoroplethLayer.....	11
1.3. MapInteractionControls.....	12
2. For Functional Requirement #2.....	13
2.1. Hawker-OpportunityScore.....	13
2.2. ShowRegionRankPercentile.....	14
3. For Functional Requirement #3.....	15
3.1. FilterByGeography.....	15
3.2. FilterByScoreQuantile.....	16
3.3. SearchByRegionName.....	17
4. For Functional Requirement #4.....	18
4.1. ShowDemographicInformation.....	18
4.2. ShowSupplyInformation.....	19
4.3. ShowAccessibilityInformation.....	20
5. For Functional Requirement #5.....	21
5.1. SelectRegionsToCompare.....	21
5.2. DisplaySide-by-SideComparison.....	22
6. For Functional Requirement #6.....	23
6.1. RefreshDatasets (Admin).....	23
6.2. ManageSnapshots (Admin).....	24
6.3. ExportMapRegion.....	25
7. For Functional Requirement #7.....	26
7.1. UserRegistration.....	26
7.2. AdminAccountCreation.....	27
7.3. User/AdminLogin.....	28
7.4. PasswordManagement.....	29
4. UI MockUps.....	30
4.1 Login and Registration.....	30
4.2 Home screen (List of all hawker centers).....	31
4.3 Regional zoom (List of hawker centers in a region).....	32
4.4 Filtering and calculating regional business score.....	32
4.5 Statistic of a region.....	33
4.6 Statistical and graphical comparison of multiple regions.....	34
4.6 Admin refreshing of data and generation of new scoremap.....	35

1. Documentation of functional and non-functional requirements

A. Functional Requirements

1. The system shall allow Users to browse a map of Singapore segmented by regions
 - 1.1. The system shall display regions as polygons.
 - 1.1.1. Each polygon shall be drawn based on URA official boundaries.
 - 1.1.2. Each polygon shall be clickable and highlighted upon selection.
 - 1.1.3. The polygon tooltip shall show the region name, ID, and basic score.
 - 1.2. The system shall provide a choropleth layer for the Hawker-Opportunity Score
 - 1.2.1. Scores shall be colour-coded using a colour-blind-safe gradient.
 - 1.2.2. The legend shall explain the score range and colour mapping.
 - 1.2.3. Colours shall update dynamically when filters are applied.
 - 1.3. The system shall provide map interaction controls.
 - 1.3.1. Users shall be able to zoom in/out with the mouse or touch controls.
 - 1.3.2. Users shall be able to pan across the map using drag gestures.
 - 1.3.3. Hovering over a polygon shall display a tooltip with score and rank percentile.
2. The system shall compute and show a Hawker-Opportunity Score per region
 - 2.1. The score shall integrate smoothed demand, competing-adjusted supply, and transport accessibility.
 - 2.1.1. Demand (Dem_i): The system shall convolve resident population counts with a spatial kernel (Gaussian or exponential) to generate a smoothed demand surface for each location i . Locations closer to more residents yield higher demand values.
 - 2.1.2. Supply (Sup_i): The system shall compute supply by convolving hawker centres with the same type of kernel, adjusting each centre's effective contribution by the demand it already faces. This prevents overscoring areas already saturated by demand.
 - 2.1.3. Accessibility (Acc_i): The system shall compute accessibility by convolving MRT stations and bus stops with their respective kernels. Weighted sums (β_{MRT} and β_{BUS}) and optional weights (e.g., number of lines or bus frequency) adjust their relative importance.
 - 2.1.4. Final Score (H_i): The system shall scale each component using robust z-scores and compute the weighted combination:
$$H_i = w_D \cdot Z(Dem_i) - w_S \cdot Z(Sup_i) + w_A \cdot Z(Acc_i), w_D + w_S + w_A = 1$$
 - 2.2. The system shall show the region rank percentile city-wide.
 - 2.2.1. Each score shall be ranked against all regions.
 - 2.2.2. The final Gap Score shall be shown for each region and ranked city-wide.
3. The system shall filter and search regions
 - 3.1. The system shall filter by Planning Area/Subzone.
 - 3.1.1. A dropdown shall allow filtering to a single planning area.
 - 3.1.2. When applied, only polygons in the chosen planning area shall remain visible.

- 3.2. The system shall filter by score quantile.
 - 3.2.1. Options shall include Top 10%, Top 25%, Top 50%, and All.
 - 3.2.2. The map shall dynamically update to show only regions that qualify.
- 3.3. The system shall allow search by region name.
 - 3.3.1. A search bar with autocomplete shall return matching planning areas or subzones.
 - 3.3.2. Selecting a search result shall zoom and highlight that region on the map.
4. The system shall display detailed region information when selected.
 - 4.1. Demographic information shall be shown.
 - 4.1.1. The total population of the region shall be displayed.
 - 4.1.2. Population breakdown by age groups shall be presented as both counts and percentages.
 - 4.1.3. A bar chart shall visualize the population distribution.
 - 4.2. Supply information shall be shown.
 - 4.2.1. The number of hawker centres within 400 m, 800 m, and 1 km shall be displayed.
 - 4.2.2. A list shall show each hawker centre name and distance from centroid.
 - 4.3. Accessibility information shall be shown.
 - 4.3.1. The nearest MRT station name and distance shall be displayed.
 - 4.3.2. The nearest bus stop code, road name, and distance shall be displayed.
5. The system shall provide a region comparison feature
 - 5.1. The system shall let Users select two regions to compare.
 - 5.1.1. Users shall add regions to a “comparison tray” by clicking “Add to Compare”.
 - 5.1.2. The tray shall support at least two selected regions.
 - 5.2. The system shall show side-by-side comparisons.
 - 5.2.1. Comparison shall include demand, supply, accessibility, and final score.
 - 5.2.2. A radar chart shall display differences visually.
 - 5.2.3. Results shall include percentile ranking for each component.
6. The system shall support data refresh and versioning (Admin)
 - 6.1. The system shall allow Admin to reload official datasets.
 - 6.1.1. Admin shall log in and trigger “Refresh Data”.
 - 6.1.2. The system shall fetch new datasets and recompute scores.
 - 6.2. The system shall store snapshots with version notes and timestamps to keep track of past versions of its calculations
 - 6.2.1. Each snapshot shall include dataset versions and formula weights.
 - 6.2.2. Snapshots shall be archived for at least 30 days.
 - 6.3. The system shall allow users to export results.
 - 6.3.1. Users shall export the current map + sidebar as PNG/PDF.
 - 6.3.2. Each export shall include metadata: dataset version and timestamp.
 - 6.3.3. Comparison view exports shall include radar charts and tables.
7. The system shall provide User and Admin authentication and account management.
 - 7.1. The system shall allow Users (general users such as entrepreneurs, planners) to create an account.

- 7.1.1. The system shall allow Users to input their name, email address, contact number, and password.
 - 7.1.2. The system shall create an account with the “User” role using the information entered.
 - 7.1.3. The system shall prevent duplicate registrations by verifying that the email is unique.
- 7.2. The system shall allow Admins to create an account with administrator privileges.
 - 7.2.1. The system shall allow Admins to input their name, email address, contact number, and password.
 - 7.2.2. The system shall create an account with the “Admin” role using the information entered.
 - 7.2.3. The system shall restrict creation of Admin accounts to existing Admins only.
 - 7.2.4. Admin accounts shall have access to dataset refresh, score recomputation, and configuration settings.
- 7.3. The system shall allow Users and Admins to sign in with their credentials.
 - 7.3.1. The login screen shall provide input fields for email and password.
 - 7.3.2. The system shall mask the password with dots by default, with an option to unmask.
 - 7.3.3. If the email and password do not match, the system shall display “Email and password do not match”.
 - 7.3.4. If the credentials are valid, the system shall log the User or Admin in and navigate to the home screen (map view).
- 7.4. The system shall allow Users and Admin to manage Password

B. Non-Functional Requirements

Usability

- The map shall load with a clear legend and color-blind-safe palette.
- All key actions (filter, select region, compare) shall be discoverable within 2 clicks.
- Content shall be legible on desktop and tablet; responsive layout for ≥ 1024 px width.

Performance

- Initial map and score tiles shall render in ≤ 3 s on broadband (p95).
- Region drill-down panel shall populate in ≤ 1 s after click (p95).
- Comparison view shall render in ≤ 1.5 s (p95).

Reliability & Availability

- Daily score snapshots retained for 30 days; recoverable within 5 minutes.
- On external-data errors, the system shall serve the last good snapshot without interruption.
- Uptime target $\geq 99\%$ during demo week.

Security

- Admin endpoints shall require API key and role-based access
- All data in transit over HTTPS; secrets stored as environment variables.

Maintainability

- Weights, radii, and thresholds shall be config-driven (no code change).
- Code shall follow layered architecture: UI, API, scoring, data adapters.
- Include unit tests for scoring and contract tests for dataset adapters.

Data Quality & Provenance

- Only official fixed lists and boundaries are used; each dataset recorded with source URL, version/date, and schema.
- The system shall validate geometry and deduplicate hawker entries by name+coords.

Transparency

- The methods page shall include equations, normalization ranges, weights, and limitations.
- Each score panel shall show component values and their contribution.

Scalability

- The system shall pre-compute per-region scores and serve as static JSON/tiles to handle concurrent users.
- Caching layer for region panels with TTL 10 min.

Compliance

- Use only publicly available government data and attribute sources per agency guidelines.

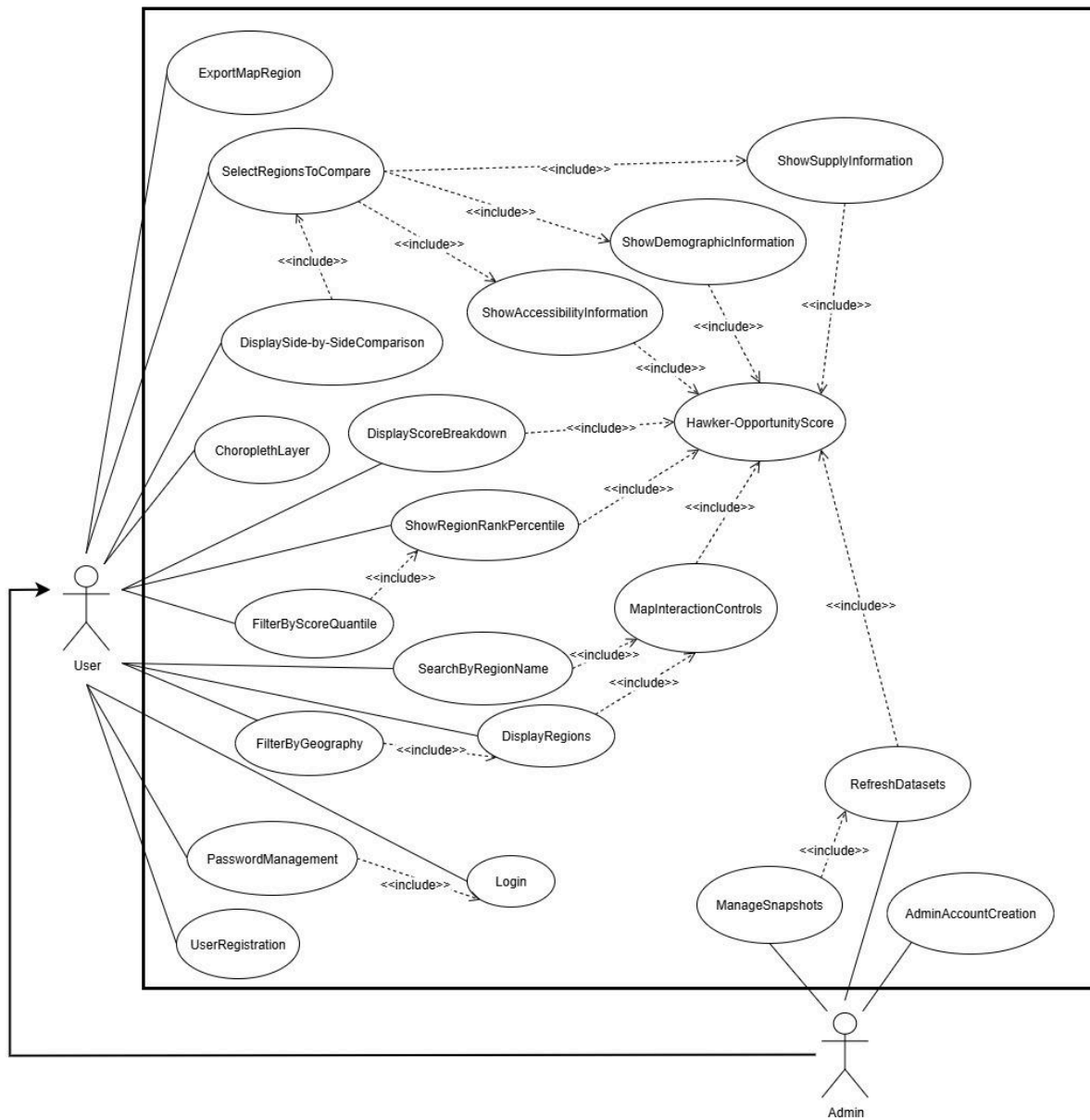
2. Data dictionary

Term	Definition
Region	A spatial analysis unit used to summarize and visualize hawker opportunity scores. Regions may be URA Planning Areas or Subzones. Each region has a polygon geometry and centroid used to aggregate kernel density results.
Subzone	A smaller geographic division within a Planning Area. Provides finer-grained representation of kernel density demand–supply gap scores.
Hawker Centre	A physical food centre listed by NEA. Each centre contributes to the supply density surface using a kernel function based on distance to surrounding regions.
Kernel Function ($K\lambda$)	A spatial weighting function (e.g., Gaussian or exponential) with bandwidth λ that smooths the influence of population, supply, or transport facilities across distance. Larger λ spreads influence farther; smaller λ localizes it.
Population Point	A demand source such as a subzone centroid weighted by resident counts. Populations are spread spatially through the kernel to produce demand density values.
Smoothed Demand (Dem_i)	The demand surface value at location i , calculated as the sum of nearby population counts weighted by distance through a kernel. Higher values indicate more nearby residents.
Competing-Adjusted Supply (Sup_i)	The supply surface value at location i , calculated by convolving hawker centres with a kernel and reducing each centre's contribution according to the demand it already faces. Reflects effective, not raw, supply.
Accessibility (Acc_i)	The transport accessibility value at location i , computed as a weighted kernel density of MRT stations and bus stops. Parameters β_{MRT} and β_{BUS} allow emphasis on rail vs bus access; weights allow scaling by service capacity.
Hawker-Opportunity Score (H_i)	The final suitability score for region i , combining demand (+), supply (–), and accessibility (+) after z-scaling. High H_i indicates areas with high demand, low competing supply, and good public transport access.
Kernel Bandwidth	The spatial smoothing parameters for demand, supply, MRT, and bus accessibility kernels. Chosen based on plausible walking catchments.
Snapshot	A record of one complete scoring run, including all

	component values, final scores, and dataset versions. Used to ensure reproducibility and historical comparison.
Dataset Version	A record of one complete scoring run, including all component values, final scores, and dataset versions. Used to ensure reproducibility and historical comparison.
UI Filter	User-selected constraints.
Compare Pair	Two regions selected by the user for side-by-side comparison of demand density, supply density, and final Gap Score.

3. Initial Use Case Model, consisting of Use Case diagram and Use Case description

A. Use Case Diagram



B. Use Case Descriptions

1. For Functional Requirement #1

1.1. DisplayRegions

Use Case ID:	1.1		
Use Case Name:	DisplayRegions		
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 the Singapore map segmented by Planning Areas and Subzones. Each region appears as a polygon boundary on the map.
Preconditions:	The system has loaded URA planning area and subzone boundary datasets.
Postconditions:	Regions 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:	<ol style="list-style-type: none">1. The user opens the application home screen.2. System loads map base layer.3. The system overlays polygons of Planning Areas and Subzones.4. The user sees the polygons drawn on the map.
Alternative Flows:	If dataset is unavailable, system shows error "Unable to load boundaries" 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.
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 region shaded according to its Hawker-Opportunity Score (normalized value).
Preconditions:	Score computation has been completed and stored in the system.
Postconditions:	Regions 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:	<ol style="list-style-type: none">1. The user accesses the home map.2. The system retrieves scores for each region.3. The system normalizes scores and maps them to a color gradient.4. The system applies shading to each polygon. Legend is displayed to explain colour ranges.
Alternative Flows:	If dataset is unavailable, system shows error "Unable to load boundaries" 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.
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:	User views the Singapore map segmented by Planning Areas and Subzones. Each region appears as a polygon boundary on the map.
Preconditions:	The system has loaded URA planning area and subzone boundary datasets.
Postconditions:	Regions 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:	<ol style="list-style-type: none">1. User zooms in/out to adjust map scale.2. User pans map to move to another area.3. User hovers over a polygon.4. The system displays tooltips with region name, score, and rank percentile.
Alternative Flows:	If map API fails, the user is restricted to default zoom level.
Exceptions:	Tooltip fails to appear if the dataset is not linked.
Includes:	Use case 1.1 and 1.2
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 region 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 region 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:	<ol style="list-style-type: none"> 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: $H_i = w_D \cdot Z(Dem_i) - w_S \cdot Z(Sup_i) + w_A \cdot Z(Acc_i)$ 7. Scores are stored in the snapshot with metadata.
Alternative Flows:	<p>A1: If capacity (C_{\square}) for a hawker centre is missing, system assumes $C_{\square} = 1$</p> <p>A2: If transport weights are not provided, system assumes $\beta_{MRT} = \beta_{BUS} = 1$</p>
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. ShowRegionRankPercentile

Use Case ID:	2.3		
Use Case Name:	ShowRegionRankPercentile		
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 region ranks relative to all other regions in terms of Hawker-Opportunity Score.
Preconditions:	Scores for all regions have been computed.
Postconditions:	Percentile rank is displayed in the tooltip and region details panel.
Priority:	Medium
Frequency of Use:	Occasionally, when comparing regions or selecting one.
Flow of Events:	<ol style="list-style-type: none">1. The user hovers or clicks on a region.2. The system retrieves the percentile rank of the region.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 Planning Areas or Subzones.
Preconditions:	The system has loaded all region polygons and computed scores.
Postconditions:	Only polygons matching the selected area(s) remain visible on the map.
Priority:	Medium
Frequency of Use:	Occasionally, when focusing on a specific area.
Flow of Events:	<ol style="list-style-type: none">1. The user opens the filter panel.2. The user selects a Planning Area from the dropdown list. System highlights and displays only subzones within that Planning Area.3. Optionally, the user selects a Subzone for more fine-grained filtering.4. System updates map view accordingly.
Alternative Flows:	If no Planning Area is selected, the system shows all regions.
Exceptions:	Dropdown fails to load due to missing dataset.
Includes:	Use case 1.1. DisplayRegions
Special Requirements:	Filter must be applied instantly without requiring page reload.
Assumptions:	Planning Areas and Subzones list matches URA official dataset
Notes and Issues:	Must handle user deselection gracefully (restore all regions).

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 regions that fall within a selected score percentile range (e.g., Top 10%, Top 25%).
Preconditions:	Scores and percentile ranks have been computed for all regions.
Postconditions:	The map updates to display only regions within the chosen percentile threshold.
Priority:	Medium
Frequency of Use:	Often, when identifying high-potential areas.
Flow of Events:	<ol style="list-style-type: none">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 regions that meet the criterion.4. System updates map to show only those regions.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 2.3. ShowRegionRankPercentile
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. SearchByRegionName

Use Case ID:	3.3		
Use Case Name:	SearchByRegionName		
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 searches for a Planning Area or Subzone by entering its name in a search bar.
Preconditions:	The list of region names is available in the system.
Postconditions:	The map zooms to and highlights the selected region.
Priority:	High
Frequency of Use:	Frequently, when users know exactly which region they want to view.
Flow of Events:	<ol style="list-style-type: none">1. The user types a region name into the search bar.2. The system provides autocomplete suggestions as user types.3. The user selects a suggested region from the dropdown.4. The system zooms into and highlights the selected region polygon.5. Tooltip appears with region name, score, and percentile rank.
Alternative Flows:	If no matches are found, the system displays "No region found".
Exceptions:	Autocomplete fails due to missing dataset or system error.
Includes:	Use case 1.3. MapInteractionControls
Special Requirements:	Autocomplete must handle both full names (e.g., "Tampines") and partial input (e.g., "Tam").
Assumptions:	Region names are stored exactly as defined in URA datasets.
Notes and Issues:	Should handle both Planning Area and Subzone search in one field.

4. For Functional Requirement #4

4.1.ShowDemographicInformation

Use Case ID:	4.1		
Use Case Name:	ShowDemographicInformation		
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:	When a region is selected, the system displays its total population and age group breakdown (0–14, 15–64, 65+).
Preconditions:	Demographic dataset is available and linked to region IDs.
Postconditions:	Demographic statistics appear in the region details panel.
Priority:	High
Frequency of Use:	Often, whenever users want to evaluate demand.
Flow of Events:	<ol style="list-style-type: none">1. The user selects a region polygon on the map.2. The system retrieves population data for that region.3. The system displays total residents and population in age groups.4. The system displays values as both raw counts and percentages.5. The system generates a bar chart visualizing the distribution.
Alternative Flows:	If demographic data for a region is missing, system shows “Demographics unavailable”.
Exceptions:	Chart rendering fails due to client browser limitations.
Includes:	Use case 2.1 - Hawker-Opportunity Score.
Special Requirements:	Chart must update instantly when a new region is selected.
Assumptions:	Census data is the latest available from SingStat.
Notes and Issues:	Large population subzones must be scaled properly to avoid misleading charts.

4.2.ShowSupplyInformation

Use Case ID:	4.2		
Use Case Name:	ShowSupplyInformation		
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:	When a region is selected, the system displays the number of hawker centres within defined distances (400 m, 800 m, 1 km).
Preconditions:	NEA hawker centres dataset is available and geocoded.
Postconditions:	Hawker counts and a list of nearby centres appear in the region details panel.
Priority:	High
Frequency of Use:	Frequently, when assessing competition.
Flow of Events:	<ol style="list-style-type: none">1. The user selects a region polygon.2. The system computes the number of hawker centres within 400 m, 800 m, and 1 km of region centroid.3. The system displays counts for each radius.4. The system lists each hawker centre with its name and distance.
Alternative Flows:	If the hawker dataset is outdated, the system shows “Data not refreshed – counts may be inaccurate”.
Exceptions:	A hawker centre has missing coordinates; the system skips it and logs error.
Includes:	Use Case 2.1 Hawker-Opportunity Score.
Special Requirements:	Distances must be computed accurately using geospatial functions.
Assumptions:	Hawker centre list from NEA is authoritative.
Notes and Issues:	If multiple centres overlap, system should de-duplicate by ID.

4.3.ShowAccessibilityInformation

Use Case ID:	4.3		
Use Case Name:	ShowAccessibilityInformation		
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:	When a region is selected, the system shows the nearest MRT station and bus stop, with their respective distances.
Preconditions:	MRT and bus stop datasets are available.
Postconditions:	Accessibility metrics appear in the region details panel.
Priority:	Medium
Frequency of Use:	Occasionally, when considering location convenience.
Flow of Events:	<ol style="list-style-type: none">1. The user selects a region polygon.2. The system computes distance from region centroid to nearest MRT.3. The system displays the MRT station name, line, and distance in meters.4. The system computes the distance to the nearest bus stop.5. The system displays bus stop code, road name, and distance.
Alternative Flows:	A1: If MRT data is unavailable, the system shows “MRT information not available”. A2: If bus stop data is unavailable, the system shows “Bus stop information not available”.
Exceptions:	Geospatial query fails due to corrupted coordinates.
Includes:	Use Case 2.1 Hawker-Opportunity Score.
Special Requirements:	Distances must be calculated in meters using GIS libraries.
Assumptions:	Distance is computed as a straight-line, not walking path.
Notes and Issues:	Walking path integration could be a future enhancement.

5. For Functional Requirement #5

5.1.SelectRegionsToCompare

Use Case ID:	5.1		
Use Case Name:	SelectRegionsToCompare		
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 selects two regions to add into the comparison tray for side-by-side evaluation.
Preconditions:	Both regions must have scores and details computed.
Postconditions:	Two regions are added into a comparison tray, ready for comparison.
Priority:	High
Frequency of Use:	Occasionally, when evaluating multiple candidate regions.
Flow of Events:	<ol style="list-style-type: none">1. The user views the map and clicks a region.2. The system shows a "+ Add to Compare" button in the region details panel.3. The user clicks the button.4. The system adds the region into the comparison tray.5. The user repeats for a second region.6. Tray confirms two selected regions.
Alternative Flows:	A1: If the user tries to add more than two regions, the system displays "Maximum two regions allowed". A2: If the same region is selected twice, the system prevents duplicate entry.
Exceptions:	If one region has incomplete data, the system prevents it from being added and shows an error message.
Includes:	Use Case 4.1 – 4.3 (Region Details).
Special Requirements:	Trays must remain visible and persistent during navigation.
Assumptions:	Users understand the limit of two regions.
Notes and Issues:	Consider allowing more than two regions in future versions.

5.2. DisplaySide-by-SideComparison

Use Case ID:	5.2		
Use Case Name:	DisplaySide-by-SideComparison		
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 demand, supply, accessibility, and scores for two selected regions displayed side-by-side.
Preconditions:	Two regions have been selected for comparison.
Postconditions:	A comparison screen with both regions' metrics is displayed.
Priority:	High
Frequency of Use:	Occasionally, during evaluation of candidate sites.
Flow of Events:	<ol style="list-style-type: none">1. User clicks "Compare" in the tray with two regions selected.2. The system opens the comparison view.3. The system retrieves demographic data, hawker supply counts, accessibility, and scores for both regions.4. The system displays both regions in two parallel columns.5. The system renders a radar chart showing differences in demand, supply, access, and final score.
Alternative Flows:	If one dataset is missing, the system displays "Data not available" for that metric.
Exceptions:	If rendering fails, the system falls back to a tabular-only comparison.
Includes:	Use Case 5.1 – SelectRegionsToCompare.
Special Requirements:	Charts must be responsive and exportable.
Assumptions:	Percentile ranks are recomputed at the same time as scores.
Notes and Issues:	UI must clearly indicate which region is which (labels, colors).

6. For Functional Requirement #6

6.1. RefreshDatasets (Admin)

Use Case ID:	6.1		
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:	<ol style="list-style-type: none">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 regions.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.

6.2. ManageSnapshots (Admin)

Use Case ID:	6.2		
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:	<ol style="list-style-type: none">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.

6.3. ExportMapRegion

Use Case ID:	6.3		
Use Case Name:	ExportMapRegion		
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 exports the current map view and region details as a PDF or PNG file with dataset version and timestamp included.
Preconditions:	A region 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:	<ol style="list-style-type: none">1. The user clicks the "Export" button on the map or region details page.2. The system collects current map state, filters applied, and visible details.3. The system generates export content with map, 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:	Use Case 4.1–4.3
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.

7. For Functional Requirement #7

7.1. UserRegistration

Use Case ID:	7.1		
Use Case Name:	UserRegistration		
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 registers for an account with the system to access standard features such as browsing the map, viewing region details, and exporting results.
Preconditions:	The user does not already have an account registered with the same email.
Postconditions:	A new User account is created with the "User" role.
Priority:	High
Frequency of Use:	Once per user (account creation).
Flow of Events:	<ol style="list-style-type: none">1. The user navigates to the registration page.2. User enters name, email address, contact number, and password.3. The system verifies that the email is not already registered.4. The system hashes the password securely.5. System creates the User account with role = "User".6. The system confirms registration success.
Alternative Flows:	If an email already exists, the system displays "Email already registered."
Exceptions:	Database error prevents account creation.
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.

7.2. AdminAccountCreation

Use Case ID:	7.2		
Use Case Name:	AdminAccountCreation		
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 registers another Admin account with elevated privileges to manage datasets and configurations.
Preconditions:	The Admin creating the account is logged in and authorized.
Postconditions:	A new Admin account is created with role = "Admin".
Priority:	High
Frequency of Use:	Occasionally, when a new Admin is required.
Flow of Events:	<ol style="list-style-type: none">1. Admin logs into the system.2. Admin navigates to "Create Admin" in the Admin Console.3. Admin enters name, email, contact number, and password.4. System validates input.5. The system hashes the password and creates the Admin account.6. The system confirms registration success.
Alternative Flows:	If a non-Admin user attempts to access this function, the system shows "Access Denied."
Exceptions:	Database error prevents account creation.
Includes:	None.
Special Requirements:	Admin accounts must be created by existing Admins only.
Assumptions:	At least one Admin exists to bootstrap the system.
Notes and Issues:	Admin actions must be logged for auditing.

7.3. User/AdminLogin

Use Case ID:	7.3		
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 or Admin
Description:	User or Admin logs into the system using email and password.
Preconditions:	A valid account exists.
Postconditions:	User or Admin is authenticated and redirected to the home map view.
Priority:	High
Frequency of Use:	Frequently, each time a session begins.
Flow of Events:	<ol style="list-style-type: none">1. The user navigates to the login page.2. The user enters an email and password.3. The system masks password input with dots (option to unmask).4. The system validates credentials against stored account data.5. If valid, the system logs the user in and redirects to the home screen.
Alternative Flows:	If credentials are invalid, the system shows "Email and password do not match."
Exceptions:	Authentication server is 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.

7.4. PasswordManagement

Use Case ID:	7.4		
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 or Admin
Description:	User or Admin resets or changes their password when forgotten or expired.
Preconditions:	The account exists with a valid email.
Postconditions:	Password is updated, and the user regains access.
Priority:	Medium
Frequency of Use:	Occasionally, when password reset is needed.
Flow of Events:	<ol style="list-style-type: none">1. The user clicks "Forgot Password" on the login page or "Changing Password" at the setting2. The system prompts users to enter registered email if forgotten.3. The system sends a reset link with a one-time token (valid 15 minutes).4. The user clicks a link and enters a new password.5. System validates new password against policy and updates account.
Alternative Flows:	If the reset token expires, the system shows "Link expired – request new reset."
Exceptions:	Email server fails to send reset link.
Includes:	Use Case 7.3. Login
Special Requirements:	Passwords must be hashed securely before storage.
Assumptions:	The user has access to the registered email inbox.
Notes and Issues:	Enforce strong password policies (≥ 12 characters, mixed types).

4. UI MockUps

4.1 Login and Registration

The image displays two UI mockups for a web application, presented as browser windows. Both mockups feature a map of Singapore on the left side, with three red location pins. The browser window includes a top bar with a 'Personal' dropdown, navigation arrows, a search bar, and a 'google.com' address. The main content area is divided into two sections: a left sidebar with a map and a right main area with a form.

Mockup 1: Create new Account

Create new Account

Already Registered? [Login](#)

NAME

EMAIL

PASSWORD

DATE OF BIRTH

[sign up](#)

Mockup 2: Login

Login

Sign in to continue

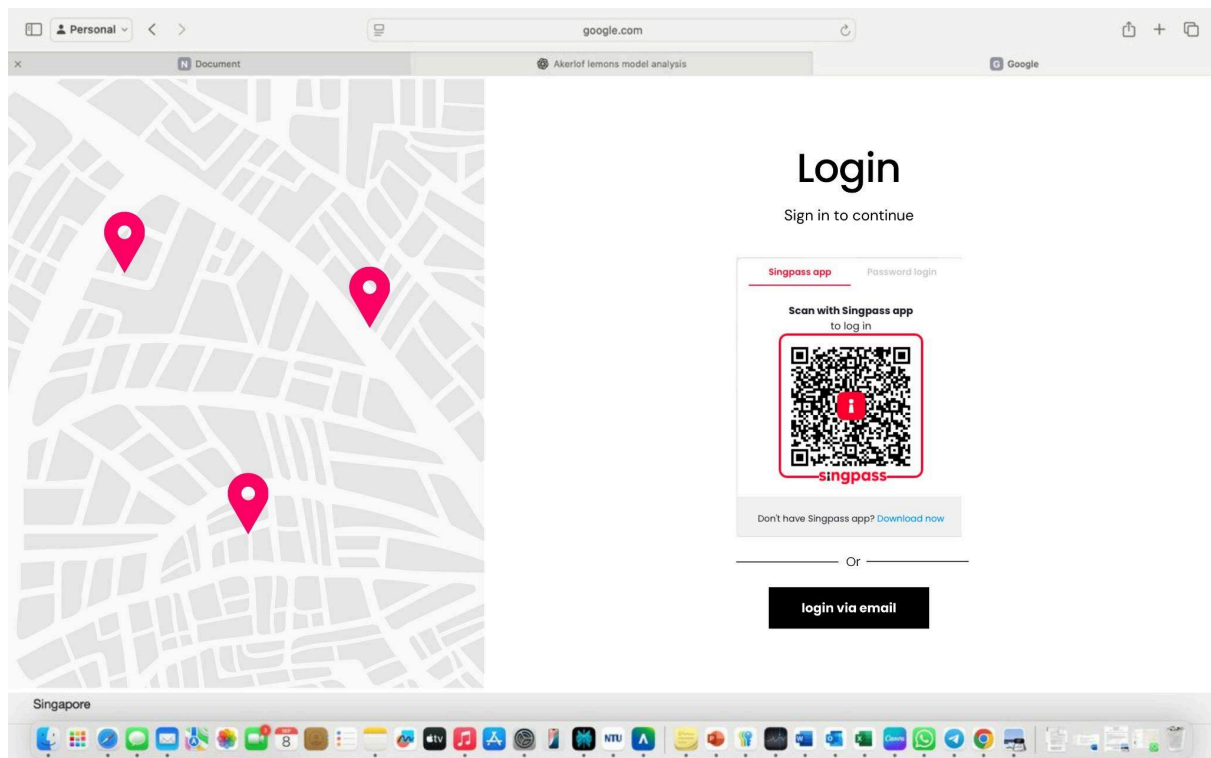
EMAIL

PASSWORD

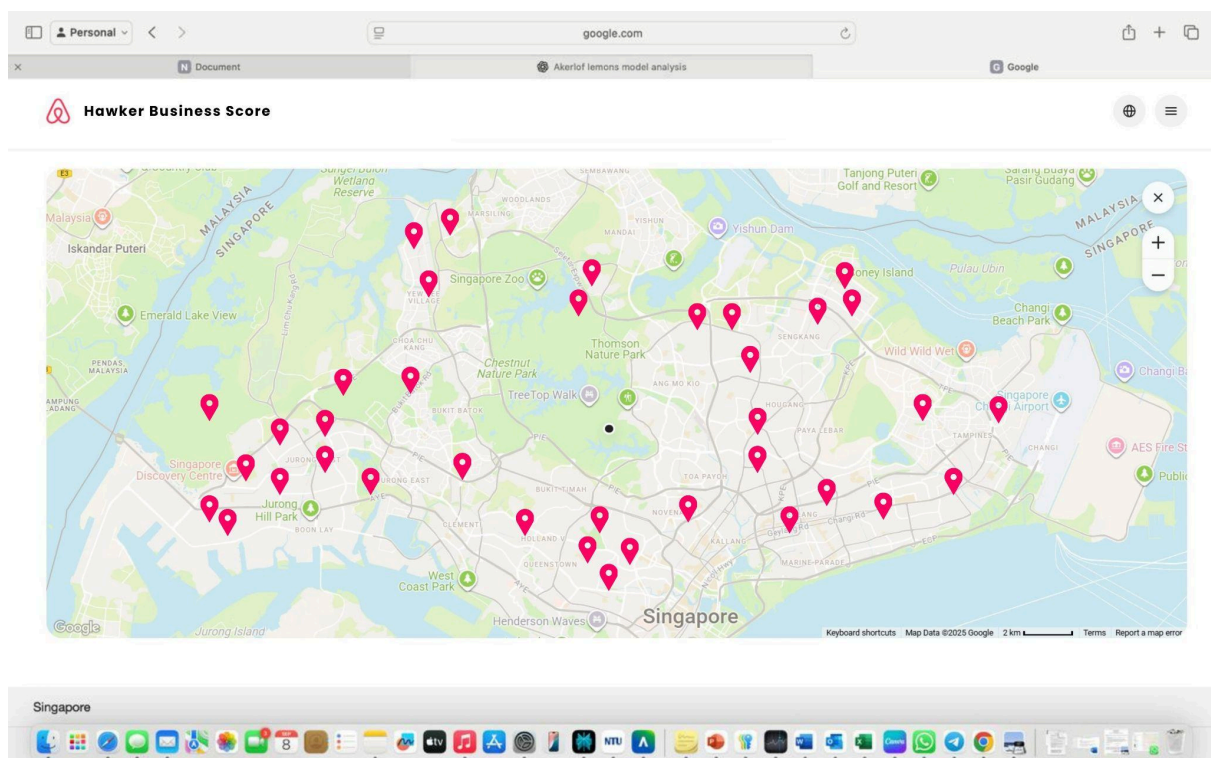
[login](#)

Or

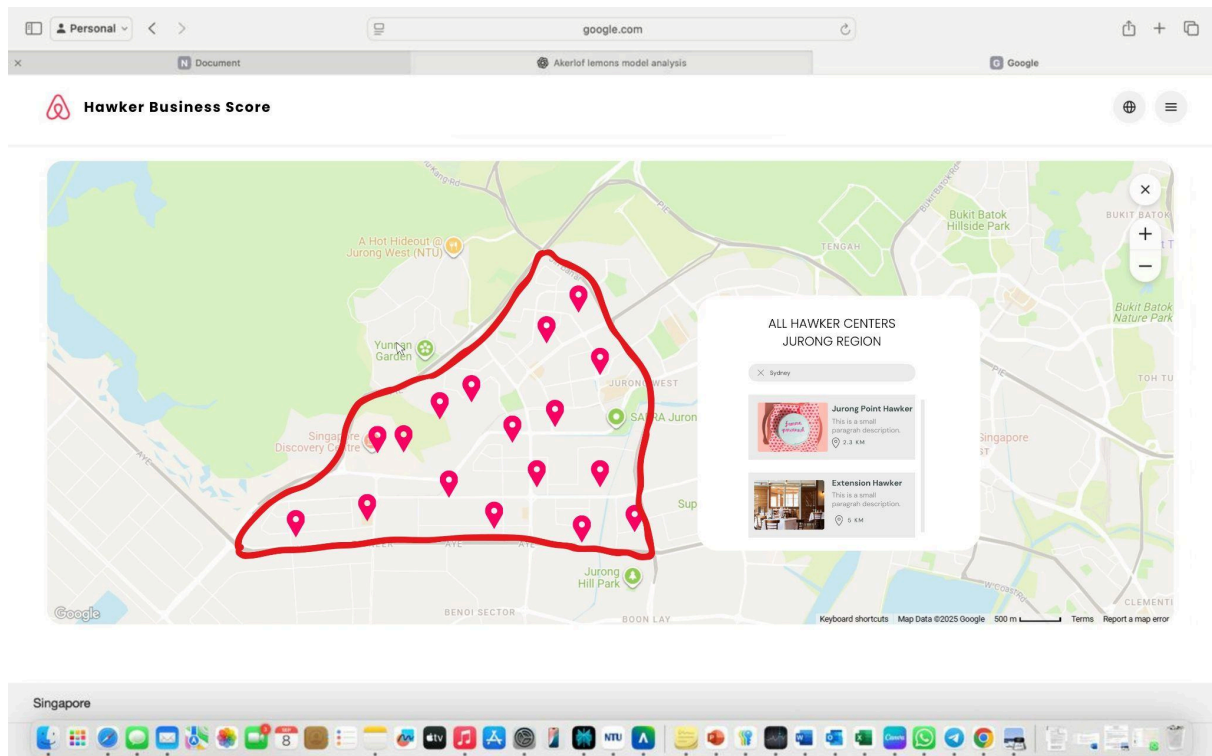
[Log in with singpass](#)



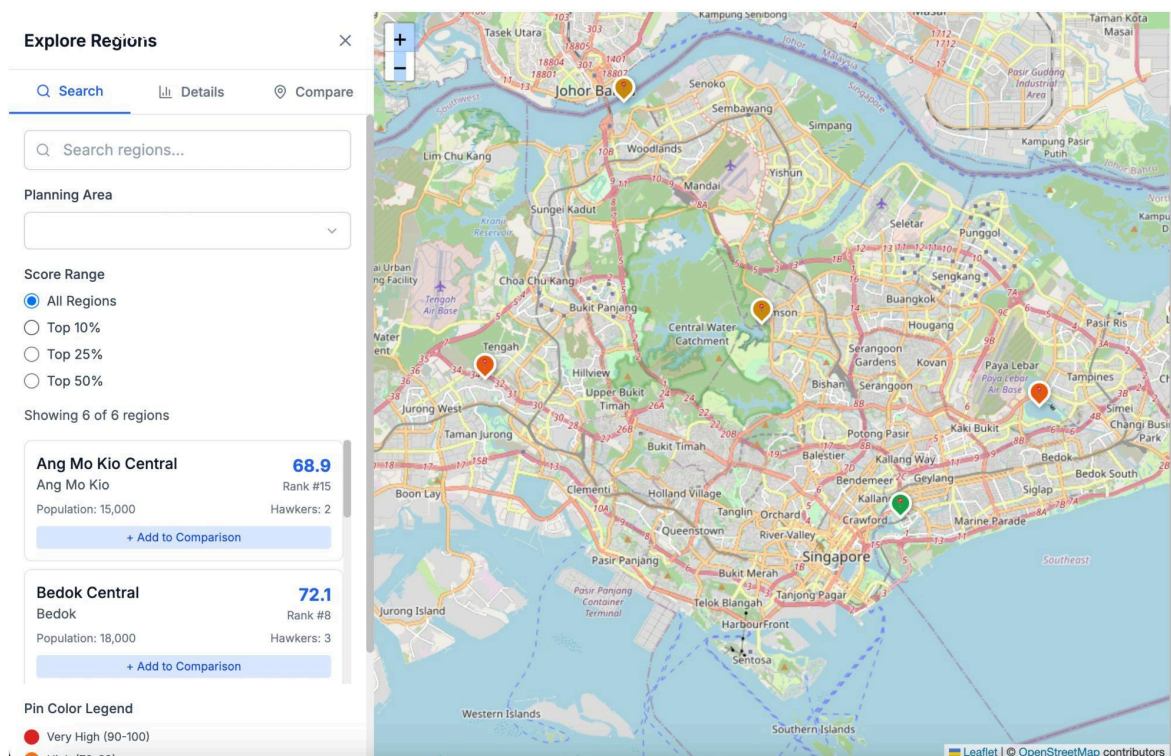
4.2 Home screen (List of all hawker centers)

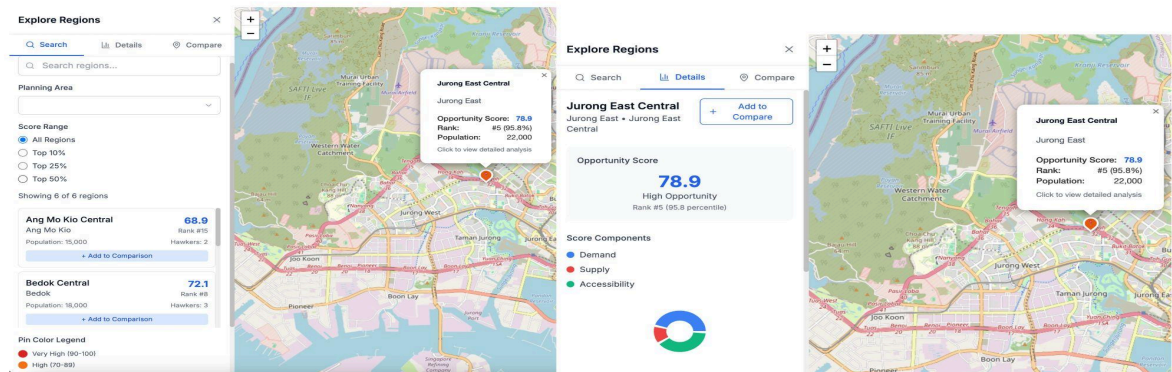


4.3 Regional zoom (List of hawker centers in a region)

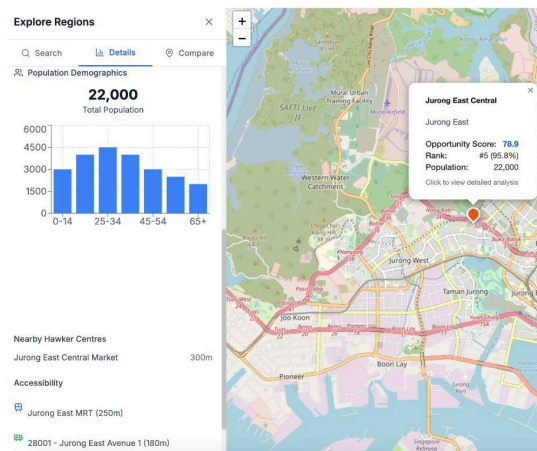


4.4 Filtering and calculating regional business score

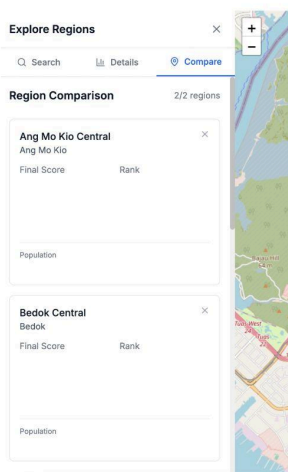
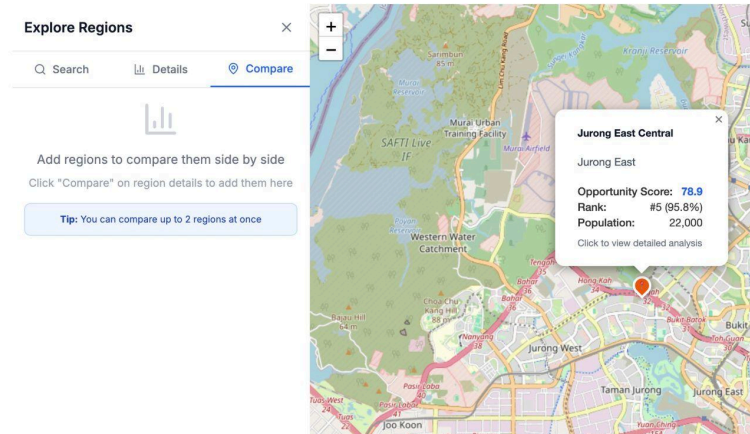




4.5 Statistic of a region



4.6 Statistical and graphical comparison of multiple regions



Key Insights

Bedok Central has a higher opportunity score (72.1 vs 68.9)

Ang Mo Kio Central has 15,000 residents vs **Bedok Central** with 18,000 residents

Ang Mo Kio Central has 2 nearby hawker centres vs **Bedok Central** with 3 nearby hawker centres

4.6 Admin refreshing of data and generation of new scoremap

