Project Overview

SmartLocate SG: Data-Driven Business Site Selector (Web App)

Note: This document is a living record of the project's conceptual and technical framework. The contents within this file will be updated iteratively throughout the **planning** and **building** phases of the project to reflect design changes, technical pivots, and feature refinements.

1. Concept Overview

Goal: Provide SMEs and MNCs a browser-based tool to score and compare potential outlet locations in Singapore using demographics, transport access, and commercial supply/vacancy data from public datasets.

Target Users

- **Small Business Owners:** (F&B, retail, tuition, gyms) planning their first or next outlet.
- **Franchise/MNC Expansion Teams:** Evaluating multiple candidate sites simultaneously.
- **Commercial Agents:** Supporting data-driven recommendations for clients.

2. Core Features (Web App Version)

A. Business Profile & Customer Wizard

- **Web Form/Wizard:**

  Captures the following:

  - **Business Sector:** Professional services, Education & training, Health & wellness, Beauty & personal care, Entertainment & leisure, Supermarket/Retail or Showrooms
  - **Price Band:** $1-$20, $21-$50, $51-$100, $100-$500, $500-$1000, $1000+
  - **Target Age Groups:** 18-24, 25-34, 35-44, 45-54, 55-64, 65+
  - **Target Income Bands:** Low (<S$3,000), Lower-Middle (S$3,000-S$5,000), Middle (S$5,000-S$8,000), Upper-Middle (S$8,000-S$12,000), High (>S$12,000)
  - **Reliance Type:** Walk-in focused, Delivery focused, Mixed
- **Storage:** Store profiles server-side (SQL) for recurring scoring runs.

B. Area & Site Scoring Engine (Backend)

Pre-computes area-level profiles by joining:

- **Demographics:** Age, household size, and economic status by planning area/subzone.
- **Transport:** MRT station density and exit proximity as footfall proxies.
- **Commercial Data:** Vacant space by type/region to proxy rental cost and supply.
  Composite Scoring logic:
- **Demographic Match:** Target age groups and income bands vs. local area distribution.
- **Accessibility Score:** Proximity to MRT stations, MRT exits, and bus stops.
- **Rental Pressure:** Vacancy levels and rental price indicators in the area.
- **Competition Density:** Count of existing similar businesses within a specified radius.

C. Interactive Map & Site Experimentation

- **Base Map:** Interactive map of Singapore with zoom and pan controls (Google Maps API).
- **Choropleth Overlay:** Visualizes composite score, demographics, accessibility, or vacancy by planning area/district/region.
- **Toggleable Layers:** MRT stations, MRT exits (zoom-dependent), bus stops.
- **Point-of-Interest (POI) Pin:** Users can drop a pin at any location to receive a detailed site score breakdown with all dimension scores.
- **Planning Area Details:** Click on districts/regions to view demographic and scoring information.
- **Portfolio:** Save candidate sites with optional names and notes for side-by-side comparison.

D. Scenario Planning

Dynamic controls (sliders/toggles) to adjust weights:

- **Normal / Holiday Peak:** Higher weights to MRT/bus stop proximity and accessibility.
- **Pandemic / Delivery Focus:** Higher weights to residential density; lower weights to MRT proximity.
- **Cost-Saving Mode:** Higher weights to vacancy/rental factors; slightly decreased accessibility weight.
- **Custom Weights:** Manual adjustment of individual scoring dimension weights (must total 100%).
- **Reset to Default:** Restore system default weight configuration.

E. Portfolio & Comparison Dashboard

- **Portfolio Management:** View, edit notes, and delete saved candidate sites for each business profile.
- **Comparison Dashboard:** Select up to **three sites** for side-by-side comparison with tables and charts (bar or radar).

- **Export:** Generate and download PDF comparison reports.

F. Score Explanation & Insights (AI Agent)

- **AI-Generated Explanations:** Custom AI Agent generates rule-based narrative explanations for site scores.
- **Dimension-Specific Insights:** Separate explanations for demographic match, accessibility, rental pressure, and competition density.
- **User-Friendly Language:** Explanations reference specific data points (age groups, income bands, nearby transport, vacancy rates, competitor counts).
- **Feedback Mechanism:** Users can provide feedback on each criterion/insight.

3. Datasets & Integration

Category	Data Source	Usage	Links
Demographics	OneMap API / SingStat	Age, income, and household size by subzone.	https://www.onemap.gov.sg/apidocs/populationquery https://tablebuilder.singstat.gov.sg/table/CT/17563 https://www.jtc.gov.sg/find-space
Transport	LTA DataMall	GEOJSON of MRT stations, exits, and bus stops.	https://datamall.lta.gov.sg/content/datamall/en/dynamic-data.html
Commercial	URA APIs / Corporate Loc.	Vacancy rates and rental price proxies.	https://www.developer.tech.gov.sg/products/categories/data-and-apis/ura-apis/overview https://www.corporatelocations.com.sg/office-rental-rates.php
Competition	Google Maps API / data.gov.sg	Counts of existing F&B/Retail licenses in a radius.	https://developers.google.com/maps

4. Web Tech Stack & Architecture

Frontend

- **Framework:** React.js
- **Routing:** React Router
- **UI Components:** Material UI (MUI)
- **Maps:** Google Maps JS API with GeoJSON overlays.

Backend

- **Framework:** Python (Flask or Django)
- **Endpoints:** RESTful API for profile management and scoring.
- **ETL:** Periodic cron jobs to ingest/update data from OneMap and LTA. (manually triggered, hence may not need a cron job for demo, but can write 1 as an example)

Database (MySQL)

- **Schema:** AREA, AREA_DEMOGRAPHICS, AREA_TRANSPORT_ACCESS, AREA_VACANCY, USER, CANDIDATE_SITE.
    - The final schema may be more complex upon working on the project.
- **Optimization:** Stored procedures for real-time scoring calculations.

Deployment

- Hosted on a local server or cloud environment for demonstration.
- Strict separation of concerns (Presentation, Application, and Data layers).

(OLD) Concept Overview

Concept Overview

Title
SmartLocate SG: Data-Driven Business Site Selector (Web App)

Goal
Provide SMEs and MNCs a browser-based tool to score and compare potential outlet locations in Singapore using demographics, transport access, and commercial supply/vacancy data from public datasets.

Target Users

- Small business owners (F&B, retail, tuition, gyms, services) planning their first or next outlet.
- Franchise/MNC expansion teams evaluating multiple candidate sites.
- Commercial agents supporting data-driven recommendations.

Core Features (Web App Version)

1. **Business Profile & Customer Wizard**
    - Web form/wizard to capture business info: sector (F&B/retail/services), price band, target age groups, desired income bands (proxy via planning-area income/household type), and reliance on walk-in vs delivery (F&B, retail).
    - Store profile server-side (SQL) for later scoring runs.
2. **Area & Site Scoring Engine (Backend)**
    - Pre-compute area-level profiles in the database by joining:
        - Demographics by planning area/subzone (age, household size, economic status).
        - MRT stations and exits (location, density as footfall proxy).
        - Vacant/available commercial space by type/region (proxy for rental cost and supply).
    - For each user profile, compute a composite score per area and per candidate site:
        - Demographic match (target age/income vs local distribution).
        - Accessibility score (distance to nearest MRT station/exit, number of MRT exits within radius, proximity from nearest bus stop).
        - Rental pressure (vacancy levels from URA datasets) and rental price.
3. **Interactive Map & Site Experimentation**
    - Web map (e.g., Google Maps JS API) showing:
        - Choropleth overlay of scores or demographics by planning area.
        - Markers for MRT stations/exits and Bus Stops.
        - Option to click on map to “drop a pin” representing a hypothetical outlet; backend returns a site-specific score and breakdown.
    - Users can save candidate sites into their “portfolio.”
4. **Scenario Planning (Web UI + Backend Logic)**

    - Web controls (sliders/toggles) to switch scenario:
        - Normal / Holiday peak (increase weight of MRT proximity, mall-proximate areas).
        - Pandemic / Delivery focus (increase weight of residential density; reduce MRT weight).
        - Cost-saving mode (increase weight of vacancy/rental proxy / increase weight on the price of the rental, tolerate lower footfall, put some weight on accessibility).
    - Backend recomputes scores using different weight sets; frontend updates ranks and map colors dynamically.
5. **Portfolio & Comparison Dashboard**
    - Logged-in users see a table and charts for all saved candidate sites:
        - Overall score, per-dimension sub-scores, and ranking.
        - Compare up to three sites side-by-side.
6. **Explanation/Insight Panel (Rule-Based "AI")**
    - Instead of a heavy LLM, implement explanation rules:
        - Example: "Strong match: large 25–34 population (+30 pts), high MRT access (+25 pts), but low vacancy (likely higher rents, −10 pts)."
    - Search-style Q&A: user selects a site and clicks "Explain score"; backend returns a pre-computed breakdown that the frontend renders as narrative bullets.

Datasets in the Web Context

Primary integration remains the same, but access switches to server-side ingestion and APIs.

1. **Demographics / Population**
    - Onemap Population Query API: age group, economic status, household size by planning area/subzone.
    - SingStat / data.gov.sg population by planning area and subzone (API).
    - Use server-side jobs (cron or manual admin action) to pull and store into tables like AREA_DEMOGRAPHICS(area_id, age_15_24, age_25_34, income_band_1, …).
    - Links: https://www.onemap.gov.sg/apidocs/populationquery , https://tablebuilder.singstat.gov.sg/table/CT/17563, https://www.jtc.gov.sg/find-space?
2. **Transport / Footfall Proxies**
    - "LTA MRT Station Point / Exits" datasets: locations of MRT stations and exits in GEOJSON.
    - Backend precomputes, for each planning area or a grid, distance to nearest station and count of stations/exits within X meters, storing an ACCESSIBILITY index.
    - Links: https://datamall.lta.gov.sg/content/datamall/en/dynamic-data.html
3. **Commercial Vacancy / Rental Proxy**
    - "Available And Vacant Commercial And Industrial Properties (End Of Period)" (quarterly) and "Available and Vacant Commercial Properties, Annually".
    - These are aggregated by property type and sector; map them to broad regions or planning areas and derive VACANCY_PRESSURE, RENTAL_RISK fields.

    - Links: https://www.developer.tech.gov.sg/products/categories/data-and-apis/ura-apis/overview, https://www.corporatelocations.com.sg/office-rental-rates.php
4. **Competition Indicators (Optional but Strong)**
    - Licensed Food Establishments / Hawker Centres to approximate competition for F&B. Licensed Retail and Services Establishments to approximate competition for Retail and Services.
    - Count establishments in radius around each candidate site and incorporate as COMPETITION_SCORE.
    - Utilize Google Maps API (Nearby Search) to find relevant competitors around the pinned location.

Web Tech Stack & Architecture

Frontend

- Web app using React (as you proposed) or another JS framework.
- Core elements:
    - React Router for pages (Dashboard, Map, Profile, Login/Settings).
    - Component library (e.g., Material UI) for forms, tables, charts.
    - Map library: Google Maps JS API with GeoJSON overlays.

Backend

- Python (Flask/Django) if allowed; previous CZ2006 projects have used Flask.
- REST endpoints, e.g.:
    - POST /api/profile – save business profile.
    - GET /api/areas/scores?scenario=holiday – return ranked area scores.
    - POST /api/site/score – accept lat/lng, return detailed score.
- Periodic ETL jobs to ingest/update data.gov.sg/OneMap datasets into SQL.

Database

- MySQL with:
    - AREA (id, name, centroid, region, …).
    - AREA_DEMOGRAPHICS.
    - AREA_TRANSPORT_ACCESS.
    - AREA_VACANCY.
    - BUSINESS_PROFILE, USER, CANDIDATE_SITE, SITE_SCORES.
- Use views or stored procedures to simplify scoring queries.

Deployment

- Single web app (backend + static frontend) hosted on a local server or cloud for demo. Will mostly be deployed locally for easier demonstration.

Clear separation of concerns for SDLC documentation (presentation, application, data layers).

⚒️ Work Allocation

Lab-1 Work Allocation

Functional & Non-Functional Requirements: Wai Yan (Group Leader)

Data Dictionary: Marcus

Use Case Model + PDF Report on the AI's Critque of the Use Case Diagram and Requirements: Ethan, Yu Hui, Marcus (after completing the Data Dictionary)

UI Mockups: Jung Kit

Work to be completed by: 4th February (Wednesday)

Wai Yan will upload the documents onto the Repo created and shared by the TA by the 5th of February.

✅ Functional Requirements (FR)

Functional Requirements (FR)

By: Wai Yan

Actor ID	Actor Name	Description
A01	Small Business Owner	F&B, retail, tuition, or gym owners planning their first or next outlet location.
A02	Franchise/MNC Expansion Team	Corporate users are evaluating multiple candidate sites simultaneously for business expansion.
A03	Commercial Agent	Real estate professionals supporting data-driven location recommendations for clients.
A04	System Administrator	Internal user responsible for managing system configurations, data updates, and user accounts.
A05	External Data Provider	External systems (OneMap API, LTA DataMall, URA API, Google Maps API) that supply data to the system.

1. Business Profile Management

FR-BPM-001: Create Business Profile

- **Actor:** A01, A02, A03
- **Description:** The system shall allow a logged-in user to create a new business profile by completing the business profile wizard.
- **Input:** Business name, business sector (F&B, Retail, Services, Showroom, etc), price band (different price ranges), target customer age groups, desired income bands, reliance type (Walk-in, Delivery, Mixed).
- **Output:** Business profile saved and confirmation message displayed.
- **Verification Criteria:** Business profile record is created and associated with the user account.

FR-BPM-002: Select Business Sector

- **Actor:** A01, A02, A03
- **Description:** The system should provide a selection of predefined business sectors during business profile creation.
- **Business Sectors :**
    - Professional services (consulting, legal, accounting, engineering)
    - Education & training (tuition centres, enrichment, private schools)
    - Health & wellness (clinics, dental, physio, gyms, spas)
    - Beauty & personal care (salons, barbers, nail studios)
    - Entertainment & leisure (arcades, escape rooms, indoor playgrounds, arts studios)
    - Supermarket/Retail or Showrooms (mini marts, car showrooms)
- **Input:** Sector selection from the dropdown menu.
- **Output:** Selected sector stored in the business profile.
- **Verification Criteria:** Only valid predefined sectors can be selected.

FR-BPM-003: Define Price Band

- **Actor:** A01, A02, A03
- **Description:** The system shall allow the user to specify the price band of different price ranges for their business. (e.g $1 - $20, $21-$50, $ 51-$100, $100 - $500, $500 - $500 - $1000, $1000+)
- **Input:** Multi-select checkbox for Price band.
- **Output:** Price band stored in the business profile.
- **Verification Criteria:** Price band value is correctly associated with the business profile.

FR-BPM-004: Specify Target Age Groups

- **Actor:** A01, A02, A03
- **Description:** The system shall allow the user to select one or more target customer age groups (e.g., 18-24, 25-34, 35-44, 45-54, 55-64, 65+).
- **Input:** Multi-select checkbox for age groups.
- **Output:** Selected age groups stored in the business profile.
- **Verification Criteria:** At least one age group must be selected.

FR-BPM-005: Specify Target Income Bands

- **Actor:** A01, A02, A03
- **Description:** The system shall allow the user to select one or more desired customer income bands based on planning area proxies.
- **Input:** Multi-select checkbox for monthly income bands (Low: >S$3,000 per month, Lower-Middle: S$3,000 – S$5,000, Middle: S$5,000 – S$8,000 , Upper-Middle: S$8,000 – S$12,000, High: Above S$12,000).
- **Output:** Selected income bands stored in the business profile.
- **Verification Criteria:** At least one income band must be selected.

FR-BPM-006: Set Customer Reliance Type

- **Actor:** A01, A02, A03
- **Description:** The system shall allow the user to specify their business's reliance on walk-in customers versus delivery services.
- **Input:** Reliance type selection (Walk-in focused, Delivery focused, Mixed).
- **Output:** Reliance type stored in the business profile.
- **Verification Criteria:** Reliance type influences the scoring algorithm weights.

FR-BPM-007: View Business Profiles

- **Actor:** A01, A02, A03
- **Description:** The system shall display a list of all business profiles created by the logged-in user.
- **Input:** User authentication token if necessary.
- **Output:** List of business profiles with name, sector, and creation date.
- **Verification Criteria:** Only profiles belonging to the authenticated user are displayed.

FR-BPM-008: Edit Business Profile

- **Actor:** A01, A02, A03
- **Description:** The system shall allow a user to modify an existing business profile.
- **Input:** Updated business profile fields.
- **Output:** Updated profile saved and confirmation message displayed.
- **Verification Criteria:** Changes are persisted in the database and reflected in the profile view.

FR-BPM-009: Delete Business Profile

- **Actor:** A01, A02, A03
- **Description:** The system shall allow a user to delete an existing business profile after confirmation.
- **Input:** Delete request with confirmation.
- **Output:** Business profile removed and confirmation message displayed.
- **Verification Criteria:** Profile record is removed from the database along with associated candidate sites.

2. Area & Site Scoring Engine

FR-SSE-001: Calculate Demographic Match Score

- **Actor:** System (triggered by A01, A02, A03)
- **Description:** The system shall calculate a demographic match score by comparing the user's target demographics (age, income) against the local area distribution.
- **Input:** Business profile target demographics, area demographic data.
- **Output:** Demographic match score (0-100 points).

- **Verification Criteria:** Score accurately reflects the percentage overlap between target and local demographics.

FR-SSE-002: Calculate Accessibility Score

- **Actor:** System (triggered by A01, A02, A03)
- **Description:** The system shall calculate an accessibility score based on the proximity to MRT stations, MRT exits, and bus stops.
- **Input:** Site coordinates, transport infrastructure data (MRT stations, exits, bus stops).
- **Output:** Accessibility score (0-100 points).
- **Verification Criteria:** Higher scores are assigned to locations closer to transport nodes.

FR-SSE-003: Calculate Rental Pressure Score

- **Actor:** System (triggered by A01, A02, A03)
- **Description:** The system shall calculate a rental pressure score based on vacancy levels and rental price indicators in the area.
- **Input:** Site coordinates, commercial vacancy data, rental price data.
- **Output:** Rental pressure score (0-100 points).
- **Verification Criteria:** Higher vacancy rates result in lower rental pressure (more favorable) scores.

FR-SSE-004: Calculate Composite Site Score

- **Actor:** System (triggered by A01, A02, A03)
- **Description:** The system shall calculate an overall composite score by combining the demographic match, accessibility, and rental pressure scores with configurable weights.
- **Input:** Individual dimension scores, weight configuration based on the user's business profile.
- **Output:** Composite site score (0-100 points).
- **Verification Criteria:** Composite score equals the weighted sum of individual scores.

FR-SSE-005: Calculate Competition Density Score

- **Actor:** System (triggered by A01, A02, A03)
- **Description:** The system shall calculate a competition density score based on the count of existing similar businesses within a specified radius.
- **Input:** Site coordinates, business sector, radius parameter.
- **Output:** Competition density score (0-100 points).
- **Verification Criteria:** Higher competitor counts result in lower (less favorable) scores.

FR-SSE-006: Pre-compute Area Profiles

- **Actor:** A04, A05
- **Description:** The system shall pre-compute and store area-level (region and districts) profiles by joining demographics, transport access (based on the number of bus stops and MRT stations), and commercial vacancy data.

- **Input:** Raw data from external APIs (demographics, transport, commercial).
- **Output:** Pre-computed area profile records in the database.
- **Verification Criteria:** Area profiles are updated according to the scheduled frequency (for project purposes, this process will be manually triggered from the administrator's dashboard to save time during the demo).

3. Interactive Map & Site Exploration

FR-MAP-001: Display Base Map

- **Actor:** A01, A02, A03
- **Description:** The system shall display an interactive map of Singapore as the primary interface for location exploration.
- **Input:** User navigation to map view (Google Maps API).
- **Output:** Interactive map centered on Singapore with zoom and pan controls.
- **Verification Criteria:** Map loads within the acceptable performance threshold.

FR-MAP-002: Display Choropleth Overlay

- **Actor:** A01, A02, A03
- **Description:** The system shall display a choropleth overlay on the map visualizing scores or demographic data by planning area.
- **Input:** Selected overlay type (composite score, demographic data, accessibility).
- **Output:** Color-coded planning areas based on selected data dimension.
- **Verification Criteria:** Colors accurately represent the data values according to the legend.

FR-MAP-003: Toggle Overlay Data Type

- **Actor:** A01, A02, A03
- **Description:** The system shall allow users to switch between different choropleth overlay types (composite score, demographics, accessibility, vacancy).
- **Input:** Overlay type selection from control panel.
- **Output:** Map overlay updated to reflect selected data type.
- **Verification Criteria:** Overlay updates without requiring page reload.

FR-MAP-004: Display MRT Station Markers

- **Actor:** A01, A02, A03
- **Description:** The system shall display markers indicating the locations of all MRT stations on the map.
- **Input:** User enables MRT stations layer.
- **Output:** MRT station icons displayed at correct geographical positions.
- **Verification Criteria:** All MRT stations from the LTA dataset are displayed.

FR-MAP-005: Display MRT Exit Markers

- **Actor:** A01, A02, A03
- **Description:** The system shall display markers indicating MRT exit locations when zoomed to sufficient detail.
- **Input:** Map zoom level exceeds threshold; MRT exits layer enabled.
- **Output:** MRT exit icons displayed at correct positions.
- **Verification Criteria:** Exit markers appear only at appropriate zoom levels.

FR-MAP-006: Display Bus Stop Markers

- **Actor:** A01, A02, A03
- **Description:** The system shall display markers indicating bus stop locations on the map.
- **Input:** User enables bus stops layer.
- **Output:** Bus stop icons displayed at correct geographical positions.
- **Verification Criteria:** Bus stops from the LTA dataset are accurately positioned.

FR-MAP-007: Toggle Map Layers

- **Actor:** A01, A02, A03
- **Description:** The system shall allow users to toggle visibility of individual map layers (MRT stations, MRT exits, bus stops, choropleth).
- **Input:** Layer toggle controls.
- **Output:** Selected layers shown or hidden on the map.
- **Verification Criteria:** Layer visibility changes immediately upon toggling.

FR-MAP-008: Drop Point-of-Interest (POI) Pin

- **Actor:** A01, A02, A03
- **Description:** The system shall allow users to drop a point-of-interest (POI) pin at any location on the map to evaluate a potential site.
- **Input:** Click/tap on map location.
- **Output:** Pin placed at selected coordinates. The site evaluation panel is triggered after the user clicks confirm for the system to start evaluating the score.
- **Verification Criteria:** Pin coordinates match the clicked location accurately.

FR-MAP-009: Display Site Score Breakdown

- **Actor:** A01, A02, A03
- **Description:** The system shall display a detailed score breakdown for a pinned location, showing individual dimension scores. The detailed score breakdown is to be shown on the left/right side of the site
- **Input:** POI pin placement or selection.
- **Output:** Panel displaying composite score and breakdown (demographic, accessibility, rental, competition).

- **Verification Criteria:** Scores match the calculation based on active business profile and scenario weights.

FR-MAP-010: View Planning Area Details

- **Actor:** A01, A02, A03
- **Description:** The system shall display detailed demographic and scoring information when a user clicks on a specific district or region.
- **Input:** Click on the district or region's Info (ℹ) icon.
- **Output:** Information panel with area name, population demographics, and area-level scores.
- **Verification Criteria:** Displayed data matches the pre-computed area profile.

4. Candidate Site Portfolio Management

FR-CSP-001: Save Candidate Site

- **Actor:** A01, A02, A03
- **Description:** The system shall allow users to save a pinned location as a candidate site to their portfolio.
- **Input:** POI pin location, site name (optional), associated business profile.
- **Output:** Candidate site saved to portfolio; confirmation message displayed.
- **Verification Criteria:** Site record created in database with coordinates, scores, and association to business profile.

FR-CSP-002a: View Candidate Site Portfolio

- **Actor:** A01, A02, A03
- **Description:** The system shall display a list of all candidate sites saved for a selected business profile.
- **Input:** Business profile selection.
- **Output:** List of candidate sites with name, address, composite score, and save date.
- **Verification Criteria:** Only sites associated with the selected business profile are displayed.

FR-CSP-002b: Delete Candidate Site

- **Actor:** A01, A02, A03
- **Description:** The system shall allow users to remove a candidate site from their portfolio.
- **Input:** Delete request for specific candidate site.
- **Output:** Site removed from portfolio; confirmation message displayed.
- **Verification Criteria:** Site record is deleted from the database.

FR-CSP-003: View Candidate Site on Map

- **Actor:** A01, A02, A03
- **Description:** The system shall center the map on a selected candidate site and display its pin and score breakdown.
- **Input:** Selection of candidate site from portfolio list.
- **Output:** Map centered on site; POI pin and score panel displayed.
- **Verification Criteria:** Map navigates to the correct coordinates.

FR-CSP-004: Add Notes to Candidate Site

- **Actor:** A01, A02, A03
- **Description:** The system shall allow users to add or edit notes for a saved candidate site.
- **Input:** Text notes for candidate site.
- **Output:** Notes saved and associated with the candidate site.
- **Verification Criteria:** Notes are persisted and displayed when viewing the site.

5. Scenario Planning

FR-SCP-001: Select Predefined Scenario

- **Actor:** A01, A02, A03
- **Description:** The system shall provide predefined scenario options (Normal/Holiday Peak, Pandemic/Delivery Focus, Cost-Saving Mode) that adjust scoring weights.
- **Input:** Scenario selection from dropdown or button group.
- **Output:** Scoring weights updated according to selected scenario; map and scores recalculated.
- **Verification Criteria:** Weight values match the predefined scenario configuration.

FR-SCP-002a: Apply Normal/Holiday Peak Scenario

- **Actor:** A01, A02, A03
- **Description:** The system shall apply higher weights to MRT proximity and mall-proximate areas when the Normal/Holiday Peak scenario is selected.
- **Input:** Selection of Normal/Holiday Peak scenario.
- **Output:** Accessibility weight increased; scores recalculated and displayed.
- **Verification Criteria:** Sites near MRT stations and bus stops show improved scores compared to baseline. Sites near carparks also show improved scores compared to baseline however, not as much as MRT stations and bus stops.

FR-SCP-002b: Apply Pandemic/Delivery Focus Scenario

- **Actor:** A01, A02, A03
- **Description:** The system shall apply higher weights to residential density and lower weights to MRT proximity when the Pandemic/Delivery scenario is selected.
- **Input:** Selection of Pandemic/Delivery Focus scenario.

- **Output:** Demographic (residential) weight increased; MRT weight decreased; scores recalculated.
- **Verification Criteria:** High-density residential areas show improved scores.

FR-SCP-002c: Apply Cost-Saving Mode Scenario

- **Actor:** A01, A02, A03
- **Description:** The system shall apply higher weights to vacancy/rental factors and higher tolerance for lower footfall (if footfall data can be obtained) when Cost-Saving Mode is selected.
- **Input:** Selection of Cost-Saving Mode scenario.
- **Output:** Rental pressure weight increased; accessibility weight slightly decreased; scores recalculated.
- **Verification Criteria:** Areas with high vacancy rates show improved scores.

FR-SCP-002d: Custom Weight Adjustment

- **Actor:** A01, A02, A03
- **Description:** The system shall allow users to manually adjust individual scoring weights using sliders or input fields.
- **Input:** Weight values for each scoring dimension (0-100% each, total must equal 100%).
- **Output:** Custom weights applied; scores recalculated and displayed.
- **Verification Criteria:** Composite scores reflect the custom weight configuration.

FR-SCP-002e: Reset to Default Weights

- **Actor:** A01, A02, A03
- **Description:** The system shall allow users to reset scoring weights to the system default values.
- **Input:** Reset to default button click.
- **Output:** Weights restored to default; scores recalculated.
- **Verification Criteria:** Weight values match system defaults after reset.

6. Site Comparison Dashboard

FR-SCD-001: Select Sites for Comparison

- **Actor:** A01, A02, A03
- **Description:** The system shall allow users to select up to three candidate sites for side-by-side comparison.
- **Input:** Selection of 2-3 candidate sites from portfolio.
- **Output:** Comparison dashboard populated with selected sites.
- **Verification Criteria:** System enforces a maximum of three sites for comparison.

FR-SCD-002: Display Comparison Table

- **Actor:** A01, A02, A03
- **Description:** The system shall display a comparison table showing key metrics for selected candidate sites.
- **Input:** Selected candidate sites.
- **Output:** Table with rows for each metric (composite score, demographic match, accessibility, rental pressure, competition) and columns for each site.
- **Verification Criteria:** All relevant metrics are displayed accurately for each site.

FR-SCD-003: Display Comparison Charts

- **Actor:** A01, A02, A03
- **Description:** The system shall display visual charts (bar or radar) comparing the score dimensions of selected sites.
- **Input:** Selected candidate sites.
- **Output:** Chart visualization of scores across sites.
- **Verification Criteria:** Chart accurately represents the numerical scores.

FR-SCD-004: Export Comparison Report

- **Actor:** A01, A02, A03
- **Description:** The system shall allow users to export the comparison data as a PDF report.
- **Input:** Export request from comparison dashboard.
- **Output:** PDF file generated and downloaded containing comparison table and charts.
- **Verification Criteria:** PDF contains all displayed comparison data in readable format.

7. Score Explanation & Insights

FR-SEI-001: Display Score Explanation Panel

- **Actor:** A01, A02, A03
- **Description:** The system shall display an explanation panel that provides rule-based narrative explanations for a site's score.
- **Input:** Request to the custom AI Agent to explain the score for a candidate site or POI pin.
- **Output:** Panel with bullet-point explanations (e.g., "Strong match: large 25-34 population (+30 pts)").
- **Verification Criteria:** Explanations accurately reflect the scoring logic and data values.

FR-SEI-002: Generate Demographic Insight

- **Actor:** System (triggered by A01, A02, A03)
- **Description:** The system shall generate a narrative explanation for the demographic match component of a score.
- **Input:** Site demographic score, target demographics, local demographics.

- **Output:** Text explanation describing demographic alignment or gaps.
- **Verification Criteria:** Explanation references specific age groups or income bands.

FR-SEI-003: Generate Accessibility Insight

- **Actor:** System (triggered by A01, A02, A03)
- **Description:** The system shall generate a narrative explanation for the accessibility component of a score.
- **Input:** Site accessibility score, nearby transport nodes.
- **Output:** Text explanation describing transport access
- **Verification Criteria:** Explanation references specific transport infrastructure.

FR-SEI-004: Generate Rental Insight

- **Actor:** System (triggered by A01, A02, A03)
- **Description:** The system shall generate a narrative explanation for the rental pressure component of a score.
- **Input:** Site rental score, area vacancy data.
- **Output:** Text explanation describing rental market conditions.
- **Verification Criteria:** Explanation references vacancy rates or rental price indicators.

FR-SEI-005: Generate Competition Insight

- **Actor:** System (triggered by A01, A02, A03)
- **Description:** The system shall generate a narrative explanation for the competition density component of a score.
- **Input:** Site competition score, competitor count.
- **Output:** Text explanation describing competitive landscape
- **Verification Criteria:** Explanation includes competitor count and impact on score.

8. Data Management & Integration

FR-DMI-001: Ingest Demographic Data

- **Actor:** A04, A05
- **Description:** The system shall periodically retrieve and store demographic data (age distribution, household size, economic status) from OneMap API and SingStat.
- **Input:** API calls to OneMap/SingStat endpoints.
- **Output:** Demographic data stored in AREA_DEMOGRAPHICS table.
- **Verification Criteria:** Data is updated according to configured ETL schedule.

FR-DMI-002: Ingest Transport Data

- **Actor:** A04, A05
- **Description:** The system shall periodically retrieve and store transport infrastructure data (MRT stations, exits, bus stops) from LTA DataMall.
- **Input:** API calls to LTA DataMall endpoints.
- **Output:** Transport data stored in AREA_TRANSPORT_ACCESS table; GeoJSON files updated.
- **Verification Criteria:** All MRT stations and bus stops from LTA dataset are captured.

FR-DMI-003: Ingest Commercial Vacancy Data

- **Actor:** A04, A05
- **Description:** The system shall periodically retrieve and store commercial vacancy and rental data from URA APIs.
- **Input:** API calls to URA API endpoints.
- **Output:** Vacancy and rental data stored in AREA_VACANCY table.
- **Verification Criteria:** Data reflects current URA published statistics.

FR-DMI-004: Ingest Competition Data

- **Actor:** A04, A05
- **Description:** The system shall retrieve business license and POI data for competition analysis from data.gov.sg and Google Maps API.
- **Input:** API calls to data.gov.sg and Google Maps Places API.
- **Output:** Competition data stored for scoring calculations.
- **Verification Criteria:** F&B and retail license data is current and geocoded.

FR-DMI-005: Execute Score Calculation

- **Actor:** A04
- **Description:** The system shall calculate the scores based on the current data from the APIs.
- **Input:** Cron job trigger.
- **Output:** Data tables updated; Log entry created.
- **Verification Criteria:** Complete within an acceptable time window and log results.

FR-DMI-006: View Data Update Status

- **Actor:** A04
- **Description:** The system shall allow administrators to view the status and timestamp of the last successful data update for each data source.
- **Input:** Navigation to admin data status page.
- **Output:** Table showing data source, last update timestamp, and status.
- **Verification Criteria:** Timestamps accurately reflect the most recent Cron job completion.

✅ Non-Functional Requirements (NFR)

Non-Functional Requirements (NFR)

By: Wai Yan

1. Usability Requirements

NFR-USA-001: First-Time User Business Profile Completion

- **Description:** First-time users shall be able to create a business profile and generate a site score within 5 minutes without external assistance.
- **Measurement:** User testing with task completion time tracking.
- **Rationale:** Ensures the business profile wizard is intuitive for non-technical SME owners.

NFR-USA-002: Map Navigation Intuitiveness

- **Description:** Users shall be able to navigate the map, toggle layers, and drop a POI pin within 2 minutes of first use. The experience must be similar to using Google Maps.
- **Measurement:** User testing with success rate measurement.
- **Rationale:** Map interaction is the core feature and must be immediately accessible.

NFR-USA-003: Mobile Responsiveness

- **Description:** The web application shall be fully functional on tablet devices.
- **Measurement:** Responsive design testing on tablet viewports.
- **Rationale:** Users may access the application on tablets during site visits.

NFR-USA-004: Score Explanation Clarity

- **Description:** Users shall correctly interpret the meaning of score explanations without requiring additional help documentation.
- **Measurement:** User comprehension testing with the ability for users to provide feedback for each criterion/insight.
- **Rationale:** Rule-based explanations must be understandable by non-technical users.

NFR-USA-005: Error Message Clarity

- **Description:** All error messages shall clearly describe the error and suggest corrective action in user-friendly language.
- **Measurement:** Review of all error message strings against clarity guidelines.
- **Rationale:** Users should understand how to resolve issues without technical support.

NFR-USA-006: Language Support

- **Description:** The application interface shall be available in English as the primary language.
- **Measurement:** Verification of all UI text in English.
- **Rationale:** English is the common business language in Singapore.

2. Reliability Requirements

NFR-REL-001: Graceful Degradation on API Failure

- **Description:** The system shall continue to function with cached data when external APIs (OneMap, LTA, URA) are temporarily unavailable, displaying a notification to users about data freshness.
- **Measurement:** Simulated API failure testing with user notification verification.
- **Rationale:** External API outages should not completely disable the application.

NFR-REL-002: Session Persistence

- **Description:** User sessions shall persist for at least 30 minutes of inactivity before requiring re-authentication.
- **Measurement:** Session timeout testing.
- **Rationale:** Prevents user frustration from frequent logouts during exploration.

NFR-REL-003: Error Recovery

- **Description:** The system shall automatically retry failed external API requests up to 3 times with exponential backoff before displaying an error to the user.
- **Measurement:** API failure handling testing.
- **Rationale:** Transient network issues should not immediately result in user-facing errors.

3. Performance Requirements

NFR-PER-001: Map Initial Load Time

- **Description:** The interactive map shall load and be interactive within 3 seconds.
- **Measurement:** Page load testing with performance monitoring tools.
- **Rationale:** Users expect fast initial load for web applications.

NFR-PER-002: Score Calculation Response Time

- **Description:** The system shall calculate and display a site score within 5 seconds of a user dropping a POI pin. The analysis of the score can take some time as it has to be generated by the AI Agent.

- **Measurement:** Response time logging for scoring requests.
- **Rationale:** Real-time feedback is essential for the site exploration workflow.

NFR-PER-003: Choropleth Overlay Rendering

- **Description:** The choropleth overlay shall render completely within 2 seconds when toggled or when the overlay type is changed.
- **Measurement:** Rendering time measurement.
- **Rationale:** Overlay changes should feel immediate to users.

NFR-PER-004: Search Response Time

- **Description:** Location search results shall be displayed within 1.5 seconds of query submission.
- **Measurement:** Search API response time logging.
- **Rationale:** Address lookup should be near-instantaneous.

NFR-PER-005: Database Query Performance

- **Description:** All database queries for scoring and retrieval shall be completed within 500 milliseconds.
- **Measurement:** Database query execution time monitoring.
- **Rationale:** Fast database access is critical for responsive scoring.

NFR-PER-006: PDF Export Generation Time

- **Description:** Comparison report PDF generation shall complete within 5 seconds.
- **Measurement:** PDF generation time logging.
- **Rationale:** Users should not wait excessively for report downloads.

NFR-PER-007: Regional/District Score Calculation Job Completion Time

- **Description:** Score Calculation Job upon data refresh shall complete within 30 minutes. The web app from the users' perspective should not be affected by the score calculation job and show continue to display the pre-refresh data. The overall score will be updated on the users' perspective upon the completion of the Score Calculation Job.
- **Measurement:** Score Calculation Job duration logging.
- **Rationale:** Data updates should not impact system performance while users are using the app.

4. Scalability Requirements

NFR-SCA-001: Database Scalability

- **Description:** The database schema shall support at least 10,000 user accounts and 100,000 candidate site records without query performance degradation.

- **Measurement:** Performance testing with scaled data volume.
- **Rationale:** Ensures long-term data growth is accommodated.

NFR-SCA-002: API Rate Limit Handling

- **Description:** The system shall implement rate limiting and queuing for external API calls to stay within provider limits (Google Maps, LTA, URA, OneMap).
- **Measurement:** API call rate monitoring and limit compliance verification.
- **Rationale:** Prevents service disruption due to API quota exhaustion.

5. Security Requirements

NFR-SEC-001: Password Encryption

- **Description:** User passwords shall be stored using bcrypt hashing with a minimum cost factor of 10.
- **Measurement:** Database inspection and hashing algorithm verification.
- **Rationale:** Protects user credentials from exposure in case of database breach.

NFR-SEC-002: Session Token Security

- **Description:** Session tokens shall be cryptographically random, stored securely (HttpOnly, Secure cookies), and expire after 24 hours of inactivity.
- **Measurement:** Session management security audit.
- **Rationale:** Prevents session hijacking and unauthorized access.

NFR-SEC-003: SQL Injection Prevention

- **Description:** All database queries shall use parameterized queries or prepared statements to prevent SQL injection attacks.
- **Measurement:** Code review and security scanning.
- **Rationale:** Protects database integrity and user data.

NFR-SEC-004: API Key Protection

- **Description:** External API keys shall be stored as environment variables or in a secure vault, never in source code or client-side code.
- **Measurement:** Code repository scanning; configuration audit.
- **Rationale:** Prevents API key exposure and unauthorized usage.

NFR-SEC-005: User Data Privacy

- **Description:** User business profiles and candidate sites shall be accessible only to the owning user account.

- **Measurement:** Authorization testing with multiple user accounts.
- **Rationale:** Ensures business data confidentiality between users.

6. Supportability & Maintainability Requirements

NFR-SUP-001: Modular Architecture

- **Description:** The system shall follow a layered architecture (Presentation, Application, Data) with clear separation of concerns.
- **Measurement:** Architecture documentation review; dependency analysis.
- **Rationale:** Enables independent updates to each layer without cascading changes.

NFR-SUP-002: API Documentation

- **Description:** All backend REST API endpoints shall be documented using OpenAPI (Swagger) specification.
- **Measurement:** API documentation completeness audit.
- **Rationale:** Facilitates frontend development and future API consumers.

NFR-SUP-003: Code Documentation

- **Description:** All public functions and classes shall include docstrings or comments describing purpose, parameters, and return values.
- **Measurement:** Code documentation coverage analysis.
- **Rationale:** Enables efficient onboarding of new developers.

NFR-SUP-004: Configuration Externalization

- **Description:** All environment-specific configurations (API endpoints, database connections, feature flags) shall be externalized in configuration files or environment variables.
- **Measurement:** Configuration audit.
- **Rationale:** Enables deployment to different environments without code changes.

NFR-SUP-005: Logging Standards

- **Description:** The system shall implement structured logging (JSON format) with severity levels (DEBUG, INFO, WARN, ERROR) for all components.
- **Measurement:** Log format verification.
- **Rationale:** Facilitates debugging and monitoring.

NFR-SUP-006: Version Control

- **Description:** All source code shall be maintained in a Git repository with meaningful commit messages and branch management.

- **Measurement:** Repository history review.
- **Rationale:** Enables collaboration, rollback, and change tracking.

7. Compatibility Requirements

NFR-COM-001: Browser Compatibility

- **Description:** The web application shall function correctly on the latest versions of Chrome, Firefox, Safari, and Edge browsers.
- **Measurement:** Cross-browser testing on specified browsers.
- **Rationale:** Accommodates diverse user browser preferences.

NFR-COM-002: External API Version Compatibility

- **Description:** The system shall be designed to adapt to version changes in external APIs (Google Maps, LTA DataMall) through abstraction layers.
- **Measurement:** API integration code review.
- **Rationale:** Minimizes disruption when external APIs are updated.

🎨 Use Case Model

Use Case Model

By: Ethan, Yu Hui, Marcus

Discussions

Focus: Based on users: Business Users, Database/API Calls, API Providers

Need an authentication system -> sign up, registration, password , forget password functionality

- Change password (use otp for resetting password) or 2fa if theres time
- Password saved in DB using a certain cryptography method

Business Users

Admin: oversees the users

Information stored in the Database

- Authentication (Name, email, password)
- Business Profile (Category, Price, Ages, Operating Method)

Pages

Dashboard - show the data

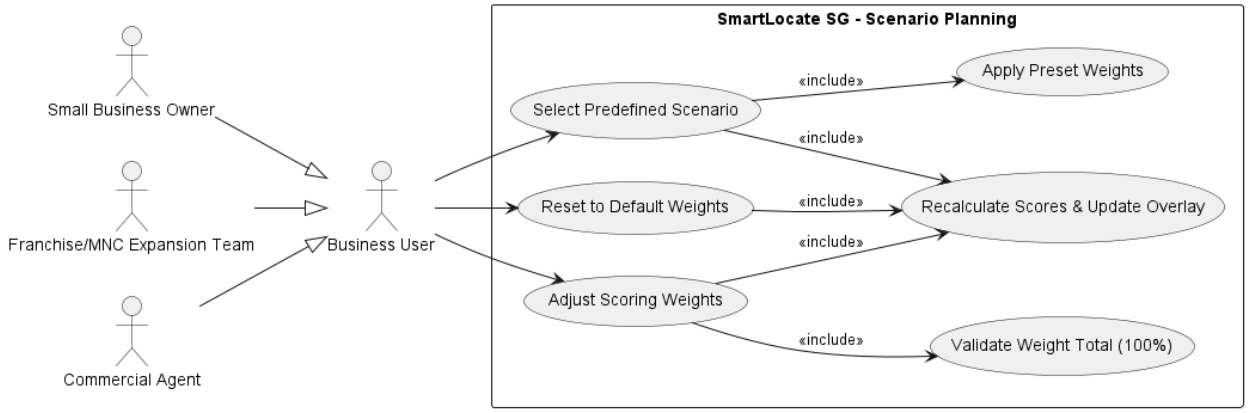
Business Profiles (Create, View)

Map

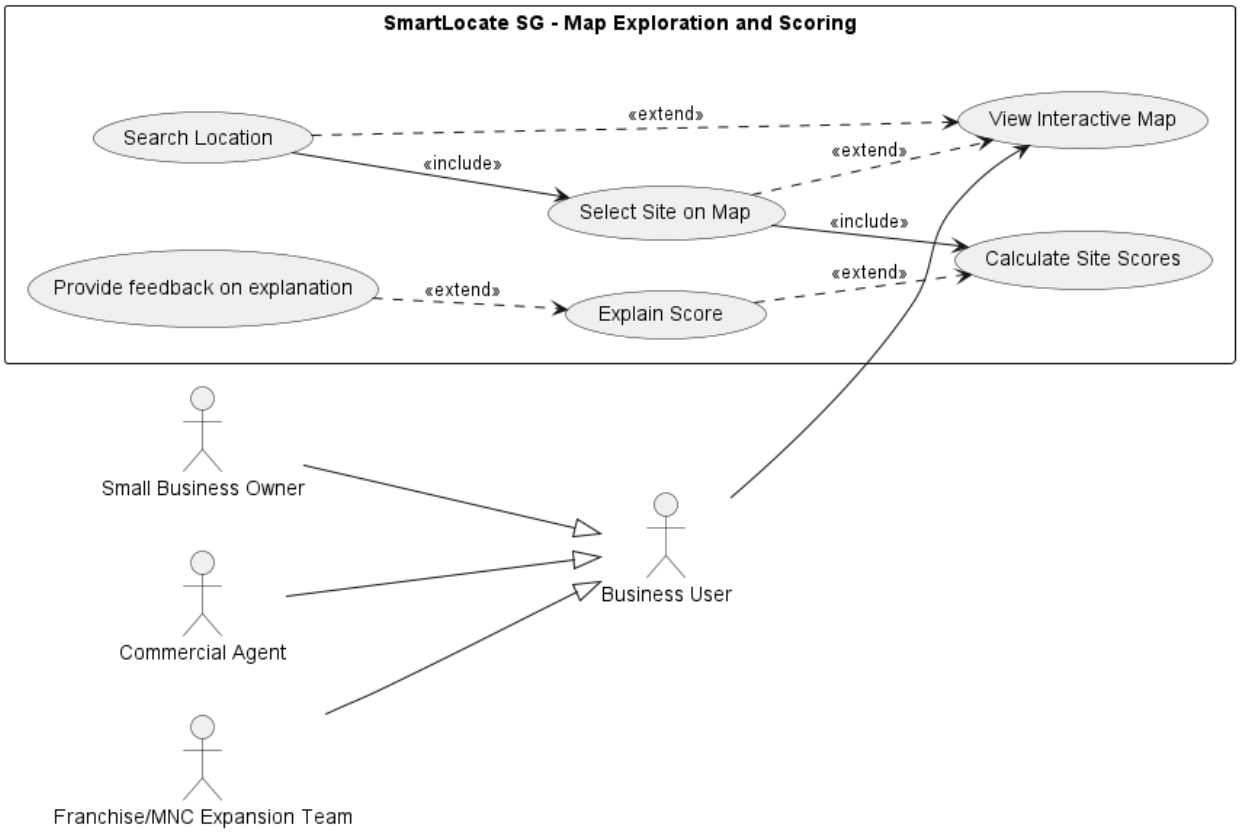
Portfolio (Different Location)

Compare (Location Scores)

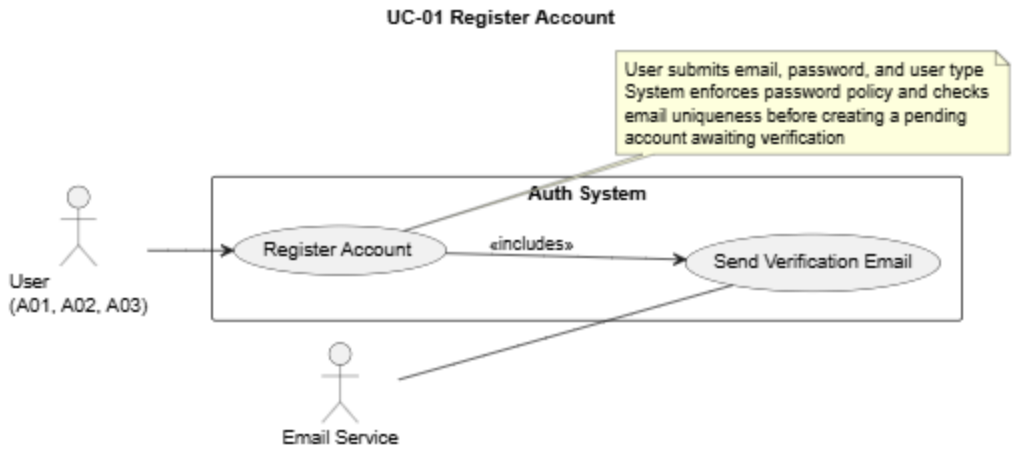
Scenario Planning



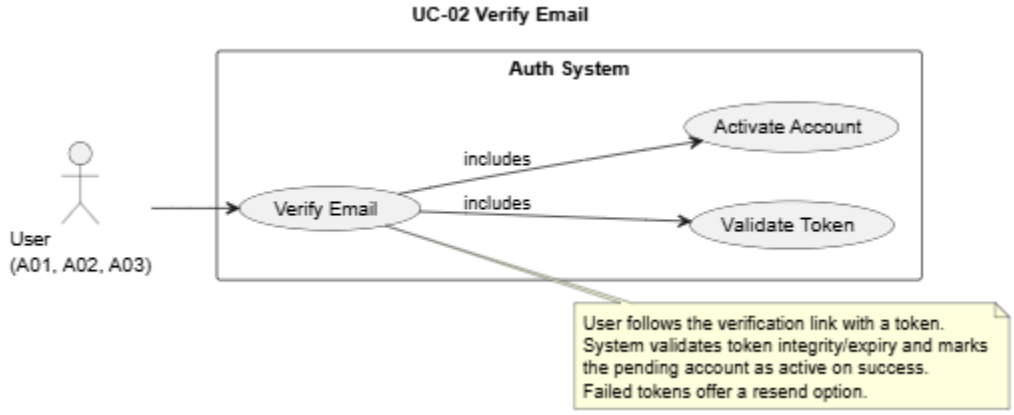
Map Exploration and Scoring



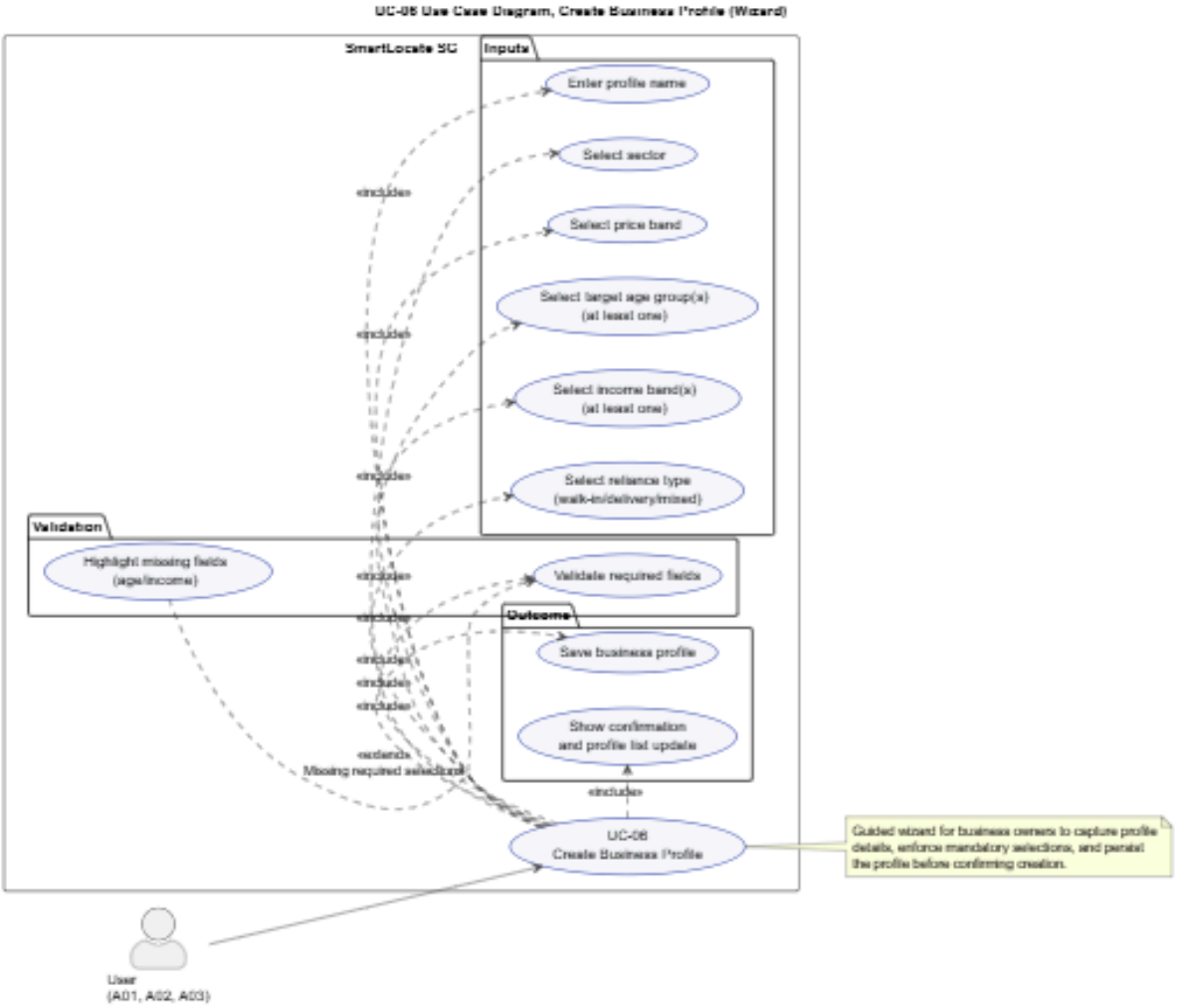
UC-01 - Register Account



UC-02 - Verify Email



UC-05 - Use Case Diagram, Create Business Profile (Wizard)



📖 Data Dictionary

Data Dictionary Table

By: Marcus

USER			
Field Name	**Data Type**	**Description**	**Example**
user_id	BIGINT	Unique user ID	100025
name	VARCHAR(80)	Display name	Dale Smith
email	VARCHAR(120)	Login email, unique	example@example.com
type_code	VARCHAR(16)	User category	SME
status	VARCHAR(16)	Account status	ACTIVE
created_at	DATETIME	Account creation time	2026-01-28 9:12:00
last_login_at	DATETIME	Last login time	2026-01-28 23:05:10
BUSINESS_PROFILE			
Field Name	**Data Type**	**Description**	**Example**
profile_id	BIGINT	Unique profile ID	200301
user_id	BIGINT	Owner user ID	100025
profile_name	VARCHAR(80)	Friendly name for profile	“Budget Gym North East”
sector_code	VARCHAR(32)	Business sector	HEALTH_WELLNESS
band_code	VARCHAR(16)	Price band	P02
reliance_code	VARCHAR(16)	Walk in vs delivery reliance	WALKIN
active_weight_config_id	BIGINT	Current weight configuration	90012
created_at	DATETIME	Created time	2026-01-28 10:00:00
updated_at	DATETIME	Last update time	2026-01-28 22:30:00

PROFILE_TARGET_AGE_GROUP			
Field Name	**Data Type**	**Description**	**Example**
profile_id	BIGINT	Business profile ID	200301
age_group_code	VARCHAR(16)	Selected age bracket	25_34
PROFILE_TARGET_INCOME_BAND			
Field Name	**Data Type**	**Description**	**Example**
profile_id	BIGINT	Business profile ID	200301
income_band_code	VARCHAR(16)	Selected income band	5000_8000
PLANNING_AREA			
Field Name	**Data Type**	**Description**	**Example**
area_id	BIGINT	Unique planning area ID	30018
area_name	VARCHAR(80)	Planning area name	“BISHAN”
area_code	VARCHAR(16)	Optional code from source	“BSN”
region_name	VARCHAR(40)	Region grouping	“CENTRAL”
geometry_geojson	LONGTEXT	Polygon boundary as GeoJSON text	{"type":"Polygon","coordinates":[...]}
centroid_lat	DECIMAL(9,6)	Area centroid latitude	1.3509
centroid_lng	DECIMAL(9,6)	Area centroid longitude	103.8485
source	VARCHAR(32)	Boundary source	URA_MP2019
source_version	VARCHAR(16)	Version or year	2019
last_updated_at	DATETIME	Last boundary refresh time	2026-01-10 3:00:00
SUBZONE (optional)			
Field Name	**Data Type**	**Description**	**Example**

subzone_id	BIGINT	Unique subzone ID	400211
subzone_name	VARCHAR(80)	Subzone name	“BISHAN EAST”
subzone_code	VARCHAR(16)	Optional code from source	“BSN_E”
area_id	BIGINT	Parent planning area ID	30018
geometry_geojson	LONGTEXT	Subzone polygon GeoJSON	{"type":"Polygon","coordinates":[...]}
centroid_lat	DECIMAL(9,6)	Subzone centroid latitude	1.3523
centroid_lng	DECIMAL(9,6)	Subzone centroid longitude	103.8499
last_updated_at	DATETIME	Last boundary refresh time	2026-01-10 3:00:00
AREA_DEMOGRAPHICS			
Field Name	**Data Type**	**Description**	**Example**
demo_id	BIGINT	Unique demographics row ID	500881
area_id	BIGINT	Planning area ID	30018
subzone_id	BIGINT	Subzone ID, null if area level	400211
effective_date	DATE	Data effective date	2025-12-31
population_total	INT	Total population	84200
age_distribution_json	JSON	Age distribution by group code	{"18_24":8200,"25_34":14500}
income_distribution_json	JSON	Income distribution by band	{"LT3000":12000,"5000_8000":21000}
household_size_json	JSON	Household size distribution	{"1_2":9000,"3_4":15000}
economic_status_json	JSON	Optional economic indicators	{"employed":41000,"students":9000}

source	VARCHAR(32)	Data source tag	ONEMAP_POPQUERY
ingested_at	DATETIME	When you stored it	2026-01-20 2:10:00
TRANSPORT_NODE			
Field Name	**Data Type**	**Description**	**Example**
node_id	BIGINT	Unique transport node ID	6001201
node_type	VARCHAR(16)	Node category	MRT_EXIT
external_id	VARCHAR(32)	ID from provider	“NS17_A”
name	VARCHAR(120)	Station or stop name	“Bishan”
label	VARCHAR(120)	Exit code or stop description	“Exit A”
lat	DECIMAL(9,6)	Latitude	1.35021
lng	DECIMAL(9,6)	Longitude	103.84802
area_id	BIGINT	Derived planning area ID	30018
source	VARCHAR(32)	Data source	LTA_DATAMALL
effective_date	DATE	Optional data date	2026-01-01
ingested_at	DATETIME	When stored	2026-01-20 2:30:00
AREA_TRANSPORT_ACCESS			
Field Name	**Data Type**	**Description**	**Example**
access_id	BIGINT	Unique access summary ID	700441
area_id	BIGINT	Planning area ID	30018
effective_date	DATE	Snapshot date	2026-01-01
mrt_station_count_1km	SMALLINT	Stations within 1 km	2
mrt_exit_count_500m	SMALLINT	Exits within 500 m	6

bus_stop_count_500 m	SMALLINT	Bus stops within 500 m	18
dist_to_nearest_mrt_ m	INT	Metres to nearest station	210
dist_to_nearest_bus_ stop_m	INT	Metres to nearest bus stop	65
accessibility_index	DECIMAL(6,2)	Normalised index 0 to 100	82.5
computed_at	DATETIME	Computation time	2026-01-20 3:10:00
AREA_VACANCY			
Field Name	**Data Type**	**Description**	**Example**
vacancy_id	BIGINT	Unique vacancy row ID	800991
area_id	BIGINT	Planning area ID, null if region level	30018
region_name	VARCHAR(40)	Region name if aggregated	“CENTRAL”
property_type	VARCHAR(40)	Property type	RETAIL
sector_type	VARCHAR(16)	Private or public	PRIVATE
period_type	VARCHAR(16)	Quarter or year	QUARTER
period_start	DATE	Period starting date	2025-10-01
metric	VARCHAR(24)	Metric name	VACANCY_RATE
value	DECIMAL(14,3)	Metric value	6.4
unit	VARCHAR(32)	Unit label	percent
source	VARCHAR(32)	Data source	URA
ingested_at	DATETIME	When stored	2026-01-18 1:00:00
WEIGHT_CONFIG			
Field Name	Data Type	Description	Example

config_id	BIGINT	Unique weight config ID	90012
owner_user_id	BIGINT	Null for system presets	100025
demographic_weight	TINYINT	Weight percent	35
accessibility_weight	TINYINT	Weight percent	35
rental_weight	TINYINT	Weight percent	20
competition_weight	TINYINT	Weight percent	10
is_preset	TINYINT(1)	1 if preset	0
preset_name	VARCHAR(40)	Preset label if used	“HOLIDAY”
created_at	DATETIME	Created time	2026-01-28 10:05:00
SITE_SCORE			
Field Name	**Data Type**	**Description**	**Example**
site_score_id	BIGINT	Unique score ID	910881
profile_id	BIGINT	Business profile used	200301
weight_config_id	BIGINT	Weights used	90012
score_target_type	VARCHAR(16)	What was scored	PIN
area_id	BIGINT	Derived planning area	30018
lat	DECIMAL(9,6)	Scored latitude	1.35041
lng	DECIMAL(9,6)	Scored longitude	103.84831
composite_score	DECIMAL(6,2)	Overall score 0 to 100	78.3
demographic_score	DECIMAL(6,2)	Demographic dimension score	81.2
accessibility_score	DECIMAL(6,2)	Accessibility dimension score	84.1
rental_pressure_score	DECIMAL(6,2)	Rental pressure dimension score	62

competition_score	DECIMAL(6,2)	Competition dimension score	70.5
computed_at	DATETIME	When computed	2026-01-28 11:30:12
inputs_snapshot_json	JSON	Optional inputs snapshot	{"nearest_mrt_m":210,"vacancy_rate":6.4}
COMPETITION_SNAPSHOT (optional but useful)			
Field Name	**Data Type**	**Description**	**Example**
comp_id	BIGINT	Unique competition snapshot ID	920010
site_score_id	BIGINT	Linked site score	910881
radius_meters	INT	Search radius	500
competitor_count	INT	Competitors found	27
competitor_types_json	JSON	Types searched or returned	["restaurant","gym","school"]
source	VARCHAR(32)	Data source	GOOGLE_PLACES
effective_date	DATE	Optional effective date	2026-01-28
queried_at	DATETIME	Query time	2026-01-28 11:30:10
CANDIDATE_SITE			
Field Name	**Data Type**	**Description**	**Example**
candidate_site_id	BIGINT	Unique saved site ID	930501
profile_id	BIGINT	Business profile owner	200301
site_name	VARCHAR(80)	User defined name	“Bishan Junction 8”
address_label	VARCHAR(160)	Display address	“9 Bishan Pl, Singapore”
notes	TEXT	User notes	“High footfall, rent risk”
lat	DECIMAL(9,6)	Latitude	1.3502
lng	DECIMAL(9,6)	Longitude	103.848

area_id	BIGINT	Derived planning area	30018
saved_site_score_id	BIGINT	Score snapshot at save time	910881
saved_at	DATETIME	Saved time	2026-01-28 11:45:00
last_updated_at	DATETIME	Last edit time	2026-01-28 12:10:00
EXPLANATION			
Field Name	**Data Type**	**Description**	**Example**
explanation_id	BIGINT	Unique explanation ID	940020
site_score_id	BIGINT	Linked score	910881
dimension_code	VARCHAR(16)	Which dimension	ACCESS
message_text	TEXT	Human friendly explanation	“Nearest MRT is 210m, strong walk in access.”
score_impact	DECIMAL(6,2)	Optional highlighted impact	6.5
referenced_data_json	JSON	Data points cited	{"nearest_mrt_m":210,"exits_500m":6}
generated_at	DATETIME	Generated time	2026-01-28 11:30:13
EXPLANATION_FEEDBACK			
Field Name	**Data Type**	**Description**	**Example**
feedback_id	BIGINT	Unique feedback ID	950088
explanation_id	BIGINT	Explanation being rated	940020
user_id	BIGINT	Who rated it	100025
rating	TINYINT	1 to 5 rating	4
comment	VARCHAR(500)	Optional comment	“Useful, but mention bus stops too.”
created_at	DATETIME	Feedback time	2026-01-28 11:55:00

DATA_REFRESH_JOB			
Field Name	**Data Type**	**Description**	**Example**
job_id	BIGINT	Unique job ID	970012
provider_name	VARCHAR(40)	Data provider	LTA_DATAMALL
job_type	VARCHAR(32)	What it refreshes	INGEST_TRANSPORT
schedule_label	VARCHAR(40)	Manual or cron label	MANUAL
status	VARCHAR(16)	Job status	SUCCESS
started_at	DATETIME	Start time	2026-01-20 2:00:00
completed_at	DATETIME	End time	2026-01-20 2:08:40
rows_ingested	INT	Rows stored	23450
error_summary	TEXT	Error details if failed	“HTTP 429 rate limited”

🤖 AI Critique - Use Case Model & Requirements

PDF Report on the AI's Critique of the Use Case Diagram and Requirements

By: Ethan, Yu Hui, Marcus

UC01RegisterAccount Critique:

“Send Verification Email” as a separate use case is only justified if it is reused by multiple base use cases (eg resend verification, change email). If not reused, keep it as internal behaviour in the UC spec.

Email Service actor is OK as a supporting external system, but you should use a normal association (solid line) to the use case, not a dependency arrow from the base use case.

UC02VerifyEmail Critique:

Validate Token and **Activate Account** are internal steps, not great as separate use cases unless reused. Use case diagrams are for user goals, not step decomposition.

The note says “Failed tokens offer a resend option” but the diagram has **no use case for resend** and no extend/include relationship to represent it.

UC06CreateBusinessProfile Critique:
Over-granular use cases: “Enter profile name”, “Select sector”, “Select price band” are **steps in a flow**, not user goal level use cases. In a use case diagram, these usually should not be separate use cases.

UI behaviour modelled as use cases: “Highlight missing fields” and “Show confirmation” are UI responses, better captured in alternate flows or post conditions.

Extend relationship is backwards in intent:

- If you keep “Highlight missing fields”, it should extend the *main* use case (Create Business Profile) under a validation failure condition, not extend “Validate required fields” in isolation.

Missing include: the note says “persist the profile before confirming”, but confirmation is included separately. Usually **Save profile** is included, and “Show confirmation” is a post condition or step after save succeeds.

Prompt:

Critique the attached use case diagram image with respect to how well it represents the high-level system functionality described in the attached requirements file, specifically the map exploration features.

The diagram is intentionally simplified to provide an overview rather than exhaustively modeling every functional requirement. Please evaluate whether the abstraction level is appropriate, whether any essential use cases are missing, and whether the relationships and actors are modeled clearly according to UML conventions.

Original Response:

The provided use case diagram for **SmartLocate SG** offers a solid high-level abstraction of the map exploration workflow, but there are opportunities to better align it with the functional requirements and UML best practices.

Below is a critique of the diagram based on the **Functional Requirements (FR)** and standard modeling conventions.

---

1. Actor Representation & Hierarchy

The diagram correctly identifies that **Small Business Owners (A01)**, **Commercial Agents (A03)**, and **Franchise Teams (A02)** share identical interactions with the map.

- **Strengths:** Using a generalized "Business User" actor is an excellent way to maintain a clean diagram.
- **Improvement:** While appropriate for this abstraction level, the requirements also mention a **System Administrator (A04)** and **External Data Provider (A05)**. While they don't interact with the *Map Exploration* front-end, adding a note or a separate boundary for "Data Ingestion" (FR-DMI) would clarify how the map gets its data.

2. Core Use Case Alignment (Map Exploration)

The diagram covers the basic "Select and Score" flow, but misses a few "essential" high-level features defined in Section 4 of the requirements.

- **Missing: Map Layer/Choropleth Management:** Requirements FR-MAP-002, 003, and 007 emphasize the ability to toggle overlays (Demographics, Accessibility, Rental). This is a primary functional differentiator of the system and should be represented as a use case like `Manage Map Overlays`.
- **Missing: Planning Area Details:** FR-MAP-010 describes viewing district-level demographics. The current diagram jumps straight to "Selecting a Site" (POI), ignoring the "Area Exploration" aspect of the tool.

- **Missing: Scenario Planning:** A major functional block (Section 6) involves applying scenarios (e.g., Pandemic Focus) to change how the map looks and scores. This is a high-level user action that significantly impacts the map interface.

3. Relationship Logic & UML Conventions

The use of `«include»` and `«extend»` relationships requires some refinement to accurately reflect the system logic:

- **Select Site on Map → Calculate Site Scores:** Currently modeled as `«include»`. This is **accurate** because, per FR-MAP-009, selecting or pinning a site automatically triggers the score breakdown display.
- **Explain Score → Calculate Site Scores:** Modeled as `«extend»`. This is **accurate** as the explanation is an optional deeper dive following the calculation.
- **Search Location → Select Site on Map:** Modeled as `«include»`. This is slightly misleading. A user can select a site by clicking the map directly (FR-MAP-008) *without* searching. It would be more accurate to have `Search Location` and `Click Map` as two separate ways to reach the `Select Site` state.
- **Extend Relationships to "View Interactive Map":** The dashed arrows pointing from `Search Location` and `Select Site` to `View Interactive Map` are technically inverted. Usually, the base use case is the "Goal." If viewing the map is the base, then searching "extends" the map's functionality.

Selected Quote:

Missing: Scenario Planning: A major functional block (Section 6) involves applying scenarios (e.g., Pandemic Focus) to change how the map looks and scores. This is a high-level user action that significantly impacts the map interface.

Justification:

Although scenario planning is part of the functional requirements, it is intentionally omitted from the diagram as it was designed to present a simplified, high-level overview of core map exploration interactions rather than to represent every functional module in the system. Including scenario planning would have increased diagram complexity and reduced clarity for its intended purpose to provide an overview of the feature

UI Mockups (Images)

UI Mockups

By: Jung Kit

Prototype Link: https://minimal-text-site--summerchoojk01.replit.app/

Screenshots uploaded onto the “UI Mockups” Folder

Tab 10

Notes

Engine or api should be another actor

More use case model regarding log in, forget password ect.

Prepare for implementation

More detail