

**NANYANG  
TECHNOLOGICAL  
UNIVERSITY**  

---

**SINGAPORE**

# SC2006 - Software Engineering

## Lab 1 Deliverables

Lab Group: SCSD

Team: FeedEmGreens

Members: Rushaidy, Rajath, Hanzhi, Girija, Yohesh

# **Table of Contents**

<b>Table of Contents.....</b>	<b>1</b>
<b>1. Project Mission Statement.....</b>	<b>1</b>
<b>2. Documentation of functional and non-functional requirements.....</b>	<b>2</b>
A. Functional Requirements.....	2
B. Non-Functional Requirements.....	10
1. Performance.....	10
2. Scalability.....	10
3. Usability.....	10
4. Reliability & Availability.....	11
5. Security.....	11
6. Data Accuracy.....	11
7. Legal & Compliance.....	11
8. Gamification & Incentives.....	11
9. Supportability.....	11
<b>3. Data Dictionary.....</b>	<b>11</b>
<b>4. Use Case Diagram.....</b>	<b>14</b>
A. Use Case Descriptions.....	14
<b>5. UI Prototype.....</b>	<b>26</b>

# **1. Project Mission Statement**

FeedEmGreens' intention is to develop an application which will help ease health-oriented individuals or people with dietary restrictions to locate and direct them towards cheap, hygienic and healthy eateries based on their personalization.

## **2. Documentation of functional and non-functional requirements**

### **A. Functional Requirements**

1. The system shall perform user authentication to allow Users to use the application.
  - 1.1. The system shall allow Users to create an account
    - 1.1.1. Users shall enter at minimum: username, email, and password.
    - 1.1.2. Additional optional fields shall include: mobile number, diet preferences, and profile picture.
    - 1.1.3. An account created shall be tagged with the "User" role.
  - 1.2. The system shall allow Administrators to create an account with Admin privileges.
    - 1.2.1. Administrators shall enter username, email, and password.
    - 1.2.2. The "Admin" role shall be tagged to the account upon creation.
  - 1.3. The system shall allow Users to sign in with the account they created.
    - 1.3.1. Users shall enter their username and password to log into the application
    - 1.3.2. The system shall mask passwords with asterisks unless the unhide option is selected.
    - 1.3.3. If the username or password is incorrect, the system shall display: *"The username or password is incorrect. Please try again."*

- 1.3.4. After 3 unsuccessful attempts within 5 minutes, the system shall lock the account for 30 minutes.
- 1.4. The system shall allow Users to log in or sign up with the Singpass application
  - 1.4.1. When Users select Singpass, the system shall request consent to retrieve permitted profile information via the Singpass API.
  - 1.4.2. If the request is granted, the application will retrieve, and store relevant User profile data as permitted by the User and in compliance with Singpass' privacy policy.
- 2. The system shall allow Users to discover the location of healthier eateries
  - 2.1. The system shall display healthier hawker centres and stalls on login.
    - 2.1.1. The system shall display results in both list and map views.
    - 2.1.2. The system shall allow Users to search by stall name, hawker centre name, or location keyword.
    - 2.1.3. The system shall display for each result: name, address (with map pin), food categories, operating hours, and price indicator.
    - 2.1.4. If any information is missing, the system shall display "Unavailable" for that field.
    - 2.1.5. In the list form, the system shall sort search results according to distance.
  - 2.2. The system shall allow Users to sort results.
    - 2.2.1. Sorting options shall include distance, average price, and popularity.
  - 2.3. The system shall allow Users to filter results via the use of checkboxes
    - 2.3.1. Users can tick or untick checkboxes to filter eateries based on dietary tags, maximum budget, distance from current or chosen location and healthy options
    - 2.3.2. The system shall output filtered results in a form of lists sorted according to distance, price or popularity

- 2.3.3. If Users opt to view filtered results in the map, the system shall display every output pinned on the map within the distance of the User or from a chosen location.
- 3. The system shall provide queue and crowd estimation for Users.
  - 3.1. The system shall provide real-time or historical estimates of the queue at each eatery
    - 3.1.1. The system shall display a **queue indicator** for the selected eatery
    - 3.1.2. The queue indicator shall display estimated waiting time for each stall.
    - 3.1.3. The system shall display the timestamp of the most recent queue update.
    - 3.1.4. If real-time queue data is unavailable, the system shall estimate waiting time using historical patterns.
  - 3.2. The system shall provide real-time or historical estimate of crowd levels at each eatery
    - 3.2.1. The system shall display a **crowd indicator** for the area around the eatery
    - 3.2.2. The crowd indicator shall classify levels as Low (< 5 mins), Medium (5-15 mins) , or High (>15 mins).
    - 3.2.3. If real-time crowd data is unavailable, the system shall display typical crowd levels based on historical data.
  - 3.3. The system shall enhance queue and crowd estimation using contextual factors.
    - 3.3.1. The system shall use time of day to refine estimation (e.g., peak lunch and dinner hours).
    - 3.3.2. The system shall use day of week to refine estimation (e.g., weekends vs weekdays).
    - 3.3.3. The system shall use weather conditions (e.g., rainy vs clear) to adjust estimates of crowd levels.
    - 3.3.4. The system shall combine contextual factors with real-time or historical data to refine queue and crowd estimates

- 3.4. The system shall handle unavailable data.
  - 3.4.1. If neither real-time nor historical data is available, the system shall display “Queue and crowd data unavailable.”
- 4. The system can direct Users to the location of their desired eatery.
  - 4.1. The system shall provide a “Get Directions” option on each hawker centre detail page
    - 4.1.1. The system shall generate optimal routes to the selected hawker centre or stall using the OneMap API.
    - 4.1.2. The system shall request location access to detect the User’s current location for route calculation.
    - 4.1.3. If location access is denied by the User, the system shall allow the User to manually enter a starting point.
    - 4.1.4. The system shall display multiple modes of transportation supported by OneMap (e.g., walking, bus, train, driving).
    - 4.1.5. For each mode, the system shall display estimated travel time and distance.
    - 4.1.6. If the OneMap API is unavailable, the system shall display the notification: “**Directions are not available at this time.**”
- 5. The system shall suggest food options based on user history and preferences.
  - 5.1. The system shall display a recommendations popup to authenticated Consumers only.
    - 5.1.1. The system shall display the popup on the Home screen after a successful login.
    - 5.1.2. The system shall not display the popup to non-authenticated Users.
    - 5.1.3. If location access is denied and a distance filter is active, the system shall not display the popup.
    - 5.1.4. The system shall display at most one popup per Consumer per day.
  - 5.2. The system shall use the Consumer’s explicit constraints as hard filters.
    - 5.2.1. The system shall respect the maximum price cap.
    - 5.2.2. The system shall respect dietary constraints (e.g., Halal, vegetarian, vegan).
    - 5.2.3. The system shall respect distance constraints from the current or chosen location.

- 5.3. The system shall generate a recommendation candidate only if stall data meets all constraint requirements.
  - 5.3.1. The system shall exclude any stall that does not have valid distance/location data.
  - 5.3.2. The system shall exclude any stall that does not have valid dietary tag data.
  - 5.3.3. The system shall exclude any stall that does not have valid price indicator data.
  - 5.3.4. The system shall exclude any stall that is not currently open.
- 5.4. The system shall present exactly one recommendation in the popup.
  - 5.4.1. The popup shall display stall name, distance (or ETA), and a price indicator (if available).
  - 5.4.2. The popup shall display applicable badges (e.g., “Under \$5”, “Halal”, “Open now”).
  - 5.4.3. The popup shall display a one-line reason (e.g., “Under \$5 • Near you” or “Halal option • Open now”).
  - 5.4.4. The popup shall provide a **Dismiss** button.
- 5.5. The system shall allow Users to dismiss a recommendation.
  - 5.5.1. Selecting Dismiss shall hide the popup for the remainder of the day.
  - 5.5.2. After Dismiss, the popup shall not reappear until the next day.
- 6. The system enables users to share feedback and see aggregated ratings.
  - 6.1. The system shall restrict review and rating actions to authenticated Users only.
    - 6.1.1. The system shall not allow non-authenticated Users to submit, edit, or delete reviews.
    - 6.1.2. The system shall prompt non-authenticated Users to log in when attempting a review action.
  - 6.2. The system shall allow a User to submit one active review per stall.
    - 6.2.1. A review shall include a **Health Score** (1–5 scale, integers only).
    - 6.2.2. A review shall include a **Hygiene Score** (1–5 scale, integers only).
    - 6.2.3. A review may include optional text feedback.
    - 6.2.4. A review may include up to 3 optional photos.
    - 6.2.5. If the User has an existing review for the stall, the system shall open the existing review for editing instead of creating a new one.
    - 6.2.6. On successful submission, the system shall confirm with “Review posted.”
  - 6.3. The system shall allow a User to edit their own review.
    - 6.3.1. The system shall allow editing of Health Score, Hygiene Score, text, and photos.

- 6.3.2. On successful update, the system shall confirm with “Review updated.”
- 6.4. The system shall allow a User to delete their own review.
  - 6.4.1. The system shall display a confirmation prompt before deletion.
  - 6.4.2. On deletion, the system shall remove the review and confirm with “Review deleted.”
- 6.5. The system shall allow Users to flag a review as inappropriate.
  - 6.5.1. A flag shall include a selected reason (e.g., spam, offensive, false information).
  - 6.5.2. On successful flag submission, the system shall confirm with “Review reported.”
  - 6.5.3. The system shall add flagged reviews to the Admin moderation queue.
- 6.6. The system shall display reviews on the stall details page.
  - 6.6.1. Each review shall show Health Score, Hygiene Score, text (if any), photos (if any), author alias, and timestamp.
  - 6.6.2. The system shall label the User’s own review as “Your review.”
  - 6.6.3. The system shall sort reviews by Most Recent by default.
  - 6.6.4. The system shall allow sorting by Highest Health Score and Highest Hygiene Score.
  - 6.6.5. If no reviews exist, the system shall display “No reviews yet.”
- 6.7. The system shall display aggregated scores for each stall.
  - 6.7.1. The system shall display the **average Health Score** (to one decimal place).
  - 6.7.2. The system shall display the **average Hygiene Score** (to one decimal place).
  - 6.7.3. The system shall display the total number of reviews used in each average.
  - 6.7.4. The system shall update averages immediately after a review is added, edited, or deleted.
- 6.8. The system shall prevent review spam and duplicates.
  - 6.8.1. The system shall restrict each User to one active review per stall.
  - 6.8.2. The system shall enforce a minimum interval of 7 days between new review submissions for the same stall by the same User (edits are allowed anytime).
- 6.9. The system shall handle validation and error states for review actions.
  - 6.9.1. If required fields (Health Score, Hygiene Score) are missing, the system shall display an inline validation message and prevent submission.



- 6.9.2. If photo upload fails, the system shall display “Photo upload failed” and allow resubmission without photos.
    - 6.9.3. If a network error occurs, the system shall display “Unable to submit review. Please try again.”
  - 6.10. The system shall expose review management actions via clear UI controls.
    - 6.10.1. The system shall provide **Write review**, **Edit**, **Delete**, and **Report** buttons where applicable.
    - 6.10.2. The system shall disable **Edit** and **Delete** for reviews not authored by the current User.
  - 6.11. The system shall ensure consistency between review list and details.
    - 6.11.1. After adding, editing, or deleting a review, the system shall refresh the review list and aggregated Health/Hygiene scores shown on the stall details page.
- 7. The system shall allow users to redeem rewards for using the application.
  - 7.1. The system shall award points to authenticated Users for specific actions.
    - 7.1.1. The system shall award points when a User purchases or selects a meal tagged as “Healthy.”
    - 7.1.2. The system shall award points when a User visits a new stall for the first time.
    - 7.1.3. The system shall award points when a User submits a Health & Hygiene review.
    - 7.1.4. The system shall prevent awarding duplicate points for the same action within a 24-hour period.
  - 7.2. The system shall display the User’s points balance.
    - 7.2.1. The points balance shall be displayed on the User’s profile page.
    - 7.2.2. The points balance shall show the current total and the date of last update.
    - 7.2.3. The system shall provide a history view of the last 5 point-earning actions.
  - 7.3. The system shall allow Users to redeem points for rewards.
    - 7.3.1. The system shall display a list of available rewards and their required point values.
    - 7.3.2. The system shall allow a User to confirm redemption with a “Redeem” button.
    - 7.3.3. On successful redemption, the system shall deduct the corresponding points from the User’s balance.
    - 7.3.4. The system shall confirm redemption with a message “Reward redeemed successfully.”
  - 7.4. The system shall handle insufficient points for redemption.

- 7.4.1. If a User attempts redemption without enough points, the system shall display “Not enough points.”
  - 7.4.2. The system shall disable the “Redeem” button for rewards where the User’s points are below the required threshold.
- 7.5. The system shall define reward expiration.
  - 7.5.1. Each redeemed reward shall display an expiry date, if applicable.
  - 7.5.2. The system shall prevent use of rewards after their expiry date.
- 7.6. The system shall ensure fairness in point awarding.
  - 7.6.1. The system shall not allow manual modification of points by Users.
  - 7.6.2. The system shall log all point transactions (award, redemption, expiry) for audit.
- 8. The system shall allow authenticated Admins to manage content quality and verify sensitive information.
  - 8.1. The system shall restrict moderation actions to authenticated Admins only.
    - 8.1.1. The system shall not allow non-Admin Users to access moderation functions.
    - 8.1.2. The system shall prompt login if a non-authenticated User attempts to access moderation.
  - 8.2. The system shall provide an Admin dashboard.
    - 8.2.1. The dashboard shall display flagged reviews requiring moderation.
    - 8.2.2. The dashboard shall display reported hygiene/health reviews requiring moderation.
    - 8.2.3. The dashboard shall provide access to dietary tag management for stalls.
    - 8.2.4. The dashboard shall display support tickets submitted by Users.
  - 8.3. The system shall allow Admins to act on flagged reviews.
    - 8.3.1. The system shall allow Admins to approve a flagged review as valid.
    - 8.3.2. The system shall allow Admins to hide a review containing offensive or inappropriate content.
    - 8.3.3. The system shall allow Admins to delete a review entirely.
    - 8.3.4. The system shall require Admins to provide a reason for any removal.
    - 8.3.5. The system shall notify the User who authored the review of the moderation outcome.
  - 8.4. The system shall allow Admins to modify dietary tags in the database.
    - 8.4.1. The system shall allow Admins to add a dietary tag (e.g., Halal, Vegetarian, Healthy) to a stall.
    - 8.4.2. The system shall allow Admins to remove a dietary tag from a stall.

- 8.4.3. The system shall allow Admins to edit existing dietary tags for accuracy.
- 8.4.4. Changes to dietary tags shall be updated in the User-facing app immediately after Admin confirmation.
- 8.5. The system shall allow Admins to manage User accounts.
  - 8.5.1. The system shall allow Admins to suspend a User account for repeated violations.
  - 8.5.2. The system shall require Admins to enter the reason and duration for suspension.
  - 8.5.3. The system shall notify the User of the suspension reason and duration.
- 8.6. The system shall ensure all Admin actions are logged.
  - 8.6.1. Each moderation action shall log: Admin ID, action type, target (review/tag/user), timestamp, and outcome.
  - 8.6.2. Logs shall be viewable only by Admins and system auditors.
- 8.7. The system shall handle error and edge cases.
  - 8.7.1. If a moderation action fails due to network or database error, the system shall display “Action could not be completed. Please try again.”
  - 8.7.2. If an Admin attempts to act on a review or tag that has already been resolved, the system shall display “This item has already been moderated.”

## B. Non-Functional Requirements

### 1. Performance

- The system shall load search results and recommendations within 2 seconds under normal network conditions.
- The system shall handle at least 500 concurrent users with no noticeable degradation in response time.
- Queue and crowd estimation data shall be updated at least every 5 minutes.
- Application startup time shall not exceed **5 seconds** under normal conditions.

### 2. Scalability

- The system shall support scaling to 10,000 registered users without requiring major architectural changes.

- The system design shall allow easy integration of additional hawker centres and stalls without downtime.

### **3. Usability**

- The system shall allow a User to locate a stall (search + filters) within 3 interactions or fewer
- The system shall provide an intuitive and mobile-friendly interface, with at least 90% of test users rating usability  $\geq 4/5$  during usability testing.
- The system shall ensure all map pins are displayed within **5 seconds** after search.
- The system shall allow smooth map zoom and pan with a delay of less than **2 seconds**.
- The system shall support switching between at least **two languages** (e.g., English and Chinese) in the User Interface
- The system design shall be extensible to support additional languages (Malay, Tamil) without major redesign.
- Users should have the ability to view, access, edit, share, or opt out of sharing specific aspects of their data.

### **4. Reliability & Availability**

- The system shall achieve 99.5% uptime per month, excluding scheduled maintenance.
- The system shall ensure no more than 1 minute of downtime per week due to unexpected failures.
- If the map API is unavailable, the system shall display a fallback message (e.g. "Map currently not available") within 2 seconds.

### **5. Security**

- All user credentials shall be stored securely using hashed and salted passwords.
- The system shall enforce multi-factor authentication (MFA) for admin  
The system shall request explicit User consent before accessing location data.
- Only authorised Admin accounts shall have access to moderation and management features.

### **6. Data Accuracy**

- Queue length and wait time predictions shall achieve  $\geq 85\%$  accuracy based on historical and live data.
- Nutritional information and stall tags (e.g., "Healthier Choice") shall be verified quarterly to ensure data reliability.

### **7. Legal & Compliance**

- The system shall comply with Singapore's PDPA for handling personal data.

- All geolocation data shall be anonymized when stored or analyzed for recommendations.
- 8. Gamification & Incentives**
- Points and rewards shall be processed within 2 seconds of meal logging or redemption.
  - Reward redemption systems shall prevent duplicate or fraudulent claims through transaction logging and validation.
- 9. Supportability**
- The database must be replaceable with any commercial product supporting standard SQL queries.

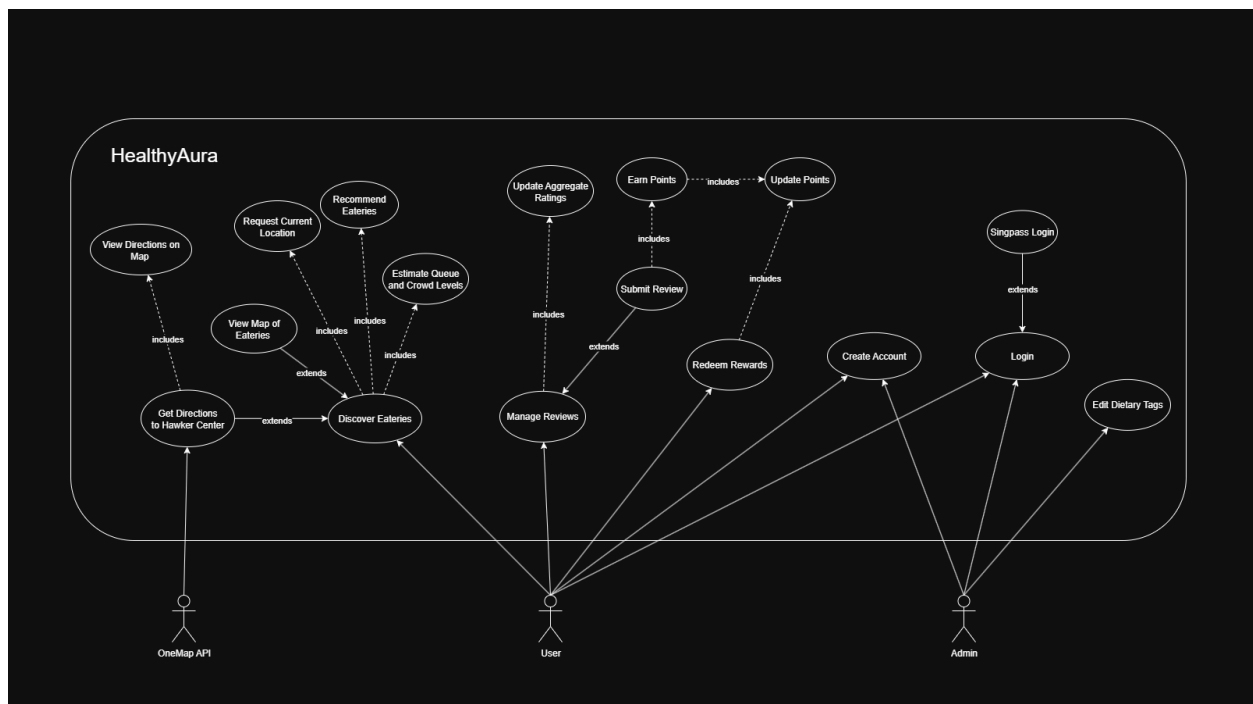
### **3. Data Dictionary**

<b>TERM</b>	<b>DEFINITION</b>
Account	A registered User profile containing login credentials and optional personal data (e.g., dietary preferences, profile picture, review history).
Admin	A User with special privileges for moderation, managing reviews, editing dietary tags, and resolving support tickets.
Application (App)	The mobile software application that enables Users to discover healthier food options in Singapore hawker centres and stalls
Consumer	A person that uses APPNAME to view Hawker centre's review, search for healthier food places, and perform consumer-specific tasks.
Crowd Indicator	A measure of busyness in the area around a hawker centre, classified as Low, Medium, or High
Dietary preference	A User's chosen food requirements (e.g., Halal, vegetarian, healthier choice) that the system uses to filter and personalise

	recommendations.
Dietary Tag	A label applied to a stall to indicate suitability for specific diets (e.g., Halal, vegetarian, vegan, healthier choice). Dietary tags are managed by Admins
Food place / Stall	Any hawker centre or stall displayed in the app with location, dietary tags, and hygiene rating.
Hawker centre	A food court in Singapore containing multiple stalls.
Healthier choice	A certification under Health Promotion Board (HPB) guidelines for healthier meals.
Hygiene rating	Cleanliness grading from user reviews and displayed in the app to help users choose cleaner and safer food outlets.
Map	An interactive feature that displays hawker centres and stalls, filtered by distance, dietary tags, and hygiene rating.
Points	A numerical value earned by Users for completing defined actions (e.g., healthy food choice, review submission), which can be redeemed for reward
Queue Indicator	An estimate of waiting time at a specific stall, shown to the User
Recommendation	A personalised suggestion for a stall or hawker centre, shown to the User based on dietary preferences, distance, and health criteria
Review	User-generated feedback on food quality, hygiene, and service at a food place, consisting of Health and Hygiene scores, optional text, and optional photos.
Search	A User-submitted request for assistance, tracked with unique ID and status
Support Ticket	A record created when a User submits an

	issue, complaint, or inquiry in the application, tracked by ID and status until resolved by an Admin.
User	<p>An individual who has created an account in the application. A User may hold one of the following roles:</p> <ul style="list-style-type: none"> <li>• <b>Admin</b> – has privileges to manage reviews, dietary tags, support tickets, and User accounts.</li> <li>• <b>Standard User</b> – can search, filter, view, and review eateries, as well as receive recommendations.</li> </ul>

## 4. Use Case Diagram



## A. Use Case Descriptions

### 1. Use Case 1 - Create Account

Use Case ID:	UC-1		
Use Case Name:	Create Account		
Created By:	Rushaidy	Last Updated By:	Rushaidy
Date Created:	30th August 2025	Date Last Updated:	2nd September 2025

Actor:	<ul style="list-style-type: none"><li>• User (primary)</li><li>• Admin (for admin account creation)</li></ul>
Description:	This use case allows a new User / Admin to register an account in the application with required and optional details.
Flow of Events:	<ol style="list-style-type: none"><li>1. The User selects Sign Up.</li><li>2. The system prompts for required fields: username, email, password.</li><li>3. The User enters the required information.</li><li>4. The system prompts for optional fields: mobile number, diet preferences, profile picture.</li><li>5. The User may enter optional information.</li><li>6. The system validates uniqueness of username/email and password complexity.</li><li>7. If valid, the system creates a new account tagged with "User" role for Users and "Admin" roles for admin.</li><li>8. The system confirms successful account creation to the User / Admin.</li></ol>
Includes:	None



## 2. Use Case 2 - Login

Use Case ID:	UC-2		
Use Case Name:	Login		
Created By:	Rushaidy	Last Updated By:	Rushaidy
Date Created:	30th August 2025	Date Last Updated:	2nd September 2025

Actor:	User or Admin
Description:	This use case authenticates user / admin with username and password.
Flow of Events:	<ol style="list-style-type: none"><li>1. User / Admin selects <b>Login</b>.</li><li>2. System prompts for username/email and password.</li><li>3. User / Admin enters credentials.</li><li>4. System masks password by default.</li><li>5. User may unhide password.</li><li>6. System validates credentials.</li><li>7. If valid, grants access.</li></ol>
Includes:	<ul style="list-style-type: none"><li>• UC2.E1 - Singpass Login</li></ul>

### 2.1. Use Case 2.E1 - Singpass Login (<<extend>> of Login)

Use Case ID:	UC-2.E1		
Use Case Name:	Singpass Login		
Created By:	Rushaidy	Last Updated By:	Rushaidy
Date Created:	30th August 2025	Date Last Updated:	2nd September 2025

Actor:	<ul style="list-style-type: none"><li>• User or Admin</li><li>• Singpass API</li></ul>
Description:	This use case allows User to sign up or log in with Singpass.
Flow of Events:	<ol style="list-style-type: none"><li>1. User selects "Login/Signup with Singpass."</li><li>2. User grants consent.</li><li>3. Singpass API returns permitted profile data.</li><li>4. System creates/updates account.</li><li>5. User authenticated.</li></ol>
Includes:	None

### 3. Use Case 3 - Discover Healthier Eateries

Use Case ID:	UC-3		
Use Case Name:	Discover Healthier Eateries		
Created By:	Rajath Krishna	Last Updated By:	Rajath Krishna
Date Created:	30th August 2025	Date Last Updated:	2nd September 2025

Actor:	<ul style="list-style-type: none"><li>• User (primary)</li><li>• Device Location Services (supporting)</li></ul>
Description:	The User searches for and views a list of healthier hawker centres or stalls. The System presents results with basic details and optional map view.
Flow of Events:	<ol style="list-style-type: none"><li>1. User opens the Discover screen after login.</li><li>2. User is prompted to give current location using included use case <b>UC-3.I1 Request Current Location</b></li><li>3. System displays a list of eateries with basic details (name, address, categories, hours, price indicator).</li><li>4. User enters a search keyword (e.g., stall name, hawker centre, or location).</li><li>5. System updates the list with matching results.</li><li>6. User optionally switches to <b>Map View</b> to see eateries on a map.</li><li>7. User selects an eatery → System shows details and includes <b>UC-4 Estimate Queue and Crowd Levels</b>.</li></ol>
Includes:	<ul style="list-style-type: none"><li>• UC-3.I1 Request Current Location</li><li>• UC-4 Estimate Queue and Crowd Levels</li><li>• UC-6 Recommend Eateries</li></ul>
Extends:	<ul style="list-style-type: none"><li>• UC-3.I2 View Map of Eateries (optional, for map-based results).t Location</li></ul>

### 3.1. Use Case 3.11 - Request Current Location

Use Case ID:	UC-3.11		
Use Case Name:	Request Current Location		
Created By:	Rajath Krishna	Last Updated By:	Rajath Krishna
Date Created:	30th August 2025	Date Last Updated:	2nd September 2025

Actor:	<ul style="list-style-type: none"><li>• User</li><li>• Device Location Services</li></ul>
Description:	The System requests permission to access the User's current location to support location-based search and map features.
Flow of Events:	<ol style="list-style-type: none"><li>1. System prompts the User for location permission.</li><li>2. User accepts or denies the request.</li><li>3. If accepted, System retrieves coordinates and returns them to the parent use case..</li></ol>
Includes:	None

### 3.2. Use Case 3.E2 — View Map of Eateries (as <<extend>> of UC-3)

Use Case ID:	UC-3.E2		
Use Case Name:	View Map of Eateries		
Created By:	Rajath Krishna	Last Updated By:	Rajath Krishna
Date Created:	30th August 2025	Date Last Updated:	2nd September 2025

Actor:	<ul style="list-style-type: none"><li>• User</li><li>• System</li></ul>
Description:	The User switches from list view to map view. The System displays eateries as pins on a map
Flow of Events:	<ol style="list-style-type: none"><li>1. User selects Map View.</li><li>2. System loads the map interface.</li><li>3. System places pins for visible eateries.</li><li>4. User can pan/zoom to explore the map.</li><li>5. User taps a pin to open that eatery's details (returns to UC-3).</li></ol>
Includes:	None
Extends:	<ul style="list-style-type: none"><li>• UC-3 Discover Eateries</li></ul>

#### 4. Use Case 4 —Estimate Queue and Crowd Levels

Use Case ID:	UC-4		
Use Case Name:	Estimate Queue and Crowd Levels		
Created By:	Yoheshvaran	Last Updated By:	Yoheshvaran
Date Created:	30th August 2025	Date Last Updated:	2nd September 2025

Actor:	<ul style="list-style-type: none"><li>• User (primary)</li><li>• System (supporting)</li><li>• Data Provider APIs / Sensors (optional, supporting)</li></ul>
Description:	The User views estimated queue waiting times and crowd levels for a hawker centre or stall. Since live data is unavailable, the System generates estimates using predefined rules based on time, day, and weather.
Flow of Events:	<ol style="list-style-type: none"><li>1. User selects a hawker centre or stall from the list.</li><li>2. System checks contextual factors (time of day, day of week, weather).</li><li>3. System applies estimation rules (e.g., peak hours = higher crowd).</li><li>4. System generates:<ul style="list-style-type: none"><li>○ Estimated queue time for stall(s)</li><li>○ Overall crowd level (Low / Medium / High)</li></ul></li><li>5. System displays estimates with a label “Estimated based on patterns.”</li><li>6. Users will use the estimated time to decide whether to visit.</li></ol>
Includes:	None

## 5. Use Case 5 — Get Directions to Hawker Centre (as <<extend>> of UC-3)

Use Case ID:	UC-5		
Use Case Name:	Get Directions to Hawker Centre		
Created By:	Rajath Krishna	Last Updated By:	Rajath Krishna
Date Created:	30th August 2025	Date Last Updated:	2nd September 2025

Actor:	<ul style="list-style-type: none"> <li>• User</li> <li>• OneMap API (supporting system)</li> <li>• Device Location Services (supporting system, optional)</li> </ul>
Description:	This use case allows a User to request directions to a selected hawker centre or stall. The system determines the starting point using the device's location (with permission) or allows manual input. The OneMap API then returns optimal routes with multiple modes of transport, ETA and distance to the user.
Flow of Events:	<ol style="list-style-type: none"> <li>1. User selects <b>Get Directions</b> on the stall or hawker centre detail page.</li> <li>2. System prompts the User to enter a starting address or postal code.</li> <li>3. User enters a valid starting point.</li> <li>4. System calls the OneMap API to generate routes to the selected destination</li> <li>5. System displays available routes with mode of transport, travel time, and distance.</li> <li>6. User selects a route to view, and the System shows it on a map using the included use case <b>UC - 4.11 View Directions on Map</b>.</li> </ol>
Includes:	<ul style="list-style-type: none"> <li>• UC-5.11 View Directions on Map</li> </ul>
Relationship:	<ul style="list-style-type: none"> <li>• UC-5 extends UC-3 Discover Eateries</li> </ul>

### 5.1. Use Case 5.11 — View Directions on Map

Use Case ID:	UC-5		
Use Case Name:	Get Directions to Hawker Centre		
Created By:	Rajath Krishna	Last Updated By:	Rajath Krishna
Date Created:	30th August 2025	Date Last Updated:	2nd September 2025

Actor:	<ul style="list-style-type: none"><li>• User (primary)</li><li>• System (supporting)</li></ul>
Description:	After routes are generated in <b>Get Directions</b> , the System must display the selected route on a map.
Flow of Events:	<ol style="list-style-type: none"><li>1. UC-5 generates available routes.</li><li>2. User selects a route to view.</li><li>3. System opens the map with the route overlaid.</li><li>4. System shows key waypoints, travel time, and distance.</li><li>5. User can zoom/pan to inspect the path.</li></ol>
Includes:	None



## 6. Use Case 6 — Recommend Eateries

Use Case ID:	UC-6		
Use Case Name:	Recommend Eateries		
Created By:	Hanzhi	Last Updated By:	Hanzhi
Date Created:	31th August 2025	Date Last Updated:	2nd September 2025

Actor:	<ul style="list-style-type: none"><li>• User (primary)</li><li>• System (supporting)</li></ul>
Description:	While browsing healthier eateries, the System may suggest one stall or dish based on user preferences and patterns. This extends the base “Discover Eateries” use case.
Flow of Events:	<ol style="list-style-type: none"><li>1. UC-3 Discover Eateries is running.</li><li>2. System checks User’s dietary tags, budget, or past activity.</li><li>3. System generates a suitable recommendation.</li><li>4. System displays the suggestion (e.g., pop-up or highlight).</li><li>5. User may accept (view details) or ignore the suggestion.</li></ol>
Includes:	None

## 7. Use Case 7 — Manage Reviews

Use Case ID:	UC-7		
Use Case Name:	Manage Reviews		
Created By:	Hanzhi	Last Updated By:	Hanzhi
Date Created:	31th August 2025	Date Last Updated:	2nd September 2025

Actor:	<ul style="list-style-type: none"><li>• User</li><li>• Admin</li><li>• System</li></ul>
Description:	Allow users to submit, edit, delete, and flag reviews for stalls, while displaying aggregated ratings.
Flow of Events:	<ol style="list-style-type: none"><li>1. User opens a stall's details page.</li><li>2. System shows existing reviews and current aggregate rating.</li><li>3. User chooses one of: <b>Submit Review, Edit Review, Delete Review, or View Reviews.</b></li><li>4. System validates the chosen action.</li><li>5. System updates reviews and recalculates aggregates.</li><li>6. System confirms the outcome (posted/updated/deleted) and refreshes the list.</li></ol>
Includes:	<ul style="list-style-type: none"><li>• UC-7.I1 – Update Aggregate Ratings</li></ul>
Extends:	<ul style="list-style-type: none"><li>• UC-7.E1 – Submit Review</li><li>• UC-7.E2 – Edit Review</li><li>• UC-7.E3 – Delete Review</li></ul>

### 7.1. Use Case 7.I1 — Update Aggregate Ratings

Use Case ID:	UC-7.I1		
Use Case Name:	Update Aggregate Ratings		
Created By:	Hanzhi	Last Updated By:	Hanzhi
Date Created:	30th August 2025	Date Last Updated:	2nd September 2025

Actor:	System
Description:	System recalculates and updates stall's aggregate rating after changes.
Flow of Events:	<ol style="list-style-type: none"><li>1. System recomputes aggregate rating and review count.</li><li>2. System updates the stall details page.</li></ol>
Includes:	None

### 7.2. Use Case 7.E1 — Submit Review (as <<extend>> of UC-7)

Use Case ID:	UC-7.E1		
Use Case Name:	Submit Review		
Created By:	Hanzhi	Last Updated By:	Hanzhi
Date Created:	31th August 2025	Date Last Updated:	2nd September 2025

Actor:	<ul style="list-style-type: none"><li>• User</li><li>• System</li></ul>
Description:	The user submits a review for a stall with required scores (Health, Hygiene) and optional text/photos.
Flow of Events:	<ol style="list-style-type: none"><li>1. User selects <b>Submit Review</b>.</li><li>2. System prompts for required inputs (e.g., rating).</li><li>3. User provides required rating (and optionally adds text or a photo).</li><li>4. System validates the input.</li><li>5. On success, System saves the review and invokes the included use case <b>UC-7.I1 Update Aggregate Ratings</b></li><li>6. System shows "Review posted."</li></ol>
Includes:	<ul style="list-style-type: none"><li>• UC-7.I1 – Update Aggregate Ratings</li></ul>

### 7.3. Use Case 7.E2 — Edit Review (as <<extend>> of UC-7)

Use Case ID:	UC-7.E2		
Use Case Name:	Edit Review		
Created By:	Hanzhi	Last Updated By:	Hanzhi
Date Created:	31th August 2025	Date Last Updated:	2nd September 2025

Actor:	<ul style="list-style-type: none"><li>• User</li><li>• System</li></ul>
Description:	User updates their existing review.
Flow of Events:	<ol style="list-style-type: none"><li>1. User selects <b>Edit Review</b></li><li>2. System retrieves existing Review</li><li>3. User updates rating/text/photo.</li><li>4. System validates and saves changes.</li><li>5. System invokes <b>UC-7.I1 Update Aggregate Ratings</b> and shows “Review updated.”</li></ol>
Includes:	<ul style="list-style-type: none"><li>• UC-7.I1 – Update Aggregate Ratings</li></ul>

### 7.4. Use Case 7.E3 — Delete Review (as <<extend>> of UC-7)

Use Case ID:	UC-7.E3		
Use Case Name:	Delete Review		
Created By:	Hanzhi	Last Updated By:	Hanzhi
Date Created:	31th August 2025	Date Last Updated:	2nd September 2025

Actor:	<ul style="list-style-type: none"><li>• User</li><li>• System</li></ul>
Description:	User removes their review.
Flow of Events:	<ol style="list-style-type: none"><li>1. User selects <b>Delete Review</b> and confirms.</li><li>2. System deletes the review.</li><li>3. System invokes <b>UC-7.I1 Update Aggregate Ratings</b> and shows “Review deleted.”</li></ol>
Includes:	<ul style="list-style-type: none"><li>• UC-7.I1 – Update Aggregate Ratings</li></ul>

## 8. Use Case 8 — Edit Dietary Tags

Use Case ID:	UC-8		
Use Case Name:	Edit Dietary Tags		
Created By:	Girija	Last Updated By:	Girija
Date Created:	31th August 2025	Date Last Updated:	2nd September 2025

Actor:	<ul style="list-style-type: none"><li>• Admin (primary)</li><li>• System (supporting)</li></ul>
Description:	The Admin updates or corrects dietary tags (e.g., Halal, vegetarian, healthy choice) for a stall to maintain accuracy in search and recommendations.
Flow of Events:	<ol style="list-style-type: none"><li>1. Admin logs in with elevated privileges.</li><li>2. Admin selects a stall record.</li><li>3. Admin edits dietary tags (add/remove/update).</li><li>4. System saves the changes.</li><li>5. System updates the stall listing and recommendations that rely on these tags.</li></ol>
Includes:	None

## 9. Use Case 9 — Earn Points

Use Case ID:	UC-9		
Use Case Name:	Earn Points		
Created By:	Girija	Last Updated By:	Girija
Date Created:	31th August 2025	Date Last Updated:	2nd September 2025

Actor:	<ul style="list-style-type: none"><li>• Admin (primary)</li><li>• System (supporting)</li></ul>
Description:	The User earns points when performing eligible actions (e.g., submitting a review, and more to be added actions). The System updates the User's points balance.
Flow of Events:	<ol style="list-style-type: none"><li>1. An eligible action completes (e.g., <b>UC-7.E1 Submit Review</b>).</li><li>2. System determines the points to award.</li><li>3. System includes <b>UC-9.I1 Update Points</b> to add points to the User's balance.</li><li>4. System stores a simple points transaction record.</li><li>5. System optionally notifies the User (e.g., "+10 points earned").</li></ol>
Includes:	<ul style="list-style-type: none"><li>• UC-9.I1 Update Points (add to balance)</li></ul>
Extends:	<ul style="list-style-type: none"><li>• UC-7.E1 Submit Review (when a review is posted)</li></ul>

### 9.1. Use Case 9.I1 — Update Points

Use Case ID:	UC-9.I1		
Use Case Name:	Update Points		
Created By:	Girija	Last Updated By:	Girija
Date Created:	31th August 2025	Date Last Updated:	2nd September 2025

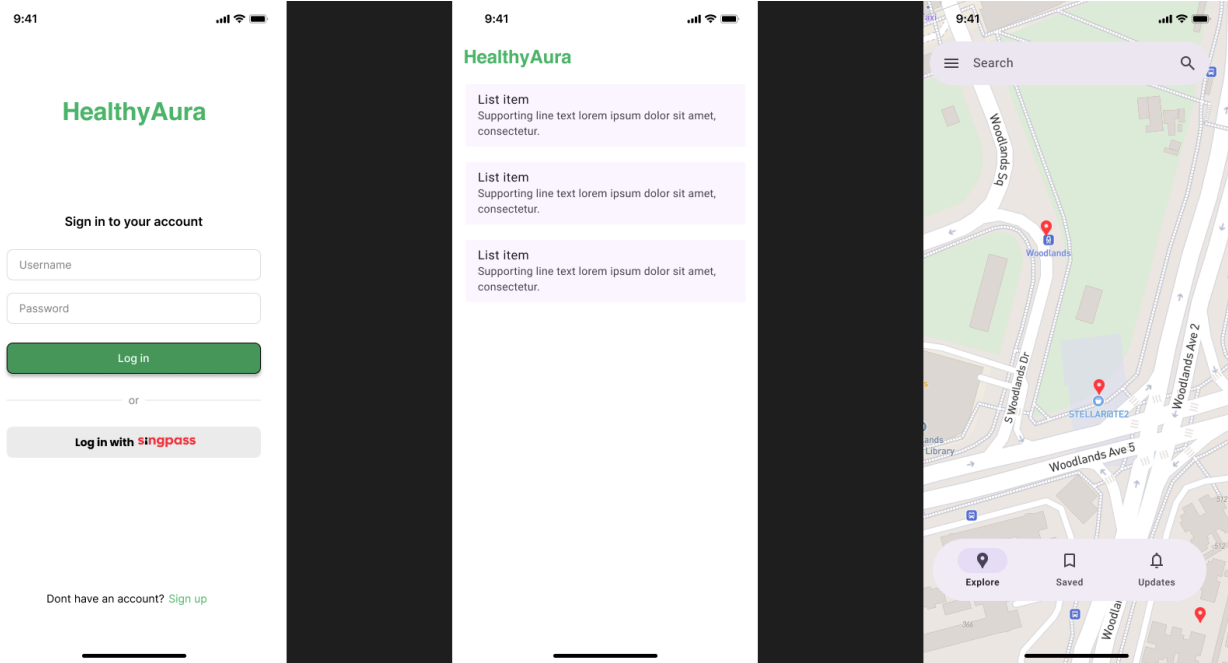
Actor:	<ul style="list-style-type: none"><li>• System</li></ul>
Description:	System updates the User's balance by either adding or deducting points, depending on the calling UC
Flow of Events:	<ol style="list-style-type: none"><li>1. System retrieves current balance.</li><li>2. System adds or subtracts points as specified.</li><li>3. System validates the new balance (rejects if negative).</li><li>4. System saves the updated balance and logs the transaction.</li></ol>
Includes:	None

### 10. Use Case 10 — Redeem Reward

Use Case ID:	UC-10		
Use Case Name:	Redeem Rewards		
Created By:	Yoheshvaran	Last Updated By:	Yoheshvaran
Date Created:	31th August 2025	Date Last Updated:	2nd September 2025

Actor:	<ul style="list-style-type: none"><li>• User (primary)</li><li>• System (supporting)</li></ul>
Description:	The User redeems accumulated points for discounts or vouchers. The System deducts points and issues the reward.
Flow of Events:	<ol style="list-style-type: none"><li>1. User opens the Rewards section.</li><li>2. System shows balance and available rewards.</li><li>3. User selects a reward.</li><li>4. System verifies sufficient points.</li><li>5. System includes <b>UC-9.I1 Update Points</b> to deduct points.</li><li>6. System issues the reward and updates balance.</li></ol>
Includes:	<ul style="list-style-type: none"><li>• UC-9.I1 Update Points (deduct from balance)</li></ul>

## 5. UI Prototype







Search

## Healthier Hawker Centres

21 results

### Tiong Bahru Hawker Centre

\$

30 Seng Poh Rd

Asian

Vegan

9.00 AM - 9.00 PM

### Amoy Street Food Centre

\$

7 Maxwell Rd

Asian

Vegetarian

6.00 AM - 9.00 PM

### S11 Curry

\$

9 Maxwell Rd, #01-19

Indian

Halal

10.00 AM - 2.00 PM

### Green Eats

\$

3 Maxwell Rd, #02-10

Vegetarian

Healthy



HealthyAura



Leave a Review

Green Eats



Your review

This place has great healthy food options!

Rating



Submit




Alex P.


This place has great healthy food options!





# HealthyAura


Search by stall name, hawker centre, or loc...

 Vegetarian



 High Protein

 Vegan

 Budget Friendly

 Healthy Options

Sort by Distance 

 Filter 



## Green Eats Hawker

123 Jurong East St., Singapore 60023

Vegetarian, Budget Friendly

 ~10 mins wait

[Get Directions](#)



## Healthy Bites

456 Bukit Merah Lane 1, Singapore 150456

Vegan, Gluten-Free

 ~5 mins wait

[Get Directions](#)



## NutriHub

789 Kallang Rd., Singapore 339328

Vegan, HALAL

# HealthyAura

← Crowd & Queue Status



## Green Bowl

123 Hawker Lane

Vegan

Halal

Healthy



### Queue Status

Estimated Wait: 10 mins

Last updated: 5 mins ago



### Crowd Status

Low (< 5 mins)

Low (< 5 mins)


Medium (5 - 15 mins)





## Recommended for You


Based on your preferences & nearby options

 Vegan

 Healthy

 Vegetarian

 High Protein

 Budget Friendly



### Tiong Bahru Market

30 Seng Poh Rd

Vegan

Healthy

Vegetarian



### ABC Food Centre

51 Ang Mo Kio Ave 3

Vegan

High Protein

Budget  
Friendly



### Jamie's Hawker Hub

3-55 Lor 4 Toa Payoh

Vegan

Healthy

Vegetarian