

# NANYANG TECHNOLOGICAL UNIVERSITY

---

## SINGAPORE

SC2006 SOFTWARE ENGINEERING

### LAB 1 DELIVERABLES

LAB GROUP: TDDB

TEAM NUMBER: 49

### GROUP MEMBERS:

CHUA YONG TAI ANTHONY

HO KING HEI

KHOO QIAN YEE

MUHAMMAD WISNU DARMAWAN

RACHEL LIM LIN

TRISTAN AMADEUS SURYA

# **Table of Contents**

1. Documentation of functional and non-functional requirements
  - A. Functional Requirements
    - I. User Account Management System (Functional Requirement #1)
    - II. Food Hunt System (Functional Requirement #2)
    - III. Admin Management System (Functional Requirement #3)
  - B. Non-Functional Requirements
2. Data Dictionary
3. Initial use case model, consisting of Use Case diagram and Use Case description
  - A. Use Case Diagram
  - B. Use Case Description
    - I. For Functional Requirement #1 (User Account Management System)
      - I.I Sign Up
      - I.II Login
      - I.III Logout
      - I.IV Update Profile
      - I.V Delete Account
      - I.VI View Notifications
    - II. For Functional Requirement #2 (Food Hunt System)
      - II.I View Map
      - II.II View Food
      - II.III Search Food
      - II.IV Filter Food
      - II.V Submit Food Review
      - II.VI Update Food Review
      - II.VII Delete Food Review
      - II.VIII Flag Food Review
      - II.IX Submit Food Listing
      - II.X Update Food Listing
      - II.XI Delete Food Listing
    - III. For Functional Requirement #3 (Admin Management System)
      - III.I Process Review
      - III.II Ban User
      - III.III Notify User
4. UI Mockup
  - A. Non-Functional Requirement
  - B. Functional Requirements
    - I Account Management
      - I.I Sign Up
      - I.II Login

- I.III Logout
- I.IV Update Profile
- I.V Delete Account
- I.VI View Notifications
- II Food Hunt System
  - II.I View Map
  - II.II View Food
  - II.III Search Food
  - II.IV Filter Food
  - II.V Submit Food Review
  - II.VI Update Food Review
  - II.VII Delete Food Review
  - II.VIII Flag Food Review
- III Hawker User Interface
  - III.I Submit Food Listing
  - III.II Update Food Listing
  - III.III Delete Food Review
  - III.IV Process Review
  - III.V Ban User
  - III.VI Notify User

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

## **A. Functional Requirements**

### **I. User Account Management System (Functional Requirement #1)**

#### **1. NewUser:**

1.1. Users must be able to register for a new account.

1.1.1. System must check that all required credentials are fulfilled.

1.1.1.1. Users must input their email address.

1.1.1.1.1. If the email address is already taken, the system displays a message “Invalid email! Email is taken by another user”.

1.1.1.2. Users must input their password.

1.1.1.3. System shall be able to check the length of the password.

1.1.1.3.1. If the password does not contain at least 8 characters, with at least 1 number, the system displays a message “Password must contain at least 8 characters, with at least 1 number”.

1.1.1.4. Users must input their display name.

1.1.1.5. System shall be able to check the length of the display name.

1.1.1.5.1. If the display name is not within 5 to 30 characters, the system displays a message “Display name should be between 5-30 characters”.

1.1.2. Users will be redirected to the login page with their registered credentials to access the application.

2. ExistingUser:

2.1. Users must be able to login to the application with their registered credentials.

2.1.1. Users will have to input their registered email address.

2.1.2. Users will have to input their registered password.

2.1.2.1. The system will display an error message “Incorrect email or password! Please try again” when the email address or password or both are incorrect.

2.2. Users must be able to logout of their account.

2.2.1. Users will have to click on their profile and click on “Sign out”.

2.2.2. The system will display a message “Bye! Hope to see you soon next time!”.

2.3. Users must be able to update their profile information.

2.3.1. Users shall be able to update their password.

2.3.2. System shall be able to check the length of the password.

2.3.2.1. If the password does not contain at least 8 characters, with at least 1 number, the system displays a message “Password must contain at least 8 characters, with at least 1 number”.

2.3.3. Users shall be able to change the display name for their account.

2.3.3.1. If the display name is not within 5 to 30 characters, the system displays a message “Display name should be between 5-30 characters”.

2.3.4. Users shall be able to upload a profile picture.

- 2.3.4.1. Users must click on the “Change Photo” on the default profile picture and upload their picture.
  - 2.4. Users shall be able to delete their accounts if they wish to.
    - 2.4.1. System displays a message “Do you want to delete your account?” for the user to confirm the deletion of the account, so that account will not be deleted by mistake.
    - 2.4.2. System will remove the user’s information from the system upon deletion of the account.
    - 2.4.3. Users will have to create a new account if they want to access the application again.

## II. Food Hunt System (Functional Requirement #2)

1. Users will be able to view the suggested/popular food choices on the Home page and the entire food list on the Food page.
  - 1.1. Users will be redirected to a detailed description of the particular canteen/stall/food they clicked on.
2. SearchFood: (under Food navigation)
  - 2.1. Users will be able to search food options by food name.
  - 2.2. Users will be able to search food options by location (by canteen/stall).
    - 2.2.1. If the user inputs a food name or location that does not exist, the system will display a message “No results found”.
3. FilterFood: (under Food navigation)
  - 3.1. Users will be able to filter food options by price.
    - 3.1.1. The system will display the food from the lowest price or highest price depending on the user’s filter choice.
  - 3.2. Users will be able to filter food options by food type which is based on their dietary restrictions (e.g. Halal / Vegetarian / Vegan etc.)

3.3. Users will be able to filter food options by cuisines (e.g. Japanese / Western/Malay etc.)

3.4. Users will be able to filter food options by food on discount.

4. ViewMap:

4.1. Users must allow permission for location sharing to access the map.

4.1.1. If location sharing is not enabled, the system displays a message “Please enable location sharing to use Map”.

4.2. Users shall be able to find the location of their choice of food by the indicators on the map.

4.2.1. Users will be able to search for a particular canteen from the map.

4.2.2. Users will be able to search for a particular stall from the map.

5. ReviewManagement:

5.1. Users can submit their food reviews.

5.1.1. Users will have to indicate overall (food, store, service etc.) rating from 1 to 5.

5.1.2. Users can optionally include a photo in the review.

5.1.2.1. Users must click “Upload Photo” and upload a photo.

5.1.3. Users can include further descriptions of their review.

5.1.3.1. Users will have to click on the empty text box with the placeholder text “Enter your review description”.

5.2. Users will be able to view the list of reviews they have submitted.

5.2.1. Users will be able to edit their food review.

5.2.1.1. Users will have to click on the three dots button beside the review that they want to update.

5.2.1.2. Users will have to click “Edit Review” to make changes to their review.

5.2.1.2.1. System will have to retain the information submitted before the update of the review.

5.2.2. Users will be able to delete their food review if they decide to do so.

5.2.2.1. Users will have to click on the three dots button beside the review that they want to delete.

5.2.2.2. Users will have to select delete to remove their review.

5.2.2.2.1. System will display a message “Delete your review?” and users will have to click “Delete” to proceed with the deletion.

5.2.2.2.2. System will remove the food review for that particular food.

5.3. Users will be able to flag the reviews submitted by other users.

5.3.1. Users will be able to flag reviews submitted by other users.

5.3.1.1. Users will have to click on the three dots button beside the review that they want to flag.

5.3.1.2. System will display a message “Flag this review?” and users will have to select “Flag” to flag the review.

5.3.1.2.1. The system will send a notification to the system admin.

## 6. FoodListing:

6.1. Store owners shall be able to submit new and/or discounted food listings.

6.1.1. Store owners must provide photo(s) when listing the food.

6.1.2. Store owners must provide the price of food when listing the food.

6.1.3. Store owners must provide the name and description of the food when listing the food.

6.2. Store owners shall be able to edit the usual food menu as a discounted item and vice versa.

6.3. Store owners shall be able to delete their food listings.

6.3.1. Store owners will have to click on the three dots button beside the food listing that they want to delete.

6.3.1.1. The system will display a message “Delete your listing?” and the store owners will have to click “Delete” to proceed with the deletion.

### III. Admin Management System (Functional Requirement #3)

#### 1. BanUser:

1.1. Admin will check the reviews provided by the users.

1.1.1. If there are more than three inappropriate reviews (e.g. unreasonably low ratings, inappropriate language use in reviews) by a particular account, the admin will ban the account from the application permanently.

1.1.1.1. Admin will be able to access a list of accounts and admin will have to click on the “Ban” beside the account to ban it.

1.1.1.1.1. The system will display a message “Do you want to ban this user?” and the admin will have to select “Yes” to proceed with the suspension of the account.

1.1.1.1.2. Users will have to contact the admin for support through the email address displayed when they attempt to login with the same credentials.

#### 2. ProcessReview:

2.1. Admin shall be able to manage the reviews flagged by users.

2.1.1. If admin finds that the flagged reviews are indeed inappropriate, admin can click on the three dots button beside each of the reviews to delete them.

### 3. NotifyUsers:

3.1. Admin will be able to create notifications which will be sent to the target users by the system.

3.1.1. Admin will have to click on the “plus” button on the Notifications page to input the required fields (e.g. title, description etc.) and click on the “Notify” button to submit the notification.

## B. Non-Functional Requirements

ID	Category	Description	Priority (H/M/L)	Remarks
NFR1	Performance	Response Time: The app should respond to user actions within 2 seconds  Scalability: The app should handle up to 10,000 concurrent users without performance degradation.  Throughput: The system should process up to 100 actions per second.	High	
NFR2	Usability	User Interface: The UI should be intuitive and easy to navigate for all users.  Accessibility: The app should be accessible to users with disabilities, following WCAG 2.1 guidelines.  Localization: The app should support the English language.	High	
NFR3	Reliability	Availability: The app should have an uptime of 99.9%.  Backup: Data should be backed up every 24 hours.  Recovery: The system should recover from failures within 1 hour.	High	

NFR4	Security	<p>Authentication: Users must authenticate using a secure method.</p> <p>Data Encryption: All sensitive data should be encrypted both in transit and at rest.</p> <p>Access Control: Different user roles should have appropriate access controls.</p>	High	
NFR5	Maintainability	<p>Code Quality: The codebase should follow best practices and be well-documented to facilitate maintenance.</p> <p>Modularity: The system should be modular to allow for easy updates and feature additions.</p> <p>Monitoring: The app should have monitoring tools in place to detect and alert on issues in real-time.</p>	Medium	
NFR6	Compliance	<p>Data Protection: The app must comply with relevant data protection regulations (e.g., GDPR, PDPA).</p> <p>Audit Logs: All user actions should be logged for auditing purposes.</p>	High	
NFR7	Interoperability	<p>API Standards: The app should use RESTful APIs to ensure compatibility with other systems.</p> <p>Data Formats: The app should support common data formats like JSON and XML for data exchange.</p>	Medium	
NFR8	Environmental	<p>Supported Platforms: The app should be compatible with major operating systems (iOS, Android, Windows, macOS).</p> <p>Browser Compatibility: The web version should support major browsers (Chrome, Firefox, Safari, Edge).</p>	Low	

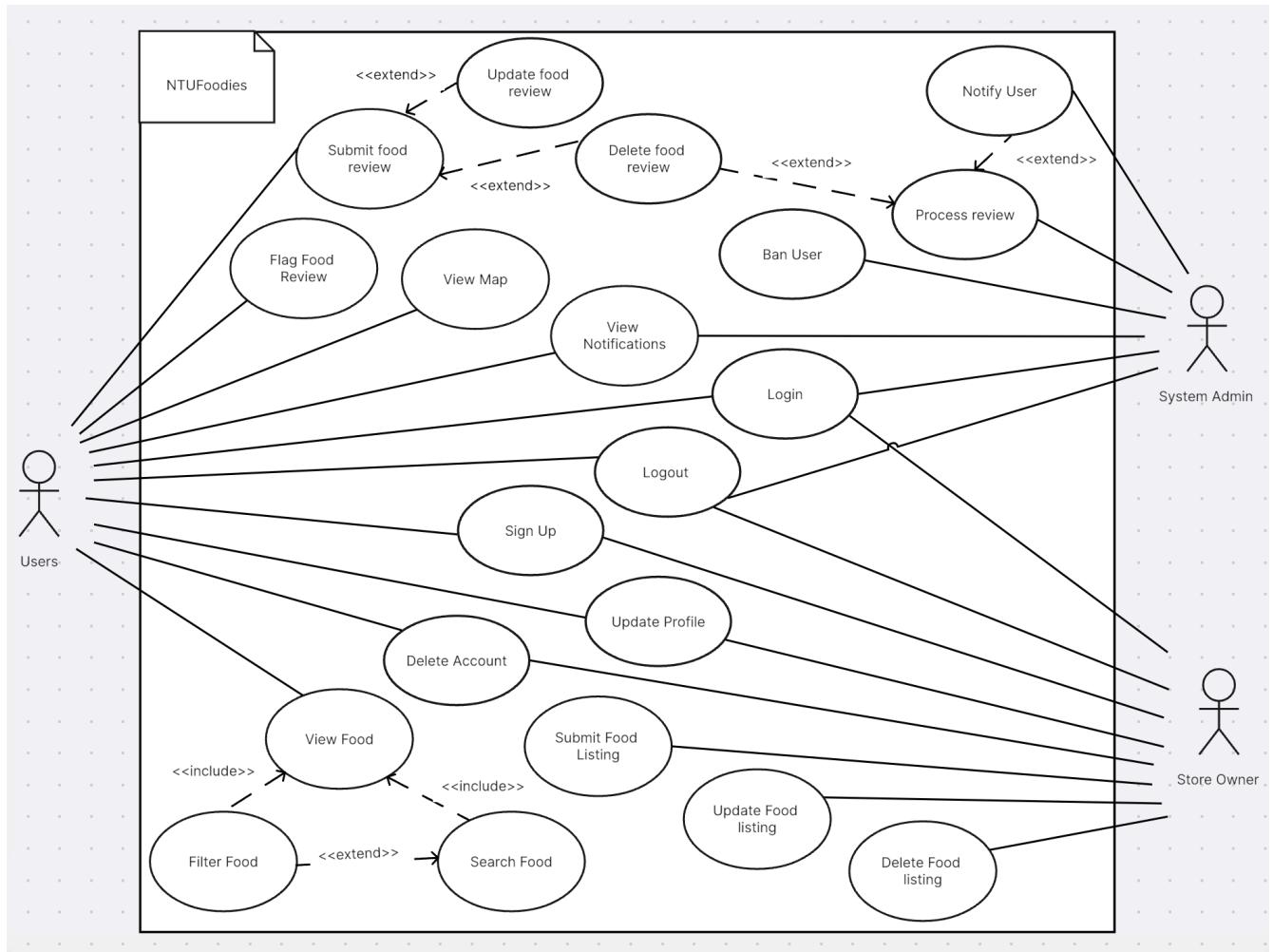
## 2. Data Dictionary

Term	Definition
Admin	A system administrator responsible for managing user reviews, banning inappropriate users, and approving food reviews for publication.
Ban	A temporary suspension of a user's account due to inappropriate reviews. Users will not be able to access the application for the duration of the ban.
Canteen	A food court/a collection of some stalls in a specific place in NTU, e.g. Canteen 11, Canteen 2, etc.
Display Name	A name chosen by the user for public display within the application. If not set, the system uses the user's registered name by default.
Email Address	A unique identifier that users must provide during account registration. The system uses this for user authentication and communication.
Food Cuisines	A category of food based on its country or region of origin. This will enable the user to filter the food based on what they want at the time of using our application.
Food Filter	A feature that allows users to filter food options based on price, dietary restrictions, or cuisines.
Food Item	A dish or meal option available for search or review within the NTU campus.
Food Listing	A feature that allows hawkers to list food items at a discount for students to purchase within a set timeframe.
Food Review	User-submitted evaluation of a food item, including a rating (1 to 5, up to 1 decimal place), optional photo, and description. Reviews are processed and verified by an admin before posting.
Food Types	A filter option to categorise food based on specific dietary needs, such as Halal, Vegetarian, or Vegan.
Inappropriate Review	A review flagged by the admin for inappropriate content, such as offensive language or unreasonable ratings, which may result in banning the user.
Password	A string of characters required for user authentication. It must contain at least 8 characters, including at least 1 number.

Profile Picture	An optional image file uploaded by the user to personalise their account. If no image is uploaded, a default profile picture is displayed.
Review Description	A textual user input detailing the user's experience with a specific food item, including taste, quality, and service.
Review Process	A system feature where an admin checks and either approves or revokes reviews before they are published on the platform.
Review Rating	A numerical score from 1 to 5 representing the overall satisfaction of a food item, store, or service.
Stall	A specific F&B establishment that sells food, where the User can give a review on the food that the stall sells. The owner of this Stall is Store Owner. Stall is also commonly known as Hawker.
Store Owner	User account that belongs to the owner of a food stall.
User Account	A registered entity in the system that allows users to access the app. Users must provide an email, password, and name to create an account.

### 3. Initial use case model, consisting of Use Case diagram and Use Case description

#### A. Use Case Diagram



#### B. Use Case Description

##### I. For Functional Requirement #1 (User Account Management System)

###### I.I Sign Up

Use Case ID:	ACCOUNT_UC_1		
Use Case Name:	Sign Up		
Created By:	Anthony	Last Updated By:	

Date Created:	03/09/2024	Date Last Updated:	
---------------	------------	--------------------	--

Actor:	User / Store Owner
Description:	This use case allows users to sign up for an account to access the relevant functions according to the user's role. The various user roles that are required to sign up are user & store owner. To sign up for the account, the user has to key in their email address and password. Once the account has been created, the user will be redirected to the login page
Preconditions:	NA
Postconditions:	The user will be redirected to the relevant home page
Priority:	High
Frequency of Use:	Once per user
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user enters their email and password</li> <li>2. The user submits their email and password</li> <li>3. The system validates and verifies the submitted email and password</li> <li>4. The system displays a message saying "Account has been created" when the sign-up is successful</li> <li>5. The system will redirect the user to the relevant home page</li> <li>6. The use case ends</li> </ol>
Alternative Flows:	<p>AF-S3a: If the email is used by another user</p> <ol style="list-style-type: none"> <li>1. The system displays a message saying "Invalid email! Email is used by another user"</li> <li>2. The use case resumes at main flow step 1</li> </ol> <p>AF-S3b: If the password is invalid</p> <ol style="list-style-type: none"> <li>1. The system displays a message saying "Invalid password! Password must contain at least 8 characters with at least 1 number"</li> <li>2. The user case resumes at main flow step 1</li> </ol> <p>AF-S3c: If the email and/or password is missing</p> <ol style="list-style-type: none"> <li>1. The system displays a message saying "Email or password must be filled"</li> <li>2. The use case resumes at main flow step 1</li> </ol>

Exceptions:	If any required information is missing or invalid, an error message is displayed.
Includes:	NA
Special Requirements:	The system needs to validate data input by the user
Assumptions:	NA
Notes and Issues:	NA

## I.II Login

Use Case ID:	ACCOUNT_UC_2		
Use Case Name:	Login		
Created By:	Anthony	Last Updated By:	
Date Created:	03/09/2024	Date Last Updated:	

Actor:	User / Store Owner / System Admin
Description:	This use case allows users to log into the system to access the relevant functions according to the user's role. The various user roles that can log in are user, store owner & system admin. To log in, all users have to enter their email and password. Upon successful login, the system displays the relevant user's home page
Preconditions:	The user has to have a valid account created beforehand
Postconditions:	The system displays the relevant homepage
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user enters their email and password</li> <li>2. The user submits their email and password</li> <li>3. The system validates and verifies the submitted email and password</li> <li>4. The system will display the user's homepage</li> <li>5. The use case ends</li> </ol>
Alternative Flows:	<p>AF-S3a: If the email and/or password is invalid</p> <ol style="list-style-type: none"> <li>1. The system displays a message saying "Invalid email or password! Please try again"</li> <li>2. The use case resumes at main flow step 1</li> </ol> <p>AF-S3b: If the email and/or password is not filled in</p>

	<ol style="list-style-type: none"> <li>1. The system displays a message saying “Email and/or password must be filled”</li> <li>2. The use case resumes at main flow step 1</li> </ol> <p>AF-S3C:</p> <ol style="list-style-type: none"> <li>1. The system displays a message saying “Your account is banned. Please contact the administrator!”</li> <li>2. The use case resumes at main flow step 1</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>1. If any required information is missing or invalid, an error message is displayed.</li> <li>2. If the user account is banned, an error message is displayed.</li> </ol>
Includes:	NA
Special Requirements:	The system needs to validate data input by the user
Assumptions:	NA
Notes and Issues:	NA

### I.III Logout

Use Case ID:	ACCOUNT_UC_3		
Use Case Name:	Logout		
Created By:	Anthony	Last Updated By:	
Date Created:	03/09/2024	Date Last Updated:	

Actor:	User / Store Owner / System Admin
Description:	This use case allows users to log out of the system. The various user roles that can log out are user, store owner & system admin. Upon successful logout, the system will display a message saying “Bye! Hope to see you next time!” and the login page will be displayed
Preconditions:	The user has to be signed in
Postconditions:	<ol style="list-style-type: none"> <li>1. The system displays a message saying “Bye! Hope to see you next time!”</li> <li>2. The system displays the login homepage</li> </ol>
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user clicks on the profile button</li> <li>2. The system will display the user’s profile page</li> </ol>

	<ul style="list-style-type: none"> <li>3. The user clicks on “Sign out”</li> <li>6. The system displays a message saying Bye! Hope to see you next time!</li> <li>7. The system redirects the user back to the Login page</li> <li>8. The use case ends</li> </ul>
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Special Requirements:	NA
Assumptions:	NA
Notes and Issues:	NA

#### I.IV Update Profile

Use Case ID:	ACCOUNT_UC_4		
Use Case Name:	Update Profile		
Created By:	Anthony	Last Updated By:	
Date Created:	03/09/2024	Date Last Updated:	

Actor:	User / Store Owner
Description:	This use case allows users to update their profiles. The various user roles that can update their profiles are the users and store owners. The user can update the display name, profile picture and password. Upon successful update, the system will display a message saying “Profile updated”
Preconditions:	The user has to be signed in
Postconditions:	<ul style="list-style-type: none"> <li>1. The system displays a message saying “Profile updated”</li> <li>2. The system redirects the user back to the Profile page</li> </ul>
Priority:	Medium
Frequency of Use:	Low
Flow of Events:	<ul style="list-style-type: none"> <li>1. The user clicks on the profile button</li> <li>2. The system will display the user’s profile page</li> <li>3. The user clicks on “Update profile”</li> <li>4. The user enters the new input on the relevant fields</li> <li>5. The user clicks on the “Update”</li> <li>6. The system displays a message saying “Profile Updated”</li> </ul>

	<p>7. The system redirects the user back to the Profile page</p> <p>8. The use case ends</p>
Alternative Flows:	<p>AF-S4a: If the password is invalid</p> <ol style="list-style-type: none"> <li>1. The system displays a message saying “Invalid password! Password must contain at least 8 characters with at least 1 number”</li> <li>2. The use case resumes at main flow step 1</li> </ol>
Exceptions:	If any password is invalid, an error message is displayed.
Includes:	NA
Special Requirements:	The system needs to validate data input by the user
Assumptions:	NA
Notes and Issues:	NA

#### I.V Delete Account

Use Case ID:	ACCOUNT_UC_5		
Use Case Name:	Delete Account		
Created By:	Anthony	Last Updated By:	
Date Created:	03/09/2024	Date Last Updated:	

Actor:	User / Store Owner
Description:	This use case allows users to delete their accounts. The various user roles that can delete their accounts are the users and store owners. When the user clicks on the “Delete Account” button, the system displays a confirmation message saying “Do you want to delete your account?”. Upon successful deletion, the system will display a message saying “Account deleted”
Preconditions:	The user has to be signed in
Postconditions:	The system redirects the user back to the login page
Priority:	Medium
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user clicks on the profile button</li> <li>2. The system will display the user’s profile page</li> <li>3. The user clicks on “Delete Account”</li> <li>4. The system displays a message saying “Do you want to delete your account?”</li> </ol>

	5. The user clicks on the “Delete” 6. The system displays a message saying “Account deleted” 7. The system redirects the user back to the login page 8. The use case ends
Alternative Flows:	AF-S5a: If the user clicks “Back” <ol style="list-style-type: none"> <li>1. The system will display the user’s profile page</li> <li>2. The use case ends</li> </ol>
Exceptions:	NA
Includes:	NA
Special Requirements:	NA
Assumptions:	NA
Notes and Issues:	NA

## I.VI View Notifications

Use Case ID:	ACCOUNT_UC_9		
Use Case Name:	View Notifications		
Created By:	Anthony	Last Updated By:	
Date Created:	04/09/2024	Date Last Updated:	

Actor:	User / System Admin
Description:	<p>This use case allows the user or system admin to view notifications.</p> <p>When the user clicks on the Bell button at the top of their Home page, the system displays all the notifications that the user has.</p> <p>When the system admin clicks on the “Notifications”, the system displays all the notifications that the system admin has.</p>
Preconditions:	The user (user and system admin) has to be signed in
Postconditions:	The system sends all the notifications the user has
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user clicks on the Bell button at the top of their Home page</li> <li>2. The system displays all the notifications the user has</li> <li>3. The use case ends</li> </ol> <p>OR</p> <ol style="list-style-type: none"> <li>1. The system admin clicks on the “Notifications” button</li> </ol>

	<ol style="list-style-type: none"> <li>2. The system displays all the notifications the system admin has</li> <li>3. The use case ends</li> </ol>
Alternative Flows:	NA
Exceptions:	If there are no notifications, the system will display “You have no notifications”
Includes:	NA
Special Requirements:	NA
Assumptions:	NA
Notes and Issues:	NA

## II. For Functional Requirement #2 (Food Hunt System)

### II.I View Map

Use Case ID:	USER_UC_1		
Use Case Name:	View map		
Created By:	Anthony	Last Updated By:	
Date Created:	03/09/2024	Date Last Updated:	

Actor:	User
Description:	This use case allows users to view the map that shows the food areas in NTU. When the user clicks on the “Map” button, the system displays a map that shows the food areas in NTU
Preconditions:	User has to enable Location sharing in their local device
Postconditions:	The system displays a map that shows the food areas in NTU
Priority:	Medium
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user clicks on the map button</li> <li>2. The system will display a map that shows the food areas in NTU</li> <li>3. The use case ends</li> </ol>
Alternative Flows:	<p>AF-S1a: If location sharing is not enabled on the user’s local device</p> <ol style="list-style-type: none"> <li>1. The system displays an error message saying “Please enable location sharing to use Map”</li> </ol>

	<p>2. The user will be redirected back to the relevant user's home page</p> <p>3. The use case ends</p>
Exceptions:	If location sharing is not enabled on the user's local device, the map will not work
Includes:	NA
Special Requirements:	Location sharing must be enabled for the user's local device
Assumptions:	NA
Notes and Issues:	NA

### II.II View Food

Use Case ID:	USER_UC_2		
Use Case Name:	View Food		
Created By:	KH	Last Updated By:	
Date Created:	04/09/2024	Date Last Updated:	

Actor:	User
Description:	This use case allows the user to view the available food in NTU. When the user clicks on the “Food button”, the system displays the available food from the various stalls in NTU
Preconditions:	The user has to be signed in
Postconditions:	The system displays the available food from the various stalls in NTU
Priority:	High
Frequency of Use:	High
Flow of Events:	<p>1. The user clicks on the “Food” button</p> <p>2. The system displays the available food from the various stalls in NTU</p> <p>3. The use case ends</p>
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Special Requirements:	NA
Assumptions:	NA
Notes and Issues:	NA

## II.III Search Food

Use Case ID:	USER_UC_3		
Use Case Name:	Search Food		
Created By:	KH	Last Updated By:	
Date Created:	04/09/2024	Date Last Updated:	

Actor:	User
Description:	This use case allows the user to search for food by food name or by location. When the user enters a specific food or location on the Search bar within the “Food” page, the system displays the relevant results based on the query
Preconditions:	The user has to be signed in
Postconditions:	The food options related to the query will be displayed
Priority:	High
Frequency of Use:	High
Flow of Events:	<ul style="list-style-type: none"> <li>1. The user clicks on the “Food” button</li> <li>2. The user clicks on the Search bar within the “Food” page</li> <li>3. The user types in a food name</li> <li>4. The system will show all stalls that sell that particular food</li> <li>5. The use case ends</li> </ul> <p>OR</p> <ul style="list-style-type: none"> <li>1. The user clicks on the “Food” button</li> <li>2. The user clicks on the Search bar within the “Food” page</li> <li>3. The user types in the location of one of the food areas in NTU</li> <li>4. The system will show the available food located in that particular food area</li> <li>5. The use case ends</li> </ul>
Alternative Flows:	<p>AF-S2a: If the user clicks “Back” within the Search page</p> <ul style="list-style-type: none"> <li>1. The system will redirect the user back to the “View Food” page</li> <li>2. The use case ends</li> </ul>
Exceptions:	If the user types in an invalid location or food name, the system will display a message saying “No results found”.
Includes:	View Food

Special Requirements:	NA
Assumptions:	NA
Notes and Issues:	NA

## II.IV Filter Food

Use Case ID:	USER_UC_4		
Use Case Name:	Filter Food		
Created By:	KH	Last Updated By:	
Date Created:	04/09/2024	Date Last Updated:	

Actor:	User
Description:	This use case allows the user to filter food by Cuisine, Type, Price or Discount. When the user clicks on the filter button, the system will display the options for the user to filter food by. Upon clicking the “Apply Filter” button after selecting the fields to filter by, the system will display the relevant results based on the query
Preconditions:	The user has to be signed in
Postconditions:	The food options related to the query will be displayed
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user clicks on the “Food” button</li> <li>2. The user clicks on the filter button</li> <li>3. The user selects one of the filters or a combination of them             <ol style="list-style-type: none"> <li>a. If the user selects Price, the system will display the food, from the cheapest/most expensive food at the top of the list</li> <li>b. If the user selects a Cuisine, the system will display the food from stalls that sell that specific cuisine</li> <li>c. If the user selects Discount, the system will display all the food that are listed on a discount</li> <li>d. If the user selects Type, only stalls that sell that specific type of food will be shown in the list</li> </ol> </li> <li>4. The use case ends</li> </ol>
Alternative Flows:	AF-S3a: If the user clicks “Cancel” within the Filter Food page

	<ol style="list-style-type: none"> <li>1. The system will redirect the user back to the “View Food” page</li> <li>2. The use case ends</li> </ol>
Exceptions:	NA
Includes:	View Food
Special Requirements:	NA
Assumptions:	NA
Notes and Issues:	NA

## II.V Submit Food Review

Use Case ID:	USER_UC_5		
Use Case Name:	Submit Food Review		
Created By:	KH	Last Updated By:	
Date Created:	04/09/2024	Date Last Updated:	

Actor:	User
Description:	This use case allows the user to submit their food review. When the user clicks the “Leave a review” button shown on the stall page, the user can rate the food, upload a photo and write a description about the food. Upon successful submission, the system displays a message saying “Review submitted”
Postconditions:	The user has to be signed in
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user clicks the “Leave a review” button</li> <li>2. The system displays a form to be filled up</li> <li>3. The user fills up relevant portions of the form and clicks “Submit Review”</li> <li>4. The system displays a message saying “Review submitted”</li> <li>5. The system redirects the user to the relevant stall page</li> <li>6. The use case ends</li> </ol>
Alternative Flows:	<p>AF-3a: If there are missing fields on the form</p> <ol style="list-style-type: none"> <li>1. The system displays a message “Please fill up relevant parts of the form”</li> <li>2. The use case resumes at main flow step 2</li> </ol>

Exceptions:	If any required information is missing or invalid, an error message is displayed.
Includes:	NA
Special Requirements:	The system needs to validate the input from the user
Assumptions:	NA
Notes and Issues:	NA

## II.VI Update Food Review

Use Case ID:	USER_UC_6		
Use Case Name:	Update Food Review		
Created By:	KH	Last Updated By:	
Date Created:	04/09/2024	Date Last Updated:	

Actor:	User
Description:	This use case allows the user to edit a review that they have previously submitted. When the user clicks on the “Edit review” button on a previously submitted review, the user can enter the new inputs in the form. Upon successful submission, the system displays a message saying “Review edited”
Preconditions:	<ol style="list-style-type: none"> <li>1. The user has to be signed in</li> <li>2. There must be a review that was previously submitted by the user</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The system displays a message saying “Review edited”</li> <li>2. The system redirects the user to the Reviews page</li> </ol>
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user clicks on the “Published Reviews” button</li> <li>2. The user clicks on the “Edit Review” button on a previously submitted review</li> <li>3. The system displays a form with details of their submitted review</li> <li>4. The user edits relevant portions of the form and selects “Save”</li> <li>5. The system displays a message saying “Review edited”</li> <li>6. The system redirects the user to the Reviews page</li> </ol>

	<p>7. The use case ends</p>
Alternative Flows:	<p>AF-S4a: If there is missing input for the mandatory fields on the form</p> <ol style="list-style-type: none"> <li>1. The system displays a message “Please fill up relevant parts of the form”</li> <li>2. The use case resumes at main flow step 2</li> </ol>
Exceptions:	If any required information is missing or invalid, an error message is displayed.
Includes:	NA
Special Requirements:	The system needs to validate the input from the user
Assumptions:	NA
Notes and Issues:	NA

## II.VII Delete Food Review

Use Case ID:	USER_UC_7		
Use Case Name:	Delete Food Review		
Created By:	Anthony	Last Updated By:	
Date Created:	04/09/2024	Date Last Updated:	

Actor:	User / System Admin
Description:	<p>This use case allows the user or system admin to delete a review.</p> <p>When the user clicks on the “Delete review” button on a previously submitted review, the system displays a confirmation message saying “Delete your review?”. Upon successful deletion, the system displays a message saying “Review deleted”.</p> <p>When the system admin has processed the review, the system admin may delete the review</p>
Preconditions:	<ol style="list-style-type: none"> <li>1. The user has to be signed in</li> <li>2. There must be a review that was previously submitted by the user</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The system displays a message saying “Review deleted”</li> <li>2. The system redirects the user to the Reviews page</li> </ol>
Priority:	Medium
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user clicks on the “Published Reviews” button</li> </ol>

	<ol style="list-style-type: none"> <li>2. The user clicks on the “Delete review” on a previously submitted review</li> <li>3. The system displays a confirmation message saying “Delete your review?”</li> <li>4. The user clicks on the “Delete” button</li> <li>5. The system displays a message saying “Review deleted”</li> <li>6. The system redirects the user to the Reviews page</li> <li>7. The use case ends</li> </ol>
Alternative Flows:	<p>AF-S5a: If the user clicks “Back”</p> <ol style="list-style-type: none"> <li>3. The system will display the Published Reviews page</li> <li>4. The use case ends</li> </ol>
Exceptions:	NA
Includes:	NA
Special Requirements:	NA
Assumptions:	NA
Notes and Issues:	NA

## II.VIII Flag Food Review

Use Case ID:	USER_UC_8		
Use Case Name:	Flag Food Review		
Created By:	Anthony	Last Updated By:	
Date Created:	04/09/2024	Date Last Updated:	

Actor:	User
Description:	This use case allows the user to flag a food review. When the user clicks on the “Flag” button on a previously submitted review, the system sends a notification to the system admin about the flagged review and the system displays a message saying “Review flagged”
Preconditions:	<ol style="list-style-type: none"> <li>1. The user has to be signed in</li> <li>2. There must be a review that was previously submitted by the user</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The system sends a notification to the system admin</li> <li>2. The system displays a message saying “Review flagged”</li> </ol>
Priority:	Medium
Frequency of Use:	Low

Flow of Events:	<ol style="list-style-type: none"> <li>1. The user clicks on a previously submitted review by another user</li> <li>2. The user clicks on the “Flag button”</li> <li>3. The system sends a notification to the system admin</li> <li>4. The system displays a message saying “Review flagged”</li> <li>5. The use case ends</li> </ol>
Alternative Flows:	<p>AF-S2a: If the user clicks “Back” on the Review page</p> <ol style="list-style-type: none"> <li>1. The system redirects the user back to the relevant page</li> <li>2. The use case ends</li> </ol>
Exceptions:	NA
Includes:	NA
Special Requirements:	NA
Assumptions:	NA
Notes and Issues:	NA

## II.IX Submit Food Listing

Use Case ID:	STORE OWNER_UC_1		
Use Case Name:	Submit Food Listing		
Created By:	Anthony	Last Updated By:	
Date Created:	04/09/2024	Date Last Updated:	

Actor:	Store Owner
Description:	This use case allows the store owner to submit food listings onto the app. The store owner can submit food listings such as new and/or discounted food onto the app for users to view. Upon successful submission, the system will display a message saying “Food submitted”
Preconditions:	The store owner has to be signed in
Postconditions:	The system displays the newly listed food submitted by the store
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. The store owner clicks on the “Submit Food” button</li> <li>2. The store owner enters the details (price, image, description and discount if any) of the food</li> <li>3. The store owner clicks on the “Submit” button</li> </ol>

	<p>4. The system displays a message saying “Food submitted”</p> <p>5. The use case ends</p>
Alternative Flows:	<p>AF-S2a: If the store owner clicks “Back” within the Submit Food page</p> <ol style="list-style-type: none"> <li>1. The system will redirect the store owner back to the relevant store owner home page</li> <li>2. The use case ends</li> </ol> <p>AF-3a: If there are missing fields on the form</p> <ol style="list-style-type: none"> <li>1. The system displays a message “Please fill up relevant parts of the form”</li> <li>2. The use case resumes at main flow step 2</li> </ol>
Exceptions:	If any required information is missing or invalid, an error message is displayed.
Includes:	NA
Special Requirements:	The system needs to validate the input from the store owner
Assumptions:	NA
Notes and Issues:	NA

## II.X Update Food Listing

Use Case ID:	STORE OWNER_UC_2		
Use Case Name:	Update Food Listing		
Created By:	Anthony	Last Updated By:	
Date Created:	04/09/2024	Date Last Updated:	

Actor:	Store Owner
Description:	This use case allows the store owner to update the submitted food listings on the app. The user can update the name, category, price, picture and discount if any. Upon successful update, the system will display a message saying “Listing updated”
Preconditions:	<ol style="list-style-type: none"> <li>1. The store owner has to be signed in</li> <li>2. There must be a food listing that was previously submitted by the store owner</li> </ol>
Postconditions:	The system displays a message saying “Listing updated”
Priority:	High

Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> <li>1. The store owner clicks on the specific food listing to update</li> <li>2. The store owner clicks on the “Edit Listing button”</li> <li>3. The store owner enters the new input on the relevant fields</li> <li>4. The store owner clicks on the “Update” button</li> <li>5. The system displays a message saying “Listing updated”</li> <li>6. The use case ends</li> </ol>
Alternative Flows:	<p>AF-S3a: If the store owner clicks “Back” within the Update Food page</p> <ol style="list-style-type: none"> <li>1. The system will redirect the store owner back to the food listing page</li> <li>2. The use case ends</li> </ol> <p>AF-S4a: If there is missing input for the mandatory fields on the form</p> <ol style="list-style-type: none"> <li>1. The system displays a message “Please fill up relevant parts of the form”</li> <li>2. The use case resumes at main flow step 3</li> </ol>
Exceptions:	If any required information is missing or invalid, an error message is displayed.
Includes:	NA
Special Requirements:	The system needs to validate the input from the store owner
Assumptions:	NA
Notes and Issues:	NA

## II.XI Delete Food Listing

Use Case ID:	STORE OWNER_UC_3		
Use Case Name:	Delete Food Listing		
Created By:	Anthony	Last Updated By:	
Date Created:	04/09/2024	Date Last Updated:	

Actor:	Store Owner
Description:	This use case allows the store owner to delete the submitted food listings on the app. The user can delete any of the submitted food

	listings. Upon successful deletion, the system will display a message saying “Listing deleted”
Preconditions:	<ul style="list-style-type: none"> <li>3. The store owner has to be signed in</li> <li>4. There must be a food listing that was previously submitted by the store owner</li> </ul>
Postconditions:	The system displays a message saying “Listing deleted”
Priority:	Medium
Frequency of Use:	Low
Flow of Events:	<ul style="list-style-type: none"> <li>1. The store owner clicks on the specific food listing to update</li> <li>2. The store owner clicks on the More (3 dots) button</li> <li>3. The system displays a message saying “Do you want to delete this listing?”</li> <li>4. The store owner clicks on the “Delete” button</li> <li>5. The system displays a message saying “Listing deleted”</li> <li>6. The use case ends</li> </ul>
Alternative Flows:	<p>AF-S2a: If the store owner clicks “Back” within the Delete Food page</p> <ul style="list-style-type: none"> <li>1. The system will redirect the store owner back to the food listing page</li> <li>2. The use case ends</li> </ul>
Exceptions:	NA
Includes:	NA
Special Requirements:	NA
Assumptions:	NA
Notes and Issues:	NA

### III. For Functional Requirement #3 (Admin Management System)

#### III.I Process Review

Use Case ID:	ADMIN_UC_1		
Use Case Name:	Process Review		
Created By:	Anthony	Last Updated By:	
Date Created:	04/09/2024	Date Last Updated:	

Actor:	System Admin
--------	--------------

Description:	This use case allows the system admin to process the food reviews. The system admin can process the flagged food reviews that are flagged by users. When the user clicks on the “Process Review” button, the system displays all the flagged food reviews, with the most recent ones at the top
Preconditions:	The system admin has to be signed in
Postconditions:	NA
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<p>1. The system admin clicks on the “Process Review” button</p> <p>2. The system displays all the flagged reviews with the most recent ones at the top</p> <p>3. The system admin can click on the More (3 dots) button to see further actions available for any specific review</p> <p>4. The system admin clicks on “Delete review” button</p> <p>5. The system displays a message saying “Review deleted”</p> <p>6. The system sends a notification to the user saying the review has been deleted</p> <p>7. The use case ends</p> <p>OR</p> <p>1. The system admin clicks on the “Process Review” button</p> <p>2. The system displays all the flagged reviews with the most recent ones at the top</p> <p>3. The system admin can click on the More (3 dots) button to see further actions available for any specific review</p> <p>4. The system admin clicks on “Resolve” button</p> <p>5. The system removes the review from the list</p> <p>6. The use case ends</p>
Alternative Flows:	<p>AF-S3a: If the store owner clicks “Back”</p> <p>1. The system will redirect the store owner back Process Review page</p>
Exceptions:	NA
Includes:	NA
Special Requirements:	NA
Assumptions:	NA
Notes and Issues:	NA

### III.II Ban User

Use Case ID:	ADMIN_UC_2		
Use Case Name:	Ban User		
Created By:	Anthony	Last Updated By:	
Date Created:	04/09/2024	Date Last Updated:	

Actor:	System Admin
Description:	This use case allows the system admin to ban users. The system admin can ban users from using the app. Upon successful banning, the system will display a message saying “User banned”
Preconditions:	<ol style="list-style-type: none"> <li>1. The system admin has to be signed in</li> <li>2. The target account must not be banned beforehand</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The system displays a message saying “User banned”</li> <li>2. The banned account system admin will not be able to log in</li> <li>3. The system redirects the system admin to the relevant page that displays the user list (users excluding store owner)</li> </ol>
Priority:	Medium
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system admin clicks on the “Ban User” button</li> <li>2. The system displays the user list for the system admin to ban</li> <li>3. The system admin clicks on the target account to ban</li> <li>4. The system displays the Ban Account page</li> <li>5. The system admin enters the relevant description</li> <li>6. The system clicks on the “Ban” button</li> <li>7. The system displays a message saying “Do you want to ban the user?”</li> <li>8. The system admin clicks on the “Yes” button</li> <li>9. The system displays a message saying “User banned”</li> <li>10. The system redirects the system admin to the page that displays the user list</li> <li>11. The use case ends</li> </ol>
Alternative Flows:	<p>AF-5a: If the system admin clicks “Back” within the Ban account page:</p> <ol style="list-style-type: none"> <li>1. The use case resumes at main flow step 2</li> </ol>

	AF-8a: If the system admin clicks the “No” button 1. The use case resumes at main flow step 5
Exceptions:	NA
Includes:	NA
Special Requirements:	NA
Assumptions:	NA
Notes and Issues:	NA

### III.III Notify User

Use Case ID:	ADMIN_UC_3		
Use Case Name:	Notify User		
Created By:	Anthony	Last Updated By:	
Date Created:	04/09/2024	Date Last Updated:	

Actor:	System Admin
Description:	This use case allows the system admin to notify users. The system admin can send notifications to users directly. When the system admin clicks the “Notify” button, the system sends a notification to the target user
Preconditions:	The system admin has to be signed in
Postconditions:	<ol style="list-style-type: none"> <li>1. The user receives a notification from the system admin</li> <li>2. The system redirects the system admin back to the Notifications page</li> </ol>
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system admin clicks on “Notifications” button</li> <li>2. The system displays the Notification page</li> <li>3. The system admin clicks on the Plus (+) button</li> <li>4. The system admin enters input on the relevant fields on the Notification form</li> <li>5. The system admin clicks the “Notify” button</li> <li>6. The system sends a notification to the target user</li> <li>7. The system redirects the system admin back to the Notifications page</li> </ol>

	8. The use case ends
Alternative Flows:	AF-3: If system admin clicks “Cancel” or Cross (x) button, 1. The system redirects the system admin back to the Notifications page
Exceptions:	If any required information is missing or invalid, an error message is displayed.
Includes:	NA
Special Requirements:	The system needs to validate the input from the store owner
Assumptions:	NA
Notes and Issues:	NA

## 4. UI Mockups

### A. Non-Functional Requirement



### B. Functional Requirements

#### I Account Management

##### I.I Sign Up

A composite image showing the sign-up process across three screens. The first screen shows a welcome message and input fields for email and password. The second screen shows the sign-up form with fields for display name, email address, and password, followed by a success message. The third screen shows a keyboard and a red 'OK' button.

**Welcome to myNTU!**

Email Address

Password

Forgot password?

Login

Not a member? [Register now](#)

Or continue with

G

**Sign up**

Create an account to get started

**Display Name**

Luc

Display name should be between 5 -30 characters

**Email Address**

name@email.com

**Password**

Create a passaword

Confirm password

Password should contain at least 8 characters, including at least 1 number

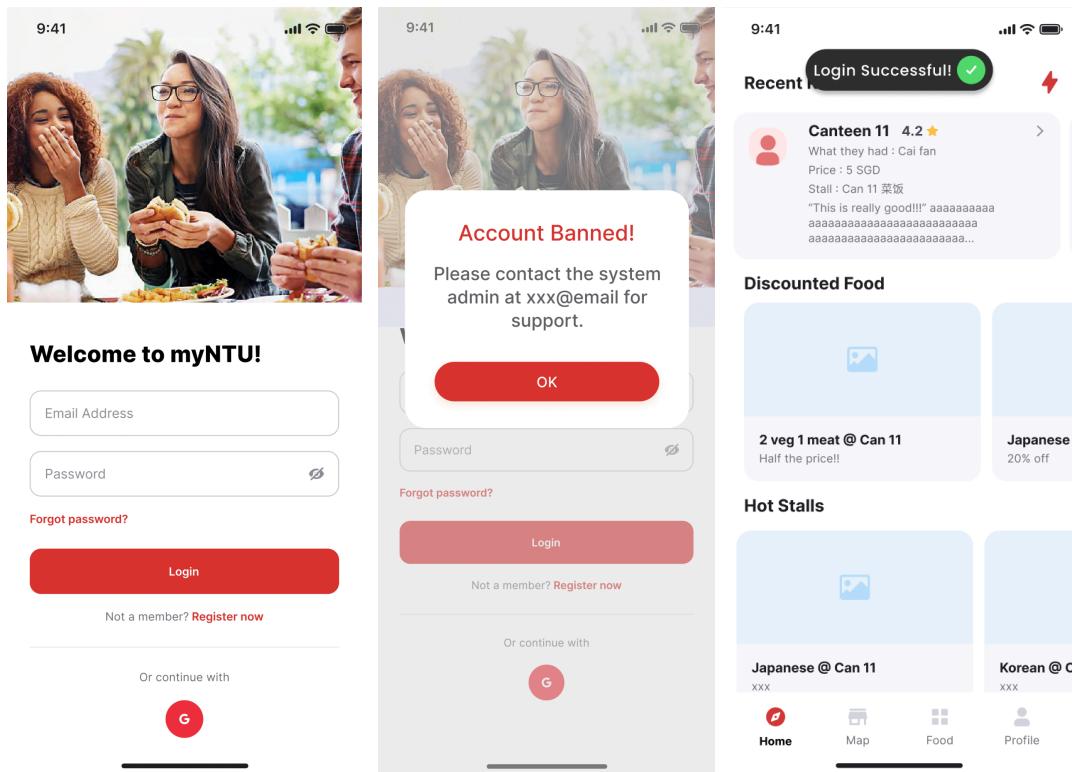
Q W E R T Y U I O P  
A S D F G H J K L  
Z X C V B N M ↻  
123 space return  
😊

**Success!**

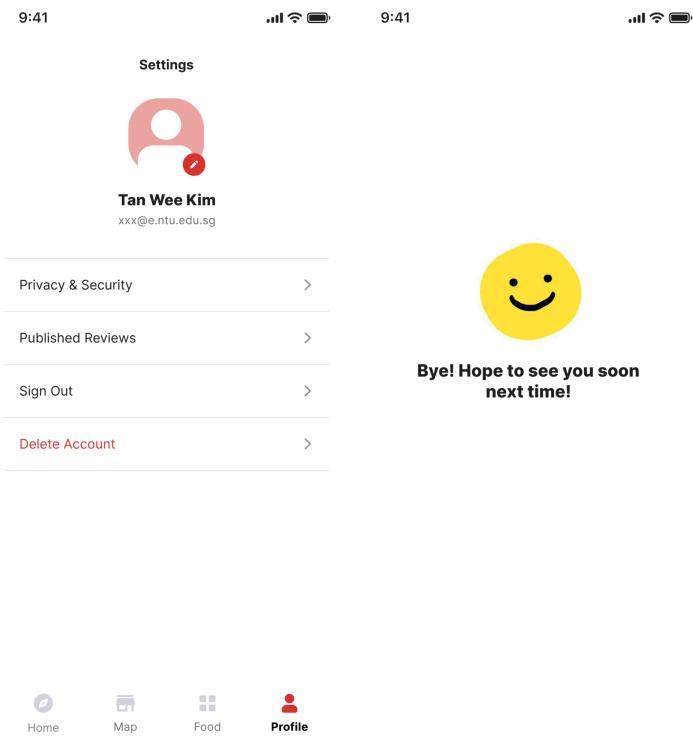
Welcome!

OK

## I.II Login



## I.III Logout

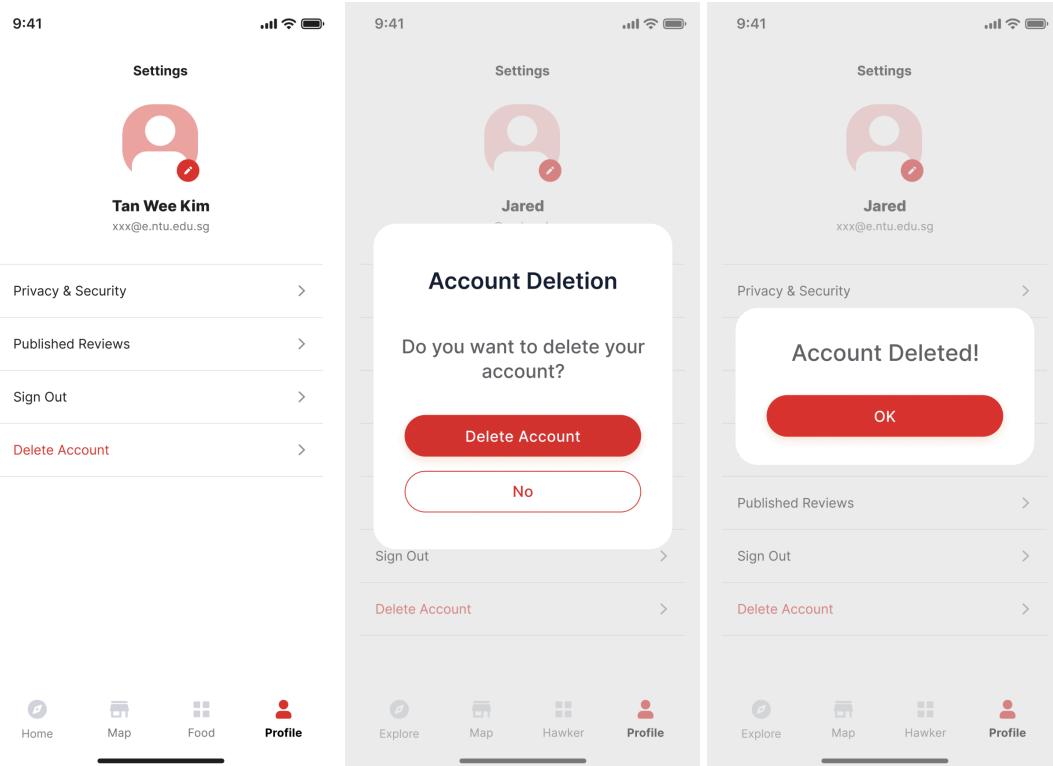


## I.IV Update Profile

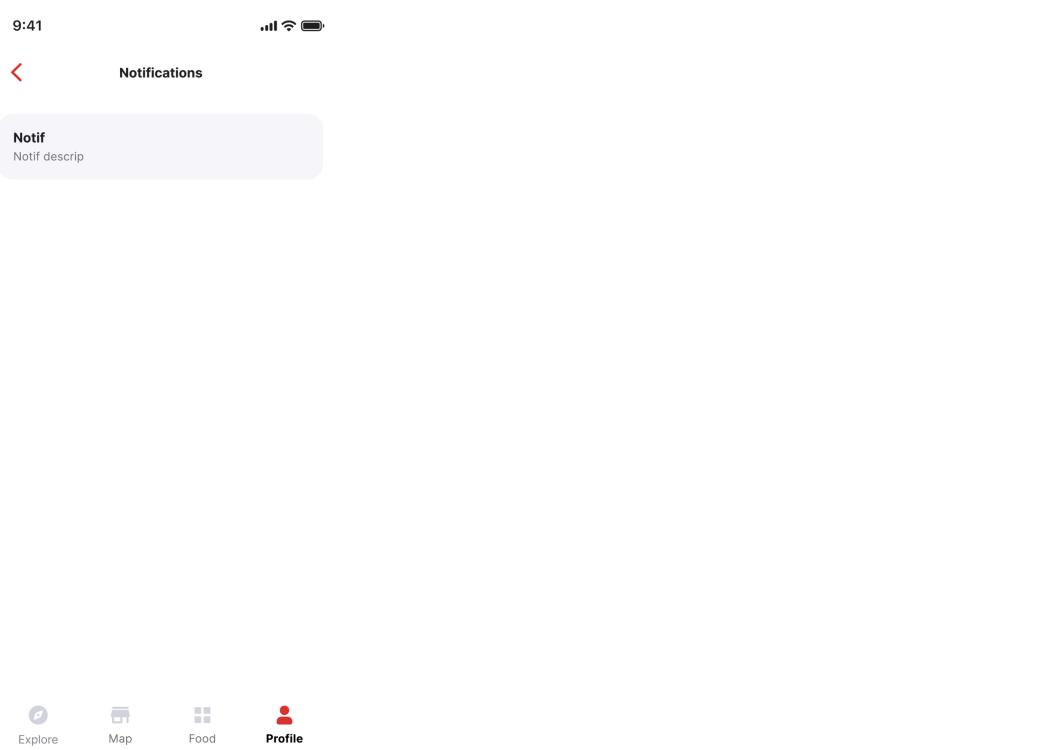
The screenshots illustrate the steps to update a user's profile:

- Initial Profile View:** Shows the user's current profile picture (Tan Wee Kim), name (Tan Wee Kim), and email (xxx@e.ntu.edu.sg). Below the profile are navigation icons for Home, Map, Food, and Profile.
- Settings Screen:** The user selects "Settings". This screen shows "Privacy & Security", "Published Reviews", "Sign Out", and "Delete Account".
- Profile Update Screen:** The user selects "Update Profile". This screen shows "Change Picture" (with a placeholder image of a person), "Display Name" (with the value "Jared" in a red-bordered field), and "Email Address" (with the value "twk@e.ntu.edu.sg").
- Confirmation Screen:** A large red "Confirm" button is displayed at the top right.
- New Password Screen:** The user is prompted to enter a new password ("Create a password") and confirm it ("Confirm password"). A note states: "Password should contain at least 8 characters, including at least 1 number." Navigation icons for Explore, Map, Food, and Profile are shown below.
- Profile Updated Confirmation:** A red banner at the top says "Profile Updated!" with a checkmark icon. The user's profile is now updated to "Jared" (xxx@e.ntu.edu.sg). The navigation icons remain the same.
- Final Profile View:** The user's profile is now "Jared" (xxx@e.ntu.edu.sg). The navigation icons for Explore, Map, Food, and Profile are shown at the bottom.

## I.V Delete Account

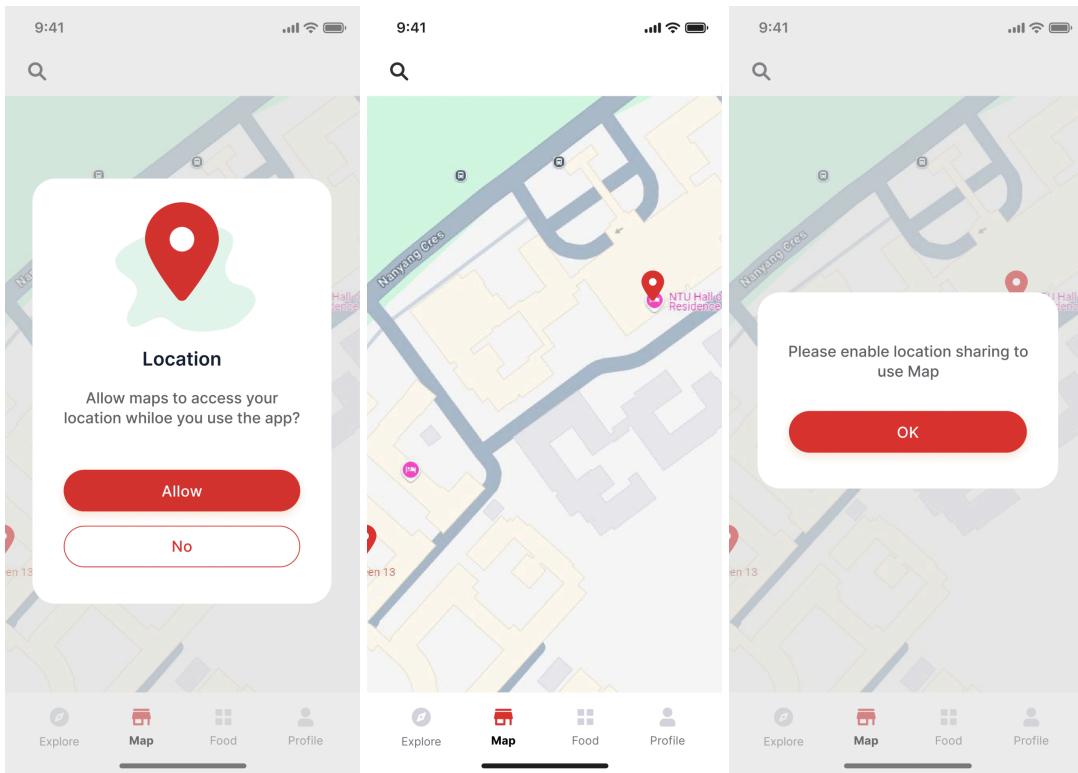


## I.VI View Notifications

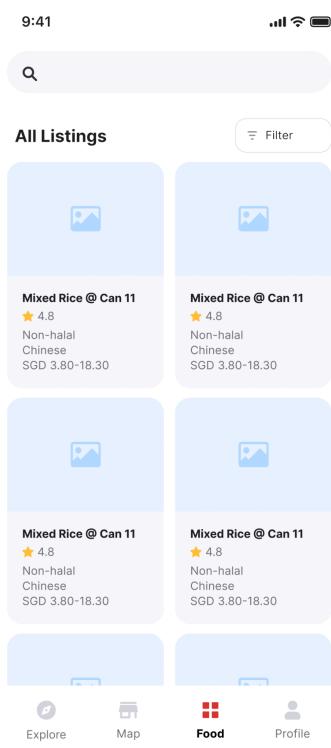


## II Food Hunt System

### II.I View Map



### II.II View Food



## II.III Search Food

The screenshots illustrate the search flow:

- Screenshot 1:** Shows a search bar with "Mi" typed in, a "Filter" button, and a "RECENT SEARCHES" section with "pasta" and "japanese". A virtual keyboard is at the bottom.
- Screenshot 2:** Shows the search results for "Mixed Rice" with four items, each featuring a photo, name, rating (4.8), cuisine type (Non-halal Chinese), and price range (SGD 3.80-18.30). A "Food" tab is highlighted in the navigation bar.
- Screenshot 3:** Shows the search results for "Canteen 11" with four items, identical to the ones in Screenshot 2. A "Food" tab is also highlighted here.

## II.IV Filter Food

The screenshot shows the filter interface with the following sections:

- Category:** Buttons for BEEF, HALAL, VEGETARIAN, NON-HALAL, VEGAN, and NON-BEEF. The NON-HALAL button has a red badge with the number 1.
- Price:** A text input field.
- Cuisines:** A dropdown menu.
- Discountes:** A dropdown menu.
- Sort by Filter:** A section titled "Non-Halal" with three items, each with a photo, name, rating (4.8), cuisine type (Non-halal Chinese), and price range (SGD 3.80-18.30).

A large red "Apply Filters" button is at the bottom, and a "Food" tab is highlighted in the navigation bar.

## II.V Submit Food Review

The figure consists of three vertically aligned screenshots from a mobile application interface.

**Screenshot 1:** Shows the initial review form. It includes a header with signal strength, battery level, and time (9:41). Below is a red 'X' button, followed by a placeholder image icon. The main form has fields for rating (3.5 stars), store name ('Store' dropdown), food item ('What did you have?'), price ('Price' dropdown), and a picture upload section ('Picture (Optional)'). At the bottom are 'Leave A Review' and 'See more' buttons.

**Screenshot 2:** Shows the review being submitted. The header now shows 'Review'. The rating is set to 3.5 stars. The store dropdown contains 'Store Name'. The food input field contains 'Waffl'. The price dropdown contains 'SGD'. The picture section says 'Click to browse or Upload your files here'. At the bottom is a large red 'Submit Review' button.

**Screenshot 3:** Shows the review submission confirmation. The header shows 'Review Submitted!' with a green checkmark. Below is a list of reviews from 'Can11 菜饭' (4.2 stars). Each review includes a profile picture, name, rating, location, and a truncated comment. The bottom navigation bar includes 'Explore', 'Map', a red 'Food' tab, and 'Profile'.

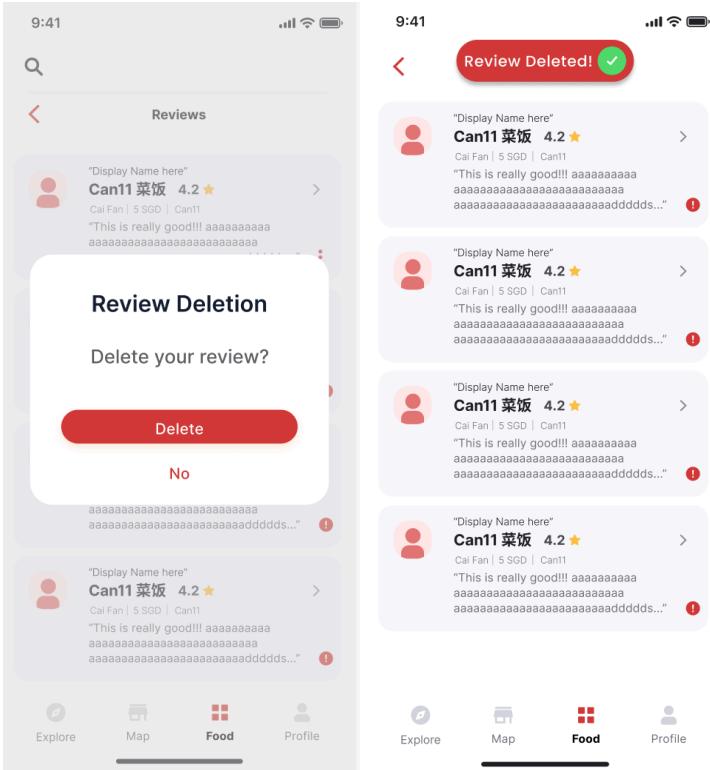
## II.VI Update Food Review

The figure consists of two vertically aligned screenshots from a mobile application interface.

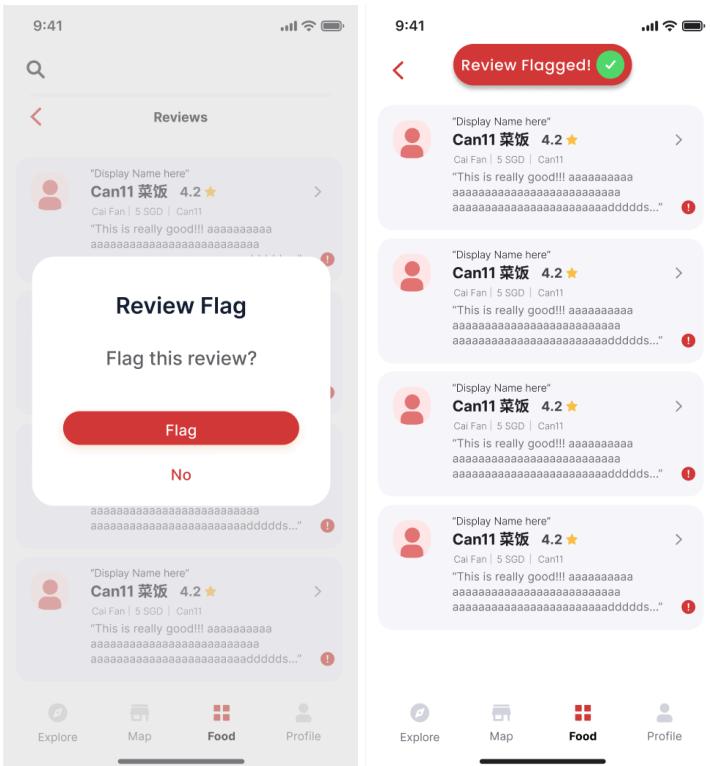
**Screenshot 1:** Shows the review edit screen. The header says 'Review Edit'. It includes a red 'X' button and a placeholder image icon. The main form is identical to the submission screen: rating (3.5 stars), store name ('Store' dropdown), food item ('What did you have?'), price ('Price' dropdown), and a picture upload section ('Picture (Optional)'). At the bottom is a red 'Save Edit' button.

**Screenshot 2:** Shows the edited review confirmation. The header shows 'Review Edited!' with a green checkmark. Below is a list of reviews from 'Can11 菜饭' (4.2 stars). Each review includes a profile picture, name, rating, location, and a truncated comment. The bottom navigation bar includes 'Explore', 'Map', a red 'Food' tab, and 'Profile'.

## II.VII Delete Food Review

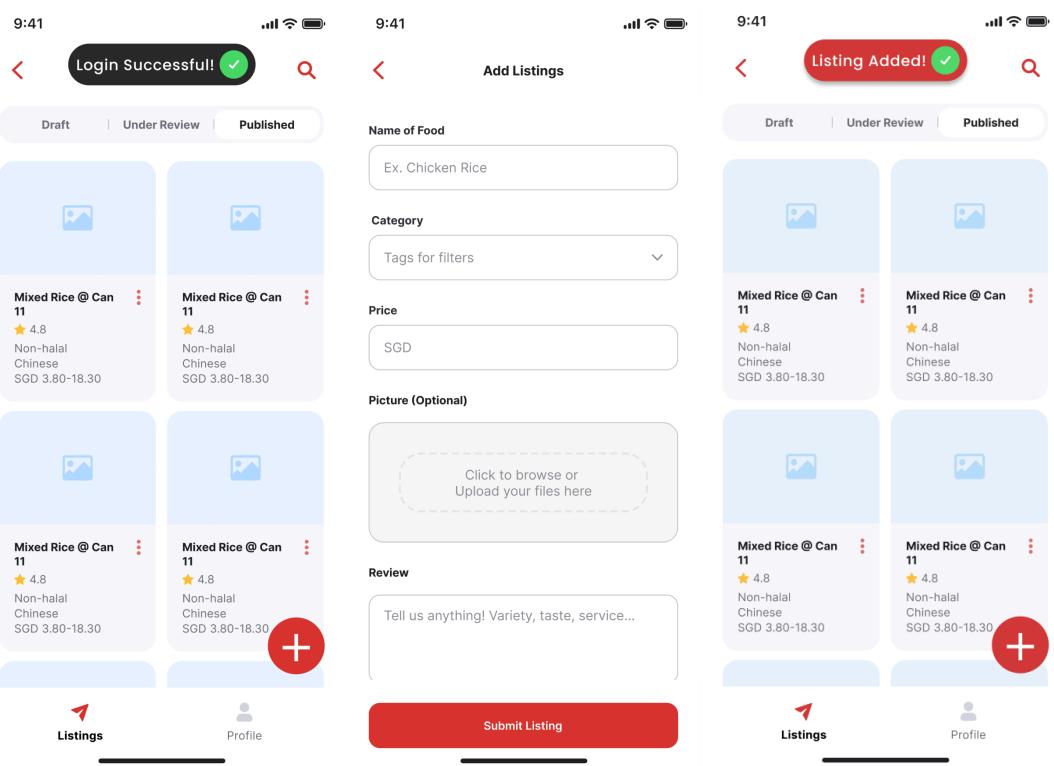


## II.VIII Flag Food Review

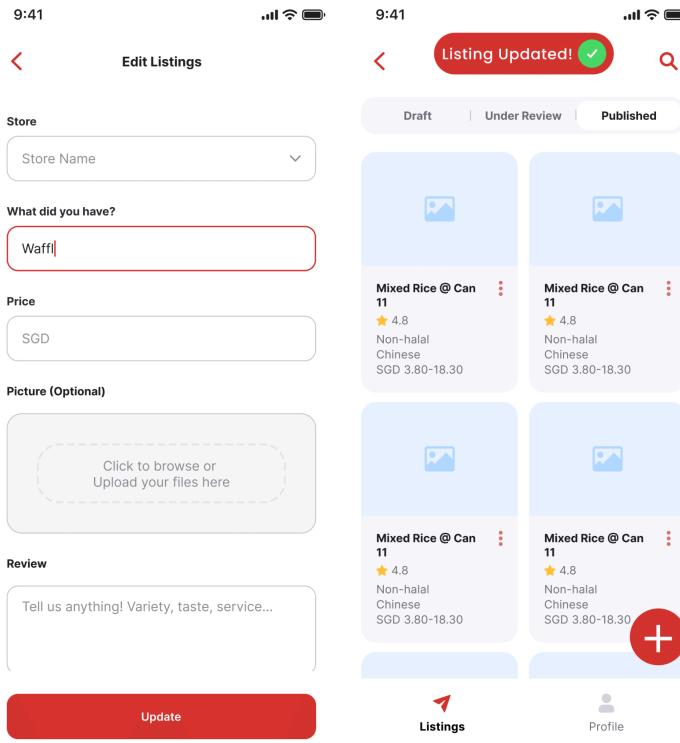


## III Hawker User Interface

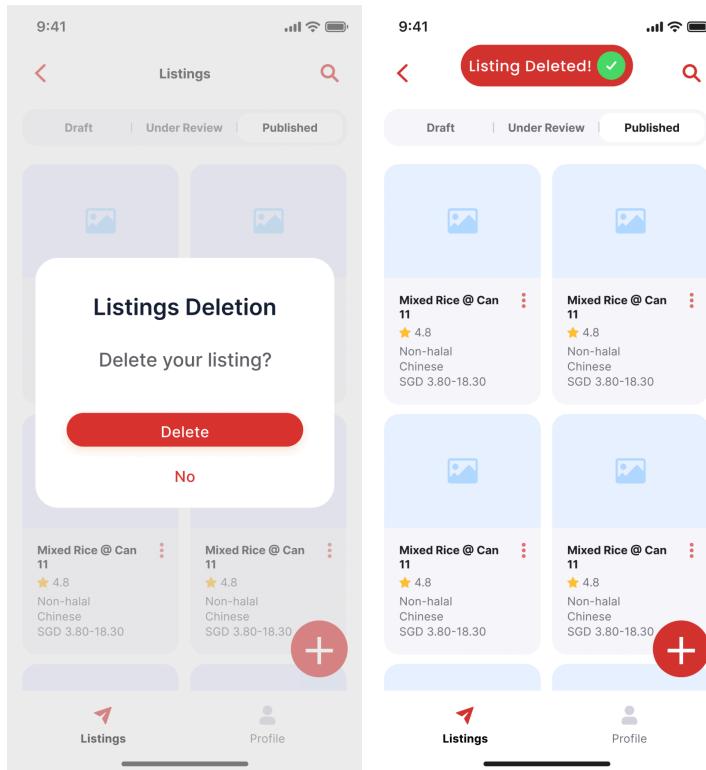
### III.I Submit Food Listing



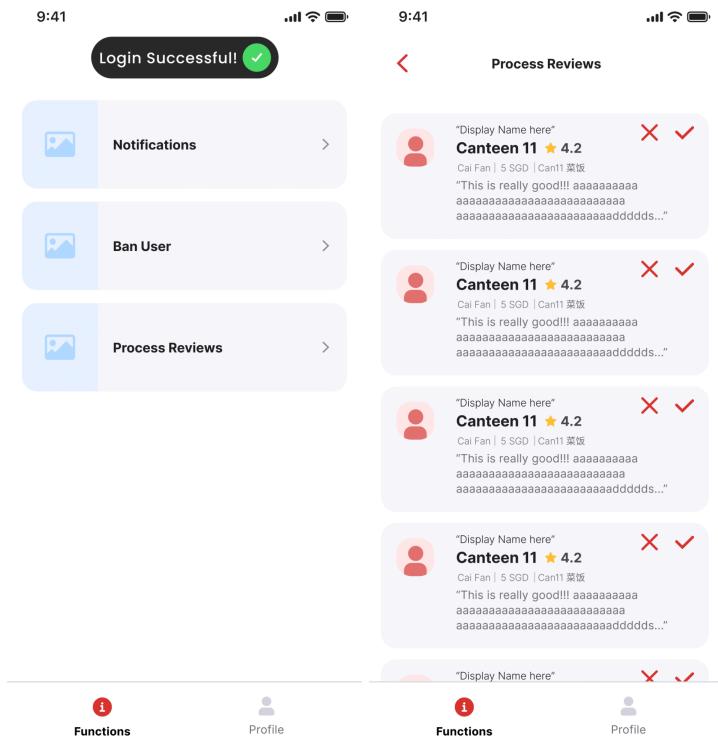
### III.II Update Food Listing



### III.III Delete Food Review



### III.IV Process Review



### III.V Ban User

The screenshots show a mobile application interface for managing users. The top row shows the initial login screen and the 'Ban User' search screen. The bottom row shows the confirmation screen after a user has been banned.

**Screenshot 1 (Top Left):** Shows a 'Login Successful!' message at the top. Below it are three navigation items: 'Notifications', 'Ban User', and 'Process Reviews'. Each item has a small image icon and a right-pointing arrow.

**Screenshot 2 (Top Middle):** Shows a 'Ban User' search screen with a magnifying glass icon at the top right. Below is a list of users with 'Display' and 'email' information and a 'Ban' button.

**Screenshot 3 (Top Right):** Shows a 'Ban User' confirmation dialog. It asks 'Do you want to ban this user?' with 'Ban' and 'No' buttons. The background shows the same list of users as the previous screen.

**Screenshot 4 (Bottom Left):** Shows a 'User Banned!' message at the top. Below is the same list of users as the other screens, with the 'Ban' button now highlighted in red.

**Screenshot 5 (Bottom Middle):** Shows the same list of users as the other screens, with the 'Ban' button now highlighted in red.

**Screenshot 6 (Bottom Right):** Shows the same list of users as the other screens, with the 'Ban' button now highlighted in red.

### III.VI Notify User

