

Lab 1

1. Overview

This document will serve as a summary of Lab 1. The summary includes the expected deliverables of the lab, as well as the detailed write-up of the respective deliverables.

2. Mission Statement

iHawker app is designed to be an all in one application for Singaporeans looking for information about hawker food. The app will allow users to search for hawker centres and food stalls based on their location, as well as read reviews and ratings from other users. This will make it easy for users to find the best hawker food in their area, and to make informed decisions about where to eat. In addition, the app enables hawker stall owners to register their establishments, making it visible to all app users, thereby increasing visibility and customer reach.

Furthermore, users will be able to share their own experiences and recommendations, which will help to build a community of food enthusiasts. This way, they could also provide feedback on the food taste, price, hygiene, and other important factors. The app will also include features such as healthy food options, and budget-friendly options, making it easy for users to find food that fits their dietary and financial needs.

3. Functional Requirements

The following are the proposed preliminary FR:

3.1 The web application has a signup and login page

3.1.1 Users can create an account and login to our system

3.1.2 Users can continue as guest but with limited available features

3.1.2.1 Guests can search and view all the information of hawker centres and stalls

3.1.2.2 Guests cannot leave a rating or a comment

3.1.2.3 Guests cannot add hawker stalls

3.2 The web application has a homepage which showcases an overview of its features

3.2.1 The homepage must display nearby hawker centres and stalls based on the user's location (km radius?)

3.2.2 The homepage must show recommended food options by other users based on the user's location

3.2.3 The homepage must show popular stalls and recent comments posted by other users

Lab 1

3.3 The web application has a search bar

3.3.1 Users can filter results based on choice of location, distance, rating, food and price

3.3.2 The search functionality will suggest the top ranked options based on the input entered by the user, to provide the most relevant and popular search suggestions

3.4 The web application has a map

3.4.1 Google map API will be used to display the hawker centres spread out across Singapore

3.4.2 Each hawker centre on the map is represented by a circular marker

3.4.3 Users can click on a marker to view additional information about the hawker centre

3.5 The web application provides users with the ability to obtain directions to the selected hawker centres

3.5.1 The website will display the most efficient route for travelling by car, public transportation, and on foot

3.6 The web application provides users with the ability to leave a rating and a comment to share information about food or food stalls

3.6.1 Users can write text and upload photos and videos as in their comment

3.6.1.1 Text descriptions must be at most 150 characters

3.6.1.2 At most 5 photos shall be uploaded

3.6.1.3 Each photo must not exceed 25 MB

3.6.1.4 At most 3 videos shall be uploaded

3.6.1.5 Each video must not exceed 7 seconds

3.7 The web application provides users (hawkers) with the ability to add a hawker stall

3.7.1 The stall added must include the location of the stall

3.7.2 The stall added must include the menu of the stall

4. Non-Functional Requirements

4.1. Usability

4.1.1. All features of the application will only be supported in the English Language.

4.1.2. Intuitive for user (5 secs)

4.2. Performance

4.2.1. Each page on the web application must load in less than 5 seconds

4.2.2. Search results must be available in less than 5 seconds

4.2.3. Recommendations of food and hawker stalls must be available in less than 5 seconds

4.3. Security

4.3.1. The password should be a length of at least 8 characters with at least an upper-case letter, a lower-case letter and a digit

4.4. Portability

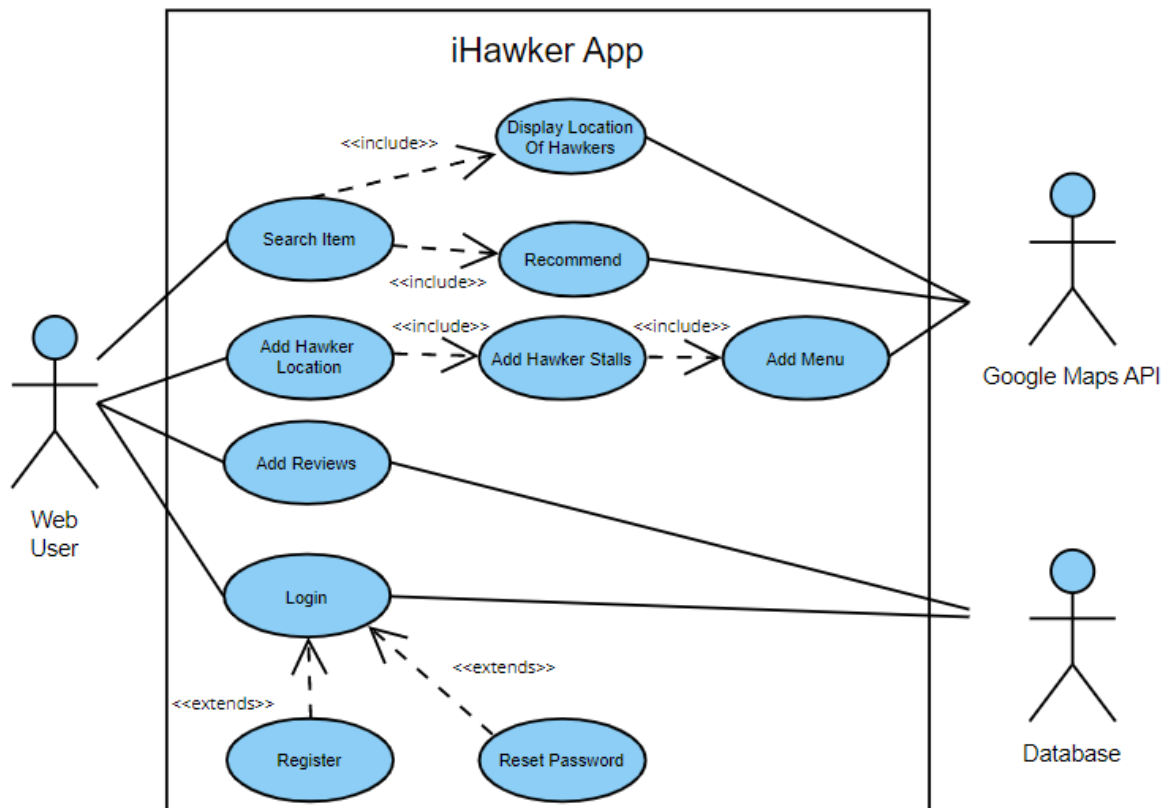
4.4.1. The web application must be able to work in any browser

4.5. Reliability

4.5.1. The web application must be able to provide accurate information to the users

Lab 1

5. Use Case Diagram



Lab 1

6. Use Case Description

Use Case ID:	001		
Use Case Name:	Register		
Created By:	Chan Kin Leong	Last Updated By:	Chan Kin Leong
Date Created:	25 th January 2023	Date Last Updated:	25 th January 2023
Actor:	Web User (Initiating), Database		
Description:	The Web User can register for an account that is stored in the Database using this use case		
Preconditions:	<ol style="list-style-type: none">1. The Database must be up and online.2. The Web User must be connected to the internet.		
Postconditions:	The Web User has successfully registered an account with a unique username and password. Or The Web User is notified of the reason(s) why the registration of an account is unsuccessful.		
Priority:			
Frequency of Use:			
Flow of Events:	<ol style="list-style-type: none">1. At the home page of the website, the Web User clicks on “Sign up” and is redirected to the registration page.2. The Web User inputs a valid email, a username, a password that contains at least an upper-case letter, a lower-case letter and a digit, and the repeated password into the respective fields in the submission form.3. The Web User clicks on “Sign Up”.4. The system verifies the username is unique and the password satisfies the constraints.5. The system stores the Web User’s information in the database securely.6. The Web User is notified that the registration is successful.		
Alternative Flows:	<u>AF-1: The Web User left input field(s) blank.</u> <ol style="list-style-type: none">1. When the Web User clicks on “Sign Up”, the system displays the message “Please ensure all fields have been filled up before submitting!” above the submission form.2. The system returns to Step 2 and waits for the Web User inputs. <u>AF-2: The Web User inputs a taken username.</u> <ol style="list-style-type: none">1. The system displays the message “Username has been taken. Please try again!” above the submission form.		

Lab 1

	<p>2. The system returns to Step 2 and waits for the App User inputs.</p> <p><u>AF-3: The Web User inputs a password that do not satisfy the given requirements.</u></p> <p>1. The system displays the message “Password does not meet the required standards” above the submission form.</p> <p>2. The system returns to Step 2 and waits for the Web User inputs.</p> <p><u>AF-4: The Web User inputs mismatched passwords.</u></p> <p>1. The system displays the message “Passwords do not match” above the submission form.</p> <p>2. The system returns to Step 2 and waits for the Web User inputs.</p>
Exception:	<p><u>EX-1: The Web User repeatedly attempts to register for an account for more than five times despite errors in input.</u></p> <p>1. On the 5th attempt, when the Web User clicks on “Sign Up”, the system displays the message “Too many attempts! Please try again in 10 minutes.” above the submission form.</p> <p>2. The “Sign Up” button is unavailable for ten minutes.</p> <p>3. The system only accepts registration from the Web User’s IP address after ten minutes.</p>
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Lab 1

Use Case ID:	002		
Use Case Name:	Login		
Created By:	Chan Kin Leong	Last Updated By:	Chan Kin Leong
Date Created:	25 th January 2023	Date Last Updated:	25 th January 2023
Actor:	Web User (Initiating), Database		
Description:	The Web User can login to his/her account with the correct credentials that are stored securely in the Database.		
Preconditions:	<ol style="list-style-type: none">1. The Database must be up and online.2. The Web User must be connected to the internet.3. The Web User has a registered account.		
Postconditions:	The Web User has successfully logged into his/her web account. Or The Web User is notified of the reason(s) why he/she is unable to login to his/her account.		
Priority:			
Frequency of Use:			
Flow of Events:	<ol style="list-style-type: none">1. At the home page of the website, the Web User clicks on “Log in” and is redirected to the registration page.2. The Web User inputs his/her username and password.3. The Web User clicks on “Log In”.4. The system verifies the credentials provided with the database.5. When the information is verified, the Web User is redirected to his/her account home page.		
Alternative Flows:	<u>AF-1: The Web User inputs an incorrect username or password</u> <ol style="list-style-type: none">1. When the Web User clicks on “Log In”, the system displays the message “Invalid username and/or password!” above the submission form.2. The system returns to Step 2 and waits for the Web User inputs. <u>AF-2: The Web User left input field(s) blank.</u> <ol style="list-style-type: none">1. When the Web User clicks on “Log In”, the system displays the message “Please ensure all fields have been filled up before submitting!” above the submission form.2. The system returns to Step 2 and waits for the Web User inputs.		
Exception:	<u>EX-1: The Web User inputs incorrect username or password for more than five times</u> <ol style="list-style-type: none">1. On the sixth attempt at logging in, when the Web User clicks on “Log In”, the system displays the message “Account suspended.		

Lab 1

	<p>Please try again after 10 minutes” above the submission form.</p> <ol style="list-style-type: none">2. The “Log In” button is unavailable for ten minutes.3. The system only accepts registration from the Web User’s IP address after ten minutes. <p><u>EX-2: The Web User forgot his/her login credentials</u></p> <ol style="list-style-type: none">1. The Web User clicks on “Forget Password?” on the login page.2. The Web User can recover his/her account using the extended use case LostAccountHelp.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Lab 1

Use Case ID:	003		
Use Case Name:	LostAccountHelp		
Created By:	Chan Kin Leong	Last Updated By:	Chan Kin Leong
Date Created:	25 th January 2023	Date Last Updated:	25 th January 2023
Actor:	Web User (Initiating), Database		
Description:	The Web User can request for help if he/she lost access to his/her account using this use case.		
Preconditions:	<ol style="list-style-type: none"> 1. The Database must be up and online. 2. The Web User must be connected to the Internet. 3. The Web User has forgotten his/her login credentials. 		
Postconditions:	<p>The Web User has successfully recovered his/her account by changing his/her credentials.</p> <p>Or</p> <p>The Web User has contacted support to seek further assistance.</p>		
Priority:			
Frequency of Use:			
Flow of Events:	<ol style="list-style-type: none"> 1. The Web User clicks on “Forgotten?” on the login page. 2. The system displays the recover account page. 3. The Web User inputs his/her registered email and clicks on “Recover Account”. 4. The system displays a “Security Check” submission form. 5. The Web User inputs the one-time password (OTP) that has been sent to his/her email inbox. 6. The system displays a “Change Security Details” submission form. 7. The Web User inputs a new set of username and password and clicks on “Change”. 8. The system verifies that the username is unique, and the password satisfies the given requirements before updating the Web User’s information in the database securely. 9. The Web User is informed of the successful change in credentials and is redirected back to the login page. 		
Alternative Flows:	<p><u>AF-1: The Web User entered an incorrect, but registered email address</u></p> <ol style="list-style-type: none"> 1. When the Web User realises that he/she has inputted an incorrect email address, the App User clicks on “Not email@serviceprovider.com?”. 2. The system returns to Step 3 and waits for the Web User inputs. 		

Lab 1

	<u>AF-2: The Web User entered an incorrect, and not registered email address</u> 1. When the Web User clicks on “Recover Account”, the system displays the message “Email not registered!” above the submission form. 2. The system returns to Step 3 and waits for the Web User inputs
Exception:	<u>EX-1: The Web User forgot his/her registered email.</u> 1. The Web User clicks on “Contact Support”. 2. The system redirects the Web User to the FAQ page that contains the support email address. 3. The Web User contacts the support via email to retrieve access of his/her account.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Lab 1

Use Case ID:	004		
Use Case Name:	SearchHawkerCatalog		
Created By:	Chan Kin Leong	Last Updated By:	Chan Kin Leong
Date Created:	25 th January 2023	Date Last Updated:	25 th January 2023
Actor:	Web User (Initiating), Database		
Description:	The Web User searches the database of hawker foods.		
Preconditions:	<ol style="list-style-type: none">1. User must be connected to WiFi/Mobile Data2. User must be on the “Homepage” or “Search page”		
Postconditions:	The Web User can view results pertaining to his/her search, with information on related hawker stalls such as Location and Price. Or The Web User can save the hawker stall to their favourites list.		
Priority:	Medium		
Frequency of Use:	High		
Flow of Events:	<ol style="list-style-type: none">1. At the home page of the website, the Web User clicks on search bar and makes a query.2. System checks user query against database.3. Information relating to the query is presented.		
Alternative Flows:	<u>AF-1: The Web User searches via the Search page</u> <ol style="list-style-type: none">1. When the Web User clicks on “Search” page for his/her query.2. The system returns to Step 2 and waits for the Web User inputs. <u>AF-2: The Web User inputs nothing and clicks on the Search icon</u> <ol style="list-style-type: none">1. The system displays a list of 10 random items.2. The system returns to Step 1 and waits for the Web User to input another keyword.		
Exception:	<u>EX-1: The system is unable to retrieve any items based on the keyword</u> <ol style="list-style-type: none">1. The system will not display any search results.2. The system will display a message “No results found! Sorry we cannot find any results for your search item.” instead.3. The system will return to Step 1 and waits for the Web User to input another keyword.		
Includes:	<u>Search Filter</u> <ol style="list-style-type: none">1. The Web User may filter search results by various metrics<ol style="list-style-type: none">a. Locationb. Pricec. Cuisine Type		

Lab 1

	<u>Display Hawker Location</u> <ol style="list-style-type: none">1. Using Google Maps API to present map of nearby hawkers2. Display optimal route to nearby hawkers<ol style="list-style-type: none">a. On footb. By public transportc. By vehicle
Special Requirements:	
Assumptions:	Database has hawker stalls to query from
Notes and Issues:	

Lab 1

Use Case ID:	005		
Use Case Name:	RateHawkerCentres		
Created By:	Chan Kin Leong	Last Updated By:	Chan Kin Leong
Date Created:	25 th January 2023	Date Last Updated:	25 th January 2023
Actor:	Web User (Initiating), Database		
Description:	The Web User rates the food from the hawker centres out of 5 stars after they have eaten the food. The Web User can upload photos of their food to share with other users.		
Preconditions:	<ol style="list-style-type: none">1. User must be connected to WiFi/Mobile Data2. User must have an account3. User must be logged in to their account		
Postconditions:	The Web User adds a food review to the database. Or Other Web User can access and view the review made by other Web User.		
Priority:	Low		
Frequency of Use:	High		
Flow of Events:	<ol style="list-style-type: none">1. User logs into their account2. User clicks on “Review” button3. User selects the location that they purchased food from4. User selects a rating from 1 to 5<ol style="list-style-type: none">a. 1 representing a bad ratingb. 5 representing a good rating5. User shares their review6. Review is saved in the database		
Alternative Flows:	<u>AF-1: The Web User shares a photo with their review</u> <ol style="list-style-type: none">1. The Web User is prompted with an upload request.2. The Web User can upload photos of their food.3. The system will return to Step 5.		
Exception:			
Includes:			
Special Requirements:			
Assumptions:	<ol style="list-style-type: none">1. User has a file manager2. User device has photos		

Lab 1

	a. Or device is capable of taking photos
Notes and Issues:	

Lab 1

Use Case ID:	006		
Use Case Name:	CommentHawkerCentres		
Created By:	Chan Kin Leong	Last Updated By:	Chan Kin Leong
Date Created:	25 th January 2023	Date Last Updated:	25 th January 2023
Actor:	Web User (Initiating), Database		
Description:	The Web User leave comments on other food reviews.		
Preconditions:	<ol style="list-style-type: none">1. User must be connected to WiFi/Mobile Data2. User must have an account3. User must be logged in to their account		
Postconditions:	The Web User can leave comments on other food reviews. Or Other Web Users can view and leave comments of their own.		
Priority:	Medium		
Frequency of Use:	High		
Flow of Events:	<ol style="list-style-type: none">1. User logs into their account2. User selects "Review" page3. User is presented with reviews submitted by other users4. User selects a review5. User clicks on the "Comment" button6. User writes a comment7. Comments can contain text and emoticons8. User submits the comment9. Comment is saved in the database and tagged to the review		
Alternative Flows:	<u>AF-1: Comment contains offensive words</u> <ol style="list-style-type: none">1. Comments that contain offensive words will be flagged2. User will be prompted to change the choice of words used3. The system will return to Step 7		
Exception:			
Includes:			
Special Requirements:			
Assumptions:	<ol style="list-style-type: none">1. Database has reviews to query from<ol style="list-style-type: none">a. Submitted by other users		
Notes and Issues:			

7. Data Dictionary

Created By:	Aloysius Tan	Last Updated By:	Aloysius Tan
Date Created:	18th January 2023	Date Last Updated:	25th January 2023
User	An individual who holds an account with the iHawker web application. The account must be formally registered via the iHawker web application and must be retrievable from the Database. The user is then entitled to use all the services provided within the web application, which includes but not limited to, searching for hawker stalls and recommended food and adding Hawker and its stalls, subject to the Terms of Use which the user has agreed upon registration. An User can be either a customer or Hawker center stall owner.		
One-Time Password (OTP)	A six-digit combination which is sent to the User via his/her registered email address. The OTP service serve as an additional layer of security in the event where an User forgets his/her login credentials. The OTP sent will expire within minutes. An User may request for an additional OTP after every 60 seconds.		
Username	A unique identification set up by the User which serves as a locator of him/her.		
Database	An online spreadsheet which contains all information of each User such as their username, hashed registered email address, hashed password as well as Hawkers, which includes the stalls, menu items and ratings.		
Hawker Centre	The physical location of the hawker centre and contains the stalls located inside the hawker center and their menu items.		
Rating	A scaling system implemented by the web application for Users to rate the quality of service in the Hawker. The scale of system is from one to five star, where one star represents the poorest experience and five star represent the best experience. The rating system will display the average of all ratings given by the Users.		
Comment	Written remark expressing an opinion or reaction of a hawker food stall by the user		
Category	Type of food		
API (Google Map)			