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. Deliverables

The following defines the deliverables to be submitted to the Lab TA prior to Lab 2:

2.1 Team formation and team leader assignment

The team will choose an appropriate team name. The team will also appoint a member of the team as the team leader. The team must fill in the Team Registration Form as instructed by Lab TA.

2.2 Documentation of Functional Requirements (FR) and Non-Functional Requirements (NFR)

The team will proceed to brainstorm a project idea which will serve as guidance for the team project. The team will also formulate the target audience of the project. The team will proceed to liaise with the stakeholders to elicit FR and NFR. The team will then formulate a documentation which contains the FR and NFR.

The team will use SRS_Template.doc, under sections *4. System Features* and *5. Other Nonfunctional Requirements* as references for the template.

2.3 Data Dictionary

The team will create a data dictionary which includes important terms used throughout the project. The team will ensure that the data dictionary contains attributes of, and the relationship between each term.

2.4 Use Case Models

The team will identify certain use cases based off the FR. The team will proceed to depict the use cases using a Use Case Diagram.

The team shall make sure that each use case is accompanied by a use case description. The description will explain the interaction between an end-user and the system to carry out a functionality.

2.5 User Interface (UI) Mock-ups

The team will use relevant software tools, or hand-drawn methods (if applicable) produce a mock-up of the project's preliminary user interface.

3. Project Ideas

The following are the proposed project ideas:

3.1 Carpark availability application

The web application will provide a heatmap of the carparks around Singapore. The heatmap will depict the number of available parking spots in each carpark.

3.1.1 Feedbacks

The complexity of the initial project idea is not sufficient to demonstrate the work of a 5-man project.

3.2 Taxi availability application

The web application will provide a heatmap of the live location of all taxi around Singapore.

3.2.1 Feedbacks

A web application for taxi hailing may not be appropriate. The team considered that a taxi hailing application should be a mobile application instead

3.3 COVID-19 cases heatmap

The web application will provide a heatmap of the live COVID-19 cases within Singapore.

3.3.1 Feedbacks

The team agrees that the project idea is overused and lacks innovation.

3.4 Dengue fever heatmap

The web application will provide a heatmap of the live Dengue fever cases within Singapore.

3.4.1 Feedbacks

The team agrees that the project idea is overused and lacks innovation.

3.5 Music recommendation web application

The web application will allow the user to compile a list of favourite music. The list shall then be used to recommend the user other music of which the system deem the user may be interested in.

3.5.1 Feedbacks

A web application that allows a user to compile a list of music is redundant, as ordinary music players such as Spotify contains said features. The project idea can retain as a supplementary feature and should be built on top of an alternate project idea.

3.6 AI Shopping Tracking System

The web application will track the user's search pattern. Based on the search pattern, the system will recommend the user a list of items that the system predicts the user may be interested in. The web application will also perform a cross-platform check to find the best deals of the items. The best deal includes the cheapest price and rebates, whichever is applicable.

The team has decided to select *3.6 AI Shopping Tracking System* as the project idea.

4. Team Name

The following are the proposed team names:

4.1 FindR

The name *FindR* mimics the pronunciation of *Finder* and provides a catchy feeling to the customers.

Since there are no alternate suggestions or proposals, the team has unanimously decided to select *4.1 FindR* as the team's name.

5. Target Audience

Based on the selected project idea, the following characteristics shall define the target audience:

5.1 Users who lack time to perform shopping physically.

We are targeting workers who have packed schedules. We strongly believe the AI shopping recommendation system will help reduce time taken to shop for goods. The cross-platform comparison feature will also help to save the time needed to navigate through multiple stores or platforms to find the best deals.

5.2 Users who live far away from physical convenience stores.

We are targeting customers whose location are inconvenient for physical shopping. An online AI shopping recommendation system will aid the customers in their daily online shopping. The cross-platform comparison feature will further serve to aid the customers in finding the best deals of their everyday online shopping.

5.3 Users who are home-bound or have mobility issues.

We are targeting elderlies or disabled customers. The customers could already be engaged with online shopping activities. Thus, an online AI shopping recommendation system with cross-platform comparison feature will certainly aid the customers to make better and informed decisions.

5.4 Tech-savvy users.

We are targeting young adults who are mostly familiar with navigating the online world. Our online AI shopping recommendation system will aid the customers to make informed decisions in their purchases.

6. Functional Requirements (FR)

The following are the proposed preliminary FR:

6.1 The user must be able to register for an account with our system and login subsequently.

6.2 If the user has forgotten his/her login credentials, he/she must be able to seek help to recover the lost account.

6.2 When the user searches for an item, our system must be able to recommend at least three other items to the user.

6.3 The system must be able to retrieve the prices of the searched items from at least one e-commerce platform.

6.4 The system must be able to retrieve at least one relevant rebate of the searched item.

6.5 The system must be able to provide a set of parameters such as price, number of purchases, form of rebate, delivery fee, payment methods and rating which can be tweaked by the user to sort the result.

6.6 The system must be able to provide the best deal for the user according to the parameters set by the user.

6.7 The user must be able to compile a wish list of items which must be sold on at least one e-commerce platform.

6.8 The user must be able to send another user a friend request using their usernames.

6.9 The user must be able to accept or reject the friend request received.

6.10 The user must be able to view his/her friends' wish list and their respective birthdays.

7. Non-Functional Requirements (NFR)

The following are the proposed preliminary NFR:

7.1 The system must be able to successfully register for an account for the user after the user fills in all the details required within 15 seconds.

7.2 The system must be able to retrieve and display the search result of the user within 30 seconds.

7.3 The system must be able to support searches from at least three e-commerce platform.

7.4 The system must not be down for more than three hours in one year.

7.5 The system must encrypt the user's credential information using AES algorithm.

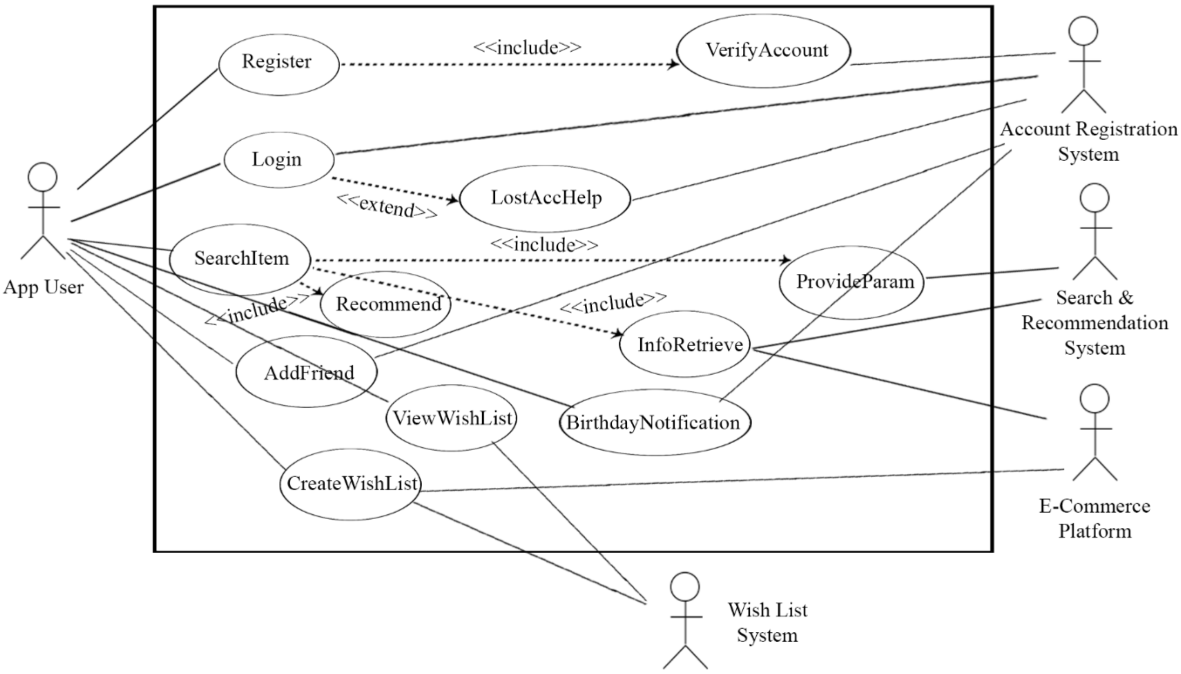
7.6 The system must be able to display help information in the local language of the user based on the user's location.

8. Data Dictionary

Created By:	Lee Juin	Last Updated By:	Lee Juin
Date Created:	19th August 2022	Date Last Updated:	28th August 2022

Voucher	An online code that entitles the holder to a discount, or that may be exchanged for goods or services.
Cashback	A form of incentive offered to buyers of certain products whereby they receive a cash refund after making their purchase.
Rebate	A form of discount applied to a product sold in the form of cashback or voucher.
Wish list	A list of desired items by the user which are available on an e-commerce platform.
E-commerce platform	An online platform where sellers advertise and sell their goods to consumers.
Price	The amount of money expected, required, or given in payment for the item sold.
Delivery fee	The cost of transporting or delivering goods.
Payment methods	A method for customers to pay for a product or a service.
App User	An individual who holds a valid account with the FindR web application.

9. Use Case Diagrams



10. Use Case Description

Use Case ID:	001		
Use Case Name:	Register		
Created By:	Lee Juin	Last Updated By:	Lee Juin
Date Created:	25th October 2022	Date Last Updated:	29th October 2022

Actor:	App User (Initiating), Account Registration System
Description:	The App User can register for an account with the Account Registration System using this use case.
Preconditions:	1. The Account Registration System must be up and online. 2. The App User must be connected to the Internet.
Postconditions:	The App User has successfully registered an account for the web application with a unique username and password. *Or* The App User is notified of the reason(s) why the registration of an account is unsuccessful.
Priority:	
Frequency of Use:	
Flow of Events:	1. The App User connects to the web application through the Internet. 2. The App User clicks on “Sign up”. 3. The Account Registration System displays the registration page that contains a “Sign Up” submission form. 4. The App User inputs a valid email, a username, a password that satisfies the given constraint and the repeated password into the respective fields in the submission form. 5. The App User checks the tick box of “I agree to the Terms of Use and Privacy Policy”. 6. The App User clicks on “Sign Up”. 7. The Account Registration System verifies the information provided using the included use case *VerifyAccount.* 8. When the information is verified, the App User is notified of the successful registration of his/her account via email and notification on the website.
Alternative Flows:	AF-1: The App User did not check the tick box of “I agree to the Terms of Use and Privacy Policy”. 1. When the App User clicks on “Sign Up”, the Account Registration System displays the registration page again. 2. The registration page further contains the message “Please tick the checkbox for acknowledging our Terms of Use and Privacy Policy!” above the “Sign Up” submission form. 3. The Account Registration System returns to Step 4 and waits for the App User inputs. AF-2: The App User left input field(s) blank 1. When the App User clicks on “Sign Up”, the Account Registration System displays the registration page again. 2. The registration page further contains the message “Please ensure all fields have been filled up before submitting!” above the “Sign Up” submission form.

	3. The Account Registration System returns to Step 4 and waits for the App User inputs.
Exceptions:	EX-1: The App User repeatedly attempts to register for an account for more than ten times despite errors in input 1. On the 11th failed attempt at registering for an account by the App User, when the App User clicks on “Sign Up”, the Account Registration System displays the registration page again. 2. The registration page further displays the message “Too many attempts! Please try again in 10 minutes.” above the “Sign Up” submission form. 3. The “Sign Up” button is unavailable for ten minutes. 4. The Account Registration System only accepts registration from the App User’s IP address after ten minutes.
Includes:	*VerifyAccount*
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	002		
Use Case Name:	Login		
Created By:	Lee Juin	Last Updated By:	Lee Juin
Date Created:	25th October 2022	Date Last Updated:	29th October 2022

Actor:	App User (Initiating), Account Registration System
Description:	The App User can login to his/her account with the correct credentials.
Preconditions:	1. The Account Registration System must be up and online. 2. The App User must be connected to the Internet. 3. The App User has a registered account with the Account Registration System.
Postconditions:	The App User has successfully logged into his/her application account. *Or* The App 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:	1. The App User connects to the web application through the Internet. 2. The App User clicks on “Login”. 3. The Account Registration System displays the login page that contains a “Login” submission form. 4. The App User inputs his/her username and password. 5. The App User clicks on “LOGIN”. 6. The Account Registration System verifies the credentials provided. 7. When the information is verified, the App User is redirected to his/her account dashboard.
Alternative Flows:	<u>AF-1: If the App User inputs an incorrect username or password</u> 1. When the App User clicks on “LOGIN”, the Account Registration System displays the login page again. 2. The login page further displays the message “Invalid username and/or password!” above the “Login” submission form. 3. The Account Registration System returns to Step 4 and waits for the App User inputs. <u>AF-2: The App User left input field(s) blank</u> 1. When the App User clicks on “LOGIN”, the Account Registration System displays the login page again. 2. The login page further displays the message “Please ensure all fields have been filled up before submitting!” above the “Login” submission form. 3. The Account Registration System returns to Step 4 and waits for the App User inputs.
Exceptions:	<u>EX-1: The App User inputs incorrect username or password for more than five times</u> 1. On the sixth failed attempt at logging in, when the App User clicks on “LOGIN”, the Account Registration System displays the login page again. 2. The login page further displays the message “Account suspended. Please try again after 10 minutes” above the “Login” submission form. 3. The “LOGIN” button is unavailable for ten minutes.

	4. The Account Registration System only accepts registration from the App User's IP address after ten minutes. EX-2: The App User forgot his/her login credentials 1. The App User clicks on "Forget Password?" on the login page. 2. The App User can recover his/her account using the extended use case *LostAccHelp*.
Includes:	
Extends:	*LostAccHelp*
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	003		
Use Case Name:	LostAccHelp		
Created By:	Lee Juin	Last Updated By:	Lee Juin
Date Created:	25th October 2022	Date Last Updated:	29th October 2022

Actor:	App User (Initiating), Account Registration System
Description:	The App User can request for help if he/she lost access to his/her account.
Preconditions:	1. The Account Registration System must be up and online. 2. The App User must be connected to the Internet. 3. The App User has forgotten his/her login credentials.
Postconditions:	The App User has successfully recovered his/her account by changing his/her credentials. *Or* The App User has contacted support to seek further assistance.
Priority:	
Frequency of Use:	
Flow of Events:	1. The App User clicks on "Forgotten?" on the login page. 2. The Account Registration System displays the recover account page with a "Help" submission form. 3. The App User inputs his/her registered email and clicks on "Recover Account". 4. The Account Registration System displays a "Security Check" submission form. 5. The App User inputs the one-time password (OTP) that has been sent to his/her email inbox. 6. The Account Registration System displays a "Change Security Details" submission form. 7. The App User inputs a new set of username and password and clicks on "Change". 8. The Account Registration System verifies the information provided. 9. The App User is redirected back to the login page.
Alternative Flows:	<u>AF-1: The App User did not receive the OTP in his/her email inbox</u> 1. The "Resend another OTP" button is available after 60 seconds. 2. The App User clicks on the "Resend another OTP" button to resend another OTP to his/her email inbox. 3. The Account Registration System returns to Step 6 and waits for the App User inputs. <u>AF-2: The App User entered an incorrect, but registered email address</u> 1. When the App User realises that he/she has inputted an incorrect email address, the App User clicks on "Not email@serviceprovider.com?". 2. The Account Registration System returns to Step 2 and waits for the App User inputs. <u>AF-3: The App User entered an incorrect, and not registered email address</u> 1. When the App User clicks on "Recover Account", the Account Registration System displays the recover account page again. 2. The recover account page further displays the message "Email not registered!" above the "Help" submission form.

	3. The Account Registration System returns to Step 3 and waits for the App User inputs
Exceptions:	EX-1: The App User forgot his/her registered email. 1. The App User clicks on "Contact Support" 2. The Account Registration System redirects the App User to the FAQ page that contains the support email address. 3. The App User contacts the support via email to retrieve access of his/her account.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	004		
Use Case Name:	VerifyAccount		
Created By:	Lee Juin	Last Updated By:	Lee Juin
Date Created:	25th October 2022	Date Last Updated:	29th October 2022

Actor:	App User (Initiating), Account Registration System
Description:	The Account Registration System can verify that the inputted information are valid using this use case.
Preconditions:	1. The Account Registration System must be up and online. 2. The App User must be connected to the Internet. 3. The App User has input his/her new username, password, repeated password, and email. 4. The App User has ticked the check box "I agree to the Terms of Use and Privacy Policy". 5. The App User has clicked "Sign Up".
Postconditions:	The Account Registration System issues no conflict warning. The App User account registration is successful. *Or* The Account Registration System issues conflict warning. The App User account registration is unsuccessful.
Priority:	
Frequency of Use:	
Flow of Events:	1. The Account Registration System checks that the submitted username is available. 2. The Account Registration System checks that the submitted password satisfies the given constraints of at least one upper case letter, one lower case letter and one digit. 3. Upon completion of Step 1 and Step 2, the Account Registration System displays the "Security Check" submission form. 4. The App User inputs the one-time password (OTP) sent to his/her email inbox to verify his/her email address. 5. The Account Registration System verifies that the input OTP matches the sent OTP. 6. The Account Registration System returns true, signalling the completion of account registration process.
Alternative Flows:	AF-1: The App User inputs a taken username 1. The Account Registration System displays the registration page again. 2. The registration page further displays the message "Username has been taken. Please try again!" above the "Sign Up" submission form. 3. The Account Registration System returns to Step 1 and waits for all the preconditions to be satisfied. AF-2: The App User did not receive the OTP in his/her email inbox 1. The "Resend another OTP" button is available after 60 seconds. 2. The App User clicks on the "Resend another OTP" button to resend another OTP to his/her email inbox. 3. The Account Registration System returns to Step 4 and waits for the App User inputs. AF-3: If the App User inputs an insecure password 1. The Account Registration System displays the registration page again.

	2. The registration page further displays the message “Password does not meet the required standards” above the “Sign Up” submission form. 3. The Account Registration System returns to Step 1 and waits for all the preconditions to be satisfied. AF-4: The App User inputs a mismatched password 1. The Account Registration System displays the registration page again. 2. The registration page further displays the message “Passwords do not match” above the “Sign Up” submission form. 3. The Account Registration System returns to Step 1 and waits for all the preconditions to be satisfied. AF-5: The App User inputs an incorrect OTP 1. The Account Registration System displays the “Security Check” submission form again. 2. A message that says, “Incorrect OTP! Please try again!” is displayed above the “Security Check” submission form. 3. The Account Registration System returns to Step 4 waits for App User input.
Exceptions:	EX-1: If the App User requests for more than three resent of OTP 1. The Account Registration System displays the registration page again. 2. The registration page further displays the message “Please try again with a different email” above the “Sign Up” submission form. 3. The Account Registration System returns to Step 1 and waits for all the preconditions to be satisfied.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	005		
Use Case Name:	SearchItem		
Created By:	Jerick Lim Kai Zheng	Last Updated By:	Lee Juin
Date Created:	22nd August 2022	Date Last Updated:	28th August 2022

Actor:	App User (Initiating), Search and Recommendation System
Description:	The App User will be able to search for items with keywords using this use case.
Preconditions:	1. The Search and Recommendation System is up and online. 2. The App User is connected to the Internet. 3. The App User registered for an account with the Account Registration System. 4. The App User has logged in to his/her account.
Postconditions:	The App User obtained a list of searched items based on the keywords inputted. *Or* The App User is unable to obtain a search result based on the keywords inputted.
Priority:	
Frequency of Use:	
Flow of Events:	1. The App User types a keyword in the search box. 2. The App User clicks on the search icon. 3. The Search and Recommendation System searches, based on the keyword, for items sold on the e-commerce platforms. 4. The Search and Recommendation System retrieves the information of the searched items using the included use case *InfoRetrieve.* 5. The Search and Recommendation System displays the searched keyword and the number of results retrieved. 6. The Search and Recommendation System displays the searched items. 7. The Search and Recommendation System recommends at least three other items to the App User using the included use case *Recommend*. 8. The Search and Recommendation System provides a set of parameters to sort the result based on the included use case *ProvideParam.*
Alternative Flows:	AF-1: The Search and Recommend System returns a list of items which includes sold out items. 1. The Search and Recommendation System displays a list of items sold on the e-commerce platforms based on the keyword inputted by the App User. 2. A “Sold Out” symbol is further displayed over the items which are sold out. 3. The Search and Recommendation System returns to Step 1 and waits for the App User to input another keyword. AF-2: The App User inputs nothing and clicks on the search icon. 1. The Search & Recommendation System displays a list of 100 random items. 2. The Search & Recommendation System displays “Random items” instead of the searched keywords. 3. The Search & Recommendation System displays 100 as the number of results. 4. The Search and Recommendation System returns to Step 1 and waits for the App User to input another keyword.

Exceptions:	EX-1: The Search and Recommendation System is unable to retrieve any items based on the keyword. 1. The Search and Recommendation System will not display any search results. 2. A message that says, “No results found! Sorry we cannot find any results for your search item.” is displayed instead. 3. The Search and Recommendation System will return to Step 1 and waits for the App User to input another keyword.
Includes:	*InfoRetrieve, ProvideParam, Recommend*
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	006		
Use Case Name:	Recommend		
Created By:	Jerick Lim Kai Zheng	Last Updated By:	Lee Juin
Date Created:	22nd August 2022	Date Last Updated:	28th August 2022

Actor:	App User (Initiating), Search and Recommendation System
Description:	Based on the searched items, the Search and Recommendation System recommends the App User other similar items using this use case.
Preconditions:	1. The Search and Recommendation System is up and online. 2. The App User is connected to the Internet. 3. The App User registered for an account with the Account Registration System. 4. The App User has logged in to his/her account.
Postconditions:	App User obtains a list of recommended items based on the searched items.
Priority:	
Frequency of Use:	
Flow of Events:	1. The App User inputs a keyword and clicked on the search icon. 2. The Search and Recommendation System returns a list of items based on the keyword. 3. The Search and Recommendation System recommends at least three other related items based on the searched item. 4. When the App User scrolls to the bottom of the page, the App User can view the section of “You may also like:” which displays the recommended items.
Alternative Flows:	AF-1: The Search and Recommendation System is unable to retrieve any items based on the keyword. 1. The Search and Recommendation System will display three items based on the keyword instead as recommended items to the App User. 2. The Search and Recommendation System returns to Step 1 and waits for the App User to input another keyword.
Exceptions:	EX-1: The App User inputs nothing and clicked on the search icon. 1. When the App User inputs nothing, the Search & Recommendation System will display 100 random items.

	2. The Search & Recommendation System will not display any recommended items.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	007		
Use Case Name:	ProvideParam		
Created By:	Jerick Lim Kai Zheng	Last Updated By:	Lee Juin
Date Created:	22nd August 2022	Date Last Updated:	28th August 2022

Actor:	App User (Initiating), Search and Recommendation System
Description:	The Search and Recommendation System provides the App User a set of parameters to sort the searched results using this use case.
Preconditions:	1. The Search and Recommendation System is up and online. 2. The App User is connected to the Internet. 3. The App User registered for an account with the Account Registration System. 4. The App User has logged in to his/her account.
Postconditions:	1. App User sorts the returned list of items based on the parameters set. 2. App User obtains the best deal of the item searched.
Priority:	
Frequency of Use:	
Flow of Events:	1. The App User inputted a keyword in the search box and clicked on the search icon. 2. The Search and Recommendation System returns a list of items based on the keyword. 3. The Search and Recommendation System provides a set of parameters such as price, number of purchases, form of rebate, delivery fee, payment methods and rating. 4. The App User sorts the results using the parameters and clicks on "Apply". 5. The Search and Recommendation System re-displays the list of times based on the parameters set. 6. The Search and Recommendation System computes the best deal for the App User based on the parameters set.
Alternative Flows:	<u>AF-1: The parameters set by the App User does not match any searched items</u> 1. The Search and Recommendation System will not display any search results. 2. A message that says, "No items that matches the filters set!" is displayed instead. 3. The Search and Recommendation System returns to Step 1 and waits for the App User to re-adjust the parameters.
Exceptions:	<u>EX-1: The Search and Recommendation System is unable to retrieve any items based on the keyword.</u> 1. The Search and Recommendation System will not display any search results. 2. A message that says, "No results found! Sorry we cannot find any results for your search item." is displayed instead. 3. The Search and Recommendation System will not allow the App User to adjust the parameters. 4. The Search and Recommendation System returns to Step 1 and waits for all the pre-conditions to be satisfied.
Includes:	
Special Requirements:	

Assumptions:	
Notes and Issues:	

Use Case ID:	008		
Use Case Name:	InfoRetrieve		
Created By:	Jerick Lim Kai Zheng	Last Updated By:	Lee Juin
Date Created:	22nd August 2022	Date Last Updated:	28th August 2022

Actor:	App User (Initiating), Search and Recommendation System, E-Commerce Platform
Description:	The Search and Recommendation System retrieves the information of an item from the e-commerce platforms using this use case.
Preconditions:	1. The Search and Recommendation System is up and online. 2. The App User is connected to the Internet. 3. The App User registered for an account with the Account Registration System. 4. The App User has logged in to his/her account.
Postconditions:	The Search and Recommendation System displays the information of each item in the list returned.
Priority:	
Frequency of Use:	
Flow of Events:	1. The App User inputted a keyword in the search box and clicked on the search icon. 2. The Search and Recommendation System returns a list of searched items. 3. The Search and Recommendation System retrieves the information of the searched items from the e-commerce platforms such as rating, price, number of items sold, payment method, rebates, and delivery fee. 4. The Search and Recommendation System displays the information of each item.
Alternative Flows:	AF-1: There is no relevant rebate of the searched item 1. The Search and Recommendation System will not show any rebates. 2. A message that says, “No relevant rebates.” is displayed instead at the rebate column.
Exceptions:	EX-1: The Search and Recommend System returns a list of items which includes sold out items. 1. The Search & Recommendation System is unable to retrieve certain information such as price and delivery fee. 2. The Search & Recommendation System only displays the item name and the platform it was once sold on to the App User. 3. A “Sold Out” symbol is further displayed over the items which are sold out.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	009		
Use Case Name:	AddFriend		
Created By:	Oi Yeek Sheng	Last Updated By:	Lee Juin
Date Created:	22nd August 2022	Date Last Updated:	28th August 2022

Actor:	App User (Initiating Actor), Account Registration System
Description:	The App User can send, accept, and reject friend request using this use case.
Preconditions:	1. The Account Registration System is up and online. 2. The App User is connected to the Internet. 3. The App User registered for an account with the Account Registration System. 4. The App User has logged in to his/her account.
Postconditions:	1. The App User successfully sent a friend request to another App User. *And* 2. The App User accepts the friend request of another App User. *Or* 3. The App User rejects the friend request of another App User.
Priority:	
Frequency of Use:	
Flow of Events:	1. The App User clicks on the profile icon in the navigation panel. 2. The App User navigates to “Add Friends” section. 3. The App User searches the username of another App User. 4. The App User sends a friend request to the searched App User by pressing on the add friend icon. 5. The App User’s friend accepts the friend request by clicking on “Accept” next to the App User’s name in the “Friend Request” section. 6. The App User is notified via email that the friend request has been accepted.
Alternative Flows:	AF-1: The other App User rejects the friend request 1. The App User’s friend rejects the friend request by clicking on “Reject” next to the App User’s name in the “Friend Request” section. 2. The App User is notified via email that the friend request has been rejected. 3. The Account Registration System returns to Step 3 and waits for the App User to search for another App User.
Exceptions:	EX-1: The searched username is not available 1. The Account Registration System displays the message that says, “The searched username is not available. Please type a valid username.” 2. The Account Registration System returns to Step 3 and waits for the App User to search for another App User.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	010		
Use Case Name:	CreateWishList		
Created By:	Oi Yeek Sheng	Last Updated By:	Lee Juin
Date Created:	22nd August 2022	Date Last Updated:	28th August 2022

Actor:	App User (Initiating Actor), Wish List System, Search and Recommendation System, E-commerce Platforms
Description:	The App User can create a wish list of items sold on e-commerce platforms using this use case.
Preconditions:	1. The Wish List System is up and online. 2. The Search and Recommendation System is up and online. 3. The App User is connected to the Internet. 4. The App User registered for an account with the Account Registration System. 5. The App User has logged in to his/her account.
Postconditions:	The App User successfully creates a wish list and adds an item to his/her wish list.
Priority:	
Frequency of Use:	
Flow of Events:	1. The App User clicks on the heart icon on the navigation panel. 2. A message that says, “You do not have a wish list yet. Do you want to create one now?” is displayed to the App User. 3. The App User selects “Sure!” to create his/her wish list. 4. The App User gives his/her wish list a name. 5. The App User inputs a keyword in the search bar and clicks on the search icon. 6. The Search and Recommendation System retrieves a list of items based on the keyword. 7. The App User adds the selected items to his/her wish list by clicking on the heart icon on the selected item.
Alternative Flows:	AF-1: The App User adds a sold-out item to his/her wish list. 1. When the App User adds the item to his/her wish list, the App User is prompted with a warning message that says “Warning! The added item is currently sold out and may not be available anytime soon!” 2. The App User clicks on “I understand.” to proceed. 3. The Wish List System returns to Step 3 and waits for the App User to add another item to his/her wish list. AF-2: The App User decides not to create a wish list. 1. After Step 2, the App User selects “Maybe later!”. 2. The Wish List System redirects the App User back to the home page. 3. The Wish List System returns to Step 2 and waits for the App User selection.
Exceptions:	EX-1: The App User attempts to add an item to his/her wish list without creating one. 1. When the App User adds the item to his/her wish list, the App User is prompted with a warning message that says, “Error! You do not have a wish list yet! Do you want to create one now?” 2. If the App User clicks on “Sure!”, the Wish List System creates a wish list for the App User with the previously added item inside. 3. The App User gives his/her wish list a name. 4. If the App User clicks on “Maybe later!”, the warning message disappears, and the item is not added to any wish list.

Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	011		
Use Case Name:	ViewWishList		
Created By:	Oi Yeek Sheng	Last Updated By:	Lee Juin
Date Created:	22nd August 2022	Date Last Updated:	28th August 2022

Actor:	App User (Initiating Actor), Wish List System
Description:	The App User can view his/her, and his/her friends' wish list using this use case.
Preconditions:	1. The Wish List System is up and online. 2. The App User is connected to the Internet. 3. The App User registered for an account with the Account Registration System. 4. The App User has logged in to his/her account.
Postconditions:	The Wish List System displays the App User's wish list. *Or* The Wish List System displays the App User friends' wish list.
Priority:	
Frequency of Use:	
Flow of Events:	1. The App User clicks on the heart icon in the navigation panel. 2. The Wish List System displays the App User's wish list with all the added items. 3. The App User clicks on the profile icon in the navigation panel. 4. The App User navigates to "Friends" section. 5. The App User clicks on "View Wishlist". 6. The Wish List System displays the App User friend's wish list.
Alternative Flows:	AF-1: The App User has not yet created a wish list. 1. When the App User clicks on the heart icon, A message that says, "You do not have a wish list yet. Do you want to create one now?" is displayed to the App User. 2. If the App User selects "Sure!", the Wish List System creates a wish list. 3. The App User gives his/her wish list a name. 4. If the App User selects "Maybe later!", the prompted message is cancelled. 5. The Wish List System redirects the App User back to the home page. 6. The Wish List System returns to Step 1 and waits for App User inputs.
Exceptions:	EX-1: The App User's friend has not yet created a wish list 1. After Step 5, a message that says, "Sorry! The user has not yet created any wish list!" is displayed to the App User. 2. The App User clicks on "Okay." 3. The Wish List System redirects the App User back to the home page.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	012		
Use Case Name:	BirthdayNotification		
Created By:	Oi Yeek Sheng	Last Updated By:	Lee Juin
Date Created:	22nd August 2022	Date Last Updated:	28th August 2022

Actor:	App User (Initiating Actor), Account Registration System, E-commerce Platforms
Description:	The App User can receive birthday notification of his/her friends using this use case.
Preconditions:	1. The Account Registration System is up and online. 2. The App User is connected to the Internet. 3. The App User registered for an account with the Account Registration System. 4. The App User has logged in to his/her account.
Postconditions:	The App User is prompted about his/her friends' birthday 7 days in advance.
Priority:	
Frequency of Use:	
Flow of Events:	1. The App User clicks on the profile icon in the navigation panel. 2. The App User opts in to the Receive Birthday Notification feature. 3. A message that says, "You will be notified of your friend's birthday 7 days in advance." will be displayed to the App User. 4. The Account Registration System registers that the App User has opted in for the feature. 5. When the App User connects with another App User, the App User is notified about his/her friend's birthday 7 days in advance.
Alternative Flows:	AF-1: The App User has not set up his/her birthday details 1. When the App User navigates to his/her Profile page for the first time, the App User is asked to set up his/her personal details, including birthday. 2. If the App User chooses not to set up his/her personal details and attempts to Step 2, an error message that says, "Please set up your personal details prior to opting in for this feature!" will display to the App User. 3. The Account Registration System returns to Step 3 and waits until the App User has set up his/her personal details.
Exceptions:	EX-1: The App User's friend has not set up his/her birthday details 1. The App User will not be notified of his/her friend's birthday.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	