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.

1. **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 stakeholdersto elicit FR and NFR. The team will then formulate a documentation which contains the FR and NFR.

The team will use [SRS\_Template.doc](https://ntulearn.ntu.edu.sg/bbcswebdav/pid-3068116-dt-content-rid-27044536_1/xid-27044536_1), 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.

1. **Project Ideas**

The following are the proposed project ideas:

* 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.

* 1. **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

* 1. **COVID-19 cases heatmap**

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

* + 1. **Feedbacks**

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

* 1. **Dengue fever heatmap**

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

* + 1. **Feedbacks**

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

* 1. **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.

* + 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.

* 1. **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.

1. **Team Name**

The following are the proposed team names:

* 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.

1. **Target Audience**

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

* 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.

* 1. **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.

* 1. **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.

* 1. **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.

1. **Functional Requirements (FR)**

The following are the proposed preliminary FR:

* 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.

1. **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.

1. **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. |

1. **Use Case Diagrams**

Diagram

Description automatically generated

1. **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 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 Register. 3. The registration page prompts the App User to input a valid email, a username and a password that satisfies the given constraint. 4. The App User inputs the credentials accordingly. 5. The App User checks and acknowledges about the Terms & Condition of using the web application before submitting his/her credentials. 6. The website verifies with the Account Registration System on the information provided using the included use case *VerifyAccount.* 7. The App User is asked to verify his/her email through a one-time password (OTP) sent to his/her email inbox. 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 tick the checkbox of acknowledgment of Terms & Conditions   1. When the App User clicks on Submit, the website redirects back to the registration page. 2. The registration page further displays the message “Please tick the checkbox for acknowledging our Terms & Conditions!” 3. The website returns to Step 3 and waits for App User inputs.   AF-2: The App User left input field(s) blank   1. When the App User clicks on Submit, the website redirects back to the registration page. 2. The registration page further displays the message “Please ensure all fields have been filled up before submitting!” 3. The website returns to Step 3 and waits for App User inputs. | | |
| **Exceptions:** | | EX-1: The App User repeatedly attempts to register for an account despite errors in input   1. On the tenth failed attempt at registering for an account by the App User, the registration page displays the message “Too many attempts! Please try again in 10 minutes.” 2. The registration page will only accept registration from the App User’s IP address after 10 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 Log In. 3. The login page prompts the App User to input his/her username and password. 4. The App User inputs the credentials accordingly. 5. The website verifies with the Account Registration System on the credentials provided. 6. When the information is verified, the App User is redirected to his/her account dashboard. | | |
| **Alternative Flows:** | | AF-1: If the App User inputs an incorrect username or password   1. The website returns to Step 2 and displays the login page again. 2. The login page further displays the message “Invalid username and/or password!” to explain to the App User the reason why the account login is unsuccessful.   AF-2: The App User left input field(s) blank   1. When the App User clicks on Submit, the website redirects back to the login page. 2. The login page further displays the message “Please ensure all fields have been filled up before submitting!” 3. The website returns to Step 3 and waits for App User inputs. | | |
| **Exceptions:** | | EX-1: If the App User inputs incorrect username or password for more than five times   1. The website returns to Step 2 and displays the login page again. 2. The login page further displays the message “Account suspended! Please login again after 15 minutes.”   EX-2: If the App User forgot his/her login credentials   1. The App User clicks on Forgotten? on the login page. 2. The App User is able to 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: | 25th 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 has forgotten his/her login credentials. | | |
| **Postconditions:** | | The App User has successfully recovered his/her account by changing his/her credentials. | | |
| **Priority:** | |  | | |
| **Frequency of Use:** | |  | | |
| **Flow of Events:** | | 1. The App User clicks on Forgotten? on the login page. 2. The website prompts the App User to input his/her registered email. 3. The App User is asked to verify his/her identity through a one-time password (OTP) sent to his/her email inbox. 4. The website prompts the App User to input a new set of username and password. 5. Once completed, the App User is redirected back to the login page. | | |
| **Alternative Flows:** | | AF-1: If the App User did not receive the OTP in his/her email inbox   1. The website allows the App User to resend another OTP after 60 seconds. | | |
| **Exceptions:** | | EX-1: If the App User forgot his/her registered email.   1. The App User clicks on Forgot Email? when prompted to input his/her registered email. 2. The website prompts the App User to contact Customer Support 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 input new username and password satisfy the given constraints. | | |
| **Preconditions:** | | 1. The Account Registration System must be up and online. 2. The App User has input his/her new username, password, and email. 3. The App User has clicked Submit after inputting the credentials. | | |
| **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 whether the submitted username is available. 2. The Account Registration System checks whether the submitted password satisfies the given constraints of at least one upper case letter, one lower case letter and one digit. 3. The Account Registration System returns true upon verifying both Step 1 and Step 2. | | |
| **Alternative Flows:** | | AF-1: If the App User inputs a taken username   1. The website reloads and displays the registration page again. 2. The registration page further displays the message “Username has been taken. Please try again!” to explain to the App User the reason why the account registration is unsuccessful.   AF-2: If the App User did not receive the OTP in his/her email inbox   1. The website allows the App User to resend another OTP after 60 seconds.   AF-3: If the App User inputs an insecure password   1. The website reloads and displays the registration page again. 2. The registration page further displays the message “Password does not meet the required standards” to explain to the App User the reason why the account registration is unsuccessful.   AF-4: If the App User inputs a mismatched password   1. The website returns to Step 2 and displays the registration page again. 2. The registration page further displays the message “Passwords do not match!” to explain to the App User the reason why the account registration is unsuccessful. | | |
| **Exceptions:** | | EX-1: If the App User requests for more than three resent of OTP   1. The website returns to the registration page. 2. The registration page further displays the message “Please try again with a different email.” | | |
| **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 & Recommendation System | | |
| **Description:** | | The App User will be able to search for items with keywords using this use case. | | |
| **Preconditions:** | | 1. App User registered for an account with the Account Registration System. 2. App User has logged in to his/her account. | | |
| **Postconditions:** | | App User obtained a list of searched items based on the keywords inputted.  *Or*  App User was unable to obtain a search result for the item. | | |
| **Priority:** | |  | | |
| **Frequency of Use:** | |  | | |
| **Flow of Events:** | | 1. The App User types a keyword in the search box of the application. 2. The Search and Recommendation System searches for the items sold on the e-commerce platforms. 3. The Search and Recommendation System retrieves the information of the searched items using the included use case *InfoRetrieve.* 4. The Search and Recommendation System displays the searched items. 5. 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: If the searched item is sold out   1. The Search and Recommendation System displays the searched items from the e-commerce platforms. 2. A “Sold Out” symbol is displayed over the items. 3. The Search and Recommendation System displays an estimated time before the item will be in stock again.   AF-2: If the App User inputs nothing and clicked on Search   1. When the App User does not input anything, the Search & Recommendation System will display a list of random items. | | |
| **Exceptions:** | | EX-1: If the searched item is not sold on the supported E-Commerce platforms   1. The Search and Recommendation System will not display any search results. 2. There will be a displayed message that says “No results found” 3. The Search and Recommendation System will return to Step 1 to wait for the App User to search for a new item. | | |
| **Includes:** | | *InfoRetrieve, ProvideParam* | | |
| **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 & Recommendation System | | |
| **Description:** | | Based on the searched item, the Search & Recommendation System can recommend the App User other similar items. | | |
| **Preconditions:** | | 1. App User has logged in to his/her account. 2. App User inputted a keyword in the search box and clicked on Search. | | |
| **Postconditions:** | | App User obtained a list of recommended items based on the searched items. | | |
| **Priority:** | |  | | |
| **Frequency of Use:** | |  | | |
| **Flow of Events:** | | 1. When the App User inputs a keyword, the Search and Recommendation System returns a list of items based on the keyword. 2. The Search and Recommendation System recommends at least three other related items based on the searched item. 3. 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: If the App User inputs non-meaningful keywords such as spam characters   1. The Search & Recommendation System will display a list of random items as recommended items to the App User. | | |
| **Exceptions:** | | EX-1: If the App User inputs nothing and clicked on Search   1. When the App User does not input anything, the Search & Recommendation System will only display a list of 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 & Recommendation System | | |
| **Description:** | | The Search & Recommendation System provides the App User a set of tuning parameters to sort the searched results using this use case. | | |
| **Preconditions:** | | 1. App User inputted a keyword in the search box and clicked on Search. 2. The Search & Recommendation System returned a list of items based on the searched keyword. | | |
| **Postconditions:** | | 1. App User can sort 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 sorts the results using a set of parameters such as price, number of purchases, form of rebate, delivery fee, payment methods and rating. 2. The Search and Recommendation System will be able to compute the best deal for the user based on the parameters set by the user. | | |
| **Alternative Flows:** | | AF-1: If the parameters set by the App User does not match any searched items   1. The Search and Recommendation System will not display any search results. 2. There will be a displayed message that says, “No items that matches the filters set!” 3. The Search and Recommendation System will wait for the App User to search for a new item. | | |
| **Exceptions:** | | EX-1: If the searched item is not sold on the supported E-Commerce platforms   1. The Search and Recommendation System will not display any search results. 2. There will be a displayed message that says, “No results found!” 3. The Search and Recommendation System will not allow the App User to adjust the parameters. 4. The Search and Recommendation System will wait for the App User to search for a new item. | | |
| **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 & Recommendation System, E-Commerce Platform | | |
| **Description:** | | The Search & Recommendation System retrieves the relevant information of a sold item from the e-commerce platforms using this use case. | | |
| **Preconditions:** | | App User inputted a keyword in the search box and clicked on Search. | | |
| **Postconditions:** | | The Search & Recommendation System displays the information of each item in the list returned. | | |
| **Priority:** | |  | | |
| **Frequency of Use:** | |  | | |
| **Flow of Events:** | | 1. The Search & Recommendation System retrieves the information of each sold items from the e-commerce platforms such as rating, price, and delivery fee. 2. The Search & Recommendation System displays the information of each sold item. | | |
| **Alternative Flows:** | | AF-2: If there is no relevant rebate of the searched item   1. The Search and Recommendation System will not show any rebates and display the message “No relevant rebates.” | | |
| **Exceptions:** | | EX-1: If the searched item is sold out   1. The Search & Recommendation System will not be able 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 displayed over the items. 4. The Search and Recommendation System displays an estimated time before the item will be in stock again. | | |
| **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:** | | The App User must be logged in to an account. | | |
| **Postconditions:** | | The App User successfully sent a friend request to another App User.  *Or*  The App User accepts the friend request of another App User.  *Or*  The App User rejects the friend request of another App User. | | |
| **Priority:** | |  | | |
| **Frequency of Use:** | |  | | |
| **Flow of Events:** | | 1. The App User adds friends by searching the unique username of each App User. 2. If the username searched is available in the Account Registration System, the App User may send a friend request to that user. 3. When the other App User accepts the friend request, the App User can now view the friend’s wish list and birthday date. 4. The App User can receive birthday notification from the Wish List System using the included use case *BirthdayNotifcation*. | | |
| **Alternative Flows:** | | AF-2: If the other App User rejects the friend request   1. The Wish-list System sends an email to the App User about the friend request being rejected. | | |
| **Exceptions:** | | AF-1: If the searched username is not available   1. The Account Registration System displays the message “The searched username is not available. Please insert a valid username.” | | |
| **Includes:** | | *BirthdayNotification* | | |
| **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, E-commerce Platforms | | |
| **Description:** | | The App User can create a wish list of sold items on e-commerce platforms using this use case. | | |
| **Preconditions:** | | The App User must be logged in to an account. | | |
| **Postconditions:** | | The App User successfully adds an item to his/her wish list. | | |
| **Priority:** | |  | | |
| **Frequency of Use:** | |  | | |
| **Flow of Events:** | | 1. The App User searches for an item using the search function provided in the Search and Recommendation System. 2. The Wish List System checks for availability of that item on all supported E-commerce platforms. 3. If the item is available on at least one E-commerce platform, the Search and Recommendation System displays the items available to the App User. 4. The App User selects the item to be added to his/her wish list. | | |
| **Alternative Flows:** | | AF-1: If the item is sold out on the supported e-commerce platforms   1. The item is displayed with a “Sold Out” message. 2. 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!” | | |
| **Exceptions:** | | AF-1: If the item is not sold on the supported e-commerce platforms   1. The Wish List system displays the message “No results found.” 2. The Wish List system recommends at least one similar item to the user. 3. If the App User selects the recommended item, the Wish List System will add the item to his/her 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:** | | The App User must be logged in to an account. | | |
| **Postconditions:** | | The Wish List System displays the App User’s or his/her friends’ wish list. | | |
| **Priority:** | |  | | |
| **Frequency of Use:** | |  | | |
| **Flow of Events:** | | 1. The App User clicks on the Profile navigation panel. 2. The App User can view his/her wish list underneath his/her profile. 3. The App User may click on the Friends navigation panel. 4. The App User can view his/her friends’ wish list by clicking on the Profile icon next to each entry on the friends’ list. | | |
| **Alternative Flows:** | | AF-1: If the App User has not yet created a wish list   1. When the App User enters his/her Profile page, the App User is prompted with a message to create his/her wish list if he/she chooses to. 2. The App User may respond with “Sure!” or “Maybe later”. 3. The former will create an empty wish list for the App User to add items later. The latter will cancel the prompted message. | | |
| **Exceptions:** | | EX-1: If the App User’s friend has not yet created a wish list   1. When the App User enters his/her friend’s Profile page, a message that says “Sorry! The user has not yet created any wish list!” is displayed underneath the profile. | | |
| **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), System, E-commerce Platforms | | |
| **Description:** | | The App User can receive birthday notification of his/her friends using this use case. | | |
| **Preconditions:** | | The App User must have enabled the Receive Birthday Notification feature. | | |
| **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 navigates to his/her Profile page. 2. The App User opts in to the Receive Birthday Notification feature. 3. 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 user. | | |
| **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:** | |  | | |