Use Cases

for

<Jiak Lah!>

Version 1.0 approved

Prepared by <Noel Wee Ee Cher>

<Foodies>

<13/2/2023>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Guidance for Use Case Template

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

# Use Case Identification

## Use Case ID

Give each use case a unique numeric identifier, in hierarchical form: X.Y. Related use cases can be grouped in the hierarchy. Functional requirements can be traced back to a labeled use case.

## Use Case Name

State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun. Some examples:

1. View part number information.
2. Manually mark hypertext source and establish link to target.
3. Place an order for a CD with the updated software version.

## Use Case History

### Created By

Supply the name of the person who initially documented this use case.

### Date Created

Enter the date on which the use case was initially documented.

### Last Updated By

Supply the name of the person who performed the most recent update to the use case description.

### Date Last Updated

Enter the date on which the use case was most recently updated.

# Use Case Definition

## Actor

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor(s) that will be performing this use case.

## Description

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

## Preconditions

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. User’s identity has been authenticated.
2. User’s computer has sufficient free memory available to launch task.

## Postconditions

Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. Document contains only valid SGML tags.
2. Price of item in database has been updated with new value.

## Priority

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

## Frequency of Use

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

## Flow of Events

Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, “How do I <accomplish the task stated in the use case name>?” This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system.

## Alternative Flows

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative course, and describe any differences in the sequence of steps that take place. Number each alternative course using the Use Case ID as a prefix, followed by “AC” to indicate “Alternative Course”. Example: X.Y.AC.1.

## Exceptions

Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. Number each exception using the Use Case ID as a prefix, followed by “EX” to indicate “Exception”. Example: X.Y.EX.1.

## Includes

List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.

## Special Requirements

Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.

## Assumptions

List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.

## Notes and Issues

List any additional comments about this use case or any remaining open issues or TBDs (To Be Determineds) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1.0 | | |
| Use Case Name: | Create Jiak Lah! Account | | |
| Created By: | Noel Wee Ee Cher | Last Updated By: | Noel Wee Ee Cher |
| Date Created: | 13/2/2023 | Date Last Updated: | 13/2/2023 |

|  |  |
| --- | --- |
| Actor: | User (Initiating Actor), Login Database |
| Description: | User creates a new Jiak Lah! account by entering a new username and password. |
| Preconditions: | * User’s device has sufficient free memory to launch Jiak Lah! and carry out the task * User’s device is connected to a reliable data network * User’s device has sufficient battery to launch Jiak Lah! and carry out the task |
| Postconditions: | 1. Jiak Lah’s Login Database has been updated with new String values (Username and Password) 2. The User has created an account |
| Priority: | Low |
| Frequency of Use: | Low |
| Flow of Events: | 1. User clicks the “Create a New Account Button”. 2. Jiak Lah! brings User to the Account Creation Page. 3. Jiak Lah! requests for the User to enter a new username and password. 4. User enters a new username and password. 5. User clicks the “Confirm Account Creation” button, 6. Upon confirmation, the Login Database stores the User’s credentials. 7. User’s account is created, and the User is brought back to the Login Page. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | 1. Create new Username 2. Create new Password |
| Special Requirements: | 1. Login Database must be secure in order to store User’s credentials |
| Assumptions: | 1. User is interested in creating a new account instead of logging in through Google Account |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1.1 | | |
| Use Case Name: | Login to Jiak Lah! | | |
| Created By: | Noel Wee Ee Cher | Last Updated By: | Noel Wee Ee Cher |
| Date Created: | 13/2/2023 | Date Last Updated: | 13/2/2023 |

|  |  |
| --- | --- |
| Actor: | User (Initiating Actor), Login Database, Jiak Lah! System, Google Accounts Database |
| Description: | User logins into Jiak Lah! through a Registered Account from Jiak Lah’s Login Database, through Google Account authentication or simply through Guest Login. |
| Preconditions: | * User’s device has sufficient free memory to launch Jiak Lah! and carry out the task * User’s device is connected to a reliable data network * User’s device has sufficient battery to launch Jiak Lah! and carry out the task * User has already created a Jiak Lah! Account, or has an existing Google Account |
| Postconditions: | User’s credentials match with the Login Database. User’s account has been authenticated. |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. User clicks the Login button 2. Jiak Lah! requests for the User’s username and password. 3. User enters his username and password 4. Upon successful authentication, User is brought to Jiak Lah’s Main Listing Page |
| Alternative Flows: | 1.1.AC.1. User logins with existing Google Account. If the User has a valid existing Google Account, User is then brought to Jiak Lah’s Main Listing Page  1.1.AC.2 User simply uses Guest Login and is then brought to Jiak Lah’s Main Listing Page. However, some functionalities will not be made available to the User. |
| Exceptions: | 1.1.EX.1. If User entered a wrong username, the prompt, “Sorry, you have entered an invalid username or password, please try again!” will be shown.  1.1.EX.2. If User entered a wrong password, the prompt, “Sorry, you have entered an invalid username or password, please try again!” will be shown. |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | 1. User has created a valid Jiak Lah account or has an existing Google Account 2. User is able to remember his account username and password |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2.0 | | |
| Use Case Name: | View All Listings | | |
| Created By: | Noel Wee Ee Cher | Last Updated By: | Noel Wee Ee Cher |
| Date Created: | 13/2/2023 | Date Last Updated: | 13/2/2023 |

|  |  |
| --- | --- |
| Actor: | User (Initiating Actor), Jiak Lah System |
| Description: | User is able to view all the listings available in Jiak Lah’s Main Listing Page. |
| Preconditions: | * User’s device has sufficient free memory to launch Jiak Lah! and carry out the task * User’s device is connected to a reliable data network * User’s device has sufficient battery to launch Jiak Lah! and carry out the task * User has logged into Jiak Lah! |
| Postconditions: | - |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. Upon successful login, User is brought to Jiak Lah’s Main Listing Page 2. User is able to scroll through all the listings available and click on any listing that piques his interest. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | 1. User is successfully logged in into Jiak Lah! |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2.1 | | |
| Use Case Name: | Search for Listing | | |
| Created By: | Noel Wee Ee Cher | Last Updated By: | Noel Wee Ee Cher |
| Date Created: | 13/2/2023 | Date Last Updated: | 13/2/2023 |

|  |  |
| --- | --- |
| Actor: | User (Initiating Actor), Jiak Lah! System |
| Description: | User searches for a specific listing |
| Preconditions: | * User’s device has sufficient free memory to launch Jiak Lah! and carry out the task * User’s device is connected to a reliable data network * User’s device has sufficient battery to launch Jiak Lah! and carry out the task |
| Postconditions: | - |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. User searches for a specific listing by typing in the search bar located in the Main Listing Page 2. If search is successful, the specific listing will be shown. User can then click on it to view further details. |
| Alternative Flows: | 2.1.AC.1. User is able to filter and search through the listings by clicking on Jiak Lah’s in-built filters |
| Exceptions: | 2.1.EX.1. If User types the listing’s name wrongly, or types in an invalid listing, the error message “Sorry, we could not locate what you are looking for, please try again!” will be shown. |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | 1. User is successfully logged in into Jiak Lah! 2. User types in the listing’s name correctly |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2.2 | | |
| Use Case Name: | View Specific Listing | | |
| Created By: | Noel Wee Ee Cher | Last Updated By: | Noel Wee Ee Cher |
| Date Created: | 13/2/2023 | Date Last Updated: | 13/2/2023 |

|  |  |
| --- | --- |
| Actor: | User (Initiating Actor), Jiak Lah! System |
| Description: | After searching through string inputs or through Jiak Lah’s filters, User is able to view a specific listing by clicking on it. |
| Preconditions: | * User’s device has sufficient free memory to launch Jiak Lah! and carry out the task * User’s device is connected to a reliable data network * User’s device has sufficient battery to launch Jiak Lah! and carry out the task * User has found the listing he is interested in through Jiak Lah’s search functions |
| Postconditions: | - |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. User has found the listing he is interested in 2. User clicks on the listing and is able to view further details such as location, price range, descriptions, ratings etc. 3. User is also able to leave a review and/or submit accessibility tags which correlate to the characteristics of the listing. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | 1. User is successfully logged in into Jiak Lah! 2. User has already found a specific listing |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3.0 | | |
| Use Case Name: | Submit Review | | |
| Created By: | Noel Wee Ee Cher | Last Updated By: | Noel Wee Ee Cher |
| Date Created: | 13/2/2023 | Date Last Updated: | 13/2/2023 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | User leaves a review and/or submit accessibility tags, which correlate to the characteristics of the listing, through the listing’s “Leave a Review” button |
| Preconditions: | * User’s device has sufficient free memory to launch Jiak Lah! and carry out the task * User’s device is connected to a reliable data network * User’s device has sufficient battery to launch Jiak Lah! and carry out the task * User is already on a specific listing |
| Postconditions: | - |
| Priority: | Medium |
| Frequency of Use: | Medium |
| Flow of Events: | 1. On a specific listing, User clicks the “Leave a Review” button. 2. User chooses the accessibility tags that best describe the listing. 3. User types out his review of the listing. 4. Once done, User clicks the “Submit Review” button. 5. User is brought back to the listing page. |
| Alternative Flows: | 3.0.AC.1. After viewing the listing, User clicks on the “Leave a Review” button, but only leaves a review without choosing accessibility tags.  3.0.AC.2. After viewing the listing, User clicks on the “Leave a Review” button, but only chooses some accessibility tags without leaving a review. |
| Exceptions: | 3.0.EX.1 If the user tries to submit review without typing anything or choosing any accessibility tags, the error message “Invalid Response! Please enter in your review or choose your tags!” will appear. |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | 1. User is successfully logged in into Jiak Lah! 2. User has already found a specific listing 3. User wants to review the listing |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 4.0 | | |
| Use Case Name: | Access Settings | | |
| Created By: | Noel Wee Ee Cher | Last Updated By: | Noel Wee Ee Cher |
| Date Created: | 13/2/2023 | Date Last Updated: | 13/2/2023 |

|  |  |
| --- | --- |
| Actor: | User (Initiating Actor), Jiak Lah! System |
| Description: | User accesses his settings to view past reviews |
| Preconditions: | * User’s device has sufficient free memory to launch Jiak Lah! and carry out the task * User’s device is connected to a reliable data network * User’s device has sufficient battery to launch Jiak Lah! and carry out the task * User must be logged in either through a Registered Account or through Google |
| Postconditions: | - |
| Priority: | Medium |
| Frequency of Use: | Medium |
| Flow of Events: | 1. User clicks the settings button. 2. User can view all his past reviews. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | 1. User is successfully logged in into Jiak Lah! as a registered user |
| Notes and Issues: | - |