Bee Lazi

Use-Case Specification: <Use-Case Name>

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 21/11/2016 | 1.0 | First draft of Use case’s document | Van Tu-Ninh |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Use-case Model 4

2. Use-case Specifications 4

2.1 Use-case: Create Account 4

2.2 Use-case: User Authentication. 4

2.3 Use-case: Log In 5

2.4 Use-case: Log out. 5

2.5 Use-case: Schedule-based Suggestion. 6

2.6 Use-case: Update Personal’s Schedule. 6

2.7 Use-case: Recieve sugesstion 6

2.8 Use-case: View detail-information: 7

2.9 Use-case: respond suggestion: 7

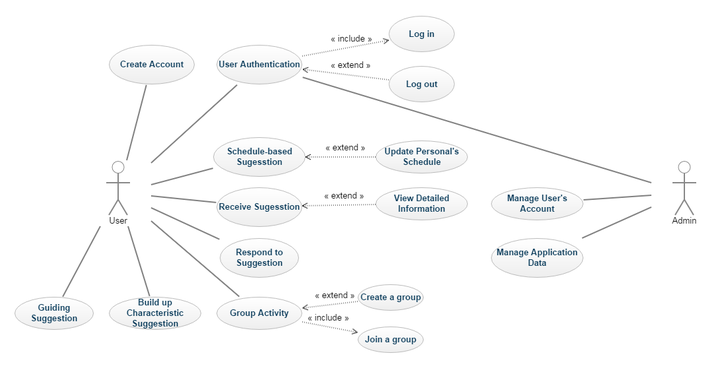
2.10 Use – case: Guiding suggestion 7

2.11 Use – case: Build up characteristic suggestion 8

2.12 Use-case: Manage User’s Account. 9

2.13 Use-case: Manage Application Data. 9

# Use-case Model



# Use-case Specifications

## Use-case: Create Account

|  |  |
| --- | --- |
| Use case Name | Create Account |
| Brief description | This use-case describes how the user create new account to log in application |
| Actors | Users, Customers |
| Basic Flow | 1. At Log In activity, user presses sign up button to access sign up session. 2. User fills personal information: email address, username, phone number, and password. 3. User clicks finish button on top-right corner of sign up session to finish. |
| Alternative Flows | **Alternative flow 1: User cancels creating new account.**   1. From #1 of the basic flow, user returns to Log In activity.   **Alternative flow 2: User clicks finish button**   1. From #3 of the basic flow, user returns to Log In activity. |
| Pre-conditions | User goes to Log In activity. |
| Post-conditions | User finishes sign up session and creates new account successfully. |

## Use-case: User Authentication.

|  |  |
| --- | --- |
| Use case Name | User Authentication |
| Brief description | This use-case describes how user-authentication works. |
| Actors | Users, Customers |
| Basic Flow | 1. Check if the password and username is are correct. |
| Alternative Flows | **Alternative flow 1: User types wrong username or password**   1. From #1 of the basic flow, a warning appears to remind users that they have just typed wrong username or password.   **Alternative flow 2: User types correct username and password**   1. From #1 of the basic flow, user access main interface of application. |
| Pre-conditions | User logs in or logs out application |
| Post-conditions | Successfully check user-authentication including username and password. |

## Use-case: Log In

|  |  |
| --- | --- |
| Use case Name | Log In |
| Brief description | This use-case describes choices for user and admin to sign in application. This use-case is a ≪include≫ of use-case User Authentication. |
| Actors | Users, Customers, Admin |
| Basic Flow | 1. User chooses methods to sign in application including: Log In with Facebook account or Log In with user’s application account. 2. If User chooses Log In with Facebook account, Facebook log in session for application appears. 3. If User chooses Log In with user’s application account, user must fill in username and password in Log In activity. |
| Alternative Flows | **Alternative flow 1: User types wrong username or password**   1. From #3 of the basic flow, a warning appears to remind users that they have just typed wrong username or password.   **Alternative flow 2: User logs in successfully**   1. From #1 of the basic flow, user access main interface of application.   **Alternative flow 3: User has no Facebook account**   1. From #2 of the basic flow, user returns to Log In activity and a warning appears to remind user creating Facebook account. User can choose to log in with application account instead. |
| Pre-conditions | User clicks Log In button. |
| Post-conditions | Successfully Log in application and user goes to main interface. |

## Use-case: Log out.

|  |  |
| --- | --- |
| Use case Name | Log out |
| Brief description | This use-case describes how user logs out application. This use-case is a ≪extend≫ of use-case User Authentication. |
| Actors | Users, Customers, Admin |
| Basic Flow | 1. User clicks Log out button. 2. Authenticate account to log out. 3. Finish main activity session and return to Log In activity |
| Alternative Flows | There is no alternative flows for this use-case |
| Pre-conditions | User clicks logs out button. |
| Post-conditions | Successfully log out and return to Log In activity. |

## Use-case: Schedule-based Suggestion.

|  |  |
| --- | --- |
| Use case Name | Schedule-based Suggestion |
| Brief description | This use-case describes how Schedule-based suggestion functions. |
| Actors | Users, Customers |
| Basic Flow | 1. User inputs his/her weekly/daily schedule. 2. System makes suitable daily activity suggestion corresponding to the free time of user in the schedule. |
| Alternative Flows | **Alternative flow 1: User does not input his/her weekly/daily schedule**   1. The application supposed that users would be free at any time in a day. The suggestion is not filtered by time. |
| Pre-conditions | User inputs his/her weekly/daily schedule. |
| Post-conditions | The suggestions are more precise for user in terms of free time according to the given schedule. |

## Use-case: Update Personal’s Schedule.

|  |  |
| --- | --- |
| Use case Name | Update Personal’s Schedule |
| Brief description | This use-case describes how user updates personal’s schedule. |
| Actors | Users, Customers |
| Basic Flow | 1. User chooses Update Schedule button. Update Schedule session appears. 2. User updates information of schedule. 3. User clicks save button to finish update session. |
| Alternative Flows | **Alternative flow 1: User cancels updating personal’s schedule**   1. From #1 or #2 or #3 of the basic flow, user exits Update Schedule session and return to main interface. |
| Pre-conditions | User clicks Update Personal’s Schedule button. |
| Post-conditions | Successfully update schedule if user saves the changes. If user cancels updating, the old information must not be changed. |

## Use-case: Recieve sugesstion

|  |  |
| --- | --- |
| Use case Name | Recieve suggestion |
| Brief description | This use case describes how the app suggest to user. |
| Actors | Customer |
| Basic Flow | 1. User choose that function on the homepage.  2. App show a action, and two view for 2 another use-case view detail-infomation and respond suggestion. |
| Alternative Flows | **--------------------------** |
| Pre-conditions | User choose this function. |
| Post-conditions | ---------------------- |

## Use-case: View detail-information:

|  |  |
| --- | --- |
| Use case Name | View detail-information. |
| Brief description | This use case describes how user view detail-information of one suggested action. |
| Actors | Customer |
| Basic Flow | 1. After receive suggestion, user click on view-detail button.  2. App show a list of information of that action depend of type of action (film/food).  3. User close the information box to return receive suggestion action. |
| Alternative Flows | **---------------------** |
| Pre-conditions | User click on view-detail button on receive suggestion use-case. |
| Post-conditions | ------------------- |

## Use-case: respond suggestion:

|  |  |
| --- | --- |
| Use case Name | Respond Suggestion |
| Brief description | This use case describes how user respond to a suggestion. |
| Actors | Customer |
| Basic Flow | 1. After receive suggestion, a small yes/no button included in the photo of action which is suggested.  2. User choose from 2 options. |
| Alternative Flows | 1. If user choose yes, action finish.  2. If user choose no, update information and suggest another action (go to receive suggestion use-case). |
| Pre-conditions | User go to receive suggestion use-case. |
| Post-conditions | Update user’s informations. |

## Use – case: Guiding suggestion

|  |  |
| --- | --- |
| Use case Name | Guiding suggestion |
| Brief description | This use case gives a suggestion based on a sequence of questions smartly chosen for customer. |
| Actors | Customer |
| Basic Flow | 1. At the application screen, there will have the first question, and two button Yes and No, or the star rating, or something else based on the type of question. 2. User reads the question and answers it by selecting on the button or rating based on the type of question. 3. The system will show the next question. Similarly, the user reads and answers the question. Until, the system is very confident to suggest the user what is useful to him. 4. The user needs to select one of three buttons, Do, Not Do, I can’t, for better suggestion in the future. 5. The system put the suggest activity in the Main Job if the user accepts the suggestion activity 6. When the user completes the activity, he can mark activity as completed by select the button Completed 7. The system gives the user many scores for a period of a month and show the user his last achievements |
| Alternative Flows | **Alternative flow 1: The system asked more than 6 questions.**   1. The system will show a message for user to know and then show a list of relevant questions. The user chooses one of them. 2. Continue step #5 in the basic flow.   **Alternative flow 2: The user marked the activity in the Main Job as not completed**   1. The user will be given a minus mark in the total score. 2. The system show a message for advising them doing better in the future.   **Alternative flow 3: The user selects button Not Do, or I can’t**   1. The system will show message “I will give you more useful suggestion”. 2. The system returns to the main screen for ending the function. |
| Pre-conditions | User need to select the guiding suggesting mode. |
| Post-conditions | The user successfully does the suggested activity |

## Use – case: Build up characteristic suggestion

|  |  |
| --- | --- |
| Use case Name | Build up characteristic suggestion |
| Brief description | The system gives the user the suggestion based on the selected characteristic |
| Actors | Customer |
| Basic Flow | 1. At the application screen, there is one suggestion appeared and 3 buttons Do, Not Do, I can’t. The user choose Do. 2. The system gives more detail information or description about the activity for the user to follow and complete the activity. 3. The user selects “marked as complete” 4. The system gives scores to the user and show his last achievements. |
| Alternative Flows | **Alternative flow 1: The user selects button Not Do, or I can’t**   1. The system will show message “I will give you more useful suggestion”. 2. The system returns to the main screen for ending the function.   **Alternative flow 2: The user marked the activity in the Main Job as not completed**   1. The user will be given a minus mark in the total score. 2. The system show a message for advising them doing better in the future. |
| Pre-conditions | User need to select the build-up characteristic suggesting mode. |
| Post-conditions | The user successfully does the suggested activity |

## Use-case: Manage User’s Account.

|  |  |
| --- | --- |
| Use case Name | Manage User’s Account |
| Brief description | This use-case describes how admin manages user’s account. |
| Actors | Admin |
| Basic Flow | 1. Admin checks if any account violates the terms and conditions of application. 2. If any account violates terms and conditions of application, admin can ban or delete account. 3. Admin updates the user information to database if there is any requests from users. |
| Alternative Flows | There is no alternative flow for this use-case. |
| Pre-conditions | Admin has to log in with administrator account to manage user’s account. |
| Post-conditions | Admin can make actions including delete, create or ban account after having logged in. |

## Use-case: Manage Application Data.

|  |  |
| --- | --- |
| Use case Name | Manage Application Data |
| Brief description | This use-case describes how admin manages application data |
| Actors | Admin |
| Basic Flow | 1. Admin removes unhealthy suggestions or any suggestions of activity that break the laws. 2. Admin adds new activity suggestions. 3. Admin updates suggestion if any suggestion is outdated. |
| Alternative Flows | There is no alternative flow for this use-case. |
| Pre-conditions | Admin has to log in with administrator account to manage user’s account. |
| Post-conditions | Admin can make actions include: removes, adds, or updates information of suggestions after having logged in. |