**Use Cases**

**for**

**<**Recreation Activities App**>**

**Version 2.0 approved**

**Prepared by <KingCode>**

**<NTU SE project>**

**<10-02-2019>**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| sehyun hwang | 10-02-2019 | Initial use case model | 1.0 |
| sehyun hwang | 24-02-2019 | modified use case model for lab2 | 2.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **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:

* View part number information.
* Manually mark hypertext source and establish link to target.
* 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: | RA001 | | |
| Use Case Name: | Search for Community Centre | | |
| Created By: | sehyun hwang | Last Updated By: | sehyun hwang |
| Date Created: | 10-02-2019 | Date Last Updated: | 24-02-2019 |

|  |  |
| --- | --- |
| Actor: | user |
| Description: | This use case allows users to search for the CCs. Users can search for CCs by location or name. If there are no CCs that match keywords or keywords are wrong, the system must show error message. |
| Preconditions: | 1. The user have to login on app. |
| Postconditions: | 1. The system displays the list of relevant CCs. |
| Priority: |  |
| Frequency of Use: | often used |
| Flow of Events: | 1. The user selects “search by location” or “search by name” 2. The system proceeds searching by using RA009 or RA010. 3. The system displays the list of relevant CCs. |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: | RA009 Search for Community Centre by name  RA010 Search for Community Centre by location |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | GPS function should be added to functional requirements.  Priorities are not determined.  Use Case IDs should be organized as logical order. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | RA009 | | |
| Use Case Name: | Search for Community Centre by name | | |
| Created By: | sehyun hwang | Last Updated By: | N/A |
| Date Created: | 24-02-2019 | Date Last Updated: | N/A |

|  |  |
| --- | --- |
| Actor: | user |
| Description: | This use case allows users to search for the CCs by their name.  If there are no CCs that match keywords or keywords are wrong, the system must show error message. |
| Preconditions: | 1. The user turns on the app. |
| Postconditions: | 1. The system shall display the list of relevant CC with correct name. |
| Priority: |  |
| Frequency of Use: | often |
| Flow of Events: | 1. The user clicks the “search by name” button. 2. The user enters the name or uses Auto-Suggest prompting. 3. The system validates the keyword. 4. The system verifies the keyword. 5. The system displays the list of relevant CCs. |
| Alternative Flows: | RA009.AC.2. Invalid keywords. (wrong name)   1. The user reenters the keyword. 2. The user clicks the “search” button again. 3. returns to RA009.4 |
| Exceptions: | N/A |
| Includes: | N/A |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | N/A |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | RA010 | | |
| Use Case Name: | Search for Community Centre by location | | |
| Created By: | sehyun hwang | Last Updated By: | N/A |
| Date Created: | 24-02-2019 | Date Last Updated: | N/A |

|  |  |
| --- | --- |
| Actor: | user |
| Description: | This use case allows users to search for the CCs by their location. The system shall show at least the 5 closest CCs by accuracy of the keyword. |
| Preconditions: | 1. The user turns on the app. |
| Postconditions: | 1. The system displays the list of relevant CCs with correct location. |
| Priority: |  |
| Frequency of Use: | often used. |
| Flow of Events: | 1. The user clicks the “search by location” button. 2. The location of the user detected by GPS is automatically entered on search engine . 3. The system validates the location. 4. The system verifies the location. 5. The system displays the list of relevant CCs. |
| Alternative Flows: | RA010.AC.3. Invalid location.   1. The user clicks “search by location” button again. 2. The GPS service detects the current location again and then the detected location is entered on the search bar. 3. returns to RA010.3 |
| Exceptions: | RA001.EX.2. Location detection failure.   1. There can be situations that the user’s current location cannot be detected. (e.g. inside the elevator or long tunnel) 2. The system shall shows error message which recommend entering another keywords. |
| Includes: | N/A |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | N/A |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | RA002 | | |
| Use Case Name: | check courses | | |
| Created By: | sehyun hwang | Last Updated By: | sehyun hwang |
| Date Created: | 10-02-2019 | Date Last Updated: | 24-02-2019 |

|  |  |
| --- | --- |
| Actor: | user, community centre |
| Description: | This use case allows users to check for courses. The system shows course list of the selected CC. If the user click specific course name, the system shall displays details of the course. |
| Preconditions: | 1. The user must be logged in. 2. The user selects the CC. |
| Postconditions: | 1. The system displays the list of courses. |
| Priority: |  |
| Frequency of Use: | very often |
| Flow of Events: | 1. The user select the specific community centre. 2. The system displays the list of courses. 3. The user clicks specific course name. 4. The system uses RA003 (display course details). |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: | RA003 display course details |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | have to check priority |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | RA003 | | |
| Use Case Name: | display course details | | |
| Created By: | sehyun hwang | Last Updated By: | sehyun hwang |
| Date Created: | 10-02-2019 | Date Last Updated: | 24-02-2019 |

|  |  |
| --- | --- |
| Actor: | user, community centre |
| Description: | This use case allows users to check for details of the course. The system shall displays details of the course such as course description, instructor, course schedule, location and vacancy. |
| Preconditions: | 1. The user search for courses. 2. The user clicks the course name. |
| Postconditions: | 1. The system displays the details of the course. |
| Priority: |  |
| Frequency of Use: | very often |
| Flow of Events: | 1. The use choose the course. 2. The system displays details of the course. 3. The user read the details precisely to decide whether he or she will register the course or not. |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: | N/A |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | N/A |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | RA004 | | |
| Use Case Name: | Login to SingPass | | |
| Created By: | sehyun hwang | Last Updated By: | sehyun hwang |
| Date Created: | 10-02-2019 | Date Last Updated: | 24-02-2019 |

|  |  |
| --- | --- |
| Actor: | user, government |
| Description: | This use case allows users to login to the SingPass. The user have to login to SingPass on the first page of the app. |
| Preconditions: | None. |
| Postconditions: | 1. The system shows the search engine page. |
| Priority: |  |
| Frequency of Use: | every time the user turns on the app |
| Flow of Events: | 1. The system requests entering ID and password from users. 2. The user enters his or her ID and password. 3. The system validates the entered ID and password. 4. The system shows search engine. |
| Alternative Flows: | RA004.AC.2.1 Forget ID or passwords   1. The user clicks “Find ID” or “Find password” button. 2. The SingPass account page opens. 3. The user tries to find ID or password via email or mobile phone. 4. returns to RA004.2   RA004.AC.2.2 Does not have SingPass account   1. The user clicks “Sign up” button. 2. The SingPass account page opens. 3. The user enters privacy information to sign up. 4. returns to RA004.2 |
| Exceptions: | RA004.EX.3. Invalid ID or password   1. The system shows the error message. 2. The system recommends trying again. 3. returns to RA004,2 |
| Includes: | N/A |
| Special Requirements: | N/A |
| Assumptions: | Users are eligible for SingPass.  Users already have SingPass account. |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | RA005 | | |
| Use Case Name: | Register course | | |
| Created By: | sehyun hwang | Last Updated By: | sehyun hwang |
| Date Created: | 11-02-2019 | Date Last Updated: | 25-02-2019 |

|  |  |
| --- | --- |
| Actor: | user, community centre |
| Description: | This use case allows users to register the course they’d like to learn. Users have to enter their personal information in all the required fields. |
| Preconditions: | 1. The user finds the course that he or she would like to learn. 2. The course has at least one vacancy left. |
| Postconditions: | 1. The registration process continues to next step “Make payment” |
| Priority: |  |
| Frequency of Use: | often |
| Flow of Events: | 1. The user requests registering the course. 2. The system check vacancy of the course. 3. The user enters personal information in all the required fields. 4. The user presses “Next” button. |
| Alternative Flows: | N/A |
| Exceptions: | RA005.EX.3. No vacancies left   1. The system shows error message. 2. The system shows the list of courses. 3. Return to RA003 (display course details) |
| Includes: | RA006 Make payment |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | N/A |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | RA006 | | |
| Use Case Name: | Make payment | | |
| Created By: | sehyun hwang | Last Updated By: | sehyun hwang |
| Date Created: | 11-02-2019 | Date Last Updated: | 25-02-2019 |

|  |  |
| --- | --- |
| Actor: | user, bank |
| Description: | This use case allows user to make payment for the course. The bank has to check whether the user has valid account and whether there are enough money in bankbook. The bank shall send payment results to the user. |
| Preconditions: | 1. The system successfully proceeds registration request. |
| Postconditions: | 1. The system shows the payment result. |
| Priority: |  |
| Frequency of Use: | not that often |
| Flow of Events: | 1. The user requests payment. 2. The user is taken to payment summary page which includes all information relating to payment. 3. The user must select a mode of payment. 4. The user inputs a CAPTCHA code, agrees to Terms and Conditions and then press “Check Out” button. 5. The system transmits request to the bank. 6. The bank verifies availability of payment. 7. The bank proceeds payment. 8. The bank transmits the payment result. 9. The system shows the payment result to the user. |
| Alternative Flows: | N/A |
| Exceptions: | RA006.EX.7. Payment failure   1. The bank shows reason of the payment failure. 2. The system transfers the result to the user. 3. The user tries another way of payment. |
| Includes: | N/A |
| Special Requirements: | Payment must be processed within a fixed time. |
| Assumptions: | The user has valid SingPass account. |
| Notes and Issues: | extended by RA007 “Cancel Payment” |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | RA007 | | |
| Use Case Name: | Cancel payment | | |
| Created By: | sehyun hwang | Last Updated By: | sehyun hwang |
| Date Created: | 11-02-2019 | Date Last Updated: | 25-02-2019 |

|  |  |
| --- | --- |
| Actor: | user, bank |
| Description: | There can be some cases that the user wants to retract his or her payment and to be given the money back. This use case allows users to cancel payment which is already processed. |
| Preconditions: | 1. The user already made the payment. |
| Postconditions: | 1. The user can be given his or her money back. |
| Priority: |  |
| Frequency of Use: | occasionally |
| Flow of Events: | 1. The user requests cancellation of the payment. 2. The bank verifies the request. 3. The bank confirms information relating to payment such as account, amount of payment, payment date, etc. 4. The bank gives the money back to user’s account. 5. The bank sends confirmation mail to the user. |
| Alternative Flows: | N/A |
| Exceptions: | RA007.EX.3. computational error of the bank   1. The bank cannot give the money back to the user. 2. The bank sends error message to the user. 3. The user checks the message and tries again later. |
| Includes: | N/A |
| Special Requirements: | The cancellation has to be processed within a fixed time after the request is accepted. |
| Assumptions: | The user already pay for the course with SingPass account. |
| Notes and Issues: | N/A |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | RA008 | | |
| Use Case Name: | Verify registration | | |
| Created By: | sehyun hwang | Last Updated By: | sehyun hwang |
| Date Created: | 11-02-2019 | Date Last Updated: | 25-02-2019 |

|  |  |
| --- | --- |
| Actor: | user, community centre |
| Description: | This use case allows the user to verify whether the payment was processed successfully or not. Also this use case informs the user of the final confirmation including course details, payment result. |
| Preconditions: | 1. The user proceeds payment. |
| Postconditions: | 1. The system shows the final result. |
| Priority: |  |
| Frequency of Use: | occasionally |
| Flow of Events: | 1. The system confirms the payment result. 2. The system puts payment information together. 3. The system shows the user to summary of the result. |
| Alternative Flows: | N/A |
| Exceptions: | RA008.EX.3. Network connection error   1. The system cannot send a confirmation email due to bad network. 2. The system displays error message. |
| Includes: | RA011. send confirmation email |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | N/A |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | RA011 | | |
| Use Case Name: | Send confirmation email | | |
| Created By: | sehyun hwang | Last Updated By: | N/A |
| Date Created: | 25-02-2019 | Date Last Updated: | N/A |

|  |  |
| --- | --- |
| Actor: | user, bank |
| Description: | This use case allows users to confirm the payment result. |
| Preconditions: | 1. The user made a payment. |
| Postconditions: | 1. The user receives confirmation email from the bank. |
| Priority: |  |
| Frequency of Use: | every time when payment proceeds. |
| Flow of Events: | 1. The user finishes making payment. 2. The bank proceeds payment process. 3. The bank finishes the process. 4. The bank send confirmation email to the user. |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: | N/A |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | N/A |