
Use Cases

for

Open-Jio

Version 3.0

Prepared by

Tio Guo Yong (Leader) (U2123181B)

Chen Zihang (Asst. Leader) (U2121486H)

Zhang Jing Wen (U2121853G)

Jayden Yeo He (U2120348J)

Nema Aarushi (U2120814C)

Team Fivver

5th November 2022

Revision History

Name	Date	Reason For Changes	Version
Use cases	30/8/2022	Created Use Cases	1.0
First check	31/8/2022	First revision	1.1
Minor addition	13/9/2022	Added some missing flows	1.2
Second revision	23/10/2022	Modified and added new functionalities	2.0
Third revision	5/11/2022	Final revision	3.0

Guidance for Use Case Template

Document each use case using the template shown in the Appendix. This section describes each section in the use case template.

1. Use Case Identification

1.1. Use Case ID

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

1.2. 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 a link to the target.
- Place an order for a CD with the updated software version.

1.3. Use Case History

1.3.1 Created By

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

1.3.2 Date Created

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

1.3.3 Last Updated By

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

1.3.4 Date Last Updated

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

2. Use Case Definition

2.1. 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.

2.2. 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.

2.3. 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 the task.

2.4. Postconditions

Describe the system's state after the use case execution. Number each postcondition. Examples:

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

2.5. 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.

2.6. Frequency of Use

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

2.7. Flow of Events

Provide a detailed description of the user actions and system responses that will take place during the execution of the use case under normal, expected conditions. This dialogue 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.

2.8. 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.

2.9. Exceptions

Describe any anticipated error conditions that could occur during the 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.

2.10. 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.

2.11. 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.

2.12. 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.

2.13. Notes and Issues

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

Use Cases

Use Case ID:	UC0001		
Use Case Name:	Register Account		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	The user creates an account by filling in information for username, name, password, birthdate, and email.
Preconditions:	-
Postconditions:	User's account successfully created.
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. The user presses the "Create account" button. 2. The user fills in the required information for username, name, password, birthdate, and email. 3. The user presses the "Create account" button. 4. App authenticates the information. 5. App redirects to the home page ui. 6. App displays the message "Account successfully created."
Alternative Flows:	<ol style="list-style-type: none"> 4. App finds the information is invalid. 5. Back to step 2.
Exceptions:	The user already has an account.
Includes:	-
Special Requirements:	App must be able to check information validity within 5 seconds.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC0002		
Use Case Name:	Login Account		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	The user logs into the app using their registered email and password.
Preconditions:	Valid email and password.
Postconditions:	The user's account successfully logged in.
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. In the login page, the user fills in their registered email and password. 2. The user presses the "Login" button. 3. App authenticates the username and password. 4. The user logs into their account's home page.
Alternative Flows:	<ol style="list-style-type: none"> 3. App finds that the email or the password is invalid. 4. Back to step 1.
Exceptions:	-
Includes:	-
Special Requirements:	Users must be able to log in to their account within 5 seconds.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC0003		
Use Case Name:	Logout Account		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	The user logs out of his/her account on the application if he/she wishes to log in to a different account
Preconditions:	-
Postconditions:	The user is able to log in using the same or another account
Priority:	-
Frequency of Use:	By the intention of the user
Flow of Events:	<ol style="list-style-type: none"> 1. In the profile page, the user presses the 'Logout' button. 2. The application system will log the user out of his/her current account.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	Users must be able to log out of their account within 5 seconds
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC0004		
Use Case Name:	Reset password		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	On the login page, if the user does not remember their password, they can reset it through email.
Preconditions:	The user has an account.
Postconditions:	The user's password is reset.
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. The user presses the "Forgot password" button. 2. App sends an email with a verification link to the user. 3. The user clicks the verification link. 4. The link opens the app. 5. The user is prompted to enter a new password. 6. The user enters a new password. 7. App checks that the password is valid. 8. Password is reset.
Alternative Flows:	<ol style="list-style-type: none"> 7. App checks that the password is invalid. 8. The user is prompted to enter a new password.
Exceptions:	6.EX.1 - The user's email is not linked to any valid account. 6.EX.2 - The user does not have access to the email he/she inputted.
Includes:	-
Special Requirements:	App must send the email within 10 seconds.
Assumptions:	The email belongs to the User's account.
Notes and Issues:	-

Use Case ID:	UC0005		
Use Case Name:	Display Activities Nearby		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	Application will display a list of activities that are located near the user's current location, available for him/her to join.
Preconditions:	-
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. Activities are queried wherever the user navigates to the home page. 2. Maps API is invoked and nearby activities within a 10km radius is displayed on the home page.
Alternative Flows:	<ol style="list-style-type: none"> 1. If the user has not previously granted permission to access location, the app will prompt the user again to grant permission. 2. If it is past 9pm, the app will display activities happening on the same day, as well as the following day.
Exceptions:	-
Includes:	UC0002
Special Requirements:	The user's device needs to have location tracking capability.
Assumptions:	The user has granted permission to the App to access location.
Notes and Issues:	-

Use Case ID:	UC0006		
Use Case Name:	Search Activity		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	User searches for upcoming activities initiated by other users (organisers).
Preconditions:	Valid activity name.
Postconditions:	List of relevant activities filtered by above.
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. The user clicks on the 'search bar' on home page. 2. The user clicks on 'activity' from drop down menu to search by activity. 3. The app displays the list of relevant activities.
Alternative Flows:	<ul style="list-style-type: none"> • In step 3, if the list is empty (no activities). <ol style="list-style-type: none"> 1. The app displays 'No activities found.'
Exceptions:	-
Includes:	-
Special Requirements:	The app must display the list of relevant activities filtered within 10 seconds.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC0007		
Use Case Name:	Search User		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	User searches for other users' profiles.
Preconditions:	Valid username.
Postconditions:	List of users filtered by above.
Priority:	-
Frequency of Use:	By the intention of the user
Flow of Events:	<ol style="list-style-type: none"> 4. The user clicks on the 'search bar' on the home page. 5. The user clicks on 'user' from the drop down menu to search by activity. 6. The app displays the list of relevant users.
Alternative Flows:	<ul style="list-style-type: none"> • In step 3, if the list is empty (no users). 2. The app displays 'No users found.'
Exceptions:	-
Includes:	-
Special Requirements:	The app must display the list of users filtered within 10 seconds.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC0008		
Use Case Name:	Create Activity		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	User (Organizer) can organize an activity that other Users can then search for and join.
Preconditions:	-
Postconditions:	The user has created an activity, and activity is added to the activities collection in the database.
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. User presses the “Create Activity” button on the navigation bar. 2. The user inputs the information for Activity title and description, category, location, time, the maximum number of participants or picture (optional). 3. App checks that all information is valid. 4. User clicks on ‘Post Activity’ button 5. Activity is created.
Alternative Flows:	<ol style="list-style-type: none"> 2. Information inputted is invalid. 3. The user is prompted to input the information again.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	<ol style="list-style-type: none"> 1. User enters one of the registered locations on the OneMap API.
Notes and Issues:	-

Use Case ID:	UC0009		
Use Case Name:	Update Activity		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	User (Organiser) can update an activity that they have created.
Preconditions:	Activity must exist in the database.
Postconditions:	The user has updated an activity, and activity is updated in the database.
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. User opens an activity that they have created. 2. User clicks on the 'update' button. 3. User can modify any field (title, description, location, time, max capacity).
Alternative Flows:	<ol style="list-style-type: none"> 4. Information inputted is invalid. 5. The user is prompted to input the information again.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	User enters one of the registered locations on the OneMap API.
Notes and Issues:	-

Use Case ID:	UC0010		
Use Case Name:	Cancel Activity		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	The user (Organiser) cancels the activity that he/she has organised.
Preconditions:	The user has created an activity that is still upcoming.
Postconditions:	Activity is cancelled.
Priority:	-
Frequency of Use:	By the intention of the user
Flow of Events:	<ol style="list-style-type: none"> 1. The user (Organizer) clicks the “Cancel Event” button in the page of the Activity he/ she has created. 2. The user is prompted to confirm the cancellation. 3. Activity is cancelled. 4. App sends a notification to all participants about the activity cancellation.
Alternative Flows:	<ol style="list-style-type: none"> 3. User chooses to not confirm the cancellation. 4. Activity is not cancelled.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	The activity being cancelled was created by the User.
Notes and Issues:	-

Use Case ID:	UC0011		
Use Case Name:	Save Activity		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	Users can save activities that they wish to attend but are not sure if they want to attend.
Preconditions:	-
Postconditions:	Activity is saved.
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on the 'save' button on the activity card. 2. Activity gets added to saved activities of the user.
Alternative Flows:	<ol style="list-style-type: none"> 1. Activity is not saved. 2. Activity gets deleted by the organiser.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	User has not joined the activity yet.
Notes and Issues:	-

Use Case ID:	UC0012		
Use Case Name:	Join Activity		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	User (Participant) joins an activity that another user posted.
Preconditions:	Other users have created activities to join.
Postconditions:	The participant joins the activity.
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. The user (participant) searches for an activity to join. 2. The user clicks the desired activity and clicks on the 'join' button. 3. App sends a notification to the organiser who will approve/reject the request. 4. When a request is approved, the user will receive a notification that they have joined the event..
Alternative Flows:	<ol style="list-style-type: none"> 1. User clicks on activity on the Home UI or Search List UI. 2. App sends a notification to the organizers and the user who just joined.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC0013		
Use Case Name:	Leave Activity		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	User (Participant) can leave an activity that he/ she has joined.
Preconditions:	The user has joined as a participant in an activity.
Postconditions:	The user is no longer a participant in the activity.
Priority:	-
Frequency of Use:	-
Flow of Events:	<ol style="list-style-type: none"> 1. User (Participant) chooses from a list of activities he/she has indicated to join. 2. User clicks the “Leave Activity” button. 3. App prompts the user to confirm the withdrawal. 4. User is no longer a participant for the activity. 5. (Assuming Organizer has approved User as Participant) App sends a notification to the Organiser to inform of participant leaving.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	Activity is still upcoming and not canceled yet. The user is a Participant and not Organiser.
Notes and Issues:	-

Use Case ID:	UC0014		
Use Case Name:	Add Friend		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	User can request to add another User as Friend.
Preconditions:	Users are not Friends with one another.
Postconditions:	Users become one another's Friends.
Priority:	-
Frequency of Use:	By the intention of the user
Flow of Events:	<ol style="list-style-type: none"> 1. User A clicks on "Add Friend" button on the profile page of User B. 2. User B receives requests from User A and clicks on "Accept". 3. User A and User B become Friends.
Alternative Flows:	<ol style="list-style-type: none"> 1. User B receives requests from User A and clicks on "Reject". 2. User A is notified that User B has rejected his/her request.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC0015		
Use Case Name:	Remove Friend		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	User can remove another User who is a 'Friend'.
Preconditions:	Users are Friends with one another.
Postconditions:	Users are no longer one another's Friends.
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. User A clicks on "Remove Friend" button on the profile page of User B. 2. User B is removed from User A's Friend list and User A is removed from User B's Friend list.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	User A and User B are friends from before.
Notes and Issues:	-

Use Case ID:	UC0016		
Use Case Name:	Reject Friend Request		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	User can remove another User who is a 'Friend'.
Preconditions:	Users are Friends with one another.
Postconditions:	Users are no longer one another's Friends.
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. User A clicks on "Reject Friend Request" button on the profile page of User B. 2. User B is not added to User A's Friend list.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC0017		
Use Case Name:	View Friends' Activities		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	User can view friends' activities.
Preconditions:	Users have friends, friends have joined an activity.
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks the "Friends" tab to view all friends and their current activity. 2. If the friend is participating in an activity, it will be reflected under his username.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC0019		
Use Case Name:	Chat		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	The user can send messages to another User.
Preconditions:	-
Postconditions:	The user sends a message to another User.
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. User selects another user to send a message to. 2. The user is prompted to type the message. 3. User presses the “Send” button. 4. The message is sent to the selected User. 5. App sends a notification to the selected User that he/ she has a new message.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC0020		
Use Case Name:	Edit Profile		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User
Description:	User edits the information displayed in their own profile page.
Preconditions:	-
Postconditions:	The user's information on the profile page is successfully updated.
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. The user presses the "Edit profile" button. 2. The user selects the required section to be changed: Profile photo, username, name and/ or description. 3. The user fills in the new information. 4. The user presses the "Save changes" button. 5. App authenticates the format of the information. 6. App redirects to profile page. 7. The user's information on the profile page is successfully updated.
Alternative Flows:	<ol style="list-style-type: none"> 5. App finds that the information is invalid. 6. Back to step 3.
Exceptions:	-
Includes:	-
Special Requirements:	App must be able to update information within 5 seconds.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UC0021		
Use Case Name:	Manage Participants		
Created By:	Fivver	Last Updated By:	Fivver
Date Created:	30th Aug 2022	Date Last Updated:	6th Nov 2022

Actor:	User (Organizer)
Description:	User (Organizer) can accept or reject requests to join the activity from other users. The user (Organizer) can also remove users that are already taking part in the activity.
Preconditions:	The user has created an activity.
Postconditions:	-
Priority:	-
Frequency of Use:	By the intention of the user.
Flow of Events:	<ol style="list-style-type: none"> 1. User(Participant) sends a request to join the activity by pressing the “Join Activity” button on the activity details page. 2. The user(Organizer) can choose to “Accept” the request from the other user. 3. The user(Participant) will be added to the list of participating Users on the activity page. 4. App sends a notification to the participant of the acceptance. 5. The user(Organizer) can choose to remove a participating User by pressing the “Remove” button next to the name. 6. App sends a notification to the user of the removal.
Alternative Flows:	<ol style="list-style-type: none"> 2. User(Organizer) can choose to “Reject” the request from the other user. 3. App sends a notification to the user of the removal.
Exceptions:	10.EX.1 - User(Organizer) cannot accept a request as the participant's limit of the activity has been reached.
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-