Use Cases

for

Yummy Tummy

Version 1.0 approved

Prepared by Grace Wong & Colin Hong

Super Smash Bros

Aug 25, 2021

Revision History

Name	Date	Reason For Changes	Version
Grace Wong	21 Oct 2021	Removed use case for “Remove Participant”	2

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.

1. Use Case Identification

1.1. 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.

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 link to 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 task.

2.4. 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.

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 of time.

2.7. 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.

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 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 Determineds) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

Use Case ID:	1		
Use Case Name:	Sign In		
Created By:	Colin Hong	Last Updated By:	Colin Hong
Date Created:	25/08/2021	Date Last Updated:	01/09/2021

Actor:	**User (Host, Invitee, Participant)**
Description:	**User** shall sign in to our website using their Google account via the Google Sign-In integration.
Preconditions:	1. Website is loaded to the Landing Page.
Postconditions:	1. **User** state changes from guest to logged in 2. Webpage displays the Dashboard which lists the upcoming events, previous events and a button to “Jio your friends!” (create a new event).
Priority:	High
Frequency of Use:	2000 times a week
Flow of Events:	1. **User** selects option to *Sign in with Google* 2. Google Sign-In integration authorizes **User’s** credentials 3. System verifies that Google account is in Firebase 4. If account is verified, System logs in the **User** and **User’s** Dashboard is displayed
Alternative Flows:	1.AC.1. New User 1. If the Google account is not found in Firebase, the System shall create a new identifier for the new **User**.
Exceptions:	1.EX.1. If **Firebase** is inaccessible 1. System displays an error message on the front end 2. System displays button to redirect **User** to the Landing Page
Includes:	N.A.
Special Requirements:	N.A.
Assumptions:	**User** uses English.
Notes and Issues:	1. Signup and sign in are the same when handled by the Google Sign-In integration 2. The System shall not directly handle the conditions when **Users** forgot their password or want to create a new Google account. All these are handled by the Google Sign-In integration 3. The Google Sign-In integration uses OAuth2.0 4. **User** has an internet browser to access webpage

Use Case ID:	2		
Use Case Name:	Create Event		
Created By:	Colin Hong	Last Updated By:	Colin Hong
Date Created:	25/08/2021	Date Last Updated:	31/08/2021

Actor:	**User (Host)**
Description:	**Host** creates an event to invite friends to a private meal gathering set for a specific time and date.
Preconditions:	1. **Host** is logged in.
Postconditions:	1. Event is created in **Firebase**. 2. URL to the event is generated for the Host to share with invitees. 3. Event Page (Host) shall be displayed
Priority:	High
Frequency of Use:	4000 times a week
Flow of Events:	1. **Host** selects option to create new event 2. **Host** enters the event title, date & time, maximum group size 3. **Host** is prompted to input their location. They may do so in the following ways: 3.1. Joining with their current location automatically by enabling location services for the Geolocation API to obtain their current location 3.2. Joining by entering their location manually using postal code 4. Host selects the “Create” option to create a new event 5. New event is stored into **Firebase** 6. System displays Event Page containing the details entered above 6.1. Curated location is displayed as “To be confirmed” until **Host** decides to *Initiate Location Search* (Use Case ID 5) 7. URL to the new event created shall be displayed and copied into the device clipboard of the **Host**
Alternative Flows:	N.A.
Exceptions:	2.EX.1. If **Firebase** is inaccessible 1. System displays an error message on the front end 2. System displays button to redirect **User** to the Landing Page
Includes:	N.A.
Special Requirements:	All date entries are recorded in DD/MM/YY, HH:MM format
Assumptions:	1. **Host’s** device is able to get their Geolocation 2. **Host** has a messaging application on their device to paste the event URL and share it with Invitees 3. **Host** has an internet browser to access webpage 4. **Host** uses English 5. **Host** location is in Singapore
Notes and Issues:	N.A.

Use Case ID:	3		
Use Case Name:	Join Event		
Created By:	Colin Hong	Last Updated By:	Colin Hong
Date Created:	26/08/2021	Date Last Updated:	26/08/2021

Actor:	**User (Invitee)**
Description:	**Invitee** receives an event URL from the **Host** and registers for gathering and inputs their location.
Preconditions:	1. Event has been created by **Host** and event URL has been sent to **Invitee**
Postconditions:	1. **Participant** and their location is appended to event details in Firebase 2. Event appended to **Participant's** upcoming events in Dashboard
Priority:	High
Frequency of Use:	20,000 times a week
Flow of Events:	1. **Invitee** receives an event URL from Host through text messaging application. 2. **Invitee** opens the URL in their browser. 3. **Invitee** is required to Sign in (see Use Case 1) 4. Browser displays the Event Page with the following details: 4.1. Event title 4.2. Date & time 4.3. A list of other Participants who are going for the event 5. If **Invitee** chooses to accept the invite by clicking a button to join with their location, they become a **Participant**. They may do so in the following ways: 5.1. Joining with their current location automatically by enabling location services for the Geolocation API to obtain their current location 5.2. Joining by entering their location manually using postal code
Alternative Flows:	3.AC.1. Waiting for other **Invitees** to respond **after** joining event 1. The Event Page is displayed on the **Participant's** browser with the list of other **Participants** who are going for the event. 1.1. This list is updated whenever the **Participant** refreshes the page 3.AC.2. **Host** has *Initiated Location Search* before **Invitee** has responded (see Use Case 5) 1. **Invitee** is redirected to the Link Expiration Page that explains to them that the link has expired and they are unable to join the event. 2. **Invitee** has an option to select to return to Dashboard.
Exceptions:	3.EX.1. **Invitee** inputs an invalid location when they join event 1. If invalid location is input, the page displays a system dialog that an invalid location has been entered

	2. **Invitee** has to re-enter their location using the process in Step 4 in the Flow of events. 3.EX.2. If **Firebase** is inaccessible 1. System displays an error message on the front end 2. System displays button to redirect **User** to the Landing Page
Includes:	*Sign In*
Special Requirements:	N.A.
Assumptions:	1. **Invitee's** device is able to get their Geolocation 2. **Invitee** has a messaging application on their device to receive the event URL from Host 3. **Invitee** has an internet browser to open event URL from Host 4. **Invitee** uses English 5. **Invitee** location is in Singapore
Notes and Issues:	N.A.

Use Case ID:	4		
Use Case Name:	Edit Event		
Created By:	Colin Hong	Last Updated By:	Colin Hong
Date Created:	30/08/2021	Date Last Updated:	01/09/2021

Actor:	**User (Host)**
Description:	**Host** decides to edit event details
Preconditions:	1. Event has been created by **Host** and event URL has been sent to **Invitees**
Postconditions:	1. Updated event details are displayed on the Event Page
Priority:	Medium
Frequency of Use:	10,000 times a week
Flow of Events:	1. In the Event Page, **Host** clicks on the "Edit" button to do either of the following: 1.1. With the existing filled in details of event title, date & time, and maximum group size, **Host** chooses to edit any of these details 2. **Host** saves the changes using the "Save" button and these changes are updated in **Firebase**.
Alternative Flows:	4.AC.1. Deleting an event 1. **Host** clicks on the "Delete" button to delete the event 2. Pop up is displayed to warn **Host** that event shall be deleted 3. Event is deleted from **Firebase** 4. Event is removed from the list of upcoming events in the Dashboard
Exceptions:	4.EX.1. **Host** reduces maximum group size to below current number of **Participants** who have joined event 1. System dialog displays on the edit page to notify **Host** that invalid maximum group size has been chosen, and prompts **Host** to re-enter valid maximum group size. 4.EX.2. If **Firebase** is inaccessible 1. System displays an error message on the front end 2. System displays button to redirect **User** to the Landing Page
Includes:	N.A.
Special Requirements:	N.A.
Assumptions:	1. **Invitee's** device is able to get their Geolocation 2. **Invitee** has an internet browser to open event URL from Host 3. **Invitee** uses English
Notes and Issues:	

Use Case ID:	5		
Use Case Name:	Initiate Location Search		
Created By:	Grace Wong	Last Updated By:	Grace Wong
Date Created:	28/08/2021	Date Last Updated:	31/08/2021

Actor:	**User (Host)**
Description:	**Host** closes the event by clicking the "Search" button, preventing more users from joining the event.
Preconditions:	1. Desired or maximum number of **Participants** has joined the event. 2. All **Participants** have inputted their location. 3. **Host** is on the Event Page
Postconditions:	1. Event State is changed to open to locked. 2. **Host** is directed to the Choosing Location Page to choose the event location. 3. Event title, number of **Participants**, date and time is displayed. 4. List of 5 Recommended Eateries, including their address, opening hours and ratings, is displayed to the **Host**.
Priority:	High
Frequency of Use:	4000 times a week
Flow of Events:	1. **Host** checks that all desired **Participants**' names are displayed on the Event Page. 2. **Host** selects "Search" option to search for location 3. Website algorithm calculates the midpoint of all **Users**' locations 4. Algorithm searches for 5 nearest dining location using Google Maps API 5. Choosing Location Page is displayed.
Alternative Flows:	N.A.
Exceptions:	5.EX.1. If **Firebase** is inaccessible 1. System displays an error message on the front end 2. System displays button to redirect **User** to the Landing Page 5.EX.2. If current time is past event's scheduled time 1. Event is deleted 5.EX.3. **Host** selects browser "back" button after selecting "Search" option 1. **Host** is redirected to Dashboard 2. If **Host** selects the event again, Host is directed to the Choosing Location Page.
Includes:	N.A.
Special Requirements:	N.A.
Assumptions:	1. **User** uses English. 2. Google Maps API is online and functional
Notes and Issues:	N.A.

Use Case ID:	6		
Use Case Name:	Choose Event Location		
Created By:	Grace Wong	Last Updated By:	Grace Wong
Date Created:	28/08/2021	Date Last Updated:	31/08/2021

Actor:	**User (Host)**
Description:	**Host** views 5 Recommended Eateries displayed and selects 1 location as the event location.
Preconditions:	1. **Host** is on the Choosing Location Page. 2. List of 5 Recommended Eateries, including their address, opening hours and ratings, is displayed to the **Host**. 3. "Select" button is displayed to the **Host**.
Postconditions:	1. The Event Page is updated to display the chosen event location to **Users** (both **Host** and **Participants**) along with a picture of the location, its address, opening hours and ratings. 2. Buttons to "Open in Google Maps" and "Copy Postal Code" are displayed below the location details for **Users** to get directions to the gathering location
Priority:	High
Frequency of Use:	4000 times a week
Flow of Events:	1. **Host** clicks to select a location displayed in the list and view its details. 2. Choosing Location Page displays the location details on the right side of the webpage, including a picture of the location, its address and ratings. 3. **Host** scrolls down the list and clicks on location suggestions to view details of the 5 locations on the list as desired. 4. **Host** clicks the "Select" button to choose the location displayed currently. 5. Event Page is updated to display Curated Location Page
Alternative Flows:	N.A.
Exceptions:	6.EX.1. If **Firebase** is inaccessible 1. System displays an error message on the front end 2. System displays button to redirect **User** to the Landing Page 6.EX.2. **Host** selects browser "back" button before selecting "Select" option 3. **Host** is redirected to Dashboard 4. If **Host** selects the event again, Host is directed to the Choosing Location Page
Includes:	N.A.
Special Requirements:	N.A.
Assumptions:	**User** uses English.
Notes and Issues:	N.A.

Use Case ID:	7		
Use Case Name:	View Directions to Event Location		
Created By:	Grace Wong	Last Updated By:	Grace Wong
Date Created:	28/08/2021	Date Last Updated:	31/08/2021

Actor:	**User**
Description:	**User** obtains directions to the event location.
Preconditions:	1. **Host** finalises location of event by selecting one location from the list of 5 locations generated by website algorithm. 2. **Host** is redirected back to Event Page 3. Event Page is updated to Curated Location Page 4. Buttons to "Open in Google Maps" and "Copy Postal Code" are displayed.
Postconditions:	1. **User** is redirected to Google Maps 2. Address of the event location is automatically loaded into Google Maps
Priority:	Medium
Frequency of Use:	10,000 times a week
Flow of Events:	1. **User** clicks on the "Open in Google Maps" button 2. **User** is redirected to Google Maps 3. Address of the event location is automatically loaded into the Google Maps
Alternative Flows:	7.AC.1. Usage of "Copy Postal Code" 1. **User** clicks on the "Copy Postal Code" button, the postal code of location is copied into the clipboard of the device 2. **User** manually opens the Google Maps and input the postal code into Google Maps, or utilises the postal code in other ways
Exceptions:	N.A.
Includes:	N.A.
Special Requirements:	N.A.
Assumptions:	1. **User** knows how to navigate and use Google Maps 2. **User** uses English. 3. **User** location is in Singapore
Notes and Issues:	1. The System shall not handle issues faced in Google Maps, such as requesting location services to be turned on, or ensuring network connection. 2. **User** is to click on the "Directions" button in Google Maps to generate possible routes to the event location. 3. The web version of Google Maps shall be displayed if the Google Maps application is not installed on the **User's** device.

Use Case ID:	8		
Use Case Name:	View Previous Events		
Created By:	Grace Wong	Last Updated By:	Grace Wong
Date Created:	28/08/2021	Date Last Updated:	31/08/2021

Actor:	**User**
Description:	**User** views details of previous events
Preconditions:	1. **User** has existing events that have expired.
Postconditions:	1. Event State changed from locked to expired 2. Dashboard is updated to display title, location, list of participants, date and time of the past event under the "Previous Events" list.
Priority:	Low
Frequency of Use:	4000 times a week
Flow of Events:	1. **User** refreshes the Dashboard or selects the "Yummy Tummy" option on the navigation bar. 2. Dashboard is updated to display title, location, list of participants, date and time of the past event under the "Previous Events" list, with the latest event at the top.
Alternative Flows:	N.A.
Exceptions:	8.EX.1. If **Firebase** is inaccessible 1. System displays an error message on the front end 2. System displays button to redirect **User** to the Landing Page
Includes:	N.A.
Special Requirements:	N.A.
Assumptions:	**User** uses English.
Notes and Issues:	1. Event shall not appear in the "Previous Events" list if the event is deleted before the time of the event. 2. Event details cannot be edited by the **Host** after the Event State is set to expired.

Use Case ID:	9		
Use Case Name:	Sign Out		
Created By:	Grace Wong	Last Updated By:	Grace Wong
Date Created:	31/08/2021	Date Last Updated:	31/08/2021

Actor:	**User**
Description:	**User** signs out of the website.
Preconditions:	1. **User** is logged in to the website.
Postconditions:	1. **User** state changes from logged in to guest 2. **User** is redirected to landing page
Priority:	Low
Frequency of Use:	2000 times a week
Flow of Events:	1. **User** selects the “Sign Out” option 2. Authentication token is deleted from browser cookies
Alternative Flows:	N.A.
Exceptions:	N.A.
Includes:	N.A.
Special Requirements:	N.A.
Assumptions:	**User** uses English.
Notes and Issues:	N.A.

Use Case ID:	10		
Use Case Name:	View Upcoming Events		
Created By:	Colin Hong	Last Updated By:	Colin Hong
Date Created:	01/09/2021	Date Last Updated:	01/09/2021

Actor:	**User**
Description:	**User** views details of upcoming events
Preconditions:	**User** has existing events with an open Event State.
Postconditions:	N.A.
Priority:	Medium
Frequency of Use:	4000 times a week
Flow of Events:	1. **User** refreshes the Dashboard or selects the "Yummy Tummy" option on the navigation bar. 2. Dashboard is updated to display title, location, list of participants, date and time of the upcoming event under the "Upcoming Events" list, with the oldest event at the top.
Alternative Flows:	11.AC.1 If upcoming event is deleted by **Host** (4.AC.1) 1. Dashboard is refreshed 2. Upcoming event is removed from **User's** "Upcoming Events" list 11.AC.2 If upcoming event is edited by **Host** (Use Case ID 4) 1. Dashboard is refreshed 2. Event details are updated in Dashboard.
Exceptions:	11.EX.1. If **Firebase** is inaccessible 3. System displays an error message on the front end 4. System displays button to redirect **User** to the Landing Page
Includes:	N.A.
Special Requirements:	N.A.
Assumptions:	**User** uses English.
Notes and Issues:	N.A.