Use Cases

for

Spotify App (TBD)

Version 1.0

Prepared by Liam DeBeasi, Robin Liu, Lang Gao, Matthew Stoll

CS411 Fall 2016

October 4th, 2016

1. 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.

2. Use Case Identification

1. Use Case ID

Give each use case a unique integer sequence number identifier. Alternatively, use a hierarchical form: X.Y. Related use cases can be grouped in the hierarchy.

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.

3. Use Case History

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

2. Date Created
Enter the date on which the use case was initially documented.

3. Last Updated By
Supply the name of the person who performed the most recent update to the use case description.

4. Date Last Updated
Enter the date on which the use case was most recently updated.

3. Use Case Definition

1. Actors

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 that will be initiating this use case and any other actors who will participate in completing the use case.

2. Trigger

Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.

3. 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.

4. 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.

5. 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.

6. Normal Flow

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. The normal flow is numbered "X.0", where "X" is the Use Case ID.

7. Alternative Flows

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative flow, and describe any differences in the sequence of steps that take place. Number each alternative flow in the form "X.Y", where "X" is the Use Case ID and Y is a sequence number for the alternative flow. For example, "5.3" would indicate the third alternative flow for use case number 5.

8. 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. If the use case results in a durable state change in a database or the outside world, state whether the change is rolled back, completed correctly, partially completed with a known state, or left in an undetermined state as a result of the exception. Number each alternative flow in the form "X.Y.E.Z", where "X" is the Use Case ID, Y indicates the normal (0) or alternative (>0) flow during which this exception could take place, "E" indicates an exception, and "Z" is a sequence number for the exceptions. For example "5.0.E.2" would indicate the second exception for the normal flow for use case number 5.

9. 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.

10. 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.

11. Frequency of Use

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

12. Business Rules

List any business rules that influence this use case.

13. 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.

14. 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.

15. 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 List

ID	*Primary Actor*	*Use Case Title*
1.0	Logged In User	Select an Action
2.0	Logged In User	Select “Party” Action
3.0	Logged In User	Enable Discover Mode
4.0	Not Logged In User	Login to app

Use Case Template

Use Case ID:	1.0		
Use Case Name:	Select an Action		
Created By:	Liam DeBeasi, Robin Liu, Lang Gao, Matthew Stoll	Last Updated By:	Liam DeBeasi
Date Created:	9/30/2016	Date Last Updated:	10/5/2016

Actors:	A logged in user
Description:	Users will be presented with a set of "actions" they want to complete. These actions could be things like "sleep", "exercise", or "party". The user can choose at most one action (i.e. Users cannot mix actions together). If the user does not select "Party", the app will generate a playlist of your own music given constraints set by the action. For example, a constraint for the "Sleep" action could be that the music must be within a low decibel range.
Trigger:	A user goes to the main page of the application
Preconditions:	1. The user must be logged in and have already authenticated with Facebook and Spotify

Postconditions:	1. Page layout changes to reflect the action the user selected 2. Page displays song information and playlist info generated for the selected action
Normal Flow:	1.0: User lands on the main page and is presented with a set of actions to choose from 1.1: User selects an action and the app generates a playlist for the action 1.2: User can begin playing music, share the playlist with friends, or leave the app
Alternative Flows:	1.1 If the user selects the "Party" option, they will be prompted to select people from a list of their Facebook friends before the app generates a playlist
Exceptions:	

Includes:	Other use cases that are included are the “Enable Discover Mode” and “Select Party Action” cases. Each use case offers slightly different functionality and implementation but share the core feature of listening to music.
Priority:	Implementing the various actions is the top priority, as it is the most basic functionality of the application
Frequency of Use:	Each user will likely choose one action per session (1-3 hours)
Business Rules:	
Special Requirements:	For “Party” mode to be fully functional, a user must have Facebook friends who have linked their accounts with Spotify in order to access public playlists and generate a “Party” playlist
Assumptions:	Assume a user has a Facebook account
Notes and Issues:	

Revision History

Name	**Date**	**Reason For Changes**	**Version**
Robin Liu	10/4/2016	Initial work	1.0
Liam DeBeasi	10/5/2016	Add trigger, flow, pre-condition, alternative flows, assumptions	1.0

Use Case ID:	2.0		
Use Case Name:	Select “Party” Action		
Created By:	Liam DeBeasi, Robin Liu, Lang Gao, Matthew Stoll	Last Updated By:	Lang Gao
Date Created:	9/30/2016	Date Last Updated:	10/5/2016

Actors:	Logged in User
Description:	The user selects the “Party” action from the set of available actions. The application proceeds to generate a playlist of songs from the Facebook / Spotify friends the user selected from a list of available friends.
Trigger:	User selects the “Party” feature
Preconditions:	1. User is logged in. 2. User has friends added on Facebook or Spotify.
Postconditions:	1. A playlist is created from songs that the user’s friends like or have been recommended 2. The playlist is accessible to the user 3. The user can share the playlist with other people via a link
Normal Flow:	2.0 User selects “Party” feature 2.1 User is presented with a list of friends to choose from 2.2 User selects a set of friends from the list and presses continue 2.3 App generates a playlist based on the music that the selected friends like and returns the playlist along with a link to share it to the user
Alternative Flows:	
Exceptions:	If the user’s friends all have fresh Spotify accounts without any music information on them the returned playlist will be generated from Spotify’s default recommended songs
Includes:	Might include “create playlist” case if we decide it deserves its own use case

Priority:	Medium. It is only one feature among the many that our app will offer, and while important, requires other pieces (friends, playlists) to be created before it will work properly
Frequency of Use:	Unknown. Will vary from user to user.
Business Rules:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Revision History

Name	**Date**	**Reason For Changes**	**Version**
Lang Gao	10/5/2016	Initial case information	1.0

Use Case ID:	3.0		
Use Case Name:	Enable “Discover” Function		
Created By:	Liam DeBeasi, Robin Liu, Lang Gao, Matthew Stoll	Last Updated By:	Lang Gao
Date Created:	9/30/2016	Date Last Updated:	10/31/2016

Actors:	Logged in User
Description:	The user selects the “Discover” action from the set of available actions. The application proceeds to generate music suggestions based on the user’s most listened to genres, with music that may or may not be in the user’s library.
Trigger:	User toggles on discover mode
Preconditions:	1. User is logged in
Postconditions:	1. The user is presented with music suggestions based on the user’s most listened to / highest rated genres
Normal Flow:	3.0. User toggles on “Discover” feature 3.1. App looks through the user’s most played songs and highest rated songs and gathers data on which genres the user likes the most 3.2. App looks through currently trending / top songs for the genres that the user likes the most 3.3. App presents the user with a few songs from each of his/her top categories to listen to 3.4. App occasionally begins the process again by itself to generate an updated list of songs to show the user
Alternative Flows:	3.0. User toggles off “Discover” feature 3.1. App removes suggested music from the user’s view
Exceptions:	
Includes:	
Priority:	Relatively high. This feature will be used for various other components of our application.

Frequency of Use:	Variable, depending on if the user decides to turn on discover mode or not, and how long they leave it on.
Business Rules:	
Special Requirements:	
Assumptions:	
Notes and Issues:	If the user has a Spotify account without any music in it the default recommendations will be shown.

Revision History

Name	**Date**	**Reason For Changes**	**Version**
Lang Gao	10/31/2016	Initial case information	1.0

Use Case ID:	4.0		
Use Case Name:	User logs in		
Created By:	Liam DeBeasi, Robin Liu, Lang Gao, Matthew Stoll	Last Updated By:	Lang Gao
Date Created:	9/30/2016	Date Last Updated:	10/31/2016

Actors:	Anonymous user (not logged in)
Description:	A user attempts to login
Trigger:	Use arrives at the login page
Preconditions:	
Postconditions:	If the user inputs a valid username and password, the user is show his or her account. If not the user is returned to the login screen and notified of a failed authentication.
Normal Flow:	4.0 The user selects to login with either their Facebook or Spotify account 4.1 The user enters their username and password 4.2 The app sends the information to the corresponding site to authenticate 4.3 The user's information is authenticated, and the app allows the user into their profile home page
Alternative Flows:	4.3 The user's information is incorrect, and the app returns them to the login page and informs them of a failed login

Exceptions:	
Includes:	
Priority:	Medium – high. This function does not directly tie in with any of the other functions of our app, but completing this would allow us to more easily get actual data to test the rest of our app on instead of creating false data sets.
Frequency of Use:	Very high. Users must login each time they access the site.
Business Rules:	
Special Requirements:	Facebook / Spotify must be accessible for user authentication.
Assumptions:	
Notes and Issues:	

Revision History

Name	**Date**	**Reason For Changes**	**Version**
Lang Gao	10/31/2016	Initial case information	1.0