# Use cases v1

James Angel, Harrison McDonagh, Kevin Ibelgauptas, Shao Wen Young

12/10/2020

Actors:

Night clubbers

Person at a bar

The DJ

Stakeholder:

Bar or Club Owner

Preconditions:

The DJ has set up a virtual room for the bar/night club on our application.

When a member of the public walks into a bar or club, they will be able to scan a QR code or copy a URL allowing them to enter a virtual room.

Once they have joined the virtual room, the user will then be asked to login to their Spotify account.

When they login, data of the user’s favourite music will be sent to our application.

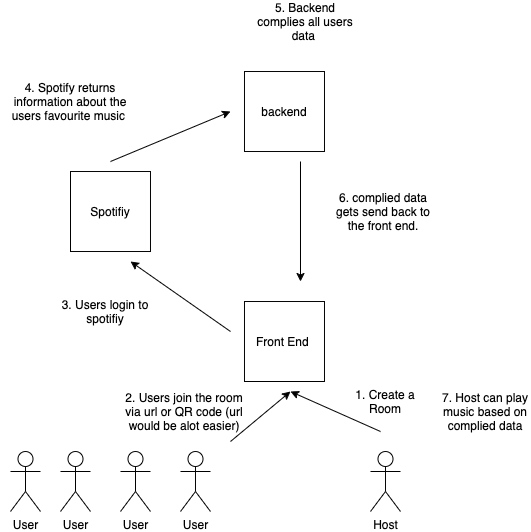
Once this is done the user will be shown a message telling them they can close the application.

Once the data is in our application, the backend of our website will compile the new data with other users in the virtual room.

This compiled data will then be sent to the host of the virtual room.

The host will then be able to play and queue songs based on the data.

Below shows a diagram visualising this process.



# User requirements

Template format

Scenario: Some determinable business situation

Given some precondition

And some other precondition

When some action by the actor

And some other action

And yet another action

Then some testable outcome is achieved

And something else we can check happens too

Scenario: User can send their Spotify data using a QR code

Given a user enters the club

When they scan the provided QR code

Then they are asked to login to Spotify

Scenario: User can send their Spotify data using a URL

Given a user enters the club

When they visit the provided URL

Then they are asked to login to Spotify

Scenario: User’s Spotify data is sent to the host

Given a user is prompted to login to Spotify

When they enter the correct credentials for their account

And accept their data being shared to the app

Then their Spotify data is sent to the host of the virtual room

Scenario: Host can access a user’s Spotify data

Given a user has sent their Spotify data using the app

When the host checks the app

Then the data displayed in the app will be updated with the user’s data

Scenario: Host can queue songs to a Spotify playlist using the app

Given the app receives and displays users’ data

When the host clicks the button to queue up a song

Then the song is added to the queue in the Spotify playlist

Scenario: Top songs that are the same for multiple users appears at the top of the host view

Given the app receives and displays users’ data

And there are top songs that are the same for more than one user

When host display is updated

Then the duplicate top songs appear at the top in order of frequency

## Actors

### Venue/Bar

The venue or bar is a location in which user would make use of our application

### Host/DJ

The host/dj is the user of the application who is in charge of playing music at the venue

### Customer

The main user of the application is anyone at the venue or bar who wants to share their music preferences. All customers of the application must have a spotify account.

### Spotify

Spotify is the application we will connect to, to get customer preferences and to allow the host to queue up songs.

For example, “As a <type of user>, I want <some goal> so that <some reason>.” ​

## User Stories

* As the DJ at a venue/bar (host) I want to start a virtual room, so that customers can join to share their music preferences.
* As a customer, I want to share my music taste via QR code or URL so that I can listen to music I enjoy in the venue/bar I am at.
* As a customer i would like to have my Spotify data shared with AllTunes so that the host can get my music preferences.
* As a host, I want to have access to a customer’s choice of genre/taste of music, so that I can queue their songs up in the Spotify queue to be played next.
* As a host, I want to be able to see the songs listed in order of most popular, so that I can play the most popular music.
* As a host, I want to queue songs displayed in the app so customers can listen to the songs I’ve queued.

## 

## Use Case

|  |  |
| --- | --- |
| *Use case name* | Host can create a virtual room |
| *Participating actors* | Host/DJ  Spotify |
| *Flow of events:*  *Normal flow* | 1. Host enters url of application 2. Host lands on landing page 3. The host clicks “create room” button 4. The customer is redirected to spotify to login 5. The customer logs into spotify 6. The customer is redirected back to application 7. The host sees a new room with no customers in and is shown a QR code to allow users to join. |
| *Flow of events:*  *Alternative flow* |  |
| *Pre-condition* | N/A first step |
| *Post-condition* | A virtual room has been created. |

|  |  |
| --- | --- |
| *Use case name* | Customers joins the room |
| *Participating actors* | Customer  Club/Venue |
| *Flow of events:*  *Normal flow* | 1. A customer arrives to a Club/Venue  2. The customer scans a QR code  3. The customer arrives at a landing page  4. The customer hits the “login” button  5. The customer is redirected to spotify to login  6. The customer logs into spotify  7. The customer is redirected back to application  8. The customer joins the virtual room. |
| *Flow of events:*  *Alternative flow* | Customer uses URL   1. A customer arrives to a Club/Venue 2. The customer enters a url 3. The customer arrives at a landing page 4. The customer hits the “login” button 5. The customer is redirected to spotify to login 6. The customer logs into spotify 7. The customer is redirected back to application 8. The customer joins the virtual room. |
| *Pre-condition* | The club/venue has set up a room, and they have made the QR or URL available. |
| *Post-condition* | The customer is then in the virtual room. |

|  |  |
| --- | --- |
| *Use case name* | Customer shares Spotify data. |
| *Participating actors* | Spotify  Host  Customer |
| *Flow of events:*  *Normal flow* | 1. Customer joins virtual room  2. Customers music preference is sent to application  2. Customer is shown room host spotify user id  3. Customer is shown a list of queued songs |
| *Flow of events:*  *Alternative flow* | Customer has not provided permissions for application to receive data from Spotify.   1. User joins the room. 2. Customer denies permission to accept application from retrieving data from the Spotify app. 3. Host does not have access to the user's Spotify data. |
| *Pre-condition* | Customer has logged into to spotify |
| *Post-condition* | Host receives the user's Spotify data. |

|  |  |
| --- | --- |
| *Use case name* | Customer’s spotify data is sent to the host |
| *Participating actors* | Host/DJ  Customer  Spotify |
| *Flow of events:*  *Normal flow* | 1. The customer has joins the virtual room  2. Application gets information about customers' music from spotify.  3. The host can see a new customer has joined  4. The host can see data about the customers music preferences. |
| *Flow of events:*  *Alternative flow* | The customer is a new spotify user with no customised user data   1. The customer has joined the virtual room 2. The host can see a new user has joined 3. No data is sent to the host/DJ as the user does not have any favorite music data. |
| *Pre-condition* | Host has created a room and the customer has scanned the QR code/used the URL and has logged into spotify and joined the virtual room. |
| *Post-condition* | The customers spotify data has been received by the host. |

|  |  |
| --- | --- |
| *Use case name* | Music is shown in order of most popular |
| *Participating actors* | Host  Customers |
| *Flow of events:*  *Normal flow* | 1. Host is shown a list of most popular tracks 2. A new Customer joins the room 3. The hosts list of tracks is updated to reflect the new users music preferences. |
| *Flow of events:*  *Alternative flow* | None of the Customers in the room listen to similar music.   1. Multiple Customers join the room 2. The system uses a popularity value given by spotify to order the list. |
| *Pre-condition* | A host has created a virtual room and customers have joined. |
| *Post-condition* | The host has a list of songs based on the popularity of the customers in the room. |

|  |  |
| --- | --- |
| *Use case name* | Queueing songs from the app to the Spotify queue to be played next |
| *Participating actors* | Host  Room  Spotify |
| *Flow of events:*  *Normal flow* | 1. The collated data (top songs) in the room is shown in the app as a list 2. The host clicks on the queue button next to a song in the list 3. The song is added to the Spotify queue to be played next 4. The song is removed from the list |
| *Flow of events:*  *Alternative flow* |  |
| *Pre-condition* | A customer has sent their data to the app |
| *Post-condition* | The songs the host has queued are present in the Spotify playlist  The songs queued are no longer in the list in the app |