|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1 | | |
| Use Case Name: | Create Account | | |
| Created By: | Bennett Dierckman | Last Updated By: | Emma Smith |
| Date Created: | 10/17/2017 | Date Last Updated: | 10/22/2017 |
| Actors: | User | | |
| Description: | Potential user seeks to create his/her account. | | |
| Trigger: | User attempts to access application features, but must sign up. | | |
| Preconditions: | MusicMedley application is open on device. | | |
| Postconditions: | 1. Default User Profile created.  2. Database updated to accommodate new user information. | | |
| Normal Flow: | **1.0 Create a new user account**  1. User accesses application and does not have an account.  2. User taps the “Create Account” button.  3. User enters their name, date of birth, email, password in each corresponding textEdits.  4. User accepts Terms and Conditions.  5. User presses the “Create” button.  6. User directed to default profile page. | | |
| Alternative Flows: | **1.1 New User decides to cancel creating account** (After Step 2,3, 4, or 5)  1. User changes mind and taps “Cancel” button to return to home screen.  **1.2 User has an existing account** (After Step 5)  1. User is informed that the username or email is already taken.  2. User taps the “Cancel” button to return to home screen.  **1.3 User is under the age of 13**. (After Step 5)  1. User is notified that they are not of age.  2. User is directed to the homepage.  **1.4 User signs up as an organization/business**. (After Step 3)  1. User checks box for business or organization.  2. User enter business or organization name.  3. User can choose to upload correctly formatted advertisements.  4. User selects advertisement package.  5. User links credit card by inputting information into secure textEdits.  6. Return to Step 5. | | |
| Exceptions: | **1.0.E.1 After Step 1.0.5, User inputs incorrect data type for password.**  1. System notifies user that the information in the password textEdit holds invalid characters.  2. User is redirected to the “Create Account” page.  **1.0.E.2 After Step 1.0.5, User enters invalid character length in password textEdit.**  1. System notifies User that the information in the password textEdit holds too few or too many characters.  2. System redirects User to the “Create Account” page.  **1.0.E.3 User taps the “Create” button with empty fields.**  1. System notifies User that there are incomplete required fields.  2. System redirects User to “Create Account” page.  **1.4.E.1 User registered as an organization/business fails to upload correctly formatted advertisements.**  1. System notifies User that they must re-upload a properly formatted advertisement.  2. System informs User how to properly format advertisements. | | |
| Includes: | NA | | |
| Priority: | High | | |
| Frequency of Use: | One instance per user. | | |
| Business Rules: | Don't let 13 and under sign up.  Terms and conditions. | | |
| Special Requirements: | User is capable of deactivating their account at any time.  User accepts Terms and Conditions. | | |
| Assumptions: | The application has been downloaded from Google Play. | | |
| Notes and Issues: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2 | | |
| Use Case Name: | User Login | | |
| Created By: | Bennett Dierckman | Last Updated By: | Emma Smith |
| Date Created: | 10/18/17 | Date Last Updated: | 10/22/17 |
| Actors: | User | | |
| Description: | User enters his/her credentials to login to his/her account via the application login page. | | |
| Trigger: | User attempts to access application features, but must login. | | |
| Preconditions: | 1. User is not currently logged-in. | | |
| Postconditions: | 1. User has access to their profile page.  2. User is capable of turning on Party Discover Mode.  3. User capable of creating a party. | | |
| Normal Flow: | **2.0 User Login**  1. User enters username in username textEdit.  2. User enters password in password textEdit.  3. User taps the “Login” button.  4. User is successfully logged in.  5. User is directed to their profile page. | | |
| Alternative Flows: | **2.1 User enters correct username but incorrect password (**After Step 3)  1**.** System notifies User thatlogin was not successful.  2. System notifies User that the password entered does not match password for username in DB.  3. User redirected to login page.  4. User re-enters login credentials.  5. User successfully logs in.  **2.2 User forgets password**  1**.** System notifies User thatlogin was not successful.  2. System notifies User that the password entered does not match password for username in DB.  3. User cannot remember their password.  4. User taps the “Recover Password“ button.  5. System notifies User that the recover password email has been sent.  6. User resets password through email.  7. User re-opens app to arrive at login page.  8. Return to Step 1. | | |
| Exceptions: | **2.0.E.1 Username does not exist**  1. System notifies User that the given username is not in the application’s database.  2. User is prompted to create account. | | |
| Includes: | Create Account | | |
| Priority: | High | | |
| Frequency of Use: | Every time account is accessed on a new device or every time a user logs out and decides to log back in (semi-frequent). | | |
| Business Rules: | NA | | |
| Special Requirements: | User is able to ‘Cancel’ or ‘Create Account’ and ‘Recover Account’ through the duration of the login process. | | |
| Assumptions: | 1. The application has been downloaded from Google Play, and is open on the device.  2. User has an account. | | |
| Notes and Issues: | 1. Potential login with Facebook.  2. User remains logged in until they decide to logout. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3 | | |
| Use Case Name: | User Logout | | |
| Created By: | Bennett Dierckman | Last Updated By: | Emma Smith |
| Date Created: | 10/18/17 | Date Last Updated: | 10/22/17 |
| Actors: | User | | |
| Description: | User decides to logout of the MusicMedley application. | | |
| Trigger: | User no longer desires to be logged into the application. | | |
| Preconditions: | 1. User is currently logged-in. | | |
| Postconditions: | 1. User is no longer logged-in.  2. User is redirected to login screen. | | |
| Normal Flow: | **3.0 User Logout**  1. User taps hamburger menu in top left corner of the application.  2. User selects the ‘Logout’ button.  3. User is asked, “Are you sure you want to logout?”.  4. User selects the “Confirm” button. | | |
| Alternative Flows: | **3.1 User decides to cancel logout** (After Step 3)  1. User selects the “Cancel” button.  2. User remains logged-in. | | |
| Exceptions: | NA | | |
| Includes: | User Login | | |
| Priority: | High | | |
| Frequency of Use: | Every time a user decides to Logout (semi-frequent depending on User). | | |
| Business Rules: | NA | | |
| Special Requirements: | NA | | |
| Assumptions: | The application has been downloaded from Google Play, and is open on user’s device. | | |
| Notes and Issues: | Logging out option *must be* apparent to the user. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 4 | | |
| Use Case Name: | Manage Profile | | |
| Created By: | Bennett Dierckman | Last Updated By: | Emma Smith |
| Date Created: | 10/18/17 | Date Last Updated: | 10/22/17 |
| Actors: | User | | |
| Description: | The user wants to manage his/her profile features. | | |
| Trigger: | 1. User desires to customize profile information.  2. User desires to be more recognizable to friends | | |
| Preconditions: | 1. The user is currently on their profile page. | | |
| Postconditions: | 1. User has an updated profile page. | | |
| Normal Flow: | **4.0 User Manages Profile**  1. User taps the “Edit” icon on their profile page. 2. User is directed to Manage Profile view. 3. User selects “Change Profile Picture” button. 4. User selects option to select from photo from their camera roll. 5. Users selection photo from their camera roll. 6. User selects the Bio editText. 7. User enters information into the bio section with a limit of 125 characters.  8. User taps the “Update Profile” button. 9. The User is directed to their updated profile page. | | |
| Alternative Flows: | **4.1 User Takes Profile Picture with Camera** (After Step 3)  1. User selects option to take photo with device camera.  2. User captures profile picture with device’s camera.  3. Return to Step 6.  **4.2 User Selects Manage Account Information** (After Step 1)  1. User selects the “Manage Account Information” button.  2. User taps the “Change Password” button.  3. User prompted to enter current password.  4. User enters current password to confirm their identity.  5. System validates that the current password is correct.  6. User enters new password in the editText box.  7. User taps the “Change Email” button.  8. User enters new email in the editText box.  9. User taps the “Update Account Information” button.  10. User is prompted to enter their password to confirm identity.  11. User enters password in the editText box.  12. User taps the “Confirm” button.  13. User is redirected to the Manage Profile view. | | |
| Exceptions: | **4.0.E.1 User Keeps Current or Default Profile Information**  1. User taps the “Cancel” button.  2. User is redirected back to their profile page.  **4.2.E.1 User Enters Incorrect Password**  1. System notifies User that the password entered is incorrect.  2. System prompts User to re-enter password.  3. User re-enters their current password in the textEdit box.  4. Return to Step 4.2.5. | | |
| Includes: | View Profile | | |
| Priority: | Medium | | |
| Frequency of Use: | Frequency will vary depending on User’s preference. Some User’s will manage their profile more frequently than others. | | |
| Business Rules: | NA | | |
| Special Requirements: | Editing any account information (alternative flow 4.2) will require the user to enter their password. | | |
| Assumptions: | 1. The application has been downloaded from Google Play, and is open on user’s device.  2. The User has an account and is logged in. | | |
| Notes and Issues: | Users are not allowed to change their birthday. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 5 | | |
| Use Case Name: | Add Friends | | |
| Created By: | Bennett Dierckman | Last Updated By: | Emma Smith |
| Date Created: | 10/18/17 | Date Last Updated: | 10/22/17 |
| Actors: | User | | |
| Description: | User has the ability to search for friends and add them to their friend list. | | |
| Trigger: | 1. The User desires to connect with their friends through the application. | | |
| Preconditions: | 1. User is on profile page. | | |
| Postconditions: | 1. User has successfully added a friend to their friend list. | | |
| Normal Flow: | **5.0 User Adds Friends**  1. User A selects the hamburger menu. 2. User A selects their friend list within the menu.  3. User A taps the search bar at the top of friends list. 4. User A types user B’s username. 5. User A selects user B from the list of similar usernames. 6. User A’s friend request is sent to User B. 7. User B receives friend request. 8. User B accepts friend request. 9. User A is notified of friend request’s approval. 10. User A now has the ability to view user B’s profile, and vice versa. | | |
| Alternative Flows: | **5.1 Friend request declined** (After Step 7) 1. User B declines user A’s friend request. 2. User A is still not able to view user B’s profile. | | |
| Exceptions: | **5.0.E.1 User Decides not to Add Any Friends**  **5.0.E.2 User Cannot Find Desired Username** (During Step 5)  1. User A does not see User B’s username among possibilities.  2. User A double checks spelling of User B’s username.  3. Return to Step 4. | | |
| Includes: | NA | | |
| Priority: | Low | | |
| Frequency of Use: | Frequency dependent on Users preferences. | | |
| Business Rules: | NA | | |
| Special Requirements: | System can curate a list of similar usernames. | | |
| Assumptions: | 1. The application has been downloaded from Google Play, and is open on user’s device.  2. The user has created an account.  3. The user is logged-in. | | |
| Notes and Issues: | Determine how username query will populate search results. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 6 | | |
| Use Case Name: | Delete Friends | | |
| Created By: | Bennett Dierckman | Last Updated By: | Emma Smith |
| Date Created: | 10/19/17 | Date Last Updated: | 10/22/17 |
| Actors: | User | | |
| Description: | User desires to remove friends from their profile’s friends list. | | |
| Trigger: | 1. User no longer desires to include a specific user in their friend list. | | |
| Preconditions: | 1. User has friends in friends list | | |
| Postconditions: | 1. User has fewer friends in friends list.  2. Specified friend is removed. | | |
| Normal Flow: | **6.0 User Deletes Friend**  1. User selects the hamburger menu. 2. User selects their friend list within the menu.  3. User selects the friend they wish to defriend. 4. User prompted, “Are you sure you want to delete [username]?”. 5. User taps the “Delete” button. 6. User’s friend list is updated to not include the recently deleted friend. 7. Deleted friend’s friend list is updated to no longer include user. | | |
| Alternative Flows: | **6.1 User Decides Not to Delete Friend** (After Step 4)  1. User taps the “Cancel” button.  2. User is redirected to their unchanged friends list. | | |
| Exceptions: | NA | | |
| Includes: | None | | |
| Priority: | Low | | |
| Frequency of Use: | Less Frequent | | |
| Business Rules: | NA | | |
| Special Requirements: | The System is able to remove friends from both users friends list. | | |
| Assumptions: | 1. The application has been downloaded from Google Play, and is open on user’s device.  2. The user has created an account.  3. The user is logged-in. | | |
| Notes and Issues: | Potential option to block user? | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 7 | | |
| Use Case Name: | View Friends List | | |
| Created By: | Bennett Dierckman | Last Updated By: | Emma Smith |
| Date Created: | 10/19/17 | Date Last Updated: | 10/22/17 |
| Actors: | User | | |
| Description: | User views their current friends list on their profile. | | |
| Trigger: | 1. User desires to view a friend’s profile.  2. User desires to see who they are friends with. | | |
| Preconditions: | 1. User has friends in friends list | | |
| Postconditions: | 1. User can see who they are friends with.  2. User visits selected friends profile | | |
| Normal Flow: | **7.0 User Views Friends List**  1. User navigates to their profile page. 2. User selects the “Friends” option from hamburger menu. 3. User scrolls through friends list. 4. User taps back Icon. 5. User directed back to their profile. | | |
| Alternative Flows: | **7.1 User Views Friends Profile** (After Step 3)  1. User selects friend from friend list. 2. User directed to selected friend’s profile. | | |
| Exceptions: | **7.0.E.1 User has no Friends…** (After Step 1)  1. User’s Friend List page displays text that displays “You have no friends.”  2. Return to Step 4. | | |
| Includes: | View Profile | | |
| Priority: | Low | | |
| Frequency of Use: | Semi-frequent depending on the user. | | |
| Business Rules: | NA | | |
| Special Requirements: | NA | | |
| Assumptions: | 1. The application has been downloaded from Google Play, and is open on user’s device.  2. The user has created an account and is logged-in. | | |
| Notes and Issues: | Implement search capabilities. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 8 | | |
| Use Case Name: | View Recent Parties | | |
| Created By: | Abigail Hackett | Last Updated By: | Emma Smith |
| Date Created: | 10/20/2017 | Date Last updated: | 10/23/2017 |
| Actors: | User | | |
| Description: | The user will have the ability to view all the parties that they have attended. Categorized by each specific party, the View Recent Parties feature will provide the user with a master list of all songs played from the beginning of the party to the end. | | |
| Trigger | A user wants to see what parties that they have attended in that past. | | |
| Preconditions: | 1. User is on a profile page. | | |
| Postconditions: | 1. User has viewed previous parties that they have attended. | | |
| Normal Flow: | **8.0 View a Party** 1. User scrolls down on their profile page. 2. User taps the desired party to find out more information. 3. User presented screen with a table of the party’s playlist in chronological order, the date, and parties description. | | |
| Alternative Flows: | * 1. **Exit current playlist and switch to another** (After Step 2 or 3)   1. User taps party name a second time to close the current party’s information. 2. Return to step 2.  **8.2 User does not see the party they are searching for in the initial party’s shown** (After Step 1) 1. User taps the “View More Parties” button. 2. System displays more parties that the user has attended. 3. Return to step 2. | | |
| Exceptions: | NA | | |
| Includes: | View Profile | | |
| Priority: | Medium | | |
| Frequency of Use: | Frequency depends on user, but a typical user may utilize this feature once per app usage. | | |
| Business Rules: | NA | | |
| Special Requirements: | 1. Query to populate a party’s detailed information once it is selected. 2. A User can view the party’s that their friends attended. | | |
| Assumptions: | 1. The User has previously attended parties that utilized MusicMedley. 2. The User is logged into MusicMedley. | | |
| Notes and Issues: | This feature is accessible at any time after the user opens MusicMedley. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 9 | | |
| Use Case Name: | View Advertisement Details | | |
| Created By: | Abigail Hackett | Last Updated By: | Emma Smith |
| Date Created: | 10/19/17 | Date Last Updated: | 10/28/17 |
| Actors: | Users | | |
| Description: | User sees advertisements in the form of a slider throughout their usage of the application. If tapped, the user will be able to see the advertisements details. | | |
| Trigger: | User sees Live Nation or hosting organization’s advertisement slider while using the app. | | |
| Preconditions: | 1. User is either on their profile, another’s profile, or in a party. | | |
| Postconditions: | 1. User is informed of nearby upcoming Live Nation events, or hosting organization’s advertisements. | | |
| Normal Flow: | **9.0 User Views Advertisement** 1. User sees advertisement in slider. 2. User taps on advertisement. 3. Shadow box containing advertisement details and link to external site appear. 4. User reads advertisement details and taps the “X” button to return to app. | | |
| Alternative Flows: | **9.1 User Follows Advertisement’s Link** (After Step 3) 1. User sees advertisement in slider. 2. User taps on advertisement. 3. Shadow box containing advertisement details and link to external site appear. 4. User decides to tap external link. 5. User directed out of application.  **9.2 User never clicks on advertisements.** (After Step 1)  1. User ignores advertisements and keeps scrolling within the application. | | |
| Exceptions: | NA | | |
| Includes: | Live Nation Advertisement Incorporation | | |
| Priority: | Low | | |
| Frequency of Use: | Semi-frequent depending on the user | | |
| Business Rules: | NA | | |
| Special Requirements: | NA | | |
| Assumptions: | 1. The application has been downloaded from Google Play, and is open on user’s device. 2. The user has created an account and is logged-in. | | |
| Notes and Issues: | Request permission to incorporate Live Nation Advertisements | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 10 | | |
| Use Case Name: | Party Discovery Mode | | |
| Created By: | Abigail Hackett | Last Updated By: | Bennett Dierckman |
| Date Created: | 10/20/2017 | Date Last Updated: | 10/23/17 |
| Actors: | Users | | |
| Description: | Users turn on Party Discovery Mode to locate parties happening nearby. | | |
| Trigger | User wants to join a party. | | |
| Preconditions: | Users device has GPS location services enabled. | | |
| Postconditions: | User can see the parties that are occurring around them. | | |
| Normal Flow: | * 1. **Locate Nearby Parties**   1. User initiates Party Discovery Mode. 2. Application Determines User’s Location. 3. Application displays nearby Parties in shadowbox. 4. User selects desired party. | | |
| Alternative Flows: | **10.1 User Does Not Locate Desired Party** (After Step 3)1. User does not find their desired party. 2. User turns off Party Discovery mode, shadowbox closes. | | |
| Exceptions: | **10.1.E.0 No Existing Parties** 1. System displays no existing parties.  2. User prompted to “Create Event”. | | |
| Includes: | Create Event, GPS tracking | | |
| Priority: | High | | |
| Frequency of Use: | Whenever a User desires to locate nearby parties. | | |
| Business Rules: | NA | | |
| Special Requirements: | 1. Parties will be displayed that are within a host specified radius of the hosting device. | | |
| Assumptions: | 1. The user is logged in. 2. User has data/WiFi to track location. | | |
| Notes and Issues: | NA | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 11 | | |
| Use Case Name: | Create Party | | |
| Created By: | Abigail Hackett | Last Updated By: | Emma Smith |
| Date Created: | 10/20/2017 | Date Last Updated: | 10/25/2017 |
| Actors: | Host | | |
| Description: | When a user starts his/her own party, the user becomes a “host”. The first step in becoming a host is creating an event. | | |
| Trigger | A user wishes to begin a Party. | | |
| Preconditions: | 1. Host is logged into MusicMedley. | | |
| Postconditions: | 1. Party information will be saved in the database. 2. Users will be able to join Party. | | |
| Normal Flow: | **11.0 Host Creates Party** 1. Host taps “Create Party”. 2. Host is brought to “Create Party” screen. 3. Host enters party title in “Party Title” editText. 4. Host selects a box indicating “18+” party or “Not 18+”. 5. Host selects “Party Radius” from available options. 6. User reaches “Set Party Time Frame” and selects “No”. 7. Host sets Party password. 8. Host adds songs to create initial playlist. 9. Host taps “Finalize Party Plans”. 10. The database saves the information. 11. The database saves the host’s device information and location. | | |
| Alternative Flows: | **11.1 Host decides to cancel creating event during Setup** (After Step 1)1. Host taps “Cancel”. 2. User is returned to their profile page. **11.2** **Host Sets a Party** **Timeframe** (After Step 5) 1. Host selects “Yes”. 2. Host selects the “Start Time” on the TimePicker. 3. Host selects the “End Time” on the TimePicker. 4. Return to Step 7. | | |
| Exceptions: | **11.0.E.1 Host’s GPS Location is Disabled**  1. System notifies user that they must turn on Location Services for MusicMedley. | | |
| Includes: | GPS Location, Suggest Song | | |
| Priority: | High | | |
| Frequency of Use: | Each time a party is created. | | |
| Business Rules: | Privacy clause within Terms & Conditions. | | |
| Special Requirements: | NA | | |
| Assumptions: | 1. MusicMedley is open on the Host’s device. 2. GPS Location Services are enabled on the Hosts device. | | |
| Notes and Issues: | This feature is *only* accessed when creating a party. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 12 | | |
| Use Case Name: | Display YouTube video | | |
| Created By: | Noah Kiser | Last Updated By: | Emma Smith |
| Date Created: | 10/20/17 | Date Last Updated: | 10/28/17 |
| Actors: | Users and YouTube | | |
| Description: | Display the YouTube video of the given song that is next on the master playlist. | | |
| Trigger: | Video is suggested or chosen, voted on, and is now at the top of the playlist. | | |
| Preconditions: | 1. User has added video *or* attendees suggest video. 2. Video has ended up at the top of the playlist based on voting. | | |
| Postconditions: | 1. Video is played for those at the party to view. | | |
| Normal Flow: | **12.0 Display YouTube Video**  1. When a video comes up on the playlist the video is translated to the display.  2. System displays video at the top of activity page.  3. When video is over the next video on the playlist is displayed. | | |
| Alternative Flows: | **12.1 Users do not like the Video displayed** (After Step 2)  1. When a video comes up on the playlist the video is translated to the display.  2. Users “dislikes” video playing and tap the “thumbs down" button.  3. System registers the dislike votes on the video.  4. System evaluates likes to dislike votes.  5. Dislikes override Likes.  6. Next video on the playlist is displayed.  **12.2 User Likes the Video displayed** (After Step 2)  1. When a video comes up on the playlist the video is translated to the display.  2. Users “likes” video playing and tap the “thumbs up" button.  3. System registers the like vote on the video.  4. System evaluates likes to dislike votes.  5. Likes override dislikes.  6. Video remains displayed. | | |
| Exceptions: | **12.0.E.1 User wants to lock device**  1. System is displaying video on user’s device.  2. User Locks phone.  3. Video remains displayed on host’s device.  4. Video proceeds. | | |
| Includes: | Like Song, Dislike song | | |
| Priority: | High | | |
| Frequency of Use: | Every time a song on the playlist is being played (frequent throughout party). | | |
| Business Rules: | None | | |
| Special Requirements: | Hosts device must be unlocked for YouTube song to remain playing due to YouTube terms and conditions. | | |
| Assumptions: | 1. The application has been downloaded from Google Play, and is open on user’s device.  2. The Host has YouTube Red. | | |
| Notes and Issues: | 1. Need to decide what it takes for a video to be skipped over after starting to play.  2. “Video” is also referred to as “song”. The content of the video displayed will be primarily music being that MusicMedley is designed for music inclined settings. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 13 | | |
| Use Case Name: | End Party | | |
| Created By: | Noah Kiser | Last Updated By: | Emma Smith |
| Date Created: | 10/20/17 | Date Last Updated: | 10/28/17 |
| Actors: | Host | | |
| Description: | This feature allows the party host to end party. | | |
| Trigger: | Host desires to end the party. In doing so, no other user can enter the playlist, add songs, or like/dislike songs. | | |
| Preconditions: | 1. Host creates party. | | |
| Postconditions: | 1. Party is ended and no longer viewable on party list to those in gps range. 2. The playlist is saved in “View recent parties”. | | |
| Normal Flow: | **13.0 End Party**  1. Host taps hamburger menu in top left corner.  2. Host selects “End Party”.  3. System asks “Are you sure you would like to end this party” followed by “Yes” or “No” that can be tapped.  4. Host selects “Yes”.  5. The party ends and the user is taken back their profile page.  6. All attendees of the party’s device will be notified and kicked out of the party. | | |
| Alternative Flows: | **13.1 Host decides not to end party.** (After Step 3)  1. Host selects “No”.  2. Host is taken back to party playlist.  **13.2 Party Playlist runs out of songs.**  1. System automatically ends the party.  2. Attendee’s kicked out of the party.  3. Attendee’s directed to their own profile pages.  4. Host is taken back to profile page.  **13.3 End Time on Party is Reached.**  1. System automatically ends the party.  2. Attendee’s kicked out of the party.  3. Attendee’s directed to their own profile pages.  4. Host is taken back to profile page. | | |
| Exceptions: | NA | | |
| Includes: | Profile Page, Party Playlist | | |
| Priority: | High | | |
| Frequency of Use: | Every time that a party ends. | | |
| Business Rules: | NA | | |
| Special Requirements: | NA | | |
| Assumptions: | The application has been downloaded from Google Play, and is open on user’s device. | | |
| Notes and Issues: | None | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 14 | | |
| Use Case Name: | Join Party | | |
| Created By: | Noah Kiser | Last Updated By: | Emma Smith |
| Date Created: | 10/20/17 | Date Last Updated: | 10/25/17 |
| Actors: | User, Attendee | | |
| Description: | Allow users to join desired party via Party Discovery Mode. | | |
| Trigger: | User wants to join a party. | | |
| Preconditions: | 1. User has turned on Party Discovery Mode.  2. User has received a password that allows them to access the party. | | |
| Postconditions: | 1. User is inside of the party and capable of suggesting songs. | | |
| Normal Flow: | **14.0 Join Party**  1. A list of the available parties based on location appears.  2. User selects desired party from available parties.  3. User is prompted to enter Party Password.  4. User taps “Join Party”.  5. User is registered as party attendee. | | |
| Alternative Flows: | **14.1 Party is not Listed** (After Step 1)  1. User continues to walk around searching for desired party.  2. Desired Party appears on Party Discovery Mode.  3. Return to Step 2.  **14.2 User Entered Wrong Password** (After Step 3)  1. User is notified that the password entered is incorrect.  2. User is prompted to enter password once more.  3. Return to Step 4. | | |
| Exceptions: | **14.0.E.1 User is not of Age**  1. System notifies user that they do not meet the age requirements for the party.  2. User is redirected to Profile. | | |
| Includes: | Party Discovery Mode | | |
| Priority: | High | | |
| Frequency of Use: | Every time a user wants to enter a party. (frequent) | | |
| Business Rules: | NA | | |
| Special Requirements: | NA | | |
| Assumptions: | It is assumed that the application has been downloaded from Google Play, and is open on user’s device. | | |
| Notes and Issues: |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 15 | | |
| Use Case Name: | Suggest Song | | |
| Created By: | Emma Smith | Last Updated By: | Emma Smith |
| Date Created: | 10/18/17 | Date Last Updated: | 10/28/17 |

|  |  |
| --- | --- |
| Actors: | Attendee, Host (upon website integration) |
| Description: | An attendee of a party wants to suggest a song to add to the party playlist. |
| Trigger: | Attendee looks at the party playlist and does not see a song that they would like to hear. |
| Preconditions: | 1. Attendee has selected a party.  2. Attendee has entered the correct password.  3. Attendee has tapped the “Join” button to enter the party.  4. Attendee is viewing party playlist. |
| Postconditions: | 1. The suggested song is added to the party playlist in the system.  2. The party playlist is updated with the suggested song. |
| Normal Flow: | 1. **Suggest Song**   1. Attendee scrolls through the playlist and does not see a song they like. 2. Attendee taps the “Suggest Song” button. 3. Attendee enters song information that they want to look up. 4. Attendee presses “Search” button. 5. System pulls results based on whether or not explicit content is allowed at party. 6. Attendee views top results. 7. Attendee selects desired video. 8. A dialogue box pops up and asks “Are you sure you want to suggest --------?” (Where ---- denotes the video title). 9. Attendee presses “Confirm” button. |
| Alternative Flows: | 1. **Attendee is satisfied with the party playlist.** (After step 1)   1. Attendee does not suggest song.  **15.1 Attendee searched for the incorrect song title or misspelled what they were looking for.** (After Step 4) 1. Attendee selects the search box. 2. Attendee re-enters information/fixes errors. 3. Return to step 4.  **15.2 Attendee does not find the song they are looking for.** (After step 6) 1. Attendee taps the “View More Results” button to search the next 10 video results. 2. Return to step 7.  **15.3 Attendee presses “Cancel”.** (After Step 7) 1. Attendee is taken back to the Results page. 2. Attendee continues to search for their desired song. 3. Return to step 6. |
| Exceptions: | **15.3.E.2 Attendee presses “Search” with an empty field.** 1. System informs Attendee that they must enter a valid search.2. Return to step 3. |
| Includes: | View Party Playlist, Search for Desired Video |
| Priority: | High |
| Frequency of Use: | Will range from each party and each user. Each user may use this feature an unlimited number of times during a part (an average of 3 suggestions per user). |
| Business Rules: | NA |
| Special Requirements: | System recognizes when there is a repeat song within the playlist or another user has suggested the same song. |
| Assumptions: | Attendee will view the playlist before suggesting a song. |
| Notes and Issues: | Potential exception  \*\*\*15.3.E.1 Attendee enters information with invalid characters**.** 1. System informs Attendee of invalid characters and asks them to re-enter search. 2. Attendee re-enters search information.\*\*\* |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 16 | | |
| Use Case Name: | Like Song | | |
| Created By: | Emma Smith | Last Updated By: | Bennett Dierckman |
| Date Created: | 10/21/17 | Date Last Updated: | 10/23/17 |

|  |  |
| --- | --- |
| Actors: | Attendee, Host (upon website integration) |
| Description: | An attendee or host either views a song on the playlist or is currently hearing it being played and wants to “like” the song. |
| Trigger: | An attendee or host wants to “like” a song within the playlist. |
| Preconditions: | 1. An attendee/host is within a party. |
| Postconditions: | 1. The system will process the like and store that the attendee/host liked the song.  2. The system will process the “Like Song” and show it in the party playlist. |
| Normal Flow: | **16.0 Like Song** 1. Attendee/Host scrolls through the playlist and sees a song that peaks their interest. 2. Attendee/Host pressed the “Like Song” button denoted by a “thumbs-up” icon.  3. System registers users action. 4. Visual confirmation feedback sent to user. |
| Alternative Flows: | **16.1** **Attendee/Host accidently pressed “Dislike Song”.** (After Step 4) 1. Attendee/Host taps the “Like Song” button to correct the mistake. **16.2 Attendee/Host accidently pressed “Love Song”.** (After Step 4) 1. Attendee/Host re-taps the “Love Song” button once more. 2. “Love Song” button deactivates. 3. Attendee/Host taps the “Like Song” button to correct the mistake. 4. Return to step 3. **16.3 Attendee/Host did not mean to “Like Song”.** (After Step 4) 1. Attendee/Host pressed on the “Like Song” button a second time. 2. “Like Song” button deactivates. 3. Return to step 3. |
| Exceptions: | NA |
| Includes: | View Party Playlist, Dislike Song, Love Song |
| Priority: | High |
| Frequency of Use: | Attendee/Host may choose this option as many times as there are number of videos in the playlist. |
| Business Rules: | NA |
| Special Requirements: | The system can handle changing the attendee/host’s “Like Song” from a “Dislike Song” or “Love Song”. |
| Assumptions: | Attendee/Host does not want to “Like Song” and “Dislike Song” at the same time. |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 17 | | |
| Use Case Name: | Dislike Song | | |
| Created By: | Emma Smith | Last Updated By: | Bennett Dierckman |
| Date Created: | 10/21/17 | Date Last Updated: | 10/23/17 |

|  |  |
| --- | --- |
| Actors: | Attendee, Host (upon website integration). |
| Description: | An attendee or host either views a song on the playlist or is currently hearing it being played and wants to “dislike” the song. |
| Trigger: | An attendee or host wants to “dislike” a song within the playlist. |
| Preconditions: | 1. An attendee/host is within a party. |
| Postconditions: | 1. The system will process the like and store that the attendee/host disliked the song.  2. The system will process the “Dislike Song” and show it in the party playlist. |
| Normal Flow: | **17.0 Dislike Song** 1. Attendee/Host scrolls through the playlist and sees a song that they do not enjoy. 2. Attendee/Host pressed the “Dislike Song” button denoted by a “thumbs-down” icon. 3. System registers users action. 4. Visual confirmation feedback sent to user. |
| Alternative Flows: | **17.1** **Attendee/Host pressed the wrong button of either “Like Song” or “Love Song”.** (After Step 4) 1. Attendee/Host taps the “Dislike Song” button to correct the mistake. 2. Return to step 3.  **17.2 Attendee/Host did not mean to “Dislike Song”.** (After Step 4) 1. Attendee/Host pressed on the “Dislike Song” button a second time. 2. “Dislike Song” button deactivates. 3. Return to step 3. |
| Exceptions: | NA |
| Includes: | View Party Playlist, Like Song, Love Song |
| Priority: | High |
| Frequency of Use: | Attendee/Host may choose this option as many times as there are number of videos in the playlist. |
| Business Rules: | NA |
| Special Requirements: | The system can handle changing the attendee/host’s “Dislike Song” from a “Like Song” or “Love Song”. |
| Assumptions: | Attendee/Host does not want to “Like Song” and “Dislike Song” at the same time. |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 18 | | |
| Use Case Name: | Love Song | | |
| Created By: | Emma Smith | Last Updated By: | Bennett Dierckman |
| Date Created: | 10/21/17 | Date Last Updated: | 10/23/17 |

|  |  |
| --- | --- |
| Actors: | Attendee, Host (upon website integration) |
| Description: | An attendee or host likes a song within the playlist so much that they want to book or “Love” the song. |
| Trigger: | An attendee or host want to “Love Song” within the playlist to bookmark the song on their profile. |
| Preconditions: | 1. An attendee/Host is within a party. |
| Postconditions: | 1. The system will process the “Love Song” and store that the attendee/host loved the song and bookmarks it on the user’s profile. |
| Normal Flow: | **18.0 Love Song** 1. Attendee/Host scrolls through the playlist and sees a song that they want to bookmark. 2. Attendee/Host taps the “Love Song” button. 3. System registers users action. 4. Visual confirmation feedback sent to user. |
| Alternative Flows: | **18.1 Attendee/Host accidently pressed the “Dislike Song” or “Like Song” button.** (After Step 4) 1. Attendee/Host presses the “Dislike Song” or “Like Song” button to deactivate the like or dislike. 2. Return to step 3. **18.2 Attendee/Host did not mean tap the “Love Song” button.** (After Step 4) 1. Attendee/Host presses the “Love Song” button a second time. 2. “Love Song” button deactivates. 3. Return to step 3. |
| Exceptions: | NA |
| Includes: | View Party Playlist, Like Song, Dislike Song |
| Priority: | Medium |
| Frequency of Use: | Each user can bookmark the number of songs within a party playlist, but mostly likely will only use this feature once or twice per party. |
| Business Rules: | NA |
| Special Requirements: | 1. Store “Loved Songs” within the database 2. Display “Loved Songs” on user’s profile. |
| Assumptions: | User will not “Love Song” and “Dislike Song” simultaneously. |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 19 | | |
| Use Case Name: | View Party Playlist | | |
| Created By: | Emma Smith | Last Updated By: | Bennett Dierckman |
| Date Created: | 10/21/17 | Date Last Updated: | 10/23/17 |

|  |  |
| --- | --- |
| Actors: | Attendee, Host (upon website integration) |
| Description: | An Attendee/Host views the list of songs that a party has on queue. |
| Trigger: | An Attendee/Host wants to know what songs are coming up or within the playlist. |
| Preconditions: | 1. The Attendee/Host is within the party. |
| Postconditions: | 1. The Attendee/Host is aware of the party’s playlist. |
| Normal Flow: | **19.0 View Party Playlist** 1. Attendee/Host opens the application.2. Attendee/Host expands playlist extension.3. Attendee/Host scrolls through the contents of the playlist. |
| Alternative Flows: | **19.1 Attendee/Host no longer wants to view the playlist**. (After Step 3) 1. Attendee/Host navigates to profile page. |
| Exceptions: | **19.0.E.1 Party Host does not have the capability of viewing the playlist due to their status of Host.** (After Step 1) 1. Host logs into website to host video. 2. Host reenter application on their mobile device. 3. Return to step 2. |
| Includes: | Stream Video |
| Priority: | High |
| Frequency of Use: | Whenever a user within a party opens up the application. |
| Business Rules: | NA |
| Special Requirements: | NA |
| Assumptions: | It is assumed that the user has closed out of the application before wanting to perform this use case. |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 20 | | |
| Use Case Name: | Leave Party | | |
| Created By: | Emma Smith | Last Updated By: | Bennett Dierckman |
| Date Created: | 10/21/17 | Date Last Updated: | 10/23/17 |

|  |  |
| --- | --- |
| Actors: | Attendee |
| Description: | An attendee is physically or electronically leaving a party, which prevents influence on the party’s playlist after attendee’s departure. |
| Trigger: | An attendee is leaving a party. |
| Preconditions: | 1. An attendee is within a party. |
| Postconditions: | 1. An attendee is no longer within the party. 2. User can no longer influence party playlist. |
| Normal Flow: | **20.0 Leave Party** 1. Attendee decides that it is time to leave the party.2. Attendee taps the “Leave Party” button.3. A dialogue box appears asking if the Attendee is sure that they want to leave the party.4. Attendee taps the “Confirm” button.5. Attendee has exited the party. |
| Alternative Flows: | **20.1 Attendee accidently pressed the “Leave Party” button.** (After Step 3) 1. Attendee taps the “Cancel” button. 2. Attendee remains within the party. |
| Exceptions: | **20.0.E.1 Attendee never presses “Leave Party” button but is still within GPS range.** 1. System kicks attendee out of party once the party ends. **20.0.E.2 Attendee never presses “Leave Party” button but is outside of GPS range.** 1. System kicks attendee out of party once attendee leaves GPS range. |
| Includes: | Set Party Timeframe |
| Priority: | Medium |
| Frequency of Use: | Expected once per user for every party attended. Exceptions occur if an attendee returns after running an errand, etc. |
| Business Rules: | NA |
| Special Requirements: | GPS Location Incorporation |
| Assumptions: | NA |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 21 | | |
| Use Case Name: | Search for Desired Video | | |
| Created By: | Emma Smith | Last Updated By: | Bennett Dierckman |
| Date Created: | 10/21/17 | Date Last Updated: | 10/23/17 |

|  |  |
| --- | --- |
| Actors: | Media Streaming Service, Attendee, Host (upon website integration) |
| Description: | The Media Streaming service will process the information that Attendees input when searching for songs in order to suggest songs for a party playlist. |
| Trigger: | Either when an attendee searches for a song that they want to suggest, or a host is building a party playlist. |
| Preconditions: | 1. The Host has created a party event.  2. The attendee is suggesting a song. |
| Postconditions: | 1. The desired video is found. |
| Normal Flow: | **21.0 Search for Desired Video.** 1. Attendee/Host select the “Search for Song” search bar.2. Attendee/Host input desired song information.3. Attendee/Host tap the “Search” button.4. System runs query through Media Streaming Service to search for desired song.5. Media Streaming Service receives input and pulls top 15 results.6. System receives results.7. System displays first 5 videos, with an option to “View More Results” expand the results 10 more videos (totaling 15).8. Attendee/Host selects intended video.9. System asks if Attendee/Host are sure they want to add/suggest the selected song.10. Attendee/Host taps the “Confirm” button.11. Song is added to the party playlist. |
| Alternative Flows: | **21.1 Attendee/Host Search for Wrong Video** (After Step 3) 1. Attendee/Host reselect the “Search for Song” search bar. 2. Attendee/Host input the correct search information. 3. Return to Step 3. **21.2 Search Results Page Does Not Display Desired Video** (After Step 7) 1. Attendee/Host taps the “Next Page” button. 2. Return to Step 8. **21.3 Attendee/Host Select the Wrong Video** (After Step 8) 1. System asks if the Attendee/Host are sure they want to add/suggest the selected song. 2. Attendee/Host taps the “Cancel” button. 3. Return to Step 7. |
| Exceptions: | **21.0.E.1 Invalid Information is Provided in the “Search for Song” search bar** 1. System informs Attendee/Host of invalid characters and asks them to re-enter search.2. Return to Step 2. |
| Includes: | Create Initial Playlist, Suggest Songs, Send Request via API |
| Priority: | High |
| Frequency of Use: | Every song that is added to an initial playlist as well as every time a song is suggested. |
| Business Rules: | NA |
| Special Requirements: | Pre-made queries that only need to handle string input from the user. |
| Assumptions: | NA |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 22 | | |
| Use Case Name: | Send Media Request Via API | | |
| Created By: | Emma Smith | Last Updated By: | Bennett Dierckman |
| Date Created: | 10/21/17 | Date Last Updated: | 10/23/17 |

|  |  |
| --- | --- |
| Actors: | Media Streaming Service |
| Description: | The System utilizes YouTube’s API in order to process video requests. |
| Trigger: | 1. An attendee attempts to suggest a song. 2. A Host begins creating an initial playlist. 3. A host is streaming a video. |
| Preconditions: | 1. A host has created a party event.  2. Songs have been added to the playlist |
| Postconditions: | 1. Display video results.  2. Media is played according to YouTube’s Terms and Conditions |
| Normal Flow: | **22.0 Send Media Request Via API** 1. Attendee/Host searches for a song. 2. System formulates query request. 3. System processes query through YouTube’s API. 4. API runs query. 5. API pulls query results temporarily stores information. 6. API sends information to system. 7. System receives information. 8. System displays first 5 results with an option to “View More Results” (displaying an additional ten videos). |
| Alternative Flows: | NA |
| Exceptions: | **22.0.E.2 No Results Found from Search Query** 1. System informs Attendee/Host that no results could be found with the current search information.2. System asks Attendee/Host to re-enter search information.3. Return to Step 1. |
| Includes: | Suggest Song, Search for Desired Video, Stream Video, Create Initial Playlist |
| Priority: | High |
| Frequency of Use: | Every time a video is searched for when creating the initial party playlist, suggesting a song, and streaming a video from the host device. |
| Business Rules: | NA |
| Special Requirements: | Specific Media Streaming Service API is obtained. |
| Assumptions: | The Media Streaming Service API is working and compatible with the system. |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 23 | | |
| Use Case Name: | Stream Video | | |
| Created By: | Emma Smith | Last Updated By: | Bennett Dierckman |
| Date Created: | 10/21/17 | Date Last Updated: | 10/23/17 |

|  |  |
| --- | --- |
| Actors: | Media Streaming Service, Host |
| Description: | YouTube API enables media to be played on hosts device. |
| Trigger: | 1. The party playlist begins. 2. A new video begins after the prior video ends. |
| Preconditions: | 1.The party has begun.  2. Videos have been added to the party’s playlist.  3. The Hosts device’s application is active. |
| Postconditions: | The streaming video has ended. |
| Normal Flow: | **23.0 Stream Video** 1. System pulls information of the next upcoming song in party playlist queue.2. System formulates query based on information.3. System runs query through YouTube’s API.4. Media Streaming Service allows access.5. YouTube’s API sends information to the System.6. System obtains streaming information.7. System streams video. |
| Alternative Flows: | **23.1 Media Streaming Service Denies Access** (After Step 3) 1. Media Streaming Service denies access. 2. Media Streaming Service sends denial through YouTube’s API. 3. YouTube’s API notifies the System of denial of access. 4. Return to Step 1. |
| Exceptions: | NA |
| Includes: | Display YouTube Video, Send Media Request via API |
| Priority: | High |
| Frequency of Use: | Each time a video is being played within the party by the Host. |
| Business Rules: | None. |
| Special Requirements: | Only the Host can display or stream the video. |
| Assumptions: | Majority of videos will not be denied access. |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 24 | | |
| Use Case Name: | Retrieve Advertisement | | |
| Created By: | Bennett Dierckman | Last Updated By: | Emma Smith |
| Date Created: | 10/19/17 | Date Last Updated: | 10/28/17 |
| Actors: | Advertisements | | |
| Description: | Advertisements will be presented to the user on their profile or in a party. They will be presented with Live Nation Entertainment advertisements based on their GPS location, unless the user is in a party hosted by an organization with paid advertisement space. | | |
| Trigger: | 1. The application opens on the device. | | |
| Preconditions: | 1. User is either on their profile, another’s profile, or in a party. | | |
| Postconditions: | 1. Advertisements are ready to be sent to user’s device. | | |
| Normal Flow: | **24.0 Retrieve Live Nation Entertainment Advertisement**  1. Android determines location of user. 2. Advertisements determined based on zip code from location.  3. Advertisements found after Android sends zip code to Live Nation Entertainment event locator.  4. Advertisements are ordered and prepared for user’s device.  5. Advertisements are sent to user’s device. | | |
| Alternative Flows: | **24.1 Retrieve Advertisement from Hosting Organization** 1. Android knows user is in an organization’s party.  2. Advertisements are pulled from organization’s uploaded advertisements.  3. Jump to Step 4. | | |
| Exceptions: | **24.0.E.1 No Connection to Internet**  1. System shows User a screen asking them to connect to internet. | | |
| Includes: | GPS Locating | | |
| Priority: | Low | | |
| Frequency of Use: | Anytime the application is open, and the device is connected to the internet. | | |
| Business Rules: | NA | | |
| Special Requirements: | Permission from Live Nation Entertainment | | |
| Assumptions: | It is assumed that the application has been downloaded from the Google Play store. | | |
| Notes and Issues: | Request permission to incorporate Live Nation Advertisements. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 25 | | |
| Use Case Name: | Display Advertisements | | |
| Created By: | Bennett Dierckman | Last Updated By: | Emma Smith |
| Date Created: | 10/19/17 | Date Last Updated: | 10/28/17 |
| Actors: | Advertisements | | |
| Description: | Determined advertisements are displayed on the user’s device. | | |
| Trigger: | 1. The application opens on the device. | | |
| Preconditions: | 1. User is either on their profile, another’s profile, or in a party. | | |
| Postconditions: | 1. Advertisements are displayed on the user’s device. | | |
| Normal Flow: | **25.0 Display Live Nation Advertisements**  1. Advertisements determined in Use Case 24 are sent to the user’s device. 2. Advertisements are displayed in slider.  3. Slider rotates through advertisements every 30 seconds. | | |
| Alternative Flows: | **25.1 Display Hosting Organizations Advertisements**  1. Advertisements from hosting organization are sent to the user’s device.  2. Return to Step 2. | | |
| Exceptions: | **25.0.E.1 User’s Device is offline**  1. Display no advertisements. | | |
| Includes: | GPS Locating, Live Nation Entertainment | | |
| Priority: | Low | | |
| Frequency of Use: | Anytime the application is open, and the device is connected to the internet. | | |
| Business Rules: | NA | | |
| Special Requirements: | NA | | |
| Assumptions: | It is assumed that the application has been downloaded from the Google Play store. | | |
| Notes and Issues: | NA | | |