UpNext

Product Backlog

Shivan Desai, Adam Johnston, Akhil Agrawal, Brian Duffy

Problem Statement:

While there are many online radios, or music platforms, we felt that few really involve the user with other people when choosing songs. We present UpNext, a music website where users can post, vote, and comment on songs with other people.

Background Information:

The goal is to make creating a playlist more of a group experience. In order to keep the playlist fresh, songs will expire after a certain time limit. We want user interaction to be at the heart of our application. Many people are afraid to share their taste of music from fear of judgement. What our app does is put that anonymity barrier to the enjoyment of music with others.

Environment:

Our app will be developed using AngularJS for modularity of our app as well as single page routing for the web interface. In addition to this, we will be utilizing the Spotify Web API as the backbone to our app’s music functionality. As for the hosting of our web app we plan on using Google Firebase as it is reliable, cheap, and relatively straightforward to use.

Functional Requirements:

|  |  |  |
| --- | --- | --- |
| Backlog Id | Functional Requirement | Hours |
| 1 | As a user, I would like to post songs. | 10 |
| 2 | As a user, I would like to delete songs that I have added. | 8 |
| 3 | As a user, I would like to vote for my favorite songs. | 8 |
| 4 | As a user, I would like to flag songs as inappropriate. | 6 |
| 5 | As a user, I would like to comment on songs (time permitting). | 10 |
| 6 | As a user, I would like to mark songs to be skipped (time permitting). | 10 |
| 7 | As a user, I would like to create a profile (time permitting). | 10 |
| 8 | As an admin, I would like to remove songs. | 6 |
| 9 | As an admin, I would like to delete users. | 8 |
| 10 | As a developer, I would like to have songs automatically removed after reaching an expiration date. | 6 |

Non-Functional Requirements:

|  |  |  |
| --- | --- | --- |
| Backlog Id | Non-Functional Requirement | Hours |
| 1 | As a user I would like to have the app formatted for mobile devices as well as desktop (time permitting). | N/A |
| 2 | As a developer I would like the app to be secure. | 12 |
| 3 | As a user I would like to retrieve songs efficiently. | N/A |

Use Cases:

Use Case 1 - Post Songs

|  |  |
| --- | --- |
| Action | System Response |
| Click on create post button | Access a page allowing user to select a song |
| Select a song from Spotify |  |
| Confirm Selection | Create a post in the database with the selected song and reload original songs page |

Use Case 2 - Delete Songs

|  |  |
| --- | --- |
| Action | System Response |
| Click a button next to one of the user’s created posts | Load a dialogue box/page confirming deletion of the post |
| Click confirm for deletion | Delete the song from the database and return to the homepage |

Use Case 3 - Vote on Songs

|  |  |
| --- | --- |
| Action | System Response |
| Click the upvote or downvote button next to a post | Increment/Decrement the number next to the post, changing it’s order with the nearby songs to reflect the change |

Use Case 4 - Flag Songs as Inappropriate

|  |  |
| --- | --- |
| Action | System Response |
| Click a report button next to one of the posts | If the post has over a preset limit of reports, remove that post from the database |
|  | Update the list accordingly |

Use Case 5 - Comment on Songs

|  |  |
| --- | --- |
| Action | System Response |
| Click on a post and pull up a comments section/page | Load the comments page |
| Type text into a comments textbox |  |
| Click submit | Add the typed text as a comment |
|  | Update the comments to reflect the change |

Use Case 6 - Mark Songs to Skip

|  |  |
| --- | --- |
| Action | System Response |
| Click a skip button next to one of the posts/songs | Remove the song from the list of songs the user’s Spotify Player will play |

Use Case 7 - Create a Profile

|  |  |
| --- | --- |
| Action | System Response |
| User clicks on create account button of the login page. | Redirects user to Spotify’s Accounts Service |
| User follows the prompt to create a new account |  |
| User creates account | Redirect to UpNext user login page |

Use Case 8 - Remove Songs (Admin)

|  |  |
| --- | --- |
| Action | System Response |
| Click a button next to a post/song to remove it | Load a confirmation box/page |
| Confirm deletion of selected post | Delete the post from the database |
|  | Update the post list to display accordingly |

Use Case 9 - Ban Users (Admin)

|  |  |
| --- | --- |
| Action | System Response |
| Type a user’s profile name into a textbox and click submit | Add the user to a banned list in the database |
| The user tries to login | Redirects the user to a page notifying them they are banned |

Use Case 10 - Playlist Garbage Collector (Developer)

|  |  |
| --- | --- |
| Action | System Response |
| A song is added to the playlist | The song is stored in the database |
|  | Time passes |
| The song is removed from the playlist | The song reaches an “expiration date” and there are not enough upvotes |