

Assignment 1 - Proposal

Proposal #1:

- Spotify API (can also use soundcloud)
 - An application that can create a playlist of the setlist of the concert (headliner and opener) you are planning to attend, so that you can get into the mood of the artists. We will have the users sign up for an account, with OAuth such as gmail, facebook, etc. that stores their basic information such as name, email, etc. and the list of setlists generated from past concerts they have attended. The setlists will be generated using the Spotify and SoundCloud API.

This is the proposal that we executed.

Change #1: We stuck with the Spotify API, but we changed the SoundCloud API to the iTunes API. The iTunes API had clearer documentation and generally easier to use.

Proposal #2:

- Plan food
 - Yelp API
 - An application that can plan a date for you by giving you a dinner, dessert, and drinks place within proximity of each other using the Yelp API, based on inputted location. We will have the users sign up for an account, with OAuth such as gmail, facebook, etc. that stores their basic information such as name, email, etc. It can choose locations based on ratings, reviews, proximity, and price.

Backup Proposal:

- Aggregate free events around the area
- Cheapest travel itinerary (best place to go based on weather & flight price)
 - Preferences (nature / cities)
 - Dates
 - List of travel restrictions (covid-19 travel restriction APIs)
 - User information to gather:
 - Nationality
 - Location (find closest airport, etc.)
 - Preferences

Assignment 2 - User Stories

User: Bob the API Builder just bought concert tickets to his favorite artist. He is trying to find a way to get ready and excited for the concert.

- **Login to Spotify**

As a not yet logged in / logged out user of the site, I want to be able to log in to my Spotify account to access the functionalities of the app, and to be able to connect with my Spotify account to add the playlist generated by the site to my playlist collection.

- Bob opens the application and sees the landing page of the app with a Spotify login button that guides Bob through login flow.
- If he has successfully logged in to his Spotify account, he sees the “Search Concert Page”.
- If he has failed to login, Bob will see a “Please try again” message and proceed to try again until successful.

Change #1: The user is able to save the playlist to their own playlist collection on our website, not Spotify.

- **Search Concert Page**

As a logged in user, I want to be able to search for my upcoming concert, so that I can have a playlist created.

- Logged In Bob is now at the search concert page. He sees a search section with an “artist” text box, “city” text box, and “date” selection UI to fill out about his upcoming concert. After he fills out all the information and clicks on the “search” button, if he didn’t fill out any of the three information, an error message of “please fill out all information” will appear. If he did fill out all the information, he sees a loading icon while the application starts generating the concert setlist(s).
- If his search is successful, Bob sees the resulting concert setlist(s). Bob could then click on the concert setlist items to proceed to “Created playlist”.
- If search fails, an error message “please try again” will be displayed.

Change #1: We only need the artist search box under the assumption that the user will not be searching for the artist unless there is a concert going on. Therefore, we do not have to verify that there is a concert.

- **Created playlist**

As a logged in user who already found a concert setlist, I want a playlist of the concert setlist to listen to, so that I can get ready for the concert.

- After Bob finds a concert setlist that he is attending, a playlist of the concert setlist will be displayed on this page.
- If the playlist generation is successful, a playlist with an expandable display of the list of songs will be displayed.
- If creating the playlist fails, Bob will receive an error message “cannot find a playlist”.

- **Add Playlist To Spotify**

As a logged in user who already found a playlist of the concert that I will be going to, I want to be able to import and save the resulting playlist to my own Spotify account.

- Logged In Bob doesn't only want to listen to the playlist created for his next concert online, he also wants to have the playlist in his Spotify account so that he can listen to the playlist on phone. On each playlist the site generated, Bob can see an "import" button. When he clicks it, he can see a message with "Do you want to import the playlist to your Spotify account?", after he clicks on "yes", the import process starts. The import process shouldn't fail as Bob now already connects the application to his Spotify account.
- If the import is successful, the playlist will be imported over to spotify, and Bob should be able to see it under his playlist collection in his Spotify account on the web or on the mobile app.
- Since Bob is already logged in, the import should not fail. If it fails, it is some bug in the implementation of the Spotify API. We should make sure it works :)))

Change #1: The user will be able to listen to their playlist only on our application. We made the change because we believe that this feature would be better implemented in our next iteration of the project.

Technology Stack

For this project, our team decided to use the MERN (MongoDB, Express,JS, React.JS, Node.JS) stack as it is very popular and demanding in the development market. Also, Spotify provides a well-rounded document on how to use its API in Node.JS. Originally, we have also considered using Django or Flask for backend. However, since React uses JavaScript and these two frameworks use Python, we think it would be better to use both JavaScript for both frontend and backend to have a better learning curve.

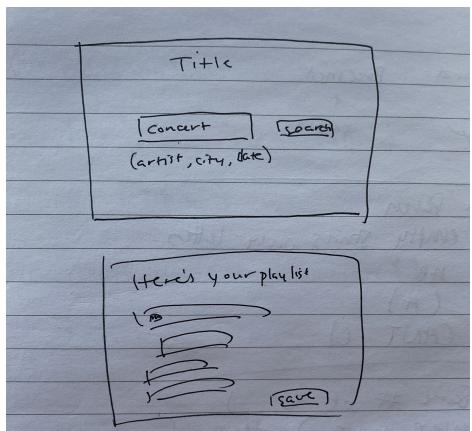
Change #1: We considered changing the database from MongoDB to Firebase, but quickly discovered that Firebase was better for mobile development. In the end, we kept using MongoDB. We used OAuth to authenticate the user login and used it to get user information for the database.

Prototype link: <https://github.com/edward0326/411-Project/tree/api/prototype>

Demonstration video link:

<https://drive.google.com/drive/u/0/folders/1SQ2tXS0OHJDuk5SOYlmXSq1EZZIAPeF8>

Artifacts



Artifact #1: A quick drawing of our MVP.