Crowdsound: A Popularity-based Music Application

Inspired by the use of music streaming services (such as Spotify) at social gatherings, the Crowdsound web application allows partygoers to collectively choose the songs they want to hear. Usage of the application begins with a party host, who creates a new playlist from the landing page. After authenticating with Rdio, the host is directed to the playlist page and can begin streaming music. Each playlist is assigned a unique code for the host to share with guests, who use it to access the playlist via their mobile devices. Anyone viewing a playlist may search for songs and add them to the collaborative list. Users can then vote for songs already in the playlist, causing them to move closer to the top of the queue. Both desktop and mobile displays use web sockets to show playlist updates in real time, making for an engaging user experience.

The project was developed in collaboration with students Mackenzie Clark, Jian Shen Tan, and Koji Yamamoto. I contributed much of the server-side components, including Rdio authentication and search, API endpoints, and socket communication. Intended only for demonstration purposes, the first iteration of the application used server main memory to store playlist data. I later improved the project by implementing MongoDB as the data store, a fundamental step toward scalability.

Playlist page as viewed by the host. The top panel contains music player controls and current song information. Search results and the current playlist appear in center left and right panels, respectively.