403 Project Proposal
Music Player Development - Queue

Samantha Brender - samb0303
Blake - blakec20
Tuan - tuanvo
Michelle Park - parkm23

Vision:

Our product is a democratic music player called the "Queue." The Queue provides a voting system, which allows all members in a certain group-playlist to get involved with the choices of music at a gathering. All users can continuously modify the list by adding new songs to the queue and by voting for or against the music being played. By following the majority rule, the Queue will solve the problems of which songs to play for any group of people based on what the majority wants to hear, not what a single person picks. Potential competitors that provide similar services would be satellite radio applications such as Pandora. Whereas the pre-existing music players focuses on personalized music selections of individuals, the Queue focuses on collective involvement through multiple feedbacks from the members of a specific group.

Software Architecture:

At a high level Queue will consist of a web application which allows users to interact with a queue which is stored on the server. We will implement Queue's functionality using a priority queue which ranks songs according to the number of votes. Users will be able to use the web application to communicate with the server and vote on the songs in the queue. We will also implement a system to keep track of who has voted on each song so that a single user can't vote for a song repeatedly. Additionally the group host will be able to start and stop the music as well as add and remove songs from the queue. From a technical point of view some interesting aspects of this project are managing simultaneous changes to the queue data structure and designing the media player for the group host to use. For simplicity, our model will have a server that handles one party. If we complete our project early, we can refactor for multiple parties. We will need some kind of server side language that persistently keeps track of this queue. When the client app interfaces with the server, we will need to define metadata such as the server and client can utilize whichever language they want. However, node.js is a possibility to simplify client/server interaction since they will both be in JS. We might also use a database so that parties can store settings, such as saving their previous Queue for a later time.

Challenge and Risk:

One challenge in this project will be figuring out how to share music between users for the playlist. Figuring out how to access and cache the shared music for playback on another device will require quite a bit of thought and effort. Another challenge will be the concurrency issues. All users will need to be able to vote on songs and change what is in the Queue at the same time and we need to ensure that these simultaneous requests don’t cause race conditions which can damage the integrity of our queue. Making sure that user requests are handled appropriately will be highly important as well. We need to make sure that our server can properly and reliably handle requests in an efficient and timely manner. We will need to specify scenarios such as when users disconnect.