Part 1

1. Allow broadcaster to automatically create streaming listening stations from their local collection of mp3 files.
2. Users will be people who want to share their music (broadcasters) and people who want to hear other people's music (listeners).
   For authentication the listener will create an account with a password on the broadcast server.
3. For the demo, the broadcaster will show their screen, and start the server. Another user will then show their web browser tuning into the original broadcaster's station.
4. We will use Sqlite database in Java to store music metadata in order to cache the results for future use. Also we will use the database when the system is figuring out what song to play next in a particular station.
5. We will be using "Deezer" which is a REST API for the music metadata.

Part 2

1. As a broadcaster, I want to share my music so that other people can enjoy what I enjoy.

   My conditions of satisfaction are: A broadcast station allows a user to join and listen.

2. As a broadcaster, I want to find people that have the same musical tastes so that I can make friends that have similar interests.

   My conditions of satisfaction are: the broadcaster gets notification when the user joins the station.

3. As a listener, I want to socially listen to the same songs at the same time as my friends so that we can bond.

   My conditions of satisfaction are: A user in the same room as another user can see each other by some unique identifier such as a user name.

Part 3 (acceptance tests)

1. When the broadcast server starts, it will output how many songs it found, how many stations it created, and a notification that the web server started. The test concludes by visiting the website and seeing the genre choices, and tuning into their own station.
2. When the user joins a channel, the broadcaster will be notified that someone joined the station.
3. Two users join the same room, and can see each other's usernames.

Part 4

For the backend plan to utilize the same stack that was introduced in the Tic tac toe. That would include Checkstyle, Junit, Emma, spotbugs.

For frontend, we will use Javascript, MaterialUI and React. Tools include Eslint for static code analysis and JEST for testing.