**Date:** 16-11-2020

**Not In Attendance:** N/A

**What’s Been Done since the Previous Meeting:**

**James Angel: continued testing and set up of kubernates, created the room-service micro service.**

**Harrison Mcdonagh:** implement front end desings

**Kevin Ibelgauptas:** Set up a search for songs feature.

**Shao Wen Yong:** Created basic server side of WebSocket.

**What’s Being Done:**

**James Angel:** set up a kubernates cluster within the cloud.

**Harrison Mcdonagh:** adding CSS code to the web app and showing track list

**Kevin Ibelgauptas:** Queueing songs and playing them through our app.

**Shao Wen Yong:** Create client side of WebSocket and add more functions and states.

**Further Discussion:**

We should pair program some more. When we are working on things, we should post it in the Discord so someone else can ask to pair up.

For getting the currently playing song, the client making calls to Spotify would max out the number of allowed requests, so the RoomService should make requests instead.

Ideal file structure is illustrated below (to be updated to include Kubernates) and is also pinned in the Discord in the `general` channel.

### Folder Structure Layout Diagram

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Backend | API | Clients | room-service.js |  |
|  |  |  | kubernates.js |  |
|  |  | Room | routes | routes.js |
|  |  |  | controllers | controller.js |
|  | roomManager.js |  |  |  |
|  | index.js |  |  |  |
|  | test |  |  |  |
| Room-service | API | Client | Spotify.js |  |
|  |  | Room | routes | routes.js |
|  |  |  | controllers | controller.js |
|  | Music-Manager |  |  |  |
|  | Websocket |  |  |  |
|  | index.js |  |  |  |
|  | test |  |  |  |
| Client | src | components |  |  |
|  |  | pages |  |  |
|  |  | data | backend-client |  |
|  |  |  | websocket-client |  |