|  |  |
| --- | --- |
| **PROJECT REQUIREMENTS** | |
| **Project Name** | SoundByte |
| **Functional Requirements** | |
| Our system must take a song as input, or preset features, or given criteria. It must return a standard result(s) (10 likely, unless filter) across all given inputs.  It must employ a reliable machine learning model. This model must be trained from an easily accessible dataset.  Our suggestions must be songs from the dataset  The user must be able to observe the results & select input to download or playback as needed  Download must consist of a compressed file of mp3 files.  Playback must not be a download | |
| **Technical/Performance Requirements** | |
| Frontend webapp UI. View & Controller.  API Gateway & Backend NodeJS Functions  Python functions called by NodeJS  Database for dataset storage. Metadata, features.  System for propagating database  Storage server for mp3 songs. used in download and playback.  Minimum response time: 30 seconds. This is due to the api gateways minimum response time of 30s. May have to sidestep api gateway for download/playback.  download time: variable. cutoff: 30 minutes.  Playback time: variable. cutoff: 8 minutes.  Python model trained outside of app.  Models in python communicate with nodejs requests.  API to handle secure requests between frontend and backend  Model must be reliable and comparable to professionals tastes/preferences. While the song combination may not be “good”, the fact that it may mix easily is enough to rely on the creator’s inspiration. Add other features like genre to narrow down. | |

**Further requirements gathering in empathy maps, affinity maps, user story mapping, use case diagrams.**