# music mirror

## Problem

- Existing DJs and stereo systems do not efficiently collect and manage user song requests, or ensure that these requests are representative of the collective event
- The quality of song sets are solely dependent on the quality of the DJ, which can be expensive and incapable of adjusting to event environments

**Team B3**: Luke Marolda, Thomas Lee, Matt Hegi

# Use-Case

- A comprehensive speaker attachment that seamlessly manages queuing, song recommendations, and crowd engagement
- Users steer the system through a distributed web app that hosts a suite of song request and consensus voting capabilities

## Existing Solutions

- Current systems are singular - they focus on one person having full control. We democratize the event listening experience for uniform enjoyment

## Areas

- Software Systems, Machine Learning, Hardware Systems

# Requirement #1

Effectively engage with the crowd environment (Engage with Users)

## Motivation

- A successful DJ engages with the audience audibly and visually

## Sub-Requirements

- System ability to mount to any functional Bluetooth speaker
- Easily usable mobile-optimized website
    - Users will be onboarded in under **1** minute on average
    - Predictable and consistent web app behavior to User inputs
- Light system colors and strobing match song genre, tone, and crowd loudness noise sensor

# Requirement #2

Accurately process the users’ collective music requests  (Listen to Users)

## Motivation

- Everyone must have an equal ability to contribute to what gets played

## Sub-Requirements

- **3** direct song request formats. Implemented with a semantic matching algorithm to map requests with queried spotify resources
  - By name of song
  - By artist or album
  - By songs that have already been played

# Requirement #2 (cont’d)

- User song requests are accurately reflected by the centralized queue within **1 second**
- Centralized concurrent queue to accept and maintain ordering of incoming song requests for a target of **100-150 users**
- Consensus voting protocol to support ‘veto’ functionality of songs on queue
- Queue can hold at least **100 songs** (6 hour reception / 3.5 min average song length)

# Requirement #3

Generate song recommendations that resonate with users  (Serve Users content)

## Motivation

- A great DJ injects creativity to introduce songs that users don't immediately think of but will enjoy

## Sub-Requirements

- Machine learning recommendation system
    - Generates song suggestions in a multimodal sense, using data from the MusicBrainz database, environment noise sensor, and user input

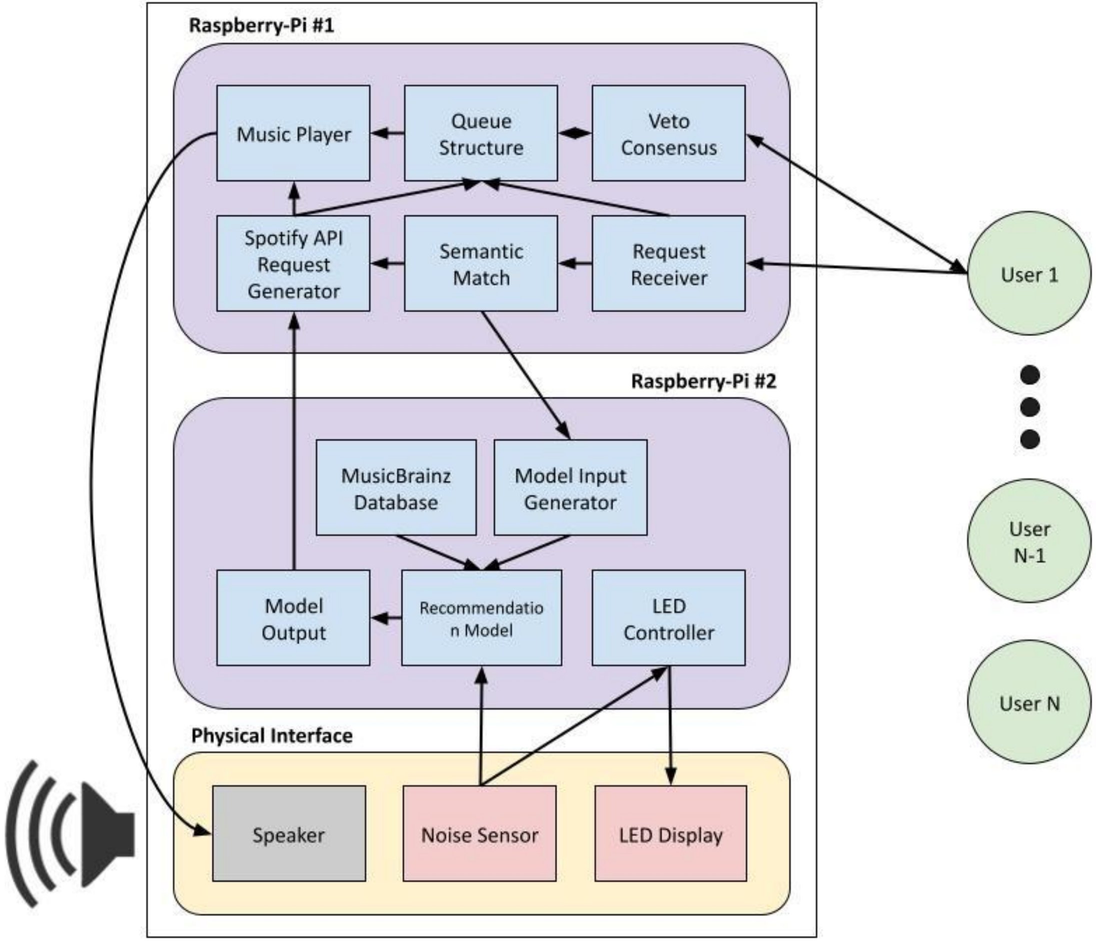
# Requirement #3 (cont'd)

- **1** additional song request format: Similarity search
  - Ability to generate song requests based on what has already been played
- Endless queue
  - Queue should never be empty
  - System can input creative song choices every **3-5** user requests

# Technical Challenges

- Accurately accepts user requests and places them in the correct order on the queue
- Ability to semantically match user request to correct song resources and play them on the speaker
- Robust protocols to manage concurrent users, maintain song queue consistency, and allow for veto mechanism
- Tuning a song recommendation model to achieve desired level of user satisfaction accuracy at both small and large user request volumes
- Efficient integration of subsystems to reduce User to System latency
- Easily understandable user interface within web app
- Noise sensors accurately detect & responds to user voice/volume inputs
- Light system colors & strobing match song genre, tone, and BPM in real time

# Solution Approach



music mirror

# Testing, Verification, and Metrics

## System Correctness

- Verify that song and sound requests are properly reflected by the centralized queue and DJ system behavior
- Semantic match for direct song requests reaches **90%** efficiency in obtaining resources for correct requests, and **70%** accuracy for incorrect requests (misspelling, etc.)

## Latency Clocking

- Use wall-to-wall clocks to time how long different User requests take to be accepted and processed by the system

## User Satisfaction

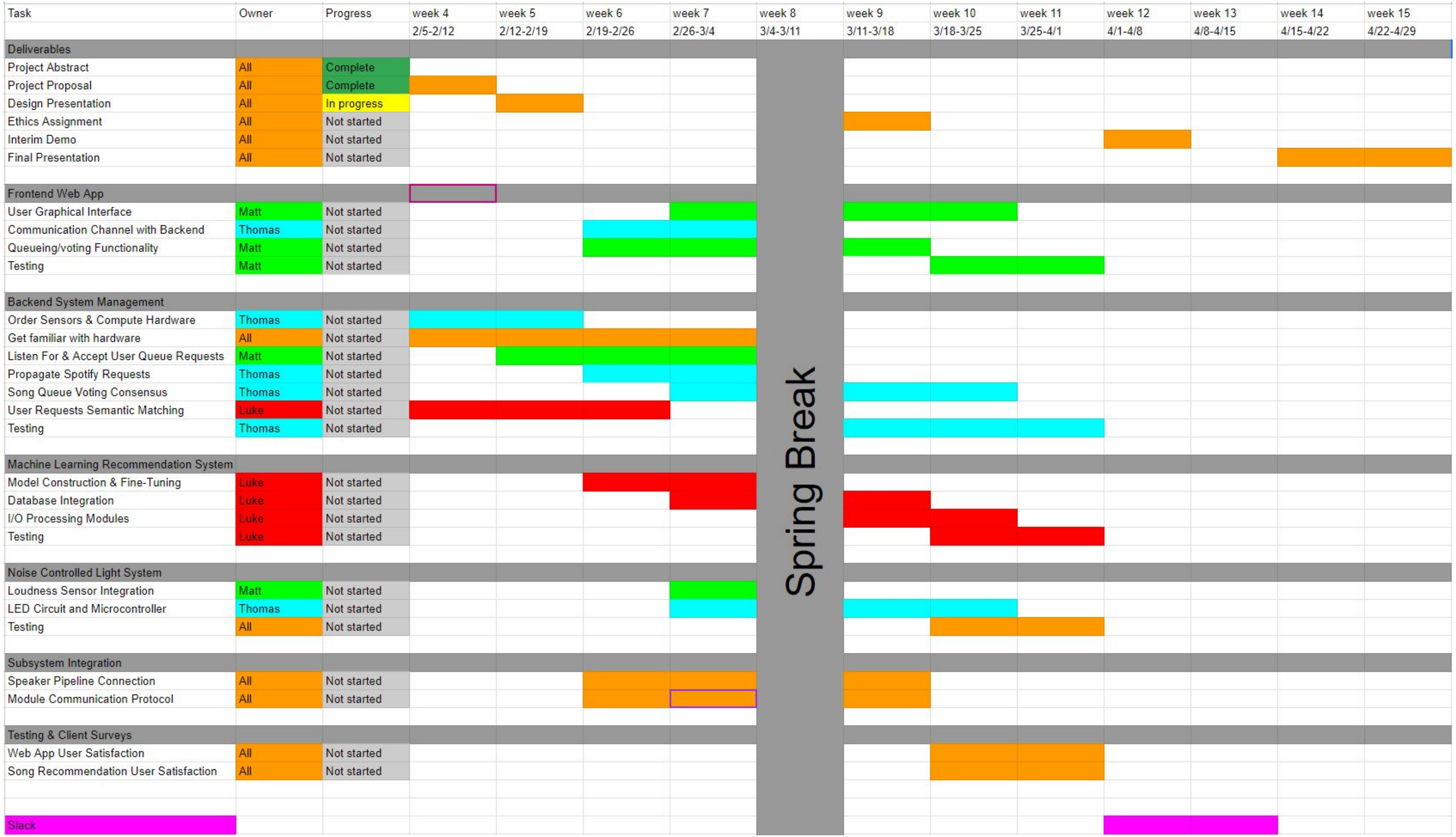
- Time Users to determine how quickly they can learn to use our web app interface
- Poll Users on how satisfied they are with DJ generated song recommendations that were based on songs they queued, aiming for **75%** approval

## Stress Testing

- Leverage scripting to simulate large user count, request volume, and queue size, and observe system stability and performance under load

# Task Distribution

- Frontend web app (Matt)
- Backend system management
    - Queuing system (Matt)
    - Spotify requests (Thomas)
    - Consensus voting (Thomas)
    - Semantic matching (Luke)
- Machine Learning Recommendation System
    - Model construction and tuning (Luke)
    - Database integration (Luke)
    - Input/output processing modules (Luke)
- Noise controlled lights
    - Loudness sensor integration (Matt)
    - LED circuit and controller (Thomas)
- Subsystem integration
    - Speaker connection (Everyone)
    - Communication protocol between modules (Everyone)
- Testing and client surveys (Everyone)



| Task | Owner | Progress | week 4 | week 5 | week 6 | week 7 | week 8 | week 9 | week 10 | week 11 | week 12 | week 13 | week 14 | week 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 2/5-2/12 | 2/12-2/19 | 2/19-2/26 | 2/26-3/4 | 3/4-3/11 | 3/11-3/18 | 3/18-3/25 | 3/25-4/1 | 4/1-4/8 | 4/8-4/15 | 4/15-4/22 | 4/22-4/29 |
| Deliverables | | | | | | | | | | | | | | |
| Project Abstract | All | Complete | | | | | Spring Break | | | | | | | |
| Project Proposal | All | Complete | 🟧 | | | | | | | | | | | |
| Design Presentation | All | In progress | | 🟧 | | | | | | | | | | |
| Ethics Assignment | All | Not started | | | | | | 🟧 | | | | | | |
| Interim Demo | All | Not started | | | | | | | | | 🟧 | | | |
| Final Presentation | All | Not started | | | | | | | | | | | 🟧 | 🟧 |
| | | | | | | | | | | | | | | |
| Frontend Web App | | | | | | | | | | | | | | |
| User Graphical Interface | Matt | Not started | | | | 🟩 | | 🟩 | 🟩 | | | | | |
| Communication Channel with Backend | Thomas | Not started | | | 🟦 | 🟦 | | | | | | | | |
| Queueing/voting Functionality | Matt | Not started | | | 🟩 | 🟩 | | 🟩 | | | | | | |
| Testing | Matt | Not started | | | | | | | 🟩 | 🟩 | | | | |
| | | | | | | | | | | | | | | |
| Backend System Management | | | | | | | | | | | | | | |
| Order Sensors & Compute Hardware | Thomas | Not started | 🟦 | 🟦 | | | | | | | | | | |
| Get familiar with hardware | All | Not started | 🟧 | 🟧 | 🟧 | 🟧 | | | | | | | | |
| Listen For & Accept User Queue Requests | Matt | Not started | | 🟩 | 🟩 | 🟩 | | | | | | | | |
| Propagate Spotify Requests | Thomas | Not started | | | 🟦 | 🟦 | | | | | | | | |
| Song Queue Voting Consensus | Thomas | Not started | | | | 🟦 | | 🟦 | 🟦 | | | | | |
| User Requests Semantic Matching | Luke | Not started | 🟥 | 🟥 | 🟥 | | | | | | | | | |
| Testing | Thomas | Not started | | | | | | 🟦 | 🟦 | 🟦 | | | | |
| | | | | | | | | | | | | | | |
| Machine Learning Recommendation System | | | | | | | | | | | | | | |
| Model Construction & Fine-Tuning | Luke | Not started | | | 🟥 | 🟥 | | | | | | | | |
| Database Integration | Luke | Not started | | | | 🟥 | | 🟥 | | | | | | |
| I/O Processing Modules | Luke | Not started | | | | | | 🟥 | 🟥 | | | | | |
| Testing | Luke | Not started | | | | | | | 🟥 | 🟥 | | | | |
| | | | | | | | | | | | | | | |
| Noise Controlled Light System | | | | | | | | | | | | | | |
| Loudness Sensor Integration | Matt | Not started | | | | 🟩 | | | | | | | | |
| LED Circuit and Microcontroller | Thomas | Not started | | | | 🟦 | | 🟦 | 🟦 | | | | | |
| Testing | All | Not started | | | | | | | 🟧 | 🟧 | | | | |
| | | | | | | | | | | | | | | |
| Subsystem Integration | | | | | | | | | | | | | | |
| Speaker Pipeline Connection | All | Not started | | | 🟧 | 🟧 | | 🟧 | | | | | | |
| Module Communication Protocol | All | Not started | | | 🟧 | 🟧 | | 🟧 | | | | | | |
| | | | | | | | | | | | | | | |
| Testing & Client Surveys | | | | | | | | | | | | | | |
| Web App User Satisfaction | All | Not started | | | | | | | 🟧 | 🟧 | | | | |
| Song Recommendation User Satisfaction | All | Not started | | | | | | | 🟧 | 🟧 | | | | |
| | | | | | | | | | | | | | | |
| Slack | | | | | | | | | 🟪 | 🟪 | | | | |