Software Project Proposal

Soundcheck!
Release 1.0

Prepared by Sankalp Doshi
Jackson Kallen
Sakin Kirti
Hyunjoon Kweon
Josh Levy
Brian Song

January 20, 2023

1. Scope and Vision

1.1 Background

Have you ever been curious about what your friends are listening to at the same time you are? Are you interested in expanding your music taste? We are proud to introduce Soundcheck! BeReal is a social media application that captures an image at a random time in the day and you post it on your feed. We have a similar idea where at a random moment in time, we will send a notification on your phone asking you “What are you listening to?”. We plan on connecting Spotify to this application so it will post the current song you are listening to. If you aren’t listening to a song at that current point in time, it will put the last song you listened to for you to post. We don’t believe we will have enough time during the semester to completely develop an application for your mobile device, so we will implement a desktop application that does all of what Soundcheck! should do. Our final deliverable will be a desktop application that allows a user to select the notification, get your current/last listened to song, and send it to your feed so your friends can view your selected track.

1.2 Vision Statement

Soundcheck! is an application that allows users to post the song they are currently listening to or the last song they listened to to a feed for their friends to see. A notification will be sent out to users at a random time each day to alert the user that it is time to post what song they are listening to. The purpose of this application is to give users insight into their friends' musical tastes, expand their own musical taste, and opportunities to interact based on these posts with “likes” and comments.

1.3 Success and Criteria Measures

Success will be measured based on several criteria.

1. We want to ensure the efficiency and effectiveness of our implemented algorithms.
    a. On the backend, we will rely heavily on the use of python unittest to test our methods and algorithms across detailed edge cases and normal use cases.
    b. On the frontend, we will rely heavily on the use of Cypress to test methods and implementation of frontend components.
2. We want to ensure the goals that we set for ourselves in this proposal are met. Therefore, in section 3, we highlight major milestones that individuals need to meet. Backend and Frontend developer teams will meet regularly to break down these major milestones and assign individual tasks. Success will be measured based on the proportion of milestones completed and their relative priorities.

2. Detailed Feature Description

2.1 Feature Descriptions

FE-1: A user can sign in to the application with their spotify account
FE-2: A user can create a post daily when a notification has been sent by the developers
FE-3: A user can edit their profile and follow friends
FE-4: A user can view the posts of those that they follow in their feed
FE-5: Each Saturday a playlist will be made of the songs your friends have posted during the week

2.2 Feature Dependencies and Constraints

DE-1: Soundcheck! relies on external APIs and the application will not function properly when the APIs are down or not updating.
CO-1: The system's design, code, and maintenance documentation shall conform to the standards dictated in the CSDS 493 Software Engineering course.
CO-2: All HTML code shall conform to the HTML 5.0 standard.

2.3 Operating Environment

OE-1: This application should be run-able on any updated Google Chrome browser on all desktop computers.

2.4 Design and Implementation

Based on early planning, the proposed web application will appear similar to the image below. Importantly, note the scrollable window on the left which displays the songs along with icons to like a post by one's friends. Additionally, on the right side, a non-scrolling window displaying the individual's profile with their song and number of likes given to their post for the day. Additionally, below that is a list of the user's friends. At the top, the bar displays the app's name as well as a button with the word user for one to click on to view their profile.

2.5 Software Interfaces

The following is a list of interfaces that we will need to implement or use to interact with users and Spotify.

SI-1	User Information Database	A cloud-based database to store all user information. This includes an individual's posts and their playlists, etc.

SI-2	Spotify Application Programming Interface (API)	The Spotify API is central to this application. From the Spotify API users can login, and our app can get current song and all details, etc.

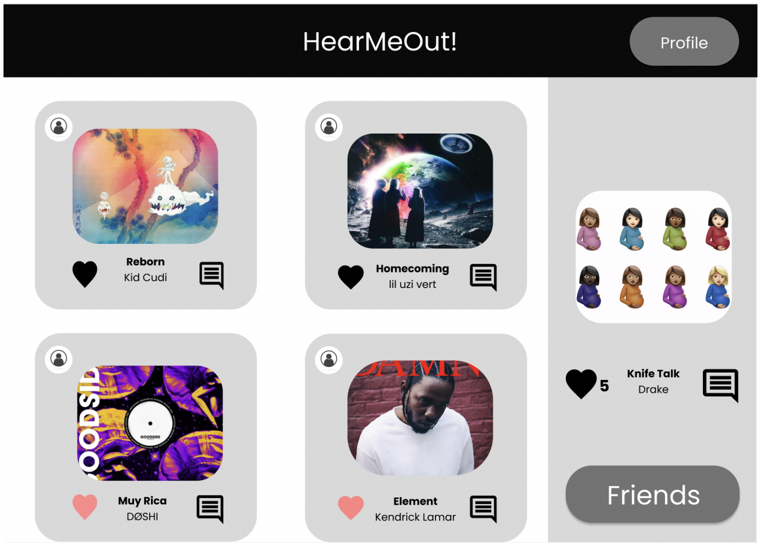

Figure 1: Soundcheck! user interface created on Figma

3. Prioritized Deliverable List and Estimates

- Anywhere where the term 'backend dev' is used, it refers to Sankalp, Brian, Sakin, and Hyunjoon.
- Anywhere where the term 'frontend dev' is used, it refers to Jackson, Brian, and Josh.
- Note that these teams are general, and a person from one team might work on another team when needed.

Priority	Name	Owner	Time Estimate	Description
1	User Interface Planning	Frontend Devs	≤1 Week	Plan out the interface look to get an idea of what the final product should look like. (In Figma)
2	Spotify API	Backend Devs	1-2 weeks	Familiarizing with the Spotify API + identifying how to retrieve necessary data from the API useful for our project.
3	Storing User Information	Backend Devs	1-2 weeks	Plan database schema, check schema with group for efficiency. Implement in cloud (AWS RDS - likely using PostgreSQL or MySQL)

4	Creating Algorithms	All	4-8 weeks	Creating algorithms for the backend portion of Soundcheck!
5	Planning Structure of Frontend	Frontend Devs	1-2 Weeks	Deciding on the libraries to be used and the desired and necessary components to implement. Plan this in a manner to efficiently reuse code.
6	Frontend Structure Implementation	Frontend Devs	3-4 Weeks (Subject to change)	Implement the desired visual features and structure without backend features. Basic backend features should be integrated.
7	Frontend and Backend Complete Integration	Frontend Devs	3-4 Weeks (Subject to change)	Integrate backend features to the user interface.
8	Quality of Life and Polishing Changes	Full Stack		Test the product with testing frameworks and users of the application to determine issues with application that need fixing or tweaking.

Management Plan

The team will be in constant communication throughout the project duration. An agile work environment will be set in place. Towards the early development and research portion of the project, the team will meet for bi weekly scrum meetings for all team members to be aware and in communication with the progress of the application. Towards the latter part of the semester, communication will be increased in effort of development. The management application Jira will be used to keep track of our week to week progress (sprints), as well as logging what features are in development/have been completed.