Team 8 Sprint 2 Retrospective Document: Kadence

Jackson Rosenberg, Avery Schaefer, Nathan Simon, Colston Streit, Raymond Xie
rosenbe7@purdue.edu, schaef35@purdue.edu, simon70@purdue.edu,
cstreit@purdue.edu, xie328@purdue.edu

What Went Well:

We believe that overall, this sprint was very successful. We were able to complete general song playback with both Apple Music and Spotify, as well as implementing those APIs in more specific applications, by completing Mood Mode and Interval Mode. Also, we completed a much-needed redesign of our User Interface, making the UI consistent and easy-to-use on every page. Additionally, we added the ability for users to search for and view other users, setting a foundation for the social features we plan to add in Sprint 3.

User Story #2: KAD-42

As a user, I would like to select from different playlist generation modes

#	Description	Estimated Time	Owner
1	Create home page	9 Hours	Nathan
2	Create place to select playlist mode	3 Hours	Nathan
3	Create button to begin playlist generation	1 Hours	Nathan
4	Create viewport for users to see generated playlist	6 Hours	Raymond
5	Using the music platform databases, take the preferences the user provided and store artist and song information to used for playlist generation (favorites and to blacklist)	6 Hours	Jack
6	Save all songs generated by the playlist algorithm into a list that the user can choose to then save as a permanent playlist to their music platform profile of choice	8 Hours	Jack
7	Write API routes and unit tests for users to save playlist to their connected music platform	4 Hours	Jack
Total User Story Time		37 Hours	

Completed:

We created a home page for the app, where users can pick from any of our four playlist modes. When they click the button for one of the two modes implemented so far, they are sent to the page associated with that mode and, after any associated preferences are input/gathered, playlist generation is started.

User Story #3: KAD-77

As a user, I would like to be able to interact with my music platform from within the Kadence app.

#	Description	Estimated Time	Owner
1	Design and create UI components for an in-app music player (including play, pause, and skip buttons).	4 Hours	Colston
2	Implement the play and pause buttons for the music platforms.	9 Hours	Colston
3	Implement the skip button for the music platforms.	9 Hours	Colston
Total User Story Time		22 Hours	

Completed:

We created an in-app player page, which allows users to see the current song they are listening to, as well as its album art and the current progress in the song. Also, users are able to pause/play the song, as well as skip to the next song. Additionally, we created a mini component version of the player, which allows for some functionality of the player while a user is viewing another page within the Kadence app. These work for both Spotify and Apple Music.

User Story #4: KAD-75

As a user, I would like to start playlist generation for the “interval” mode.

#	Description	Estimated Time	Owner
1	Design UI elements required for user to start playlist generation	9 Hours	Raymond
2	Create API endpoint that uses the “long” and “short” times stored in a user’s preferences to control generation	10 Hours	Avery
3	Create an algorithm that uses user preferences to generate a requested number of songs to add to the queue	8 Hours	Jack
4	Use the backend API endpoint to add the songs returned by the algorithm to the user’s play queue	7 Hours	Avery
5	Write unit tests for “interval” generation	2 Hours	Jack
Total User Story Time		36 Hours	

Completed:

The user can now access interval mode from the home page. Upon reaching the page for the generation, the user can select the times they prefer for the “high” and “low” energy intervals and then move from a “pre-interval” staging area to the player featuring actual generation. The current song for the user serves as the seed song for generation, and depending on the current interval, a single song with filtered criteria is added to the queue. When the song changes, a new song is queued and the generation continues like this until the user chooses to end the session. Lastly, the user selects whether or not to save the items generated during this session to their profile in a playlist.

User Story #5: KAD-38

As a user, I would like to start playlist generation for the “mood” mode.

#	Description	Estimated Time	Owner
1	Design UI elements required for user to start playlist generation	5 Hours	Nathan
2	Create API endpoint that uses the user’s selected mood as a seed to generate songs	7 Hours	Avery
3	Create an algorithm that uses user preferences to generate a requested number of songs to add to the queue	5 Hours	Jack
4	Use the backend API endpoint to add all songs returned by the algorithm to the user’s play queue	5 Hours	Jack
5	Write unit tests for “mood” generation	2 Hours	Jack
Total User Story Time		24 Hours	

Completed:

The user can now access “mood” mode from the home page. The user can select the mood they want the playlist to be centered around from a list and use a seed song that they are currently listening to generate a playlist of their desired length. The user can specify to save the playlist to their profile before generation or have the option to choose to save after their session is over. The songs generated also will be put into the user’s queue so they can listen in our app instead of having to switch over to Spotify to begin playback.

User Story #6: KAD-D6, 66

As a developer, I would like the app to have an intuitive and consistent user interface.

#	Description	Estimated Time	Owner
1	Redesign registerInfo page	6 Hours	Nathan
2	Add profile picture to account	8 Hours	Raymond
3	Make UI consistent across all pages	6 Hours	Nathan
4	Redesign settings page	4 Hours	Raymond
5	Write automated tests for frontend components	10 Hours	Raymond
Total User Story Time		34 Hours	

Completed:

We completed a comprehensive overhaul of our User Interface. Every page in the app received a new design, featuring new components and new visual layouts. These both serve the purpose of making the app easier to use and more visually appealing to users.

User Story #7: KAD-53, 57

As a user, I would like to search for other users and view their profile pages.

#	Description	Estimated Time	Owner
1	Add an API endpoint to search for other users by their username.	4 Hours	Colston
2	Create UI components for the search page.	10 Hours	Colston
3	Display the searched user's profile information according to their profile visibility settings.	2 Hours	Colston
4	Write tests for the Search API endpoint	2 Hours	Colston
Total User Story Time		18 Hours	

Completed:

We implemented a search page, which allows users to search for profiles by their username. The search page will return any profile with a username containing the searched term. Users are able to tap on a result item and view the associated profile, which contains some basic information about the profile. However, if the selected user has their profile visibility set to "private", some information about the user (such as their connected platforms and devices) is hidden.

What Did Not Go Well:

The primary struggle with this sprint was with the Fitbit integration. In the original version of our Sprint Planning document, we planned on adding implementation with Huawei fitness watches in Kadence. However, due to unforeseen data privacy concerns, we elected to switch our supported fitness watch brand from Huawei to Fitbit. Switching this mid-sprint caused some issues with completing our original goals, due to the differences between the two APIs. Additionally, applying for Fitbit's developer access required additional work, including having fully written Terms of Service and Privacy Policy documents for our app. This extra overhead work caused us to have to modify our goals, moving the task of gathering fitness data from the paired device to Sprint 3.

User Story #1: KAD-18, 20, 21, 22, 23, 68, D3

As a user, I would like to link, update, and delete my fitness watch in the app.

#	Description	Estimated Time	Owner
1	Create backend functionality for users to link their Fitbit account (and in turn device) to their Kadence account	8 Hours	Avery
2	Design relevant UI components to connect a new Fitbit account	5 Hours	Nathan
3	Design relevant UI components with proper formatting to display the current connected Fitbit account	5 Hours	Nathan
4	Create functionality to allow users to update and delete linked account	3 Hours	Nathan
5	Write a Terms of Service and Privacy Policy for Kadence, and use it to apply for Fitbit Developer Access	10 Hours	Avery
6	Test device connection and functionality	3 Hours	Raymond
Total User Story Time		38 Hours	

Not Completed:

As earlier stated, the primary issue with this user story stemmed from the mid-switch from using Huawei smart watches to Fitbit smart watches, and the additional work required to make this switch. Because of that extra work, we needed to push back the task of pinging the paired Fitbit account for heart rate data to Sprint 3. However, we were able to complete the work required for the switch, meaning that we are in a good position to complete that task in this upcoming sprint.

How We Will Improve:

For the next sprint, the primary focus will be on eliminating the gap in knowledge between frontend and backend developers, making communication more effective. We will work on planning out more before jumping in to developing, as that seemed to be something that caused us issues throughout this second sprint. We will be making sure to work out exactly how each page will work and look, since that will mean that both the frontend and backend developers will be on the same page before starting work, eliminating time spent debating what a page is intended to look like.