Team 8 Sprint 1 Planning 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

Sprint Overview:

This first sprint will revolve around the base groundwork for the project. To start, we'll lay the groundwork for our project, adding the required frameworks and beginning the work on hosting the project. We will be making a skeleton UI, building out the base structure of the pages that can then be given functionality later in the project. Additionally, we will be working on the basic functionality for account creation, reading, updating, and deletion (CRUD) operations, as well as the processes for logging in/out and creating a new profile. Finally, we will implement the functionality required for users to link their music platform accounts and their fitness devices.

Scrum Master: *Nathan Simon*

Meeting Schedule: *MWF 3:00-4:00pm*

Risks and Challenges:

This first sprint will mainly consist of the base setup for the project, which we will be building on in the subsequent sprints. Because of that, we need to take extra care this sprint to ensure everything works as it should to prevent compounding issues down the road. Additionally, some of the documentation and functionality for the APIs that are required for this project are murky at best, so we will have to be flexible to adapt to any issues that may arise from that. Lastly, we will be working on learning about each other's development styles, figuring out what communication and collaboration methods work best.

User Story #1:

As a developer, I would like a MongoDB database connected to the backend.

#	Description	Estimated Time	Owner
1	Read MongoDB Query Documentation	3 hours	Avery
2	Create and format a MongoDB database	2 hours	Avery
3	Test CRUD operations on database using mock data and Postman	3 hours	Avery
4	Link MongoDB database to Node/Express backend	3 hours	Avery
Total User Story Time		11 Hours	

Acceptance Criteria

- ☑ ~~Given that the MongoDB database is created and formatted correctly, we will be able to manually add/update data elements to it~~
- ☐ Given that the MongoDB database is properly set up, we will be able to perform basic CRUD operations on it using Postman
- ☑ ~~Given that the MongoDB database is created, we will be able to link it properly to the backend of the app.~~

User Story #2:

As a developer, I would like a functional Next.js frontend.

#	Description	Estimated Time	Owner
1	Research and familiarize with Next.js framework	4 hours	Avery
2	Create a blank Next.js frontend	2 hour	Avery
3	Connect Capacitor to blank Next.js frontend	2 hour	Avery
Total User Story Time		8 Hours	

Acceptance Criteria

- ☑ ~~Given that a functional Next.js frontend is created, we should be able to build the project without issue~~
- ☑ ~~Given that Capacitor is connected properly to the Next.js frontend, we should be able to compile and deploy the project to an Android device~~
- ☐ Given that Capacitor is connected properly to the Next.js frontend, we should be able to compile and deploy the project to an iOS device

User Story #3: KAD-69

As a developer, I would like a functional Node/Express backend.

#	Description	Estimated Time	Owner
1	Research and familiarize ourselves with how to use Node and Express.	2 Hours	Avery, Raymond, Jack
2	Create a runnable Node/Express backend server with an organized file structure.	6 Hours	Raymond
3	Sketch out a list of all the API endpoints that we want our backend to expose to our frontend.	3 Hours	Avery
4	Implement a skeleton version of each of these endpoints that returns some placeholder data (to be filled in later when functionality is added).	8 Hours	Jack
Total User Story Time		23 Hours	

Acceptance Criteria

- ☐ Given that the backend is configured correctly, when I run the command to run the server, the server should begin running with no errors.
- ☐ Given that our list of API endpoints to include is conclusive, as we move forward through the project, we should not have to add any additional API endpoints onto that list and implement them on the backend.
- ☐ Given that a skeleton version of each API endpoint is correctly implemented, when I use an external tool (like Postman) to issue GET/POST/DELETE requests to the server while it is running, the relevant code handlers should be executed and the appropriate placeholder response should be returned.

User Story #4: KAD-71

As a developer, I would like to integrate the frontend and backend.

#	Description	Estimated Time	Owner
1	Add functionality to the relevant UI components to make requests to each of the backend API endpoints.	9 Hours	Colston
2	On the frontend, display the data returned by the backend for each API endpoint in some way.	5 Hours	Colston
3	Implement preliminary error handling for each of the backend endpoints, as well as for each of the requests made from the frontend.	4 Hours	Colston
4	Test that all of the backend API endpoints are reachable from the frontend and return the expected placeholder data.	6 Hours	Colston
Total User Story Time		24 Hours	

Acceptance Criteria

- □ Given that all backend API endpoints are reachable from our frontend, whenever I want to make a request to a given backend API endpoint, there should be some UI component that I can use to make that request.
- □ Given that a given UI component correctly queries its corresponding backend API endpoint, when a request to the endpoint is made by the component, the appropriate data is returned and displayed in some way on the screen (even if the method of display isn’t final).
- □ Given that error handling for the frontend is handled correctly, if a request is made from the frontend but cannot go through because of (for example) Internet issues, the error should be displayed on the screen for the user to see.
- □ Given that error handling for the backend is handled correctly, if a request is made from the frontend that does not match the format expected by the backend, an appropriate error status code will be sent back and displayed on the screen for the user to see.

User Story #5: KAD-5, 6, 7, 8

As a user, I would like to register an account for Kadence, log-in, and delete my account if desired.

#	Description	Estimated Time	Owner
1	Create tables in the database for storing user login information and implement security features / hashing algorithm for properly storing passwords	6 hours	Jack
2	Create algorithm to receive login information from database and check entered password	8 hours	Avery
3	Develop forgotten password retrieval functionality	5 hours	Jack
4	Develop password strength checking algorithm	2 hours	Jack
5	Create unit tests for checking password strength	2 hours	Jack
6	Design relevant UI components to assist registering for new accounts	3 hours	Colston
7	Design relevant UI components to assist logging in and logging out of already made and stored accounts	3 hours	Colston
8	Add querying in order to remove accounts from the database on request	2 hours	Jack
9	Create manual testing sequences for account management	2 hours	Jack
Total User Story Time		33 Hours	

Acceptance Criteria

- ☐ Given that the database is configured correctly, when a user registers for a new account, their information should persist and Kadence should allow the user to login with their username and password.
- ☐ Given that the password strength algorithm is implemented correctly, when the user enters a weak password that doesn't meet Kadence's standards when registering, Kadence will not allow the user to create their account.

- □ Given that the database stores the account information correctly, when a user enters the correct username and password, they should be brought to their profile page with the navigation bar ready to use the app.
- □ Given that the database is configured correctly and the querying is properly done, when a user requests to have their account deleted, they should not be able login to the app using their old credentials.
- □ Given that the authentication is integrated correctly, when a user requests to log-out, they should be prompted to log-in again and have no functionality available to them.
- □ Given the forgotten password algorithm is implemented correctly, when a user requests a password change, the user should receive a redirection link to update it in an email associated with their account.

User Story #6: KAD-12, 13, 14, 15

As a user, I would like to link, update, or remove my music platform of choice within the app.

#	Description	Estimated Time	Owner
1	Use the Spotify and Apple Music APIs to connect and retrieve data about user accounts (with a focus on third party account authentication)	11 hours	Jack
2	Design relevant UI components to connect a new platform upon registering	4 hours	Nathan
3	Design relevant UI components with proper formatting to display the current connected account	4 hours	Nathan
4	Allow users to remove current connected platform	1 hour	Avery
5	Allow users to update current connected platform	1 hour	Avery
6	Allow users to leave Kadence and access their specified music platform	2 hours	Raymond
Total User Story Time		23 Hours	

Acceptance Criteria

- ☐ Given that the connections are implemented correctly, when I register my account, I would like to be prompted to connect to a music platform of my choosing.
- ☐ Given that the UI is implemented correctly, when I visit the settings of my profile, I should be able to see the platform I have connected to, including some minor details about the account.
- ☐ Given that the updating platform functionality is implemented correctly, when I switch the subscription platform I use, I should be able to remove / update the platform from my Kadence profile and begin playlist generation again.
- ☐ Given that the connections are implemented correctly, when I visit my current connections in my settings, I should have the option to click a link to take me to either my Apple Music or Spotify account

User Story #7: KAD-18, 20, 21, 22, 23, 68

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

#	Description	Estimated Time	Owner
1	Use bluetooth to identify nearby devices.	4 Hours	Raymond
2	Use the Huawei API to connect to and retrieve data from the device.	6 Hours	Avery
3	Create a default function that will select the preferred device if two are detected.	2 Hours	Avery
4	Design relevant UI components to connect a new device registering	5 Hours	Nathan
5	Design relevant UI components with proper formatting to display the current connected devices	4 Hours	Nathan
6	Create functionality to allow users to update and delete devices	3 Hours	Nathan
7	Test device connection and functionality	3 Hours	Raymond
Total User Story Time		27 Hours	

Acceptance Criteria

- ☐ Given that the device discovery is set up correctly, when I search for nearby devices, then the device I am trying to connect should pop up if it is in range.
- ☐ Given that the Huawei API is integrated correctly, when a user connects a device, information about that device should be sent to the backend and associated with that user.
- ☐ Given that the default device functionality is set up correctly, when a user connects two devices, the app will only retrieve fitness data from the default device that the user selects.
- ☐ Given that the update and deletion fields are set up correctly, when a user updates or removes a device, those changes will be reflected in both the backend and the frontend.

User Story #8: KAD-10, 11, 65, 66, 67

As a user, I would like to have and be able to update a profile.

#	Description	Estimated Time	Owner
1	Design layout of profile page.	6 Hours	Nathan
2	Create a form to request user information upon signing up for the app.	4 Hours	Nathan
3	Create a backend request to get user info.	4 Hours	Nathan
4	Link form with the backend request	3 Hours	Nathan
5	Allow users to update information on their profile.	3 Hours	Nathan
Total User Story Time		20 Hours	

Acceptance Criteria

- □ Given that the profile prompt is set up correctly, when a user signs up for the app, they will be prompted to input information about themselves that will populate their profile page.
- □ Given that the profile page is set up correctly, when a user views their profile page, the correct information will be displayed.
- □ Given that the updating information is implemented correctly, when a user updates their information, those changes will be made on both the frontend and the backend.

User Story #9: KAD-17

As a user, I would like to be able to update my default music and device preferences.

#	Description	Estimated Time	Owner
1	Expand upon design of user settings page from mockup	3 Hours	Colston
2	Create form UI for default music preferences page	2 Hours	Colston
3	Create form UI for default device preferences page	2 Hours	Colston
4	Implement storage of default music preferences in database	3 Hours	Colston
Total User Story Time		10 Hours	

Acceptance Criteria

- ☐ Given that the front-end app is up, when a user navigates to the music preferences page, a form that allows a default genre and artist to be set is displayed.
- ☐ Given that the front-end app is up, when a user navigates to the device preferences page, a form that allows a default device to be set is displayed.
- ☐ Given that the user has filled out their default preferences, when they press “Save”, the changes are persisted to the database.

User Story #10: KAD-73

As a developer, I would like the repository to have a Continuous Integration pipeline.

#	Description	Estimated Time	Owner
1	Research Docker and containerize the back-end app with Docker	6 Hours	Raymond
2	Set up ESLint to check for code quality	2 Hours	Raymond
3	Create a Github Action workflow to automatically build the app upon PR creation	4 Hours	Raymond
4	Add a step to the above workflow that automatically runs the test suite upon PR creation	2 Hours	Raymond
Total User Story Time		14 Hours	

Acceptance Criteria

- ☐ Given that the back-end is containerized, when a developer locally builds and starts the backend container, the API will be available from a Docker container.
- ☐ Given a GitHub Action workflow is in place, when a pull request is created, the application is automatically built to check for build errors.
- ☐ Given a GitHub Action workflow is in place, when a pull request is created, an automated unit test suite is run against the new build, which must pass in order to merge.
- ☐ Given that ESLint is set up in our project, when a pull request is created, merging is prevented unless all code style rules are followed.

User Story #11: KAD-73

As a developer, I would like a way to automatically deploy the back-end API with a Continuous Deployment pipeline.

#	Description	Estimated Time	Owner
1	Create a Github Action workflow that pushes the app image to a container registry upon merging a branch	4 Hours	Raymond
2	Test the manual creation of an Azure Container Instance running our API	2 Hour	Raymond
3	Set up a continuous deployment pipeline in Azure	3 Hours	Raymond
Total User Story Time		9 Hours	

Acceptance Criteria

- □ Given a GitHub Action workflow is in place, when a pull request is merged in, the latest Docker image build of our back-end API is automatically pushed to a container registry.
- □ Given that a CD pipeline is set up in Azure, when a new image is pushed to the container registry, the back-end API is automatically deployed with the new code changes.
- □ Given that the app is redeployed, when API endpoints are hit, the backend API is available and responds with the new code changes.

Total Time This Sprint: 202 Hours

Name	Estimated Time
Jackson	40
Avery	42
Nathan	40
Colston	40
Raymond	40

Remaining Product Backlog

Story ID	Story Name
Developer Stories	
KAD-D1	As a user, I would like the app to be cross platform (iOS and Android).
KAD-D2	As a developer, I would like user profile data to be safely and securely stored.
KAD-D3	As a developer, I would like user health / fitness data to be safely and securely stored.
KAD-D4	As a developer, I would like informative error messages to be shown both on the frontend of the app and in the developer console.
KAD-D5	As a developer, I would like the app to be properly documented with an informative README.
KAD-D6	As a user, I would like the app to have an intuitive and consistent user interface.
~~KAD-D7~~	~~As a developer, I would like the repository to have a CI/CD pipeline.~~
~~KAD-D8~~	~~As a developer, I would like a Node/Express backend.~~
~~KAD-D9~~	~~As a developer, I would like a Next.js frontend.~~
~~KAD-D10~~	~~As a developer, I would like to integrate the frontend and backend.~~
~~KAD-D11~~	~~As a developer, I would like a MongoDB database connected to the backend.~~

Accounts and Profiles	
~~KAD-5~~	~~As a user, I would like to create an account.~~
~~KAD-6~~	~~As a user, I would like to login and logout of my account.~~
~~KAD-8~~	~~As a user, I would like to be able to delete my account.~~
~~KAD-9~~	~~As a user, I would like to be able to reset my password~~
~~KAD-10~~	~~As a user, I would like to view my profile.~~
~~KAD-11~~	~~As a user, I would like to update my profile.~~
~~KAD-65~~	~~As a user, I would like to have and edit a username.~~
~~KAD-66~~	~~As a user, I would like to have and display a profile picture.~~
~~KAD-67~~	~~As a user, I would like to display a short bio about myself and my music taste.~~
Device and Music Connections	
~~KAD-12~~	~~As a user, I would like to link a music platform.~~
~~KAD-13~~	~~As a user, I would like to be able to update my music platform.~~
~~KAD-14~~	~~As a user, I would like to be able to remove a music platform.~~
~~KAD-15~~	~~As a user, I would like to see what music platforms are currently linked.~~
~~KAD-17~~	~~As a user, I would like to be able to update my default music and device preferences.~~
~~KAD-18~~	~~As a user, I would like to connect to my fitness device.~~
~~KAD-68~~	~~As a user, I would like to be able to remove my connected device.~~
~~KAD-20~~	~~As a user, I would like to view my connected devices.~~
~~KAD-21~~	~~As a user, I would like to be able to choose a default device.~~
~~KAD-22~~	~~As a user, I would like to be able to allow / disallow the app to access my fitness data.~~
~~KAD-23~~	~~As a user, I would like to be able to rename a connected device.~~

Song Selection / Playlist Preferences	
KAD-24	As a user, I would like to select the genre of the playlist.
KAD-16	As a user, I would like to create default preferences on genres, artists, etc. that affect playlist generation.
KAD-25	As a user, I would like to blacklist songs and artists from my playlists.
KAD-26	As a user, I would like to be able to create a preference between lyrical and instrumental songs.
KAD-27	As a user, I would like to choose whether explicit songs are added to the playlist or not.
KAD-28	As a user, I would like to set a lyric language preference.
KAD-29	As a user, I would like to set a minimum and maximum song length.
KAD-52	As a user, I would like to set minimum and maximum time limits on the playlist being generated.
Playlist / Mode Functionality	
KAD-38	As a user, I would like to be able to start playlist generation.
KAD-39	As a user, I would like to specify whether or not I want to save the playlist right away or allow Kadence to build a queue of songs to listen to first.
KAD-40	As a user, I would like to be able to provide feedback about the generated playlist.
KAD-41	As a user, I would like to delete a playlist after initial generation and automatically regenerate / reshuffle.
KAD-42	As a user, I would like to be able to select different playlist creation modes.
KAD-43	As a user, I would like to set / edit the short and long interval times in interval mode.
KAD-44	As a user, I would like to select ramp up or ramp down settings in the fitness mode.
KAD-45	As a user, I would like to change my default location to explore different local music scenes.
KAD-46	As a user, I would like to skip a song in a generated playlist / queue if I do not enjoy it.

KAD-47	As a user, I would like to blacklist additional songs if I strongly dislike them during playback.
KAD-48	As a user, I would like to save a playlist generated to either Spotify or Apple Music profiles after listening to it.
KAD-49	As a user, I would like to customize the name of the playlist before I save it.
KAD-51	As a user, I would like to have the ability to alter the playlist (remove songs) after it has been generated (if time permits).
Social	
KAD-53	As a user, I would like to view other people's profile pages.
KAD-54	As a user, I would like to hide my own profile page from other users by declaring my profile private.
KAD-55	As a user, I would like to ensure that all of my sensitive fitness information remains private from all users, including friends.
KAD-56	As a user, I would like other people to find and view my profile by declaring my profile public.
KAD-57	As a user, I would like to search for other users.
KAD-58	As a user, I would like to send friend requests to view private profiles.
KAD-59	As a user, I would like to view and listen to playlists that public users or friends generate.
KAD-61	As a user, I would like to view what song that a public user or friend is listening to in real-time (if time permits).
Activity Log (if time permits)	
KAD-62	As a user, I would like to see an activity log of all of my own actions.
KAD-63	As a user, I would like to see an activity log of new playlists my friends are generating and saving to their profiles.
KAD-64	As a user, I would like to see statistics regarding my listening history in each mode and about my playlists.