

Meta University Eng Project Plan Template

Fill in blanks (enclosed by brackets []) and remove red text as you work through writing your project plan. Your project plan should be a living document and can be changed as you progress through the internship. Make sure to work on this document together with your manager to get feedback, as well as ensuring your project meets the requirements and expectations in the [Project Guide](https://sites.google.com/meta.com/metau-intern/capstone-project/project-guide).

# [MOOD TUNE]

Intern**: Frank Dornberger Wember**

Intern Manager: Yunlong Zhang

Intern Director: [Name]

Peer(s): [Arpit Sood, Yidan Ni, Vraj Parikh]

GitHub Repository Link: [https://github.com/FrankWember]

## **Overview**

Mood-Based Music Recommendation Web App

* **Category:** Entertainment
* **Story:** [My project aims to build a web application that can recommend different music to its user based on the user's current mood and location data obtained from their Apple Watch. The app will provide an ideal audio experience tailored to the user's emotional state and physical environment.
* **Market**: The app is aimed at tech-savvy music enthusiasts, particularly youths, adults and professionals who use wearables and are looking for a seamless and personalized music experience. This can also be applicable to anyone who likes music.
* **Habit:** The users are likely to use the app daily, especially during commutes, workouts, leisure time and even during studying to get music recommendations that match their current mood and activity.
* **Scope:** The initial scope of my idea was a mood-based music recommendations, integration with Apple Watch for heart rate and location data, and the user’s Spotify for playlist creation. Out of my scope for now will be maybe dding advanced features like user-generated content or detailed analysis on music preferences.]

## **Product Spec**

* User account creation and login
* Apple Watch integration for heart rate and location data
* Spotify integration for music playback
* Mood detection and music recommendation algorithm
* Playlist creation and playback
* Save and share playlists.

### **User Roles**

* **User Roles:**
* **Listener:** A user who is seeking a personalized music recommendations based on their mood and location.
* **User Personas:**
* **Listener Persona 1:**
* **Name:** Emma Joyce
* **Origin: Cameroon**
* **Age:** 20
* **Tech Usage:** Heavy, primarily uses an iPhone and Apple Watch.
* **Motivation:** Emma is a student and goes to University of Yaounde so she has a pretty busy day from Classes to public transport to working out or running and she wants music that matches her mood during different times of her day.
* **Pain Points:** Finds it hard to create the perfect playlist for her varying moods.
* **Listener Persona 2:**
* **Name:** Franck Smith
* **Origin:** France
* **Age:** 32
* **Tech Usage:** Moderate, uses both smartphone and wearable devices.
* **Motivation:** Looking for music to enhance his workout sessions.
* **Pain Points:** Franck gets bored with repetitive music and seeks variety because he does not know how to find the ideal musical experience for his mode.

### **User Stories**

* As a listener, I want to create an account, so that I can save my preferences and playlists.
* As a listener, I want to be able to log in, so that I can access my personalized recommendations.
* As a listener, I want to connect to my Apple Watch, so that the app can gather my heart rate and location data.
* As a listener, I want the app to detect my mood based on my heart rate, so that I get music that matches my current state.
* As a listener, I want the app to recommend playlists based on my location, so that the music fits my environment.
* As a listener, I want to play music directly within the app, so that I can listen to my recommended playlists.
* As a listener, I want to save my favorite tracks, so that I can easily access them later.
* As a listener, I want to share my playlists with friends, so that they can also enjoy the music I like.
* As a listener, I want to view and manage my saved playlists, so that I can edit or delete them as needed.
* As a listener, I want to receive notifications about new playlist recommendations, so that I stay updated with fresh music.

### **Screen Archetypes**

A screenshot of a login screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a music player

Description automatically generated

### **Data Model**

[Describe the data you’re going to need to back your application. This can include database models (like tables), or external data you’ll require from some API.]

**User Model**:

"id": "integer",

"name": "string",

"email": "string",

"appleUserId": "string",

"spotifyUserId": "string",

"createdAt": "timestamp",

"updatedAt": "timestamp"

**Health Data Model**

"id": "integer",

"userId": "integer",

"heartRate": "integer",

"stepsCount": "integer",

"timestamp": "timestamp"

**Location Data Model**

"id": "integer",

"userId": "integer",

"latitude": "float",

"longitude": "float",

"timestamp": "timestamp"

}

**Spotify Data Model**

"id": "integer",

"userId": "integer",

"playlistId": "string",

"trackId": "string",

"trackName": "string",

"artistName": "string",

"timestamp": "timestamp"

**Music Recommendation Model**

"id": "integer",

"userId": "integer",

"recommendedTrackId": "string",

"recommendedTrackName": "string",

"artistName": "string",

"reason": "string",

"timestamp": "timestamp"

### **Server Endpoints**

**Endpoints**

1. **Authentication Endpoints**
   * GET /auth/apple: This will be use to redirect to Apple OAuth for authentication.
   * GET /auth/spotify: Same OAuth for Spotify
   * GET /auth/callback/apple:Will handle the callback from Apple OAuth.
   * GET /auth/callback/spotify: Will handle the callback from Spotify OAuth.
   * GET /auth/logout: Logs out the user by ending the session and/or clearing cookies.
2. **User Endpoints**
   * GET /users/:id - Retrieve the user’s profile.
   * PUT /users/:id - Update the user’s profile.
   * DELETE /users/:id - Delete the user’s profile.
3. **Music Data Endpoints**
   * GET /spotify/playlists - Retrieve user playlists.
   * GET /spotify/recommendations - Get music recommendations.
   * POST /spotify/recommendations - Save music recommendations.
4. **Real-Time Data Endpoints**
   * 1. GET /realtime/health - Fetch real-time health data.
     2. GET /realt
     3. ime/location - Fetch real-time location data.
5. **Miscellaneous Endpoints**
   * GET /dashboard - Fetch data for the dashboard.

Possibility of using ML

**Location Data Endpoints**

* + GET /location-data - Retrieve historical location data.
  + POST /location-data - Upload historical location data.

**Health Data Endpoints**

* + GET /health-data – Retrieve historical health data from watch.
  + POST /health-data - Upload historical health data received.

### Navigation

## Project Requirements

[Based on the [Project Guide](https://docs.google.com/document/d/1TvGTVGsH0b3HSVh_tRvQZDizWwBSQVCfRiS4sqMZY6Y/edit#heading=h.92pf0mt6mhlq), describe how your project is going to be fulfilling each of the base project requirements.]

## Technical Challenges

For your project, you should demonstrate that you can apply what you’ve learned so far and expand on that knowledge to write code and implement features that go beyond the scope of the projects you worked on during Code Path.

Based on the general idea and direction of your project requirements, your intern manager will create at least two (2) Technical Challenges for you. This section is all about explaining what they are and how you’re planning to tackle them - you’ll work together with your manager to fill it out.

### Technical Challenge #1 - [Name/Small Description]

#### What

What problem are you solving, and what parts go beyond what you learned in CodePath?

#### How

Explain in words how you’ll solve this problem.

You’re encouraged to expand on this section with pseudo-code, links to external frameworks, architecture / design diagrams, anything that you can use to explain this to others!

### Technical Challenge #2

#### What

#### How

### Database Integration

[Describe what you are using for database storage. For example, Parse, MongoDB, Sequelize, etc.]

### External APIs

[Describe at least one external API you’re using for your project. For example, Google Maps, Spoonacular, OpenWeather, etc.]

### Authentication

[Describe how user authentication is handled for your project, including logging in and signing up. Also describe any kind of cookie / session management you’re doing and how you’re implementing it, and how this affects navigation between different screens by the same user.]

### Visuals and Interactions

[Provide details on how your app is fulfilling the following UI craft requirements, and how these are technically accomplished.]

* Interesting Cursor Interaction
* UI Component with Custom Visual Styling
* Loading State

## Timeline

Project execution will start in Week 4 of MU. Based on the previously defined requirements, user stories and technical challenges, use the following table to scope out and plan a timeline for deliverables over Week 4 - 9. You can be as detailed as you need, ranging from simply mentioning the user stories, or dividing them into sub-tasks.

You are free to modify the table, add / remove rows or columns, whatever fits your style! The important thing here is that you focus and prioritize certain aspects of your project so you don’t get behind and are ready to deliver the MVP - remember your required features should be code complete before the end of Week 8, including both technical challenges!

We also encourage you to leverage project tracking tools such as GitHub Issues or Meta’s internal Tasks / GSD tooling to keep manage individual units of work.

|  |  |  |  |
| --- | --- | --- | --- |
| MU Week | Project Week | Focus | User Stories |
| 4 | 1 | Focus on the components that will serve as the skeleton of your project. You will probably be using most of what you learned in CodePath to set up things like the client and server repositories, initial routing, login / registration, creating a database with object models, etc. | Example:   * User can login * User can create an account * [Optional] User passwords are encrypted in the database for security |
| 5 | 2 | Week 5 and 6 should be where you focus on the specific requirements of your project. | Example:   * User can create / edit / delete posts * User can chat with other users in real-time (e.g. technical challenge) |
| 6 | 3 | By this point, you should be getting started with your technical challenges as well. |  |
| 7 | 4 | You should focus on finishing your MVP and core requirements. By this point, you should be done with at least one of your technical challenges. |  |
| 8 | 5 | Continue work on finishing touches and stretch goals for your MVP. By this point, your core functionality and both TAPs should all be in place. It is also a good point to start working on stretch goals that could further expand on the functionality (and technical complexity) of your project.  This week you also have to submit your self-review, make sure you allocate enough time for this alongside your final submission for your project! |  |
| 9 | 6 | It’s time to show others what you have built! Work on a presentation and demo that you will present to other interns to showcase your work. You are also free to continue polishing and expanding on your project! |  |
| 10 | 7 | For this week, we have a bunch of extra activities prepared to give you a quick dive of what it is to work at Meta. You will find activities around using internal tools and frameworks, and even committing code to our internal repositories. |  |