Software Engineering Report

GitHub link : https://github.com/swanjln/Moodify/

Introduction

Welcome to our Software Engineering project report, a project entitled Moodify. This report is a written record of our progress throughout the semester, leading up to the project as it stands today. You'll find a presentation of our project, details of the site design and a section on how we organized ourselves and divided up the tasks. Enjoy your reading!

Moodify presentation

Moodify

First of all, what is Moodify? Very simply, the name comes from the fusion of Spotify and Mood. The idea evolved over the course of six months, and finally took shape as we see it today.
Moodify is a site that will suggest music to the user, according to a selected genre and mood. As its name suggests, the site uses Spotify's API to suggest music that already exists on Spotify. The aim is to help users discover new music that matches their mood.

Why Spotify?

We chose Spotify for two reasons. Firstly, it's one of the most widely used streaming platforms, and it has a well-stocked catalog with over 80 million tracks. What's more, Spotify's API is very easy to access, and extremely well documented, as there's a whole site explaining how to use it with examples. Here's the link:
https://developer.spotify.com/documentation/web-api

Discontinued ideas

At first, we wanted to make a C# application, with a WPF interface. We quickly changed our minds and decided to make a website instead, which made it much easier for us to make API requests.
We also wanted to bring a social aspect to our project, and had thought of having the music suggestions be music listened to by the user's friends. Unfortunately, this turned out to be too complex for the API's authentication tokens.

Organisation

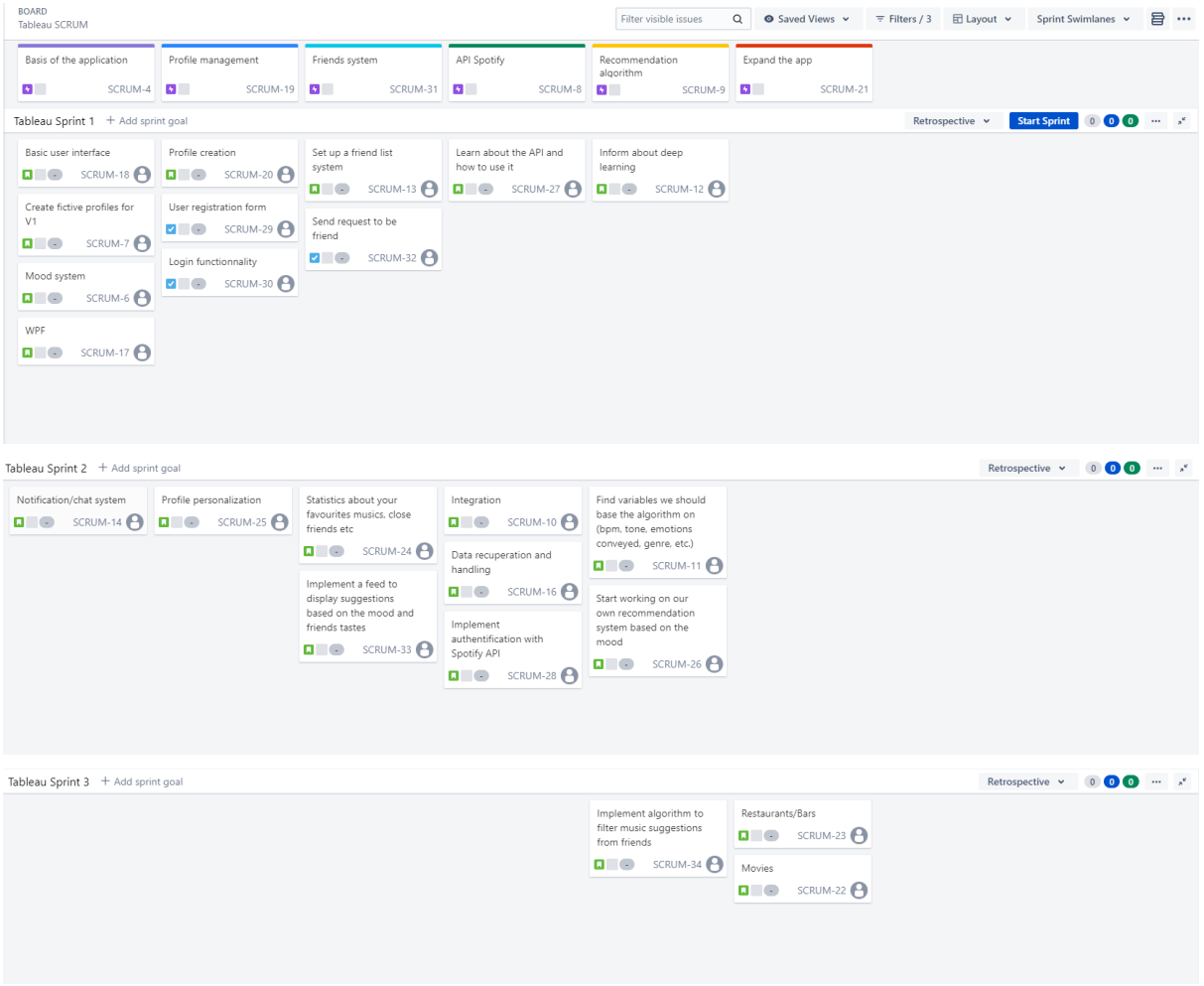
Tools Used

SCRUM

We used SCRUM sprints throughout the semester to make steady progress on our project. The tasks we had to do during these sprints were planned in advance.

Jira

Jira is a tool we discovered during the semester. It enabled us to define which tasks needed to be done, and in what order.



This is our Story Map. It's not up to date because, as previously mentioned, we had originally planned to make an application. However, only the "Basis of the Application" Epic was not taken into account, as the other Epics were similar for a site and an application.

GitHub

So that we could all work on the code on our own and on the various tasks we had to perform, we set up a GitHub.

Communication channels

We communicated with two media. Discord was our spearhead for the project over the long term. This is where we shared documents and discussed issues throughout the project. We also used a WhatsApp group to circulate the most important information.

Task allocation

The distribution was quite natural, and fairly balanced. We all took part in programming the site to varying degrees. Those who worked a little less on the technical side compensated by focusing on the organization (the Jira) and the various outputs, including this report.

Design details

Tools and languages used

JavaScript

The back-end of the site was built in JavaScript. This is where all requests to the Spotify API are managed. Some buttons are also procedurally generated in JavaScript, making them easy to modify if you wish to add additional moods or genres, for example.

HTML

This is where the site interface is managed.

CSS

CSS is used to improve the aesthetics of our buttons and the overall look of the site.

Code overview

API connection

```
// Get access token
const accessToken = await getAccessToken();
```

In order to connect to the spotify API, you need a "token" which identifies you and determines what you are authorized to request from the API.

Assigning values to parameters

```
// Define target values based on mood
let targetDanceabilityMin, targetDanceabilityMax,
targetEnergyMin, targetEnergyMax,
targetKey, targetLoudness,
targetPopularityMin, targetPopularityMax,
targetTempoMin, targetTempoMax,
targetValence;

switch (mood) {
    case 'happy':
        targetDanceabilityMin = 0.6;
        targetDanceabilityMax = 1;
        targetEnergyMin = 0.6;
        targetEnergyMax = 1;
        targetKey = 7;
        targetLoudness = 0.8;
        targetPopularityMin = 20;
        targetPopularityMax = 80;
        targetTempoMin = 110;
        targetTempoMax = 140;
        targetValence = 0.85;
        break;
```

In this first version of the code, we've determined some parameters ourselves, corresponding to the different moods we've chosen.

Request Url to Spotify

```
// Make a request to Spotify API to get recommended songs
const url = `https://api.spotify.com/v1/recommendations?limit=100&market=FR
&seed_genres=${seedGenres}
&target_danceability_min=${targetDanceabilityMin}
&target_danceability_max=${targetDanceabilityMax}
&target_energy_min=${targetEnergyMin}
&target_energy_max=${targetEnergyMax}
&target_key=${targetKey}
&target_loudness=${targetLoudness}
&target_popularity=${targetPopularityMin}
&target_tempo=${targetTempoMin}`;
```

The various parameters and the genre contained in seed_genres are used to send a recommendation request to spotify.

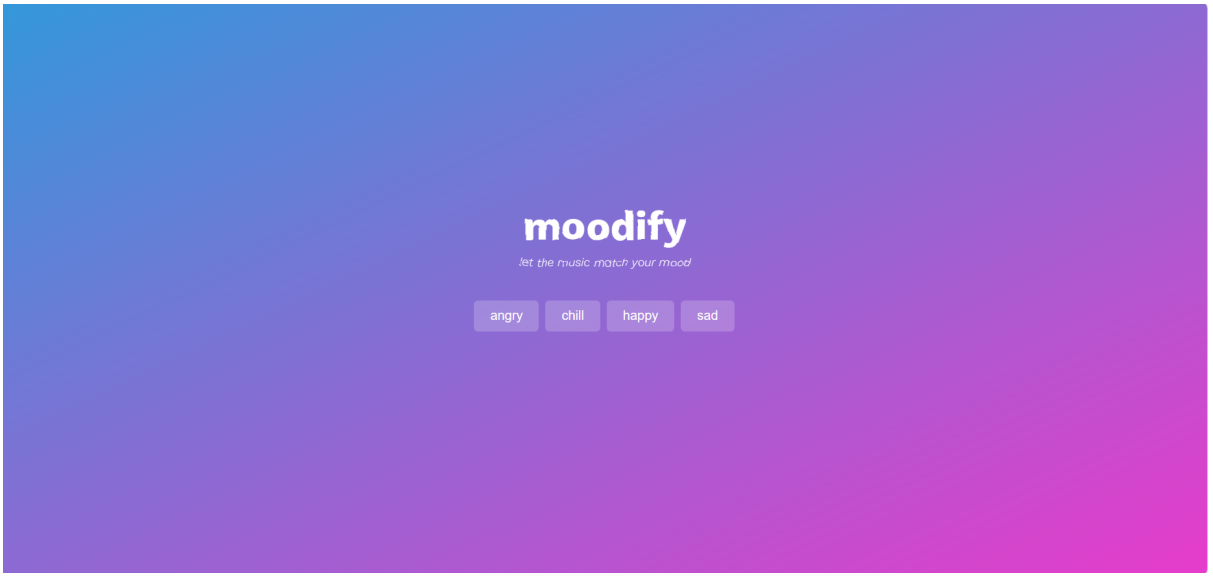
Moods and Genres

```
const moods = ['angry','chill','happy', 'sad'];
const genres = ['ambient', 'classical', 'country', 'electro', 'jazz', 'hip-hop', 'pop', 'rock', 'r-n-b', 'soul'];
```

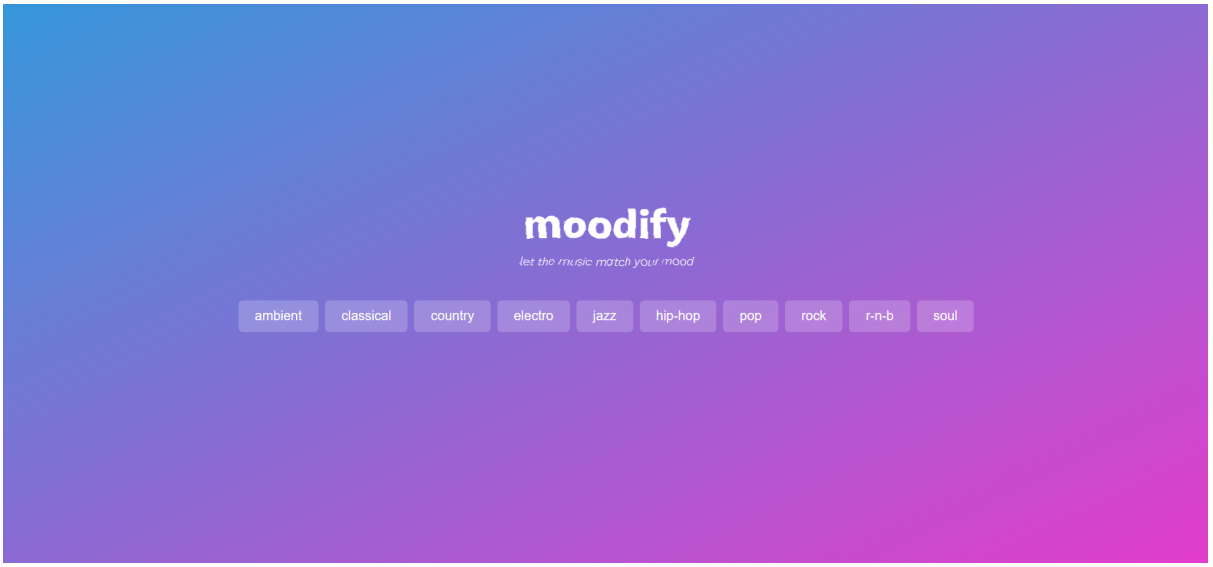
Our different moods and genders are instantiated once as constants.

Design

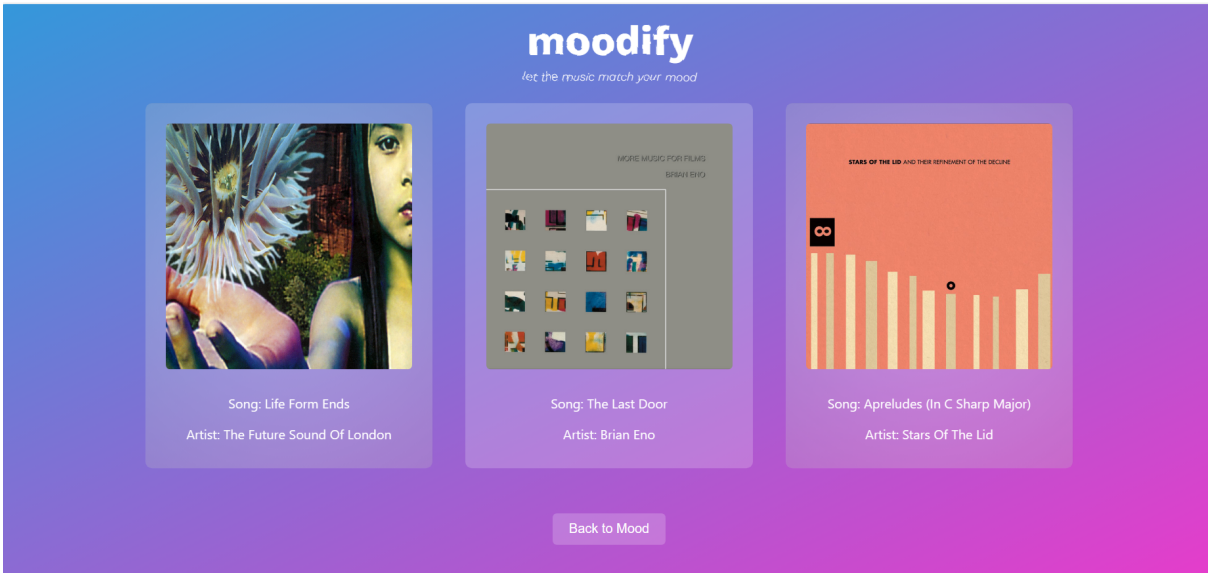
General Aspect



→ Choose a mood



→ Choose the genre



→ Music linked to Spotify

The visual direction aimed for a clean, uncluttered UI, minimizing buttons for a seamless experience. A soothing gradient of purple and blue dominated the background, accentuating the simplicity. White text ensured clarity against this backdrop. Tailored for music, the design emphasized minimalism to keep the focus on the application's core purpose.

Effects

Considerable effort was invested in crafting visually engaging effects, including hover effects, translations, shadows, glows, and animated squiggles. These elements were meticulously designed to enhance user interaction and elevate the visual appeal. Each effect aimed to add depth and interactivity, contributing to a more dynamic and captivating user experience.

Retrospective:

Initially, our team faced challenges due to the project's complexity. We had attempted to incorporate multiple features and functionalities into our application, which led to various issues and made the project too complex to work with effectively. However, we recognized this problem and made the decision to narrow down our focus to a specific feature, personalized music recommendations based on user mood.

By shifting our focus and pivoting to a website, we were able to overcome these challenges and make significant progress. This change allowed us to streamline our development process and deliver a more refined product for our prototype. Integrating the Spotify API into our site was a major success, as it provided us with easy access to Spotify's music data, simplifying the recommendation algorithm development.

Conclusion:

In conclusion, we are pleased with the progress we have made in developing our personalized music “recommendation” algorithm. The successful integration of the Spotify API into our site has significantly enhanced our capabilities and allowed us to create a valuable product for users. By focusing on a smaller scope and delivering a more polished prototype, we are confident that our website will meet the needs and expectations of our target audience.

Moving forward, our next steps involve expanding our service to include other categories of content, such as movies, to provide a more comprehensive recommendation experience. We also aim to optimize our recommendation algorithm further to improve its accuracy.

Additionally, we plan to enhance the design of our site, particularly the menu and user interface, to ensure a seamless and visually appealing user experience.

Furthermore, we intend to develop new functionalities, such as user profiles and a social aspect, to foster user engagement and interaction. Integration with additional APIs, such as Deezer, will also be explored to broaden the range of music options available to our users.

Overall, we are confident in the potential of our personalized music “recommendation” (not properly a recommendation algorithm) algorithm and look forward to further refining and enhancing our product to deliver an exceptional user experience.