CSCI-3308-Recitation-15-Team-02

METEOR MELODIES MAPPER

Presented by 2Kewl4Skewl

Who

- Emma Worthington
- Evan Rodenburg
- Francisco Villanueva
- Jared Press
- Jonathan Wu

Description

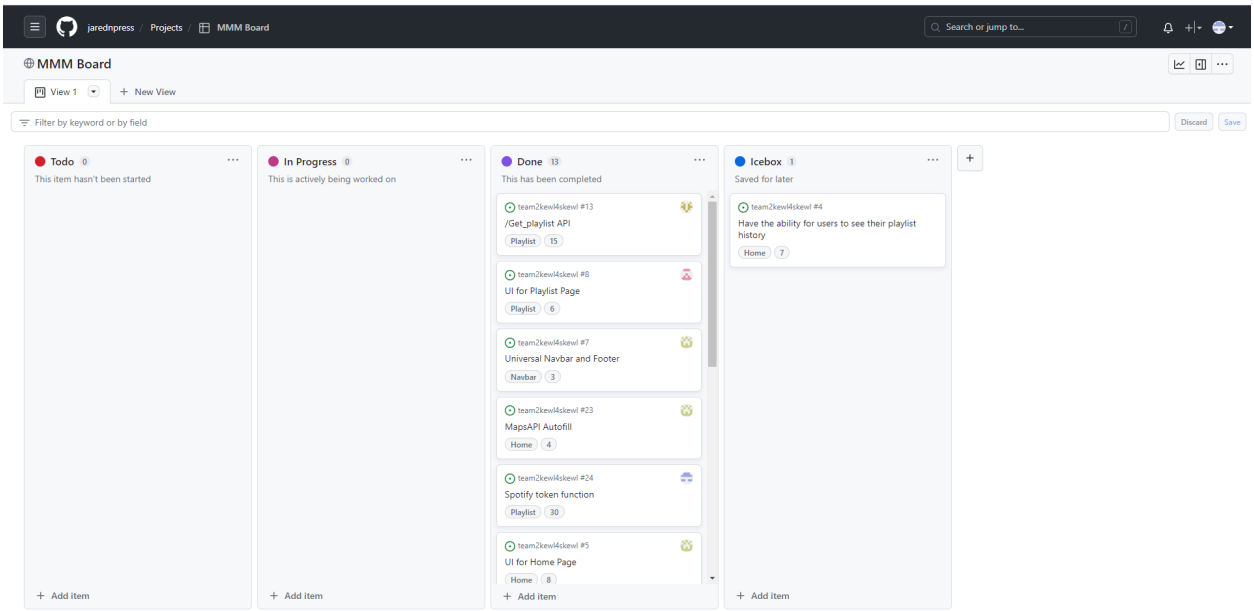
With our app, users can easily register and log in to start planning their perfect road trip. Once logged in, they can input their origin and destination. Using the Google Maps API, the application will determine the estimated duration of the route. Then, using this information, the app will also retrieve the weather at these locations with the Open Weather Maps API and generate a playlist matching the weather conditions by calling the Spotify API. This means that users can enjoy a playlist that is tailored to the weather conditions they will encounter on their journey. For example, if it's sunny, the playlist will feature upbeat, summer music. Whereas if it's rainy, the playlist will include softer, calmer music.

Meteor Melodies Mapper is designed to offer an unparalleled road trip experience, providing users with features that traditional road trip planning applications lack. Our focus is on providing a uniquely generated musical experience that encompasses a wide range of tastes. In this iteration of the applications users will be able to effortlessly generate playlists after registering for a free account. In the future, we hope the application could be expanded to recall historical playlists to relive the memory of a road trip. With Meteor Melodies Mapper, users can relax and enjoy their journey without the hassle of searching for the right music.

Project Tracker

The following link directs to the project board:

https://github.com/users/jarednpress/projects/1



CSCI-3308-Recitation-15-Team-02

Video Presentation

https://drive.google.com/file/d/1DuEL99zDhqVhDkiiGnp6giqQ_rHLfI4I/view?usp=share_link

VCS/GitHub Repository

The following link redirects to the remote repository: https://github.com/jarednpress/team2kewl4skewl.git

Contributions

Emma Worthington

I collaborated alongside my teammates to flesh out several frontend features. Additionally, I implemented a couple of asynchronous functions to enable API calls to Google Maps and Open Weather Maps. The first function passes in a city as an argument and returns the latitudinal and longitudinal coordinates. In the second function, both values, latitude and longitude, get passed in as arguments and return the corresponding weather at the specified coordinates at the time of the request. I also took the lead on other tasks such as preparing the template for the presentation slides; managing deadlines for the group; and other similar duties.

Evan Rodenburg

I assisted my teammates by managing the project board by writing user stories and acceptance criteria as well as helping to manage who was working on what parts of the project at any given time. Additionally, I helped by writing and bug fixing some of the on startup bug test cases of our website. My largest coding contribution was writing the frontend of the playlist page which takes information from the playlist API and displays every song from the playlist along with the information associated with a given song. I also worked on many smaller things associated with the project, whether that be small edits required for project related labs, or contributions to our presentation.

Francisco Villanueva

In contributing to the group project, I participated in supporting my team members by reviewing code; acquiring the necessary API tokens; digesting the API documentation; implementing the initial iteration of the functions to retrieve the Spotify API token and display the playlist endpoint; and preparing several documents. As a part of the code review process, I helped ensure that several features were enabled and displayed properly. An example of debugging, was when I fixed a bug wherein the redirect from the login page was broken. With the exception of Spotify, retrieving the API tokens and digesting the documentation was straightforward. For Spotify, my team and I had to use a *client ID* and *client secret* to retrieve an API key, which required the implementation of a function to get the token to enable features of the playlist endpoint. Finally, I handled documentation such as the *readme* file, this project report and other documentation and notes.

Jared Press

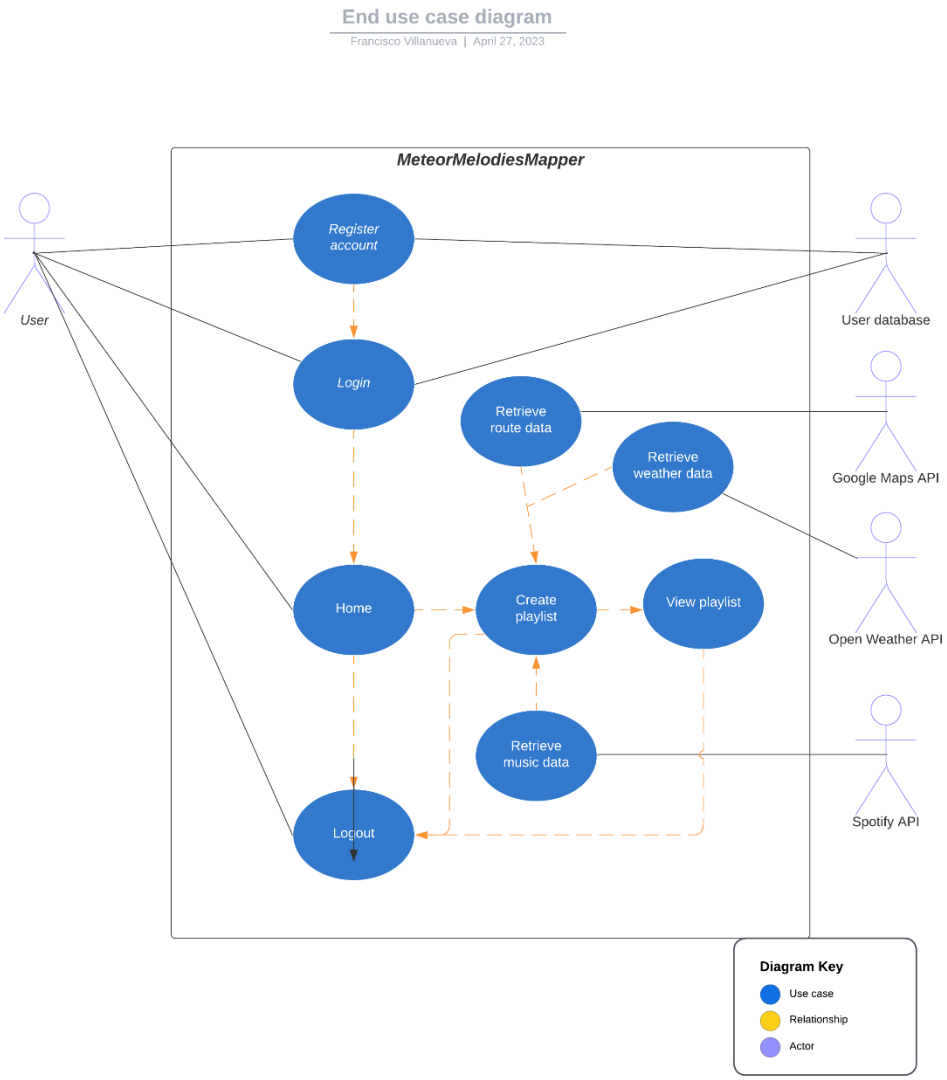
I was responsible for the Google Maps API integration, as well as styling, structuring, and other front-end tasks. For Google Maps, I implemented the places search so that we could get the location for our weather data and limited the search to only cities and only the US. I made the wireframe, and then used styling to make the vision from the drawing be accurately represented to the user. The visual functionality of the front-end. I was an active member of the team, fixing bugs here and there to keep the project going. Also, I wrote all of maps.js.

Jonathan Wu

I contributed to the project in 3 major ways: project management, backend engineering and integration. In terms of project management, I was responsible for organizing group meetings and finding solutions to scheduling conflicts. Additionally, I set goals and timelines for group work so that the project can be delivered on time and within scope. I engineered critical backend software such as the get playlist endpoint (including helper functions that were used in the endpoint), login endpoint, database design and writing test cases. When individual group members finished their work, I was responsible for merging code, resolving conflicts, and debugging integration problems. I supported my team members by helping explain technologies, methodologies and by debugging.

CSCI-3308-Recitation-15-Team-02

Use Case Diagram



Test Results

All tests were written into the code base to test various features of the application such as creating a user, logging into the application, etc. The application managed to successfully pass all current test cases. In future iterations of the application, a feature to recall historical playlists should be implemented and tested by checking if the application properly stores the list in a new table within the database. Otherwise, all other functionality seems to work appropriately.

Deployment

http://recitation-015-team-02.eastus.cloudapp.azure.com:3000/welcome