Emilia Skoins, Yizhou Zhang, Tianxin Wang, Xiangyu Sha, Rebecca Wright, Cameron McCurdy

Match Melody: A User Interaction-Based Music Recommendation Algorithm and Matching Platform

Product Description

Our project aims to bring people together over a shared interest in music. By letting the user 'swipe' on music they enjoy, our software will match them to other songs, albums and genres that might suit them and in turn match them with other users with similar tastes. We will use Spotify's available datasets and user data (when the user logs in and gives their permission) to make a connection between the music that the user currently listens to and what the software predicts will fall into that pattern. It will take the form of a website which will be more accessible to a variety of users, compared to creating a mobile application that may only work on one operating system.

Aims and Objectives

The aims of this project are:

1. To create a website that enables a personalised music recommendation system.
2. To provide interactive features that increase the user's experience of music on the site.

In order to achieve the above aims, the objectives listed below will be followed:

1. Extracting music data from existing open source databases, such as Spotify, and retrieving it to collect data on users' music preferences.
2. Implementing music playback functionality on the website and enabling users to interact with the currently playing song.
3. Implement music recommendation functionality using an algorithm created by a combination of content-based filtering and collaborative filtering.
4. Implementation of personalised music recommendations based on users' music preferences and historical interaction data.
5. Perform security and reliability tests on the website.
6. Evaluate the accuracy of the music recommendation system and determine its performance through testing and evaluation.

Key Literature and Background Reading

Spotify allows developers to use tools they have already created, such as interactive embedded music and Spotify Connect using documentation they provide on their platform. Many other websites utilise Spotify user data and analyse the results to provide information on the listeners habits, for example Stats for Spotify, How Bad is Your Spotify and Instafest. The Spotify Web API can return on request metadata about music artists, albums, songs and genre data directly from the Spotify Data Catalogue. The Spotify Web API can also provide access to user related data with user authorisation, such as playlists or music that the user saves in their library. All access to data is selectively authorised by the user.

Development and Implementation Summary

The project will exist in the form of a website. For front-end development, we will use a combination of HTML, CSS, and JavaScript. We chose these languages because they are the standard front-end languages, and we require JavaScript for integrating the Spotify API into the site and making API calls.

On the server-side, we will run a MySQL database to store user data as well as that retrieved from Spotify. We have chosen to use Python for our server-side development, such as interfacing with the database and the algorithms which deliver content to the user.

One of the main features of the site will be 'smart' song recommendations. We will create an algorithm using a combination of content-based filtering, and collaborative filtering. The algorithm will use user data to find new songs with similar characteristics to their preferences. The other main feature of the site is the social media style interaction between users. User activity in the discovery section will be stored in the database and an algorithm will retrieve and display some of this data in the social section.

Team members will focus on a specific area of development to increase productivity and allow for better organisation of the system components. All project files will be stored in cloud storage during the development process. During the early stages of development, components of the whole system can be run locally (such as the server-side components). The implementation of the project will require a dedicated machine to host the website.

Data Sources

1. **Personal Information of Users**
   Personal information will be collected when users register for a new account. The data will be used to establish a unique profile for each user in the background of our website. Meanwhile, it will provide credentials for the users to log in.

2. **Usage Information of Users**
   Usage information, such as songs the users prefer and whether they like the songs that our website recommends to them, will be collected while users are using the website. This data will be used to help the website recommend more songs that are in tune with the users' preferences.

3. **Music Resource of Spotify**
   We will utilise the open API of Spotify for developers. The music that our website will recommend to users and all song information will be derived from Spotify. The song information will be analysed and divided into different groups so that our website can decide which songs may be to the users' liking.

4. **Evaluation Questionnaires Data**
   The data will be collected by asking users to answer questionnaires to evaluate whether the music recommended by our website is to their liking. This will help to evaluate the results of our website.

Confidentiality Assurance:

- The website needs to be created and used in accordance with the GDPR.
- The database that stores the data should be well-designed and well-maintained.
- Before users register we will display the relevant user notice contract to users, listing the need to obtain personal information data and usage record data, how the website will collect this data and what the data will be used for. Only with users' knowledge and consent can users continue to register and use our website (Voigt, 2017).
- Before obtaining questionnaire survey data, users will be informed of the purpose of the data and will be asked whether they are willing to participate in the survey. Data will be collected only with the consent of users.
- The data will be securely stored in the background and will not be released to the public.
- All website developers need to follow a code of ethics and keep data confidential.

Testing and Evaluation

During software development, we will test our code using the unit testing method of white box testing. After all the parts are completed, we need to integrate them and perform system testing using black box testing. After the system is developed, it is important to use proper testing methods to evaluate the performance of our music recommendation system. There are various evaluation metrics that can be used including accuracy, variety and novelty.
To assess the accuracy, we can split the data into training and test sets using some data splitting strategies. We train the model using the training set and then calculate the accuracy of the data in the test set. The data we use here can be obtained by performing data processing on the playcounts (the number of listening

records) (Chou et al. 2015). In addition to accuracy, we can also evaluate our system in other aspects such as the variety and novelty of the recommended songs, with the help of some mathematical models (Chou et al. 2015). Hence we can gain more insights into the performance of the system, which makes our evaluation more comprehensive. In the meantime, we can optimise the system by trying different models and tuning parameters to achieve the best results.

However, it will be difficult to assess the true success of our website without receiving feedback from users. Thus, we can design a questionnaire to obtain users' opinions and preferences. A series of questions can include the number of recommended songs added to the playlist by the user, how often the user uses the site, how easy the site is to use, user satisfaction, etc. We confirm that all testing will be performed ethically.

Ethical Considerations

We are developing a software that processes data collected from external sources and will have outside participants to test and give feedback on our product. Throughout the project we will uphold the following ethical guidelines:

- All data presented in our project is genuine and our own work.
- All external data will be obtained legally and with the correct permissions.
- The project will be carried out in the UK, with only UK participants.
- Any participants involved in the project will be aware of our research and will not be offered any financial incentives.

Collecting data for our project will involve obtaining music data from existing open-source databases to retrieve data on users' music preferences. To do this we will be using the Spotify Web API (developer.spotify.com, n.d.), which requires us to first obtain authorisation from the user by asking permission to access their protected resources such as playlists. We can only access the user's Spotify account if they explicitly give us their permission, which keeps in line with our ethical guidelines.

We plan to run a questionnaire to evaluate our software performance. Any data that is obtained from participants who test our software will be protected by storing it securely on a university file store and then deleting it at the end of the project. Participants will be informed that this research is for a University of Liverpool project and that any data obtained will be anonymised to protect their identity. We will ensure to only involve participants who have given us formal permission to use their results.

BCS Project Criteria

As our degree programme is accredited by BCS, the Chartered Institute for IT, the software project we create is required to meet specific outcomes.

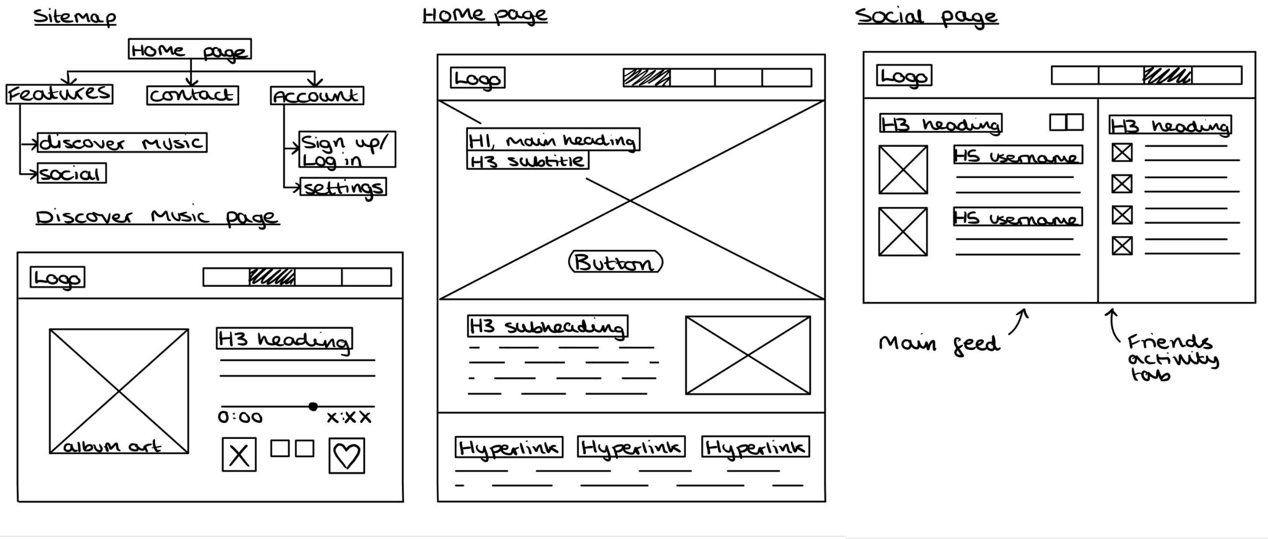
Apply practical and analytical skills acquired through previous university modules, including self-management and self-evaluation.

Throughout the two years spent studying at the University of Liverpool there have been many modules dedicated to coding such as COMP101 (Python) and COMP207 (MySQL). The team members will draw on their knowledge from these modules to complete the software project successfully. From COMP201, we learned the basics of software engineering and project management which taught us the important skill of self-management. Each team member needs to be responsible for their own part and ensure that tasks are completed within certain time constraints.

Union of information and practices to create a quality project as a team.

It is imperative that we work together well as a team and communicate effectively with each other. By sharing ideas and research, the team members can learn from each other and become more knowledgeable and skilled individuals which will help when doing our individual tasks for the project. If a team member is struggling with a particular task assigned to them, they should feel confident in being able to ask other team members for assistance.

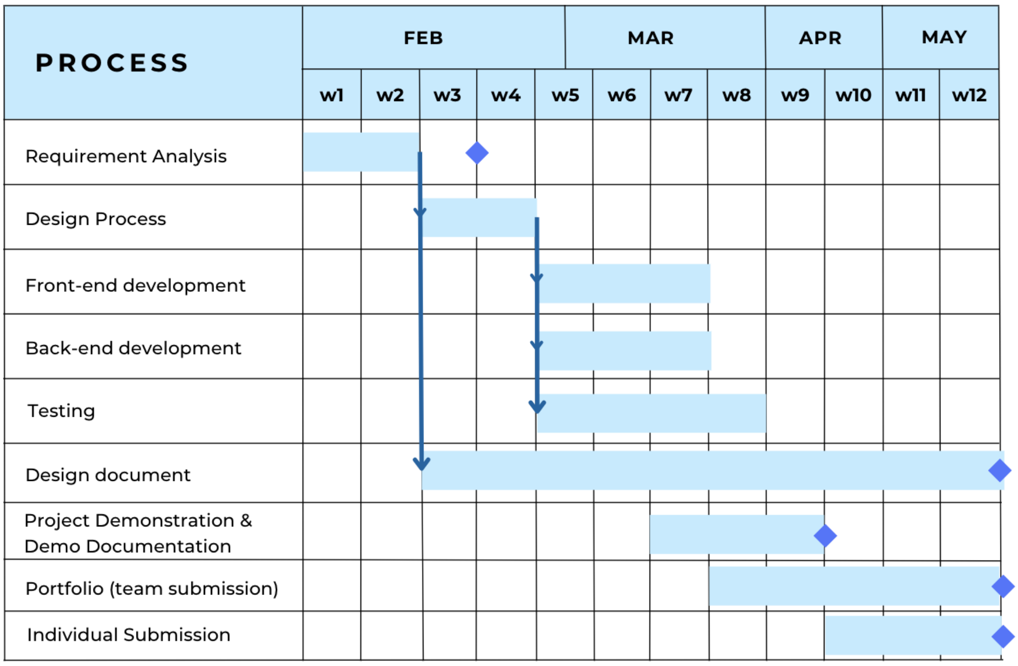
UI/UX Mockup



- The ‘discover music’ page has a simple user interaction to skip or like the current song. The design encourages quick discovery of new songs. The user will also be able to share and pause the current playing song.
- The home page will have a minimalist design, as its job is to point new users towards connecting to their Spotify and registering. It will have some brief information about the purpose of the site.
- The social page allows the user to share the music they have discovered, as well as look at recently liked songs from users they follow.

Project Plan

We use the Gantt chart, shown below, to display our whole project plan.



Risks & Contingency Plan

Risks	**Contingencies**	**Likelihood**	**Impact**
Serious computer hardware problems occurred during the project.	Back up data on multiple devices in a timely manner or use a platform that can share data in real-time.	Low	This will have a serious impact on the project, resulting in lost data that cannot be recovered and projects that cannot be delivered on time.
The tools used are no longer available.	Find and use alternative tools with similar functionality.	Low	The serious impact on project progression could lead to data loss and the risk that the project could be started from scratch.
Members are not familiar with the software to be used.	Members spend additional time learning.	Medium	The impact on the project is moderate and the planned completion time will be extended.
Reusable software components have functional defects.	Recode the component to fill in its functional gaps.	Medium	The impact on the project is more severe and the re-coding may also lead to bugs in other components and extended planning times.
The completion time of the project was underestimated.	Fully consider the possible risks faced and adopt the maximum time that the project may require to anticipate the project completion time.	High	This impact is moderate for the whole project and may result in a late arrival at the final deadline.
The time-consuming repair of software defects has been underestimated.	Avoid unnecessary errors by using good programming habits during the coding process.	High	The impact on the project is tolerable.

References

Chang, E. (2021). *Building a Song Recommendation System with Spotify*. [online] Medium. Available at: https://towardsdatascience.com/part-iii-building-a-song-recommendation-system-with-spotify-cf76b52705e7.

Chou, S.-Y., Yang, Y.-H. & Lin, Y.-C. (2015), Evaluating music recommendation in a real-world setting: On data splitting and evaluation metrics, *in* ‘2015 IEEE International Conference on Multimedia and Expo (ICME)’, pp. 1–6.

developer.spotify.com. (n.d.). *Authorization | Spotify for Developers*. [online] Available at: https://developer.spotify.com/documentation/general/guides/authorization/.

Voigt, P. and von dem Bussche, A. (2017). *The EU General Data Protection Regulation (GDPR)*. [online] Cham: Springer International Publishing. doi:https://doi.org/10.1007/978-3-319-57959-7.

developer.spotify.com. (n.d.). *Documentation | Spotify for Developers*. [online] Available at: https://developer.spotify.com/documentation/.