Software Requirements and Design Document

For

Group 4

Version 2.0

Authors:
Ranzley B
Lucas R
Nicholas M
Nicholas J
Veronica W

1. Overview (5 points)

Utilizing Spotify's web API to produce a collage of albums and artists based on the end user's Spotify listening activity. By allowing a more personalized experience for users we intend to allow specific time frames to be inputted to display users' music history. Presenting this in a visually pleasing collage will allow users to establish their "music moods" or stats that will be broken down into various categories for better viewing; The visual element is a key part of what we aim to develop.

2. Functional Requirements (10 points)

*List the **functional requirements** in sentences identified by numbers and for each requirement state if it is of high, medium, or low priority. Each functional requirement is something that the system shall do. Include all the details required such that there can be no misinterpretations of the requirements when read. Be very specific about what the system needs to do (not how, just what). You may provide a brief design rationale for any requirement which you feel requires explanation for how and/or why the requirement was derived.*

Completed - High Priority

1. API Integration: Connect website application to Spotify's API, enabling access to its data retrieval features. This will allow us to begin implementation of the collage-generation button.

2. User Login Authentication: Allow users' to login to Spotify account using username and password. Upon successful login, users will have access to the seasonsFM personalized homepage. Upon an unsuccessful login, the screen will display an error message along with the login page.

Completed - Medium Priority

1. Drop-Down Menu: Allow users' to input a timestamp that will display their personalized collage for that designated window frame. Secondary drop-down option will allow users to filter collage for genres or artists that will be displayed in alphabetical order.

Todo - High Priority

1. Collage Generation: Generate a collage of artists or tracks based on a user's listening activity over the selected timeframe. This represents the core functionality of the application.

Todo - Medium Priority

1. Drop-Down Menu Updates: Add a filter for the desired time period from which to pull listening data. Spotify's API supports pulls based on the last four weeks, last six months, and account life-time.

2. Box Physics: Create an interactive set of boxes using the collage elements. This is still up in the air, but we intend to implement a physics system on the boxes.

3. Personalized Background: Depending on the current season (Winter, Spring, Summer, Fall), background will change to a different gradient/image.

Todo - Low Priority

1. Mood Collage: Collect listening data from the user's Spotify account that will be used to create a defaulted collage of music taste for the month. The collage, in chronological order, will display the picture of the mostly commonly listened to artist/album of that day. This wide range of visuals will create a “mood” calendar of users music taste that can be filtered through using the Menu Feature.

3. Non-functional Requirements (10 points)

*List the **non-functional requirements** of the system (any requirement referring to a property of the system, such as security, safety, software quality, performance, reliability, etc.) You may provide a brief rationale for any requirement which you feel requires explanation as to how and/or why the requirement was derived.*

Security:

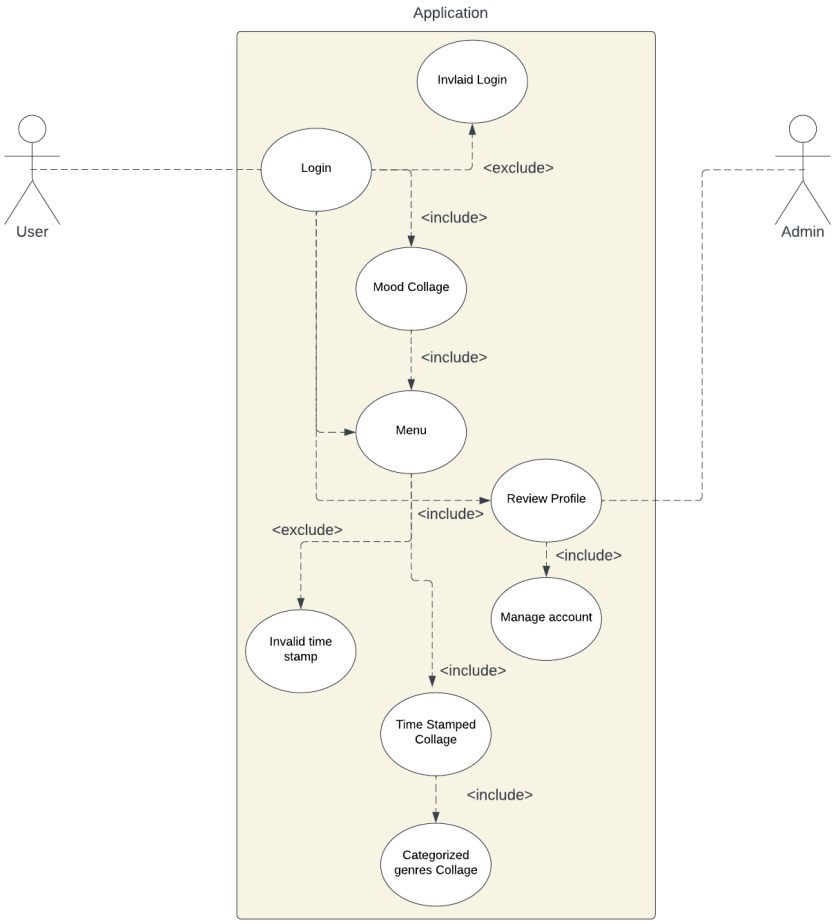
- Spotify API: Only Spotify users will be able to access the main features. The API also deals with the Spotify login authentication, preventing Spotify account compromisation.
- Role-Based Access:
    - Regular users will only be able to access the create account and login page, as well as the mood collage generation features.
    - Admin role will be able to access the above features, along with an account management page that has the ability to add/remove accounts, edit account information, and edit site features.

Performance: Application should be able to handle simultaneous activity between users without degradation (6+ users at a time). Expected response time between application interactions are between ~100-500 milliseconds

Usability: Application should be easy to navigate and understand without any extensive information. This includes having panels and designs that are legible and consistent amongst all users. In addition, having a straightforward and simplistic interactive system is one of our main focuses. We aim to have a visually-appealing color scheme that is complementary and attracts users to the personalized platform, followed by error prevention and feedback messages that will aid users on invalid input.

Compatibility: Ensuring the application is compatible with at least two web browsers and displays the same consistency across platforms. Also ensuring the application runs across multiple operating systems like Windows and MacOS, along with mobile OS’ like Apple’s iOS and Android’s Linux OS.

4. Use Case Diagram (10 points)



Textual Descriptions

Use Case Name: Login

Actors: User

Description: Allows user to login to their Spotify account which allows the application to gain access of music data

Flow of Events:

- User inputs valid Spotify Account information
- System retrieves data and sends user to home page
- User gains access to music data displayed in a designed mood collage

Alternate Flows: User is directed to the Invalid Login Use Case

Use Case Name:Invalid Login

Actors: User

Description: Displays to user that login information is invalid and they must re-enter valid account information

Use Case Name: Mood Collage

Actors: User

Pre-condition: User must have successfully logined

Description: Defaulted mood collage will be created displaying top artist & songs in organized aesthetically pleasing boxes

Use Case Name: Menu

Pre-condition: User must have successfully logined

Actors: User

Flow of Events:

- User inputs valid time stamps in drop down menu
- User inputs valid collage scale
- System retrieves data and modifies the mood collage to the users inputs
- Modified mood collage is displayed on home page

Alternate Flows: User inputs invalid time stamp directing them to an invalid error marked in the drop down menu (invalid time stamp use case).

Description: Drop Down Menu allows users to modify the mood collages dates and scale. Giving users customiablity on how much music data they want and how they want it displayed

Use Case Name: Invalid time stamp

Actors: User

Description: If users input an invalid input for the date in the drop down menu, it will allow users to input an valid date

Use Case Name: Review Profile

Actors: User, Admin

Pre-condition: User must have successfully logined

Description: Allows users to review Spotify account information including their emails, their username, etc

Use Case Name: Manage Profile

Actors: User, Admin

Pre-condition: User must have successfully logined

Description: Allows users to make light changes to their account, like changing their username

Use Case Name:Time Stamped Collage

Actors: User

Pre-condition: User must have successfully logined

Descrption: Depending on the menu options, users will be shown a collage on the time stamps inputted

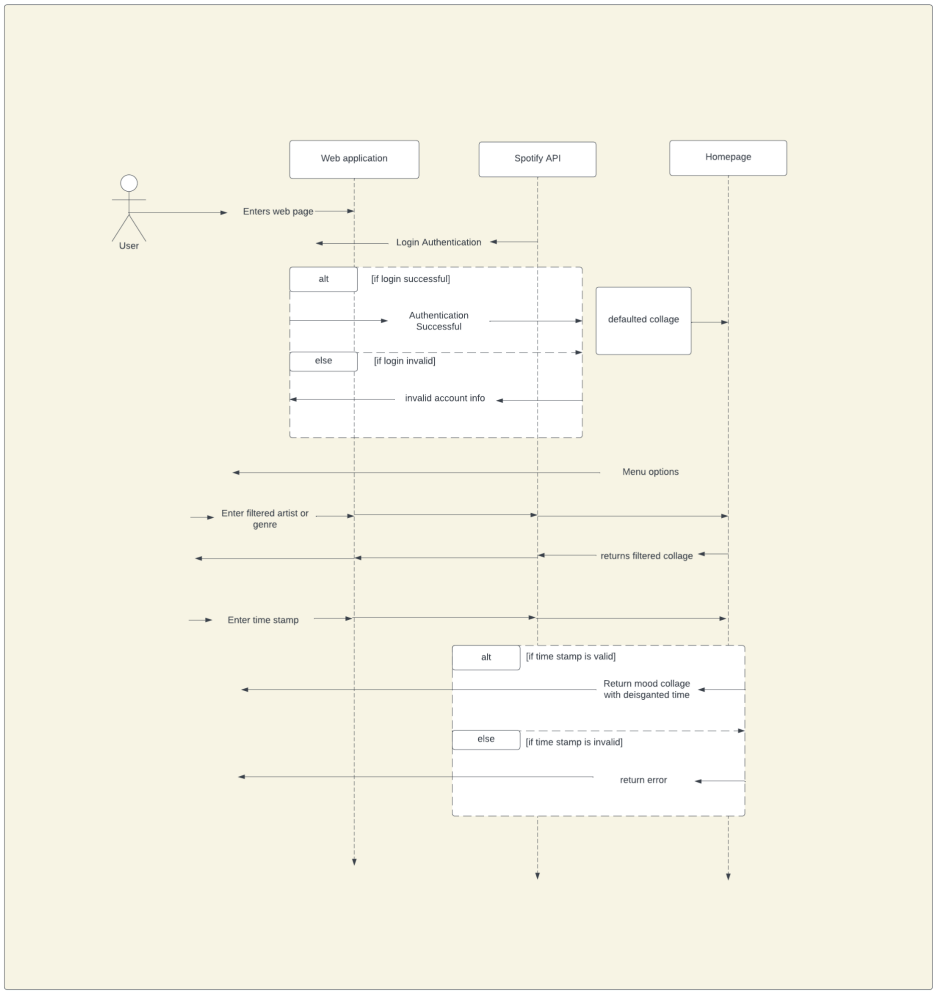
Use Case Name:Catergorized Genre Collage

Actors: User

Pre-condition: User must have successfully logined

Description:Depending on the menu options, users will be shown a collage with only a specific genre.

5. Class Diagram and/or Sequence Diagrams (15 points)



6. Operating Environment (5 points)

Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.

The primary environment will be a desktop setting with added support for mobile devices. The application should be compatible with all browsers including but not limited to: Opera, Edge, Safari, Chrome, and Firefox, assuming javascript is enabled.

7. Assumptions and Dependencies (5 points)

List any assumed factors (as opposed to known facts) that could affect the requirements stated in this document. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.

Assumptions:

- User has a mobile or desktop device connected to the internet
- User has Javascript enabled on their browser
- Spotify API services are up and running
- User already has a Spotify account

Dependencies:

- Spotify API Uptime: Since the backbone of our features uses user data gathered from the API, uptime of the API is important.
- Bootstrap: Bootstrap assists with dynamic and responsive web pages. It also helps with window resizing, streamlining mobile support.
- JQuery: It will be used alongside JavaScript to ease the coding process
- Matter.js: It will be used to implement the physics engine on the boxes for the music collage.

Testing