Project Charter: MusiConnect

## Project Overview:

The goal of this project is to develop a mobile application that enables users to synchronize multiple Spotify accounts, allowing them to listen to the same music simultaneously and share their music queue. The application will allow users to create groups and join the synchronization and queue sharing at any time.

## Stakeholders:

**Project Manager:** Davide Giovanetti

**Development team**

**End-user**

## Objectives:

* Develop a user-friendly mobile application that integrates with the Spotify API to synchronize multiple Spotify accounts.
* Allow users to sign up and log in using their existing Spotify credentials.
* Enable users to access their music libraries and playback controls.
* Enable users to create groups with an unlimited number of participants and invite others to join.
* Allow users to join and leave the synchronization and queue sharing at any time.
* Develop a synchronization mechanism that ensures all group members' Spotify accounts play the same music at the same time.
* Enable real-time updates of playback status across all connected devices.
* Implement a queue-sharing feature that allows users to add songs to a shared playlist.
* Enable all group members to see and contribute to the shared queue.
* Ensure the privacy and security of user accounts and data.

## Deliverables:

* A mobile application compatible with iOS and Android platforms.
* User registration and authentication system.
* Group creation and management functionality.
* Synchronization mechanism for simultaneous playback.
* Queue sharing feature with real-time updates.
* Privacy and security measures.
* Comprehensive testing and quality assurance.

## Precondition:

* The app will rely on the availability and stability of the Spotify API for seamless integration with users' Spotify accounts.
* Users are already registered on Spotify
* Users will have a reliable internet connection to ensure real-time synchronization and queue sharing.

## Input:

* Spotify account authorization code (given at login time).
* Group creation and management requests.
* Song selection and queue management requests.
* Playback control commands (play, pause, skip, etc.)

## Output:

* Synchronized playback of music across multiple Spotify accounts
* Real-time updates of playback status
* Shared queue displaying the songs added by all group members
* Notifications and alerts for group invitations, synchronization updates, etc.

## Historical Variables:

The only historical variables handled by MusiConnect are the joined groups with their song queue.

Music preferences along with every other account information are handled by Spotify services.

## Scale Factors:

* **User Growth:** As the popularity of the MusiConnect application increases, the number of registered users and active groups may grow significantly. The system should be designed to handle a large number of concurrent users and group interactions without compromising performance.
* **Group Size:** The application should support groups with an unlimited number of participants. The scalability of the synchronization and queue-sharing mechanisms should be optimized to ensure smooth playback and real-time updates, regardless of the group size.

## Project Risks:

Technical challenges may arise when integrating with the Spotify API, which could impact the synchronization and queue-sharing functionality.

The project may face delays if the Spotify API undergoes significant changes or disruptions.