

Milestone 1 - Project Proposal

Team Information

Team Number: 024_2

Team Name: PlaySyncers

Team Members:

- Gaukas Wang
- Logan O'Brien
- Kaitlin McConnell
- Erica Lowrey

Application Information

Application Name: PlaySync

Application Description:

PlaySync, as proposed, is a software tool to migrate/synchronize playlists across multiple music streaming platforms. These platforms may include but are not limited to, Spotify, Apple Music, Pandora, Amazon Music, and YouTube Music. The tool will prompt users to log in to their respective platform accounts and specify which platform they wish to migrate from and which to migrate to. It will then ask the user to specify which playlists they wish to migrate. It will also allow users to view the contents of each playlist before transferring.

Additionally, the software will allow users to sync their playlists across platforms and give them the option to do a one-time sync or have their playlists sync automatically during a given time interval whenever changes to their playlist are made. Users will also be able to share playlists with other users.

Vision Statement

For music streaming service users who use more than one platform or need to switch to a new platform, PlaySync offers the seamless playlist migration experience.

Version Control

We will use GitHub for version control of this project.

Link to the repository: https://github.com/CSCI-3308-CU-Boulder/3308SP21_024_2

This repository will include:

- Team Meeting Notes/Logs (with TA)
- Milestone Submissions (like this one, but in PDF)
- All codework or other design components related to the project.

Development Method

The software development methodology of this project should follow an agile-based hybrid methodology.

Link to the project dashboard:

<https://csci-3308-spring21-024-2.atlassian.net/jira/software/projects/PLS/boards/1>

Communication and Meeting Plan

The communication will be fully-remote for safety purposes.

The team will hold 2 All-Hands meetings every week, online via Zoom(Default)/Google Meet(Backup):

- Thursday 4 PM - 5 PM (For the first 10 minutes, meet with TA)
- Friday 5:10 PM - 6:10 PM

In addition to all-hands meetings, the team will communicate via Slack for all other updates. We welcome our TA/instructor to join our Slack Channel:

https://join.slack.com/t/3308sp210242/shared_invite/zt-1kht802t-5CYsnlj5PYsbNdu1r4xgCg

Proposed Architecture Plan

Frontend:

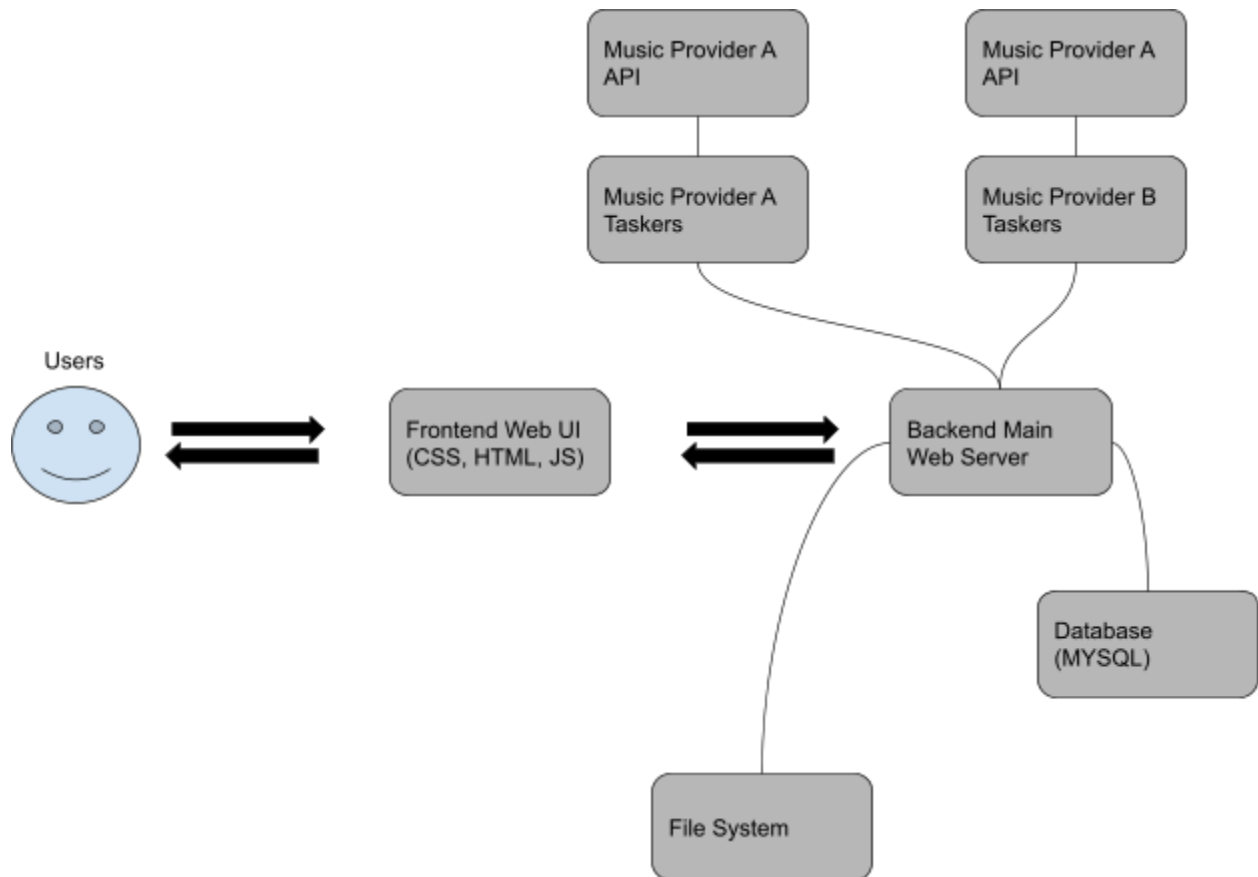
- A website providing Web UI (CSS, HTML, JS)

Backend:

- 1 Main server program to read/write database and process requests.
- Several taskers (GetPlaylist, GetSongInfo, SearchSong, CreatePlaylist, EditPlaylist, etc) for every platform supported, build in Python/NodeJS/C++.
- A task coordinator who communicates between taskers and the main server.

Database:

- A general-purpose database (MySQL?)



Use Case Diagram

Extremely basic use case diagram. Feel free to edit. I gave “First time user” essentially the same use cases as the normal “user” to try to visualize the incorporation of the seamless log-in for both new and existing users, however, this may not be the best way to portray that idea.

