Team Number	105-4
Team Name	Team 3308
Team Members	Max Macaluso, Keyu Wu, Zongyi Huang, Jiahao Wang, Peter Martin, Robert Sarno
Application Name	Monosic
Application Description	Our program will merge different playlists across platform. It helps mainly with playlist migration.
Vision Statement	General: To make listening to listening as easy as possible. Specifically: To merge all your music in one place, so users can listen seamlessly across platforms.
Version Control	Three folders: (which are submodules all under one repository, named Monosic) MileStone Submissions Meeting Log Code - Back end (committed in separate branch) - Front end (committed in separate branch)
Development Method	Our team will mainly be using agile methodology. We are doing this system because we plan to build out simple features first, and then we will build upon that iteratively. The simple features to focus on are copying playlists, starting from ones with easier web services like Spotify. Then making sure we are able to merge two web services into one playlist smoothly with no issues. Once we have the merging of two web services we will focus on adding the wider array of web services. We will continue iterating upon this

	until we are satisfied, and test out as we go (that's why we use agile). The backend will be built out simultaneously. All we will really need for backend is a login page which links you up to all your associated music accounts, but through the agile process, we can expand on this as we see fit.
Communication Plan	Our team plans to communicate using the application, Discord. Discord is an application available for smartphones, tablets, as well as desktops and laptops. It is a revolutionary platform that allows for voice chat as well as direct messaging. Additionally, you can have different channels of communications within each group chat, for specific topics. We are choosing this because it allows us to text regardless of device, and it has great capabilities.
Proposed Architecture Plan	Front end: - HTML - CSS - Javascript - Angular? Back end: - Node.js - SQL? Integration layer: TBD This whole application will be a web app hosted by heroku. It will be accessible from mobile or desktop. We will have a front end (what user sees) and back end (database and webserver). The webserver will have a login that should be associated with your logins to other music services. This is all the backend needs to do. The frontend and backend will communicate with an integration layer, so the frontend now has access to all of the person's accounts. The frontend will have a pretty UI that allows the user to clearly see which playlist they are migrating from what, what they are merging, etc. We will use various web services, like amazon web services, spotify web services, apple music web services, etc. to be able to copy playlists across platform. That's all that will need to happen. If we finish this, a stretch goal would be having a hybrid playlist, which can stream stuff from various platforms.

Meeting Plan	Time: Tuesday and Sunday after 5:00 PM Location: CSEL or Library study rooms Mode: In person, or voice chat via Discord.