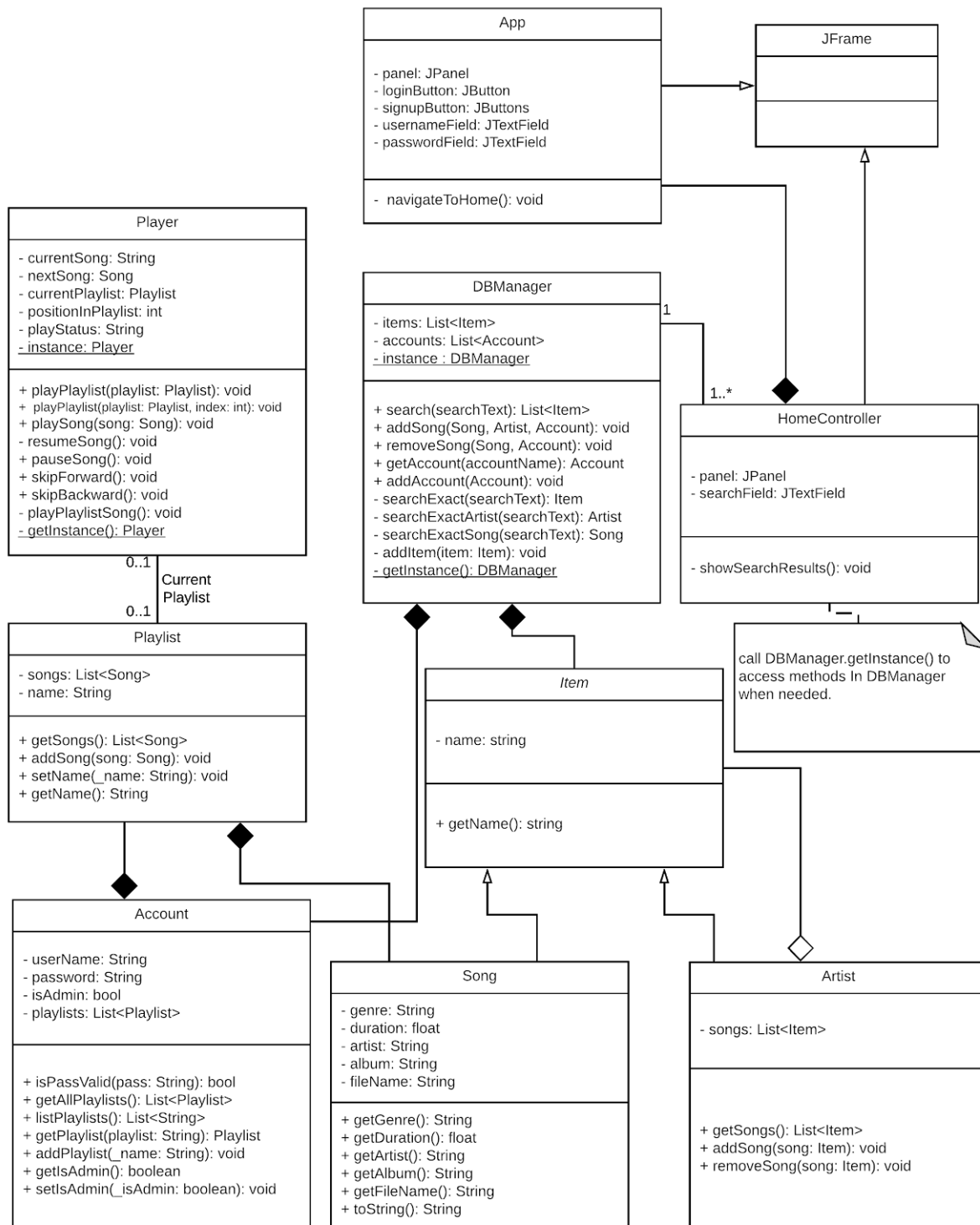


1. Team 05 Names (Github):

- Scott Young (scottlyoung)
- Amjad Alharbi (amjaad)
- Joe Alsko (JAlsko)
- Jonah Jacobsen (jjjacobsen)
- **Project Title:** Web-based Music Player
- **Project Summary:** A simple web-based Music player system that allows users to listen to their favourite songs, create and edit their own playlists.

2. Previous class diagram

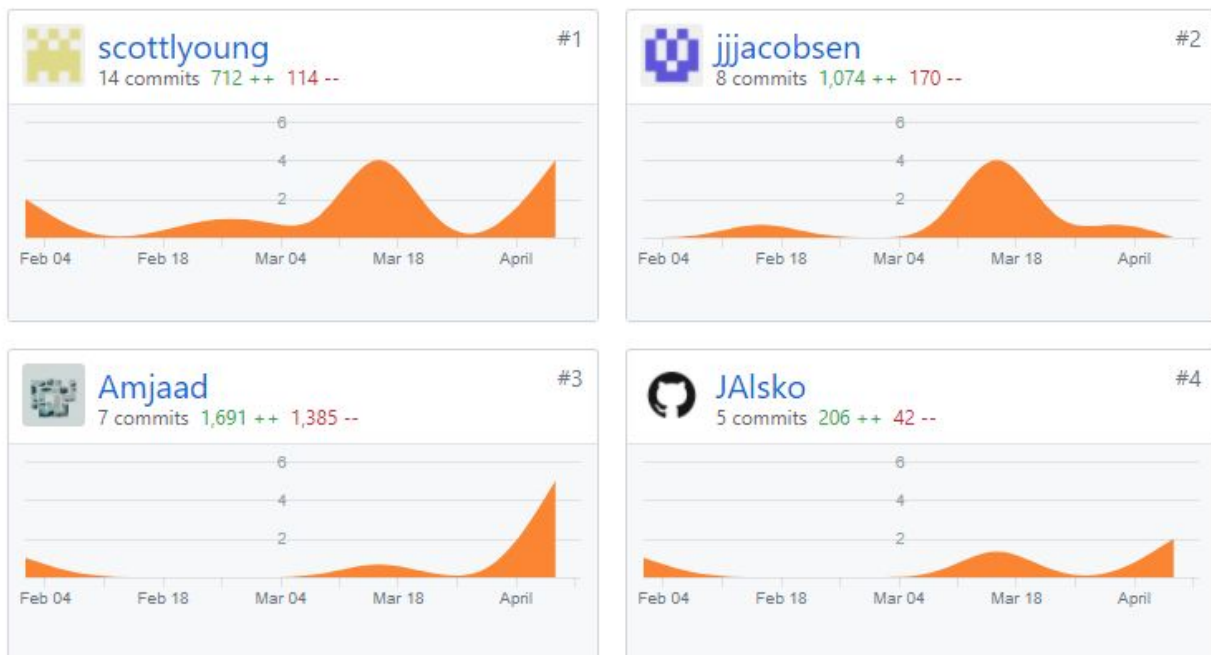
4. Summary: These past weeks, we've worked on fleshing out some of the components of our system, including the audio player, the account setup, and the main input class. We also began implementation of the client and server as separate programs.



5. Breakdown of work

- Scott Young
 - Added account storage to database
 - Started Client Server Networking
- Amjad Alharbi
 - Implementing search for songs views/controllers
- Joe Alsko
 - Beginning implementation for audio file handling
 - Finished Player class and integrated it with input loop
- Jonah Jacobsen
 - Implemented testing stand in for MVC

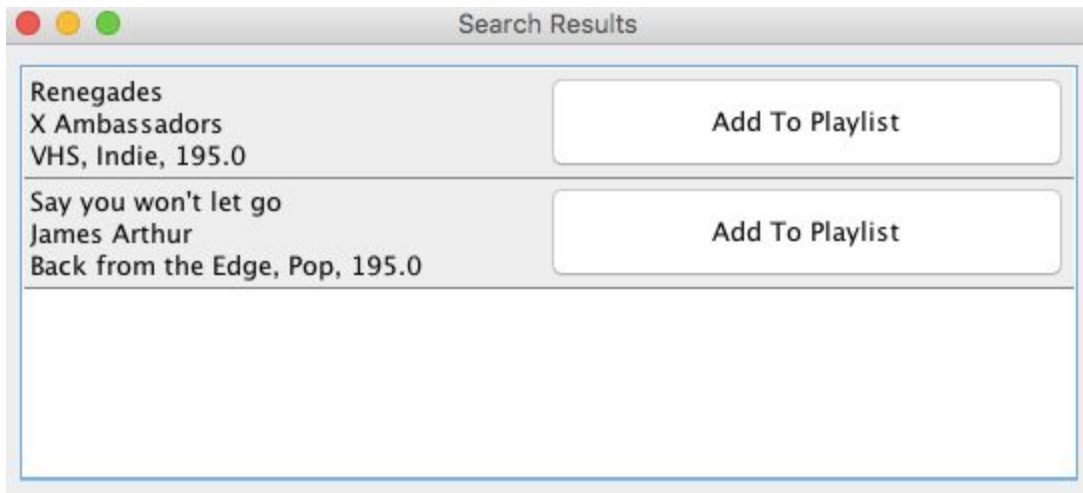
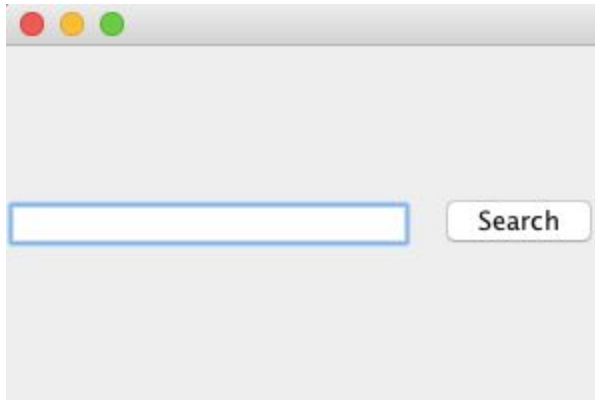
6. GitHub Graph: Graph in Github of each person's contribution.



7. Estimate Remaining Effort:

A screenshot of a web form with a light gray background. At the top, there are three colored circles (red, yellow, green) in a row. Below them, the form has the following elements:

- A text input field labeled "Username" containing the text "user55".
- A text input field labeled "Password" containing seven black dots.
- A button labeled "Login" with a light gray background and a thin border.
- A button labeled "Signup" with a light gray background and a thin border.



To Do

- Need to implement server data storing when offline
- Need to finish MVC
- Need to finish client/server communication

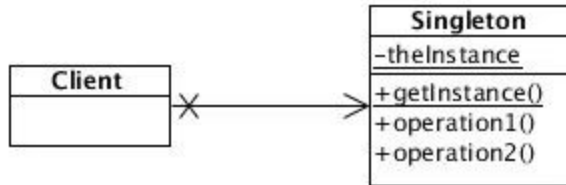
8. Design Patterns:

Singleton: We used the singleton pattern for our audio player class and database manager class. There should only be one audio player and one database manager at any time, so this pattern seemed appropriate.

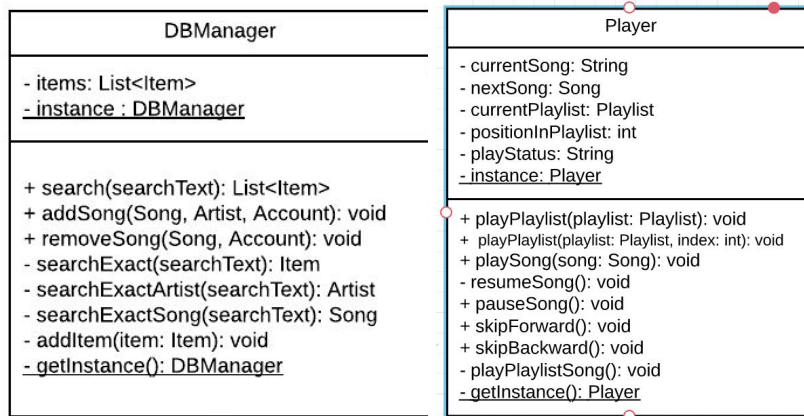
Composite: To better search our list of songs and artists, we used the composite pattern to reference artists and songs simultaneously, while also using a list containing the Item class to keep track of an artist's existing songs.

9. Show the portion of your class diagram with each design pattern implemented:

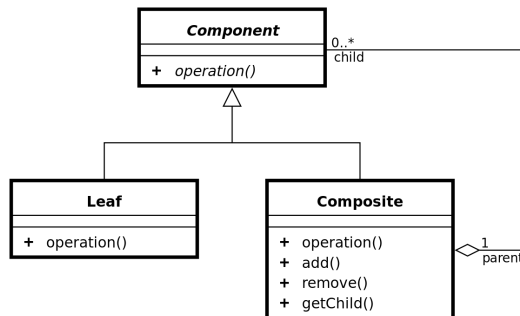
Singleton



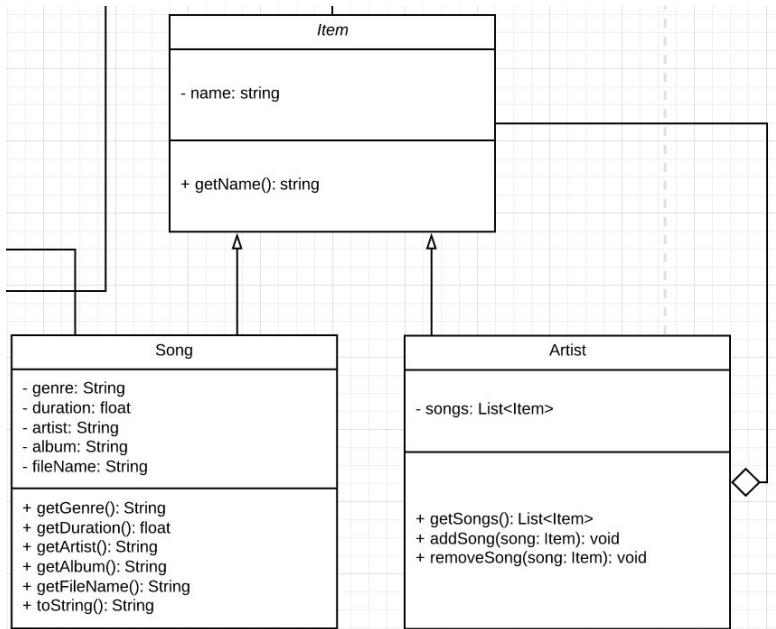
Our Implementations



Composite



Our Implementation



- Designate the participants in your own diagram

10. Final Iteration:

Finish implementing all views and controllers for the second use case, adding songs into the database. Finish client server communications.