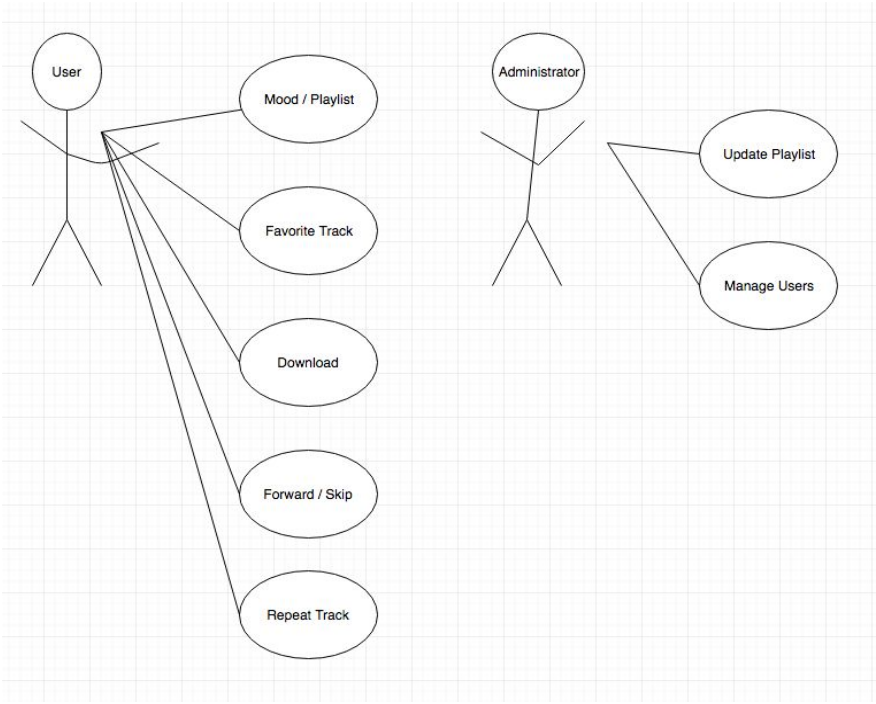
SCHOOL OF INFORMATICS, COMPUTING, AND CYBER SYSTEMS
CS 386 – Software Engineering
Prof. Marco Gerosa
Team Project – D.4

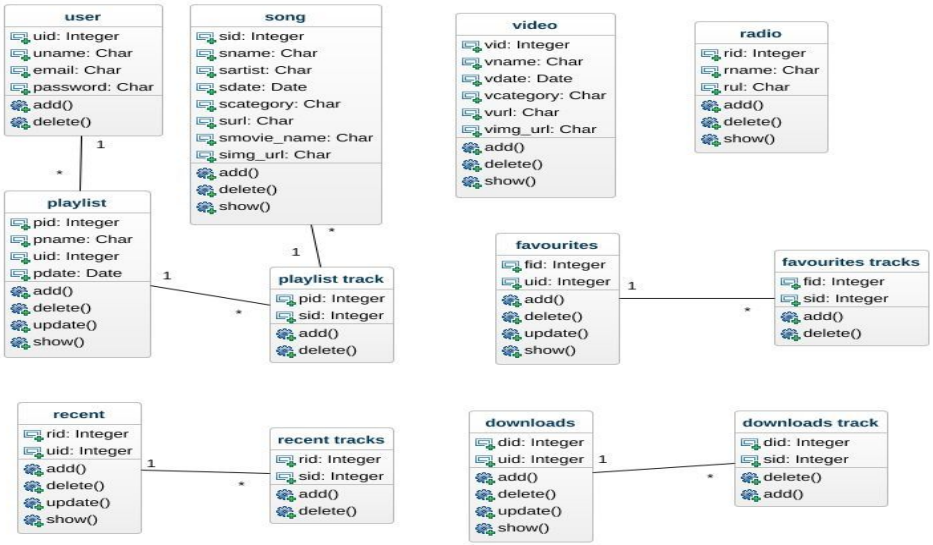
1. Description Provide 1-2 paragraphs to describe your system. This will help understand the context of your design decisions.

Our system will help users listen to music without having to discover music. It will lets users broaden their horizon with good music, without having to search through random phishing websites. The problem of not having a music player that satisfies all the needs of the modern user affects populations of various taste and age; the impact of which is not having a reliable music streamer and a place to listen to the music of their choice.

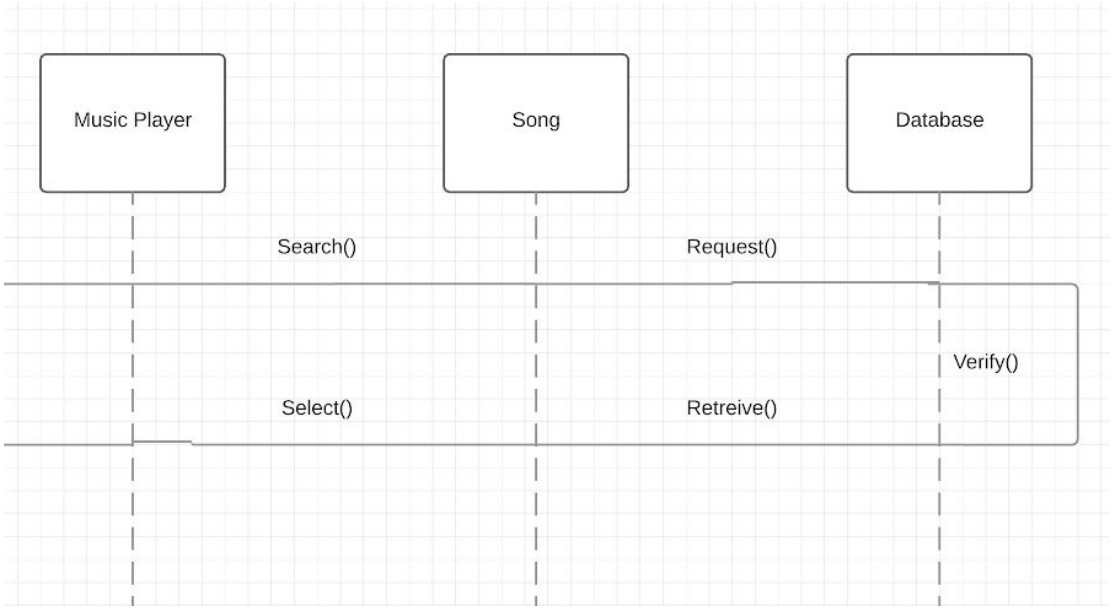
Users will have the chance to broaden their horizon with good music without having to search through random phishing websites. For music lovers of all age who are struggling to find a centralized place to put all their music, *Mood* music streaming app that allows the users to play music based on their current mood hence creating a playlist off that. Mood has the ability to create and modify easily and have an easy way to download their songs unlike Spotify or Pandora our product will not rely on premium plans to let the user create playlists or listen to music without having to be interrupted. The user will have the choice of favoriting the track and add it to their own playlist or downloading it. They may also watch the music video or audio of the track. Another feature our application will have is a music blog side. If the user like keeping up to date on music culture, they will have chance to read up and watch videos on our music blog.

2. Architecture Present a diagram of the high-level architecture of your system. Use a UML package diagram to describe the main modules and how they interrelate. See some examples at: https://www.uml-diagrams.org/index-examples.html Make clear the layers of your architecture (if they exist) as described in: https://www.uml-diagrams.org/multi-layered-application-uml-model-diagram-example.html Provide a brief rationale of your architecture explaining why you designed it that way.





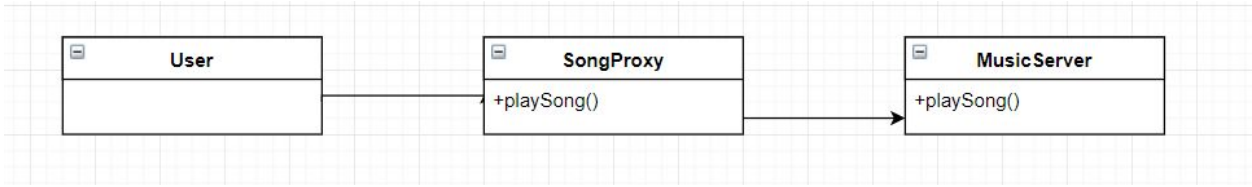
4. Sequence diagram Present a sequence diagram that represents how the objects in your system interact for a specific use case. Also include the use case description in this section. The sequence diagram helps to understand the dynamic aspect of your design.



This specific use case is when a song is being searched and selected. First, the song is searched by the application among many databases. Once it has been found, it is requested by the specific database and is then verified that it is the correct song in playable form and does not contain any corrupted data. If the verification is successful, it is retrieved from the database by the music player and waits to be selected by the user. When it is selected by the user, its audio file is then played back from the music player for the user to enjoy. From there, many other choices can be made but do not apply specifically to this use case.

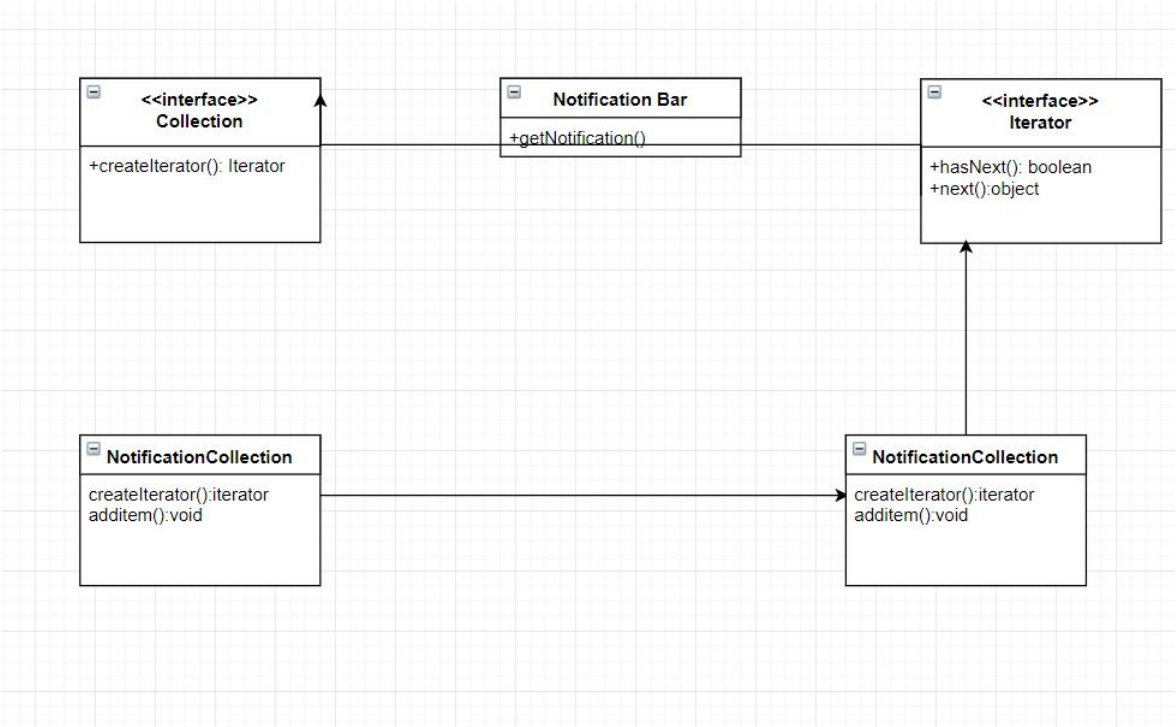
5. Design Patterns Split this section into 3 subsections. For each subsection, present a UML class diagrams showing the application of a design pattern to your system (a different pattern for each section). Each class diagram should show only the classes involved in the specific pattern (you don't need to represent the whole system). Choose patterns from at least two of these categories: Behavioral, Structural, and Creational. You are not limited to design patterns 2 studied in class. Maybe your system is not appropriate for any design pattern. In this case, for didactic purpose, be creative and extend a little bit the scope of your system to make 3 design patterns appropriate.

Proxy



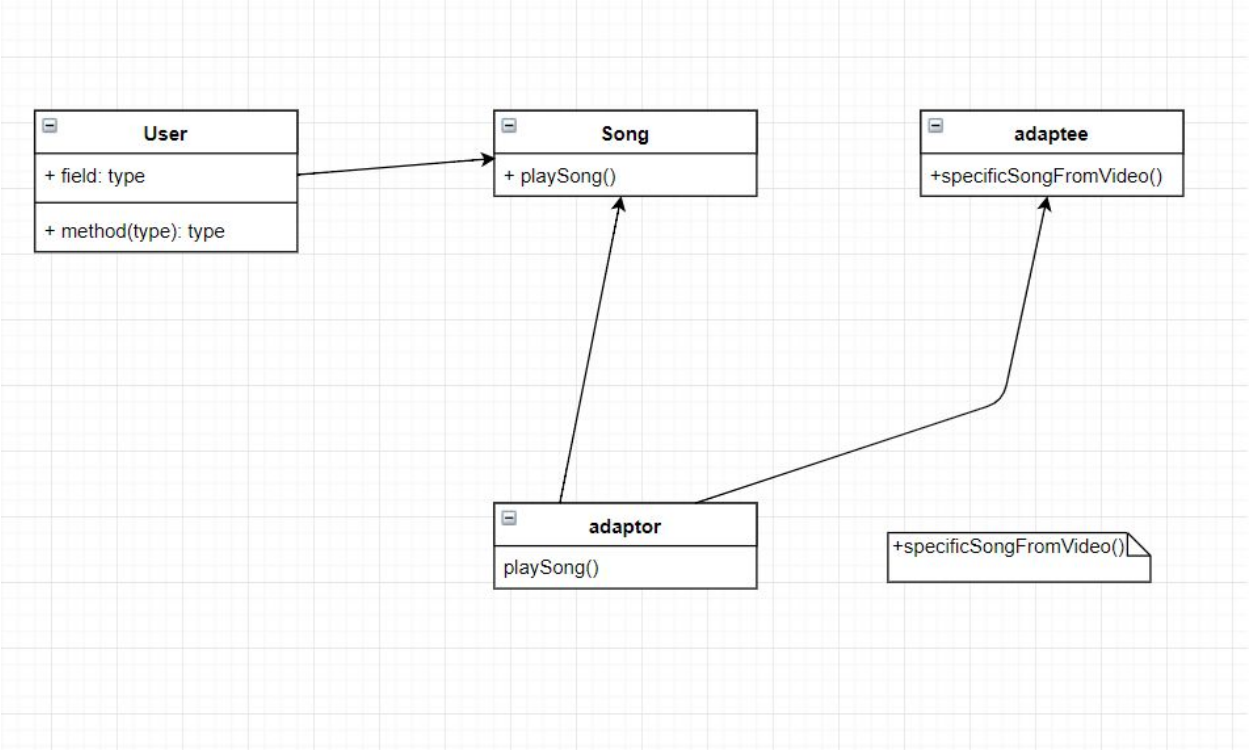
Our songs will be stored on a server and a proxy will help mediate and act as an object for the user to the remote object which is the song.

Iterator



One of the functions we want to implement in our final product is a notification bar that lets users know of any new songs, playlists or albums released. All of these notifications are stored in the notification collection and the NotificationCollection provides an iterator to iterate over its element without having to expose how it has implemented the array.

Adapter



User wants to play a song in the background that is a video file but our app does not support playing video songs in the background. We use an adapter to let the user play the song by calling the method playSong from the song class interface. The adapter converts the request on the adaptee using the adaptee interface. User gets to listen to the song without being aware of the adaptor's presence.

6. Design Principles How does your design observe the SOLID principles? Provide a short description of followed principles giving concrete examples from your classes.
By using the design patterns we are eliminating the need to have many tasks cramped together in one class. All of our classes do one task only. Our objects can be extended but they are not modifiable.

7. Group participation Provide this section as described in the "Team Project Instructions."

Turan (20%), Karsten(20%), Austin C(20%), Austin T(20%) and Mike (20%)