MyTunes - Compulsory Assignment #4



Put your own MyTunes program here!

**Handed-in by Group E:**

1. **Christian Hansen**
2. **Frederik Flagstad**
3. **Mario Ampudia Valdes**
4. **Roula Bakr**
5. **Tienesh Kanagarasan Sivasubremaniyam**

**Date: 18/12-2019**

# Table of Content

Table of Content 2

1. Introduction 2

2. State of delivery 2

3. Functionality 2

4. process documentation 2

5. Application Structure 2

6. Data storage 2

7. Implementation details 2

8. Source details 2

9. Source code 2

10. Evaluation 2

# Introduction

The fourth compulsory assignment is about designing and constructing a JavaFXML application, able to provide administration of various songs and playlists and capable of playing songs. Furthermore, is the application expected to contain songs and playlists, allowing the user to choose between various playlists and songs alike.

For the duration of this assignment, groups of 3-4 members are put together. Though, a few groups were not able to be fully composed of the designated number of members, which led to said groups’ dispersal and they were put into other groups, resulting in groups to be composed of five members instead.

# State of Delivery

Which requirements are met?

A GUI is made, a graphical user interface is important for a media player to have. Our MyTunes

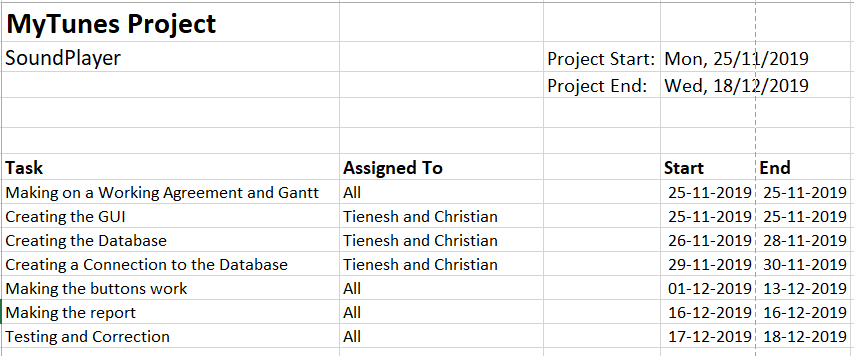
# Functionality

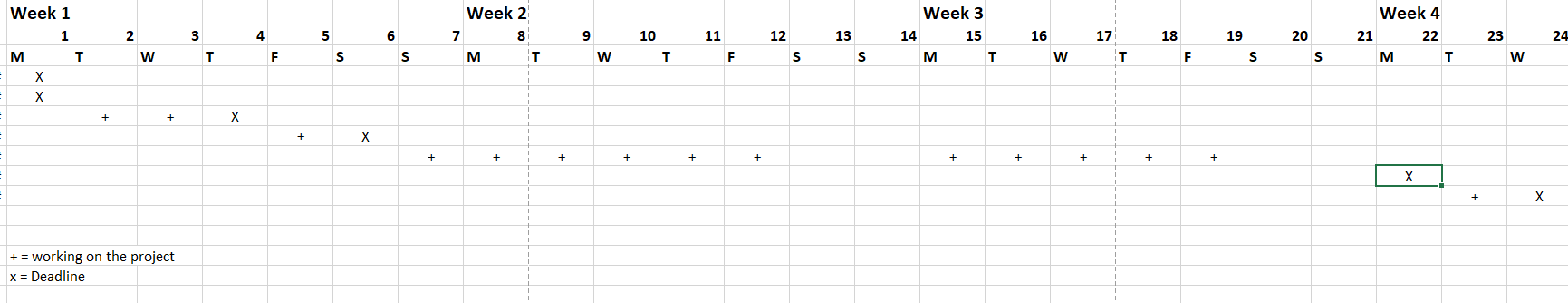
The functionality of the player must be described. You may refer to requirements in the assignment (1-7 mandatory and a) to g) optional).

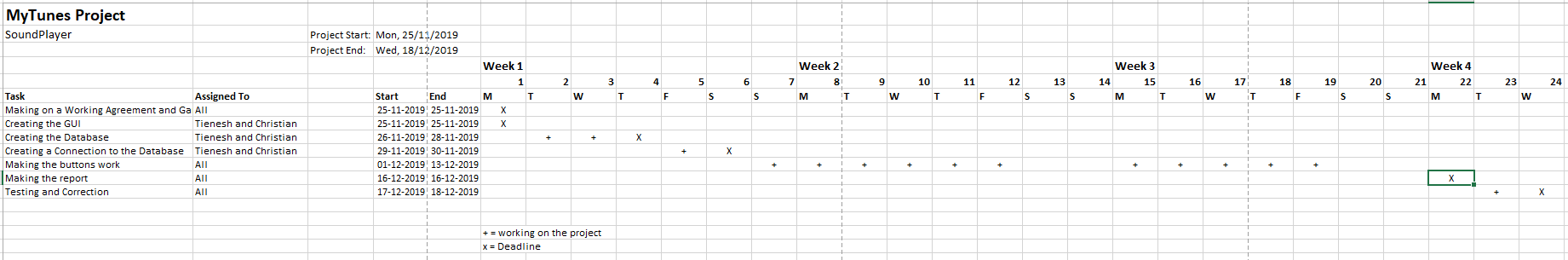
1. The MyTunes project is expected to be a desktop application with a graphical user interface.
2. MyTunes is a traditional music player application, allowing the user to manage the songs and playlists by themselves. For this to be possible, it is expected that there is a song table.
   1. All songs are to be displayed.
   2. The user can sort through this via build-in functionality
   3. The user is allowed access to both edit and create a new song to be added.
      1. A dialog is expected to appear to state the user’s choice and aid them. This is to be the same when the user wants to delete a song.
3. A filter function is to be added and allows the user to sort through their playlists for songs by either *artist* or *title*.
4. A dialog is to appear to ensure that the user is permitted to create and edit a new playlist.
5. When a playlist is chosen, the songs contained in that playlist is to be displayed.
6. It has to have the most common function of a music player, meaning that it has to be able to play the next song in the playlist if the previous one is done playing.
7. Playlists and the song lists must be saved to a database using JDBC. This ensures that the songs can be loaded when the program opens.
8. The filetypes the program is allowed to open and play are music files like “.wav” and “.mp3” files.

# Process Documentation

For the project’s duration, we followed the process design, the Gantt Model, allowing the group to assign tasks, to whom should complete said tasks and assign deadlines for the individuals and their work. The Gantt Model also allows us to make a starting date for the tast and the end date of said task.





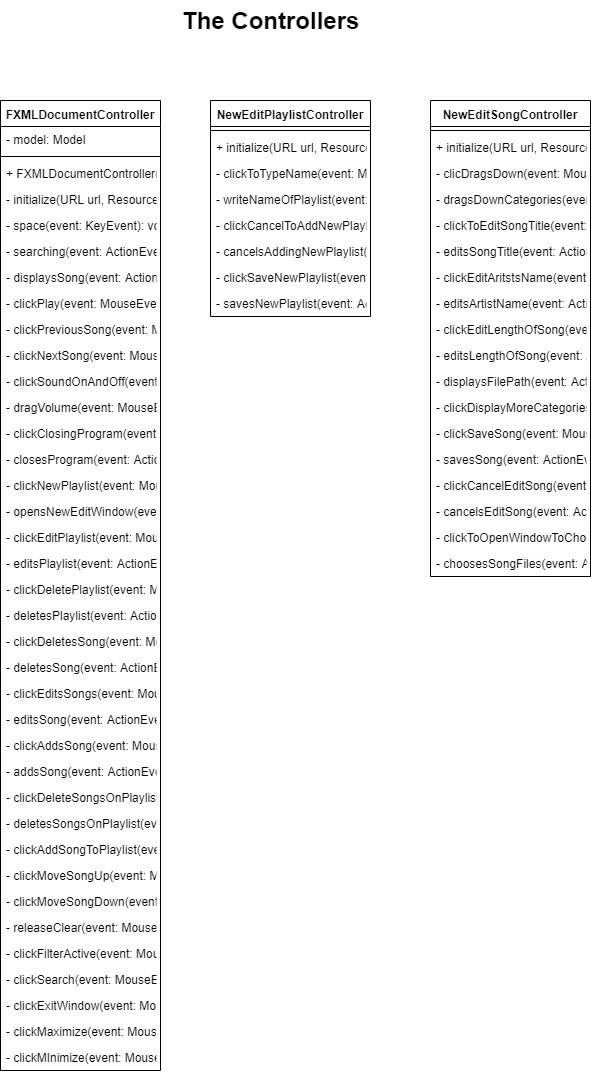
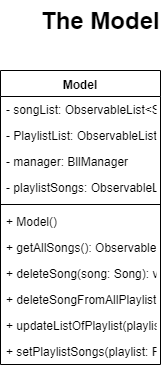


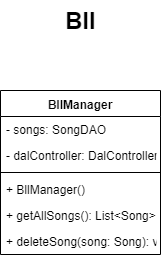
# Application Structure

Present one or more UML class diagrams for your mediaPlayer. Remember to comment your diagrams.

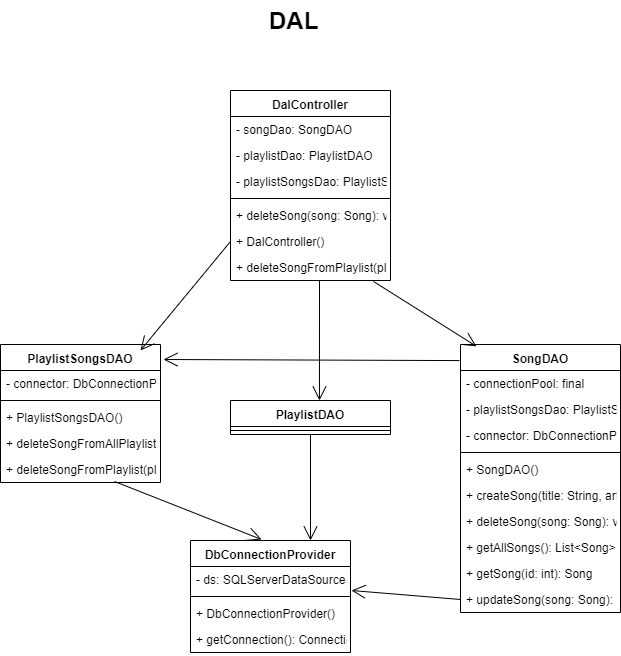


GUI



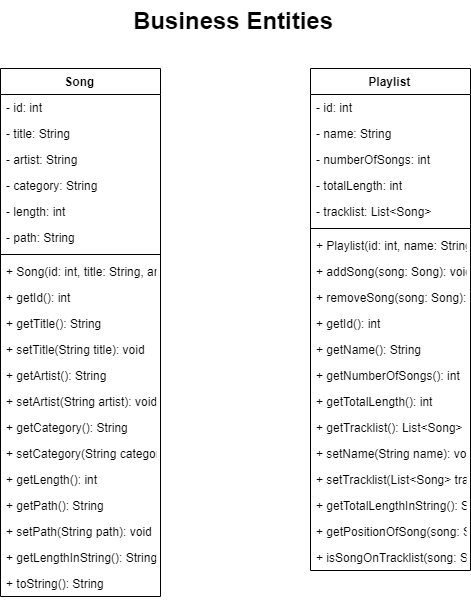


Bll

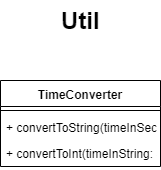


Dal

The GUI, Bll and Dal are depending on the Business Entities.

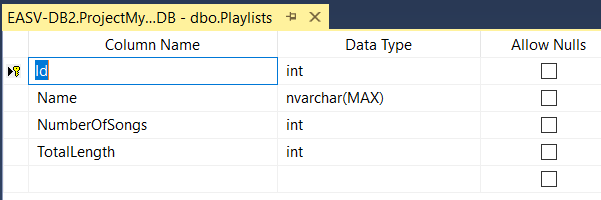


The Song and Playlist classes are depending on the TimeConverter Class.

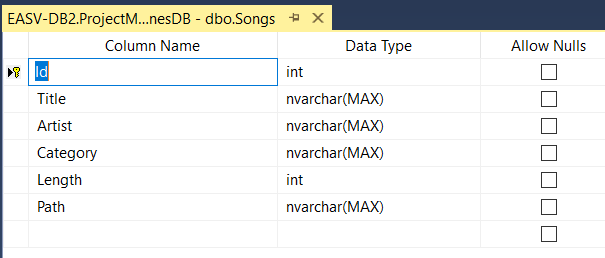


# 6. Data Storage

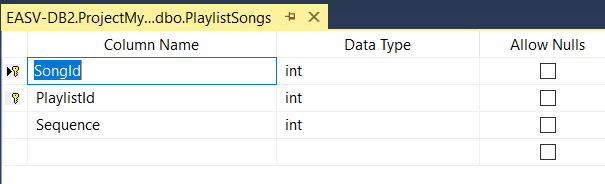
Playlists:



Songs:



PlaylistSongs:



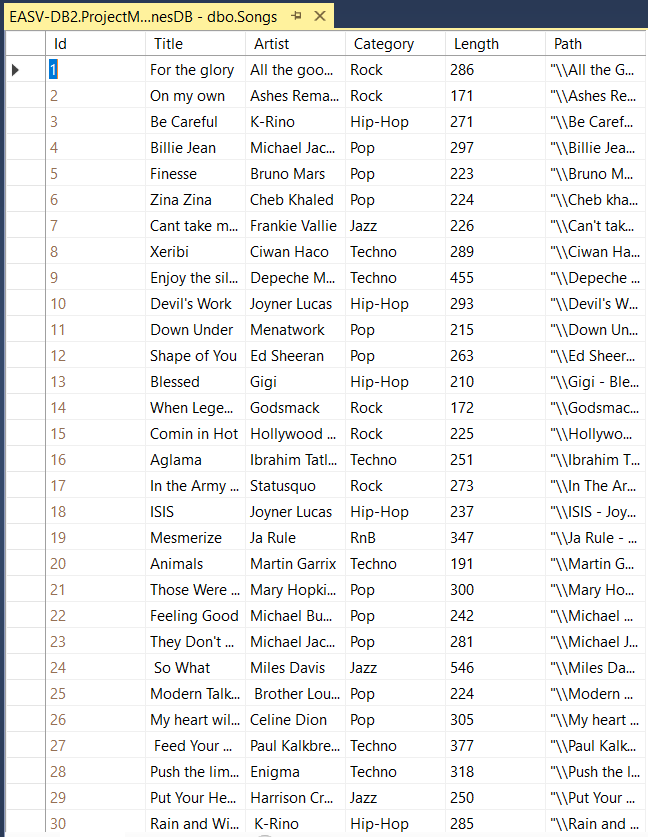
There’s a foreign key from the table “Songs” to the table “PlaylistSongs”.

There is a foreign key between “id” of the table “Songs” and “SongId” of the table “PlaylistSongs”.

There is a foreign key between “id” of the table “Playlists” and “PlaylistId” of the table “PlaylistSongs”.

The “Id” of the songs will be assigned automatically by the database.

Database with Songs, including, id, title, artist, category(genre), length of song in seconds and filepath.



# 7. Implementation Details

In case there are issues that caused you troubles or solutions that you find smart you must describe these. You may insert code snippets whenever you find it beneficial.

# 8. Source Control

This section describe which system has been used for source control (Git), the name of the repository and how the team has used this through the project.

Throughout the duration of the compulsory assignment, the group utilized the tool GitHub for sharing the coding of the project. Each group member connected to GitHub and downloaded the desktop application, “GitHub Desktop”. Via this, the group was able to download the original MyTunes project created in Christian’s repository. Any improvements done by any member will be committed to Github, allowing the other members to receive the improved project with the new added changes.

The name of the repository is: “MyTunes”

# 9. Source Code

Remember to add a link for the teachers to your GitHub repository. Do not change the repository after hand-in.

The link for Group E’s GitHub repository during the fourth compulsory assignment:

“<https://github.com/ChristianH321/MyTunes.git>”

# 10. Evaluation

Was it a good project? What did you learn? What to do better, to improve next time?