Willian Bernatzki Woellner

Student ID 30021175

Programming 3

Question 1 – Predesign Specification

Table of Contents

[Question 1 – Predesign Specification 2](#_Toc54258259)

[Data Structures 2](#_Toc54258260)

[Hashing Techniques 2](#_Toc54258261)

[Sorting Algorithm 2](#_Toc54258262)

[Search Technique 2](#_Toc54258263)

[Third-Party Libraries 3](#_Toc54258264)

[Graphical User Interface 3](#_Toc54258265)

[Source Control 4](#_Toc54258266)

[Coding Standards 4](#_Toc54258267)

[Tests 5](#_Toc54258268)

[References 6](#_Toc54258269)

# Question 1 – Predesign Specification

Jupiter Mining Corporation requires an advanced music player that allows the ability to sort and search the songs stored in a binary tree. The GUI should display the sorted tracklist and highlight and play the searched track, it should save the tracklist to a CSV using a 3rd party library.

## Data Structures

The Music Player program store the songs on a Binary Tree, it is a data structure where each node has up to two child nodes, creating the branches of the tree. The two children are usually called the left and right nodes. (Techopedia, 2018)

## Hashing Techniques

Hashing Techniques are using to insert, sort, and search a song in the Binary Tree. Each Tree node is created using a key and value that represents the original information about the song. For example title, album, and singer.

## Sorting Algorithm

The program will use the Tree sort algorithm, it is an online sorting algorithm that builds a binary search tree from the elements input to be sorted, and then traverses the tree, in-order, so that the elements come out in sorted order. (Baeldung, 2020)

## Search Technique

The program will use the Binary Search technique to search a song on the Binary Tree. Binary search is a fast search algorithm with the run-time complexity of Ο(log n). This search algorithm works on the principle of divide and conquer. For this algorithm to work properly, the data collection should be in the sorted form. (Tutorials Point, 2020)

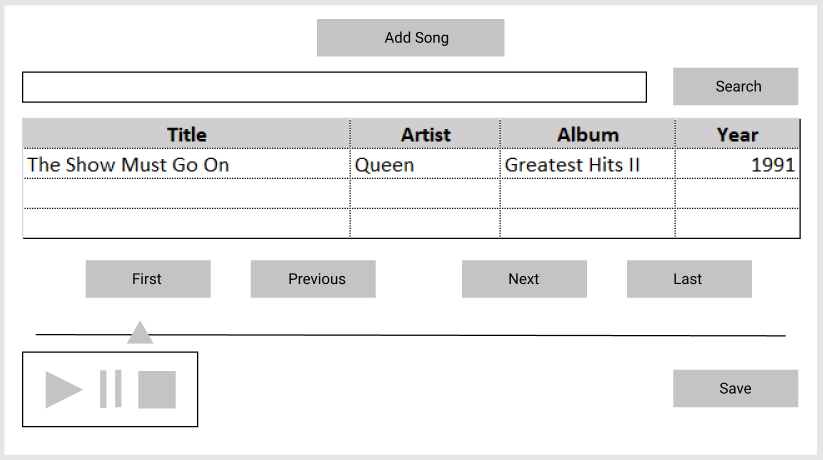
## Third-Party Libraries

CSVHelper is a .NET library for reading and writing CSV files. It will be used to save the tracklist on a CSV file. Its documentation is available on the website: <https://joshclose.github.io/CsvHelper/api>

TagLib is a .NET platform-independent library for reading and writing metadata in media files, including video, audio, and photo formats. It will be used to collect the song information to display. Its documentation is available on the website: <https://github.com/mono/taglib-sharp>

Windows Media Player SDK provides information and tools to customize Windows Media Player and Windows Media Player Mobile and to use the Windows Media Player ActiveX control and the Windows Media Player Mobile ActiveX control. Its documentation can be access on the website: <https://docs.microsoft.com/en-us/windows/win32/wmp/about-the-windows-media-player-sdk>

## Graphical User Interface

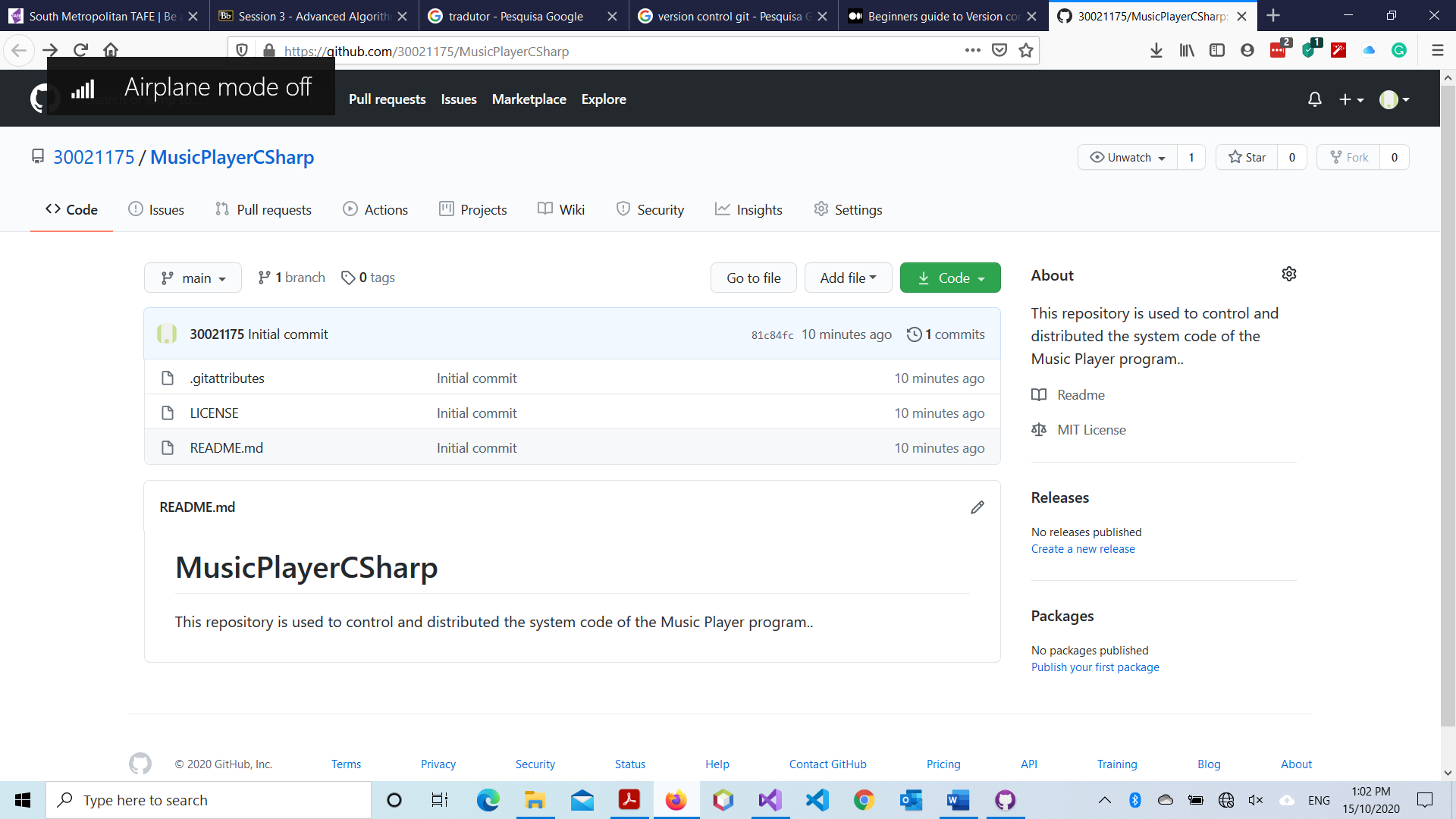


* The Search will be done by the song title.
* All songs added will be displayed on the table.
* All songs will be displayed sorted by title.
* The Save button will be used to save the tracklist on the CSV file.

## Source Control

The project will be used GitHub to control and distributed the versions of the system. Git is a free and open-source distributed version control system or protocol. (Bhushan, 2019)

The repository is available on <https://github.com/30021175/MusicPlayerCSharp>



## Coding Standards

The program will be used some coding standards listed below:

* Indentation
* Naming conventions for local variables, global variables, constants, and functions
* Code will be well documented
* Standard headers for different modules (Name, description, parameters, and returns)
* Code will be developed using Camel Case

## Tests

The program will be tested using two categories: Black Box Testing and White Box Testing. All tests will be documented on a table with screenshots. The main functionalities listed below will be tested:

* Add a song to play
* Search a song
* Sort the tracklist
* Save the tracklist on CSV file

# References

Baeldung. (2020, july 24). *Sorting the Elements in a Binary Tree*. Retrieved from Baeldung: https://www.baeldung.com/cs/sorting-binary-tree

Bhushan, M. (2019, July 30). *Beginners guide to Version control using Git and GitHub*. Retrieved from Medium: https://medium.com/faun/beginners-guide-to-version-control-using-git-and-github-8bf44b421140

Techopedia. (2018, September 5). *Binary Tree*. Retrieved from Techopedia: https://www.techopedia.com/definition/1145/binary-tree

Tutorials Point. (2020, October 15). *Data Structure and Algorithms Binary Search*. Retrieved from Tutorials Point: https://www.tutorialspoint.com/data\_structures\_algorithms/binary\_search\_algorithm.htm