|  |
| --- |
| Jupiter Mining Corp |
| Software Project Outline |
| Project: Advanced Music Player |

|  |
| --- |
| Josh Macaulay  6-11-2020 |

Contents

[Data Structures in Use 1](#_Toc42767685)

[Binary Tree 1](#_Toc42767686)

[Hashing Techniques 1](#_Toc42767687)

[Profile Password 1](#_Toc42767688)

[Sorting Algorithm 1](#_Toc42767689)

[Merge Sort 1](#_Toc42767690)

[Searching Algorithm 1](#_Toc42767691)

[Binary Search 1](#_Toc42767692)

[Third-Party Libraries 1](#_Toc42767693)

[CSV Helper 1](#_Toc42767694)

[Documentation for CSV Reader/Writer 1](#_Toc42767695)

[Graphical User Interface 2](#_Toc42767696)

[Prototype 2](#_Toc42767697)

[Source Control 3](#_Toc42767698)

[GitHub 3](#_Toc42767699)

[Coding Standards 3](#_Toc42767700)

[C# Coding Conventions 3](#_Toc42767701)

[Testing 3](#_Toc42767702)

[Types of testing to be done 3](#_Toc42767703)

[References 4](#_Toc42767704)

# Data Structures in Use

## Binary Tree

For the storage and searching of individual tracks in the playlist, I will be implementing a Binary Tree. This will allow tracks to be added to the list in a sorted manner. Binary Trees allow fast searches based on a logarithmic proportion.

# Hashing Techniques

## Profile Password

The application will use passwords for each profile, with each profile having its own track listings. The passwords will be hashed and stored locally in the applications data files. Having the entered password typed in plaintext, then hashed and compared to the stored hash provides decent security. The application will have an implementation of a “Secure Hash Algorithm” or SHA.

# Sorting Algorithm

## Merge Sort

After storing the tracks in a Binary Tree, the output list of tracks can be sorted alphabetically. The type pf sorting algorithm used will be the Merge Sort algorithm. Merge Sort is useful for its efficiency and its general-purpose.

# Searching Algorithm

## Binary Search

Since the application will be using a Binary Tree to store its tracks, the binary search is an easy implementation of a search function that utilises the architecture of the tree to perform its search.

# Third-Party Libraries

## CSV Helper

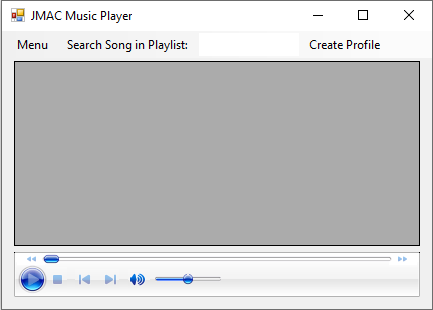
The application will use the 3rd party .NET library called CsvHelper. It allows fast and flexible function of reading csv files and writing them.

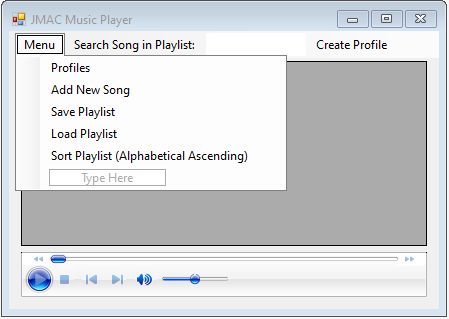
## Documentation for CSV Reader/Writer

The documentation for the CsvHelper library can be found at the libraries GitHub page <https://joshclose.github.io/CsvHelper/api/>.

# Graphical User Interface

## Prototype





# Source Control

## GitHub

For the development of this application, GitHub will be used as the source control. During its project development, every new function will be committed to the main source files. With every main function implemented in iterations, there will be a period of testing to find any bugs in its operation. The bugfixes will be committed and pushed to GitHub and after fixing, a new branch will be created for the next iteration (or version).

# Coding Standards

## C# Coding Conventions

The coding convention and standards that will be applied to this development are the C# coding conventions. This will ensure a consistent look to the code, enable readers to understand the code more quickly by allowing assumptions based on previous experience, and to facilitate copying, changing, and maintaining the code (Microsoft, 2015).

# Testing

## Types of testing to be done

This application will have System Testing and Acceptance Testing performed.

System Testing in the form of white box testing where the tester will use the application in its complete form to test the various functions within the application. The tester knows how the program was implemented and allows them to check all the possible paths through the algorithm.

Acceptance Testing is a validation test, where the tester will compare the functionality of the application with the requirements set by the client. If the project satisfies the requirements, it will have passed the Acceptance phase.

# References

Microsoft. (2015, July 20). *C# Coding Conventions (C# Programming Guide)*. Retrieved from docs.microsoft.com: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions