*<Advanced Music Player>*

Product Design Specification

Version *<1.0>*

*<19/06/2020>*

VERSION HISTORY

Development of the Advanced Music Player for Jupiter Mining Corp is designed by Joshua Macaulay to a range of specifications and requirements. The project will track its development with source control and documentation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 1.0 | Joshua Macaulay | *19/06/2020* |  | *<mm/dd/yy>* | Initial Design Definition draft |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

TABLE OF CONTENTS

[1 Introduction 4](#_Toc42771505)

[1.1 Purpose of The Product Design Specification Document 4](#_Toc42771506)

[2 General Overview and Design Guidelines/Approach 4](#_Toc42771507)

[2.1 Assumptions / Constraints / Standards 4](#_Toc42771508)

[3 Architecture Design 5](#_Toc42771509)

[3.1 Software Architecture 5](#_Toc42771510)

[3.2 Security Architecture 5](#_Toc42771512)

[3.3 Performance 5](#_Toc42771513)

[4 System Design 6](#_Toc42771514)

[4.1 Use-Cases 6](#_Toc42771515)

[4.2 Data Conversions 6](#_Toc42771516)

[4.3 User Interface Design 7](#_Toc42771517)

[5 Product Design Specification Approval 8](#_Toc42771518)

[Appendix A: References 9](#_Toc42771519)

[Appendix B: Key Terms 10](#_Toc42771520)

# Introduction

## Purpose of The Product Design Specification Document

The Product Design Specification document documents and tracks the necessary information required to effectively define architecture and system design in order to give the development team guidance on architecture of the system to be developed. The Product Design Specification document is created during the Planning Phase of the project. Its intended audience is the project manager, project team, and development team. Some portions of this document such as the user interface (UI) may on occasion be shared with the client/user, and other stakeholder whose input/approval into the UI is needed.

# General Overview and Design Guidelines/Approach

This section describes the principles and strategies to be used as guidelines when designing and implementing the system.

## Assumptions / Constraints / Standards

The system will be implemented in NetBeans, using JavaFx for its user interface and overall design. It will be programmed to the Java Coding Conventions standards to provide a common look in its programming.

## Client Business Domain

The purpose of this project for the client is that their employees can use a music player application created by the company. As the company is Jupiter Mining, they are not intending on marketing this software to those outside of the company as their domain involves mining.

## Industry Accepted Hardware and Software

The industry accepted hardware is computers, laptops, mobile devices, routers, servers, databases, WiFi transmitters, and network infrastructure. The software is operating systems, document handling software, and firmware of the hardware.

## Role of Stakeholders

The role of the stakeholders in this environment is to remain in contact with the development team to ensure at each step in the development cycle, the developers can confirm requirements and allow correspondence.

## Quality Assurance

Refining the requirements is an important part of quality assurance. This ensures that the project being developed is what the clients and stakeholders expect. Reviewing code, repeated testing and following code conventions ensure the code created is of decent quality.

## Target Systems Specifications

The software implemented in this task can be executed on any PC that runs on Windows 10. It will not have compatibility with any other operating system except potentially legacy windows systems, but testing those systems is outside of scope.

# Architecture Design

This section outlines the system and hardware architecture design of the system that is being built.

The main part of the system architecture is the main Java class Application that interacts with the functions that are separated into different class structures. These class structures will become a library for the main class to utilize. It will also implement a 3rd party library for its Csv Reader and Writer.

## Software Architecture

## 

## Security Architecture

The encryption software used will be a Secure Hash Algorithm, which will encrypt the stored passwords of the login passwords. This allows the application to take an input plaintext password and hash it to compare with the stored one.

## Performance

The application will be using Merge Sort and Binary Search algorithms for efficient performance during sort and search operations.

# System Design

## Use-Cases

1: User selects play music option.

2: User selects stop music option.

3: User selects add song option.

4: User selects export option.

5: User selects import option.

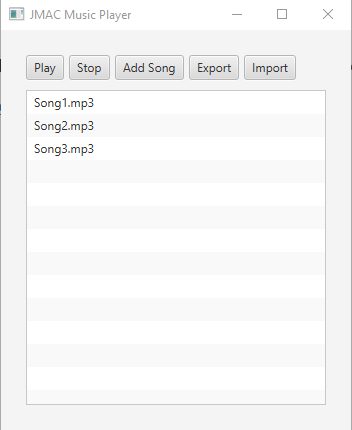
6: User selects Sort option.

7: User uses Search Option.

## Data Conversions

The Java CSV Library by ‘shriop’ will convert array data into csv format for storage. Java CSV Library will also load data from csv file into array format.

## User Interface Design



## Version Control

This application will be developed using GitHub source and version control.

<https://github.com/JMac118/Advanced-Music-Player-Java>

# Testing and Deployment

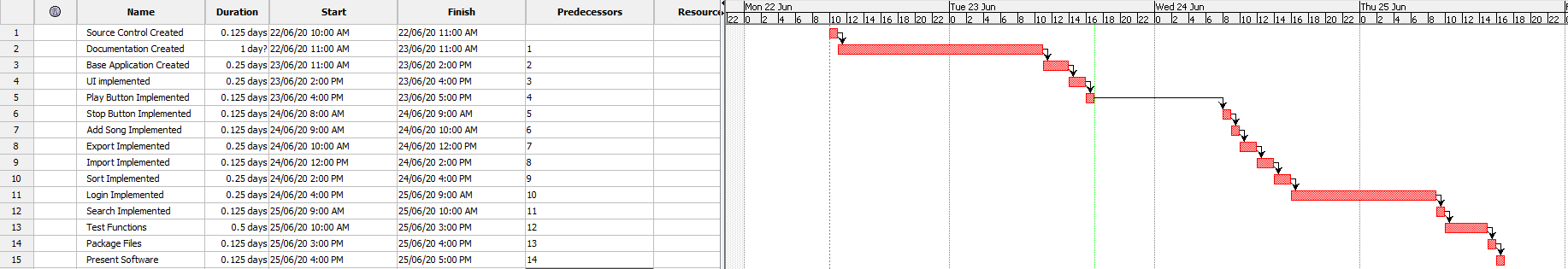
## Test Documentation

The testing process will be following the testing style that was outlined in the Software Project Outline document. The testing will be displayed in a table in the document titled ‘Test Documentation’ with screenshot evidence provided in the ‘Screenshots’ folder. All documentation will be available on the GitHub source. There will also be Junit tests performed on the functions.

## Implementation Plan/Deployment

Implementation of the software will follow the Gantt Chart structure provided in the documentation. For deployment of the software it will be packaged in a .exe installer file with the source files in a zip file uploaded to the central server.

## Gannt Chart

Product Design Specification Approval

The undersigned acknowledge they have reviewed the *Advanced Music Player* **Product Design Specification** document and agree with the approach it presents. Any changes to this Requirements Definition will be coordinated with and approved by the undersigned or their designated representatives.

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

Appendix A: References

The following table summarizes the documents referenced in this document.

|  |  |  |
| --- | --- | --- |
| **Document Name and Version** | **Description** | **Location** |
| UML Image | An image file of the UML chart of the project. | Advanced-Music-Player-Java\Screenshots |
| UI Prototype | Image file of the prototype UI | Advanced-Music-Player-Java\Screenshots |
| Gannt Chart | An image of the Gannt chart | Advanced-Music-Player-Java\Screenshots |

Appendix B: Key Terms

The following table provides definitions for terms relevant to this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
|  |  |