EECS2311: SOFTWARE DEVELOPMENT PROJECT

Systems Requirement Specification

### February 2, 2022

### PREPARED FOR

Vassilios Tzerpos

Lassonde School of Engineering

### PREPARED BY

Hiba Jaleel - 215735020

Kuimou Yi - 216704819

Kamsi Idimogu - 216880288

Maaz Siddiqui - 216402927

# Table of Contents

[**1. Introduction**](#_kwgv17tdyjyh) **2**

[1.1 Purpose](#_tsujb6bdbrox) 2

[1.2 Intended Audience](#_ogu2a0admwfv) 2

[1.3 Intended Use](#_4g91yr33a2og) 2

[**2. Overall Description**](#_fhrqfww32v11) **3**

[2.1 Product Perspective](#_cgv1fy5zt8kx) 3

[2.2 Product Overview](#_75rgmghqjjc5) 3

[2.3 User Class Characteristics](#_9d8alilbkztz) 3

[2.3.1 Use Case #1](#_3t1sjqm15dei) 3

[2.3.2 Use Case #2](#_xbqppwfu74r5) 3

[2.3.3 Use Case #3](#_t3mnngpuhpme) 3

[**3. System Features and Requirements**](#_p59z0qdj3yme) **4**

[3.1 Functional Requirements](#_7w9a4pcaflbk) 4

[3.1.1 Tablature to MusicXML conversion](#_4lqbf5s89ipb) 4

[3.1.1.1 Description](#_lebi81ut36iq) 4

[3.1.1.2 Stimulus/Response Sequences](#_p2zv1ga1b9uu) 4

[3.1.2 MusicXML Visualization](#_e590pguqmsou) 4

[3.1.2.1 Description](#_3zp2rbn60r50) 4

[3.1.2.2 Stimulus/Response Sequences](#_yyscdg93nj9t) 4

[3.1.3 MusicXML playing](#_msag7pkuyog0) 5

[3.1.3.1 Description](#_2l18r46cva9k) 5

[3.1.3.2 Stimulus/Response Sequences](#_bxi0tvovfopz) 5

[3.2 External Interface Requirements](#_8leh106ita89) 5

[3.2.1 User Interface](#_wtrlvor5ita) 5

[3.3 Nonfunctional Requirements](#_acfuhvfsn8ki) 5

[3.3.1 Software Abilities](#_4vhz1sclyg26) 5

# 1. Introduction

## 1.1 Purpose

The purpose of the Software Requirement Specifications document is to describe the functions our software, the MusicXML previewing system will be able to perform and how exactly our software will be able to achieve the desired results.

## 1.2 Intended Audience

This document is intended for:

*Customer/Client:* The customer or client will be expressing what features they would like the software to have, and through this document ensure that the software will cover the aspects they would like.

*Project Manager:* The project manager will be a point of contact for any conveyance between the developers and the customer/client and communicate any new demands or changes in the customer/client’s requirements. The project manager will also be setting deadlines for the developers to follow and assist in maximizing developer efficiency.

*Developers:* The developers will be working on implementing the requirements of this software as laid out by the customer/client through their development of the software.

## 1.3 Intended Use

The software development timeline of the project is from January 10, 2022 to April 11, 2022. The main costs involved in this project are time, energy, and electricity. The software’s ability to perform the required tasks will be the measure of success. Our team has determined the core features/functionality, important features, and nonfunctional requirements to be the deliverables of this project. Risk will be evaluated by grading it’s exposure, probability of occurrence, and loss size (in days). Based on the context and severity of the risk, effective mitigation measures will be implemented.

# 2. Overall Description

## 2.1 Product Perspective

This software package (application) will be accessed when run through Eclipse. The application will connect to the user interface of MusicXML Converter.

## 2.2 Product Overview

This software will allow users to convert their text-based tablatures into music sheets. Additionally, it will allow users to play those music tablatures after the conversion.

## 2.3 User Class Characteristics

### 2.3.1 Use Case #1

**Title***:* Preview Music Sheet

**Primary Actor**: Who desires it? User/user/Client

**Preconditions***:* The user has started the application.

**Success Scenario***:* The user inputs the text-based tablature. The user previews the music sheet.

### 2.3.2 Use Case #2

**Title**: Saving Music Sheet

**Primary Actor**: Who desires it? User/user/Client

**Preconditions**: The user has started the application.

**Success Scenario**: The user inputs the text-based tablature. The user previews the music sheet. The user saves the music sheet.

### 2.3.3 Use Case #3

**Title**: Play Music

**Primary Actor**: Who desires it? User/user/Client

**Preconditions**: The user has started the application.

**Success Scenario**: The user inputs the text-based tablature. The user previews the music sheet. The user plays music.

# 3. System Features and Requirements

## 3.1 Functional Requirements

### 3.1.1 Tablature to MusicXML conversion

#### 3.1.1.1 Description

This feature allows the customer to convert their text-based tablature into a MusicXML file.

#### 3.1.1.2 Stimulus/Response Sequences

* The user inputs their text-based tablature into the designed text input area in the graphic user interface.
* The user inputs essential metadata of a MusicXML file (e.g title, creator..)
* The system identifies the type of instrument that the inputted text-based tablature represents.
* The system outputs it into a MusicXML file

### 3.1.2 MusicXML Visualization

#### 3.1.2.1 Description

This feature allows the customer to visualize the MusicXML file. The visualization features include a preview of the visualization result in the form of a music sheet and the option to export the result as a .pdf file.

#### 3.1.2.2 Stimulus/Response Sequences

* The user will select desired MusicXML-based tablature.
* The user will have a window to preview the visualized music notations.
* The user can export it as a PDF file in order to print or share.

### 3.1.3 MusicXML playing

#### 3.1.3.1 Description

This feature allows the customer to play the MusicXML file.

#### 3.1.3.2 Stimulus/Response Sequences

* The user will select desired MusicXML-based tablature
* The user will have a window to preview the visualized music notations, and this windows will have a “Play Music” button
* The music sheet will be played with the desired instrument.

## 3.2 External Interface Requirements

### 3.2.1 User Interface

The user interface will be designed to ensure simplicity and ease of use. The user interface will prioritise efficiency in both speed and use.

## 3.3 Nonfunctional Requirements

### 3.3.1 Software Abilities

The software should:

* handle at least one text-based tablature input at a time
* handle multiple requests back to back
* contain interactive buttons that enable features such as the ‘Preview Music Sheet’ button or the ‘Play Music’ button
* be able to handle large text-based tablatures
* handle the users request within seconds
* be available to use at all times

### 