Software Requirements Specification (SRS)

Project TAB2MXL

Authors: Patrick Alexander Baciu, Sanchit Duggal, Kevin Hua,Venkatessh Kumar

Customer: Vassilios Tzerpos

Instructor: Vassilios Tzerpos

Table of Contents

1. Introduction

This Software Requirements Specification document provides an overview of the functionality of a TAB2MXL (T2M) software system designed for people interested in learning to play a particular piece of music. This document will cover the scope and requirements of the T2M.

1.1 Purpose

The objective of this document is to describe the specifications and functionality of the design of T2M. The expected viewers for this document are developers and users of the T2M application.

1.2 Scope

TAB2MXL will be designed to run on general computer systems. The user interested in MusicXML format can effortlessly upload a text file or directly paste text containing the guitar, bass, or drums tablature for a particular song and have it converted into a MusicXML file which can be used to precisely denote a piece of music or be played by another application.

1.3 Definitions, acronyms, and abbreviations

- Tab (Tablature) - a form of musical notation indicating instrument fingering rather than musical pitches.
- XML (Extensible Markup Language) - A widely used type of text data organization and storage language that uses „<“ and „>“ to label and distinguish sections of data or instructions from each other.
- MusicXML - An XML-based file format for representing Western musical notation.

- GUI (Graphical User Interface) - The part of the application that the user sees and interacts with.
- PC (Personal Computer) - A desktop or laptop.

2. Functionality

2.1 Capability

This product will be capable of receiving plain text, or .txt files of tablature and converting them into a .musicxml file.

2.2 Reusability

This product will be a reusable desktop application that will be entirely reusable. There are infinitely many conversions provided by T2M, that are only limited by the amount of persistent storage on the system. T2M will be able to be reused for various types of tablature, including guitar, bass guitar, and percussion instruments.

2.3 Security

Since T2M is a native desktop application that does not interface with any external APIs or web services, there are no risks to user security. Users are not required to register an account, as there is no data persistence or history of converted tabs. Each time the application is launched, it is as if the user is running T2M for the first time.

3. Usability

3.1 Human Factors

There are several human factors at play when considering the usability of our application. After discussion with our client,

3.2 Aesthetics

T2M is meant to be simple, clean, and easy to use. This will be accomplished by a well organized, and pleasant looking user interface.

3.3 Consistency

T2M is a consistent software. At various stages of the conversion process, the user will be prompted to clarify/correct any unclear information in the tab (inserting bar lines, clarifying time signature, clarifying tempo, etc). As a result, our software will be able to reliably generate .musicxml files from tablature text.

3.4 Documentation

This application will contain a detailed README file with instructions as to how to use the application. The codebase is clean and commented for future developers who wish to use, upgrade, or maintain the software.

3.5 User Cases

3.5.1 Primary Actor: Music Instructors

Success Scenario: Jorge is a music instructor. He noticed that his guitar students have trouble understanding sheet music and relied on tabs to play pieces. To teach his students formal sheet music, Jorge uses TabToMXL to convert tablature into musicxml and displays them side by side for his students when teaching them. Jorge hopes to see improvement in his students' ability to read sheet music.

3.5.2 Primary Actor: Industry Professionals

Success Scenario: Axl is a guitar player who has written several books of guitar tablature for his hit albums. Axl wishes there was a way to easily convert this tablature into sheet music for distribution to a wider audience. TabToMXL will help Axl convert his tablature to sheet music which can be added to his future publications alongside his tablature.

3.5.3 Primary Actor: Amateur Musicians

Success Scenario: Ella wants to learn the guitar. She knows the basics but has trouble understanding music theory. “Tabs don’t really teach me music in the way I’d like”. To deepen her understanding of music theory, Ella can use TabToMXL to convert her tabs into sheet music and practice her sight reading.

3.6 User Stories

- Music instructors can make use of this software to turn tablature into sheet music for their students. This can make arrangements for jazz and orchestral pieces much easier, since tablature tends to be more widely available than sheet music.

- Industry professionals will be able to use T2M to create sheet music from their tablature. Most bands already sell “tab books”, which could be expanded to contain sheet music if our software is used. This would reach a wider audience and could be profitable for bands.

- Amateur musicians who wish to learn sheet music can use our software to easily convert their tablature into practice material. This would simplify writing and learning formal sheet music for the amateur musician, as only basic knowledge of tablature is needed.

4. Reliability

4.1 Availability

This software will be usable on any device running MacOS, Windows, or any flavour of Linux with Java installed. No internet connection is required for functionality as all conversion is done client-side.

4.2 Failure Rate and Predictability of Failures

The purpose of this application is to limit the rate of failure by consulting the user at various stages of the conversion process. We can expect failure when tab formatting is inconsistent, spacing is incorrect, or non-tablature data is entered into the program. Failure should be handled gracefully and not met with a program crash.

5. Performance

5.1 Speed

This application will ideally operate quickly. The conversion process should take less than 10 seconds to complete; this is a goal that will be adjusted in the future.

5.2 Efficiency

The efficiency of the program will be directly proportional to the length of a given piece of music. Longer tabs will take longer to convert, shorter tabs, shorter. The exact efficiency can be computed once the algorithm is finished.

5.3 Resource Consumption

This program is limited by the system's memory and CPU. Exact constraints will be determined in the future.

5.4 Scalability

T2M will support guitar and bass tabs in beta. Drum tabs will be added in the future, ideally before a full release.

6. Supportability

6.1 Testability

Our program will be tested with jUnit tests. We will use a serializer to compare our parsed models to the original tablature before converting those models into musicXML format. Once the conversion is complete we will conduct another series of tests by parsing the musicXML into tablature and comparing it to the original tab. (this is an ideal feature and will be added depending on time)

6.2 Extensibility

This application will support tablature for other instruments. In the future, adjustments may be implemented to support 7 string or 8 string guitars. The code written will be extensible, upgradable, and maintainable to ensure future functionality of the application.

7. Plus Category

7.1 Design Constraints

The design is constrained by limitations in weekly work hours, and experience developing full scale applications.

7.2 Implementation

This software is written using Java 15, JavaFX, Gradle, and possibly some external libraries (yet to be decided).

8. References

1) “Francisco Tárrega - Capricho Arabe (Tab),” Guitar.Com, 16-Jun-2020. [Online]. Available: https://tabs.ultimate-guitar.com/tab/86734. [Accessed: 03-Feb-2021].

2) “MusicXML,” Wikipedia, 21-Dec-2020. [Online]. Available: https://en.wikipedia.org/wiki/MusicXML. [Accessed: 03-Feb-2021].

3) Template based on IEEE Std 830-1998 for SRS. Modifications (content and ordering of information) have been made by Betty H.C. Cheng, Michigan State University (chengb at cheng.cse.msu.edu)