

Tab to Music Score

User Manual

Version 1.1 (Prototype)
2022-02-20

Group 7:
Aleksander Weinberger
Harsimran Saini
Chirag Sardana
David Hanna

Table of Contents

User Manual	3
1. Introduction	3
2. Background	3
2.1. Music Theory	3
2.2. MusicXML	4
3. Overview	4
4. Conventions	4
5. Getting Started	6
5.1. System Requirements	6
5.2. Run the System	6
6. Close the System	10

Images

Image 1	5
Image 2	6
Image 3	7
Image 4	7
Image 5	8
Image 6	9
Image 7	10
Image 8	10

1. Introduction

The intended purpose of this document is to provide the client and its users an overview of the continued development of the software the course provided and will serve as a step-by-step guide in installing, using, and troubleshooting the latest version of Group 7's application. Additionally, this document should reflect any new features and changes in later versions. It is important to note that this document will not cover (insert elements).

2. Background

This document is better understood if the reader reviewed the following noteworthy concepts and definitions about music theory and MusicXML:

1. Music theory:

- 1.1. The **staff/stave** is a set of five horizontal lines and four spaces that each separately represent a different musical pitch.
- 1.2. The **clef** is a musical symbol used to indicate which notes are represented by the lines and space on a musical staff/stave (i.e., G-clef (treble), F-clef (bass), and so forth).
- 1.3. The **note value** is the duration of a note when played.
- 1.4. The **note head**, either filled (black) or open (white), indicates which musical note to play and its duration (note value).
- 1.5. The **stem** serves the purpose of making it easier to read musical notes while allowing them to fit neatly on the staff.
- 1.6. The **note flag** is a curvy mark to the right of the note stem. Its purpose is to tell you how long to hold a note.
- 1.7. **Beams** serve the same function as note flags. However, it helps musicians read music more clearly by keeping the notation less cluttered.
- 1.8. The **time signature** is a fraction where the top number states how many beats are in a measure, and the bottom number states the note value of each beat.
- 1.9. **Pitch** is how high or low a specific note sounds.
- 1.10. An **octave** is composed of eight notes, where the interval between those notes is one musical pitch.

- 1.11. A **chord** is a set of three or more musical notes.
- 1.12. A **fret** is a space on the neck of a stringed instrument and represents a semitone, where one octave has twelve semitones.
- 1.13. The Guitar has the following notes ordered and arranged from lowest-pitch to highest-pitch (bottom string to top string): E, A, D, G, B, and E.

2. MusicXML:

- 2.1. The **<divisions> element**, presented inside a measure's attributes object, serves as a **unit of measure** for the duration element in terms of divisions per quarter note.
- 2.2. The **<duration>** element reflects the intended note duration when played. For example, for a division value of one and duration value of four, the resulting note duration is four quarter notes.

Note: This section will update for later versions of the application.

3. Overview

The purpose of the software is to provide an easy way to access preview sheets for musicians after providing the text sheet . The text tablature is converted to music xml which in turn is converted to sheet music . There is also a graphical user interface which provides easy access for this functionality .The application is based on java programming language which uses eclipse IDE . It also uses gradle build to build and run the application.

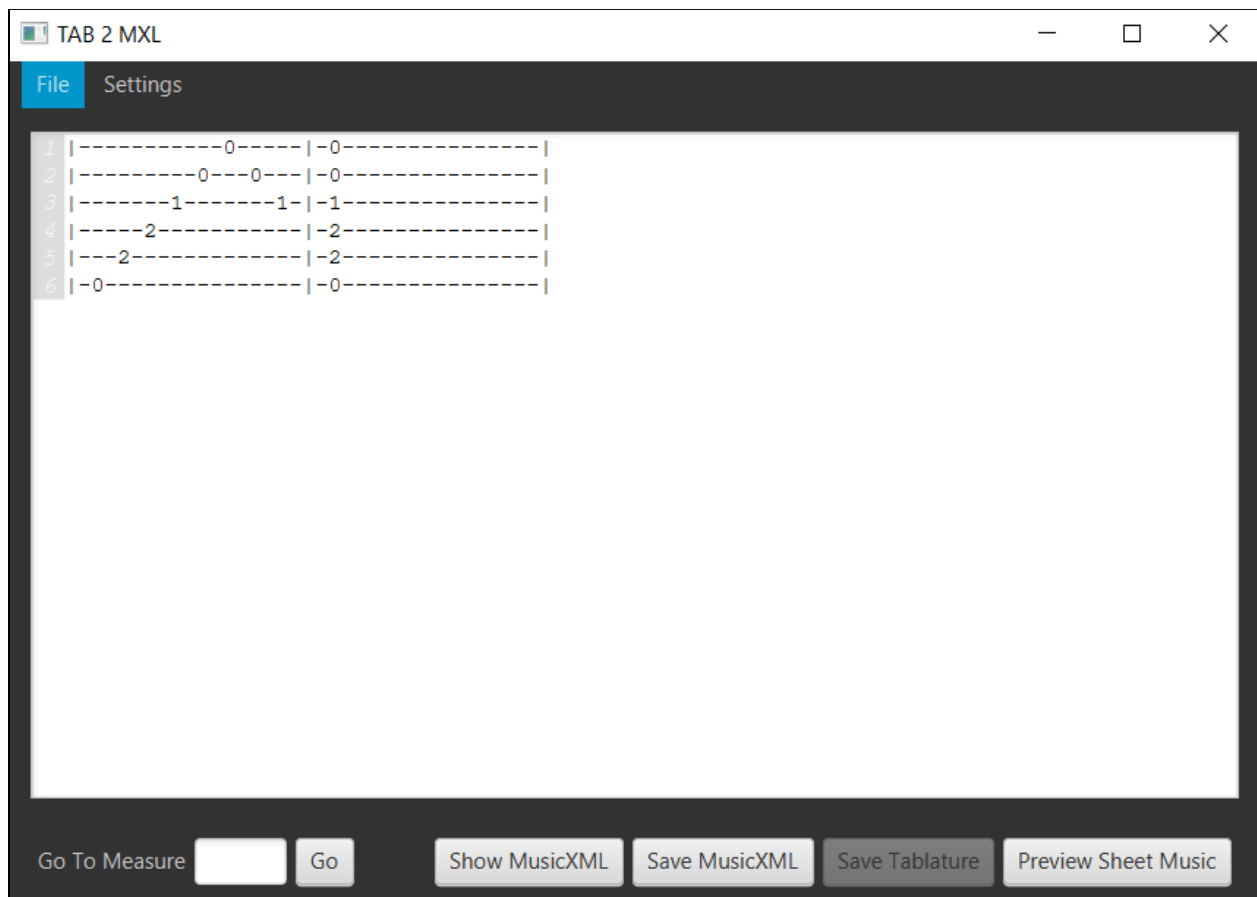
4. Conventions

In the latest version of this application, the main features that would support the extended functionality are still in development. For that reason, the specific conventions for the extended functionality of the application have not been documented yet. However, there exists a set of guidelines for the tablature a user may supply the software. Since the application does support the preview

of basic guitar tablatures (see image 1), it is highly recommended to the reader to review the following conventions for tablature:

1. Each row must have at least one measure.
2. Each measure must have at least one character.
3. Measures are the same length.
4. Vertical bars separate measures.
5. Spacing and comments between rows are allowed.
6. Representation of special characters (such as "h" or "p" for hammer-ons and pull-offs, respectively) can exist between two notes.

It is important to note that other conventions exist for the provided system TAB 2 MXL. It is up to the user to review those conventions by inspecting the `src/main/resources` folder in the TAB 2 MXL package.



(Image 1 - A guitar tablature that the 'preview sheet music' feature supports)

5. Getting Started

1. System Requirements

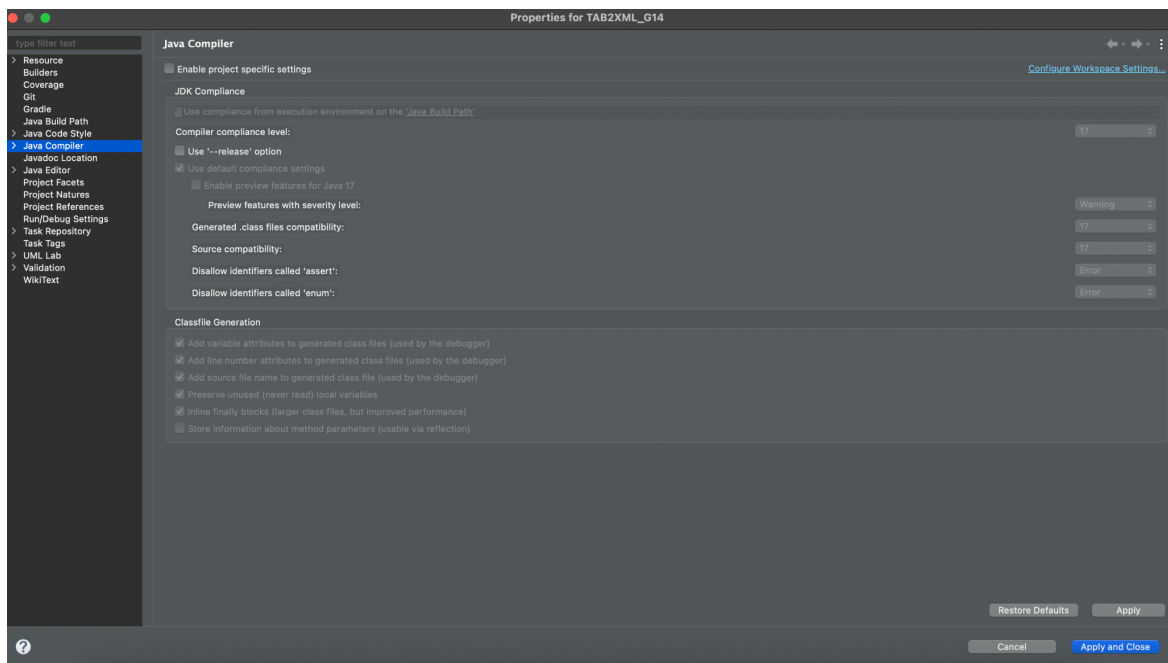
1.1. Java Compiler

In order to install the software, Java -17 compiler needs to be installed. If Java -15 can't be found please install it using this link.

<https://www.oracle.com/ca-en/java/technologies/javase-downloads.html>

File → Properties → Java Compiler → Compiler Compliance Level → 17

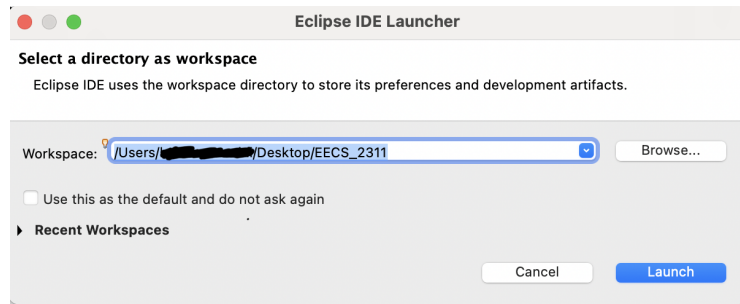
In Eclipse IDE the compiler settings for Java looks like the following (see image 2).



(Image 2 - Compiler settings in Eclipse IDE)

2. Run the System

Open eclipse and open a new workspace (see image 3).

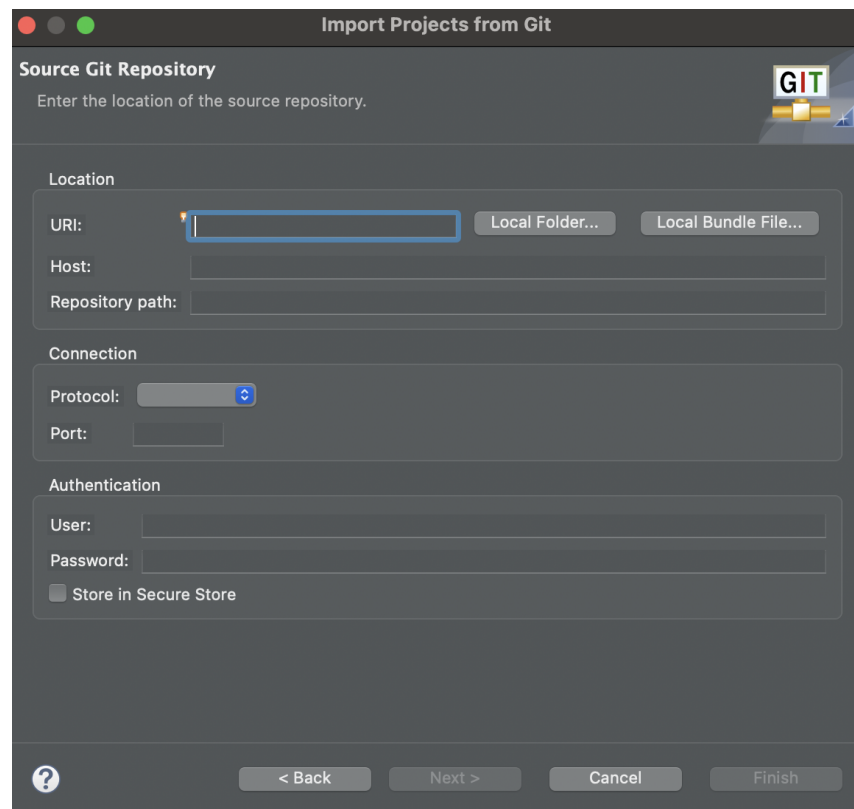


(Image 3 - Eclipse IDE Workspace selector)

Now, once eclipse is opened, then

Go to **File** → **import** → **Git** → **Projects from Git** → **Clone URI**

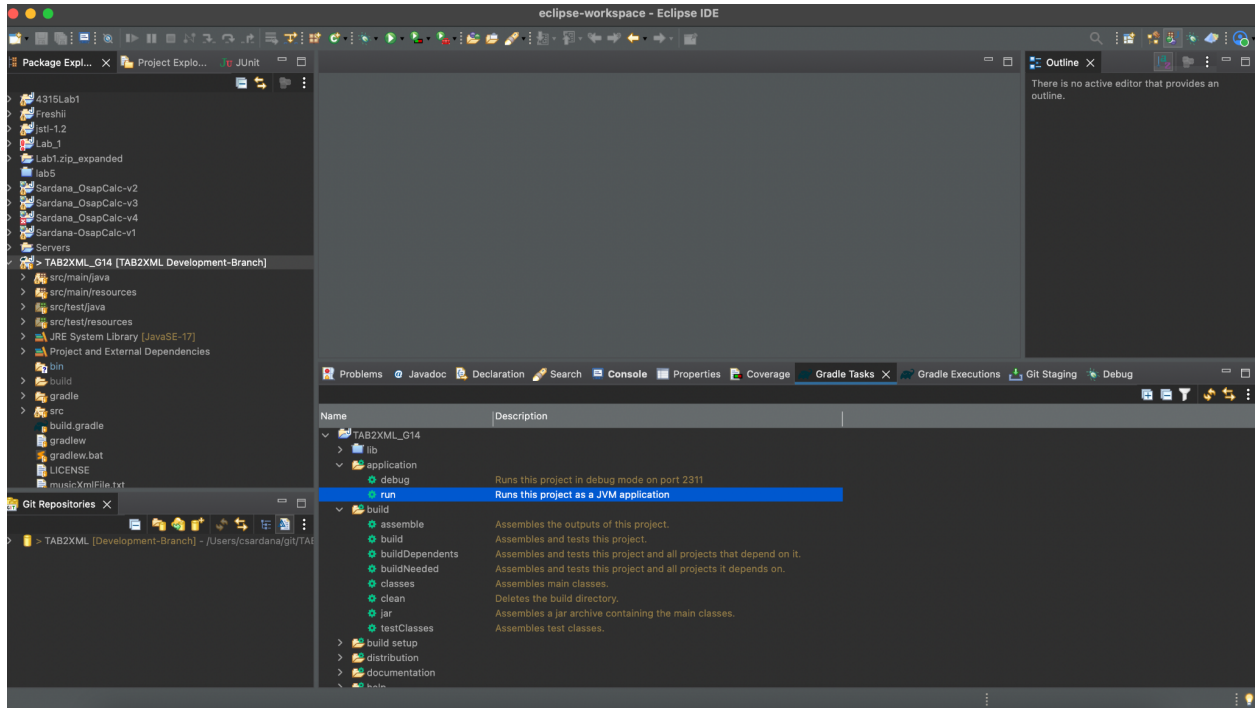
In order to run the application a user has to download the source code from <https://github.com/csardana/TAB2XML> . The code can only be downloaded from the master branch which contains the latest functionality. Copy this link above into the URI and click the **finish** button (see image 4).



(Image 4 - Import Project from Git interface)

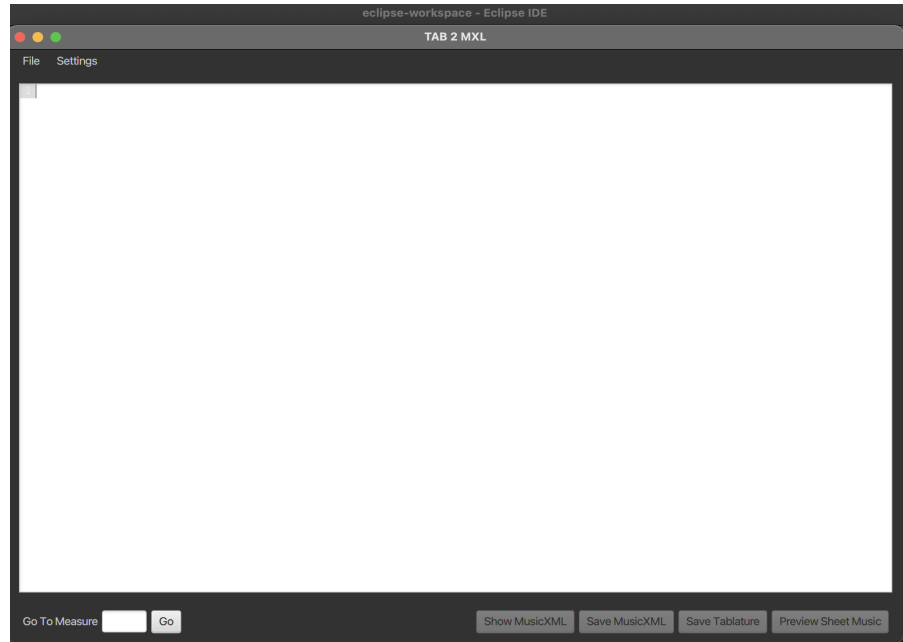
When filling in the Authentication, we have to use your username of github and password that you use to login with, if you don't have a github account, please sign up for one. Proceed by clicking on the **Finish** button. The code needs to be run on Eclipse IDE. It needs to be built by grade tasks . Select build and run the application (see image 5).

Folder → Application → Run (double click)



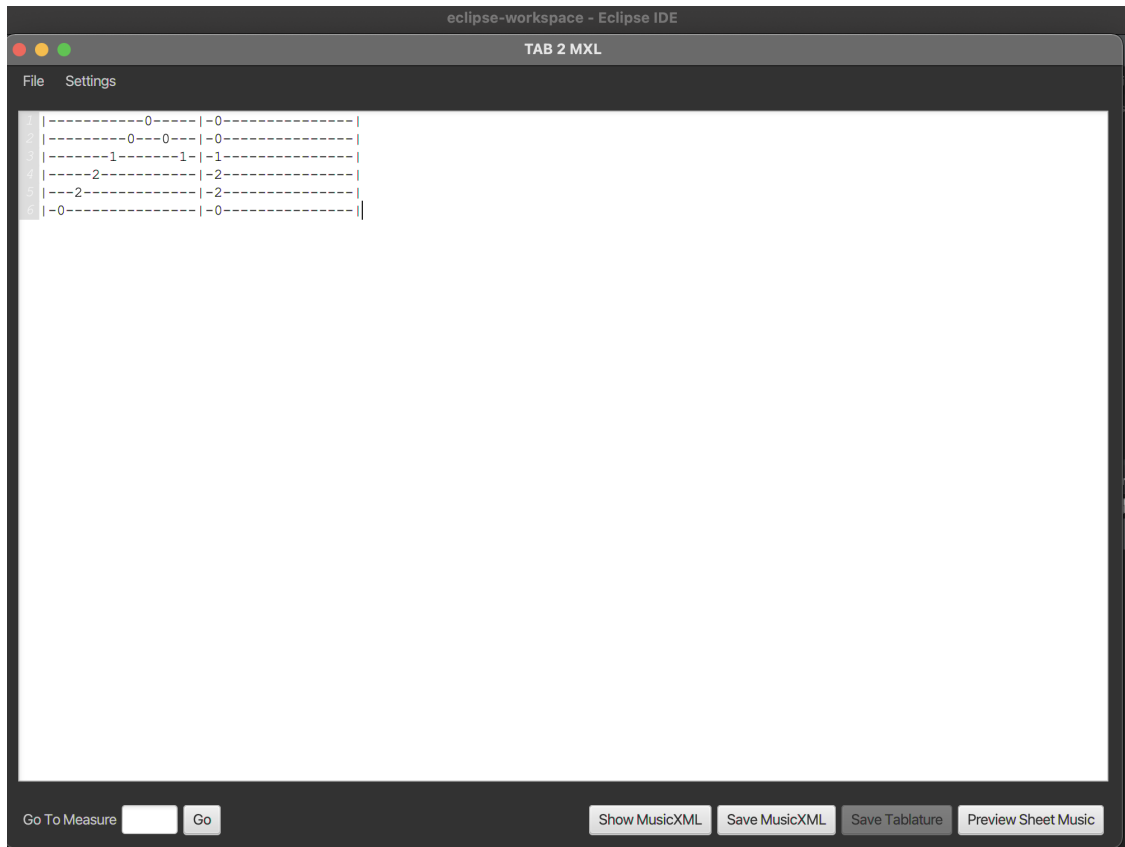
(Image 5 - To run the application through Gradle)

A Java graphic user interface (GUI) will appear on the screen once Gradle launches the application (see image 6).

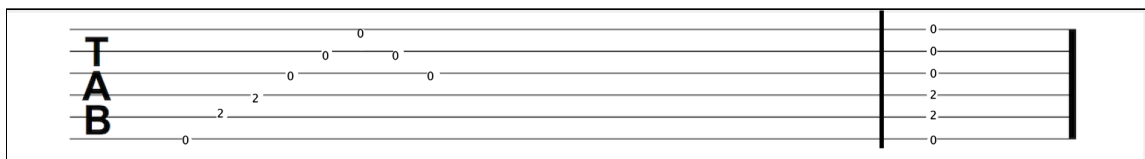


(Image 6 - Java GUI after Gradle launches the application)

Paste the any text-based tablature and click the **preview** button (see image 7 and 8).



(Image 7 - An acceptable guitar tablature in the text area)



(Image 8 - The resulting preview of a guitar tablature as a music score)

3. Exit the System

In order to exit the system, you will have to close it by clicking the “X” button situated at the top left for MacOS users or top right for Windows users.