Tab2XML Music Score

User Manual

Version 1.1 (Prototype)
2022-03-06

Group 7:
Aleksander Weinberger
Harsimran Saini
Chirag Sardana
David Hanna

Table of Contents

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).

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.

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 the Java programming language which uses eclipse IDE. It also uses gradle build to build and run the application.

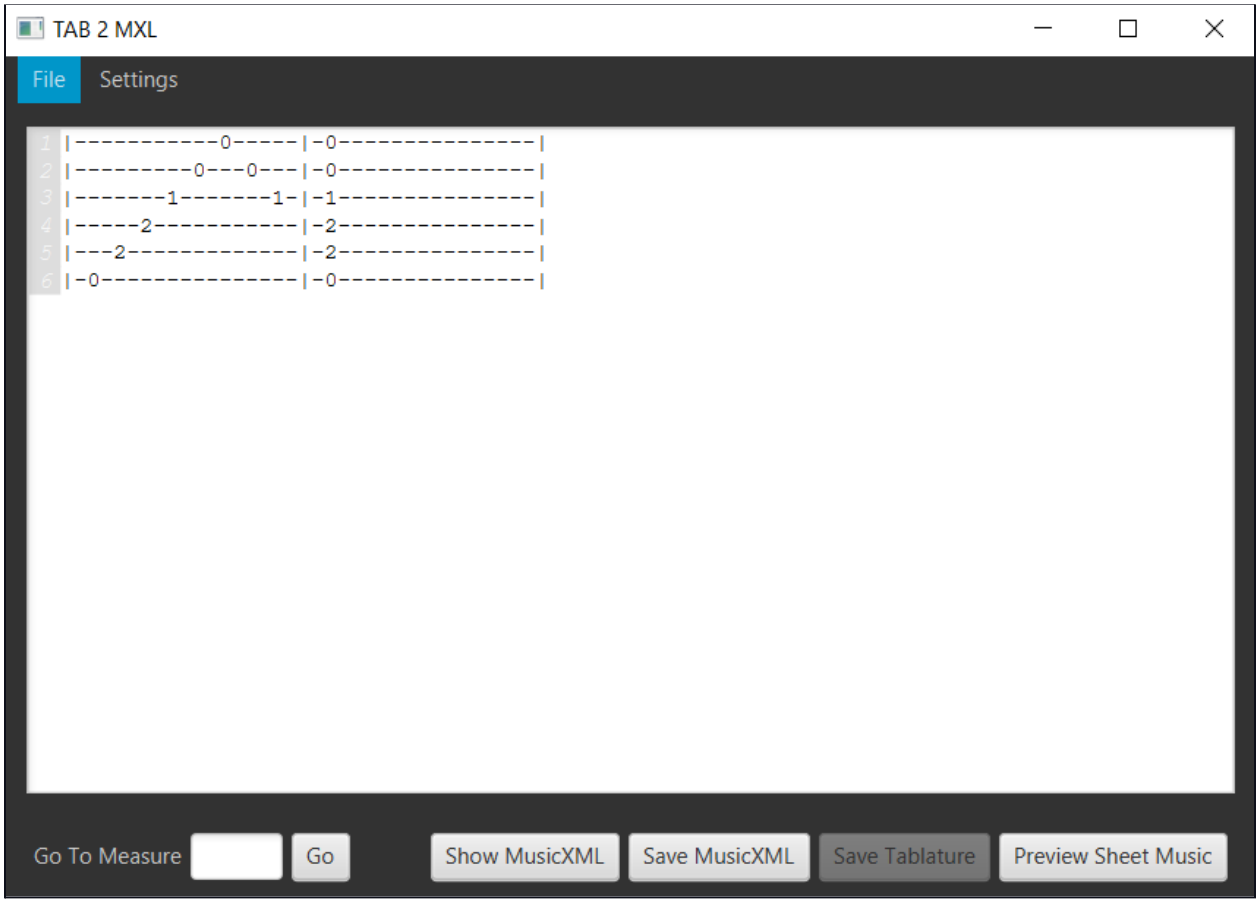
Conventions

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)

Getting Started

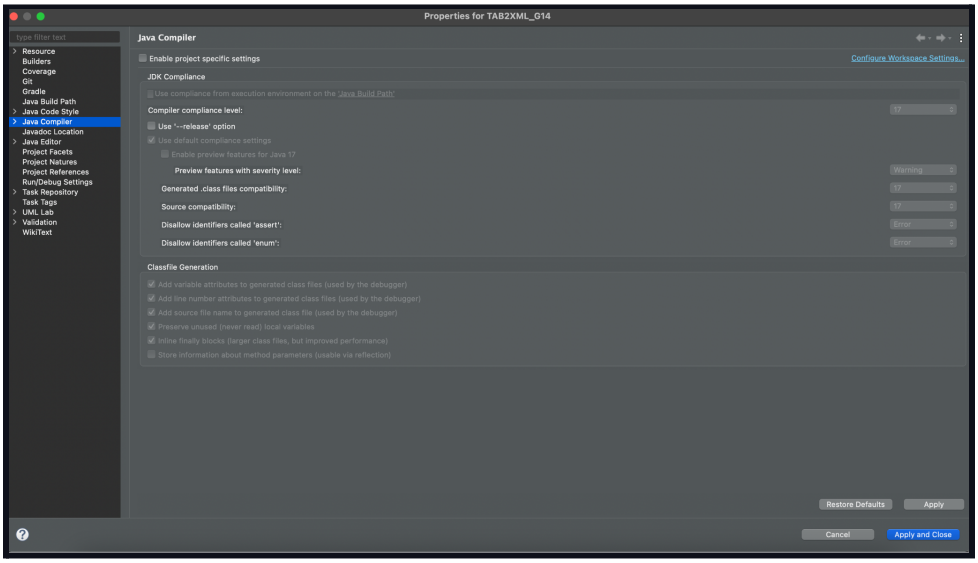
1. System Requirements

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

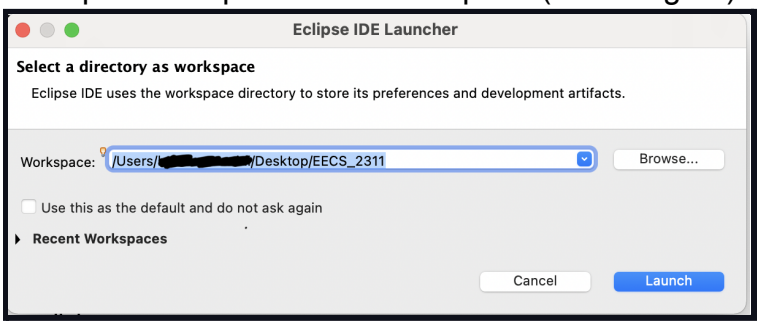
In Eclipse IDE the compiler settings for Java looks like the following (see image 2).



(Image 2 - Compiler settings in Eclipse IDE)

2. Run the System

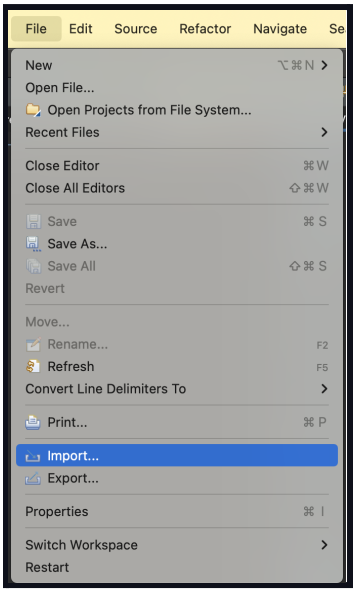
Open eclipse and open a new workspace (see image 3).



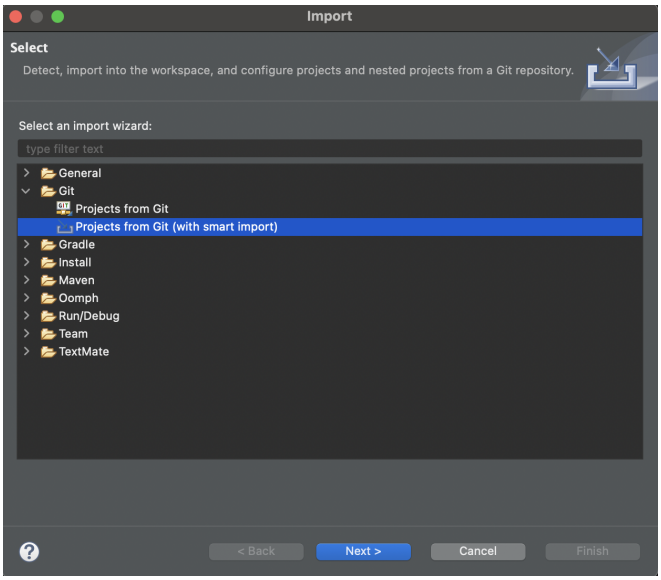
(Image 3 - Eclipse IDE Workspace selector)

Now, once eclipse is opened, then

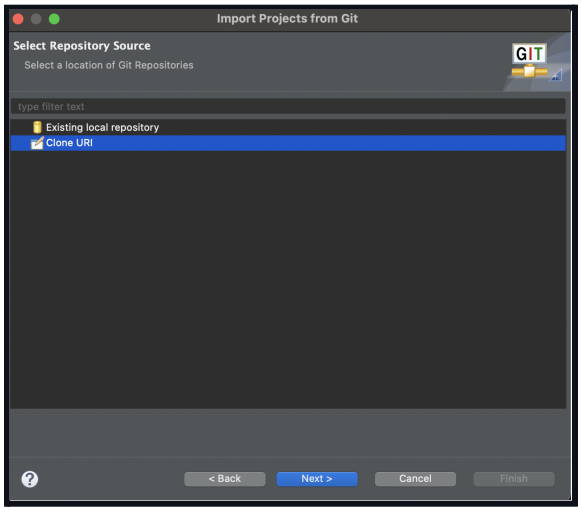
Go to **File → import → Git → Projects from Git → Clone URI**



(Image 3.1 - Importing file)

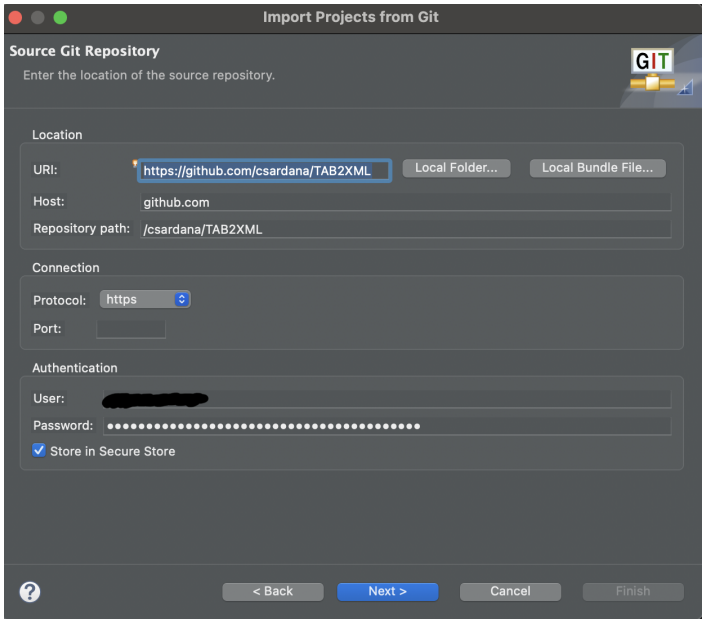


(Image 3.2 - choosing project from git)



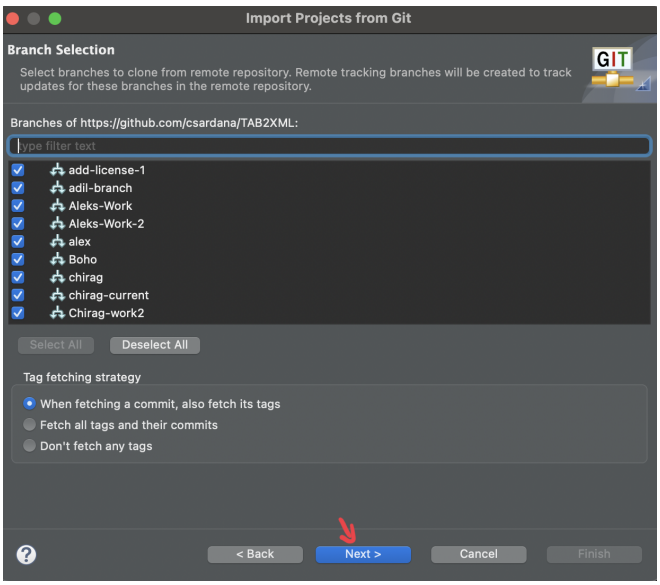
(Image 3.2 - cloning URI)

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 **next** button (see image 4).



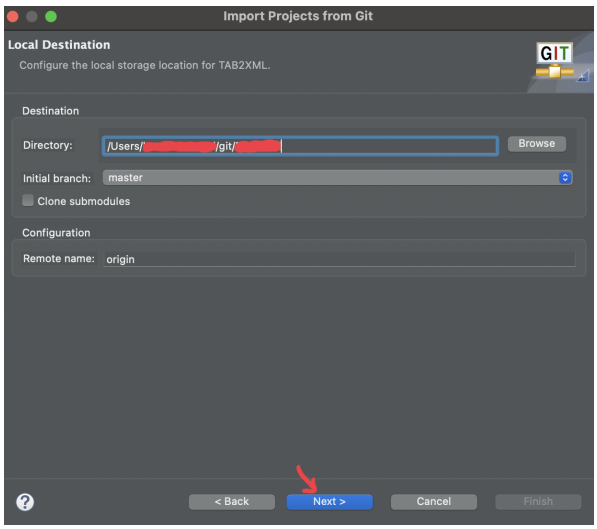
(Image 4 - Import Project from Git interface)

When filling in the Authentication, we have to use your username of github and for password that you use is the **personal access token created in your github.** If needed instructions on how to create personal access token please click here.
Proceed by clicking on the **next** button.



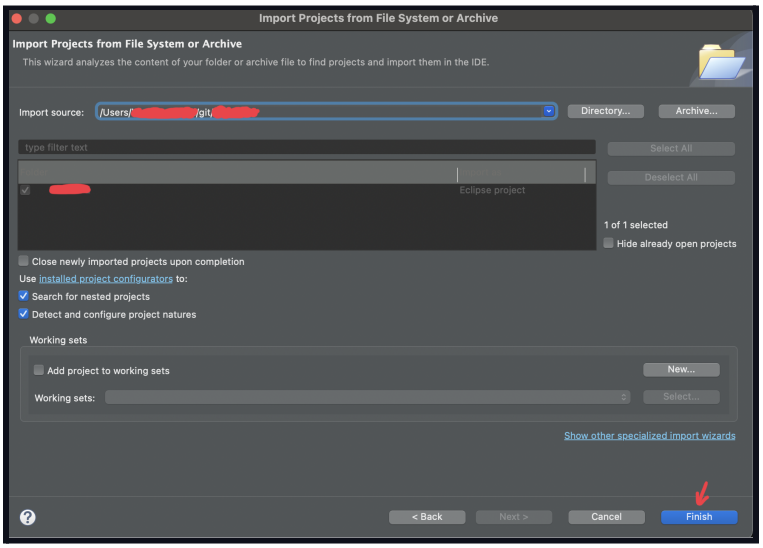
(Image 4.1 - Import Project from Git interface)

For this step make sure the project is being stored in the correct path and then may proceed further by clicking **next.**



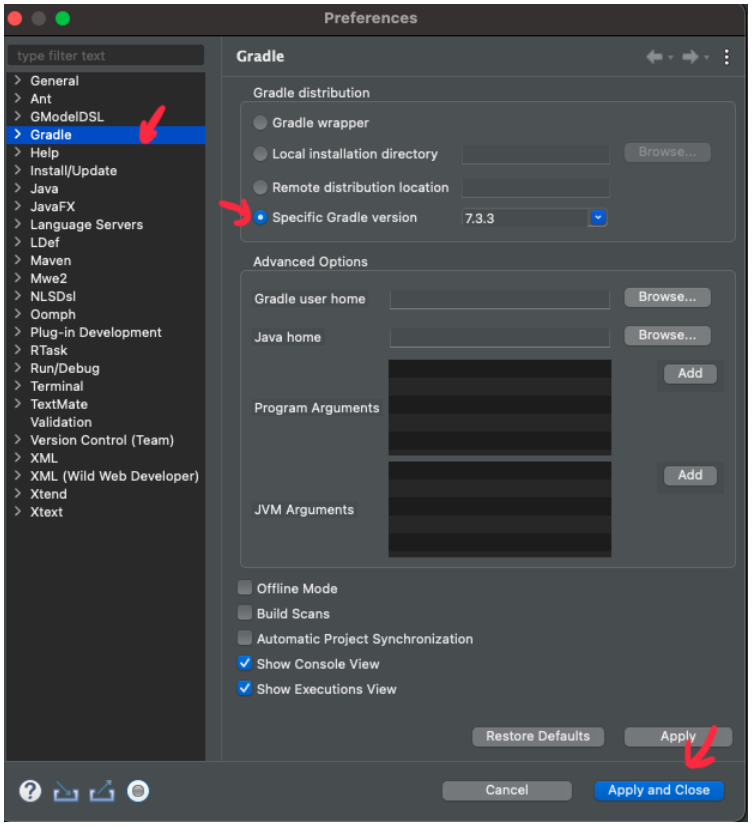
(Image 4.2 - Import Project from Git interface)

After seeing everything is in order, you may finish importing and the project will import into your workspace.



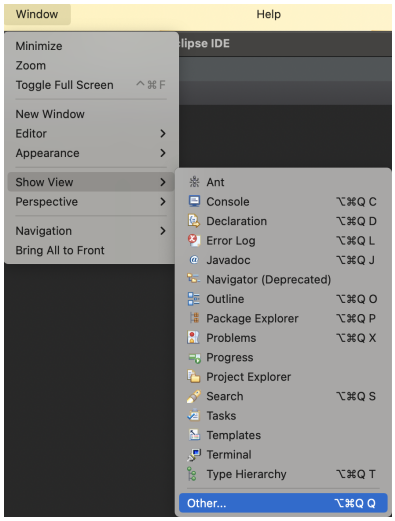
(Image 4.3 - Import Project from Git interface)

The code needs to be run on Eclipse IDE. It needs to be built by grade tasks. Firstly we need to see the specific gradle version. So go to preferences → Gradle → select specific Gradle version.
Note: The Gradle version has to be 7.3.3.



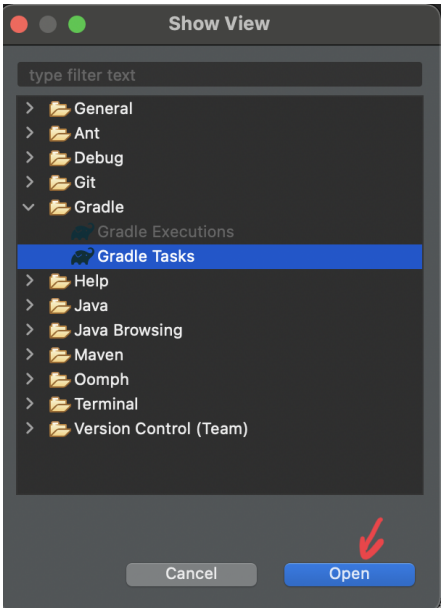
(Image - 5 To run the application through Gradle)

To get gradle tasks so the application can run, "Run → Show View → Other" (see Image 5.1).



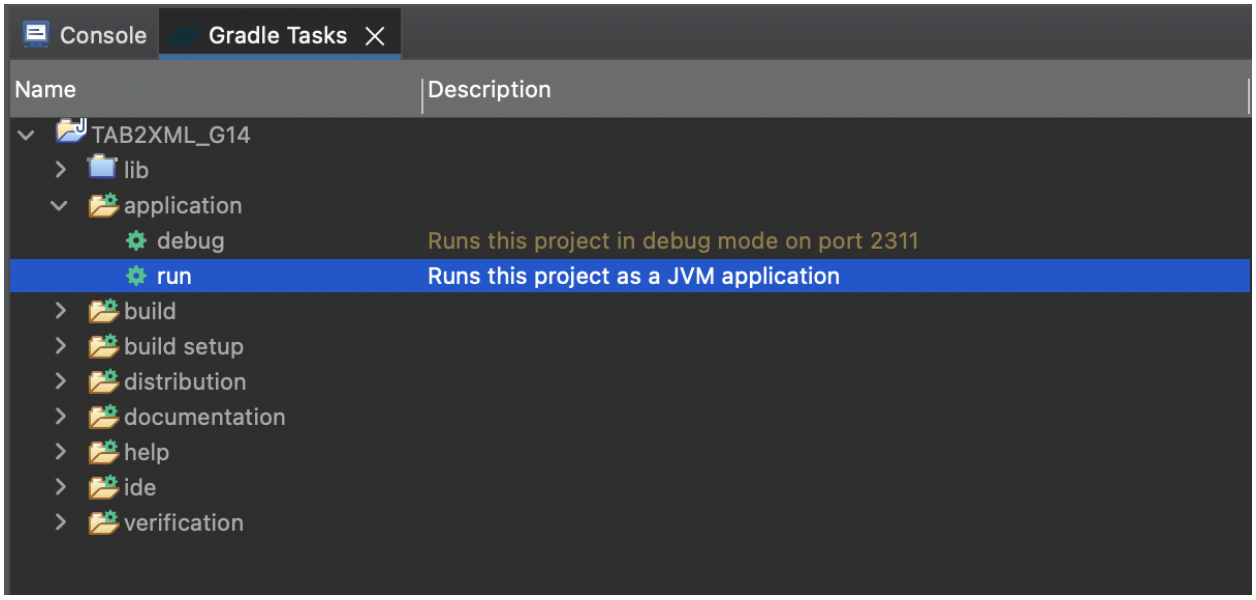
(Image 5.1 - To run the application through Gradle)

Now in the show view select "Gradle -> Gradle Tasks"



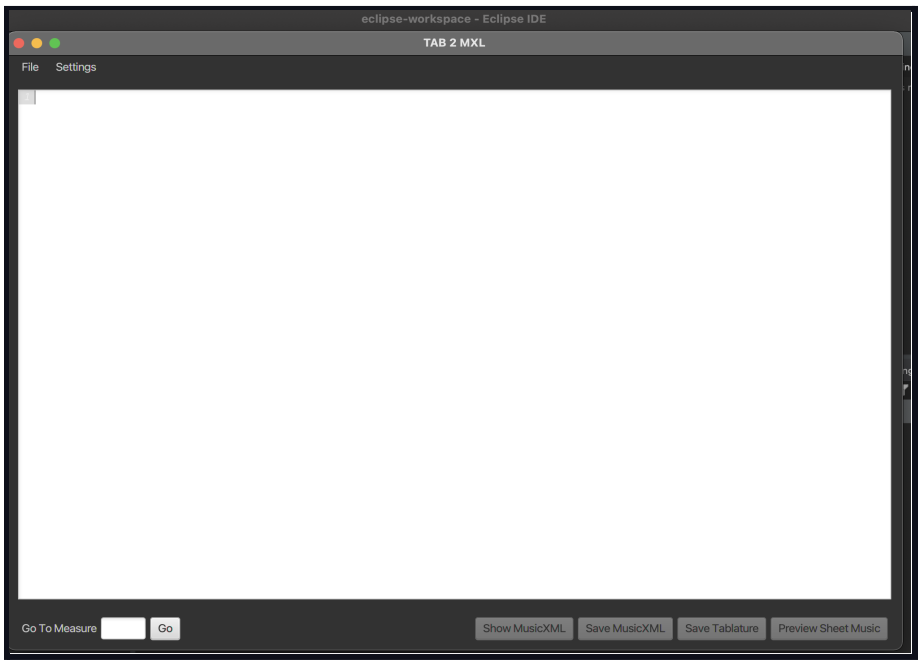
(Image 5.3 - To run the application through Gradle)

After doing these steps, there would be a window named Gradle tasks and underneath that the project name "Project_name -> application -> run"
Note: Double click "run"



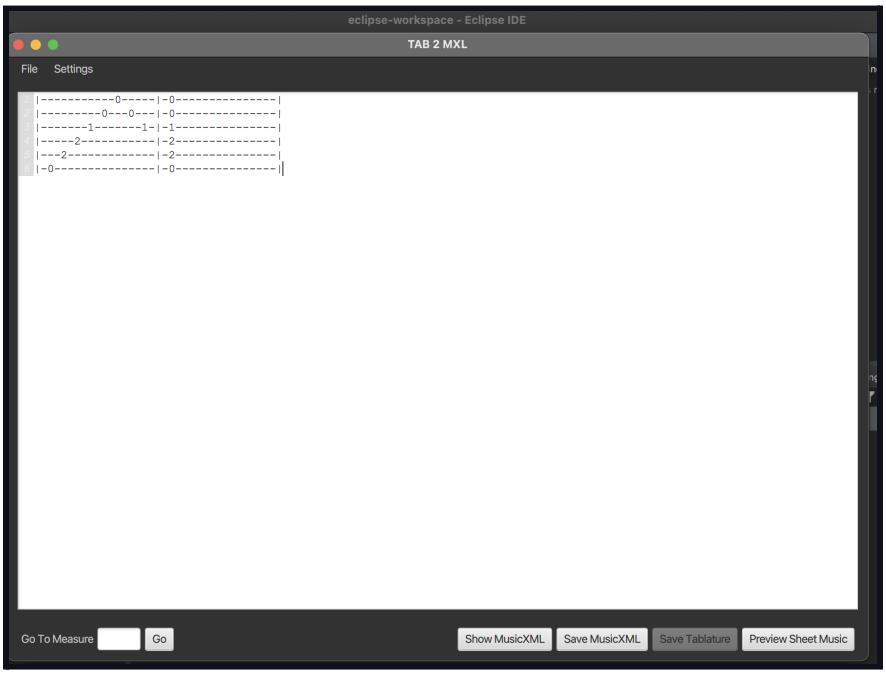
(Image 5.4 - To run the application through Gradle)

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

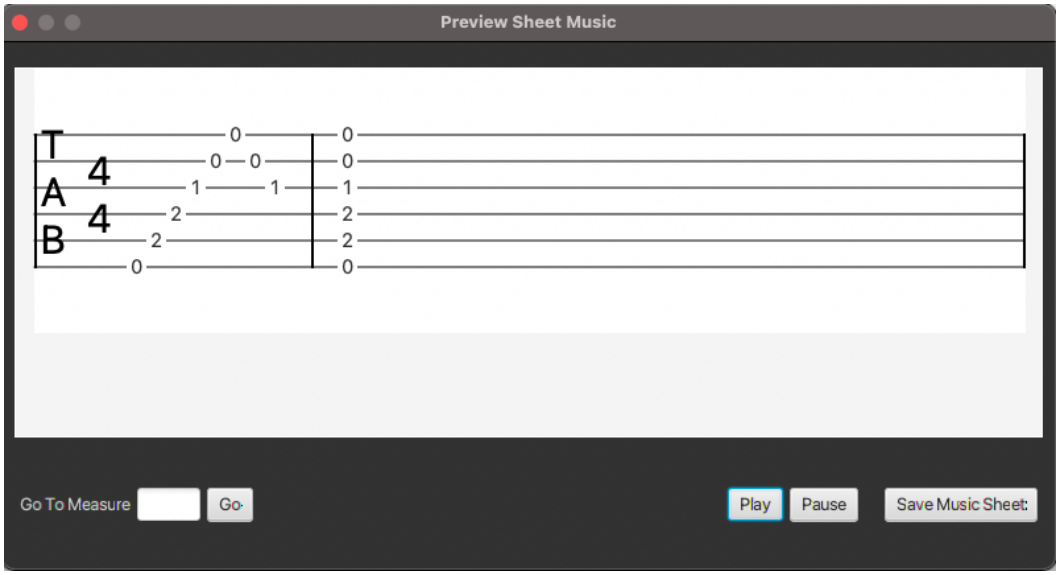


(Image 6 - Java GUI after Gradle launches the application)

Paste 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)

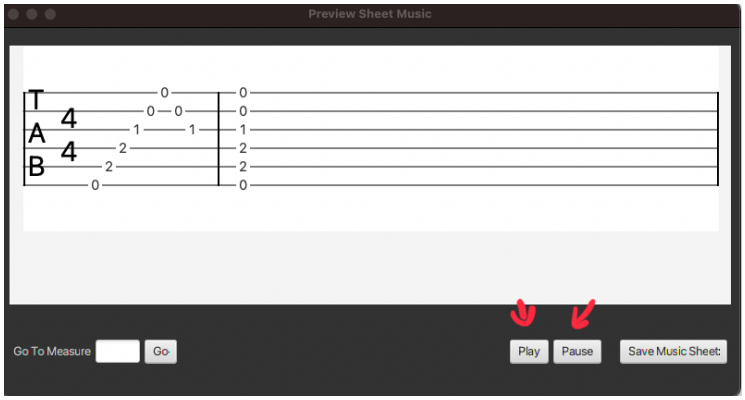
Exit the System

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.

Features

Play Music

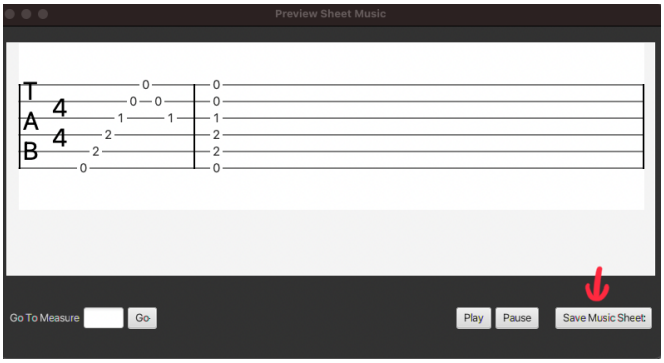
In the Preview Music Sheet window, there is a play and pause button at the bottom right which allows the user to play the sheet music on a virtual instrument. The type of virtual instrument used to play the music sheet is dependant on the tablature’s instrument the user supplied the application. It should be noted that the pause button can only be used when the user clicks the play button, and pressing the play button after pausing restarts the play function.



(Image 9.1 - Play and Pause Music Buttons)

Save Music Sheet

This feature allows the user to save their music sheet as a *.png file on their computer.
Note: The save *.pdf file is under development.



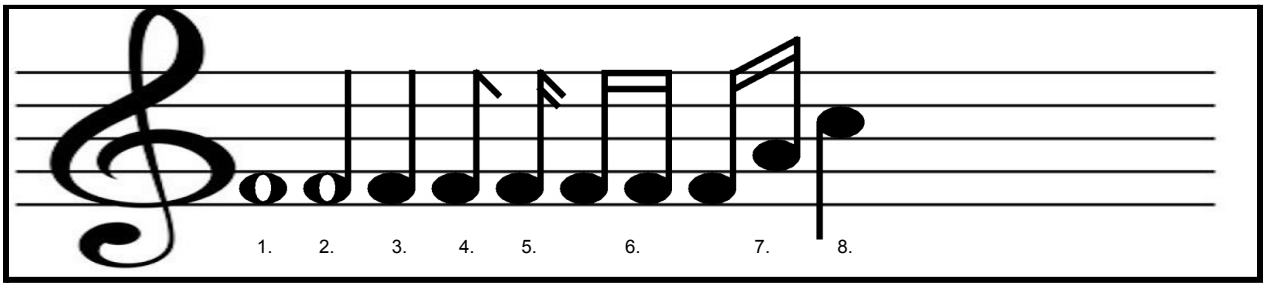
(Image 9.2 - Save Music Sheet)

Limitations

- In the latest version of the application, the software does not support previewing a drums music sheet as this feature is still in development.
- For a provided guitar tablature, the application does not support pull-off's, hammer-ons, or any other advanced guitar techniques.
- The go to measure function in the preview sheet window is not supported is still under development.
- Music sheet's can only be saved as .png images, but options of choosing other formats such as PDF and .jpg are in development.
- There is no functionality to resume playing the music sheet once paused.

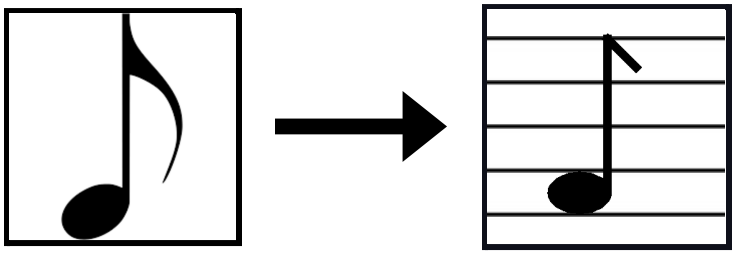
Coming Soon…

TAB2XML to display drum tablature in the treble clef as seen in the image below



The above picture describes how we plan to display different types of notes using Java GUI. Please note that this is a feature still in development. Notice how for the lone eighth and sixteenth notes they have a line striking out as the flag. Each note stem is the length of the bar. Please see Java Graphics API for more info on the implementation of this GUI.

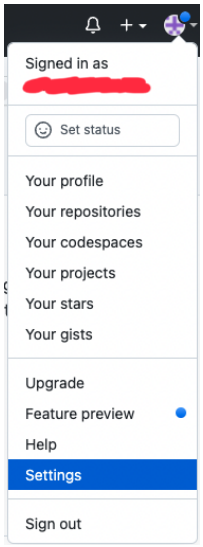
Number	Note
1.	Whole.
2.	Half.
3.	Quarter.
4.	Eighth.
5.	Sixteenth.
6.	Barred 16th.
7.	Slanted-Barred 16th.
8.	Upside Down Quarter.



How to Create a Personal Access Token Via Github

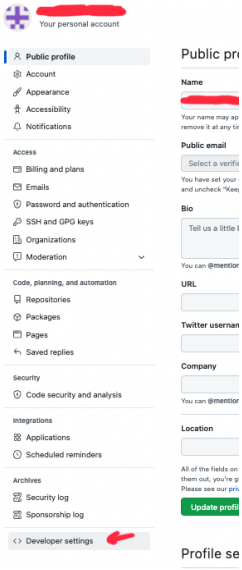
This Personal Access Token will be used when importing our project into eclipse via Github. This portion will take you step by step in creating the Personal Access Token.

Go to your github settings



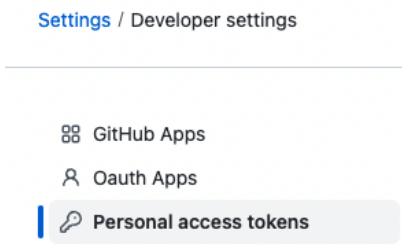
(Image 10.1 - Creating Personal Access Token)

Scroll down and select "Developer settings" in the side menu bar.



(Image 10.2- Creating Personal Access Token)

Now from these three options select "Personal access tokens"

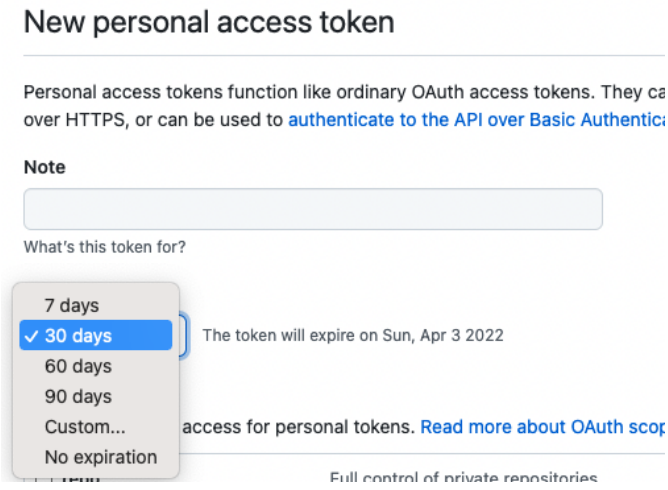


(Image 10.3 - Creating Personal Access Token)

To generate a new token, select "Generate new token"

(Image 10.4 - Creating Personal Access Token)

Now it will prompt you to enter your password for verification, after doing that, select a date for how long you want this token to work for and make a note for what this token is for. Then scroll all the way at the bottom to click generate.
Note: When the token is generated, copy and paste it somewhere (e.g notes or notpad) so you can enter when importing a project from github.



(Image 10.5- Creating Personal Access Token)

Note: You will see many other options, but we will not select any of those and just select "Generate key" at the bottom of the page!!