Project Requirements

ℹ️ **EECS 2311 W22 - Group 9**

Members:
- Vivek Wadhwani (*vivek121@my.yorku.ca*)
- Kris Singh (*ksingh7@my.yorku.ca*)
- Kingsley Okon (*King808@my.yorku.ca*)
- Lan Zhang (*zhanglala0208@gmail.com*)

Project: github.com/devivekw/TAB2XML

1 Requirements
    1.1 Functional Requirements
    1.2 Non-Functional Requirements
2 Uses
    2.1 User Stories
    2.2 Use Cases

1 Requirements

This section will detail in which capacity the application will be able to work and its intended functionalities.

1.1 Functional Requirements

The functional requirements of the application are the features present and what the user can do within the application.

- The user must be able to input valid guitar or drum tablature into the application.
- The user should be able to choose which features they wish to use through the push of a button.

- The user must be able to convert the provided tablature into sheet music.
- The user must be able to play the music of the tablature they provided.
- The user must have access to playback features such as play and pause.
- The user should be able to change the beats per minute of the playback.

1.2 Non-Functional Requirements

The non-functional requirements of the application are less technical than their functional counterparts, but are important nonetheless.

- The application should be user friendly and easy to understand.
- There should be a user manual to explain the application to new users.
- The application should be well tested to ensure it works in the correct user scenarios.

2 Uses

As there exist requirements for this application, user stories and cases are important for understanding the requirements and the functionality of the application.

2.1 User Stories

Different user stories help the development team envision what the intended uses are for the application.

1. A music teacher finds tablature for a song they would like their students to play.
    a. In this scenario, the teacher will require the ability to convert the tablature into readable sheet music for the students.
    b. In addition, it would be useful for the music teacher to be able to play the music for the students so that they have a better understanding of the song.
    c. If the students are continuously practicing, the playback feature will be useful as it will allow them to rewind back if they need to practice a specific part.
2. A musician who creates their own tablature wants to easily convert their work into sheet music.

a. The musician will be able to enter their tablature and convert it into the desired sheet music.

b. They will be able to listen to their song before actually attempting to play it in person, and can use the beats per minute functionality to determine what bpm their song should be.

2.2 Use Cases

Similar to the user stories, the use cases will help the development understand the technical aspect of what the users should be able to do.

1. The user wants to play the provided tablature.
    a. The user launches the application and is greeted by the launch screen.
    b. They either paste in the tablature or import a text file containing it.
    c. The user clicks on the `Play Music` button.
    d. The application launches the Play Music window.
    e. The user clicks on the play button.
    f. The application plays the music until the user decides to press the pause button.
    g. The user can customize the bpm using the bpm slider which determines the length of the music being played.
    h. The user finishes playing and closes the application.
2. The user wants to preview the sheet music of the provided tablature.
    a. The user launches the application and is greeted by the launch screen.
    b. They either paste in the tablature or import a text file containing it.
    c. The user clicks on the `Preview Sheet Music` button.
    d. The application launches the Sheet Music window.
    e. The user is able to preview the sheet music for however long they wish.
    f. Once they are satisfied, the user will close the application.