Da Capo

Requirements Document

Version 1.0
Date: 02/28/2021

Gong-Fan (Billy) Bao - 213563150
Syed Quadri - 216204554
Elijah Nnorom - 214502504
Miguel Mesa Gonzalez - 216138323
Ayub Osman Ali - 217612847

Table of Contents

1. Introductions

1.1 Purpose

Da Capo serves to help musicians with the conversion of tablature into the musicXML format. It offers an easy to use interface and the ability to convert text files with various tab formats for multiple instruments.

1.2 Scope

Da Capo takes a text file containing music tablature for drums, guitar or bass and translates this into the musicXML format. The user is given the option to save the new text file and can Then feed it into a program to be played. The system requirements dictate that Da Capo Must accept, display and allow the user to edit a text file containing tablature for drums, bass Or guitar. Then, the system must produce a correct musicXML translation which can be saved and renamed by the user. In the case a translation fails or the file saving is cancelled, the system will display an error message.

1.3 Definitions, Acronyms and Abbreviations

Term	**Definition**
Simple Guitar Tab/Tablature	A Guitar tablature that contains no double-digit frets as well as no special symbols. An example can be found in the Da Capo User Manual.
Bass	Abbreviation of Bass Guitar
GUI	Acronym for Graphical User Interface
MusicXML	XML-based file format for representing Western music notation
Tab	Abbreviation of Tablature

Tablature	A form of musical notation with ASCII characters
.txt	Extension of a plain text file

1.4 References

Da Capo User Manual
https://docs.google.com/document/d/1IhHdD-Nd9ZIJpAWSL3IuWYTfuE3LoL_WjTELXTAvqtw

EECS2031 Course Website
https://wiki.eecs.yorku.ca/course_archive/2020-21/W/2311/proj

2. Software Description

2.1 Product Perspective

Da Capo was built for the EECS2311 Software Development Class at York University in 2021. The aim of the project was to develop a software system that allows the user to input a text file containing the guitar, bass, or drums tablature for a song, and produces a MusicXML file, which can be used in other software.

2.2 Product Features

In the future, Da Capo will be able to convert more complex Guitar Tabs, as well as Bass and Drum tabs. Da Capo will also prompt the user for input when it cannot translate a given tablature, instead of only displaying a generic error message. Da Capo will also give the user the option to choose between two different formats for Drum musicXML.

2.3 User Characteristics

Da Capo's intended user is able to recognize music tablature, knows how to copy and paste, can navigate through a computer's files and has very basic knowledge of how to use java or eclipse. This might limit the product's audience to those who have some programming experience or general computer knowledge.

2.4 Assumptions and Dependencies

Da Capo assumes that the Operating System that it is installed on has Eclipse with Gradle installed, and can run Java.

3. Specific Requirements

3.1 Functional Requirements

3.1.1: The system must accept a text file containing tablature, given by the user.

3.1.2: The system must display the plaintext version of a text file, if given by the user.

3.1.3: The system must allow the user to edit the paintext displayed within the Tablature Editor display.

3.1.4: The system must accept tablature formatted for guitar, bass guitar and drums.

3.1.5: The system must produce a translated MusicXML file of the given tablature.

3.1.6: The system must produce a correctly translated MusicXML file, if the given tablature is a simple Guitar tablature.

3.1.7: The system must allow the user to customize the name of the translated MusicXML file.

3.1.8: The system must allow the user to save the translated MusicXML file to a location of their choosing.

3.1.9 The system must present an appropriate error message to the user if the file saving action is canceled.

3.1.10: The system must present an appropriate error message to the user if it is unable to translate the given tablature.

3.2 Performance Requirements

3.2.1: The system should be able to perform its function within 5 clicks, at minimum.

3.2.2: The system should be able to translate tablature to MusicXML within 1 second.

3.2.3: The system should return an error message whenever there is an error or failure.

3.3 Other Requirements

3.3.1: The system should be compatible with any Operating System that can run Eclipse (with Gradle) and Java.

3.3.2: The system should have text that is legible to most users.

4. Use Cases

4.1 Converting Copied Plaintext Guitar Tab

Actor: User
Success Scenario:

1. User inserts the copied guitar tablature into the Tablature display.
2. System displays the guitar tablature as text on the screen, and identifies the tablature as a guitar tablature.
3. User clicks the “convert” button.
4. System converts the guitar tablature into MusicXML.
5. System creates a file on the User’s computer containing the translated MusicXML.
6. User locates file and closes System.

Precondition: User found a guitar tablature somewhere online and has it saved to their clipboard on their computer.

4.2 Prompting for More Information

Actor: User
Success Scenario:

1. User selects the drum tablature file from their computer.
2. System displays the drum tablature as text on the screen, and identifies the tablature as a drum tablature.
3. System fails to determine the time signature for the tablature, and creates a prompt for more information.
4. User enters the correct time signature into the prompt.
5. System takes the information inside the prompt and reprocesses the tablature.
6. User clicks the “convert” button.

Precondition: User is aware of types of instruments and has a drum tablature stored in a file on their computer.

4.3 Overriding an Incorrect Instrument Interpretation

Actor: User
Success Scenario:

1. User pastes the tablature into the application.
2. System correctly identifies the guitar tab as a drum tab.
3. User is given the option to choose the correct type of tab.
4. User clicks the convert button
5. System converts the tab to musicXML
6. System creates a musicXML file on the user's computer
7. User locates file andcloses system

Precondition: User found a guitar tablature somewhere online and copied it into the application.

4.4 Modifying Text Tablature Directly before Conversion

Actor: User
Success Scenario:

1. User selects text file or pastes a snippet of text into the GUI
2. Before converting the file, the user would like to modify the music tab by changing the text
3. User will directly modify the tab in the GUI and then begin the conversion process
4. System begins the conversion process, if the tab has become unreadable (incorrect formatting), the system will alert the user.
5. User can correct formatting and attempt to convert once again
6. System creates a MusicXML file once conversion is possible.

Precondition: Musician has acquired a tab, had knowledge of how to read and edit the tab, and is using the app completely aware of its features.