Design Document

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to visualize the system of this application using UML class diagrams and sequence diagrams. This document follows closely with the requirements document in order to meet all application requirements.

1.2 Scope

The Music Assistant is a web-based application that allows choir members (students) to practice their music. The main goal of the application is to provide real-time graphical feedback for pitch, duration, dynamic, tempo, rhythm, and phrasing. This main goal differentiates us from all other existing applications. We would also like to include features from existing applications such as statistical analysis and communication between users. Choir instructors (teachers) can communicate with students for assistance.

We are focusing on a web application so that many users can have access to an uncompromised, cohesive experience. The application will be available to desktop, tablet, and smartphone users alike. In order to group members of a choir and to protect the copyright of pieces of music, users

will authenticate before they can use the application. Once authenticated, students can practice their music, work on practice exercises, and communicate with their teacher. The teacher will also authenticate, but will have access to student statistics and will be able to communicate with all of their students through a messaging system.

1.3 Overview

This software design document describes the system and architecture design of The Music Assistant. The document provides details on how the software will be constructed. The document describes these details using use cases, sequence diagrams, class diagrams, and textual descriptions.

1.4 Reference Material

https://sovannarith.files.wordpress.com/2012/07/sdd_template.pdf

1.5 Definitions and Acronyms

Web Application (Web App): A cross-platform application accessible on the web using web technologies such as HTML, CSS, and JavaScript.

Duration - an amount of time or how long or short a note, phrase, section, or composition lasts

Dynamic - the volume of a sound or note

Pitch - the degree of highness or lowness of a tone

Phrasing - the manner in which a musician shapes a sequence of notes in a passage of music, in order to express an emotion or impression

Rhythm - the pattern of regular or irregular pulses caused in music by the occurrence of strong and weak melodic and harmonic beats

Tempo - the pace or speed at which a section of music is played

2. System Overview

The Music Assistant was born out of a frustration of choir members who could not easily practice their piece of music outside of designated practice time with a choir instructor. While choir members have always been able to practice their piece at home, they have had no way to determine how well they are doing beyond basic intuition. The Music Assistant changes that - choir members will be able to practice their piece with real-time feedback for pitch, duration, dynamic, tempo, rhythm, and phrasing. This real-time feedback allows the student (choir member) to fix their problems as they sing the piece. Algorithms will show where the student is in relation to the actual music note at any given time. The more a student practices, the better they will become, and this will be shown visually through the use of trend graphs and charts. Not

only will these charts be visible to the student, but the teacher will also be able to view all students' performances. An integrated communication tool will allow students to chat with their teacher about their current performance and struggles.

3. System Architecture

3.1 Architectural Design

Music Assistant ...ture Diagram.pdf

3.2 Decomposition Description

The User class is the superclass of Teacher and Student. The User class contains the user's unique identifier (id) and email. The id will be used to uniquely identify a user for database access permissions. The email is used for authentication and any communications that use email. Each user will also be linked to one Name instance. Each user can only have one name, but their first name and/or last name can be changed at any time. Each user will be linked to 0 or more Conversation instances. This link from User to Conversation allows a user to see all of the conversations that they've been in.

The Teacher class inherits from the User class. Each teacher teaches 0 or more students, and therefore has a link to each Student instance that they teach. Each teacher instructs 0 or more choirs. Each choir that a teacher is part of is also linked back to them. Each Choir has 1 or more teachers.

The Student class also inherits from the User class. Each student has 0 or more teachers, and therefore has a link to each Teacher instance that teaches them. Each student can be in 0 or more choirs, and therefore has a link to each Choir instance that that they are a part of. Each choir that a student is part of is also linked back to them.

The Choir class has a name string and a description string. Each choir is linked to 1 or more teachers and 1 or more students. Each choir contains 0 or more music pieces. Due to copyright laws, we do not intend to share pieces of music across choirs.

The Conversation class is contains a created time attribute which holds a date and time. Each conversation contains 0 or more messages. Each conversation is created by one User instance, and therefore has a link to the user. Each conversation also contains one student and one teacher. Note that students can only message their teachers and that each conversation only contains two people. Each conversation also contains 0 or more MusicPiece attachments.

The Message class contains the message text as a string, the sent time (DateTime format), and an isDelivered boolean which indicates that the message was delivered to the other user's device. Each message is sent by one user, and therefore has a link to the user that sent the message.

The MusicPiece class contains the piece's title as a string, a string description of the piece, a release date (DateTime format) and a copyright string. Each piece has 1 or more composers. We only store the composer's name in the Name class. In order to hold choirs responsible for copyright infringement, each piece of music is used by only one choir, and each piece has a link to the choir that uses it.

The Name class contains a first name string and a last name string.

3.3 Design Rationale

No other architectures were considered.

4. Data Design

4.1 Data Description

When a user logs into the web application, the system will determine if you are a student or a teacher. Your id, name, and email will be fetched from browser storage or the database. Your id and email will be stored in a User class and your name will be stored in a Name class. The User class will reference the instance of the Name class. A type variable will hold "student" if the user is a student or "teacher" is the user is a teacher. Note that there are subclasses Teacher and Student in the System Architecture diagram, but they are not currently being used for any reason other than for visual reference.

If a user opens their conversations, all conversations will be retrieved from the database. Each conversation is added to an instance of the Conversation class and the createdTime variable is initialized to the correct value. The last message to be sent or received from each conversation is retrieved from the database, along with the other user's id, email, and name. If the user opens up a specific Conversation, all other messages (other than the first message) from that conversation will be retrieved. Each Message instance will hold the message text, sentTime, and isDelivered boolean. Note that each Message instance is added to a list of messages in the correct Conversation instance. This list (array) of messages will hold references to each Message instance for that conversation.

If the user opens up the "Choir" tab, all choirs that they are in will be retrieved from the database. For each choir, a new Choir instance will be created. Each Choir instance will contains the choir ID, name, and description.

If the user selects a specific choir, the system will need to retrieve all pieces of music associated with the Choir instance from the database. Each MusicPiece instance will hold the title, description, release date, and copyright information for that piece of music. The system will also need to retrieve the name of the composer of each piece of music. Each composer's first name and last name will be stored in Name class instances. Each MusicPiece instance will have a reference to the corresponding Name instance.

Notes:

1. There are still pieces of data that we need to determine how to store. These include
    1. The actual piece of music (string representation [AlphaText] or file?)
    2. The statistics for a student
    3. A piece of music that is drawn on digitally (used in conversations)
    4. Gradebook for a teacher

4.2 Data Dictionary

1. User
    1. id: String
    2. email: String
    3. profilePicture: Image
    4. getID(): String
    5. getEmail(): String
    6. setEmail(String)
    7. getProfilePicture(): Image
    8. setProfilePicture(Image)
2. Name
    1. firstName: String
    2. lastName: String
    3. getFirstName(): String
    4. setFirstName(String)
    5. getLastName(): String
    6. setLastName(String)
3. Choir
    1. id: String
    2. name: String
    3. description: String
    4. getId(): String
    5. getName(): String
    6. setName(String)
    7. getDescription(): String
    8. setDescription(String)

4. MusicPiece
    1. title: String
    2. description: String
    3. releasedDate: DateTime
    4. copyright: String
    5. getTitle(): String
    6. getDescription(): String
    7. getReleasedDate(): DateTime
    8. getCopyright(): String
    9. setCopyright(String)
5. Conversation
    1. createdTime: DateTime
    2. getCreatedTime(): DateTime
6. Message
    1. text: String
    2. sentTime: DateTime
    3. isDelivered: Boolean
    4. getText(): String
    5. getSentTime(): DateTime
    6. getIsDelivered(): Boolean

5. Component Design

Current component design is simplistic and all method algorithms are trivial. This will change as we determine the complex parts of this product (e.g. music representation, real-time feedback, and statistics).

6. Human Interface Design

6.1 Overview of User Interface

When the user enters the web application and there is no log in data stored, they will be presented with a log in screen. This screen will allow them to log in with an email and password or use a third-party authenticator (e.g. Facebook or Google).

If the user does not already have an account, they will have the option to sign up. The sign up process requires the user's email and name at a minimum, as well as a password. If the user chooses to sign up with Facebook or Google, they will have no required information to enter. We will be able to get their email and name from the respective API. The user will also be able to upload a profile picture if they choose to do so. The user will then select if they are a student or a teacher. If the user is a student, they will then select the Choir that they are in by entering their choir id. If the user is a teacher, they will then create a new choir and receive a choir id that they will pass along to their students.

Once the user is authenticated, they will be brought to the home screen. The home screen for students will display the last piece of music that they were practicing, statistics, and

conversations. The home screen for teachers will display the last piece of music that they uploaded or created, student statistics, and conversations.

The student practice screen allows the student to practice a chosen piece of music. As the student sings, the real-time feedback system will provide graphical information to the student. The page will also provide statistics for the current piece of music selected.

The teacher practice screen allows the teacher to create new exercises and modify existing exercises.

When a student views their statistics, they can view general statistics (e.g. how often their pitch is correct) or drill down into individual statistics for a specific piece of music (e.g. at what points their duration was incorrect).

When a teacher views student statistics, they can view aggregated statistics for all students or drill down into an individual student's statistics.

Teachers will also have a simple grade sheet. This is a new (potential) requirement and we have not yet worked out the details.

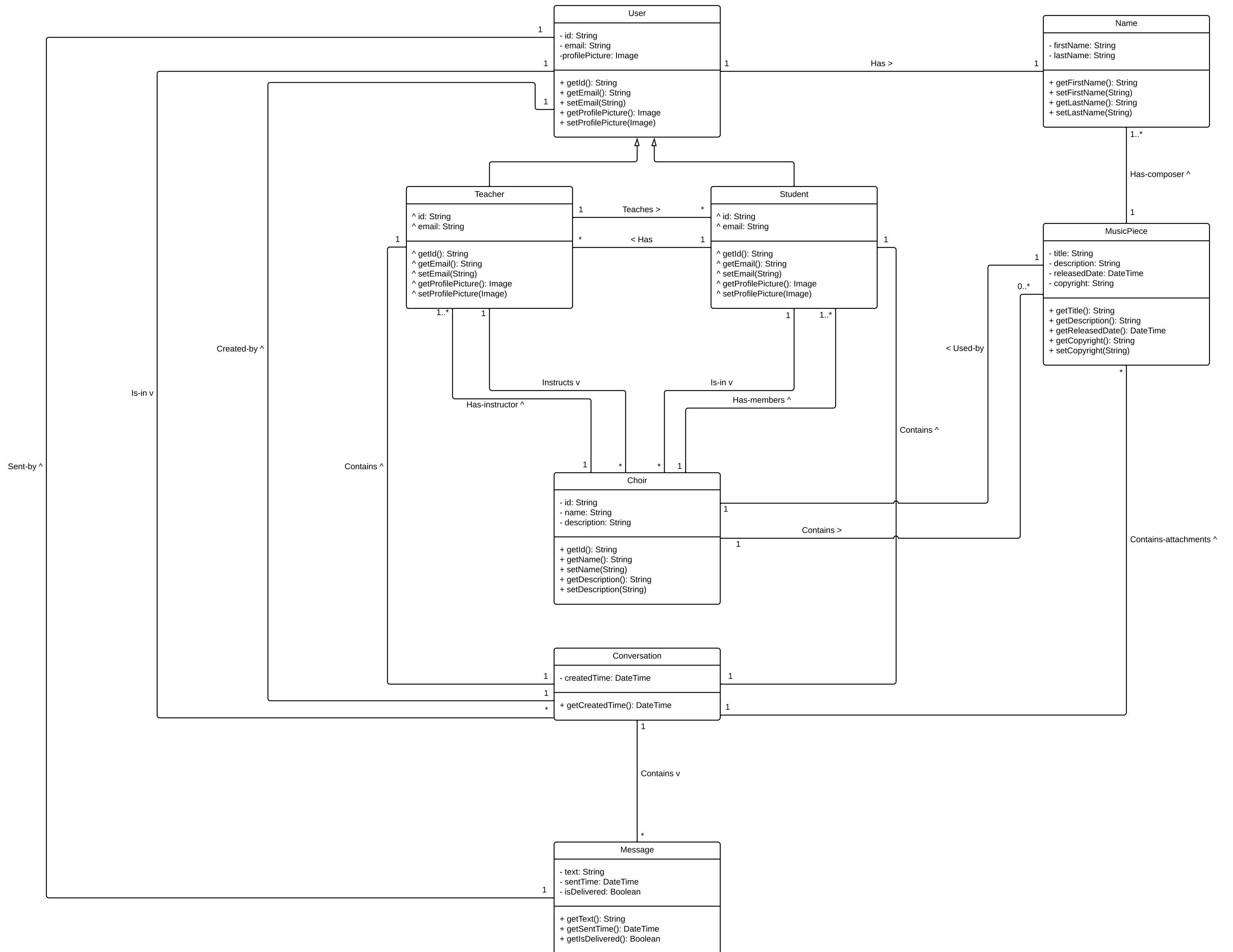6.2 Screen Images

Music Assistent Wireframe.pdf
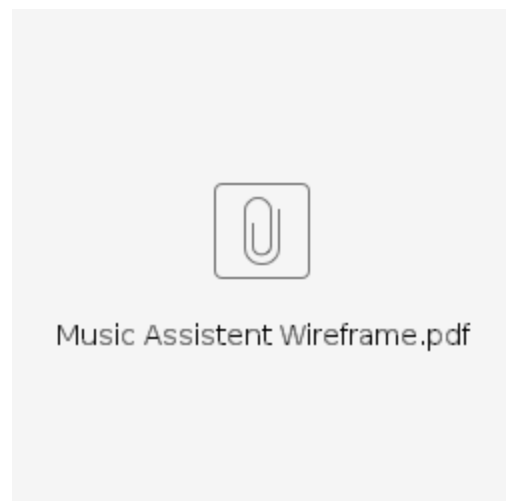
6.3 Screen Objects and Actions

See 6.1

7. Requirements Matrix

The requirements matrix will be completed once all functionalities are known. In particular, we need to determine how to store some data as noted in 4.1.

User

- id: String
- email: String
- profilePicture: Image

+ getId(): String
+ getEmail(): String
+ setEmail(String)
+ getProfilePicture(): Image
+ setProfilePicture(Image)

Name

- firstName: String
- lastName: String

+ getFirstName(): String
+ setFirstName(String)
+ getLastName(): String
+ setLastName(String)

Has >

Teacher

^ id: String
^ email: String

^ getId(): String
^ getEmail(): String
^ setEmail(String)
^ getProfilePicture(): Image
^ setProfilePicture(Image)

Student

^ id: String
^ email: String

^ getId(): String
^ getEmail(): String
^ setEmail(String)
^ getProfilePicture(): Image
^ setProfilePicture(Image)

Teaches >

< Has

MusicPiece

- title: String
- description: String
- releasedDate: DateTime
- copyright: String

+ getTitle(): String
+ getDescription(): String
+ getReleasedDate(): DateTime
+ getCopyright(): String
+ setCopyright(String)

Has-composer ^

< Used-by

Created-by ^

Is-in v

Instructs v

Is-in v

Has-instructor ^

Has-members ^

Contains ^

Contains ^

Choir

- id: String
- name: String
- description: String

+ getId(): String
+ getName(): String
+ setName(String)
+ getDescription(): String
+ setDescription(String)

Contains >

Conversation

- createdTime: DateTime

+ getCreatedTime(): DateTime

Contains v

Contains-attachments ^

Sent-by ^

Message

- text: String
- sentTime: DateTime
- isDelivered: Boolean

+ getText(): String
+ getSentTime(): DateTime
+ getIsDelivered(): Boolean

Home for Students and Teachers

Sheet Music List for Practice

Statistics of Currently Selected Piece

Messages

Teacher | Message
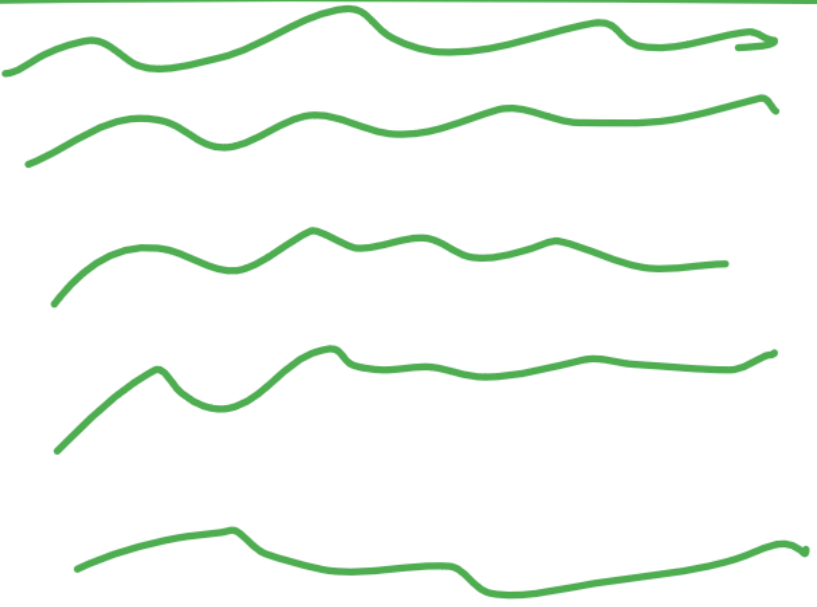
Group Message 1
Group Message 2
...
...

Type...

Messages for Students and Teachers

Messages

Conversation 1

Conversation 2

Conversation 3

Type:

Calibration for Students
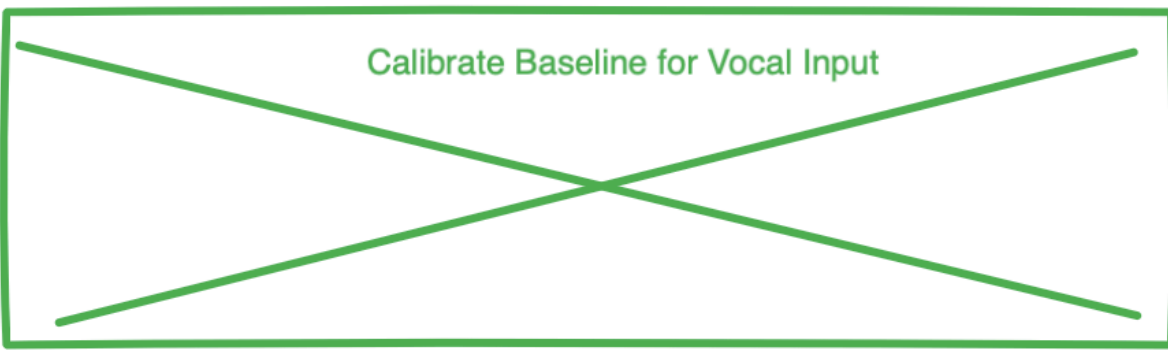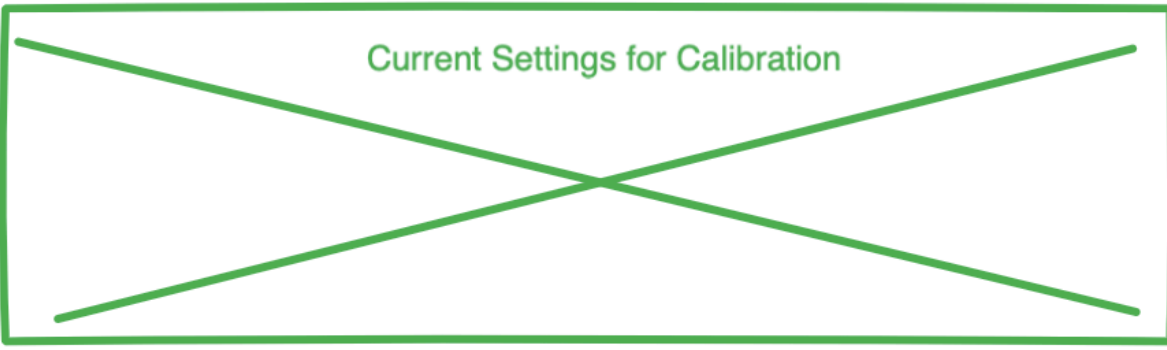
Calibration

Calibrate Baseline for Vocal Input

Current Settings for Calibration

Practice for Students

Sheet Music - Horizontal Scrolling

blah-   di   blah-   di   blah-   di   blah

Play/Pause/Record/Music Navigation

Analysis Stats

Tips from past singing

Previous Stats for this songs

Messages

Teacher   Message

Group Message 1
Group Message 2
...
...

Type...

Practice for Teachers

Create New

Generated Exercise

Generated Exercise

Generated Exercise

Statistics for Students

Song overlayed with notes
Basically highlighting times in song that are noteworthy

Pitches and Duration and Phrasing - detected vs actual. Line graph or on sheet music by red and green notes for pitch and duration. Marks phrasing

Dynamic - shows detected vs actual. Dynamic key using color key

Tempo and Rhythm - shows detected vs actual. Shows unaltered sheet music and lines up detected music where it can

Messages

Statistics Cont. for Students

Select Statistic for Practice - Select area of song which is currently causing the student trouble

Generated exercises matching statistic selected and trouble area(s)

Previous best performance statistic info - includes info about performance against other students

Current performance statistic info

Messages

Choir for Students

Assigned Exercises

Sheet Music - Progress

Sheet Music - Progress

Messages

Choir for Teachers

Student 1
Statistic
Summary

Student 2
Statistic
Summary

Student 3
Statistic
Summary

Student 4
Statistic
Summary

Messages

Choir Cont. for Teachers

Student 1

Sheet Music 1

Sheet Music 2

Sheet Music 3

Sheet Music 4

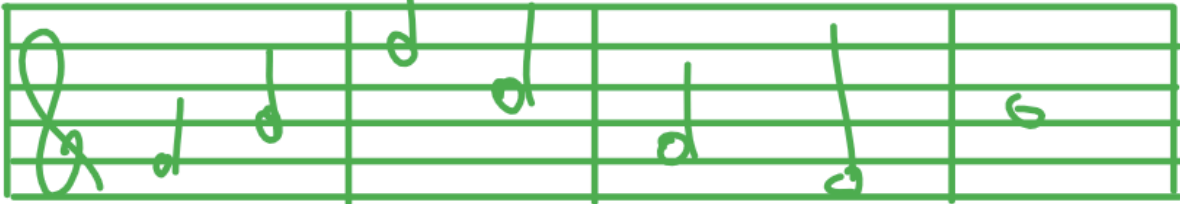Messages

Choir Cont. 2 for Teachers

Student 1

Sheet Music - Horizontal Scrolling

blah-   di   blah-   di   blah-   di   blah

Play/Pause/Record/Music Navigation

Select Statistic

Select Performance on Statistic

Messages

Grading for Teachers

	Practice	Exercises	Assigned Exercise 1	
Student 1				
Student 2				
Student 3				