Introduction

# Purpose

* This documentation seeks to provide its reader with the information needed to use and maintain the Concert Tracker software. This document is therefore broken up into each component and the corresponding subcomponents. Each subcomponent will show both how to use and maintain the subcomponent.

# Authors

* Thaddeus Yurek
* Thomas Burgess
* Peter Greene
* Kevin Ginta
* Lorenzo Macias

Section 1 - System Overview & Key Terms

# 1.1 - Description

The Concert Tracker System is composed primarily of three main components. The first component focuses on the Database, specifically the storing and importing of the concert data. The second component focuses on what a faculty member will interact with, that being the Faculty UI. The third component focuses on what the student will interact with, that being the Student UI.

# 1.2 - List of Components and Subcomponents

**1.2.1 - Database**

* [Concert Database](#_1pahksscof1y)
* [Data-Migration Script](#_dybxk83b472n)

**1.2.2 - Faculty UI**

* [Progress Search Tool](#_xhs5bafjk6v8)
* [Individual Search Tool](#_7zuilzdmy4hu)
* [Requirements Editor](#_8msnid1fqz23)

**1.2.3 - Student UI**

* [Email Script](#_fajagrpopa2)
* [Student Form](#_xn6lo9jwv3v7)

Section 2 - Database

# 2.1 - Overview

**2.1.1 - Description**

1. This component manages the importing and storing of all concert data. This component is essential to the system as every other component depends on the structure defined by this component. This component consists of two subcomponents. The first subcomponent is the *Concert Database* which defines the data structure for the data storage. The second subcomponent is the *Data-Migration Script* which takes data from CSV files and maps it to the data structure found in the Master sheet.

**2.1.2 - Key Terms and Definitions**

1. *Concert Database*

* The *Google Spreadsheet* used to store data.

1. *Data-Migration Script*

* The *Google Apps Script* used to import CSV data into the *Concert Database*.

1. *Google Spreadsheet*

* Consists of one or more *Google Sheets* which store data in a table format.

1. *Google Sheet*

* Individual table of cells used to store data. They appear as tabs on the bottom of a *Google Spreadsheet.*

1. *Google Apps Script*

* Provides scripting functionality to Google Drive software including *Google Spreadsheets*.

# 2.2 - Concert Database

## 2.2.1 - Description

1. The *Concert Database* functions exactly like a database would in a normal system. Therefore the function of the *Concert Database* is to allow data to be written to it, store the data, and allow the data to be retrieved. The *Concert Database* is a *Google Spreadsheet* that consists of five *Google Sheets* that contain logically separated data. These five *Google Sheets* store the (1) student information, (2) concert information, (3) attendance information, (4) category types, (5) semester requirements.

## 2.2.2 - Use & Maintenance

1. Adding Concert Categories
   1. In order to add a new concert category, navigate first to the “ConcertCategories” tab. Go to the next available blank row. In the “Concert Category ID” column, add the 1+ the integer used in the above row. In the “Category Name” column, add the plaintext name of the category. This column represents what will be shown in the UI components. In the “Category Filename” column, add the category name as shown in the filename format for the .csv files. **Do not include the dash before the performer in the column**.
2. Changing Concert Category Filename
   1. If the filename format changes for the concert category, it will need to be updated in the “ConcertCategories” tab. Locate the row which contains the category and update the “Category Filename” column to match what the filename is. The filename should only contain up to the category name, **do not include the dash before the performer**.
3. Adding a new student
   1. **Before adding a new student, be sure there isn’t an upload in progress**. In case a student needs to be added who was not listed in the .csv upload files, one can be added by first navigating to the “Students” tab, and then going to the next available blank row. In the “Students ID” column, add the student’s Biola ID number. In the “First Name” column, add the students first name. In the “Last Name” column, enter the student’s last name. In the “Email” column, add the student’s Biola email address.
4. Adding an attendance record
   1. **Before adding any new attendance records, be sure there isn’t an upload in progress**. In order to add an attendance record, first find the chosen student’s Biola ID number either in the “Students” tab. Then find the concert ID that the student attended in the “Concerts” tab. Then navigate to the “Attendance” tab and find the next available blank row. In the “Date” Column, add the date in which the student attended the concert. In the “Concert ID” column, add the concert ID that the student attended. In the “Student ID” column, add the student’s Biola ID number. In the semester column, add the semester, **making sure to follow the format shown in the above rows**.
5. Adding a Concert
   1. Due to the required “ConcertHash” value generated upon uploads, **it is not recommended to add a new concert to the “Concerts” tab**. If one needs to be added, it is better to make a .csv file, matching the format of previous .csv files.
6. Modifying tabs
   1. Due to the dependence of the scripts upon the tab names **it is not recommended to modify the tab names**. If a tab name is accidentally changed, simply revert it back to what it originally was.

# 2.3 - Data-Migration Script

## 2.3.1 - Description

1. The *Data-Migration Script* provides functionality to the *Concert Database* to allow one to upload CSV documents with the concert data and have them automatically imported into the *Concert Database*. Once a CSV file is uploaded to the designated “To Upload” folder, the script will grab the data and once finished, move the CSV file to a designated “Archive” folder. Should a duplicate CSV file be accidentally put in the “To Upload” folder, the script will, instead of importing the duplicate data, move the CSV file into a designated “Duplicates” folder. Due to Google limiting the amount of time a *Google Apps Script* can run in a day, the *Data-Migration Script* will instead run every 15 minutes.

## 2.3.2 - Use & Maintenance

1. Upload Data
   1. In order to upload data, simply upload the .csv file into the “To Upload” folder and within 15 minutes, the data will be uploaded.
2. What to do if the folders change
   1. If the “To Upload”, “Archive”, or “Duplicate” folders are deleted and have to be added again, then the folder IDs will need to be updated in the script. First get the ID of the folder that needs to be updated. This can be done by navigating to the folder in Google Drive. Looking at the URL, one should see that after the last “/” there is a series of random letters and number with a dash. Everything after that “/” is the ID of the folder. Copy or save that value. Then navigate to the *Concert Database*, and click on “Tools” → “Script Editor”. This will navigate the screen to the *Data-Migration Script*. Make sure the “Data-Migration.gs” side tab is selected. In the code one will see 3 constants which are labelled “UPLOAD\_FOLDER ID”, “ARCHIVE\_FOLDER\_ID”, and “DUPLICATE\_FOLDER\_ID”. Look for the appropriate constant and change the String value located to the right of the constant name and put the new folder ID in there.

Section 3 - Faculty UI

# 3.1 - Overview

**3.1.1 - Description**

1. This component is the primary component the administrators will interact with and are located in the *Concert Database*. The faculty UI consists of three subcomponents, the *Progress Search Tool*, *Individual Search Tool*, and the *Requirements Editor*. These three subcomponents both allow the admin to view and edit the commonly used data within the *Concert Database*.

**3.1.2 - Key Terms and Definitions**

1. *Concert Database*

* The *Google Spreadsheet* used to store data.

1. *Google Spreadsheet*

* Consists of one or more *Google Sheets* which store data in a table format.

1. *Google Sheet*

* Individual table of cells used to store data. They appear as tabs on the bottom of a *Google Spreadsheet.*

1. *Google Apps Script*

* Provides scripting functionality to Google Drive software including *Google Spreadsheets*.

1. *Progress Search Tool*

* *Google Sheet* which provides the ability to search multiple students and view a summary of their progress towards the requirement of that semester.

1. *Individual Search Tool*

* *Google Sheet* which provides the ability to search a student based on their Biola Student ID and see a list of concerts they have attended.

1. *Requirements Editor*

* *Google Sheet* which provides the ability to view and modify the requirements for the selected semester.

# 3.2 - Progress Search Tool

## 3.2.1 - Description

1. The *Progress Search Tool* allows the user to view the progress of multiple students towards the current semester’s requirement. In the *Google Sheet* there is a semester selection option, 4 student identification columns, and a search button to trigger the script. If the search is run on a semester that has requirements set for that semester, then the requirements will show up on the search. If there are no requirements set for the semester selected, then only the attended concerts will be shown in the table.

## 3.2.2 - Use & Maintenance

1. Search Students
   1. To search multiple students progress, first navigate to the “Progress Search Tool” tab. Then choose a semester in the cell located in B1. Then enter the students information in the first four columns labelled “Student ID”, “Email”, “Last Name”, and “First Name”. After entering all of the student information, click the search button to trigger the script which will search the *Concert Database* for the data. **The data is prioritized from left to right**. For instance, if Student A’s ID number is entered on row 5 and Student B’s Email address is also entered on row 5, Student A will be chosen and the row will update accordingly.
2. Clear Table
   1. In order to clear the table and reset it, first navigate to the “Progress Search Tool” tab. Then clear out the chosen semester in B1. Then clear out the student data in the first four columns. Then click the search button to run the script which will in turn clear the data previously shown.

# 3.3 - Individual Search Tool

## 3.3.1 - Description

1. The *Individual Search Tool* is used to get more information on a specific student. This is useful when more detailed information is required than the *Progress Search Tool* provides.

## 3.3.2 - Use & Maintenance

1. Search a Student
   1. To search a student, first navigate to the “Individual Search Tool” tab. Then enter the student’s Biola ID number in the cell A2. The student’s concert information should show up in the rest of the columns.

# 3.4 - Requirements Editor

## 3.4.1 - Description

1. The *Requirements Editor* allows an admin to view and modify the requirements for the selected semester. This will likely be used once a semester to set the requirements for that semester, however it can be used in advance or retroactively.

## 3.4.2 - Use & Maintenance

1. Create Requirements for Semester
   1. To add new requirements for a semester first navigate to the “Requirements Editor” tab. Then enter the chosen semester in the cells A2 and B2. Then click the “LOAD REQUIREMENTS” button. This will generate new requirements since the selected semester does not have requirements. Edit the values for each concert category to the desired values. Once finished click the “SAVE REQUIREMENTS” button to save the values to the *Concert Database*.
2. Modify Requirement for Semester
   1. To modify requirements for a semester first navigate to the “Requirements Editor” tab. Then enter the chosen semester in the cells A2 and B2. Then click the “LOAD REQUIREMENTS” button. This will load the requirement values for the selected semester. Edit the values for each concert category to the desired values. Once finished click the “SAVE REQUIREMENTS” button to save the values to the *Concert Database*.

Section 4 - Student UI

# 4.1 - Overview

**4.1.1 - Description**

1. This component allows students to request their concert attendance data and have it emailed to them. This *Google Form* is automatically updated and completely autonomous.

**4.1.2 - Key Terms and Definitions**

1. *Concert Database*

* The *Google Spreadsheet* used to store data.

1. *Google Form*

* Google provided form which allows people to fill it out and their responses saved.

1. *Google Spreadsheet*

* Consists of one or more *Google Sheets* which store data in a table format.

1. *Google Apps Script*

* Provides scripting functionality to Google Drive software including *Google Spreadsheets*.

1. *Email Script*

* *Google Apps Script* which triggers when a student submits the *Student Form* which sends an email with the students concert attendance data.

1. *Student Form*

* *Google Form* which triggers the *Email Script*.

# 4.2 - Email Script

## 4.2.1 - Description

1. The *Email Script* is a *Google Apps Script*, which is triggered whenever a *Student Form* is submitted, sends an email to the student with the attendance data found in the *Concert Database*.

## 4.2.2 - Use & Maintenance

1. The *Email Script* is automatically triggered upon *Student Form* submission and therefore requires no maintenance.

# 4.3 - Student Form

## 4.3.1 - Description

1. The *Student Form* is a *Google Form* which is used to send students their concert attendance data.

## 4.3.2 - Use & Maintenance

1. Using the Form
   1. In order to allow students to use the *Google Form*, one must first navigate to the *Student Form*. Then click the “Send” button located in the top right and either listing out the student emails or copying the share link and making it available to the students.
2. Viewing the history of requests
   1. In order to view the amount of students who have used the *Student Form*, navigate to the “Student Form (Responses)” *Google Spreadsheet*. Here one can view the history of responses.