NEA DOCUMENTATION

Ben Bromley

Project: Sheet Music manager and editor (annotator),
for musicians from novice to pro

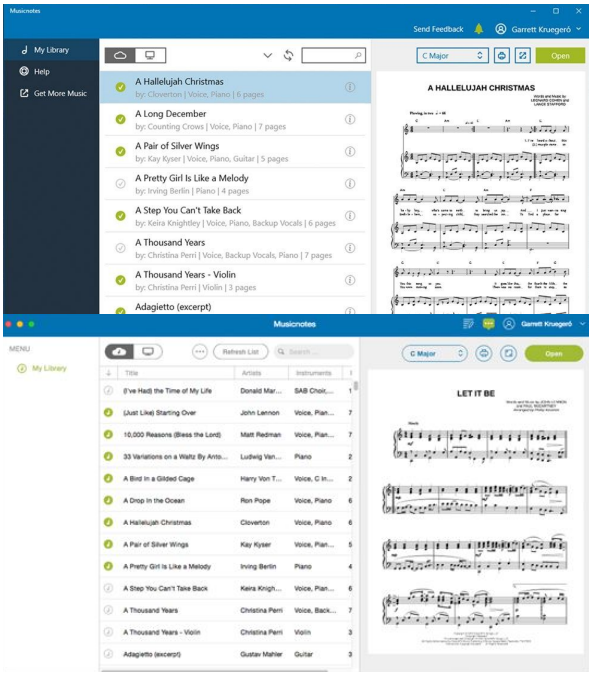
Contents:

ANALYSIS SECTION:

1. Research, Background & Existing Systems

In order to develop a more comprehensive understanding of the problem I face, I have researched different Sheet Music Managers to see how they differ from each other, as well as my own solution.

‘MUSIC NOTES’



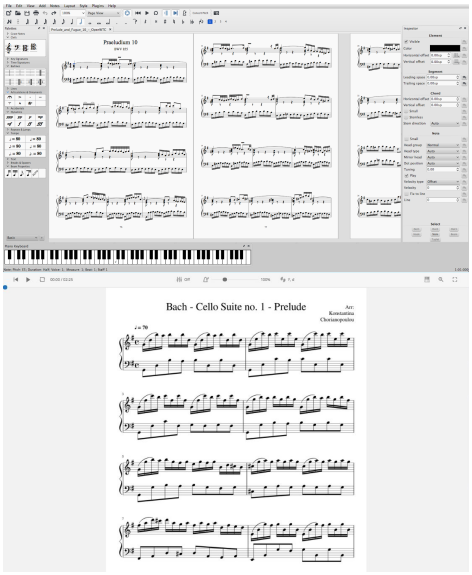
Music Notes is one of the leading services in the world. Their online sheet music store has lots of options, all at a relatively low price.
The part to focus on of course, is their desktop application. You can view your synced library of sheet music, and even import your own PDFs. It displays PDFs page by page, but doesn’t allow for annotation on its windows and mac app.

This system is more suited to beginners or those with basic abilities, because a lot of the music in

store is a simplified version of full scores. That is also combined with the metronome feature, and the ability to listen to your sheet music. Whilst these features are useful, they are not geared towards the audience that my system will be. The reason being that anyone playing the more difficult pieces available, will be able to play simply by reading the music. If necessary, the learner or performer would be more inclined to find a live recording of the piece, because it would be a much more 'musical' performance, than the robotic one delivered by the music notes app.

'Muse Score'

Muse score has one of the best music writing tools, however, it's only viewing capabilities are either in a browser, or on a mobile device. As with music notes, you can listen to your saved scores and with a metronome if you wish. However with muse score, you can only view music published by individual creators on musescore, so there is no quality or accuracy guarantee. You also cannot import external PDFs to use on their online viewer.



2. Interview

What is the purpose of this System?
This system's purpose is to allow musicians to access and modify their own personal database of sheet music that they can organize and use for both study, and use for practicing and rehearsal.

What created the desire for this system?
I have seen so many systems just like this but none of them are practical enough whilst still having a modern and user friendly interface whilst still being free.

How will my system differ from those similar to it?
Most systems like this are used purely for performance and practice, but my system will have added structures and features that allow for it to be used a learning and educational tool for musical studies

In what context or capacity will my system be used?
The beauty of this system is its immense versatility, it will be intuitive enough for younger students to use it in both education and practice, but also good enough for amateur and professional musicians to use on their devices for events. For me personally, this system will be used for studies as a student, as well as for practicing as an amateur musician. It is a key tool for anyone involved in music, and the system will greatly benefit all it's users.

Is my system something that I could or would want to generate profit from?
Most systems similar to this have either a free version with reduced features, or simply require you to use the majority of your sheet music through in app purchases. So, in theory yes I could sell this software, but only in the former sense, through charging for a full version of the software and offering a trial. However due to the nature of my own skills, it will not reach commercial use.

Once the system is complete will it stay bespoke or will it be made open source? Why?
Now this is actually undetermined and will have to be decided later on in the process. Whilst I said that I do have the option of selling it, I also have the option of keeping it to myself or revealing it for all the world to collaborate and build upon. If I feel like the program fulfills its potential, I will keep it bespoke and consider publishing it whilst retaining some form or copyrights. If the system is a failure or highly inadequate, I will keep it bespoke without releasing it to the public. If the system shows promise but has not fully maximized its potential, then

I will release it as open source and see what improvements the programming community can make to it.

How vital is this system to me or the music industry?
Because this system is not entirely original, it will not be life changing for most. A lot of professional musicians could use this system, but would rather use a more well known system. That being said, this kind of system, especially at a low price or as open source, would be of great help to students and educators looking for software that doesn't blow up the budget.

How popular is this kind of system?
So this system is not hugely popular, it is rarely used in classical, or pop music, in live situations. However, for studio musicians, and/or gigging musicians, especially jazz and rock musicians or theatre and pit musicians, this would be highly popular as digital sheet music is extremely helpful and useful in this area of the music industry, because these musicians are travelling from and between many venues and events, and need capable software that makes managing all of their sheet music easy for them.

3. Client

The background of the project is discussed at length in the interview, however a brief summation would be that I feel that there is a need for a system that is widely available and capable to help musicians in a practical way, and give music educators another tool for themselves and their students.

Program Requirements

The System must have a simple but practical user interface, with the ability to import any PDF sheet music, as well as the ability to organise the music into playlists, or by composer, genre, or name. The system must **not** be reliant on an online library of music.

4. Existing System(s) & Problem Identification

Problem Identification

Imports

When people use these systems for events, they are either using their own arrangements of sheet music, or licensed sheet music for official performances, and so it is important that PDFs can be imported and used without having to use a roundabout strategy to be able to use the viewer and editor.

Price

Many of these systems are being increasingly monetised, either through ads, or through trial versions of the system, however my aim is to make this available to educators and performers, without adding incurrences on their already dwindling budget (thanks to the recent decline of the musical industry, as well as government support).

Graphical User Interface

The interface needs to be simple, it needs to be something streamlined that doesn't interfere with a live performance, and that doesn't risk slowing down students or educators.

Independence

This must be a standalone system, without any dependence on an online store to charge its users.

Platform

The system should be dedicated to working on Mac OS and Windows, there are dozens and dozens of mobile apps out there, but not a single satisfactory one for either of the specified platforms.

Features

Annotation should be a primary feature, the ability to add notes to a score is essential in most, if not all use cases. In education this would just be in the form of circling aspects, highlighting and/ or writing notes in margins. In performance and practice this would be similar.

It is common in theatre to cut entire sections out of songs, so the ability to authentically cross or scribble sections out is important. It is also essential that the system be able to quickly move through the document as fast as turning a physical page (or faster).

It should also have a login / user system, that allows different users to access their own sheet music through the system.

It is my aim to solve each of these problems with my system.

5. Objectives

Programming Objectives

Save/ Load: The user should be able to save their changes and annotations as either a copy of the original or a replacement.
Backup: there should be periodical backups of the PDFs or annotations
User Database: There should be a stored database of users, and what PDFs are in their ‘account’
Import: The system must get its music from imported PDFs
Export: You should be able to export any annotated PDFs
Backdoor Access: Ideally users shouldn’t be able to access their PDFs through

a file manager in order to protect the integrity of the user system
Guests: Guest option allows users without accounts to use the system, but without saving any of their files, unless they export it.

Qualitative Objectives

GUI: The interface should be simple and practical.
Annotation: there should be an option to write, type or highlight the PDF
Annotations: There should be options for size and font of text annotations, colour for highlight annotations, and colour and size (different to text size) for drawn annotations.
Page View: The user should be able to view part of a page, a whole page, or two pages at any one time whilst in the view window.
Page Turns: There should be either a scroll bar, or page turn buttons on screen whilst in the view (or edit) window.

Quantitative Objectives

Library: The main window should show a view of all music, with options to sort and organise the music.
Folders: The user should be able to organise their PDFs in their chosen playlists, or leave on the main library view.
Other Sorts/ Filters: The user should be able to sort or filter the view of PDFs by: Genre, Composer/Artist, Name, Date Last Used.

DESIGN SECTION:

Documented Design: Evidence Checklist

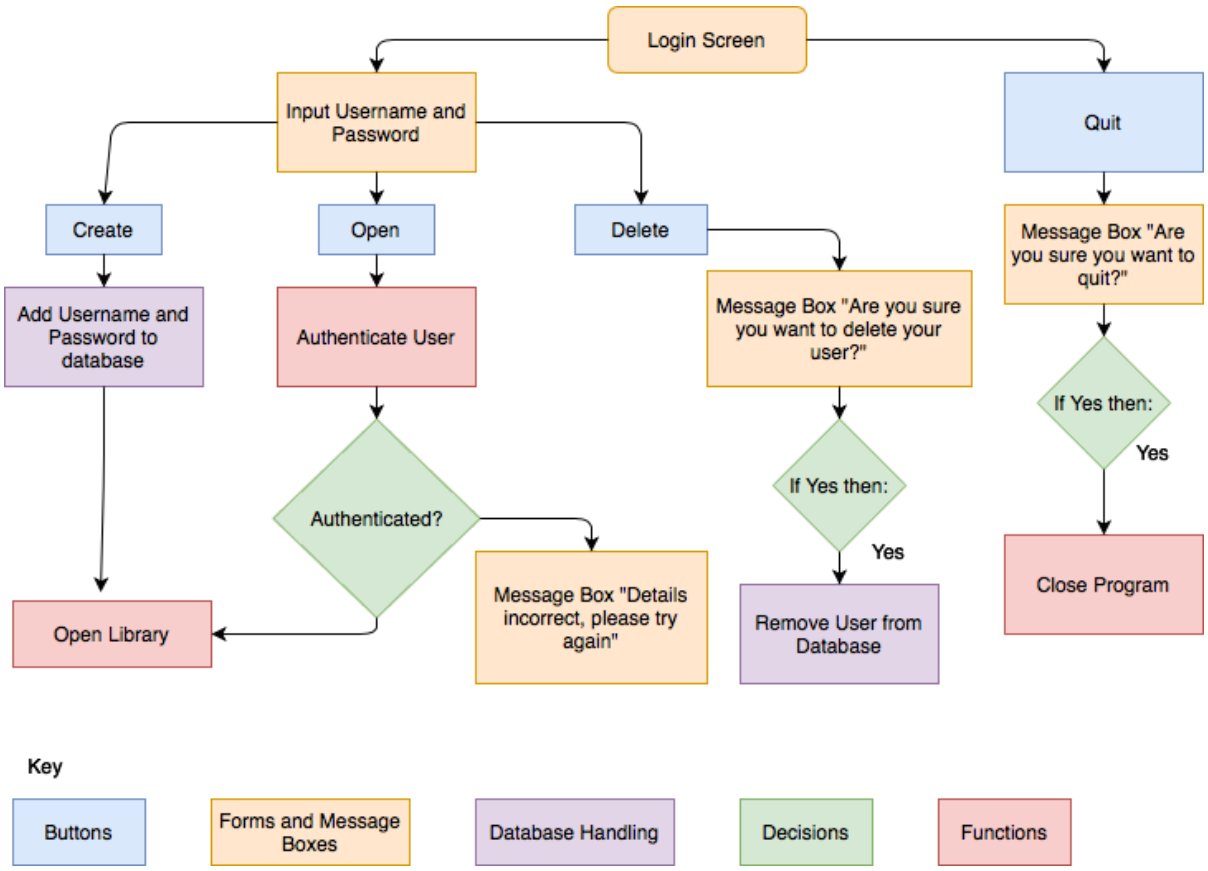
Key to the importance of sections of evidence:

- Any item marked with an E is essential and must be included if you are to gain reasonable marks in this section
- Any item marked with a D is desirable and should be included, it will help to gain good marks but is not always essential
- Any item marked with a C is complementary. It may help explain aspects of your project but is not essential

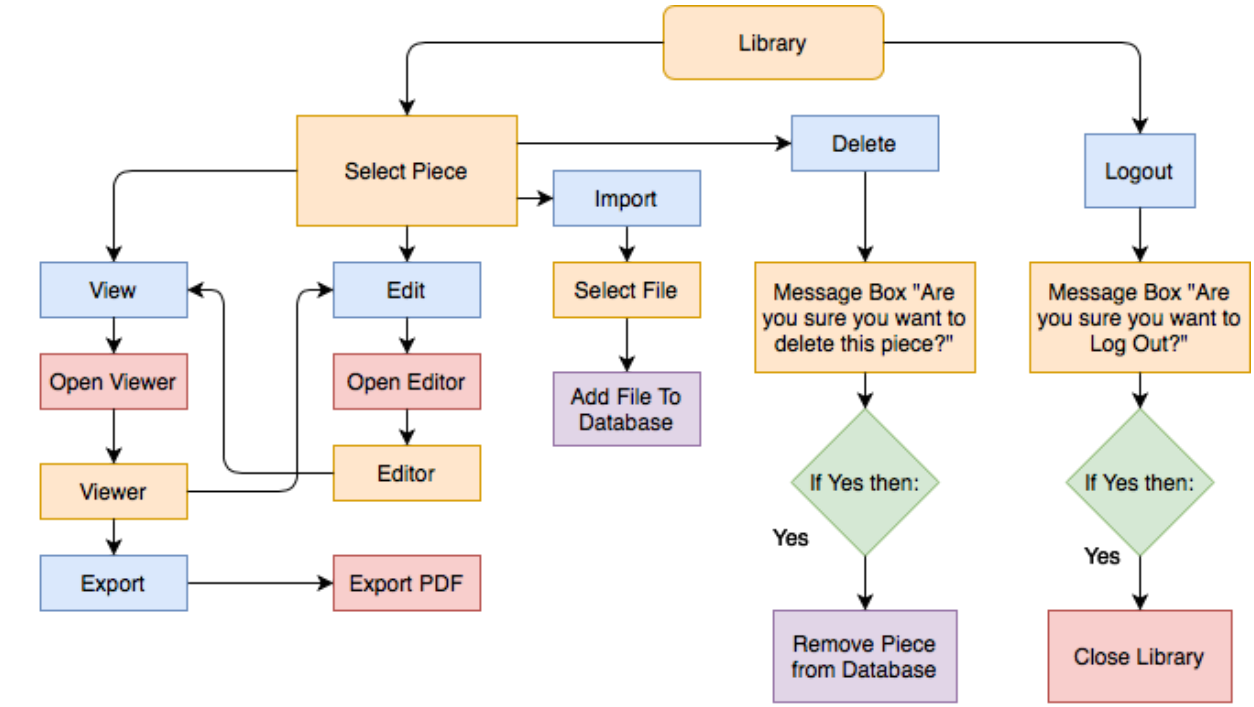
Item No	Have you included???	Tick
1	For Data-Handling projects only: • ERD and EAM [E] • Normalisation [E] • Data Dictionaries [E] • SQL statements [E] • Data Flow Diagrams (to show main data processing) [D]	☐
2	Overall System Design • Top Down diagram [E] • IPSO chart [E] • Flowcharts to show main processing (non-data handling projects) [D]	☐
3	Data dictionaries / Record Structure / Data Structures [E]	☐
4	OOP Class Design (if OOP is used) [E]	☐
5	Interface Design [E]	☐
6	Algorithms [E]	☐
7	Hardware Selection / Design (if appropriate) [D]	☐
8	Security and Integrity of Data [C]	☐

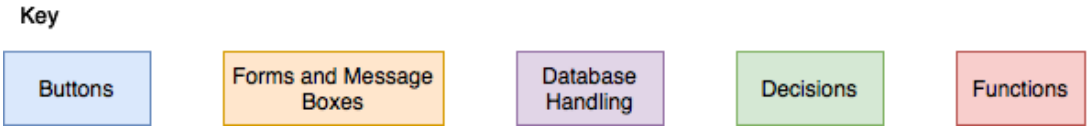
6. Hierarchy Diagrams

Login Hierarchy Diagram



Library Hierarchy Diagram





IPSO Chart

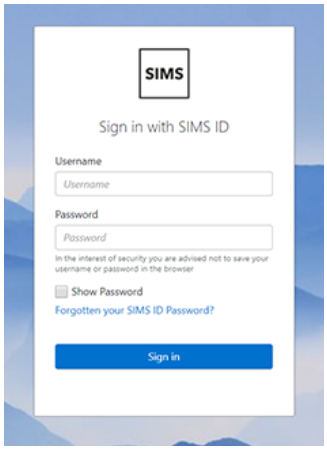
This chart helps me identify the Input and Output, as well as storage and process of each element in each form

Element:	INPUT	PROCESS	STORAGE	OUTPUT
Login Form	Login Button	Encrypt details and check against database, authorise login if details match	User Database	Open User Library
	Create user button	Encrypt details and add to database	User Database	Open New Library
	Quit button	Run Quit Sub Message Box	N/a	Confirmation Form and close program if user clicks yes.
View Form	Export menu button	Opens file window to select export location Exports file	Device Storage	File manager form to save document
	Next Page Button or right arrow key or space bar	Moves to next page	N/a	Next page displayed
	Previous Page Button to left arrow key	Moves to previous page	N/a	Previous page displayed
	Edit button	Opens piece in the editor form	Music Database	Editor form displayed with

				selected piece
	Exit menu button	Closes viewer	N/a	Close View Form and Opens library Form
Edit Form	Save menu button	Saves annotations onto PDF, duplicating the file if it is the first edit, to preserve original in case of error	Music Database	No form change, but piece annotations saved
	Save & Exit Menu Button	Saves the annotations and closes the form	Music Database	Form Closed, but piece annotations saved
	Exit menu Button	Closes the form	Music Database	Form Closed, but piece annotations not saved
Library Form	Delete Button	Removes file and its data from the database	Music Database User Database	Form updated and the library no longer contains deleted item.
	View Button	Opens viewer form with selected file	Music Database	Open Piece Viewer
	Import Menu Button	Opens a message box with a form for all the piece details and adds the data to te data base	Music Database User Database	Message box with a series of entry boxes
	Export Menu Button	Opens message box confirmation		Message box confirmation, exports file if Yes

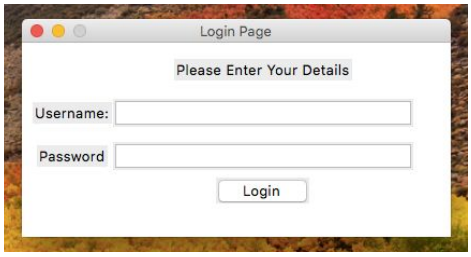
	Edit button	Opens piece in the editor form	Music Database	Editor form displayed with selected piece
	Quit Menu Button	Opens Confirmation message box	N/a	Message box asking if the user wants to quit and quits program if yes

7. Interface Inspirations



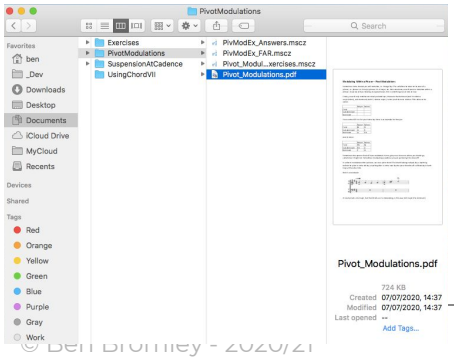
Login Inspiration

I'm using the SIMS ID login as inspiration. It is an effective and clean design, with only a few elements. It has 2 inputs, a checkbox, a sign in button, some text, and a logo at the top.



Login Prototype

Here is a prototype of my login screen. It is an abstracted version of the SIMS login. There are only inputs, and an enter button.



Library Inspiration

This is one of the four file browsing modes on Mac OS, and as soon as I visualised my Library

form, this layout came to mind. I plan to strip it down to only 3 panes, playlist, files, and info.

Library Prototype

Here is a prototype of my Library, it has the three specified panes, with the lists on the left, documents centre, and metadata pulled from each file.

Viewer Inspiration

I wanted a simple interface with just scrolling, and options to exit or open the editor

Viewer Prototype

Here is the viewer, it focuses on one page at a time for now, and allows users to scroll with hardware keys as well as on screen buttons

Editor Inspiration

The mobile app sheet music reader was my inspiration. It's more complicated than I intend mine to be but functions nonetheless.

Editor Prototype

Here is my prototype, and as you can see it is more abstracted than the inspiration, but still allows for both text and drawn annotations to be added and saved to the file.

8. Database

Database Design

There are plenty of attributes to be used, however as this is a rather uncommon system, it will likely undergo multiple revisions as I better understand how best to program the system. The attributes are:

- UserID: Integer
- Username: string
- Password: string
- PieceTitle: string
- PieceID: integer
- DateLastUsed: datetime
- Genre: string
- GenreID: integer
- Composer/Artist/Band: string
- Type: string
- Instrument: string

Unnormalised Form

UserID	Username	Password	PieceTitle	PieceID	DateLastUsed	Genre	Composer/Artist/Band	Type	Instrument
1	Bob	12345	The Four Seasons: Winter The Four Seasons: Summer Cello Suite in G major	125 345 457	12/3/20 12/3/20 11/3/20	Classical Classical Classical	Vivaldi Vivaldi Bach	Full Score Full Score Solo	String Ensemble String Ensemble Cello
2	Tom	password	One Dance Piano Cover Someone You Loved Cover	423 348	28/6/18 31/12/19	Pop	Drake Lewis Capaldi	Solo Solo	Piano Piano
3	Matt	vegman	The Watermelon Man Birdland Pass The Peas	564 798 123	25/9/20 25/9/20 26/9/20	Jazz Jazz Jazz	Herbie Hancock Weather Report The JB's	Lead Sheet Lead Sheet Lead Sheet	Trumpet 1A Trumpet 2B Trumpet 1B
4	Sam	babyman	Sister Act	575	14/3/20	Musical	Alan Menken	Piano Score	Piano 1
5	Dave	pass0000	Eine Feste Burg Die Zauberflöte	143 222	12/11/20 12/11/20	Classical Classical	Bach Mozart	Full Score Full Score	SATB and Cont. Voice(s) and G.Orch
6	Jim	wl1$0n	Messiah Ich Elender Mensch	957 492	9/11/20 9/11/20	Classical Classical	Handel Bach	Full Score Full Score	SATB on Orchestra SATB (Chorale)

First normalised Form

UserID	Username	Password	PieceTitle	PieceID	DateLastUsed	Genre	Composer/Artist/Band	Type	Instrument
1	Bob	12345	The Four Seasons: Winter	125	12/3/20	Classical	Vivaldi	Full Score	String Ensemble
1	Bob	12345	The Four Seasons: Summer	125	12/3/20	Classical	Vivaldi	Full Score	String Ensemble
1	Bob	12345	Cello Suite in G major	457	11/3/20	Classical	Bach	Solo	Cello
2	Tom	password	One Dance Piano Cover	423	28/6/18	Pop	Drake	Solo	Piano
2	Tom	password	Someone You Loved Cover	348	31/12/19	Pop	Lewis Capaldi	Solo	Piano
3	Matt	vegman	The Watermelon Man	564	25/9/20	Jazz	Herbie Hancock	Lead Sheet	Trumpet 1A
3	Matt	vegman	Birdland	798	25/9/20	Jazz	Weather Report	Lead Sheet	Trumpet 1A
3	Matt	vegman	Pass The Peas	123	26/9/20	Jazz	The JB's	Lead Sheet	Trumpet 1B
4	Sam	babyman	Sister Act	575	14/3/20	Musical	Alan Menken	Piano Score	Piano 1
5	Dave	pass0000	Eine Feste Burg	143	12/11/20	Classical	Bach	Full Score	SATB and Cont.
5	Dave	pass0000	Die Zauberflöte	222	12/11/20	Classical	Mozart	Full Score	Voice(s) and G.Orch
6	Jim	wl1$0n	Messiah	957	9/11/20	Classical	Handel	Full Score	SATB on Orchestra
6	Jim	wl1$0n	Ich Elender Mensch	492	9/11/20	Classical	Bach	Full Score	SATB (Chorale)

2nd/3rd normalised Form

User Table

UserID	Username	Password
1	Bob	12345
2	Tom	password
3	Matt	vegman
4	Sam	babyman
5	Dave	pass0000
6	Jim	w!1$0n

Library Table

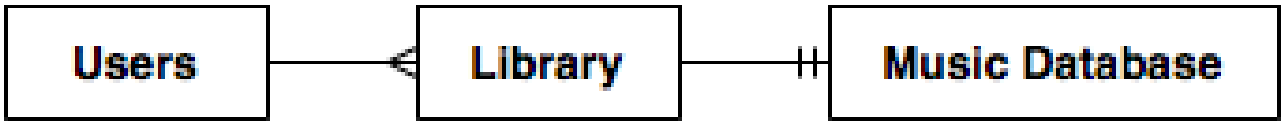
UserID	PieceID
1	125
1	126
1	457
2	423
2	348
3	564
3	798
3	123
4	575
5	143
5	222
6	957
6	492

Music Database

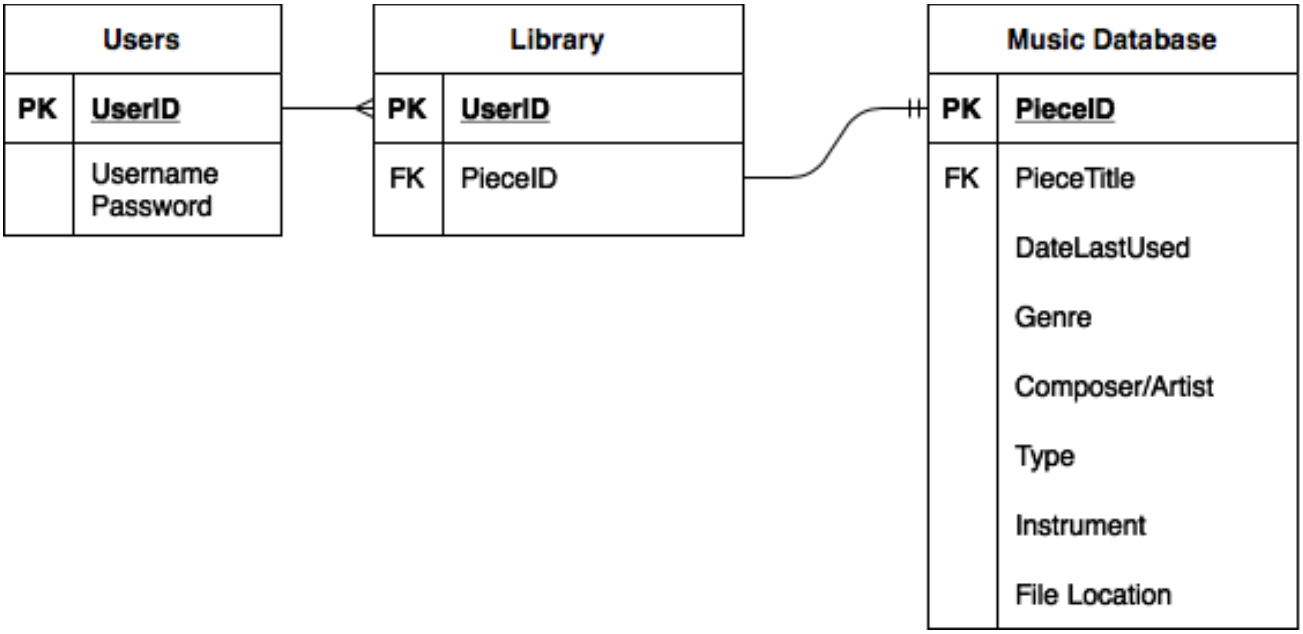
PieceID	PieceTitle	DateLastUsed	Genre	Composer/Artist/Band	Type	Instrument	File Location
125	The Four Seasons: Winter	12/3/20	Classical	Vivaldi	Full Score	String Ensemble	
126	The Four Seasons: Summer	12/3/20	Classical	Vivaldi	Full Score	String Ensemble	
457	Cello Suite in G major	11/3/20	Classical	Bach	Solo	Cello	
423	One Dance Piano Cover	28/6/18	Pop	Drake	Solo	Piano	
348	Someone You Loved Cover	31/12/19	Pop	Lewis Capaldi	Solo	Piano	
564	The Watermelon Man	25/9/20	Jazz	Herbie Hancock	Lead Sheet	Trumpet 1A	
798	Birdland	25/9/20	Jazz	Weather Report	Lead Sheet	Trumpet 1A	
123	Pass The Peas	26/9/20	Jazz	The JB's	Lead Sheet	Trumpet 1B	
575	Sister Act	14/3/20	Musical	Alan Menken	Piano Score	Piano 1	
143	Eine Feste Burg	12/11/20	Classical	Bach	Full Score	SATB and Cont.	
222	Die Zauberflöte	12/11/20	Classical	Mozart	Full Score	Voice(s) and G.Orch	
957	Messiah	9/11/20	Classical	Handel	Full Score	SATB on Orchestra	
492	Ich Elender Mensch	9/11/20	Classical	Bach	Full Score	SATB (Chorale)	

ERD and EAM

Entity Relationship Diagram



Entity Attribute model



Data Dictionary

The Primary Key of the User table is synonymous with that of the Library table, which is why the Primary Key for both tables. This allows me to connect both databases without having any extra unnecessary foreign keys.

User Table			
Data Item	**Data Type**	**Validation**	**Example**
UserID	Integer	Unique identifier for each user their corresponding library	1
Username	String	A personal identifier for the users account	Drew
Password	String	Secures users files	password123
Library Table			
Data Item	**Data Type**	**Validation**	**Example**
UserID	Integer	Unique identifier for each Library and its corresponding user	1
PieceID	Integer	Unique identifier for each piece in the database	243
Library Table			
Data Item	**Data Type**	**Validation**	**Example**
PieceTitle	String	Tells the user the name of piece	Suite in G major
DateLastUsed	Datetime	Tells the user the last date the piece was opened	12/3/20
Genre	String	Tells the user the genre of a piece	Classical

Composer/Artist	String	Tells the user the person who wrote the piece	J.S. Bach
Type	String	Tells the user the kind of score it is	Solo score
Instrument	String	Tells the user the instrument the music is written for	Cello
File Location	String	Tells the program where the file is	C:\Users\Music\SheetMusic\file.pdf

SQL Statements

Creating a new user

```
SELECT * FROM tblUsers WHERE Username=@Username
INSERT INTO tblUsers (Username,[Password])
VALUES (@Username,@Password);
```

Logging-in and Deleting a user

```
SELECT * FROM tblUsers WHERE Username=@Username AND Password=@Password
```

Deleting a user

```
DELETE FROM tblUsers WHERE UserID=@UserID;
```

Data Flow Diagram

9. Class Design

10. Algorithms

11. Testing

Test No.	Testing:	Type of Test:	Test Data	Expected Outcome	Actual Outcome	Comments
1	Login (console)	Normal	Username: benjamin Password: 12345	Access Granted	Access Granted	Building backend to program
2	Login (console)	Boundary	Username: Benjamin Password: 12345	Access Granted	Access Granted	"
3	Login (console)	Erroneous	Username: jerry Password: 56789	Access Granted	Access Granted	"
1	Repeat login on negative result	Normal	Username: jerry Password: 1234	Access Denied (and repeat login screen)	Access Denied (and repeat login screen)	This Feature isn't absolutely necessary but is essential to later improving security
2	Repeat login on negative result	Boundary	Username: Benjamin Password: 12345	Access Denied (and repeat login screen)	Access Denied (and repeat login screen)	"
3	Repeat login on negative result	Erroneous	Username: 12345 Password: benjamin	Access Denied (and repeat login screen)	Access Denied (and repeat login screen)	"
1	sql_const connection to main.py	Normal				
1	Creating User Table	Normal				
1	Creating Library Table	Normal				
1	Creating Music Table	Normal				

1	Creating User	Normal				
1	Deleting User	Normal				
1	Changing User Details	Normal				
1	Importing file	Normal				
1	Exporting file	Normal				
1	Viewing file	Normal				
1	Editing file	Normal				
	Extracting Metadata	Normal				
	Changing File name within program	Normal				

12. Evaluation