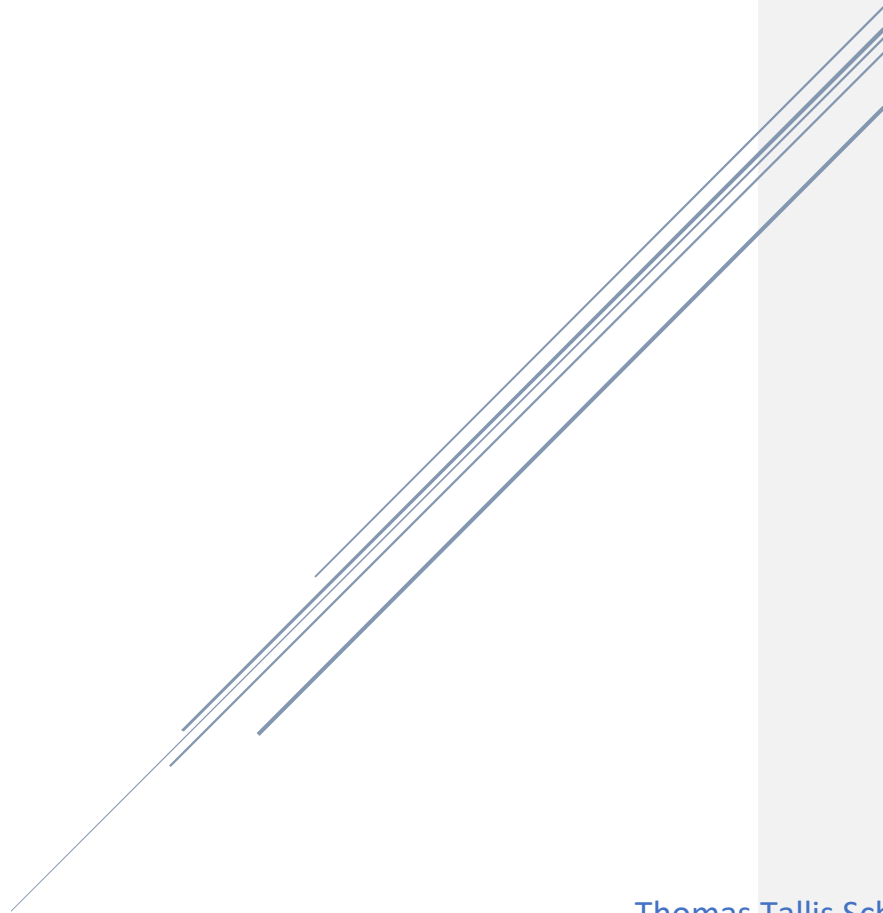


# A LEVEL PROGRAMMING PROJECT

Music Composing Website



Thomas Tallis School  
Beauty Oluokun

# Requirements & Analysis

## Researching Existing Applications and Websites

### MuseScore

MuseScore is a free music-writing application for Windows, MacOS and Linux that is hosted on your device however, it is connected to musescore.com where the user can upload their scores and download other people's scores. The website has some features which are 'pro' and must be paid; for example, some scores can only be accessed by users with 'pro' membership. The main connection between the application is being able to upload and download scores to and from the website's cloud.

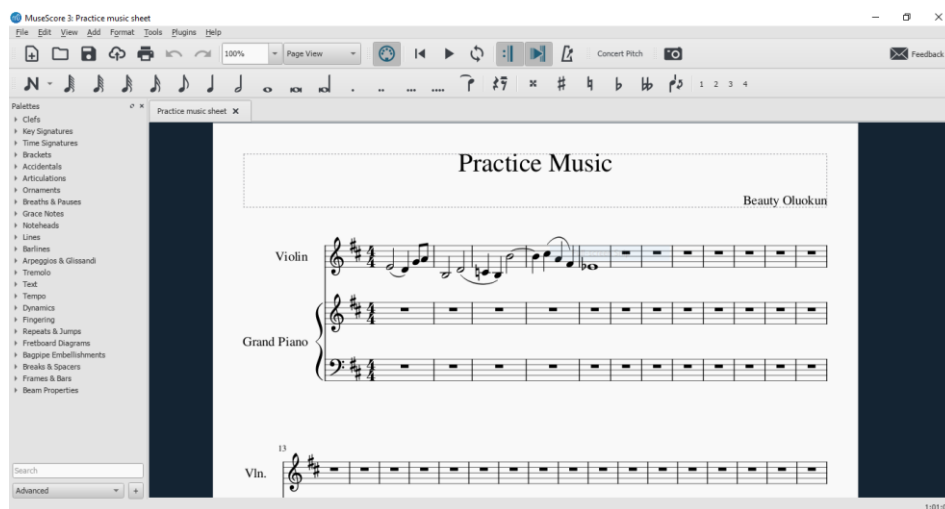
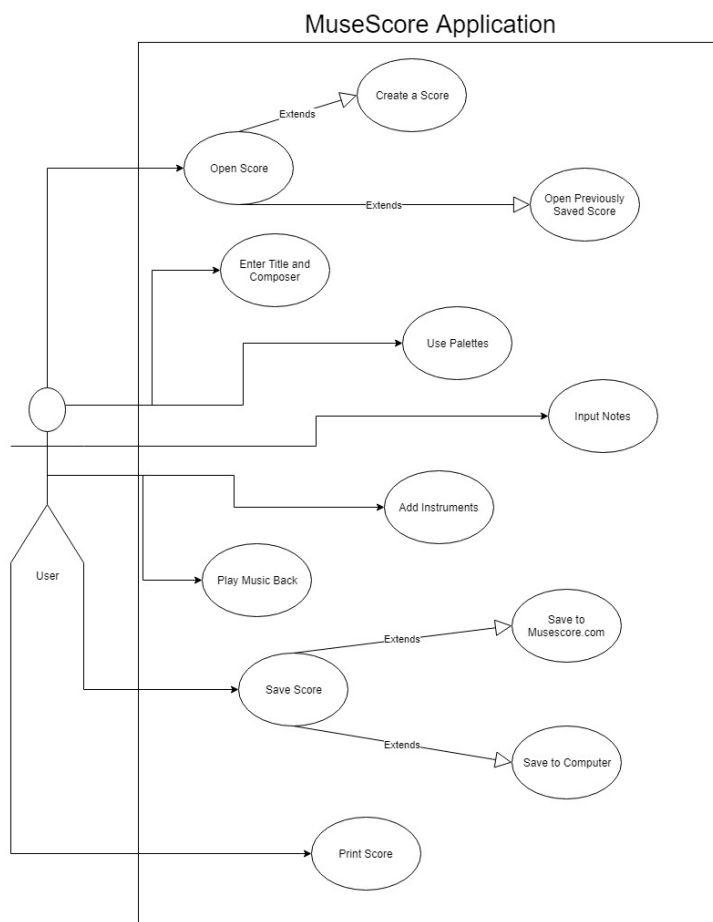


Fig 1. MuseScore application interface

Commented [p1]: Fig 1. Musecore application interface

You can use the mouse and keyboard to interact with the application easily. Note inputs are located along the top bar quite visibly and palettes – which refer to symbols like key signatures and dynamics markings – can be found in a menu on the left. The menu-driven interface uses both text and images to identify each symbol and a search bar to help the user find the symbol they would like to use. The palette menu can be switched between a basic and advanced version, additionally, there is a master palette available which have a massive range of symbols.

This allows users with different levels of musical experience to use the application in a way that's comfortable to them.



**Fig 2.** MuseScore application use-case diagram

**Commented [p2]:** Etc.

To enter notes into the score, note input must be activated first which can be done by either pressing the 'n' key on the keyboard or clicking the note input icon. If note input is off, you can't add any notes to the score, this is quite strange compared to some of the most popular score-writing software but is quite effective as it prevents the user from accidentally editing the notes when note input is off. The pitch can be chosen by dragging the note onto the correct position or by typing in the letter of the note. The duration of the notes can be chosen by clicking the

duration icons or by typing in their assigned numbers. The flexibility of which peripheral to use with the software helps the user explore the application more and allows them to find a way of using the application that works very well for them.

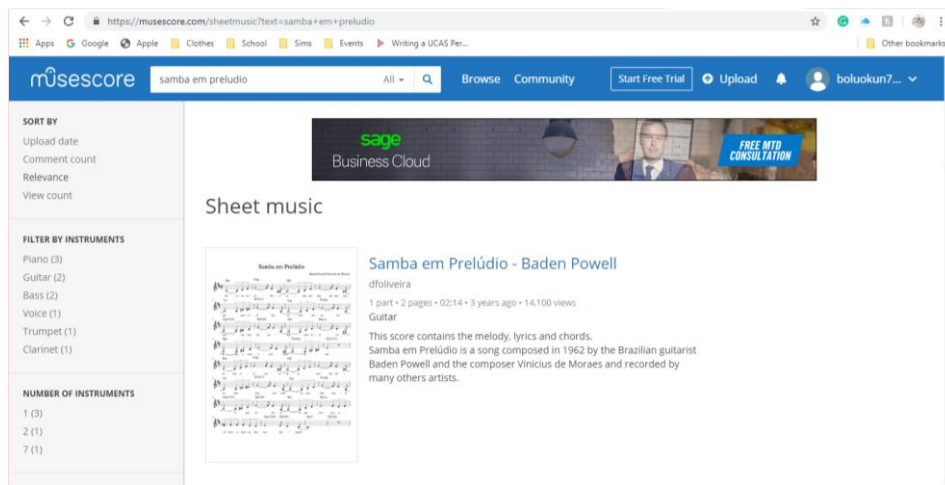


Fig. 3 – Musescore website interface

The MuseScore website mainly houses the social networking aspect; along with sharing scores, users can communicate on multiple forums and join various groups. The website is mainly menu-driven, and the top bar stays the same on all webpages. The top bar contains a search bar which is used to search for scores saved in the cloud, the MuseScore logo which links to the home page and other icons and text that link to other webpages.

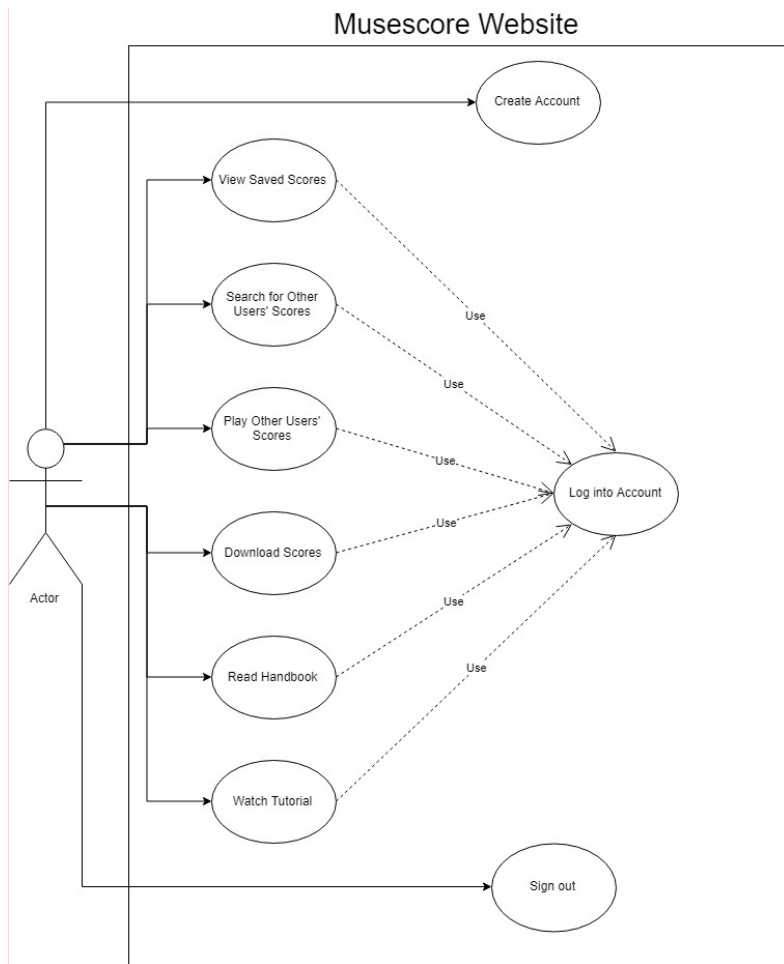


Fig. 4 – MuseScore website use-case diagram

## Noteflight

Noteflight is a website (not an application) where you can create, edit and share music scores. As it is completely web-hosted, it can be used on any device and OS with a browser. As the software and the files aren't stored on the user's device, they don't have to be concerned with storage space or the risk of malware – however, the user must have internet connection to use

the software which can be inconvenient. To use the website, you must create an account and there is an option of two accounts: the free account or the 'premium' (paid) account. There are many features that are only available to premium users like access to some webpages such as the dashboard or features like being able to print scores for each instrument used separately. This can be very frustrating for a free user as you will only know these features are premium if you click on them.

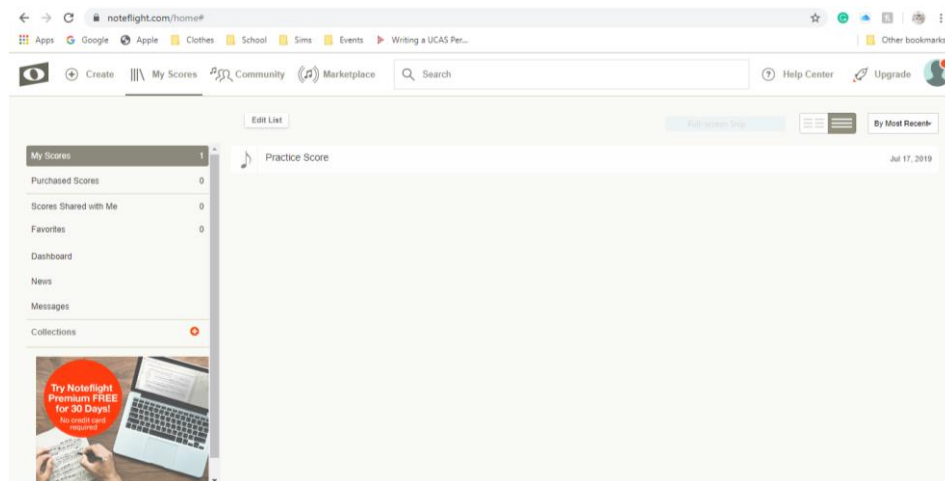


Fig. 5 – Noteflight score library interface

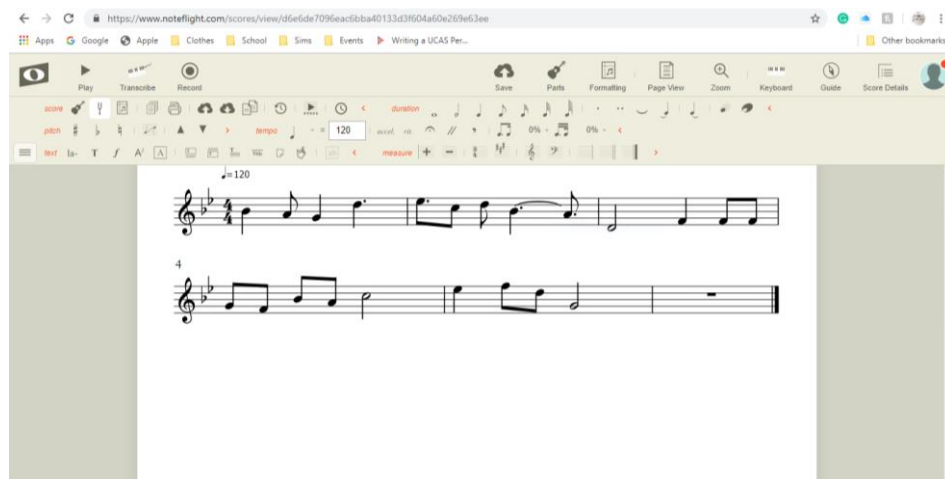


Fig. 6 – Noteflight score-editor interface

All symbol and note input options can be found in the top bars which can make searching for the right icon quite difficult – especially as some options are originally hidden and you must click the ‘show more’ icon to reveal them. But it also gives more room beneath the top bar for sheets which maximizes the space as it’s a website and on laptops or tablets the screen will probably be small. You can input notes by clicking the stave lines or typing in the notes, then you change the durations after entering the first note. Note input can be very frustrating as you can’t select the duration of the note before entering it, once entered the next note space isn’t selected so you can easily change the previous note when you wanted to enter a new note. What makes this even worse is that there are no undo or redo icons so once you make a mistake it’s very hard to rectify it.

To add techniques or expressions, you don’t use a menu like most music-writing software, instead you click the technique or expression icon then type in the technique or expression you want. This is a very inconvenient and time-consuming method as most music expressions and techniques are in Italian or Latin so are hard to spell properly for most. Additionally, beginner musicians probably won’t know that many techniques or expressions so they would benefit from having a list of options to choose from – allowing them to learn. Once symbols like ties are attached to a note, they can’t be selected individually, they are stuck to that note so to delete the symbol you must delete the note also which can become infuriating.

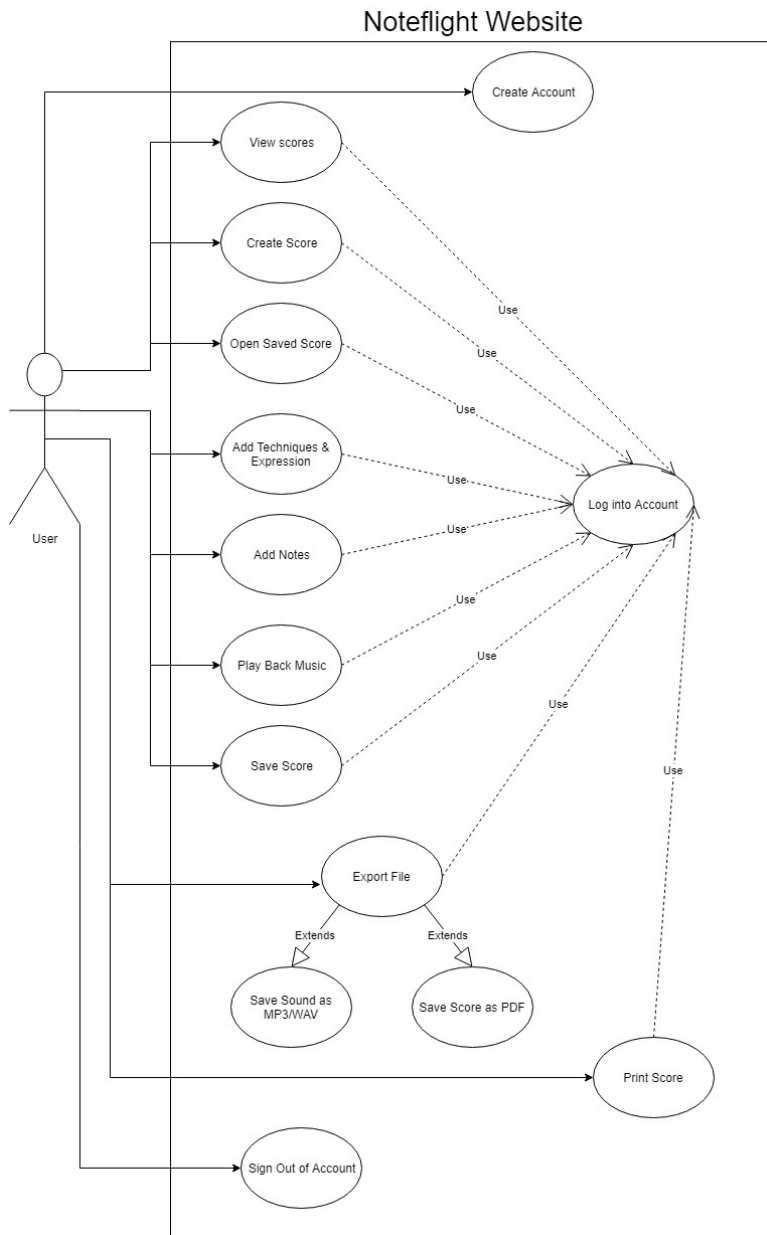


Fig. 7 – Noteflight use-case diagram



## Summary

Overall, both MuseScore and Noteflight have features that I would like to apply to my project. My project will be completely web-hosted like Noteflight as I feel my target audience (young people) would rather use a website than download an application. Like both MuseScore and Noteflight, the users would need to create an account but there will only be a free version as young people are unlikely to pay to use a 'premium' version but will probably be frustrated that not all of the features are available to them. The layout of the actual score editor will be like the one in the MuseScore application as it was much more straightforward to use.

## Features of Proposed Solution

**Commented [p6]:** Missing. DFD and description of features goes here. Link to previous section

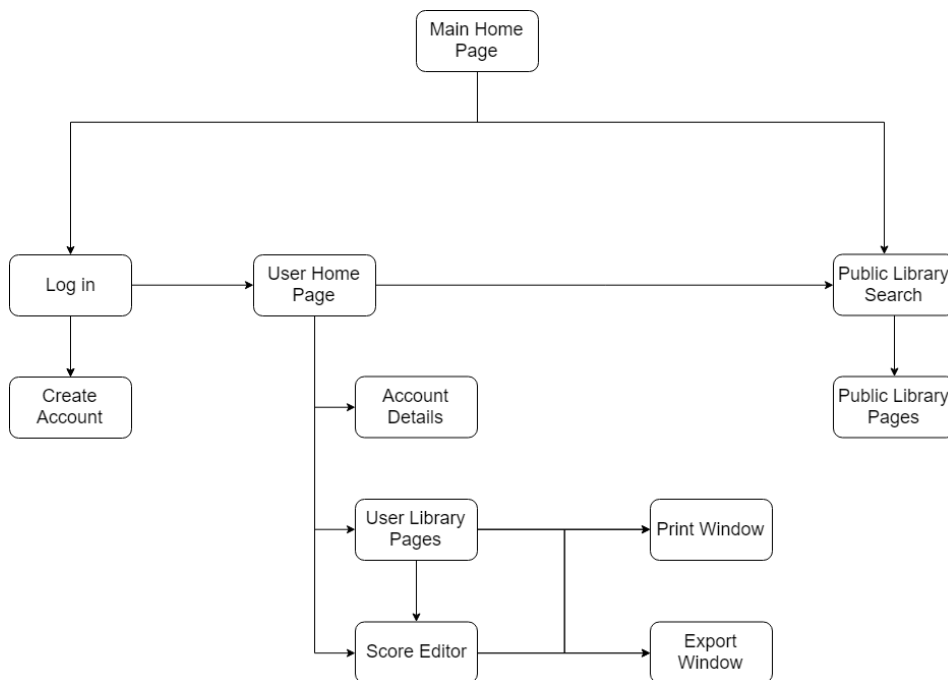
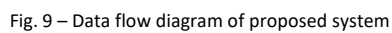
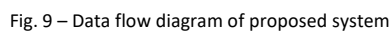


Fig. 8 – Site map of proposed website



## ***Description of Website***

The main functions of my website are outlined in the data flow diagram above (fig. 9) and the main webpages are shown in the site map above (fig. 8). Like MuseScore and Noteflight, when the user wants to first use the website, they must create an account using an email address and secure password – this is done on the ‘create account’ webpage. Their login details are then saved to the user accounts data store. When the user wants to ‘log in’ they go to the login webpage; the details they enter are compared with the details in the user accounts data store and if they match the user can log in and is taken to the ‘user log in’ webpage. This webpage is restricted to users with accounts and allows them to access the ‘user library’ and ‘account details’ pages.

On the ‘account details’ webpage, the user is able to edit details such as their email address and password; the ‘user accounts’ data store is then updated. On the ‘user library’ webpage, the user can view their saved scores. If they click to have one of their saved scores uploaded to the public library, the selected score’s file will be copied from the ‘user scores’ data store to the ‘public scores’ data store. If the user clicks to edit a saved score or create a new score, they are taken to the ‘edit score’ webpage and the selected saved score will be uploaded from the ‘user scores’ data store to the ‘current score’ data store or a new file will be created and moved to the ‘current score’ data store. The user is then able to edit the score using the score editor interface. If the user clicks to save the score, the file in the ‘current score’ data store is copied to the ‘user scores’ data store. If the user clicks to return to ‘user library’ webpage, the file in the ‘current score’ data store is deleted as it is only temporary – so the user must ensure they have saved their score before exiting the page.

When on the ‘user library’ webpage, the user can select a score and click the option to print the score or export the score. The user also has the option to print or export their current score on the ‘score editor’ webpage. If printing is selected, the score’s file is converted to a PDF and the user’s browser’s printing protocol is started. If exporting is selected, the ‘export’ webpage is opened in a smaller separate window; on the page, the user can select whether they want to export the score as a PDF or the music as a MP3 file. The score file is then converted to the chosen file format and downloaded to the user’s computer – the ‘export’ window will then close.

The user can access the ‘public library search’ webpage from the ‘main homepage’ and the ‘user homepage’. On the webpage, there is a searching tool that allows the user to do either a basic or advanced search of the public library of scores. Once they enter their search, they’re taken to the ‘public library pages’ and the scores that appear will be based on the user’s search. Only 20 scores will be shown per page so the user will be able to navigate through a variable

number of pages. If the user selects for a score to be saved to their user library, that score's file will be copied from the 'public library' data store to the 'user library' data score.

### ***Justification of Features***

I have decided to make the user have an account so they can easily access and edit their own scores from several devices not just the device the music score was created on. The account system also helps the website to only make use of the internet through cloud storage and reduces the need for the score files to be stored on the user's device. It is essential that the user is able to update their details after they've created an account hence the addition of the 'account details' webpage.

Like Noteflight, I've created two separate data stores and webpages for the public scores and the user's personal scores. It makes the webpage much easier to navigate as the user can easily distinguish scores they have saved and created from the scores that have been shared by all users. The user score data store also decreases the data that must be uploaded from the cloud when the user logs in as only scores linked to that specific user need to be uploaded alongside the public scores. The score sharing feature creates a community of users as one user can view and improve a score written by another user. I believe this makes my website more appealing to my target audience (young people) as it brings an aspect of social networking.

Like MuseScore and Noteflight, the score editor is like a mini app within the whole system – but implemented as a webpage like Noteflight – that has a static layout and fixed functions that can be used with any score file that is uploaded. This allows the score editors various functions to be programmed once and yet work generally on any score file as their formats will be the same.

The use of the temporary data store 'current score' allows the user to edit one of the scores from the 'user scores' data store without directly changing the file – instead modifying the copy in the temporary data store. This is very useful because if the user decides they don't want the changes saved, the temporary file can be discarded, and the original file is left untouched. It also allows several copies of the score under different file names to be saved from the same original score when on the 'score editor' page.

Both MuseScore and Noteflight have the 'print' and 'export' features that would like to use in my system. I feel they help the system seem more useful to the user as they can print out paper copies of their scores, share PDF versions of scores and listen to a MP3 audio file of their piece of music. My score-writing website is supposed to aid users in writing music but, ultimately, the scores aren't meant to stay within the website.

### ***Limitations***

As my starting amount of cloud storage is likely to be very limited, I have decided not to implement a forum feature like MuseScore. As more people create accounts, make scores and share them to the public library, the 'public library' data score will become quite large and take a lot of bandwidth to upload from the cloud server . Therefore, it would be easier and more efficient to minimise the amount of data that needs to be transferred over the internet as the website would run more smoothly for a user with a weaker internet connection.

Likewise, the user's current score will not automatically save while the user is on the 'score editor' page. Neither MuseScore nor Noteflight do this however, other cloud-based websites such as Google drive and Office 365 support automatic saving of files. Including automatic saving would also requires a lot of bandwidth as the updated data would continuously have to be transported over the Internet to the cloud server. More importantly, automatic saving would remove the main advantage of 'current score' temporary data store as the file in the 'user scores' data store would have to be updated directly.