Music Library Application Report

The User will have the ability to register an account and update the details.

They will then be able to login and be directed to the Add a song Page

Now they can add a song of their choice.

It will then be added to the Playlist Page.

They can delete a song post if they wish.

They will also be able to logout.

**Requirements**

* Trello Board
* Relational Database#
* Documentation
* Functional Python application
* Testing
* Front-end website using Flask
* Integration with Version Control System
* Deployed by CI Server

**Technologies Used**

Trello

Google Console Platform

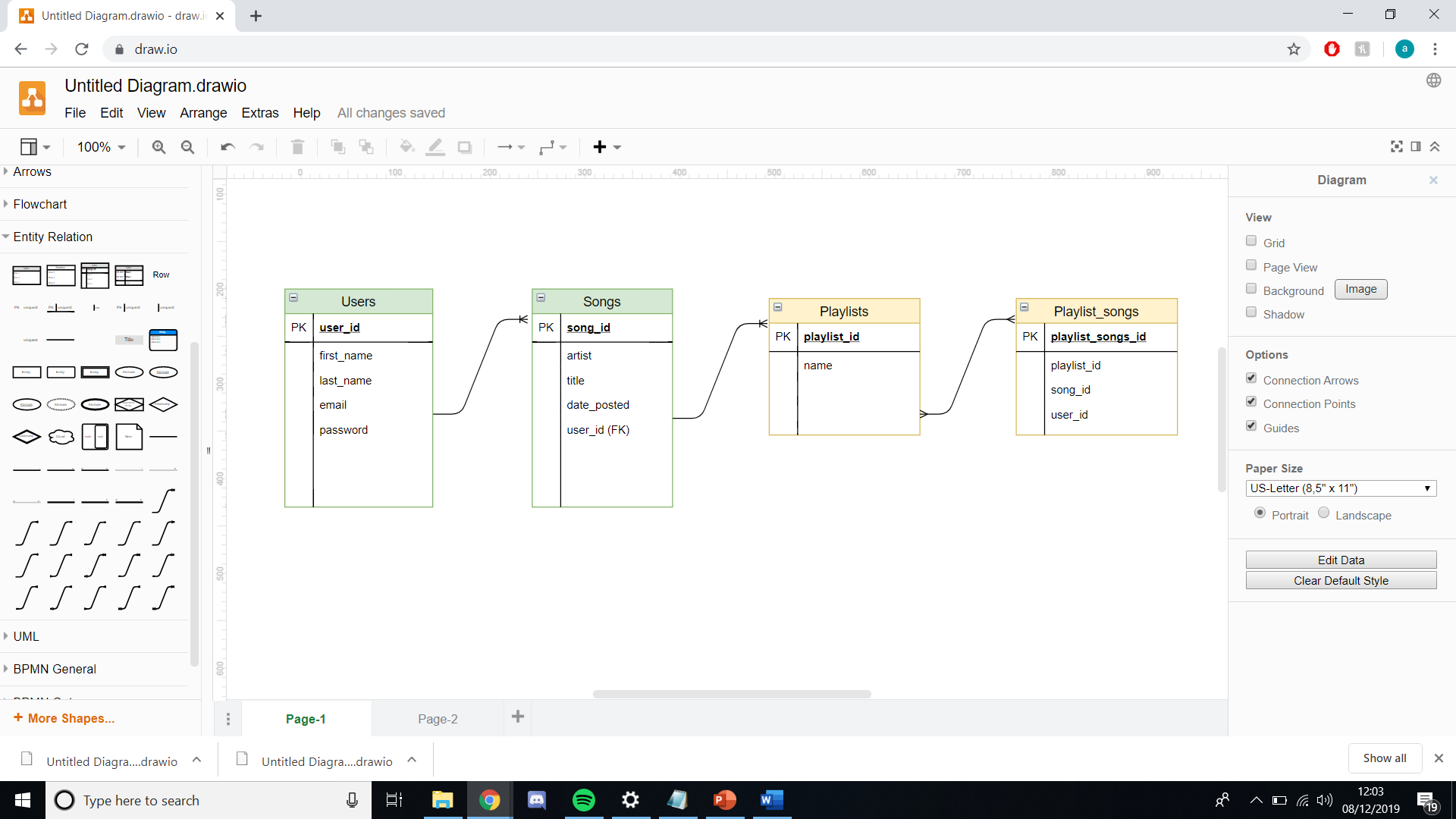
Python

Flask

Version Control - Git

CI Server - Jenkins

**Entity Relationship Diagram**



This is the ERD showing the tables that were used for this project. The two tables that were used were the Users and Songs tables with a 1 to many relationship.

The other two tables which solved the many to many relationships which would have occured between the playlist and songs tables were not implemented as the minimum viable product was reached with the must have tables.

**Risk Assessment**

Poor Planning:

With poor planning there may be a low likelihood of it occurring but if it happened then the impact would be very high the project. Having poor planning would be a major danger to the project. So to combat this I used trello to make sure all requirements are down and all major goals are planned out.

GCP issues:

Having a problem with the GCP instances would have a major impact on my project however it has a low likelihood of happening. To make sure the likelihood is low i will keep the number of instances down so there is no conflict or confusion between any virtual machine

Presentation:

With the presentation there is a low likelihood as well as a medium impact on the project. If the presentation was to have problems this shouldnt impact the project greatly if there is sufficient documentation to support the web application.

Automates Test:

As having automated tests is an important feature of the project. It would have an impact on the project if there was no automated test. However if there is a problem with the automation there is always the manual unit testing to fall back on which does the same job just not automated.

Version Control:

Having a version control system in place is essential to having a successful project. Working on different features and pushing the complete code up using git gives the project a high chance of succeeding. It's better than uploading all work at the last minute which could have massive bugs.

Documentation:

Documentation is very important so not having sufficient documentation would have a high impact on the project. If there is no evidence of the work you did then the credibility of the work is not great.

**Crud Functionality**

* Create - Create songs
* Read - read the database and show on playlist page
* Update- update account details
* Delete - delete songs

**Testing**

Test coverage report using pytest.

