Final Report

This summary will be a bit different from the summary I wrote for the midterm as the flow here needs to be a bit different. This document will consist of two main parts, the first will be about issues that I came across trying to download and run this project. It is as comprehensive as I could make but it should aid in any issues with downloading and running the code. The second half will cover my problems and overall opinion of the project.

## Readme

This project was made in C# 6.0 on visual studio 2017 community edition. Upon downloading the project the path to the db in all of SqlCommand statements will need to be updated as well as the path to the model in the web.config file. I took some time to test it on my laptop before submitting it.

The path to the database file will look like the following on download:

C:\Users\otherUser\Desktop\WebApplication1\WebApplication1\userAndClassPJT.mdf

Everything before the first occurrence of WebApplication1 will need to replace with the local path that can be copied and pasted in from windows explorer. This new path will need to be added to the following places:

* Web.config line 21
* ClassSelect.aspx.cs line 24, 60,
* Main.aspx.cs line 14
* Login.aspx.cs line 31

The database itself and the scripts needed to generate it are zipped and included in this file if any issues arise with the db running the **finalModel.edmx.sql** should make a new one from scratch and the **addClasses.sql** script can be used to populate the class table it can also be modified to add more or remove some classes. I zipped it with 3 classes.

## Reflection

I must say that this project was a lot harder than I initially thought. It seemed simple in theory to add another database, a new page and an extra feature onto an existing page. But binding is a difficult subject to tackle especially if you have limited database experience. Thankfully I work with them a bit at my internship so I could relate the two but if you don’t know how to write SQL queries or use the sql library in c sharp it’s basically impossible to do this. On top of that the class covered several topics but not many that directly covered exactly how to do this or how to manipulate PJTs in ADO.net. For clarity, PJT in my context is pure join table and it makes it incredibly simple to do the correlation between the users and classes tables but we didn’t cover the concept too well and it made things require quite a bit of research.

On one hand that makes in difficult but on the other hand I’m kind of lead to believe that the point of projects is to give students that difficulty and force them to find alternative ways of doing things. I personally find this a bit unappealing because it doesn’t lead to good ways of doing things. I personally feel that my approach was pretty brute-force-hack-and-slash but given the time crunch I could manage full functionality. Also on top of that this project was meant to built out of the midterm project which hasn’t been graded or given any feedback so if there were issues or anything like that I didn’t catch this greatly minimized my chances of fixing them.

In hindsight, I feel like because I had to spend so much time getting wired on the back-end I couldn’t do what I really wanted with the UI. I had no time to spend on the css or JavaScript side which I feel is a bit of shame but the classes focus is primarily backend so I can understand the lack of focus on that but with JavaScript being such an important factor in web development I was slightly disappointed at the lack of attention from the project.

In conclusion, I’m glad that I decided to do the project, it was hard but I learned a lot from working on it about SQL, and how c# can work so closely with it. Plus, I walked away with a project that I can tweak in my personal time to do more things.