Matthew O'Brien
CS300 Final Write Up
6/4/2015

The overall structure of my code was very simple thanks to how Netbeans handles the GUI java classes. Since this was the first time I've ever created any sort of GUI, I'm not sure if this is a normal set up for GUIs or if it was even anything good. I found it to be very easy to use, and was able to keep it nice and simple. I changed my coding significantly from my original design. I took it upon myself to try and use a database. I have never used any sort of database before, so I originally strayed away from trying it. But after some research and suggestion from classmates, I decided to give it a shot. It made everything so much easier, after learning how it all works that is. I broke the project down into three simple classes. The first class handles my database, and all of the methods dealing with the database. Then there is the GUI class to create the GUI and use the methods of the previous class. And the final class is just the main class that creates/connects to the database, and creates the GUI for the user. My original plan was considerably more complex, and had it's own data structure with external files to save and load from, but with mySQL and the online database, I had no reason for any of those. My Storage.java class handles all of the functionality of the project, and the GUI class handles all of the interactions with the user for those methods.

I actually had a lot of problems initially with the GUI. Since I had never made my own GUI before, it took a little bit of time to really understand what I was doing. I watched a lot of tutorials to figure out how they actually worked. I made a few different test programs, which did some very basic math in order to figure out how a GUI was actually used. I connected the idea of selecting a number from a text menu, which would then perform the function to the buttons of the GUI. After I understood that concept of placing those methods within the button methods, it was easy from there. Netbeans allowed for the creation of the frames, and since I am only returning strings for each field, it was very easy to display and change the data. Another rather large issue I had was actually using mySQL. That took the majority of the time for me to get this project up and running. I actually couldn't get the bin file to load into my libraries for a long time, and that led me down a crazy rabbit hole. After getting the libraries to add correctly, it was just a matter of learning the syntax for the given methods of mySQL. I only need to make a basic add, retrieve, and delete methods. I was able to manipulate those methods or slightly change the retrieve, to fit all of my basic needs for the buttons of the GUI. Since the table is stored online, I never needed to deal with any sort of local saves for the information, and it was all easy to access from anywhere with internet access. It allows for multiple users very easily, and each user has their own set of tasks.

My testing was quite simple. All of the functions were just passing around strings or concatenated strings. Because of this, I was able to just feed into strings directly into the methods, and connect to the database to make sure they were there. I also implemented an automatic refresh for all of the lists whenever a user logs in, adds a task, or moves a task. So at any point in time, whenever anything changes, the new set of information is given to the user. Granted, I've read that that is not always the best idea. The least amount of interactions with the database is considered the best. But since this was on such a small scale, I sort of ignored that idea.

I didn't really add anything additional or special to the project outside of the requirements. I decided to challenge myself with using MySQL instead of trying to add anything outside of the requirements. The original idea of learning how to make GUIs work and learn MySQL sort of led me away from trying anything else new. I'm very glad I went with the online database, it was a very interesting concept to learn, and I'm looking forward to altering this project in the future for something a little more useful.