Thea Gregory 5291240 -- Java Project Final Report

Introduction

This project was intended to create an application to manage a restaurants, clients, orders, and deliveries. It is interconnected and runs on using the powerful MVC system and OOP concepts. It was highly complex, and required fine attention to detail and a great amount of resourcefulness to complete.

This report chronicles the creation of this project. It includes class diagrams, screenshots, and my own thoughts about the three week design and building process. In this report I hope to show my diligence and hard work. This project was one of the most complex problems I’ve ever had to solve, and I want to document it as such.

Analyzing the Project

My plan was to do one use case at a time. I began with Login at the suggestion of the instructor. It gave me a feel for the MVC logic and let me create a look and feel for the SWAG portion of the project. I only moved on to another use case when the one before was completed to my satisfaction. I used a MySQL database to store and manage my data because I was familiar with it, and I enjoy working with it.

Class Diagram attached with this report due to size. Each use case is its own MVC unit and there are multiple pages of UML diagrams.

Implementation of the Project

I began by spending the first two days of the project planning and drawing UML diagrams. I also planned my database and what needed to be in it. This helped me greatly because I had a clear image in my head of what was needed to move forward with the project and what it would entail.

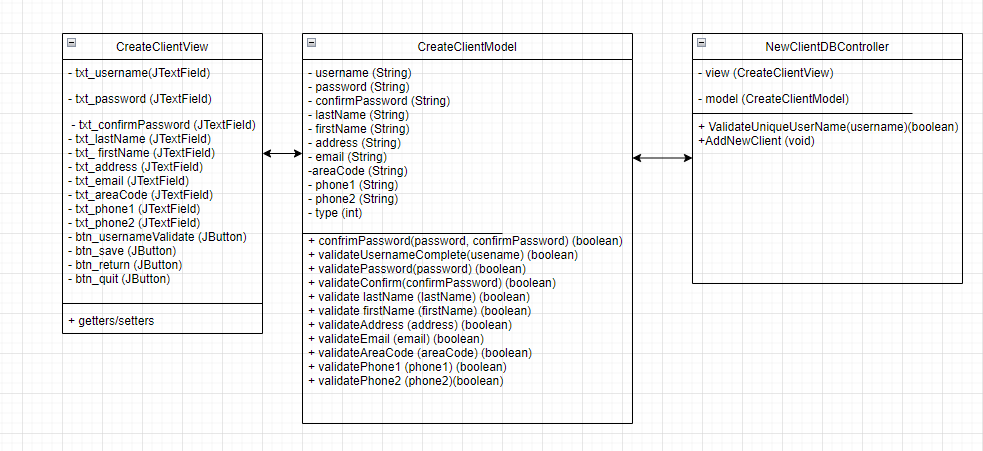
After login, I worked on adding/editing/deleting a client, because of its simplicity. I wanted to practice my MVC skills, as I was not confident in them at first. I spent a good deal of time doing research, as a great deal of this project resembles itself.

I enjoyed working on my DBManager class, which holds all of my database related functions. It was incredible to see what I learned, as I’d had to teach myself how to operate mysql in Java from online resources. The longest part were the add/edit/delete restaurant use cases, as the comboboxes took quite a bit of time to validate and arrange in the view and controller. The model also had quite a big of logic in it that needed to be kept comprehensible.

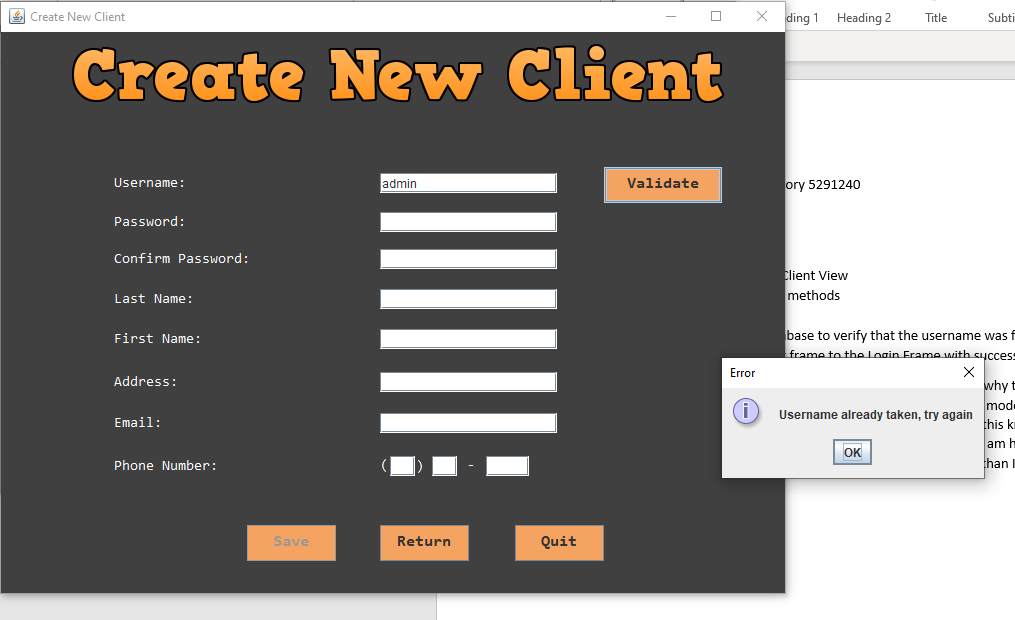
I began to pick up speed at the end, as I became familiar with JTables. I taught myself how to user ListSelectionListeners, so I wouldn’t need an extra button to make selections from the JTable. It is much cleaner and more efficient. There was a bit of a learning curve, but I was able to pull it off.

Screenshots:

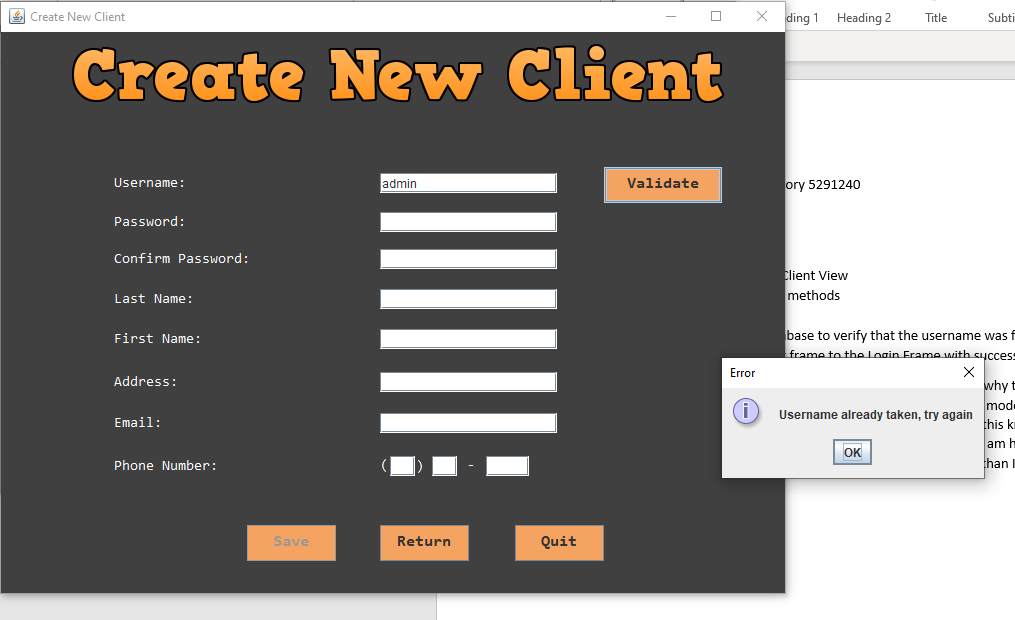
Some UML (complete file to be attached)



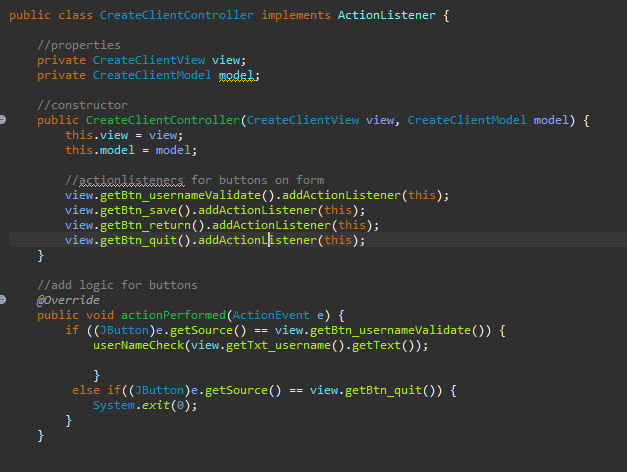
An example of my username validation feature:



A code sample for my early validation methods:



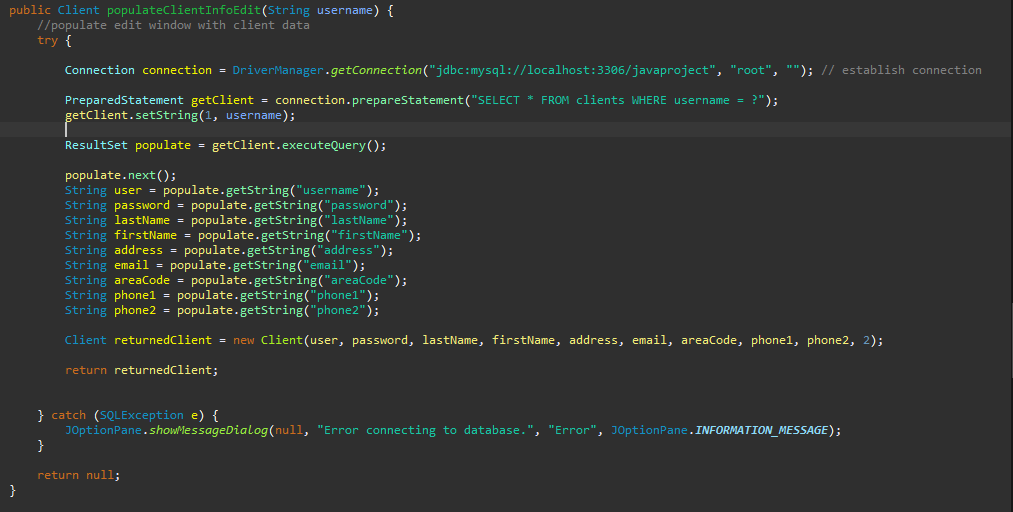
Actionlisteners!



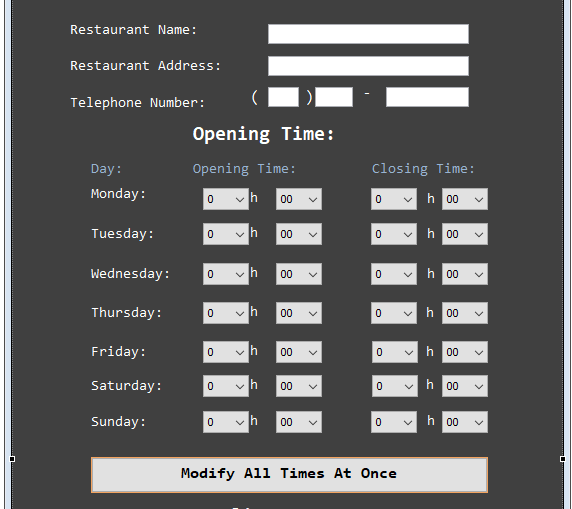
The check username DB function:



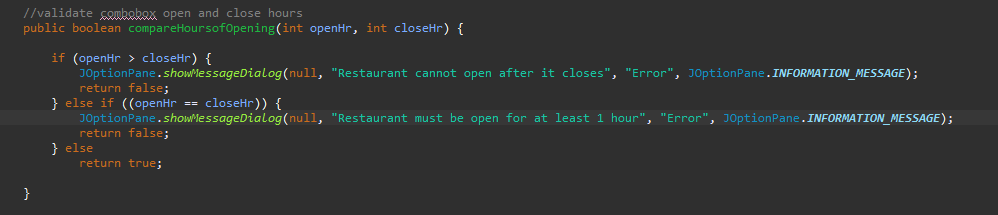
DB Code to populate textboxes in the edit client window:



Comboboxes for opening hours!



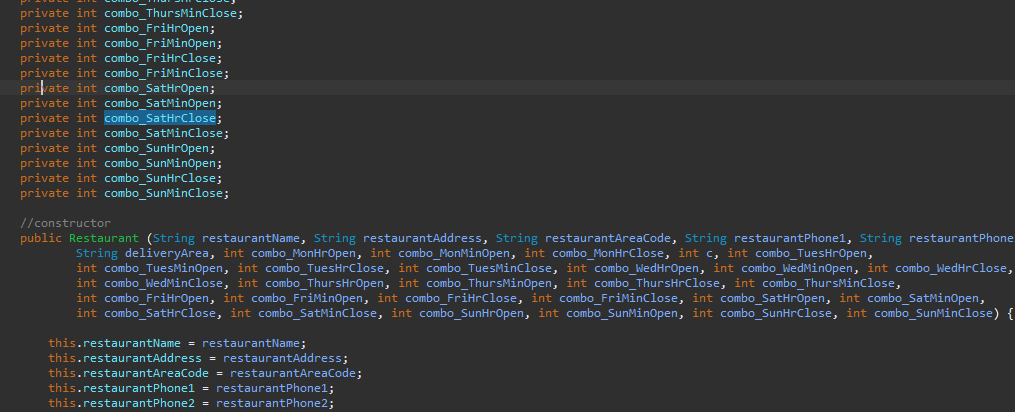
Combobox validation:



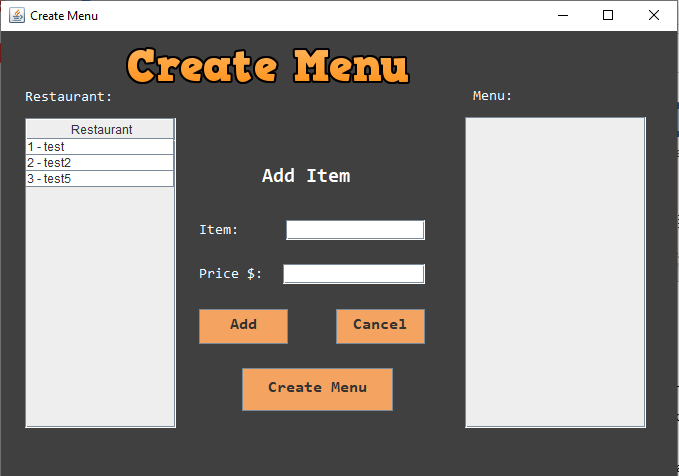
A sample from the restaurants table:



The constructor for the Edit Client use case



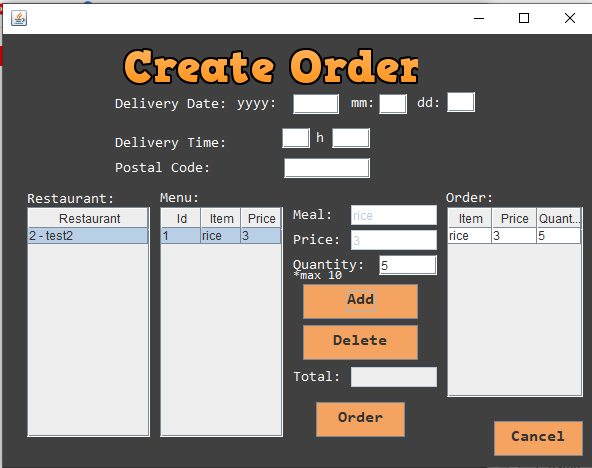
Create Menu, with JTables



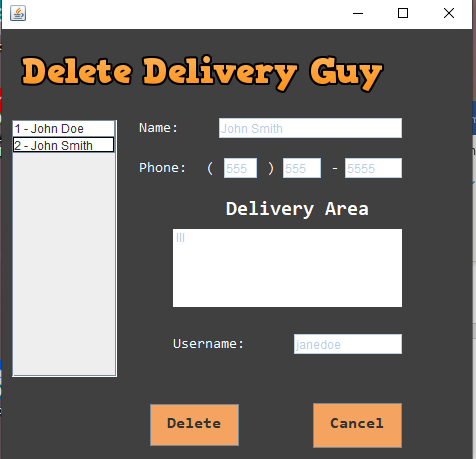
Code to display the restaurant’s menu in the right screen:



Create Order



Delete Delivery Guy:



Add new Order code:



What Happened Throughout the Project?

I learned a great deal. There was frustration, and there was triumph. I found that I could be self-reliant, and gained a great deal more confidence in my abilities as a programmer. I struggled, and spent time searching the Internet for hints and solutions. I read documentation. I experimented. I can now enter a foreign key from memory and handle a database with many tables.

Problem, Encounter, Solution?

I hit many problems along the way. Here is a sampling:

* I had issues with brackets sometimes. I’d have code outside of its scope and then wonder why it wouldn’t work. The debugger was very well used during this project.
* JTables have a learning curve. I spent a lot of time reading Oracle’s documentation and Stack Overflow to get the hang of it.
* Starting out, I was still having trouble with the concept of MVC. With some exposure and starting small, as well as some reading on the internet I was able to understand the concept.
* I forgot to set one table to InnoDB, and realized my mistake when I googled error 1215.
* JTextArea formatting can be challenging if you’re not used to working with them. It also helps to name them properly.
* Reading the use cases is essential, as the provided guide images often lacked complete information and it can be easy to be put into a false sense of security.
* Manage your alertness and fatigue because it will slow you down. Rest and take care of yourself!

Conclusion

I finished the project. I would have liked more time to give it more polish, but it works and the look is decent. It was a sprint to the finish and a very challenging undertaking, but I’m glad I stuck it out and completed it.

I would have spent less time on the UML and dove into the use cases a day earlier. That would have given me more time to finish and made the final week less intense. I devoted a lot of time and energy to this project, but I would like to be more efficient and fast in the future. I’m a slow coder and it showed in this project.