Matthew Ottomanelli

Prof. Arias

Software Development 1

2 May 2018

Final Writeup

Abstract:

For my final project I initially chose to make a Nike bot that would automatically add a sneaker to your cart. However, after doing extensive research on my topic, I realized it would be too difficult to make in java. I decided to change my topic to a healthcare database in which you can add, search, and pull information on both clients and companies. I initially struggled with the MySQL part of my database but after doing more research I quickly learned how to use it to effectively store all my data.

Introduction:

After realizing my initial topic was too difficult, I chose a healthcare database. This topic was important to me because my dad is in the healthcare industry and has extensive knowledge on the business behind it. I did not have much knowledge of a database or how it works at first but then, after researching, I discovered MySQL. I used MySQL because it is free, well-documented, and has a nice Graphical User Interface (GUI). I had many problems throughout the development of my project but, through trial and error, I eventually came to create a database that I can honestly say I am proud to have made.

Detailed System Description:

My java project includes four classes including Person, Company, Driver, and Connector. The Person class collects data on people, or clients, of the individual healthcare company it is used by. It collects data such as, the client's name, gender, salary, age and whether or not they have health insurance. This was done by creating String for the name, a Boolean for the gender (true=male, false=female), a double for salary, an int for age, and a Boolean for hasHealthInsurance. From there, I began creating getters and setter for each variable. For example, the getter for name is:

```
String getName() {
      return this.name; }.
```

While the setter for name is:

```
void setName(String updatedName) {
      this.name = updatedName; }.
```

I then did the exact same for the Company class; which collects data on the health insurance company itself like the name of the company, the company's budget, the amount of clients they have, the location of their office, and the number of employees they have. Just as I did for the Person class, I created variables for each of these data fields including a String for name, a double for budget, an int for clientNum, a String for officeLoc, and an int for employeeNum. Then, I made the getters and setters for each of these.

Next, I began working on a way to store the information that the user can input. I found that the best way for me to accomplish this was to use MySQL. I went onto MySQL's website and downloaded a java connector. The installation was easy because of their instructions on the website. In order to import 'java.sql.Connection', I downloaded a .jar file from their website,

right clicked on my project name, clicked java build path, selected the .jar file, then imported the packages and called for the import. Then I used the examples on their website to establish a connection with my database by copying and pasting a link on line 22 of my Connector class. However, I had to change the end of the link to make it more easily accessible. The link initially ended in '=greatsqldb' but I addeded '&useSSL=false' so that MySQL's warning about not using encryption would disappear. This would normally be found upon in a real company's database due to security encryption issues; however, I don't need to worry about that since my database isn't going into market any time soon. I created the Connector class to establish a connection between the user input and my database on MySQL. In this class, I also greatly extended my knowledge of try/catch blocks. This helped me to understand all my mistakes from the error messages.

At first, I struggled with databases in general due to a lack of knowledge on the subject, however, websites like stack overflow gave me simple solutions to most of the errors I encountered throughout my development. In MySQL, I created a table in my workbench for both Persons and Companies. Then, in my Connector class I queried my data into the SQL table by using the 'INSERT INTO' command in a text file. Then, in my Driver class I used the scanner class to store my user input. I prompted the user to create either a person or company and stored them into the accompanying variable using the appropriate data type. The new person or company was created using the person or company constructor; whichever they chose. Then after the user answers the questions they are prompted to answer, I went into the workbench and refreshed to see if the data was successfully stored in the tables.

Requirements & Literature Survey:

My program effectively stores a company's data on their clients and individual company information. It solves the problem of trying to store a large amount of company records into one place while quickly finding any kind of data you are looking for. My program eliminates the issue of storing several paper files into a large warehouse and going through, trying to find whatever you're looking for. All the data is in one program and is much quicker to find. There are many databases like mine on the market, so it doesn't necessarily fix any problems that are not already solved but it does do what it is supposed to efficiently. Companies like IBM and Google have their own types of cloud storage that can easily hold the same data that my program can hold, however mine is specific to a Healthcare company.

User Manual:

My program is meant to be used to store a company's data in which you can insert information like a person's name, gender, salary, health insurance status (whether they have it or not), and their age. You can also insert information of a company like the company's name, budget, number of clients they have, their office location, and the number of employees they have. This information gets stored into MySQL and can easily be searched through for whatever information you wish to see. The data can also be altered easily. For example, if somebody who used to not have health insurance ends up getting it, you can easily update their data and change the 'hasHealthInsurance' variable to true.

Conclusion:

As a result of this project, I have developed far more knowledge in the software development field. I gained much more knowledge in databases, MySQL, try/catch blocks, and the switch command. I created a database that I am proud of making that efficiently does exactly

what it is meant to do; prompt the user to insert data and store it into a table on MySQL. Given more time, I would probably add more data fields like the person's home city and home state and possibly even more. I am happy with what I created and can honestly say I've learned more about developing a program because of this project.