Assignment 2: Networking, Databases, and Multithreading (90 points; due Tuesday, May 26th at 11:59 PM)

So far, so good! The bank is impressed that you've taken an existing GUI done in Swing and made the successful conversion to JavaFX. In fact, they took your conversion and made it into a single user system and have the start of what will act like an ATM system. They're looking for a little more work from you though, as they had no knowledge of networking, databases, or multithreading. The design of a bank account was done where the output was written into a file. However, the system is in serious need of an upgrade to the 21st century and, more specifically, the year 2020. Here's what they'll need:

- For the Java-based client, it will first connect to the server and provide a username and password to login. This login will be checked against the MySQL database for validity. Here is the MySQL database login information:

Server Name	sql3.freemysqlhosting.net
Database Name	sql3336223
User Name	sql3336223
Password	C$23Ass!gnment#2
Table	users
Fields	username(varchar(20)), password(varchar(50))

  There were two dummy accounts created to test this:

User Name	**Password**
John Doe	1234
Jane Doe	5678

  The ATM removed the File features, but left in the Add An Account feature so you could add new accounts to test on. This will need to be removed at the end. The bank account update should be sent to the server after Add An Account feature, as well as when any deposit or check transaction is made.
- The interfacing for the new ATM client will be done to a Java-based server that you will need to develop. A suggestion will be to make two threads: one for handling the user name/password and the other to handle the updating of the bank account objects.
- Rather than data being written into a file, the output needs to be written into a MySQL database. The Java-based server will connect to the database to store the information. This Java-based server should also be taking in connections from Java-based clients through using the Socket

and ServerSocket classes. Here is the information for the **account** table which will hold the data for the checking accounts:

Table Name	account
Field 1	username (varchar(20))
Field 2	acct (blob)

As a suggestion, looking at folders 4 and 5, specifically the projects for the Tic Tac Toe client and server, as well as the Simple JDBC should be a good starting point. Also, you'll need the attached library mysql-connector-java-8.0.17.jar to add to your project for using Java with the MySQL database.

For storing the blob, here's a starting point:

http://www.java2s.com/Code/Java/Database-SQL-JDBC/Storeandretrieveanobjectfromatable.htm

Finally, the conceptual diagram discussed in class is also available.