בס"ד

Yossi Silberhaft 210028924 89-281-04
Daniel Blinick WN654676 89-281-04
Binyamin Greenberg 200220671 89-281-04
Steven Lapp 204785240 89-281-05

READ_ME

The idea of our project is to create a GUI that connects to a SQL server provided by the user and allows the user to execute both DDL and DML queries on the server. In order to provide this capability we created 4 classes:

1) DBProject – class that holds the main, in charge or using the various other classes to, read the config file, create the GUI, and start it running
2) ReadConfig – class that reads the config file and returns the information in the form of a DBConnectionInfo object to the main
3) DBConnectionInfo- class that holds the information about the server
4) FrontPage-class that is used by the main to receive queries from the user and send them to server to be executed, returning the answer to the user where appropriate

In order to get a better understanding of how the program runs, we will now take a closer look at how each class operates and accomplishes its task.

The first of the aforementioned classes, DBProject, is home to the main method. The main method begins by creating a GUI in the form of a JFrame object. It then creates a readConfig object, to which it passes the config file that it received as a parameter, and the readConfig object extracts the information from the file, returning it to the main in the form of a DBconnectionInfo object. The main then opens a try and attempts to connect to the server provided by the config file, using the name and password, also provided by the config file. If the connecting succeeds, it then creates a FrontPage object which runs the GUI and allows for the passing of queries to the server and the posting of the results. If the main fails to connect to the server, however, it moves to the catch and prints the stack trace. The main is the only method that this class has.

The ReadConfig class is of course in charge of reading the configuration file. Its getConnectionInfoFromFile method receives the name of the config file as a parameter and creates in inputStream from the file. It then opens a try and attempts to create a buffered reader from that stream. If it succeeds, it then reads the 3 lines of the config file and stores the url, user name, and password in 3 strings. If it fails it prints the stack trace. Finally, it closes the file from which it read and returns a newly created DBConnectionInfo to which it passes the 3 strings holding the url to the server, the username, and the password.

The DBConnectionInfo is a very simple class. It is only used for storing the url, username, and password of the SQL server that we are going to use. All it consists of are 3 strings, a constructor in which it sets the 3 strings, and 3 getters.

The FrontPage class is what provides the functionality to the GUI. When first created, this class runs a method called initComponents when it sets up the GUI with a text editor to write in, 2 buttons for DDL and DML queries, and a table for displaying results. The buttons are programmed so that when they are clicked they will execute their respective commands.

When the DDL button is clicked, it retrieves the text from the text editor and checks it to see if the words SELECT or SHOW are in the string. If they are, then it returns a WRONG QUERY STRUCTURE error, because clearly the user meant to execute a DML query. Barring this scenario, the method then opens a try in which it sends the query to the server. If

Yossi Silberhaft 210028924 89-281-04
Daniel Blinick WN654676 89-281-04
Binyamin Greenberg 200220671 89-281-04
Steven Lapp 204785240 89-281-05

everything works fine and the server does not throw and exception, then the method simply ends. However, if the server does throw and exception, then we check whether the exception is a result of a syntax error and if it is we return a WRONG QUERY STRUCTURE error. Otherwise we know that the syntax is correct so it must be a logical error and we return a LOGICAL ERROR error.

When the DML button is clicked, something similar happens, with a few differences. Before we send anything to the server we first clear the table of all its pervious data. Then we get the string from the text editor, but here we make sure that the string contains either the word SHOW or the word SELECT, because we want to make sure the user did indeed intend for this to be a DML query. If it does not contain those words, then we return a WRONG QUERY STRUCTURE. Otherwise send the query to the SQL server. If it returns and exception then we do the same thing as we did with the DDL query, checking if it is a problem with the syntax or the logic and responding appropriately. Otherwise, since we executed a DML query, we receive a table in response from the server as a string. We then copy the table that we received as a string into vectors and create a new JTable with those vectors. After that we display the JTable in the GUI.

That explains the classes that we created and how they each run. Below the UML shows the structure visually: