File - DatabaseTools.java

```
 1 import java.sql.*;
 2 import java.util.ArrayList;
 3
 4 public class DatabaseTools {
 5     public static Connection getConnected() {
 6         Connection conn = null;
 7
 8         try {
 9             //Load MySQL database driver
10             Class.forName("com.mysql.jdbc.Driver");
11
12             // Connection string
13             String dbURL = "jdbc:mysql://localhost:
   3306/max";
14
15             // Credentials
16             String username = "root";
17             String password = "mysql";
18
19             // Set connection to database to
   Connection object
20             conn = DriverManager.getConnection(
   dbURL, username, password);
21         } catch (ClassNotFoundException e) {
22             e.printStackTrace();
23         } catch (SQLException e) {
24             e.printStackTrace();
25         }
26
27         return conn;
28     }
29
30     public static void insert(String fn, String ln,
```

File - DatabaseTools.java

```
30  String un, String pw) {
31          // Connection to database;
32          Connection myConn = DatabaseTools.
   getConnected();
33
34          PreparedStatement ps = null;
35
36          // Insert query
37          String queryInsert = "INSERT INTO tbl_users
    (user_fn, user_ln, user_un, user_pw) VALUES
    (?, ?, ?, ?);";
38
39          try {
40              // Set PreparedStatement object to
   instance with query
41              ps = myConn.prepareStatement(
   queryInsert);
42
43              // Set values for parameters
44              ps.setString(1, fn);
45              ps.setString(2, ln);
46              ps.setString(3, un);
47              ps.setString(4, pw);
48
49              // Execute insert on database
50              ps.execute();
51          } catch (SQLException e) {
52              e.printStackTrace();
53          } finally {
54              DatabaseTools.closePreparedStatement(ps
   );
55              DatabaseTools.closeConnection(myConn);
56          }
```

File - DatabaseTools.java

```
57      }
58
59      public static ArrayList<User> selectAllUsers()
    {
60          // Connection to database
61          Connection conn = DatabaseTools.
    getConnected();
62
63          PreparedStatement ps = null;
64          ResultSet rs = null;
65          ArrayList<User> myUserList = new ArrayList<
    User>();
66
67          // Select statement
68          String selectQuery = "SELECT user_id,
    user_fn, user_ln, user_un, user_pw FROM tbl_users;"
    ;
69
70          try {
71              // Create prepared statemnent object
72              ps = conn.prepareStatement(selectQuery)
    ;
73
74              // Execute query and return result set
75              rs = ps.executeQuery();
76
77              // Loop through all rows in result set
78              while (rs.next()) {
79                  // Create user object to add to
    list.
80                  User user = new User();
81
82                  // Add values for row of result set
```

File - DatabaseTools.java

```
 82  to properties of User object
 83                  user.setIdUser(rs.getInt("user_id"
    ));
 84                  user.setFnUser(rs.getString("
    user_fn"));
 85                  user.setLnUser(rs.getString("
    user_ln"));
 86                  user.setUnUser(rs.getString("
    user_un"));
 87                  user.setPwUser(rs.getString("
    user_pw"));
 88
 89                  // Add user to list
 90                  myUserList.add(user);
 91              }
 92          } catch (SQLException e) {
 93              e.printStackTrace();
 94          } finally {
 95              DatabaseTools.closePreparedStatement(
    ps);
 96              DatabaseTools.closeResultSet(rs);
 97              DatabaseTools.closeConnection(conn);
 98          }
 99
100          return myUserList;
101      }
102
103      public static void closePreparedStatement(
    Statement ps) {
104          try {
105              if (ps != null) {
106                  ps.close();
107              }
```

File - DatabaseTools.java

```
108         } catch (SQLException e) {
109             e.printStackTrace();
110         }
111     }
112 
113     public static void closeConnection(Connection
    conn) {
114         try {
115             if (conn != null) {
116                 conn.close();
117             }
118         } catch (SQLException e) {
119             e.printStackTrace();
120         }
121     }
122 
123     public static void closeResultSet(ResultSet rs
    ) {
124         try {
125             if (rs != null) {
126                 rs.close();
127             }
128         } catch (SQLException e) {
129             e.printStackTrace();
130         }
131     }
132 }
133 
```