

## Write-Up

### Assignment No. 3

Title : Application development using JDBC and multi-threading

Problem Statement :

Develop an application using JDBC, multi-threading concurrency, synchronous and asynchronous callbacks, threadpools using executor service.

Objectives: To

- 1) Learn database connectivity
- 2) Learn concurrency

Outcomes: We will be able to:

- 1) Implement various types of JDBC drivers
- 2) Implement multi-threading in the application.

Software and Hardware Requirements:

64-bit Fedora/Windows, Eclipse IDE, Intel i5 processor, 4 GB RAM.

Theory:

Java Database Connectivity (JDBC):

JDBC is an application programming interface (API) for java which defines how a client may access a database.

It is a part of Java Standard Edition platform from oracle corporation. It provides methods to query and

is oriented towards relational database. A JDBC to ODBC  
accessive data to source in the JAVA virtual machine  
(JVM) host environment.

### Multi-threading :

It is a technique that allows a program or a process  
to execute many tasks concurrently.

It allows a process to run its tasks in a parallel  
mode on a single process system.

In multi-threading concept, several multiple light weight  
processes run in a single process/taskbar program by  
a single processor.

Multi-threading software treats each process as a  
separate program.

eg: When you use word processor, you perform different  
tasks such as printing, spell checking, etc.

### Life cycle of a thread :

- 1) New/Born : Begins its lifecycle in a new state.
- 2) Runnable : After a newly born thread is started,  
it becomes runnable. This is considered to be  
executing its tasks.
- 3) Waiting : Thread waits for another to perform a task.
- 4) Time waiting : A runnable thread can enter time waiting  
stable state for a certain time interval.



Q. No.

Q. No.

Q. No.

5) Dead: Enters terminated state after task completion.

### Threadpool:

Collection of thread is taken from a threadpool and task is allocated.

Whichever thread gets allocated task, private creation of new thread impacts scalability.

### JDBC Driver:

JDBC Driver is a software component that enables java application to interact with the database.

The four types of JDBC drivers are -

- 1) JDBC - ODBC bridge driver
- 2) Native - API driver (partially java driver)
- 3) Network protocol driver (fully java driver)
- 4) Thin driver (fully java driver)

### Steps to connect JDBC

- 1) Register to driver class

```
Class.forName("com.mysql.jdbc.Driver")
```

- 2) Create connection object

```
Connection conn = DriverManager.getConnection(
    "jdbc:mysql://localhost/text?" + "user=
    & password= ")
```

- 3) Create statement object

```
Statement stmt = conn.createStatement()
```

Fr.  
No.

6) Execute query  
stmt.executeQuery()  
or  
stmt.executeUpdate()

5) close connection  
conn.close();

Steps to connect database:

1) Register to driver class  
class.forName("com.mysql.jdbc.Driver")

2) create connection object:  
public static connection getConnection  
(String url) throws Exception  
connection con = DriverManager.getConnection  
(String url)  
public static connection getConnection (String url, String  
name, String password) throws SQLException.

3) create statement object:  
Statement stmt = conn.createStatement()

4) Execute query:  
stmt.executeQuery (String sql);

**Sample Code:**

// Server.java

package multithread;

```
import java.io.IOException;

import java.net.ServerSocket;

import java.net.Socket;

import java.sql.*;

import java.util.concurrent.ExecutorService;

import java.util.concurrent.Executors;


public class Server {

    private static final int PORT = 9169;

    private static final ExecutorService pool = Executors.newFixedThreadPool(5);


    public static void main(String[] args) throws IOException, SQLException, ClassNotFoundException {

        String name, pass, url;

        Connection con;

        Class.forName("com.mysql.cj.jdbc.Driver");

        url = "jdbc:mysql://localhost:3306/moviedb";

        name = "root";

        pass = "mysql";

        con = DriverManager.getConnection(url, name, pass);

        ServerSocket listener = new ServerSocket(PORT);

        while(true){

            System.out.println("Waiting For Client");

            Socket client = listener.accept();

            System.out.println("Client Accepted");

            ClientHandler clientThread = new ClientHandler(client,con);

            pool.execute(clientThread);

        }

    }

}
```

```
}
```

```
//ClientHandler.java
```

```
package multithread;
```

```
import java.sql.*;
```

```
import java.io.*;
```

```
import java.net.Socket;
```

```
import java.util.ArrayList;
```

```
import java.util.Random;
```

```
public class ClientHandler implements Runnable {
```

```
    private Socket client;
```

```
    private BufferedReader in;
```

```
    private PrintWriter out;
```

```
    Connection con;
```

```
    ArrayList<String> usernames = new ArrayList<>();
```

```
    Statement stmt;
```

```
    ArrayList<String> movieswatched = new ArrayList<>();
```

```
    ArrayList<String> moviesnotwatched = new ArrayList<>();
```

```
    boolean contain, passeq;
```

```
    public ClientHandler(Socket clientSocket, Connection con) throws IOException, SQLException {
```

```
        this.client = clientSocket;
```

```
        this.con = con;
```

```
        stmt = con.createStatement();
```

```
        in = new BufferedReader(new InputStreamReader(client.getInputStream()));
```

```
        out = new PrintWriter(client.getOutputStream(), true);
```

```
    }
```

```
@Override
```

```

public void run() {

    try {

        // Getting the usernames

        ResultSet rs = stmt.executeQuery("SELECT username FROM users");

        while (rs.next()) {

            usernames.add(rs.getString(1));

        }

        while (true) {

            Statement stmt = con.createStatement();

            String request = in.readLine();

            // Checking for socket close case

            if (request.equals("911")) {

                try {

                    in.close();

                    out.close();

                    client.close();

                } catch (IOException e) {

                    e.printStackTrace();

                }

                break;

            }

            // Log In request

            if (request.equals("1")) {

                String username = in.readLine();

                String password = in.readLine();

                // Check if username exists

                contain = usernames.contains(username);

                if (contain) {

                    // Username existed. check for password

                    out.println("1");
                }
            }
        }
    }
}

```

```

        rs = stmt.executeQuery("SELECT password FROM users WHERE username=\"\" +
username + "\"");
        rs.next();
        passeq = password.equals(rs.getString(1));
        if (!passeq) {
            // Password didn't match
            out.println("0");
        } else {
            // Password matched
            out.println("1");

            while (true) {
                String clientresp;
                clientresp = in.readLine();
                // Sign out case
                if (clientresp.equals("3")) break;
                // Rewatch Case
                else if (clientresp.equals("2")) {
                    rs = stmt.executeQuery("SELECT MovieName FROM movietable WHERE " +
username + "\"Y\"");
                    while (rs.next()) {
                        movieswatched.add(rs.getString(1));
                    }
                    String randomMovie;
                    String clientres;
                    while (!movieswatched.isEmpty()) {
                        int index = new Random().nextInt(movieswatched.size());
                        randomMovie = movieswatched.get(index);
                        out.println(randomMovie);
                        clientres = in.readLine();
                        if (clientres.equals("1")) break;
                        movieswatched.remove(index);
                    }
                }
            }
        }
    }
}

```



```

    }

    if (movieswatched.isEmpty()) out.println("404");
}

// New Recommendation Case
else {
    rs = stmt.executeQuery("SELECT MovieName FROM movietable WHERE " +
username + "=\"N\"");
    while (rs.next()) {
        moviesnotwatched.add(rs.getString(1));
    }
    String randomMovie = "";
    String clientres;
    int flag = 0;
    while (!moviesnotwatched.isEmpty()) {
        int index = new Random().nextInt(moviesnotwatched.size());
        randomMovie = moviesnotwatched.get(index);
        out.println(randomMovie);
        clientres = in.readLine();
        if (clientres.equals("1")) {
            flag = 1;
            break;
        }
        moviesnotwatched.remove(index);
    }
    if (moviesnotwatched.isEmpty()) out.println("404");
    if (flag == 1) {
        stmt.executeUpdate("UPDATE movietable set " + username + "=\"Y\" where
MovieName=\"" + randomMovie + "\"");
    }
}
}
}

```

```

        }
    }
    // Username didn't match
    else {
        out.println("0");
    }
}

// Sign Up Case
if (request.equals("2")) {
    String usernamesignup = in.readLine();
    String passwordsignup = in.readLine();
    if (usernames.contains(usernamesignup)) {
        // Username Already Exists
        out.println("0");
    } else {
        usernames.add(usernamesignup);
        stmt.executeUpdate("ALTER TABLE movietable ADD " + usernamesignup + "
varchar(1);");
        stmt.executeUpdate("UPDATE movietable SET " + usernamesignup + "=\"N\"");
        stmt.executeUpdate("INSERT INTO users values(\"" + usernamesignup + "\",\"" +
passwordsignup + "\");");
        out.println("1");
    }
}

}

} catch (Exception e) {
    e.printStackTrace();
} finally {
    try {
        in.close();
        out.close();
    }
}

```



```

out.println(keyboard.readLine());

System.out.print("Enter password: ");

out.println(keyboard.readLine());

String usernameresponse = reader.readLine();

if (usernameresponse.equals("0"))

    System.out.println("Account with this username doesn't exist!");

else {

    String passwordresponse = reader.readLine();

    if (passwordresponse.equals("0"))

        System.out.println("Incorrect Password");

    else {

        System.out.println("Logged In Successfully");

        String loginchoice;

        while (true) {

            System.out.println("\n1. A new movie recommendation.\t\t2. A rewatch
recommendation\t\t3.Sign out and exit");

            loginchoice = keyboard.readLine();

            switch (loginchoice) {

                case "3":

                    out.println("3");

                    choice = "911";

                    break label;

                case "2":

                    out.println("2");

                    String rewatch;

                    String rewatchchoice;

                    while (true) {

                        rewatch = reader.readLine();

                        if (rewatch.equals("404")) {

                            System.out.println("We're out of movies :(");

                            break;

```



```

    }
    System.out.print("\nThe rewatch recommendation is: ");
    System.out.println(rewatch);
    System.out.print("Would you like a different recommendation?(y/n) ");
    rewatchchoice = keyboard.readLine();
    if (rewatchchoice.equals("y")) out.println("0");
    else {
        out.println("1");
        break;
    }
}
break;
case "1":
    out.println("1");
    String recomm;
    String recommchoice;
    while (true) {
        recomm = reader.readLine();
        if (recomm.equals("404")) {
            System.out.println("We're out of movies :(");
            break;
        }
        System.out.print("\nThe movie recommendation is: ");
        System.out.println(recomm);
        System.out.print("Would you like a different recommendation?(y/n) ");
        recommchoice = keyboard.readLine();
        if (recommchoice.equals("y")) out.println("0");
        else {
            out.println("1");
            System.out.println("We hope you enjoy the movie!!!!");
            break;
        }
    }
}

```

```

        }
    }
    break;
}
}
}

}

break;
case "2":
    out.println(2);
    System.out.print("Enter username: ");
    out.println(keyboard.readLine());
    System.out.print("Enter password: ");
    out.println(keyboard.readLine());
    String serverresp = reader.readLine();
    if (serverresp.equals("0")) {
        System.out.println("Username Already Exists!");
    } else {
        System.out.println("Account Created Successfully!!");
    }

    break;
case "911":
    out.println(911);
    keyboard.close();
    reader.close();
    out.close();
    socket.close();

    break;
default:

```

```

        System.out.println("Enter Valid Input");
    }

    } while (!choice.equals("911"));

    out.println(911);

    keyboard.close();

    reader.close();

    out.close();

    socket.close();

}

}

```

## Outputs

```

"D:\Program Files\Java\jdk-12.0.1\bin\java.exe" "-javaagent:D:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea
Assignment3 \src\ multithread ClientHandler run
Server.java ClientHandler.java Client.java
Run: Server Client Client
1.Login 2.Sign Up 911.Close the Socket
Enter username: username
Enter password: password
Account Created Successfully!!
-----WELCOME-----
1.Login 2.Sign Up 911.Close the Socket
Enter username: username
Enter password: password
Logged In Successfully

1. A new movie recommendation. 2. A rewatch recommendation 3.Sign out and exit
The movie recommendation is: Ouija
Would you like a different recommendation?(y/n)
We hope you enjoy the movie!!!!

1. A new movie recommendation. 2. A rewatch recommendation 3.Sign out and exit
The movie recommendation is: Pirates of The Caribbean

```

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Assignment3 - ClientHandler.java
Assignment3 src \multithread ClientHandler Client.java
Run: Server Client Client
We hope you enjoy the movie!!!!
1. A new movie recommendation. 2. A rewatch recommendation 3.Sign out and exit
The movie recommendation is: Pirates Of The Carribean
Would you like a different recommendation?(y/n)
We hope you enjoy the movie!!!!
1. A new movie recommendation. 2. A rewatch recommendation 3.Sign out and exit
The movie recommendation is: Once Upon A Time In Hollywood
Would you like a different recommendation?(y/n)
The movie recommendation is: Django Unchained
Would you like a different recommendation?(y/n)
We hope you enjoy the movie!!!!
1. A new movie recommendation. 2. A rewatch recommendation 3.Sign out and exit
The rewatch recommendation is: Django Unchained
Would you like a different recommendation?(y/n)
```

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Assignment3 - ClientHandler.java
Assignment3 src \multithread ClientHandler Client.java
Run: Server Client Client
1. A new movie recommendation. 2. A rewatch recommendation 3.Sign out and exit
The movie recommendation is: Once Upon A Time In Hollywood
Would you like a different recommendation?(y/n)
The movie recommendation is: Django Unchained
Would you like a different recommendation?(y/n)
We hope you enjoy the movie!!!!
1. A new movie recommendation. 2. A rewatch recommendation 3.Sign out and exit
The rewatch recommendation is: Django Unchained
Would you like a different recommendation?(y/n)
The rewatch recommendation is: Ouija
Would you like a different recommendation?(y/n)
1. A new movie recommendation. 2. A rewatch recommendation 3.Sign out and exit
Process finished with exit code 0
```



```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Assignment3 - ClientHandler.java
Assignment3 src \ multithread ClientHandler Client.java
Run: Server Client Client
"D:\Program Files\Java\jdk-12.0.1\bin\java.exe" "-javaagent:D:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea
-----WELCOME-----
1.Login      2.Sign Up      911.Close the Socket
Enter username: user1
Enter password: user1
Logged In Successfully

1. A new movie recommendation.      2. A rewatch recommendation      3.Sign out and exit
The movie recommendation is: Once Upon A Time In Hollywood
Would you like a different recommendation?(y/n)
We hope you enjoy the movie!!!!

1. A new movie recommendation.      2. A rewatch recommendation      3.Sign out and exit
Process finished with exit code 0
```

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Assignment3 - ClientHandler.java
Assignment3 src \ multithread ClientHandler Client.java
Run: Server Client Client
"D:\Program Files\Java\jdk-12.0.1\bin\java.exe" "-javaagent:D:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea
Waiting For Client
Client Accepted
Waiting For Client
Client Accepted
Waiting For Client
```