Description

The goal of this exercise is to write a simple Java Client / Server program with database access through which you will obtain hands-on experience in fundamental communication concepts. Specifically, you are to implement a basic information exchange functionality between two remote hosts (client / server) communicating over the Internet where the server has access to a database for storing and retrieving data. You are to assume the following scenario:

- A server program waits for up to 5 (thread pool size) concurrent clients to connect to it.
- A client program that generate a series of requests towards the server based on user interaction.
- When the client program starts it establishes a new connection (socket) to the server. At the same time, the client provides a menu of options to the user as shown below:

  Welcome to Students Registration Service
  Choose one of the following options:
  1. Insert student.
  2. Show all students.
  3. Delete student.
  4. Quit service.

- Using the phpMyAdmin, please create the students table within the phpmyadmin database.
1. Choose phpmyadmin database

2. Insert table name: students

- In the students table, create the same columns as shown below:
Below, we describe each one of the menu options:

1. **Insert student**

   When the user takes the first option, the program asks for the following information: first name, last name, username and average grade as shown below:

   Give student's first name, last name, username and average grade separated by space:

   If any of the requested information is missing (i.e. only 3 input strings were given) the program replies with the following message

   Wrong input!

   followed by the previous message requesting user information.

   After successful user input, the program creates a JSON-based request (terminated by new line) to be sent to the server as shown, for example, below:

   ```json
   {"type": "insert", "firstname": "John", "lastname": "Smith", "username": "jsmith", "average": 8.7}
   ```

   Once the server receives the request, it **inserts the new student to the database** \(^1\), prepares a response message (terminated by new line) and sends it back to the client. After successful insertion to database the response is:

   ```json
   {"status": "success", "action": "tuple inserted to db"}
   ```

2. **Show all students**

   When the user takes the second option the program creates the following request to be sent to the server:

   ```json
   {"type": "select"}
   ```

   Once the server receives the request, it **retrieves all user data from the database** \(^2\), prepares the following response message, for example, and sends it to the client:

   ```json
   {"status": "success", "action": "student data retrieved", "students": [{"id": 1, "firstname": "John", "lastname": "Smith", "username": "jsmith", "average": 8.7}, {"id": 2, "firstname": "Pavlos", "lastname": "Antoniou", "username": "pantoniou", "average": 5.3}]}
   ```

   The client should print the data received as shown below:

---

\(^1\) Use classes presented in lab 3 (use insert sql query)

\(^2\) Use classes presented in lab 3 (use select sql query)
3. **Delete student**

When the user takes the third option the program asks for the id of the student to be deleted:

Give the id of the student to be deleted:

and creates the following request, for example, to be sent to the server:

```json
{"type": "delete", "id": 3}
```

Once the server receives the *request*, it **deletes the user from the database**\(^3\) and returns the following *response* message to the client:

```json
{"status": "success", "action": "student deleted"}
```

4. **Disconnect**

When the user takes the last option, the program creates the following *request* to be sent to the server:

```json
{"type": "disconnect"}
```

Once the server receives the *request*, it returns the following *response* message to the client and closes the socket:

```json
{"status": "success", "action": "connection closed"}
```

Upon reception of the server response, the client closes its own socket and terminates the program.

In case of an error in communication between the server and the database in all the cases above, the response of the server to the client will be:

```json
{"status": "error"}
```

**Notes:**

1. **Database table in MySQL must be named as students and the column names must be the same (and of the same type) as in Figure 2.**
2. For **json manipulation** we strongly recommend using the *org.json* class. Therefore, for compiling and running both the client and server programs the *org.json.jar*\(^4\) need to be in the same folder as *Client.java* and *Server.java* and in java classpath (see below).

---

\(^3\) Use classes presented in lab 3 (use delete sql query)

\(^4\) Download from [http://www.java2s.com/Code/Jar/o/Downloadorgjsonjar.htm](http://www.java2s.com/Code/Jar/o/Downloadorgjsonjar.htm) (final release API)
3. For connecting server to the mysql database, we strongly recommend using the JDBC connector which needs the mysql-connector-java-8.0.19.jar to be present in the same folder as Server.java and in Java classpath (see below).
4. You will compile Server source code as shown below

   javac -cp .:/mysql-connector-java-8.0.19.jar:/org.json.jar Server.java

5. You will compile Client source code as shown below

   javac -cp .:/org.json.jar Client.java

6. The server will have as an address the IP of the machine it is running on and the port will be given to it as a command line argument e.g.

   java -cp .:/mysql-connector-java-8.0.19.jar:/org.json.jar Server 9999

7. The client must know the server’s IP address and port given as command line arguments. Thus, the client program must be executed as follows:

   java -cp .:/org.json.jar Client 217.45.32.45 9999

8. Both client and server programs must print on the screen the contents of every message received as well as the reception datetime (using java.util.Date class) as shown in the program execution example in the last page of this assignment.

General Instructions:

1. Create a folder named Assignment1 that will contain your Java code (only Server.java and Client.java files) as well as the .sql file exported from phpmyadmin that contains the students table. The process of exporting the mysql table is shown in the next figure.

---

5 Download from https://dev.mysql.com/downloads/connector/j/ (platform independent file)
2. Zip the folder Assignment 1 and submit it via Moodle by the deadline of the assignment. Any submissions and commits after that will not be graded.

3. Avoid plagiarism. Your programs will be checked by plagiarism detection software.
   Plagiarists will be given the mark of 0 and risk further disciplinary actions.
<table>
<thead>
<tr>
<th><code>java -cp .:/mysql-connector-java-8.0.19.jar ./org.json.jar MultiThreadedTCPServer</code></th>
<th><code>java -cp .:/org.json.jar TCPClient 168.63.127.40 8083</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome to Students Registration Service</td>
<td>Choose one of the following options:</td>
</tr>
<tr>
<td></td>
<td>1. Insert student.</td>
</tr>
<tr>
<td></td>
<td>2. Show all students.</td>
</tr>
<tr>
<td></td>
<td>3. Remove student.</td>
</tr>
<tr>
<td></td>
<td>4. Quit service.</td>
</tr>
<tr>
<td></td>
<td>Given your option: 1</td>
</tr>
<tr>
<td></td>
<td>Give student's first name, last name, username and average grade separated by space:</td>
</tr>
<tr>
<td></td>
<td>John Smith jsmith 9.3</td>
</tr>
<tr>
<td></td>
<td>[Thu Jun 30 01:35:29 EET 2020] Received: {&quot;action&quot;: &quot;tuple inserted to db&quot;, &quot;status&quot;: &quot;success&quot;}</td>
</tr>
<tr>
<td></td>
<td>Choose one of the following options:</td>
</tr>
<tr>
<td></td>
<td>1. Insert student.</td>
</tr>
<tr>
<td></td>
<td>2. Show all students.</td>
</tr>
<tr>
<td></td>
<td>3. Remove student.</td>
</tr>
<tr>
<td></td>
<td>4. Quit service.</td>
</tr>
<tr>
<td></td>
<td>Given your option: 1</td>
</tr>
<tr>
<td></td>
<td>Give student's first name, last name, username and average grade separated by space:</td>
</tr>
<tr>
<td></td>
<td>Paul Antoniou panton 5.3</td>
</tr>
<tr>
<td></td>
<td>[Thu Jun 30 01:35:41 EET 2020] Received: {&quot;action&quot;: &quot;tuple inserted to db&quot;, &quot;status&quot;: &quot;success&quot;}</td>
</tr>
<tr>
<td></td>
<td>Choose one of the following options:</td>
</tr>
<tr>
<td></td>
<td>1. Insert student.</td>
</tr>
<tr>
<td></td>
<td>2. Show all students.</td>
</tr>
<tr>
<td></td>
<td>3. Remove student.</td>
</tr>
<tr>
<td></td>
<td>4. Quit service.</td>
</tr>
<tr>
<td></td>
<td>Given your option: 2</td>
</tr>
<tr>
<td></td>
<td>[Thu Jun 30 01:35:42 EET 2020] Received: {&quot;action&quot;: &quot;student data retrieved&quot;, &quot;students&quot;: [{&quot;average&quot;: 9.300000190734863, &quot;firstname&quot;: &quot;John&quot;, &quot;id&quot;: 1, &quot;lastname&quot;: &quot;Smith&quot;, &quot;username&quot;: &quot;jsmith&quot;}, {&quot;average&quot;: 5.300000190734863, &quot;firstname&quot;: &quot;Paul&quot;, &quot;id&quot;: 2, &quot;lastname&quot;: &quot;Antoniou&quot;, &quot;username&quot;: &quot;panton&quot;}], &quot;status&quot;: &quot;success&quot;}</td>
</tr>
<tr>
<td></td>
<td>ID      FName   LName   UName   Average</td>
</tr>
<tr>
<td></td>
<td>1       John    Smith   jsmith  9.300000190734863</td>
</tr>
<tr>
<td></td>
<td>2       Paul    Antoniou        panton  5.300000190734863</td>
</tr>
<tr>
<td></td>
<td>Choose one of the following options:</td>
</tr>
<tr>
<td></td>
<td>1. Insert student.</td>
</tr>
<tr>
<td></td>
<td>2. Show all students.</td>
</tr>
<tr>
<td></td>
<td>3. Remove student.</td>
</tr>
<tr>
<td></td>
<td>4. Quit service.</td>
</tr>
<tr>
<td></td>
<td>Given your option: 3</td>
</tr>
<tr>
<td></td>
<td>Give the id of the student to be deleted:</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>[Thu Jun 30 01:36:20 EET 2020] Received: {&quot;action&quot;: &quot;student deleted&quot;, &quot;status&quot;: &quot;success&quot;}</td>
</tr>
<tr>
<td></td>
<td>Choose one of the following options:</td>
</tr>
<tr>
<td></td>
<td>1. Insert student.</td>
</tr>
<tr>
<td></td>
<td>2. Show all students.</td>
</tr>
<tr>
<td></td>
<td>3. Remove student.</td>
</tr>
<tr>
<td></td>
<td>4. Quit service.</td>
</tr>
<tr>
<td></td>
<td>Given your option: 1</td>
</tr>
<tr>
<td></td>
<td>Give student's first name, last name, username and average grade separated by space:</td>
</tr>
<tr>
<td></td>
<td>Tim Black tim 9.9</td>
</tr>
<tr>
<td></td>
<td>[Thu Jun 30 01:36:38 EET 2020] Received: {&quot;action&quot;: &quot;tuple inserted to db&quot;, &quot;status&quot;: &quot;success&quot;}</td>
</tr>
<tr>
<td></td>
<td>Choose one of the following options:</td>
</tr>
<tr>
<td></td>
<td>1. Insert student.</td>
</tr>
<tr>
<td></td>
<td>2. Show all students.</td>
</tr>
<tr>
<td></td>
<td>3. Remove student.</td>
</tr>
<tr>
<td></td>
<td>4. Quit service.</td>
</tr>
<tr>
<td></td>
<td>Given your option: 2</td>
</tr>
<tr>
<td></td>
<td>[Thu Jul 31 01:36:40 EET 2020] Received: {&quot;action&quot;: &quot;student data retrieved&quot;, &quot;students&quot;: [{&quot;average&quot;: 9.300000190734863, &quot;firstname&quot;: &quot;John&quot;, &quot;id&quot;: 1, &quot;lastname&quot;: &quot;Smith&quot;, &quot;username&quot;: &quot;jsmith&quot;}]</td>
</tr>
<tr>
<td></td>
<td>Choose one of the following options:</td>
</tr>
<tr>
<td></td>
<td>1. Insert student.</td>
</tr>
<tr>
<td></td>
<td>2. Show all students.</td>
</tr>
<tr>
<td></td>
<td>3. Remove student.</td>
</tr>
<tr>
<td></td>
<td>4. Quit service.</td>
</tr>
</tbody>
</table>
### Student Data Management

**Choose one of the following options:**
1. Insert student.
2. Show all students.
3. Remove student.
4. Quit service.

**Give your option:** 1

**Give student's first name, last name, username and average grade separated by space:**
Bob Flux bflux 8.2

**Choose one of the following options:**
1. Insert student.
2. Show all students.
3. Remove student.
4. Quit service.

**Give your option:** 2

**Give student's first name, last name, username and average grade separated by space:**
John Smith jsmith 9.3
Tim Black tim 9.9
Bob Flux bflux 8.1

**Choose one of the following options:**
1. Insert student.
2. Show all students.
3. Remove student.
4. Quit service.

**Give your option:** 3

**Give the id of the student to be deleted:**
3

**Choose one of the following options:**
1. Insert student.
2. Show all students.
3. Remove student.
4. Quit service.

**Choose one of the following options:**
1. Insert student.
2. Show all students.
3. Remove student.
4. Quit service.

**Give your option:** 4

---

**ID** | **FName** | **LName** | **UName** | **Average**
---|---|---|---|---
1 | John | Smith | jsmith | 9.3
3 | Tim | Black | tim | 9.9
4 | Bob | Flux | bflux | 8.1

<table>
<thead>
<tr>
<th>ID</th>
<th>FName</th>
<th>LName</th>
<th>UName</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John</td>
<td>Smith</td>
<td>jsmith</td>
<td>9.3</td>
</tr>
<tr>
<td>3</td>
<td>Tim</td>
<td>Black</td>
<td>tim</td>
<td>9.9</td>
</tr>
<tr>
<td>4</td>
<td>Bob</td>
<td>Flux</td>
<td>bflux</td>
<td>8.1</td>
</tr>
</tbody>
</table>

**Choose one of the following options:**
1. Insert student.
2. Show all students.
3. Remove student.
4. Quit service.

**Give your option:** 4

---

**ID** | **FName** | **LName** | **UName** | **Average**
---|---|---|---|---
1 | John | Smith | jsmith | 9.3
4 | Bob | Flux | bflux | 8.1

**Choose one of the following options:**
1. Insert student.
2. Show all students.
3. Remove student.
4. Quit service.

**Give your option:** 4

---

**ID** | **FName** | **LName** | **UName** | **Average**
---|---|---|---|---
1 | John | Smith | jsmith | 9.3
4 | Bob | Flux | bflux | 8.1

**Choose one of the following options:**
1. Insert student.
2. Show all students.
3. Remove student.
4. Quit service.

**Give your option:** 4

---

**ID** | **FName** | **LName** | **UName** | **Average**
---|---|---|---|---
1 | John | Smith | jsmith | 9.3
4 | Bob | Flux | bflux | 8.1

**Choose one of the following options:**
1. Insert student.
2. Show all students.
3. Remove student.
4. Quit service.

**Give your option:** 4

---

**ID** | **FName** | **LName** | **UName** | **Average**
---|---|---|---|---
1 | John | Smith | jsmith | 9.3
4 | Bob | Flux | bflux | 8.1

**Choose one of the following options:**
1. Insert student.
2. Show all students.
3. Remove student.
4. Quit service.

**Give your option:** 4

---

**ID** | **FName** | **LName** | **UName** | **Average**
---|---|---|---|---
1 | John | Smith | jsmith | 9.3
4 | Bob | Flux | bflux | 8.1

**Choose one of the following options:**
1. Insert student.
2. Show all students.
3. Remove student.
4. Quit service.

**Give your option:** 4

---

**ID** | **FName** | **LName** | **UName** | **Average**
---|---|---|---|---
1 | John | Smith | jsmith | 9.3
4 | Bob | Flux | bflux | 8.1

**Choose one of the following options:**
1. Insert student.
2. Show all students.
3. Remove student.
4. Quit service.

**Give your option:** 4