Multithreaded Client/Server - Student Database

Instructions: The assignment is based on the Area Of Circle and JDBC Labs.
Java threads covered In (Threads and Distributed Processing in Java).
The Area-Circle Multithreaded Client/server lab exercise covered in Lab4: Multithreaded Server.
Pseudo Code (Area Of Circle)
The assignment also uses work from the Java JDBC MySQL Labs #1/#2

1. Create a server hosted database (Assign2) including TWO tables (students and users). Use the following structures: (Populate each table with 4 records.)

students

- SID (type int(2), unique key)
- STUD_ID (type int(8))
- FNAME (type varchar(20))
- SNAME (type varchar(20))

users

- UID (type int(2), unique key)
- UNAME (type varchar(20))

2. Create a socket based Multithreaded Client/Server GUI application. Your application GUI should have two screens (1 User Login, 2 Database support) . The Layout is up to you, but please use standard JDK GUI commands & components.

1 User Login.

The server only accepts requests from registered Users in the users table. The client enters their UserID and submits request to the Server. The server creates a new thread for the client and validates that the Student exists in a database table. Invalid logins will result in an appropriate message being sent to the client and the socket is closed. Eg.

Client-1: Enter User ID: 12345
Server: Processing .....
Client-1: Welcome JOE BLOGGS... You are now connected to the Server.
OR
Client-1: Sorry 12345. You are not a registered student. Bye.

Clearly identify all communication between Server and Client by displaying HostName and IPAddress in the Client/Server windows with all messages sent

2 Database support

After a valid login the user can access the database review window. Create a basic database review GUI with the following features:

1. Forward/Back controls: Allow users to scroll forward and backwards through all records in the database
2. Search by SNAME: A simple search function based on Surname.
3. Clear control: Clear all fields.
4. Exit control: Exit / Logout

Submission Details:

Format: Eclipse Project (Assign2_yourName) containing all relevant java source code. Your project will be unzipped and imported into an Eclipse workspace. From there we will run your application and grade your documentation and/or models. For best results, it would be a good idea to have all external resources that you need for your project (like various external JARs) in a lib/ directory in your project. We will be willing to do minor build path changes to get your project to work with no penalty, but it is good form to have the project ready to deploy as soon as we import it into Eclipse

Marking Scheme

- Valid Submission: 20%
- JAVA Threaded Client/Server: 45%
- DBase/JDBC: 15%
- Quality: 20%
- Plagarism: Fail awarded 0%

1. Valid Submission

- On time
- Error free and executes correctly
- All requested files
- Correct Naming and Format conventions: (JAVA Project)
- Valid Database naming/tables/fields/contents (as specified)
- All elements of the assignment are inter-dependent and will be graded as a completed entity. ie
- An invalid database and a valid java section will not be accepted.

2. Java Multi Threaded GUI Client/Server

- GUI
- Event Handling
- Sockets
- Threads
- Client Request Processing

3. DBase/JDBC

- Valid and error free
- Valid Setup/Connection to WAMP/MAMP
- Valid Database, Tables, Fields and Content.
- Valid Submit functionality using JDBC
- Valid RETRIEVE functionality

4. Quality

- Overall quality of assignment. inc
- Design, Understanding, Execution, Coding, Testing/Validation, Documentation, Delivery and Submission

5. Plagiarism

- Identified Plagiarism will result in zero mark.