Software Development III - Object Oriented Architectures & Secure Development

Client-server communication: lab

General assignment

- Continue work on your *jBamaflex* application. Whereas uptil now it was one large application, we will split it up in a server and a client component for the retrieval of the various courses.
- The easiest is to create two copies of your existing application: one will become server, the other client.

Building the server

- Using the `ServerSocket` class, make sure the server listens on a free port (e.g. `1234` or `32768`) of your localhost.
- When there is an incoming connection, respond to the client with a collection of the various courses. Make sure to retrieve these from the database.
- In case the connection crashes, make sure the server stays up and running, so it can answer another client.
- There is no need for multi-threaded support.
- Make sure you remove all UI related packages/classes and dependencies from this project. This is purely a network server application.

Building the client

- Note:
    - This assignment assumes you have split up your data access code for courses and students in seperate repositories.

    - If this is not the case in your solution (e.g. you have one and the same repo for both students and courses), split up in separate repositories first.

- Create a new `NetworkCourseRepository` that implements your `CourseRepository` interface.
- When executing the method `getCourses` in this repository, use a `Socket` class to connect to the server on the same port you chose in the server part. It should give you all courses from the server.
- Make sure you switch to the `NetworkCourseRepository` when querying the various courses from within the service layer.

General tips

- Apply the various techniques studied during the classes.
- There isn't one "single solution". Make sure you can motivate your choices.
- It is your software, take ownership.