Enes Altuntas if21b115

SEB – Protocol

1. Design Description

The Java project is designed to serve as a reliable backend system that manages web services through an HTTP server. This system consists of three main parts: the HTTP server, database utilities, and JSON utilities, each fulfilling unique and interconnected roles. The HTTP server, created in HttpServerSetup.java, handles communications with clients. It uses com.sun.net.httpserver.HttpServer to listen on port 10001 and routes incoming HTTP requests to specific handlers based on the URL path, which makes it efficient and easy to expand.

The database utility in DbUtil.java plays a critical role in managing data interactions. It uses a single connection to talk to the database, which makes the system more efficient by reusing this connection for different tasks. It loads database settings from a properties file, setting up connections with the necessary details like URL and user credentials.

Lastly, the JSON utilities in JsonUtil.java are responsible for converting data between JSON format and Java objects. This is essential for sending data to and receiving data from the front end, ensuring consistency and ease of handling data. Together, these components make the system well-organized, easy to manage, and ready for future updates. This straightforward design ensures that different parts of the system work together smoothly, providing a strong foundation for the project's services.

2. Lessons Learned

- Concurrency with threads: The importance of utilizing “newSingleThreadScheduledExecutor to handle more requests simultaneously was learned. This allows for increased efficiency and responsiveness in the server.

- Secure database connection: The importance of securing the connection to the database was learned. This includes implementing secure authentication methods and properly handling sensitive information. Furthermore, experience with PostgreSQL was acquired.

- User authorization through token: The use of tokens for user authorization was implemented. This allows for secure and efficient management of user access to the system.

- General good practices: The use of design patterns, such as the singleton pattern, was implemented to improve the overall design and maintainability of the code.

3. Unit Testing

The unit tests are conducted in the “test”-folder and are working without any issues. A total of 13 tests are being performed. This includes testing whether the pushup record is being updated so the user records are kept fresh. Additionally, there is a whole file to test tournaments. These tests go through the tournament creation process, finding an active tournament and updating a tournament, which is the key factor of this project and should work without any flaws. To make sure that user profile handling is working properly, and the game mechanics are in place, multiple tests for the user was created. These tests check the login, register function which looks into both incorrect and correct credentials to make sure that users provide the correct username and password or not. Furthermore, it checks a valid or invalid token assigned to the token.

4. Estimated Tracked Time

Approximately 45-50 hours were spent working on the whole project. This includes:

- Configuring the database and establishing a connection to the program.
- Configuring the server
- Developing each class inside the manager folder.
- Developing each class in the project folder.
- Writing and running tests for the project.

5. GitHub Link

https://github.com/Enes3854/SportsExerciseBattle.git