Server Starten:

Unter src -> main -> java -> ShrimpServerApplication ist die Startdatei für den Springbootserver zu finden. Der Server kann über Microsoft Visual Studio Code gestartet werden indem das Run, welches sich oberhalb der public static void main befindet, angeklickt wird.

Anschließend kann der Server im Browser über : localhost:8080 aufgerufen werden

Auf der Login Seite bitte folgende Benutzer für die Tests verwenden

Username: [nabend@moin.de](mailto:nabend@moin.de)

PW: hallo123

Username: [moin@nabend.de](mailto:moin@nabend.de)

PW: hallo123

Oben Links stehen nun drei Namen. Dies sind die unterschiedlichen Clients.

Der Client unter dem Namen „Lennart“ ist meiner. Bitte diesen verwenden.

Im Hauptmenü kann sich theoretisch über die Button ausloggen und einloggen neu angemeldet werden. Registrieren ist auch möglich. Es wird jeweils auf die REST Endpunkte vom Server dafür zugegriffen.

REST Endpunkte und Websockets: Unter src -> main -> java -> controller sind die Controller Klassen für den Client zu finden.

ShrimpServerAuthenticationController: ist für die Realisierung der Registrierung da. Es wird auf den Endpunkt /process\_register zugegriffen um sich mit einem neuen Spieler in der Datenbank zu registrieren.

ShrimpServerLobbyController: Dieser Controller ist für die Erstellung der Lobby da. Über den Endpunkt /myPlayer werden dem eingeloggten Benutzer seine eigenen Accountdaten ausgegeben.

Der Endpunkt “/lobby/get\_private\_messages/“ wird so wie ich das verstanden habe für das empfangen von privaten Nachrichten verwendet. Dies habe ich bereits begonnen bzw. versucht umzusetzen. Zu finden in meiner OnlineLobby.js.

(“/lobby/send\_message“) benötigen wir wohl nicht laut den Aussagen der Kollegen. Da verwenden wir für die Kommunikation schon Websockets.

(“/lobby/top20LobbyChat“) mit diesem Endpunkt lasse ich mir die letzten zwanzig Nachrichten des Chats ausgeben. Ist schon realisiert. Dieser reicht wohl auch, ich werde glaube ich erstmal nicht unbedingt den Endpunkt (“/lobby/getAllLobbyChat“) benötigen

Mit dem Endpunkt lobby/onlineUsers ist dafür da um mich als User Online zu schalten. Das der server mir eine Online Meldung mitgibt. Ist schon realisiert.

Lobby/getPlayerHistory/player id ist für das abrufen einer Playerhistory. Diese ist wie ich es verstanden habe bisher noch nicht im spiel realisiert daher kann das wohl erstmal weggellassen werden. BZW Was ich jetzt gerade sehe: ganz unten ist eine schnittstelle /lobby/leaderboard, ich schätze fast diese soll playerHistory ersetzen.

Die websockets socket.addUser und socket.sendMessage habe ich in der Lobby für das versenden und empfangen von nachrichten bereits realisiert und implementiert.

ShrimpsServerGameController.java:

Diese Controllerklasse ist für das anzeigen des spiels und mit Fortführenden Endpunkten für das Einladen eines Spielers. Hier liegen meine hauptbrennpunkte und da würde ich gerne soweit kommen wie es geht.

Meine Client Dateien aufrufen:

Meine Client dateien sind zu finden unter : src -> main -> resources -> static -> lennart

Das Hauptaugenmerk liegt auf der onlinelobby.html , RestAbfragen.js, WebSocket.js und jetzt neu der spielEinladungUndVerwaltung.js

Ich hatte bis vorhin den Chat und die Abfrage der Online Benutzer komplett in Takt. Da ich nun angefangen habe zu versuchen die Lobby mit Einladungen zu verbinden habe ich mir das leicht zerschossen.

Um das ganze einmal funktionstüchtig zu sehen kann die RestAbfragen.js durch die oldRestAbfragen.js ausgetauscht werden. Die spielEinladungUndVerwaltung.js wäre dann nichtig.

Hauptpunkte:

Ein eingeloggter Spieler muss auf definierte Weise (z. B. über eine Lobby) einen Mitspieler finden können, bevor das eigentliche Spiel beginnen kann.

• Für jeden Spieler wird eine Historie der gespielten Spiele gespeichert und mittels einfacher Auswertungen dargestellt (gewonnene und verlorene Spiele, Highscores o. ä.).

• Es sollen an mindestens einer sinnvollen Stelle auch Bilder über die API an den Server gesendet bzw. vom Client angefragt und dargestellt werden (z. B. die Profilbilder der Spieler).

* Alles Schick machen am Ende
* Und falls irgend ein Puffer dann an die Spieleentwicklung :D Dafür sind dann auch Endpunkte in dem GameController gegeben

Ist nur leider auch wichtig das ich das alles einigermaßen nachvollziehen kann, da ich es am Mittwoch erläutern bzw vorstellen muss.

Weiterhin:  
Eine Große Hilfe kann es sein, da möchte ich aber möglichst so gut es geht abwandeln, damit ich nachher keinen ärger bekomme, ein Client von einem Kollegen.

Dieser hat nämlich schon mit javascript einen Client gebaut für seinen Server. Ich muss allerdings zugeben das dieser aus meinser sicht fortgeschrittene Konstrukte benutzt die ich nicht sofort verstanden habe.

Zu finden unter:

Src -> main -> resources -> static -> js in den Ordnern game und lobby.