

Übungsblatt 3: Schach als Web-Anwendung

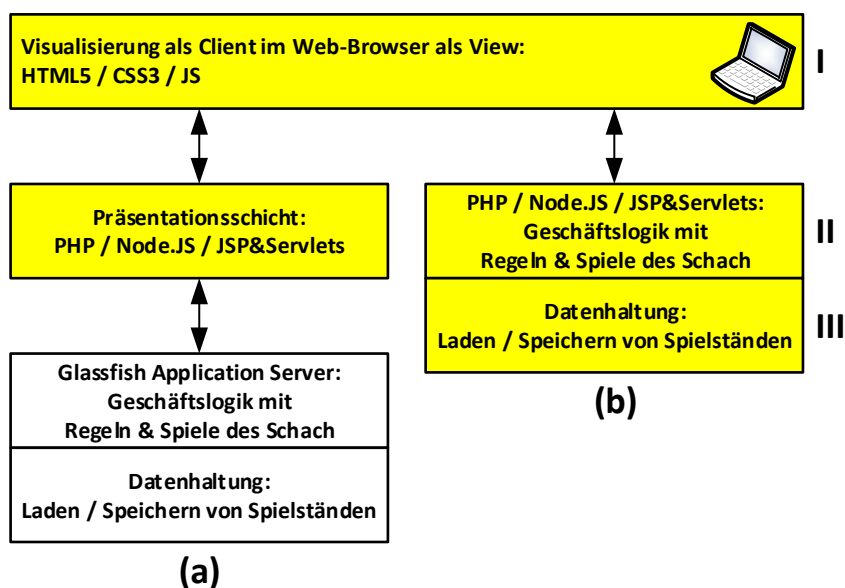
300 Punkte möglich

Prof. Dopatka erhält die gesamten Codes als ZIP-Datei im Format A-GRUPPENNUMMER.ZIP bzw. A-GRUPPENNUMMER.PDF bis zum Beginn der Deadline. Kommt die Datei nicht an oder ist sie nicht lesbar, so gilt sie generell als nicht abgegeben. Arbeiten Sie also niemals last-minute; es gibt keine Ausreden!

Bereiten Sie sich in Ihrer Gruppe darauf vor, die Ergebnisse spätestens zu Beginn der Deadline zu präsentieren. Dabei muss jeder Teilnehmer der Gruppe jede Frage zum Code zügig beantworten können. Die automatische Generierung von Quellcode ist verboten!

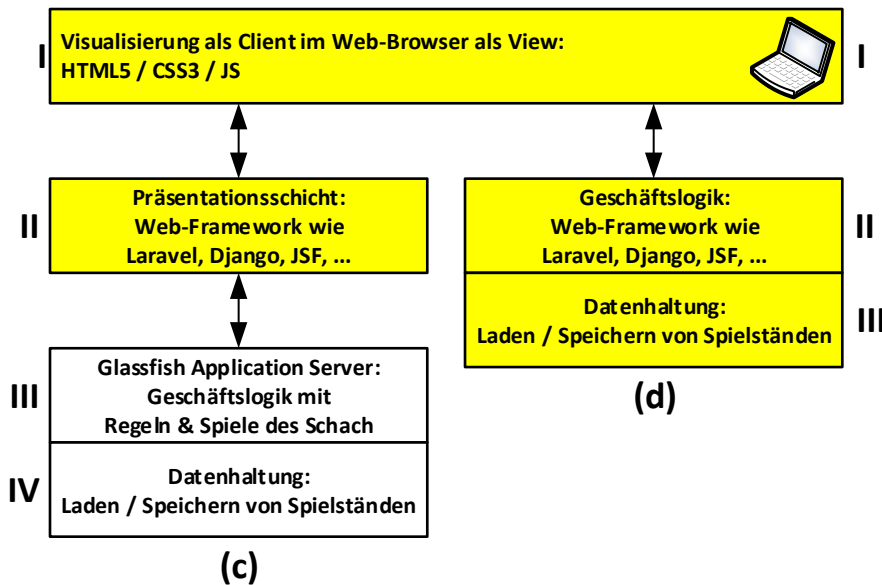
Das Ziel dieses Übungsblatts besteht darin, Schach als Web-Anwendung im Frontend anzubieten. Der User sitzt vor einem Internet-Browser als Client und kann die Funktionalität des bestehenden REST-Interfaces des Schach-Spiels voll nutzen, so dass man an 2 verschiedenen PCs eine gemeinsame Partie Schach spielen kann. In diesem Internet-Browser ist HTML5, CSS3 und JavaScript zu verwenden.

Bei der Gestaltung der Serverseite haben Sie die im Folgenden dargestellten Freiräume. Sie können beispielsweise den vorhandenen Glassfish Application Server weiterhin verwenden, der bereits die Geschäftslogik mit den Regeln und die Schach-Spiele des Schach enthält und via REST Level 1 kommuniziert. Dies sind die Optionen (a) und (c).



Um das Frontend realisieren zu können, erstellen Sie eine Präsentationsschicht, die entweder direkt in einer server-seitigen Programmiersprache wie PHP, Node.JS oder JSPs & Servlets realisiert ist (a), oder über ein Web-Framework (c).

Statt dessen können Sie auch den bestehenden Code aus dem Glassfish Application Server mit voller Funktionalität incl. Datenhaltung in eine server-seitige Programmiersprache wie PHP, Node.JS oder JSPs & Servlets portieren. Eine alternative Portierung kann in ein Web-Framework Ihrer Wahl erfolgen. Dies sind die Optionen (b) und (d). Die REST-Kommunikation entfällt in diesen Fällen.



Aufgabe 1: Schriftliche Skizzierung der Web-Anwendung und vertikaler Prototyp [60 Punkte]

Skizzieren Sie Ihre geplante Umsetzung schriftlich in Form einer PDF-Datei. Welchen Ansatz von (a) bis (d) wollen Sie verfolgen und welche konkreten Technologien bzw. Frameworks wollen Sie einsetzen? Welches Entwurfsmuster verwenden Sie?

Stellen Sie für sich als Gruppe sicher, dass Sie diesen Plan auch erfolgreich umsetzen können, denn von dieser geplanten Umsetzung dürfen Sie nicht mehr abweichen!

Liefern Sie dazu einen vertikalen Prototyp ab, einen sogenannten Durchstich: Dabei wird eine einzelne Funktionalität wie das Erstellen eines neuen Spiels über alle Schichten implementiert.

Aufgabe 2: Prüfung der Funktionalität der Web-Anwendung [120 Punkte]

Stellen Sie die gesamte Funktionalität der Web-Anwendung bei der Abgabe vor, Prof. Dopatka wird die Anwendung durch eigene Eingaben testen. Weisen Sie nach, dass Sie die geplante Umsetzung vollständig implementiert haben.

Aufgabe 3: Präsentation der Funktionalität der Web-Anwendung [120 Punkte]

Präsentieren Sie Ihre Web-Anwendung dem Kunden in Form eines ca. 20-minütigen Vortrags Ihrer Gruppe an das Auditorium per Beamer. Stellen Sie sowohl Ihre Idee zur Umsetzung aus Aufgabe 1 vor, also auch die Funktionalität aus Aufgabe 2. Diese Präsentation wird entweder am MI 09.01 oder am DO 10.01 gehalten; dies entscheidet Prof. Dopatka.