

W4DSKI 104 - Fortgeschrittene Informatik

Dozent: Prof. Dr. Michael Eichberg
Kontakt: michael.eichberg@dhbw.de, Raum 149B
Version: MA-WDSKI24A

Verteilte Systeme - Kerninhalte gem. MHB

- Terminologie, Konzepte, Architekturen, Anforderungsprofile und Architekturmodelle für verteilte Systeme
- Synchrone und asynchrone Kommunikation und entfernter Methodenaufruf (RMI, RPC, Web-Services, ...), Übertragungsprotokolle und APIs zwischen Client und Server (z.B. HTTP, HTTPS, WebSockets, Fetch API, etc.), Einführung in das RESTful API-Design
- Herausforderungen in verteilten Systemen
- High Performance Computing und Distributed Computing
- Sicherheitsaspekte bei der Verarbeitung von verteilten Anwendungen; Authentifizierung, Autorisierung, Rollenkonzepte
- Grundlagen Ubiquitous Computing, Internet der Dinge, MQTT, Edge Computing, Streaming & Messaging

Web-Entwicklung - Kerninhalte gem. MHB

- Frontend-Technologien, HTML, CSS, JavaScript
- Konzepte, Entwurfsmuster und Werkzeuge für die Entwicklung von Web-Anwendungen - Entwurf und Umsetzung von Responsive Web-Design und zustandsbehafteten Web-Anwendungen
- Abgrenzung client-side und Server-Side-Rendering

Labor Web Projekt

Die theoretischen Inhalte sollen jeweils auch mit aktuellen Technologien beispielhaft umgesetzt werden. Es soll eine übergreifende Anwendung entwickelt werden anhand derer das Zusammenspiel deutlich wird.

Prüfungsleistung - Portfolio

Hintergrund

- Das Modul hat 5 ECTS
- Insgesamt gibt es max. 120 Punkte

Bewertung

- Kurztest 25%
- Portfolioaufgabe 75% in Teams von 5 bis 6 Studierenden. Die Aufgabe umfasst Präsentationen, Code Reviews und die Entwicklung eines Projekts.

Gegenstand der Portfolioaufgabe ist die Entwicklung eines verteilten Web-Spiels für mehrere Spieler. Die Wahl des Spieles ist weitgehend frei.

Folgende Anforderungen sind zu erfüllen:

- Ihr Projekt muss am Ende spielbar sein in dem Sinne, dass alle Kursteilnehmer die Möglichkeit haben gleichzeitig zu spielen; ggf. in mehreren Kleingruppen gegeneinander.
- Das Spiel muss rundenbasiert sein.

Achtung!

Echtzeitspiele sind aufgrund der damit verbundenen Komplexität ausgeschlossen.

- Sie können sich von klassischen Brettspielen oder Kartenspielen inspirieren lassen.
- Die Entwicklung erfolgt in Gruppen von 5 bis 6 Personen und muss immer einen Webclient und eine Serverkomponente umfassen.
- Die eingesetzten Technologien sind auf JavaScript, CSS und HTML sowie ggf. Python für die Serverseite beschränkt. Frameworks, um ggf. die Kommunikation zwischen Web-Client und Server zu vereinfachen, dürfen eingesetzt werden.
- Die Webanwendung muss responsive sein und braucht nur auf den neuesten Browsern laufen: Safari, Chrome und Firefox. Es sollten also die neuesten Web-Technologien verwendet werden.

Termine - 1. Block

(Klassische Vorlesungen und Übungen)

- 7. May 2025 von 09:30 bis 12:45 und 13:15 bis 16:30, CEST
- 12. May 2025 von 09:30 bis 12:45, CEST
- 19. May 2025 von 09:30 bis 12:45, CEST

Termine - 2. Block

(Vorlesungen und Übungen zusätzlich zur Begleitung des Projekts sowie ein Test.)

18. Jun 2025 von 09:30 bis 12:45 und 13:30 bis 16:45, CEST

Ablauf

1. Vorstellung der Projekte

(8-10 Minuten pro Gruppe - 1 Personen)

An dem Tag muss das Spielkonzept weitgehend fertig sein und auch das Design der Oberfläche muss weitestgehend fertig sein.

Die eigentliche Spiellogik, insbesondere die Interaktion mit dem Server - muss weder angefangen noch fertig sein!

2. Durchführung eines Code Reviews (HTML und CSS) eines anderen Projektes (2 * 45 Minuten)

Die Code Reviews erfolgen in zwei Runden, damit jede Gruppe ein Review bekommt!

Welche Gruppe welche andere Gruppe reviewt wird *nach* den Vorträgen bekannt gegeben.

3. Erstellung eines Reports, der konstruktive Vorschläge enthält (30 Minuten).

4. Präsentation der Reports (max 2 Personen pro Gruppe!)

5. Danach wird es weitere Vorlesungsinhalte und Übungen geben.

25. Jun 2025 von 09:30 bis 12:45, CEST

27. Jun 2025 von 09:30 bis 12:45 und 13:30 bis 16:45, CEST

Vorlesungen und Übungen; ggf. Unterstützung bei der Durchführung des Portfolioprojekts.

9. Jul 2025 von 09:30 bis 12:45 und 13:30 bis 16:45, CEST

Ablauf

■ Vormittag

Vorlesung und Übung

■ Nachmittag

Unterstützung der Gruppen bei der Entwicklung:

Gruppe1: 13:30 bis 14:00

Gruppe2: 14:00 bis 14:30

Gruppe3: 14:30 bis 15:00

Gruppe4: 15:00 bis 15:30

Gruppe5: 15:30 bis 16:00

Gruppe6: 16:00 bis 16:30

14. Jul 2025 von 09:30 bis 12:45 und 13:30 bis 16:45, CEST

Ablauf

0. Kurztest (20 Minuten mit max. 30 Punkten)

1. Vorstellung der Projekte mit Fokus auf der Implementierung (Architektur, Build Skripte, JavaScript und ggf. Python)

(8-10 Minuten pro Gruppe - 1 Person)

2. Durchführung gegenseitiger Code Reviews (Fokus: JavaScript und ggf. Python Code) eines

anderen Projektes (2 * 60 Minuten)

Welche Gruppe welche andere Gruppe reviewt wird *nach* den Vorträgen bekannt gegeben.

3. Erstellung eines Reports, der konstruktive Vorschläge enthält (30 Minuten).
4. Präsentation der Reports (max 2 Personen pro Gruppen).
5. Umsetzung von Verbesserungsvorschlägen und bei Bedarf Beratung der Gruppen.

16. Jul 2025 von 09:30 bis 12:45 und 13:30 bis 16:45, CEST

Ablauf

1. Jede Gruppe stellt kurz die finale Version des Spiels inkl. der Regeln und der Bedienung vor (ca. 5-10 Minuten). Danach spielen wir das Spiel ca. 20 Minuten.
2. Besprechung der Anforderungen an die finale Abgabe.

Prüfungsleistungsbestandteile

Bestandteil	Benotung	Punkte
ein bewerteter Kurztest	Individualnote	30
initiale Projektpräsentation	Gruppennote	05
Qualität des Reviews (d.h. des Reports)	Gruppennote	15
zweite Projektpräsentation	Gruppennote	05
Qualität des Reviews (d.h. des Reports)	Gruppennote	20
Spielbarkeit (inkl. Abschlusspräsentation)	Gruppennote	10
Projekt	Gruppennote	35
		7

Abgabe (Ergänzt am 14. Juli 2025)

- Abgabe des Projekts am 3.8.2025 bis 23:59 Uhr per Moodle.

(Die Abgabemöglichkeit ist bereits eingerichtet und bis zur Deadline jederzeit möglich.)

- Abzugeben sind:

- **die vollständigen Quellen des Projekts**; sollten externe Libraries verwendet worden sein, so ist eine entsprechende `package.json` bzw. `requirements.txt` Datei beizulegen, die alle Abhängigkeiten des Projekts enthält.
- eine Beschreibung der Architektur in Hinblick auf den Nachrichtenaustausch.
D. h. wann werden welche Nachrichten vom Client zum Server und wieder zurück zum Client gesendet. Wie ist das Datenformat der Nachrichten. Neben einer Überblicksdarstellung ist eine exemplarische Darstellung der Nachrichten anzugeben.
- das Projekt muss eine **License.txt** enthalten, die die Lizenzbedingungen des Projekts beschreibt.
(Für eine Opensourcelizenz: <https://opensource.org/licenses>)
- das Projekt muss eine **README.md** enthalten, die beschreibt wie der Server zu konfigurieren und zu starten ist.
- ein kurzes Dokument, das besagt **wer an welchem Teil gearbeitet hat**
- eine kurze **Erklärung welche KI Tools wo und wie verwendet wurden.**

Das bezieht alle Aspekte ein: Generierung von Code, Icons, Musik und Dokumentation.

Benotungsgrundlage für das Projekt (Ergänzt am 14. Juli 2025)

- (max. 25 Punkte) die Qualität des Sourcecodes (HTML, CSS, JavaScript und ggf. Python)
Hierbei werden insbesondere die Kriterien beachtet, die auch bei den Reviews Gegenstand waren. Darüber hinaus wird auf die Verständlichkeit der Netzwerkkommunikation geachtet.
- (max. 10 Punkte) die Qualität der Dokumentation (Readme und Architektur)
- die anderen Dokumente (License.txt, Erklärung bzgl. KI Tools, Beteiligung an der Entwicklung) führen ggf. zum Aufschub der Notenmeldung