

Modul 223

Multiuser - Apps



Block	Inhalt
1	Intro Multiuser, Software-Architektur, Datenpersistenz
2	Projekteinrichtung Github, Multiuser-Stories, Multiuser-Datenmodell
3	Spring-Security-Einführung, JPA-Datenmodell + Transaktionen
4-5	JWT-Backend
6-7	JWT-Frontend
8-9	Arbeiten am eigenen Projekt
10	Präsentation und Fachgespräch

Unterrichtsziel heute

- einrichten Projektumgebung
- erstellen User Stories für Multiuser-Software
- konzipieren RBAC

Bewertungsraster	Max-Punkte
Dokumentation sinnvoll strukturiert und durchgängig einheitlich formatiert	2
notwendige User Stories mit Akzeptanzkriterien aussagekräftig und korrekt dokumentiert	2
Arbeitsplan enthält sinnvolle Arbeitspakete und Aufwandschätzung	2
Backend funktional und korrekt implementiert und im Code dokumentiert	3
mind. 2 sinnvolle Backend-Tests automatisiert und protokolliert	2
Backend-Architektur aussagekräftig, korrekt und mit eigenen Worten beschrieben (ggf. illustriert)	2
Transaktionen begründet eingesetzt	2
Frontend funktional und korrekt implementiert und im Code dokumentiert	3
mind. sinnvolle 2 Frontend-Tests automatisiert und protokolliert	2
Frontend-Architektur aussagekräftig, korrekt und mit eigenen Worten beschrieben (ggf. illustriert)	2
Git korrekt eingesetzt (mit Branches, mehreren ordentlich kommentierte Commits)	2
JWT-Authentifizierung im Front- und Backend korrekt implementiert und im Code dokumentiert	2
Sicherheitskonzept dokumentiert	2
Arbeitsjournal je Block (referenziert Arbeitspakete Arbeitsplan, Dauer und allfällige Probleme)	2
Auswertung enthält objektiven Soll-Ist Vergleich und Problemanalyse	2
Projekt ist nach Anleitung deploybar und ausführbar	2
Live Produktpräsentation (max. 10 Minuten)	6
Fachgespräch (individuell)	10
Summe	50

Projektideen?

Github Projekt einrichten

- Online-Repo erstellen
- Projektidee in README beschreiben (ggf. online)
- Berechtigung für Repo: privat, Dozent braucht Zugriff
- lokal clonen (PULL/PUSH Funktionalität prüfen)
- ...

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

Repository template

No template ▾

Start your repository with a template repository's contents.

Owner * greenorca ▾ / **Repository name *** M223-WissQuiz

✓ M223-WissQuiz is available.

Great repository names are short and memorable. Need inspiration? How about [studious-garbanzo](#) ?

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

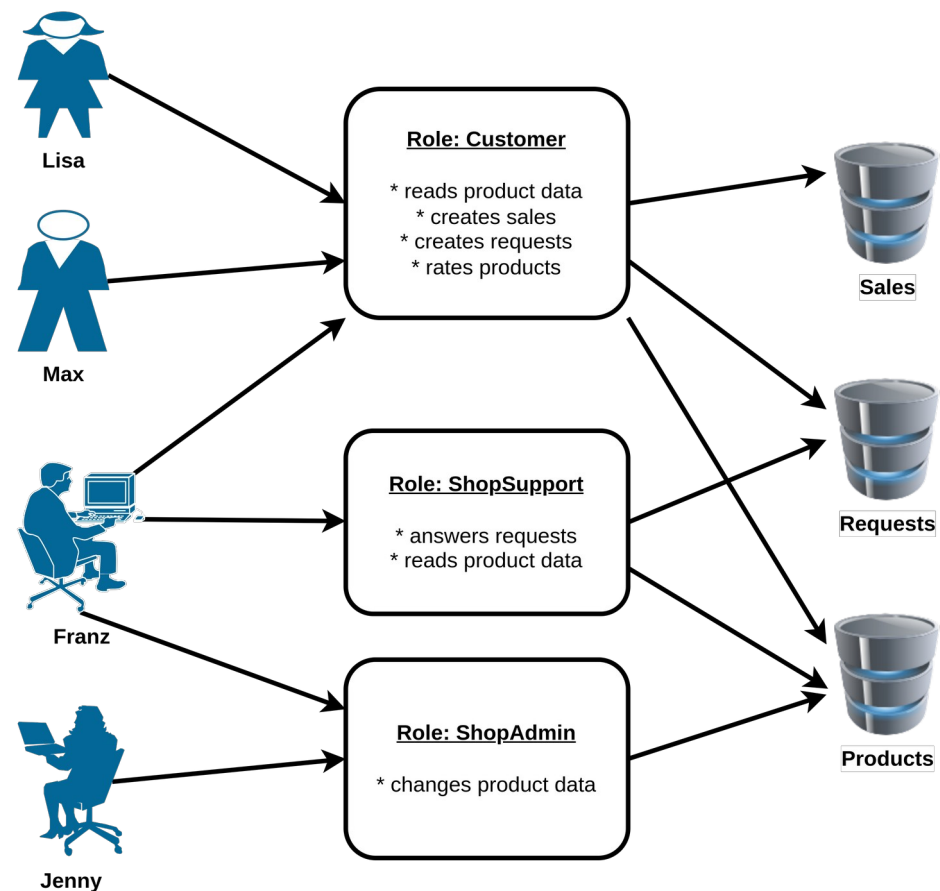
☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Zugriffssteuerung in Mehrbenutzersystemen

- Idee: bestimmte Benutzer dürfen nur bestimmte Ressourcen:
 - lesen
 - ändern
- Beispiele:
 - Zutrittskontrollsysteme
 - Active Directory
 - Shopping-Systeme (Kunden kaufen und lesen Produktdaten, nur Shop-Manager ändern Produktdaten)

RBAC - Role Based Access Control

- Zugriffsteuerung über Rollen ist flexibler und einfacher wartbar als Steuerung auf Benutzerebene
- siehe https://de.wikipedia.org/wiki/Role_Based_Access_Control



User Stories

- Wer? Was? Warum?
- Akzeptanzkriterien als Grundlage für spätere Akzeptanztests
- Hands on: Erstellen User Stories für unterschiedliche Benutzer
- Zeit: 45 Minuten

Rahmenbedingungen

- Abgabetermin Doku und Code (als Github-Repo):
07.06.2025, 19:00 Uhr
- Präsentation + Produktdemo: Dauer max. 10 Minuten
- Fachgespräch im Anschluss: Dauer max. 10 Minuten
- **Wichtig:**
 - Alle Klassen, Funktionen und Komponenten sind aussagekräftig auf Deutsch dokumentiert.
 - Allfällige Hilfestellungen (Internet, ChatGPT etc.) sind dokumentiert. Nicht dokumentierte Hilfestellung gilt als Betrugsversuch