

OTH Regensburg – Softwareentwicklung Kurs – SW WS 2026/27 Prof. Dr. Alixandre Santana Präambel

Als Bewertungspaket dienen:

- Architekturdokumente (Canvas Model, Product Backlog mit User Stories, Klassendiagramm nur mit den geschäftsbezogenen Klassen)
- Der komplette Quellcode des Projekts (Klassen sollten kommentiert werden)
- Das Deployment sowie die Ausführung und Nutzbarkeit der Anwendung
- Link für die Docker-Datei
- **Interview***

* Sie sollten alle Fragen des Professors korrekt beantworten. Wenn Sie Ihr Wissen über das Projekt nicht unter Beweis stellen können, könnten Sie den Kurs nicht bestehen.

Über die Anwendung:

- Anwendung meint das Gesamtergebnis Ihres Projekts i. S. eines Software-Systems
- Bereiche der Kunden (Vorschlag): Arbeitskräftemangel, Health, E-Gov, Nachhaltigkeit, Company-specific.

Als Basistechnologie **MUSS Springboot** genutzt werden. **Programmiersprache ist nur Java.**

Ein Projektteam muss aus vier Personen bestehen.

Die Evaluation ist individuell (90%): jedes Mitglied muss individuelle Anforderungen/Aktivitäten innerhalb des Projekts haben. Die 4 Teile müssen ein Gesamtbild ergeben.

Die Evaluation ist auch Gruppenbewertung (10%): Dabei wird auch die Gesamtfunktion und der Umfang der laufenden Anwendung berücksichtigt. Stellen Sie sicher, dass die Teile des Mitglieds ein Ganzes bilden.

Über das Product Backlog

Für jedes Mitglied des Teams wird erwartet:

1) Um mindestens 4 Features und ALLE ihre notwendigen CRUDs (CREATE, UPDATE, DELETE, SEARCH) für mindestens zwei verschiedene BENUTZERPROFILE (Bsp.: USER, ADMIN, MANAGER usw.) zu implementieren.

Beispiel:

Ein „Sekretär“-PROFIL:

Feature 1: „das Management von Workshop“-Angebot in dem System (CRUD). Das Feature benötigt eine „Workshop“-CRUD und „Fakultät“-CRUD.

Feature 2: „Studentenregistrierung in Workshop“. Das Feature benötigt eine „Workshop“-CRUD, eine „Student“-CRUD und eine „Registrierung“ CRUD, inklusive Registrierungsbestätigung per E-Mail.

Ein „Student“-PROFIL:

Feature 3: Management von dem Konto (CRUD), inklusive Aktualisierung des E-Mail-Feldes (mit Validierung per E-Mail).

Feature 4: Teilnahmebescheinigung für den Workshop. Für diese Funktion muss die Sekretärin die Anwesenheit des Studenten bei der Workshop-Anmeldung bestätigen. Es wird ein PDF mit den Informationen generiert.

Ein „ADM“-PROFIL:

Feature 5: Management von Users und ihren Profiles (CRUD).

- Die CRUDs sollten unterschiedliche Komplexitätsgrade aufweisen.

Betrachten Sie das Feature F2: einen Studenten in einem Workshop anzumelden: Das Sekretariat muss zuerst einen Studenten auswählen, dann das Workshop, dann prüfen, ob in diesem Workshop noch Plätze frei sind, und schließlich die Anmeldung vornehmen. Bestätigung wird per E-Mail geschickt.

Betrachten Sie das Feature F3: zum Erstellen eines Students. Das könnte viel einfacher als F2 sein: einfach ein Studentenformular auszufüllen und die E-Mail zu bestätigen).

Diese beiden Beispiele, F2, F3, geben uns eine Vorstellung davon, dass wir unterschiedliche Aufwandsstufen benötigen, um Funktionen mit unterschiedlicher Komplexität zu implementieren.

ACHTUNG: Bitte implementieren Sie nicht nur grundlegende und isolierte CRUDs von Geschäftseinheiten! Diese Systeme würden keinen Sinn ergeben. Denken Sie bei allen CRUD-Implementierungen an Felder, die aktualisiert werden können (wie die Anzahl der freien Plätze in einem Workshop, das E-Mail-Feld eines Benutzers, der Ort des Workshops usw.); und an die Beziehungen zu anderen Tabellen in der Datenbank (z. B. Workshop-Fakultät, Registrierung-Student, Student-Bachelor usw.) Einfache und isolierte CRUDs mit nur drei oder fünf Feldern (Attributen) werden nicht akzeptiert.

- Für das „Product Backlog“, verwenden Sie die in Elo (Über das Projekt) hochgeladene "Product Backlog Template".

Folien über Product Backlog:

<https://www.dropbox.com/scl/fi/fkcxchildme0b26umlnllc/04-SPM-Class-03-Part-1-Product-Backlog.pdf?rlkey=pfxxs4sqjt79992io0o3hhiyc&dl=0>

Product Backlog Template: https://docs.google.com/spreadsheets/d/17U2-X1Lt7_Xqg1YzuPnG2TNmaaGL-4G4O4s5C7dciOc/edit?usp=sharing

Individual Product Backlog Example:

[https://www.dropbox.com/scl/fi/ovbdsrokdmdkvdiom4prl6/Individual Product Backlog Example.xlsx?rlkey=gvg3dda4rsczr05gxxx5zc446&dl=0](https://www.dropbox.com/scl/fi/ovbdsrokdmdkvdiom4prl6/Individual%20Product%20Backlog%20Example.xlsx?rlkey=gvg3dda4rsczr05gxxx5zc446&dl=0)

2) Für jedes Mitglied: Bereitstellung mindestens einer REST-API (mit allen CRUD-Operationen) einer Geschäftseinheit UND Nutzung mindestens einer externen REST-API (Endpunkt mit oder ohne Authentifizierung)

die angebotene API sollte eine one-to-Many-Beziehung enthalten.

Beispiel: Student und seine Workshop-Registrations.

Achtung: Bitte implementieren Sie nicht nur grundlegende und isolierte CRUDs von Geschäftseinheiten!

3) „Paginierung“ zu implementieren, um die Suchergebnisse anzuzeigen.

mindestens in einer Feature, mit Parametern (z. B. Name, Title, Ort usw.) und Befehlen wie „ORDER BY“ (z. B. Datum, Name usw.)

4) Kommunikation.

z.B.: eine Bestätigungs-E-Mail zu schicken, eine SMS senden usw.

Achtung: Verwenden Sie dazu die Servicekomponente der MVC-Architektur. Rufen Sie E-Mail-APIs (JARs) nicht direkt auf!

5) Thymeleaf oder eine gleichwertige Technologie zu benutzen

Sie können auch andere Frontend-Frameworks wie Angular auswählen.

6) Anmeldung mit Spring Boot Security zu implementieren

Ein(e) Student(in) soll die Anmeldefunktion für das gesamte System implementieren. Die anderen Studenten sollen es lernen und während der Projektpräsentation erklären, wie die Anmeldung funktioniert.

7) 1 Non-functional requirement (NFR) wie zum Beispiel die Internationalisierung der Nachrichten und Bezeichnungen der Ansichten ODER jede andere nicht geschäftsbezogene Anforderung.

Mehr Info über NFRs: https://link.springer.com/content/pdf/10.1007/978-3-319-18344-2_3.pdf

8) Ein spezifisches Feature mit generative KI zu integrieren.

Beispiele: Zusammenfassung des Kenntnissen eines Students.

ACHTUNG:

Die entwickelten „User Stories“/Features aller Mitglieder müssen zusammen einen Sinn ergeben. Es gibt keine Bestellartikel, wenn Sie die verkauften Artikel nicht registrieren. Es gibt keine Kundenhistorie, wenn Sie die Bestellungen für diesen Kunden nicht speichern.

WICHTIG:

Das sind die **zufriedenstellenden Anforderungen (~ 2,0 – 2,3) !**

Sie **garantieren nicht** die beste Noten (1.0).

Je mehr Anforderungen Sie umsetzen/je komplexer die Features sind/je mehr Aufwand Sie zeigen, desto besser ist Ihre Note!

Über Deadlines

1) Ideenvorstellung bis 07.10.2026:

Teilen Sie hier mit:

[https://syncandshare.lrz.de/getlink/fiA61jLydindbWPfMVXXzy/LeanCanvas Ideen Datasheet SW2026.xlsx](https://syncandshare.lrz.de/getlink/fiA61jLydindbWPfMVXXzy/LeanCanvas%20Ideen%20Datasheet%20SW2026.xlsx)

2) Team Formation bis 12.10.2026:

3) Product Backlog Vorstellung (Datasheet format): 15.10, 16.10, 22.10, 23.10.

Wählen Sie ein Zeitfenster:

[https://syncandshare.lrz.de/getlink/fiE8A79doaxBdzLySukfP4/Timeslots-Product%20Backlog%20TEAM%20Presentation SW-WS2026.xlsx](https://syncandshare.lrz.de/getlink/fiE8A79doaxBdzLySukfP4/Timeslots-Product%20Backlog%20TEAM%20Presentation%20SW-WS2026.xlsx)

4) Bis 24.01.2027 um 23:59 laden Sie eine ZIP-Datei hoch, die enthält:

2.1) Eine PDF-Datei mit:

- Name: „Titel des Projekts,_vollständige Namen der Mitglieder.PDF“.
- Lesbares Lean Canvas. Teilen Sie das Canvas bei Bedarf in Abschnitte auf.
- Produkt-Backlog, unterteilt nach Teammitgliedern (einschließlich der Spaltenmarkierung „Done“ oder „Not Done“/(grün oder rot markiert) für jede geplante Feature.
- Klassendiagramm mit den Geschäftsdomänenklassen (Klassen von Spring Boot sind nicht erforderlich).

2.2 Eine Zip-Datei hoch, die enthält:

- der als **ZIP-Datei** exportierte Projektcode (einschließlich .java-Dateien)
- .TXT mit Benutzeranmeldeinformationen, um als normaler Benutzer und als ADMIN auf die Anwendung zuzugreifen.
- Link für die Docker-Datei/Image.

5) Vorstellung der Anwendung/ Prüfung

25.01.2027 bis 27.01.2027