

Softwaretechnik [Hausaufgabe Blatt 1. Requirements]

Aufgabe 1. [Steakholder]

A) 1. Studenten:

Die Studierenden möchten sich einfach und effizient für Übungsgruppen anmelden, die in ihren Stundenplan passen. Als Hauptnutzer haben sie daher ein großes Interesse an dem System, da ihre Kapazitäten direkt von den Funktionalitäten des Systems beeinflusst werden und sie den Anmeldeprozess effizienter und fairer abschließen und Konflikte mit anderen Kursen vermeiden können.

2. Dozenten:

Dozenten möchten Übungsgruppen erstellen und diese einfach verwalten. Sie haben ein großes Interesse an dem System, da sie eine zuverlässige Methode benötigen, um die Studierenden den Gruppen zuzuordnen, und zwar in Übereinstimmung mit ihren Lehrplänen.

3. Systemverwalter(Admins)

Die Administratoren sorgen dafür, dass das System reibungslos funktioniert, die Daten sicher verarbeitet werden und alle technischen Probleme gelöst werden. Sie haben ein großes Interesse am System, da die benutzerfreundliche Schnittstelle des Systems es ihnen ermöglicht, die Zuweisung von Gruppen, die Einschreibung von Studierenden und andere mögliche Aufgaben effizienter zu überwachen.

4. Lehrstuhl (Chair of Software & Systems Engineering)

Der Lehrstuhl möchte die Ausbildungsziele und den allgemeinen Auftrag des Fachbereichs unterstützen. Das System kann helfen, die Ziele zu erreichen und zu unterstützen. Der Lehrstuhl ist auch für die Genehmigung des Projektbudgets und des Zeitplans verantwortlich.

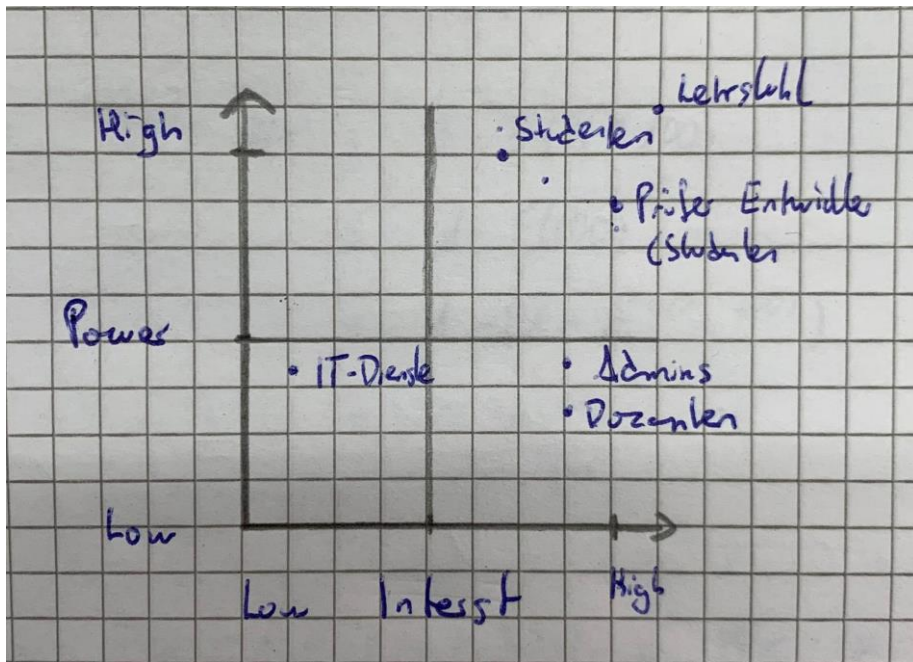
5. IT-Dienste der der Universität

Die IT-Dienste sind für die technische Infrastruktur des Systems verantwortlich, einschließlich der Sicherheit und der Integration mit bestehenden Universitätsanwendungen wie Shibboleth. Sie müssen sicherstellen, dass das System skalierbar ist und den Sicherheitsanforderungen entspricht.

6. Prüfer (Studentische Entwickler/ Anwender)

Die studentischen Entwickler haben die Möglichkeit, sich an der Entwicklung eines Systems zu beteiligen und praktische Erfahrungen zu sammeln (ggf. ihrem Interesse an der Entwicklung nachzugehen). Dabei können sie praktisches Feedback zur Benutzerfreundlichkeit und Funktionalität des Systems geben. Diese Hilfe der Studierenden kann dazu beitragen, Probleme besser und schneller zu erkennen und durch ihre Erfahrung Verbesserungen vorzunehmen, was wiederum für sie selbst besser ist

B)



Aufgabe 2 [Requirements]

A) Funktionale Anforderungen

1. Erstellung von Gruppen:
Dozenten können Übungsgruppen mit spezifischen Sitzungsdetails wie Zeiten und Kapazität erstellen.
2. Anmeldung für Studenten:
Die Studierenden können sich für alle Kurse anmelden, die sie in einem Semester besuchen möchten.
3. Eingabe der Verfügbarkeit:
Die Schüler können Zeiten markieren, in denen sie aufgrund anderer Umstände NICHT verfügbar sind.
4. Gruppenzuweisung:
Das System weist Studenten automatisch Gruppen zu, basierend auf ihrer Verfügbarkeit und Terminkonflikten.
5. Benachrichtigungssystem:
Das System benachrichtigt die Schüler über die Gruppen, denen sie zugewiesen wurden.
6. Manuelle Verwaltung:
Wenn ein Schüler keiner Gruppe zugewiesen werden kann, wird eine manuelle Verwaltung eingeleitet.

B) Qualitätsanforderungen und Qualitätsattribute

1. Benutzerfreundlichkeit:

Das System sollte für Studierende, Dozenten und Systemadministratoren einfach zu bedienen sein.

2. Sicherheit:

Das System muss sicherstellen, dass personenbezogene Daten vor unbefugtem Zugriff geschützt sind, indem es Anmeldedaten der Universität (Shibboleth) verwendet.

3. Skalierbarkeit:

Das System sollte in der Lage sein, eine steigende Zahl von Nutzern zu bewältigen, insbesondere während der Anmeldezeiträume, bis hin zu Tausenden von Studenten.

C) Einschränkung

1. Begrenzung des Budgets:

Für die Entwicklung des Systems steht ein Gesamtbudget von 70.000 Euro zur Verfügung.

D) Projektanforderung

1. Zeitplan für die Einführung:

Das System soll im Wintersemester 2026/27 in Betrieb genommen werden, erste Testversionen sollen zu Beginn des Wintersemesters 2025/26 fertig sein.

E) Prozessanforderung

1. Einbindung der Studenten:

Studierende sollten am Entwicklungsprozess sowohl als Entwickler als auch als Tester teilnehmen, um Feedback und Erkenntnisse aus der Praxis zu sammeln.

Aufgabe 3 [Requirements Validation]

A) Funktionale Anforderungen

Gruppenbildung

Präzision: Erfüllt; es wird festgelegt, was die Dozenten tun können.

Konsistenz: Erfüllt; steht im Einklang mit der Gesamtfunktionalität des Systems.

Überprüfbarkeit: Erfüllt; kann getestet werden, indem überprüft wird, ob Dozenten Gruppen erstellen können.

Gültigkeit: Erfüllt; steht im Einklang mit den Projektzielen.

Beispiel für eine Verbesserung: „Dozenten können Übungsgruppen erstellen, indem sie die Sitzungszeiten, die Kapazität und den Gruppennamen in der Systemschnittstelle angeben.“

Registrierung von Studenten

Präzision: Erfüllt; beschreibt eine klare Handlung.

Konsistenz: Erfüllt; stimmt mit dem Gesamtziel des Systems überein.

Überprüfbarkeit: Erfüllt; kann durch den Registrierungsprozess getestet werden.

Gültigkeit: Erfüllt; unterstützt die Bedürfnisse der Schüler.

Verbesserungsbeispiel: „Studierende können sich über eine spezielle Anmeldeseite für alle Übungsgruppen ihrer gewählten Lehrveranstaltungen in einem Semester anmelden.“

Verfügbarkeit Eingabe

Genauigkeit: Erfüllt; definiert klar, was Studierende tun können.

Konsistenz: Erfüllt; entspricht dem Bedürfnis nach fairen Gruppenzuweisungen.

Überprüfbarkeit: Erfüllt; kann durch Beobachtung des Eingabemechanismus überprüft werden.

Gültigkeit: Erfüllt; berücksichtigt Planungskonflikte der Schüler.

Verbesserungsbeispiel: „Studierende können ihre nicht verfügbaren Zeiten in eine Zeitauswahloberfläche eingeben, die Konflikte mit bestehenden Kursen hervorhebt.“

Gruppenzuordnung

Genauigkeit: Erfüllt; spezifiziert den Zuweisungsmechanismus.

Konsistenz: Erfüllt; unterstützt das Ziel einer fairen Verteilung.

Überprüfbarkeit: Erfüllt; kann durch Überprüfung der Zuweisungen anhand der Verfügbarkeit der Studierenden getestet werden.

Gültigkeit: Erfüllt; berücksichtigt Fairness bei Gruppenzuweisungen.

Verbesserungsbeispiel: „Das System weist den Studierenden automatisch Übungsgruppen zu, basierend auf ihren angegebenen verfügbaren Zeiten und eventuellen Konflikten mit anderen registrierten Kursen.“

Benachrichtigungssystem

Genauigkeit: Erfüllt; beschreibt den Benachrichtigungsmechanismus.

Konsistenz: Erfüllt; notwendig für die Kommunikation.

Überprüfbarkeit: Erfüllt; kann durch Überprüfung der Zustellung der Benachrichtigung getestet werden.

Gültigkeit: Erfüllt; hilft Schülern, informiert zu bleiben.

Verbesserungsbeispiel: „Das System sendet E-Mail-Benachrichtigungen an die Studierenden, um sie innerhalb von 24 Stunden nach Abschluss der Aufgaben über die ihnen zugewiesenen Übungsgruppen zu informieren.“

Manuelle Verwaltung

Genauigkeit: Teilweise erfüllt; es fehlen Details darüber, was die manuelle Verwaltung beinhaltet.

Konsistenz: Erfüllt; entspricht der Notwendigkeit von Ausweichprozessen.

Überprüfbarkeit: Teilweise erfüllt; es ist nicht klar, wie dies gemessen werden soll.

Gültigkeit: Erfüllt; notwendig für den Umgang mit Zuordnungsproblemen.

Beispiel für eine Verbesserung: „Wenn ein Schüler nicht einer Gruppe zugewiesen werden kann, überprüft ein Systemadministrator das Profil des Schülers und weist ihn manuell der nächsten verfügbaren Gruppe zu.“

B) Qualitätsanforderungen und Qualitätsattribute

Benutzerfreundlichkeit

Genauigkeit: Teilweise erfüllt; erfordert spezifische Kriterien für die Benutzerfreundlichkeit.

Konsistenz: Erfüllt; entspricht dem Bedürfnis nach benutzerfreundlichem Design.

Überprüfbarkeit: Teilweise erfüllt; es fehlen Metriken für Usability-Tests.

Gültigkeit: Erfüllt; entspricht den Bedürfnissen der Benutzer.

Beispiel für eine Verbesserung: „Das System muss über eine intuitive Benutzeroberfläche verfügen, die es den Studierenden ermöglicht, die Registrierung mit nur drei Klicks abzuschließen.“

Sicherheit

Genauigkeit: Erfüllt; spezifiziert die Sicherheitsanforderungen.

Konsistenz: Erfüllt; steht im Einklang mit den Datenschutzzielen.

Überprüfbarkeit: Erfüllt; kann durch Sicherheitsprüfungen getestet werden.

Gültigkeit: Erfüllt; wesentlich für den Schutz von Benutzerdaten.

Beispiel für eine Verbesserung: „Das System muss die Shibboleth-Authentifizierung für einen sicheren Zugang nutzen und alle personenbezogenen Daten während der Speicherung und Übertragung verschlüsseln.“

Skalierbarkeit

Genauigkeit: Teilweise erfüllt; benötigt spezifische Skalierbarkeitskriterien.

Konsistenz: Erfüllt; wichtig für zukünftige Anwendungsfälle.

Verifizierbarkeit: Teilweise erfüllt; es fehlen spezifische Metriken.

Gültigkeit: Erfüllt; unterstützt langfristige Nutzung.

Beispiel für eine Verbesserung: „Das System muss bis zu 10.000 gleichzeitige Benutzer während der Spitzenzeiten bei der Registrierung ohne Leistungseinbußen unterstützen.“

C) Einschränkung

Budgetbegrenzung

Genauigkeit: Erfüllt; gibt das Budget klar an.

Konsistenz: Erfüllt; steht im Einklang mit der Projektplanung.

Überprüfbarkeit: Erfüllt; kann anhand der Projektausgaben überprüft werden.

Gültig: Erfüllt; wichtig für die Durchführbarkeit des Projekts.

Verbesserungsbeispiel: Keine Verbesserung erforderlich, da bereits klar.

D) Projektanforderung

Budgetbegrenzung

Genauigkeit: Erfüllt; gibt das Budget klar an.

Konsistenz: Erfüllt; steht im Einklang mit der Projektplanung.

Überprüfbarkeit: Erfüllt; kann anhand der Projektausgaben überprüft werden.

Gültig: Erfüllt; wichtig für die Durchführbarkeit des Projekts.

Verbesserungsbeispiel: Keine Verbesserung erforderlich, da bereits klar.

E) Prozessanforderung

Studentisches Engagement

Genauigkeit: Teilweise erfüllt; es fehlen spezifische Angaben zur Beteiligung.

Konsistenz: Erfüllt; steht im Einklang mit den Zielen der gemeinsamen Entwicklung.

Überprüfbarkeit: Teilweise erfüllt; benötigt klarere Metriken für die Beteiligung.

Gültigkeit: Erfüllt; unterstützt nutzerzentriertes Design.

Beispiel für eine Verbesserung: „Mindestens 20 Studierende sind an der Entwicklungs- und Testphase beteiligt und geben zu bestimmten Meilensteinen Feedback zur Benutzerfreundlichkeit und Funktionalität.“

Durch die Verfeinerung der Anforderungen anhand der gemachten Vorschläge kann das Team die Klarheit verbessern und sicherstellen, dass alle Kriterien erfüllt werden, was einen erfolgreichen Entwicklungsprozess ermöglicht.

Aufgabe 4 [Use Case]

Title	Registrierung von Übungsgruppen		
Actor	Student		
Preconditions	Der/die Studierende ist in mindestens einem Kurs eingeschrieben, dem Übungsgruppen zugeordnet sind. Der/die Studierende verfügt über gültige Zugangsdaten zur Anmeldung am EGD-System.		
Postconditions	Der Student wird auf der Grundlage seiner Eingabeverfügbarkeit den verfügbaren Übungsgruppen zugewiesen. Der Student wird über Gruppenzuweisungen oder Konflikte benachrichtigt.		
	Step	Description	
Main success scenario	1	Log In: Der Studierende meldet sich mit den Zugangsdaten der Universität beim EGD-System an	
	2	View Available Groups: Das System zeigt die verfügbaren Übungsgruppen für die Kurse an, für die der Student eingeschrieben ist.	
	3	Enter Availability: Der Schüler gibt seine nicht verfügbaren Zeiten in das System ein	
	4	System Checks Availability: Das System vergleicht die Angaben des Schülers mit den verfügbaren Übungsgruppen und es gibt keinen Konflikt	
	5	Assignment of Groups: Das System ordnet den Studenten automatisch den verfügbaren Gruppen zu	
	6	Notification: Die Studierenden erhalten eine Bestätigungs-E-Mail, in der sie über die ihnen zugewiesenen Gruppen informiert werden oder die Notwendigkeit, Terminkonflikte manuell zu lösen	
Alternative Flows	4a.	No Available Slots: wenn das System nach Auswertung der Eingaben des Schülers feststellt, dass keine geeigneten Übungsgruppen vorhanden sind	
	*4b.	Choosing a Group even without availability: Der Student wählt eine Gruppe aus um angemeldet zu sein und die Unterlagen zu bekommen und im Nachgang möglicherweise zu wechseln.	
	*4c.	Canceling: der Student bricht den laufenden Prozess ab, da er keine Passende Zeit findet	
	6a.	Manual Resolution Required: Wenn ein Student einen Dozenten für eine besondere Vereinbarung kontaktieren muss/ Änderung im Nachgang	
Trigger	Der Student möchte sich für eine Übungsgruppe anmelden		

Assumptions:

- Studenten können sich für mehrere Kurse einschreiben, deren Stundenpläne sich überschneiden können.
- Das System kann mehrere Kurse mit unterschiedlicher Verfügbarkeit der Übungsgruppen verwalten.
- Kommunikationsmethoden (E-Mail) sind für Benachrichtigungen eingerichtet.

Extensions:

- Gruppenanmeldung als Team: In zukünftigen Iterationen können sich Studierende möglicherweise als Gruppe für bestimmte Kurse oder Sitzungen anmelden, was zusätzliche Funktionen für die Teamverwaltung erfordert.