

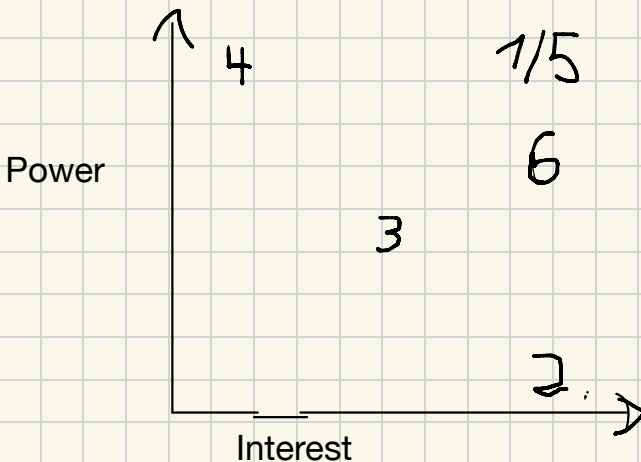
Nr.1

a)

Stakeholder:

1. Universität als Institution: Durch effektiv laufende Systeme ist der Workload durch Fehler, Studenten anfragen und Management geringer. Der Ruf ist ebenfalls besser. Sie Stellen das Budget.
2. Studenten: System erleichtert die Planung des Stundenplans und ermöglicht allen an ihren Kursen teilzunehmen
3. Professoren: Diese sind an einer benutzerfreundlichen Bedingung interessiert, da sie über das System ihre Kurse bereitstellen müssen.
4. Shibboleth: Da Sicherheit und Datenschutz eine wichtige Rolle in der Software Entwicklung spielen ist Shibboleth ein wichtiger Stakeholder.
5. Department of CS: Als Administratoren des Projektes sind diese an einer Reibungslosen und sicheren Entwicklung besonders interessiert
6. Chair of SE: Da dieser explizite an der Entwicklung der Software teilnehmen möchte indem er seine Studenten programmieren lässt ist dieser aush ein Stakeholder in dem Projekt

b)



Nr.2

a) Functional Requirements

1. Gruppenerstellung durch Dozenten
2. Studenten müssen in der Lage sein die Zeiten, wo sie nicht können, einzutragen
3. Automatische Verteilung der Studenten
4. Automatische Benachichtigung
5. Login über Shibboleth (für Sicherheit)
6. Manuelle Nachbearbeitung bei komplikationen

b) Quality Requirements

1. Benutzerfreundlichkeit (Usability)
2. Leistungsstark (Performance)
3. Sicherheit (Security)

c) Constrain

Das System muss in Java geschrieben werden.

d) Projekt Requirements

Das Budget ist 70000

e) Process Requirements

Testphase im WS 2025/26 und live gehen WS 2026/27

Nr.3

1. Gruppenerstellung durch Dozenten:

Precision: Es ist festgehalten, dass Uhrzeit sowie Teilnehmeranzahl festgelegt werden kann. Wie es um die Räume steht ist nicht festgehalten und könnte im Konflikt mit den Zeiten stehen (wenn kein raum frei ist aber Uhrzeit trotzdem angeboten wird).

Consistency: Stimmt mit anderen Requirements überein (e.g Verfügbarkeit)

Verifiability: Durch einfaches Testen in dem Program kann man dies überprüfen

Validity: Stimmt mit den echten Bedürfnissen überein, da Dozenten ein Weg brauchen solche Gruppen zu erstellen um ihren job zu machen

Verbesserung: Dozenten müssen in der Lage sein den Raum in Echtzeit zu überprüfen, ob dieser bereits geblockt ist, um keine Probleme im nach hinein zu verursachen.

2. Zeitangabe der Studenten:

Precision: Angabe das Studenten nur ihre Zeiten angeben müssen ist eindeutig, zu hinterfragen ist es, wie das System damit umgeht wenn es keine Übereinstimmungen gibt.

Consistency: Stimmt mit den anderen Anforderung überein (e.g Studienplanerstellung)

Verifiability: Es kann getestet werden ob Studenten ihr nicht erreichbaren zeiten eingeben können und ob diese auch richtig gespeichert und verarbeitet werden

Validity: Dies ist wichtig für die gleichmäßige verteilung der Studenten auf die Kurse

Verbesserung: Wenn dieses System auf einen Studenten für mehrere Kurse angewendet wird muss sichergestellt werden, das das System nicht eine Zeit Doppelt belegt, weil es einen der neuen Kurse noch nicht einbezieht.

3. Automatische Verteilung der Studenten

Precision: Der begriff "fair" ist ein bisschen weit gefasst und könnte sowohl auf minimal Konflikte im terminkalnder hinweisen aber auch auf gleichmäßige Gruppenverteilung.

Consistency: Stimmt mit dem vorherigen Anforderungen überein (Kann immerhin nur unter den da genannten Rahmenbedingung arbeiten)

Verifiability: Unter dem Vagen begriff "fair" schwer zu Testen (was ist fair), Messbar wäre die Konfliktmenge oder eine gleichmäßige Aufteilung

Validity: Das Ziel stimmt mit dem momentanen Problem überein, das Studenten die momentane Aufteilung unfair finden

Verbesserung: Ziel nach dem die Aufteilung durchgeführt wird genauer Definieren z.B. durch möglichst kleine Konflikt menge

4. Benachichtungssystem

Precision: Das Zeil ist klar, aber der Kommunikationskanal ist nicht explizit genannt

Consistency: Überschneidet sich nicht mit anderern Anforderungen

Verifiability: Es kann überprüft werden, ob Nachrichten losgeschickt werden und ankommen.

Validity: Wichtig, damit Studenten wissen ob noch Probleme und Überschneidungen bestehen

5. Login mit Shibboleth

Precision: Ist klar Daten existieren bereits

Consistency: Stimmt mit den Sicherheitsanforderungen überein

Verifiability: Einfach zu Testen per Login Daten der betroffenen

Validity: Spiegelt Sicherheitsanforderung gut wieder

Verbesserung: Keine Notwendig

6. Manuelle Nachbearbeitung

Precision: Sinn ist zwar klar, Ausführung aber nicht (muss sich der Student kümmern oder der Dozent etc.)

Consistency: Stimmt mit der Anforderung der Automatisierung überein in den Fällen, wo diese an ihre Limits gerät

Verifiability: Kann getestet werden ob Administratoren manuell Studenten Gruppen zuweisen können, aber da das Ziel der Nachbearbeitung nicht klar definiert ist, schwer

Validity: Wichtig um Randfälle wo Automatische Prozesse nicht greifen

Verbesserung: Admins bzw vielleicht auch Dozenten müssen in der Lage sein Gruppen bearbeiten zu können um manuell Leute hinzufügen zu können und Gruppengrößen anzupassen

7. Benutzerfreundlichkeit

Precision: Einfach zu Bedienen ist vage und kann für verschiedene Leute unterschiedliche Sachen bedeuten

Consistency: Geht gegen keine andere Voraussetzung

Verifiability: Umfragen zur Benutzerfreundlichkeit können gemacht werden. Richtige Richtwerte / Matrix gibt es dennoch nicht

Validity: Da viele dieses System benutzen müssen stimmt es mit den echten Bedürfnissen überein

Verbesserung: Ablauf muss intuitiv sein, test Beispiel: 90% der User in 5 Minuten fertig

8. Leistung

Precision: Anforderung Tausende Studenten gleichzeitig zu Bearbeiten etwas ungenau
(keine Zeitsapnne gegeben in welcher diese tausend anfragen abgeabeitet werden müssen)

Consistency: Keine Anforderung setzt eine Langsame Rechenleistung voraus

Verifiability: Überlastungs Test können die Maximale anzahl an anfragen Überprüfen

Validity: Wichtig Für Zeiten mit viel Verkehr

Verbesserung: Beispiel Test matrix wäre 10000 Anfragen die alle nicht Länger als 3 Sekunden brauchen sollen

9. Sicherheit

Precision: Schutz vor unberechtigten ist klar höchsten kann man noch anforderung an die verschlüsselungsart stellen

Consistency: Stimmt mit dem Login über Shibboleth überein

Verifiability: Pen Tests können die sicherheit überprüfen

Validity: Sicherheit ist wichtig da mit Sensiblen persönlichen Daten umgegangen wird

Verbesserung: Perse keine, höchsten die anforderung das keine Verwundbaren stellen gefunden werden dürfen und das alles Datenschutzkonform bearbeitet wird

10. Constraint

Precision: Java also eindeutig

Consistency: Steht keiner Anforderung im Weg

Verifiability: Selbsterklärend

Validity: Ermöglicht Studenten die Teilnahme am Projekt

Verbesserung : Keine

11. Projekt Requirement

Precision: Klar gestellt 70000

Consistency: Stimmt mit der Reichweite des Projekts überein

Verbesserung: Keine notwendig

12. Process Requirement

Precision: Testzeitraum und Umfang klar, aber Feedback mechanismus ungeregelt

Consistency: Stimmt mit dem zeitplan überein

Verifiability: Demo muss geplant vor WS 2025/26 live gehen

Validity: Wichtig um Qualitäts Standards einzuhalten

Verbesserung: Zeitraum für Feedback entgegennahme und Datenauswertung
bereitstellen

Nr.4

Title: Kurs belegen

Actors: Studenten, Professoren

Preconditions: Student hat Shibboleth Login, Student muss in Kursen eingeschrieben sein

Trigger: Student öffnet Entsprechende Internetseite

Main Success Scenario:

1. Student. Einloggen in Shibboleth
2. System: Verifiziert Student
3. Student: Gibt die Zeiten an wo dieser nicht kann
3. System: Überprüft ob alle Kurse Konfliktfrei vergeben werden können und teilt den Studenten zu
4. System. Schickt bestätigungs Mail mit Übersicht der Kurse

Alternative paths:

- 2a1. System: Shows Error Falsche Daten
- 2a2. Return zu Schritt 1
- 4a. System: Kann für einen Kurs keine konfliktfreie Lösung finden und schickt dementsprechend eine Nachricht an den Betroffenen. Dieser muss nun den Entsprechenden Dozenten informieren um eine Lösung zu finden
- 4b. Das System erkennt Konflikt zwischen zwei Gruppen und wählt automatisch eine der beiden aus. Erneut wird eine Mail rausgeschickt, um den Zustand aufzuklären, damit sicher der Student an einen Dozenten wenden kann