

Homework 1 - Bearbeitung

Nr 1 a)

Marc Rennefort 7421290
Yves-Pierre Weselek 7478442
Daniel Schnicke Vaz 7424876

1. Studenten: Die Studenten sind als spätere primäre Nutzengruppe direkt von der erfolgreichen Entwicklung des Systems betroffen. Sie sind direkt von der Funktionalität des Systems abhängig, da sie über dieses in zukünftige Kurse belegen sollen. Zudem besteht bei ihnen das Hauptinteresse für die Entwicklung des Systems, da sie das momentane System als ungerecht betrachten.

2. Dozenten: Die Dozenten sollen im späteren System Übungsguppen und deren Rahmenbedingungen förmigen können. Für sie ist die Funktionalität und die einfache Handhabung des Systems entscheidend, da es direkt das Management ihrer Kurse beeinflusst. Ein dysfunktionales oder schwer zu bedienendes System würde für sie erheblichen Mehraufwand bedeuten.

3. Universitätverwaltung: Die Universitätverwaltung hat ein großes Interesse daran, dass bei der Entwicklung des Systems alle vorgegebenen Regeln und Gesetze (z.B. Datenschutz) eingehalten werden. Zudem hat die Universitätverwaltung als Geldgeber ein großes Interesse daran, dass ihre Mittel sinnvoll genutzt werden und später ein funktionales System entsteht mit dem die Studenten und Dozenten gut und gezielt arbeiten können.

4. Systemadministratoren: Als Betreuer des Systems haben die Systemadminis ein großes Interesse daran ein sicheres, stabiles und benutzerfreundliches System zu entwickeln. Dieses ist entscheidend für die erfolgreiche Implementierung und den Betrieb des Systems. Zudem ist es für sie entscheidend, dass das System im späteren Verlauf einfach zu warten und wie gefordert einfach skalierbar ist.

5. Lehrstuhl Software & Systems Engineering: Der Lehrstuhl ist an der Entwicklung des Systems interessiert, da er so seinen Studierenden Einblicke in ein echtes Softwareentwicklungsprojekt geben will und ihnen auch die Möglichkeit geben will daran mitzuwirken.

6. IT-Entwicklungsteam: Die Entwickler sind am Erfolg ihrer Entwicklung interessiert, da

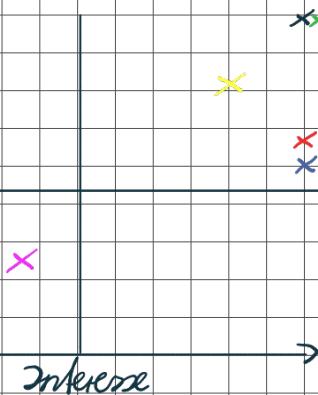
sie ein möglichst gutes Produkt auf den Markt bringen wollen

und da sie für die Arbeit bezahlt werden.

e)

1

Macht



legende

● = IT-Entwicklungsteam

● = Studenten

● = Dozenten

● = Systemadministratoren

● = Universitätverwaltung

● = Lehrstuhl

Erklärung der Zuordnung:

Dozenten / Studierende: Sie sind die Hauptzielgruppe des Systems und haben ein dementsprechend hohes Interesse an dessen erfolgreicher Entwicklung. Dadurch, dass sie die späteren Benutzer des Systems sind haben sie auch die Macht auf die Entwicklung des Systems in einem gewissen Rahmen Einfluss zu nehmen

IT-Entwicklungsteam / Systemadminim: Als Entwickler können sie direkte Änderungen am System vornehmen und haben entsprechend hohe Macht. Zudem sind sie sehr interessiert an einer erfolgreichen Entwicklung, für die sie schließlich auch bezahlt werden.

Universitätverwaltung: Als Geldgeber ist die Universitätverwaltung ein sehr mächtiger Stakeholder. Entsprechend hoch ist auch sein Interesse an der erfolgreichen Entwicklung bzw. der sinnvollen Verwendung ihrer Mittel.

Lehrstuhl: Der Lehrstuhl hat zwar ein gewisses Interesse daran seinen Studierenden das entwickeln eines solchen Systems nahe zu bringen er könnte dies aber auch anders machen. Er kann auch nur über die studentischen Entwickler einen kleinen Einfluss auf das Projekt ausüben.

Nr 2

Funktionsanforderungen:

1. Gruppenzuweisung: Das System muss in der Lage sein die Studierenden gerecht auf die Gruppen zu verteilen.
2. Markierung: Studierende müssen in der Lage sein Zeiten zu markieren, zu denen sie nicht können.
3. Gruppenverstellung: Dozenten müssen in der Lage sein Gruppen zu erstellen und Rahmenbedingungen (wie LpW/GruppengröÙe) für diese festzulegen.
4. Anmeldung: Studierende und Dozenten müssen sich mit ihren Universitätsanmeldedaten (Shibboleth) anmelden können.
5. Benachrichtigungen: Nach der Zuweisung sollen die Studenten über ihre entsprechend Zuweisung informiert werden.
6. Rollen: Das System sollte Rollen (Studenten, Dozenten etc.) mit unterschiedlichen Berechtigungen zur Verfügung stellen.

Qualitätsanforderung:

1. Skalierbarkeit: Das System sollte in der Lage sein mehrere tausende Benutzer gleichzeitig zu verwalten.
2. Benutzerfreundlichkeit: Das System sollte für die Benutzer möglichst einfach zu bedienen sein.
3. Sicherheit: Das System sollt die persönlichen Daten der Studierenden vor Unbefugten Zugriffen schützen.

Constraints:

Das Projekt soll in Java geschrieben werden.

Projektanforderung:

Das System soll zum WS 2026/27 veröffentlicht werden. Eine erste Test version sogar bereits zum WS 2025/26

Probenanforderung:

Die Studierenden sollen aktiv in die Entwicklung und das Testen miteinbezogen werden.

Ur. 3)

Requirements Validation

Die funktionalen Anforderungen erfüllen, bis auf 1 und 6, die Präzision, da sie sehr spezifisch und strukturiert sind.

Alle funktionalen Anforderungen erfüllen zudem die Konsistenz, da es keine Widersprüche bzw. Redundanzen gibt.

Dieser Punkt erfüllt sie die Verifizierbarkeit, da man die Anforderungen leicht testen kann, um so die Richtigkeit zu überprüfen.

Auch wird die Validität erfüllt, weil alle Punkte reale Anforderungen sind, die benötigt werden.

1. Gruppenzuweisung:

Die Präzision wird nicht erfüllt, da nicht beschrieben wird, was unter einer gerechten Aufteilung zu verstehen ist.

Man könnte ergänzen, dass die verfügbaren Zeiten der Studenten und Konflikte mit parallelen Kursen berücksichtigt werden. Zudem sollte man das Ziel ergänzen, dass das System die Anzahl an Studenten, die keinem Kurs zugeordnet werden können, minimiert wird.

6. Rollen:

Die Präzision wird hier ebenfalls nicht erfüllt, da nicht darauf eingegangen wird, welche unterschiedlichen Funktionen die einzelnen Rollen haben und wie die Rollen vergaben werden.

Man könnte die Rollen bei der Registrierung verteilen. Das System erkennt dann anhand des Benutzernamens ob es sich um einen Dozenten oder Studenten handelt und weist entsprechende Rollen zu.

Die Qualitätsanforderungen erfüllen die Präzision, da sie ebenfalls spezifisch und strukturiert sind.

Sie erfüllen zudem die Konsistenz, da es keine Widersprüche bzw. Redundanzen gibt.

Den Punkt der Verifizierbarkeit erfüllen sie jedoch nicht, da man die Skalierbarkeit und Benutzerfreundlichkeit nur schwer testen kann.

Da die Sicherheitsanforderung lässt sich nur begrenzt testen.
Die drei Anforderungen sind jedoch reale Anforderungen, die benötigt werden, und erfüllen somit die Validität.

Die Rahmenbedingung, dass das Projekt in Java geschrieben werden soll erfüllt alle Punkte.

Es ist eine spezifische und strukturierte Anforderung und somit präzise. Des Weiteren ist diese korrekt, da es keine Widersprüche bzw. Redundanzen gibt. Auch ist sie verlässlich, da es leicht zu überprüfen bzw. offensichtlich ist. Abschließend ist sie valide, da so auch Studenten mitarbeiten können.

Die Projektanforderung erfüllt ebenfalls alle Punkte.

Sie ist eine spezifische und strukturierte Anforderung und somit präzise. Des Weiteren ist diese korrekt, da es keine Widersprüche bzw. Redundanzen gibt. Auch ist sie verlässlich, da es leicht zu überprüfen bzw. offensichtlich ist. Abschließend ist sie valide, da es eine reale, trennende Anforderung ist, die benötigt wird.

Auch die Prozessanforderung erfüllt alle Punkte.

Sie ist eine spezifische und strukturierte Anforderung und somit präzise. Des Weiteren ist diese korrekt, da es keine Widersprüche bzw. Redundanzen gibt. Auch ist sie verlässlich, da es leicht zu überprüfen bzw. offensichtlich ist. Abschließend ist sie valide, da es eine reale Anforderung ist, die Steuerende benötigen.

Szenario 1:

Titel: Kurswahl

Aktion: Student

Preconditions: 1. Student hat einen Account für das EGD-System

2. Student ist in mind. einen Kurs eingeschrieben

Postconditions: Der Student erhält eine Benachrichtigung über seine Terminbuchungen oder eventuelle Konflikte und wird ggf. in die entsprechenden Übungsgruppen eingetragen

Main success scenario: 1. Student: Eingabe der Logon Daten

2. System: Überprüfung der Logon Daten

3. System: Anzeigen der verfügbaren Übungsgruppen pro Kurs

4. Student: Eingabe der Präferenzen durch Markierung der Zeiten zu denen der Student nicht kann

5. System: Erstellung eines passenden Zeitplans und Eintragung in die entsprechenden Gruppen

6. System: Bestätigt die Terminbelegung und sendet einen Zeitplan per email

Alternative paths: 2a 1: [Benutzer verwendet falsche Anmelddaten]

System: Zeigt Fehlermeldung an

2a 2: Student: Bestätigt Fehlermeldung

→ Zurück zu Schritt 1

5a 1: [System konnte mind. eine Übunggruppe nicht zuordnen]

System: Informiert Student über den Konflikt

5a 2: Student: Informiert Dozent

→ Use Case endet