

Homework 2

- 1)
 1. Einführung
 2. Nutzungsprofile
 3. Anforderungen
 - 3.1 Funktionalo Anforderungen
 - 3.2. Qualitäts Anforderungen
 - 3.3. Constraints
 - 3.4. Projekt Anforderungen
 - 3.5. Prozess Anforderungen
 4. Use Cases
 - 4.1 Use Case: upload a file
 - 4.2. Use Case Diagramm

2) 1. Einführung

Ziel von Klipsias ist es die Organisation und zu erleichtern und zu vereinfachen. Dazu sollen die beiden System der Universität zu Köln, Llias und Klips, zu einem einheitlichen System zusammengeführt werden, in welchen alle Funktionalitäten beider Systeme enthalten sind. Das Klipsias bietet in einem übersichtlichen alle essentiellen Funktionen für Organisation und Verwaltung an der Universität zu Köln, so dass nicht zwei Systeme parallel genutzt werden müssen.

2) a) 2. Nutzungsprofile

Studenten: Die Studenten sind die größte Nutzergruppe von Klipsias. Sie benötigen das System täglich. Kursanmeldung, Materialbeschaffung, etc.

Professoren: Die Professoren nutzen beide Systeme um ihre Systeme um entweder Kurse anzubieten oder Lösungsmaterialien bereit zu stellen.

Tutorien: Tutorien die Übungen/Tutorien halten haben auch Interesse Materialien hochzuladen aber auch Abgaben & zu Aufgaben von Studenten zu bekommen

Fakultäten: Fakultäten haben insbesondere in Form des Prüfungsamts ein Interesse daran die Noten der Studentin zu erfahren

Rectorat: Als Auftraggeber vor allem daran Interesse das Klipsias den Angestellten und Studierenden den Alltag erleichtert

Sekretariat: Vor allem interessiert daran den Studierenden Dokumente schnell und unkompliziert digital zur Verfügung zu stellen.

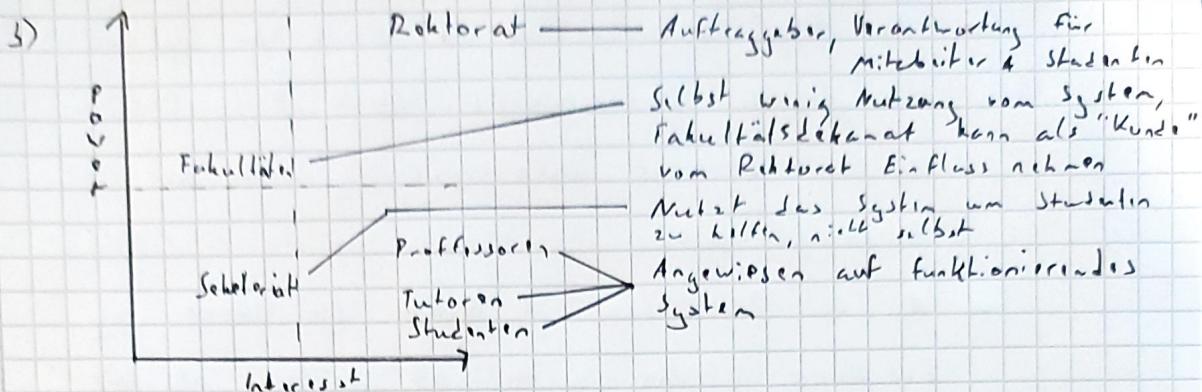


Abb. 1 Stakeholder Power/Interest Grid

4) 3. Anforderungen

a) 3.1. Funktionale Anforderung

- When a exam is finished the Klipsias -System shall request the ^{professor} teacher to assign a grade (1)
- The Klipsias -System allow a user to be student and tutor at the same time. (2)
- While uploading material the Klipsias -System shall run a security check on the files (3)
- Where tutor book room the Klipsias -System shall hide unavailable rooms (4)
- If a user is located in a course room he has no acces to then the Klipsias -System shall relocate him the user to the startpage (5)
- If a student tries to sign up for a course while he already completed this note the Klipsias -System shall stop the sign up and show the student a error message. (6)
- As a the Fakultät I want to see the grades for all completed courses from all students of the Fakultät (7)
- As the Rektorat I want to create new users (8)

- As the Sekretariat I want to upload template documents that fills for each student automatically with his personal data. (9)

- As a professor I want to share my lecture slides with all students of the course. (10)

b) 3.2 Qualitäts Anforderungen

- Klipsias shall be available 99,99% of the time (m)
- Klipsias shall be useable in german and english (12)
- Klipsias shall have a secure communication between user and client (13)
- Klipsias shall be accessible from every web browser (14)

c) 3.3 Constraints

- Klipsias must be written in Java. (15)
- The source code of Klipsias must be stored at github (16)

d) 3.4 Projekt Anforderungen

- Klipsias shall be released on 5th February 2024 (17)
- The amount costs of developers shall not exceed 1 (18)

e) 3.5 Prozess Anforderungen

- Every two weeks a interim state shall be shown (19)
- Every new feature shall be developed on its own branch (20)

(Nachtrag Wichtigkeit Requirements)

(1) - Noten sollen schnell eingetragen werden um Fehlertücher zu verhindern.

(2) - Studenten seien manchmal Übungsguppen und sollen auch die Rechte von TutorInn, sich aber weiterhin über die Studentenrolle und nicht TutorInnrolle einloggen

(3) - Um zu verhindern dass über eine schadhafte Datei Massen an Studenten von Viren befallen werden

(4) - Für eine bessere Übersicht von verfügbaren Räumen

(5) - Für den Fall das ein User bspw. über einen Link in einer für ihn nicht zugelassenen Bereich kommt

(6) - Zu verhindern, dass sich Personen ausserhalb für bereits abgeschlossene Module erneut anmelden

(7) - Das Prüfungsamt sollte Zugriff auf die Noten haben.

(8) - Das Rektorat (als Admin agierend) soll neue Mitarbeiter, Professoren, und Studenten Accounts anlegen können.

(9) - Da nicht häufig tausende Dokumente erstellt werden soll das Sekretariat Templates hochladen können die dann entsprechend von Klipsilis neu ausgefüllt werden

(10) - Materialien sollen wie in illias weiterhin normal geteilt werden können

(11) - Klipsilis soll immer erreichbar sein, da es keine festen Arbeitszeiten für Studenten gibt

(12) - Für internationale soll die Klipsilis auch auf Englisch verfügbar sein

(13) - Bei mit https sollte die Standard Sicherheit gegeben sein

(14) - Sollte nun überall zugreifbar sein ~~außer~~ auf die Klipsilas Webseite

- (15/16) - Java und GitHub Voraussetzung für JUnit-Tests
- (17) - Gesetztes Relogidatum, hier bspw. Datum der Klausur
- (18) - Festgelegte Kanten die nicht überschritten werden darf
- (19) - Abgabe zur Qualitätskontrolle durch den Auftraggeber oder Tester, hier Abschluß der Aufgabe
- (20) - Jedis Feature kann eigenen Branch, damit diese sich nicht überschneiden

5) Use Cases (2) Upload a File:

- a) (2) Das folgende Use Case leitet sich aus den folgenden zwei funktionalen Anforderungen ab.
 - (3) - While uploading material the Klipsias-System shall run a security check on the file.
 - (10) - As a professor I want to share my lecture slides with all students of the course
- ~~4.1 - Use Case: Upload a file~~

Grade Exams:

- (1)

Book a room:

- (4)

Enter a course room:

- (5)

Sign up a course:

- (6)

Create user:

- (8)

Fill templates:

- (9)

Login / Assign New Permissions

- (2)

- (8)

(Explanation: If a user logs in to the system, sich ein user anmeldet soll nach der Verifizierung dem User seine Rolle (Student, Professor, etc) zugewiesen und damit auch seine Berechtigungen.)

View study progress:

- (207)

b) 4.1 Use Case: Upload a file

Title: Upload a file

Actor: User

Preconditions: - User did login

- User is located in a room

Trigger: - User uploads a file
prior execution to

Main Success scenario: Step Description

1 System: Checks users permission

2 System: Opens File Explorer

3 User: Selects a file

4 System: Runs a security check on the file

5 User: Confirms the upload

6 System: Uploads the file to the room
→ Use Case Ends

Alternative Paths	Step	Description
	1b1	[User has no permission to upload] System: Shows error message
	1b2	User: Confirms error message → Use Case Ends
	4b1	[Security check fails] System: Sends a notification to the server administration
	4b2	System: Shows error message User: Confirms error message → Return to step 2
	5b1	User cancels the upload] System: Shows a message successfully canceled → Use Case Ends

c) 4.2 Use case Diagramm

