

SEP Projektmappe

PROJEKTMAPPE DES PROJEKTES

Project X

Dokumentation des Projektes

Gruppe X:

Frederik Gnass 3099469

Hana Habibi 3100348

Noemi Kallweit 3098817

//Thorben Kronewald-Lamken 3091481

Vera Theodora Mitschulat 3093828

Kurinchinilavan Thangeswaran 3098849

Inhalt

Projektbeschreibung	3
Zyklus I.....	9
Definition der Verantwortlichen für die Artefakte der Spezifikation.....	9
User-Stories	11
Papierprototypen	18
[OBJ]	
Szenarien (MSCs).....	24
[OBJ]	
Strukturdiagramm (Klassendiagramm)	39
Funktionalitätsplanung.....	42
Systemtests	50
Zyklus II	56
Definition der Verantwortlichen für die Artefakte der Spezifikation.....	Fehler! Textmarke nicht definiert.
User-Stories	59
Papierprototypen	64
Szenarien (MSCs).....	73
Strukturdiagramm (Klassendiagramm)	86
Funktionalitätsplanung.....	89
Unitests	94
Systemtests	95
Zyklus III.....	100
Definition der Verantwortlichen für die Artefakte der Spezifikation.....	101
User-Stories	102
Papierprototypen	104
Szenarien (MSCs).....	108
Strukturdiagramm (Klassendiagramm)	113
Funktionalitätsplanung.....	115
Unitests	117
Systemtests	118
Nutzerhandbuch.....	121
Technische Anforderungen	121
Installationsanleitung	121
Bedienungsanleitung.....	121

Projektbeschreibung

Student Education Platform

Einleitung

Die Student Education Platform soll ein Programm werden, das die universitäre Lehre für zwei wesentliche Rollen unterstützt: Studierende und Lehrende. Studierende sollen sie als Lernplattform nutzen, um bspw. auf Lehrmaterial zugreifen, mit anderen Studierenden zusammenarbeiten, Quizfragen beantworten und ihren Kalender verwalten zu können. Lehrende sollen bspw. Lehrveranstaltungen erstellen und Lehrmaterial zur Verfügung stellen können. Das Programm soll auf einer Client-Server-Architektur aufbauen; der Server soll von mehreren Clients gleichzeitig erreichbar sein.

Login Daten - Lehrender

olga.meyer@mein.gmx

ProjectX!!!

Login Daten - Student

max.mustermannSEP@gmx.de

ProjectX!!!

Zyklus1

Im ersten Zyklus soll zunächst die Grundstruktur des Programms, bestehend aus einem Client und einem Server, implementiert werden. Sämtliche Daten des Programms (Daten über Studierende und Lehrende, Daten über Lehrveranstaltungen etc.) sollen persistent gespeichert werden.

(Registrierung der Nutzer)

Es soll Nutzern möglich sein, sich entweder als Lehrende oder als Studierende zu registrieren und damit ein Nutzerkonto zu erstellen. Diese Konten sollen persistent gespeichert werden und über folgende Information verfügen: Jeder Nutzer hat ein Profil, das aus Vor- und Nachname, Passwort, E-Mailadresse, Profilbild und privater Adresse besteht. Darüber hinaus können Lehrende einen zugehörigen Lehrstuhl und ein Forschungsgebiet angeben. Studierende haben eine Matrikelnummer und ein Studienfach. Dabei soll die Matrikelnummer aus 7 Stellen bestehen, eindeutig sein und dem Studierenden beim Erstellen seines Profils automatisch vom System zugewiesen werden. Mit Ausnahme von Vor- und Nachname, Mailadresse und Matrikelnummer sollen alle Informationen nachträglich bearbeitet werden können.

(Verwaltung von Lehrveranstaltungen)

Jeder Lehrende hat die Möglichkeit, neue Lehrveranstaltungen in der Student Education Plattform anzulegen. Hierzu kann er jede Lehrveranstaltung einzeln durch eine manuelle Eingabe in einem entsprechenden Formular erstellen. Alternativ kann er eine beliebige Zahl von Lehrveranstaltungen durch das Einlesen einer vordefinierten CSV-Datei erstellen. Jede Lehrveranstaltung hat einen eindeutigen Titel, ist entweder eine Vorlesung oder ein Seminar und ist einem bestimmten Semester zugeordnet (bspw. Sommersemester 2021). Jede Lehrveranstaltung hat eine eigene Übersichtseite im System, in der alle zugehörigen Informationen für teilnehmende Nutzer angezeigt werden. Lehrende können hierüber Lehrmaterialien bereitstellen, auf die alle Teilnehmer zugreifen können.

(Zuordnung von Nutzern zu Lehrveranstaltungen)

Die Liste aller in der Student Education Platform angelegten Lehrveranstaltungen soll für alle Nutzer sichtbar sein. Lehrende und Studierende können Lehrveranstaltungen beitreten. Lehrende werden den Lehrveranstaltungen, die sie selbst angelegt haben, automatisch als Lehrende zugeordnet. Lehrende können darüber hinaus Studierende anhand ihres Namens oder ihrer Matrikelnummer suchen und manuell in ihre Veranstaltung einschreiben. Für jede

Lehrveranstaltung soll es eine Teilnehmerliste geben, die von den Teilnehmern eingesehen werden kann. Die Teilnehmerliste soll die eingetragenen Nutzer in den zwei Kategorien „Lehrende“ und „Studierende“ aufteilen.

(Nutzerprofile)

Jeder Nutzer verfügt über ein Profil, das durch einen Klick auf den entsprechenden Eintrag in der Teilnehmerliste aufgerufen werden kann. Der eigene Eintrag in der Teilnehmerliste soll entsprechend visuell hervorgehoben werden. Im Profil werden die oben genannten Informationen sowie die Lehrveranstaltungen, an denen der Studierende teilnimmt/die der Lehrende organisiert, angezeigt. Für Studierende sind jedoch die privaten Adressen und Matrikelnummern der übrigen Teilnehmer nicht sichtbar. Der Lehrende kann alle Informationeneinsehen. Außerdem kann jeder Nutzer die eigenen Informationen sehen.

(Login-Vorgang und Startfenster)

Für das Einloggen soll der Nutzer ein Feld nutzen können, in dem er entweder seine Matrikelnummer oder seine E-Mailadresse eingibt. Außerdem muss er sein Passwort eingeben. Bei einem erfolgreichen Login hat er Zugriff auf sein Profil, kann die Lehrveranstaltungen sehen, in die er eingeschrieben ist, und weitere Lehrveranstaltungen suchen und diesen beitreten. Die Lehrveranstaltungen werden nach zugeordneten Semestern gruppiert. Diese Gruppen sind chronologisch abwärts aufgelistet, d.h. die Lehrveranstaltungen des aktuellen Semesters erscheinen ganz oben.

Zyklus 2

(Projektgruppen)

Im zweiten Zyklus soll das Programm der Lehrveranstaltungen durch Projektgruppen erweitert werden. Diese sollen sowohl von Lehrenden als auch von Studierenden erstellt werden können. Lehrende können die Studierenden zu einer Projektgruppe manuell hinzufügen, Studierende können diese nur für die Suche durch andere Studierende bereitstellen. Jede Projektgruppe verfügt über einen Titel, einen Chatraum, in dem sich alle Mitglieder der Gruppe austauschen können, über eine gemeinsame ToDo-Liste, die von jedem Mitglied bearbeitet werden kann und über die Möglichkeit, Dateien untereinander auszutauschen. Todos auf der ToDo-Liste sollen einzelnen Gruppenmitgliedern zugeordnet werden können.

(Kommunikation)

Die Mitglieder einer Lehrveranstaltung (Vorlesung, Seminar, Projektgruppe) sollen sich gegenseitig private Nachrichten schicken und diese beantworten können. Außerdem sollen Studierende Freundschaftsanfragen an die Mitglieder ihrer gemeinsamen Lehrveranstaltungen schicken können. Nach deren Bestätigung, erscheinen die Freunde in der jeweiligen Freundesliste. Die Freundesliste soll wie die persönliche Lehrveranstaltungsliste, auf dem Startbildschirm nach Login angezeigt werden.

(Quiz)

Lehrende sollen die Möglichkeit haben, Tests für ihre Lehrveranstaltungen zu erstellen, die aus einer beliebigen Zahl von Quizfragen bestehen. Quizfragen können manuell erstellt werden oder durch Einlesen einer xml-Datei integriert werden. Dabei weisen Quizfragen das Multiple-Choice-Format auf und der Lehrende definiert die korrekten Antworten. Nach der Erstellung können Studierende die Tests beliebig oft bearbeiten und Lösungen einreichen. Das System stellt ihnen dann automatisch ein Feedback zu ihren Antworten bereit, aus dem hervorgeht, welche Fragen korrekt beantwortet wurden und welche nicht. Allerdings soll keine Musterlösung angezeigt werden, wenn eine Antwort falsch war. Lehrende können die Statistik über die durchgeführten Tests einsehen. Dabei können sie die Beteiligung der Studierenden im Verhältnis zur Gesamtteilnehmerzahl einer Lehrveranstaltung, die Bestehensquote(mindestens einmal mindestens 50% der Quizfragen korrekt beantwortet), die Anzahl der Versuche der einzelnen Studierenden und die Anzahl der korrekten Antworten zu einer einzelnen Frage beobachten.

(Kalenderfunktionen)

Jeder Nutzer soll über einen persönlichen Kalender verfügen. Der Lehrende kann für seine Lehrveranstaltungen Termine vorgeben (bspw. Einreichungsfristen für Hausarbeiten), die in seinen eigenen Kalender sowie in die Kalender der Teilnehmer eingetragen werden. Dabei soll er Reminder für die Termine erstellen können, durch die die Studierende vor dem Erreichen eines solchen Termins gewarnt werden. Der Lehrende soll einstellen können, in welchem Abstand vor dem Termin der Reminder erscheint und in welcher Form (Pop-up-Fenster/E-Mail) der Studierende diesen erhält. Um diese Funktion effektiv testen zu können, soll das Datum im Programm verändert werden können.

(Zwei-Faktor-Authentisierung)

Der Login-Vorgang soll durch eine Zwei-Faktor-Authentisierung erweitert werden, bei der, neben dem Passwort, ein Sicherheitscode eingegeben werden muss, der per E-Mail versendet wird.

Zyklus 3

(Projektgruppe)

Neben den beschriebenen Funktionen sollen die Mitglieder einer Projektgruppe nun Lernkarten erstellen und diese mit den übrigen Mitgliedern teilen können.

(Bewertung von Lehrveranstaltungen)

Studierende sollen die Möglichkeit haben, die von ihnen belegten Lehrveranstaltungen bewerten zu können. Zu diesem Zweck erstellt der Lehrende einen weiteren Test mit Multiple-Choice Fragen für seine Lehrveranstaltungen. Studierende können diese beantworten, sofern diese mindestens die Hälfte der vorherigen Tests bearbeitet und somit an der Veranstaltung teilgenommen haben. Der Lehrende soll sich auch hierzu eine anonymisierte Statistik ansehen können. Dabei soll er sehen können, wie häufig eine bestimmte Antwort zu einer Frage ausgewählt wurde. Außerdem soll er auswählen können, ob er die Bewertungsstatistik aller Teilnehmer einsehen will oder spezifischere Bewertungsstatistiken entweder nur über Studierende, die bestanden haben oder nur über diejenigen, die durchgefallen sind.

(Themenangebote für studentische Arbeiten)

Lehrende sollen Themenangebote für Abschlussarbeiten bereitstellen können. Ein Thema ist durch einen Titel, eine kurze Beschreibung und eine Literaturliste gegeben. Die Literaturliste kann der Lehrende durch das Einlesen einer bibtex-Datei erstellen, die die Daten über die Veröffentlichungen enthält. Die Themenangebote sollen im Nutzerprofil des Lehrenden angezeigt werden und nur für diejenigen Studierenden einsehbar sein, die einen Kurs des jeweiligen Lehrenden belegt haben.

(Mitteilung der Ergebnisse)

Studierende sollen automatisch eine E-Mail erhalten, wenn Sie eine Lehrveranstaltung bestanden haben oder durchgefallen sind. Eine Lehrveranstaltung gilt als bestanden, wenn bei Ablauf des Semesters mindestens die Hälfte der Tests einmal erfolgreich absolviert wurde. Eine Lehrveranstaltung gilt als Nicht-bestanden, wenn weniger als die Hälfte der Tests erfolgreich absolviert wurde.

Zyklus I

Definition der Verantwortlichen für die Artefakte der Spezifikation

Id	Artefakt	Art des Artefakts	Verantwortlicher	Status
1.	Registrierung der Nutzer		Noemi	
1.1	User Stories	User Story		Erl.
1.1.1	User Stories - Nutzer			Erl.
1.1.1.1	Datenangabe bei Profilerstellung - Nutzer			Erl.
1.1.1.2	Passwortvergabe bei Profilerstellung			Erl.
1.1.2	User Stories - Studierende			Erl.
1.1.2.1	Angabe von Studienfach im Profil			Erl.
1.1.2.2	Automatische Zuweisung von Matrikelnummer			Erl.
1.1.2.3	Bearbeitung der Profilangaben			Erl.
1.1.3	User Story - Lehrende			Erl.
1.1.3.1	Angabe von Lehrstuhl und Forschungsgebiet im Profil			Erl.
1.1.3.2	Ansicht der Profildaten für andere Nutzer			Erl.
1.2	Papierprototypen	Papierprototyp		Erl.
1.2.1	StartWindow (Registrierung/Login)			Erl.
1.2.2	CreateProfile Ansicht			Erl.
1.2.3	RoleWindow Ansicht			Erl.
1.2.4	EditProfile Ansicht			Erl.
1.2.5	Password Ansicht			Erl.
1.3	MSCs	MSC		Erl.
1.3.1	Registrierungsprozess – User Story 1.1.1			Erl.
1.3.2	Passwortvergabe - User Story 1.1.2			Erl.

1.3.3	Überprüfung der Rolle des Nutzers - User Stories 1.2.1, 1.2.2 und 1.3.1			Erl.
1.3.4	Bearbeitung der Angaben im Profil – User Story 1.2.3			Erl.
2.	Verwaltung von Lehrveranstaltungen		Kuri	
2.1	User Story	User Story		Erl.
2.1.1	User Story 2.1 (Nutzer)			Erl.
2.1.2	User Story 2.2 (Lehrende)			Erl.
2.2	MSC	MSC		Erl.
2.2.1	Eine Lehrveranstaltung erstellen – bMSC 2.2.1			Erl.
2.2.1	Mehrere Lehrveranstaltungen erstellen – bMSC 2.2.2			Erl.
2.2.2	Material Bereitstellung – bMSC 2.2.3			Erl.
2.2.3	Nach Veranstaltung suchen – bMSC 2.2.4			Erl.
2.3	Papierprototyp	Papierprototyp		Erl.
2.3.1	Übersicht-Lehrveranstaltung – Fenster für Nutzer			Erl.
2.3.2	Übersicht-Lehrveranstaltung - Fenster für Lehrende			Erl.
2.3.3	Übersicht-Lehrveranstaltung - Fenster für eingetragene Studenten			Erl.
3.	Zuordnung von Nutzern zu Lehrveranstaltungen		Vera	
3.1	User Story	User Story		Erl.
3.1.1	User Story 3.1.1 (Nutzer meldet sich an)			Erl.
3.1.2	User Story 3.1.2 (Lehrende meldet Nutzer an)			Erl.
3.1.3	User Story 3.1.3 (Nutzereinsicht Teilnehmerliste)			Erl.
3.1.4	User Story 3.1.4 (Nutzereinsicht alle Veranstaltungen)			Erl.
3.2	Übersicht-Lehrveranstaltungen-Fenster	Papierprototyp		Erl.
3.2.1	Manuelle Eintragung (Selbst)	MSC		Erl.
3.2.2	Manuelle Eintragung (Lehrperson)	MSC		Erl.
3.2.3	Aufruf Teilnehmerliste	MSC		Erl.
3.3	Übersicht-Teilnehmerliste-Fenster	Papierprototyp		Erl.
4.	Nutzerprofile			
4.1	User Story	User Story	Thorben	Erl.

4.1.1	User Story 4.1 (Nutzer)			Erl.
4.1.2	User Story 4.2 (Lehrende)			Erl.
4.1.3	User Story 4.3 (Studierende)			Erl.
4.2	Profil-Übersicht	Papierprotoyp	Hana	Erl.
4.2.1	Profil- Übersicht-begrenzt			Erl.
4.2.2	Profil- Übersicht-vollständig			Erl.
4.3	Profil-Aufruf	MSC	Hana	Erl.
5.	Login-Vorgang und Startfenster		Fred	
5.1	User Story 5.1 (Nutzer)	User Story		Erl.
5.2	Login-Fenster	Papierprotoyp		Erl.
5.2.1	Login-Vorgang	MSC		Erl.
5.3	Start-Übersicht-Fenster	Papierprotoyp		Erl.
5.3.1	Profil anzeigen	MSC		Erl.
5.3.2	Veranstaltung suchen	MSC		Erl.

User-Stories

User Story-ID	1.1.1
User Story-Beschreibung	Als Nutzer möchte ich bei der Erstellung meines Profils meinen Vor- und Nachnamen, E-Mailadresse und ein Profilbild festlegen können, damit andere Nutzer diese Informationen einsehen können.
Geschätzter Realisierungsaufwand	low
Priorität	high
Autor(en)	Noemi Kallweit
Abhängigkeiten zu anderen User Stories	Keine Abhängigkeiten

User Story-ID	1.1.2
User Story-Beschreibung	Als Nutzer möchte ich bei der Erstellung meines Profils ein Passwort festlegen können, sodass man sich über mein Profil nur mit dem zugehörigen Passwort einloggen kann.
Geschätzter Realisierungsaufwand	middle
Priorität	high
Autor(en)	Noemi Kallweit
Abhängigkeiten zu anderen User Stories	Keine Abhängigkeiten

User Story-ID	1.2.1
User Story-Beschreibung	Als Studierende möchte ich bei der Erstellung meines Profils mein Studienfach angeben können, damit andere diese Information einsehen können, wenn sie mein Profil öffnen.
Geschätzter Realisierungsaufwand	low
Priorität	middle
Autor(en)	Noemi Kallweit
Abhängigkeiten zu anderen User Stories	4.1.3

User Story-ID	1.2.2
User Story-Beschreibung	Als Studierende möchte ich, dass mir meine Matrikelnummer beim Erstellen meines Profils automatisch vom System zugewiesen wird, um sicherzugehen, dass meine Matrikelnummer einzigartig ist.
Geschätzter Realisierungsaufwand	middle
Priorität	high
Autor(en)	Noemi Kallweit
Abhängigkeiten zu anderen User Stories	Keine Abhängigkeiten

User Story-ID	1.2.3
User Story-Beschreibung	Als Studierende möchte ich alle Angaben in meinem Profil mit Ausnahme von Vor- und Nachname, Mailadresse und Matrikelnummer nachträglich bearbeiten können, sodass ich meine Angaben jederzeit aktualisieren kann.
Geschätzter Realisierungsaufwand	low
Priorität	high
Autor(en)	Noemi Kallweit
Abhängigkeiten zu anderen User Stories	Keine Abhängigkeiten

User Story-ID	1.3.1
User Story-Beschreibung	Als Lehrende möchte ich in meinem Profil meinen Lehrstuhl und mein Forschungsgebiet angeben können, damit andere diese Information einsehen können, wenn sie mein Profil öffnen.
Geschätzter Realisierungsaufwand	low
Priorität	middle
Autor(en)	Noemi Kallweit
Abhängigkeiten zu anderen User Stories	Keine Abhängigkeiten

User Story-ID	1.3.2
User Story-Beschreibung	Als Lehrende möchte ich nicht, dass Studierende ihren Vor- und Nachnamen, Emailadresse oder Matrikelnummer in ihrem Profil ändern können, weil diese Änderungen nur von der Universität selber vorgenommen können werden sollen.
Geschätzter Realisierungsaufwand	low
Priorität	middle
Autor(en)	Noemi Kallweit
Abhängigkeiten zu anderen User Stories	Keine Abhängigkeiten

User Story-ID	2.1.1
User Story-Beschreibung	Als Nutzer möchte ich die Lehrveranstaltungen eindeutig identifizieren können, um sie voneinander zu unterscheiden.
Geschätzter Realisierungsaufwand	low
Priorität	hoch
Autor(en)	Kurinchinilavan Thangewaran
Abhängigkeiten zu anderen User Stories	

User Story-ID	2.1.2
User Story-Beschreibung	Als Nutzer möchte ich für jede Lehrveranstaltung, an der ich beteiligt bin, eine Übersicht haben, um Informationen der Veranstaltung einsehen zu können.
Geschätzter Realisierungsaufwand	low
Priorität	hoch
Autor(en)	Kurinchinilavan Thangewaran
Abhängigkeiten zu anderen User Stories	

User Story-ID	2.2.1
User Story-Beschreibung	Als Lehrender möchte ich <i>einzelne</i> Lehrveranstaltungen erstellen können, um diese im SEP einsehen zu können.
Geschätzter Realisierungsaufwand	low
Priorität	hoch
Autor(en)	Kurinchinilavan Thangewaran
Abhängigkeiten zu anderen User Stories	

User Story-ID	2.2.2
User Story-Beschreibung	Als Lehrender möchte ich <i>mehrere Lehrveranstaltungen</i> erstellen können, um diese im SEP einsehen zu können.
Geschätzter Realisierungsaufwand	low
Priorität	hoch
Autor(en)	Kurinchinilavan Thangewaran
Abhängigkeiten zu anderen User Stories	2.1.1.1

User Story-ID	2.2.3
User Story-Beschreibung	Als Lehrender möchte ich in der Übersicht der Veranstaltung Lehrmaterial zur Verfügung stellen und diese verwalten, damit die Teilnehmer darauf zugreifen können.
Geschätzter Realisierungsaufwand	middle
Priorität	hoch
Autor(en)	Kurinchinilavan Thangewaran
Abhängigkeiten zu anderen User Stories	2.1.1.1, 2.1.1.2

User Story-ID	3.1.1
User Story-Beschreibung	Als Nutzer möchte ich mich an Lehrveranstaltungen anmelden, um an diesen teilnehmen zu können und um auf die Materialien zugreifen zu können
Geschätzter Realisierungsaufwand	low
Priorität	mittel
Autor(en)	Vera Mitschulat
Abhängigkeiten zu anderen User Stories	Keine Abhängigkeiten

User Story-ID	3.1.2
User Story-Beschreibung	Als Lehrender möchte ich andere Nutzer manuell an meiner Lehrveranstaltung anmelden können, um nur eine bestimmte ausgewählte Gruppe zu meiner Lehrveranstaltung zuzulassen.
Geschätzter Realisierungsaufwand	low
Priorität	niedrig
Autor(en)	Vera Mitschulat
Abhängigkeiten zu anderen User Stories	3.1.3

User Story-ID	3.1.3
User Story-Beschreibung	Als Nutzer möchte ich eine Teilnehmerliste für belegte Lehrveranstaltung einsehen können, damit ich auf die Kontaktdaten von anderen Teilnehmern dieses Kurses zugreifen kann.
Geschätzter Realisierungsaufwand	low
Priorität	Mittel
Autor(en)	Vera Mitschulat
Abhängigkeiten zu anderen User Stories	

User Story-ID	3.1.4
User Story-Beschreibung	Als Nutzer möchte ich eine Liste von allen Lehrveranstaltungen einsehen können, um die von mir gesuchte Lehrveranstaltung zu finden
Geschätzter Realisierungsaufwand	low
Priorität	Mittel
Autor(en)	Vera Mitschulat
Abhängigkeiten zu anderen User Stories	Keine Abhängigkeiten

User Story-ID	4.1.1
User Story-Beschreibung	Als Nutzer möchte Ich, dass mein Eintrag in der Teilnehmerliste visuell hervorgehoben wird, damit ich leicht auf mein Profil komme.
Geschätzter Realisierungsaufwand	low
Priorität	mittel
Autor(en)	Hana Habibi
Abhängigkeiten zu anderen User Stories	4.1.2, 4.1.3

User Story-ID	4.1.2
User Story-Beschreibung	Als Studierende möchten wir nicht, dass unsere Matrikelnummer und private Adresse von anderen Studierenden eingesehen werden können, um unsere Privatsphäre und unsere Daten zu schützen.
Geschätzter Realisierungsaufwand	low
Priorität	hoch
Autor(en)	Thorben Kronewald-Lamken
Abhängigkeiten zu anderen User Stories	4.1.1

User Story-ID	4.1.3
User Story-Beschreibung	Als Lehrende möchten wir, dass wir die Informationen aller Studenten einsehen können, um diese einwandfrei zuordnen zu können.
Geschätzter Realisierungsaufwand	low
Priorität	mittel
Autor(en)	Thorben Kronewald-Lamken
Abhängigkeiten zu anderen User Stories	4.1.1

User Story-ID	5.1.1
User Story-Beschreibung	Als Nutzer möchte ich entweder meine Martrikelnummer oder meine E-Mailadresse für den Login nutzen, damit ich meine Matrikelnummer nicht auswendig kennen muss.
Geschätzter Realisierungsaufwand	low
Priorität	hoch
Autor(en)	Frederik Gnass
Abhängigkeiten zu anderen User Stories	

User Story-ID	5.1.2
User Story-Beschreibung	Als Nutzer möchte ich zusätzlich mein Passwort für den Login angeben können, um meinen Account zu schützen.
Geschätzter Realisierungsaufwand	low
Priorität	hoch
Autor(en)	Frederik Gnass
Abhängigkeiten zu anderen User Stories	5.1.1

User Story-ID	5.1.3
User Story-Beschreibung	Als Nutzer möchte ich bei einem erfolgreichen Login Zugriff auf mein Profil haben, damit ich meine Daten einsehen und bearbeiten kann.
Geschätzter Realisierungsaufwand	middle
Priorität	hoch
Autor(en)	Frederik Gnass
Abhängigkeiten zu anderen User Stories	

User Story-ID	5.1.4
User Story-Beschreibung	Als Nutzer möchte ich bei einem erfolgreichen Login Zugriff die Lehrveranstaltungen sehen, in die ich eingeschrieben bin, damit ich die Kursmaterialien einsehen kann.
Geschätzter Realisierungsaufwand	middle
Priorität	hoch
Autor(en)	Frederik Gnass
Abhängigkeiten zu anderen User Stories	

User Story-ID	5.1.5
User Story-Beschreibung	Als Nutzer möchte ich bei einem erfolgreichen Login Zugriff weitere Lehrveranstaltungen suchen können, um diesen beizutreten.
Geschätzter Realisierungsaufwand	middle
Priorität	hoch
Autor(en)	Frederik Gnass
Abhängigkeiten zu anderen User Stories	5.1.2.1, 5.1.2.2

User Story-ID	5.1.6
User Story-Beschreibung	Als Nutzer möchte ich bei einem nicht erfolgreichen Login Zugriff benachrichtigt werden, um andere Zugangsdaten einzutragen.
Geschätzter Realisierungsaufwand	middle
Priorität	hoch
Autor(en)	Frederik Gnass
Abhängigkeiten zu anderen User Stories	

Papierprototypen

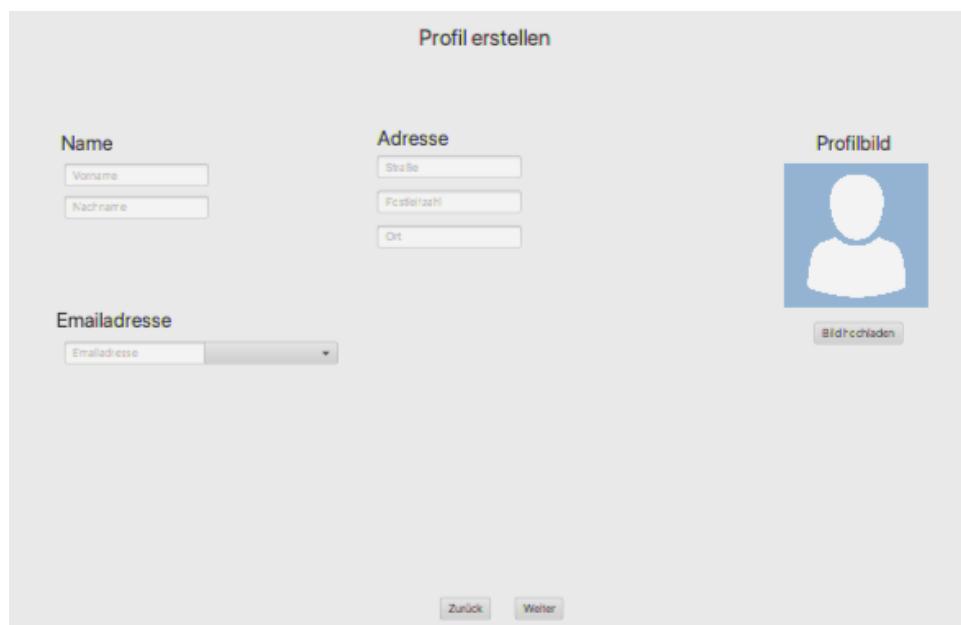
Ansicht StartWindow

Dieses Fenster wird angezeigt, wenn der Nutzer das Programm startet. Über den Button "Login" wird der Nutzer zur Hauptseite weitergeleitet. Wenn man auf "Registrieren" klickt, kommt man zum Fenster "CreateProfile".



CreateProfile – Ansicht

Dieses Fenster wird über Button "Registrierung" im Start Fenster erreicht. Hier kann ein Nutzer Vor- und Nachname, private Adresse und Emailadresse anlegen und ein Profilbild festlegen. Über den "Weiter" Button gelangt man zum RoleWindow.



RoleWindow – Ansicht

Dieses Fenster folgt auf das Registrierungsfenster. Es hat ein unterschiedliches Aussehen, je nach Rolle des Nutzers. Studierende bekommen nur ein Textfeld angezeigt mit der Überschrift "Studienfach". In diesem Fenster kann der Nutzer seine Profilangaben um Forschungsgebiet und Lehrstuhl bzw. Studienfach ergänzen.

The screenshot shows a light gray window titled 'RoleWindow'. At the top center is the text 'Forschungsgebiet' in red. Below it is a large, empty white input field. In the center of the window is the text 'Lehrstuhl' in red. Below it is another large, empty white input field. At the bottom left is a gray button labeled 'Zurück' (Back). At the bottom right is a blue button labeled 'Weiter' (Forward).

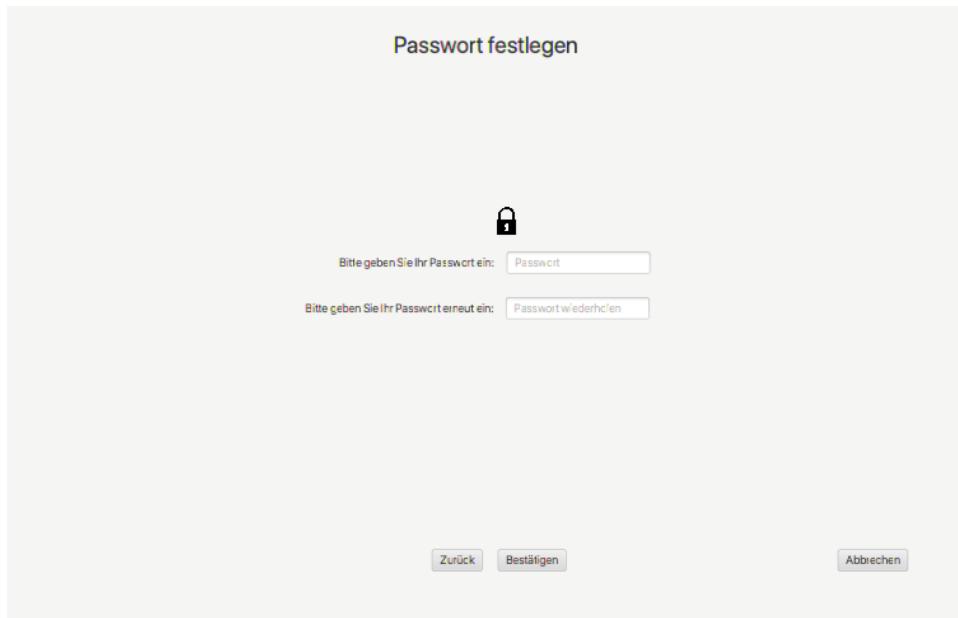
EditProfile – Ansicht

Dieses Fenster erreicht ein Nutzer über den "Bearbeiten"-Button im Profil Tab. Hier können die eigenen Angaben bearbeitet werden. Studierenden wird die Matrikelnummer und das Eingabefeld für das Studienfach angezeigt. Lehrenden werden dementsprechend die Eingabefelder für den Lehrstuhl (Doppelbelegung mit Studienfach) und das Forschungsgebiet angezeigt. Über "Ändern" gelangt man zum "Password"-Fenster und über "Schließen" zurück zum Profil Tab.

The screenshot shows a window titled 'Profil' in blue at the top center. The interface is divided into several sections: 'Max Mustermann' (Name), 'Matrikelnummer 1234567' (Matriculation Number), and a placeholder 'Profilbild' (Profile Picture) with a 'Bearbeiten' (Edit) button. Below these are sections for 'Adresse' (Address) with fields for street ('Musterstraße, 123') and zip code ('12345'), and 'Studienfach' (Study Field) with a placeholder 'Studienfach'. There is also a 'Forschungsgebiet' (Research Area) section with a placeholder 'Forschungsgebiet' and a 'Bearbeiten' button. At the bottom left is an 'Emailadresse' (Email Address) section with 'max.mustermann@stud.uni.de' and a 'Passwort ändern' (Change Password) button with an 'Ändern' (Change) button. At the bottom right is a 'Schließen' (Close) button.

Password - Ansicht

Dieses Fenster wird angezeigt, wenn ein Nutzer ein Passwort bei der Registrierung anlegen oder später ändern möchte. Das Fenster wird über den “Ändern” Button im Bearbeitungsfenster erreicht oder über den Button “Weiter” im RoleWindow. Die Buttons “Zurück” und “Abbrechen” sind nicht gleichzeitig sichtbar. Der Button “Abbrechen” ist nur sichtbar und benutzbar, sobald der User das Passwort ändern möchte. Der Button “Zurück” ist nur sichtbar, wenn man einen neuen Account anlegt.



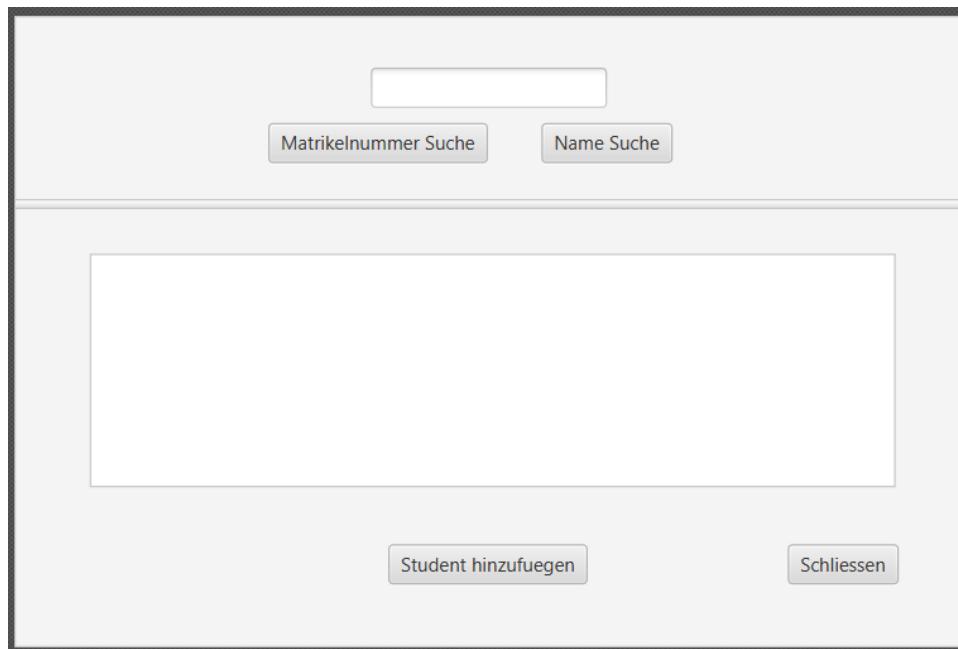
Übersicht-Lehrveranstaltung - Ansicht

Diese Ansicht wird Nutzern angezeigt, die als Lehrende dieses Kurses eingetragen sind. Über den “Add Student” Button können Studenten dem Kurs hinzugefügt werden und über den Button “Teilnehmerliste” kann eine Liste mit allen Teilnehmern dieses Kurses aufgerufen werden. Die beiden unteren Buttons dienen dazu Lehrmaterialien hochzuladen bzw. zu löschen. Diese Materialien werden dann in der Liste angezeigt. Die Button “Hinzufügen” und “Löschen” werden nur Lehrpersonen angezeigt.

A screenshot of a user interface for managing course participants. At the top, there is a navigation bar with tabs: "Profil", "Veranstaltungen", and "Das Spätmittelalter" (selected). Below the tabs, the title "Das Spätmittelalter" is displayed. On the left, there is a button "Add Student". On the right, there is a button "Teilnehmerliste". The main area shows a list titled "021Modellierung.pdf" with several empty rows below it. At the bottom of this list area are two buttons: "Hinzufügen" (Add) and "Loeschen" (Delete). A vertical scroll bar is visible on the right side of the list area.

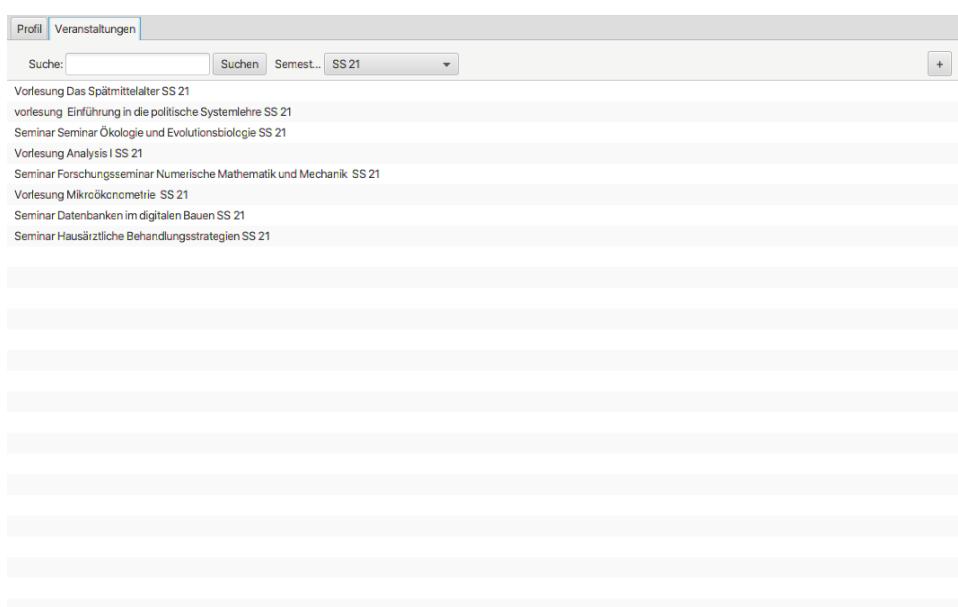
Übersicht-Student suchen und hinzufügen - Ansicht nur für Lehrende

Dieses Fenster gibt es nur für Lehrpersonen. Es wird über den “AddStudent” Button in der CourseView erreicht. Hier können Lehrende Studenten nach ihrem Namen oder Matrikelnummer suchen und einem Kurs hinzufügen.



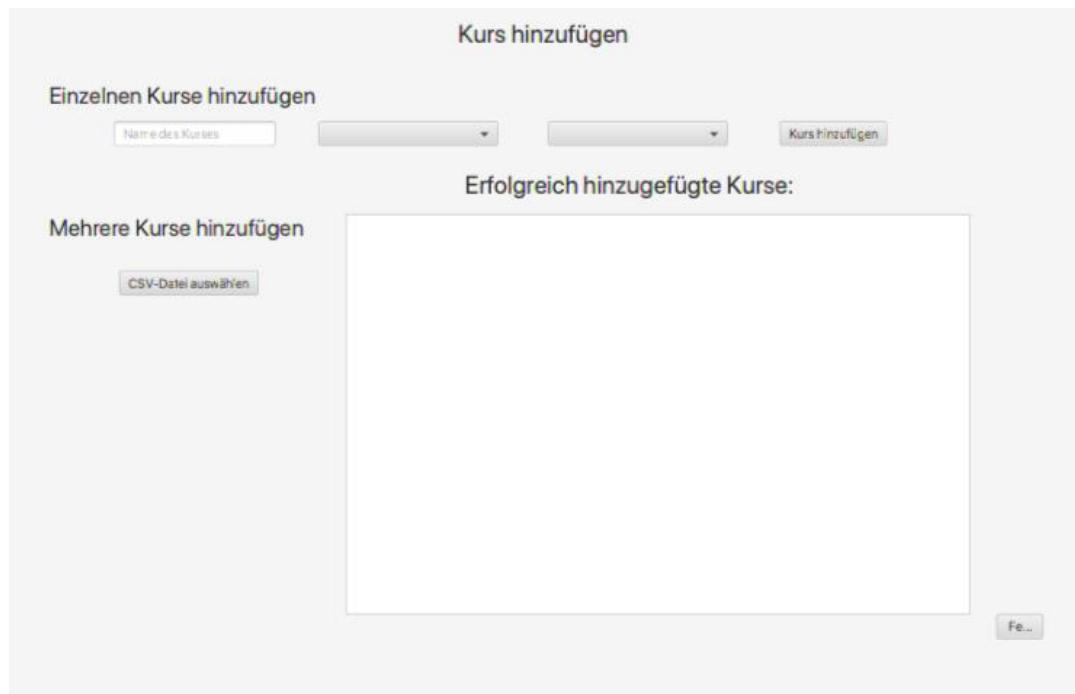
CourseViewAll – Ansicht

Eine Liste aller Lehrveranstaltungen. Ein Klick auf die Bezeichnung einer Lehrveranstaltung führt zu den näheren Details der Lehrveranstaltung und der Möglichkeit des Beitritts zur Lehrveranstaltung. Der “+” Button in der oberen rechten Ecke wird nur Lehrpersonen angezeigt. Darüber können neue Kurse erstellt werden. Über die ChoiceBox können Kurse nach Semestern gefiltert werden. Über die Suchleiste kann nach Veranstaltungen in dem ausgewählten Semester gesucht werden.



AddCourse – Ansicht nur für Lehrende

Dieses Fenster wird angezeigt, wenn in der CourseViewAll auf den “+” Button geklickt wird. Hierüber können einzelne oder mehrere Kurse angelegt werden.

**ParticipantsList – Ansicht**

Eine Liste aller aktuellen Teilnehmer einer Lehrveranstaltung. Die Liste ist in Lehrende und Teilnehmer unterteilt. Der eigene Name wird grün hervorgehoben. Mit einem Klick auf einen Namen in der Liste gelangt man zum zugehörigen Profil.

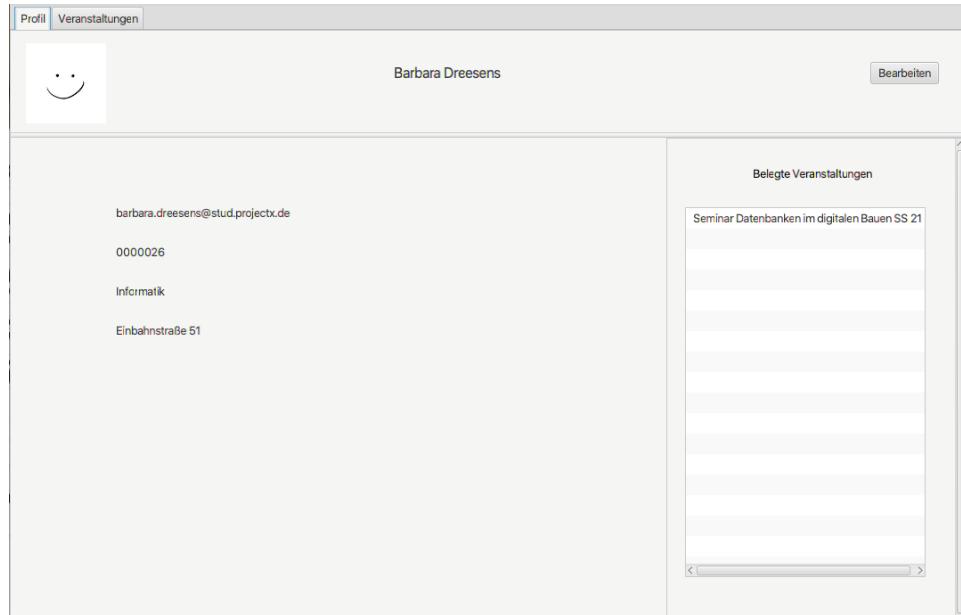
The screenshot shows a list of participants divided into 'Lehrende' (Teaching staff) and 'Teilnehmer' (Students). The 'Teilnehmer' section is highlighted with a green background for the row containing 'Barbara Dreesens'. Other names listed include Olga Meyer, David Hollenberg, and Barbara Dreesens. The top navigation bar includes tabs for 'Profil', 'Veranstaltungen', 'Datenbanken im digitalen Bauen', and 'Teilnehmerliste'.

Profile – Ansicht eigenes Profil

Dieses Fenster sehen Studierende, wenn der Login-Vorgang erfolgreich war. Über den “Bearbeiten” Button gelangt man zum EditProfile Fenster.

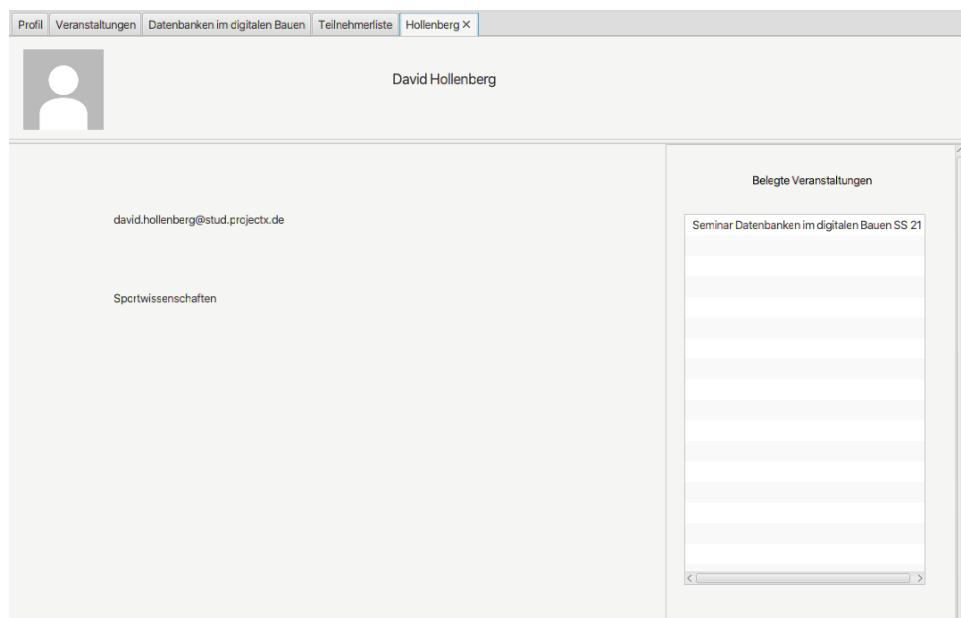
Beim Klick als Student auf sein eigenes Profil in der Teilnehmerliste, wird dieses Fenster als Profilansicht geöffnet.

Beim Klick als Lehrender auf sein eigenes Profil in der Teilnehmerliste, wird dieses Fenster als Profilansicht geöffnet. Anstatt der Matrikelnummer und Studienfach, werden der Lehrstuhl und das Forschungsgebiet gezeigt.



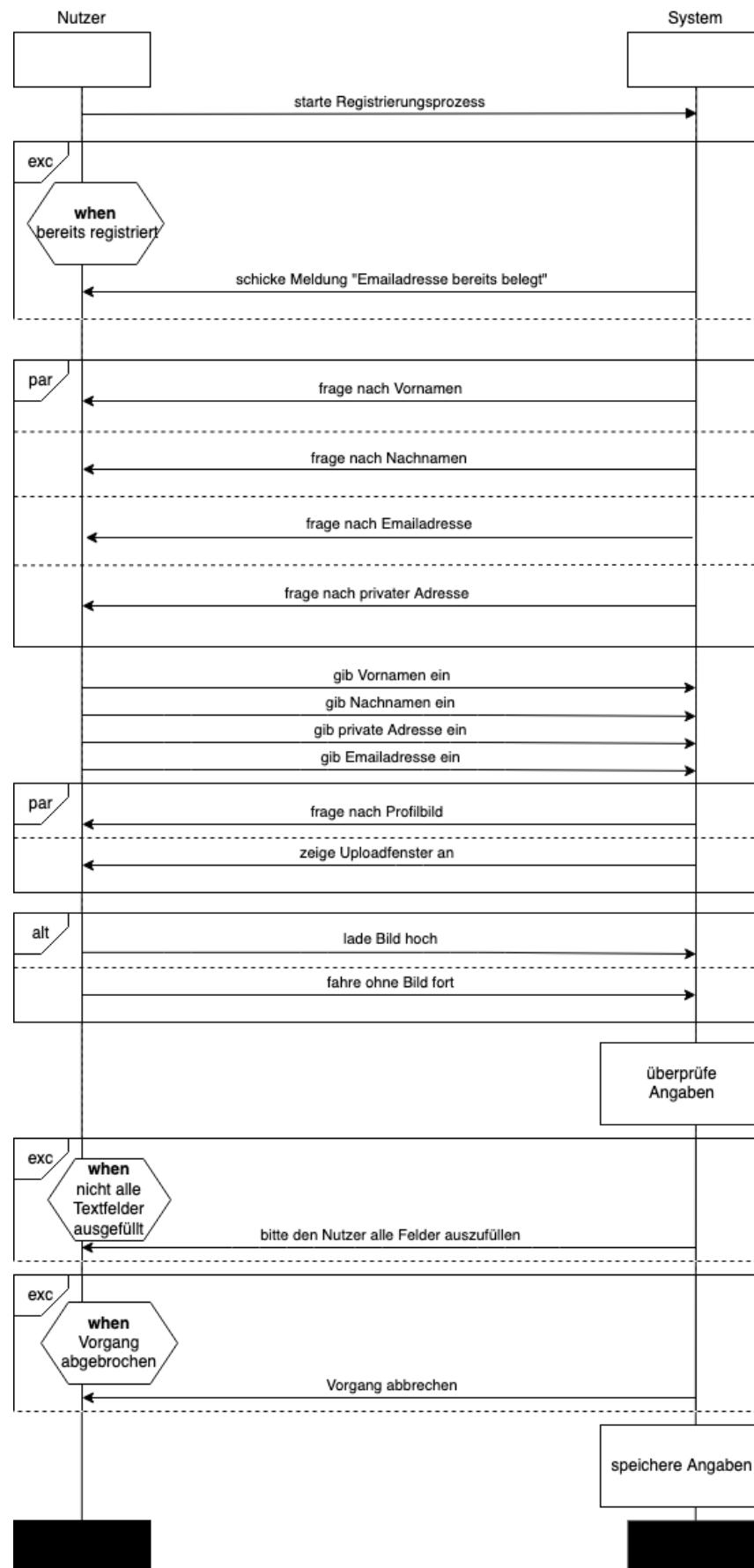
Profile – Ansicht fremdes Profil

Beim Klick als Studierender auf das Profil eines Studierenden in der Teilnehmerliste, wird dieses Fenster als Profilansicht geöffnet. Es wird nur das Studienfach gezeigt. Lehrende können alle Angaben eines Studenten sehen. Studenten können umgekehrt alle Angaben von Lehrenden sehen, mit Ausnahme der privaten Adresse.



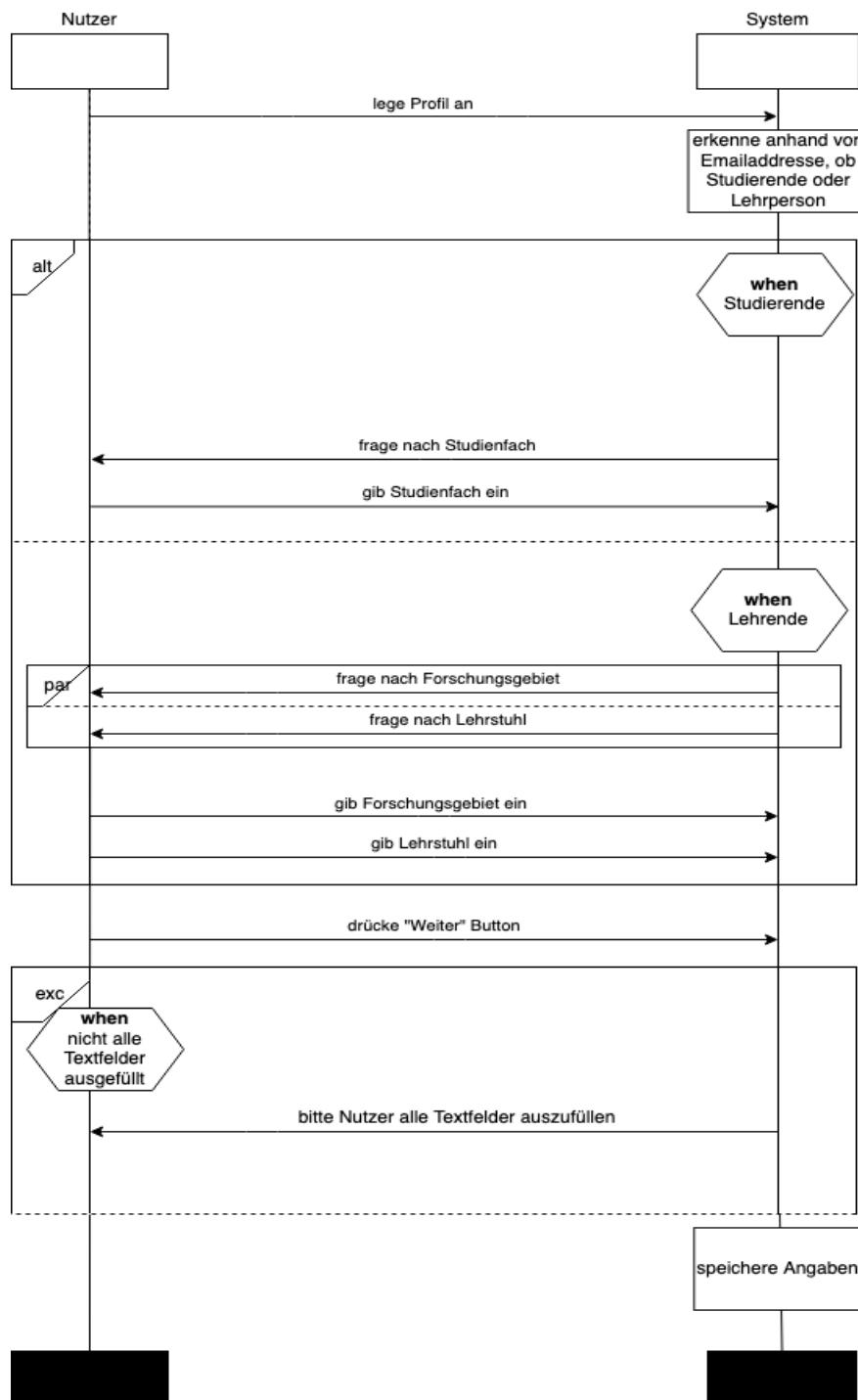
Szenarien (MSCs)

bMSC 1.1.1



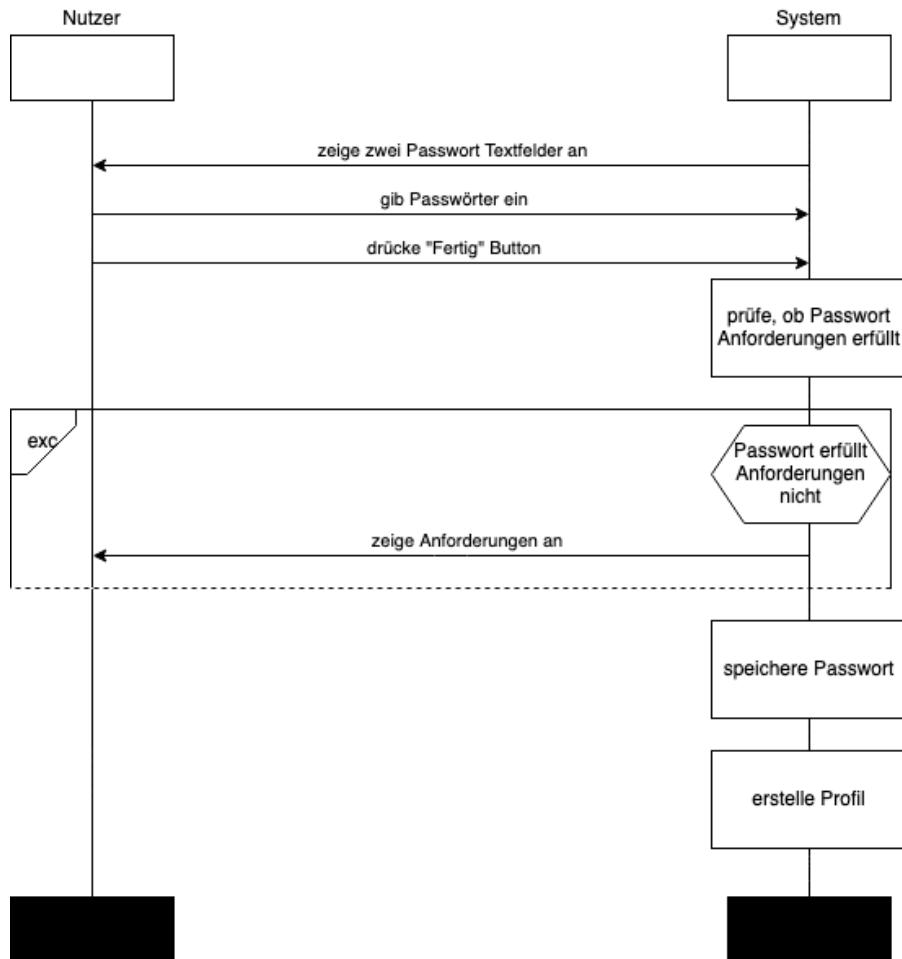
Der Registrierungsprozess wird gestartet, indem der Nutzer auf den "Registrieren" Button im Startfenster klickt. Danach bekommt er ein Fenster angezeigt, in dem Eingabefelder für Name, Emailadresse und private Adresse angezeigt werden. Außerdem wird ein Upload-Fenster für ein Profilbild angezeigt. Nachdem der Nutzer auf "Weiter" geklickt hat, werden die Angaben überprüft u. A., ob alle Felder mit Ausnahme des Profilbildes ausgefüllt sind. Wenn dies nicht der Fall ist, wird der Nutzer solange darauf hingewiesen, bis er alle Felder ausgefüllt hat. Falls der Nutzer den Registrierungsprozess abbricht, indem er z. B. die Seite schließt, wird der Vorgang beendet. Sonst werden die Angaben gespeichert und der Nutzer wird zum nächsten Fenster weitergeleitet.

bMSC 1.2.1, 1.2.2 und 1.3.1

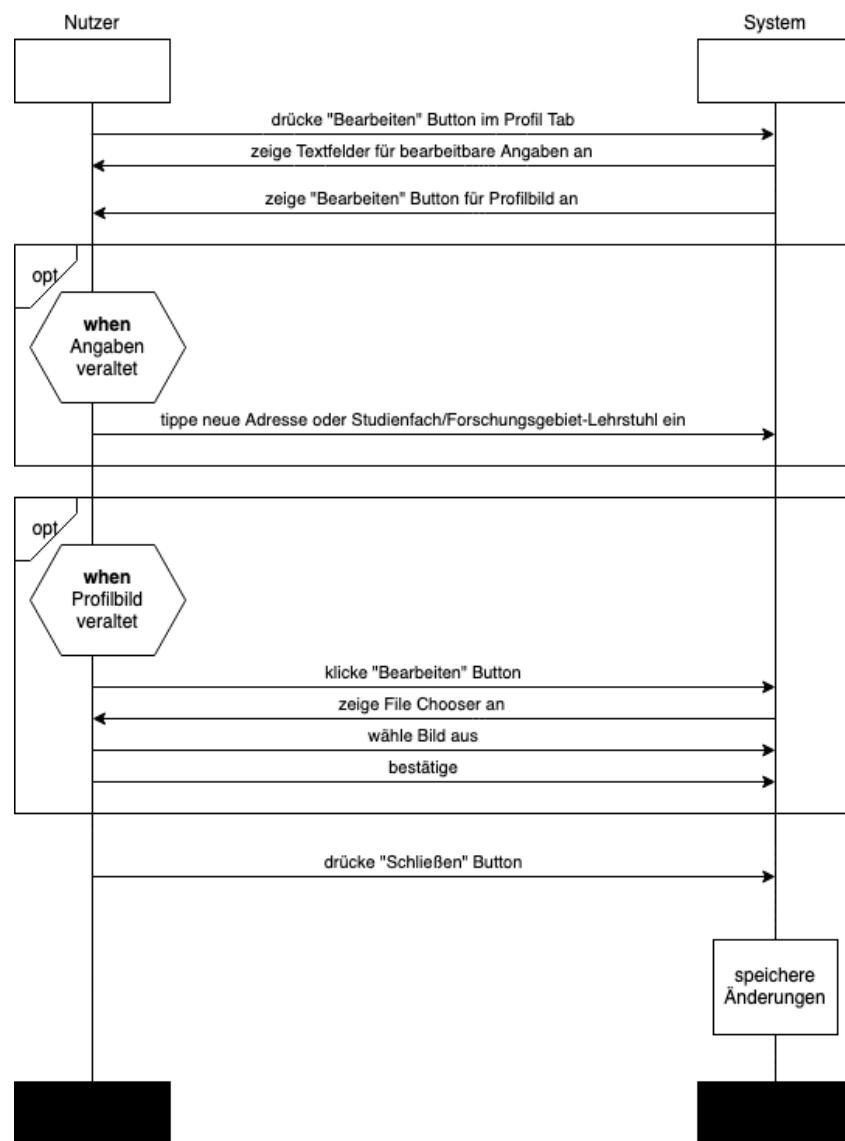


Nach den ersten Registrierungsschritten überprüft das System anhand der E-Mail-Adresse, ob es sich um eine Lehrperson oder einen Studierenden handelt. Die E-Mail-Adresse für Studierende ist nach dem Schema „vorname.nachname@stud.projectx.de“ aufgebaut. Die für Lehrende „vorname.nachname@projectx.de“. Anhand der Endung der E-Mail-Adresse wird den Nutzern ihre Rolle zugewiesen. Wenn es sich um einen Studierenden handelt, wird ein Eingabefeld angezeigt, in dem er sein Studienfach angeben kann. Eine Lehrperson bekommt Eingabefelder für den Lehrstuhl und das Forschungsgebiet angezeigt. Wenn der Nutzer seine Angaben gemacht hat, werden diese gespeichert.

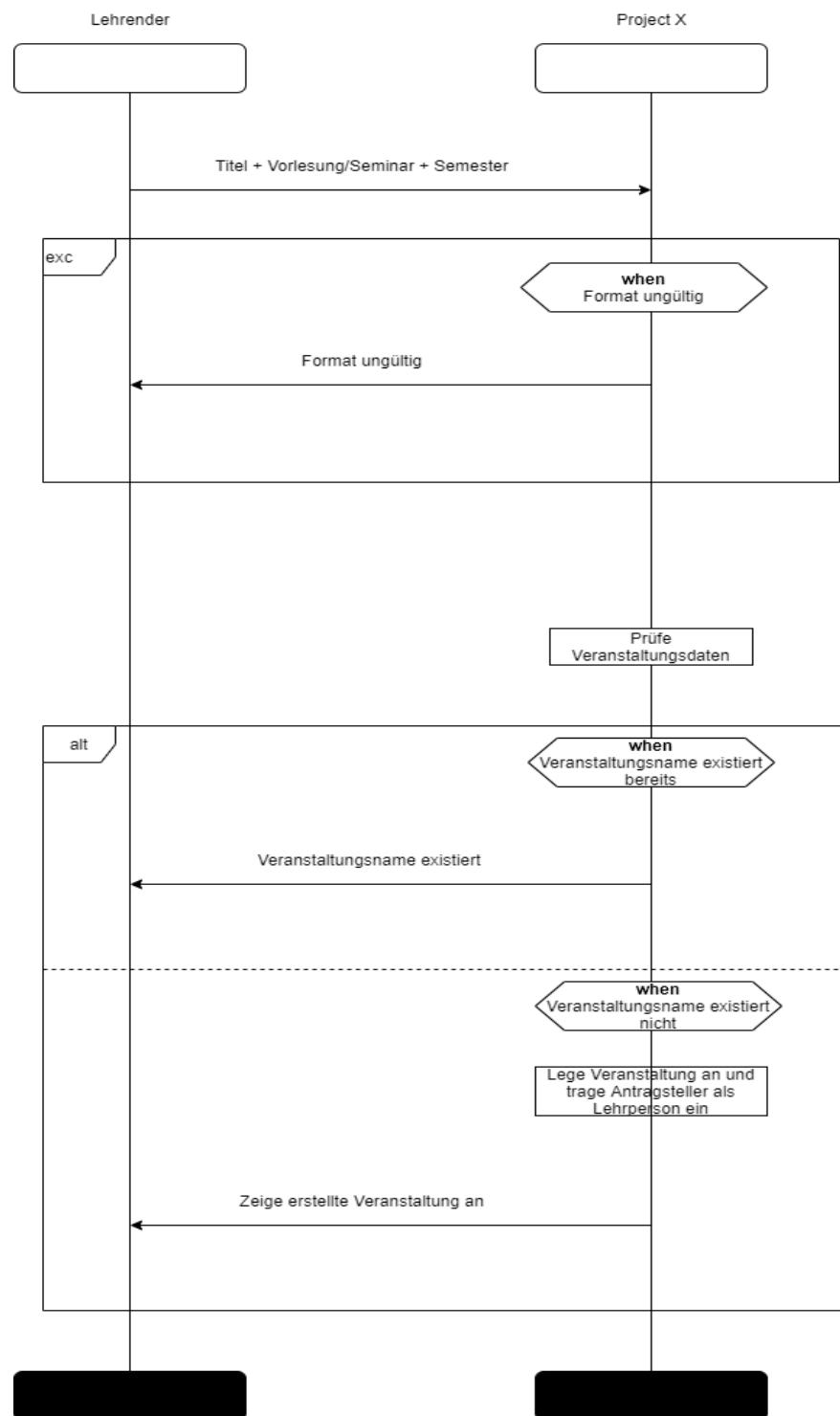
bMSC 1.1.2



Nach den ersten Registrierungsschritten wird der Nutzer dazu aufgefordert ein Passwort festzulegen. Dazu wird ihm zwei Eingabefelder gezeigt. Nachdem der Nutzer sich ein Passwort vergeben hat, überprüft das System, ob das Passwort die Anforderungen erfüllt. Falls nicht wird der Nutzer solange gebeten ein anderes Passwort zu wählen, bis es die Anforderungen erfüllt. Wenn dies der Fall ist, wird das Passwort gespeichert und das Profil wird angelegt.

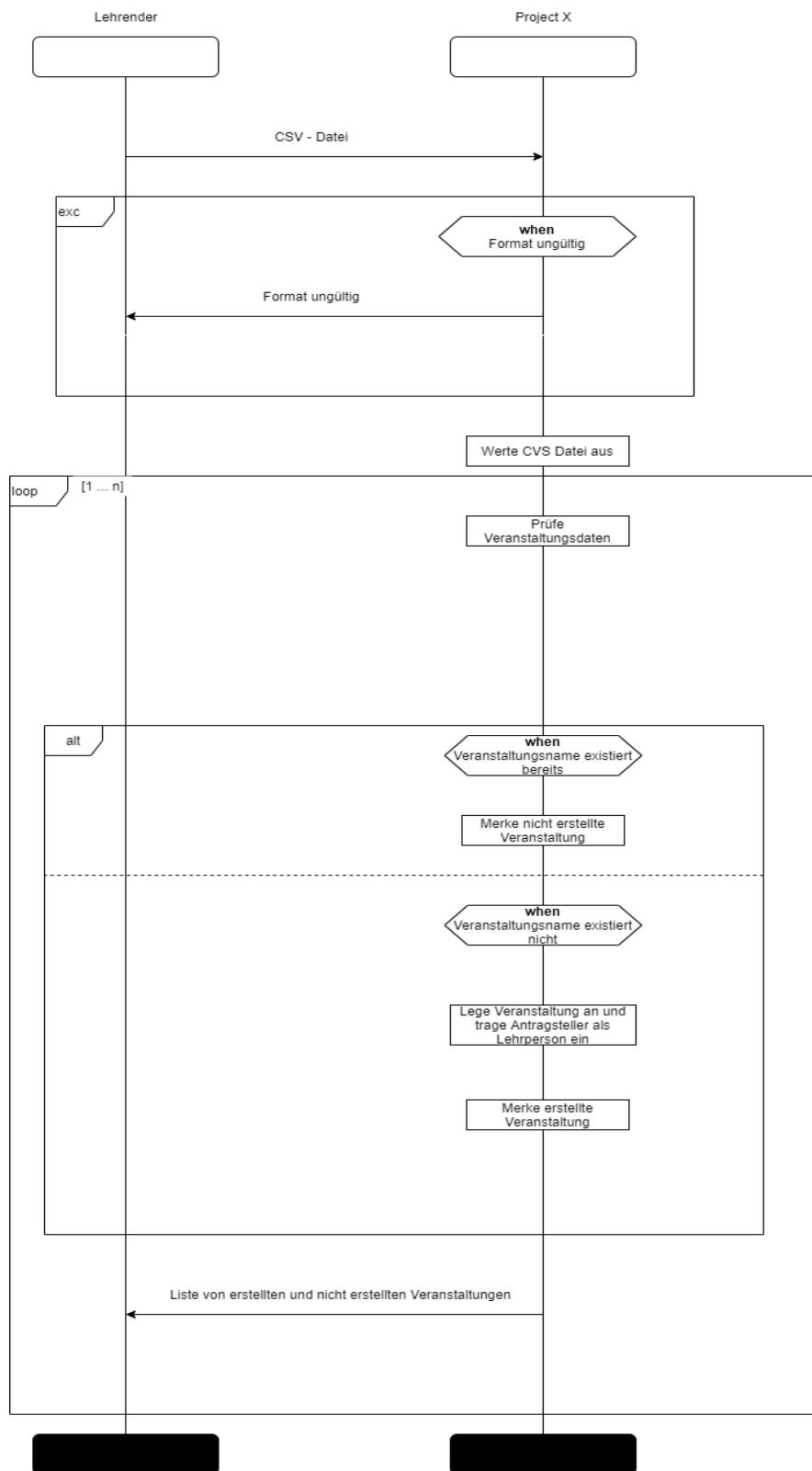
bMSC 1.2.3

Wenn ein Student sein Profil aufruft, werden ihm alle seine Angaben angezeigt. Da er nur die private Adresse, das Profilbild oder das Passwort ändern kann, werden nur für diese Angaben Textfelder bzw. FileChooser angezeigt. Die vorgenommenen Änderungen werden gespeichert, sobald der Nutzer den "Schließen" Button drückt.

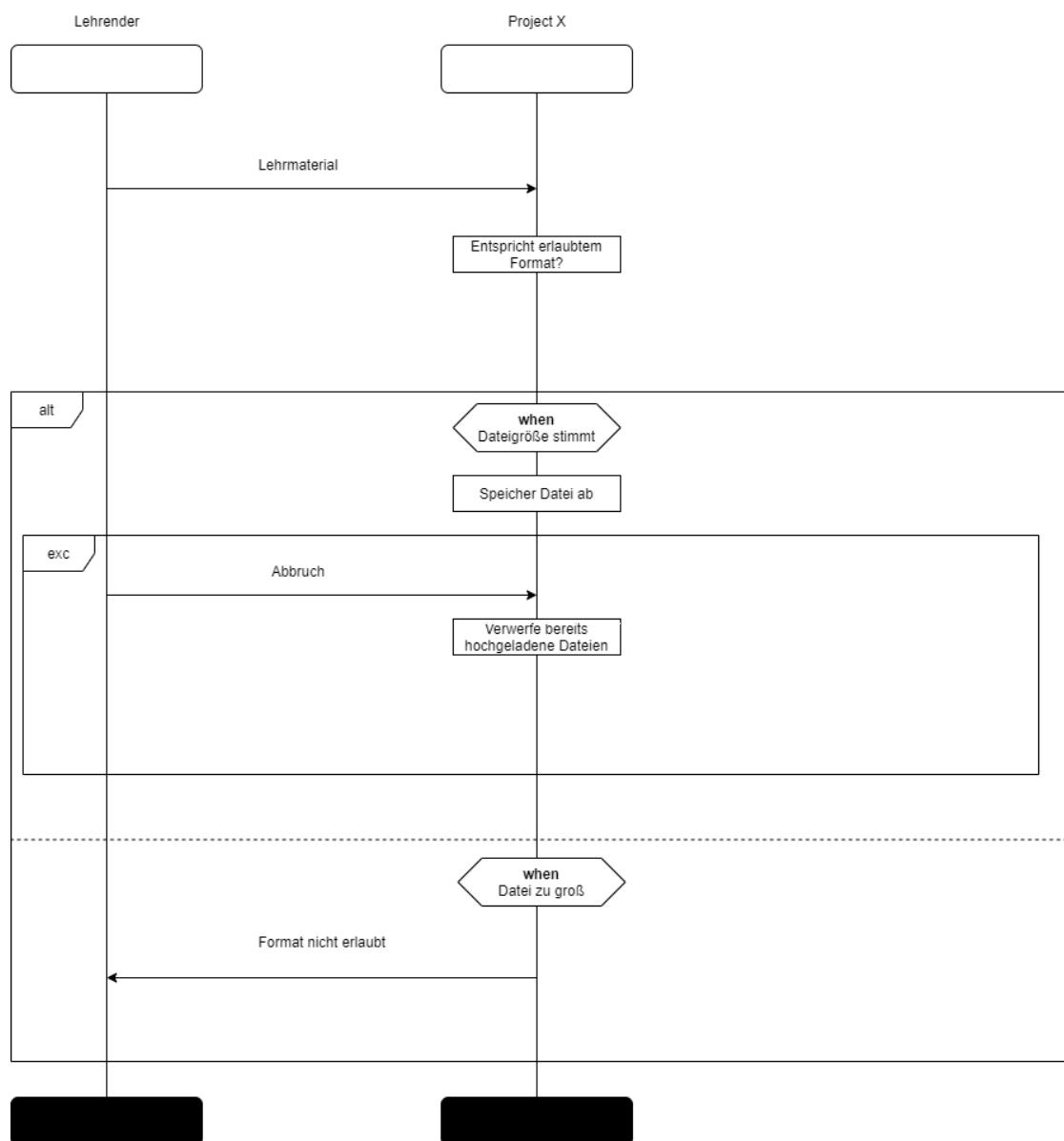
bMSC 2.2.1

Der Lehrende möchte eine Veranstaltung anlegen und gibt die benötigten Daten ein.

Das System prüft erst ob die Eingabe Korrekt ist und im Anschluss, ob bereits eine Veranstaltung mit dem Namen existiert. Existiert diese, gibt es eine Meldung an den Lehrenden, sonst wird die Veranstaltung im System angelegt.

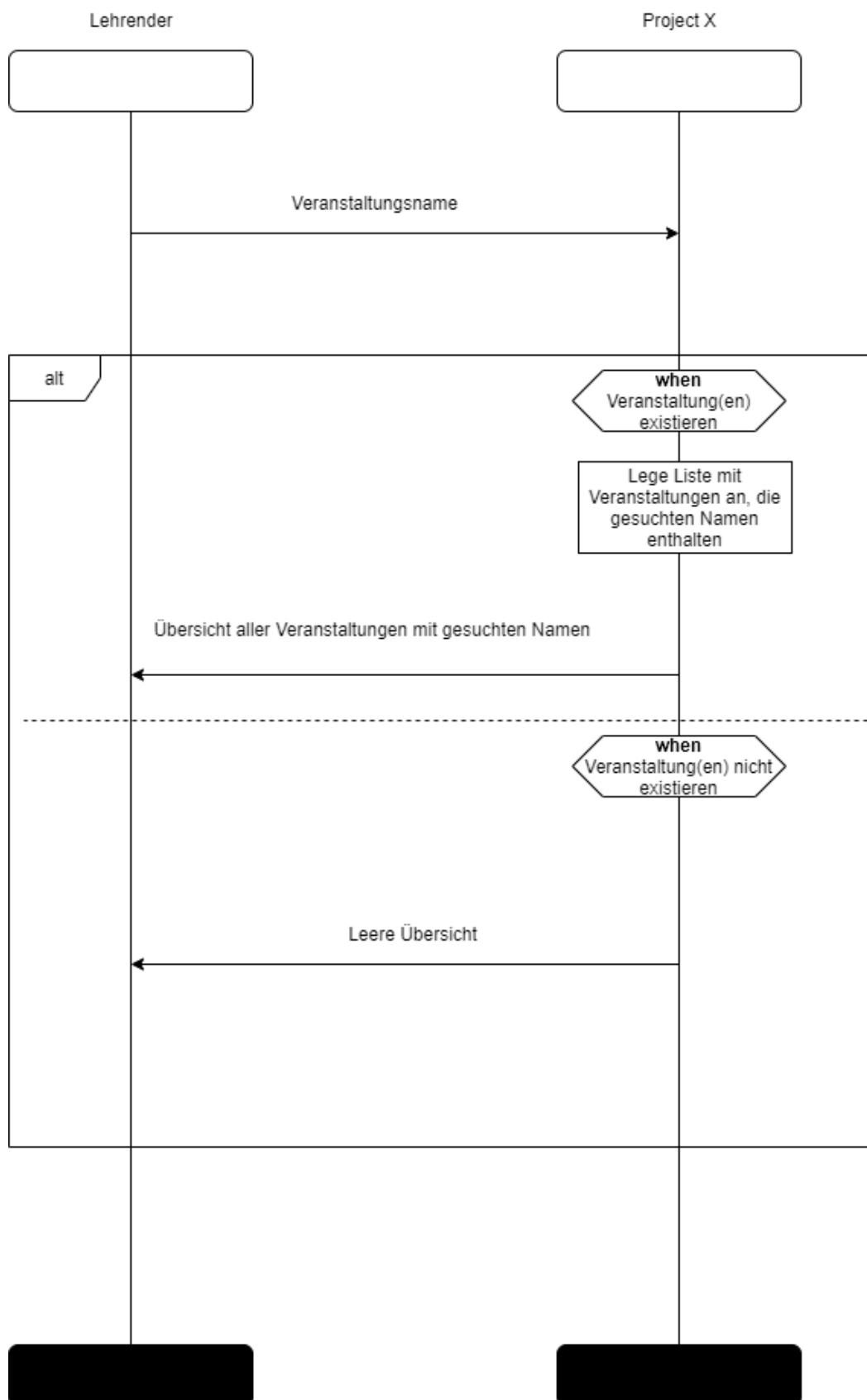
bMSC 2.2.2

Der Lehrende möchte mehrere Veranstaltungen anlegen und lädt dafür eine CSV-Datei hoch. Das System wertet die Einträge 1 bis n aus der Datei aus und prüft dabei, ob bereits eine Veranstaltung mit dem Namen existiert. Das System erstellt mit den gegebenen Daten, Veranstaltungen, die noch nicht existieren. Es merkt sich dabei angelegte und nicht angelegte Veranstaltungen und gibt diese zum Schluss an den Lehrenden weiter.

bMSC 2.2.3

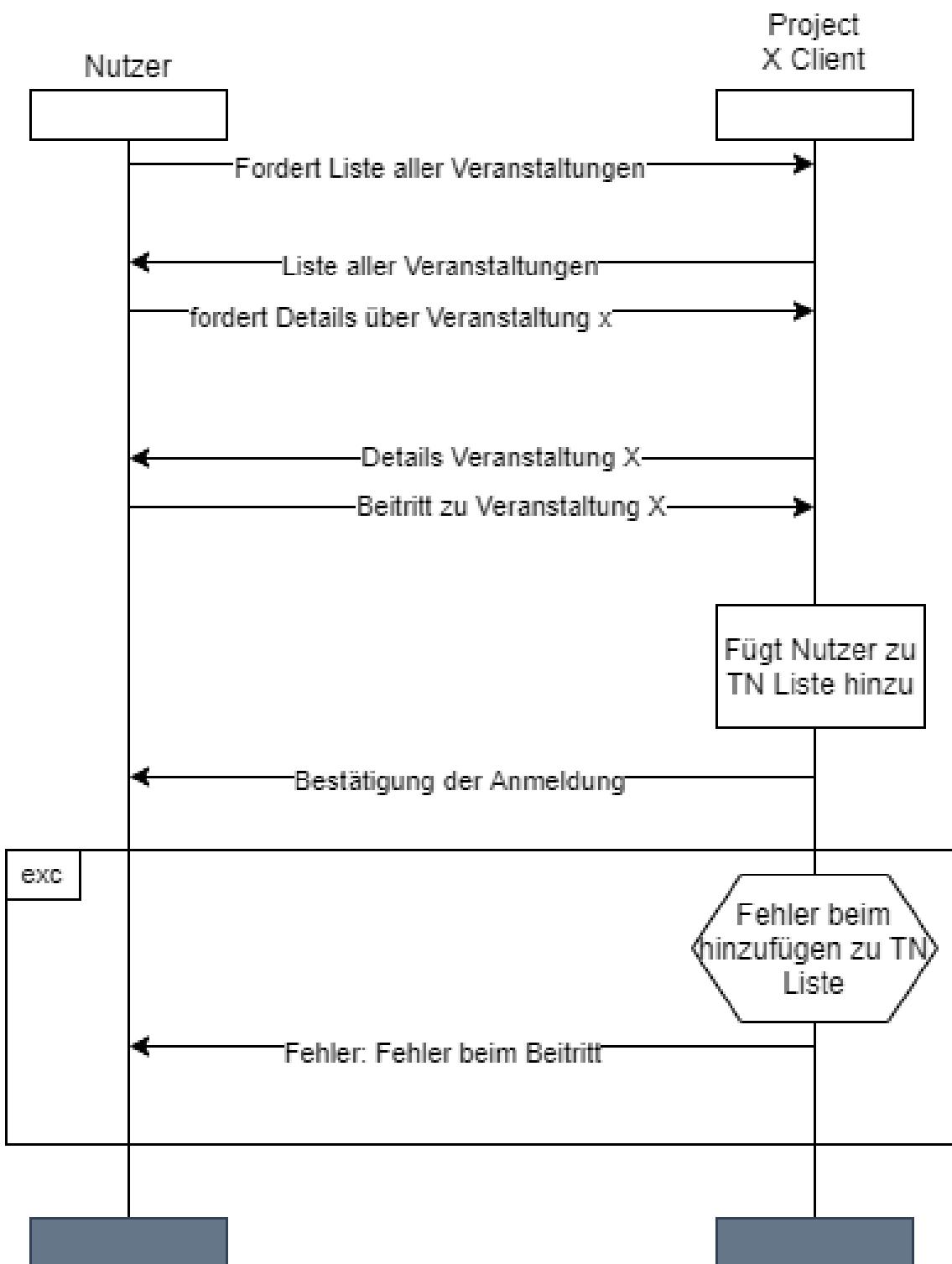
Der Lehrende möchte Lehrmaterial zur Verfügung stellen und lädt diese im System hoch.

Wenn das Material vom Lehrenden das richtige Größe hat, speichert das System die Datei ab und stellt diese in der Veranstaltungsübersicht zur Verfügung. In Ausnahmefällen, kann es zu einem Abbruch kommen, durch getrennte Verbindung oder der Lehrende bricht den Upload ab. Dann wird die Datei nicht hochgeladen.

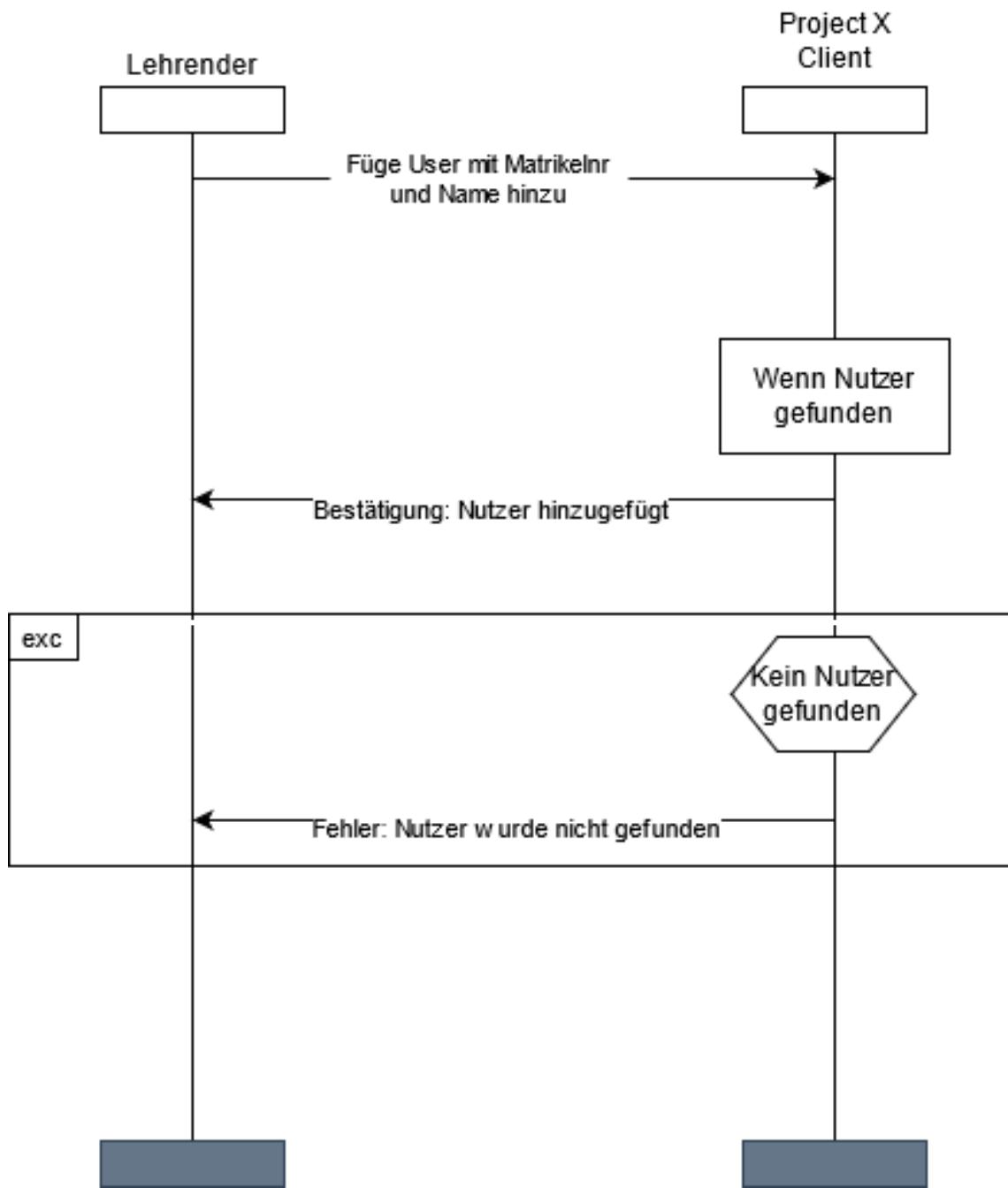
bMSC 2.2.4

Der Nutzer möchte eine Veranstaltung suchen und gibt diese in der Suche ein.

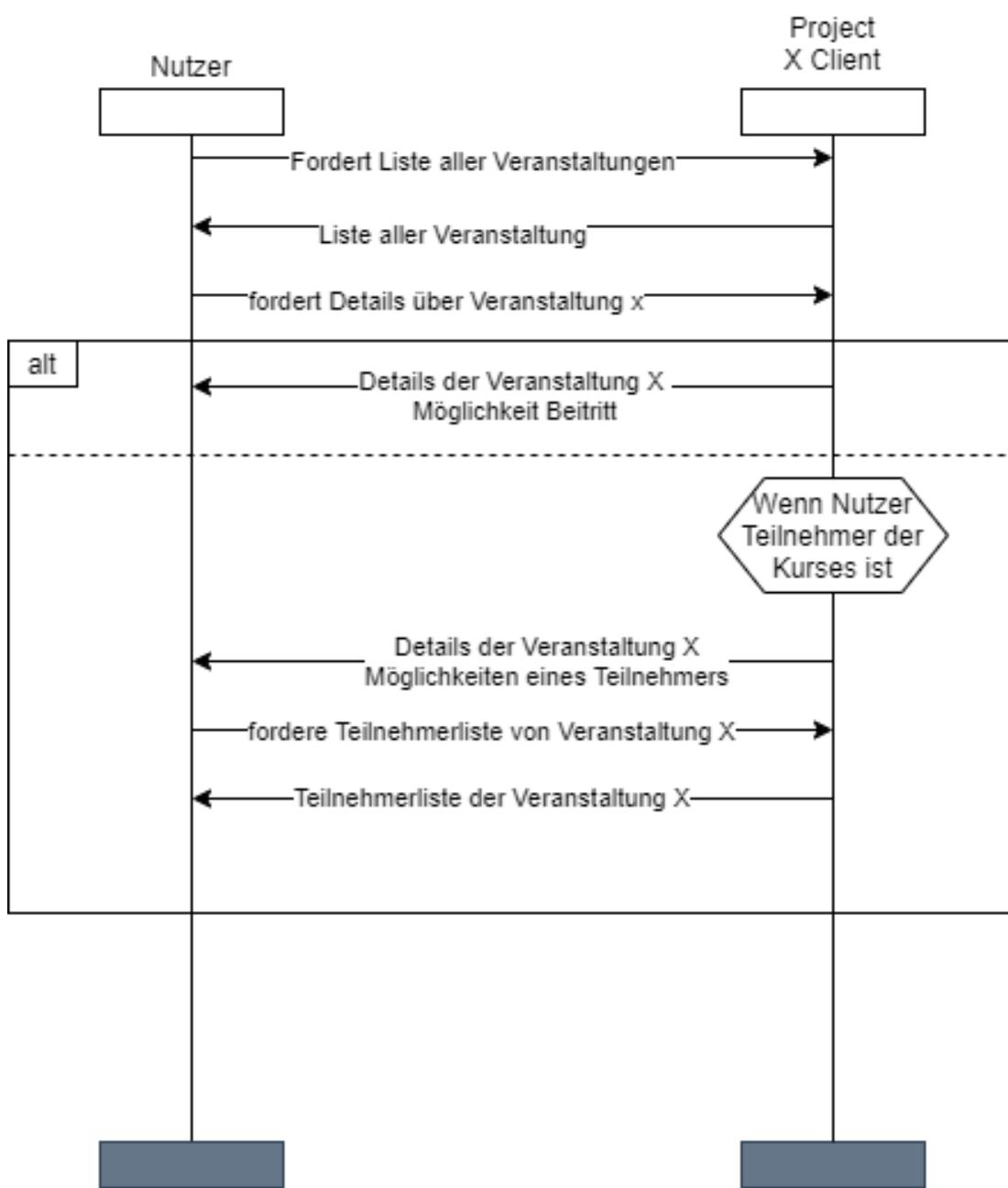
Das System sucht alle Veranstaltungen mit dem meisten Übereinstimmungen heraus und zeigt diese an. Gibt es keine Treffer, so wird eine leere Übersicht angezeigt.

bMSC 3.1.1, 2.1.1

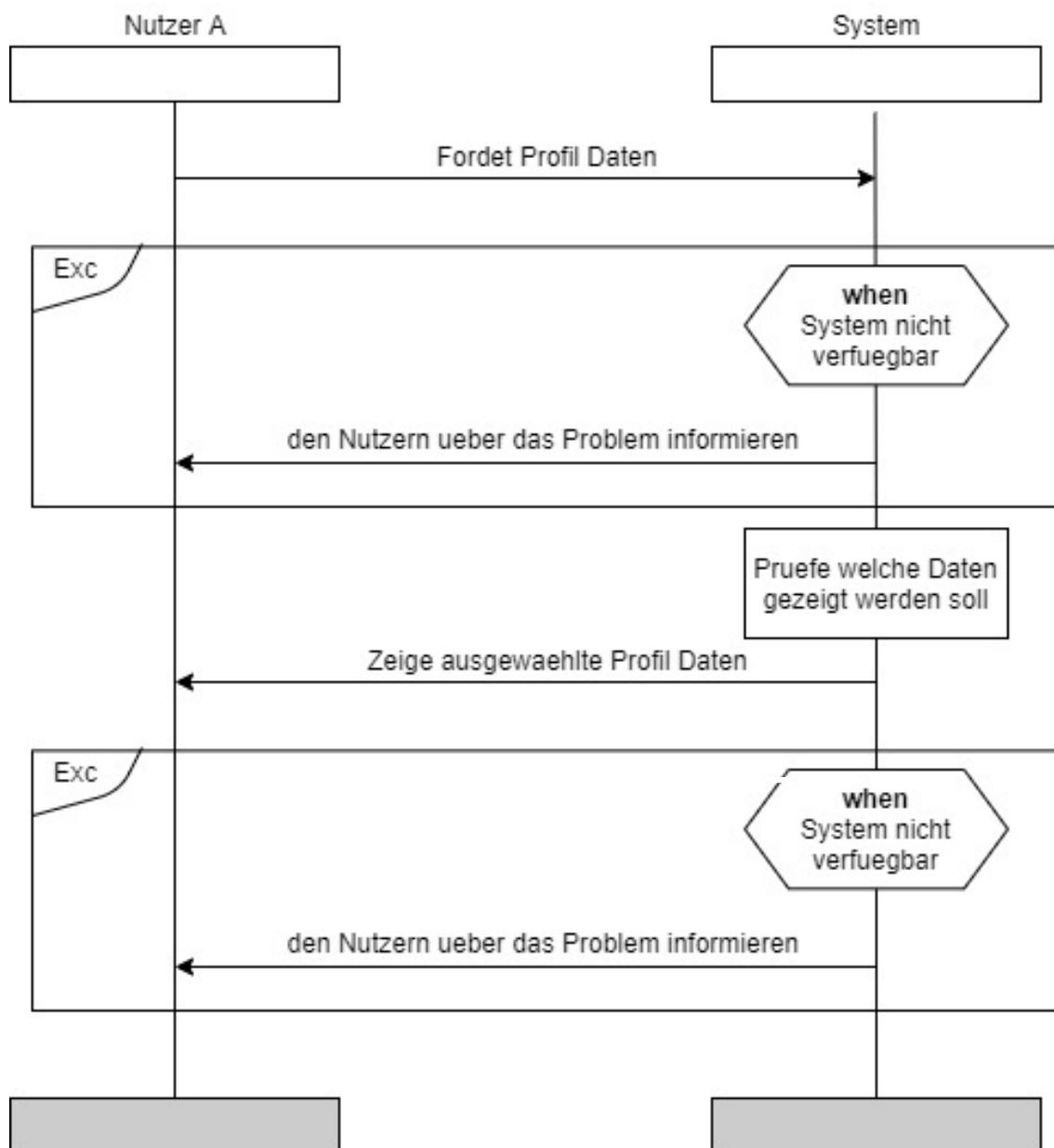
Der Nutzer fordert die Liste aller Veranstaltungen an. Aus der Liste aller Veranstaltungen will der User Details über eine beliebige Veranstaltung, in diesen Fall Veranstaltung X erfahren. Nachdem der User die Veranstaltung X genauer betrachten konnte, entscheidet sich der User dafür der Veranstaltung X beizutreten. Wenn der Beitritt erfolgreich ist wird der Nutzer der Teilnehmerliste hinzugefügt und erhält eine Bestätigung. Wenn ein Fehler auftritt und der Nutzer nicht der Teilnehmerliste hinzugefügt wird erhält er einen Hinweis das ein Fehler aufgetreten ist.

bMSC 3.1.2

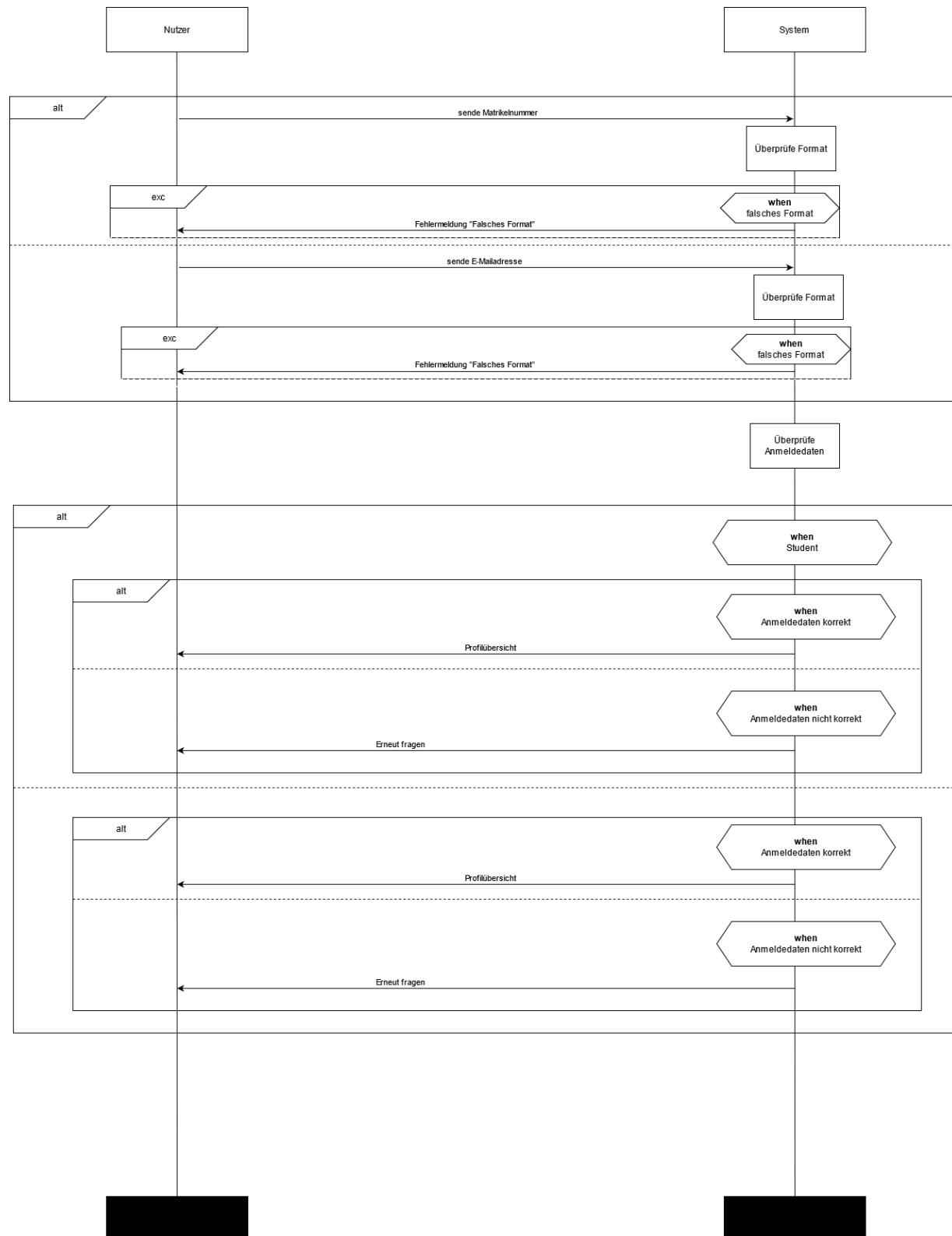
Ein Lehrender will manuell einen Nutzer mithilfe seiner Matrikelnummer und seines Namens zu seiner Lehrveranstaltung hinzufügen. Dafür gibt der Lehrende die Matrikelnummer und Namen des Hinzuzufügenden ein. Wenn ein Nutzer mit passenden Daten gefunden wird, wird dieser zu der Teilnehmerliste hinzugefügt und eine Bestätigung wird dem Lehrenden angezeigt. Falls kein Nutzer mit passenden Informationen gefunden werden konnte, wird dem Lehrenden angezeigt das der Nutzer nicht gefunden werden konnte.

bMSC 3.1.3

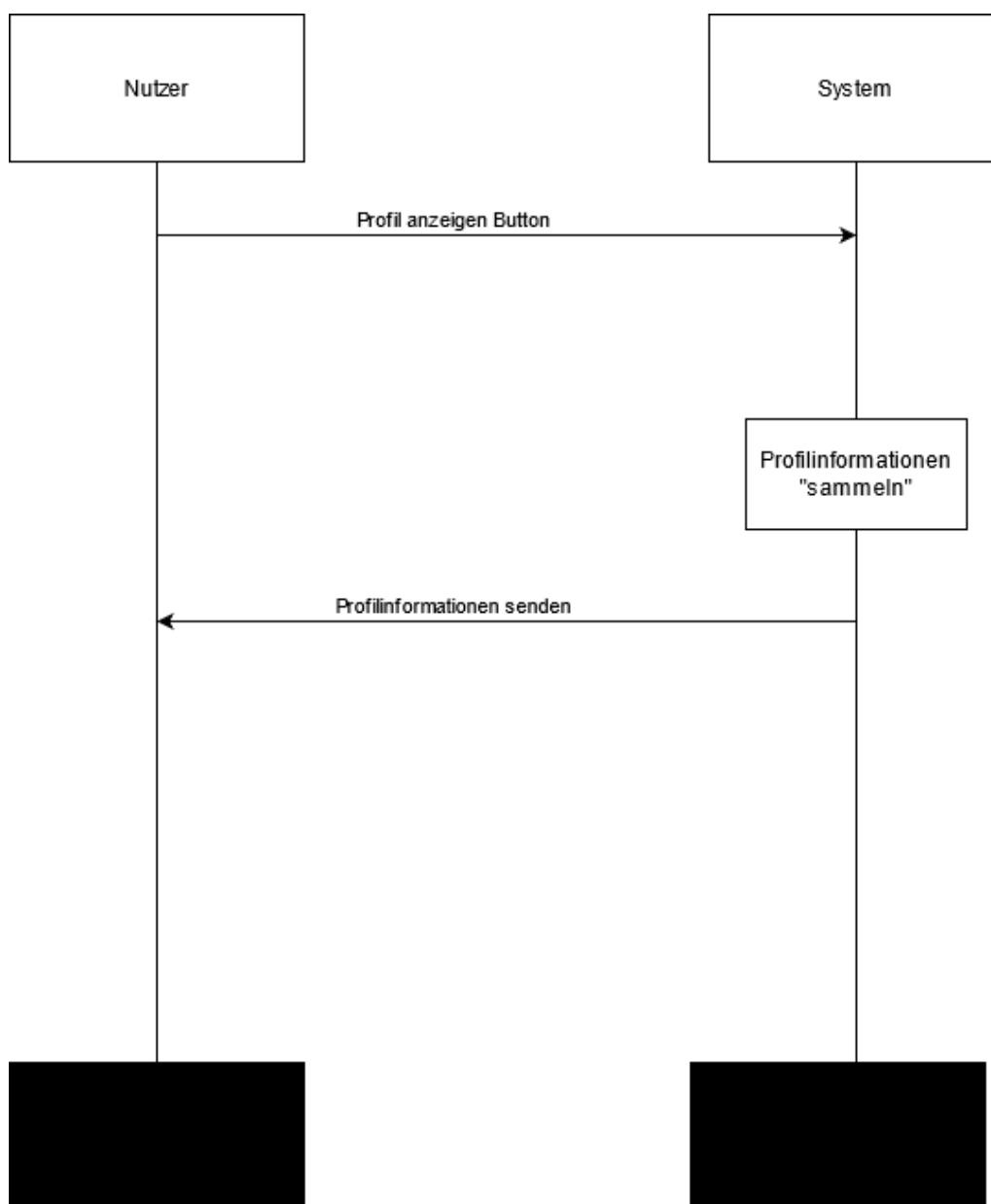
Der Nutzer fordert die Liste aller Veranstaltungen an. Aus der Liste aller Veranstaltungen will der User Details über eine beliebige Veranstaltung, in diesen Fall Veranstaltung X erfahren. Wenn der Nutzer Teilnehmer der Veranstaltung X ist hat er die Möglichkeit auf die Teilnehmerliste der Veranstaltung X zuzugreifen.

bMSC 4.3

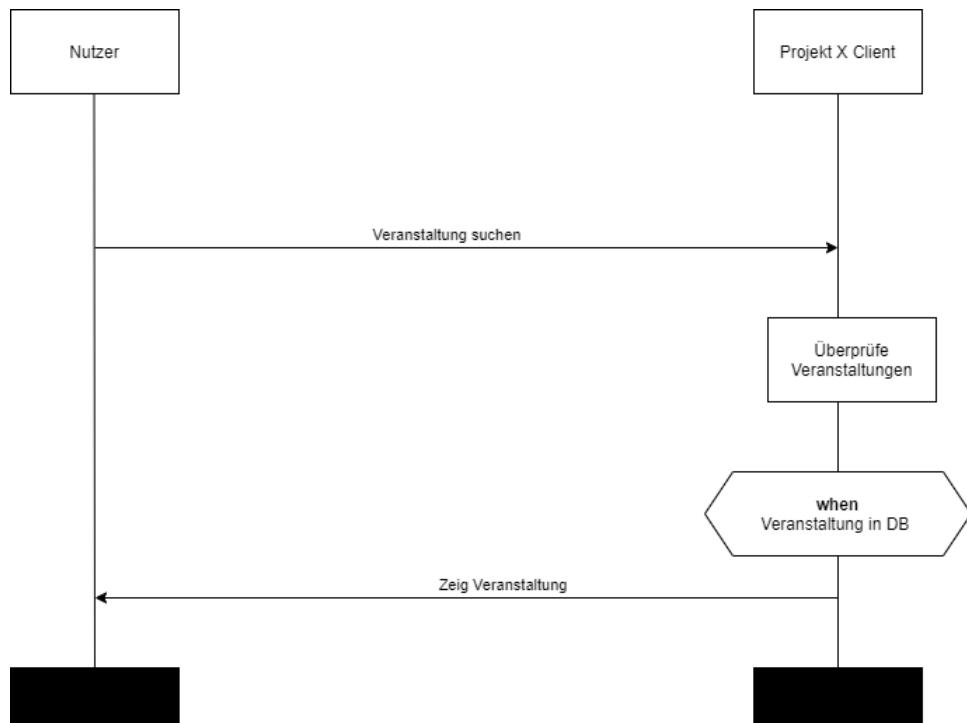
Wenn ein Nutzer auf eine Nutzerlink (Name) klickt, dann werden die Profildaten aufgerufen. Das System wählt welche Daten gezeigt werden soll aus, z.B. ein Student kann nicht die Matrikelnummer und die Adresse von anderem Nutzer sehen. Das System zeigt die ausgewählten Daten dem Nutzer.

bMSC 5.2.1

Wenn der Nutzer sich anmelden möchte, sendet der Nutzer seine Matrikelnummer oder seine E-Mailadresse. Dazu sendet er noch sein Passwort. Zuerst wird überprüft, ob der Nutzer in der Datenbank existiert. Wenn ja, wird sein Passwort aus der Datenbank abgerufen und gegen das Passwort verglichen, welches er im Client eingegeben hat. Sollte sich herausstellen, dass der Nutzer ein Lehrender ist, wird er andere Ansicht und andere Optionen zum Interagieren mit dem System haben.

bMSC 5.3.1

Im gezeigten bMSC wird ein Profilauftrag durchgeführt. Wenn der Nutzer auf den "Profil anzeigen"-Button gedrückt hat, wird erstmal geguckt, ob der Nutzer ein Lehrender, oder ein Studierender ist. Daraufhin werden die Profilinformationen gesammelt und bei den verschiedenen Personen angezeigt.

bMSC 5.3.2, 3.1.4

In diesem bMSC wird modelliert, wie man eine Lehrveranstaltung suchen kann. Der Nutzer tippt den gewünschten Kurs in das System ein. Daraufhin wird geguckt, ob der Kurs in der Datenbank schon vorhanden ist, oder nicht. Wenn das der Fall sein sollte, wird der gewünschte Kurs ausgegeben.

Strukturdiagramm (Klassendiagramm)

Ein Strukturdiagramm dient der grafischen Darstellung von Klassen, Schnittstellen und deren Beziehungen. Es hilft dabei, Quellcode und Implementierungsarbeiten zu strukturieren bevor diese starten und ermöglicht eine Aufteilung der Programmieraufgaben.

Client

Erläuterung des Strukturdiagramms:

In dem Source Paket des Clients befinden sich 7 Packages.

Im Model Package befinden sich alle Klassen, die Objekte erstellen, welche wir zwischen den views übergeben müssen.

Die views sind alle als fxml Dateien im fxml Package gespeichert.

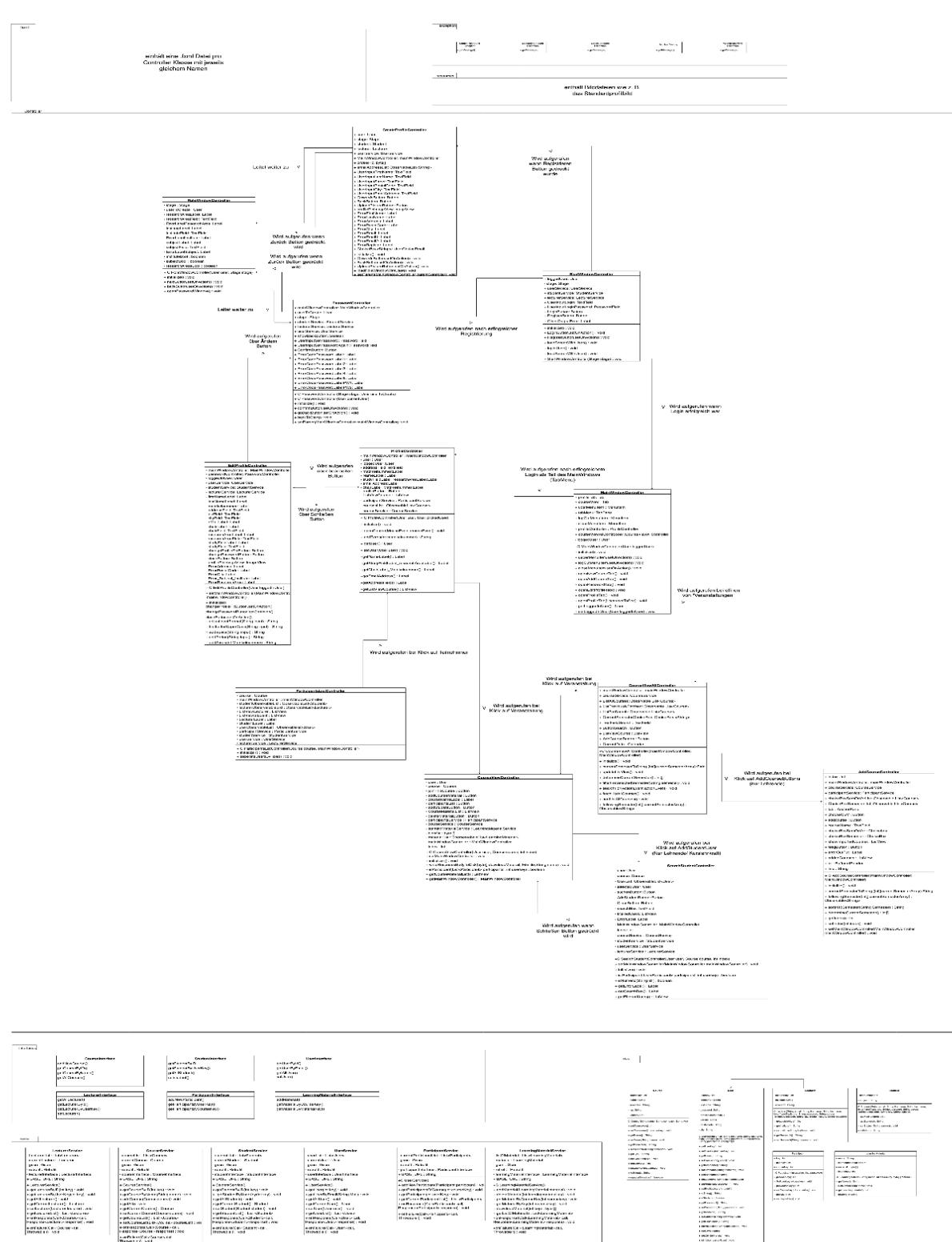
Im Controller Package befinden sich alle Controller, die die Funktionen der .fxml Dateien steuern. Zu jeder view im .fxml Package gibt es einen Controller mit dem selben Namen.

Im Interface Package befinden sich die zugehörigen Interfaces zu den models.

Im Service Package befinden sich die Services zu den Models.

Im Exception Package sind alle Exceptions enthalten, die aufgerufen werden, sobald ein Fehler vorliegt bspw. Nutzer wurde nicht in Datenbank gefunden.

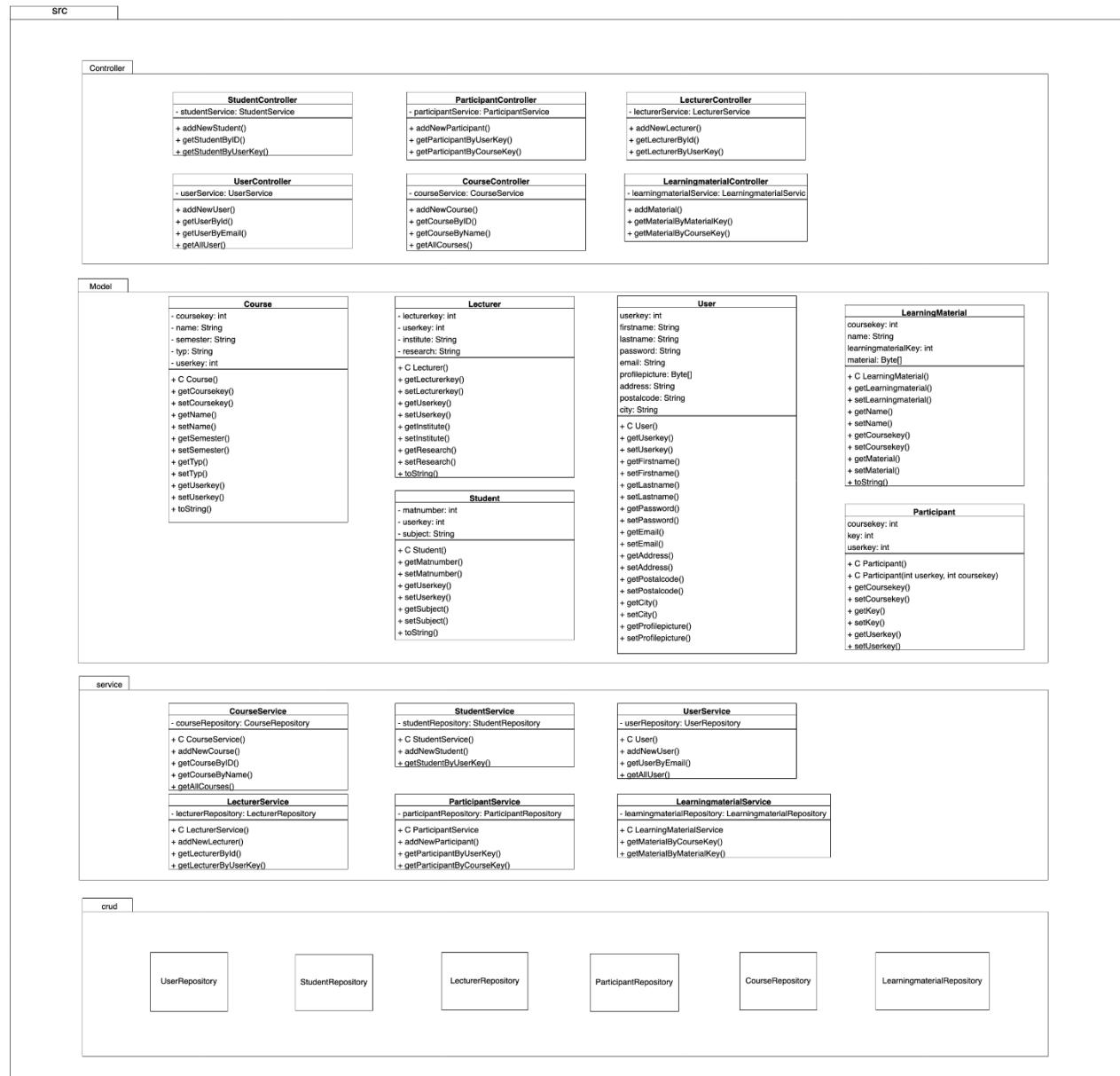
Im resources Package befinden sich Bilddateien für die views wie das Standardprofilbild.



Server

In jedem Package gibt es eine Klasse für unsere fünf verschiedenen Objekte: User, Student, Lecturer, Participant und Course.

Jeder von ihnen hat sein eigenes Repository auf das über den zugehörigen Service und Controller zugegriffen werden kann. Im model Package befinden sich die Klassen der jeweiligen Objekte.



Funktionalitätsplanung

Anhand der Funktionalitätsplanung werdet ihr geprüft. Diese sollte dementsprechend **immer** aktuell gehalten werden und pro Funktionalität **nur einen** Verantwortlichen enthalten. Als Quellcodereferenz solltet ihr immer euer Package, eure Klasse und die dazugehörige Methode angeben. Ggf. könnt ihr auch Zeilenangaben machen. Folgend erhalten Sie ein Template.

Id	Funktionalität	Geschätzter Aufwand	Verantwortlicher	Abhängige Funktionalitäten	Verknüpfte User-Stories	Quellcode-referenz	Status
1.	Login						
1.1	Startfenster						
1.1.1	Prüfe, ob Login Name eine Mail Adresse oder eine Matrikelnummer ist	1 Tag	Kurinchin ilavan Thanges waran		5.1.1	Client.Controller.StartWindowController.loginUser()	Fertig
1.1.2	Prüfe Passwort und melde User an	2 Tag	Kurinchin ilavan Thanges waran	1.2	5.1.1, 5.1.2	Client.Controller.StartWindowController.loginUser()	Fertig
1.1.3	User informieren, wenn Login Daten falsch sind	1 Tag	Kurinchin ilavan Thanges waran	1.1.2	5.1.6	Client.Controller.StartWindowController.loginUser()	Fertig
1.1.4	Unterscheidung der Rollen anhand von Email-Adressen Endung	1 Tag	Kurinchin ilavan Thanges waran		5.1.1, 5.1.3	In User Model – Client IsStudent()	Fertig
1.1.5	User anhand der Email aus Datenbank raussuchen	1 Tag	Kurinchin ilavan Thanges waran	1.1.4	5.1.1	In UserService - Server	Fertig
1.1.6	Login über "login" Button	1 Tag	Kurinchin ilavan Thanges waran		5.1.1, 5.1.2	Client.Controller.StartWindowController.Initialize() LoginButton.setOnAction()	Fertig

1.1.7	Login über Username Textfeld	1 Tag	Noemi Kallweit		5.1.1, 5.1.2	StartWindowController Initialize() UserInputLogin.setOnAction()	Fertig
1.1.8	Login über Passwort Textfeld	1 tag	Noemi Kallweit		5.1.1, 5.1.2	StartWindowController Initialize() UserInputPassword.setOnAction()	Fertig
1.2	Passwort						
1.2.1	Passwort nach Beschränkungen prüfen	1 Tag	Frederik Gnass	1.3, 1.4	1.1.2	Client.Controller.Password Controller Initialize() ConfirmButton.setOnAction()	Fertig
1.2.2	Überprüfung, ob beide Passwörter gleich sind	1 Tag	Frederik Gnass		1.1.2	Client.Controller.Password Controller Initialize() ConfirmButton.setOnAction()	Fertig
1.2.3	Vom Passwortfenster zurück zum RoleWindow (bei Registrierung)	1 Tag	Frederik Gnass		1.2.1, 1.3.1	Client.Controller.Password Controller Initialize() goBackButton.setOnAction()	Fertig
1.2.4	Nach erfolgreicher Registrierung zurück zum Startfenster wechseln	1 Tag	Frederik Gnass		5.1.1, 5.1.2	Client.Controller.Password Controller BackToStart()	Fertig
1.2.5	Ohne Änderung zurück zum EditProfile Fenster wechseln (wenn Nutzer bereits eingelogged war)	1 Tag	Frederik Gnass		1.2.3	Client.Controller.Password Controller Initialize() CancelButton.setOnAction()	Fertig
2.	Verwaltung von Lehrveranstaltungen						

2.1	Eingabe Formate überprüfen						
2.1.1	Eingabe Format prüfen - Client	1 Tag	Thorben Kronewald-Lamken		2.2.1	AddCourseController.initialize()	Fertig
2.1.2	Eingabe Daten Format prüfen - Client	1 Tag	Thorben Kronewald-Lamken		2.2.1	AddCourseController.initialize()	Fertig
2.1.2.1	Eingabe Format Vereinheitlichung und Erkennung des Semesters	1 Tag	Vera Mitschulat		2.1.1	AddCourseController.sortInToSemester(String semester)	Fertig
2.2	Veranstaltung prüfen	1 Tag	Thorben Kronewald-Lamken	2.1.1.,2.1.2	2.2.1	AddCourseController.initialize().addCourse.setOnAction()	Fertig
2.3	Lege Veranstaltung an - Server	1 Tag	Thorben Kronewald-Lamken	2.2	2.2.1,2.2.2	AddCourseController.initialize().addCourse.setOnAction()	Fertig
2.4	Werte CSV-Datei aus	1 Tag	Thorben Kronewald-Lamken		2.2.2	AddCourseController.initialize().chooseCSV.setOnAction()	Fertig
2.5	Veranstaltungen View für verschiedene Nutzer anpassen	1 Tag	Hana Habibi	2.6, 2.7, 2.8	2.7	Client.Controller.CourseViewController Initialize()	Fertig
2.6	Lehrmaterialen – Server und Client	2 Tage	Hana Habibi		2.2.3	Server: LearningMaterial Controller, LearningMaterialService, LearningMaterial (class), LearningMaterial Repository Client: LearningMaterial Controller, LearningMaterialService, LearningMaterial (class), LearningMaterial.fxml	Fertig
2.7	Prüfe Format vom	1 Tag	Hana Habibi		2.6	LearningMaterialController	Fertig

	Lehrmaterial - Client						
2.8	Speicher Lehrmaterial ab – Server und Client	1 Tag	Hana Habibi	2.6	2.6	LearningMaterialControllerLearninigMatrialService, LearningMaterial Repository	Fertig
2.9	Löschen des Lernmaterial – Server und Client	1 Tag	Hana Habibi		2.6	LearningMaterialControllerLearninigMatrialService, LearningMaterial Repository	
2.10	Suche nach Veranstaltung - Server	1 Tag	Vera Mitschulat		2.1.1	In Server: .Course Repository, Controller, Service CourseService: -getCourseByName() -getCoursebyID() -getAllCourses()	Fertig
2.11	Liste der Veranstaltungen - Client	1 Tag	Vera Mitschulat	2.8	2.1.1	CourseViewAllController (Alles) AddCourseController: -SortInToSemester()	Fertig
2.11.1	Anzeigen von Allen Veranstaltungen aller Semester Chronologisch	1 Tag	Vera Mitschulat	2.9.2. 8	3.1.4	CourseViewAllController: -fetchListAllCourses() -sortListAllCourses() -updateListView()	Fertig
2.11.2	Anzeigen aller Lehrveranstaltungen eines Semesters	1 Tag	Vera Mitschulat	2.9.2. 8	3.1.4	CourseViewAllController: -fetchListAllCourses() - determineCurrentSemester() -initialize() -updateListView()	fertig
2.11.3	Such Funktion innerhalb aller Lehrveranstaltungen	1 Tag	Vera Mitschulat	2.9.2. 8	3.1.4	CourseViewAllController: -fetchListAllCourses() -updateListView() -searchFor()	Fertig
2.11.4	Such Funktion innerhalb eines bestimmten Semesters	1 Tag	Vera Mitschulat	2.9.2, 2.9.2. 8	3.1.4	CourseViewAllController: -fetchListAllCourses() -updateListView() -searchFor() - determineCurrentSemester() -initialize()	Fertig

2.11.5	Öffnen des gewünschten Kurses durch Doppelklick auf ListView Eintrag	1 Tag	Vera Mitschulat	2.9.2, 2.9.2.8, 2.9.4	3.1.1	CourseViewAllController: -fetchListAllCourses() -updateListView() -searchFor() - determineCurrentSemester() -initialize()	Fertig
2.12	Die belegte Veranstaltung in Profil anzeigen	1 Tag	Hana Habibi		3.1.1	Client.Controller.ProfileController initialize()	Fertig
3.	Zuordnung von Nutzern zur Lehrveranstaltung						
3.1	Participant Server und Client Struktur	1 Tag	Vera Mitschulat		3.1.3	Participant Repository, Service, Controller, Model (Server) Participant Model, Interface , Service	Fertig
3.2	Lehrende fügt Studenten in einem Kurs hinzu	2 Tage	Hana Habibi		3.1.2	Client.Controller.SearchStudentController	Fertig
3.3	Studenten suchen	1 Tag	Hana Habibi		3.1.2	SearchStudentController	Fertig
3.3.1	Studenten mit Matrikelnummer suchen	1 Tag	Hana Habibi		3.1.2	SearchStudentController	Fertig
3.3.2	Studenten mit Namen suchen	1 Tag	Hana Habibi		3.1.2	SearchStudentController	Fertig
4.	Nutzerprofil						
4.1	Eigenes Profil beim Klick auf den eigenen Namen anzeigen	1 Tag	Hana Habibi		4.1.1	Client.Controller.ProfileController	Fertig
4.2	Profil von anderen Studenten, bei Klick auf deren Namen anzeigen (aus Sicht Student)	1 Tag	Hana Habibi		4.1.2	Client.Controller.ProfileController	Fertig

4.3	Profil von anderen Studenten, bei Klick auf deren Namen anzeigen (aus Sicht Lehrender)	1 Tag	Hana Habibi		4.1.3	Client.Controller.ProfileController Initialize()	Fertig
4.4	Profil von Lehrenden, bei Klick auf deren Namen anzeigen	1 Tag	Hana Habibi		3.1.3	Client.Controller.ProfileController Initialize()	Fertig
4.5	Profil bearbeiten						
4.5.1	EditProfile-Fenster Labels mit Angaben des loggedInUsers belegen	1 Tag	Noemi Kallweit	4.5.5	1.2.3	Client.Controller.EditProfileController Initialize()	Fertig
4.5.2	Textfelder auf Vollständigkeit überprüfen - keine leeren Textfelder erlaubt	1 Tag	Noemi Kallweit		1.2.3	Client.Controller.EditProfileController Initialize() CloseButton.setOnAction() {...}	Fertig
4.5.3	Textfeldeingabe formatieren vor dem Abspeichern	1 Tag	Noemi Kallweit		1.1.1	Client.Controller.EditProfileController AdjustInputFormat() AddSpaces() AddPeriod() FirstLetterUpperCase()	Fertig
4.5.4	Eingabeformate der Textfelder prüfen - Fehlermeldung anzeigen, falls falsch	1 Tag	Noemi Kallweit		1.1.1	Client.Controller.EditProfileController Initialize() CloseButton.setOnAction() {...}	Fertig
4.5.5	Über "Ändern" Button Password-Fenster öffnen	1 Tag	Noemi Kallweit		1.1.2	Client.Controller.MainWindowController openPasswordTab()	Fertig

4.5.6	Über "Schließen" Button Profile- Fenster öffnen	1 Tag	Noemi Kallweit		5.1.3	Client.Controller.MainWin- dowController openProfileTab()	Fertig
4.5.7	Von Profilfenster zum EditProfileWind- ow wechseln	1 Tag	Noemi Kallweit		5.1.3, 1.2.3	Client.Controller.MainWin- dowController openEditProfileTab()	Fertig
5.	Startfenster						
5.3	Lehrveranstaltu- ng beitreten	4 Tage	Hana Habibi		5.1.5	Client.Controller.CourseVi- ew Initialize() joinThisCourse.setOnAction()	Fertig
5.4	Material herunterladen/ anschauen	4 Tage	Hana Habibi		5.1.4	Client.Controller.CourseVi- ew DownloadMaterial()	Fertig
6	Registrierung						
6.1	Neuen Nutzer anlegen						
6.1.1	Textfelder auf Vollständigkeit überprüfen - keine leeren Textfelder erlaubt	1 Tag	Frederik Gnass		1.1.1	Client.Controller.CreatePro- fileController Initialize() OnwardsButton.setOnAction()	Fertig
6.1.2	Eingabeformate der Textfelder prüfen - Fehlermeldung anzeigen, falls falsch	2 Tage	Frederik Gnass		1.1.1	Client.Controller.CreatePro- fileController Initialize() OnwardsButton.setOnAction()	Fertig
6.1.3	Textfelder mit Nutzerangaben belegen, falls Nutzer != null	2 Tage	Frederik Gnass		1.1.1	Client.Controller.CreatePro- fileController Initialize()	Fertig
6.1.4	Vom CreateProfile Fenster zum RoleWindow wechseln	1 Tag	Frederik Gnass		1.1.1, 1.2.1, 1.3.1	Client.CreateProfileControl- ler LoadRoleWindowWithUser() ()	Fertig
6.1.5	Vom CreateProfile Fenster zum Startfenster wechseln	1 Tag	Frederik Gnass		1.1.1	Client.Controller.CreatePro- fileController Initialize() BackButton.setOnAction()	Fertig

6.2	RoleWindow						
6.2.1	RoleWindow an Nutzer (Student oder Lehrperson) anpassen	1 Tag	Noemi Kallweit		1.2.1, 1.3.1	Client.Controller.RoleWindowController Initialize()	Fertig
6.2.2	Textfelder auf Vollständigkeit überprüfen - keine leeren Textfelder erlaubt	1 Tag	Noemi Kallweit		1.2.1, 1.3.1	Client.Controller.RoleWindowController Initialize() nextButton.setOnAction()	Fertig
6.2.3	Zurück zum CreateProfile-Fenster gehen	1 Tag	Noemi Kallweit		1.1.1	Client.Controller.RoleWindowController Initialize() backButton.setOnAction()	Fertig
6.2.4	Weiter zum Password-Fenster gehen	1 Tag	Noemi Kallweit		1.1.2	Client.Controller.RoleWindowController openPasswordWindow()	Fertig
7	Profilbild						
7.1	Profilbild auswählen (BLOB)	2 Tage	Frederik Gnass		1.1.1	Client.Controller.CreateProfileController UploadPictureButton.setOnAction{...} Client.Controller.EditProfileController Initialize() “changeProfilePicButton.setOnAction{...}”	Fertig
7.2	Formatmöglichkeiten für Profilbild nach .png, .PNG, .jpg, .JPG filtern	2 Tage	Frederik Gnass		1.1.1	CreateProfileController, Client.Controller.EditProfileController Initialize() “changeProfilePicButton.setOnAction{...}”	Fertig

Systemtests

Datum	19.05.2021		
Tester	Noemi Kallweit		
SW-Version	V 0.1.2		
Vorbedin-gung(en)	Nutzer „Holger Müller“ ist am System eingeloggt und hat den „EditProfile“ Tab geöffnet		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Nutzer drückt den „Ändern“ Button.	Es öffnet sich der „Password“ Tab	✓
2	Der Nutzer gibt ein zwei unterschiedliche Passwörter ein und bestätigt die Eingabe	Das System prüft, ob Eingabe den Vorgaben entspricht und zeigt ein Error Label an weil die Passwörter nicht übereinstimmen	✓
3	Der Nutzer gibt ein Passwort ein, welches nicht den Vorgaben entspricht und bestätigt die Eingabe	Das System prüft, ob Eingabe den Vorgaben entspricht und zeigt mehrere Error Label mit den Vorgaben an.	✓
4	Der Nutzer gibt ein zweimal ein gleiches Passwort ein, welches den Vorgaben entspricht und bestätigt die Eingabe	Das System prüft, ob Eingabe den Vorgaben entspricht und zeigt kein Error Label an. Der Password Tab wird zum EditProfileTab	✓
Nachbe-dingung(en)	Das neue Passwort wird in der Datenbank gespeichert		
Testurteil	Test bestanden.		

Datum	20.05.2021		
Tester	Kurinchinilavan Thangewaran		
SW-Version	V 0.1.2		
Vorbedin-gung(en)	Student mit Matrikelnummer "0000007" bzw. Email " max.musterman@stud.projectx.de " und Passwort "Test123!" existiert		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der User gibt als Login "asd4544" und als Passwort "Test123!" ein und klickt auf "Login"	Das System Zeigt "Fehlerhafter Login Name " an.	✓
2	Der User gibt als Login "1234567" und als Passwort "Test123!" ein und klickt auf "Login"	Das System Zeigt "Kein Nutzer mit Email/Matrikelnummer gefunden! " an.	✓
3	Der User gibt als Login "0000007" und als Passwort "Test123!" ein und klickt auf "Login"	User wird eingeloggt, es wird dem User sein Profil angezeigt.	✓
Nachbe-dingung(en)	User ist angemeldet und sieht das Tab Menü		✓
Testurteil	Test bestanden.		

Datum	20.05.2021		
Tester	Vera Mitschulat		
SW-Version	V 0.1.2		
Vorbedin-gung(en)	Es gibt Veranstaltungen in der Datenbank, User ist angemeldet		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Ein User greift auf die Übersicht zu und fordert so eine Übersicht aller Veranstaltungen an	Das System zeigt alle Veranstaltungen des aktuellen Semesters an.	✓
2	Der User lässt sich mit der Auswahl "Alle Semester" in der Choicebox, alle Veranstaltungen in allen Semestern anzeigen	Das System zeigt alle Veranstaltungen an, die in der Datenbank vorhanden sind	✓
3	Der User gibt ein Suchwort im Suchfeld ein und bestätigt seine Suche mit einem Klick auf den Button "Suchen"	Das System zeigt alle Veranstaltungen an dessen Namen das Suchwort, ungeachtet Groß- oder Kleinschreibung beinhalten.	✓
Nachbe-dingung(en)	Die Veranstaltungen werden in der ListView angezeigt		
Testurteil	Test bestanden.		

Datum	20.05.2021		
Tester	Hana Habibi		
SW-Version	V 0.1.2		
Vorbedin-gung(en)	Nutzer(Lehrer) Edgar Langbein ist erfolgreich eingeloggt und öffnet eine Veranstaltung, die er selbst erzeugt hat.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Benutzer öffnet den Veranstaltung Tab.	Das System zeigt das Kurs View vollständig an. Anstatt dem beitreten-Button wird hier ein AddStudent-button gezeigt. Der Lehrer kann Lernmaterial hinzufügen und löschen.	✓
2	Der Benutzer klickt auf den Materialhinzufügen-Button	Der File Chooser wird geöffnet damit Edgar die gewünschte files hochlädt.	✓
3	Der Benutzer wählt den pdf oder docx file aus und klickt das okay-button	Das File soll auf die Lernmaterialliste gezeigt werden	✓
4	Der Benutzer doppelklickt auf einem Material der Liste	Das Lernmaterial wird heruntergeladen.	✓
5	Der Benutzer klickt einmal auf einem Material der Liste.	Das geklickte Material wird ausgewählt.	✓
6	Der Benutzer klickt auf den löschen-Button	Der ausgewählte File wird gelöscht.	✓
7	Der Benutzer klickt auf den AddStudent-Button	Auf dem gleichen Tab wird eine neue Seite geöffnet. Hier kann Edgar einen User suchen und sie/ihn in dem Kurs hinzufügen.	✓
8	Der Benutzer schreibt der Name des gewünschten Users und klickt auf NameSuche-Button	Eine Liste von Nutzer mit denselben Namen ist gezeigt	✓
9	Der Benutzer klickt einmal auf dem Nutzer,	Der Nutzer von der Liste wird ausgewählt.	✓

	den er im Kurs hinzufügen will		
10	Der Benutzer klickt den hinzufügen-Button	Der Nutzer wird in diesem Kurs hinzugefügt. Ein Alert Window wird geöffnet um zu zeigen, dass der Prozess erfolgreich war und dass der Nutzer jetzt ein Teilnehmer von diesem Kurs ist	✓
11	Der Benutzer klickt den schließen-Button	Die searchStudent Seite wird geschlossen. Die Veranstaltung View Seite wird wieder geöffnet	✓
Nachbedingung(en)	Der Lehrende kann Lernmaterial hinzufügen und Löschen. Er kann Teilnehmer in einem Kurs hinzufügen.		
Testurteil	Test bestanden.		

Datum	20.05.2021		
Tester	Frederik Gnass		
SW-Version	V 0.1.2		
Vorbedingung(en)	Nutzer soll sein eigenes Profilbild auswählen		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Nutzer drückt auf den Registrieren Button	Das Registrierungsfenster öffnet sich.	✓
2	Der Nutzer wählt den Button "Bild hochladen"	Der File-Explorer/Finder sollte sich öffnen und der User kann nach einem Profilbild in JPG- oder PNG-Format suchen	✓
3	Der Nutzer wählt ein Bild aus und drückt auf "öffnen"	Das Profilbild sollte dem Nutzer sofort angezeigt werden.	✓
4	Der Nutzer drückt auf "weiter"	Das Profilbild sollte in der Stage gespeichert werden.	✓
Nachbedingung(en)	Das Profilbild wird beim Profil-Fenster und Edit-Profilfenster gesetzt.		
Testurteil	Test bestanden.		

Datum	25.05.2021		
Tester	Thorben Kronewald-Lamken		
SW-Version	V 0.1.2		
Vorbedin-gung(en)	Lehrender möchte eine einzelne Lehrveranstaltung hinzufügen		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Lehrende gibt den Namen des hinzuzufügenden Kurses ein	Name wird lokal für die Veranstaltung abgespeichert	✓
1	Der Lehrende klickt auf den Dropdown-Pfeil der Semester Choicebox	Das Dropdown Menü öffnet sich und das gewünschte Semester für den Kurs kann ausgewählt werden.	✓
2	Der Lehrende klickt auf den Dropdown-Pfeil der Veranstaltungsart Choicebox	Das Dropdown Menü öffnet sich und es kann zwischen den verschiedenen Veranstaltungsarten gewählt werden.	✓
3	Der Lehrende klickt auf den "Kurs hinzufügen" Button	Die Auswahlen der Choiceboxen und der ausgewählte Name werden als lokales Kursobjekt gespeichert, es wird überprüft ob der Kurs schon existiert und falls das nicht der Fall ist, wird er der Liste aller Lehrveranstaltungen hinzugefügt.	✓
Nachbe-dingung(en)	Der Kurs erscheint in der Liste aller Lehrveranstaltungen		
Testurteil	Test bestanden.		

Zyklus II

Id	Artefakt	Art des Artefakts	Verantwortlicher	Status
1.	Projektgruppen			
1.1	Projektgruppe erstellen		Hana	Fertig
1.1.1	Projektgruppe erstellen	User Story		Fertig
1.1.2	Projektgruppe	Papierprototyp		Fertig
1.1.3	Projektgruppe erstellen	MSC		Fertig
1.2	Mitglieder hinzufügen		Thorben	
1.2.1	Mitglieder hinzufügen	User Story		
1.2.2	Mitglieder hinzufügen	Prototyp		
1.2.3	Mitglieder hinzufügen	MSC		
1.3	Chatraum		Noemi	Fertig
1.3.1	Chatraum als Teil vom Projektgruppen - Fenster	Papierprototyp		Fertig
1.3.2	User Story: Chatraum	User Story		Fertig
1.3.3	BMSC: Chatraum	MSC		Fertig
1.4	ToDo Liste		Thorben	
1.4.1	ToDos einfügen und zuteilen	User Story		
1.4.2	ToDo Liste	MSC		
1.5	Dateien austauschen		Thorben	
1.5.1	Datei austauschen	User Story		
1.5.2	Dateien austauschen	MSC		
2.	Kommunikation			
2.1	Privater Chat		Noemi	Fertig
2.1.1	Chatfenster	Papierprototyp		Fertig
2.1.2	User Story: Chatfunktion	User Story		Fertig

2.1.3	BMSC: Chatfunktion	MSC		Fertig
2.2	Freundschaftsanfragen		Hana	Fertig
2.2.1	Freundesliste	Papierprototyp		Fertig
2.2.2	Freundschaftsanfragen	User Story		Fertig
2.2.3	Freundschaftsanfragen	MSC		Fertig
3.	Quiz (Multiple Choice)		Fred	
3.1	Quiz-Fenster			Erl.
3.1.1	Einzelnes Quiz erstellen	Papierprototyp		Erl.
3.1.1.1	User Story: Einzelnes Quiz erstellen	User Story		Erl.
3.1.1.2	BMSC: Einzelnes Quiz erstellen	MSC		Erl.
3.1.2	Quiz aus XML-Datei auslesen lassen	Papierprototyp		Erl.
3.1.2.1	User Story: Quiz aus XML-Datei auslesen lassen	User Story		Erl.
3.1.2.2	BMSC: Quiz aus XML-Datei auslesen lassen	MSC		Erl.
3.1.3	Quiz-Fenster (Student)	Papierprototyp		Erl.
3.1.3.1	User Story: Quiz-Fenster (Student)	User Story		Erl.
3.1.3.2	BMSC: Quiz-Fenster (Student)	MSC		Erl.
3.2	Quiz-Feedback	Papierprototyp		Erl.
3.2.1	User Story: Quiz-Feedback	User Story		Erl.
3.2.2	BMSC: Quiz-Feedback	MSC		Erl.
3.3	Statistik-Fenster	Papierprototyp		Erl.
3.3.1	User Story: Statistik	User Story		Erl.
3.3.2	BMSC: Statistik berechnen	MSC		Erl.
4.	Kalenderfunktionen		Vera	
4.1	Kalender einsehen (User)			Erl.
4.1.1	Kalender einsehen User Story	User Story		Erl.
4.1.2	Kalender View	Papierprototyp		Erl.
	Kalender einsehen (User) MSC	MSC		Erl.
4.2	Event hinzufügen (Lehrender)			Erl.

4.2.1	Event hinzufügen User Story	User Story		Erl.
4.2.2	Event hinzufügen View	Papierprototyp		Erl.
4.2.3	Event hinzufügen MSC	MSC		Erl.
4.3	Reminder zu Event hinzufügen (Lehrender)			Erl.
4.3.1	Reminder zu Event hinzufügen User Story	User Story		Erl.
4.3.2	Reminder zu Event hinzufügen View	Papierprototyp		Erl.
4.3.3	Reminder zu Event hinzufügen MSC	MSC		Erl.
4.4	Reminder erhalten (Student) (Pop-up oder Email)			Erl.
4.4.1	Reminder erhalten (Pop-up) User Story	User Story		Erl.
4.4.2	Reminder erhalten (Email) User Story	User Story		Erl.
4.4.3	Reminder erhalten (Pop-up) View	Papierprototyp		Erl.
4.4.4	Reminder erhalten (Pop-up) MSC	MSC		Erl.
4.4.5	Reminder erhalten (Email) MSC	MSC		Erl.
4.5	(Testfunktion) Aktuelles Datum im Programm ändern (User)			Erl.
5.	Zwei-Faktor-Authentisierung		Kuri	Fertig
5.1	User-Story	User-Story		
5.2	Auth.-Fesnter	Papierprototyp		
5.3	2-Faktor- Auth. MSC	MSC		

User-Stories

User Story-ID	1.1.1
User Story-Beschreibung	Als Nutzer möchte ich eine Projektgruppe erstellen können, als eine Möglichkeit für die Teilnehmer einer Gruppe, um sich auszutauschen und die Aufgaben über eine To-Do-Liste zu verteilen
Geschätzter Realisierungsaufwand	5 Tage
Priorität	hoch
Autor(en)	Thorben Kronewald-Lamken
Abhängigkeiten zu anderen User Stories	

User Story-ID	1.2.1
User Story-Beschreibung	Als Lehrer möchte ich Studenten in unseren Projektgruppen hinzufügen, damit wir miteinander leichter kommunizieren können.
Geschätzter Realisierungsaufwand	2 Tage
Priorität	mittel
Autor(en)	Hana Habibi
Abhängigkeiten zu anderen User Stories	1.1.1

User Story-ID	1.3.2
User Story-Beschreibung	Als Nutzer möchten wir Chaträume in unseren Projektgruppen zur Verfügung stehen haben, damit wir uns mit anderen Mitgliedern dieser Projektgruppe austauschen können.
Geschätzter Realisierungsaufwand	2 Wochen
Priorität	hoch
Autor(en)	Noemi Kallweit
Abhängigkeiten zu anderen User Stories	

User Story-ID	1.4.1
User Story-Beschreibung	Als Nutzer möchte ich zu erledigende Aufgaben der To-Do-Liste hinzufügen und die einzelnen Aufgaben verteilen können, um eine klare Rollenverteilung zu gewährleisten
Geschätzter Realisierungsaufwand	3 Tage
Priorität	Mittel
Autor(en)	Thorben Kronewald-Lamken
Abhängigkeiten zu anderen User Stories	1.1.1

User Story-ID	1.5.1
User Story-Beschreibung	Als Nutzer möchte ich mit den anderen Mitgliedern der Gruppe Dateien austauschen, damit wir die Projekte machen können
Geschätzter Realisierungsaufwand	4 Tage
Priorität	mittel
Autor(en)	Hana Habibi
Abhängigkeiten zu anderen User Stories	1.1.1, 1.3.2

User Story-ID	2.1.2
User Story-Beschreibung	Als Nutzer möchte ich mit anderen Nutzern einen privaten Chat starten können, damit wir eine einfache Möglichkeit haben uns auszutauschen.
Geschätzter Realisierungsaufwand	2 Wochen
Priorität	hoch
Autor(en)	Noemi Kallweit
Abhängigkeiten zu anderen User Stories	

User Story-ID	2.2.2
User Story-Beschreibung	Als Nutzer möchte ich Freundschaftsanfragen senden, damit ich mit anderen Nutzern chatten kann
Geschätzter Realisierungsaufwand	2 Tage
Priorität	hoch
Autor(en)	Hana Habibi
Abhängigkeiten zu anderen User Stories	

User Story-ID	3.1.1.1
User Story-Beschreibung	Als Nutzer möchte ich ein Quiz erstellen können, damit sich die anderen Nutzer testen können.
Geschätzter Realisierungsaufwand	2 Tage
Priorität	hoch
Autor(en)	Frederik Gnass
Abhängigkeiten zu anderen User Stories	

User Story-ID	3.1.2.1
User Story-Beschreibung	Als Nutzer möchte ich ein Quiz aus einer XML-Datei auslesen können, damit ich nicht jede Frage eintippen muss.
Geschätzter Realisierungsaufwand	2 Tage
Priorität	hoch
Autor(en)	Frederik Gnass
Abhängigkeiten zu anderen User Stories	

User Story-ID	3.1.3.1
User Story-Beschreibung	Als Nutzer möchte ich die Quizze bearbeiten können, damit ich mich testen kann.
Geschätzter Realisierungsaufwand	2 Tage
Priorität	hoch
Autor(en)	Frederik Gnass
Abhängigkeiten zu anderen User Stories	

User Story-ID	3.2.1
User Story-Beschreibung	Als Nutzer möchte ich am Ende des Quizzes ein Feedback bekommen, welche Antworten richtig und falsch waren, damit ich mein Wissensstand überprüfen kann.
Geschätzter Realisierungsaufwand	2 Tage
Priorität	hoch
Autor(en)	Frederik Gnass
Abhängigkeiten zu anderen User Stories	

User Story-ID	3.3.1
User Story-Beschreibung	Als Nutzer möchte ich eine Statistik des Quizzes haben, damit ich mein Quiz und Wissensstand besser einschätzen kann.
Geschätzter Realisierungsaufwand	2 Tage
Priorität	hoch
Autor(en)	Frederik Gnass
Abhängigkeiten zu anderen User Stories	

User Story-ID	4.1.1
User Story-Beschreibung	Als User möchte ich auf einen Kalender zugreifen, sodass ich eine Übersicht über anstehende Termine habe, die mich betreffen
Geschätzter Realisierungsaufwand	2 Tage
Priorität	Hoch
Autor(en)	Vera Mitschulat
Abhängigkeiten zu anderen User Stories	4.2.1

User Story-ID	4.2.1
User Story-Beschreibung	Als Lehrender möchte ich Ereignisse (für eine bestimmte Veranstaltung) in den Kalender hinzufügen, um betreffende Personen an wichtige Ereignisse zu erinnern
Geschätzter Realisierungsaufwand	1 Tag
Priorität	Hoch
Autor(en)	Vera Mitschulat
Abhängigkeiten zu anderen User Stories	4.3.1 , 4.1.1.

User Story-ID	4.3.1
User Story-Beschreibung	Als Lehrender möchte ich zu Ereignissen Erinnerungen hinzufügen können, damit betreffende Personen wichtige Ereignisse nicht vergessen
Geschätzter Realisierungsaufwand	<1 Tag
Priorität	Mittel
Autor(en)	Vera Mitschulat
Abhängigkeiten zu anderen User Stories	4.2.1

User Story-ID	4.4.1
User Story-Beschreibung	Als Student möchte ich eine Erinnerung als Pop-Up im Programm erhalten, um mich an wichtige Ereignisse zu erinnern
Geschätzter Realisierungsaufwand	<1 Tag
Priorität	Mittel
Autor(en)	Vera Mitschulat
Abhängigkeiten zu anderen User Stories	4.3.1 , 4.2.1

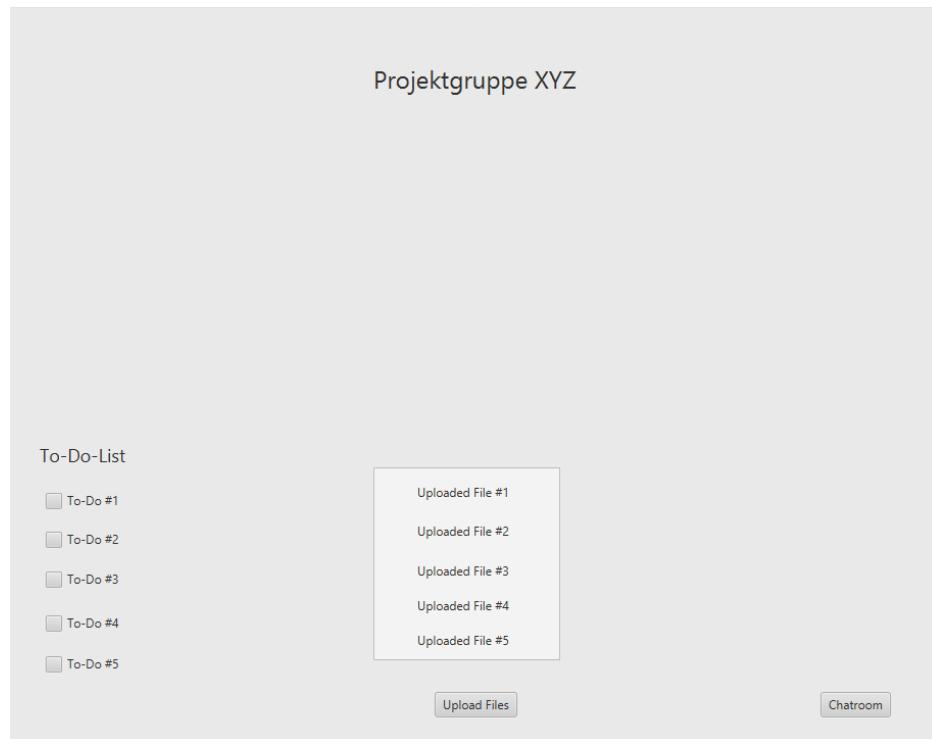
User Story-ID	4.4.2
User Story-Beschreibung	Als Student möchte ich eine Erinnerung als E-Mail erhalten, um mich an wichtige Ereignisse zu erinnern
Geschätzter Realisierungsaufwand	<1 Tag
Priorität	Mittel
Autor(en)	Vera Mitschulat
Abhängigkeiten zu anderen User Stories	4.3.1 , 4.2.1

User Story-ID	5.1
User Story-Beschreibung	Als User möchte ich per E-Mail ein Code gesendet bekommen um mich zu verifizieren.
Geschätzter Realisierungsaufwand	1 Tag
Priorität	Mittel
Autor(en)	Kurinchinilavan Thangeswaran
Abhängigkeiten zu anderen User Stories	

Papierprototypen

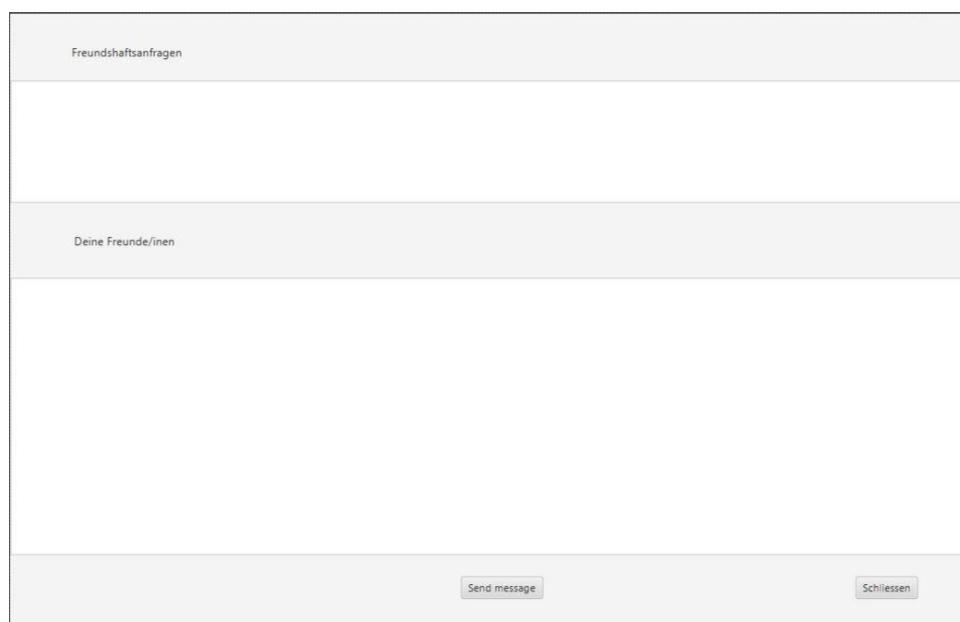
Ansicht Projektgruppe

Die Projektgruppe verfügt über einen Titel, einen Chatraum, eine To-Do-Liste und über die Möglichkeit Daten auszutauschen.



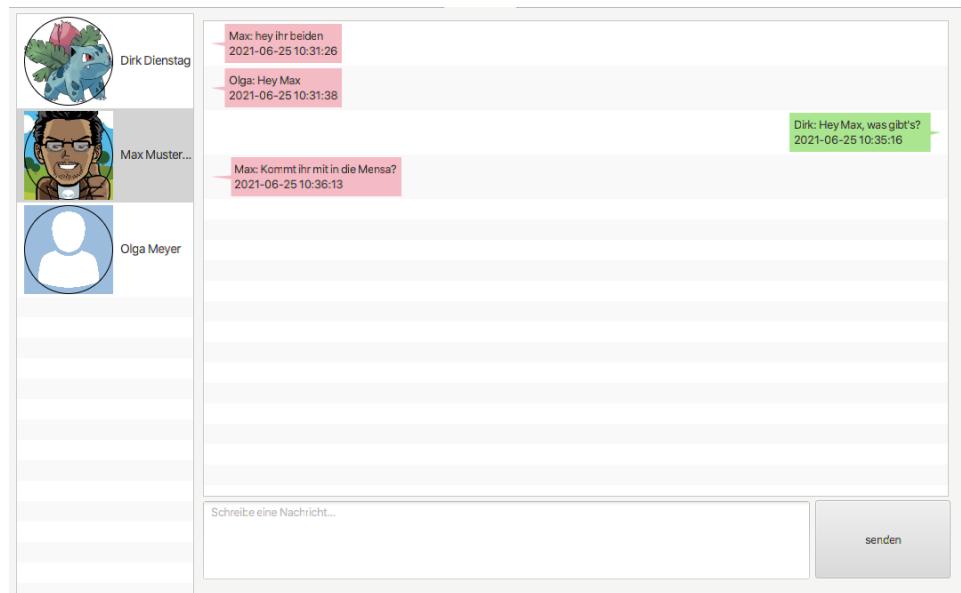
Ansicht Freundschaftsliste

Die Ansicht hat die Liste von Freunden. Freunde können gesucht werden und wenn man ein Freund auswählt und Nachricht senden kann mit dem chatten



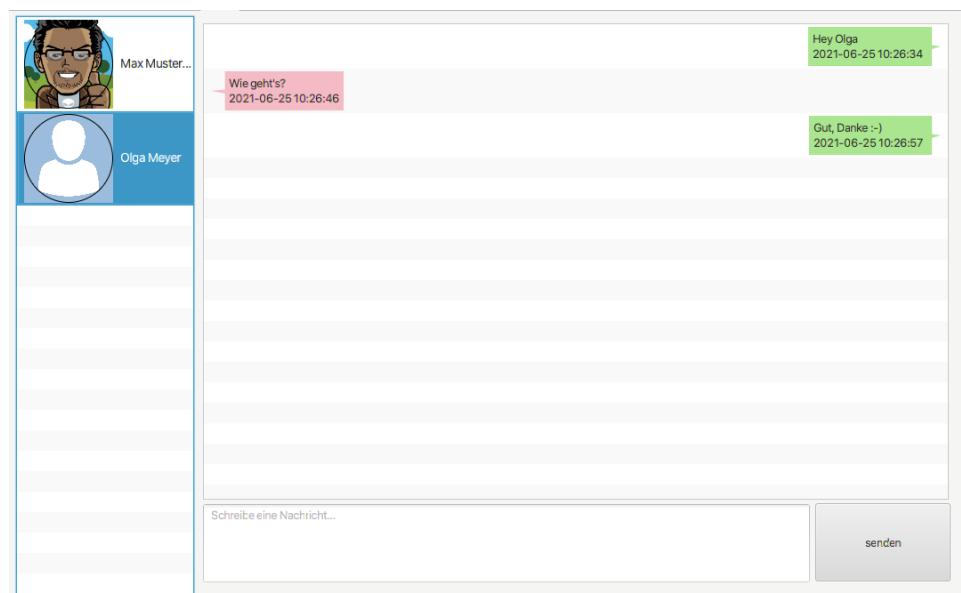
Ansicht Gruppenchat

Die Ansicht sieht ein Nutzer, wenn er einen Gruppenchat öffnet. Es wird immer kenntlich gemacht welcher Teilnehmer des Chats die Nachricht geschrieben hat. In dem Textfeld können Nachrichten geschrieben werden und über den “send” Button gesendet werden. Links werden die Mitglieder des Gruppenchats angezeigt.



Ansicht privater Chat

Diese Ansicht hat ein Nutzer, wenn er das Chatfenster öffnet. Links sind Chats mit anderen Nutzern aufgeführt, mit denen der Nutzer bereits gesprochen hat. Rechts wird der aktuell ausgewählte Chatverlauf angezeigt. Über das Textfeld können Nachrichten geschrieben und über den “send” Button verschickt werden. Wenn auf einen anderen Nutzer geklickt wird, wird der zugehörige Chatverlauf geladen.



Gruppenmitglieder hinzufügen Ansicht

In dieser Ansicht kannst du ein Nutzer mit seiner/ihr Name/Matrikelnummer suchen. Die Liste der gefundenen Nutzer wird im ViewList gezeigt. Von da kann man Nutzern in der Gruppe hinzufügen. Wenn da ein Fehler gibt oder der Nutzer erfolgreich hinzugefügt ist, dann wird das System uns Bescheid. Am Ende schließt man die Seite durch den Schließen-Button



Einzelnes Quiz erstellen Ansicht

In dieser Ansicht kannst du als Lehrender ein Quiz erstellen. Du kannst mehrere Antwortmöglichkeiten generieren und auch deine Frage im oberen Feld stellen. Auf der linken Seite erkennt man, wie viele Fragen schon erstellt worden sind.

Frage 1
Frage 2
Frage 3

Frage 3

Scotty, der Ingenieur des Raumschiffs Enterprise, war bei den "Ausflügen" meist nicht dabei, weil er ...?

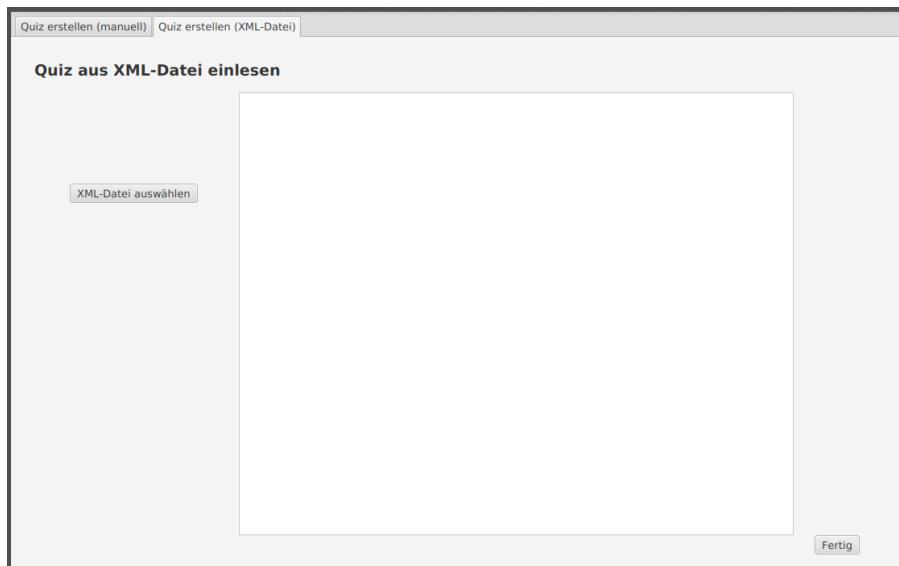
angestellte beamte

selbstständige vorgesetzte

Zurück Weitere Frage

Quiz aus XML-Datei auslesen Ansicht

In dieser Ansicht kannst du als Lehrender das Quiz aus einer XML-Datei einlesen lassen. Auf der rechten Seite bekommst du einen kleinen Überblick, welche Fragen korrekt formatiert sind und ob diese erfolgreich eingelesen wurden. Auf dem Button kannst du eine XML-Datei aus deinem Datei-Browser auswählen.



Quiz-Fenster Student Ansicht

In dieser Ansicht kannst du als Nutzer das Quiz beantworten. Auf der linken Seite findet man, bei welcher Frage man ist und wie viele Fragen das Quiz insgesamt hat. Auf der rechten Seite sieht man die Fragen und kann auch seine Antwortmöglichkeit abgeben und zur nächsten Frage springen.

The screenshot shows a window titled 'Quiz'. On the left side, a vertical list shows the current question 'Frage 3' and the total number of questions: 'Frage 1', 'Frage 2', 'Frage 3', 'Frage 4', 'Frage 5', and 'Frage 6'. The main area displays Question 3: 'Scotty, der Ingenieur des Raumschiffs Enterprise, war bei den "Ausflügen" meist nicht dabei, weil er ...?'. Below the question are four answer options: 'angestellte' (with an empty checkbox), 'beamte' (with an empty checkbox), 'selbstständige' (with an empty checkbox), and 'vorgesetzte' (with an empty checkbox). At the bottom left is a 'Zurück' (Back) button, and at the bottom right is a 'Weiter' (Next) button.

Quiz-Feedback Ansicht

In dieser Ansicht kannst du als Nutzer deine Gesamtwertung des gerade absolvierten Quizes anschauen. Man bekommt eine Staffelung, die zeigt, welche Fragen korrekt beantwortet wurden und welche Fragen falsch beantwortet wurden.

The screenshot shows a 'Quiz Feedback' window. At the top, it displays 'Richtig beantwortet (6/10)' and '60%' next to 'Bestanden:' and 'ja'. Below this, a list of 10 questions is shown, each with its status: 'Frage 1: Korrekt', 'Frage 2: Falsch', 'Frage 3: Korrekt', 'Frage 4: Korrekt', 'Frage 5: Falsch', 'Frage 6: Falsch', 'Frage 7: Korrekt', 'Frage 8: Korrekt', 'Frage 9: Falsch', and 'Frage 10: Korrekt'.

Statistik Ansicht

In dieser Ansicht kannst du als Lehrender die Statistik zu deinem Quiz verfolgen. Es werden einige Daten angezeigt.

The screenshot shows a 'Quiz Ergebnisse' window. It displays two main sections: 'Anzahl der Versuche:' and 'Anzahl korrekte Antworten pro Frage:'.

Anzahl der Versuche:

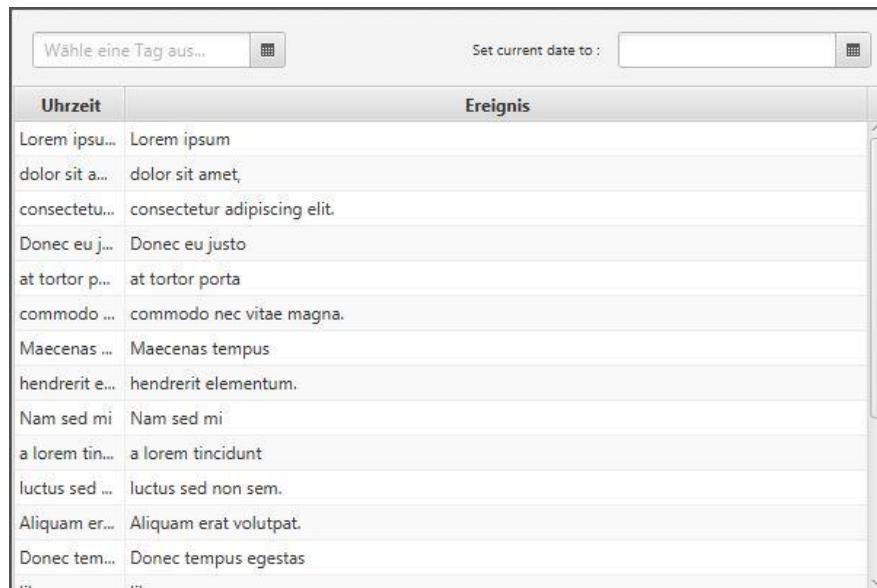
- Evi Noel: 5
- Abdullo Solo: 7
- Alija Bunschoten: 3
- Bilal Albrechtsson: 4
- Katharyn Abspoel: 1

Anzahl korrekte Antworten pro Frage:

- Frage 1: 9/20 (45%)
- Frage 2: 12/20 (60%)
- Frage 3: 18/20 (90%)
- Frage 4: 14/20 (70%)
- Frage 5: 10/20 (50%)

Kalender Ansicht

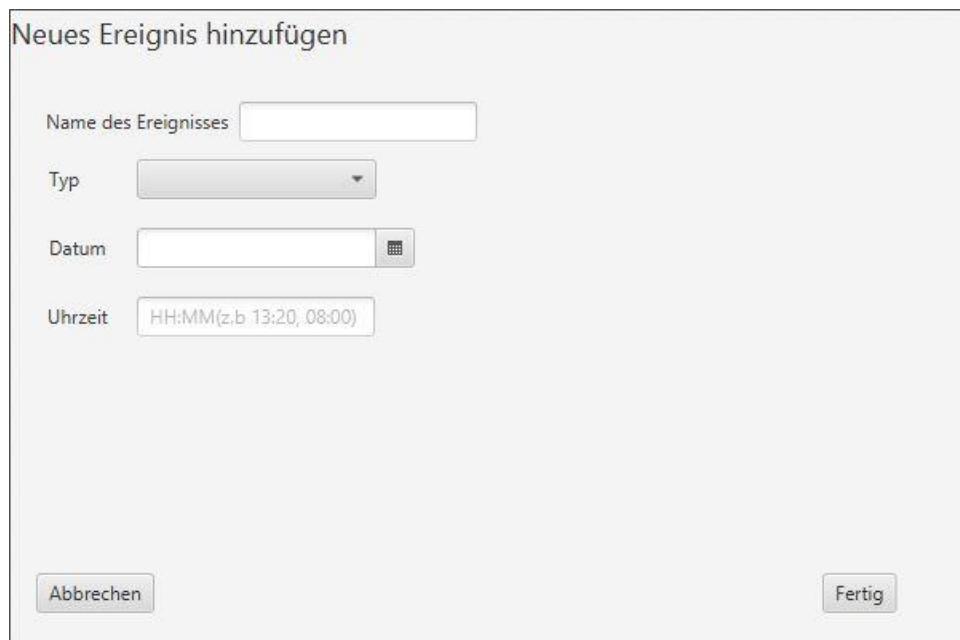
Die Kalender Ansicht aller Nutzer. Über die linken Datepicker lässt sich der Kalendereintrag eines gewünschten Tages aufrufen. In der Tabelle werden die Uhrzeit des Ereignisses, sowie der Name angezeigt. Bei dem rechten Datepicker handelt es sich um eine Testfunktion um das aktuelle Datum des Systems beliebig zusetzen. Sie ist nur für Vorführzwecke gedacht und ist nicht in der finalen Version enthalten.



Uhrzeit	Ereignis
Lorem ipsum...	Lorem ipsum
dolor sit a...	dolor sit amet,
consectetu...	consectetur adipiscing elit
Donec eu j...	Donec eu justo
at tortor p...	at tortor porta
commodo ...	commodo nec vitae magna.
Maecenas ...	Maecenas tempus
henderit e...	henderit elementum.
Nam sed mi	Nam sed mi
a lorem tin...	a lorem tincidunt
luctus sed ...	luctus sed non sem.
Aliquam er...	Aliquam erat volutpat.
Donec tem...	Donec tempus egestas
...	...

Ansicht vom Hinzufügen eines Ereignisses

Dieses Fenster wird der Lehrperson angezeigt, wenn sie ein neues Ereignis in einer Veranstaltung erstellt. Die Lehrperson muss den Namen, Typ (Vorlesung, Klausur, Abgabefrist, Testat oder sonstiges angeben), das Datum und die Uhrzeit im gegebenen Format (HH:MM) angeben. Mit dem Klick des Fertig Buttons wird das Ereignis finalisiert und hinzugefügt. Mit Abbrechen wird die Erstellung des Ereignisses gelöscht.



The dialog box contains the following fields:

- Name des Ereignisses: Input field
- Typ: Drop-down menu
- Datum: Input field with calendar icon
- Uhrzeit: Input field with time format (HH:MM) placeholder "HH:MM(z.B 13:20, 08:00)"
- Buttons at the bottom: "Abbrechen" (Cancel) and "Fertig" (Done)

Ansicht vom Hinzufügen eines Reminders

Dieses Fenster wird der Lehrperson angezeigt, wenn sie einen neuen Reminder in einem Ereignis erstellt. Die Lehrperson muss den Namen, das Datum, die Uhrzeit (im gegebenen Format (HH:MM)) und die Art des Reminders (Pop-up oder E-Mail) angeben. Mit dem Klick des Fertig Buttons wird der Reminder finalisiert und hinzugefügt. Mit Abbrechen wird die Erstellung des Reminders gelöscht.

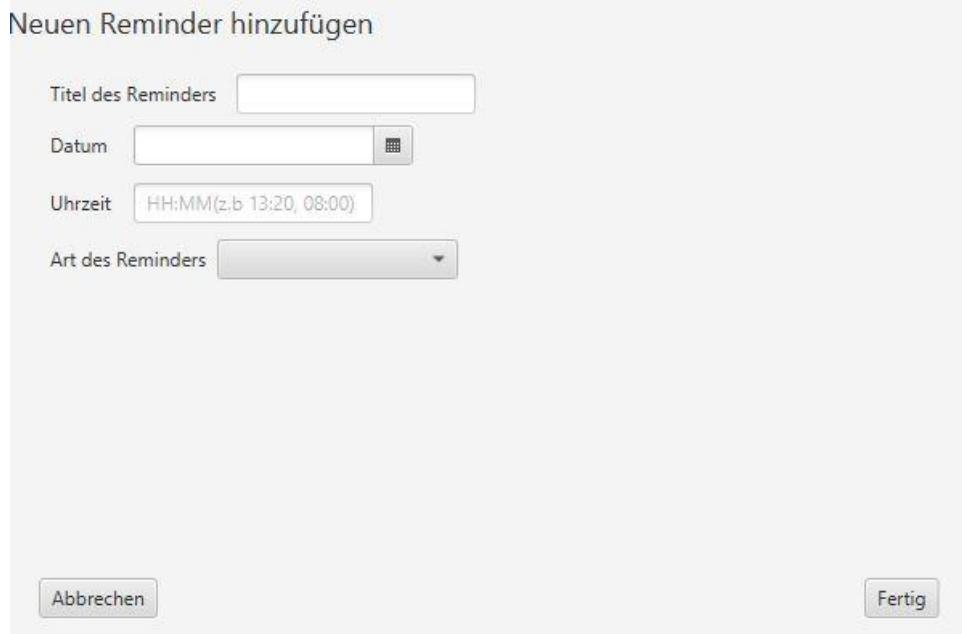
Neuen Reminder hinzufügen

Titel des Reminders

Datum

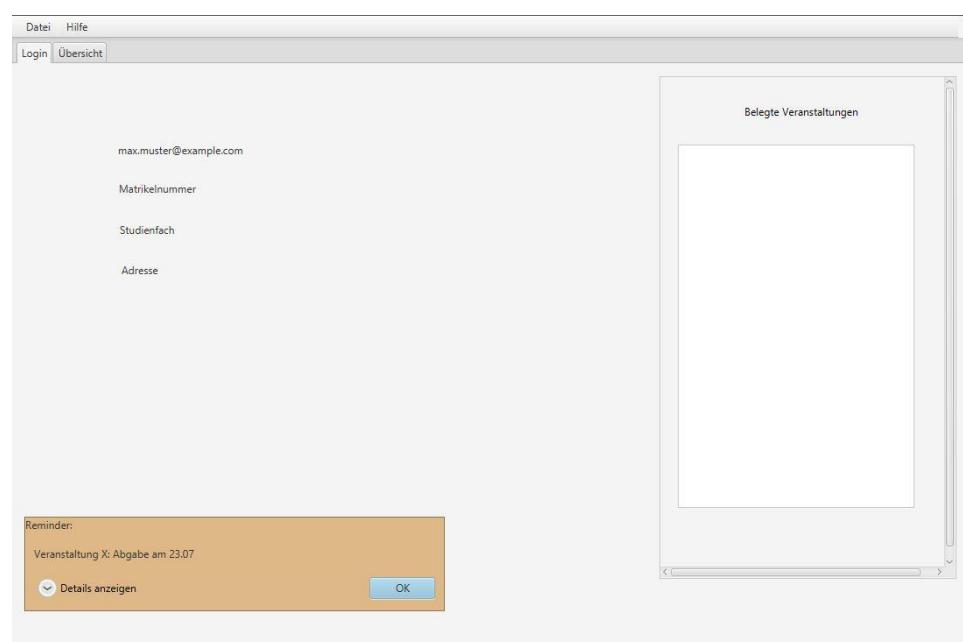
Uhrzeit HH:MM(z.B 13:20, 08:00)

Art des Reminders



Ansicht eines Pop-Up Reminder (Student)

Wenn man als Student einen Reminder bekommt wird dieser unten links in der Ecke des Fensters angezeigt. Mit einem Klick auf den OK Button wird der Reminder entfernt. Mit einem Klick auf Details wird das Fenster des Ereignisses geöffnet.



Ereignis Ansicht

Wenn ein Ereignis geöffnet wird, wird dieses Fenster angezeigt. Der Name der Veranstaltung zu dem das Ereignis gehört wird angezeigt. Der Name, das Datum und die Uhrzeit des Ereignisses werden hier angezeigt. Zudem ist hier die Liste aller dazugehörigen Reminder sowie. Der Button für Lehrende um neue Reminder hinzuzufügen

Ereignis
Veranstaltung: Name der Veranstaltung
Name des Ereignis: Abgabe der Veranstaltung X
Datum: XX.XX.XXXX
Uhrzeit: HH:MM
Reminder hinzufügen
Liste aller Reminder:
Ereignis löschen

Reminder Ansicht

Wenn ein Reminder geöffnet wird, wird diese Ansicht angezeigt. Der Name der Veranstaltung zu dem das Reminder gehört wird angezeigt. Der Name, das Datum und die Uhrzeit des Ereignisses zu dem der Reminder gehört werden hier angezeigt. Zudem wird das Datum, die Uhrzeit und der Typ des Reminders angezeigt.

Reminder
Veranstaltung: Name der Veranstaltung
Daten des Ereignis: Abgabe der Veranstaltung X, am XX.XX.XXXX, um HH:MM
Datum des Reminders: XX.XX.XXXX
Uhrzeit des Reminders: HH:MM
Typ des Reminder: Pop-Up oder E-Mail
Reminder löschen

5.2

Zwei-Faktor-Auth.-Fenster

Hier wird im Textfeld der Code eingegeben, welcher vorher per E-Mail versandt wurde.

Mit dem “Bestätigen” Button wird die Eingabe mit dem Code überprüft der versandt wurde.

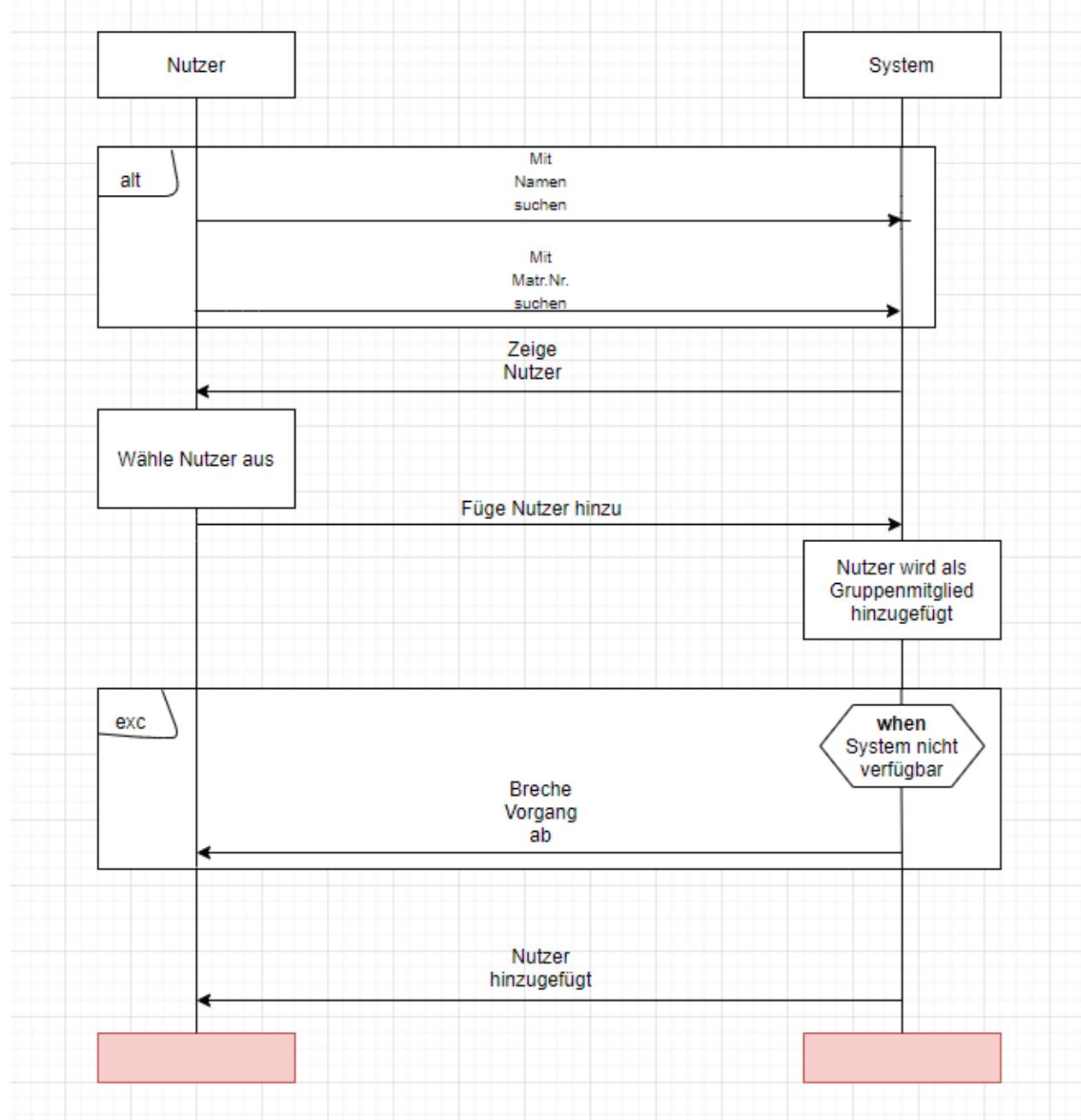
Der “Code Erneut senden” Button sendet einen neuen Code an den Nutzer.

Mit dem “Abbrechen” Button kehrt man zurück zum Start-Fenster

The screenshot shows a user interface for two-factor authentication. At the top, there is a label in German: "Geben sie den Code aus ihrer E-Mail ein:". Below this label is a large, empty rectangular input field. At the bottom of the window, there are three buttons arranged horizontally: "Abbrechen" (Cancel) on the left, "Code Erneut senden" (Send new code) in the middle, and "Bestätigen" (Confirm) on the right. The entire window has a dark gray border.

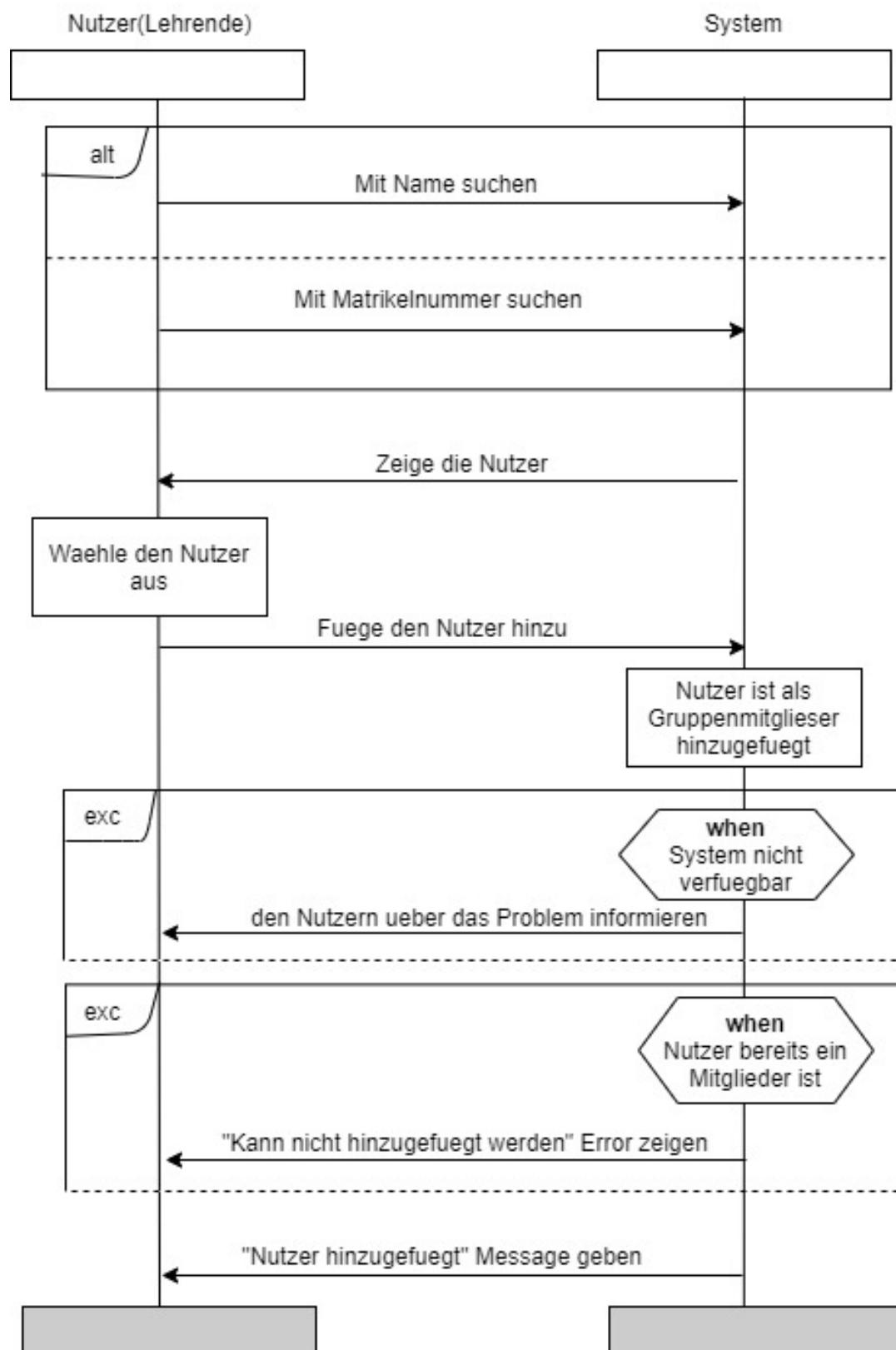
Szenarien (MSCs)

1.1.3 Projektgruppe erstellen



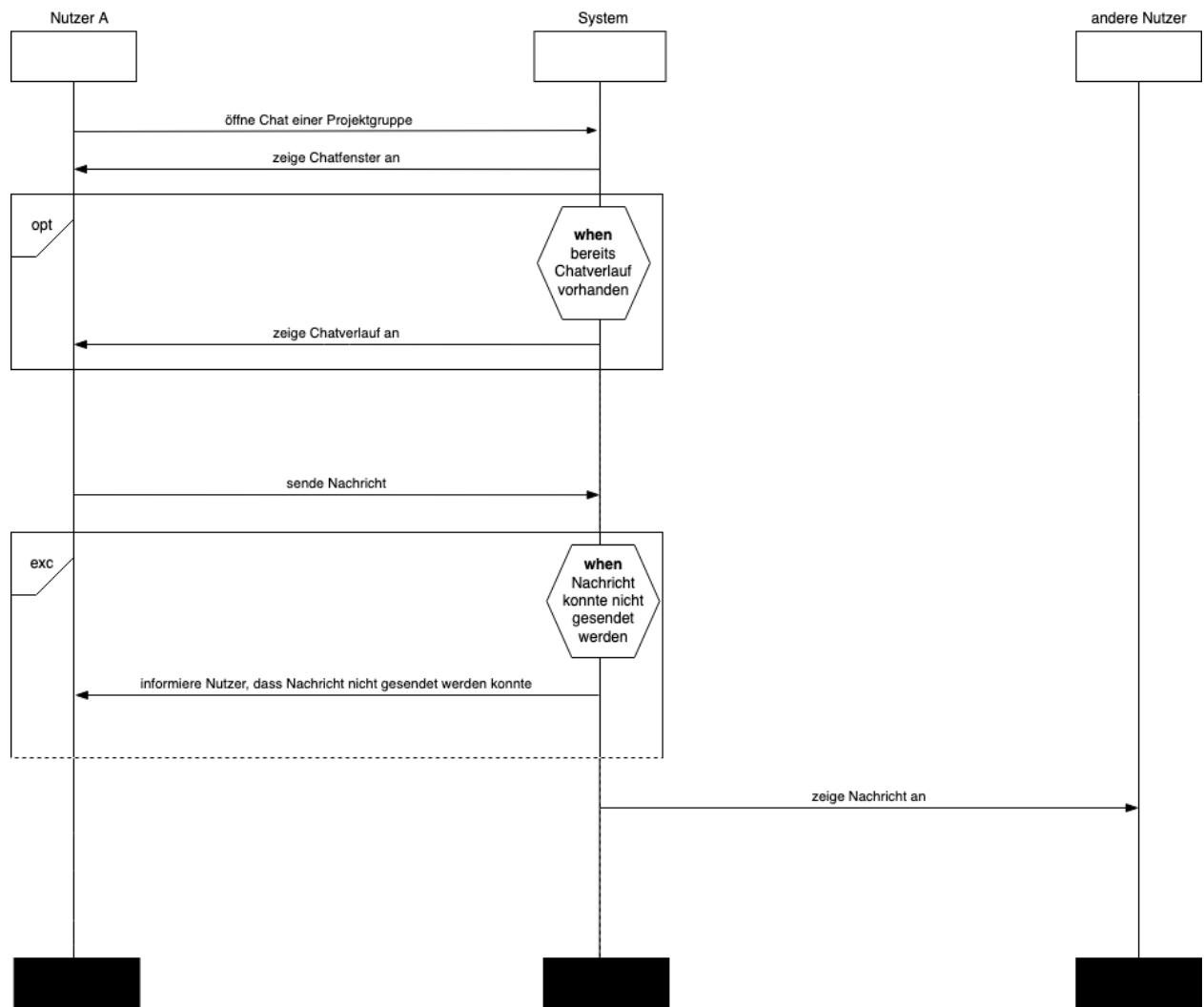
Der Nutzer sucht einen anderen Nutzer über sein/ihren Namen oder Matrikelnummer. Die Ergebnisse werden vom System gezeigt. Der Nutzer wählt einen Nutzer und fügt ihm/ihr hinzu. Wenn das System nicht verfügbar ist, dann wird der Vorgang abgebrochen. Sonst fügt das System den Nutzer hinzu.

1.2.3 Mitglieder hinzufügen



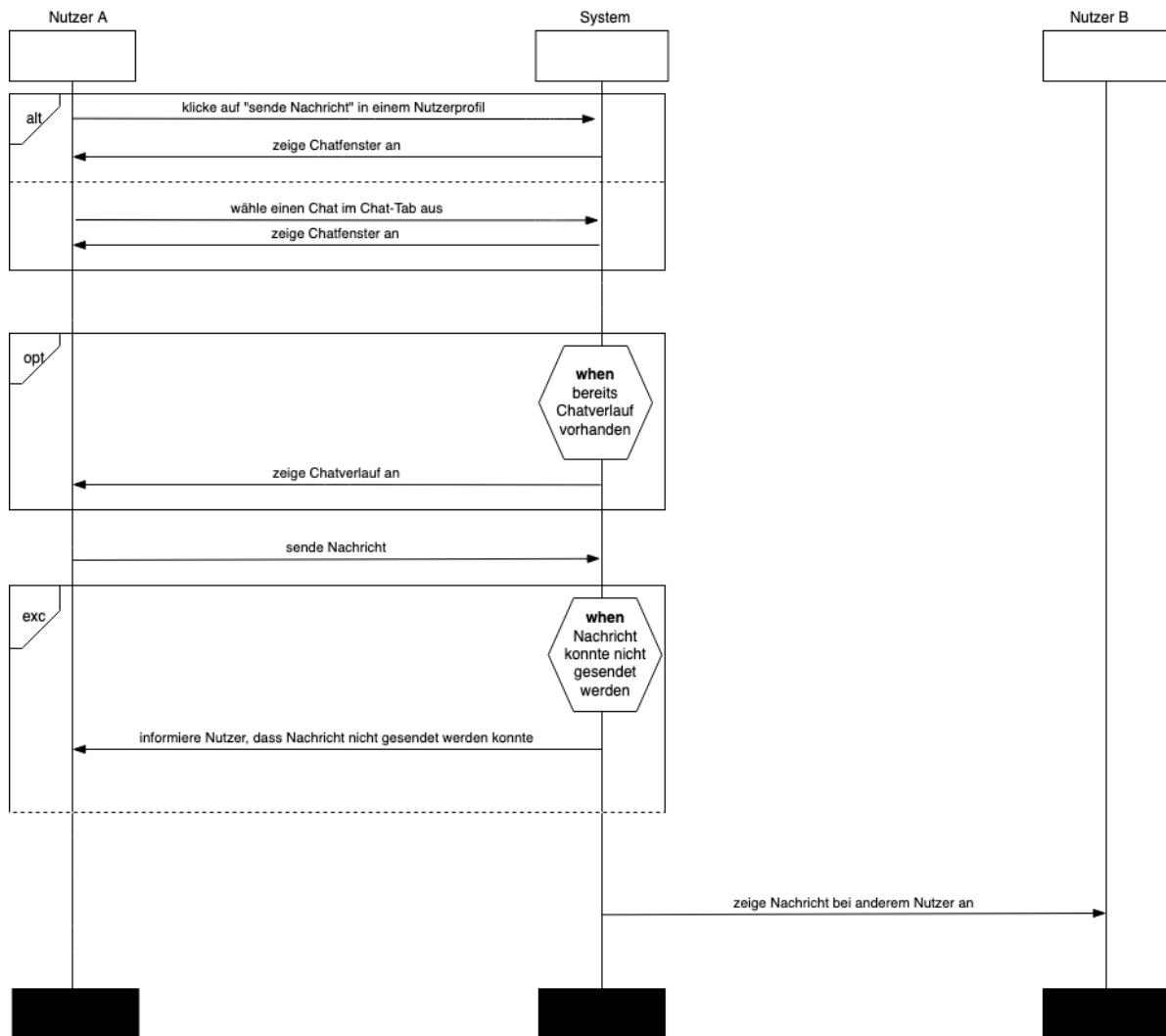
Als Lehrende darf man Mitglieder in einer Projektgruppe hinzufügen. Über den addStudentButton können Lehrende Studenten mit Namen oder Matrikelnummer zur Projektgruppe hinzufügen.

1.3.3 Gruppenchat in Projektgruppe



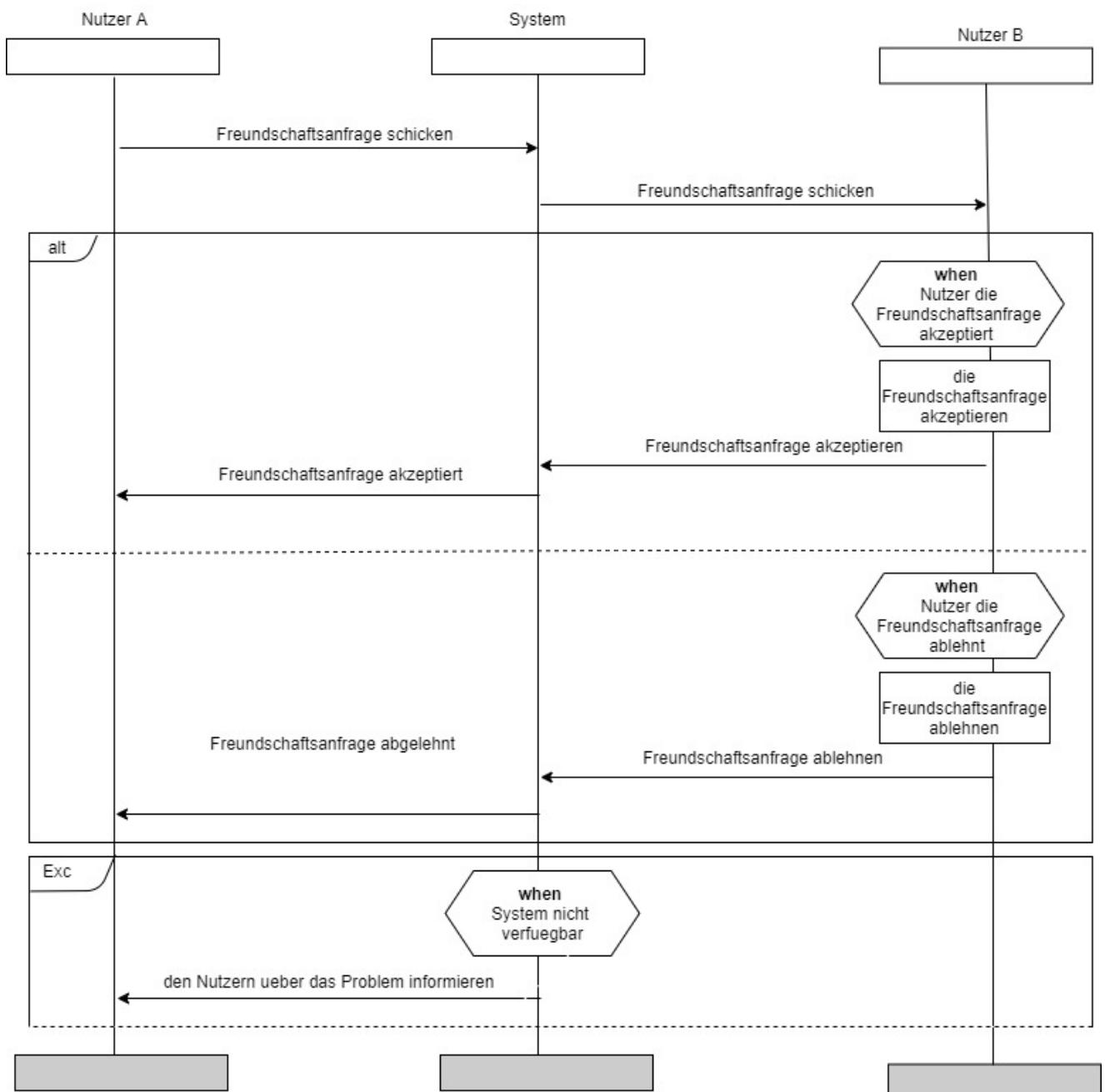
Ein Nutzer öffnet in der Projektgruppe den Gruppenchat. Der Chatverlauf wird geladen und der Nutzer kann Nachrichten schreiben und sie über den “Senden” Button abschicken. Diese werden bei den anderen Nutzern empfangen/angezeigt(impliziert, dass auch der sendende User Nachrichten von anderen Nutzern empfangen kann). Falls die Nachricht nicht versendet werden konnte wird der Sender informiert.

2.1.1 privater Chat



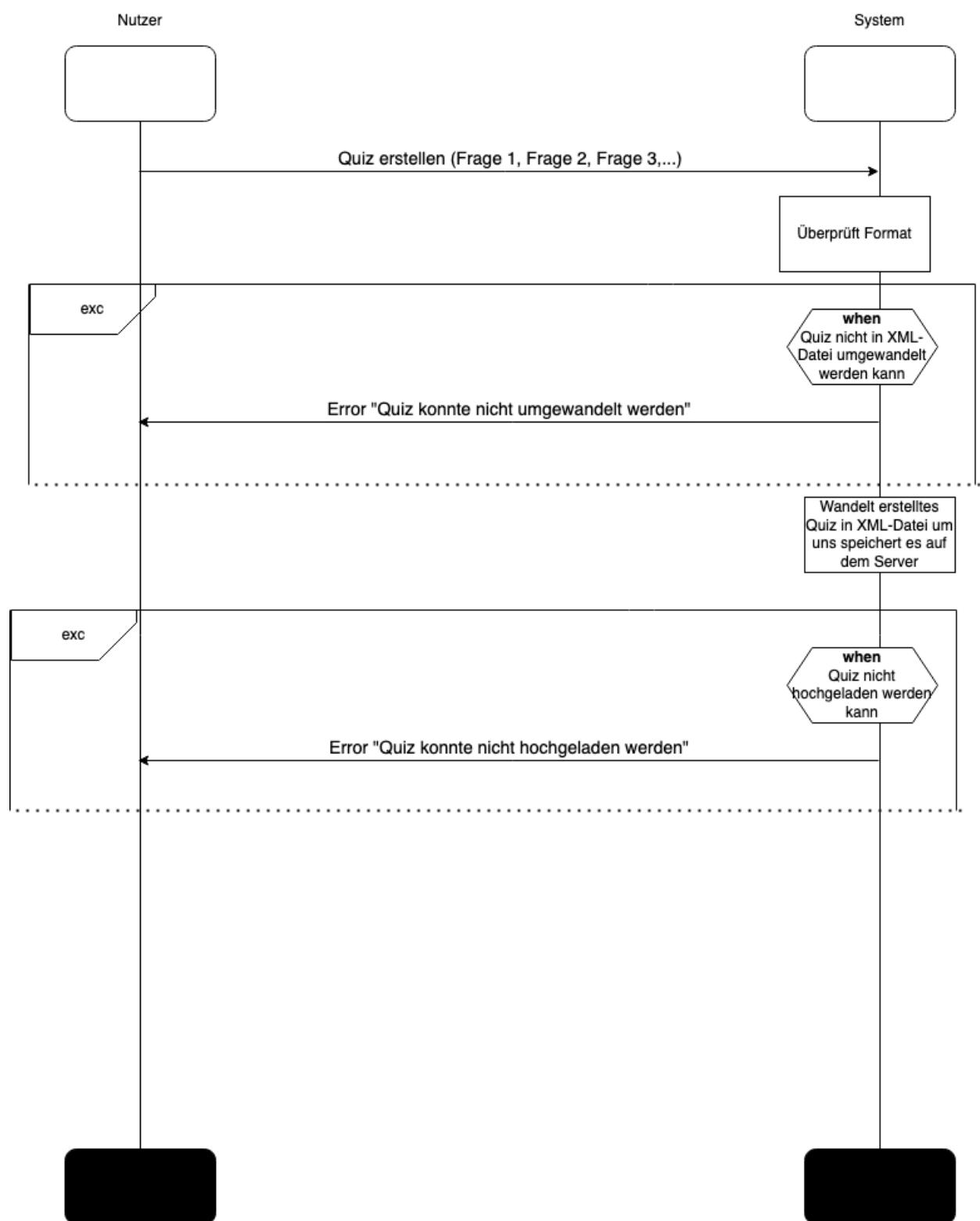
Sobald Freundschaftsanfragen ausgetauscht wurden, kann ein Nutzer einen privaten Chat mit einem Freund starten, indem er z. B auf den Namen des Freundes in der Kurs-Teilnehmerliste klickt. Falls schon ein Chatverlauf existiert wird dieser geladen. Der Nutzer kann Nachrichten schreiben und senden. Der andere Nutzer kann diese empfangen (impliziert, dass auch der sendende User Nachrichten von anderen Nutzern empfangen kann). Falls die Nachricht nicht versendet werden konnte wird der Sender informiert.

2.2.2 Freundschaftsanfrage akzeptieren



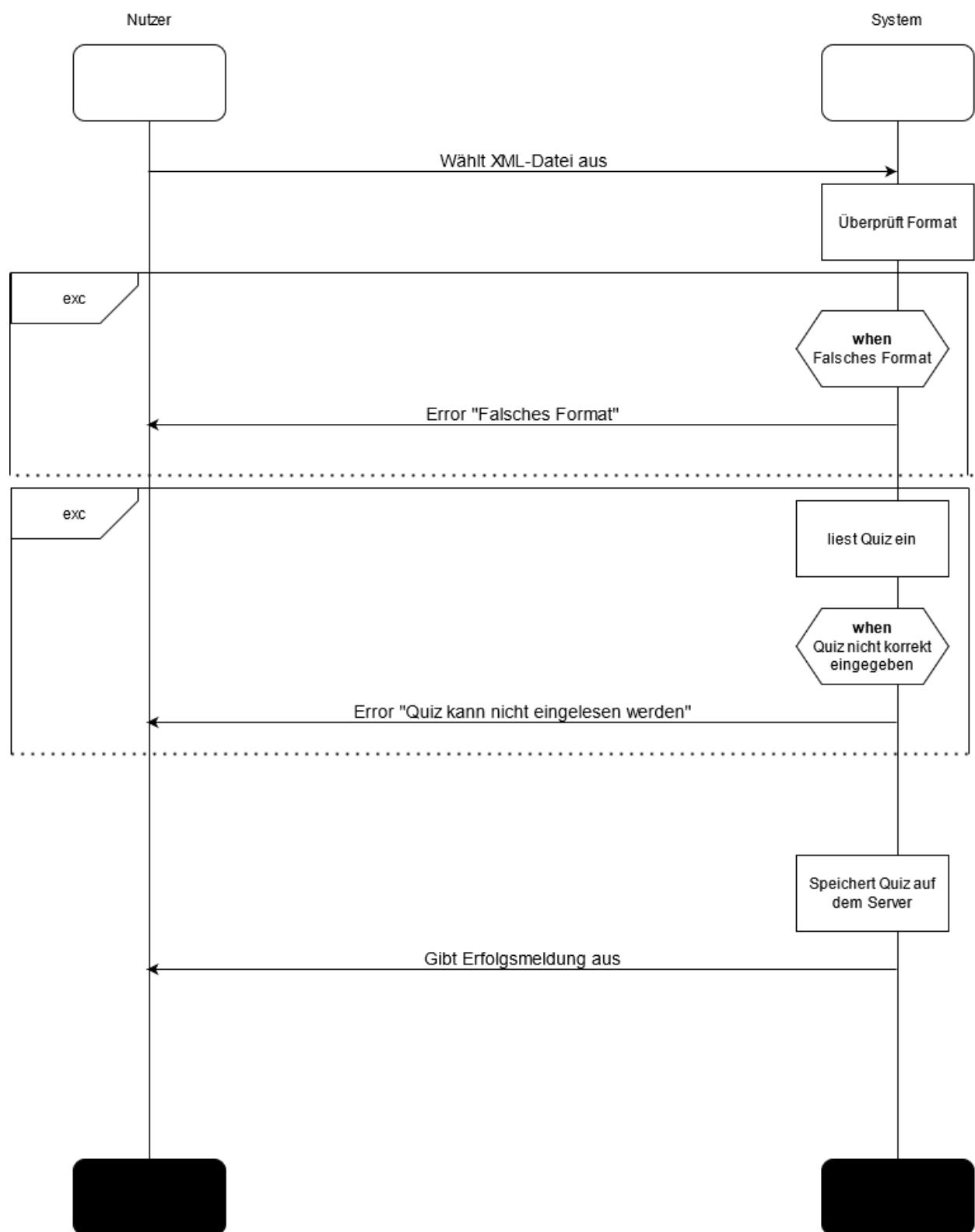
Damit die Nutzer miteinander kommunizieren können, sollen sie Freundschaftsanfragen austauschen. Ein Nutzer schickt ein Freundschaftsanfrage, das wird mithilfe des Systems weitergeleitet und der andere Nutzer darf diese Freundschaftsanfrage akzeptieren oder ablehnen.

3.1.1.2 Einzelnes Quiz erstellen



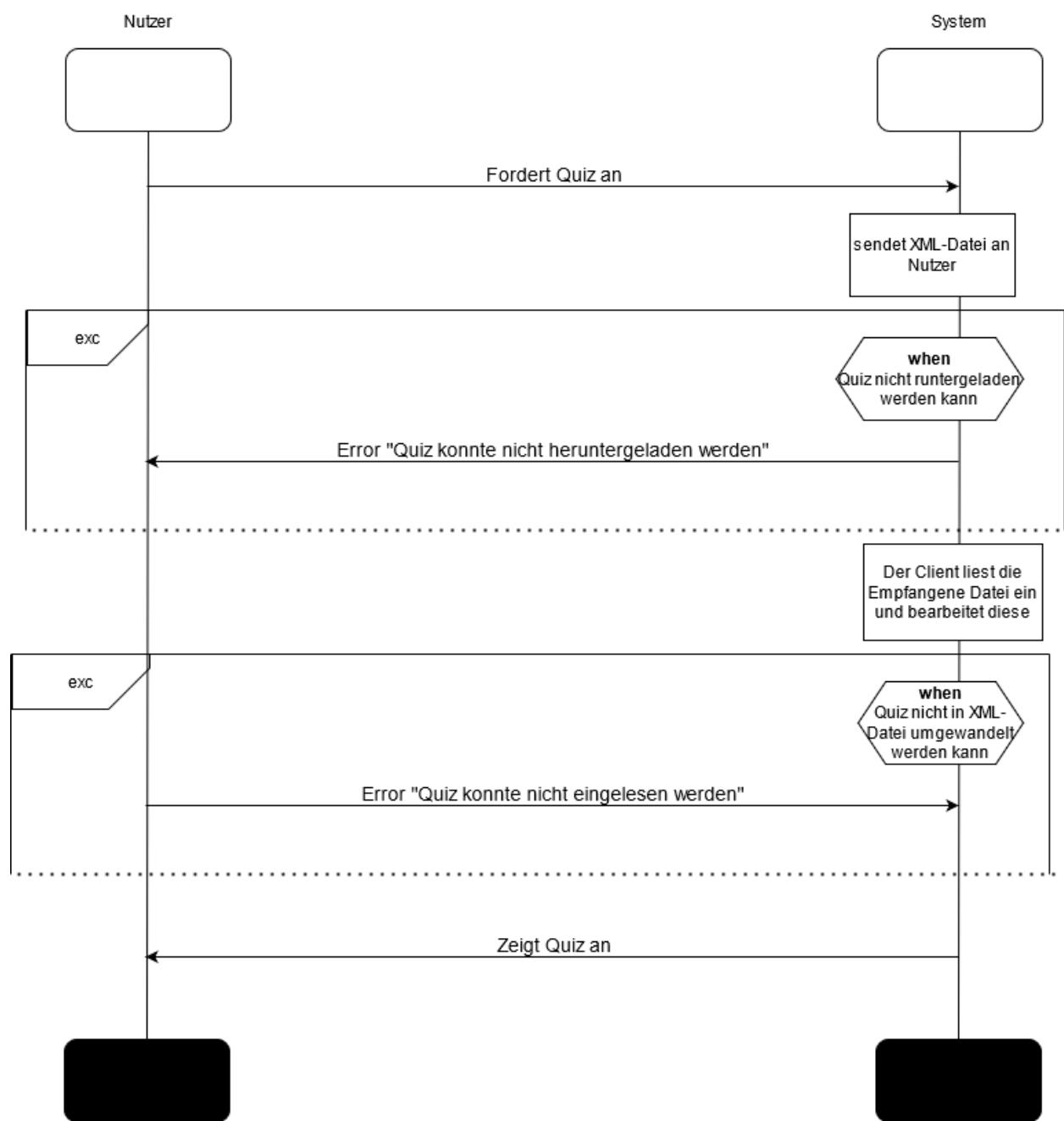
Damit die Lehrende Quize für die Studierenden erstellen können, sollen diese mit Hilfe des Systems ein eigenes Quiz erstellen können. Dieses Quiz wird dann als XML-Datei umgewandelt und auf dem Server gespeichert.

3.1.2 Quiz aus XML-Datei einlesen



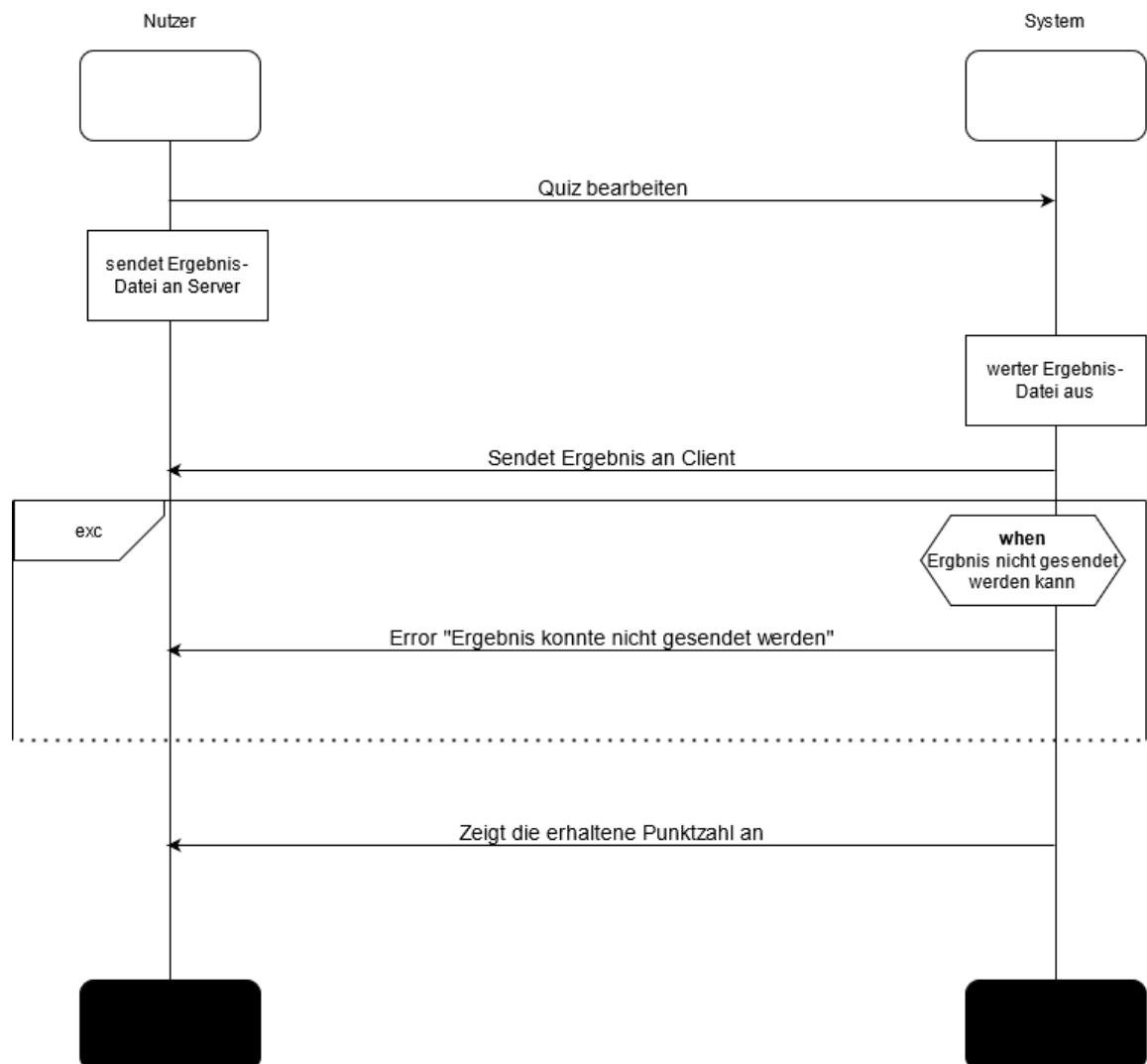
Alternativ können die Lehrenden auch ein Quiz aus einer XML-Datei einlesen lassen können. Dazu wird die XML-Datei ausgewählt und dann an den Server gesendet.

3.1.3 Quiz



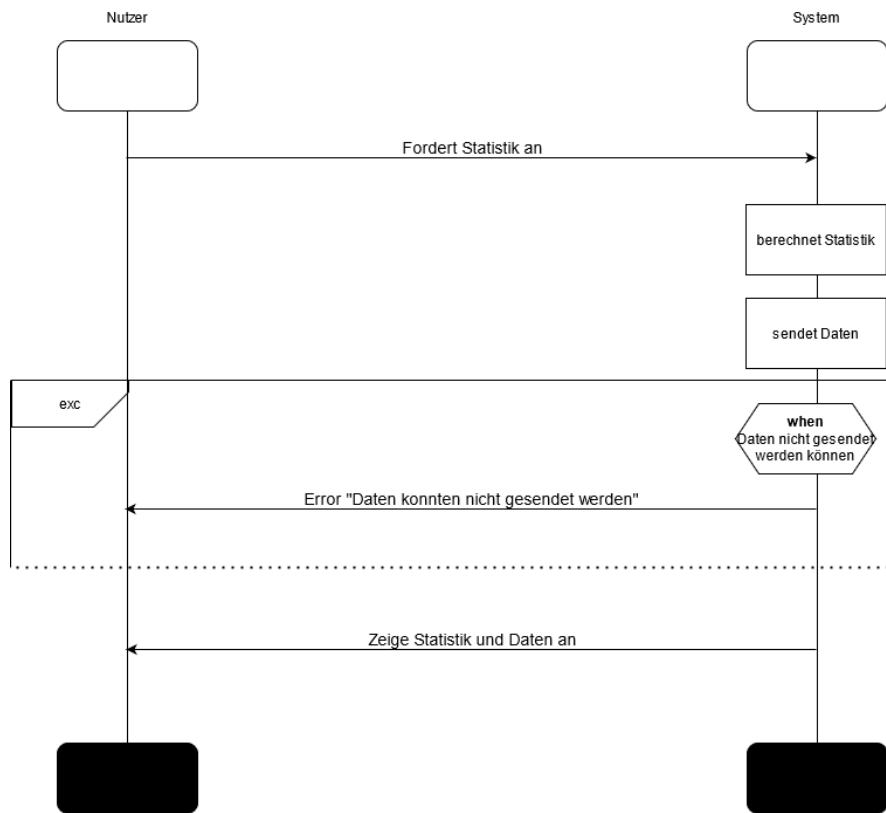
Damit die Nutzer das gewünscht Quiz bearbeiten können, wird die Quiz-XML-Datei vom Server heruntergeladen und im Client visualisiert.

3.2.2 Quiz-Feedback



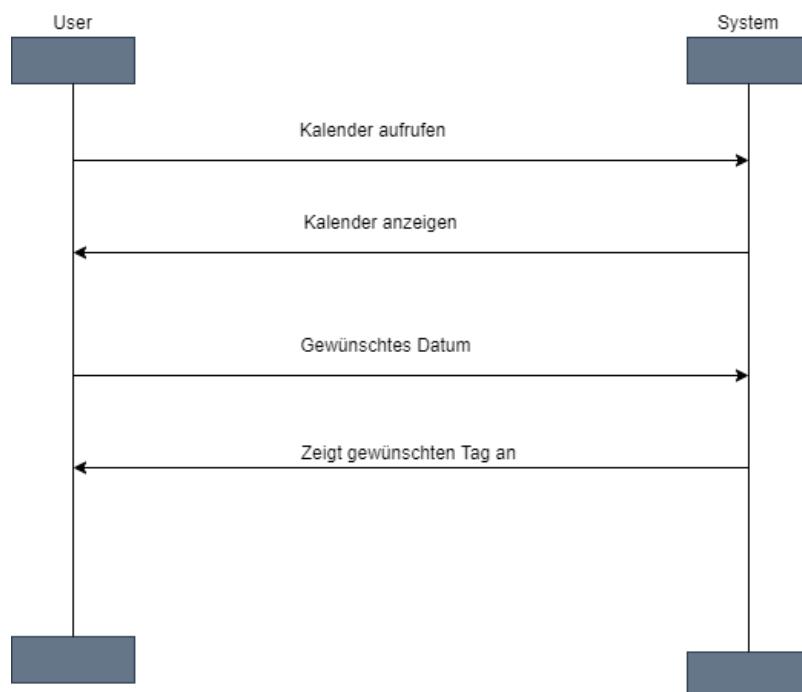
Damit die Nutzer ein Blick darüber haben, welche Fragen er richtig und welche falsch beantwortet hat, gibt es am Ende jeden Quizes ein Feedback, wie sich der Nutzer geschlagen hat. Dort wird aufgezeigt, welche Fragen richtig und welche falsch beantwortet wurden.

3.3.2 Statistik berechnen



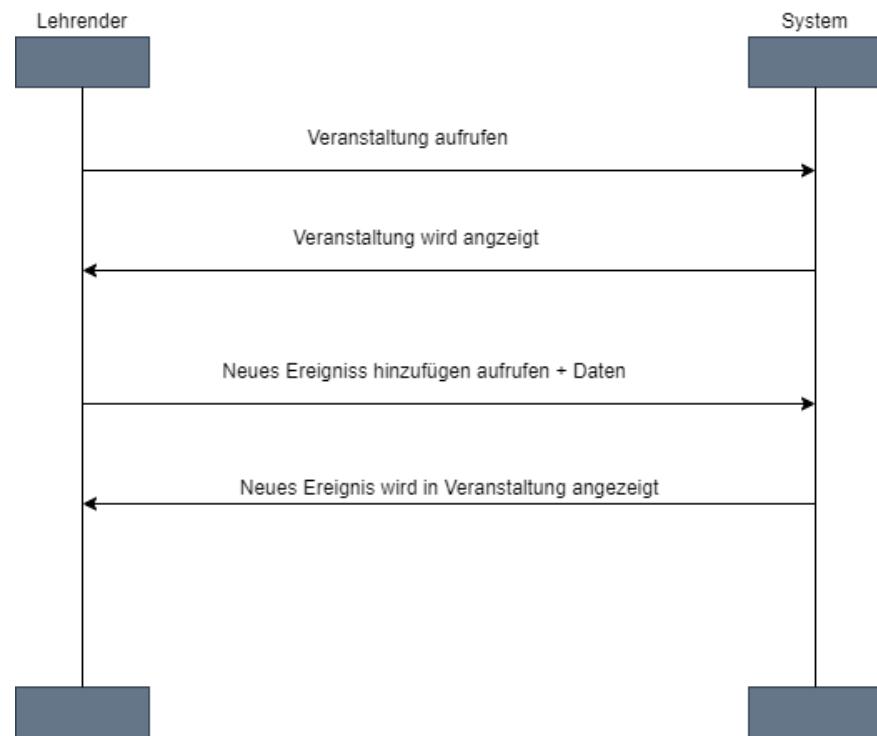
Damit der Lehrende einen Überblick über die Schwierigkeit Fragen hat, wird eine gesamt-Statistik für jedes Quiz erstellt. Dazu ruft der Lehrende einige Daten vom Server ab, die dann vom Client berechnet und ausgegeben werden.

4.1.1 User ruft Kalender auf



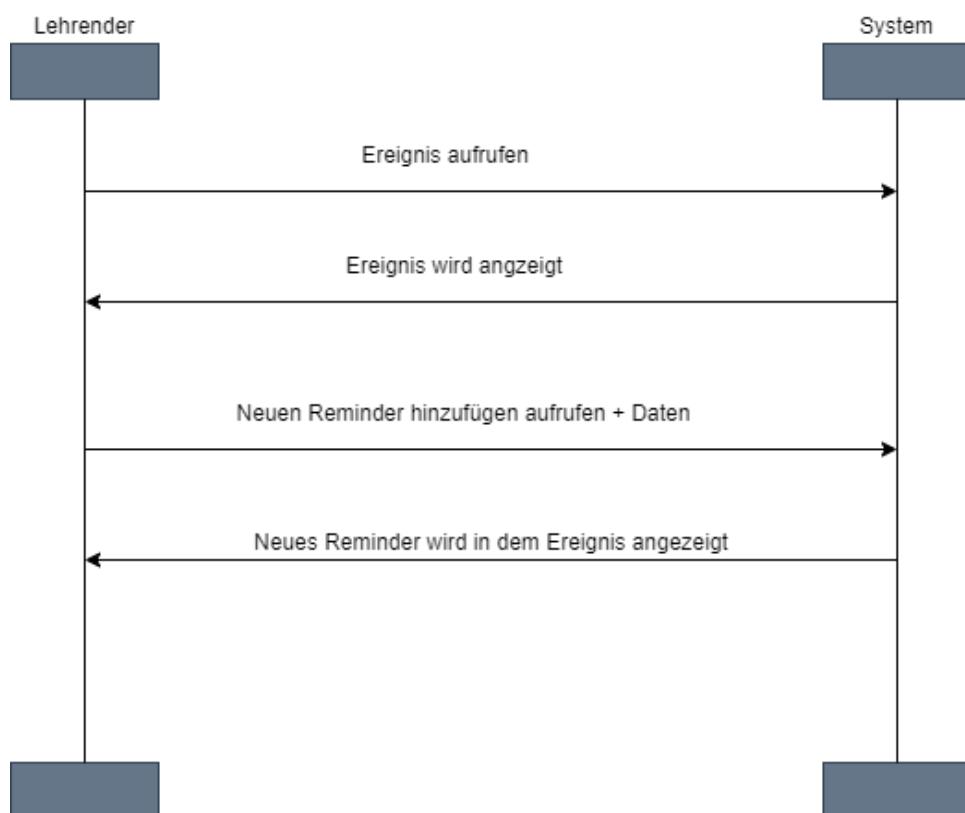
Ein User ruft den Kalender und gelangt durch Auswählen eines beliebigen Tages an die Termine des gewünschten Tages

4.2.1 Lehrender fügt Ereignis hinzu



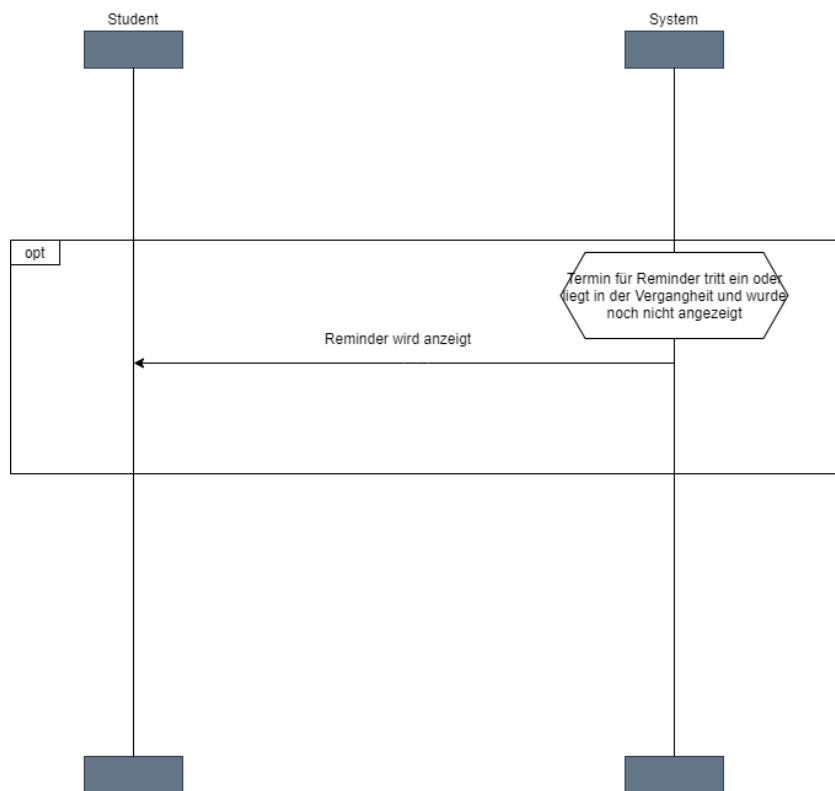
Wenn eine Lehrperson ein Ereignis zu einer Veranstaltung hinzufügen will, muss diese zuerst die Veranstaltung aufrufen. Dort kann ein neues Ereignis angelegt werden.

4.3.1 Lehrender fügt Reminder zu Ereignis hinzu



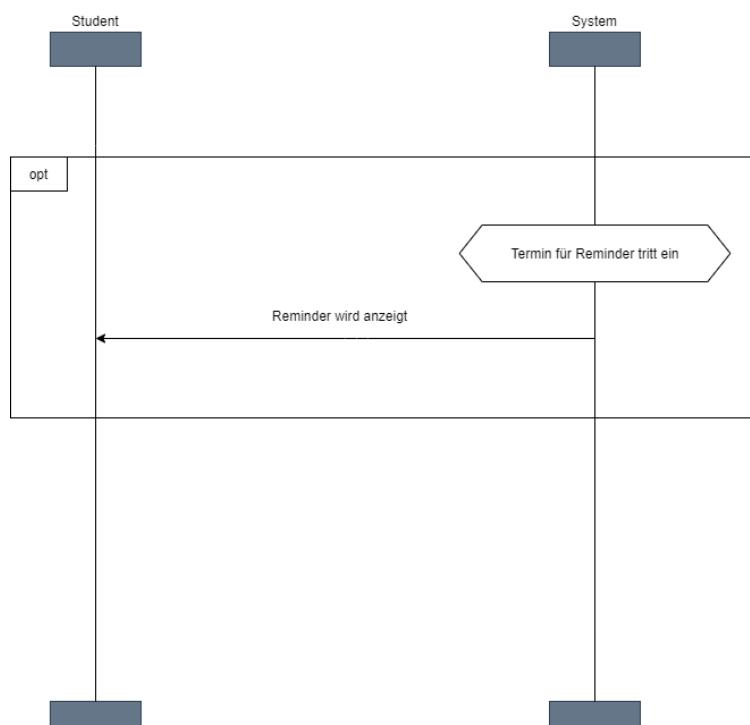
Wenn eine Lehrperson einen Reminder zu einem Ereignis hinzufügen will, muss diese zuerst das Ereignis aufrufen. Dort kann ein neuer Reminder angelegt werden.

4.4.1 Student bekommt Reminder als Pop-Up



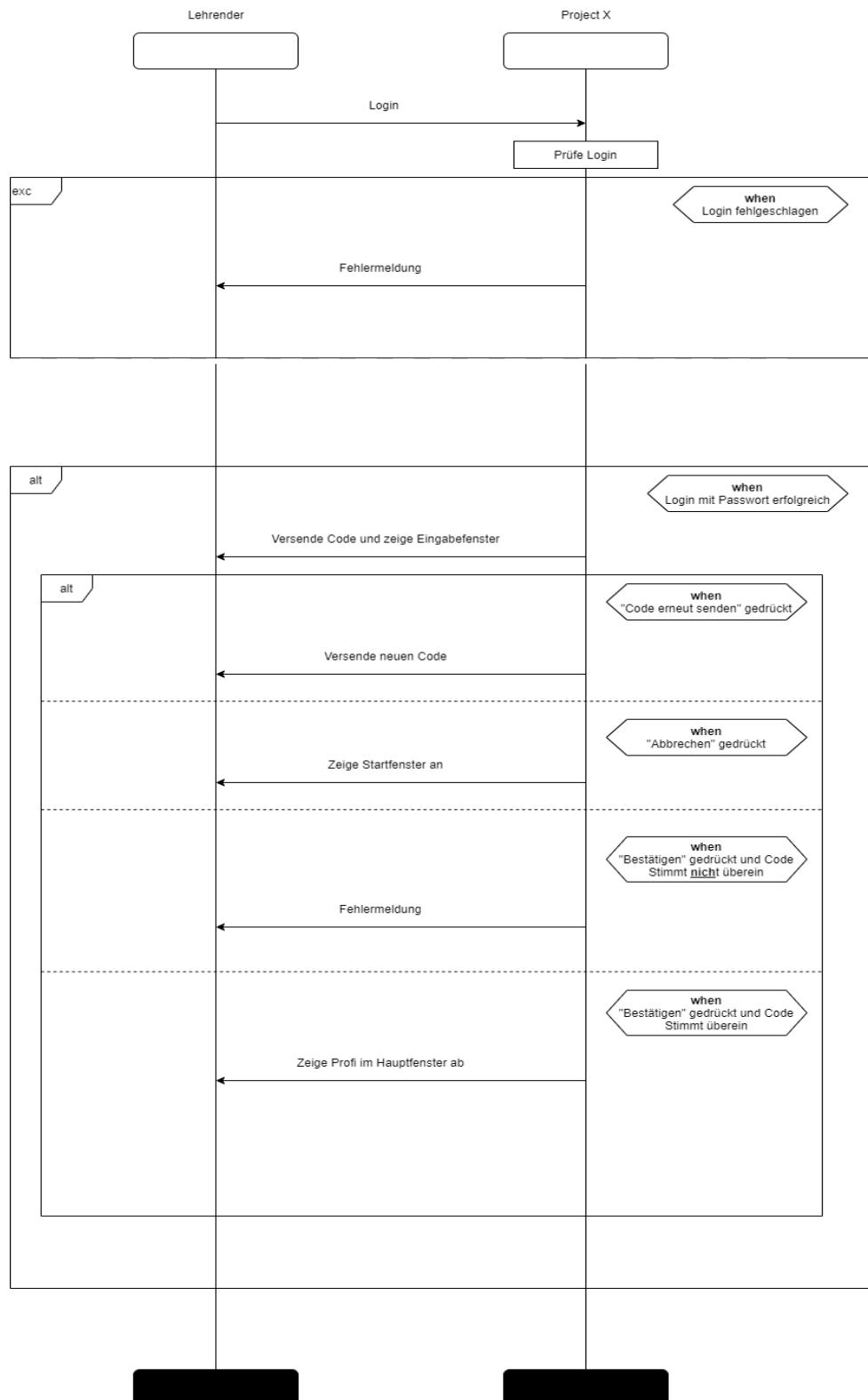
Wenn ein Student die Education Plattform Project X öffnet, wird geprüft, ob ein Termin für einen Reminder eintritt oder seit dem letzten Login eingetreten ist. Falls dem so ist wird der Reminder (als Pop-Up) angezeigt

4.4.1 Student bekommt Reminder als E-Mail



Wenn ein Reminder eintritt, wird eine E-Mail mit dem Reminder an den Nutzer verschickt.

5.3



Wenn der Nutzer im Startfenster auf Login klickt, öffnet sich ein Authentifizierungsfenster. In diesem Fenster muss der Nutzer einen Code eingeben, den er per Mail zugeschickt bekommen hat. Wenn er keine Mail bekommen hat, kann er sich erneut einen Code zusenden lassen oder den Vorgang abbrechen. Wenn der Code richtig ist, wird der User eingeloggt. Sonst bekommt er eine Fehlermeldung.

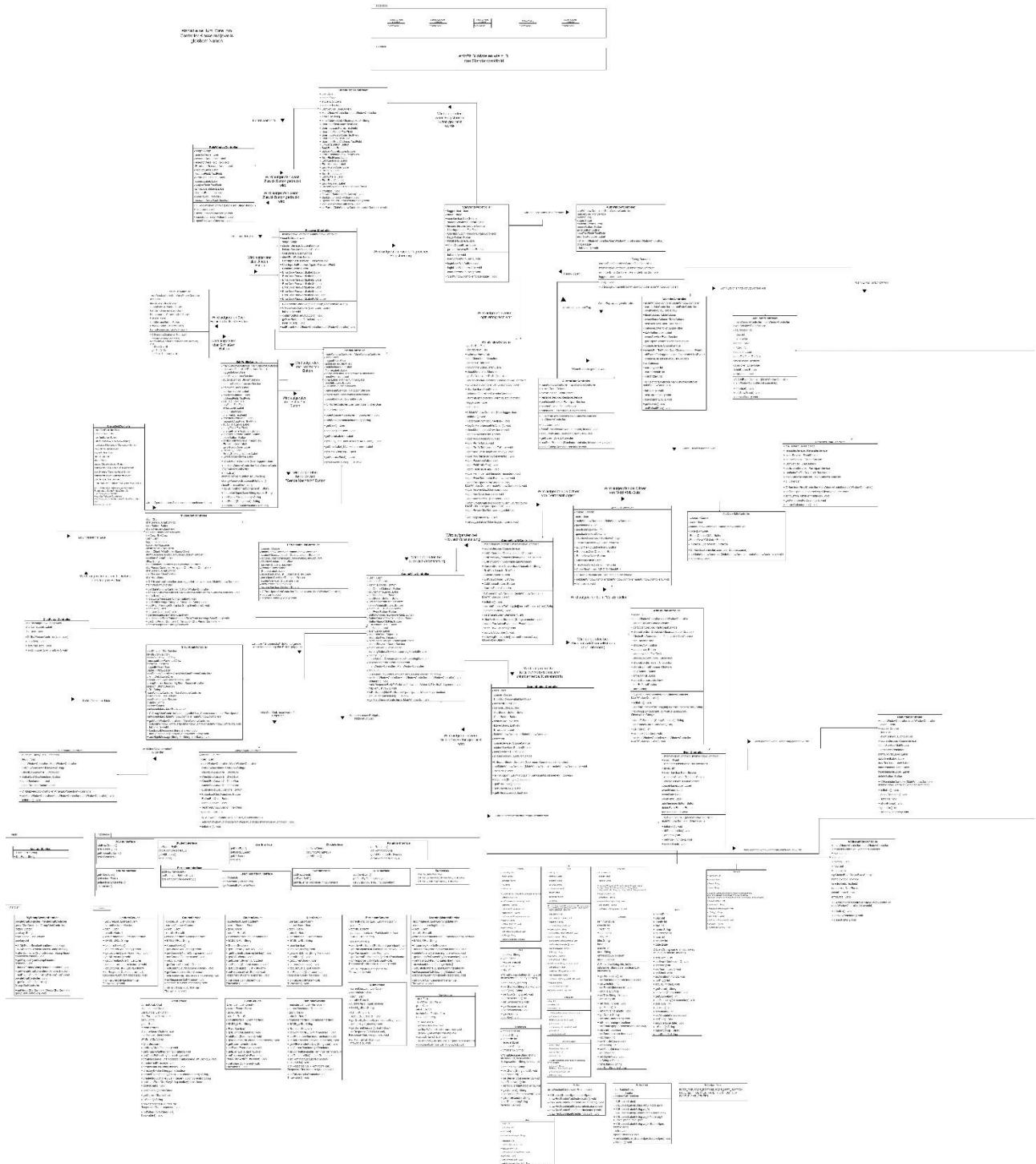
Strukturdiagramm (Klassendiagramm)

Client

Der Client besteht aus neun Packages:

- Controller
- FXML
- Service
- Model
- Exception
- Interfaces
- Handler
- Resources
- Value

Es gibt für jeden Controller eine FXML, die ein Fenster oder Tab darstellt. Nur der Gruppenchat und der PrivateChat teilen sich eine FXML Datei. Zu jedem Model gibt es ein Interface und einen Service. Der Handler enthält eine Klasse über den die Chat Sessions verwaltet werden. In Resources, Values und Exceptions sind Werte und Dateien gespeichert, die von den Controllern aufgerufen werden.

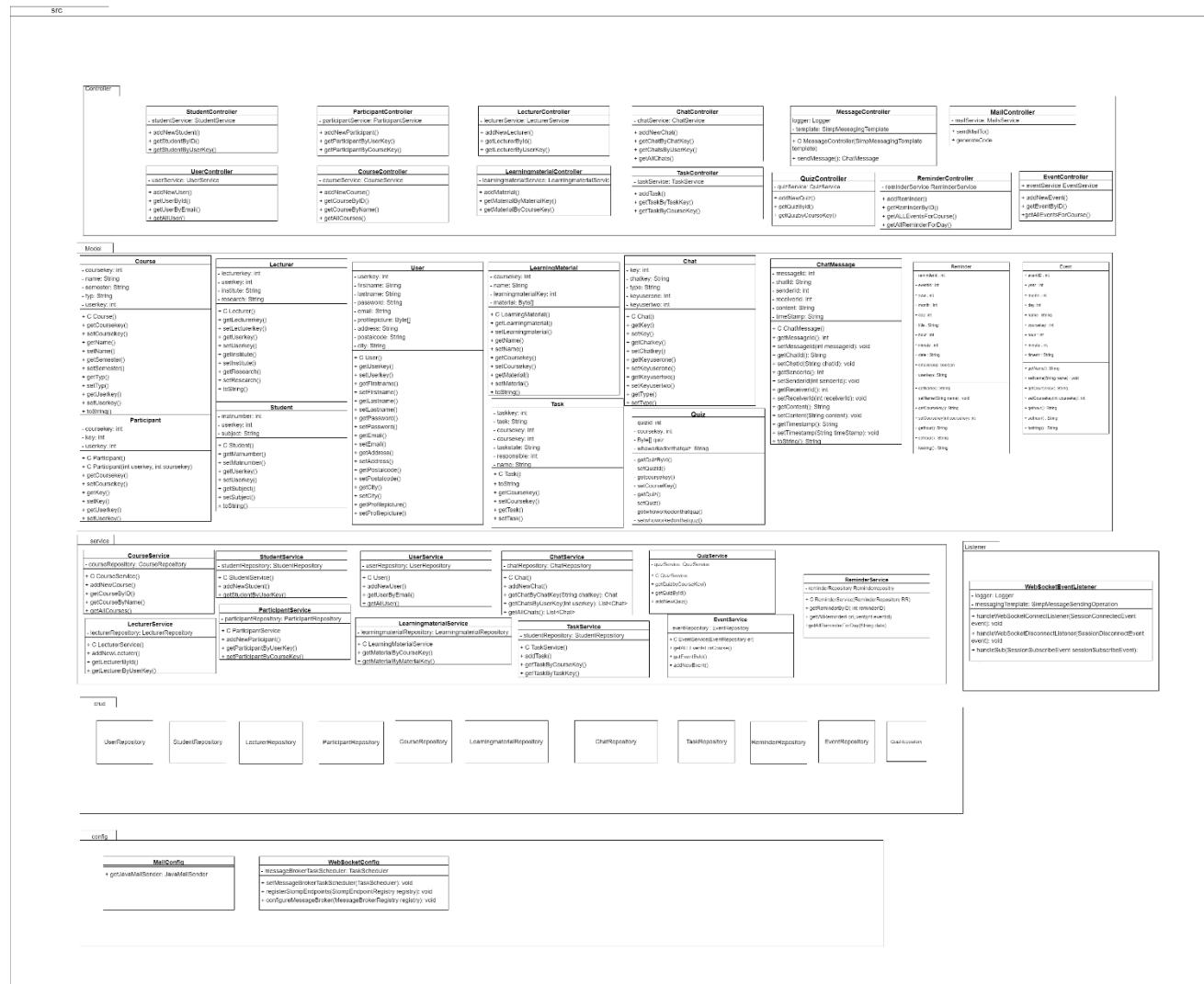


Server

Der Server besteht aus sechs Packages:

- Config
- Controller
- Crud
- Listener
- Model
- Service

Das Config Package enthält die Konfigurationen für die Mails und WebSockets. Zu jedem Controller im Controller Package gibt es einen Service, ein Repository im crud Package und ein Model. Sie sind die Gegenstücke zu den Klassen im Client. Ausnahmen davon sind der Message Controller und der Mail Controller. Das Listener package enthält den Listener, der die WebSocket Connections abhört.



Funktionalitätsplanung

Id	Funktionalität	Geschätzter Aufwand	Verantwortlicher	Abhängige Funktionalitäten	Verknüpfte User-Stories	Quellcode-referenz	Status
1.	Projektgruppe erstellen						
1.1	Projektgruppenansicht inklusive aller Button etc.	1 Tag	Hana Habibi		1.1.1	CourseViewController,	Fertig
1.1.2	To-Do-Listen Mit Möglichkeit der Aufgabenzuweisung	3 Tage	Hana Habibi		1.4.1	Task(Client&Server), TaskService (Client&Service), TaskController, TaskRepository (Service) CourseViewController, CreateTaskController(Client)	Fertig
1.1.3	Status von Todos aktualisieren	1 Tag	Hana Habibi		1.4.1	CourseViewController, Task(Client&Server)	Fertig
1.1.3	Dateien uploaden für Gruppenmitglieder	1 Tag	Thorben Kronewald-Lamken				In Bearbeitung
1.1.4	Nutzer zur Gruppe hinzufügen	1 Tag	Thorben Kronewald-Lamken				In Bearbeitung
1.2	Gruppenchat						
1.2.1	Chattab über Button in Projektgruppe öffnen	1 Tag	Noemi Kallweit		1.3.2	MainWindowController openGroupChat()	fertig
1.2.2	Chat pro Projektgruppe erstellen	1 Tag	Noemi Kallweit		1.3.2	Client: Service. ChatService	fertig
1.2.3	Nachrichtenaustausch zwischen Projektgruppen Teilnehmern in Chat	14 Tage	Kurinchinilavan Thangeswaran		1.3.2	Client: MyStompSessionhandler Server: WebSocketEventListerner,	fertig

						MessageController	
1.2.4	Anzeigen von Chatnachrichten in View	3 Tage	Noemi Kallweit		1.3.2	AddLeftMessage() bzw. AddRightMessage()	fertig
2.	Chat und Freundschaftsanfragen						
2.1	Privater Chat						
2.1.1	Chattab über "Nachricht senden" in Profiltab öffnen	1 Tag	Noemi Kallweit		2.1.2	Client: Controller.MainWindowController openPrivateChatTab()	fertig
2.1.2	Nachrichtenaustausch zwischen beiden Teilnehmern	14 Tage	Kurinchinil avan Thangeswaran		2.1.2	Client: MyStompSessionhandler Server: WebSocketEventlistener, MessageController	fertig
2.1.3	Nachrichten in View anzeigen	3 Tage	Kurinchinil avan Thangeswaran	2.1.2	2.1.2	AddLeftMessage() bzw. AddRightMessage()	fertig
2.1.4	Nachrichten lokal speichern	3 Tage	Noemi Kallweit	2.1.6	2.1.2	Client: Controller.PrivateChatController: createMessage()	fertig
2.1.5	Chatverlauf laden	3 Tage	Noemi Kallweit	2.1.4, 2.1.6	2.1.2	Client: Controller.PrivateChatController: loadLocalMessages()	fertig
2.1.6	Chatdateien lokal anlegen	3 Tage	Noemi Kallweit		2.1.2	Client: Service.ChatService: DirectoryExists(), CreateChatDirectory(), CreateChatFile()	fertig

2.1.7	Chatkey erstellen	1 Tag	Noemi Kallweit		2.1.2	Client: Service.ChatService CreateChatKey()	fertig
2.2	Freundschaftsanfragen						
2.2.1	Freundschaftsanfragen schicken	1 Tag	Hana Habibi		2.2.2	Client: Controller.ProfileController, Student:Client & Server	Fertig
2.2.2	Freundschaftsanfragen akzeptieren	2 Tage	Hana Habibi		2.2.1	Client: Controller.ProfileController Student:Client & Server	Fertig
3	Quiz						
3.1	Einzelnes Quiz erstellen	2 Tage	Frederik Gnass		3.1.1.1	AddQuizController	Fertig
3.2	Quiz in XML-Datei schreiben	3 Tage	Frederik Gnass		3.1.1.1	AddQuizXMLController	Fertig
3.3	Quiz aus XML-Datei auslesen	2 Tage	Frederik Gnass		3.1.2.1	AnswerQuizController	Fertig
3.4	Quiz bearbeiten	2 Tage	Frederik Gnass		3.1.3.1	AnswerQuizController	Fertig
3.5	Quiz Feedback ausgeben	1 Tag	Frederik Gnass		3.2.1	QuizFeedBackController	Fertig
3.6	Statistik berechnen	2 Tage	Frederik Gnass		3.3.1	QuizStatistikController	Fertig
4.	Kalender						
4.1	Jeder Nutzer hat einen eigenen Kalender	3 Tage	Vera Mitschulat		4.1.1	CalendarController()	Fertig
4.2	Im Kalender lassen sich für den Nutzer relevante Termine einsehen.	3 Tage	Vera Mitschulat		4.1.1	CalendarController() GetEvents()	fertig
4.3	Der Kalender lässt sich nach Tagen sortiert einsehen	2 Tage	Vera Mitschulat		4.1.1	CalenderController()	fertig

4.4	(Testfunktion) Das Datum des gesamten Systems lässt sich manuell setzen	3 Tage	Vera Mitschulat			CalendarContr oller() SetCurrentDat e() CurrentDateDa tePicker CurrentDateCa lendar()	Fertig
4.5	Über den Kalender lassen sich, für den Nutzer relevante Ereignisse öffnen	3 Tage	Vera Mitschulat		4.1.1.	CalendarContr oller EventTableVie w()	fertig
5.	Ereignisse und Reminder						
5.1	Ereignisse können von der Lehrperson einer Veranstaltung zu einer Veranstaltung hinzugefügt werden	1 Tag	Vera Mitschulat		4.2.1	AddEventContr oller()	fertig
5.2	Zu Ereignissen können von der zuständigen Lehrperson Reminder hinzugefügt werden	4 Tage	Vera Mitschulat	4.2.1	4.3.1.	AddReminder()	Fertig
5.3	Reminder werden als Pop-up für den betroffenden User angezeigt, wenn der Reminder eintritt	7 Tage	Vera Mitschulat		4.4.1	CurrentDateCo ntroller() DialogRunnabl e()	Fertig

5.4	Reminder werden als Email an den User verschickt wenn der Reminder eintritt	5 Tage	Vera Mitschulat		4.4.2	DialogRunnable() CurrentDateController() ReminderEmailController()	Fertig
6.	2-Fach-Authentification						
6.1	Auth-Code erstellen - Server	1 Tag	Kurinchinil avan Thangeswaran		5.1	AuthService.generateCode()	fertig
6.2	Auth Code an User per Mail senden	1 Tag	Kurinchinil avan Thangeswaran		5.1	MailController.sendMailTo()	fertig

Unitests

Unitest oder auch Modultests sind Komponententests. Diese werden in der Softwareentwicklung angewendet, um die funktionalen Einzelteile (Units) von Computerprogrammen zu testen.

Id	Getestete Funktionalität	Quellcode Referenz	Status
1	Abspeichern der empfangenen Nachricht in einer lokalen Datei als Empfänger	Client/test/service/ChatServiceTest.createMessagePrivatChat()	Test Bestanden
2	Abspeichern der empfangenen Nachricht in einer lokalen Datei als Sender	Client/test/service/ChatServiceTest.createMessagePrivatChat()	Test Bestanden
3	Abspeichern der Gruppennachrichten in den jeweiligen lokalen Dateien des Empfängers	Client/test/service/ChatServiceTest.CreateMessageGroupChat()	Test Bestanden

Systemtests

Datum	25.06.2021		
Tester	Hana Habibi		
SW-Version	V 0.1.2		
Vorbedin-gung(en)	Nutzer „Max Mustermann“ ist am System mit Passwort „Test123!“ registriert		
Schritt	Aktion (User)	Erwartete Reaktion (System)	v / X
1	Der Benutzer gibt den Benutzernamen „Max Mustermann“ auf der Tastatur ein.	Das System zeigt „Max Mustermann“ auf dem Display an.	v
2	Der Benutzer gibt das Passwort „geheim“ auf der Tastatur ein.	Das System zeigt das Passwort durch „*“-Symbole zensiert an.	v
3	Der Benutzer klickt auf „Anmelden“.	Das System schickt ein Code an Max Mustermanns Mail und öffnet den Authentifikation Window	v
4	Der Benutzer klickt auf Code erneut senden-Button	Das System schickt ein anderer Code zu Max Mustermanns Mail und öffnet den Authentifikation Window	v
5	Der Benutzer schreibt der letzte Code, der er auf seinem Mail bekommen hat und klickt auf Bestätigen-Button	Das System öffnet das Profil von Max Mustermann. Da werden die belegten Veranstaltungen, generelle Infos, Profilbild Profil Bearbeiten – Button und Freundschaftsliste-Button gezeigt	v
6	Der Benutzer klickt auf Projektgruppe “Weniger Plastikverpackungen im Alltag” in der Veranstaltungen Liste	Das System öffnet die Projektgruppe “Weniger Plastikverpackungen im Alltag” in einem neuen Tab	v
7	Der Benutzer klickt auf den Materialhinzufügen-Button	Der File Chooser wird geöffnet damit Max die gewünschte files hochlädt.	v
8	Der Benutzer wählt den pdf oder docx file aus und klickt das okay-button	Das System zeigt das File auf die Lernmaterialielliste.	v

9	Der Benutzer klickt auf den ToDoTask hinzufügen - Button	Das System öffnet das Create Task Window. Das System fragt da die Aufgabe, der Status der Aufgabe und der Verantwortlicher für die Aufgabe an.	v
10	Der Benutzer schreibt die Aufgabe "Hausaufgaben machen" und wählt den Status "NA - Nicht angefangen" der Aufgabe und der Verantwortlicher "Max Mustermann" aus. Am Ende klickt der Benutzer auf speichern-Button	Das System öffnet der Kurs Window wieder und jetzt kann Max seine erstellte Aufgabe in der ToDoListe sehen	v
11	Der Benutzer klickt auf der erstellte Aufgabe	Das System ändert der Status der Aufgabe von "NA - Nicht angefangen" zu "IB - In Bearbeitung"	v
Nachbedingung(en)	Nutzer ist am System mit Authentifizierung angemeldet, und kann in der Projekt Gruppe Lernmateriale austauschen und Aufgaben in ToDoList hinzufügen.		
Testurteil	Test bestanden.		

Datum	25.06.2021		
Tester	Noemi Kallweit		
SW-Version	V 0.1.2		
Vorbedin-gung(en)	Nutzer „Max Mustermann“ und „Dirk Dienstag“ sind am System angemeldet und haben den Tab Chat geöffnet		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Benutzer (Max Mustermann) schreibt die Nachricht „hi“ in der Text Box und klickt auf den senden-Button	Das System schickt die Nachricht „hi“ an Dirk Dienstag. Auf Max Mustermanns ChatWindow kann man diese Nachricht auf der rechten Seite sehen, währenddessen kann Dirk Dienstag die Nachricht auf der linken Seite sehen	✓
2	Dirk Dienstag antwortet Max Mustermann.	Diesmal wird die Nachricht bei Dirk auf der rechten Seite angezeigt und bei Max auf der Linken.	✓
3	Max Mustermann klickt den Kalender Tab an	Der Kalender Tab wird geöffnet	✓
4	Max Mustermann wechselt wieder in den Chat tab und klickt auf Dirk Dienstag	Der Chatverlauf mit Dirk Dienstag wird geladen	✓
Nachbe-dingung(en)	Max und Dirk haben Nachrichten ausgetauscht und diese wurden gespeichert.		
Testurteil	Test bestanden.		

Datum	25.06.2021		
Tester	Vera Mitschulat		
SW-Version	V 0.1.2		
Vorbedin-gung(en)	Nutzer „Max Mustermann“ ist am System angemeldet.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Benutzer klickt auf den „Kalender“ Tab	Das System zeigt den Kalender Tab an.	✓
2	Der Benutzer benutzt den linken DatePicker um einen Tag im Kalender auszuwählen	Das System zeigt den gewählten Tag im Kalender an. Falls es Ereignisse gibt sind diese in der TableView sichtbar. Es ist ein Event sichtbar	✓
3	Der Benutzer klickt das Event an.	Das System zeigt das Ereignis an	✓
4	Der Benutzer wechselt zurück auf den Kalender Tab.	Der Kalender Tab mit dem zuletzt ausgewählten Tab wird angezeigt.	✓
5	Der Benutzer nutzt den rechten DatePicker um das Datum des Systems umzusetzen	Das System setzt das aktuelle Datum auf das gegebene Datum. Wenn dadurch ein Reminder aktiviert wird. Wird dieser gezeigt	✓
Nachbe-dingung(en)	Das Datum des Systems wurde umgesetzt		
Testurteil	Test bestanden.		

Datum	25.06.2021		
Tester	Kurinchinilavan Thangeswaran		
SW-Version	V 0.1.2		
Vorbedin-gung(en)	Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ registriert		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Benutzer gibt den Benutzernamen „Max Mustermann“ auf der Tastatur ein.	Das System zeigt „Max Mustermann“ auf dem Display an.	✓
2	Der Benutzer gibt das Passwort „geheim“ auf der Tastatur ein.	Das System zeigt das Passwort durch „*-Symbole zensierte an.	✓
3	Der Benutzer klickt auf „Anmelden“.	Das System zeigt die Meldung „Anmeldung erfolgreich“ auf dem Display an.	✓
Nachbe-dingung(en)	Nutzer ist am System angemeldet, Anmeldezeitpunkt ist im System gespeichert.		
Testurteil	Test bestanden.		

Zyklus III

Zyklus 3

(Projektgruppe)

Neben den beschriebenen Funktionen, sollen die Mitglieder einer Projektgruppe nun Lernkarten erstellen und diese mit den übrigen Mitgliedern teilen können.

(Bewertung von Lehrveranstaltungen)

Studierende sollen die Möglichkeit haben, die von ihnen belegten Lehrveranstaltungen bewerten zu können. Zu diesem Zweck erstellt der Lehrende einen weiteren Test mit Multiple-Choice Fragen für seine Lehrveranstaltungen. Studierende können diese beantworten, sofern diese mindestens die Hälfte der vorherigen Tests bearbeitet und somit an der Veranstaltung teilgenommen haben. Der Lehrende soll sich auch hierzu eine anonymisierte Statistik ansehen können. Dabei soll er sehen können, wie häufig eine bestimmte Antwort zu einer Frage ausgewählt wurde. Außerdem soll er auswählen können, ob er die Bewertungsstatistik aller Teilnehmer einsehen will oder spezifischere Bewertungsstatistiken entweder nur über Studierende, die bestanden haben oder nur über diejenigen, die durchgefallen sind.

(Themenangebote für studentische Arbeiten)

Lehrende sollen Themenangebote für Abschlussarbeiten bereitstellen können. Ein Thema ist durch einen Titel, eine kurze Beschreibung und eine Literaturliste gegeben. Die Literaturliste kann der Lehrende durch das Einlesen einer bibtex-Datei³ erstellen, die die Daten über die Veröffentlichungen enthält. Die Themenangebote sollen im Nutzerprofil des Lehrenden angezeigt werden und nur für diejenigen Studierenden einsehbar sein, die einen Kurs des jeweiligen Lehrenden belegt haben.

(Mitteilung der Ergebnisse)

Studierende sollen automatisch eine E-Mail erhalten, wenn Sie eine Lehrveranstaltung bestanden haben oder durchgefallen sind. Eine Lehrveranstaltung gilt als bestanden, wenn bei Ablauf des Semesters mindestens die Hälfte der Tests einmal erfolgreich absolviert wurde. Eine Lehrveranstaltung gilt als Nicht-bestanden, wenn weniger als die Hälfte der Tests erfolgreich absolviert wurde.

Rot = Hana , Blau = Fred , Gelb = vorerst Kuri , Rosa = Noemi , Grün = Vera

Definition der Verantwortlichen für die Artefakte der Spezifikation

Id	Artefakt	Art des Artefakts	Verantwortlicher	Status
1.	Mitteilung der Ergebnisse			
1.1	Bewertung per E-Mail schicken	User Story	Vera Mitschulat	Erledigt
1.1.1	Bewertung per E-Mail schicken	MSC	Vera Mitschulat	Erledigt
2.	Bewertungen von Lehrveranstaltungen			
2.1	Lehrveranstaltung bewerten	User Story	Frederik Gnass	Erledigt
2.1.1	BMSC: Lehrveranstaltung bewerten	MSC	Frederik Gnass	Erledigt
2.1.2	Papierprototyp: Lehrveranstaltung bewerten	Papierprototyp	Kurinchinilavan Thangeswaran	Erledigt
2.2	Lehrveranstaltung Statistik	User Story	Frederik Gnass	Erledigt
2.2.1	BMSC: Lehrveranstaltung Statistik	MSC	Frederik Gnass	Erledigt
2.2.2	Papierprototyp	Papierprototyp	Kurinchinilavan Thangeswaran	Erledigt
3.	Lernkarten (Projektgruppen)		Hana Habibi	
3.1	Lernkarten	User Story		Erledigt
3.1.1	Lernkarten	Papierprototyp		Erledigt
3.2	Lernkarten erstellen	User Story		Erledigt
3.2.1	Lernkarten erstellen	MSC		Erledigt
3.2.2	Lernkarten erstellen	Papierprototyp		Erledigt
4.	Themenangebote für studentische Arbeiten		Noemi Kallweit	
4.1	Themenangebote bereitstellen	User Story		Erledigt
4.1.1	View Themenangebot erstellen	Papierprototyp		Erledigt
4.1.2	bMSC: Themenangebot erstellen	bMSC		Erledigt
4.2	Themenangebote einsehen	User Story		Erledigt
4.2.1	View Lehrerprofil	Papierprototyp		Erledigt
4.2.2	View Themenangebot	Papierprototyp		Erledigt
4.2.3	BMSC: Themenangebote einsehen	User Story		Erledigt

User-Stories

User Story-ID	1.1
User Story-Beschreibung	Als Nutzer möchte ich eine E-Mail erhalten, ob ich in einer Veranstaltung, in der ich eingeschrieben bin, durchgefallen bin oder bestanden habe, damit ich weiß ob ich durchgefallen bin oder bestanden habe
Geschätzter Realisierungsaufwand	3 Tage
Priorität	hoch
Autor(en)	Vera Mitschulat
Abhängigkeiten zu anderen User Stories	

User Story-ID	2.1
User Story-Beschreibung	Als Studierender möchte ich eine Möglichkeit haben, eine Lehrveranstaltung zu bewerten, damit ich dem Dozenten Tipps und Empfehlungen geben kann
Geschätzter Realisierungsaufwand	5 Tage
Priorität	hoch
Autor(en)	Frederik Gnass
Abhängigkeiten zu anderen User Stories	

User Story-ID	2.2
User Story-Beschreibung	Als Lehrender möchte ich eine erweiterte Statistik haben können, damit ich meine Lehrveranstaltung besser analysieren kann
Geschätzter Realisierungsaufwand	3 Tage
Priorität	hoch
Autor(en)	Frederik Gnass
Abhängigkeiten zu anderen User Stories	

User Story-ID	3.1
User Story-Beschreibung	Als Student möchte ich die Lernkarten, die die andere erstellt haben, sehen, damit ich effektiver lernen kann.
Geschätzter Realisierungsaufwand	1 Tag
Priorität	mittel
Autor(en)	Hana Habibi
Abhängigkeiten zu anderen User Stories	3.2
Zugehörige Szenarien	-

User Story-ID	3.2
User Story-Beschreibung	Als Student möchte ich Lernkarten erstellen, damit die anderen Studenten besser lernen können.
Geschätzter Realisierungsaufwand	1 Tag
Priorität	mittel
Autor(en)	Hana Habibi
Abhängigkeiten zu anderen User Stories	3.1
Zugehörige Szenarien	-

User Story-ID	4.1
User Story-Beschreibung	Als Lehrperson möchte ich Themenangebote für studentische Arbeiten in meinem Profil bereitstellen können, damit Studierende darauf zugreifen können
Geschätzter Realisierungsaufwand	14 Tage
Priorität	mittel
Autor(en)	Noemi Kallweit
Abhängigkeiten zu anderen User Stories	
Zugehörige Szenarien	-

User Story-ID	4.2
User Story-Beschreibung	Als Lehrperson möchte ich zu einem Themenangebot eine bibtex Datei hochladen können, damit die Studierenden bei der Bearbeitung die Literaturliste als Hilfestellung benutzen können.
Geschätzter Realisierungsaufwand	14 Tage
Priorität	mittel
Autor(en)	Noemi Kallweit
Abhängigkeiten zu anderen User Stories	
Zugehörige Szenarien	-

User Story-ID	4.3
User Story-Beschreibung	Als Studenten möchten wir nur die Themenangebote zu Kursen sehen in denen wir eingeschrieben sind, damit wir keine Themen sehen, die uns nicht betreffen
Geschätzter Realisierungsaufwand	3 Tage
Priorität	mittel
Autor(en)	Noemi Kallweit
Abhängigkeiten zu anderen User Stories	4.1
Zugehörige Szenarien	-

Papierprototypen

Evaluation erstellen – Ansicht

In dieser Ansicht kann der Lehrende sich für seinen Kurs eine eigene Evaluation erstellen. Es kann hier nur eine Frage in die Text-Box eingetragen werden. Die Antwortmöglichkeiten sind bei jeder Frage gleich gestaltet. Mit dem Knopf “Evaluation speichern” wird die Evaluation abgespeichert und direkt bei den Studenten angezeigt.

The paper prototype shows a user interface for creating an evaluation. On the left is a large empty white box. To its right is a rectangular area containing a text input field with the placeholder "Hier kommt Ihre Frage rein". Below this are four radio button options: "Stimme zu", "Stimme nicht zu", "Neutral", and "Kann ich nicht beurteilen". At the bottom are three buttons: "Zurück", "Weitere Frage", and "Evaluation speichern".

Evaluation beantworten – Ansicht

In dieser Ansicht kann der Studierende seine Antworten für die Evaluation abgeben. Oben (im momentan leeren Kasten) wird die Frage angezeigt. Dazu kann dann eine von 6 Antworten gewählt werden.

The paper prototype shows a user interface for answering an evaluation. On the left is a large empty white box. To its right is a rectangular area containing a question placeholder. Below it are six radio button options arranged in two rows of three: "Stimme zu", "Stimme nicht zu", "Neutral", and "Kann ich nicht beurteilen". At the bottom are three buttons: "Zurück", "Weiter", and "Evaluation beenden".

Auswertung Evaluation – Ansicht

In dieser Ansicht kann der Lehrende sich sein Feedback für seinen Kurs ansehen. Oben im Dropdown-Menü kann ausgewählt werden, ob alle Bewertungen angezeigt werden sollen, oder ob das Feedback gefiltert nach “bestanden”-Studenten und “durchgefallen”-Studenten angezeigt werden soll.

Statistik für: ▾

Frage	Stimme komplett zu	Stimme zu	Stimme nicht zu	Stimme gar nicht zu	Neutral	Nicht beurteilt

Kein Inhalt in Tabelle

Lernkarten - Ansicht

Diese Fenster wird angezeigt, wenn man auf den Lernkarten-Button von CourseView klickt. Wenn man auf die Lernkarte (das gefärbte Rechteck) klickt, dann wird die Beschreibung des Wortes gezeigt und auch andersrum. Der Nächste Karte-Button öffnet die nächste Karte und der Letzte Karte-Button öffnet die letzte Karte. Mit dem Schließen-Button kann man die CourseView wieder öffnen.

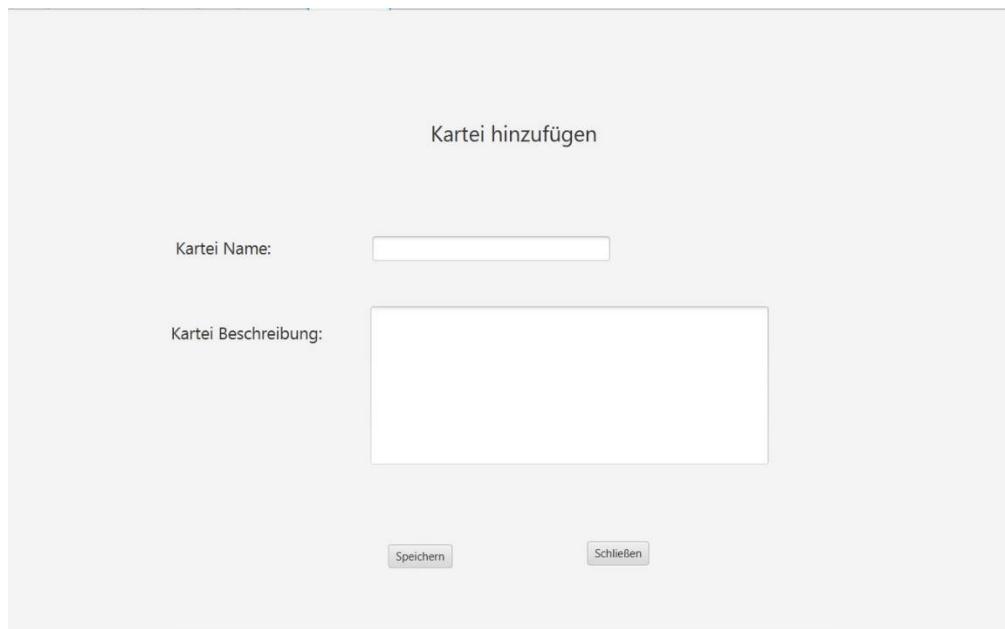
Profil Veranstaltungen Kalender Chats TestGruppe X

Software

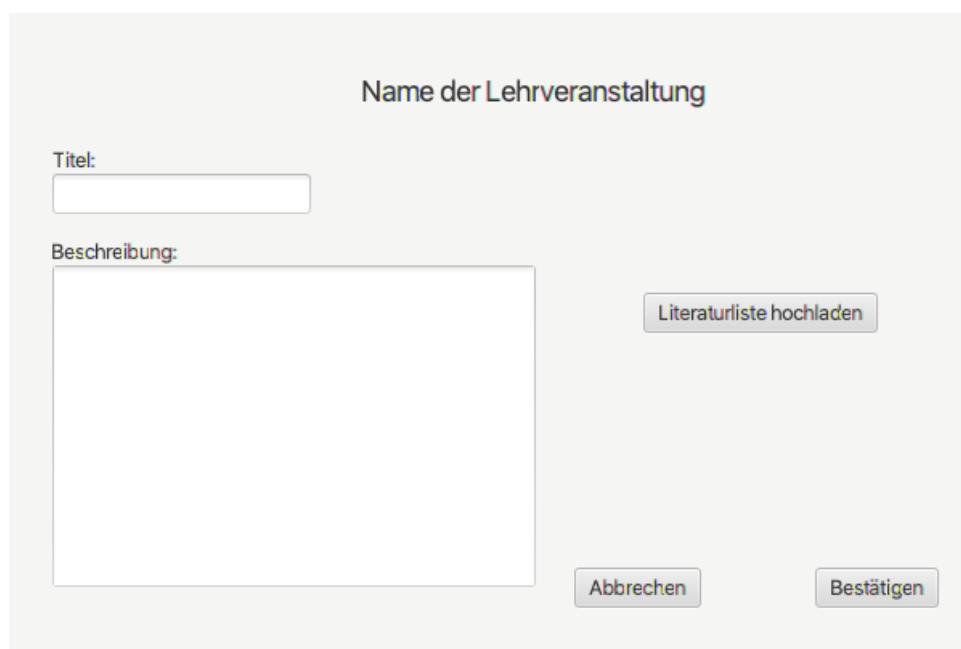
Letzte Kartei Nächste Kartei Schließen

Lernkarten erstellen – Ansicht

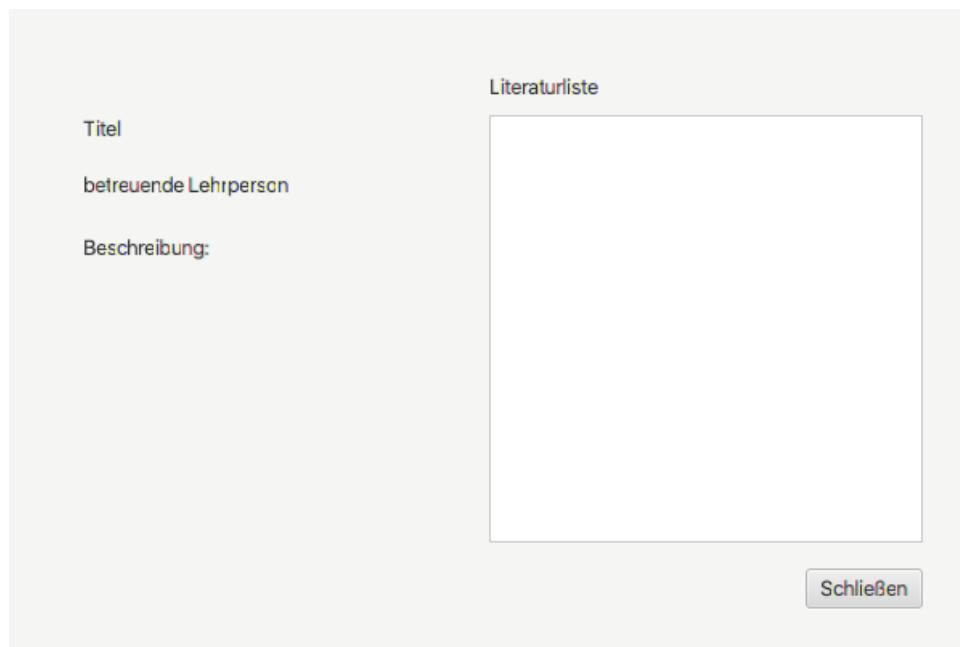
Dieses Fenster wird dann geöffnet, wenn der Kurs keine Lernkarten enthält oder wenn man auf den Nächste Karte-Button klickt (nur wenn es keine weiteren Karten gibt). Auf der ersten TextBox muss der Name der Karte geschrieben werden und auf dem Zweiten die Beschreibung dieser Karte. Die kann man durch den Speicher-Button speichern und dann wird diese Fenster automatisch geschlossen. Wenn man seine Inputs nicht speichern will dann kann man das Fenster einfach mit dem Schließen-Button schließen und die CourseView öffnen.



Themenangebot erstellen – Ansicht



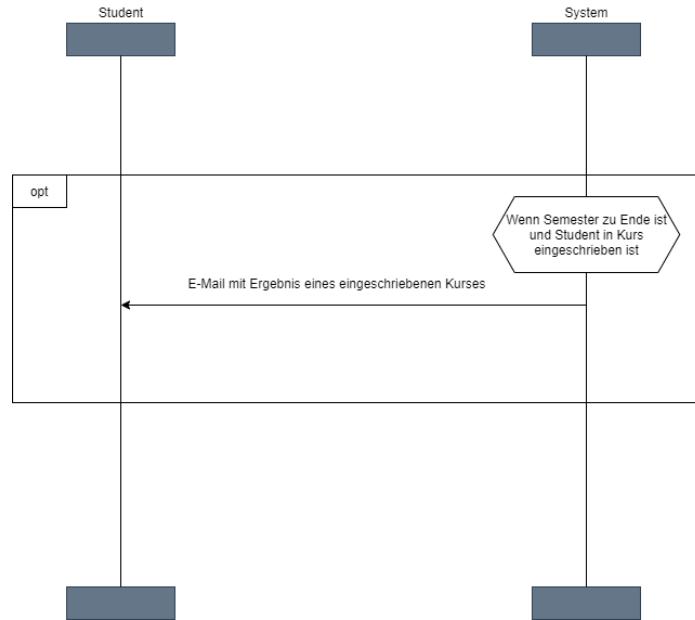
In diesem Fenster kann die Lehrpersonen alle nötigen Angaben für das Themenangebot machen. Der Button "Listeraturliste hochladen" öffnet ein Upload Fenster über das die Literaturliste hochgeladen werden kann. Über "Bestätigen" wird das Themenangebot gespeichert.

Themenangebot einsehen – Ansicht

Dieses Fenster sehen Studierende, wenn sie auf ein Themenangebot im Profil der Lehrperson klicken. Hier sehen sie alle benötigten Angaben zum Themenangebot.

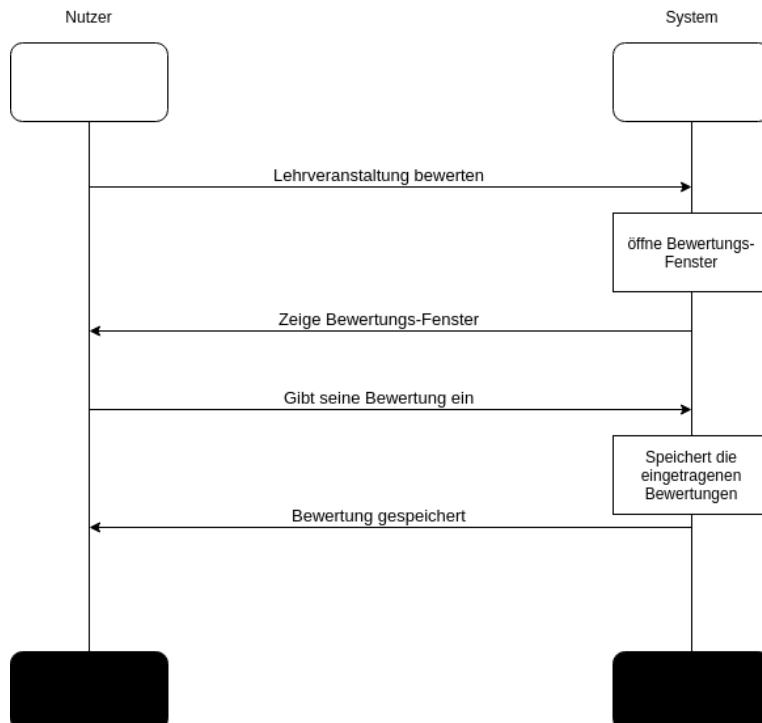
Szenarien (MSCs)

1.1 Erhalten einer E-Mail zur Bewertung



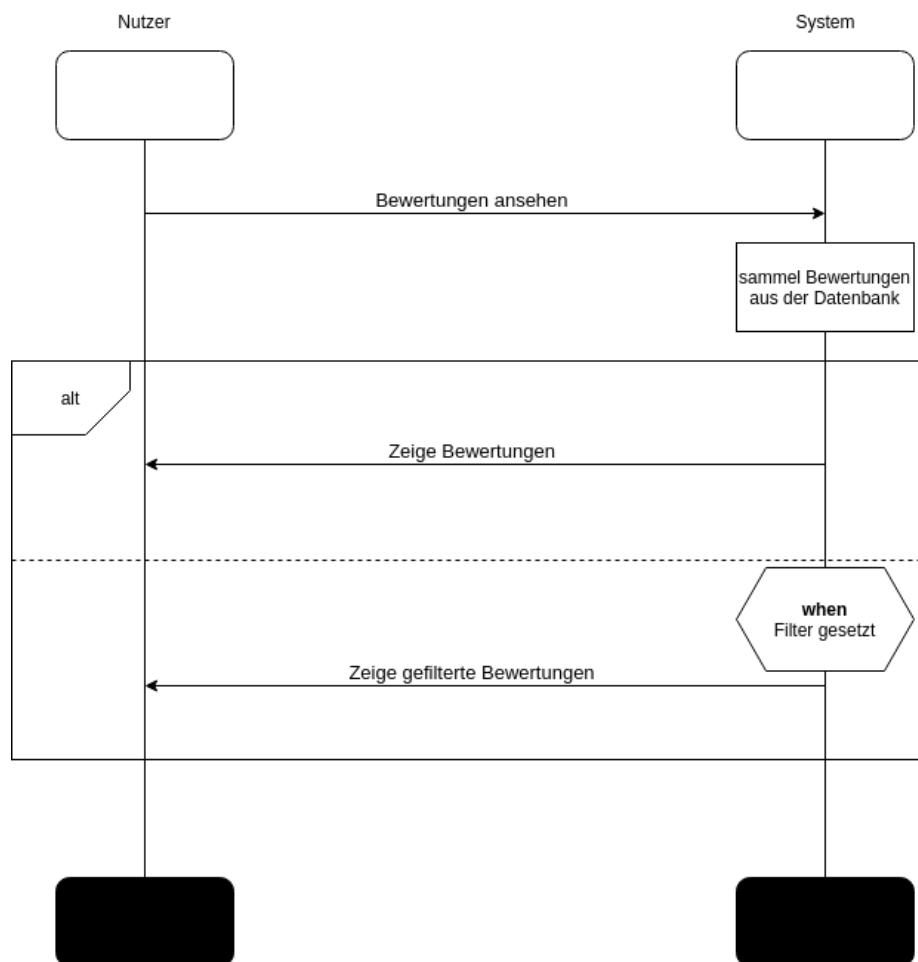
Wenn das Semester zu Ende ist und ein Student in einer Veranstaltung eingeschrieben ist, soll der Student eine E-Mail erhalten, ob er die Veranstaltung bestanden hat oder durchgefallen ist.

2.1 Veranstaltung bewerten



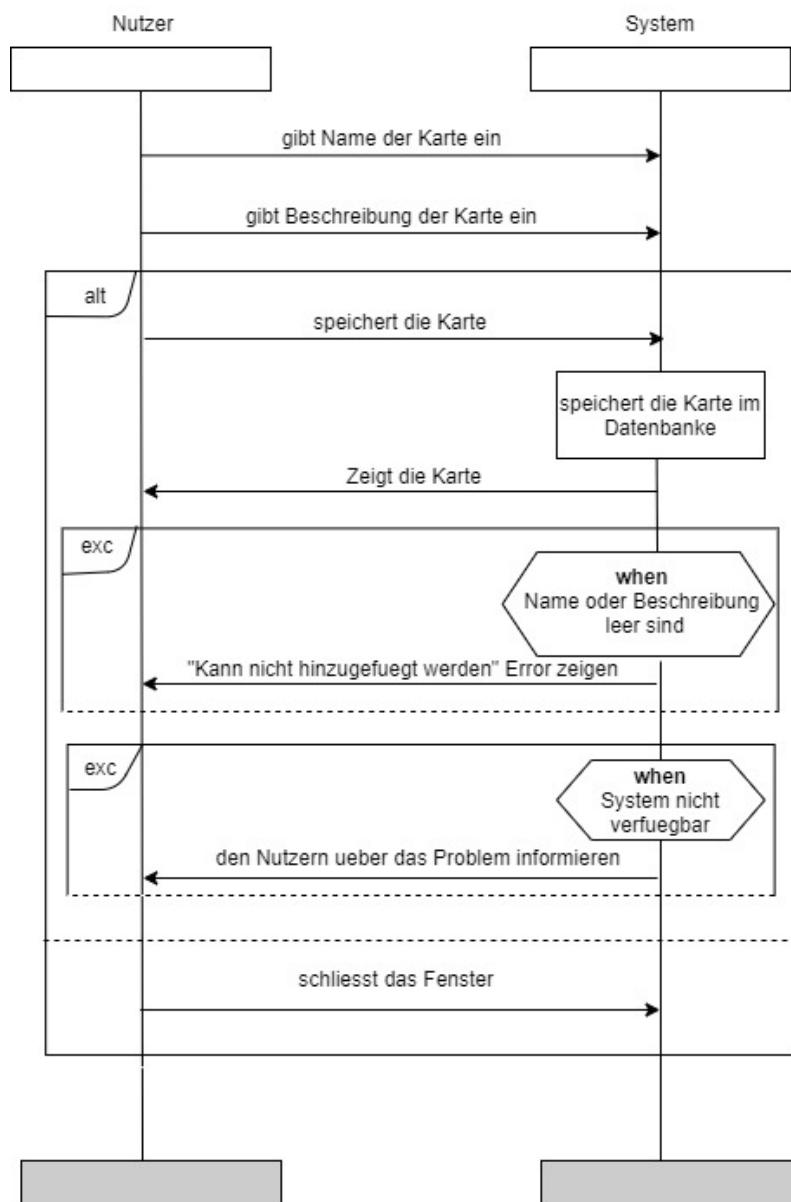
Wenn ein Nutzer eine Lehrveranstaltung bewerten möchte, um konstruktive Kritik zu äußern, kann er das über den entsprechenden Knopf im Kurs machen. Der Nutzer bekommt ein Multiple Choice Quiz und kann darüber die Lehrveranstaltung bewerten.

2.2 Bewertungen abrufen



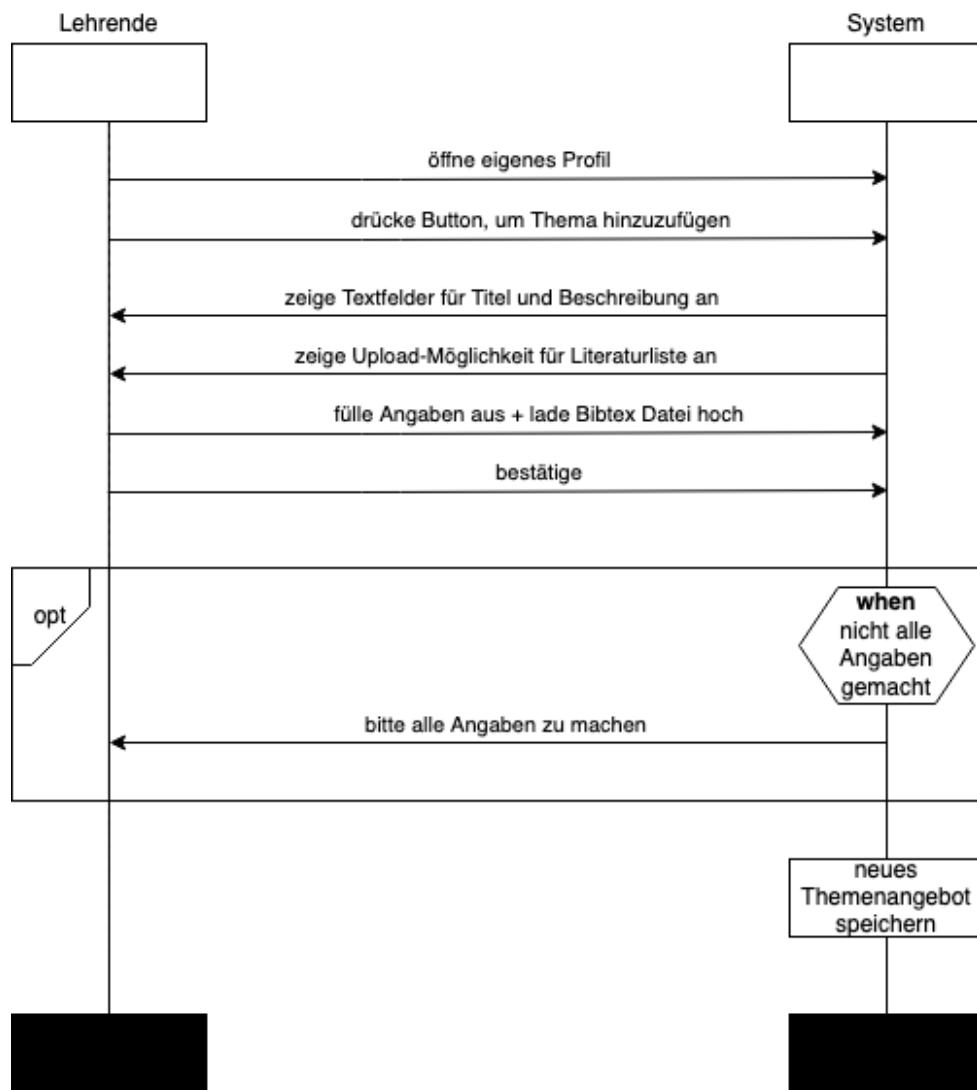
Als Lehrender habe ich die Möglichkeit die Bewertungen anzusehen und auch zu filtern. Wenn ich die Bewertungen ansehen möchte, gehe ich auf den entsprechenden Button im Kurs. Hier werden mir alle Bewertungen der Studierenden angezeigt. Ferner kann ich auch nach den Bewertungen filtern, die Studenten geschrieben haben, die den Kurs bestanden und nicht bestanden haben.

3.2.1 Lernkarte erstellen



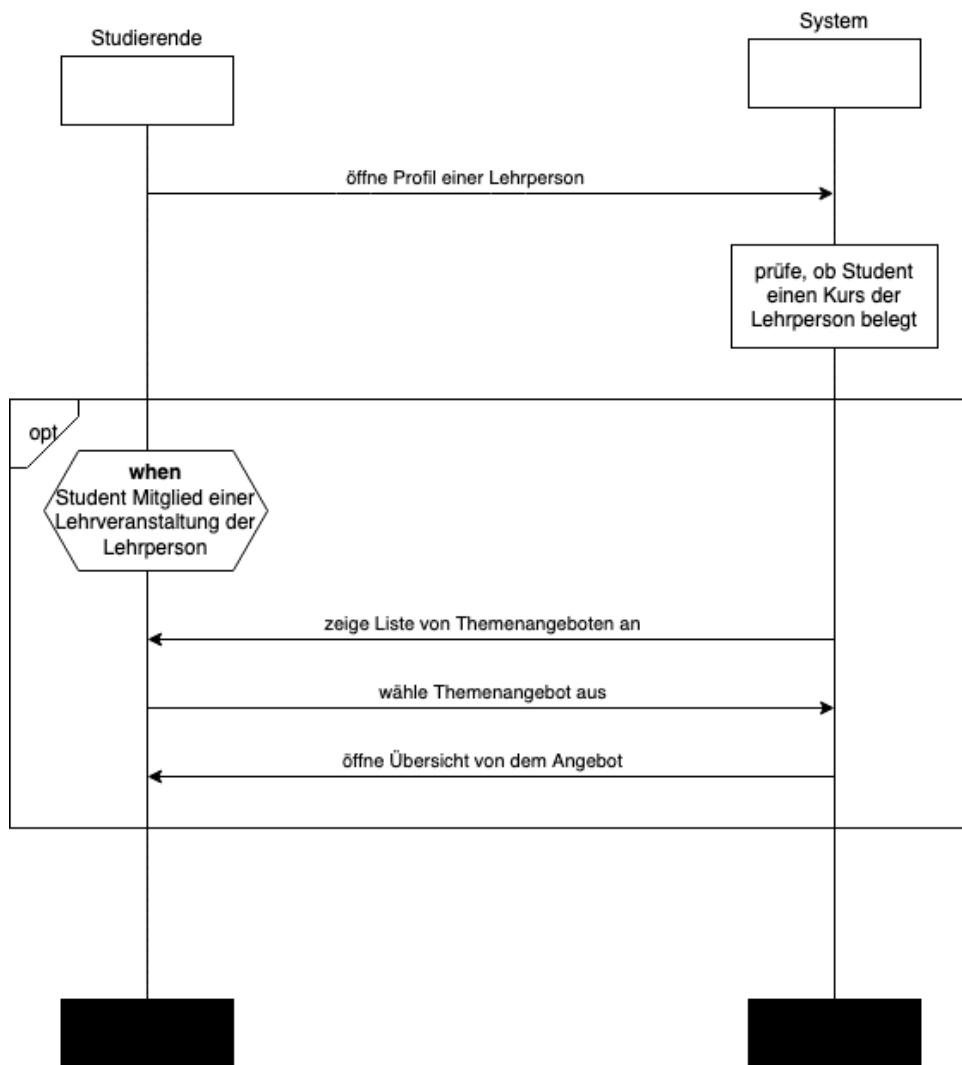
Wenn ein Nutzer eine Karte erstellt, muss er einen Namen und eine Beschreibung für die Karte eingeben und dann speichern. Wenn man nichts da eingegeben hat, dann wird das System einen Error zeigen.

4.1 Themenangebot für studentische Arbeiten erstellen



Wenn Lehrpersonen ihr Profil geöffnet haben, können sie dort über einen Button ein Dialogfenster öffnen, in dem sie Angaben zu dem Themenangebot machen können, welches sie gerne erstellen würden. Nachdem sie alle Felder ausgefüllt haben und eine bibtex Datei hochgeladen haben wird das Themenangebot gespeichert.

4.2 Themenangebote einsehen

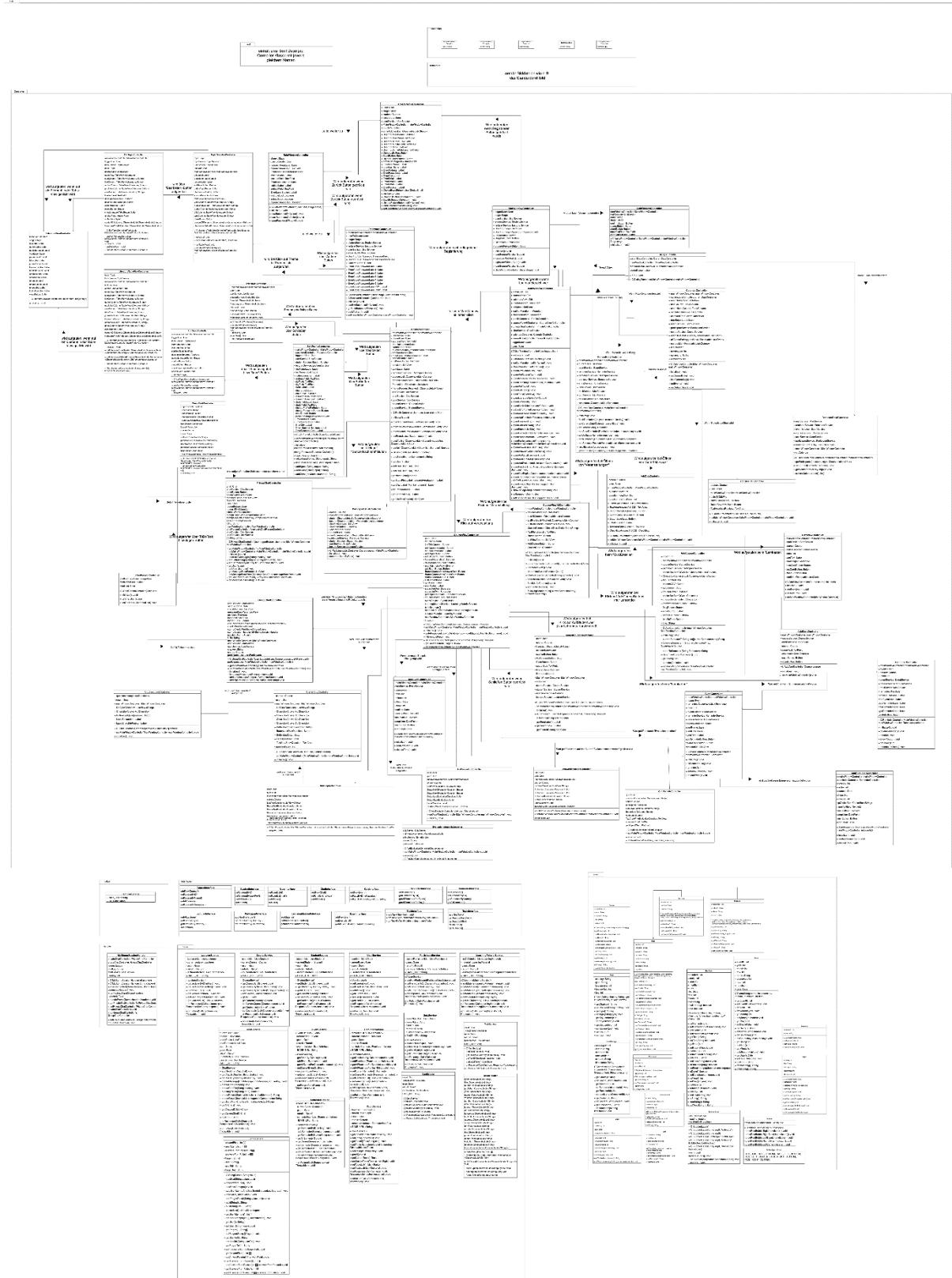


Wenn Studierende das Profil einer Lehrperson öffnen, prüft das System zuerst, ob sie in Kursen von dieser Person eingeschrieben sind. Danach wird eine Liste von allen Themen angezeigt, die den Studierenden betreffen.

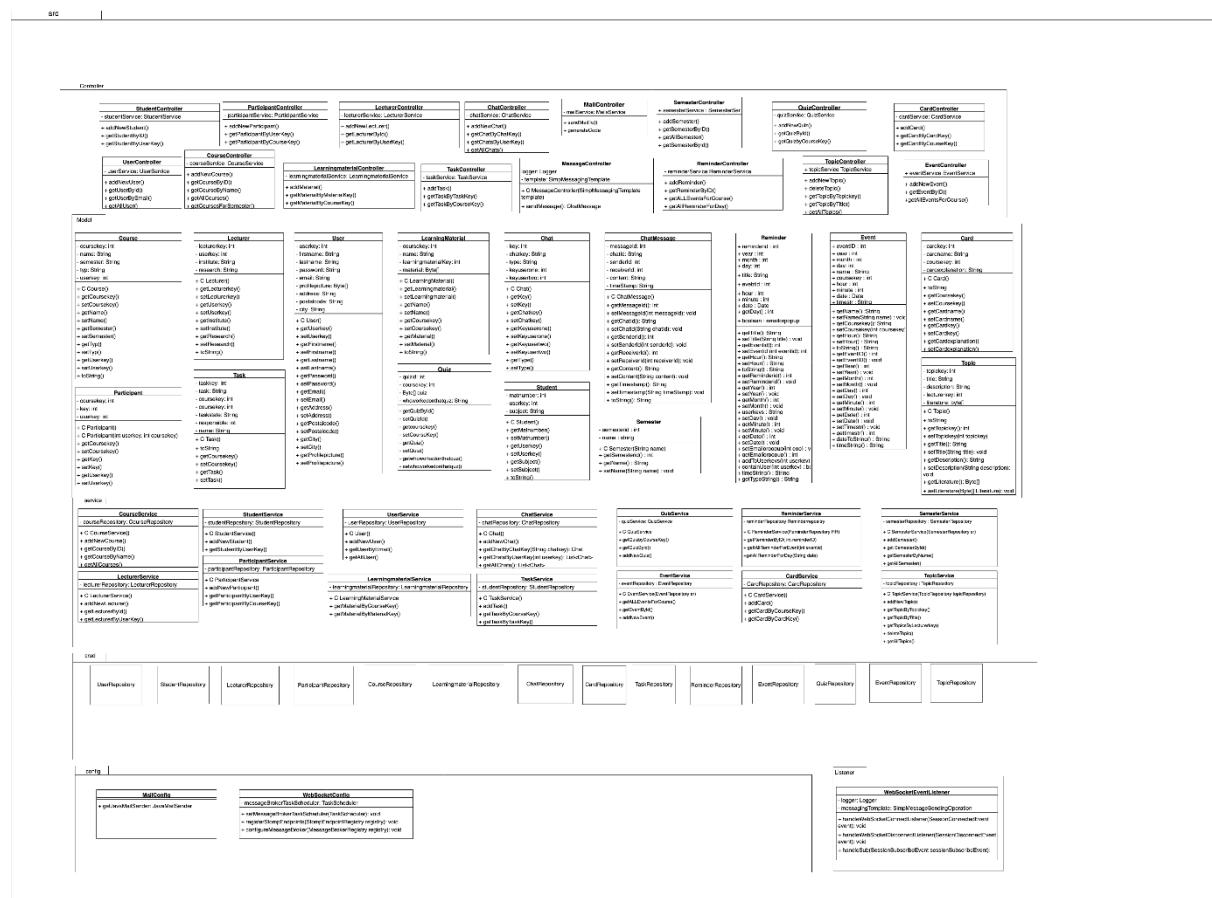
Strukturdiagramm (Klassendiagramm)

Erläuterung des Strukturdiagramms:

Client:



Server:



Funktionalitätsplanung

Id	Funktionalität	Geschätzter Aufwand	Verantwortlicher	Abhängige Funktionen	Verknüpft e User-Stories	Quellcode-referenz	Status
1.	Lernkarten						
1.1	Lernkarten View	1 Tag	Hana Habibi	1.2	3.1	Card: Client & Server CardService: Client & Server CardInterface: Client & Server CardRepository, CardViewController	Fertig
1.2	Erstellen von Lernkarten	1 Tag	Hana Habibi	1.1	3.2	Card: Client & Server CardService: Client & Server CardInterface: Client & Server CardRepository, AddCardController	Fertig
2.	Bewertungen von Lehrveranstaltungen						
2.1	Einlesen der Statistik	1 Tag	Kurinchinilavan Thangeswaran	2.2, 2.3, 2.4		Client: service.StringService.loadValue()/loadAntwortField()/loadUserAttempts()	Fertig
2.2	Hinzufügen einer Veränderung in der Statistik (Quiz und Evaluation)	1 Tag	Kurinchinilavan Thangeswaran			Client: service.StringService.addFragenResult()/userAttempts[]	Fertig
2.3	Hinzufügen eines Evaluationsstrings	1 Tag	Kurinchinilavan Thangeswaran			StringService()	Fertig
2.4	Ausrechnen der gefragten Werte	1 Tag	Kurinchinilavan Thangeswaran			EvaluationStatistikController	Fertig
2.7	Anzeigen der erstellten Evaluationsfragen	1 Tag	Frederik Gnass			Client: Controller.AnswerEvaluationController	Fertig

2.8	Quiz-Auswahl	1 Tag	Frederik Gnass			Client: Controller.QuizPopupController	Fertig
2.9	Hinzufügen des Evaluationsquizes	2 Tage	Frederik Gnass			Client: Controller.addEvaluationController	Fertig
3.	Mitteilung der Ergebnisse						
3.1	Überprüfung ob ein Semester beendet wurde/bereits beendet wurde		Vera Mitschulat			Client CurrentDate Controller pastSemester() hasSemesterPassed() Semester, SemesterService(Client & Server), SemesterInterface, Semesterepository, SemesterController	Fertig
3.2	E-Mails für jede Veranstaltung in dem beendeten Semester, an alle Teilnehmer, verschicken		Vera Mitschulat			Client CurrentDateController PastSemester() HasSemesterPassed() ResultEmail() gradeForUser()	Fertig
4.	Themenangebote für studentische Arbeiten						
4.1	Themenangebote bereitstellen						
4.1.1	Textfelder auslesen	1 Tag	Noemi Kallweit		4.1	Client.Controller. AddTopicController Initialize()	Fertig
4.1.2	Bibtex Datei auslesen	7 Tage	Noemi Kallweit		4.2	Client.Service BibtexReader	Fertig
4.1.3	Themenangebot in Datenbank speichern	1 Tag	Noemi Kallweit	4.2.1, 4.2.2, 4.2.2.1, 4.2.2	4.1	Client.Controller AddTopicController Initialize()	Fertig
4.1.4	Themenangebote bearbeiten		Noemi Kallweit	4.1.3	4.1	Client.Controller EditTopicController	Fertig

4.2	Themenangebot einsehen						
4.2.1	Themen in Profil anzeigen	2 Tage	Noemi Kallweit	4.2.2	4.1	Client.Controller ProfileController Initialize()	Fertig
4.2.2	Themenübersicht öffnen	1 Tag	Noemi Kallweit	4.2.2.1, 4.2.2.2	4.1	Client.Controller MainWindowController openTopicOverview()	Fertig
4.2.2.1	Literatur in Tabelle anzeigen	3 Tage	Noemi Kallweit	4.2.2.2	4.1	Client.Controller EditTopicController SetTableView() Client.Controller TopicOverview SetTableView() Client.Controller LiteratureTableViewController SetTableView()	Fertig
4.2.2.2	Pop-Up bei Klick auf Eintrag in Tabelle	2 Tage	Noemi Kallweit		4.1	Client.Controller EditTopicController Initialize() Client.Controller TopicOverview Initialize() Client.Controller LiteratureTableViewController Initialize()	Fertig

Unitests

Id	Getestete Funktionalität	Quellcode Referenz	Status
1	Laden der Userantworten und unterteilen in antworten von bestandenen und durchgefallenen	Client/test/StringServiceTest.tesLoadEval()	Test Bestanden
2	Speichern der Anzahl der beantworteten Fragen	Client/test/StringServiceTest.testAnswerQuantity()	Test Bestanden

Systemtests

Datum	30.06.2021		
Tester	Hana Habibi		
SW-Version	v3.01		
Vorbedin-gung(en)	Der Nutzer Max Mustermann ist erfolgreich registriert und das System zeigt gerade die Projektgruppe "TestGruppe" Fenster		
Schritt	Aktion (User)	Erwartete Reaktion (System)	v / X
1	Der Nutzer drückt den Lernkarten - Button	Das System zeigt die erste Lernkarte (Software).	v
2	Der Nutzer klickt auf der Lernkarte	Das System zeigt die Beschreibung von der erste Lernkarte (zusammenfassende Bezeichnung für die Programme, die auf einem Computer ausgeführt werden können. Ohne Software ist die Hardware nicht betriebsfähig. Unterschieden wird zwischen Anwendungsprogrammen und Systemprogrammen.)	v
3	Der Nutzer klickt auf dem nächsten Kartei-Button	Das System zeigt die nächste Kartei (Wasser)	v
4	Der Nutzer klickt auf der Lernkarte	Das System zeigt die Beschreibung von der erste Lernkarte (H2O)	v
5	Der Nutzer klickt auf dem nächsten Kartei-Button	Weil es keine weiteren Karten gibt, wird das System die Lernkarte Erstellen - Fenster öffnen	v
6	Der Nutzer gibt der Name und die Beschreibung der Karte ein	Die Inputs werden im System gezeigt(Name: "Test" und Beschreibung: "Test")	v
7	Der Nutzer drückt den Speichern-Button	Das System erstellt eine neue Lernkartei und geht zurück zu der Kurs Fenster	v
Nachbe-dingung(en)	Der Nutzer kann Lernkarten anschauen und erstellen		
Testurteil	Bestanden		

Datum	16.07.2021		
Tester	Noemi Kallweit		
SW-Version	v3.01		
Vorbedin-gung(en)	Die Lehrperson Olga Meyer ist erfolgreich registriert und sieht in ihrem Profil alle Themenangebote, die sie erstellt hat.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Olga klickt auf das Thema "Neues Thema"	Das System öffnet das Übersichtsfenster dieses Themas.	✓
2	Olga klickt in der Tabelle auf das Buch "An Integrative Model Of Organizational Trust"	Das System öffnet ein Pop-up mit Informationen zu dem Buch.	✓
3	Olga klickt auf den "Schließen"-Button	Das Pop-Up schließt sich.	✓
4	Olga klickt auf den "Thema bearbeiten"-Button	Das Bearbeitungsfenster öffnet sich. Alle vorhandenen Angaben sind ausgefüllt.	✓
5	Olga ändert den Titel des Themas in "Neues Thema 2.0" und klickt auf den "Bestätigen"-Button.	Das System zeigt ein Pop-Up, dass die Änderung bestätigt und wechselt zurück zum Themenübersichtsfenster. Der neue Titel wird im Übersichtsfenster angezeigt.	✓
Nachbe-dingung(en)	Der Titel des Themas "Neues Thema" wurde in der Datenbank zu "Neues Thema 2.0" verändert.		
Testurteil	Bestanden		

Datum	16.07.2021		
Tester	Vera Mitschulat		
SW-Version	v3.01		
Vorbedin-gung(en)	Ein Semester wurde noch nie zuvor beendet, um beendet zu werden. Ein User ist registriert und am System angemeldet		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Nutzer öffnet seinen Kalender	Das System öffnet den Kalender	✓
2	Der Nutzer benutzt den rechten Datepicker um das Datum des gesamten Systems zu einem Datum zu ändern, welches im nächsten Semester liegt	Das System ändert das Datum für das gesamte System und beendet ein oder mehrere Semester	✓
Nachbe-dingung(en)	Ein Semester wurde beendet und die E-Mails über die Ergebnisse wurden an die Teilnehmer der betreffenden Veranstaltungen verschickt		
Testurteil	Bestanden		

Nutzerhandbuch

Technische Anforderungen

Technische Mindestanforderungen, welche das Programm benötigt, um wie gewünscht bedienbar zu sein.

Installationsanleitung

Genaue Erläuterung, wie das entwickelte Programm vollkommen funktionsfähig auf einem Rechner in Betrieb genommen werden kann.

Bedienungsanleitung

Genaue Erläuterung, wie das entwickelte Programm zu bedienen ist.