SEP Projektmappe

**Projektmappe des Projektes**

**eTeach**

**Dokumentation des Projektes**

**Gruppe W**

Budiselic, Antonella

Krolik, Sammy-Steve

Metwally, Omar

Ridder, Christopher

Weingarth, Winston Julius

**Hinweis**

An vielen Stellen findet ihr im Dokument folgendes Kästchen:

*Dies ist eine Hilfestellung.*

Diese Kästen dienen dazu, euch kurze Informationen über Ziele und Inhalte der jeweiligen Abschnitte zu geben. Auch die Beispiele und Templates dienen dazu, euch bei der Dokumentation eures Projektes zu unterstützen. **Sowohl die Kästchen als auch die Beispiele und Templates sind spätestens zur finalen Abgabe der Projektmappe vollständig zu entfernen.** Betrachtet dieses Dokument bitte nicht als Aufgabe, die man von oben nach unten abarbeiten soll; es soll vielmehr als durchgängige Dokumentation eurer Projektarbeit dienen und fortlaufend erweitert bzw. angepasst werden, sodass am Ende des SEPs der Entwicklungsprozess eurer Software vollständig dokumentiert ist.

Das SEP-Team wünscht euch  
**viel Erfolg**  
bei der Bearbeitung der Hauptaufgabe!!!

Inhalt

[Projektbeschreibung 5](#_Toc76818059)

[Zyklus I 8](#_Toc76818060)

[Definition der Verantwortlichen für die Artefakte der Spezifikation 8](#_Toc76818061)

[User-Stories 10](#_Toc76818062)

[Papierprototypen 13](#_Toc76818063)

[**Papierprototyp Registration Student** 13](#_Toc76818064)

[**Papierprototyp Registration Lehrender** 14](#_Toc76818065)

[**Papierprototyp Lehrveranstaltung anlegen** 15](#_Toc76818066)

[**Papierprototyp Lehrveranstaltungsübersicht** 15](#_Toc76818067)

[**Papierprototyp Lehrveranstaltung suchen** 16](#_Toc76818068)

[**Papierprototyp Teilnehmer zu LV hinzufügen** 16](#_Toc76818069)

[**Papierprototyp Teilnehmerliste** 17](#_Toc76818070)

[**Papierprototyp Profilansichten** 17](#_Toc76818071)

[**Papierprototyp Login** 19](#_Toc76818072)

[**Papierprototyp Dashboard** 19](#_Toc76818073)

[Szenarien (MSCs) 20](#_Toc76818074)

[**Szenario Registration Student** 21](#_Toc76818075)

[**Szenario Registration Lehrender** 22](#_Toc76818076)

[**Szenario Profil bearbeiten Student** 23](#_Toc76818077)

[**Szenario Profil bearbeiten Lehrender** 24](#_Toc76818078)

[**Szenario Lehrveranstaltug anlegen** 25](#_Toc76818079)

[**Szenario Lehrveranstaltung mit CSV anlegen** 26](#_Toc76818080)

[**Szenario Material Upload** 27](#_Toc76818081)

[**Szenario Lehrveranstaltungsübersicht** 28](#_Toc76818082)

[**Szenario CSV Template** 28](#_Toc76818083)

[**Szenario Sortierte LV-Liste** 29](#_Toc76818084)

[**Szenario Lehrveranstaltung beitreten** 30](#_Toc76818085)

[**Szenario Student zu LV hinzufügen** 31](#_Toc76818086)

[**Szenario Teilnehmerliste** 32](#_Toc76818087)

[**Szenario Profilansicht** 33](#_Toc76818088)

[**Szenario Login** 34](#_Toc76818089)

[**Szenario LV suchen** 35](#_Toc76818090)

[Strukturdiagramm (Klassendiagramm) 36](#_Toc76818091)

[**Server** 36](#_Toc76818092)

[**Server Controller** 37](#_Toc76818093)

[**Server Services** 38](#_Toc76818094)

[**Server Repository** 38](#_Toc76818095)

[**Client** 39](#_Toc76818096)

[**Client Endpoints** 40](#_Toc76818097)

[**Verbindung Server-Client** 40](#_Toc76818098)

[Funktionalitätsplanung 41](#_Toc76818099)

[Systemtests 46](#_Toc76818100)

[Zyklus II 49](#_Toc76818101)

[Definition der Verantwortlichen für die Artefakte der Spezifikation 49](#_Toc76818102)

[User-Stories 50](#_Toc76818103)

[Papierprototypen 52](#_Toc76818104)

[**Papierprototyp Projektgruppe erstellen** 52](#_Toc76818105)

[**Papierprototyp Teilnehmer hinzufügen** 52](#_Toc76818106)

[**Papierprototyp Projektgruppe suchen** 53](#_Toc76818107)

[**Papierprototyp Quiz erstellen** 54](#_Toc76818108)

[**Papierprototyp Quiz mit CSV erstellen** 54](#_Toc76818109)

[**Papierprototyp Quiz bearbeiten** 55](#_Toc76818110)

[**Papierprototyp Quiz Ergebnisse** 55](#_Toc76818111)

[**Papierprototypen Statistik** 56](#_Toc76818112)

[Szenarien (MSCs) 57](#_Toc76818113)

[**Szenario Projektgruppe erstellen** 57](#_Toc76818114)

[**Szenario Studenten hinzufügen** 58](#_Toc76818115)

[**Szenario Projektgruppe suchen** 59](#_Toc76818116)

[**Szenario Quiz erstellen** 60](#_Toc76818117)

[**Szenario Quiz mit xml-Datei erstellen** 61](#_Toc76818118)

[**Szenario Quiz bearbeiten** 62](#_Toc76818119)

[**Szenario Quiz Ergebnisse** 63](#_Toc76818120)

[**Szenario Statistik** 63](#_Toc76818121)

[Strukturdiagramm (Klassendiagramm) 64](#_Toc76818122)

[**Server** 64](#_Toc76818123)

[**Server Controller** 65](#_Toc76818124)

[**Server Services** 65](#_Toc76818125)

[**Server Repository** 66](#_Toc76818126)

[**Client** 67](#_Toc76818127)

[**Client Endpoints** 68](#_Toc76818128)

[Funktionalitätsplanung 69](#_Toc76818129)

[Unittests 70](#_Toc76818130)

[Systemtests 71](#_Toc76818131)

[Zyklus III 74](#_Toc76818132)

[Definition der Verantwortlichen für die Artefakte der Spezifikation 74](#_Toc76818133)

[User-Stories 75](#_Toc76818134)

[Papierprototypen 77](#_Toc76818135)

[**Papierprototyp Privatnachricht versenden** 77](#_Toc76818136)

[**Papierprototyp Freundschaftsanfragen** 77](#_Toc76818137)

[**Papierprototyp Kalender** 78](#_Toc76818138)

[**Papierprototyp Ereignis erstellen** 78](#_Toc76818139)

[**Papierprototyp Reminder versenden** 79](#_Toc76818140)

[**Papierprototyp 2-Faktor Authentifizierung** 79](#_Toc76818141)

[Szenarien (MSCs) 80](#_Toc76818142)

[**Szenario Privatnachricht versenden** 80](#_Toc76818143)

[**Szenario Freundschaftsanfrage** 81](#_Toc76818144)

[**Szenario Ereignis in Kalender** 82](#_Toc76818145)

[**Szenario LV-Ereignis zu Kalender** 83](#_Toc76818146)

[**Szenario Reminder** 84](#_Toc76818147)

[**Szenario Zwei Faktor Authentisierung** 85](#_Toc76818148)

[Strukturdiagramm (Klassendiagramm) 86](#_Toc76818149)

[**Server** 86](#_Toc76818150)

[**Server Controller** 87](#_Toc76818151)

[**Server Services** 88](#_Toc76818152)

[**Server Repository** 89](#_Toc76818153)

[**Client** 90](#_Toc76818154)

[**Client Endpoints** 91](#_Toc76818155)

[Funktionalitätsplanung 92](#_Toc76818156)

[Unittests 93](#_Toc76818157)

[Systemtests 94](#_Toc76818158)

[Nutzerhandbuch 97](#_Toc76818159)

[Technische Anforderungen 97](#_Toc76818160)

[Installationsanleitung 97](#_Toc76818161)

[Bedienungsanleitung 97](#_Toc76818162)

# Projektbeschreibung

**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.

**Zyklus 1**

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 Platform 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-Datei1 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 Informationen einsehen. 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-Datei2 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 anzeigt 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 Hälfe 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 Hälfe der Tests einmal erfolgreich absolviert wurde. Eine Lehrveranstaltung gilt als Nicht-bestanden, wenn weniger als die Hälfte der Tests erfolgreich absolviert wurde.

**Anmerkungen**

* Das Programm ist in der Programmiersprache Java zu entwickeln.
* 3 Eine Beispiel-bibtex-Datei wird in Moodle bereitgestellt
* Die Datenbank ist als lokale Datenbank anzulegen (z.B. via XAMPP - https://www.apachefriends.org/de/index.html)
* Wir veröffentlichen besonders gelungene Software auf unserer SEP-Webseite. Hierzu ist es unbedingt erforderlich

# Zyklus I

## Definition der Verantwortlichen für die Artefakte der Spezifikation





## User-Stories

(Registrierung der Nutzer)

|  |  |
| --- | --- |
| **User Story-ID** | 1 |
| **User Story-Beschreibung** | Als Studierender möchte ich bei der Registrierung einen Vor- und Nachnamen, Passwort, E-Mail-Adresse, Profilbild, private Adresse und ein Studienfach angeben, so dass ein Profil mit diesen Angaben für mich erstellt wird. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | hoch |
| **Autor(en)** | Krolik, Sammy-Steve |
| **Abhängigkeiten zu anderen User Stories** | keine |

|  |  |
| --- | --- |
| **User Story-ID** | 2 |
| **User Story-Beschreibung** | Als Lehrender möchte ich bei der Registrierung einen Vor- und Nachnamen, Passwort, E-Mail-Adresse, Profilbild, private Adresse, Lehrstuhl und ein Forschungsgebiet angeben, so dass ein Profil mit diesen Angaben für mich erstellt wird. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | hoch |
| **Autor(en)** | Krolik, Sammy-Steve |
| **Abhängigkeiten zu anderen User Stories** | keine |

|  |  |
| --- | --- |
| **User Story-ID** | 3 |
| **User Story-Beschreibung** | Als Studierender möchte ich mein Passwort, das Profilbild, die private Adresse und mein Studienfach bearbeiten können, so dass ich die Daren aktuell halten kann. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | hoch |
| **Autor(en)** | Krolik, Sammy-Steve |
| **Abhängigkeiten zu anderen User Stories** | 8, 14, 15 |

|  |  |
| --- | --- |
| **User Story-ID** | 4 |
| **User Story-Beschreibung** | Als Lehrender möchte ich mein Passwort, das Profilbild, die private Adresse, den Lehrstuhl und das Forschungsgebiet bearbeiten können, so dass ich die Daten aktuell halten kann. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | hoch |
| **Autor(en)** | Krolik, Sammy-Steve |
| **Abhängigkeiten zu anderen User Stories** | 8, 14, 15 |

(Verwaltung von Lehrveranstaltungen)

|  |  |
| --- | --- |
| **User Story-ID** | 5 |
| **User Story-Beschreibung** | Als Lehrender möchte ich eine Lehrveranstaltung mit Hilfe eines Formulars anlegen können, um unabhängig von technischem Support meine Lehre planen zu können. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | hoch |
| **Autor(en)** | Ridder, Christopher |
| **Abhängigkeiten zu anderen User Stories** | keine |

|  |  |
| --- | --- |
| **User Story-ID** | 6 |
| **User Story-Beschreibung** | Als Lehrender möchte ich eine oder mehrere neue Lehrveranstaltungen mit Hilfe einer CSV-Datei anlegen können, um bereits vorhandene Informationen zu Lehrveranstaltungen wiederverwenden zu können. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | hoch |
| **Autor(en)** | Ridder, Christopher |
| **Abhängigkeiten zu anderen User Stories** | 10 |

|  |  |
| --- | --- |
| **User Story-ID** | 7 |
| **User Story-Beschreibung** | Als Lehrender möchte ich veranstaltungsbezogen Lehrmaterialien bereitstellen können, um die Zusammenarbeit mit den Teilnehmern zu vereinfachen. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | hoch |
| **Autor(en)** | Ridder, Christopher |
| **Abhängigkeiten zu anderen User Stories** | 5, 6, 9 |

|  |  |
| --- | --- |
| **User Story-ID** | 8 |
| **User Story-Beschreibung** | Als LV-Teilnehmer möchte ich zu einer Lehrveranstaltung, an der ich teilnehme, eine Übersichtsseite einsehen können, um nicht an verschiedenen Stellen nach den relevanten Informationen suchen zu müssen. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | hoch |
| **Autor(en)** | Ridder, Christopher |
| **Abhängigkeiten zu anderen User Stories** | 5, 6, 7, 9 |

|  |  |
| --- | --- |
| **User Story-ID** | 9 |
| **User Story-Beschreibung** | Als Lehrender möchte ich ein CSV-Template, zum Anlegen einer oder mehrerer Veranstaltungen, herunterladen können, um die Anforderungen, die an einen CSV-Upload gestellt werden, erfüllen zu können. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | hoch |
| **Autor(en)** | Ridder, Christopher |
| **Abhängigkeiten zu anderen User Stories** | keine |

(Zuordnung von Nutzern zu Lehrveranstaltungen)

|  |  |
| --- | --- |
| **User Story-ID** | 10 |
| **User Story-Beschreibung** | Als Nutzer möchte ich eine (sortierte) Liste aller Lehrveranstaltungen einsehen können, um meine Lehrveranstaltungen zu kennen. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | hoch |
| **Autor(en)** | Metwally, Omar |
| **Abhängigkeiten zu anderen User Stories** | 5, 14 |

|  |  |
| --- | --- |
| **User Story-ID** | 11 |
| **User Story-Beschreibung** | Als Nutzer möchte ich einer Lehrveranstaltung beitreten können, um an den Lehrveranstaltungen teilnehmen zu können. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | hoch |
| **Autor(en)** | Metwally, Omar |
| **Abhängigkeiten zu anderen User Stories** | 5, 10, 14 |

|  |  |
| --- | --- |
| **User Story-ID** | 12 |
| **User Story-Beschreibung** | Als Lehrender möchte ich die Studierenden über ihren Namen oder der Matrikelnummer suchen können, um sie somit manuell in ihre Veranstaltungen einzuschreiben. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | hoch |
| **Autor(en)** | Metwally,Omar |
| **Abhängigkeiten zu anderen User Stories** | 1, 2, 4, 14 |

|  |  |
| --- | --- |
| **User Story-ID** | 13 |
| **User Story-Beschreibung** | Als Nutzer möchte ich eine eingeteilte Teilnehmerliste in zwei Kategorien einsehen können, wobei mein eigener Eintrag visuell hervorgehoben wird, um eine bessere Übersicht zu erhalten. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | hoch |
| **Autor(en)** | Metwally, Omar |
| **Abhängigkeiten zu anderen User Stories** | 5, 11, 14 |

(Nutzerprofile)

|  |  |
| --- | --- |
| **User Story-ID** | 14 |
| **User Story-Beschreibung** | Als Nutzer möchte ich mit einem Klick in der Teilnehmerliste ein Profil aufrufen können und dieses daraufhin angezeigt bekommen, um alle Informationen einsehen zu können. |
| **Geschätzter Realisierungsaufwand** | 3 Tage |
| **Priorität** | hoch |
| **Autor(en)** | Budiselic, Antonella |
| **Abhängigkeiten zu anderen User Stories** | 1, 3, 13 |

(Login-Vorgang und Startfenster)

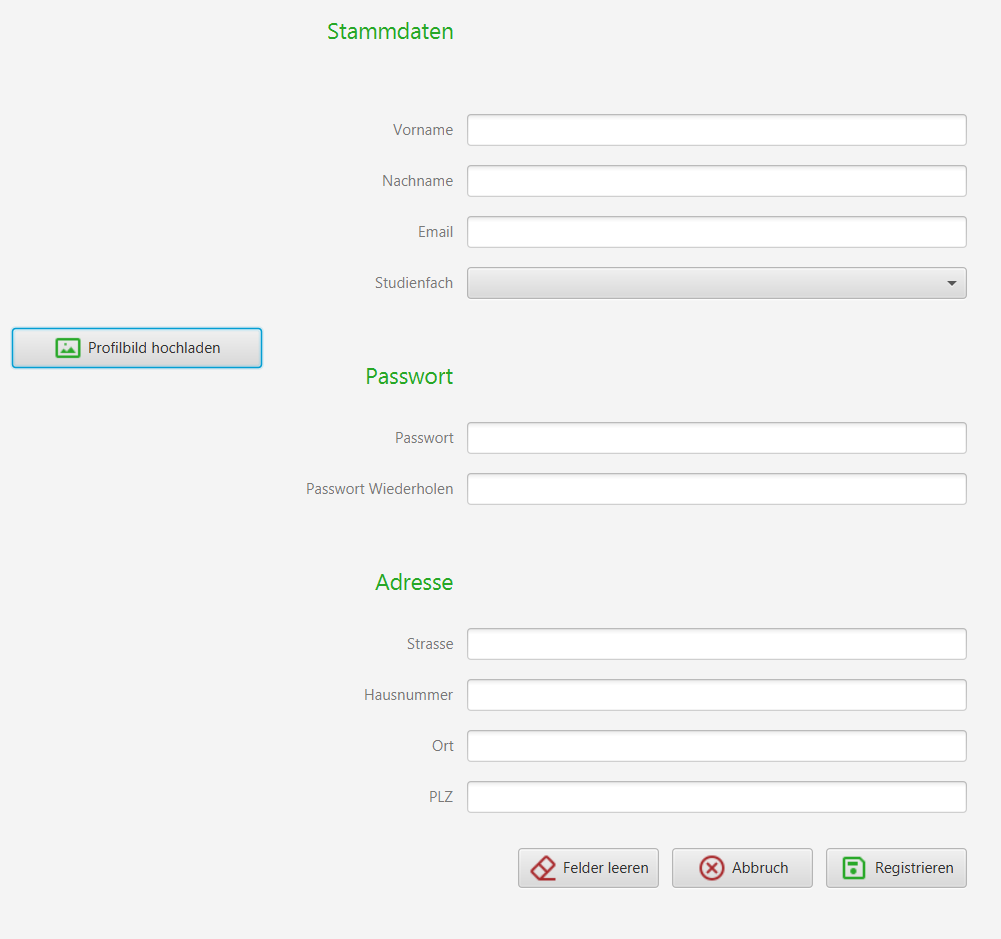
|  |  |
| --- | --- |
| **User Story-ID** | 15 |
| **User Story-Beschreibung** | Als Nutzer möchte ich eine Möglichkeit haben, mich entweder mit der E-Mail oder meiner Matrikelnummer und einem Passwort anmelden zu können, um die Software nutzen zu können. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | hoch |
| **Autor(en)** | Weingarth, Winston Julius |
| **Abhängigkeiten zu anderen User Stories** | keine |

|  |  |
| --- | --- |
| **User Story-ID** | 16 |
| **User Story-Beschreibung** | Als Nutzer möchte ich Lehrveranstaltungen suchen können, um mich zu informieren. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | hoch |
| **Autor(en)** | Weingarth, Winston Julius |
| **Abhängigkeiten zu anderen User Stories** | keine |

## Papierprototypen

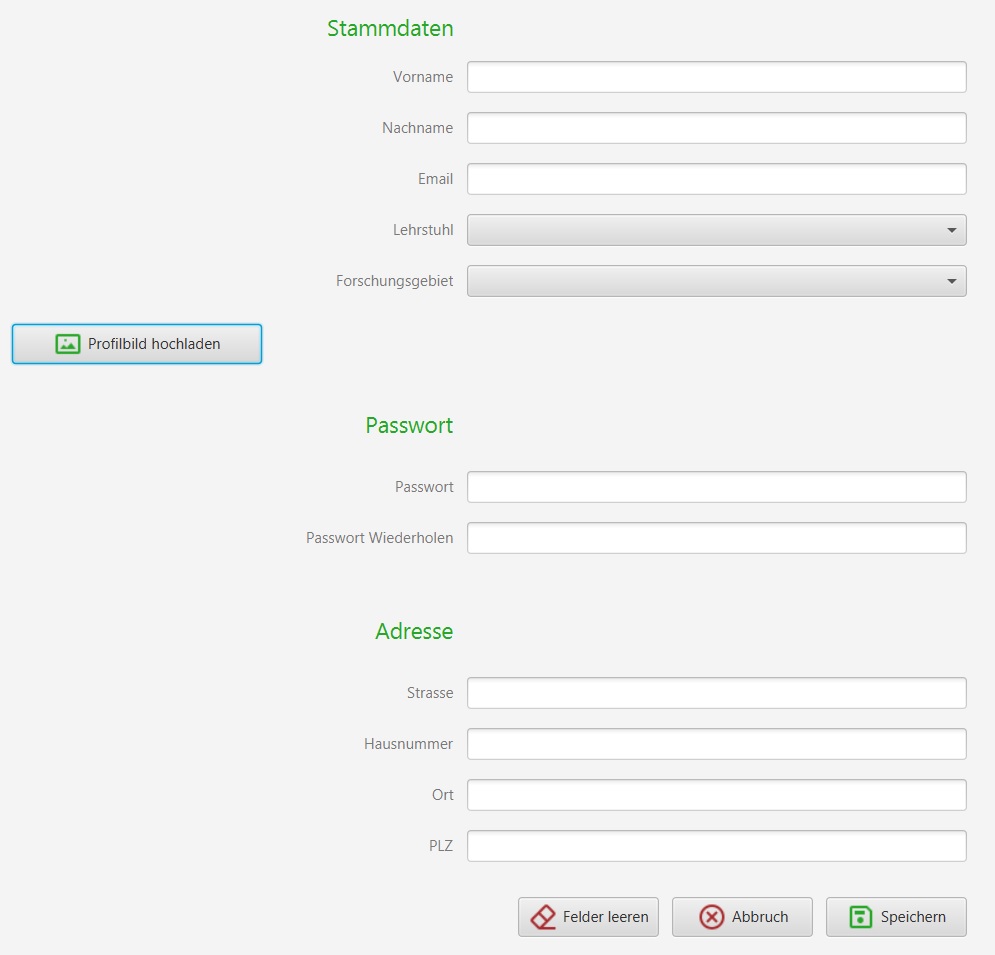
Das Erstellen eines Papierprototypen dient als Methode des Brainstormings, Designs, Herstellens, Testens und des Kommunizierens von Benutzer Interfaces.

### **Papierprototyp Registration Student**



Erläuterung: Das Fenster das man sieht wenn man auf dem LOGIN - Screen auf registrieren als Studierender auf Registrieren drückt

### **Papierprototyp Registration Lehrender**

****

Erläuterung: Das Fenster sieht man wenn man als Lehrender sich Registrieren möchte, dabei wird ein anderer Registrieren Link benötigt als beim Studierenden

### **Papierprototyp Lehrveranstaltung anlegen**

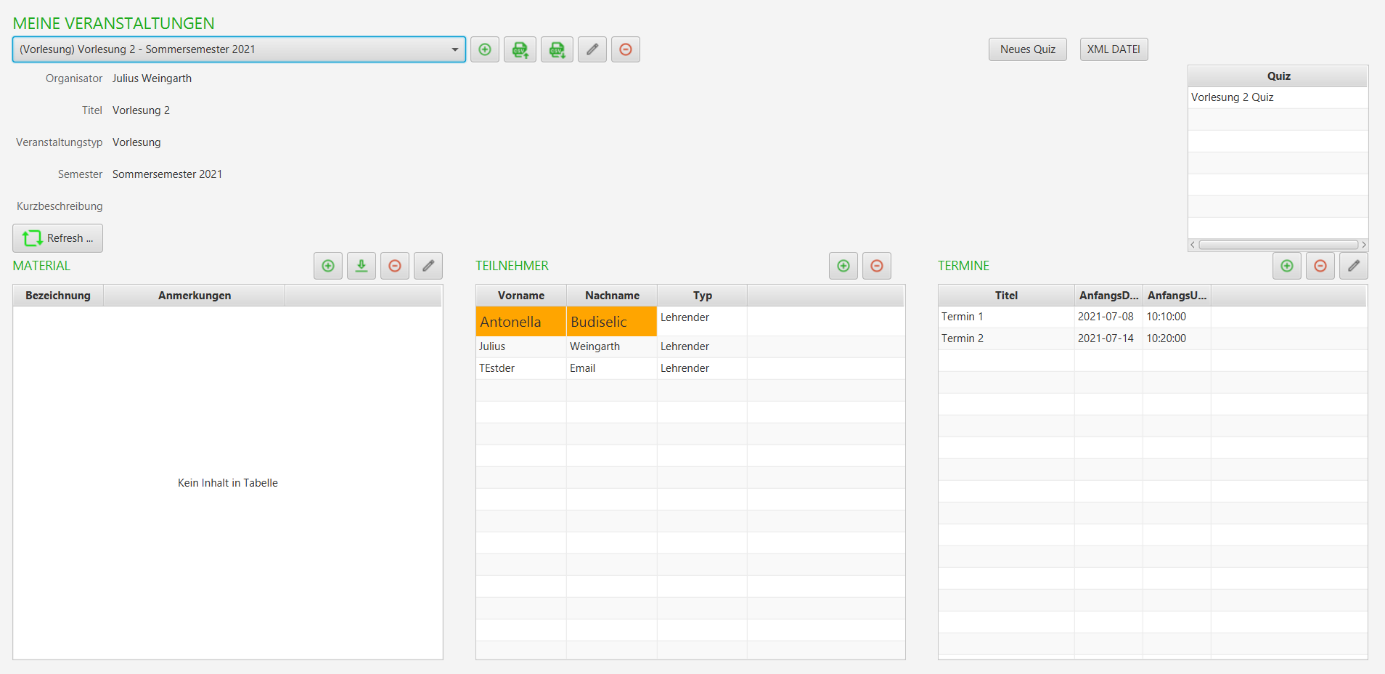


**Ein Bild, das Text enthält.

Automatisch generierte Beschreibung**

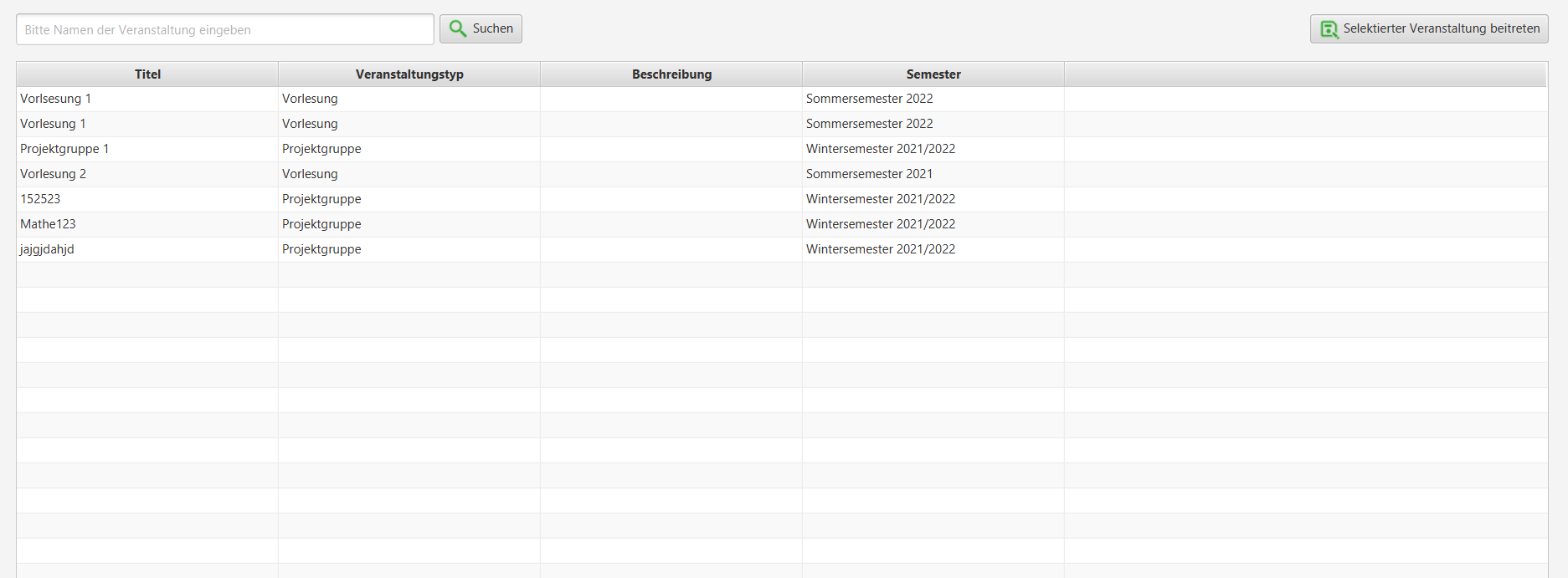
Erläuterung: Der Lehrende hat die Möglichkeit auf der Lehrveranstaltungsansicht eine Veranstaltung manuell zu erstellen, wie hier zu sehen, oder als CSV-Datei über den CSV-Upload Button.

### **Papierprototyp Lehrveranstaltungsübersicht**

****

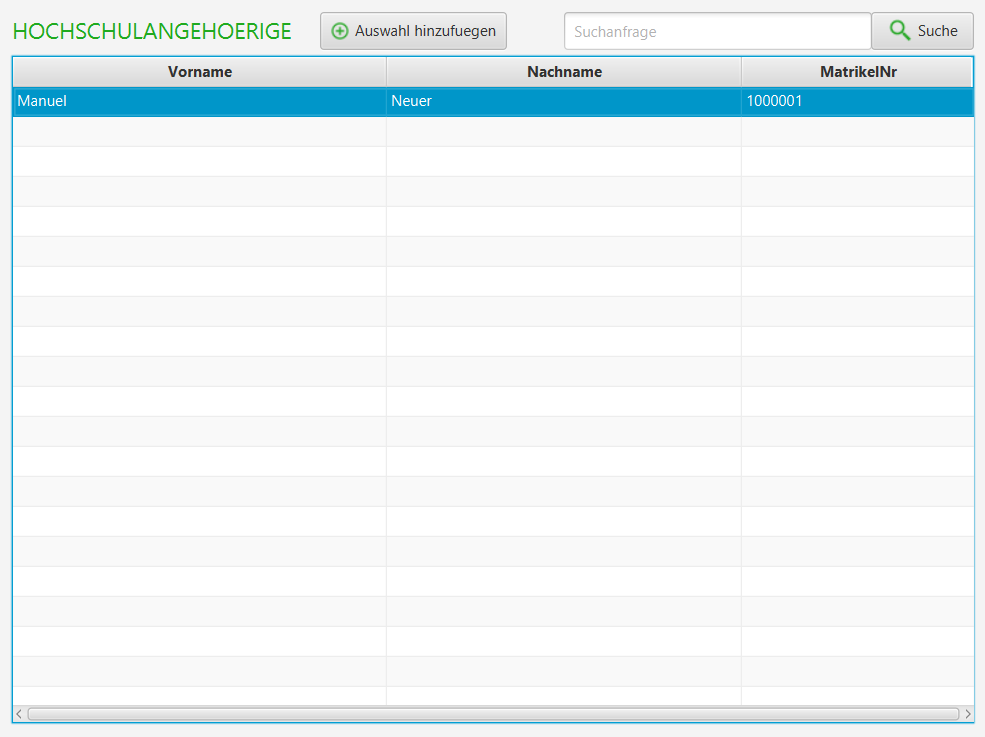
Erläuterung: In der LV Übersicht gibt es die Möglichkeit als Lehrender Material hochzuladen. Außerdem zeigt der Prototyp den Aufbau der Übersicht, man sieht die Teilnehmerliste, die Lernmaterialien und man hat als Lehrender die Möglichkeit neue Dateien hochzuladen. Die anderen Funktionalitäten, wie Termine und Quizerstellung werden in Zyklus 1 und 2 erläutert.

### **Papierprototyp Lehrveranstaltung suchen**

****

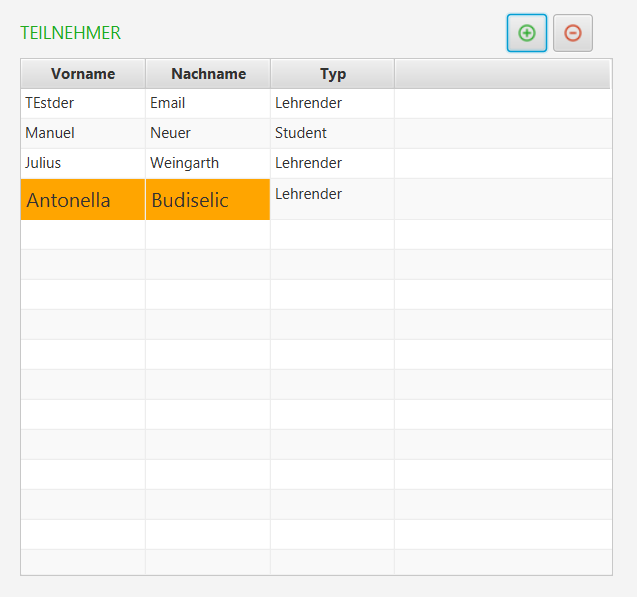
Erläuterung: Man hat die Möglichkeit alle Lehrveranstaltungen einzusehen. Diese sind sortierbar nach dem Alphabet, dem Veranstaltungstyp oder dem Semester. Es besteht außerdem die Möglichkeit nach einer bestimmten Lehrveranstaltung zu suchen, indem man in dem Textfeld den Namen der Veranstaltung eingibt. Daraufhin werden die Veranstaltungen mit den enthaltenen Namen angezeigt und können über den "Selektierter Veranstaltung beitreten" Button beigetreten werden.

### **Papierprototyp Teilnehmer zu LV hinzufügen**

****

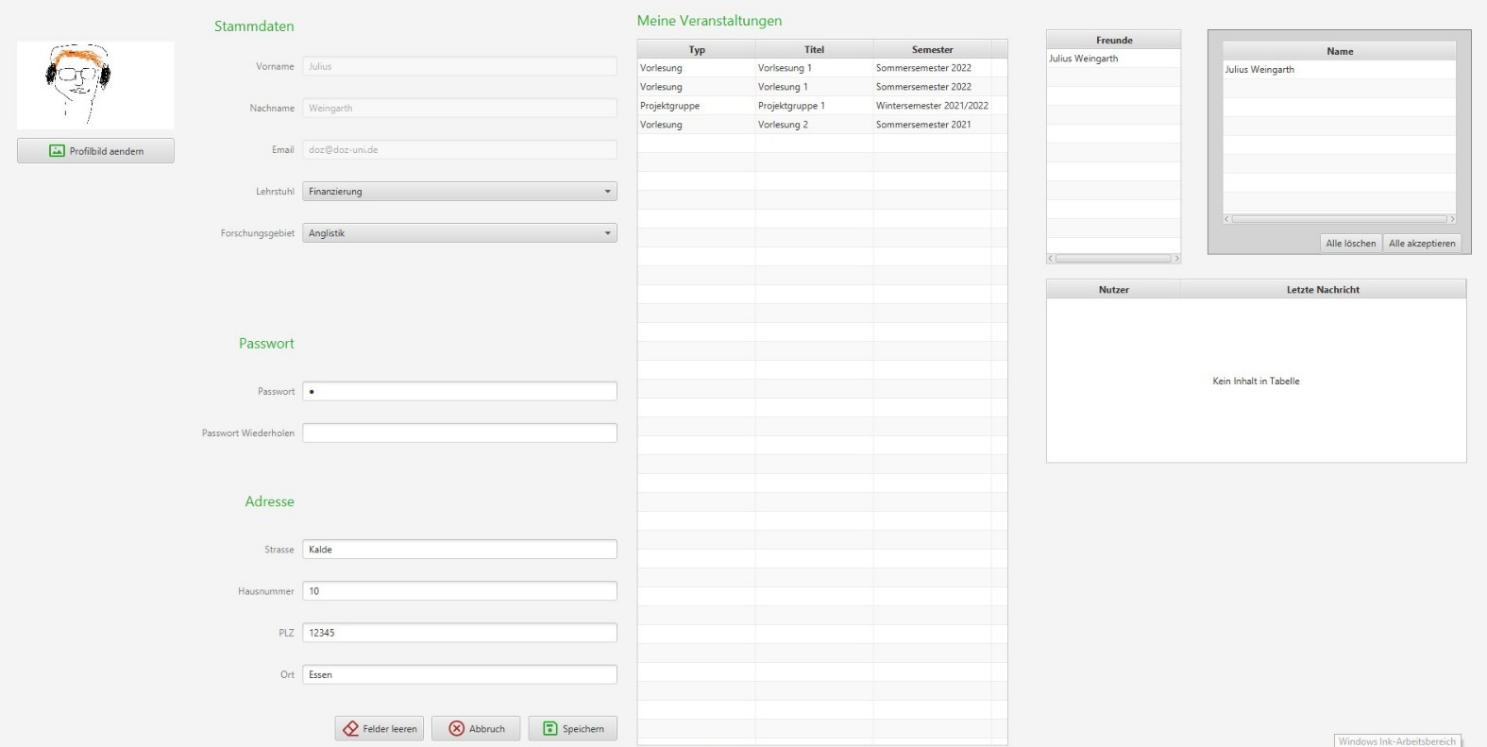
Erläuterung: In dem Prototypen für die Übersicht der Lehrveranstaltung kann man sehen, dass sich über der Teilnehmerliste ein Button (+) befindet. Klickt man auf diesen erhält man alle Studierenden. Nun kann man anhand der Matrikelnummer oder anhand des Namens die jeweiligen Studierenden filtern. Möchte man eine Person hinzufügen klickt man auf dessen Namen und wählt „Auswahl hinzufügen“.

### **Papierprototyp Teilnehmerliste**

****

Erläuterung: Dieser Prototyp zeigt die Teilnehmerliste von der Lehrveranstaltungsübersicht. Der eigene Name wird orange hervorgehoben, es wird angezeigt, ob es sich um einen Lehrenden oder Studierenden handelt und es besteht die Möglichkeit wie zuvor erwähnt neue Leute zu der Veranstaltung hinzuzufügen.

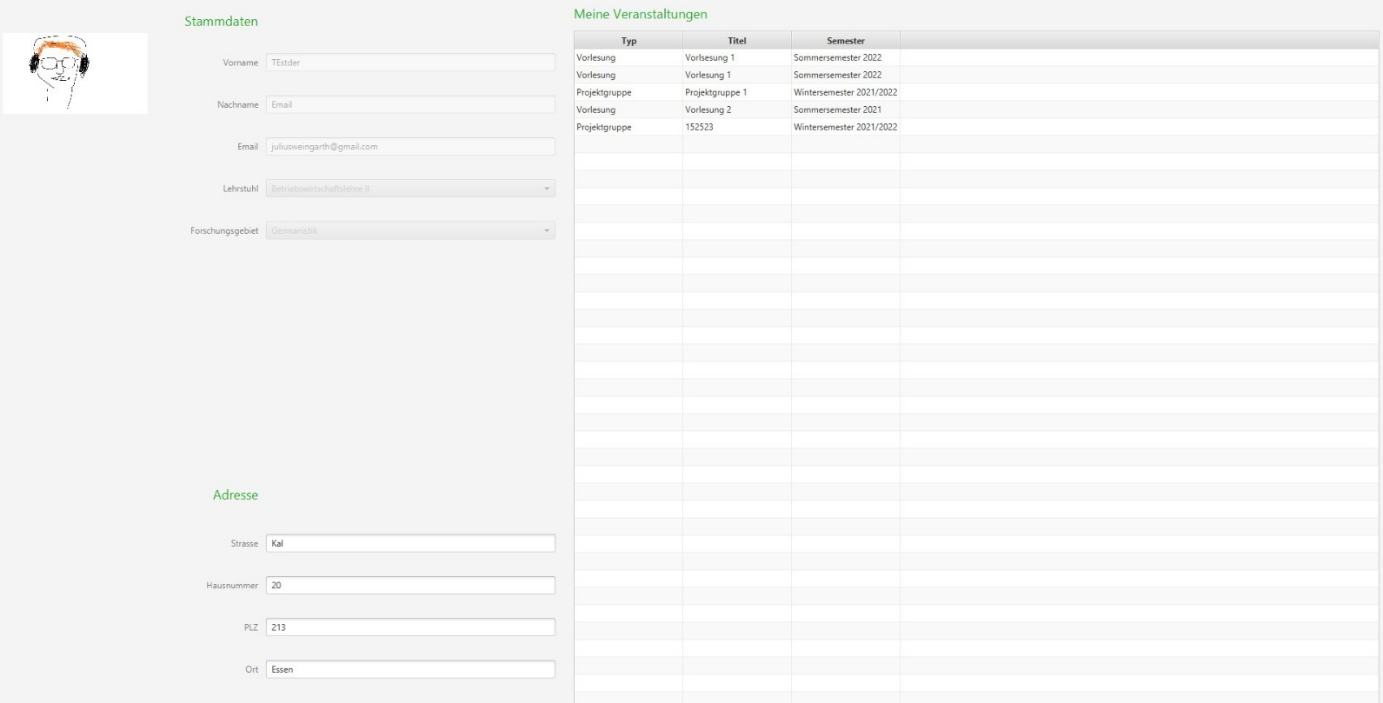
### **Papierprototyp Profilansichten**

****

Erläuterung: Wenn der Nutzer auf sein eigenes Profil klickt hat er eine uneingeschränkte Sicht auf alle Informationen, die gegeben sind. Somit auch auf seine Adresse und Matrikelnummer, die für andere Nutzer nicht sichtbar sein sollten (außer für Lehrende). Außerdem besteht die Möglichkeit ein Profilbild einzufügen/zu ändern, seine Adresse, Studiengang und das Passwort zu ändern und alle teilnehmenden/organisierenden Lehrveranstaltungen einzusehen. Die Funktion der Freundschaften und des Privatchats werden in Zyklus 3 erläutert.

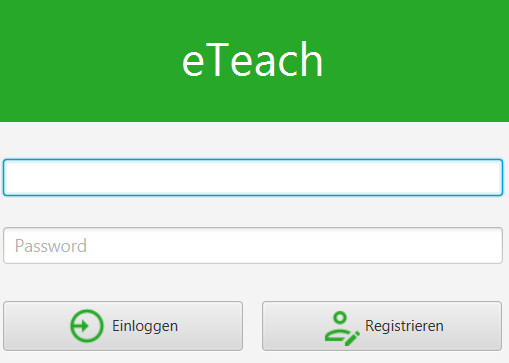
****

Erläuterung: Wenn ein Studierender das Profil eines anderen Nutzers öffnet, sollte eine eingeschränkte Sicht gegeben sein. Hier werden die persönlichen Daten wie z.B. die Adresse oder die Matrikelnummer nicht angezeigt. Die jeweiligen Veranstaltungen werden dennoch angezeigt.

****

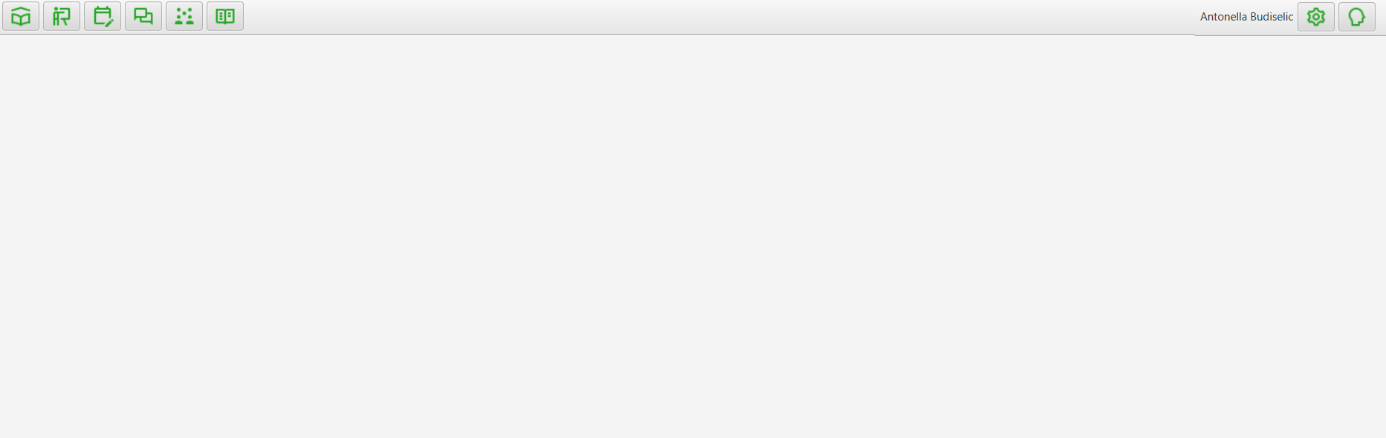
Erläuterung: Ein Lehrender hat eine absolut uneingeschränkte Sicht sowohl auf sein eigenes Profil, das im Gegensatz zu den studentischen Profilen teilweise andere Daten beinhaltet, als auch auf die Profile der Studenten. Diese werden wie im Prototyp für die eigene Ansicht für den Lehrenden angezeigt, abgesehen von der Möglichkeit Änderungen vorzunehmen.

### **Papierprototyp Login**

****

Erläuterung: Ein simples Login Fenster wo die Mail-Adresse oder die MatrikelNr. + das Passwort abgefragt wird.

### **Papierprototyp Dashboard**

****

Erläuterung: Die Startseite der Plattform nachdem man sich eingeloggt hat. Von hier aus kann man sein eigenes Profil, die Lehrveranstaltungsansicht, die Kalenderfunktion (im Zyklus 3), die Projektgruppenansicht (im Zyklus 2) und die Veranstaltungssuchfunktion öffnen. Oben rechts besteht außerdem die Möglichkeit sich wieder abzumelden.

## Szenarien (MSCs)

MSC-Diagramme inkl. Bezeichner

**Hauptszenario:** Typische Folge von Interaktionsschritten zur Erreichung des Ziels bzw. der Ziele einer oder mehrerer User Stories

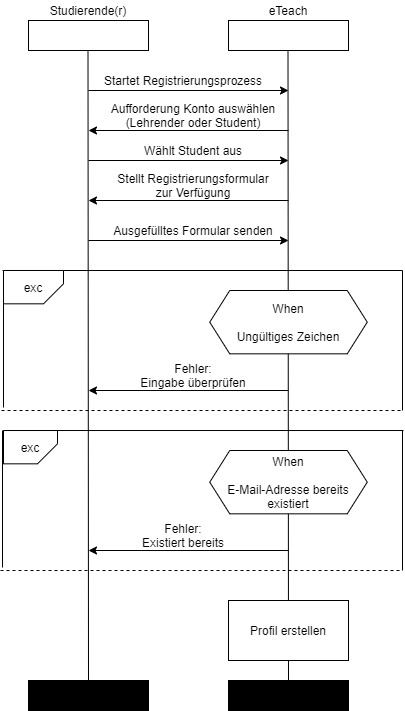
**Alternativszenario:** Alternative Interaktionsschritte, die das Hauptszenario (oder Teile davon) ersetzen und ebenfalls zur Erfüllung der Ziele hinter den User Stories führen

**Ausnahmeszenario:** Interaktionen in Ausnahmefällen, die dazu führen, dass nicht alle Ziele hinter den abgebildeten User Stories erreicht werden

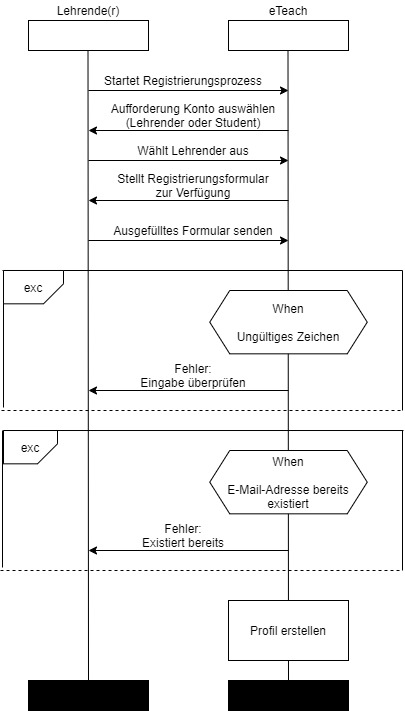
Szenarien dienen als Grundlage zur Definition von Testfällen

Erläuterung der MSCs (beispielsweise getroffene Annahmen

### **Szenario Registration Student**

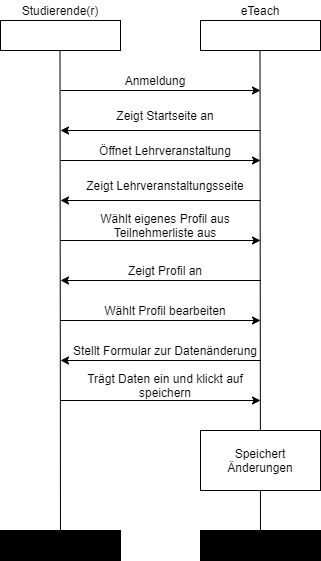
  
Erläuterung: Der Studierende startet das Programm und wählt registrieren aus. Das Programm fragt ob ein Nutzerkonto für einen Studierenden oder Lehrenden erstellt werden soll. Der Studierende wählt Konto für Student(in) erstellen aus. Das Programm stellt das Registrationsformular für Studierende zur Verfügung. Der Studierende trägt in dem Formular den Vor- und Nachnamen, die private Adresse, das Studienfach, ein Profilbild, die E-Mail-Adresse und ein Passwort ein. Anschließend wird der Button "Registrieren" gedrückt. Das Programm erstellt ein Profil mit den angegebenen Daten und einer vom Programm vergebenen eindeutigen 7 -stelligen Matrikelnummer. [Ausnahmefall 1: Der Studierende hat bei der Eingabe des Vor- oder Nachnamens ein ungültiges Zeichen (kein Buchstabe) eingegeben. Nach Betätigung des "Registrieren" Knopfes, gibt das Programm die Fehlermeldung: "Ungültiger Vor- oder Nachname eingegeben. Bitte prüfen Sie ihre Eingaben" aus.] [Ausnahmefall 2: Der Studierende hat bei der Eingabe der E-Mail-Adresse eine ungültige E-Mail-Adresse eingegeben. (Domain [@...]) fehlt. Nach Betätigung des "Registrieren" Knopfes, gibt das Programm die Fehlermeldung: "Ungültige E-Mail-Adresse eingegeben. Bitte prüfen Sie ihre Eingaben" aus] [Ausnahmefall 3: Das Programm prüft vor dem Anlegen des Profils ob bereits ein Profil mit der angegebenen E-Mail-Adresse existiert. Wenn mit der E-Mail-Adresse bereits ein Profil angelegt wurde gibt das Programm die Fehlermeldung: "Es existiert bereits ein Profil mit dieser E-Mail-Adresse" aus und beendet der Registrationsvorgang.

### **Szenario Registration Lehrender**



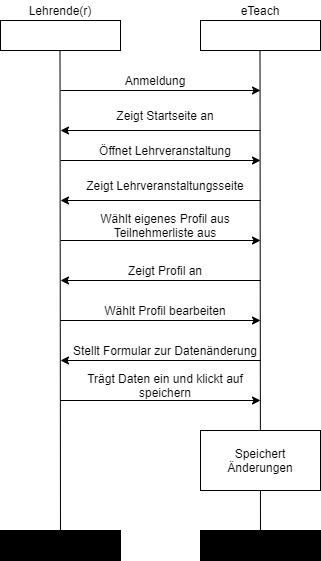
Erläuterung: Der Lehrende startet das Programm und wählt registrieren aus. Das Programm fragt ob ein Nutzerkonto für einen Studierenden oder Lehrenden erstellt werden soll. Der Lehrende wählt Konto für Lehrende(r) erstellen aus. Das Programm stellt das Registrationsformular für Lehrende zur Verfügung. Der Lehrende trägt in dem Formular den Vor- und Nachnamen, die private Adresse, den Lehrstuhl, das Forschungsgebiet, ein Profilbild, die E-Mail-Adresse und ein Passwort ein. Anschließend wird der Button "Registrieren" gedrückt. Das Programm erstellt ein Profil mit den angegebenen Daten. [Ausnahmefall 1: Der Lehrende hat bei der Eingabe des Vor- oder Nachnamens ein ungültiges Zeichen (kein Buchstabe) eingegeben. Nach Betätigung des "Registrieren" Knopfes, gibt das Programm die Fehlermeldung: "Ungültiger Vor- oder Nachname eingegeben. Bitte prüfen Sie ihre Eingaben" aus.] [Ausnahmefall 2: Der Lehrende hat bei der Eingabe der E-Mail-Adresse eine ungültige E-Mail-Adresse eingegeben. (Domain [@...]) fehlt. Nach Betätigung des "Registrieren" Knopfes, gibt das Programm die Fehlermeldung: "Ungültige E-Mail-Adresse eingegeben. Bitte prüfen Sie ihre Eingaben" aus] [Ausnahmefall 3: Das Programm prüft vor dem Anlegen des Profils ob bereits ein Profil mit der angegebenen E-Mail-Adresse existiert. Wenn mit der E-Mail-Adresse bereits ein Profil angelegt wurde gibt das Programm die Fehlermeldung: "Es existiert bereits ein Profil mit dieser E-Mail-Adresse" aus und beendet der Registrationsvorgang.]

### **Szenario Profil bearbeiten Student**

****

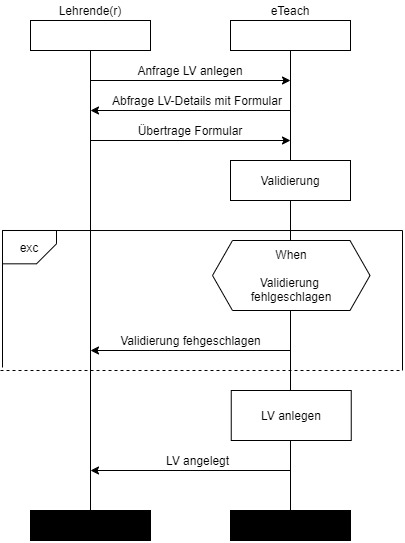
Erläuterung: Der Studierende meldet sich in seinem Profil an. Das Programm zeigt die Startseite an. Der Studierende öffnet eine Teilnehmerliste von der Startseite. Das Programm zeigt die Teilnehmerliste an. Studierende wählt eigenes Profil in der Teilnehmerliste aus. Das Programm zeigt dem Studierenden das eigene Profil an. Studierender wählt "Daten bearbeiten" Button aus. Das Programm stellt die änderbaren Daten mit einem Feld zum Eingeben der neuen Daten zur Verfügung. Änderbare Daten sind Profilbild, private Adresse, Studienfach und das Passwort. Der Studierende trägt die Daten, die geändert werden sollen in die entsprechenden Felder ein. Studierende wählt Button "Speichern" aus. Das Programm speichert die eingetragenen Änderungen.

### **Szenario Profil bearbeiten Lehrender**

****

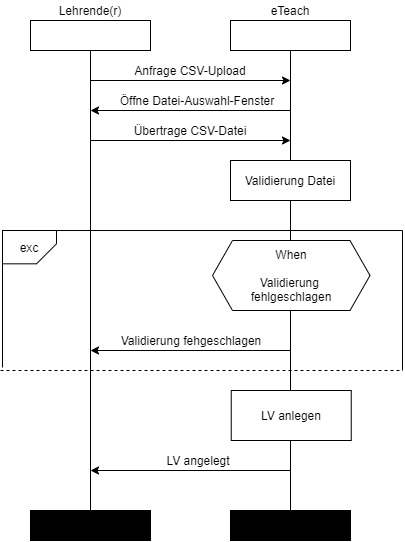
Erläuterung: Der Lehrende meldet sich in seinem Profil an. Das Programm zeigt die Startseite an. Der Lehrende öffnet eine Lehrveranstaltung von der Startseite. Daraufhin öffnet er die Teilnehmerliste. Das Programm zeigt die Teilnehmerliste an. Lehrender wählt eigenes Profil in der Teilnehmerliste aus. Das Programm zeigt dem Lehrenden das eigene Profil an. Lehrender wählt "Daten bearbeiten" Button aus. Das Programm stellt die änderbaren Daten mit einem Feld zum Eingeben der neuen Daten zur Verfügung. Änderbare Daten sind Profilbild, private Adresse, Lehrstuhl, Forschungsgebiet und das Passwort. Der Lehrende trägt die Daten, die geändert werden sollen in die entsprechenden Felder ein. Lehrender wählt Button "Speichern" aus. Das Programm speichert die eingetragenen Änderungen

### **Szenario Lehrveranstaltug anlegen**



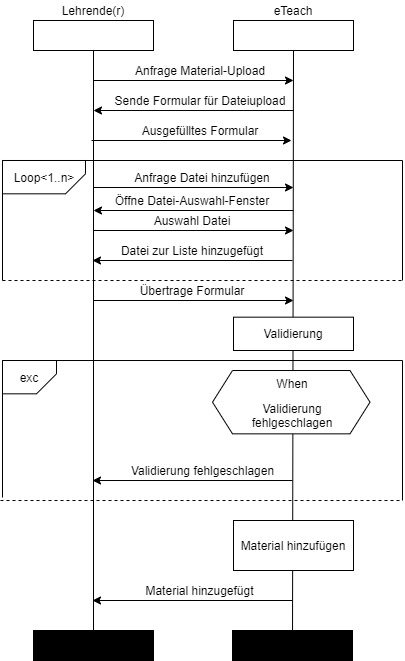
Erläuterung: Nachdem der Lehrende durch Klick auf den Button für das formularbasierte Anlegen einer Veranstaltung geklickt hat, öffnet sich die entsprechende Formularansicht. In dieses trägt der Lehrende alle für die Neuanlage einer Lehrveranstaltung erforderlichen Informationen ein. Ferner kann durch Klick auf den entsprechenden Button das 'Datei-Upload Formular' geöffnet werden, um der Lehrveranstaltung ein oder mehrere Lehrmaterialien hinzuzufügen (s. User Story 2.3: 'Lehrmaterial bereitstellen'). Durch Klick auf den Button "Speichern" sendet der Lehrende das Formular zusammen mit den Lehrmaterialien an das System. Dieses validiert die Formulare. Werden alle Anforderungen erfüllt, so erhält der Lehrende eine Bestätigung, dass die Veranstaltung erfolgreich angelegt wurde. Bei einem negativen Ausgang der Validierung wird der Lehrende darüber informiert, dass das Anlegen nicht erfolgreich war und welche Angaben fehlen bzw. fehlerhaft sind. Die fehlenden bzw. fehlerhaften Angaben werden im Formular hervorgehoben angezeigt und alle anderen Angaben beibehalten, so dass keine Mehrfacheingabe nötig ist.

### **Szenario Lehrveranstaltung mit CSV anlegen**



Erläuterung: Nachdem der Lehrende durch Klick auf den Button für das CSV-basierte Anlegen einer Veranstaltung geklickt hat, öffnet sich ein Datei-Auswahl-Fenster in dem die hochzuladende Datei ausgewählt werden kann. Sobald der Lehrende die Datei-Auswahl bestätigt hat, wird die CSV-Datei an das System gesendet. Das System validiert die CSV-Datei. Entspricht die CSV-Datei den Anforderungen, so erhält der Lehrende eine Bestätigung, dass die Veranstaltung(en) erfolgreich angelegt wurde(n), wobei die Anzahl der erfolgreich gespeicherten Veranstaltungen angezeigt wird. Bei einem negativen Ausgang der Validierung wird der Lehrende darüber informiert, dass das Anlegen aufgrund einer von Fehlern in der CSV-Datei nicht erfolgreich war und der Vorgang abgebrochen wurde.

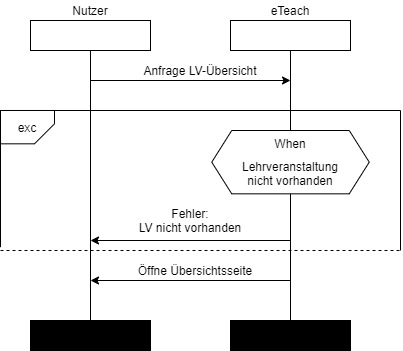
### **Szenario Material Upload**



Erläuterung: Der Lehrende wählt in seinem Profil (s. Szenario 4.2) aus der Gesamtliste aller von ihm organisierten Lehrveranstaltungen die Veranstaltung aus, zu der Material hinzugefügt werden soll. Es öffnet sich das Formular für den Upload von Lehrmaterial, das der Lehrende ausfüllt. Ferner kann der Lehrende mindestens eine oder auch mehrere Dateien für den Upload auswählen.

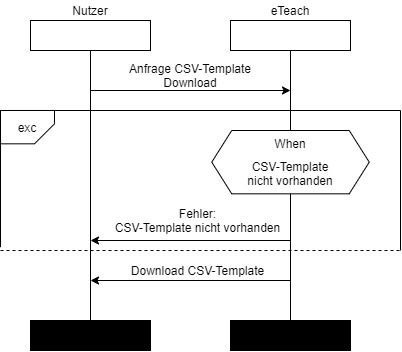
Durch Klick auf den Button "Lehrmaterial hinzufügen" werden das ausgefüllte Formular sowie die angegebenen Dateien an das System übertragen. Dieses validiert sowohl, ob alle Pflichtangaben im Formular gemacht wurden als auch die Zulässigkeit der hochgeladenen Dateien anhand des Dateiformats. Bei einem negativen Ausgang der Validierung wird der Lehrende darüber informiert, dass das der Upload des Lehrmaterials nicht erfolgreich war und welche Angaben fehlen bzw. fehlerhaft sind oder welche der Dateien nicht den Anforderungen entsprechen. Die fehlenden bzw. fehlerhaften Angaben werden im Formular hervorgehoben angezeigt und alle anderen Angaben beibehalten, so dass keine Mehrfacheingabe nötig ist.

### **Szenario Lehrveranstaltungsübersicht**



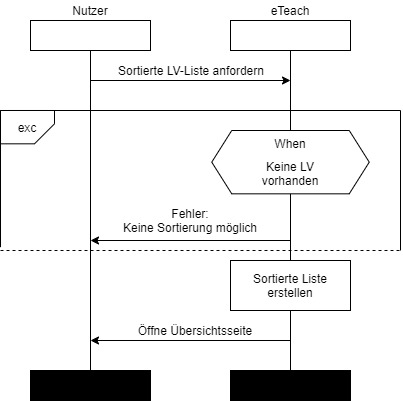
Erläuterung: Der Teilnehmende einer Lehrveranstaltung (Nutzer) wählt in seinem Profil aus der Gesamtliste aller Veranstaltungen, an denen er teilnimmt, eine aus durch Klick auf den Listeneintrag. Daraufhin öffnet sich eine Übersichtsseite, die alle zu Veranstaltung gehörenden Informationen beinhaltet.

### **Szenario CSV Template**



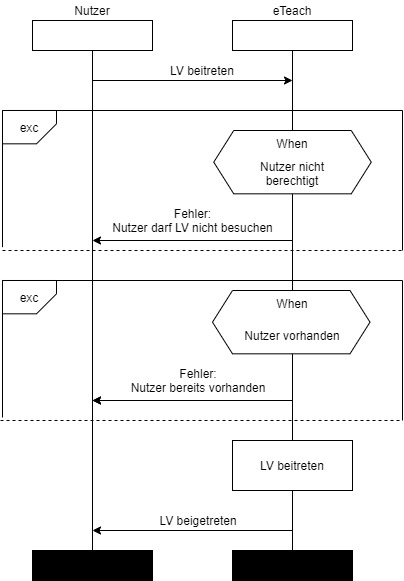
Erläuterung: Sobald der Nutzer auf den Button für das Herunterladen des CSV-Templates für das Anlegen einer oder mehrerer Veranstaltungen geklickt hat, wird der Download des Templates gestartet.

### **Szenario Sortierte LV-Liste**



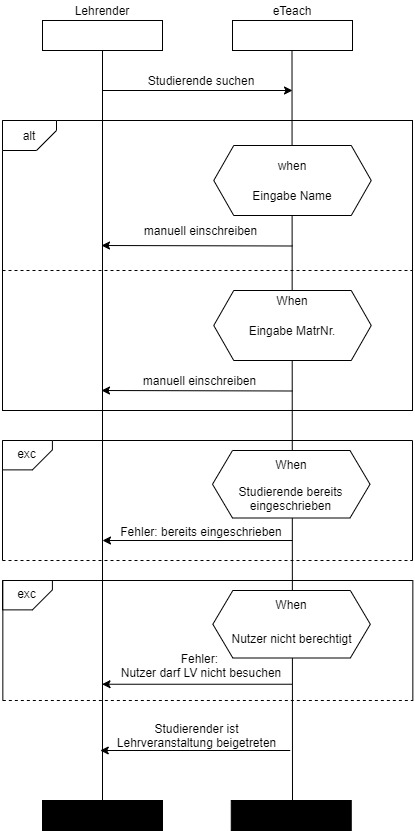
Erläuterung: Der Nutzer will eine sortierte Liste aller Lehrveranstaltungen einsehen. Der Nutzer fordert vom System, über einen Button eine sortierte Liste an. Das System gibt den Nutzer entweder die sortierte Liste aus oder wenn diese Liste nicht vorhanden ist, dann wird eine Exception geworfen

### **Szenario Lehrveranstaltung beitreten**



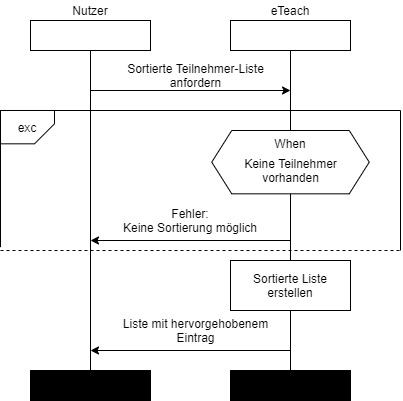
Erläuterung: Über einen Button neben der Lehrveranstaltung, kann der Nutzer der Lehrveranstaltung beitreten. Wenn der Nutzer schon beigetreten ist oder nicht berechtigt ist an der Veranstaltung teilzunehmen, bekommt er eine Fehlermeldung mit einer spezifischen Nachricht. Ist dieser jedoch berechtigt oder noch nicht beigetreten, fügt ihn das System hinzu.

### **Szenario Student zu LV hinzufügen**



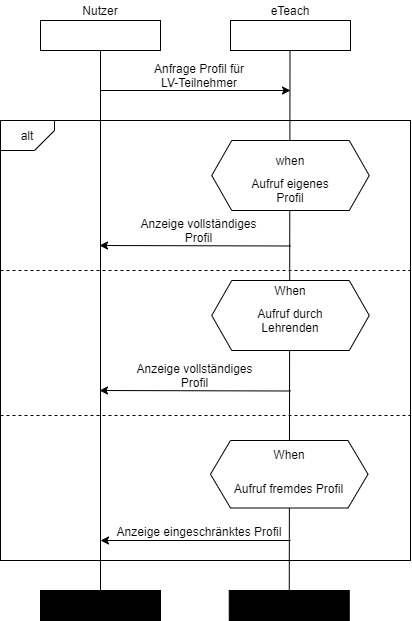
Erläuterung: Der Lehrende besitzt einen Button, wo er spezifisch nach Studierenden suchen kann. Ihm steht die Möglichkeit offen, ob er über den Namen oder Matrikelnummer suchen will. Findet er den gewünschten Studierenden, kann dieser hinzugefügt werden. Ist der Studierende jedoch schon hinzugefügt wurden, oder nicht berechtigt, bekommt der Lehrende eine Fehlermeldung.

### **Szenario Teilnehmerliste**



Erläuterung: Der Nutzer besitzt die Möglichkeit eine Teilnehmerliste mit zwei Kategorien einzusehen, wobei sein eigener Beitrag hervorgehoben wird. Dazu sendet er eine Anfrage an das System, worauf hin er die gewünschte Teilnehmerliste einsehen kann.

### **Szenario Profilansicht**



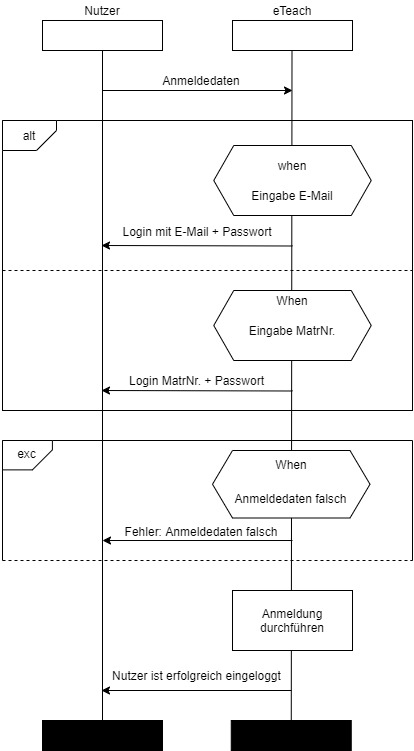
Erläuterung: Nachdem der Nutzer auf einen Namen in der Teilnehmerliste einer Lehrveranstaltung geklickt hat, wird eine Anfrage zur Ansicht eines Teilnehmerprofils an das System gesendet. Dieses prüft die Berechtigungen des Anfragenden. Es sind drei Fälle möglich:

1. Anfragender LV-Teilnehmer entspricht ausgewähltem LV-Teilnehmer: Das System bietet eine uneingeschränkte Sicht auf das Profil (alle Informationen sind einsehbar)

2. Anfragender LV-Teilnehmer unterscheidet sich von ausgewähltem LV-Teilnehmer: Das System bietet eine eingeschränkte Sicht auf das Profil

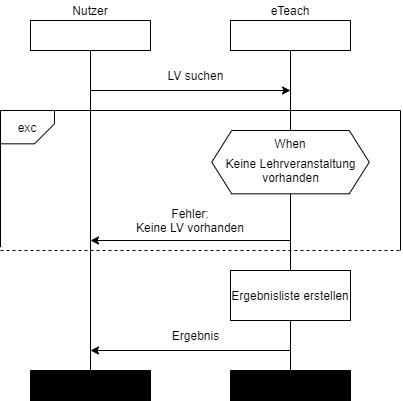
3. Anfragender LV-Teilnehmer ist ein Lehrender: Das System bietet eine uneingeschränkte Sicht auf das Profil (alle Informationen sind einsehbar; äquivalente Sicht wie bei 1.)

### **Szenario Login**



Erläuterung: Der Nutzer besitzt die Möglichkeit über ein Anmeldefenster, sich entweder über der Matrikelnummer oder der E-Mail anzumelden. Sind die Anmeldedaten korrekt gelangt er ins System, sind die Anmeldedaten jedoch unkorrekt bekommt er eine Fehlermeldung.

### **Szenario LV suchen**



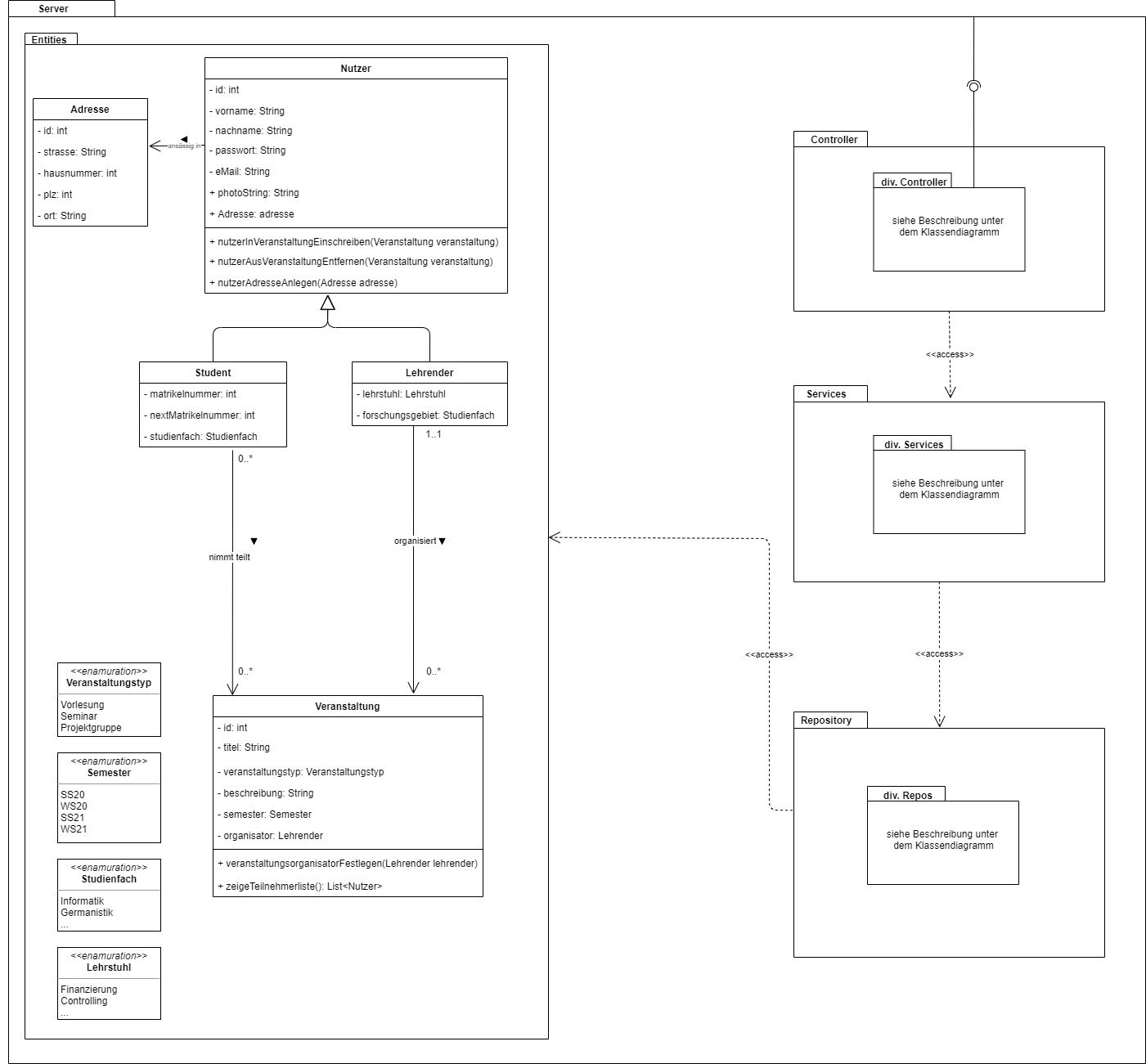
Erläuterung: Über eine Suchfunktion gibt der Nutzer die gesuchte Lehrveranstaltung ein, wobei das System eine Ergebnisliste generiert. Die generierte Ergebnisliste wird dem Nutzer angezeigt.

## 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.

Verantwortlicher: Antonella Budiselic

### **Server**



Beschreibung: Wir haben 4 Haupt-Packages im Server. Das erste Package bildet die Entities ab. Der Nutzer kann entweder ein Student sein oder ein Lehrender. Je nachdem nimmt der Nutzer an einer Lehrveranstaltung teil oder organisiert sie. Des weiteren besitzt die Klasse Lehrveranstaltungen 4 Enums. Zum einen den Lehrveranstaltungstyp Vorlesung, Seminar und Projektgruppe. Das zweite Enum stellt das Semester dar. Das dritte Enum stellt das Studienfach da, worunter sich beliebig viele befinden können. Das letzte Enum besteht aus dem Lehrstuhl, in dem sich ebenfalls beliebig viele befinden können. Die anderen Packages stellen die Controller, Services und Repositories dar. Die jeweiligen Controller, Services und Repositories sind der Anschaulichkeitshalber in einem extra Model dargestellt. Hierbei sei nur angemerkt, dass die Controller auf die Services zugreifen, diese auf die Repositories und diese wiederum auf die Entities. Zusätzlich stellen die Controller im Controller Package eine Verbindung zu den Endpoints auf Clientseite her.

### **Server Controller**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Anmerkung: hier sind alle Controller im Server von Zyklus 1 enthalten.

### **Server Services**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Anmerkung: hier sind alle Services im Server von Zyklus 1 enthalten.

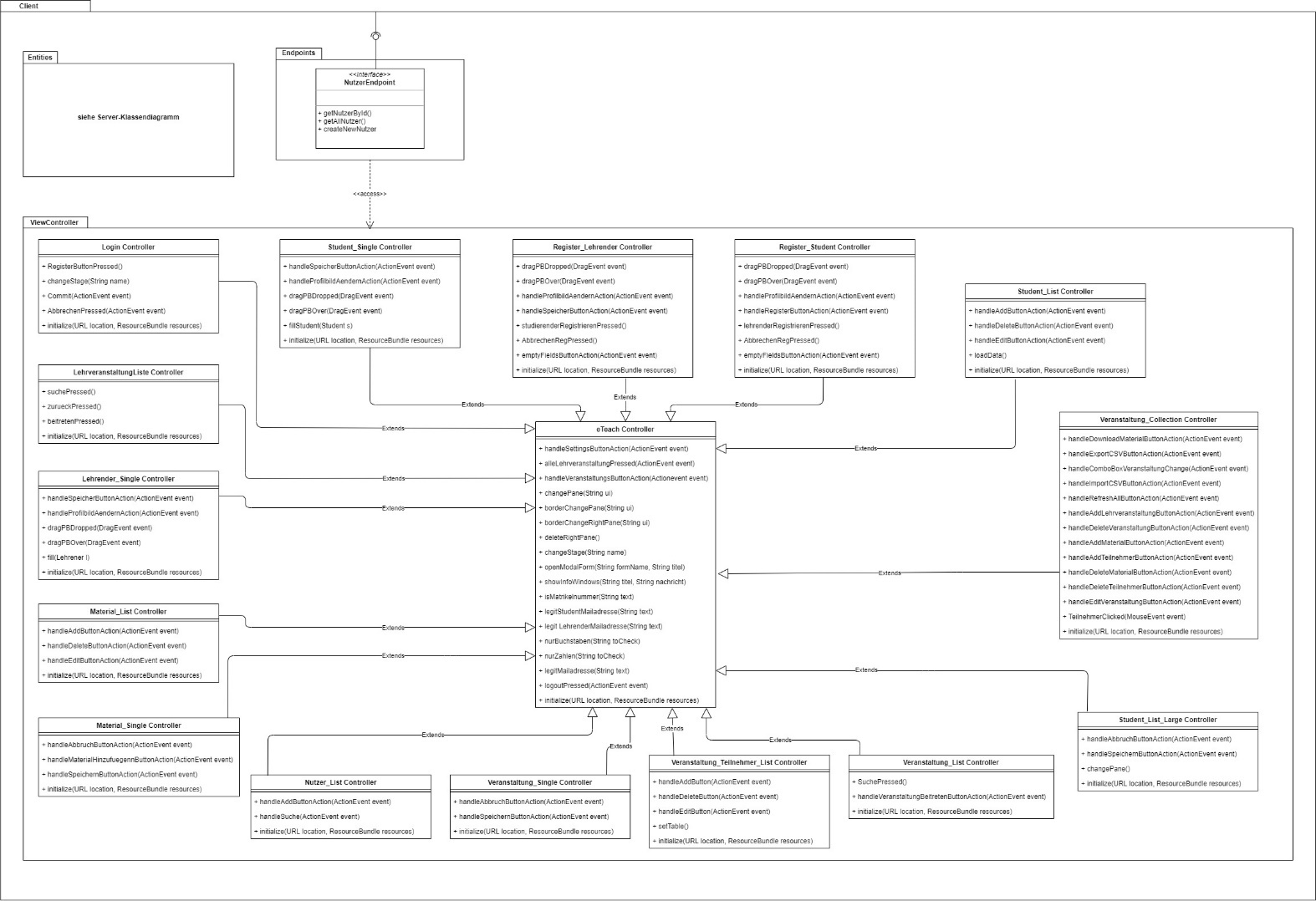
### **Server Repository**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Anmerkung: hier sind alle Repositories im Server von Zyklus 1 enthalten.

### **Client**



Beschreibung: Äquivalent zum Klassendiagramm auf Serverseite befindet sich auf Clientseite ebenfalls das Package Entities. Um Redundanzen zu vermeiden wurde das Model auf einen Verweis auf das Server-Klassendiagramm abstrahiert. Wie auch schon vorher erwähnt ist auch hier die Verbindung zu den Endpoints aufgezeigt. Zuletzt haben wir hier noch die View Controller. Vom Hauptcontroller, auch eTeach Controller genannt, erben alle anderen Controller.

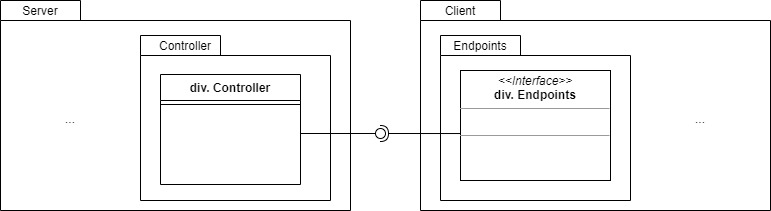
### **Client Endpoints**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Anmerkung: hier sind alle Endpoint im Client von Zyklus 1 enthalten.

### **Verbindung Server-Client**



Anmerkung: hier sieht man die Verbindung zwischen den Controllern auf Serverseite und den Endpoints auf Clientseite.

## Funktionalitätsplanung











## Systemtests

Systemtests sind Tests des Gesamtsystems gegen die Anforderungen nach erfolgreicher Integration. Eingaben und Sollverhalten werden dabei aus der Anforderungs-spezifikation abgeleitet.

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 23.05.2021 | | |
| **Tester** | Omar Metwally | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Der Lehrende „Omar Metwally“ ist am System mit Passwort „1“ und der E-Mail “omar@doz-uni.de registriert | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt den Benutzername  „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 zensiert an. | **√** |
| 3 | Der Benutzer klickt auf „Anmelden“. | Das System zeigt die Meldung „Anmeldung erfolgreich“ auf dem Display an. | **X** |
| **Nachbe-dingung(en)** | Nutzer ist am System angemeldet, Anmeldezeitpunkt ist im System gespeichert. | | **X** |
| **Testurteil** | Test nicht bestanden. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datum** | | 23.05.2021 | | |
| **Tester** | | Omar Metwally | | |
| **SW-Version** | |  | | |
| **Vorbedingung(en)** | | Der Studierende „Thomas Müller“ ist am System mit Passwort „1“ und der E-Mail “thomas@stud-uni.de“ registriert | | |
| **Schritt** | | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | | Der Benutzer gibt die E-Mail „thomas@stud-uni.de“ in das vorgesehene Feld mit der Bezeichnung „MatrikelNr. /MailAddrese“ ein. | Das System zeigt „thomas@stud-uni.de“ auf dem Display an. | **√** |
| 2 | | Der Benutzer gibt das Passwort „1“ auf der Tastatur ein. | Das System zeigt das Passwort durch „\*“-Symbole zensiert an. | **√** |
| 3 | | Der Benutzer klickt auf „Einloggen“. | Es erscheint ein Fenster mit der Meldung „Anmeldung erfolgreich! Weiterleitung an eTeach“. | **√** |
| 4 | | Der Benutzer bestätigt die Meldung mit einem Klick auf „OK“ | Die grafische Oberfläche der Applikation öffnet sich | **√** |
| (Ausnahmeszenario)  1a | | Der Benutzer gibt die E-Mail „thomas@t-online.de in das vorgesehene Feld mit der Bezeichnung „MatrikelNr. /MailAddrese“ ein. | Das System zeigt „thomas@t-online.de“ auf dem Display an. | **√** |
| 1a1 | | Der Benutzer gibt das Passwort „1“ auf der Tastatur ein. | Das System zeigt das Passwort durch „\*“-Symbole zensiert an. | **√** |
| 1a2 | | Der Benutzer klickt auf „Einloggen“. | Es erscheint ein Fenster mit der Meldung „Mailadresse oder Passwort falsch“. | **√** |
| **Nachbe-dingung(en)** | |  | |  |
| **Testurteil** | Test bestanden. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datum** | | 23.05.2021 | | |
| **Tester** | | Omar Metwally | | |
| **SW-Version** | |  | | |
| **Vorbedingung(en)** | |  | | |
| **Schritt** | | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | | Der Benutzer klickt auf „Registrieren“ | Das System öffnet das Fenster „Registrierung Studenten“ | **√** |
| 2 | | Eingabe der Daten vom Benutzer  „Vorname“ = „Mats“,  „Nachname“ = „Hummels“,  ,,Password“ = „1“,  „Password wiederholen“ = „1“,  ,,Straße“ = „Natostr“,  ,,HausNr“ = „3“  „PLZ“ = „52452“,  Stadt“ = „Essen“,  „E-Mail“ = „mats@stud-uni.de“ | Das System zeigt die gesamten Eingaben auf dem Display an | **√** |
| 3 | | Der Benutzer wählt in der Checkbox den Studiengang „Germanistik“ aus | Das System zeigt ,,Germanistik“ auf dem Display an | **√** |
| 4 | | Der Benutzer klickt auf „Registrieren“ | Es wird zurückgesprungen in das Log-In Fenster | **√** |
| (Ausnahmeszenario)  2a | | Der Benutzer gibt die E-Mail „mats@t-online.de“ in das vorgesehene Feld mit der Bezeichnung „MatrikelNr. /MailAddrese“ ein. | Das System zeigt „mats@t-online.de“ auf dem Display an | **√** |
| 2a1 | | Der Benutzer klickt auf „Registrieren“ | Es erscheint ein Fenster mit der Meldung „Mailadresse nicht korrekt“. | **√** |
| 2b1 | | Der Benutzer gibt als Strasse ,,45essener“ ein | Das System zeigt „45essener“ auf dem Display an | **√** |
| 2b2 | | Der Benutzer klickt auf „Registrieren“ | Es erscheint ein Fenster mit der Meldung „Strasse nicht korrekt“. | **√** |
| **Nachbe-dingung(en)** | |  | |  |
| **Testurteil** | Test bestanden. | | | |

# Zyklus II

## Definition der Verantwortlichen für die Artefakte der Spezifikation



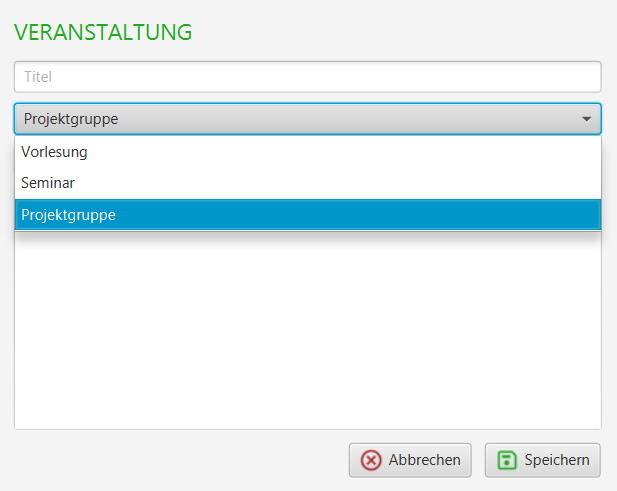
## User-Stories

|  |  |
| --- | --- |
| (1) Projektgruppen |  |
| User-StoryID | 17 |
| User Story-Beschreibung | Als Nutzer(in) möchte ich Projektgruppen erstellen können, die über einen Titel, Chatraum, ToDo-Liste und einer Datenaustauschmöglichkeit verfügen, um mit anderen Nutzern kommunizieren zu können. |
| Geschätzter Realisierungsaufwand | 4 Tage |
| Priorität | hoch |
| Autor(en) | Weingarth, Winston Julius |
| Abhängigkeiten zu anderen User Stories | 5,7 |
|  |  |
| User-StoryID | 18 |
| User Story-Beschreibung | Als Lehrende(r) möchte ich Studierende manuell zur Projektgruppe hinzufügen können, um sie an der Projektgruppe teilnehmen zu lassen. |
| Geschätzter Realisierungsaufwand | 1 Tag |
| Priorität | niedrig |
| Autor(en) | Weingarth, Winston Julius |
| Abhängigkeiten zu anderen User Stories | 12,17 |
|  |  |
| User-StoryID | 19 |
| User Story-Beschreibung | Als Studierende(r) möchte ich nach einer Projektgruppe suchen können, um dieser beitreten zu können. |
| Geschätzter Realisierungsaufwand | 1 Tag |
| Priorität | mittel |
| Autor(en) | Weingarth, Winston Julius |
| Abhängigkeiten zu anderen User Stories | 5,8,10,16,17 |

|  |  |
| --- | --- |
| (3) Quiz |  |
| User-StoryID | 20 |
| User Story-Beschreibung | Als Lehrende(r) möchte ich Multiple Choice Tests für die LV erstellen können, die sowohl manuell als auch durch eine xml-Datei erstellt werden können, um mehr Lernmöglichkeiten zu bieten. |
| Geschätzter Realisierungsaufwand | 5 Tage |
| Priorität | hoch |
| Autor(en) | Ridder, Christopher |
| Abhängigkeiten zu anderen User Stories | 5,7,8 |
|  |  |
| User-StoryID | 21 |
| User Story-Beschreibung | Als Studierende(r) möchte ich diese Tests beliebig oft bearbeiten und meine Resultate ohne Musterlösung einsehen können, damit ich meine Fehler nachvollziehen kann. |
| Geschätzter Realisierungsaufwand | 3 Tage |
| Priorität | hoch |
| Autor(en) | Ridder, Christopher |
| Abhängigkeiten zu anderen User Stories | 5,7,8,22 |
|  |  |
| User-StoryID | 22 |
| User Story-Beschreibung | Als Lehrende(r) möchte ich eine Statistik über die Ergebnisse einschließlich der Beteiligung, der Bestehensquote, die Versuchszahl und der Korrektheitsanzahl einzelner Fragen einsehen können, um auf mögliche Probleme der Inhalte eingehen zu können. |
| Geschätzter Realisierungsaufwand | 2 Tage |
| Priorität | mittel |
| Autor(en) | Ridder, Christopher |
| Abhängigkeiten zu anderen User Stories | 5,7,8,22,23 |

## Papierprototypen

### **Papierprototyp Projektgruppe erstellen**



Erläuterung: Wie auch schon durch Zyklus 1 bekannt, ist das die Ansicht um eine Veranstaltung zu erstellen. Es ist dem Nutzer möglich einen Titel einzugeben, sich zwischen Vorlesung, Seminar und Projektgruppe zu entscheiden (bei Studierenden besteht nur die Option der Projektgruppe). So lässt sich auch das Semester auswählen und mögliche Anmerkungen angeben.

### **Papierprototyp Teilnehmer hinzufügen**



Erläuterung: Auf der Übersichtseite für Projektgruppen wird ein Button mit „Teilnehmer hinzufügen“ angezeigt. Klickt man auf diesen, kann man den ausgewählten Nutzer zur Projektgruppe hinzufügen. Dies ist nur dem Lehrenden möglich.

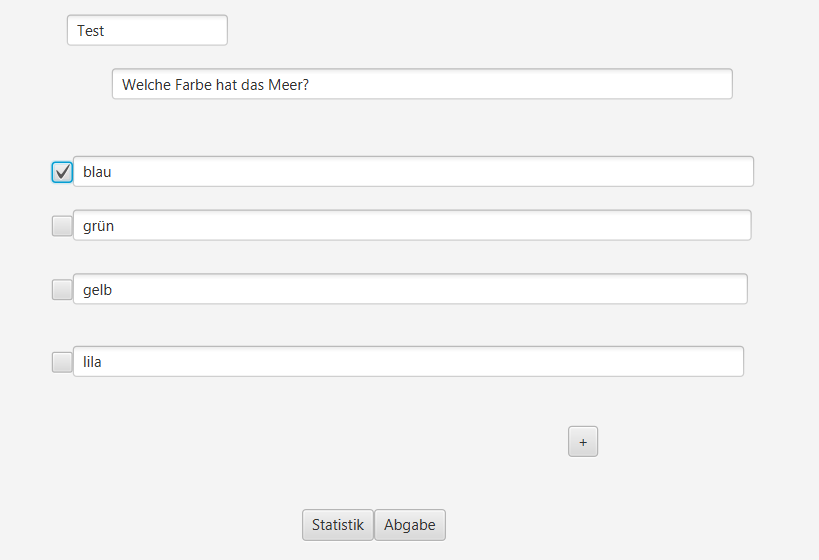
### **Papierprototyp Projektgruppe suchen**

**Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung**

Erläuterung: Wie auch bei der allgemeinen Suchfunktion von Veranstaltungen sind auch Projektgruppen unter dieser Funktion zu finden. Sortiert man man Veranstaltungstyp erhält man alle Projektgruppen.

### **Papierprototyp Quiz erstellen**



Erläuterung: Dieser Prototyp stellt die Funktion der Quiz-Erstellung manuell dar. Es besteht die Möglichkeit dem Test einen Namen zu geben, eine Frage hinzuzufügen, zu löschen oder Antwortmöglichkeiten hinzuzufügen. Der Lehrende kreuzt hier die richtigen Antwortmöglichkeiten an und kann diese über das (+) speichern. Letztlich lässt sich auch noch die Statistik des Tests öffnen.

### **Papierprototyp Quiz mit CSV erstellen**

Ein Bild, das Text, Screenshot, Computer, drinnen enthält.

Automatisch generierte Beschreibung

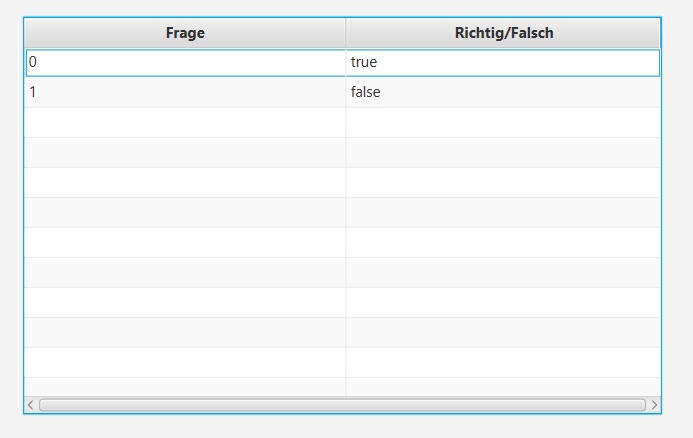
Erläuterung: Wie auch mit der CSV-Datei, wird die xml-Datei auf die gleiche Art hochgeladen. Klickt man auf „XML DATEI“ öffnet sich ein Fenster, in dem man die Datei auswählen und hochladen kann.

### **Papierprototyp Quiz bearbeiten**



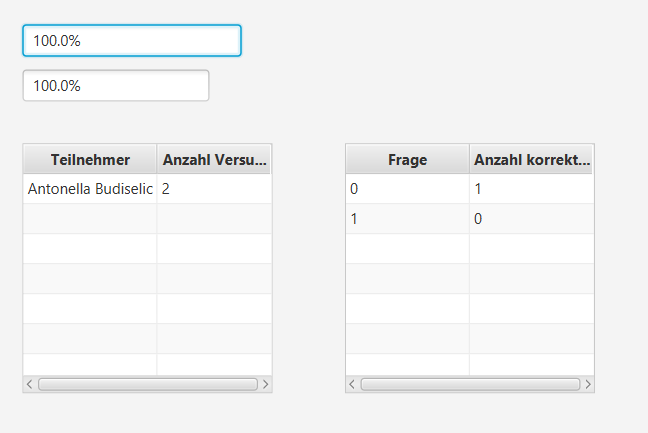
Erläuterung: Die Studierenden hingegen können nur ihre Tests bearbeiten, indem sie die richtigen Antworten auswählen, die Antwort speichern, die Pfeile nach rechts und links zur nächsten/vorigen Frage nutzen und letztlich auf Abgabe klicken.

### **Papierprototyp Quiz Ergebnisse**



Erläuterung: Nach der Abgabe wird automatisch das Ergebnis angezeigt. Dieses bietet keine Musterlösung, da im Allgemeinen nur angezeigt wird, dass die Frage falsch beantwortet wurde.

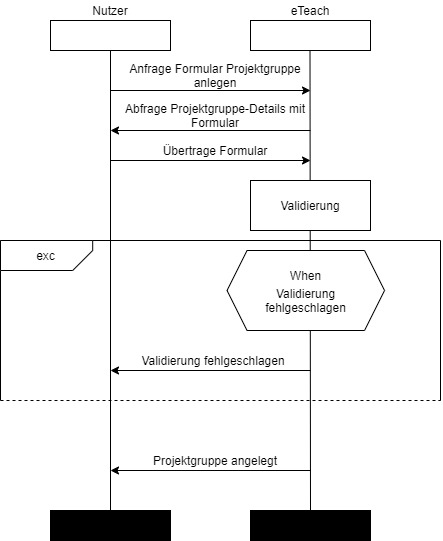
### **Papierprototypen Statistik**



Erläuterung: Hier wird die Statistik der Tests aus Sicht des Lehrenden angezeigt. Alle nötigen Fakten werden in zwei Tabellen dargestellt. Außerdem wird angezeigt wieviel Prozent der Teilnehmer das Quiz bearbeitet haben und wieviel Prozent bestanden haben (mind. 50% des Quiz korrekt).

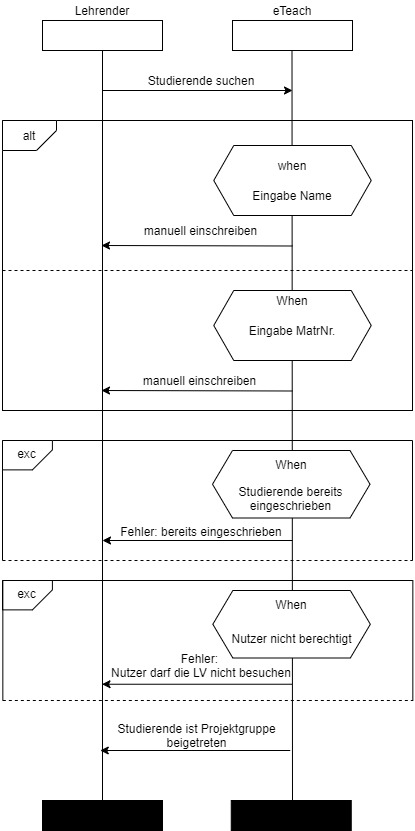
## Szenarien (MSCs)

### **Szenario Projektgruppe erstellen**



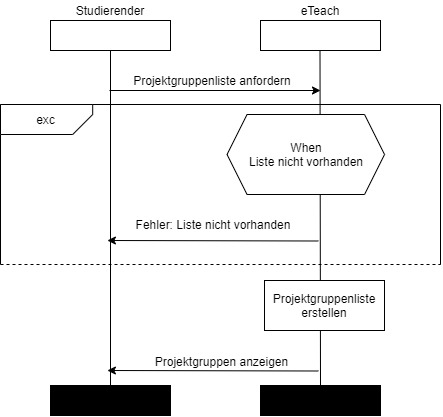
Erläuterung: Der Nutzer stellt eine Anfrage an eTeach, dass er eine Projektgruppe anlegen möchte. Dieser gibt im das Formular, welches ausgefüllt zurück geht. Hier findet die Validierung statt. Schlägt die Validierung fehl aufgrund eines Fehlers, z.B. die Projektgruppe existiert bereits, wird das dem Nutzer mitgeteilt und abgebrochen. War die Validierung erfolgreich wird die Projektgruppe in der Datenbank gespeichert und daraufhin dem Nutzer mitgeteilt, dass diese erfolgreich angelegt wurde.

### **Szenario Studenten hinzufügen**



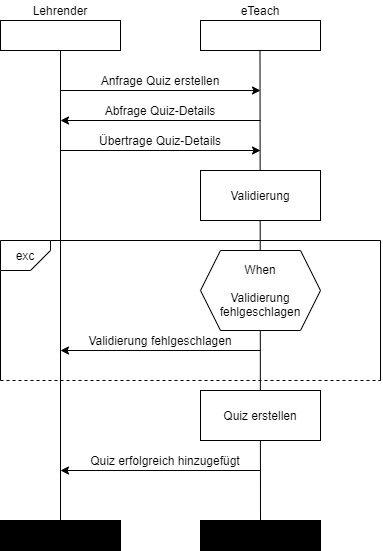
Erläuterung: Der Lehrende stellt eine Anfrage an das eTeach, dass er nach einem Studierenden suchen möchte. Nun wird die Datenbank nach Studierenden abgefragt und alle Studierenden dem eTeach übermittelt. Nun bestehen 2 Alternativen. Entweder es wurde anhand des Nachnamens oder nach der Matrikelnummer gesucht. In beiden Fällen wird der Studierende manuell der Projektgruppe hinzugefügt. Es existiert jedoch eine Möglichkeit, in der der Studierende sich bereits in der Gruppe befindet. In diesem Fall wirft das System einen Fehler an den Lehrenden, dass der Studierende bereits eingeschrieben ist oder nicht berechtigt und es wird nichts weiter vorgenommen. Kommt es jedoch nicht zu dieser Ausnahme wird der eingetragene Studierende in der Teilnehmerliste auf der Datenbank gespeichert und dem Lehrenden wird mitgeteilt, dass der Studierende hinzugefügt wurde.

**Szenario Projektgruppe suchen**

****

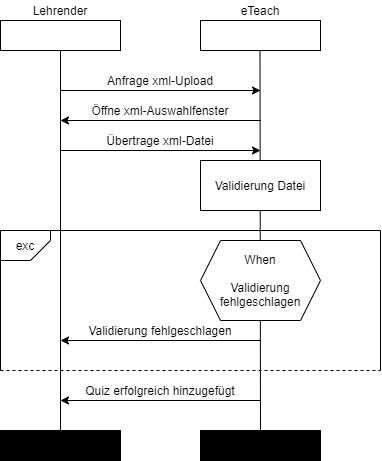
Erläuterung: Der Studierende stellt eine Anfrage an das eTeach, dass er nach Projektgruppen suchen möchte. Daraufhin fragt das eTeach in der Datenbank nach vorhanden Daten ab. Anschließend werden alle Projektgruppen an das eTeach übergeben. Hier werden alle Projektgruppen in einer Liste angezeigt und dem Studierenden übermittelt.

**Szenario Quiz erstellen**



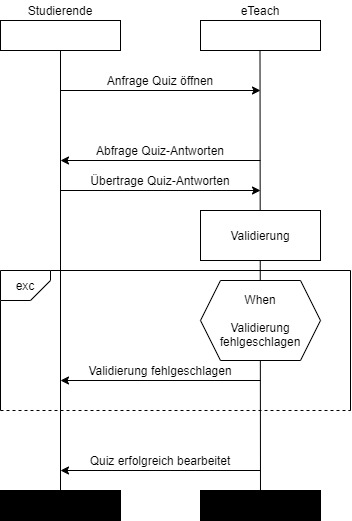
Erläuterung: Der Lehrende stellt eine Anfrage an das eTeach um ein Quiz zu erstellen. Nun ruft dieser von der Datenbank die Quizinformationen ab, die wiederum zurückübergeben werden. Der Lehrende wird jetzt dazu aufgefordert die nötigen Details anzugeben und diese zurückzugeben. Es erfolgt eine Validierung. Schlägt sie fehl, wird dem Lehrenden das mitgeteilt und es wird kein Quiz erstellt. Ist die Validierung erfolgreich wird das Quiz erstellt, in der Datenbank gespeichert und eine Mitteilung gesendet, dass das Quiz erfolgreich hinzugefügt wurde.

**Szenario Quiz mit xml-Datei erstellen**

****

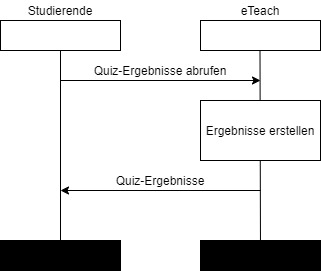
Erläuterung: In diesem Fall wird eine Anfrage für den Upload der xml-Datei an das eTeach gesendet. Dieser Öffnet dem Lehrenden das Auswahlfenster. Nun hat der Nutzer die Möglichkeit die Datei lokal auszuwählen und diese dem eTeach zu übertragen. Daraufhin wird die Datei validiert. Die darauffolgenden Schritte sind äquivalent zu dem vorherigen Szenario (Quiz manuell erstellen).

**Szenario Quiz bearbeiten**

****

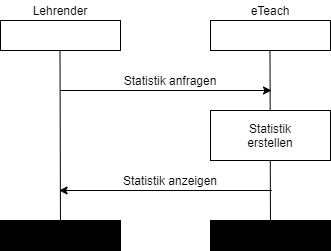
Erläuterung: Der Studierende stellt eine Anfrage das Quiz zu öffnen an das eTeach. Dieser ruft die Quizinformationen von der Datenbank ab, die ihm daraufhin übermittelt werden. Der Studierende wird nun dazu aufgefordert das Quiz zu bearbeiten. Anschließend überträgt er die Antworten an das eTeach weiter. Es erfolgt eine Validierung. Schlägt diese aufgrund unvorhergesehener Fehler fehl wird das dem Studierenden mitgeteilt. War die Validierung erfolgreich werden die Antworten in der Datenbank gespeichert und eine Mitteilung an den Studierenden wird übermittelt, dass die Abgabe erfolgreich war.

**Szenario Quiz Ergebnisse**

****

Erläuterung: Nach erfolgreicher Beendigung des Quiz wird automatisch eine Anfrage nach den Ergebnissen gestellt. Das eTeach stellt die Ergebnisse nun für den Studierenden dar, der die Ergebnisse übermittelt bekommt.

**Szenario Statistik**

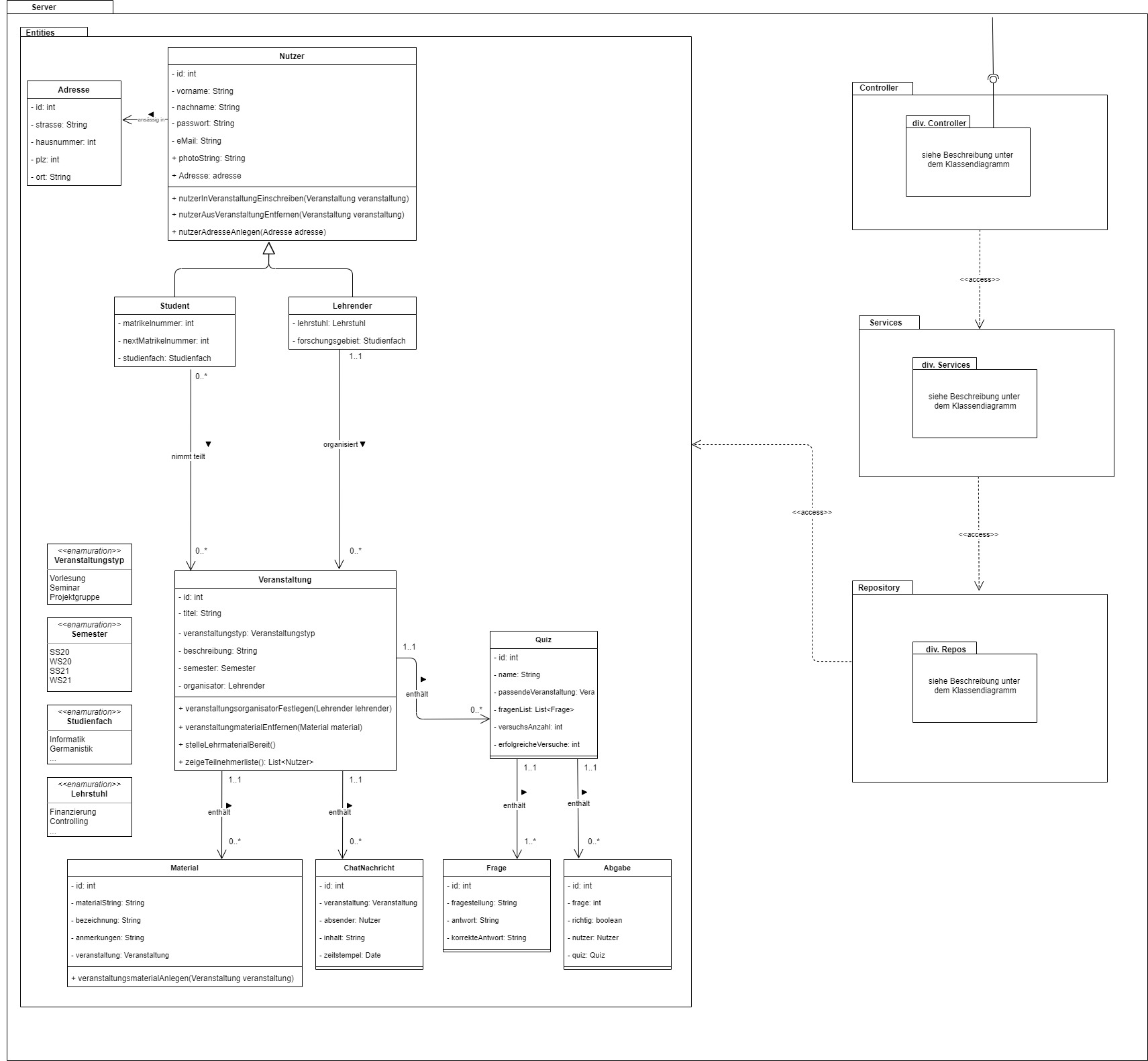
****

Erläuterung: Nach dem gleichen Prinzip des vorherigen Szenarios stellt der Lehrende eine Anfrage um die Statistik für ein Quiz zu erhalten. Auch hier wird durch das eTeach die Statistik erstellt und dem Lehrenden angezeigt.

## Strukturdiagramm (Klassendiagramm)

Verantwortlicher: Antonella Budiselic

### **Server**



Beschreibung: Wir haben 4 Haupt-Packages im Server. Das erste Package bildet die Entities ab. Der Nutzer kann entweder ein Student sein oder ein Lehrender. Je nachdem nimmt der Nutzer an einer Lehrveranstaltung teil oder organisiert sie. Eine Lehrveranstaltung kann Material, Chatnachrichten und Quizes enthalten. Die Quizes bestehen aus mindestens einer Frage und es besteht die Möglichkeit das Quiz abzugeben. Des weiteren besitzt die Klasse Lehrveranstaltungen 4 Enums. Zum einen den Lehrveranstaltungstyp Vorlesung, Seminar und Projektgruppe. Das zweite Enum stellt das Semester dar. Das dritte Enum stellt das Studienfach da, worunter sich beliebig viele befinden können. Das letzte Enum besteht aus dem Lehrstuhl, in dem sich ebenfalls beliebig viele befinden können. Die anderen Packages stellen die Controller, die Services und die Repositories dar. Die jeweiligen Controller, Services und Repositories sind der Anschaulichkeitshalber in einem extra Model dargestellt. Hierbei sei nur angemerkt, dass die Controller auf die Services zugreifen, diese auf die Repositories und diese wiederum auf die Entities. Zusätzlich stellen die Controller im Controller Package eine Verbindung zu den Endpoints auf Clientseite her.

### **Server Controller**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Anmerkung: hier sind die Controller im Server von Zyklus 1-2 enthalten.

### **Server Services**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Anmerkung: hier sind die Services im Server von Zyklus 1-2 enthalten.

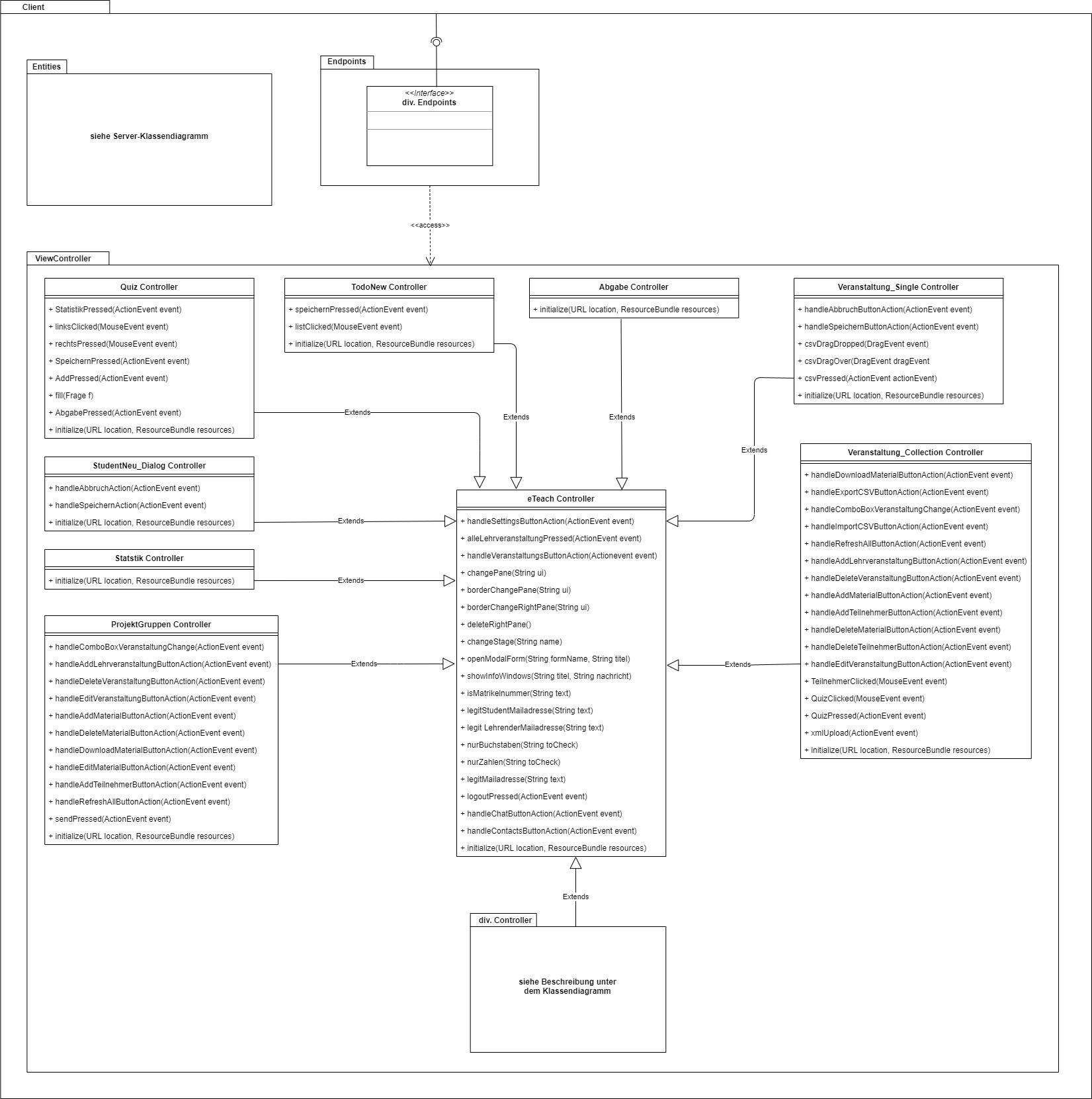
### **Server Repository**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Anmerkung: hier sind die Repositories im Server von Zyklus 1-2 enthallten.

### **Client**



Beschreibung: Äquivalent zum Klassendiagramm auf Serverseite befindet sich auf Clientseite ebenfalls das Package Entities. Um Redundanzen zu vermeiden wurde das Model auf einen Verweis auf das Server-Klassendiagramm abstrahiert. Wie auch schon vorher erwähnt ist auch hier die Verbindung zu den Endpoints aufgezeigt. Zuletzt haben wir hier noch die View Controller. Vom Hauptcontroller, auch eTeach Controller genannt, erben alle anderen Controller. Der Übersicht halber wurde im Model darauf verzichtet alle einzelnen Controller anzuzeigen. Deshalb hier nochmal eine kleine Liste über die vorhandenen Controller

(aus Zyklus 1):

Login Controller, RegisterLehrender Controller, RegisterStudent Controller, Lehrender\_Single Controller, LehrveranstaltungListe Controller, Material List Controller, Material Single Controller, Nutzer List Controller, Student\_Single Controller, Student List Controller, Student List Large Controller, Veranstaltung List Controller, Veranstaltung Teilnehmer List Controller

### **Client Endpoints**

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Anmerkung: hier sind die Endpoints im Client von Zyklus 1-2 enthalten.

## Funktionalitätsplanung





## Unittests

Unittest 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** | Nachricht senden | ChatRaumControllerTest | Tests passed |
| **2** | Nachricht löschen | ChatRaumControllerTest | Tests passed |
| **3** | Nachrichten abfragen | ChatRaumControllerTest | Tests passed |

## Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 21.06.2021 | | |
| **Tester** | Antonella Budiselic | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedingung(en)** | Der Lehrende „Herbert Groenemeyer“ ist am System mit Passwort „a“ und der E-Mail “hg@doz-uni.de“ registriert und eingeloggt. Außerdem ist die Veranstaltung „Testen“ mit dem Veranstaltungstyp „Projektgruppe“ erstellt. | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer klickt auf den Button „Meine ProjektGruppen“. | Das System zeigt das Layout von Projektgruppen allgemein an. | **√** |
| 2 | Der Benutzer wählt aus der Auswahlbox die Veranstaltung „Testen“ aus. | Das System zeigt alle vorhandenen Informationen einschließlich Materialien, dem Chat und den ToDo´s an. | **√** |
| 3 | Der Benutzer klickt auf „Teilnehmer hinzufuegen“. | Das System zeigt alle Hochschulangehörigen an. | **√** |
| 4 | Der Benutzer wählt die gewünschte Person „Max Mustermann“ aus und klickt auf „Auswahl hinzufuegen“. | Es öffnet sich ein neues Fenster mit der Meldung „Der Veranstaltung wurde 1 Teilnehmer hinzugefuegt“. | **√** |
| (Ausnahmeszenario)  4a | Der Benutzer wählt die gewünschte Person „Max Mustermann“ aus, die bereits eingeschrieben ist und klickt auf „Auswahl hinzufuegen“. | Es öffnet sich ein Fenster „Studierender bereits eingeschrieben“. | **√** |
| (Ausnahmeszenario)  4b | Der Benutzer wählt die gewünschte Person „Max Mustermann“ aus, die keine Berechtigung hat und klickt auf „Auswahl hinzufuegen“. | Es öffnet sich ein Fenster „Nutzer nicht berechtigt“. | **√** |
| **Nachbedingung(en)** | Der Studierende „Max Mustermann“ wurde durch einen Lehrenden der Veranstaltung „Testen“ hinzugefügt. | | **√** |
| **Testurteil** | Test bestanden. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datum** | | 21.06.2021 | | |
| **Tester** | | Antonella Budiselic | | |
| **SW-Version** | | V 0.1.2 | | |
| **Vorbedingung(en)** | | Der Lehrende „Herbert Groenemeyer“ ist am System mit Passwort „a“ und der E-Mail “hg@doz-uni.de“ registriert und eingeloggt. Außerdem ist die Veranstaltung „Programmierung“ mit dem Veranstaltungstyp „Vorlesung“ erstellt. | | |
| **Schritt** | | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | | Der Benutzer klickt auf den Button „Meine Veranstaltungen“. | Das System zeigt das Layout von Veranstaltungen allgemein an. | **√** |
| 2 | | Der Benutzer wählt aus der Auswahlbox die Veranstaltung „Programmierung“ aus. | Das System zeigt alle vorhandenen Informationen einschließlich Materialien, Teilnehmer und Quiz an. | **√** |
| 3 | | Der Benutzer klickt auf „Neues Quiz“. | Es erscheint ein Fenster, in dem man Testname, Frage und Antwort eingeben kann. | **√** |
| 4 | | Der Benutzer füllt alle Felder aus und wählt die richtige Antwort aus und klickt auf das „+“. | Es erscheint ein Fenster mit der Meldung „Quiz wurde erstellt“ | **√** |
| **Nachbedingung(en)** | | Der Lehrende hat ein Quiz angelegt und es im System gespeichert. | | **√** |
| **Testurteil** | Test bestanden. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datum** | | 21.06.2021 | | |
| **Tester** | | Antonella Budiselic | | |
| **SW-Version** | | V 0.1.2 | | |
| **Vorbedingung(en)** | | Der Studierende „Max Mustermann“ ist am System mit Passwort „a“ und der E-Mail “mm@stud-uni.de“ registriert und eingeloggt. Außerdem ist die Veranstaltung „Programmierung“ mit dem Veranstaltungstyp „Vorlesung“ erstellt und in „Meine Veranstaltungen“ gespeichert. | | |
| **Schritt** | | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | | Der Benutzer klickt auf den Button „Meine Veranstaltungen“. | Das System zeigt das Layout von Veranstaltungen allgemein an. | **√** |
| 2 | | Der Benutzer wählt aus der Auswahlbox die Veranstaltung „Programmierung“ aus. | Das System zeigt alle vorhandenen Informationen einschließlich Materialien, Teilnehmer und Quiz an. | **√** |
| 3 | | Der Benutzer klickt in der Quiz-Liste das Quiz „Test“ an. | Das System öffnet das gespeicherte Quiz mit allen Fragen und Antwortmöglichkeiten. | **√** |
| 4 | | Der Benutzer wählt die Antwort „ab“ aus und klickt auf „Antwort speichern“. | Es öffnet sich ein Fenster mit der Meldung „Antwort gespeichert“. | **√** |
| 5 | | Der Benutzer klickt auf den Pfeil nach rechts. | Das System zeigt die nächste Frage an. | **√** |
| 6 | | Der Benutzer wählt die Antwort „bb“ aus und klickt auf „Antwort speichern“. | Es öffnet sich ein Fenster mit der Meldung „Antwort gespeichert“. | **√** |
| 7 | | Der Benutzer klickt auf „Abgabe“. | Das System öffnet ein neues Fenster mit den Ergebnissen. | **√** |
| **Nachbedingung(en)** | | Eine neue Lösung wurde eingereicht und die Statistik aus Lehrender Sicht aktualisiert sich. | | **√** |
| **Testurteil** | Test bestanden. | | | |

# Zyklus III

## Definition der Verantwortlichen für die Artefakte der Spezifikation



## User-Stories

|  |  |
| --- | --- |
| (1) Kommunikation |  |
| User-StoryID | 23 |
| User Story-Beschreibung | Als Nutzer(in) möchte ich private Nachrichten an Mitglieder einer LV (Vorlesung, Seminar, Projektgruppe) verschicken können, um mit anderen LV Mitgliedern kommunizieren zu können. |
| Geschätzter Realisierungsaufwand | 5 Tage |
| Priorität | hoch |
| Autor(en) | Metwally, Omar |
| Abhängigkeiten zu anderen User Stories | 8,11,14 |
|  |  |
| User-StoryID | 24 |
| User Story-Beschreibung | Als Studierende(r) möchte ich Freundschaftsanfragen versenden, annehmen und diese auf meinem Startbildschirm sehen können, damit diese in meiner Freundesliste schnell zugänglich sind. |
| Geschätzter Realisierungsaufwand | 4 Tage |
| Priorität | hoch |
| Autor(en) | Metwally, Omar |
| Abhängigkeiten zu anderen User Stories | 14 |

|  |  |
| --- | --- |
| (2) Kalenderfunktionen |  |
| User-StoryID | 25 |
| User Story-Beschreibung | Als Nutzer(in) möchte ich über einen persönlichen Kalender verfügen, in dem ich Termine eintragen/auslesen kann, um alle Termine auf einem Blick zu haben. |
| Geschätzter Realisierungsaufwand | 4 Tage |
| Priorität | hoch |
| Autor(en) | Budiselic, Antonella |
| Abhängigkeiten zu anderen User Stories | - |
|  |  |
| User-StoryID | 26 |
| User Story-Beschreibung | Als Lehrende(r) möchte ich Termine in den Kalender der LV Teilnehmer eintragen können, um die Termine leichter zu übermitteln. |
| Geschätzter Realisierungsaufwand | 3 Tage |
| Priorität | mittel |
| Autor(en) | Budiselic, Antonella |
| Abhängigkeiten zu anderen User Stories | 25 |
|  |  |
| User-StoryID | 27 |
| User Story-Beschreibung | Als Lehrende(r) möchte ich Erinnerungen an die LV-Teilnehmer versenden können, die über ein Pop-up Fenster oder E-Mail übermittelt wird, um die Studierenden an ihre Termine zu erinnern. |
| Geschätzter Realisierungsaufwand | 3 Tage |
| Priorität | mittel |
| Autor(en) | Budiselic, Antonella |
| Abhängigkeiten zu anderen User Stories | 25,26 |

|  |  |
| --- | --- |
| (3) Zwei-Faktor-Authentisierung |  |
| User-StoryID | 28 |
| User Story-Beschreibung | Als Nutzer(in) möchte ich während des Einloggens eine Zwei-Faktor-Authentisierung haben, die einen Sicherheitscode an meine E-Mail versendet, um sicherer vor unerlaubtem Zugriff zu sein. |
| Geschätzter Realisierungsaufwand | 3 Tage |
| Priorität | niedrig |
| Autor(en) | Krolik, Sammy-Steve |
| Abhängigkeiten zu anderen User Stories | 1,2 |

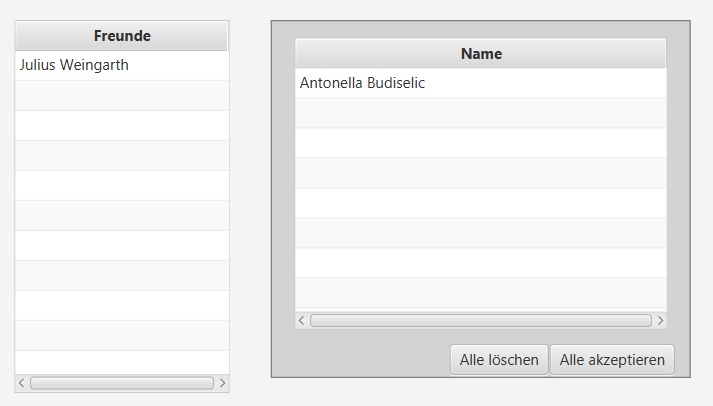
## Papierprototypen

### **Papierprototyp Privatnachricht versenden**



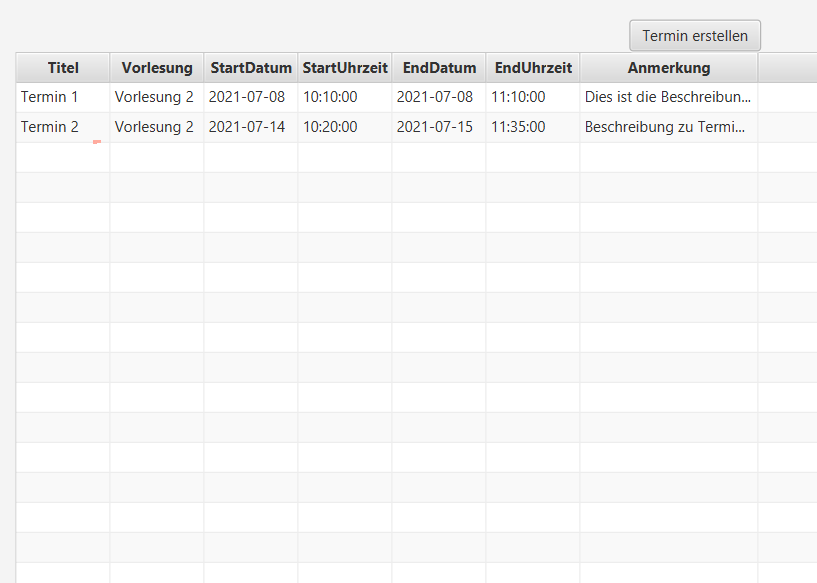
Erläuterung: Die Privatnachrichten-Funktion wird über das Profil aufgerufen, das man anklickt. Dort befindet sich ein Button „Privatchat“. Klickt man auf diesen erhält man den Chat. Auf der eigenen Profilansicht (im Zyklus 1) erhält man alle vergangenen Chatnachrichten.

### **Papierprototyp Freundschaftsanfragen**



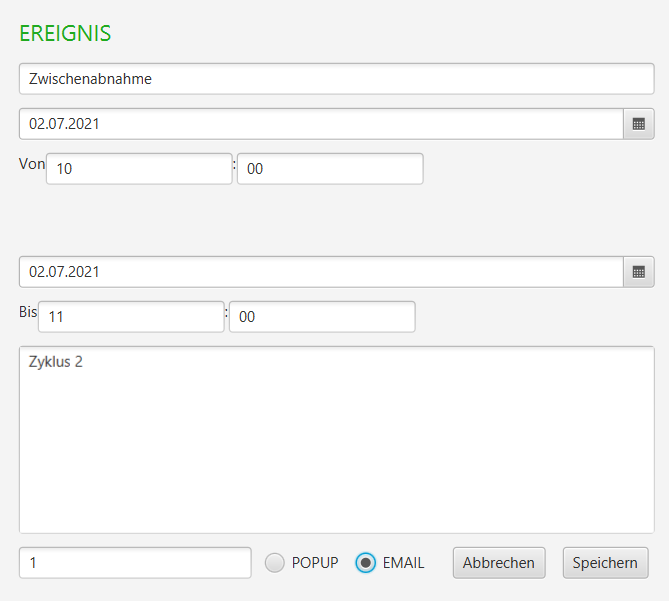
Erläuterung: In diesem Prototypen wird links die Freundesliste in einer Tabelle angezeigt. Rechts befinden sich die ausstehenden Anfragen, die separat akzeptiert werden können oder durch einen Klick auf „Alle akzeptieren“ alle Anfragen anzunehmen. Dementsprechend lassen sich auch alle Anfragen gleichzeitig löschen bzw. ablehnen. Freundschaftsanfragen werden versendet, indem man auf das jeweilige Profil klickt, „Privatchat“ auswählt und auf „Freundschaftsanfrage“ klickt.

### **Papierprototyp Kalender**



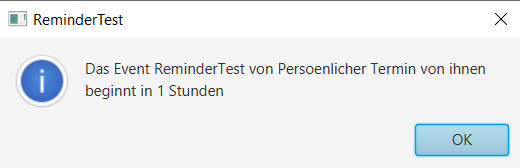
Erläuterung: Der persönliche Kalender beinhaltet alle Termine aus den teilnehmenden Lehrveranstaltungen und verfügt über die Funktion einen neuen Termin („Termin erstellen“) nur für sich selbst einzutragen. Jeder Termin besitzt einen Titel, falls vorhanden, die zugehörige Vorlesung, ein Start- und Enddatum, eine Start- und Endzeit und Anmerkungen.

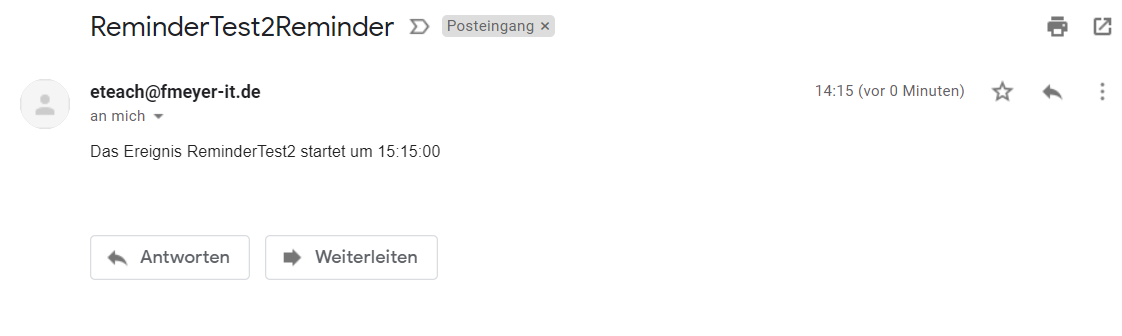
### **Papierprototyp Ereignis erstellen**



Erläuterung: Dies ist die Ansicht um Termine einzutragen. Man kann dem Termin einen Titel geben, das Datum auswählen, die Zeit angeben und Anmerkungen hinzufügen. Hier besteht auch die Möglichkeit anzugeben wie viele Stunden vor Ablauf des Termins ein Reminder versendet werden soll und auf welche Art (Pop-up oder E-Mail).

### **Papierprototyp Reminder versenden**





Erläuterung: Hier angezeigt werden die Reminder. Das obere Bild zeigt den Reminder über ein Pop-up-Fenster an. Das untere Bild zeigt die E-Mail, die man vom eTeach erhält.

### **Papierprototyp Zwei-Faktor Authentifizierung**



Erläuterung: Dieses Fenster wird nach der Login-Funktion angezeigt. Der Nutzer erhält einen Code per E-Mail, den er eintragen muss um sicher eingeloggt zu werden und unerlaubten Zugriff zu vermeiden. Nach Bestätigung des Codes gelangt man auf das Dashboard.

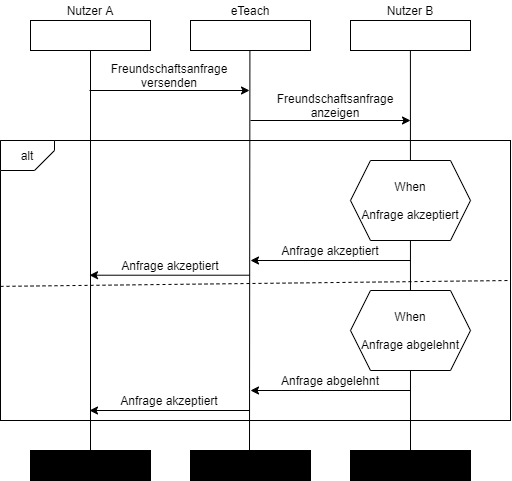
## Szenarien (MSCs)

**Szenario Privatnachricht versenden**

****

Erläuterung: Der Nutzer stellt eine Anfrage an das eTeach, dass er eine Nachricht versenden möchte. Hier kann es zu 2 alternativen Ausnahmen kommen. Entweder der ausgewählte Nutzer ist nicht vorhanden, dann wird dem Nutzer dies mitgeteilt. Die andere Alternative wäre, dass die Nachricht leer ist, dann wird das auch dem Nutzer mitgeteilt. Tritt keine dieser Szenarien auf wird die Nachricht erstellt und an den Adressaten übermittelt. Dieser zeigt nun dem Adressaten die Nachricht an.

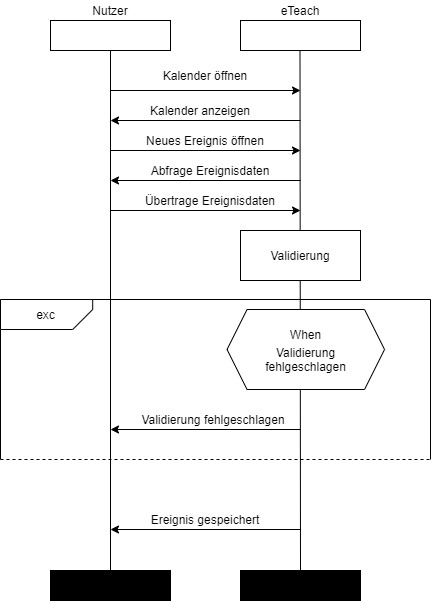
**Szenario Freundschaftsanfrage**

****

Erläuterung: **Vorbedingung:** Nutzer A ist eingeloggt und befindet sich auf der Profilansicht von Nutzer B.

Der Nutzer gibt dem eTeach an, dass er eine Freundschaftsanfrage versenden möchte. Daraufhin wird die Anfrage an den Nutzer gesendet und diesem angezeigt. Nun besteht für den Adressaten die Möglichkeit die Anfrage anzunehmen, dann wird diese über das eTeach an Nutzer1 übermittelt. Dieser gibt der Datenbank die Daten, die in der Freundschaftsliste gespeichert wird. Die andere Alternative besteht darin die Anfrage abzulehnen. Nach Übermittlung an das eTeach wird die Anfrage gelöscht.

### **Szenario Ereignis in Kalender**



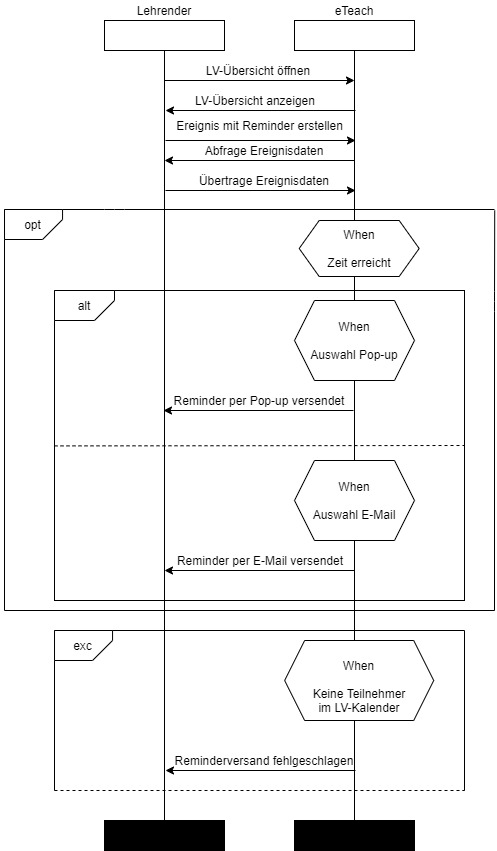
Erläuterung: Der Nutzer stellt eine Anfrage an das eTeach den Kalender zu öffnen. Der Nutzer möchte im aufgerufenen Kalender ein neues Ereignis erstellen und stellt dazu eine Anfrage an das eTeach. Nach Abfrage der Ereignisdaten werden diese zurück übertragen. Es erfolgt eine Validierung, schlägt diese fehl wird ein neues Ereignis hinzugefügt. Ist es erfolgreich wird es in der Datenbank abgespeichert und daraufhin dem Nutzer mitgeteilt, dass das Ereignis erfolgreich erstellt wurde.

### **Szenario LV-Ereignis zu Kalender**



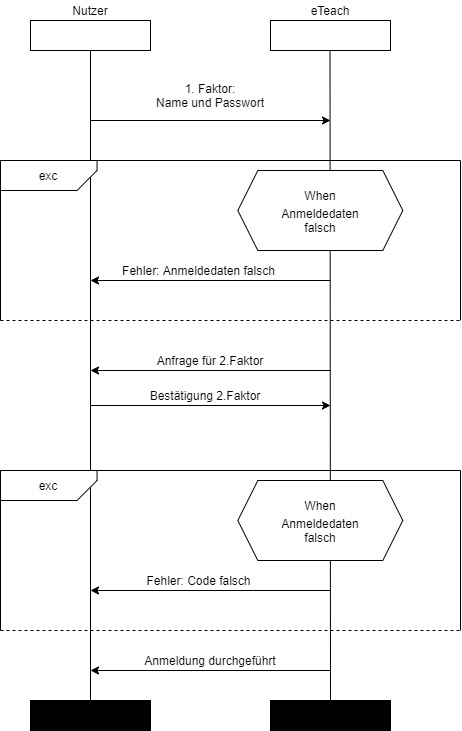
Erläuterung: Äquivalent zu dem Szenario zuvor sind die Abläufe identisch, allerding handelt es sich hier nicht um den Nutzer, in dessen persönlichen Kalender das Ereignis eingetragen wird, sondern ausschließlich um den Lehrenden. Da er der Einzige ist, der durch die ToDo Ereigniss-Erstellung in der Lehrveranstaltung das Speichern im persönlichen Kalender der Studierenden ermöglichen kann.

### **Szenario Reminder**



Erläuterung: Es erfolgen zunächst die gleichen Schritte wie in den vorherigen Szenarien. Ist nun die Zeit für den Reminder erreicht kann der Lehrende den Reminder entweder per E-Mail oder durch ein Pop-up-Fenster versenden. Sollten keine Teilnehmer in der Lehrveranstaltung eingeschrieben sein wird der Reminderversand fehlgeschlagen.

**Szenario Zwei-Faktor-Authentisierung**

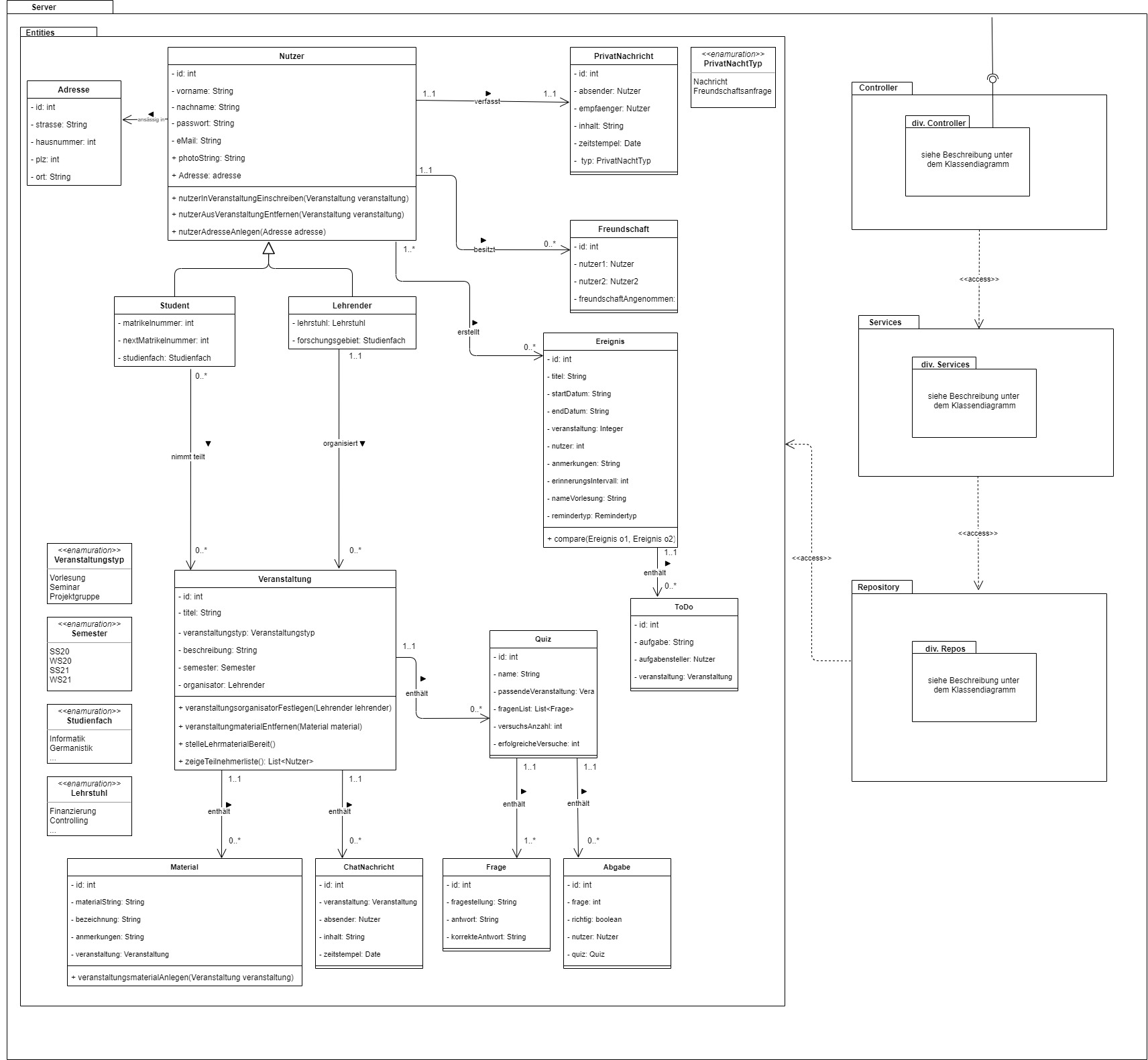
****

Erläuterung: Zuerst stellt der Nutzer die Login Anfrage für den 1. Faktor an das eTeach. Dieser fragt in der Datenbank nach den gewünschten Daten. Daraufhin werden die Daten übermittelt. Besteht nun der Fall, dass die Anmeldedaten falsch sind, wird der Anmeldevorgang abgebrochen und dem Nutzer wird mitgeteilt, dass die Daten falsch waren. Waren die Daten korrekt wird dem Nutzer die Anfrage für den 2. Faktor gestellt. Der Nutzer gibt nun den erhaltenen Code in der Anfrage ein. Erneut wird in der Datenbank nach den richtigen Daten gefragt, die zurückübergeben werden. Ist der Code falsch, wird auch dies dem Nutzer mitgeteilt und der Login-Vorgang ist fehlgeschlagen. War der Code korrekt erfolgt die Authentifizierung für die Anmeldung und der Nutzer erhält die Nachricht erfolgreich angemeldet worden zu sein.

## Strukturdiagramm (Klassendiagramm)

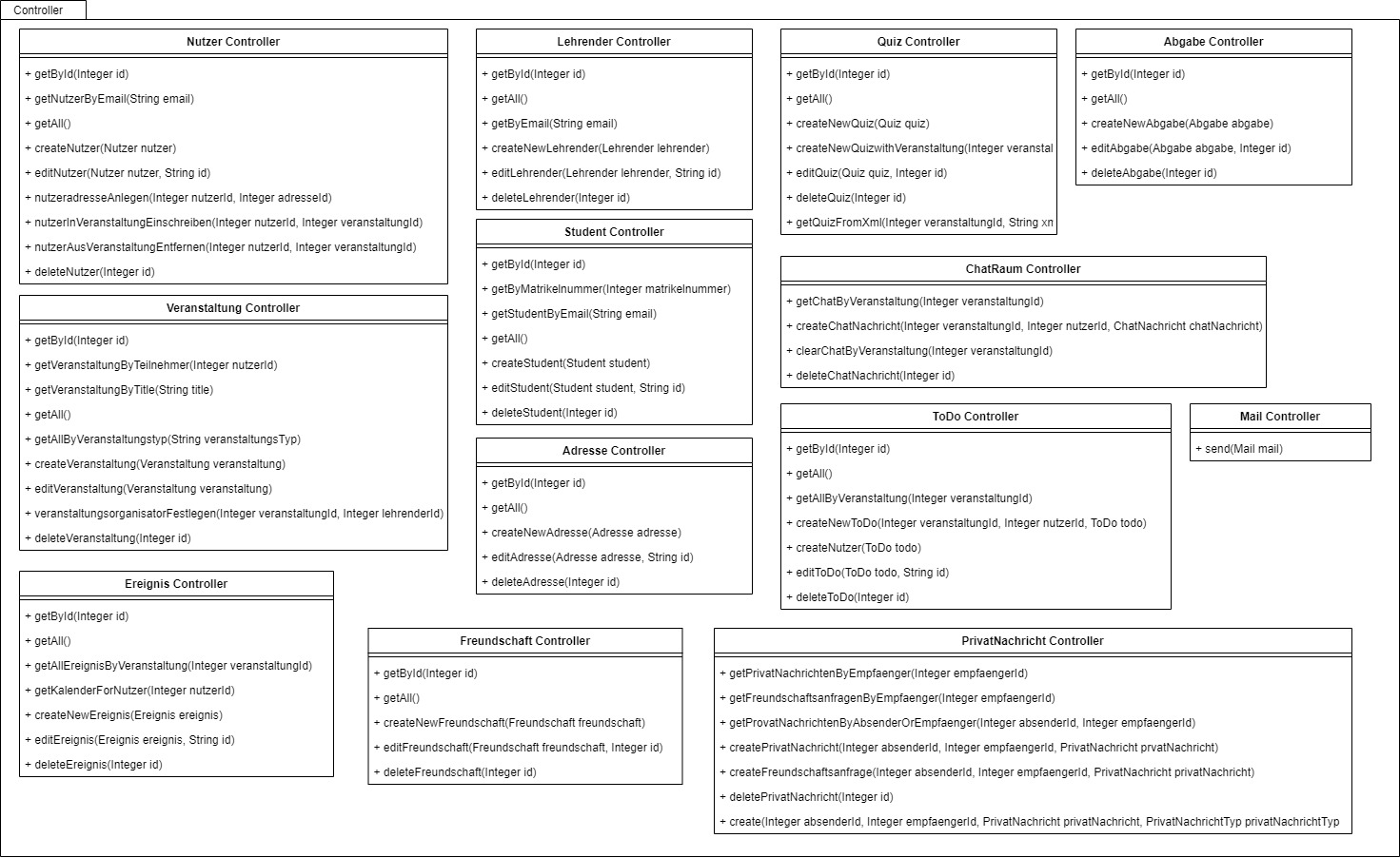
Verantwortlicher: Antonella Budiselic

### **Server**



Beschreibung zum Klassendiagramm: Wir haben 4 Haupt-Packages im Server. Das erste Package bildet die Entities ab. Wir sehen hier, dass der Nutzer dazu in der Lage ist eine Privatnachricht zu verfassen bzw. zu senden. Außerdem ist er dazu in der Lage Freundschaftsanfragen zu versenden und anzunehmen. Der Nutzer kann entweder ein Student sein oder ein Lehrender. Je nachdem nimmt der Nutzer an einer Lehrveranstaltung teil oder organisiert sie. Eine Lehrveranstaltung kann Material, Chatnachrichten und Quizes enthalten. Die Quizes bestehen aus mindestens einer Frage und es besteht die Möglichkeit das Quiz abzugeben. Der Nutzer besitzt außerdem einen persönlichen Kalender, der aus Ereignissen besteht. Es ist außerdem möglich ein ToDo aus der Lehrveranstaltung in den persönlichen Kalender des Nutzers einzutragen. Des weiteren besitzt die Klasse Lehrveranstaltungen 4 Enums. Zum einen den Lehrveranstaltungstyp Vorlesung, Seminar und Projektgruppe. Das zweite Enum stellt das Semester dar. Das dritte Enum stellt das Studienfach da, worunter sich beliebig viele befinden können. Das letzte besteht aus dem Lehrstuhl, in dem sich ebenfalls beliebig viele befinden können. Die anderen Packages stellen die Controller, die Services und die Repositories dar. Die jeweiligen Controller, Services und Repositories sind der Anschaulichkeitshalber in einem extra Model dargestellt. Hierbei sei nur angemerkt, dass die Controller auf die Services zugreifen, diese auf die Repositories und diese wiederum auf die Entities. Zusätzlich stellen die Controller im Controller Package eine Verbindung zu den Endpoints auf Clientseite her.

### **Server Controller**



Anmerkung: Hier sind alle Controller der Serverseite von Zyklus 1-3 enthalten.

### **Server Services**

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Anmerkung: hier sind alle Services vom Server von Zyklus 1-3 enthalten.

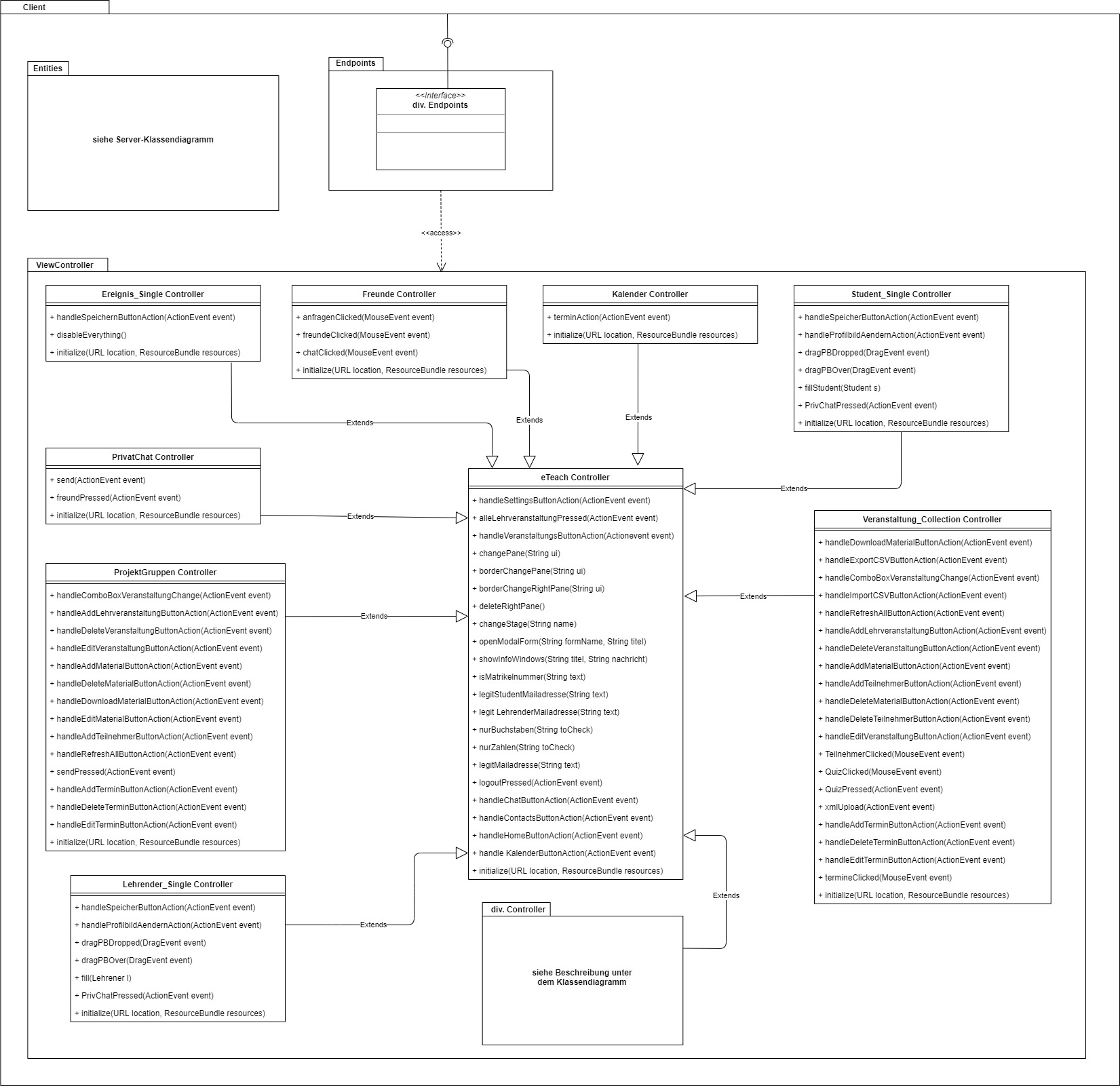
### **Server Repository**

Ein Bild, das Text, Tisch enthält.

Automatisch generierte Beschreibung

Anmerkung: hier sind alle Repositories vom Server aus Zyklus 1-3 enthalten.

### **Client**



Beschreibung: Äquivalent zum Klassendiagramm auf Serverseite befindet sich auf Clientseite ebenfalls das Package Entities. Um Redundanzen zu vermeiden wurde das Model auf einen Verweis auf das Server-Klassendiagramm abstrahiert. Wie auch schon vorher erwähnt ist auch hier die Verbindung zu den Endpoints aufgezeigt. Zuletzt haben wir hier noch die View Controller. Vom Hauptcontroller, auch eTeach Controller genannt, erben alle anderen Controller. Der Übersicht halber wurde im Model darauf verzichtet alle einzelnen Controller anzuzeigen. Deshalb hier nochmal eine kleine Liste über die vorhandenen Controller

(aus Zyklus 1): Login Controller, RegisterLehrender Controller, RegisterStudent Controller, LehrveranstaltungsListe Controller, Material List Controller, Material Single Controller, Nutzer List Controller, Student List Controller, Student List Large Controller, Veranstaltung List Controller, Veranstaltung Teilnehmer List Controller

(aus Zyklus 2): Statistik Controller, Quiz Controller, StudentDialog Controller, ToDo Controller, Abgabe Controller

### **Client Endpoints**



Anmerkung: hier sind alle Endpoint vom Client aus Zyklus 1-3 enthalten.

## Funktionalitätsplanung





## Unittests

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Getestete Funktionalität** | **Quellcode**  **Referenz** | **Status** |
| **1** | Ereignis erstellen | EreignisControllerTest | Tests passed |
| **2** | Ereignis bearbeiten | EreignisControllerTest | Tests passed |
| **3** | Ereignis löschen | EreignisControllerTest | Tests passed |

## Systemtests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datum** | | 16.07.2021 | | |
| **Tester** | | Antonella Budiselic | | |
| **SW-Version** | | V 0.1.3 | | |
| **Vorbedingung(en)** | | Der Lehrende „Herbert Groenemeyer“ ist im System mit Passwort „a“ und der E-Mail “hg@doz-uni.de“ registriert und eingeloggt. Außerdem ist die Veranstaltung „Programmierung“ mit dem Veranstaltungstyp „Vorlesung“ erstellt. | | |
| **Schritt** | | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | | Der Benutzer klickt auf den Button „Meine Veranstaltungen“. | Das System zeigt das Layout von Veranstaltungen allgemein an. | **√** |
| 2 | | Der Benutzer wählt aus der Auswahlbox die Veranstaltung „Programmierung“ aus. | Das System zeigt alle vorhandenen Informationen einschließlich Materialien, Teilnehmer, Quiz und Termine an. | **√** |
| 3 | | Der Benutzer klickt auf den Teilnehmer „Manuel Neuer“ in der Teilnehmerliste | Es erscheint ein Fenster, mit dem Profil des Nutzers. | **√** |
| 4 | | Der Benutzer klickt auf den Button „PrivatChat“. | Es erscheint ein Fenster mit dem Privatchat. | **√** |
| 5 | | Der Benutzer tippt die Nachricht „hallo“ ein und klickt auf den senden Button. | Die Nachricht wird nun im Chat angezeigt und dem Studierenden übermittelt. | **√** |
| (Ausnahmeszenario)  3a | | Der Benutzer klickt auf den Teilnehmer „Manuel Neuer“ in der Teilnehmerliste, der währenddessen nicht mehr in der Veranstaltung ist. | Es erscheint ein Fenster mit der Fehlermeldung „der Nutzer existiert nicht“. | **√** |
| (Ausnahmeszenario)  5a | | Der Benutzer tippt keine Nachricht ein und klickt auf den senden Button. | Es wird keine Nachricht versendet. | **√** |
| **Nachbedingung(en)** | | Der Lehrende hat eine Privatnachricht an den Studierenden versendet. | | **√** |
| **Testurteil** | Test bestanden. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datum** | | 16.07.2021 | | |
| **Tester** | | Antonella Budiselic | | |
| **SW-Version** | | V 0.1.3 | | |
| **Vorbedingung(en)** | | Der Lehrende „Herbert Groenemeyer“ ist im System mit Passwort „a“ und der E-Mail “hg@doz-uni.de“ registriert und eingeloggt. Außerdem ist die Veranstaltung „Programmierung“ mit dem Veranstaltungstyp „Vorlesung“ erstellt. | | |
| **Schritt** | | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | | Der Benutzer klickt auf den Button „Meine Veranstaltungen“. | Das System zeigt das Layout von Veranstaltungen allgemein an. | **√** |
| 2 | | Der Benutzer wählt aus der Auswahlbox die Veranstaltung „Programmierung“ aus. | Das System zeigt alle vorhandenen Informationen einschließlich Materialien, Teilnehmer, Quiz und Termine an. | **√** |
| 3 | | Der Benutzer klickt unter der Rubrik Termine auf (+). | Es erscheint ein Fenster, in dem man Termindaten wie Name, Datum, Zeit und Anmerkungen eintragen kann. | **√** |
| 4 | | Der Benutzer füllt alle Felder aus und klickt auf „Speichern“. | Es erscheint ein Fenster mit der Meldung „Termin gespeichert“. | **√** |
| 5 | | Der Benutzer klickt auf den Button „Kalender“. | Der angelegte Termin wird im Kalender angezeigt. | **√** |
| **Nachbedingung(en)** | | Der Lehrende hat ein Termin angelegt, der sowohl in der Terminliste als auch dem persönlichen Kalender der teilnehmenden Studierenden eingetragen wurde. | | **√** |
| **Testurteil** | Test bestanden. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datum** | | 16.07.2021 | | |
| **Tester** | | Antonella Budiselic | | |
| **SW-Version** | | V 0.1.3 | | |
| **Vorbedingung(en)** | | Der Studierende „Manuel Neuer“ ist im System mit Passwort „a“ und der E-Mail “mn@stud-uni.de“ registriert. | | |
| **Schritt** | | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | | Der Benutzer gibt im Login Fenster seine Anmeldedaten ein und klickt auf „Login“. | Das System versendet eine E-Mail an die hinterlegte E-Mail-Adresse mit einem Code und öffnet die Ansicht für die Authentisierung. | **√** |
| 2 | | Der Benutzer öffnet die E-Mail, gibt den angegebenen Code ein und klickt auf „Bestätigen“. | Der Nutzer wird eingeloggt und ihm wird das Dashboard angezeigt. | **√** |
| (Ausnahmeszenario)  1a | | Der Benutzer gibt im Login Fenster seine falschen Anmeldedaten ein und klickt auf „Login“. | Es erscheint ein Fenster mit der Meldung „Mail-Adresse oder Passwort falsch.“. | **√** |
| (Ausnahmeszenario)  2a | | Der Benutzer öffnet die E-Mail, gibt den angegebenen Code falsch ein und klickt auf „Bestätigen“. | Es erscheint ein Fenster mit der Meldung „Code ist falsch“. |  |
| **Nachbedingung(en)** | | Der Studierende hat sich erfolgreich eingeloggt. | | **√** |
| **Testurteil** | Test bestanden. | | | |

# Nutzerhandbuch

## Technische Anforderungen

Voraussetzung: Java  
Empfehlungen: Java SE 13.0.7 und Javafx-sdk-16

## Installationsanleitung

Das gesamte Programm besteht aus den zwei Dateien „Server-0.0.1-SNAPSHOT.jar“ und „Client-1.0-SNAPSHOT.jar“. Diese können einfach mit einem Doppelklick ausgeführt werden, über die beigelegten Dateien „ClientStart.bat“ und „Start.bat“.

## Bedienungsanleitung

Bitte starten sie zuerst den Server. Dann starten sie den Client.