SEP Projektmappe

**Projektmappe des Projektes**

Student Education Platform

**Dokumentation des Projektes**

Gruppe M:

Serena Lavinia Böker

Nguyen-Le Lam

Georg Orfali

Natalie Schmidt

Yafet Zehaie

**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 4](#_Toc69504733)

[Zyklus I 5](#_Toc69504734)

[Definition der Verantwortlichen für die Artefakte der Spezifikation 5](#_Toc69504735)

[User-Stories 5](#_Toc69504736)

[Papierprototypen 6](#_Toc69504737)

[Szenarien (MSCs) 6](#_Toc69504738)

[Strukturdiagramm (Klassendiagramm) 7](#_Toc69504739)

[Funktionalitätsplanung 7](#_Toc69504740)

[Systemtests 7](#_Toc69504741)

[Zyklus II 10](#_Toc69504742)

[Definition der Verantwortlichen für die Artefakte der Spezifikation 10](#_Toc69504743)

[User-Stories 10](#_Toc69504744)

[Papierprototypen 10](#_Toc69504745)

[Szenarien (MSCs) 10](#_Toc69504746)

[Strukturdiagramm (Klassendiagramm) 10](#_Toc69504747)

[Funktionalitätsplanung 11](#_Toc69504748)

[Unittests 11](#_Toc69504749)

[Systemtests 11](#_Toc69504750)

[Zyklus III 14](#_Toc69504751)

[Definition der Verantwortlichen für die Artefakte der Spezifikation 14](#_Toc69504752)

[User-Stories 14](#_Toc69504753)

[Papierprototypen 14](#_Toc69504754)

[Szenarien (MSCs) 14](#_Toc69504755)

[Strukturdiagramm (Klassendiagramm) 15](#_Toc69504756)

[Funktionalitätsplanung 15](#_Toc69504757)

[Unittests 15](#_Toc69504758)

[Systemtests 15](#_Toc69504759)

[Nutzerhandbuch 16](#_Toc69504760)

[Technische Anforderungen 16](#_Toc69504761)

[Installationsanleitung 16](#_Toc69504762)

[Bedienungsanleitung 16](#_Toc69504763)

# 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.
* 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, dass das System keine urheberrechtlich geschützten Inhalte (Bilder, Musik, etc.) enthält.

# Zyklus I

## Definition der Verantwortlichen für die Artefakte der Spezifikation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Artefakt** | **Art des Artefakts** | **Verantwortlicher** | **Status** |
| **1.** | Registrierung der Nutzer |  |  |  |
| 1.1 | Nutzerregistrierung und Kontoerstellung | User Story | Natalie Schmidt | Tbd |
| 1.2 | Nutzerregistrierung und Kontoerstellung | bMSC | Natalie Schmidt | Tbd |
| 1.3 | Registrierungsseite | Papierprototyp | Natalie Schmidt | Tbd |
| 1.4 | Kontoeditierung | User Story | Natalie Schmidt | Tbd |
| 1.5 | Kontoeditierung | bMSC | Natalie Schmidt | Tbd |
| 1.6 | Profilbearbeitung | Papierprototyp | Natalie Schmidt | Tbd |
| 1.7 | Profileinstellungen | Papierprototyp | Natalie Schmidt | Tbd |
| 1.8 | Kontrollfluss | Aktivitätsdiagramm/Modell | Natalie Schmidt | Tbd |
| 1.9 | Datenspeicherung | bMSC | Georg Orfali | Tbd |
| 1.10 | Passwortveränderung | bMSC | Georg Orfali | Tbd |
| **2.** | Erstellung einer Lehrveranstaltung |  |  |  |
| 2.1 | Erstellung einer Lehrveranstaltung | User Story | Serena Lavinia Böker | Tbd |
| 2.2 | Erstellung einer Lehrveranstaltung | bMSC | Serena Lavinia Böker | Tbd |
| 2.3 | Fenster für die Erstellung einer Lehrveranstaltung | Papierprototyp | Serena Lavinia Böker | Tbd |
| **3.** | Übersichtsseite einer Lehrveranstaltung |  |  |  |
| 3.1 | Aufrufen der Übersichtsseite einer Lehrveranstaltung | User Story | Serena Lavinia Böker | Tbd |
| 3.2 | Aufrufen der Übersichtsseite einer Lehrveranstaltung | bMSC | Serena Lavinia Böker | Tbd |
| 3.3 | Übersichtsseite einer Lehrveranstaltung | Papierprototyp | Serena Lavinia Böker | Tbd |
| **4.** | Hochladen von Lehrmaterial |  |  |  |
| 4.1 | Hochladen von Lehrmaterial | User Story | Serena Lavinia Böker | Tbd |
| 4.2 | Hochladen von Lehrmaterial | bMSC | Serena Lavinia Böker | Tbd |
| 4.3 | Fenster zum Hochladen von Lehrmaterial | Papierprototyp | Serena Lavinia Böker | Tbd |
| **5.** | Liste aller Lehrveranstaltungen aufrufen |  |  |  |
| 5.1 | Liste aller Lehrveranstaltungen aufrufen | User Story |  | Tbd |
| 5.2 | Liste aller Lehrveranstaltungen aufrufen | bMSC |  | Tbd |
| 5.3 | Liste aller Lehrveranstaltungen | Papierprototyp |  | Tbd |
| **6.** | Beitritt zu Lehrveranstaltungen |  |  |  |
| 6.1 | Lehrveranstaltung beitreten | User Story | Yafet Zehaie | Tbd |
| 6.2 | Lehrveranstaltung beitreten | bMSC | Yafet Zehaie | Tbd |
| 6.3 | Fenster dazu | Papierprototyp | Yafet Zehaie | Tbd |
| **7.** | Zuordnung der Lehrenden |  |  |  |
| 7.1 | Automatische Zuordnung der Lehrenden | User Story | Yafet Zehaie | Tbd |
| 7.2 | Automatische Zuordnung der Lehrenden | bMSC | Yafet Zehaie | Tbd |
| **8.** | Hinzufügen von Teilnehmern zu einer Lehrveranstaltung |  |  |  |
| 8.1 | Suchen und Hinzufügen von Teilnehmern zu einer Lehrveranstaltung | User Story | Yafet Zehaie | Tbd |
| 8.2 | Suchen und Hinzufügen von Teilnehmern zu einer Lehrveranstaltung | bMSC | Yafet Zehaie | Tbd |
| 8.3 | Suchfenster in einer Lehrveranstaltung | Papierprototyp | Yafet Zehaie | Tbd |
| **9.** | Teilnehmerliste einer Veranstaltung |  |  |  |
| 9.1 | Einsicht der Teilnehmerliste einer Veranstaltung | User Story | Yafet Zehaie | Tbd |
| 9.2 | Teilnehmerliste einer Veranstaltung aufrufen | bMSC | Yafet Zehaie | Tbd |
| 9.3 | Teilnehmerliste einer Veranstaltung | Papierprototyp | Yafet Zehaie | Tbd |
| **10** | Nutzerlogin |  |  |  |
| 10.1 | Einloggen | User Story | Georg Orfali | Tbd |
| 10.2 | Einloggen | User Story | Georg Orfali | Tbd |
| 10.3 | Login-Fenster | Papierprototyp | Georg Orfali | Tbd |
| **10.** | Nutzerprofil |  |  |  |
| 10.1 | Eigenes Nutzerprofil ansehen | User Story | Nguyen-Le Lam | Tbd |
| 10.2 | Eigenes Nutzerprofil aufrufen | bMSC | Nguyen-Le Lam | Tbd |
| 10.3 | Nutzerprofil | Papierprototyp | Nguyen-Le Lam | Tbd |
| 10.4 | Hervorheben des Nutzerprofils in Teilnehmerliste | User Story |  | Tbd |
| 10.5 | Nutzerprofil aufrufen durch Anklicken des Eintrags in Teilnehmerliste | bMSC |  | Tbd |
| 10.6 | Sichtbarkeit der Daten für Studierende/Lehrende | User Story |  | Tbd |
| 10.7 | Sichtbarkeit der Daten für Studierende/Lehrende | bMSC |  | Tbd |
| 10.8 | Sichtbarkeit der Daten für Studierende/Lehrende | Papierprototyp |  | Tbd |
| 11. | Verwaltung von Lehrveranstaltungen |  |  |  |
| 11.1 | eigene Lehrveranstaltungen sehen | User Story | Nguyen-Le Lam | Tbd |
| 11.2 | Lehrveranstaltungen suchen | User Story | Nguyen-Le Lam | Tbd |
| 11.3 | Lehrveranstaltungen beitreten | User Story | Nguyen-Le Lam | Tbd |
| 11.4 | Lehrveranstaltungsliste öffnen | bMSC | Nguyen-Le Lam | Tbd |
| 11.5 | Lehrveranstaltungen beitreten | bMSC | Nguyen-Le Lam | Tbd |
| 11.6 | Lehrveranstaltungen suchen | bMSC | Nguyen-Le Lam | Tbd |
| 11.7 | Lehrveranstaltungsliste | Papierprototyp | Nguyen-Le Lam | Tbd |
| 11.8 | Lehrveranstaltungssuche | Papierprototyp | Nguyen-Le Lam | Tbd |
| 12 | Klassendiagramm Client | Klassendiagramm | Georg Orfali | Tbd |
| 13 | Klassendiagramm Server | Klassendiagramm | Georg Orfali | Tbd |
| 14 | Datenbankdiagramm | (Klassen)Diagramm | Georg Orfali | Tbd |
| 15 | Homescreen | Papierprototyp | Georg Orfali | Tbd |

## User-Stories

Template:

|  |  |
| --- | --- |
| **User Story-ID** | <Eindeutiger Identifizierer> |
| **User Story-Beschreibung** | <Text der User Story mittels Satzschablone:  Als <Rolle> möchte ich <Ziel> [, um/sodass <Nutzen>]  (s. Foliensatz „Anforderungen“)> |
| **Geschätzter Realisierungsaufwand** | <Einschätzung der Zeit, die benötigt wird, um die Userstory zu implementieren> |
| **Priorität** | <Wichtigkeit der User Story hinsichtlich der Aufgabenstellung z.B. hoch, mittel niedrig> |
| **Autor(en)** | <Hier bitte nur einen Zuständigen eintragen z.B. Max Mustermann> |
| **Abhängigkeiten zu anderen User Stories** | <Auflistung verwandter User Stories> |

**Schlechtes Beispiel:**

|  |  |
| --- | --- |
| **User Story-ID** |  |
| **User Story-Beschreibung** | Ich möchte ich rechtzeitig informiert werden, wenn ein Patient einen Termin nicht wahrnimmt. |
| **Geschätzter Realisierungsaufwand** | 1337 |
| **Priorität** | - |
| **Autor(en)** | Emmett Brown, Rick Sanchez, Amelia Pond |
| **Abhängigkeiten zu anderen User Stories** |  |

**Gutes Beispiel:**

|  |  |
| --- | --- |
| **User Story-ID** | 1.6 |
| **User Story-Beschreibung** | Als Arzt möchte ich mindesten fünf Minuten vor dem Termin informiert werden, wenn ein Patient einen Termin nicht wahrnimmt, sodass ich andere Patienten vorziehen kann. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor(en)** | Emmett Brown |
| **Abhängigkeiten zu anderen User Stories** | 1.3, 1.5 |

## 

Papierprototypen

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

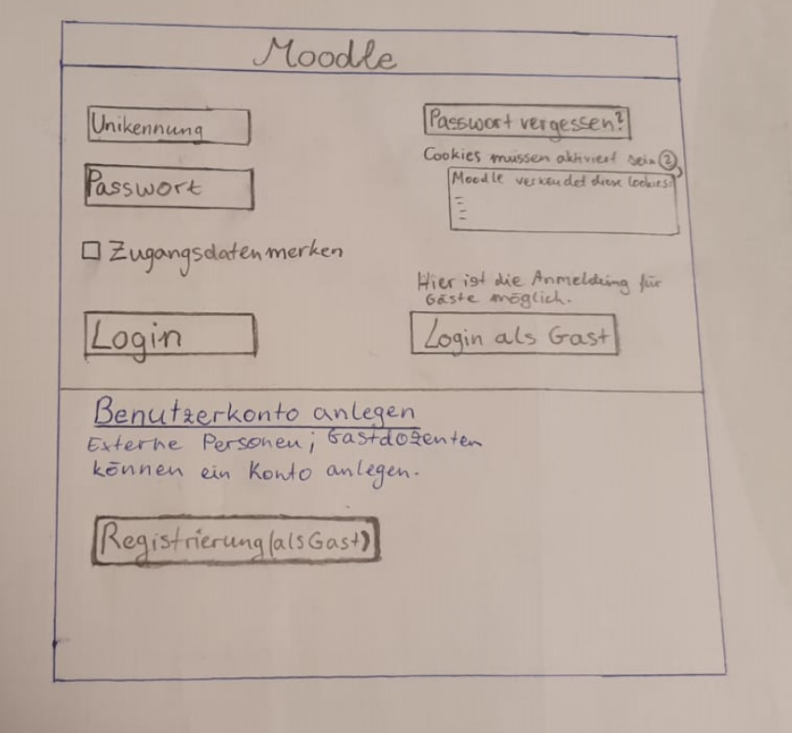
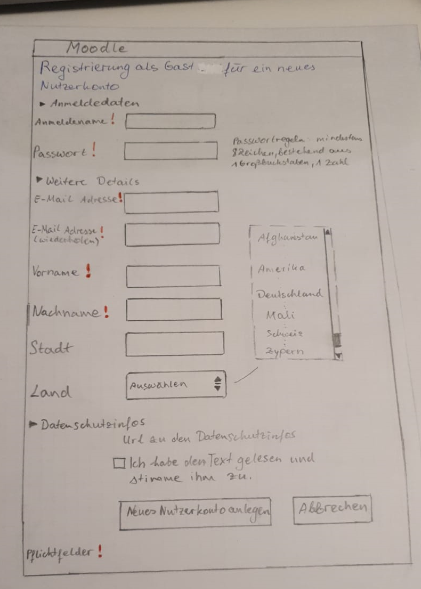
Erläuterung des Papierprototypen:

**1.3 Registrierungsseite:**

Ich habe die Oberfläche der Registrierungsseite unserer Lernplattform mit einem Papierprototypen bildlich dargestellt.

Wenn man unsere Lernplattform aufruft, entsteht die erste Seite. Der Nutzer kann sich mit seiner Unikennung und seinem zugehörigen Passwort einloggen. Dabei kann er anklicken, ob sich die Seite die Zugangsdaten merken darf oder nicht, indem er ein Häkchen in das kleine Quadrat setzt.

Rechts davon kann der Student oder das Lehrpersonal auf den Button „Passwort vergessen“ klicken, wenn er das Passwort vergessen würde. Wie bei jeder Seite müssen Cookies aktiviert werden. Wenn Fragen auftreten, ist es möglich auf das kleine Fragezeichen zu klicken, dabei entsteht ein kleines Fenster und der Nutzer kann sich die Informationen durchlesen. Rechts neben dem normalen Login, kann sich ein Gast einloggen. Damit der Gast sich einloggen kann, muss er sich erst einmal registrieren. Dieser Prozess gelingt, wenn er auf den Registrierungsbutton unten links klickt. Daraufhin öffnet sich ein zweites Fenster. Dort kann sich der Gast mit seinen Nutzerdaten registrieren. Bei der Eintragung von seinem Land, aus dem er stammt, kann sich der Nutzer über „Auswählen“ sein Land über das geöffnete Feld auswählen. Einige Felder sind mit einem roten Ausrufezeichen gekennzeichnet, dieses stellt die Erforderlichkeit für das Abschließen der Registrierung für das eigene Nutzerkonto dar. Nach dem erfolgreichen Anlegen eines Nutzerkontos, kann sich der Gast über den zugehörigen Login für den Gast einloggen und kann die Lernplattform besuchen.



## 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)

## 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.

Erläuterung des Strukturdiagramms:

## Funktionalitätsplanung

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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktiona-litäten** | **Verknüpfte**  **User-Stories** | **Quellcode-**  **referenz** | **Status** |
| **1.** | **Route** |  |  |  |  |  |  |
| 1.1 | Suche nach POI/ Straßen | 1 Tag | Max Muster | 1.2 |  |  | fertig |
| 1.2 | Erstellen von Routen | 2 Tage | Max Muster | 1.3, 1.4 |  |  | In Bearbeitung |
| 1.2.1 | …. |  |  |  |  |  |  |
| **2.** | **Anzeige** |  |  |  |  |  |  |
| 2.1 | Lokales Speichern der Routen | 4 Tage | Anja Muster | 2.4, 2,7 |  |  | fertig |
| … |  |  |  |  |  |  |  |

## Systemtests

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 03.03.2019 | | |
| **Tester** | Martina Musterfrau | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ registriert | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt den 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** | 03.03.2019 | | |
| **Tester** | Martina Musterfrau | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ registriert | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt den 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. | **√** |
| **Nachbe-dingung(en)** | Nutzer ist am System angemeldet, Anmeldezeitpunkt ist im System gespeichert. | | **√** |
| **Testurteil** | Test bestanden. | | |

# Zyklus II

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Artefakt** | **Art des Artefakts** | **Verantwortlicher** | **Status** |
| **1.** |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **2.** |  |  |  |  |
|  |  |  |  |  |
| … |  |  |  |  |

## Definition der Verantwortlichen für die Artefakte der Spezifikation

## User-Stories

Template:

|  |  |
| --- | --- |
| **User Story-ID** | <Eindeutiger Identifizierer> |
| **User Story-Beschreibung** | <Text der User Story mittels Satzschablone:  Als <Rolle> möchte ich <Ziel> [, um/sodass <Nutzen>]  (s. Foliensatz „Anforderungen“)> |
| **Geschätzter Realisierungsaufwand** | <Einschätzung der Zeit, die benötigt wird, um die Userstory zu implementieren> |
| **Priorität** | <Wichtigkeit der User Story hinsichtlich der Aufgabenstellung z.B. hoch, mittel niedrig> |
| **Autor(en)** | <Hier bitte nur einen Zuständigen eintragen z.B. Max Mustermann> |
| **Abhängigkeiten zu anderen User Stories** | <Auflistung verwandter User Stories> |

Papierprototypen

Erläuterung des Papierprototypen:

## Szenarien (MSCs)

Erläuterung der Szenarien:

## Strukturdiagramm (Klassendiagramm)

Erläuterung des Strukturdiagramms:

## Funktionalitätsplanung

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktiona-litäten** | **Verknüpfte**  **User-Stories** | **Quellcode-**  **referenz** | **Status** |
| **1.** | **Route** |  |  |  |  |  |  |
| 1.1 | Suche nach POI/ Straßen | 1 Tag | Max Muster | 1.2 |  |  | fertig |
| 1.2 | Erstellen von Routen | 2 Tage | Max Muster | 1.3, 1.4 |  |  | In Bearbeitung |
| 1.2.1 | …. |  |  |  |  |  |  |
| **2.** | **Anzeige** |  |  |  |  |  |  |
| 2.1 | Lokales Speichern der Routen | 4 Tage | Anja Muster | 2.4, 2,7 |  |  | fertig |
| … |  |  |  |  |  |  |  |

## 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** |
| **…** |  |  |  |

## Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 03.03.2019 | | |
| **Tester** | Martina Musterfrau | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ registriert | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt den 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** | 03.03.2019 | | |
| **Tester** | Martina Musterfrau | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ registriert | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt den 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. | **√** |
| **Nachbe-dingung(en)** | Nutzer ist am System angemeldet, Anmeldezeitpunkt ist im System gespeichert. | | **√** |
| **Testurteil** | Test bestanden. | | |

# Zyklus III

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Artefakt** | **Art des Artefakts** | **Verantwortlicher** | **Status** |
| **1.** |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **2.** |  |  |  |  |
|  |  |  |  |  |
| … |  |  |  |  |

## Definition der Verantwortlichen für die Artefakte der Spezifikation

## User-Stories

Template:

|  |  |
| --- | --- |
| **User Story-ID** | <Eindeutiger Identifizierer> |
| **User Story-Beschreibung** | <Text der User Story mittels Satzschablone:  Als <Rolle> möchte ich <Ziel> [, um/sodass <Nutzen>]  (s. Foliensatz „Anforderungen“)> |
| **Geschätzter Realisierungsaufwand** | <Einschätzung der Zeit, die benötigt wird, um die Userstory zu implementieren> |
| **Priorität** | <Wichtigkeit der User Story hinsichtlich der Aufgabenstellung z.B. hoch, mittel niedrig> |
| **Autor(en)** | <Hier bitte nur einen Zuständigen eintragen z.B. Max Mustermann> |
| **Abhängigkeiten zu anderen User Stories** | <Auflistung verwandter User Stories> |
| **Zugehörige Szenarien** | <Zuordnung zu Szenarien>   * Hauptszenario: * Alternativszenarien: * Ausnahmeszenarien: |

Papierprototypen

Erläuterung des Papierprototypen:

## Szenarien (MSCs)

Erläuterung der Szenarien:

## Strukturdiagramm (Klassendiagramm)

Erläuterung des Strukturdiagramms:

## Funktionalitätsplanung

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktiona-litäten** | **Verknüpfte**  **User-Stories** | **Quellcode-**  **referenz** | **Status** |
| **1.** | **Route** |  |  |  |  |  |  |
| 1.1 | Suche nach POI/ Straßen | 1 Tag | Max Muster | 1.2 |  |  | fertig |
| 1.2 | Erstellen von Routen | 2 Tage | Max Muster | 1.3, 1.4 |  |  | In Bearbeitung |
| 1.2.1 | …. |  |  |  |  |  |  |
| **2.** | **Anzeige** |  |  |  |  |  |  |
| 2.1 | Lokales Speichern der Routen | 4 Tage | Anja Muster | 2.4, 2,7 |  |  | fertig |
| … |  |  |  |  |  |  |  |

## Unittests

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Getestete Funktionalität** | **Quellcode**  **Referenz** | **Status** |
| **…** |  |  |  |

## Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** |  | | |
| **Tester** |  | | |
| **SW-Version** |  | | |
| **Vorbedin-gung(en)** |  | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| **Nachbe-dingung(en)** |  | |  |
| **Testurteil** |  | | |

# Nutzerhandbuch

## Technische Anforderungen

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

## Installationsanleitung

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

## Bedienungsanleitung

Genaue Erläuterung, wie das entwickelte Programm zu bedienen ist.