

Fachhochschule der Wirtschaft
-FHDW-
Paderborn

Dokumentation

Student resource planning

Verfasser:

David Langen, Marius Bürck,
Philipp Grosch, Sebastian Pietrzak

Eingereicht am:

04.06.2018

Inhaltsverzeichnis

Inhaltsverzeichnis.....	II
Abbildungsverzeichnis.....	III
Tabellenverzeichnis.....	IV
Abkürzungsverzeichnis.....	V
1 Einleitung	1
2 Fachkonzept.....	2
2.1 User-Stories.....	2
2.2 Use Cases	4
2.3 Wireframes	5
2.4 Datenmodell	12
3 Projektplan	13
3.1 Aufgabenaufteilung.....	14
4 Implementierung.....	16
5 Test	17
6 Projektmanagement.....	18
6.1 Netzplan	18
6.2 Aufgabenaufteilung.....	21
7 Bewertung und Fazit.....	23
Anhang	24
Anhangsverzeichnis.....	24
Anhang Tabellenverzeichnis.....	25

Abbildungsverzeichnis

Abbildung 1 Architektur	1
Abbildung 2 Use Case Diagramm: student-resouce-planning	4
Abbildung 3 Wireframe: Login	6
Abbildung 4 Wireframe: Index	7
Abbildung 5 Wireframe: show student	8
Abbildung 6 Wireframe: edit student	9
Abbildung 7 Wireframe: create student	10
Abbildung 8 Wireframe: create course	11
Abbildung 9 ERM	12
Abbildung 10 Netzplan	13
Abbildung 11 Netzplan final teil 1	19
Abbildung 12 Netzplan final teil 2	20

Tabellenverzeichnis

Tabelle 1 UC Tabelle: Student anlegen	5
Tabelle 2 Aufgabenaufteilung Planung	15
Tabelle 3 Testfall: Login Daten korrekt eingeben	17
Tabelle 4 Arbeitsaufteilung final	22

Abkürzungsverzeichnis

AWE	Anwendungsentwicklung
ERM	Entity Relationship Model
GUI	Graphical User Interface
JAR	Java Archiv
PDF	Portable Document Format
UC	Use Case

1 Einleitung

Im Rahmen des Studiums im Fachbereich Angewandter Informatik müssen wir eine Java-basierte Web-Anwendung zur Verwaltung von Studierenden konzeptionieren, realisieren und dokumentieren.

Für die Aufgabe wurden uns einige Anforderungen, sowie Rahmenbedingungen vorgegeben.

Folgende Rahmenbedingungen liegen vor

- Arbeit in Gruppen von 3-4 Personen
- Programmiersprache Java
- Angemessener Einsatz von Methoden des Software Engineering
- Server lässt sich bei einem fiktiven Kunden installieren (z.B. Export als JAR)

Architekturskizze:

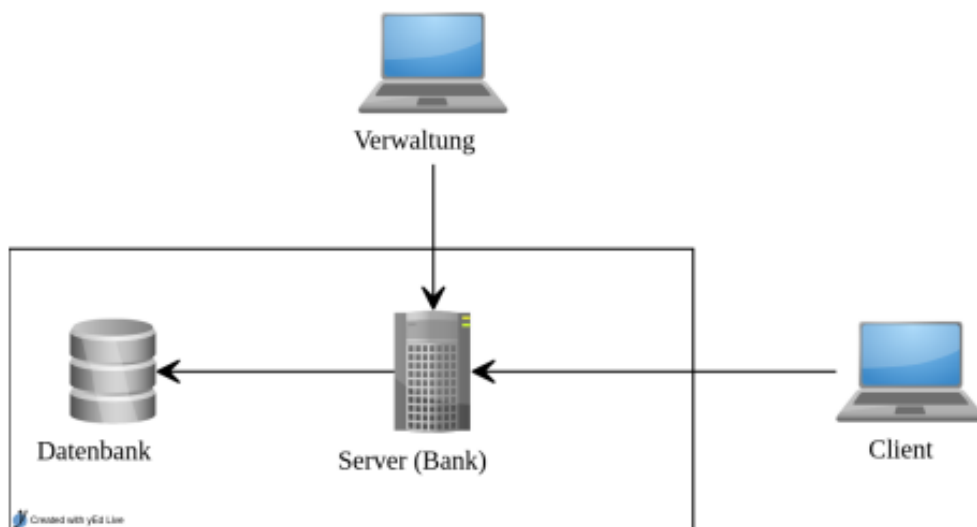


Abbildung 1 Architektur

Quelle: PDF AWE I Projektvorgaben, Christian Heuermann

2 Fachkonzept

Die folgenden Kapitel zeigen eine erweiterte Beschreibung der Anforderungen sowie das konzeptionelle Design für die Oberfläche der Anwendung und die Struktur der Datenbank.

2.1 User-Stories

Im Folgenden werden die von uns, anhand von den vorgegebenen Anforderungen, erstellten User-Stories aufgelistet. Die User-Stories sollen dabei, dank der speziellen Formulierung, ein leichteres Verständnis für die Kunden sowie für die Entwickler bieten und es ermöglichen, sich einen besseren Eindruck aus der Nutzersicht zu verschaffen.

Haupt-Stories

- Als Verwalter möchte ich...
 - den Zugriff auf das System einschränken, sodass keine Unbefugten Zugang erlangen.
 - Datensätze von Studenten im System hinterlegen, sodass diese weiterverarbeitet werden können.
 - über eine Suchfunktion nach Studenten suchen, um schneller auf die zugeordneten Daten zugreifen zu können.
 - Datensätze von Studenten löschen können, sodass sie nicht länger im System angezeigt werden.
 - über eine Weboberfläche von überall auf das System zugreifen können, um flexibel bei der Arbeit zu sein.
 - eine automatisierte Überprüfung der Eingaben, um grobe Eingabefehler zu vermeiden.
 - eine Detailansicht auf den Datensatz eines Studenten.
 - neue Kurse anlegen können, sodass diese den Studenten zugeordnet werden können.
 - Kurse den Studenten zuordnen können.
 - Kurse löschen, sodass sie nicht mehr im System existieren.
 - Studentendaten bearbeiten, sodass Änderungen ins System übernommen werden können.

Optionale-stories

- Als Verwalter möchte ich...
 - Zeugnisse als PDF-Datei exportieren, um Zeit bei der Ausstellung zu sparen.
 - Noten von Studenten in Kursen verwalten, sodass diese weiterverarbeitet werden können.
 - neue Benutzer anlegen können, damit diese Zugang zum System erlangen.
 - Benutzer sperren können, um diesen den Zugang zum System zu entziehen.
 - die Durchschnittsnote von allen Noten innerhalb eines Kurses, um diese Kennzahl sichtbar zu machen.

2.2 Use Cases

Die nächste Abbildung zeigt das Use Case Diagramm fürs „student-resource-planning“. Die Abbildung soll das Verhalten des Systems aus der Sicht der Anwender beschreiben, indem das Diagramm die Beziehungen zwischen den Akteuren und den Anwendungsfällen darstellt.

Eine Vorbedingung für alle Use Cases ist, dass der entsprechende Nutzer im System angemeldet ist.

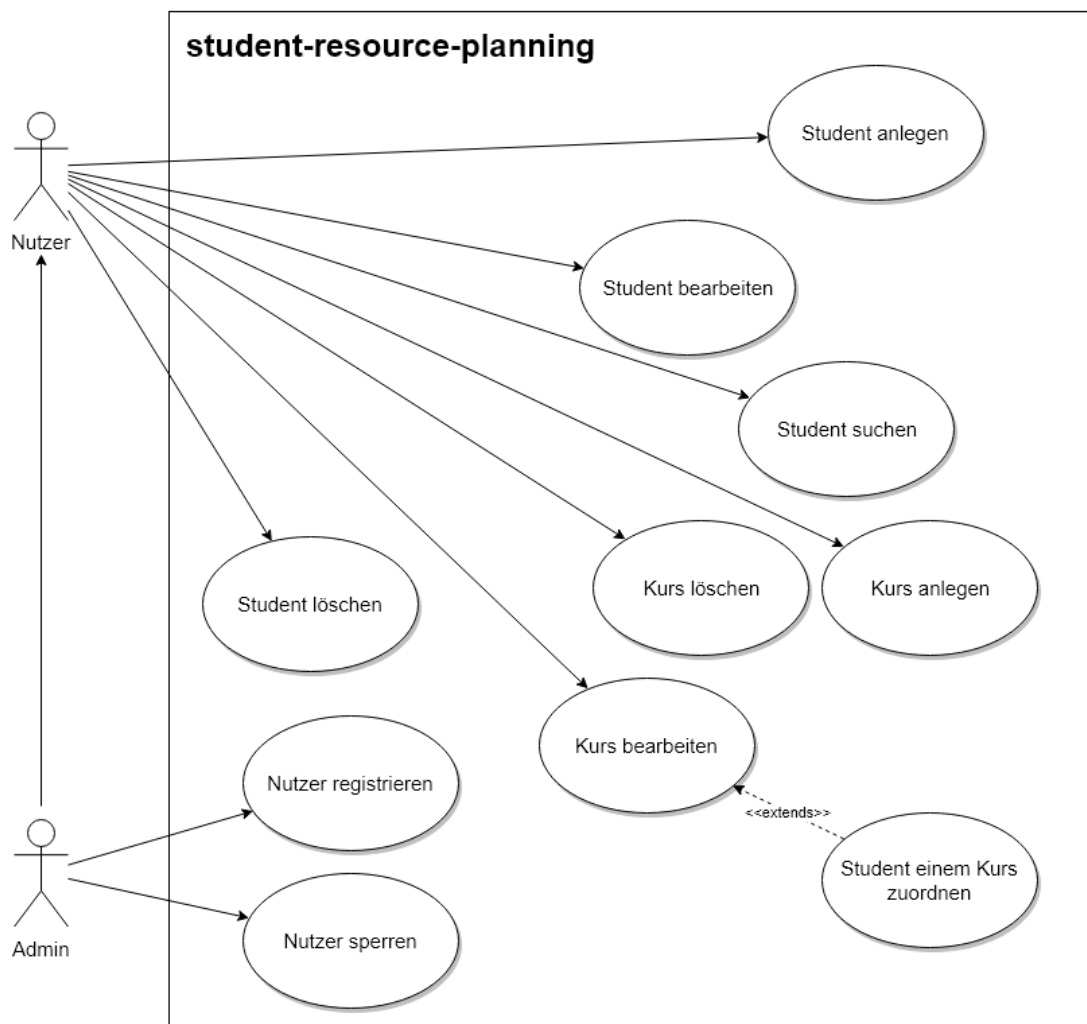


Abbildung 2 Use Case Diagramm: student-resouce-planning

Des Weiteren wurde für jeden Anwendungsfall eine Use Case Tabelle angefertigt.

Diese Tabellen dienen zur Beschreibung der einzelnen Anwendungsfälle.

Die folgende Tabelle zeigt die Beschreibung des Anwendungsfalls „Student anlegen“.

Alle weiteren Use Case Tabellen befinden sich im Anhang.

Bezeichnung	Student anlegen
Beschreibung	Der Nutzer soll neue Studenten erfassen können.
Ebene	Hauptfunktion
Akteur	Nutzer
Vorbedingung	
Nachbedingung	Ein neuer Student ist im System hinterlegt
Hauptablauf	<ul style="list-style-type: none">• Das System bietet die Möglichkeit Studenten zu erfassen.• Der Nutzer gibt die Studenten-Daten ein und bestätigt diese.• Das System überprüft die Daten und erfasst den Studenten.
Ausnahmen	Falls Eingaben fehlerhaft sind wird die Eingabemaske erneut angeboten und eine entsprechende Mitteilung angezeigt.
Anmerkung	

Tabelle 1 UC Tabelle: Student anlegen

2.3 Wireframes

Für eine konzeptionelle Darstellung der Benutzeroberfläche der Anwendung wurden Wireframes angefertigt. Diese werden nun in den nächsten Abbildungen dargestellt.

Die folgende Abbildung stellt die Login-Seite der Anwendung dar.

The image shows a wireframe of a login page within a web browser window. The browser's address bar displays the URL `https://www.studentenverwaltung.com`. The login form is centered on the page and consists of a rounded rectangle containing the following elements:

- A label **Benutzername:** followed by a text input field with the placeholder text "Benutzername".
- A label **Passwort:** followed by a password input field with placeholder text "*****".
- A "Login" button positioned to the right of the password field.

Abbildung 3 Wireframe: Login

In diesem Fenster wird die Index Seite der Anwendung dargestellt und zeigt die einzelnen Studenten an, deren Daten der Nutzer bearbeiten ansehen und löschen kann.

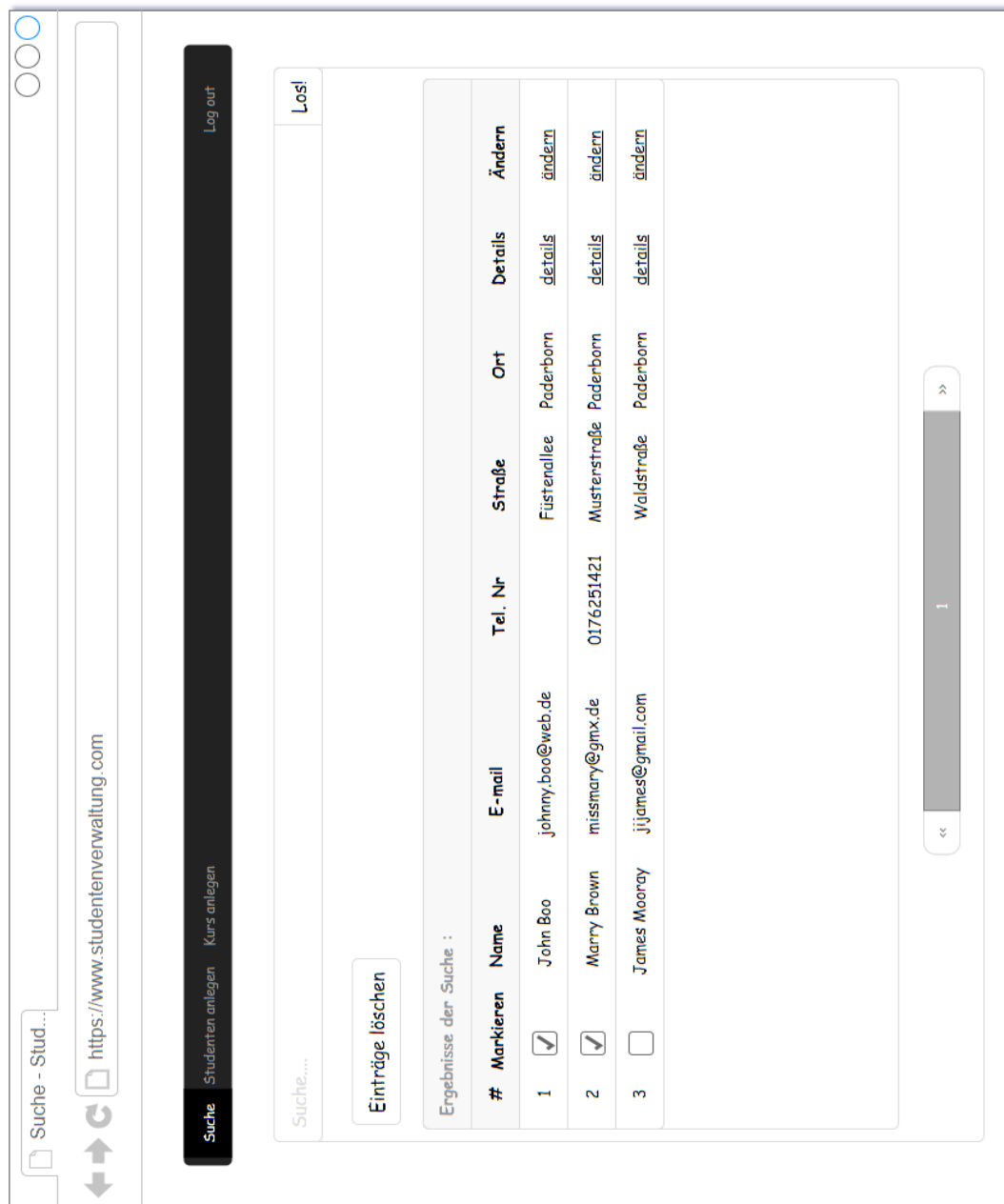


Abbildung 4 Wireframe: Index

Das Fenster zeigt dem Benutzer die Details eines Studenten an.

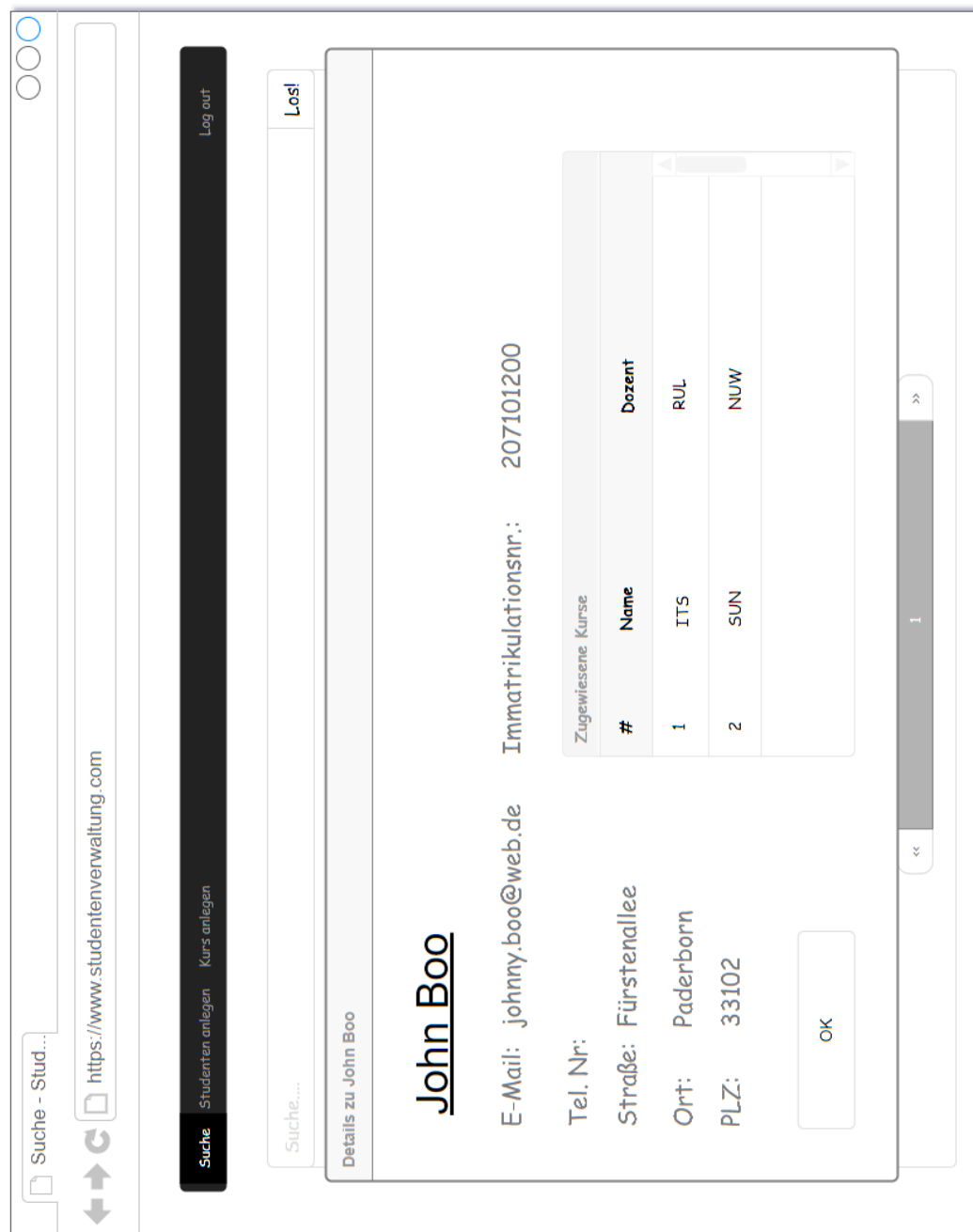


Abbildung 5 Wireframe: show student

Im Folgenden Fenster kann der Nutzer die einzelnen Attribute eines Studierenden ändern und speichern.

Suche - Stud...
<https://www.studentenverwaltung.com>

Eintrag von John Boo ändern

John Boo

Immatrikulationsnr.: 207101200

Vorname: John
 Nachname: Boo
 Straße und Hausnummer: Fürstenallee 3
 Ort: Paderborn
 PLZ: 33102
 Geburtsort: 10 Juni 1996
 E-Mail: johnny.boo@web.de
 Tel. Nr.:
 Kurs: ITS
 Kurs zuweisen
 Kurs entfernen

Zugewiesene Kurse			
#	Markieren	Name	Dozent
1	<input checked="" type="checkbox"/>	ITS	RUL

Änderungen übernehmen

Abbildung 6 Wireframe: edit student

In diesem Fenster kann der Nutzer neue Studenten anlegen.

Suche Studenten anlegen Kurs anlegen Log out

Neuen Studenten anlegen

Vorname Nachname

Straße und Hausnummer Ort PLZ

Geburtsjahr 10 1996

E-Mail Tel. Nr.

Kurs ITS

Kurs zuweisen Kurs entfernen

Zugewiesene Kurse

#	Markieren	Name	Dozent
1	<input checked="" type="checkbox"/>	ITS	RUL

Eintrag anlegen

Abbildung 7 Wireframe: create student

Diese Ansicht ist dazu da, um einen neuen Kurs anzulegen.

The wireframe depicts a web application for creating a new course. It features a browser window at the top with a tab labeled 'Kurs anlege...', a search bar, and a URL 'https://www.studentenverwaltung.com'. Below the browser is a dark navigation bar with links for 'Suche', 'Studenten anlegen', and 'Kurs anlegen' (the latter being highlighted), and a 'Log out' button. The main content area is titled 'Neuen Kurs anlegen' and contains three input fields: 'Kursbezeichnung:', 'Dozent:', and 'Beschreibung:'. A large 'Eintrag anlegen' button is positioned at the bottom right of the form.

Abbildung 8 Wireframe: create course

2.4 Datenmodell

Die folgende Abbildung zeigt das zugrundeliegende Datenmodell in Form eines Entity Relationship Models, dass die Verbindung zwischen den Tabellen darstellt.

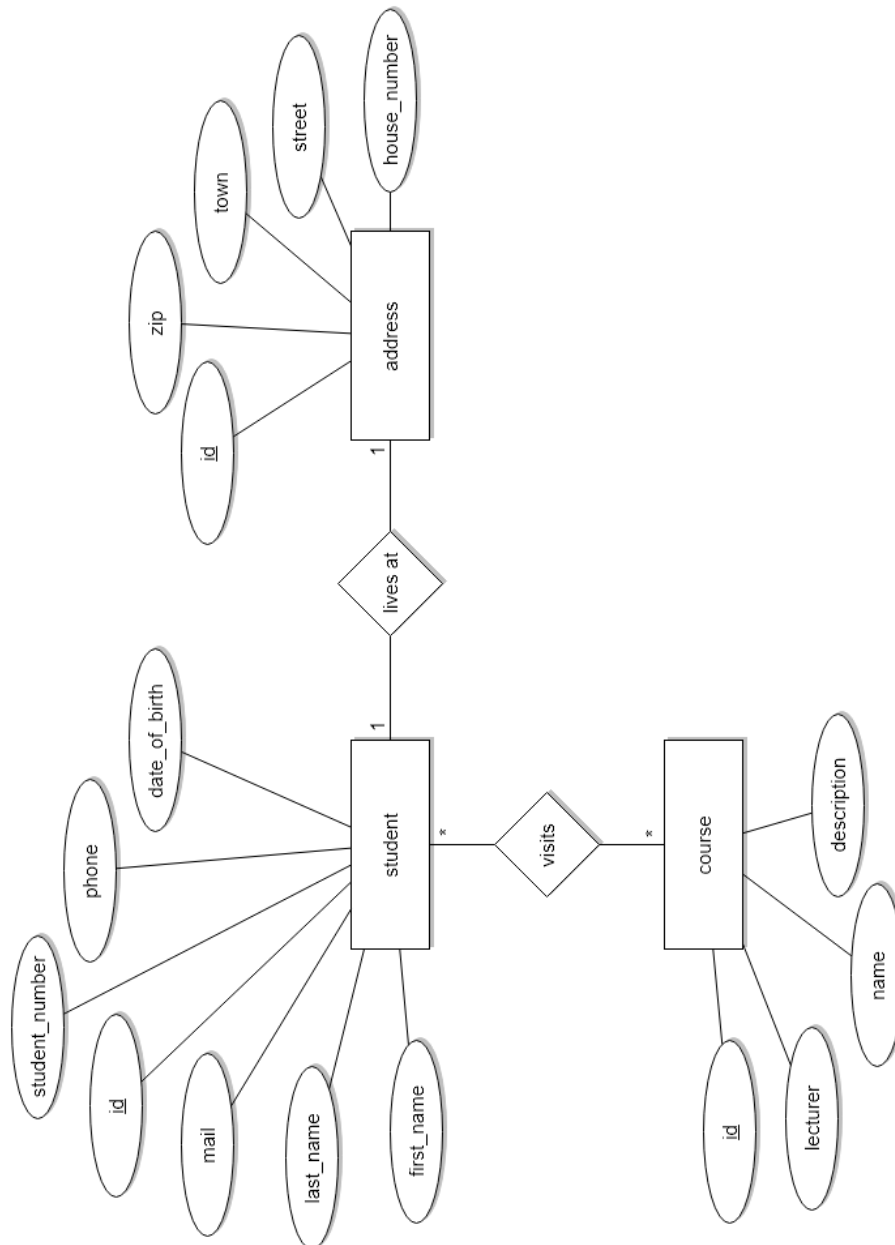


Abbildung 9 ERM

3 Projektplan

Für die Planung des Projekts wurde ein Netzplan erstellt. Dieser zeigt die geplante Aufteilung der Aufwände. An den Meilensteinen (durch grüne Linien gekennzeichnet) sieht das Team die kritischen Punkte, an denen über den Projektfortschritt reflektiert werden soll. Falls es starke Abweichungen gibt, sieht das Team sofort, dass gehandelt werden muss, mit z.B. Überstunden.

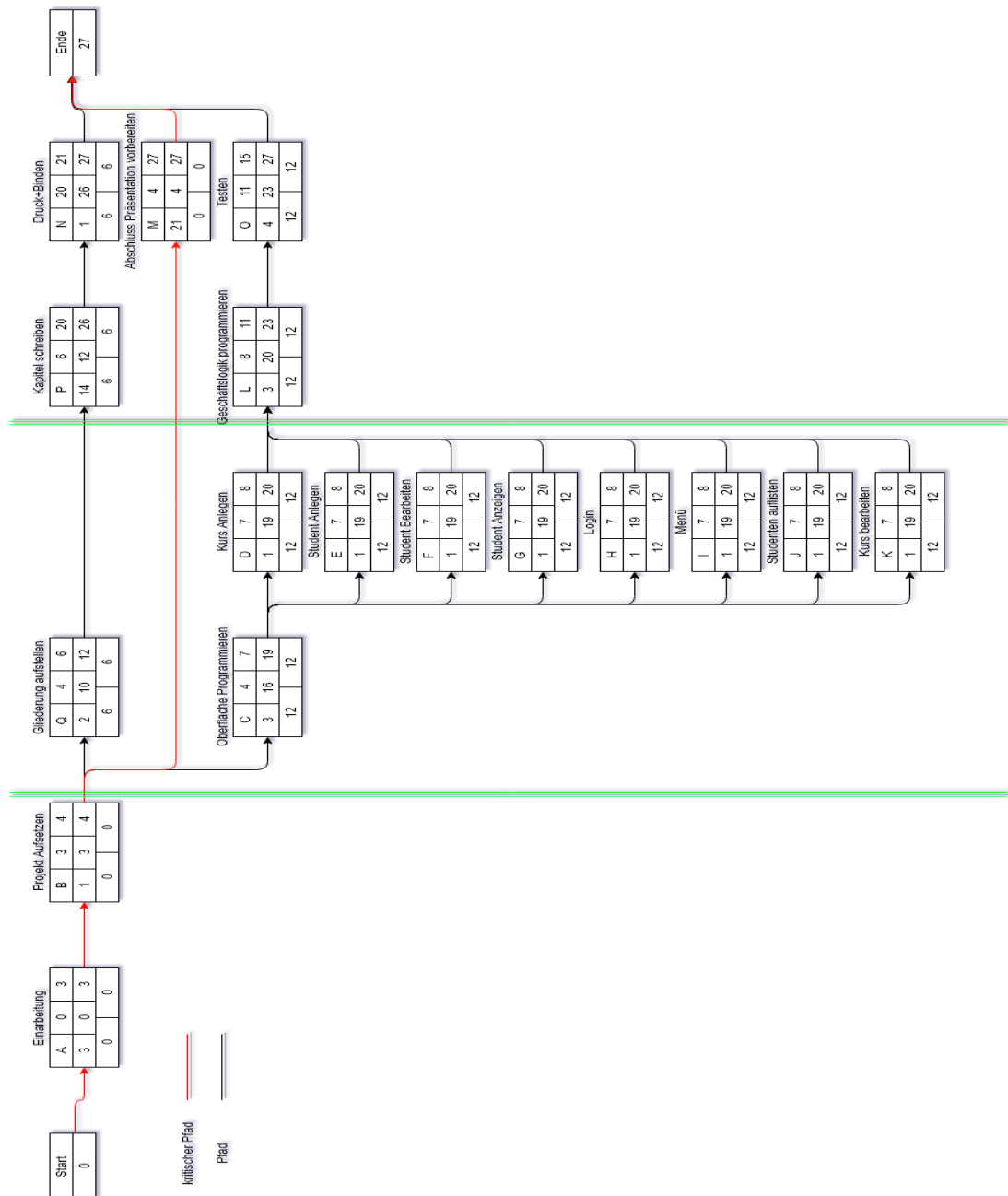


Abbildung 10 Netzplan

3.1 Aufgabenaufteilung

Im Folgenden wird aufgelistet wem welches Aufgabenpaket bei der Projektplanung zugeteilt wurde.

Arbeitspaket	Zuteilung
Einarbeiten	<ul style="list-style-type: none">• David Langen• Marius Bürck• Philipp Grosch• Sebastian Pietrzak
Projekt aufsetzen	<ul style="list-style-type: none">• David Langen• Marius Bürck• Philipp Grosch• Sebastian Pietrzak
Gliederung Aufstellen	<ul style="list-style-type: none">• Philipp Grosch
Oberfläche Programmieren	<ul style="list-style-type: none">• David Langen• Marius Bürck• Sebastian Pietrzak
Student anlegen	<ul style="list-style-type: none">• Sebastian Pietrzak
Student anzeigen	<ul style="list-style-type: none">• David Langen
Student auflisten	<ul style="list-style-type: none">• David Langen
Student bearbeiten	<ul style="list-style-type: none">• Philipp Grosch
Login	<ul style="list-style-type: none">• David Langen
Menü	<ul style="list-style-type: none">• David Langen
Kurs anlegen	<ul style="list-style-type: none">• Marius Bürck
Kurs bearbeiten	<ul style="list-style-type: none">• Marius Bürck
Geschäftslogik programmieren	<ul style="list-style-type: none">• David Langen• Marius Bürck• Philipp Grosch• Sebastian Pietrzak

Kapitel schreiben	<ul style="list-style-type: none">• Marius Bürck• Philipp Grosch• Sebastian Pietrzak
Testen	<ul style="list-style-type: none">• Philipp Grosch• Sebastian Pietrzak
Abschlusspräsentation vorbereiten	<ul style="list-style-type: none">• David Langen• Marius Bürck• Philipp Grosch• Sebastian Pietrzak
Druck/bindern	<ul style="list-style-type: none">• Sebastian Pietrzak

Tabelle 2 Aufgabenaufteilung Planung

4 Implementierung

Die Anwendung basiert auf dem Spring¹ Framework. Spring ist ein quelloffenes Java Framework und bietet einen großen Funktionsumfang an, wodurch weitgehend vermieden werden kann zusätzliche Programmbibliotheken hinzuziehen zu müssen.

Darüber hinaus wird Maven eingesetzt um das Softwareprojekt und dessen Abhängigkeiten zu verwalten. Die Entscheidung zu diesem Projektaufbau viel aufgrund von Recherchen und Anraten eines externen Experten, der bereits weitgreifende Erfahrungen mit Java-Webentwicklung gemacht hat.

Da das Projektteam in diesem Bereich nur begrenzt über Erfahrung verfügt, soll durch den Einsatz von Spring auch ein Rahmenwerk geschaffen werden um schlechtem Programmierstil vorzubeugen und Qualitätskriterien wie Erweiterbarkeit unterstützt werden.

Es wird für die Anwendung kein Klassendiagramm entworfen, da der Sinn und Zweck der Klassen durch die entsprechenden Basisklassen von Spring erklärt sind. Alle konkreten Erweiterungen, werden durch die beiliegende Implementationsdokumentation² erklärt. Die Entitäten, die letztlich durch Klassen abgebildet werden, sind durch das Entity Relationship Model³ beschrieben.

Zu dem Basisframework werden über Maven⁴ weitere Abhängigkeiten hinzugefügt, um Funktionen hinzuzufügen. Dazu gehört die H2⁵ Database, die als relationale Datenbank genutzt und zurzeit "in-memory" betrieben wird. Außerdem wird Thymeleaf⁶ als Templating Engine hinzugefügt, um dynamische HTML-Seiten zu ermöglichen. Weiterhin wichtig ist das Bootstrap⁷ CSS-Framework, das genutzt wurde um leichter und schneller GUI-Prototypen zu erzeugen und die Oberfläche grundsätzlich zu strukturieren. Es wird sich an die Dokumentationen und Empfehlungen von Spring bzw. den jeweiligen Paketen gehalten.

¹ <https://spring.io/>

² Anhang CD javadoc

³ Siehe Abbildung 9 ERM

⁴ <https://maven.apache.org/>

⁵ <http://www.h2database.com/html/main.html>

⁶ <https://www.thymeleaf.org/>

⁷ <https://getbootstrap.com/>

5 Test

Zum Testen der Anwendung wurden Akzeptanztests anhand der Use Cases durchgeführt. Als Beispiel folgt der Testfall „Login Daten korrekt eingeben“

Testfall	Login Daten korrekt eingeben
Vorbedingung	Keine
Auszuführende Testschritte	<ol style="list-style-type: none">1. Username eingeben2. Passwort eingeben3. „Log In“-button betätigen
Erwartetes Ergebnis	Nutzer loggt sich ein und wird zur Hauptseite weitergeleitet
Verwendete Testdaten	<ul style="list-style-type: none">• Benutzername: david• Passwort: password
Ergebnis	Nutzer wurde eingeloggt und zur Hauptseite weitergeleitet
Zeitpunkt des Tests	02.06.2018

Tabelle 3 Testfall: Login Daten korrekt eingeben

6 Projektmanagement

Der Verlauf des Projekts verlief summa summarum Positiv. Vergleicht man die Aufgabenpakete in dem Netzplan in „Abbildung 10“ mit der fertigen Anwendung erkennt man, dass alle geplanten Aufgaben erledigt wurden.

Im Laufe des Projekts wurden weitere Aufgabenpakete in das Projekt aufgenommen, an die beim Anfang der Projektplanung nicht gedacht wurden, z.B. „Bugfixing“. Diese Aufgaben wurden später zur finalen Version des Netzplans hinzugefügt.

Jeden Mittwoch und vor jedem Meilenstein wurde ein Meeting abgehalten indem wir die Anwendung nochmal durchgehen, um festzustellen welche Arbeitspakete zu erledigen sind und welche neuen Aufgaben dazugekommen sind und um zu schauen ob das Projekt noch realisierbar ist.

6.1 Netzplan

Die Folgende Abbildungen stellen die Finale Version des Netzplans dar.

Um den Netzplan lesbar auf den Seiten darzustellen wurde der Netzplan in zwei Teile aufgeteilt die untereinander vergrößert dargestellt wurden.

Der zusammengesetzte Netzplan befindet sich im Anhang in der beigelegten CD.

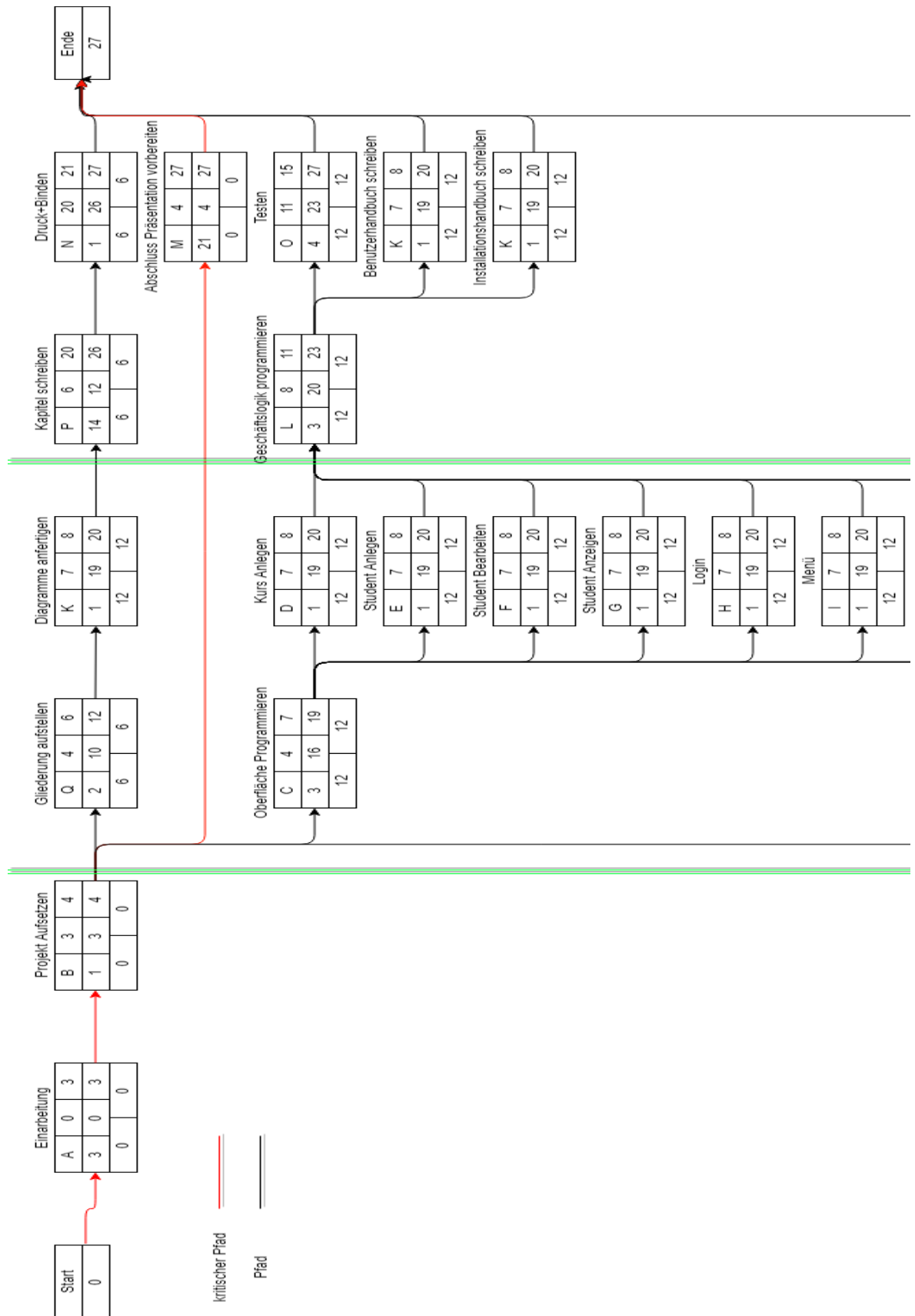


Abbildung 11 Netzplan final teil 1

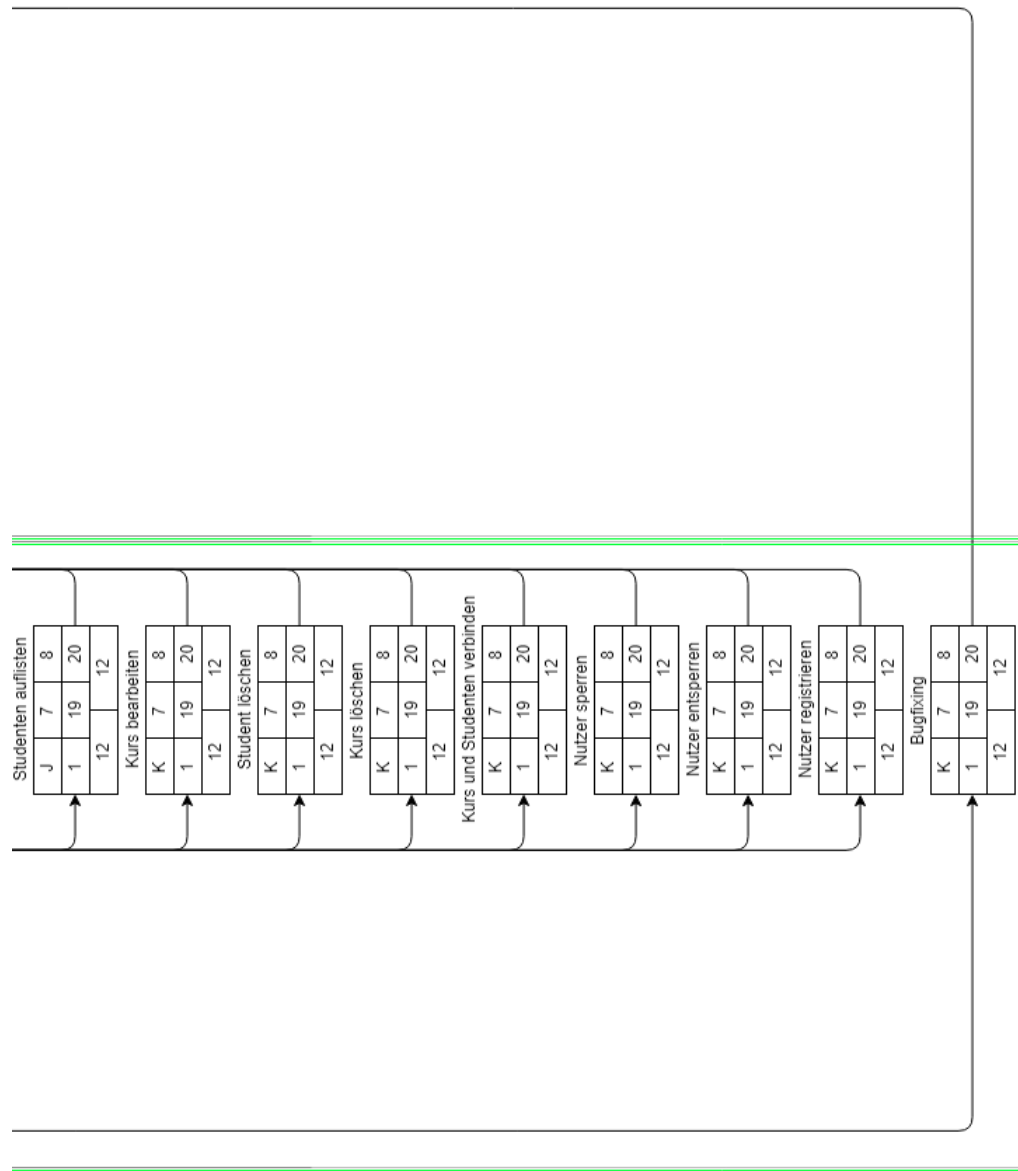


Abbildung 12 Netzplan final teil 2

6.2 Aufgabenaufteilung

Die nächste Tabelle zeigt die finale Aufgabenaufteilung mit den dazugekommenen Arbeitspaketen.

Arbeitspaket	Zuteilung
Einarbeiten	<ul style="list-style-type: none">• David Langen• Marius Bürck• Philipp Grosch• Sebastian Pietrzak
Projekt aufsetzen	<ul style="list-style-type: none">• David Langen• Marius Bürck• Philipp Grosch• Sebastian Pietrzak
Gliederung Aufstellen	<ul style="list-style-type: none">• Philipp Grosch
Oberfläche Programmieren	<ul style="list-style-type: none">• David Langen• Marius Bürck• Sebastian Pietrzak• Philipp Grosch
Student anlegen	<ul style="list-style-type: none">• Sebastian Pietrzak
Student anzeigen	<ul style="list-style-type: none">• David Langen
Student auflisten	<ul style="list-style-type: none">• David Langen
Student bearbeiten	<ul style="list-style-type: none">• Philipp Grosch
Login	<ul style="list-style-type: none">• David Langen
Menü	<ul style="list-style-type: none">• David Langen
Kurs anlegen	<ul style="list-style-type: none">• Marius Bürck
Kurs bearbeiten	<ul style="list-style-type: none">• Marius Bürck
Geschäftslogik programmieren	<ul style="list-style-type: none">• David Langen• Marius Bürck• Philipp Grosch• Sebastian Pietrzak

Kapitel schreiben	<ul style="list-style-type: none"> • Marius Bürck • Philipp Grosch • Sebastian Pietrzak
Testen	<ul style="list-style-type: none"> • Philipp Grosch • Sebastian Pietrzak
Abschlusspräsentation vorbereiten	<ul style="list-style-type: none"> • David Langen • Marius Bürck • Philipp Grosch • Sebastian Pietrzak
Druck/Bindung/CD	<ul style="list-style-type: none"> • Marius Bürck
Student Löschen	<ul style="list-style-type: none"> • Philipp Grosch • David Langen
Kurs Löschen	<ul style="list-style-type: none"> • Marius Bürck
Kurs und Studenten verbinden	<ul style="list-style-type: none"> • Marius Bürck
Bugfixing	<ul style="list-style-type: none"> • David Langen • Marius Bürck • Philipp Grosch • Sebastian Pietrzak
Nutzer sperren	<ul style="list-style-type: none"> • David Langen
Nutzer entsperren	<ul style="list-style-type: none"> • David Langen
Nutzer registrieren	<ul style="list-style-type: none"> • David Langen
Benutzerhandbuch schreiben	<ul style="list-style-type: none"> • Philipp Grosch
Installationshandbuch schreiben	<ul style="list-style-type: none"> • David Langen
Diagramme anfertigen	<ul style="list-style-type: none"> • Philipp Grosch • Marius Bürck • Sebastian Pietrzak

Tabelle 4 Arbeitsaufteilung final

7 Bewertung und Fazit

Während der Projektlaufzeit wurden alle geplanten Ziele erfüllt. Am Ende dieser ersten Iteration existiert ein ausbaufähiger Prototyp. In Zukunft kann das System um weitere Funktionen erweitert werden, wie z.B. eine Notenverwaltung für Studierende.

Die Projektdurchführung verlief reibungslos. Das Team hat gut harmoniert. Besonders zu Beginn stellten die neuen Technologien für das Team eine Herausforderung dar. Da dies aber bereits zu Beginn berücksichtigt wurde, hat das Team sich ausschließlich auf die Muss-Anforderungen konzentriert, um den Projekterfolg nicht zu gefährden. In einer weiteren Ausbaustufe traut sich das Team jetzt in einen höheren Arbeitsumfang zu.

Anhang

Anhangsverzeichnis

1	Use Case Tabellen	26
2	Tests	35
3	CD	41

Anhang Tabellenverzeichnis

Anhang Tabelle 1 UC Student bearbeiten.....	26
Anhang Tabelle 2 UC Student löschen	27
Anhang Tabelle 3 UC Student suchen	28
Anhang Tabelle 4 UC Kurs anlegen.....	29
Anhang Tabelle 5 UC Kurs löschen	30
Anhang Tabelle 6 UC Kurs bearbeiten.....	31
Anhang Tabelle 7 UC Student einem Kurs zuordnen.....	32
Anhang Tabelle 8 UC Nutzer registrieren	33
Anhang Tabelle 9 UC Nutzer sperren	34
Anhang Tabelle 10 Testfall: Student anlegen.....	35
Anhang Tabelle 11 Testfall: Student bearbeiten	36
Anhang Tabelle 12 Testfall: Student löschen	36
Anhang Tabelle 13 Testfall: Student suchen.....	37
Anhang Tabelle 14 Testfall: Kurs anlegen	37
Anhang Tabelle 15 Testfall: Kurs bearbeiten	38
Anhang Tabelle 16 Testfall: Kurs löschen.....	38
Anhang Tabelle 17 Testfall: Kurs einem Studenten zuordnen.....	39
Anhang Tabelle 18 Testfall: Nutzer registrieren	39
Anhang Tabelle 19 Testfall: Nutzer sperren	40

1 Use Case Tabellen

Bezeichnung	Student bearbeiten
Beschreibung	Der Nutzer soll die Daten von erfassten Studenten bearbeiten können.
Ebene	Hauptfunktion
Akteur	Nutzer
Vorbedingung	Mindestens ein Student muss sich im System befinden
Nachbedingung	Die Änderungen sind im System hinterlegt.
Hauptablauf	<ul style="list-style-type: none">• Das System bietet die Funktion Studenten-Daten zu ändern.• Der Nutzer bearbeitet die Daten und bestätigt diese.• Das System überprüft die Daten und ändert diese.
Ausnahmen	Falls Eingaben fehlerhaft sind wird die Eingabemaske erneut angeboten und eine entsprechende Mitteilung angezeigt.
Anmerkung	

Anhang Tabelle 1 UC Student bearbeiten

Bezeichnung	Student löschen
Beschreibung	Der Nutzer soll erfasste Studenten löschen können.
Ebene	Hauptfunktion
Akteur	Nutzer
Vorbedingung	Mindestens ein Student ist im System hinterlegt
Nachbedingung	Der entsprechende Student ist aus dem System gelöscht.
Hauptablauf	<ul style="list-style-type: none">• Das System bietet die Möglichkeit erfasste Studenten zu löschen.• Der Nutzer betätigt die Funktion zum Löschen der Studenten.• Das System löscht die Studenten.
Ausnahmen	
Anmerkung	

Anhang Tabelle 2 UC Student löschen

Bezeichnung	Student suchen
Beschreibung	Der Nutzer soll nach Studenten suchen können.
Ebene	Nebenfunktion
Akteur	Nutzer
Vorbedingung	Mindestens ein Student ist im System hinterlegt.
Nachbedingung	Die Suchergebnisse werden angezeigt.
Hauptablauf	<ul style="list-style-type: none">• Das System bietet die Möglichkeit nach Studenten suchen zu können.• Der Nutzer gibt einen Suchbegriff ein und bestätigt diesen.• Das System liefert die Studenten mit den übereinstimmenden Daten zurück.
Ausnahmen	
Anmerkung	

Anhang Tabelle 3 UC Student suchen

Bezeichnung	Kurs anlegen
Beschreibung	Der Nutzer soll Kurse anlegen können.
Ebene	Hauptfunktion
Akteur	Nutzer
Vorbedingung	
Nachbedingung	Ein neuer Kurs ist im System abgespeichert.
Hauptablauf	<ul style="list-style-type: none">• Das System bietet die Möglichkeit Kurse anzulegen.• Der Nutzer gibt benötigten Daten für die Kurse ein und bestätigt diese.• Das System legt den Kurs an.
Ausnahmen	
Anmerkung	

Anhang Tabelle 4 UC Kurs anlegen

Bezeichnung	Kurs löschen
Beschreibung	Der Nutzer soll Kurse löschen können.
Ebene	Hauptfunktion
Akteur	Nutzer
Vorbedingung	Mindestens ein Kurs ist im System hinterlegt.
Nachbedingung	
Hauptablauf	<ul style="list-style-type: none">• Das System bietet die Möglichkeit Kurse zu löschen.• Der Nutzer betätigt die Funktion zum Löschen der Kurse.• Das System löscht die Kurse.
Ausnahmen	
Anmerkung	

Anhang Tabelle 5 UC Kurs löschen

Bezeichnung	Kurs bearbeiten
Beschreibung	Der Nutzer soll Kurse bearbeiten können.
Ebene	Hauptfunktion
Akteur	Nutzer
Vorbedingung	Mindestens ein Kurs ist im System hinterlegt.
Nachbedingung	
Hauptablauf	<ul style="list-style-type: none">• Das System bietet die Möglichkeit Kurse zu bearbeiten.• Der Nutzer bearbeitet die Daten und bestätigt diese.• Das System überprüft die Daten und ändert diese.
Ausnahmen	
Anmerkung	

Anhang Tabelle 6 UC Kurs bearbeiten

Bezeichnung	Student einem Kurs zuordnen
Beschreibung	Der Nutzer soll Studenten den Kursen zuordnen können.
Ebene	Hauptfunktion
Akteur	Nutzer
Vorbedingung	Mindestens ein Kurs ist im System hinterlegt. Mindestens ein Student ist im System hinterlegt.
Nachbedingung	Die Zuordnung ist im System hinterlegt.
Hauptablauf	<ul style="list-style-type: none"> • Das System bietet die Möglichkeit Kurse den Studenten zuzuordnen. • Der User betätigt die Funktion um Kurse den Studenten zuzuordnen. • Das System ordnet die Kurse den Studenten zu.
Ausnahmen	
Anmerkung	

Anhang Tabelle 7 UC Student einem Kurs zuordnen

Optional

Bezeichnung	Nutzer registrieren
Beschreibung	Der Admin soll neue Nutzer registrieren können.
Ebene	Hauptfunktion
Akteur	Admin
Vorbedingung	
Nachbedingung	Ein neuer Nutzer ist im System registriert und kann sich mit seinen Zugangsdaten anmelden.
Hauptablauf	<ul style="list-style-type: none">• Das System bietet dem Admin die Funktion neue Nutzer zu registrieren• Der Admin gibt die Daten des Nutzers ein und bestätigt diese.• Das System registriert den neuen Nutzer.
Ausnahmen	Falls Eingaben invalide sind wird eine entsprechende Meldung angezeigt und die Eingabe erneut angeboten.
Anmerkung	

Anhang Tabelle 8 UC Nutzer registrieren

Bezeichnung	Nutzer sperren
Beschreibung	Der Admin soll Nutzer sperren können.
Ebene	Hauptfunktion
Akteur	Admin
Vorbedingung	Mindestens ein zusätzlicher Nutzer ist im System registriert.
Nachbedingung	Der entsprechende Nutzer ist gesperrt und kann sich nicht mehr anmelden.
Hauptablauf	<ul style="list-style-type: none">• Das System bietet dem Admin die Funktion Nutzer zu sperren.• Der Admin nutzt die Funktion zum Sperren der User.• Das System sperrt den Nutzer.
Ausnahmen	
Anmerkung	

Anhang Tabelle 9 UC Nutzer sperren

2 Tests

Testfall	Student anlegen
Vorbedingung	Nutzer muss als Administrator oder Benutzer angemeldet sein.
Auszuführende Testschritte	<ol style="list-style-type: none"> 1. Auf der Studentenseite den „neuer Student“ Knopf drücken. 2. Alle nötigen Informationen zu dem Studenten eintragen. 3. Den Speichern Knopf betätigen
Erwartetes Ergebnis	Der Nutzer wird auf die Studentenseite weitergeleitet und der neue Student befindet sich in der Tabelle
Verwendete Testdaten	<ul style="list-style-type: none"> • Student Number: 007 • First Name: James • Last Name: Bond • Mail: JamesBond@SecretService.co.uk • Phone: 007007007007 • Date of Birth: 07.07.1997 • Zip Code 00700 • Town: London • Street: Secret • Number: 007
Ergebnis	Ein neuer Student mit den gegebenen Daten ist im System hinterlegt.
Zeitpunkt des Tests	03.06.2018

Anhang Tabelle 10 Testfall: Student anlegen

Testfall	Student bearbeiten
Vorbedingung	Testfall „Student anlegen“ muss ausgeführt worden sein. Nutzer muss als Administrator oder Benutzer angemeldet sein.
Auszuführende Testschritte	<ol style="list-style-type: none"> 1. Bei dem Studenten James Bond auf den „Ändern“ Knopf drücken. 2. Alle Informationen, die geändert werden müssen korrigieren. 3. Den Speichern Knopf betätigen
Erwartetes Ergebnis	Der Nutzer wird auf die Studentenseite weitergeleitet und der Student befindet sich in der Tabelle mit den neuen Daten.
Verwendete Testdaten	<ul style="list-style-type: none"> • Town: Secret
Ergebnis	James Bond wohn jetzt nicht mehr in London.
Zeitpunkt des Tests	03.06.2018

Anhang Tabelle 11 Testfall: Student bearbeiten

Testfall	Student löschen
Vorbedingung	Testfall „Student anlegen“ muss ausgeführt worden sein. Nutzer muss als Administrator oder Benutzer angemeldet sein.
Auszuführende Testschritte	<ol style="list-style-type: none"> 1. Bei dem Studenten James Bond auf das Kästchen in der Spalte „Markieren“ drücken. 2. Den „löschen“ Knopf drücken
Erwartetes Ergebnis	Der Nutzer wird auf die Studentenseite weitergeleitet und der Student James Bond erscheint nicht mehr in der Tabelle.
Ergebnis	James Bond erscheint nicht mehr in der Tabelle
Zeitpunkt des Tests	03.06.2018

Anhang Tabelle 12 Testfall: Student löschen

Testfall	Student suchen
Vorbedingung	Nutzer muss als Administrator oder Benutzer angemeldet sein. Es muss mindestens mehr als ein Student vorhanden sein. Testfall „Student anlegen“ muss ausgeführt worden sein.
Auszuführende Testschritte	1. Auf der Studentenseite etwas in die Suchleiste schreiben
Erwartetes Ergebnis	Es wird eine kleinere Auswahl angezeigt die Bond im Namen haben
Verwendete Testdaten	<ul style="list-style-type: none"> Suchleiste: Bond
Ergebnis	James Bond wird angezeigt
Zeitpunkt des Tests	03.06.2018

Anhang Tabelle 13 Testfall: Student suchen

Testfall	Kurs anlegen
Vorbedingung	Nutzer muss als Administrator oder Benutzer angemeldet sein.
Auszuführende Testschritte	<ol style="list-style-type: none"> 1. Nutzer drückt auf der Kursseite auf den „Neuer Kurs“ Knopf. 2. Der Nutzer gibt die geforderten Daten ein. 3. Der Nutzer drückt speichern.
Erwartetes Ergebnis	Es erscheint ein neuer Kurs mit den eingegebenen Daten.
Verwendete Testdaten	<ul style="list-style-type: none"> • Name: Wirtschaftsrecht 1 • Dozent: Christian Pelke • Beschreibung: Jeder ist mal Verbrecher
Ergebnis	Es erschien ein neuer Kurs mit den gegebenen Einträgen.
Zeitpunkt des Tests	03.06.2018

Anhang Tabelle 14 Testfall: Kurs anlegen

Testfall	Kurs bearbeiten
Vorbedingung	Nutzer muss als Administrator oder Benutzer angemeldet sein. Testfall „Kurs anlegen“ muss ausgeführt worden sein.
Auszuführende Testschritte	<ol style="list-style-type: none"> 1. Nutzer drückt auf der Kursseite, bei einem der Kurse, auf das Notizblocksymbol. 2. Der Nutzer ändert die geforderten Daten. 3. Der Nutzer drückt speichern.
Erwartetes Ergebnis	Der Kurs erscheint mit den geänderten Daten
Verwendete Testdaten	<ul style="list-style-type: none"> • Dozent: Dr. iur. Christian Pelke
Ergebnis	Es erschien ein neuer Kurs mit den gegebenen Einträgen.
Zeitpunkt des Tests	03.06.2018

Anhang Tabelle 15 Testfall: Kurs bearbeiten

Testfall	Kurs löschen
Vorbedingung	Nutzer muss als Administrator oder Benutzer angemeldet sein.
Auszuführende Testschritte	<ol style="list-style-type: none"> 1. Nutzer drückt auf der Kursseite, bei einem der Kurse, auf das Mülleimersymbol.
Erwartetes Ergebnis	Der Kurs, bei dem das Symbol gedrückt wurde ist nicht mehr in der Liste.
Ergebnis	Der Kurs ist nicht mehr in der Liste vorhanden.
Zeitpunkt des Tests	03.06.2018

Anhang Tabelle 16 Testfall: Kurs löschen

Testfall	Kurs einem Studenten zuordnen
Vorbedingung	
Auszuführende Testschritte	<ol style="list-style-type: none"> 1. Ändern-button anklicken 2. Kurs auswählen 3. Kurs hinzufügen-button betätigen 4. Speichern-button betätigen
Erwartetes Ergebnis	Kurs wird dem Studenten hinzugefügt
Verwendete Testdaten	<ul style="list-style-type: none"> • Student Patrick Star • Kurs Wirtschaftsrecht
Ergebnis	Kurs wurde dem Studenten hinzugefügt
Zeitpunkt des Tests	03.06.2018

Anhang Tabelle 17 Testfall: Kurs einem Studenten zuordnen

Testfall	Nutzer registrieren
Vorbedingung	Nutzer muss als Administrator angemeldet sein.
Auszuführende Testschritte	<ol style="list-style-type: none"> 1. Der Administrator geht auf Userseite. 2. Der Administrator drückt auf den „Neuer User“ Knopf. 3. Der Administrator gibt die geforderten Daten ein.
Erwartetes Ergebnis	Es wird ein neuer Benutzer hinzugefügt mit den eingegebenen Daten.
Verwendete Testdaten	<ul style="list-style-type: none"> • Username: Philipp • Passwort: Philipp • Rolle: Admin
Ergebnis	Es wird ein neuer Administrator angezeigt mit den eingegebenen Daten.
Zeitpunkt des Tests	03.06.2018

Anhang Tabelle 18 Testfall: Nutzer registrieren

Testfall	Nutzer sperren
Vorbedingung	Nutzer muss als Administrator angemeldet sein.
Auszuführende Testschritte	<ol style="list-style-type: none">1. Der Administrator geht auf Userseite.2. Der Administrator drückt auf den „sperren“ Knopf neben einem Benutzer, der noch nicht gesperrt ist.3. Der Administrator meldet sich ab und versucht sich als der gesperrte Nutzer anzumelden.
Erwartetes Ergebnis	Der gesperrte Benutzer kann sich nicht mehr anmelden.
Ergebnis	Man kann sich nicht mit dem gesperrten Nutzer anmelden.
Zeitpunkt des Tests	03.06.2018

Anhang Tabelle 19 Testfall: Nutzer sperren

3 CD

Inhalt:

- Javadocs
- Dokumentation PDF
- netzplan png
- Installationshandbuch PDF
- Benutzerhandbuch PDF
- Wireframes png
- ERM png