

**Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering**

Prozessoptimierung einer Anmeldeprozedur für Lehrveranstaltungen

Process optimization of a registration platform for courses

Bachelorarbeit

im Studiengang Informatik

von

Lena Pape

**Prüfer: Herr Prof. Dr. Kurt Schneider
Zweitprüfer: Frau Dr. Jil Klünder
Betreuer: M. Sc. Wasja Brunotte**

Hannover, 02.08.2019

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 02.08.2019

Lena Pape

Zusammenfassung

Die vorliegende Arbeit mit dem Titel "Prozessoptimierung einer Anmeldeprozedur für Lehrveranstaltungen" befasst sich mit der Optimierung des Anmeldeprozesses zur Lehrveranstaltung "Softwareprojekt". Diese wird vom Fachgebiet Software Engineering (SE) an der Leibniz Universität Hannover jährlich angeboten.

Die Anmeldung der Studierenden zur Lehrveranstaltung erfolgt online.

Der bisherige Prozess der Verwaltung durch Administrierenden und die Anmeldung von Studierenden, wird in dieser Arbeit zunächst beschrieben und mögliche Optimierungsvorschläge diskutiert. Zudem werden Anforderungen an ein neues System erhoben, dokumentiert und priorisiert.

Im Hauptteil wird ein entwickeltes Konzept vorgestellt, welches das Ziel hat, die erhobenen Anforderungen umzusetzen und den Prozess zu optimieren. Fokus dabei liegt auf der Verringerung des Arbeits- und Zeitaufwandes der bei der Verwaltung des Systems entsteht.

Anschließend erfolgt eine Vorstellung des in dieser Arbeit entwickelten Prototyps. Eine Abgrenzung zu verwandten Arbeiten schließt den Hauptteil ab.

Weitere Aussichten und sowie Optimierungen, die an dem Prototypen vorgenommen werden können, werden abschließend präsentiert.

Abstract

Process optimization of a registration platform for courses

This bachelor thesis named “Process optimization of a registration platform for courses“ is on the subject of optimizing an online registration process for the course “Softwareprojekt“, offered by the Software Engineering Group (SE) of the Leibniz University Hannover.

The registration for the course is carried out online. A previously used tool maintained and administrated by the employees of the SE is supposed to be optimized in this bachelor thesis.

First of all the whole process of maintaining and operating of the system is described and different possible ways of optimizing are discussed. Following requirements on the new system, that are risen, documented and prioritized. A new concept with the objective to realize the given requirements and optimizations is presented in the main body. The focus of the concept lays on reducing the time and effort for managing the system. After this the developed prototype is showcased. Finishing the main body is a differentiation between this thesis and related works.

Lastly a future perspective and further optimization on the prototype is given.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problemstellung	1
1.2	Zielsetzung	2
1.3	Lösungsansatz	2
1.4	Struktur der Arbeit	3
2	Grundlagen	4
2.1	Entity-Relationship-Modell	4
2.2	XML	5
2.3	Requirements Engineering	6
2.3.1	Stakeholder	6
3	Prozessoptimierung	7
3.1	Prozessbeschreibung	7
3.1.1	Administrative Vorbereitung der Anmeldung	7
3.1.2	Registrierung eines Studierenden	7
3.1.3	Änderung eingegebener Daten	8
3.1.4	Beobachtung der Anmeldung	9
3.1.5	Beenden der Anmeldung	9
3.2	Prozessoptimierung	10
3.2.1	Eingabe der Projektdaten	10
3.2.2	Hinzufügen neuer Optionen in der Anmeldung	10
3.2.3	Entfernung von Daten aus der Datenbank	10
3.2.4	Export von Daten im XML-Format	11
3.2.5	Verifizierung Studierender	11
4	Anforderungen	15
4.1	Stakeholder	15
4.2	Anforderungen an das System	15
4.3	Priorisierung der Anforderungen	18
5	Konzept	19
5.1	Datenbankentwurf	19
5.2	Import und Export	21

5.3	Design	21
5.3.1	MockUps	25
6	Implementierung	27
6.1	Architektur	27
6.2	Umsetzung des Designkonzeptes	27
6.2.1	Administrationsoberfläche	28
6.2.2	Studierendenoberfläche	32
6.3	Qualitätssicherung bei der Implementierung	34
6.4	Herausforderung bei der Implementierung	35
7	Verwandte Arbeiten	36
8	Zusammenfassung und Ausblick	40
8.1	Zusammenfassung	40
8.2	Ausblick	42
A	Anhang	43

Kapitel 1

Einleitung

“Für Hochschulen bietet die Digitalisierung erhebliche Potenziale zur Profilbildung, Steigerung der Reputation, Qualitätsverbesserung, Studierendengewinnung sowie dem langfristigen Aufbau von Beziehungen zu den Zielgruppen, jedoch auch zahlreiche Herausforderungen wie die Entwicklung der notwendigen Kompetenzen, Steigerung der Akzeptanz der digitalen Bildungsangebote sowie die Heterogenität und Diversität der potenziellen Nutzer.“ ([3]). Nutzung von digitalen Tools kann also zu einer Verbesserung der Lehre an Hochschulen beitragen, bringt aber viele Herausforderungen in der Entwicklung mit sich. Die Optimierung der Bildungsverwaltung und Bildungsinfrastruktur stellt hierbei ein Hauptproblem dar, mit welcher sich die Entwicklung von E-Learningsystemen beschäftigen muss [2].

Das Fachgebiet Software Engineering (kurz: SE) der Leibniz Universität Hannover bietet jährlich eine Lehrveranstaltung namens *Softwareprojekt* an. Studierende sollen in dieser Lehrveranstaltung in Gruppen von ca. 8 Personen eine Software selbstständig in einem Zeitraum von einem Semester implementieren. Um die Studierenden auf die Projekte in ausgeglichene Gruppen verteilen zu können, erfolgt die Anmeldung zur Lehrveranstaltung online über ein Anmeldetool.

1.1 Problemstellung

Der Verwaltungsaufwand des Onlinetools für die Anmeldung zur Lehrveranstaltung soll optimiert werden. Da sich jährlich die Projekte, zugehörige Programmiersprachen und auch Lehrveranstaltungen sich verändern können, muss das Anmeldetool regelmäßig aktualisiert werden, um eine angepasste Anmeldung zu ermöglichen. Dies ist im aktuell genutzten System nicht über eine Benutzeroberfläche, sondern nur über die Modifizierung des Codes möglich.

Studierende müssen bei der Anmeldung neben ihren persönlichen Daten eine Einschätzung ihrer Kenntnisse in verschiedenen Programmiersprachen und Softskills abgeben, aber auch Angaben zu bestandenen Lehrveranstaltungen tätigen. Diese Informationen sind sowohl relevant für die Zulassung zur Lehrveranstaltung als auch für die Gruppenzuteilung von Bedeutung. Dabei findet im bisher genutztem System keine Überprüfung der Pflichtangaben statt, so dass fehlerhafte Anmeldungen aber auch Mehrfachanmeldungen möglich sind.

1.2 Zielsetzung

Der Prozess der Verwaltung der Onlineanmeldung soll im Rahmen dieser Bachelorarbeit optimiert werden und eine Entwicklung eines neuen Prototypen für das Anmeldetool erfolgen.

Ziel soll es hierbei sein, den Prozess, sowohl für die Verwaltung des Anmeldetools auf der Seite der Administrierenden, als auch die Anmeldung auf der Seite der Studierenden zu verbessern. Im Vordergrund steht hierbei die Verringerung des Arbeits- und Zeitaufwandes für das Verwalten der Anmeldung.

1.3 Lösungsansatz

Die Umsetzung der Optimierungen sollen in einem Prototypen erfolgen. Dieser soll als Webapplikation umgesetzt werden. Eine Administrationsoberfläche ermöglicht die Datenbankverwaltung und soll das Hinzufügen, Ändern und Entfernen von Elementen in der Anmeldung über eine Oberfläche ermöglichen. Dadurch soll der Verwaltungsaufwand verringert werden. Import und Exportoptionen im XML-Format sollen hierbei weitere Arbeitsschritte ersparen, da Informationen importiert statt von einem Mitarbeitendem des SE eingegeben werden.

Angaben welche Studierende bei der Anmeldung tätigen, sollen zudem vom Prototypen validiert werden, um fehlerhafte Anmeldungen zu verhindern. Außerdem soll die Anmeldeoberfläche für mobile Geräte angepasst werden.

1.4 Struktur der Arbeit

Im ersten Teil dieser Arbeit erfolgt eine Beschreibung des bisherigen Prozesses, auf der anschließend eine Optimierung durchgeführt wird, wobei Verbesserungsmöglichkeiten diskutiert werden (Kapitel 3 *Prozessoptimierung*).

Basierend auf der Prozessoptimierung werden Anforderungen an den Prototypen in Kapitel 4 *Anforderungen* dokumentiert. Unter Berücksichtigung dieser, wird ein Konzept entwickelt, welches in Kapitel 5 erläutert wird. Die konkrete Umsetzung des Prototyps wird anschließend in Kapitel 6 *Implementierung* vorgestellt.

Abschließend erfolgt eine Abgrenzung zu verwandten Arbeiten in Kapitel 7 *Verwandte Arbeiten* und eine Zusammenfassung der Ergebnisse in Kapitel 8 *Zusammenfassung*.

Kapitel 2

Grundlagen

2.1 Entity-Relationship-Modell

Nach Edwin Schicker "Datenbanken und SQL -Eine praxisorientierte Einführung mit Anwendungen in Oracle, SQL Server und MySQL" [12] ist das Entity-Relationship-Modell (kurz: ER-Modell) im nachfolgendem Abschnitt wie folgt definiert:

Das ER-Modell behandelt die Modellierung zwischen Entitäts (*deutsch: Entitäten*) durch Relationships (*deutsch: Beziehungen*). Als Entitäten werden alle identifizierbaren Objekte bezeichnet, welche wiederum *Eigenschaften* besitzen können. Im ER-Modell werden Entitäten als Rechtecke dargestellt. Die Eigenschaften werden an der entsprechende Entität verbunden und mit Ellipsen umfasst.

Entitäten werden in Beziehungen zueinander gesetzt.

So gibt es verschiedene Arten von Beziehungen die Entitäten untereinander besitzen können: Die *1 zu 1 Beziehung*, *m zu 1 Beziehung* und *m zu n Beziehung*.

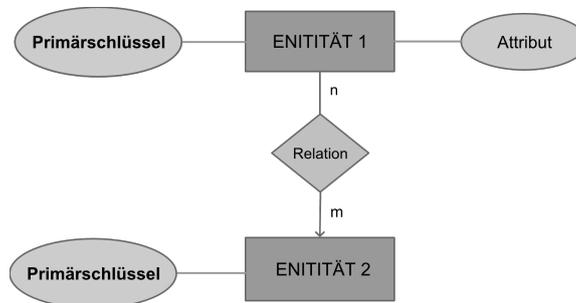
Konkreter bedeutet die 1 zu 1 Beziehung, dass zwischen zwei Entitäten jeder Eintrag einer Entität mit jeweils einem Eintrag der anderen Entität verbunden ist.

Bei m zu 1 Beziehungen besitzt ein Eintrag von einer Entität mit mehreren Einträgen in der anderen Entität verknüpft sein, allerdings nicht umgekehrt. M zu n Beziehungen bedeuten, dass beide Entitäten mit verschieden vielen Einträgen der jeweils anderen in Verbindung stehen.

Im ER-Modell werden die Entitäten mit Linien verbunden, dazwischen eine Raute die die Beziehung darstellt.

Folgende Abbildung 2.1 zeigt die Umsetzung im ER-Modell:

Abbildung 2.1: Beispiel ER-Modell



2.2 XML

In folgendem Abschnitt wird die Extensible Markup Language *XML* nach [1] definiert: XML-Dokumente werden unter anderem genutzt um Daten zu sichern und auszutauschen. Sie können plattformunabhängig eingesetzt werden und werden von verschiedensten Anwendungen unterstützt. Der strukturierte Aufbau der Dokumente basiert auf internationalen Standards und macht die Dokumente sowohl für den Menschen als auch maschinell auslesbar. Dies ist ein bedeutender Vorteil von XML.

Der Aufbau von XML-Dokumenten wird nach [6] beschrieben: XML-Dokumente bestehen aus Elementen, welche wiederum weitere Elemente in sich tragen. Ein Element fängt immer mit einem Start-Tag an und endet immer mit einem End-Tag, solange es kein Empty-Element-Tag besitzt. Wenn Element B sich in Element A öffnet, muss zuerst das Element B geschlossen bevor das Element A geschlossen werden darf. Ein Beispielexport ist folgendem Codeabschnitt zu sehen 2.1:

Listing 2.1: Beispielexport

```

<?xml version="1.0" encoding="UTF-8"?>
<Root>
  <ElementA>
    <ElementB attribut="1">
      <ElementC attribut="4">
        <ElementD>Beispiel1</ElementD>
      </ElementC>
      <ElementC attribut="2">
        <ElementD>Beispiel2</ElementD>
      </ElementC>
    </ElementB>
  </ElementA>
</Root>
  
```

Hier wird gezeigt, dass ein Element mehrere Elemente in sich tragen kann, aber zu erst die inneren Elemente geschlossen werden müssen, bevor sich das oberste Element schließen darf.

2.3 Requirements Engineering

Nach Pohl und Rupp wird Requirements Engineering im folgendem Absatz definiert [9]:

Requirements Engineering ist ein systematischer und disziplinierter Ansatz zur Spezifikation und zum Management von Anforderungen mit den folgenden Zielen:

1. Die relevanten Anforderungen zu kennen, Konsens unter den Stakeholdern über die Anforderungen herzustellen, die Anforderungen konform zu vorgegebenen Standards zu dokumentieren und die Anforderungen systematisch zu managen.
2. Die Wünsche und Bedürfnisse der Stakeholder zu verstehen und zu dokumentieren.
3. Die Anforderungen zu spezifizieren und zu managen, um das Risiko zu minimieren, dass das System nicht den Wünschen und Bedürfnissen der Stakeholder entspricht.

Wichtiger Bestandteil des Requirements Engineering stellen also die Stakeholder dar, da sie als direkte oder indirekte Informationslieferanten für Anforderungen gelten [7].

2.3.1 Stakeholder

Stakeholder sind also Personen oder Organisationen mit verschiedenen fachlichen Hintergründen, welche unterschiedliche Interessen an einem geplantem System haben und in der Regel auch die Anforderungen an das System stellen [8]. Dies können sowohl Mitwirkende an der Entwicklung des Systems sein, aber auch Personen und Organisationen welche von der Nutzung des Systems betroffen sind [7].

Eine zusätzliche Unterscheidung von Stakeholdern in internal und external Stakeholder kann getroffen werden:

Unter internal Stakeholder werden alle Personen verstanden, die Teil der Organisation sind und so einen direkten Einfluss auf die Entwicklung haben, wie die Mitarbeitenden, Besitzenden, das Management etc. der Organisation [14]. External Stakeholder sind von den Aktivitäten der Organisation betroffen, sind aber kein direkter Teil dieser, wie Kunden, Investoren, die Gesellschaft uvm.[14].

Kapitel 3

Prozessoptimierung

Im folgendem Kapitel wird zunächst der bisherige Prozessablauf beschrieben, um diesen analysieren und optimieren zu können. Im zweiten Kapitel erfolgt eine Optimierung des Prozesses, in der mögliche Verbesserungswege diskutiert werden.

3.1 Prozessbeschreibung

3.1.1 Administrative Vorbereitung der Anmeldung

Das Softwareprojekt findet jährlich im Wintersemester statt. Als Vorbereitung wird von den zuständigen Mitarbeitenden der SE eine Auswahl an Projekten, die angeboten werden, beschlossen. Ein Projekt besitzt einen *Namen*, ein passendes *Kürzel* und den Namen des *Kunden*, der die Software anfordert. Diese Daten müssen aufgrund der Struktur des Altsystems mehrmals in verschiedenen Codedateien eingetragen werden.

Die Kunden und ihre angebotenen Projekte wechseln jedes Jahr. Somit ändern sich z.B. auch die zum Einsatz kommenden Programmiersprachen, wodurch die Mitarbeitende die bisherigen Auswahloptionen in der Anmeldung an die neuen Projekte anpassen.

Nach dem die Anmeldung auf die neuen Projekte angepasst wurde, wird das Programm kompiliert. Sobald der Anmeldezeitraum für die Lehrveranstaltung startet ist die Webseite unter "swp.se.uni-hannover.de" zu erreichen.

3.1.2 Registrierung eines Studierenden

Als ersten Schritt nach der Erstellung eines Kontos, müssen Studierende Angaben zu ihrer Person tätigen. Hierfür werden *Vor-* und *Nachname*, *Matrikelnummer* und *Studiengang* abgefragt. Diese Eingaben dienen zur Identifikation des Studierenden. Zusätzlich kann die Option getroffen werden, ob die Person einen schriftlichen Prüfungsnachweis benötigt.

Diese Option ist für die Studierende relevant, für die die Lehrveranstaltung eine Wahl- und keine Pflichtveranstaltung ist.

Anschließend wird der Studierende auf eine neue Seite weitergeleitet: Hier soll angegeben werden, welche der aufgelisteten Lehrveranstaltung der Studierende besucht und bestanden hat, da die Teilnahme am Softwareprojekt den erfolgreichen Abschluss von einigen Lehrveranstaltungen voraussetzt. So wird vorausgesetzt, dass der Studierende eine Lehrveranstaltung, in welcher Grundlagen des Programmierens und der Softwaretechnik oder der Softwarequalität gelehrt wurden, bestanden hat. In der Anmeldung selber wird keine Überprüfung dessen vorgenommen, so ist es möglich keiner der Veranstaltungen anzukreuzen, da diese keine Pflichtfelder sind.

Der nächste Schritt der Anmeldung beinhaltet das Bewerten der eigenen *Programmierkenntnisse* zu den aufgeführten Programmiersprachen. Dabei wird für unterschiedliche Sprachen eine Einschätzung der Kenntnis zwischen *keine*, *wenig*, *mittel* bis *hohe* Kenntnis getroffen. Falls der Studierende noch weitere Kenntnisse einer nicht aufgeführten Programmiersprache besitzt, kann er diese unter "Sonstige" angeben. Jedes Feld, bis auf das Feld "Sonstige", ist ein Pflichtfeld.

Die Kenntnisse über Programmiersprachen ist entscheidend für die spätere Aufteilung der Teams auf die Projekte, da einige Projekte nur mit einer bestimmten Programmiersprache entwickelt werden. Hierfür ist es unter anderem wichtig, dass genug Teammitglieder Kenntnisse in der entsprechenden Programmiersprache besitzen.

Neben der Abfrage der Programmierkenntnisse, wird noch eine Selbsteinschätzung der eigenen *Führungsqualitäten*, *Projekterfahrungen* und *Programmiererfahrung* durchgeführt. Auch diese Daten werden später für die Aufteilung der Teams genutzt.

Als vorletzten Schritt der Anmeldung wird die Priorisierung der zur auswahlstehenden Projekte vorgenommen. Dabei wird zwischen *Erst-*, *Zweit-* und *Drittwunsch* unterschieden. Projekte müssen priorisiert werden und dürfen auch nicht doppelt gewählt werden.

Abschließend erhält der Studierende eine Übersicht über seine eingegebenen Daten. Diese kann er bei Bedarf bearbeiten. Nach dem klicken auf *Absenden*, ist die Registrierung abgeschlossen.

3.1.3 Änderung eingegebener Daten

Nach dem der Studierende ein Konto angelegt hat, kann sich dieser über seine E-Mail Adresse und sein zuvor gewähltes Passwort in sein bestehendes Konto einloggen. Sollte der Studierende seine Daten bereits abgesendet haben, erfolgt nach dem Login eine Übersicht seiner Angaben und kann diese nach Bedarf ändern. Sollte der Studierende die Registrierung noch nicht abgeschlossen haben, muss er diese zunächst abschließen.

3.1.4 Beobachtung der Anmeldung

Während des Anmeldezeitraums können Mitarbeitende des Fachgebietes auf eine in Tabellenform angelegte Übersicht zugreifen. Diese umfasst die Namen, ihre E-Mail Adresse und eine persönliche ID der bereits angemeldeten Studenten.

Zusätzlich wird die Verteilung der Projekte auf den Erst-, Zweit- und Drittwunsch aller angemeldeten Studierenden als Kreisdiagramm dargestellt. Außerdem gibt es eine Tabelle in der alle Projekte aufgeführt sind, inklusive der Anzahl der Studierende die dieses Projekt als Erst-, Zweit- oder Drittwunsch angegeben haben.

3.1.5 Beenden der Anmeldung

Die Mitarbeitenden der SE schließen nach Ablauf des Anmeldezeitraums die Anmeldung. Ab diesem Zeitpunkt ist es nicht mehr möglich auf das Anmeldeportal zuzugreifen. Folglich ist es als Studierender nicht mehr möglich, sich zu registrieren oder sich in seinen Account einzuloggen und Änderung seiner Daten vorzunehmen. Die Mitarbeitenden erhalten in der Administrationsansicht weiterhin eine Übersicht, wie in 3.1.4 *Beobachtung der Anmeldung* beschrieben wurde.

Nachfolgend wird von den Mitarbeitenden geprüft, ob die Studierenden korrekte Angaben gemacht haben und die Anforderungen erfüllen. Nach dem Studierende entfernt worden sind, die nicht für die Lehrveranstaltung zugelassen werden, erfolgt ein Export der Daten im CSV-Format. Mit den exportierten Daten werden die Studierende auf die Projekte zugeteilt. Dies erfolgt über interne Software des Institutes und gehört nicht mehr zu den Aufgaben des Anmeldetools.

Das Leeren der Daten aus der Datenbank ist aktuell noch nicht über die Oberfläche möglich.

3.2 Prozessoptimierung

In diesem Abschnitt werden die in Abschnitt 3.1 Prozessbeschreibung genannten Probleme zusammengefasst und mögliche Lösungswege diskutiert.

3.2.1 Eingabe der Projektdaten

Die Eingabe neuer Projektdaten erfolgt, indem der Code des bestehenden System modifiziert wird. Dies muss an mehreren Stellen im Code geschehen. Anschließend muss das System erneut kompiliert werden. Um dieses zu vereinfachen, soll ein Import einer XML-Datei erfolgen, der die einzutragenden Daten ausliest und in einer Datenbank hinterlegt.

3.2.2 Hinzufügen neuer Optionen in der Anmeldung

Aktuell lassen sich bestehende Optionen in der Anmeldung, wie Programmiersprachen, Lehrveranstaltungen oder Softskills, nur sehr aufwendig ändern oder entfernen. Das Hinzufügen neuer Optionen ist nicht möglich und durch den stetigen Wechsel der Anforderungen an die Projekte bringt dies die Problematik mit sich, dass abgewogen werden muss, welche Option geändert wird, um eine Neue hinzuzufügen. Dies soll über eine neue Oberfläche der Webseite optimiert werden. So sollen über eine Administrationsoberfläche neue Optionen beliebig oft angelegt, bearbeitet und auch entfernt werden, um so die Anmeldung vor ihrer Freischaltung anpassen zu können.

3.2.3 Entfernung von Daten aus der Datenbank

Um Daten aus der Datenbank zu entfernen, muss sich ein Mitarbeitender des SE über einen SSH Client mit der Datenbank verbinden und über die Konsole manuell Einträge aus der Datenbank entfernen.

Die Entfernung der Daten einzelner Studierender soll nun über die Administrationsoberfläche erfolgen. Hier soll in der tabellarischen Auflistung aller Studierenden, hinter jedem Studierenden, die Option zur Verfügung stehen diesen zu entfernen. Zusätzlich sollte es auch möglich sein, alle Daten der Studierenden eines Jahres mit einem Klick zu entfernen. Dies kann mithilfe eines Attributes in der Datenbank, in der das Jahr in dem sich der Studierende anmeldet gespeichert wird, erfolgen. So können anhand der Jahreszahl alle Studierenden aus der Datenbank entfernt werden, die im Vorjahr teilgenommen haben.

3.2.4 Export von Daten im XML-Format

Der Export der Daten ist besonders für die interne Weiterverarbeitung der Daten wichtig, damit eine Projektzuteilung erfolgen kann. Aktuell erfolgt der Export im CSV-Format. Da die Daten von einer intern verwendeten Anwendung ausgelesen und verarbeitet wird, würde sich hier der Export in ein XML-Format besser eignen:

XML-Dokumente sind maschinell auslesbar, wodurch viele Programmiersprachen Bibliotheken, die XML-Dokumente seriell auslesen und verarbeiten können, besitzen. CSV-Dateien wiederum müssen zuerst überprüft werden, um die Formatierung der Daten und Trennzeichen herauszufinden. Das XML-Format hat eine standardisierte Formatierungen und würde so den Schritt des Auslesens einsparen.

3.2.5 Verifizierung Studierender

Bei den Anmeldungen der Studierenden wird bislang nicht überprüft, ob diese von realen, immatrikulierten Studierenden sind. Bislang gab es Anmeldungen von Studierenden, die sich zwar angemeldet haben, aber nicht an der Lehrveranstaltung teilnehmen wollten, zudem gab es Anmeldungen die keinem Studierenden zugewiesen werden konnten oder auch Studierende, die sich mehrfach angemeldet haben.

Da Angaben dieser Anmeldungen teils so realistisch waren, wurden diese nicht als falsche Anmeldung erkannt und die Daten wurden weiterverarbeitet. Dies führte dazu, dass Fake-User zu Projekten zugeordnet worden sind. Dadurch könnten Plätze von falschen Anmeldungen in Projekten belegt werden, welche z.B. von anderen Studierenden als Erstwunsch angegeben worden sind und diese ihren Erstwunsch dadurch nicht bekommen haben.

Die Entfernung dieser Fake-User erfolgt aktuell von den Mitarbeitenden des Fachgebietes und bedeutet zusätzlichen Arbeitsaufwand und kann im Nachhinein dazu führen, dass einige Gruppen zu wenig Mitglieder haben. Eine ungefähr gleiche Anzahl an Gruppenmitgliedern in allen Projekten ist erwünscht und wichtig für die erfolgreiche Durchführung des Projektes. Im schlechtesten Fall kommt es zu einer Neuverteilung, welches wiederum zu einem Mehraufwand führen würde.

Um diese Problematik zu lösen, könnten folgende Optionen in Betracht gezogen werden:

- i. **Anmeldung über eine verifizierte E-Mail Adresse der Leibniz Universität:** Die erste Möglichkeit wäre die Anmeldung nur über eine verifizierte E-Mail Adresse der Leibniz Universität zu ermöglichen. Studierende können sich einen E-Mail Account über den Leibniz Universität IT Service (kurz: LUIS) einrichten. Sie erhalten bei Immatrikulation einen Account über den sie über verschiedene Onlinedienste der Universität verwalten können. So kann sich jeder

Studierende der Leibniz Universität eine verifizierte studentische E-Mail Adresse anlegen, die den Vor- und Nachnamen des Studierenden als lokalen Teil und *stud.uni-hannover.de* als Domänenteil beinhaltet. So könnte die zur Anmeldung genutzte E-Mail Adresse nach dem Domänenteil¹ zu überprüft werden, ob dieser mit dem der für E-Mail Adressen der Universität vergebenen Domänenteil übereinstimmt. Als nächstes müsste noch überprüft werden, ob die angegebene E-Mail Adresse wirklich eine aktive E-Mail Adresse ist. Ansonsten ist es möglich mit dem Wissen, wie die E-Mail Adresse zur erfolgreichen Anmeldung aufgebaut werden muss, eine beliebige Adresse mit dem richtigen Domänenteil anzugeben.

Um dies zu verhindern, muss eine Bestätigungs-E-Mail an die angegebene E-Mail Adresse versandt und vom Studierenden bestätigt werden, um die Anmeldung freizuschalten.

Nachteil dieser Methode liegt darin, dass viele Studierende die universitätsinterne E-Mail Adresse nicht aktiv nutzen oder überhaupt nicht eingerichtet haben. Hier würde für die Studierende ein höherer Aufwand im Prozess entstehen, da diese sich vor der Anmeldung einen E-Mail Account an der Universität anlegen müssen und dies auch Wartezeit mit sich bringt, bis der Account freigeschaltet wird.

Zudem war es in der Vergangenheit möglich, die E-Mails welche an den universitären E-Mail Account versandt werden, auf eine andere private Mail Adresse weiterzuleiten. Diese Funktion wurde aber entfernt, mit der Begründung, dass zu viele Provider die weitergeleiteten E-Mails als Spam melden.²

Durch die wenige Nutzung des Universitäts-E-Mail Accounts auf Seite der Studierenden und der nicht vorhandenen Weiterleitung auf eine private, mehrgenutzte E-Mail Adresse kann es dazu kommen, dass Studierende schlecht über die universitäre E-Mail Adresse erreichbar sind. Die Kommunikation zwischen dem Fachgebiet und den Studierenden erfolgt während des Projektes hauptsächlich über E-Mail, weswegen eine gültige und gut erreichbare E-Mail Adresse nötig ist. Zusätzlich werden die angegebenen E-Mail Adressen auch später für einen Mail-Verteiler für die interne Kommunikation in den Projekten genutzt. Werden hier nicht gut erreichbare oder gar ungültige E-Mail Adressen genutzt, könnte dies dazu führen, dass wenig Kommunikation über diese Verteiler und dadurch unter den Studierenden entsteht.

¹Domänen- und Lokalteil nach RFC822 "Standard for the format of arpa internet text messages" definiert: <https://tools.ietf.org/html/rfc822section-4.1>

²<https://www.luis.uni-hannover.de/email.html>

ii. **Hochladen einer Immatrikulationsbescheinigung:** An der Leibniz Universität Hannover wird das Onlineportal namens QIS³ genutzt, für welches jeder Studierende einen Account bei seiner Immatrikulation bekommt. Über dieses Portal können sich Studierende z.B. für Prüfungen anmelden, sofern eine Online-Anmeldung für ihren Studiengang freigeschaltet worden ist, aber auch das Herunterladen von Bescheinigungen, wie eine aktuelle Immatrikulationsbescheinigung im PDF-Format, ist möglich.

Bei der heruntergeladenen Immatrikulationsbescheinigung handelt es sich um eine maschinell erstellte Bescheinigung. Diese ist ohne Unterschrift gültig und enthält eine Verifikationsnummer, welche auf der Webseite des QIS eingegeben und verifiziert werden kann.

Eine mögliche Option der eindeutigen Identifikation von Studierenden wäre es, dass jeder teilnehmende Studierende vor der Anmeldung seine aktuelle Immatrikulationsbescheinigung herunterlädt und bei der Anmeldung zum Softwareprojekt mit hochlädt. Es könnten so maschinell die Daten wie Name und Matrikelnummer aus der Immatrikulationsbescheinigung ausgelesen und für die Anmeldung genutzt werden. Zudem wäre es möglich, bei einer Anmeldung mithilfe der hinterlegten Immatrikulationsbescheinigung falsche Anmeldungen zu identifizieren oder zu verhindern.

Das Format dieser Immatrikulationsbescheinigung ist für alle Studierenden an Leibniz Universität bekannt, so wäre es möglich eine Immatrikulationsbescheinigung zu fälschen oder auf Falschangaben zu modifizieren. Da so die Integrität der PDF-Datei nicht gegeben ist, müsste jede einzelne hochgeladene Bescheinigung über die Verifikationsnummer überprüft werden. Eine Verifizierung jeder einzelnen Immatrikulationsbescheinigungen durch die Mitarbeitende des SE würde sich als sehr zeitintensiv herausstellen.

Eine maschinelle Verifizierung und Sicherstellung der Integrität der hochgeladenen Immatrikulationsbescheinigungen gestaltet sich als schwer umsetzbar, da der Zugriff auf die Verifizierungsmechanismen von der Universität aus ermöglicht sein muss.

Alle hochgeladenen Bescheinigungen müssen zu dem gespeichert werden und auch der Upload daraufhin gesichert werden, dass keine unerwünschten Dateien in das Anmeldesystem geladen werden können. Ein weiterer Nachteil dieser Methode ist ein Mehraufwand auf Seiten der Studierenden, da ein zusätzliches Anmelden im QIS, Herunterladen einer Immatrikulationsbescheinigung und erneutes Hochladen dieser im Anmeldetool von Nöten ist.

³Onlineportal QIS: <https://qis.verwaltung.uni-hannover.de>

iii. **Nutzung von WebSSO:** Eine weitere Option wäre die Anbindung an das bereits bestehende Identitätsmanagement (IdM) des LUIS der Leibniz Universität Hannover. Diese benutzen den Authentifizierungsdienst WebSSO (Web-Single-Sign-On)⁴, welches für verschiedene Anwendungen der Universität genutzt wird.

Es handelt sich dabei um einen Authentifizierungsdienst der mithilfe einer intern an Studierenden vergebene Kennung und einem Passwort den Studierenden eindeutig identifiziert. Durch die Nutzung dieses Dienstes, kann man Anmeldungen von Externen verhindern, so wie eine mehrfache Anmeldung eines einzelnen Studierenden nicht mehr möglich machen. Auch die Fälschung von Daten ist nicht mehr möglich und der Studierende ist eindeutig identifizierbar.

Da die Nutzung von WebSSO für viele Anwendungen der Universität verpflichtend ist, besitzt jeder Studierende eine Kennung und ein Passwort. Somit würde sich kein Mehraufwand für die Studierenden, im Vergleich zu den vorherigen Optionen (i) und (ii), ergeben.

Ausgehend von den Vor- und Nachteilen der drei Optionen, bietet die Nutzung von WebSSO, vor allem in dem Punkt Sicherheit vor gefälschten Angaben, die meisten Vorteile. Für die Studierende wäre diese Option mit dem geringsten Aufwand verbunden und bringt keinen zusätzlichen Arbeitsaufwand für die Mitarbeitenden des SE mit.

⁴ Identitätsmanagement und WebSSO: <https://www.luis.uni-hannover.de/idm.html>

Kapitel 4

Anforderungen

In diesem Kapitel werden die Anforderungen an das Anmeldetool dokumentiert. Anschließend werden die Stakeholder im nächsten Abschnitt erläutert. Darauf folgend werden die Anforderungen an das System spezifiziert und priorisiert.

4.1 Stakeholder

Die größte Gruppe der Stakeholder sind die Studierende, die sich über das Anmeldetool zum Softwareprojekt anmelden. Diese benutzen das Tool meist einmalig, um sich für die Lehrveranstaltung anzumelden. Ein weiterer Stakeholder sind die wissenschaftlichen Mitarbeitenden des Institutes Software Engineering der Leibniz Universität Hannover. Diese administrieren die Anmeldeprozedur für die Lehrveranstaltung *Softwareprojekt* und verwalten das Anmeldetool. Dieser Stakeholder ist der entscheidungstragende Stakeholder bei der Entwicklung und hat somit den größten Einfluss beider Stakeholder. Zudem besitzt dieser Stakeholder das größte Interesse an dem System, da dieses den Verwaltungsaufwand bei der Anmeldung verringern soll.

4.2 Anforderungen an das System

In diesem Abschnitt werden die Anforderungen an das Anmeldetool spezifiziert.

[R01] Ein Administrator muss einen neuen Administrator hinzufügen können.

Ein Administrator soll einen neuen Account für einen weiteren Administrator anlegen können.

[R02] Die Anwendung soll eine XML-Dokument importieren können und aus diesem Projektinformationen lesen und aktualisieren können.

Die Projektinformationen sollen per Import eines XML-Dokumentes ausgelesen werden und im System automatisch eingetragen werden.

[R03] In der Anwendung muss es möglich sein Optionen zur Anmeldung über die Oberfläche hinzuzufügen, zu ändern und zu entfernen.

Die Anmeldeoberfläche soll das Hinzufügen, Ändern und Entfernen von Optionen ermöglichen. Spezifisch handelt es sich dabei um die Änderungen der in der Anmeldung gezeigten Programmiersprachen, Lehrveranstaltungen, Softskills und die auswählbaren Projekte.

[R04] Ein Administrator kann die Anmeldung für die Studierenden freischalten.

Die Anmeldung darf nur für einen gegebenen Anmeldezeitraum erreichbar sein. Die Anmeldung muss von einem Administrator freigeschaltet und wieder beendet werden können.

[R05] Ein Administrator kann die Anmeldung für die Studierenden beenden.

Nach Ende des Anmeldezeitraums kann in der Administrationsansicht die Anmeldung geschlossen werden. Die Erstellung eines neuen Kontos oder das Einloggen und Ändern von Daten soll nun nicht mehr möglich sein.

[R06] Die Anwendung kann Studierende eindeutig identifizieren und authentifizieren.

Ein Studierender muss eindeutig identifiziert und authentifiziert werden.

[R07] Studierende können einen neuen Account in der Anwendung anlegen.

Studierende können einen Account mithilfe einer E-Mail Adresse und einem selbstgewähltem Passwort anlegen. Es ist verpflichtend ein Passwort anzugeben.

[R08] Die Anwendung validiert Eingabedaten von Studierenden.

Die eingegebenen Daten von Studierenden sollen validiert werden. So müssen Name, E-Mail Adresse, Studiengang, Matrikelnummer für jeden Studierenden verpflichtend ausgefüllt werden. Zu dem muss jeder Studierende genau drei, ungleiche Projekte favorisieren.

[R09] Der Studierende kann eine Übersicht über seine eingegebenen Daten erhalten.

Nach dem alle Angaben eingegeben worden sind, erhält der Studierende eine Übersicht all seiner eingegebenen Daten. Diese kann er nach Bedarf editieren.

[R10] Die Anwendung speichert die bei der Anmeldung eingegebenen Daten des Studierenden ab.

Nach dem der Studierende alle Daten angegeben hat, kann dieser die Anmeldung abschließen, wodurch seine Daten in der Datenbank abgespeichert werden.

[R11] Während das System aktiv geschaltet ist, kann sich ein Studierender mit einem bestehenden Konto im System einloggen.

Ein Einloggen ist während des Anmeldezeitraums, nach Abschluss der erfolgreichen Erstellung des Kontos ([R06]) mit der angegebenen E-Mail Adresse und Passwort, möglich. Hier erhält der Studierende nach Abschluss der Anmeldung nach [R09] *Absenden der Daten* eine Übersicht seiner eingegebenen Daten ([R08] Übersicht eingegebener Daten).

Wenn [R08] noch nicht abgeschlossen, gelangt der Studierende zu Beginn der Anmeldung.

[R12] Der Administrierende kann sich eine Übersicht über die bislang angemeldeten Studierenden anzeigen lassen.

In der Administrationsansicht werden alle relevanten Daten der bislang angemeldeten Studierenden angezeigt. Dies geschieht in Form einer Tabelle, die den Namen, E-Mail Adresse und eine interne ID des Studierenden beinhaltet. Zusätzlich werden die Anzahl der Priorisierungen von Projekten, aufgeteilt nach Erst-, Zweit- und Drittwunsch, angezeigt

[R13] Der Administrator kann über die Oberfläche Studierende entfernen.

In der Administrationsansicht ist das Entfernen einzelner Studierender möglich.

[R14] Die Anwendung kann alle geforderten Daten im XML-Format exportieren.

Die Daten der Studierenden können im XML-Format exportiert werden.

[R15] Die Anwendung wird als Webapplikation realisiert.

Die Implementierung erfolgt als Webapplikation über einen Webserver mit Anbindung an eine MySQL Datenbank. Zu dem werden HTML5, CSS und Javascript genutzt.

[R16] Das Design der Anwendung soll auf mobile Endgeräte angepasst sein.

Das Design soll an mobile Endgeräte angepasst werden.

4.3 Priorisierung der Anforderungen

Wie in Tabelle 4.1 zu sehen wurden die Anforderungen in einem Gespräch mit einem Mitarbeitenden des SE folgendermaßen priorisiert:

Anforderung	Priorität
[R01]	Mittel
[R02]	Gering
[R03]	Hoch
[R04]	Hoch
[R05]	Hoch
[R06]	Hoch
[R07]	Hoch
[R08]	Hoch
[R09]	Hoch
[R10]	Hoch
[R11]	Hoch
[R12]	Hoch
[R13]	Hoch
[R14]	Hoch
[R15]	Hoch
[R16]	Mittel

Tabelle 4.1: Priorisierung der Testfälle

Kapitel 5

Konzept

Um die in die in Kapitel 3 genannten Optimierungen durchzuführen, wird in diesem Kapitel das entwickelte Konzept erläutert.

In den nächsten beiden Abschnitten wird das Konzept der Datenbank und des Im- und Exportes genauer dargelegt. Anschließend wird ein Konzept für die Umsetzung des Designs genauer erläutert. Ziel dieses Konzeptes ist es, das Hinzufügen, Ändern und Entfernen von Daten über eine Benutzeroberfläche zu ermöglichen, sodass diese Aktionen direkt über die Oberfläche in der Datenbank umgesetzt werden können. Dies verringert im Vergleich zum bisher genutztem System den Zeitaufwand, da alle Datenbankänderungen über eine einzige Oberfläche durchführbar sind und auch ohne Kenntnisse von Datenbankprogrammierung umgesetzt werden können.

5.1 Datenbankentwurf

Zunächst wird ein Konzept eines Datenbankschemas vorgestellt, welches alle Anforderungen an die Datenbank erfüllen soll.

Das Anmeldetool wird über ein *Konto* zugänglich, welches über eine *E-Mail Adresse* und einem *Kennwort* angelegt wird. Zusätzlich wird der *Vor- und Nachname* des Kontoinhabers gespeichert. Als Primärschlüssel wird eine *Konto-ID* vergeben, da die E-Mail Adresse sich nicht als Primärschlüssel eignet: Es kann sein, dass die Datenbank nicht vor dem nächsten Anmeldezeitraum geleert wird und Studierende erneut an dem Softwareprojekt teilnehmen wollen. Dadurch könnten sich Studierende nicht mit der gleichen E-Mail Adresse wie im Vorjahr anmelden, da die E-Mail Adresse als Primärschlüssel dient.

Ein Konto kann entweder einem *Admin* oder einem *Studierenden* angehören. Für Studierende werden zusätzlich die Angaben der *Matrikelnummer*, der *LUH-ID*, des *Studiengangs* und ob der Studierende einen *Schein* für die Teilnahme an der Veranstaltung benötigt, gespeichert.

Der Studierende muss zu dem angeben, welche Lehrveranstaltungen er aus den möglichen Auswahloptionen belegt hat. Die Lehrveranstaltungen werden mit einer *Lehrveranstaltungs-ID* identifiziert und besitzen einen *Namen*.

Zusätzlich werden *Programmierkenntnisse* zu verschiedenen Programmiersprachen und *Softskills* abgefragt. Beide werden in einer separaten Tabelle gespeichert und erhalten eine individuelle *ID* sowie den *Namen* der entsprechenden Programmiersprache oder des Softskills. Ein weiteres Attribut namens *isvisible* bietet die Möglichkeit Optionen in der Anmeldung zu verstecken, wenn dieses auf *true* oder *false* gesetzt wird.

Zu jeder Programmiersprache oder jedem Softskill werden die Einschätzung der *Fähigkeiten* in je einer extra Tabelle in der die *ID* des Studierenden, die *ID* der entsprechende Programmiersprache oder Softskills und die angegebene Einschätzung seiner Kenntnisse gespeichert. Für die Kenntnisse werden Integer-Werte vergeben, wobei null für keine Kenntnisse, eins für wenig, zwei für mittlere und drei für hohe Kenntnisse stehen.

Zuletzt werden von den Studierenden die Prioritäten der Projekte gespeichert. Ein *Projekt* besitzt eine Datenbank interne *ID* und einen *Namen*. Die Priorisierung der Projekte von den Studierenden werden in einer separaten Tabelle gespeichert. Diese besitzt 4 Spalten: *ID* des Studierenden und in den drei Weiteren wird die Projekt-*ID* des Erst-, Zweit- und Drittwunsch gespeichert. Jeder Studierende darf dabei nur einmal in dieser Tabelle vorkommen.

In der Datenbank werden die Passwörter für den Login verbunden mit einer Email Adresse gespeichert. Die Passwörter dürfen nicht im Klartext gespeichert werden, da sie so einsehbar und verbunden mit der dazugehörigen E-Mail Adresse sind. Das gespeicherte Passwort darf auch nicht reproduzierbar sein, sondern soll nur für den Login mit dem eingegebenen Passwort verglichen werden oder zurückgesetzt werden.

Die Datenbank wurde in einem ER-Schema mithilfe von Pencil¹ entwickelt und wird in Abbildung A.1 im Anhang A.

¹Pencil: <https://pencil.evolus.vn>

5.2 Import und Export

Eine neue Funktion die der Prototyp des Anmeldetools beinhalten soll, ist der Import einer XML-Datei zum Auslesen der Projektinformationen und der Export aller Daten im XML-Format.

Import: Der Import erfolgt über die Administrationsoberfläche. Hierbei muss sichergestellt werden, dass nur eine XML-Dateien und keine anderen Formate importiert werden dürfen.

Das hochgeladene XML-Dokument soll die Projektnamen enthalten, welche ausgelesen und in der Datenbank gespeichert werden sollen.

Export: Der Export der Daten soll in ein XML-Dokument erfolgen. Exportrelevanten Daten sind die Projekte mit ID und Namen, alle sichtbaren Programmiersprachen, Softskills und Lehrveranstaltungen mit Namen und datenbankinterner ID. Zusätzlich müssen alle Studierende mit allen Angaben und gewählten Optionen in dem XML-Dokument vermerkt werden.

Zuerst sollen alle Projekte, Programmiersprachen, Lehrveranstaltungen und Softskills mit ID und Namen in dem XML-Dokument aufgelistet werden. Anschließend erfolgen alle Studierenden. Die Wahl, welche Lehrveranstaltungen die Studierenden beispielsweise gewählt haben, erfolgt über ein Mapping der Lehrveranstaltungs-ID und der ID des Studierenden. Das Mapping erfolgt analog zu den Programmiersprachen und Softskills. Die Einschätzung der Fähigkeiten erfolgt mithilfe von Integer-Werten. So wird zu jedem Studierenden die Programmiersprache oder Softskill mit ID und die Selbsteinschätzung der Fähigkeiten gespeichert.

5.3 Design

Das Tool soll zwei separate Oberflächen besitzen. Die erste Oberfläche ist die Administrationsoberfläche, über die die Mitarbeitenden des SE die Anmeldung verwalten und beobachten können, die zweite ist die Oberfläche über die sich die Studierenden an dem Softwareprojekt anmelden können.

Die Administrationsoberfläche soll ein Menü erhalten über welches folgende Optionen immer sichtbar und erreichbar sind:

Benutzerverwaltung: Unter diesem Menüpunkt befindet sich eine Seite über die die aktuellen Anmeldungen beobachtet werden können.

Als oberstes Element soll eine grafische Übersicht über die gewählte Priorisierung der Projekte erscheinen: Die im Erst-, Zweit- und Drittwunsch gewählten Projekte, sollen in drei Kreisdiagrammen prozentual dargestellt werden. Eine grafische Übersicht in Diagrammen verhilft einen schnellen Eindruck über die Projekte zu bekommen, die besonders häufig priorisiert werden. Eine Tabelle darunter soll zusätzlich für jedes Projekt die Summe derer Studierenden wiedergeben, die jenes Projekt als Erst-, Zweit- und Drittwunsch ausgesucht haben.

Neben der Projektwahl sollen alle angemeldeten Studierenden aufgeführt werden. Dabei eignet sich eine Tabelle in der die LUH-ID, Name und E-Mail Adresse jedes Studierenden aufgelistet wird. Die Auflistung dient dazu ggf. Studierende wieder finden und kontaktieren zu können. Andere Daten der Studierenden sind hierbei nicht relevant, da die Angaben später in einer intern genutzten Software weiterverarbeitet werden. Es soll hier auch möglich sein, jeden Studierenden einzeln zu entfernen, da diese Option im bisher verwendeten Tool noch nicht über die Oberfläche möglich war.

Projekte: Zur Projektübersicht werden ebenfalls alle Projekte in einer Tabelle aufgelistet. Eine Tabelle bietet den Vorteil, eine schnelle und strukturierte Übersicht zu erschaffen. In jeder Zeile soll der Name des Projektes und die Datenbank interne Projekt-ID stehen.

Zudem soll für jedes Projekt die Möglichkeit bestehen, dieses zu bearbeiten oder zu entfernen.

Unter der Projektübersicht soll sich auch die Möglichkeit befinden, ein neues Projekt manuell über die Oberfläche hinzuzufügen sowie ein XML-Dokument zu importieren, welches die Projektdaten beinhaltet.

Anmeldung verwalten: Unter diesem Menüpunkt befinden sich alle weiteren möglichen Einstellungen, die an dem Anmeldetool getroffen werden können. Ein weiteres Untermenü soll hierbei helfen, das Menü nicht zu überladen.

Das Untermenü beinhaltet zunächst den Punkt *Allgemeines*: Hier soll es möglich sein den Namen der angezeigten Webseite, die die Studierenden bei Anmeldung sehen, zu ändern. Zu dem soll das Start- und Enddatum der Anmeldung hier festgelegt werden.

Die anderen drei Punkte des Untermenüs sollen Verlinkungen auf die Seiten *Programmiersprachen verwalten*, *Lehrveranstaltungen verwalten* und *Softskills verwalten* enthalten. Diese drei Seiten sollen dabei äquivalent aufgebaut werden:

Es erfolgt eine tabellarische Übersicht mit entsprechender ID, dem Namen der Programmiersprache, Lehrveranstaltung oder Softskill und zusätzlich einer Spalte die anzeigt, ob diese Option sichtbar in der Anmeldung geschaltet ist. Die Möglichkeit Optionen sichtbar und nicht sichtbar zu schalten, soll helfen Arbeitsaufwand zu verringern: So können beispielsweise jedes Jahr unter einer Auswahl von Programmiersprachen nur jene auf sichtbar geschaltet werden, die für das aktuelle Jahr benötigt werden. Dadurch wird das wiederholte Entfernen und erneutes Hinzufügen von Optionen über mehrere Jahre gespart. Zudem soll es zu jedem Tabelleneintrag die Möglichkeit geben, diesen zu bearbeiten oder zu entfernen und auch das Hinzufügen eines neuen Tabelleneintrages soll über die Oberfläche stattfinden. In Abbildung 5.1 wird ein MockUp zu der Seite gezeigt, auf der die Programmiersprache verwaltet werden sollen.

Administration: Hier soll es möglich sein die Administrierenden zu verwalten. In einer Tabelle werden alle Administrierenden mit Namen und ihrer E-Mail Adresse aufgelistet. Im gleichen Schema, wie bei den Projekten, Programmiersprachen, Lehrveranstaltungen oder Softskills, soll es möglich sein, einen neuen Administrierenden hinzuzufügen, zu bearbeiten und zu entfernen. Dabei soll sichergestellt werden, dass ein Administrator sich nicht selbst entfernen kann. Dies soll verhindern, dass im Fall eines einzigen Administrierenden in der Datenbank, dieser sein Konto entfernt und das System unzugänglich wird.

Export: Der letzte Menüpunkt soll die Möglichkeit bieten die Daten der Datenbank im XML-Format zu exportieren. Alle Projekte, alle sichtbaren Programmiersprachen, Lehrveranstaltungen und Softskills und alle angemeldeten Studierenden und ihre Eingaben bezüglich ihrer persönlichen Angaben, ihrer Kenntnisse und ihre Projektpriorisierung sollen hier in ein XML-Dokument exportiert und heruntergeladen werden.

Die zweite Oberfläche ist für die Studierenden, die sich für das Softwareprojekt anmelden, konzipiert:

Nachdem die Studierenden mit ihrem Namen, einem Passwort und ihrer E-Mail Adresse ein Konto erstellt haben, sollen sie auf die erste Seite der Anmeldung weitergeleitet werden: Auf dieser sollen weitere Angaben zur Person möglich sein, wie die verpflichtenden Angaben der Matrikelnummer, LUH-ID, des Studiengangs und die optionale Abfrage nach einem Schein.

Auf der nächsten Seite wird nach dem erfolgreich bestandenen Lehrveranstaltungen gefragt, welche mit einer Checkbox ausgewählt werden.

Die weiterführende Seite erfragt eine Selbsteinschätzung über die Kenntnisse in der, vom Administrierenden sichtbar geschalteten, Programmiersprachen ab. Bislang wurden die Programmiersprachen links auf der Seite aufgelistet und in der gleichen Zeile daneben mit einem Single-Choice die Kenntnisse, unterteilt in *wenig*, *mittel*, *hoch* und *keine*, angezeigt. Um die Seite auch für mobile Geräte zu optimieren würde sich hier ein platzsparendes Design eignen, da die Punkte des Single-Choices viel Platz verbrauchen und die Seite verbreitern. Eine Möglichkeit wäre hier je ein Dropdown-Menü für die Kenntnisse in der jeweiligen Programmiersprache zu nutzen, um auch für kleine Bildschirme eine möglichst übersichtliche Seite zu erhalten, in der die einzelnen Auswahlpunkte sich nicht über mehrere Zeilen erstrecken. Ein MockUp für diese Oberflächen im Vergleich befinden sich in Abbildung 5.4, 5.3 und 5.2.

Eine Auswahl ihres Erst-, Zweit- und Drittwunsches der Projekte treffen die Studierende auf der nachfolgenden Seite. Auch hier eignet sich je ein Dropdown-Menü in der je nach Priorisierung alle Projekte aufgelistet sind, um wieder eine platzsparendes Design für kleine Bildschirmbreiten zu erhalten.

Eine Selbsteinschätzung über weitere Softskills werden auf einer neuen Seite abgefragt. Dabei soll die Seite äquivalent zur Selbsteinschätzung der Kenntnisse der Programmiersprachen aufgebaut sein. Neben den Softskills soll ein Dropdown-Menü die Abfrage der Einschätzung ermöglichen.

Zuletzt gelangt der Studierende auf eine Übersichtsseite, auf der alle getätigten Angaben kontrolliert und bearbeitet werden können. Diese Angaben sollen wieder tabellarisch aufgelistet werden. Dabei werden persönliche Angaben, bestandene Lehrveranstaltungen, Kenntnisse der Programmiersprachen, die Projektpriorisierung und Softskills in eigenen Tabellen aufgeführt, um diese sichtbar voneinander zu trennen. Für alle Angaben soll es zusätzlich eine Möglichkeit der Bearbeitung geben. Mit dem Abspeichern wird Studierender mit Angaben in der Datenbank gespeichert. Meldet sich ein Studierender mit seinem bereits bestehenden Konto an, soll dieser direkt auf die Übersichtsseite weitergeleitet werden.

5.3.1 MockUps

Alle MockUps wurden mit Pencil erstellt².

Der erste MockUp 5.1 zeigt die Administrationsoberfläche der Programmiersprachenverwaltung. Auf dieser befindet sich links die Navigationsleiste, über die alle wichtigen Funktionen erreicht werden können. Die Programmiersprachen werden in einer Tabelle dargestellt, wo je die ID, Name und Sichtbarkeit jeder Programmiersprache erfasst sind. An der Seite kann je die Programmiersprache bearbeitet oder entfernt werden. Über den Radiobutton in der Tabelle kann die Sichtbarkeit verändert werden und unter der Tabelle kann in dem Textfeld eine neue Programmiersprache hinzugefügt werden.

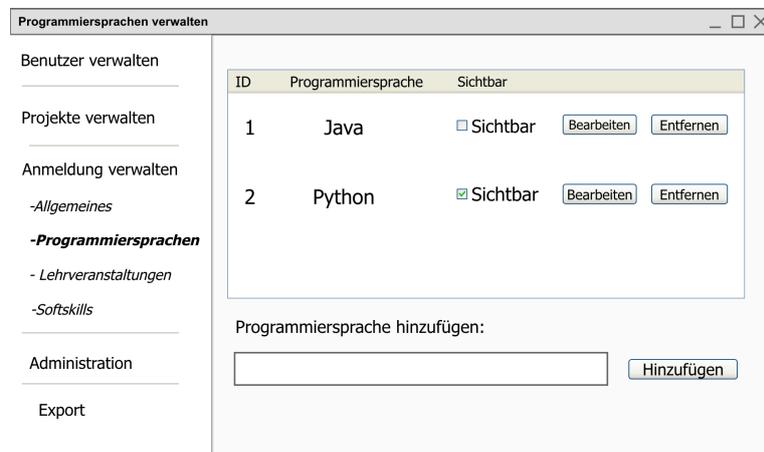


Abbildung 5.1: MockUp Administrationsoberfläche: Programmiersprachen verwalten

Es folgen drei weitere MockUps die die Ansicht der selben Seite darstellen: Auf dieser Seite können Studierende ihre Selbsteinschätzung ihrer Kenntnisse zu Programmiersprachen vornehmen. Zunächst erfolgt ein MockUp als Desktop-Ansicht 5.2, anschließend als mobile Ansicht mit Single Choice für die Kenntnisse in Abbildung 5.3 und im Vergleich dazu die selbe Ansicht mit der Realisierung Kenntnisse mit Dropdownmenüs (Abbildung 5.4).

Die Seite erscheint mit der Umsetzung von Dropdown-Menüs platzsparender und würde sich besser für die Ansicht mobiler Geräte eignen.

²Erstellt mit Pencil: <https://pencil.evolus.vn>

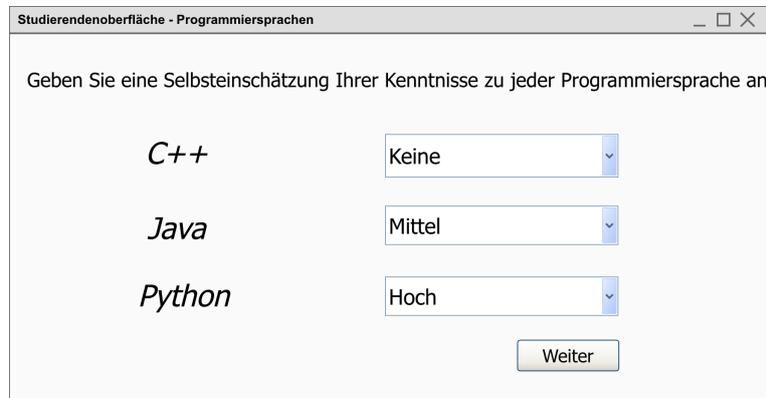


Abbildung 5.2: MockUp Studierendenoberfläche: Selbsteinschätzung der Programmierkenntnisse, Desktop Ansicht

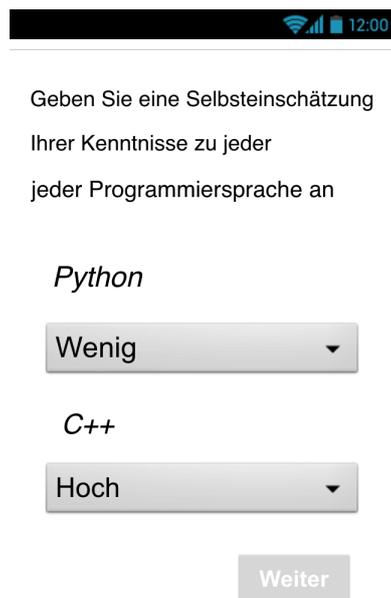


Abbildung 5.3: MockUp Studierendenoberfläche: Selbsteinschätzung der Programmierkenntnisse, mobile Ansicht mit Dropdownmenü

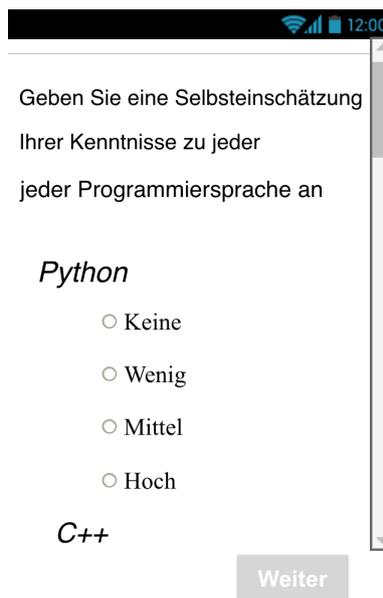


Abbildung 5.4: MockUp Studierendenoberfläche: Selbsteinschätzung der Programmierkenntnisse, mobile Ansicht mit Single Choice

Kapitel 6

Implementierung

Nach dem Erstellen der Konzepte wird im folgenden Kapitel der Ablauf der Implementierung dokumentiert. Zuerst werden einige Details der Architektur erläutert, darauffolgend wird die Umsetzung des Designkonzeptes veranschaulicht.

6.1 Architektur

Als Hilfsmittel für eine grundlegende Struktur des Anmeldetools wurde das auf PHP basierende Framework Laravel in der Version 5.8 genutzt.¹

Das Framework ist ein aktuell stetig weiterentwickeltes kostenloses Open-Source Framework, welches basierend auf dem MVC-Pattern eine vorgegebene Dateistruktur für Webapplikationen bietet. Bootstrap vereinfacht hierbei den Prozess der Frontend-Entwicklung und hat die Eigenschaft auf allen Browsern kompatibel dargestellt zu werden.

Laravel bietet außerdem Schutz vor SQL injection, Cross-Site-Request-Forgery und Cross-Site-Scripting [5]. Um das Speichern im Klartext von Daten, wie Passwörtern in der Datenbank, zu verhindern, bietet Laravel verschiedene Hashalgorithmen an.

6.2 Umsetzung des Designkonzeptes

In diesem Abschnitt wird die Umsetzung des in Kapitel 5 vorgestellten Konzeptes genauer erläutert. Mithilfe von Screenshots der Applikationen werden Funktionen weiter beschrieben.

¹<https://laravel.com>

6.2.1 Administrationsoberfläche

Die Administrationsoberfläche ist nur mit einem gültigem Admin-Account einsehbar. Eine Menüleiste im oberen Bereich der Seite ist für die Navigation für alle grundlegenden Funktionen zuständig.

i. Benutzerübersicht

Nach erfolgreicher Anmeldung als Administrierender ins System kann sich der Benutzende über die Navigationsleiste unter dem Punkt *Benutzer* eine Übersicht aller Studierenden mit Konto anzeigen lassen. Diese Übersicht ist in der folgender Abbildung 6.1 zu sehen:

Nach Anforderung [R11] zeigt diese Seite oben eine tabellarische

The screenshot shows a web interface for user management. At the top, there is a navigation menu with the following items: Softwareprojekt, Benutzer, Projekte, Registrierung, Administration, Export, and a dropdown menu for 'admin'. The main content area is titled 'Benutzerverwaltung' and contains two sections. The first section, 'Übersicht Projekte', displays a table with the following data:

Projekt	Erstwahl	Zweitwahl	Drittwahl
Projekt1	2	0	0
Projekt2	0	0	2
Projekt3	0	2	0
Projekt4	0	0	0

The second section, 'Angemeldete Studierende', displays a table with the following data:

LUH-ID	Name	Email	Entfernen
DGB-HUG	Student	student@test.com	Entfernen

Abbildung 6.1: Administrationsoberfläche: Benutzerübersicht

Übersicht der Anzahl von Priorisierungen jedes Projektes. Alle bereits angemeldeten Studierenden werden mit Namen, LUH-ID und E-Mail Adresse in einer Tabelle darunter aufgeführt. Jeder Studierende kann einzeln in der gleichen Zeile mit Hilfe des Buttons “Entfernen“ entfernt werden. Dies erfolgt aus der Anforderung [R12].

Studierende, die die Anmeldung angefangen und nicht vollständig ausgefüllt haben, werden in einer extra Tabelle darunter aufgelistet. Auch hier ist das Entfernen jedes einzelnen Benutzender über die Oberfläche möglich.

ii. Projektübersicht

Die nachfolgende Abbildung 6.2 zeigt die Projektübersicht, welche über die Navigationsleiste mit einem Klick auf den Punkt Projekte erreichbar ist.

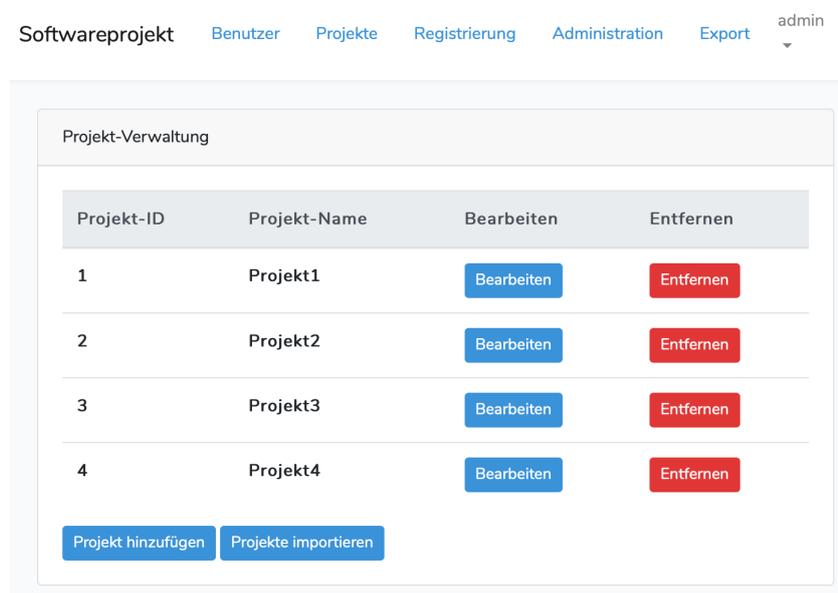


Abbildung 6.2: Administrationsoberfläche: Projektübersicht

Alle Projekte werden in einer Tabelle mit der Datenbank internen ID und ihrem Namen aufgelistet. In jeder Zeile ist über den Button “Bearbeiten“ die Bearbeitung des entsprechenden Projektes möglich. Der Benutzende wird hier auf eine neue Seite verlinkt, auf der er den Namen des Projektes über die Oberfläche bearbeiten kann [R03]. Nach erfolgreicher Bearbeitung wird der Benutzende wieder auf die Projektübersicht zurückgeleitet, auf der eine Rückmeldung erscheint, wenn das Projekt bearbeitet werden konnte. Es darf kein Projekt mit leerem Namen angegeben werden.

Das Entfernen der Projekte erfolgt über den “Entfernen“ Button in der Projektübersicht. Unter der Tabelle befindet sich ein Button zum Hinzufügen eines neuen Projektes [R03]. Daneben befindet sich ein Button für den Import eines XML-Dokumentes,

welches ausgelesen wird und die Projektinformationen aktualisiert [R02].

iii. Bearbeitung der Anmeldung

Über die Navigationsleiste ist unter dem Punkt Registrierung die Bearbeitung der Anmeldung möglich. Nach dem der Benutzer auf diesen Punkt geklickt hat, wird er auf eine neue Seite weitergeleitet, auf der sich ein Untermenü mit den Punkten "Allgemeines", "Programmiersprachen", "Lehrveranstaltungen" und "Softskills". In Abbildung A.2 im Anhang A wird das Untermenü gezeigt.

Das Erreichen der Bearbeitung der Registrierung über ein weiteres Untermenü soll verhindern, dass die Navigationsleiste überladen wirkt. Unter dem Untermenüpunkt "Allgemeines", kann der Benutzende den Namen des Softwareprojektes ändern und die Anmeldung aktiv [R04]oder inaktiv [R05] schalten.

Unter den drei Untermenüpunkten "Programmiersprachen", "Lehrveranstaltungen" und "Softskills" lässt sich die Anmeldung weiter editieren. Da alle drei Seiten äquivalent aufgebaut sind erfolgt die Erläuterung anhand des Beispiels der "Programmiersprachen".

Die Seite ist ähnlich zur bereits erklärten Projektübersicht aufgebaut: Eine Tabelle zeigt die in der Datenbank hinterlegten Programmiersprachen mit einer ID an. Zusätzlich lassen sich Programmiersprachen sichtbar oder nicht sichtbar in der Anmeldung schalten; der aktuelle Status ist in der Tabellenspalte "Sichtbar" erkennbar. Das Bearbeiten und Entfernen der Programmiersprachen ist erneut über die Buttons in der jeweiligen Zeile, sowie das Hinzufügen einer neuen Sprache über einen Button unterhalb der Tabelle möglich. Somit sind die in [R03] genannten Möglichkeiten des Hinzufügen, Ändern und Entfernen der Programmiersprachen erfüllt.

Unterhalb der Seite befindet sich das Untermenü, um so das Navigieren zwischen den Seiten zur Verwaltung der Anmeldung zu erleichtern.

In folgender Abbildung 6.3 wird die Seite zur Verwaltung der "Programmiersprachen" gezeigt:

iv. Administration

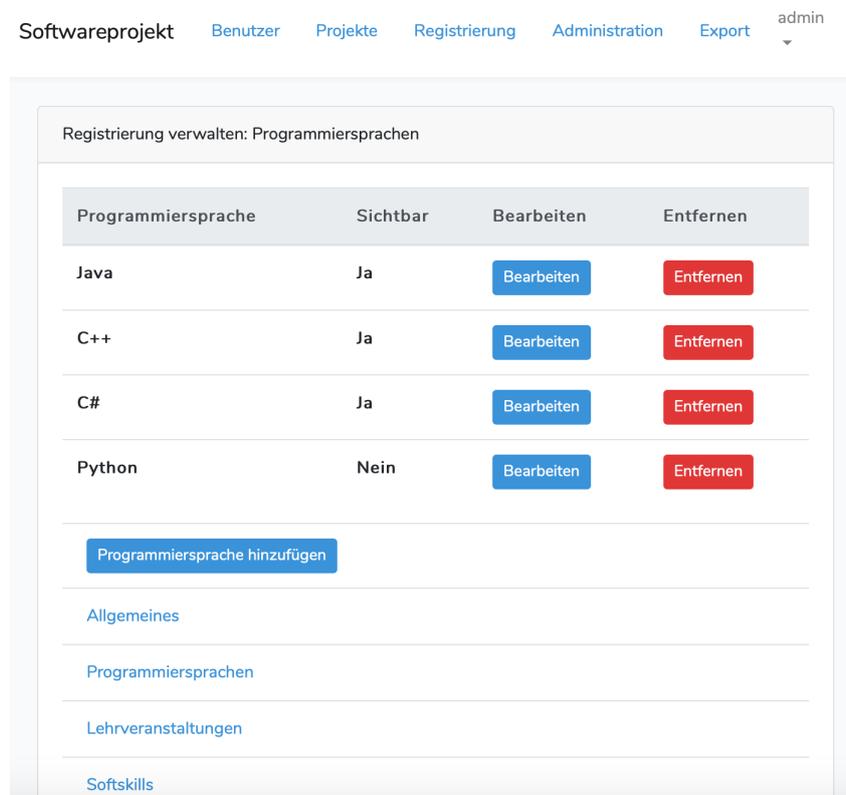


Abbildung 6.3: Administrationsoberfläche: Programmiersprachen verwalten

Hinzufügen und Verwalten der Administrierenden ist über die Navigationsleite unter dem Punkt “Administration“ möglich [R01].

v. Export

Der letzte zu erreichende Punkt ist “Export“, welcher in Anforderung [R14] erhoben wird. Nach einem Klick auf den Button, werden die Daten aus der Datenbank exportiert und im XML-Format heruntergeladen. Ein Export mit Beispieldaten wird in der Listing A.1 im AnhangA gezeigt.

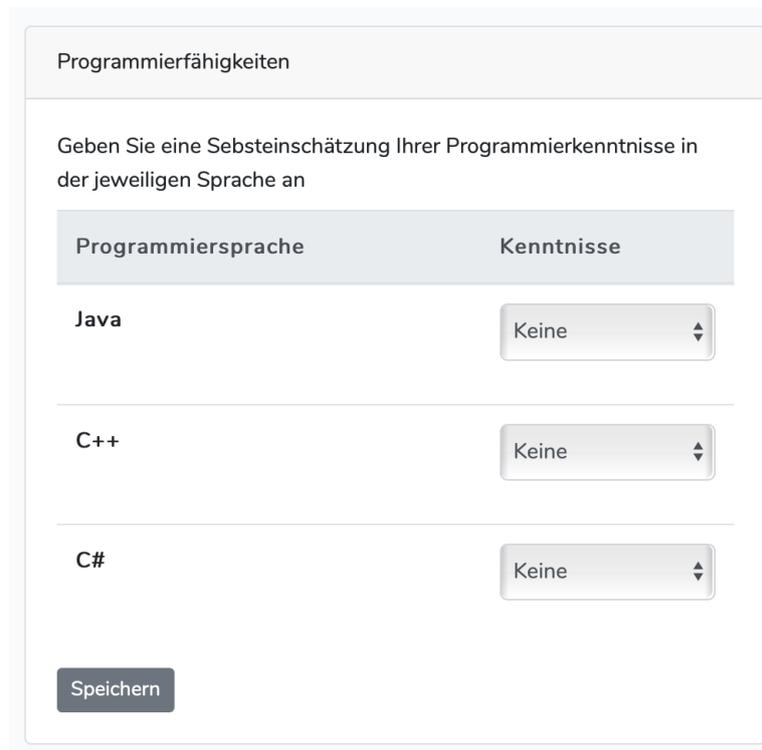
6.2.2 Studierendenoberfläche

Die Erstellung eines Accounts für Studierende erfolgt über ein Anmeldeformular, in dem lediglich der Name, das Passwort und eine E-Mail-Adresse abgefragt wird.

Zuerst wird der Studierende auf eine Seite weitergeleitet, auf der seine persönlichen Angaben wie Matrikelnummer, LUH-ID, Studiengang eingetragen werden können. Die eingegebenen Daten werden nach [R08] validiert.

Danach wird der Anwender auf eine Seite weitergeleitet, auf der Angaben zu den absolvierten Lehrveranstaltungen gemacht werden können.

Nachdem der Studierende diese Angaben getroffen hat, wird er nach einem Klick auf einen "Weiter"-Button auf eine Seite weitergeleitet, auf der eine Selbsteinschätzung der Kenntnisse zu den genannten Programmiersprachen vorgenommen werden soll. Dabei werden nur die von den Administratoren auf sichtbar geschalteten Programmiersprachen angezeigt, wie in folgender Abbildung 6.4 zu sehen:



The screenshot shows a web form titled "Programmierfähigkeiten". Below the title, there is a text prompt: "Geben Sie eine Selbsteinschätzung Ihrer Programmierkenntnisse in der jeweiligen Sprache an". The form contains a table with two columns: "Programmiersprache" and "Kenntnisse". The table lists three programming languages: Java, C++, and C#. For each language, there is a dropdown menu currently set to "Keine". At the bottom left of the form, there is a "Speichern" button.

Programmiersprache	Kenntnisse
Java	Keine
C++	Keine
C#	Keine

Speichern

Abbildung 6.4: Studierendenoberfläche: Angabe zur Selbsteinschätzung der Programmierfähigkeiten

Nach dem der Studierende eine Einschätzung getroffen hat, wird er auf eine analog aufgebaute Seite für weitere Softskills weitergeleitet. Zuletzt wird dem Studierenden eine Seite zur Priorisierung der Projekte angezeigt. Dabei muss eine Auswahl zum Erst-, Zweit- und Drittwunsch getroffen werden, wobei ein Projekt nicht mehrmals wählbar ist [R08]. Wenn trotzdem versucht wird mindestens einer der Auswahlen leer zu halten oder ein Projekt mehrmals gewählt wurde, erscheint eine Fehlermeldung, wie in Abbildung A.3 im Abbildungsverzeichnis A dargestellt.

Zum Abschluss der Anmeldung wird der Studierende auf eine Übersicht seiner eingegebenen Daten weitergeleitet. Hier ist es möglich, unter der entsprechenden Auswahl über den Button "Bearbeiten", die Angaben zu korrigieren [R09] und die Anmeldung über den Button "Speichern" zu beenden [R10]. Wenn ein Studierender die Anmeldung bereits beendet hat und sich mit seinem bestehenden Account anmeldet, wird er direkt auf die Übersicht (Abbildung 6.5) weitergeleitet:

The screenshot shows a web interface titled "Übersicht" (Overview). Below the title is a section "Übersicht Ihrer Angaben" (Overview of your data). The data is organized into three main sections:

- Zur Person** (About you):
 - Name: Student2
 - LUH-ID: DGB-HUG
 - Matrikelnummer: 24245421
 - Studiengang: Informatik
 - Schein benötigt: Nein
- Besuchte Lehrveranstaltungen** (Attended courses):
 - Programmieren 1
 - A "Bearbeiten" (Edit) button is located below this entry.
- Programmiererfahrungen** (Programming experience):
 - Java: Mittel
 - C++: Hoch

Abbildung 6.5: Studierendenoberfläche: Übersicht eingegebener Daten

6.3 Qualitätssicherung bei der Implementierung

Um die Korrektheit der umgesetzten Anforderungen zu überprüfen, wurden bei der Implementierung Testfälle erstellt. Es folgen zwei beispielhafte Testfälle für die Anforderung “[R08] Die Anwendung validiert Eingabedaten von Studierenden.“. In der Tabelle 6.1 werden Testfälle für die Validierung der Eingabe von Matrikelnummern dokumentiert. In Tabelle 6.2 werden Testfälle für die Validierung der Priorisierung der Projekte aufgeführt. Die nachfolgenden Testfälle wurden nach [13] dokumentiert:

ID	Eingabe	Sollresultat	Kommentar
TD8.1	“ABC“	“Fehleingabe“	Es dürfen keine Buchstaben in der Matrikelnummer vorkommen
TD8.2	”12563“	“Fehleingabe“	Matrikelnummern sind mindestens 7-Stellig
TD8.3	”1234567“	“Akzeptiert“	

Tabelle 6.1: Testfälle für die Validierung von Matrikelnummern

ID	Eingabe	Sollresultat	Kommentar
TD8.4	“Erstwahl: Projekt1, Zweitwahl: Projekt2, Drittwahl: Projekt1“	“Fehleingabe“	Ein Projekt darf höchstens einmal gewählt
TD8.5	“Erstwahl: Projekt1, Zweitwahl: Projekt2, Drittwahl:“	“Fehleingabe“	Es müssen genau drei Projekte priorisiert werden.
TD8.6	“Erstwahl: Projekt1, Zweitwahl: Projekt2, Drittwahl: Projekt3“	“Akzeptiert“	

Tabelle 6.2: Testfall für die Validierung der Projektwahl

6.4 Herausforderung bei der Implementierung

Laravel bietet eine eigene Struktur basierend auf den MVC Pattern. Die bereits vorgegebene Struktur ist hilfreich beim Implementieren, bereitet aber einige Einstiegsschwierigkeiten. Eine Dokumentation von Laravel ist vorhanden, allerdings ist diese sehr grundlegend gehalten.

Wenn man die neuste Version Laravels nutzt besteht ein Problem darin, dass die Hilfestellung in Literatur und im Internet sehr gering ist. Grund dafür ist, das Laravel ein stetig weiterentwickeltes Framework ist, welches von Version zu Version viele neue grundlegende Optionen bietet. Dadurch entstanden beim Start der Implementierung Schwierigkeiten, die Struktur des Frameworks zu verstehen, da die zu implementierenden und modifizierten Dateien in vielen verschiedenen Ordnern zu finden sind.

Ein Vorteil, den Laravel bietet, ist die Ausgabe eigener Fehlermeldungen. Allerdings sind diese gerade im Bereich der Verlinkungen unter den Seiten nicht sonderlich aussagekräftig, da sie Fehler in Dateien des Frameworks aufzeigt und nicht in den selbst implementierten Dateien. Dadurch ist es teilweise sehr schwer nachvollziehbar, welche Datei und welche Zeile im Code die Fehler produzieren.

Nach einer gewissen Einarbeitungszeit ist das Framework in vielen Bereichen sehr hilfreich, allerdings war die Fehlersuche teils vor allem zu Beginn der Implementierung zeitintensiv, wodurch es nicht mehr möglich war, die Anbindung von WebSSO an das System zu realisieren.

Kapitel 7

Verwandte Arbeiten

Dieses Kapitel widmet sich der Relevanz dieser Ausarbeitung im Vergleich mit anderen verwandten Arbeiten. Hierbei fällt auf, dass es Überschneidungen in Teilaspekten gibt, aber die Grundlegende Thematik unter diesem Gesichtspunkt noch nicht bearbeitet wurde.

Zunächst wird auf drei verschiedene Arbeiten eingegangen, die sich mit der Stakeholderfindung und -analyse für die Implementierung von E-Learning Systemen und Learn Management Systemen beschäftigen. Dabei wird der Fokus auf den verschiedenen gefundenen Stakeholdern und den Analyseprozess gelegt. Anschließend erfolgt eine Differenzierung zu den Stakeholdern, die für den Prototypen dieser Bachelorarbeit ermittelt worden sind.

Der Vergleich von dem zu entwickeltem Prototypen dieser Arbeit zu E-Learning System basiert darauf, dass beide Systeme von Lehreinrichtungen angeboten werden und für Lehrende ausgelegt werden.

In der Arbeit von Romero, Ballejos et al. namens *Stakeholder's analysis in e-learning software process development* [11] erfolgte eine Analyse von Stakeholdern in fünf verschiedenen Schritten, wie in folgendem Absatz erläutert:

Im ersten Schritt der Analyse werden Eigenschaften von Stakeholdern anhand von vier verschiedenen Aspekten analysiert:

- i.) Geografische und kulturelle Aspekte
- ii.) Wissenstand der Stakeholder
- iii.) Funktionen die die Stakeholder direkt oder indirekt im zukünftigen System erfüllen sollen und
- iv.) hierarchische Ebenen, die die Entwicklung beeinflussen.

Im zweiten Schritt werden die Rollen der Stakeholder in der Projektentwicklung spezifiziert und im nachfolgenden Schritt auf Basis der bisherigen Kenntnisse, eine Beschreibung zu den Stakeholdern erstellt. Diese werden anschließend mit ihren Rollen zusammen in der Projektentwicklung zusammengeführt, um abschließend eine Sortierung nach Einfluss und Interessen der Stakeholder für die Projektentwicklung vorzunehmen. Zusätzlich erfolgt eine Aufteilung der Stakeholder in internal und external Stakeholder.

Im Vergleich dazu wird in der Arbeit *Requirements' Elicitation Needs for eLearning Systems* von Nasr, Gheith und Rizk [10] Eigenschaften und Unterschiede zwischen ELearning Systemen getroffen und darauf basierend der Umgang mit der Stakeholder weiter spezifiziert. Der Aspekt der geografischer und kultureller Unterschiede zwischen den Benutzern werden aufgegriffen und des weiteren wird in dieser Arbeit davon ausgegangen, dass Stakeholder lange Zeit während des Entwicklungsprozesses unerkannt bleiben oder nie identifiziert werden [10] .

Eine weitere Arbeit namens *How to survive a learning management system implementation? A Stakeholder Analysis Approach* von Ekuase-Anwansedo, Craig et al. [4] identifiziert wiederum vier verschiedene Stakeholder Gruppen, welche in folgendem Abschnitt erläutert werden:

Als erster Stakeholder wird der *Decision Making Stakeholder* aufgeführt, dieser ist der entscheidungstreffende Stakeholder, worunter sich finanzielle Unterstützende und die Administrierenden des System, zu dem die Mitarbeitende der Fakultät und der CIO gehören. Der zweite Stakeholder umfasst *End-User Stakeholder*, hierzu gehören die Nutzenden und die Mitarbeitenden der Fakultät. Die *Vendor Stakeholder* werden hier als Entwickler und Verkäufer des Systems bezeichnet. Als letzter Stakeholder wird der *Support Stakeholder* genannt, welcher den technischen Support und Helfenden für die User umfasst. Bei der Entscheidung nach internal und external Stakeholder werden alle Stakeholdergruppen als internal und die Vendor Stakeholder als external Stakeholder aufgefasst.

Die Stakeholder Analyse dieser Arbeit erfolgte in kleinerem Umfang, da die E-Learning Systeme der oben genannten Arbeiten Systeme sind, deren Endnutzende nicht eindeutig identifizierbar sind.

Zur Entwicklung des Prototypen dieser Arbeit existiert ein bestehendes System an dem sich orientiert wurde. Die Nutzung des Systems ist zudem sehr spezifisch: Es erfolgt eine universitätsinterne Anmeldung für eine Lehrveranstaltung, an welcher nur einige Studiengänge teilnehmen. Diese wird von einem Fachgebiet durchgeführt und richtet sich nur an die Studierenden, die an der Lehrveranstaltung teilnehmen wollen.

Der größte Stakeholder hier sind die Studierenden der Leibniz Universität Hannover und weisen so ein eindeutiges Benutzerprofil auf: Auf den Bezug zu den kulturellen und geografischen Aspekten die in [11] und [10] genannt werden, beziehen sich folgende Aspekte zu den Studierenden als Stakeholder des Prototyps: Studierende, die an der Leibniz Universität Hannover studieren müssen bis auf wenige weiterführende Studiengänge Deutsch mindestens auf B1 Niveau erlernt haben, zur erfolgreichen Immatrikulation Deutschkenntnisse auf C1 Niveau vorweisen ¹. Diese Kenntnisse sind ausreichend, um die Anmeldung über den Prototypen auf deutsch durchführen zu können, sodass auf keine weitere sprachlichen Aspekte eingegangen werden muss.

Mitarbeitenden des SE wurden in dieser Arbeit als zweite Stakeholder identifiziert. Sie übernehmen Aufgaben des technischen Supports, der Administration und treffen Entscheidungen die das System beeinflussen. So sind sie sowohl entscheidungstragende, teils entwickelnde und helfende Stakeholder, um die [4] genannte Unterscheidung zwischen Stakeholdern aufzugreifen. Die Studierenden gehören zu den Endnutzenden.

Die in [10] genannte Problematik, dass Stakeholder während der Entwicklung unerkannt bleiben, ist durch den spezifischen Nutzungszweck des Anmeldetools und der Orientierung an dem bisherigen System nicht gegeben.

Eine umfassende Analyse wie sie in der Arbeit von [11] vorgenommen wird, erweist sich bei mehreren Stakeholdern als eine mögliche Methode der Stakeholderanalyse. Für den Prototypen des Anmeldetools wurden zwei Stakeholder analysiert, wobei die entscheidungstragende Stakeholder die Mitarbeitende des SE sind und somit den größten Einfluss und Interesse am System besitzen, während die Studierenden weniger Interesse und kaum Einfluss an dem System besitzen.

¹<https://www.uni-hannover.de/de/studium/vor-dem-studium/bewerbung-zulassung/voraussetzungen-zum-studium/sprachnachweise/>

Insgesamt gibt es verschiedene Ansätze Stakeholder für E-Learning und Learning Management Systemen zu finden, dabei eignen die vorgestellten Methoden für Systeme deren Nutzer unspezifischer sind. In dieser Arbeit sind die Stakeholder eindeutig zu identifizierbar gewesen, durch die Orientierung an einem vorherigem System und den spezifischen Nutzungszweck.

Kapitel 8

Zusammenfassung und Ausblick

Ziel dieser Bachelorarbeit war es, den Prozess der Anmeldeprozedur für eine Lehrveranstaltung zu optimieren. Anschließend wurde ein Prototyp entwickelt mit dem Ziel, die zuvor diskutierten Schwierigkeiten des bisher genutzten Systems zu verbessern.

8.1 Zusammenfassung

Der bisherige Prozess wurde in Kapitel 3 *Prozessoptimierung* beschrieben, um basierend darauf, mögliche Schwierigkeiten und fehlende Funktionen herauszuarbeiten und mögliche Optimierungen entwickeln zu können. Dabei haben sich vor allem Optimierungen in folgenden Bereichen ergeben:

Die Anmeldung im bisherigen System konnte nur mit hohem Arbeitsaufwand durch Modifizierung des Codes verändert werden. Das Hinzufügen neuer Optionen und auch Entfernen von Daten aus der Datenbank ist arbeitsaufwendig. Die Editierung der Anmeldung und das Entfernen von Daten aus der Datenbank über eine Oberfläche ist hierzu eine Optimierungsmöglichkeit.

Desweiteren besitzt das bisherige Anmeldetool einen Export im CSV-Format. XML-Dokumente bieten, vor allem für die interne Weiterverarbeitung der Daten mit einer intern genutzten Software, mehrere Vorteile.

Eine Validierung der Anmeldedaten der Studierenden findet noch nicht im bisher genutztem System statt. Zudem ist das Tool nicht vollständig für mobile Geräte optimiert.

Das Anmelden von Nutzern, die nicht am Projekt teilnehmen wollen oder auch die Mehrfachanmeldung von Nutzern, stellten eine weitere Schwierigkeit dar. Mögliche Lösungswege, um dies zu verhindern, wurden im Kapitel 3 Prozessoptimierung weiter beschrieben. Neben der Nutzung von universitätsinternen E-Mail Adressen zur Anmeldung oder dem Hochladen von Immatrikulationsbescheinigungen, hat sich die Nutzung des Authentifizierungsdienstes WebSSO als besonders vorteilhaft erwiesen.

Basierend auf den herausgearbeiteten Optimierungsvorschlägen wurden in Kapitel 4 *Anforderungen* Anforderungen an das System erhoben und priorisiert.

Ein genaues Konzept zur Umsetzung der Datenbank, des Exportes und Importes wie zum Design wurde in Kapitel 5 *Konzept* erläutert. Eine Neuheit zum alten System ist die Unterteilung in eine Administrationsoberfläche und eine Studierendenoberfläche. Über die Administrationsoberfläche sollen alle möglichen Änderungen, so wie der Im- und Export von Dateien möglich sein.

Die Umsetzung des Prototypen, dem das zuvor entwickelte Konzept zu Grunde liegt, wird in Kapitel 6 *Implementierung* weiter beschrieben. Neben der Vorstellung der Oberfläche, werden Herausforderungen bei der Implementierung und das Vorgehen der Qualitätssicherung des Prototyps weiter dokumentiert.

Abschließend wurde auf verwandte Arbeiten in Kapitel 7 *Verwandte Arbeiten* eingegangen und eine Differenzierung zu dieser Arbeit vorgenommen.

Insgesamt kann durch die Optimierung der Anmeldeprozedur Arbeitsaufwand vor allem auf der Seite der Administrierenden eingespart werden, da eine Administrationsoberfläche eine schnellere Bearbeitung der Anmeldung im Vergleich zur Modifizierung des Codes möglich macht.

Im Bereich der Editierbarkeit der Anmeldung bietet der Prototyp nun alle nötigen Funktionen, sodass das Hinzufügen, Bearbeiten und Entfernen von Optionen in der Anmeldung vollständig über die Benutzeroberfläche möglich ist.

Auch der Import einer XML-Datei, die Projektinformationen ausliest und speichert, wurde im Prototypen realisiert, so auch der Export der Daten im XML-Format.

Zudem werden die von Studierenden bei der Anmeldung eingegebene Daten auf Korrektheit geprüft und durch die Nutzung des Frameworks und in Zusammenhang mit Bootstrap soll die Anwendung optimiert für mobile Geräte funktionieren.

8.2 Ausblick

Dem Prototypen fehlt bislang die Möglichkeit einer eindeutigen Identifizierung von Studierenden, um Falsch- oder Mehrfachanmeldungen zu verhindern. Wie in Kapitel 3 diskutiert, würde die Benutzung von WebSSO sich hier als eine gute Möglichkeit erweisen. Die Implementierung von WebSSO hat aus zeitlichen Gründen nicht mehr stattgefunden, würde allerdings eine große Verbesserung des Anmeldetools darstellen, da hierdurch keine Falschanmeldungen mehr möglich sind und personenbezogene Informationen wie den Namen, Matrikelnummer etc. schon über WebSSO bezogen werden können. Dies würde zusätzlichen Arbeitsaufwand reduzieren.

Anhang A

Anhang

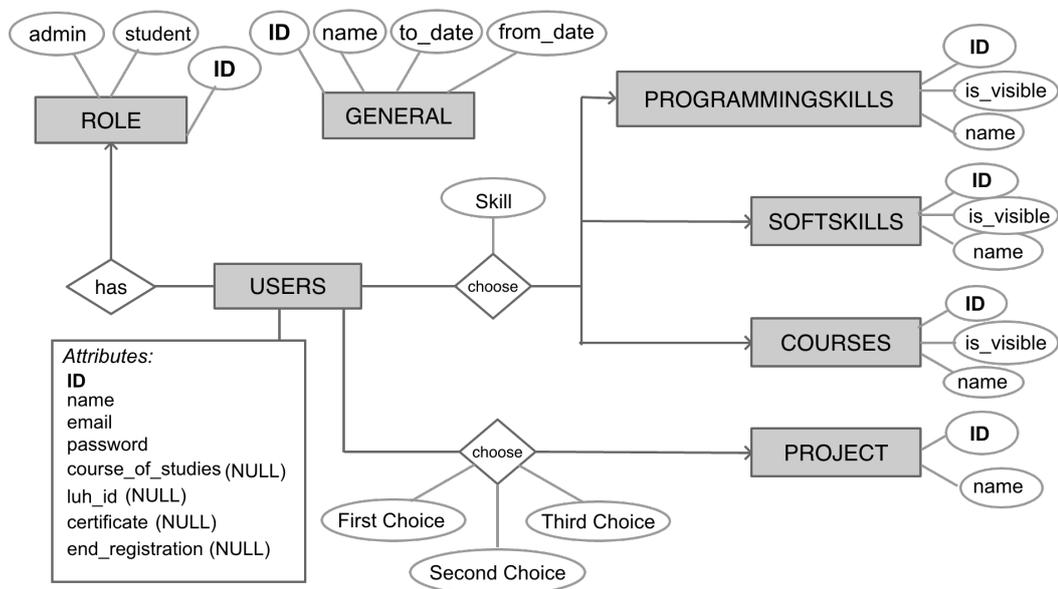


Abbildung A.1: ER-Model der Datenbank

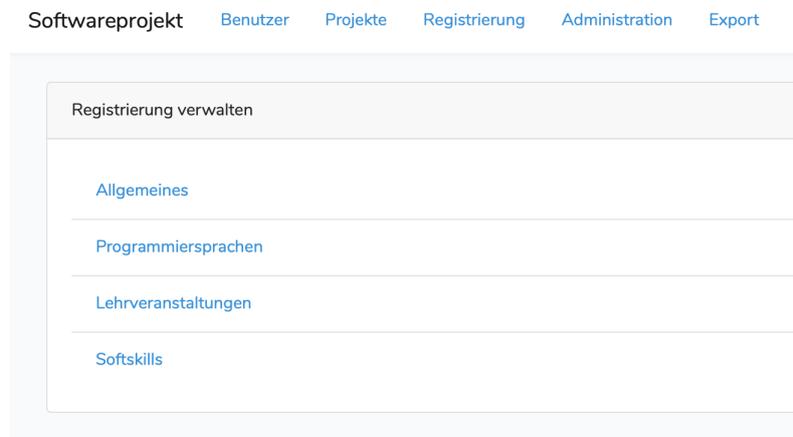


Abbildung A.2: Administrationsoberfläche: Untermenü

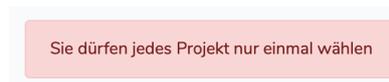


Abbildung A.3: Fehlermeldung bei mehrfacher Projektwahl

Listing A.1: Export mit Beispieldaten

```

<?xml version="1.0" encoding="UTF-8"?>
<Softwareprojekt>
  <export>
    <projects count="4">
      <project id="1">
        <name>Projekt1</name>
      </project>
      <project id="2">
        <name>Projekt2</name>
      </project>
      <project id="3">
        <name>Projekt3</name>
      </project>
      <project id="4">
        <name>Projekt4</name>
      </project>
    </projects>
    <programmingskills count="3">
      <programmingskill id="8">
        <language>Java</language>
      </programmingskill>
      <programmingskill id="9">
        <language>C++</language>
      </programmingskill>
      <programmingskill id="10">
        <language>C#</language>
      </programmingskill>
    </programmingskills>
    <softskills count="2">
      <softskill id="5">
        <softskill>
          Programmierfertigkeiten
        </softskill>
      </softskill>
      <softskill id="7">
        <softskill>
          Gruppenprojekterfahrungen
        </softskill>
      </softskill>
    </softskills>
    <courses count="2">
      <course>

```

```

        <id>5</id>
        <course>Programmieren1</course>
    </course>
    <course>
        <id>7</id>
        <course>Softwaretechnik</course>
    </course>
</courses>
<students count="1">
    <student>
        <id>2</id>
        <luh_id>DGBF-HUWG</luh_id>
        <name>Student</name>
        <email>student@test.com</email>
        <student_number>
            24245421222
        </student_number>
        <course_of_studies>
            Informatik
        </course_of_studies>
        <certificate>0</certificate>
        <course>Programmieren 1</course>
        <first_choice>1</first_choice>
        <second_choice>3</second_choice>
        <third_choice>2</third_choice>
        <programmingskill id="9">
            <skill>3</skill>
        </programmingskill>
        <programmingskill id="10">
            <skill>1</skill>
        </programmingskill>
        <programmingskill id="11">
            <skill>0</skill>
        </programmingskill>
        <softskill id="7">
            <skill>3</skill>
        </softskill>
    </student>
</students>
</export>
</Softwareprojekt>

```

Abbildungsverzeichnis

2.1	Beispiel ER-Modell	5
5.1	MockUp Administrationsoberfläche: Programmiersprachen verwalten	25
5.2	MockUp Studierendenoberfläche: Selbsteinschätzung der Programmierkenntnisse, Desktop Ansicht	26
5.3	MockUp Studierendenoberfläche: Selbsteinschätzung der Programmierkenntnisse, mobile Ansicht mit Dropdownmenü	26
5.4	MockUp Studierendenoberfläche: Selbsteinschätzung der Programmierkenntnisse, mobile Ansicht mit Single Choice	26
6.1	Administrationsoberfläche: Benutzerübersicht . . .	28
6.2	Administrationsoberfläche: Projektübersicht	29
6.3	Administrationsoberfläche: Programmiersprachen verwalten	31
6.4	Studierendenoberfläche: Angabe zur Selbsteinschätzung der Programmierfähigkeiten . .	32
6.5	Studierendenoberfläche: Übersicht eingegebener Daten	33
A.1	ER-Model der Datenbank	43
A.2	Administrationsoberfläche: Untermenü	44
A.3	Fehlermeldung bei mehrfacher Projektwahl	44

Listings

2.1	Bespielexport	5
A.1	Export mit Beispieldaten	45

Literaturverzeichnis

- [1] R. Alnaqeib, F. Alshammari, M. Zaidan, A. Zaidan, B. Bahaa, and Z. M. Hazza. An overview: Extensible markup language technology. 06 2010.
- [2] H. Barthelmeß. *E-Learning - bejubelt und verteufelt - Lernen mit digitalen Medien, eine Orientierungshilfe*. wbv Media GmbH, Bielefeld, 1. Aufl. edition, 2015.
- [3] M. Bruhn and K. Hadwich. *Dienstleistungen 4.0 - Geschäftsmodelle - Wertschöpfung - Transformation. Band 2. Forum Dienstleistungsmanagement*. Springer-Verlag, Berlin Heidelberg New York, 1. Aufl. edition, 2017.
- [4] A. Ekuase-Anwansedo, S. F. Craig, and J. Noguera. How to survive a learning management system (lms) implementation?: A stakeholder analysis approach. In *Proceedings of the 2018 ACM on SIGUCCS Annual Conference, SIGUCCS '18*, pages 165–168, New York, NY, USA, 2018. ACM.
- [5] D. Garbar. The top 10 advantages of using laravel php framework. [https:// belitsoft.com/blog/10-benefits-using-laravel-php-framework](https://belitsoft.com/blog/10-benefits-using-laravel-php-framework), 2017. Letzter Zugriff: 15.03.2019, 11:15Uhr.
- [6] V. Geroimenko. *Dictionary of XML Technologies and the Semantic Web*. Springer Science Business Media, Berlin Heidelberg, Aufl. edition, 2004.
- [7] H. Partsch. *Requirements-Engineering und -Management - Aus der Praxis von klassisch bis agil*. Carl Hanser Verlag GmbH Co KG, M, 6. Aufl. edition, 2014.
- [8] K. Pohl. *Requirements Engineering: Grundlagen, Prinzipien, Techniken*. dpunkt, Heidelberg, 2. Aufl. edition, 2007.
- [9] K. Pohl and C. Rupp. *Basiswissen Requirements Engineering - Aus- und Weiterbildung zum Certified Professional for Requirements Engineering“: Foundation Level nach IREB-Standard*. dpunkt.verlag, 2. Aufl. edition, 2010.
- [10] N. M. Rizk, M. H. Gheith, and E. S. Nasr. Requirements' elicitation needs for elearning systems. In *2016 12th*

- International Computer Engineering Conference (ICENCO)*, pages 142–147, Dec 2016.
- [11] L. Romero, L. Ballejos, M. Gutiérrez, and M. Caliusco. Stakeholder’s analysis in e-learning software process development. *EAI Endorsed Transactions on e-Learning*, 2, 03 2015.
- [12] E. Schicker. *Datenbanken und SQL - Eine praxisorientierte Einführung mit Anwendungen in Oracle, SQL Server und MySQL*. Springer-Verlag, Berlin Heidelberg New York, 5. Aufl. edition, 2017.
- [13] K. Schneider. *Abenteuer Softwarequalität - Grundlagen und Verfahren für Qualitätssicherung und Qualitätsmanagement*. dpunkt.verlag, Heidelberg, 2. Aufl. edition, 2012.
- [14] S. Surbhi. Difference between internal and external stakeholders. <https://keydifferences.com/difference-between-internal-and-external-stakeholders.html>, 2017. Letzter Zugriff: 31.08.2019, 13:38Uhr.