



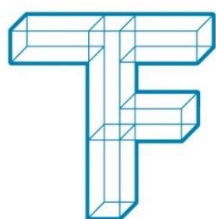
Teil 1: IPA Dokumentation

IPA-Daten

Projektname	Webbasiertes Projektportfolio		
Firmenname	Technische Fachschule Bern		
Berufsfachschule	Technische Fachschule Bern		
Autor	Danyyil Luntovsky		
Ausgabedatum	Freitag 08. März 2019		
Abteilung	Informatik		
Fachrichtung	BET		
Projektvorgehensmodell	HERMES 5		
Jahrgang und Kanton	IPA 2019, Kanton Bern		
Status	In Arbeit	In Prüfung	Zur Nutzung
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	genehmigt <input type="checkbox"/>

Beteiligter Personenkreis

Berufsbildner	Giulio Iannattone
Verantwortliche Fachkraft	Özcan Altin
Experten	VEX: Müller Andreas
	HEX: Pia Schmid
	NEX: Edgar Gmür
Projektleiter	Danyyil Luntovsky
Fachspezialist	Danyyil Luntovsky
Tester	Özcan Altin



Kurzfassung der IPA

Info

Diese Zusammenfassung richtet sich an Leser mit Fachwissen in der Informatik und vermittelt eine erste Übersicht, welche zur Erleichterung dienen soll, um die Arbeit und deren Inhalt verständlicher zu machen.

Kurze Ausgangssituation

Den Lehrpersonen und den Lernenden steht am Ende dieser Arbeit, ein Web Tool zur Verwaltung und Dokumentation von Projekten zum Gebrauch frei. Die Applikation muss auf einem Server mit vorinstallierter Umgebung geladen werden, um erreichbar sein für alle Parteien zu gewährleisten. Bei der Verwaltung von Projekten beteiligen sich nur Lehrpersonen. Die Lernenden verwenden das Tool, um die selektierten Projekte zu dokumentieren.

Umsetzung

Die Entwicklung und das Testing des Projekts wurden im IPA-Netz der TF Bern durchgeführt. Der Projektaufbau wurde mittels Projektmethode HERMES 5 vollzogen. Bei der Abwicklung des Projekts wurden die Ziele auf PkOrg befolgt.

Ergebnis

Das Projekt erfüllt alle auf PkOrg vorgeschriebenen Ziele. Somit konnte die Web-Applikation zu einem zufriedenstellenden Ergebnis erarbeitet werden. Alle Tests wurden sorgfältig durchgeführt und deren Mängel überarbeitet.

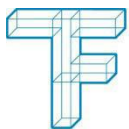
Die Anmeldung erfolgt über den Netzwerkinternen Active Directory Server. Den Lehrkräften steht es frei, Projekte zu erfassen und zu bearbeiten. Das Projekt übermittelt visuell, klare Informationen mit der Verwendung von Material Design.

Das Endresultat wurde innerhalb 84 Arbeitsstunden erstellt und dokumentiert.



Inhaltsverzeichnis

Kurzfassung der IPA	2
Info	2
Kurze Ausgangssituation	2
Umsetzung	2
Ergebnis	2
Inhaltsverzeichnis	3
1. Aufgabenstellung (kopiert und verfeinert von PkOrg)	10
1.1. Titel der Arbeit	10
1.2. Thematik	10
1.3. Ausgangslage	10
1.4. Detaillierte Aufgabenstellung	10
1.4.1. Zweck der entwickelten Lösung	10
1.4.2. Anforderungen	11
1.4.3. Erwartete Ergebnisse	11
1.4.4. Kontext	12
1.4.5. Weiterführende Arbeiten zu einem späteren Zeitpunkt	12
1.5. Mittel und Methoden	13
1.5.1. Projektmethode	13
1.5.2. Technologien	13
1.5.3. Entwicklungsumgebung:	13
1.6. Vorkenntnisse	13
1.7. Vorarbeiten	13
1.8. Neue Lerninhalte	14
1.9. Arbeiten in den letzten 6 Monaten	14
2. Standards	15
3. ISDS	16



3.1.	Zugriff auf das TF Bern Netzwerk	16
3.2.	Zugriff auf die Arbeitsstation und Testumgebung	16
3.3.	Datensicherung der IPA	16
3.4.	IPA Daten	17
3.5.	Projekt Daten	17
3.6.	Virenschutz	17
4.	Organisation der IPA	18
4.1.	Dokumentenablage	18
4.1.1.	Ordnerstruktur Tabelle	19
4.2.	Arbeitsplatz	20
4.2.1.	Hauptarbeitsplatz	20
4.2.2.	Nebenarbeitsplatz	21
5.	Detailliertes Projektvorgehen	22
5.1.	Projektmethode	22
5.1.1.	Phasen	22
5.1.2.	Szenario	23
5.1.3.	Module	23
6.	Projektorganisation	25
6.1.	Projektorganisation (Organigramm)	25
6.1.1.	Kontaktdaten	26
6.1.2.	Projektrollen	27
7.	Risikoanalyse	28
7.1.	Legende	29
7.1.1.	Schadensausmass	29
7.1.2.	Eintrittswahrscheinlichkeit	29
7.1.3.	Vor Massnahmen	30
7.1.4.	Nach Massnahmen	31
8.	Zeitplan	32



9.	Arbeitsjournal	33
9.1.	Erster Tag: Mittwoch 20.02.2019	33
9.2.	Zweiter Tag: Donnerstag 21.02.2019	36
9.3.	Dritter Tag: Freitag 22.02.2019	38
9.4.	Vierter Tag: Montag 25.02.2019	40
9.5.	Fünfter Tag: Mittwoch 27.02.2019	42
9.6.	Sechster Tag: Donnerstag 28.02.2019	44
9.7.	Siebter Tag: Freitag 01.03.2019	46
9.8.	Achter Tag: Montag 04.03.2019	48
9.9.	Neunter Tag: Mittwoch 06.03.2019	50
9.10.	Zehnter Tag: Donnerstag 07.03.2019	52
9.11.	Elfter Tag: Freitag 08.03.2019	53
10.	Abschlussbericht	54
10.1.	Fazit zur IPA	54
10.2.	Soll – Ist Vergleich	54
10.3.	Persönliches Fazit	54
10.4.	Schlussreflexion	54
	Selbständigkeitserklärung und Rechtliches für Teil 1 und Teil 2	55
	Teil 2: Projektdokumentation	56
	Hinweise zur Formatierung	57
	Allgemein	57
	Tabelle	57
	Codeeingabe	57
11.	Initialisierung	58
11.1.	IST-Situation (Umfeld, Abgrenzung, Problemverständnis)	58
11.2.	SOLL-Situation (Systemziele, Systemanforderungen / erwartete Ergebnisse)	58
11.3.	IST-SOLL Vergleich	59
11.4.	Persönliche Vorgehensziele	60



11.5.	Projektziele _____	61
11.6.	Variantenvergleich _____	62
12.	Konzept _____	63
12.1.	Netzwerkkonzept _____	63
12.1.1.	IP-Adresskonvention _____	63
12.1.2.	Netzwerkschema _____	63
12.2.	Namenskonzept _____	64
12.2.1.	Server Typ und Funktion _____	64
12.2.2.	Benutzer _____	65
12.2.3.	Laptops _____	66
12.2.4.	Switches _____	66
12.2.5.	Access Points _____	66
12.3.	ERM _____	67
12.4.	Package Diagramm _____	68
12.4.1.	Backend _____	68
12.4.2.	Frontend _____	69
12.5.	Mockup _____	70
12.5.1.	Login _____	70
12.5.2.	Dashboard _____	71
12.5.3.	Projekt Ansicht _____	72
12.5.4.	Projekt erstellen _____	73
12.6.	Testkonzept _____	74
12.6.1.	Testszenario _____	74
12.6.2.	Testrahmen _____	74
12.6.3.	Testvorgehen _____	74
12.6.4.	Testmethode _____	74
12.6.5.	Testziele _____	74
12.6.6.	Re-Testing _____	74



12.6.7.	Fehlerklassen	75
12.6.8.	Testtabelle	75
12.6.9.	Testfälle	76
13.	Realisierung	92
13.1.	Ordnerstruktur	92
13.2.	Datenbank	95
13.3.	Package Diagram	96
13.3.1.	Backend	96
13.3.2.	Frontend	97
13.4.	Migration Skript	98
13.5.	Sequenzdiagramm	103
13.6.	Vue Components (Frontend)	105
13.6.1.	Login	105
13.6.2.	AddProject	106
13.6.3.	Dashboard & MyProjects	109
13.6.4.	ViewProject	110
13.7.	Vue Routing (Frontend)	112
13.8.	Vuex Store (Frontend)	113
13.9.	Axios (Frontend)	114
13.10.	User Model (Backend)	115
13.11.	Projekt Model (Backend)	117
13.12.	Sicherheit (Backend)	119
13.13.	Design	121
13.13.1.	Login	121
13.13.2.	Project Erstellung	122
13.13.3.	Dashboard & myProjects	125
13.13.4.	ViewProject	126
13.13.5.	ViewProject bearbeiten	127



13.14.	Rechtliche Grundlagen	129
13.15.	Testprotokoll	130
13.15.1.	Re-Testing	139
13.15.2.	Testabnahme	140
	Abbildungsverzeichnis	141
	Tabellenverzeichnis	143
	Quellenverzeichnis	146
	Dokumenteninformationen	147
	Änderungskontrolle, Prüfung	147
	Freigabe der Phasen	148
	Abkürzungsverzeichnis	149
	Glossar	150
	Softwareliste	153
14.	Anhang	154
14.1.	Sitzungsprotokolle	154
14.2.	Nachweise der Versionierung und Datensicherung	156
	(ohne Nachweise keine Punkte im Beurteilungskriterium „Organisation der IPA Ergebnisse“)	
		156
14.3.	Sourcecode/Skripte	157
	Migration Skript Anleitung	158
	Db-create.sql	159
	Db-insert.sql	161
	server.js (Migration)	162
	Knexfile.js (Migration)	167
	Package.json (Migration)	168
	Config.js (Backend)	169
	Knexfile.js (Backend)	170
	AuthenticationController.js (Backend)	171



ProjectsController.js (Backend)	173
Project.js (Backend)	178
User.js (Backend)	183
Knex.js (Backend)	186
validateRequest.js (Backend)	187
Passport.js (Backend)	188
App.js (Backend)	189
routes.js (Backend)	190
Package.json (Backend)	191
AddProject.vue (Frontend)	192
Navbar.vue (Frontend)	201
showProjects (Frontend)	203
Api.js (Frontend)	206
AuthenticationService.js (Frontend)	207
ProjectsService.js (Frontend)	208
Store.js (Frontend)	209
Dashboard.vue (Frontend)	211
Login.vue (Frontend)	212
MyProjects.vue (Frontend)	214
ViewProject.vue (Frontend)	216
App.vue (Frontend)	225
Main.js (Frontend)	226
Router.js (Frontend)	227
Package.json (Frontend)	229

1. Aufgabenstellung (kopiert und verfeinert von PkOrg)

1.1. Titel der Arbeit

Webbasiertes Projektportfolio

1.2. Thematik

In den Lehrwerkstätten der Technischen Fachschule Bern fehlt eine Plattform zur Sammlung von Projektideen, sowie die Möglichkeit zur strukturierten Dokumentation von Projekten. Im Moment arbeiten die Lehrkräfte mit Excel-Tabellen. Somit muss eine Webapplikation zur Verwaltung von Projekten entstehen.

1.3. Ausgangslage

Neben dem Berufskundeunterricht haben die Lernenden der TF Bern Werkstatt-Zeit, die sie zu ihren eigenen Gunsten verwenden können. Es ist üblich, dass den Lernenden in dieser Zeit Projekte aufgetragen werden.

Jetzt soll das Webtool Abhilfe schaffen und die Lehrpersonen sowie die Lernenden bei der Projektführung unterstützen.

1.4. Detaillierte Aufgabenstellung

1.4.1. Zweck der entwickelten Lösung

Das Endprodukt soll eine Basis für die Verwaltung und Dokumentation von Projekten bereitstellen. Ebenfalls sollen die oben aufgeführten Excel-Tabellen abgeschafft und durch diese Webapplikation ersetzt werden.

1.4.2. Anforderungen

- Die Authentifizierung soll über eine Active Directory laufen (LDAP)
- Lehrkräfte müssen Projekte erfassen und bearbeiten können.
- Abhängigkeiten unter den Projekten soll ersichtlich sein
- Projekte sollen über das GUI gelöscht werden können. (Sie sollen jedoch nicht ganz aus der DB verschwinden, sondern sie sollen als "gelöscht" markiert werden)
- Es können immer alle Lehrkräfte alle Projekte bearbeiten/löschen.
- Lehrkräfte können den Projektstand dokumentieren.
- Ausserdem können den Projekten Lernende zugeteilt werden.
- Ist ein Lernender einem Projekt zugeteilt, kann er den Projektstand ebenfalls dokumentieren.

1.4.3. Erwartete Ergebnisse

Ein Projekt muss folgende Daten beinhalten:

- Titel
- Beschreibung
- Status (NEU, AKTIV, BEENDET)
- Zugeteilte Lernende
- Abhängigkeiten zu anderen Projekten
- Erstellt durch (automatisch berechnet)
- Erstellungsdatum (automatisch berechnet)
- Textfeld zur Dokumentation des Projektstandes (siehe unten)

Lernende die einem Projekt zugeteilt sind, können nur das Textfeld bearbeiten.

Lehrkräfte können Abhängigkeiten erfassen. Das heisst, dass zu jedem Projekt eine oder mehrere Projekte hinzugefügt werden können, welche die Abhängigkeiten darstellen. (Die Art der Abhängigkeit muss nicht dargestellt werden.)

Um das bestehende Personal in die DB zu importieren, erstellt der Kandidat ein Skript. Die Ausführung des Skripts geschieht über die Konsole des Servers. Dazu muss eine kurze/übersichtliche Benutzeranleitung entstehen.

- Das Skript soll aber nur die Personen importieren, welche noch nicht in der DB sind.

Der Import der Personaldaten stellt sicher, dass alle Lernende für die Zuteilung zur Verfügung stehen.

1.4.4.Kontext

Die Authentifizierung erfolgt im Labor-Netz der Lehrwerkstätte der Technischen Fachschule Bern. Das Anzapfen des produktiven AD ist nicht möglich. Als kantonale Einrichtung ist das Netzwerk der Schule dem Kanton angeschlossen. Die TF Bern baut sich nun eine eigene Infrastruktur im Labor-Netz. Die Abteilung stellt einen Active Directory Server zur Verfügung über den, die LDAP Authentifizierung laufen soll.

Während der Durchführung des Projektes werden die ICT-Bern Coding-Conventions eingehalten, da die Technische Fachschule keine derartigen Regelungen bietet.

1.4.5.Weiterführende Arbeiten zu einem späteren Zeitpunkt

Nach der Durchführung der IPA sollte der im Labor-Netz stehende Active Directory mit den produktiven Schuldaten synchronisiert werden. Damit erfolgt dann auch die interne Verwendung des Produktes unter den Beteiligten Lernenden und Lehrpersonen.

Ebenfalls soll im Rahmen dieser IPA für die Projektdokumentation, nur ein Textfeld zur Verfügung stehen. (Später muss das erweitert werden. Aber im Moment ist es wichtig, dass das Grundgerüst der Applikation steht.)

Auch das Hosting der Datenbank und der Applikation soll im späteren Verlauf, von einem Firmeninternen Server übernommen werden.

Wenn die Applikation live geht, sollen bereits laufende Projekte eingetragen werden.

Das zu entwickelnde Produkt, sowie die dazugehörige Datenbank wird im Rahmen der IPA auf dem Client des IPA Kandidaten gehostet.

1.5. Mittel und Methoden

1.5.1. Projektmethode

Als Projektmetode wird HERMES 5 angewandt.

1.5.2. Technologien

- Frontend: HTML, CSS, JS (Framework: Vue.js)
- Backend: Node.js, erweitert durch express.js & knex.js
- DB: MySQL
- Durch die Verwendung von Vue.js implementiert die Applikation (automatisch) das MVVM Design Pattern.

1.5.3. Entwicklungsumgebung:

Die Entwicklung der Webapplikation wird auf einem, von der Technischen Fachschule bereitgestellten Laptop durchgeführt.

Als Source Code Editor wird Visual Studio Code verwendet.

1.6. Vorkenntnisse

- HTML
- CSS
- JavaScript
- SQL

1.7. Vorarbeiten

Es erfolgte die Einarbeitung in folgende Themen:

- Geeignetes JavaScript-Framework (Vue.js)
- JavaScript-Backend mit DB-Anbindung (Express.js & knex.js)
- Einlesen in die Theorie des LDA-Protokolls
- Erstellung der Test User auf dem AD

1.8. Neue Lerninhalte

- LDAP & Active Directory (Personalimport)
- Einsatz von Vue.js & Express in einem konkreten Projekt

1.9. Arbeiten in den letzten 6 Monaten

- Hardwareausbau PC / Server
- Web Design
- JavaScript

2. Standards

Wie folgt sind die Standards der TF Bern sowie zusätzliche Vorschriften ersichtlich.

Standard	Beschreibung
Namenskonvention	Das Labor-Netz verfügt zurzeit noch über kein eigenes Namenskonzept. Darum wird vom Kandidaten diese selbst unter dem Punkt «Namenskonzept» definiert. Diese Namenskonvention ist jedoch nicht für das ganze Labor-Netz, sondern bezeichnet nur die Gerät, welche im Rahmen der IPA benutzt werden.
Dokumentvorlage	Die Dokumentvorlage wurde anhand der Dokumentationsvorgaben von PkOrg erstellt und wurde an den Technische Fachschule Bern Standard angepasst (Kopf- Fusszeile & Farbstyle).
Projektabwicklung	Die Technische Fachschule Bern verfügt über keine vorgeschriebene Projektabwicklungsmethode. Es ist einem Selbs überlassen, welche man benutzt.
Sicherheitskonzept	Das Sicherheitskonzept wird vom IPA Kandidaten erstellt und ist in der Dokumentation unter dem Punkt «Datensicherung der IPA» ersichtlich.
Programmcode/Skripte	Die TF Bern besitzt noch keine Standards bezüglich der Coding-Convention. Die Coding-Conventions, welche von der ICT-Berufsbildung Bern zur Verfügung gestellt wurden, werden daher berücksichtigt.

Tabelle 1 Standards

3. ISDS

Die Abteilung Informatik verfügt über zwei verschiedene Netze, die je nach Aufgabe unterschiedlich gebraucht werden. Dazu gehört das eigentliche produktive Netz der Abteilung Ressort Informatik und das Labor-Netz den Betriebsinformatiker.

3.1. Zugriff auf das TF Bern Netzwerk

Der Zugriff auf das eigentliche Netz der TF Bern ist nur angestellten gewährt. Der Authentifizierungsvorgang wird durch einen Active Directory geregelt. Das Netz verfügt ebenfalls über ein WLAN, dass auch von Besuchern genutzt werden kann.

3.2. Zugriff auf die Arbeitsstation und Testumgebung

Für die Durchführung dieser IPA wurde extra ein Subnetz (ipa.local) im Labor-Netz erstellt. Auf dieses Netz kann per LAN und WLAN zugegriffen werden. Wobei der WLAN Empfang nicht weiter als vor die Tür reicht. Es wird dauerhaft von der Arbeitsstation aus auf das Labor-Netz und das IPA-Netz zugegriffen.

3.3. Datensicherung der IPA

Die IPA Dokumente werden täglich zwei Mal während der Verfassung vom Arbeitsjournal gesichert. Das Backup wird Manuell durchgeführt, da die Betriebsinformatiker im Labor-Netz über kein automatisiertes Backupsystem verfügen. Die Vollsicherung wird auf einer externen SSD im Verzeichnis 2_backup Verschlüsselt abgelegt. Nebst der Vollsicherung wird ebenfalls eine Versionierung vom Zeitplan und von der Dokumentation erstellt.

Während der Arbeit befinden sich die IPA Daten auf dem C: Laufwerk.

3.4. IPA Daten

Die Vollsicherung wird jedes Mal in ein ZIP Archiv verwandelt und mit einem Passwort verschlüsselt. Nach jeder erfolgreichen Sicherung wird die vorgängige gelöscht.

Zugriff auf die IPA Daten hat nur der IPA Kandidat und die ober ihm stehenden Glieder (Siehe Projektorganisation Organigramm).

3.5. Projekt Daten

Die Projektdaten werden ebenfalls auf der externen SSD im Verzeichnis

2_backup/Projektportfolio_backup gesichert. Dabei werden alle generierten Dateien bis auf den *node_modules* Ordner abgelegt.

Name	Änderungsdatum	Typ	Größe
Projektportfolio_backup	07.03.2019 19:07	Dateiordner	
IPA2019-03-07.zip	07.03.2019 19:05	zip Archive	10'893 KB

Abbildung 1 Backup

3.6. Virenschutz

Alle von der Technischen Fachschule Bern bereitgestellten Clients werden von einer Schutzsoftware (TREND MICRO Security Agent) geschützt.

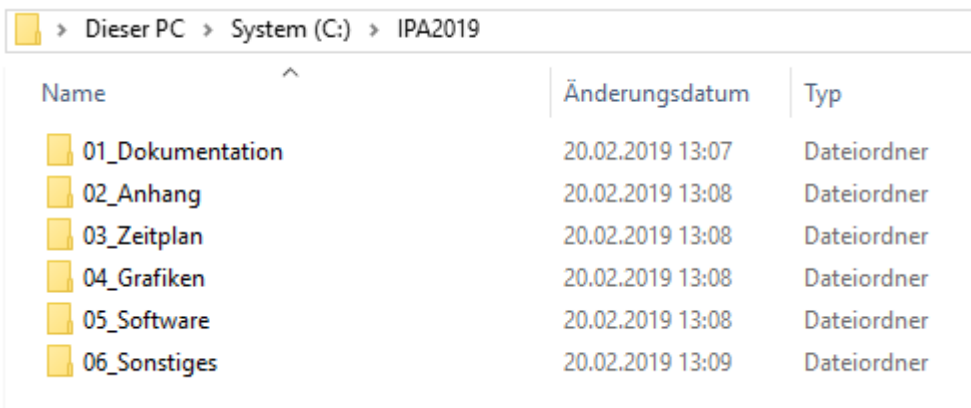
Diese Software bietet folgende Eigenschaften:

- Schutz vor Ransomware
- Schutz vor E-Mail-Angriffen
- Schutz und Verwaltung von Passwörtern

4. Organisation der IPA

4.1. Dokumentenablage

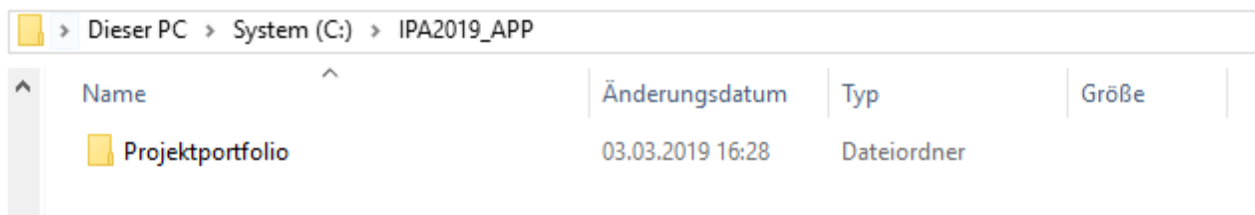
Alle IPA relevanten Dokumente werden auf dem Lokalen Laufwerk C: im Verzeichnis IPA2019/ abgelegt. Die Zugriffsrechte sind nur dem IPA Kandidaten Danyyil Luntovsky verschrieben.



Name	Änderungsdatum	Typ
01_Dokumentation	20.02.2019 13:07	Dateiordner
02_Anhang	20.02.2019 13:08	Dateiordner
03_Zeitplan	20.02.2019 13:08	Dateiordner
04_Grafiken	20.02.2019 13:08	Dateiordner
05_Software	20.02.2019 13:08	Dateiordner
06_Sonstiges	20.02.2019 13:09	Dateiordner

Abbildung 2 Dokumentenablage

Das erstellte Projekt wird ebenfalls auf dem C: Laufwerk im Ordner IPA2019_APP/Projektportfolio abgespeichert.



Name	Änderungsdatum	Typ	Größe
Projektportfolio	03.03.2019 16:28	Dateiordner	

Abbildung 3 Projektablage



Abbildung 4 Dokumentenablage

4.1.1.Ordnerstruktur Tabelle

Ordner	Beschreibung
01_Dokumentation	Beinhaltet alle Versionen der IPA-Dokumentation.
02_Anhang	Dokumente/Dateien, welche in den Anhang gehören (zum Beispiel: Kommentierter Code).
03_Zeitplan	Beinhaltet alle Versionen des Erstellten Zeitplans.
04_Bilder	Beinhaltet alle Screenshots und Bilder, die während der IPA gemacht wurden.
05_Software	In diesem Ordner befinden sich die benötigten Programme.
06_Sonstiges	In diesem Unterordner befinden sich Dokumente, welche nicht zu den Definitionen, der oben genannten Ordner passen

Tabelle 2 Ordnerstruktur

4.2. Arbeitsplatz

Während der IPA, arbeitet der IPA Kandidat an zwei Arbeitsplätzen. Der Hauptarbeitsplatz befindet sich in der Werkstatt und hat im grossen und ganzen eine bessere Ausstattung. Als der zweite Arbeitsplatz, der hauptsächlich für die Entwicklung der Applikation dient. Der zweite Arbeitsplatz ist ebenfalls direkt an den AD Server angeschlossen.

4.2.1. Hauptarbeitsplatz

Dies ist der Werkstatt Arbeitsplatz. Er befindet sich im Annex, im Aussengebäude der Technischen Fachschule Bern, im Zimmer LOA-009:



Abbildung 5 Hauptarbeitsplatz

4.2.2.Nebenarbeitsplatz

Dieser Arbeitsplatz wurde für die Entwicklung der Applikation eingerichtet:



Abbildung 6 Nebenarbeitsplatz

5. Detailliertes Projektvorgehen

5.1. Projektmethode

Als Projektmethode wird «Hermes 5» verwendet.



Abbildung 7: Projektmethode

5.1.1. Phasen

Phase	Beschreibung
Initialisierung	Die Initialisierung definiert die Ausgangslage und stellt sicher, dass die definierten Projektziele mit denen auf PkOrg übereinstimmen. Die Projektgrundlagen und der Projektauftrag werden erarbeitet.
Konzept	Die Variante, welche in der Initialisierung gewählt wird, wird präzisiert und weitere benötigte Konzepte werden erstellt. Die Ergebnisse werden so detailliert erarbeitet, dass eine aussenstehende Person sämtliche Schritte nachvollziehen kann.
Realisierung	Hier erfolgt die Umsetzung des Produkts inklusive Datenbank und Migration Skriptes. Testfälle werden durchgeführt, um die Sicherheit und die Zuverlässigkeit des Projektes zu gewährleisten.
Einführung	Die erarbeitete Lösung wird angepasst und den Kunden vorgestellt. Ebenfalls wird die Anleitung zum Migration Skript den Lehrkräften überreicht. Nach circa einer Woche uptime des Projektportfolios in einem beschränkten Ausmass an Personen, werden Umfragen an die beteiligten User versendet. Es wird auf die Kritik eingegangen und gegebenenfalls verbessert/ergänzt.

Tabelle 3: Projektphasen

5.1.2.Szenario

Die IPA behandelt das Thema der Erstellung einer Webapplikation. Es wurde dementsprechend das Szenario «Dienstleistung / Produkt» gewählt. Die untenstehenden Informationen wurden der Hermes Seite (Quellenverzeichnis HERMES Module) entnommen.

5.1.3.Module

Modul	Beschreibung
Projektsteuerung	<ul style="list-style-type: none"> - Das Projekt initialisieren, kontinuierlich steuern und mit den übergeordneten Zielen und Vorgaben der Stammorganisation in Übereinstimmung halten - Anliegen der Stakeholder berücksichtigen und integrieren, Risiken managen und Entscheide treffen - Das Projekt abschliessen
Projektführung	<ul style="list-style-type: none"> - Das Projekt planen, führen und in den definierten Rahmenbedingungen von Zeit und Kosten mit dem geforderten Ergebnis zum Ziel bringen - Die Interessen der Stakeholder kennen, die Kommunikation führen und Entscheide sicherstellen - Risiken managen, Probleme bewältigen und Erfahrungen berücksichtigen - Leistungen vereinbaren und steuern, das Änderungsmanagement und die Qualitätssicherung führen
Projektgrundlagen	<ul style="list-style-type: none"> - Die Studie erarbeiten, damit der Variantenentscheid gefällt werden kann - Die Rechtsgrundlagen klären und den Schutzbedarf analysieren - Die Voraussetzungen schaffen, um den Projektmanagementplan und den Projektauftrag zu erarbeiten



Produkt	<ul style="list-style-type: none">- Das Konzept erarbeiten und das Produkt erstellen oder beschaffen- Realisierung und Integration von IT-Systemen werden über das Modul IT-System abgewickelt
Geschäftsorganisation	<ul style="list-style-type: none">- Eine Geschäftsorganisation mit Aufbau- und Ablauforganisation neu konzipieren oder verändern, realisieren und einführen
Einführungsorganisation	<ul style="list-style-type: none">- Organisatorische Aufgaben und Massnahmen durchführen, um den Übergang vom alten zum neuen Zustand zu unterstützen- Umfasst das Organisations-Change-Management- Enthält die Vorabnahme und Abnahme

Tabelle 4 HERMES Module

6. Projektorganisation

6.1. Projektorganisation (Organigramm)

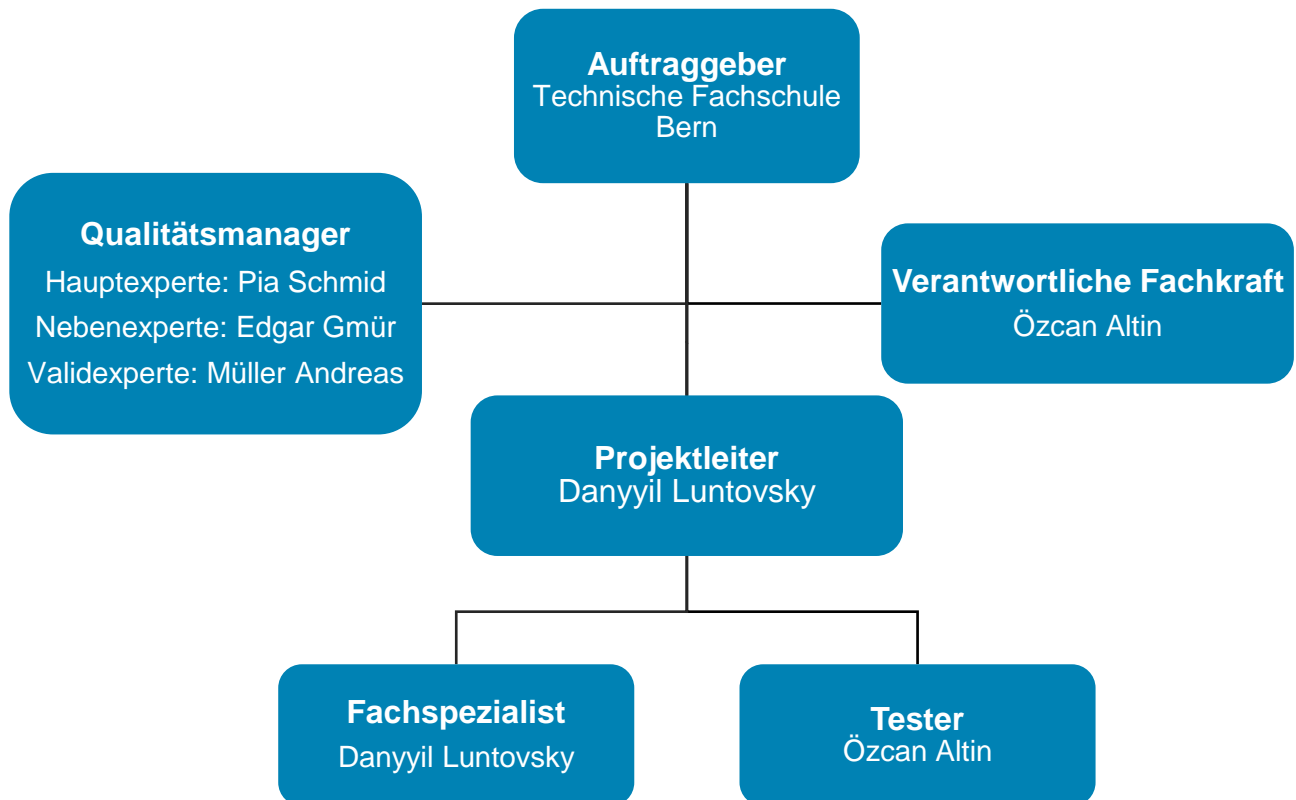


Abbildung 8: Projektorganisation

6.1.1.Kontaktdaten

Person	Kontaktdaten
Auftraggeber	Technische Fachschule Bern Lorrainestrasse 3 3013 Bern Telefon: 031 337 37 37 E-Mail: info@tfbern.ch
VEX	Müller Andreas
HEX	Pia Schmid Telefon: 079 203 08 83 E-Mail: pia.schmid@fin.be.ch
NEX	Edgar Gmür Telefon: 079 704 06 92 E-Mail: edgar.gmuer@teso.ch
Berufsbildner	Giulio Iannattone Telefon: 031 337 38 28 E-Mail: giulio.iannattone@tfbern.ch
Verantwortliche Fachkraft	Özcan Altin Telefon (P): 076 417 26 26
Tester	E-Mail: oezcan.altin@tfbern.ch
Projektleiter	Danyyil Luntovsky Bernstrasse 173A 3072 Ostermundigen
Fachspezialist	Telefon: 079 174 45 43 E-Mail: danyyil.luntovsky@gmail.com

Tabelle 5 Kontaktdaten

6.1.2. Projektrollen

Rolle	Beschreibung
Auftraggeber	Der Auftraggeber erteilt den Auftrag. Bei der IPA-Durchführung ist der Auftraggeber gleichzeitig der Fachvorgesetzte und arbeitet an der Bewertung mit.
HEX / NEX	Die praktische Prüfung wird von den Experten begleitet und beurteilt. Die Experten bewerten die Präsentation sowie das Fachgespräch. Sie verantworten gemeinsam die Gesamtnote der PA.
VEX	Die Gesamtüberprüfung des Projektes wird durch den Projektausschuss gemacht. Gleichzeitig stellt der die fachliche Richtigkeit sicher.
Verantwortliche Fachkraft	Die Verantwortliche Fachkraft ist für die Betreuung und Unterstützung des IPA Kandidaten verantwortlich. Er ist ebenfalls die erste Ansprechperson bei anfälligen Fragen und begleitet den Kandidaten während der Durchführung der IPA.
Projektleiter	Die Projektführung übernimmt der Projektleiter. Er ist das Bindeglied aller beteiligten Personen. Er ist verantwortlich, dass der Zeitplan eingehalten wird und die Qualität den Anforderungen entspricht.
Fachspezialist	Der Fachspezialist ist für die Projektumsetzung verantwortlich. Sein Ziel ist es das Projekt nach Vorgaben des Projektleiters durchzuführen.
Tester	Die Testperson übernimmt das durcharbeiten des Testkonzepts. Sie dokumentiert Mängel und Unklarheiten in Stichworten.

Tabelle 6: Projektrollen

7. Risikoanalyse

Mit Hilfe der Risikoanalyse werden Risiken und dessen Auswirkungen auf das Projekt aufgezeigt. Diese werden in der nachfolgenden Tabelle mit den potenziellen Risiken und deren Schadensausmassen und Eintrittswahrscheinlichkeiten, sowie das Schadensausmass und die Eintrittswahrscheinlichkeit aufgezeigt.

Nr.	Risikobeschreibung	Auswirkung	Vor Massnahme		Massnahmen	Nach Massnahme	
			Schadensausmass	Eintrittswahrscheinlichkeit		Schadensausmass	Eintrittswahrscheinlichkeit
R1	Krankheit / Unfall	Die verhinderte Person kann das Projekt nicht Zeitgerecht abgeben.	S4	W3	Es wird als erstes der HEX kontaktiert. Nach der Aufklärung wird das weitere Vorgehen entschieden. (Im Normalfall wird die Zeit zurückerstattet)	S1	W3
R2	Zeitprobleme	Durch den Mangel der Zeit wird der Kandidat das Projekt nicht fristgerecht einreichen können.	S3	W2	Die Erstellung eines Zeitplans für das Strukturierte Abwickeln der IPA. Damit werden Zeitmängel ersichtlich und können kontrolliert werden. Falls es trotz des Zeitplans zu Zeitproblemen kommt, werden die Auslöser genau dokumentiert und begründet.	S3	W2
R3	Datenverlust	IPA relevante Daten wiesen Mängel von Informationen auf.	S4	W2	Die IPA Dokumente werden Täglich versioniert. Ebenfalls entsteht ein Backup von den versionierten Dateien.	S2	W2
R4	Verbindungsprobleme zum Internet	Die Internetverbindung ist unzuverlässig und es können keine Recherchen angestellt werden.	S3	W3	Der Kandidat verfügt über ein mobiles Datenvolumen, dass im Falle eines Ausfalls einspringen kann.	S1	W3
R5	Systemausfall der Arbeitsstation	Alle Daten auf der Arbeitsstation sind verloren.	S4	W2	Es wird der VF kontaktiert und ein neuer Client zur weiteren Durchführung der IPA bezogen. Dank der regelmässigen Backups ist Maximal mit einem Verlust von 4h zu rechnen.	S1	W2
R6	Ausfall des für die IPA eingerichteten Routers	Es können vorerst keine LDAP Abfragen zustande kommen.	S3	W3	Es wird der VF kontaktiert und die Arbeitsstation wird direkt über den Switch an den Server Angeschlossen.	S1	W3
R7	Ausfall vom Active Directory	Es können keine LDAP Abfragen getätigt werden. Dies legt das Login sowie das Migration Skript lahm.	S4	W2	Es wird der VF kontaktiert und es wird der im Vorfeld eingerichtete AD au der GibbiX für die Entwicklung verwendet.	S1	W2

Tabelle 7: Risikoanalyse



7.1. Legende

7.1.1. Schadensausmass

Abkürzung	Beschrieb
S1	führt zu keiner Abwertung
S2	geringe Abwertung
S3	hohe Abwertung
S4	führt zu nicht bestehen

Tabelle 8: Risikoanalyse - Schadensausmass

7.1.2. Eintrittswahrscheinlichkeit

Abkürzung	Beschrieb
W1	unvorstellbar
W2	unwahrscheinlich
W3	eher vorstellbar
W4	wahrscheinlich
W5	sehr wahrscheinlich

Tabelle 9: Risikoanalyse – Eintrittswahrscheinlichkeit



7.1.3. Vor Massnahmen

Der nachfolgende Risikograph zeigt die Risiken vor dem Ergreifen der Massnahmen.

Eintrittswahrscheinlichkeit	sehr wahrscheinlich				
	wahrscheinlich				
	eher vorstellbar			R4, R6	R1
	unwahrscheinlich			R2	R3, R5, R7
	unvorstellbar				
		führt zu keiner Abwertung	geringe Abwertung	hohe Abwertung	führt zu Nichtbestehen
	Schadensausmass				

Tabelle 10: Risikograph - Vor Massnahmen



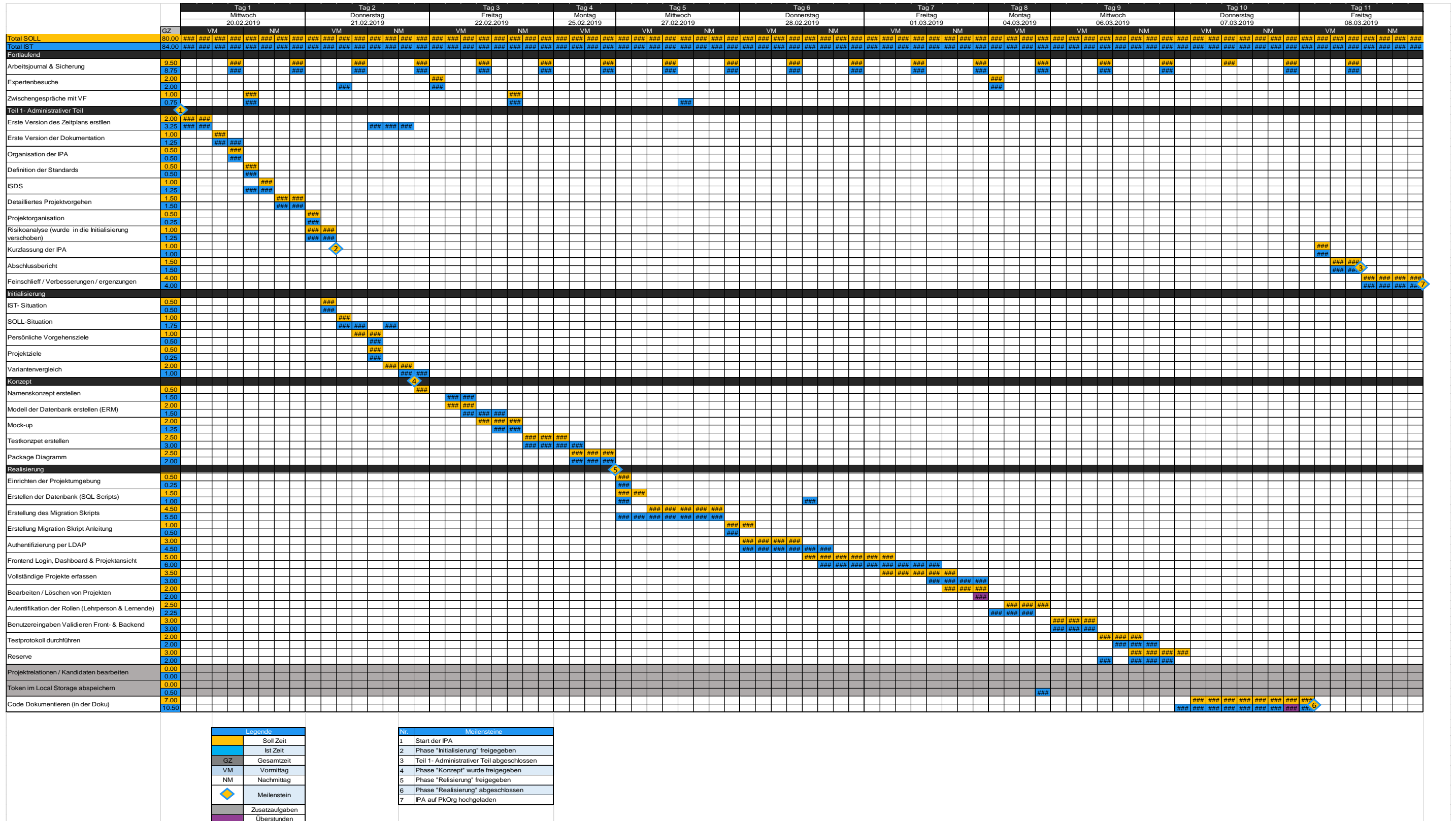
7.1.4. Nach Massnahmen

Der nachfolgende Risikograph zeigt die Risiken nach dem Ergreifen der Massnahmen.

Eintrittswahrscheinlichkeit	sehr wahrscheinlich				
	wahrscheinlich				
	eher vorstellbar	R1, R4, R6	R2		
	unwahrscheinlich	R5, R7	R3		
	unvorstellbar				
		führt zu keiner Abwertung	geringe Abwertung	hohe Abwertung	führt zu Nichtbestehen
	Schadensausmass				

Tabelle 11: Risikograph - Nach Massnahmen

8. Zeitplan

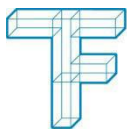




9. Arbeitsjournal

9.1. Erster Tag: Mittwoch 20.02.2019

Aufgabe	Beteiligte Personen	Aufwand geplant	Aufwand effektiv
Erste Version des Zeitplans erstellt Mit Hilfe der PkOrg Vorgaben habe ich die erste Version des Zeitplans erstellt. Zu Gunsten der Übersicht habe ich Einstündige Blöcke zur Zeitdarstellung gewählt.	Danyyil Luntovsky	2.00	2.00
Erste Version der Dokumentation Hier habe ich die bereits definierten Ziele in der Zeitplanung sowie die Detaillierte Aufgabenstellung von PkOrg als Vorlage für die Erste Version der Dokumentation verwendet.	Danyyil Luntovsky	1.00	1.25
Organisation der IPA Da ich bereits mit der Organisation des Projektes vertraut war gelang mir die Dokumentierung der Organisation.	Danyyil Luntovsky	0.5	0.5
Statusgespräch mit Özcan Altin Das Gespräch verlief rasch. Özcan Altin beantwortete mir meine Fragen klar und deutlich.	Danyyil Luntovsky Özcan Altin	0.5	0.25
Definition der Standards Ebenfalls die Standards hatte ich bereits im Kopf und musste nicht gross recherchieren.	Danyyil Luntovsky	0.5	0.5
ISDS Ich habe den Sicherungsplan entworfen und Sicherungsvorkehrungen definiert.	Danyyil Luntovsky	1.0	1.25



Detailliertes Projektvorgehen Diesen Punkt bewerkstelligte ich mit meinem Wissen und den Informationen auf der Hermes Webseite.	Danyyil Luntovsky	1.5	1.5
Arbeitsjournal führen Das Arbeitsjournal schrieb sich schnell deswegen blieb noch ein wenig Zeit übrig.	Danyyil Luntovsky	1.0	0.75
Total		8.0	8.0
Tages Ablauf			
<p>In der Schule angekommen startete ich mit viel Elan meine IPA. Direkt am Anfang machten wir mit Altin Özcan Termine für die Statusgespräche ab. Darauf folgte auch schon das Erstellen des Zeitplans mit Hilfe der Detaillierten Aufgabenstellung. Ich wurde mit dem Zeitplan nicht komplett fertig, jedoch war die Zuteilung für die nächsten drei Tage aussagekräftig.</p> <p>Beim Gespräch sowie bei der Führung der Arbeitsjournals blieb mir jeweils noch Zeit übrig, die ich dann sinnvoll in den Punkt ISDS und den Teil 1 der Doku investierte.</p>			
Problem		Lösung	
Während der Erstellung des Zeitplanes wurde ich mehrfach durch die Anwesenheit von Menschen gestört. Dieser Platz war neu für mich und ich stellte fest, dass ich mich hier nicht wohl fühlte.		Ich packte meine Sachen und zog in das Labor-Klassenzimmer neben an. Ich plante eigentlich in diesen Labor-Raum erst während der Realisierung zu ziehen.	
Ich musste die HERMES Module aufzählen und erklären.		Die HERMES Webseite hat mir bei dieser Aufgabe geholfen. (Quellenverzeichnis HERMES)	
Reflexion			
<p>Der Start meiner IPA verlief ganz nach Plan. Ich konnte mich am neuen Arbeitsplatz gut konzentrieren.</p> <p>Ich fand es jedoch schade mich von meinem Hauptarbeitsplatz zu trennen, da ich dort die bessere Ausrüstung hatte.</p> <p>Obwohl ich heute im Zeitplan war, bin ich der Überzeugung, dass es schneller voran geht könnte (ohne einen Qualitätsverlust aufzuweisen). Ich werde versuchen Morgen mehr Tempo rein zu bringen.</p>			



Weiteres Vorgehen
<ul style="list-style-type: none">- Zeitplan fertigstellen- Projektorganisation- Risikoanalyse- Initialisierung

Tabelle 12: Arbeitsjourna Tag 1

9.2. Zweiter Tag: Donnerstag 21.02.2019

Aufgabe	Beteiligte Personen	Aufwand geplant	Aufwand effektiv
Projektorganisation Ich habe ein Organigramm erstellt und die Kontaktdaten aller Beteiligten des Projektes aufgeführt.	Danyyil Luntovsky	0.5	0.25
Risikoanalyse Aufzählung aller mir bekannten Risiken, die das Projekt angehen. Mit diesem Schritt habe ich den 2. Meilenstein erreicht.	Danyyil Luntovsky	1.00	1.25
IST-Situation	Danyyil Luntovsky	0.5	0.5
SOLL-Situation Auflistung der SOLL Elemente, Vergleich zum IST Stand und jeweilige Begründung.	Danyyil Luntovsky	1.0	1.75
E-Mail an Experten versendet	Danyyil Luntovsky	-	0.25
Persönliche Vorgehensziele Definition der Meilensteine und weiterer Zeile.	Danyyil Luntovsky	1.0	0.5
Projektziele Die Ziele wurden hauptsächlich von PkOrg übernommen.	Danyyil Luntovsky	0.5	0.25
Variantenvergleich Ich habe mich online über sämtliche Node.js Dependencies informiert. 4. Meilenstein erreicht.	Danyyil Luntovsky	2.0	1.0
Zeitplan Der Zeitplan wurde fertiggestellt.	Danyyil Luntovsky	-	1.25
Arbeitsjournal	Danyyil Luntovsky	1.0	1.0
Namenskonzept erstellen Ich hatte leider keine Zeit für das Namenskonzept.	Danyyil Luntovsky	0.5	-
Total		8.0	8.0



Tages Ablauf	
<p>In den Tag startete ich mit der Erstellung der Projektorganisation. Darauf folgte die Risikoanalyse die ich mit der geschenkten Zeit von der Projektorganisation, gerade so rechtzeitig fertigstellen konnte. Ich erreichte somit den 2. Meilenstein des Projekts und fuhr mit der Initialisierung fort. Bei der IST-Situation habe ich meine Lehrperson (Marco Schmalz) für einen Screenshot vom derzeitigen Excel Projektportfolio gebeten. Nachdem ich meine Mails checkte bemerkte ich einen neuen History-Eintrag unter meiner IPA, der mir gestern wohl entwischt ist. Rasch beantwortete ich die Frage der Experten zum Thema Erster Besuchstag und fuhr mit der Arbeit fort. Das Excel-Sheet half mir enorm die IST-Lage nachzuvollziehen. Bei der SOLL-Situation habe ich viele Punkte aufgelistet und Begründet, dies rechtfertigt den Zeitüberschuss den ich für dieses Ziel verwendet habe. Die Kompensation der Zeit erfolgte durch die persönlichen Vorgehensziele. Als nächstes, bei den Projektzielen habe ich einen Grossteil von PkOrg übernommen, dies ging ziemlich Schell.</p> <p>Der Arbeitstag näherte sich dem Ende zu und für morgen wäre die erste Sitzung mit den Experten geplant. Ich priorisierte den Zeitplan für den morgigen besuch fertig zu stellen. Danach habe ich noch Recherchen zu Node.js Dependencies getätigt die ich in dem Project brauchen werde.</p>	
Problem	Lösung
Bei den persönlichen Vorgehenszielen habe ich mich mit der Zeit verschätzt, ich meinte sie genauer beschreiben zu müssen. Dies war aber nicht notwendig.	Die gewonnene Zeit habe ich mit der Überzeit von der SOLL-Analyse beglichen. Es ging zeitlich perfekt auf.
Der Zeitplan nahm zu viel Zeit in Anspruch, dass mir es nicht gereicht hat alle Zeile auf heute zu erreichen.	Ich werde probieren Morgen den Rückschlag aufzuholen.
Reflexion	
<p>Der Tag fühlte sich sehr produktiv an, trotz der Abweichung des Zeitplans.</p> <p>Ich nehme mir vor, öfters meine Mails zu checken und die Aufgaben nicht aufschieben.</p>	
Weiteres Vorgehen	
<ul style="list-style-type: none">- Variantenvergleich- Konzept	

Tabelle 13 Arbeitsjourna Tag 2



9.3. Dritter Tag: Freitag 22.02.2019

Aufgabe	Beteiligte Personen	Aufwand geplant	Aufwand effektiv
Expertenbesuch Treffen mit den beiden Experten und dem Fachvorgesetzten.	Pia Schmid Edgar Gmür Özcan Altin Danyyil Luntovsky	1.0	1.0
Namenskonzert Wurde gestern nicht behandelt.	Danyyil Luntovsky	-	1.5
ERM Die Erstellung des normalisierten Diagramms der Datenbank.	Danyyil Luntovsky	2.0	1.5
Mockup Die Designzüge der Webseite erstellen	Danyyil Luntovsky	2.0	1.25
Zwischengespräch mit VF	Özcan Altin Danyyil Luntovsky	0.5	0.25
Testkonzept Die Testfälle die nach der Realisierung von der Testperson durchgeführt werden.	Danyyil Luntovsky	1.5	1.5
Arbeitsjournal	Danyyil Luntovsky	1.0	1.0
Total		8.0	8.0

Tages Ablauf

Der dritte Tag und somit mein erster Expertenbesuch waren Angebrochen. Ich war ein wenig nervös als ich den Zeitplan und das Arbeitsjournal für die Experten gedruckt habe. In der Sitzung behandelten wir IPA relevante Themen, wie dass die Risikoanalyse in die Initialisierungsphase gehörte und dass es für mich nicht nötig wäre ein Sitzungsprotokoll zu führen (Edit: Durch spätere Absprache mit den Experten, stellte sich das Sitzungsprotokoll als doch nötig hervor). Nach der Verabschiedung fühlte ich mich erleichtert und konnte das Gestern angefangene Namenskonzept mit viel Mühe beenden. Ebenfalls beim Variantenvergleich, den ich am Vortag begonnen hatte, einigte ich mich mit den Experten, es sein zu lassen. Der Grund Dafür ist, dass die Grundlinien für das Projekt bereits vorgegeben sind und es nicht wirklich viel Raum für sämtliche Entscheide übrig hat.

Der nächste Anhaltspunkt war das ERM, mit dem ich nicht so lange beschäftigt war wie geplant. Mir fiel es leicht die DB Struktur aufzuzeichnen, wobei ich bei einem Punkt nicht ganz sicher war (Bei der OU Tabelle, mit der Aufteilung in Klassen und Rollen).

Jetzt war das Mockup an der Reihe, ich habe mich heute besonders auf dieses gefreut. Da ich mich mit dem Programm bereits auskannte und der grobe Umriss meiner Webapplikation mir schon bekannt war, konnte ich diesen Punkt schnell abhacken.

Ich führte ein Kurzes Gespräch mit dem VF und begann mit dem Testkonzept. Bei der schriftlichen Fassung ist mir erst bewusst geworden wie viele Sachen getestet werden müssen. Dies nahm viel Zeit in Anspruch und ich muss es wohl oder übel Morgen fertigstellen.

Problem	Lösung
Ich merkte dass ich zu wenig Zeit für das Testkonzept eingeplant hatte.	Ich werde mir die Zeit dafür nehmen, da das Testing ein wichtiger Bestandteil von Webapplikationen ist.

Reflexion

Während dem Namenskonzept war ich kurz demotiviert, da es so viel zu tun gab. Ich reiste mich zusammen und machte so gut wie möglichst weiter. Tatsächlich schaffte ich dennoch die zurückliegenden Aufgaben aufzuholen. Besonders Freude hatte ich heute am Mockup, da es schnell von statten ging und zugleich meiner Meinung nach schön aussah.

Weiteres Vorgehen

- Testkonzept fertigmachen
- Package Diagramm
- Sequenzdiagramm

Tabelle 14 Arbeitsjourna Tag 3

9.4. Vierter Tag: Montag 25.02.2019

Aufgabe	Beteiligte Personen	Aufwand geplant	Aufwand effektiv
Testkonzept Die bereits angefangenen Testfälle beenden	Danyyil Luntovsky	1.0	1.5
Package Diagramm & Sequenzdiagramm Grafische Darstellung der Applikation 5. Meilenstein erreicht	Danyyil Luntovsky	2.5	2.0
Arbeitsjournal	Danyyil Luntovsky	0.5	0.5
Total		4.0	4.0
Tages Ablauf			
<p>In die neue Woche startete ich mit dem unvollendeten Testkonzept, dass wie befürchtet mehr Zeit in Anspruch nahm als ursprünglich geplant war. Das schwierige am Testkonzept war, dass ich Funktionen von meiner Webseite bereits durchdenken musste, um so dann genaue Angaben dem Tester zu vermitteln.</p> <p>Heute erstellte ich ebenfalls das Package Diagramm. Auch wenn es ein wenig länger gedauert hatte, war ich komplett zufrieden mit dem Endresultat. Bei der Erstellung wusste ich schon in etwa, welche Packages und Dependencies von mir verwendet werden. Es ist nicht unwahrscheinlich falls noch ein paar weitere Tools, während der Entwicklung dazukommen.</p> <p>Ich habe mich entschieden das Sequenzdiagramm wären der Realisierung zu machen. Es wäre in meine Augen Produktiver und Zeitsparender das Diagramm im Nachhinein (während der Realisierung) zu erstellen. Das Diagramm dient ja letztendlich zu Veranschaulichung der Logik für eine aussenstehende Person.</p>			
Problem	Lösung		
Das Testkonzept war ein wenig unübersichtlich, da sehr viele Informationen vermittelt wurden.	Ich versuchte die Seite Quer zu formatieren, um horizontal mehr Platz zu bekommen. Dies sah aber im Endeffekt schlechter aus als das Hochformat, also liess ich es sein.		
Ich wusste nicht wie man ein Package Diagramm erstellt.	Ich habe auf den Webseiten visual-paradigm.com und uml-diagrams.org mich ausfindig gemacht was die Bestandteile eines Package Diagramms sind.		



Reflexion
Meiner Meinung nach habe ich das Testkonzept sehr ausführlich und detailliert erstellt. Es hat mir sogar geholfen die Funktionen, die auf meiner Webseite vorhanden sein müssen, für mich selbst im Kopf zu definieren. Jedoch muss ich diese Vorgaben jetzt ebenfalls während der Realisierung einhalten.
Weiteres Vorgehen
- Realisierung starten

Tabelle 15 Arbeitsjourna Tag 4

9.5. Fünfter Tag: Mittwoch 27.02.2019

Aufgabe	Beteiligte Personen	Aufwand geplant	Aufwand effektiv
Einrichtung der Projektumgebung Ordnerstruktur erstellen & Module installieren.	Danyyil Luntovsky	0.5	0.25
Erstellung der Datenbank Schreiben und ausführen vom SQL Script im MySQL.	Danyyil Luntovsky	1.5	0.5
Gespräch mit dem VF Wurde auf die Testfälle hingewiesen.	Özcan Altin Danyyil Luntovsky	-	0.25
Migration Skript Der erste Coding Teil der ganzen Realisierung	Danyyil Luntovsky	4.5	5.5
Erstellung Migration Skript Anleitung Anleitung für die Verwendung vom Skript durch eine Fachperson.	Danyyil Luntovsky	0.5	0.5
Arbeitsjournal	Danyyil Luntovsky	1.0	1.0
Total		8.0	8.0
Tages Ablauf			
<p>Das Einrichten der Projektumgebung war schnell erledigt.</p> <p>Bei der Erstellung der Datenbank habe ich mehr Zeit eingeplant, falls noch Änderungen aufkämen, dies war aber nicht der Fall.</p> <p>In der neuen Projektumgebung startete ich mit dem Migrationsskript. Zuerst überlegte ich den Ablauf des Skripts, wie genau der Zugriff ablaufen wird. Ständig behielt den Punkt Sicherheit und Performance im Hinterkopf. Zu Beginn kam ich gut voran und alles lief im Rahmen der Zeitplanung. Im weiteren Vorgehen entpuppten sich Syntax Schwierigkeiten, die ich lösen konnte, jedoch Zeit beanspruchten.</p> <p>Zwischendurch hatte ich noch ein Gespräch mit Özcan Altin, ich fragte ob die Grafik mit den JSON Objekten, die zwischen dem Front- und Backend herumgeschickt werden in die Realisierung gehörte, er stimmte mir zu. Ebenfalls einigten wir uns auf die Grösse des Protokoll Textfeldes, das Limit an Zeilen soll 30 Tausend betragen.</p> <p>Wie geplant fing ich an die Anleitung zum Skript zu erstellen und wurde noch heute fertig.</p>			

Problem	Lösung
Im Migration Skript musste ich zwei Arrays zusammenfügen.	Ich habe mich im Internet schlau gemacht wie concat funktioniert. Der link befindet sich im Quellenverzeichnis (Stackoverflow flatten Array).
Ich hatte zwei Arrays und musste sie vergleichen, ich wusste im Vorfeld, dass JavaScript diese Funktion bietet mit jedoch hatte ich die Syntax nicht im Kopf.	Die Webseite im Quellenverzeichnis hat mir Hilfe geboten (Stackoverflow filter compare arrays).
Nach der Erstellung des Skripts ist mir aufgefallen, dass sehr viele Anfragen einzeln an die DB gemacht werden. Da ich die User per for-Loop an die DB gesendet habe, hat Knex für jeden User ein Query abgeschickt.	Ich habe auf der Webseite (Quellenverzeichnis: Knex insert) mich genauer über das insert Statement in Knex informiert. Ich bin zum Entschluss gekommen, dass man ganze Objekte auf einmal inserten kann.
Das Thema LDAP war allgemein etwas neu für mich und ich konnte die Syntax nicht auswendig.	Ich bezog Hilfestellungen von den Webseiten (Quellenverzeichnis ldapjs.org) und (Quellenverzeichnis node-ldapjs).
Reflexion	
Ich habe mich heute sehr gefreut als ich mit der Realisierung angefangen habe. Beim Coden kamen mehrheitlich kleine Probleme vor, diese haben mich hin und wieder Zeit gekostet. Ich fand das Endresultat zufriedenstellend. Gegen Ende des Tages konnte ich mich nicht mehr so gut konzentrieren.	
Weiteres Vorgehen	
<ul style="list-style-type: none"> - Migration Skript Anleitung - Authentifizierung per LDAP 	

Tabelle 16 Arbeitsjourna Tag 5

9.6. Sechster Tag: Donnerstag 28.02.2019

Aufgabe	Beteiligte Personen	Aufwand geplant	Aufwand effektiv
Migration Skript Anleitung fertigstellen	Danyyil Luntovsky	0.5	-
Authentifizierung per LDAP Es wurde die Login Funktion implementiert.	Danyyil Luntovsky	3.0	4.5
Datenbank überarbeiten	Danyyil Luntovsky	-	0.5
Frontend Login, Dashboard & Projektansicht Das GUI für Login wurde heute beendet.	Danyyil Luntovsky	3.5	2.0
Arbeitsjournal	Danyyil Luntovsky	1.0	1.0
Total		8.0	8.0
Tages Ablauf			
<p>Am Morgen startete ich direkt mit der Authentifizierung der User, da ich die Anleitung gestern schon beendet hatte.</p> <p>Zur Implementierung verwendete ich das Programm Postman, dass Frontend Anfragen simulierte. Ich bemerkte eine Redundanz in der DB als ich die Daten abrief. Die Authentifizierung programmierte ich mit den Dependencies knex.js und ldap.js. Dabei war ldap.js herausfordernder, da ich mich nicht so gut damit auskannte.</p> <p>Insgesamt brauchte ich ein wenig länger Zeit für die Authentifizierung, ohne das DB Problem dazu zu nehmen.</p> <p>Erfolgreich aber im Verzug befand ich mich, als ich mit der Gestaltung der Seite begann. Ich schaffte es das Login fertig zu stellen und das Dashboard anzufangen.</p>			



Problem	Lösung
Die Tabelle OrganiastionalUnit im der DB beinhaltete eine Redundanz. Dies bemerkte ich anhand des DN's den ich vom AD bekam.	Ich musste das Migraion Skript & die Datenbank nochmals anpassen.
Die Erweiterung die ich verwende (ldap.js) verschickt alle Anfragen über LDAP (unverschlüsselt).	Ich probierte das LDA-Protokoll über TLS zu schicken, jedoch gab es mir ein Timeout vom Server zurück. Ich glaube es lag daran, dass man die Sichere Verbindung auf dem AD noch einrichten muss. Dazu hatte ich allerdings keine Zeit. Dies ist ein Punkt der in Betracht gezogen werden muss, nachdem die IPA vorüber ist.
Reflexion	
Von der Produktivität her, war der Tag heute durchschnittlich. Bis auf den Redundanz Fehler in der DB kamen keine grösseren Probleme vor. Es dauerte mich 30 Minuten um die Redundanz zu beseitigen.	
Weiteres Vorgehen	
<ul style="list-style-type: none">- Dashboard & Projektansicht fertig gestalten- Bearbeiten & Löschen von Projekten	

Tabelle 17 Arbeitsjourna Tag 6

9.7. Siebter Tag: Freitag 01.03.2019

Aufgabe	Beteiligte Personen	Aufwand geplant	Aufwand effektiv
Frontend Login, Dashboard & Projektansicht Es wurde der grösste Teil des Frontends fertiggestellt.	Danyyil Luntovsky	1.5	4.0
Vollständige Projekte erfassen Die Lehrpersonen können nun vollständige Projekte samt Abhängigkeiten und Kandidaten erstellen.	Danyyil Luntovsky	3.5	3.0
Bearbeiten & Löschen von Projekten Die Bearbeitung beschränkt sich auf die Vorgaben vom Detailbeschrieb. Es ist noch nicht möglich die Abhängigkeiten & die Kandidaten zu bearbeiten.	Danyyil Luntovsky	2.0	2.0
Arbeitsjournal	Danyyil Luntovsky	1.0	1.0
Total		8.0	10.0
Tages Ablauf			
<p>Der letzte Tag dieser Woche war angebrochen und ich hatte vieles zu tun. Ich fuhr mit meiner Arbeit fort das Frontend fertig zu gestalten. Dazu verwendete ich das CSS-Framework Vuetify. Das Framework half mir enorm, Komplexe Darstellungen wie die Userauswahl vereinfacht zu codieren. Danach traten die ersten grossen Probleme. Ich bezog Hilfestellungen aus dem Internet, die mir grösstenteils geholfen haben.</p> <p>Nach der schweren Zeit im Frontend, begab ich mich zur Projekterfassung. Dabei baute ich die Backend Anfragen aus und verbindet sie dann mit dem Frontend. Bei diesem Vorgehen hatte ich deutlich weniger Schwierigkeiten.</p> <p>Leider waren da, dann auch die acht Schunden Vorbei und ich hatte noch eine Aufgabe zu erledigen. Ich entschloss mich über die Zeit hinaus zu arbeiten um den Zeitrückschlag aufzuholen. Letztendlich gelang es mir dann auch.</p>			



Problem	Lösung
Ich wusste nicht wie ich mache Elemente darstellen soll.	Im Allgemeinen war die Vuetify Dokumentation sehr hilfreich. (Quellenverzeichnis Vuetify)
Die Projektansicht hat ein Modul, dass es ermöglicht Benutzer oder auch Abhängige Projekte auszuwählen. Ich hatte Probleme meine abgefragten Daten auf das Modul zuzuschneiden. Und die Daten demnach korrekt anzuzeigen.	Als Hilfestellung habe ich die Webseite (Quellenverzeichnis Vuetify autocomplete) verwendet. Komplette konnte ich das Problem aber nicht lösen, da es immer noch einen anzeige Bug gibt. Dies behindert aber nicht die Funktionalität des Moduls.
Ich wusste nicht wie man einen Zurück Knopf implementiert.	Nach einer kurzen Recherche fand ich folgende Lösung (Quellenverzeichnis: stackoverflow go route back)
Reflexion	
Wie ich es mir gewohnt war, brauchte das Frontend viel mehr Zeit als das Backend. Ich plante präventiv im Zeitplan mehr ein, doch diese reichte dennoch nicht, wie man es an den Überstunden sehen kann. Das Frontend gestaltete ich anhand des Mockups und optimierte die Benutzerfreundlichkeit sowie das Aussehen. Sonst bin ich in den zusätzlichen zwei Stunden gut vorangekommen und konnte erleichtert ins Wochenende einsteigen.	
Weiteres Vorgehen	
<ul style="list-style-type: none">- Zweiten Expertenbeuch durchführen- Authentifikation der Rollen	

Tabelle 18 Arbeitsjourna Tag 7

9.8. Achter Tag: Montag 04.03.2019

Aufgabe	Beteiligte Personen	Aufwand geplant	Aufwand effektiv
Gespräch mit Experten	Pia Schmid Edgar Gmür Özcan Altin Danyyil Luntovsky	1.0	0.75
Authentifikation der Rollen	Danyyil Luntovsky	2.5	2.75
Arbeitsjournal	Danyyil Luntovsky	0.5	0.5
Total		4.0	4.0
Tages Ablauf			
<p>Die Woche startete ich mit dem Expertenbesuch, es wurden Kriterien zur Abgabe verdeutlicht und auf organisatorische Sachen hingewiesen. Einer der wichtigsten Punkte war, dass der Source Code doch in den Anhang der Dokumentation muss. Ebenfalls wurde während dem Gespräch der aktuelle Stand des Projekts präsentiert.</p> <p>Das Gespräch war schneller vorbei als geplant, also fuhr ich mit der Realisierung etwas früher fort. Bei der Authentifikation der Rollen ging es grundsätzlich um das Zugriffsrecht im Backend. Dabei wurden Sicherheitsmechanismen mit dem JSON-Web-Token eingebaut, die Anfragen von der Client Seite aus nach der Rolle (Lernende, Lehrperson) überprüften. Mir reichte es sogar den Zusatzauftrag «Token im Local-Storage abspeichern» zu erledigen.</p>			
Problem	Lösung		
Ich kannte mich bereit mit dem JSON Web Token aus, jedoch beherrschte ich nicht die Syntax der Erweiterung passport-jwt.	Ich informierte mich auf der folgenden Webseite (Quellenverzeichnis passport-jwt)		
Das Vorgehen mit dem mit dem Token im Local-Storage abspeichern war mir neu.	Ich habe mich auf der Webseite (Quellenverzeichnis github vuex-persistedstate) über die Funktion schlau gemacht.		



Reflexion
Der heutige Tag verlief sehr produktiv. Ich habe einerseits durch das Gespräch, mehr Selbstvertrauen gewonnen und zugleich einen klaren Kopf schaffen können. Dem heutigen Stand zufolge, glaube ich dass die IPA erfolgreich abgegeben und präsentiert werden kann.
Weiteres Vorgehen
<ul style="list-style-type: none">- Validierung im Front- & Backend- Testing

Tabelle 19 Arbeitsjournale Tag 8

9.9. Neunter Tag: Mittwoch 06.03.2019

Aufgabe	Beteiligte Personen	Aufwand geplant	Aufwand effektiv
Benutzereingabe validieren (Back- & Frontend) Es reichte mir leider nur für die Frontend Validierung.	Danyyil Luntovsky	3.0	3.0
Testprotokoll durchführen Die Applikation wurde auf Chrome & Firefox getestet.	Özcan Altin Danyyil Luntovsky	2.0	2.0
Reserve Wurde für das Re-Testing angewandt.	Danyyil Luntovsky	2.0	2.0
Arbeitsjournal	Danyyil Luntovsky	1.0	1.0
Total		8.0	8.0
Tages Ablauf			
<p>Ich begann mit der Validierung der Benutzereingaben im Frontend. Mein Gedanke war, dass wenn ich nicht zur Backend-Validierung kommen sollte, das Testing trotzdem durchgeführt werden kann. Nach den drei Stunden, war es wie befürchtet, die Zeit reichte nicht um die Validierung im Backend zu schreiben. Immerhin konnte aber voraussichtlich das Testing durchgeführt werden. Mir war natürlich bewusst, dass es eine Sicherheitslücke darstellte, die Usereingaben nur im Frontend zu validieren. Diesen Schritt werde ich höchst wahrscheinlich nach der IPA als ersten implementieren.</p> <p>Da ich mich mit Özcan Altin abgemacht hatte, die Tests am Nachmittag durchzuführen, stand mir noch eine halbe Stunde zur Verfügung. Es hätte sich nicht gelohnt diese, für die Backend-Validierung aufzuwenden. Ich investierte diese Zeit um Codeoptimierungen vorzunehmen.</p> <p>Danach kam auch schon das Testing. Der Tester führte die Testfälle im Firefox durch und ich war beschäftigt mit dem Chrome Browser. Das Testing hat leichte Fehler aufgezeigt und ich konnte sie im Nachhinein, wären der Reservezeit beheben. Nach dem die Fehler behoben waren führte ich das Re-Testing erfolgreich durch.</p>			



Problem	Lösung
Immer zwischendurch, währen der Entwicklung ist mein Node Server gecrasht. Dies hatte den Nebeneffekt, dass die Applikation nicht mehr sinngemäss funktionieren konnte. Heute wollte ich mich dem Problem widmen. Obwohl ich schon in der frühen Entwicklungsphasen nach dem Fehler gesucht hatte, fand ich nichts.	Ich bat meinen Fachlehrer Marco Schmalz, um Hilfe. Er wusste auch nicht direkt von wo der Fehler kam. Jedoch gab er mir den Ratschlag im Hauptserver-File (app.js) den Error mit der Node-Erweiterung <i>longjohn</i> abzufangen. Er vermutete, dass es an einer abgeschickten Asynchronen Anfrage lag, die den Verbindungsabbruch verursachte. Ich befolgte seinen Ratschlag und es war tatsächlich eine Asynchrone Abfrage an den AD Server, die den Fehler verursachte.
Reflexion	
Diese Nacht habe ich schlecht geschlafen und war daraufhin am Morgen sehr Müde, bei meiner Arbeit. Trotzdem habe ich versucht, so produktiv wie möglichst zu sein, da der Abgabetermin vor der Tür steht. Ich bin nicht so weit gekommen wie ich wollte, dies veranlasst mich morgen mehr Gas zu geben.	
Weiteres Vorgehen	
<ul style="list-style-type: none">- Dokumentation der Realisierung	

Tabelle 20 Arbeitsjourna Tag 9

9.10. Zehnter Tag: Donnerstag 07.03.2019

Aufgabe	Beteiligte Personen	Aufwand geplant	Aufwand effektiv
Reserve Die Reserve wurde für die Realisierung verwendet.	Danyyil Luntovsky	1.0	-
Dokumentation der Realisierung Es wurde ausführlich mit Code Ausschnitten dokumentiert	Danyyil Luntovsky	6.0	9.5
Arbeitsjournal	Danyyil Luntovsky	1.0	0.5
Total		8.0	10.0
Tages Ablauf			
Heute wurde den ganzen Tag die Realisierung dokumentiert. Gegen Ende des Tages wurde ich nicht komplett fertig und beschloss, wiederum ein paar Überstunden zu leisten. Vollkommen fertig durch ich aber auch mit der Überzeit nicht. Dies ist aber nicht so schlimm, da ich noch für morgen eine Stunde eingeplant habe. Ich vermute Morgen nicht mehr als 1,5 Stunden zu benötigen.			
Problem	Lösung		
Ich habe ein Grafik über den vuex Store aus dem Internet bezogen.	Das Bild habe ich von der Seite: (Quellenverzeichnis medium vuex)		
Reflexion			
Heute habe ich sehr konzentriert und genau gearbeitet. Meiner Meinung nach vielleicht ein wenig zu genau. Ich konnte nach einer Zeit nicht mehr abschätzen wie tief ich ins Detail bei der Dokumentation soll.			
Weiteres Vorgehen			
<ul style="list-style-type: none">- Dokumentation der Realisierung fertig setllen- Kurzfassung der IPA- Abschlussbericht- Feinschliff- Abgabe auf PkOrg- Drucken / Binden			

Tabelle 21 Arbeitsjourna Tag 10



9.11. Elfter Tag: Freitag 08.03.2019

Aufgabe	Beteiligte Personen	Aufwand geplant	Aufwand effektiv
Dokumentation der Realisierung	Danyyil Luntovsky	1.0	1.0
Kurzfassung der IPA	Danyyil Luntovsky	1.0	1.0
Abschlussbericht	Danyyil Luntovsky	1.5	1.5
Arbeitsjournal	Danyyil Luntovsky	0.5	0.5
Feinschliff / Verbesserungen / Ergänzungen	Danyyil Luntovsky	4.0	4.0
Total		8.0	8.0
Tages Ablauf			
Heute habe ich die Realisierung abgeschlossen, den Code eingefügt und die Abschlussberichte geschrieben.			
Problem		Lösung	
Alle Tabellenumrisse sind verschwunden.		Ich musste allen Tabellen manuell wieder ein Raster zuweisen.	
Reflexion			
Heute war ich extrem im Stress, weil mir Kleinigkeiten auffielen die ich buhen musste.			
Weiteres Vorgehen			
- Auf PkOrg hochladen			

Tabelle 22 Arbeitsjourna Tag 11

10. Abschlussbericht

Das Dokument ist mit dem Verfassen dieses Textes nun zu ende. Es wurden 10 Arbeitstage beziehungsweise 84 Stunden für die Realisierung und Dokumentation der Arbeit aufgewendet.

10.1. Fazit zur IPA

Das Projekt konnte innerhalb des vorgegebenen Zeitraums fertiggestellt werden. Es wurden hin und wieder Hilfestellungen aus dem Internet sowie von Personen bezogen. Das Resultat spiegelt die Projektziele auf PkOrg wieder. Die Dokumentation hilft dem Leser das ganze Projektgeschehen nachzuvollziehen.

10.2. Soll – Ist Vergleich

Es wurden alle Vorgegebenen Ziele erreicht. Ein Zusatz den ich machen wollte war, das bearbeite von Abhängigen Projekten und den zugeteilten Benutzern. Dies hat leider Zeitlich nicht gepasst. Ebenfalls reichte es für die Backend Validierung der Usereingabe nicht.

10.3. Persönliches Fazit

Meiner Meinung nach wurde die Individuelle Praktische Arbeit sinngemäss, nach den Vorgaben erfüllt. Das Endresultat ist befriedigend und ich bin stolz diese Arbeit vollzogen zu haben.

Es gab hin und wieder Probleme, wo ich auf eine Fremdhilfe angewiesen war und sie dementsprechend bezog.

Als schwierigstes Vorgehen, fand ich das führen und befolgen des Zeitplans. Ich war mir nicht gewohnt so strikt nach Zeitvorgabe zu arbeiten. Ebenfalls musste die Arbeitsqualität ständig oben gehalten werden, was mir nicht immer gelungen ist.

Der Auftrag ist ausführlich gestellt, jedoch war es meiner Meinung nach zu viel Arbeit für die vorgegebene Zeit. Ich musste mich sehr anstrengen um den Zielen gerecht zu werden. Ebenso hatte ich ständigen druck, was das Dokumentieren anging.

10.4. Schlussreflexion

Trotz den negativen Aspekten, kann ich erfolgreich nach hinten blicken und von mir behaupten, dass ich vieles gelernt habe. Diese Erfahrungen sind für mein jetziges und zukünftiges Ich sehr wertvoll. Ich beende jetzt dieses Dokument und freue mich und zukünftige Projekte in meiner Laufbahn als Informatiker.

Selbständigkeitserklärung und Rechtliches für Teil 1 und Teil 2

Die lernende Person bestätigt mit ihrer Unterschrift diese IPA aus Eigenleistung erbracht und nach den Vorgaben der Prüfungskommission Informatik Kanton Bern erstellt zu haben. Die Angaben im Arbeitsjournal entsprechen dem geleisteten Arbeitsaufwand. Es ist der lernenden Person bewusst, dass Falschaussagen, nicht korrekt deklarierte Arbeitsleistungen, nicht korrekt deklarierte Fremdinhalte (Plagiate), mit der Note 1 sanktioniert werden.

Die lernende Person bestätigt mit ihrer Unterschrift ebenso, alle erforderlichen Mittel, Benutzer, Systeme, Betreuung durch die verantwortliche Fachkraft, die obligatorische Informationsveranstaltung, sowie die zwei Expertenbesuche erhalten/besucht zu haben.

Unterschrift Lernender / e



Unterschrift VF

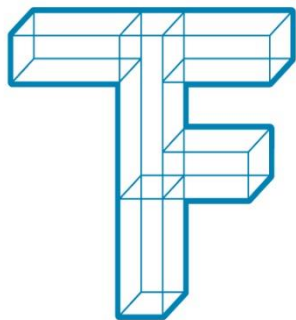




Teil 2: Projektdokumentation

IPA Projektname: Webbasiertes Projektportfolio

Autor: Danyyil Luntovsky



TECHNISCHE
FACHSCHULE
BERN



Express 



Hinweise zur Formatierung

Allgemein

Hier wird die Formatierung der Schrift im Dokument festgelegt.

Formatierung	Wert
Schriftart	Arial
Schriftgrösse	11 Pt.
Zeilenabstand	1,5 Zeilen

Tabelle 23 Formatierung Allgemein

Tabelle

Hier wird die Formatierung für die Schrift und für die Größe der Tabellen festgelegt.

Formatierung	Wert
Farbe (Titelzeile)	0, 130, 180
Zeilenhöhe	0,6
Zeilenabstand	Einfach
Textausrichtung	Links, Mittig
Rahmenlinien	Alle

Tabelle 24 Formatierung Tabelle

Codeeingabe

```
import Vue from 'Vue';
```

11. Initialisierung

11.1. IST-Situation (Umfeld, Abgrenzung, Problemverständnis)

Die Abteilung Informatik übernimmt interne sowie auch externe Projekte von Kunden. Die Führung dieser Projekte wird in Excel-Tabellen geregelt. Das derzeitige Vorgehen schränkt die Arbeitsqualität und Das Arbeitstempo enorm ein. Man hat nicht die Möglichkeit ein Projekt geregelt zu dokumentieren oder sämtliche Abhängigkeiten darzustellen. Ebenfalls ist die Lösung nicht Langlebig, da sie sehr viel Platz einnimmt und es schnell unübersichtlich werden kann.

Projektname	TFBern-App	Mastermind	Wetterstation	Grobkonzept & Offerte für Tierarztpraxis	Projekt "www.sportundausbildung.ch"	Projekt Konzept Serverraum	Projekt Startsituation	Projekt Nextcloud (Evaluation Collabora / OnlyOffice)
kurze Beschreibung	- Notenübersicht - Absenzenübersicht	eigenes Mastermind fertigstellen	Aufzeichnen von Temperaturdaten	Grobkonzept und Offerte nach Anforderungen	Website auf ein CMS umziehen, Design anpassen, Benutzeranleitung schreiben, Präsentation vor dem Kunden	Konzept für Serverraum erarbeiten	Erstellen einer Webapp zur Tagesplanung in der Werkstatt	Installation Nextcloud auf Server, Einrichten der Plugins Collabora / OnlyOffice, Evaluation
Lehrkraft(e)	özan	ams	özan	hmg	hmg, zof	AG System	hmg	hmg
Technologie, Tools	Java, Android Studio Moodle	Java	Arduino od Raspberry Pi Moodle	Moodle	CMS auf vm Moodle	Moodle	25.02.2019	LAMP Server, Nextcloud, Docker Moodle
Abgabedatum	Sommer 2019		Ende Februar 2019	21.12.2018	01.03.2019	22.03.2019		
ungefährer Aufwand pro Woche/pro Person (von insgesamt 16h/ Woche)	2		2	6	6	6	6	
Ali Ülgen		x		x	x	x		
David-Samuel Lehmann		x		x	x			
Dominik Keller	x	x		x	x	x		
Erik Mosegaard		x		x	x			
Gabriel Renaut		x		x	x			
Halid Vakuyev		x		x	x			
Hezekiah Bakare-Johnson		x	x	x	x			
James Chibuzor		x		x	x			
Jonas Gerber		x		x	x			
Josiah Schiess		x		x	x			
Julian Perren		x		x	x			
Kabilan Sivanamam	x	x		x	x		x	
Mattias Binggeli		x		x	x	x		
Michel Kaufmann		x		x	x			
Mirco Neuenschwander		x		x	x	x		
Nico Sieber	x	x		x	x	x		
Nowlan Birbaumer	x	x		x	x			
Rayan Henschli		x		x	x			
Supatra Reber		x		x	x			
Yannis Anderegg		x		x	x			

Abbildung 9 IST-Situation Projektportfolio

11.2. SOLL-Situation (Systemziele, Systemanforderungen / erwartete Ergebnisse)

Nun soll Abhilfe mittels eins Webtools geschaffen werden. Das Projektportfolio soll eine Webbasiere Variante zur Verwaltung der Aufgetragenen Projekte bieten. Die Zentralisierte Lösung soll von mehreren Personen gleichzeitig verwendet werden können. Ebenso werden die Lehrlinge in das Projektgeschehen durch das Protokollieren eingebunden. Durch die Lösung werden noch folgende Punkte behandelt:

11.3. IST-SOLL Vergleich

IST	SOLL	Begründung für SOLL
Das Dokument ist auf einem Laufwerk nur für die Lehrpersonen zugänglich.	Es soll die Anmeldung mit dem vordefiniertem TF Bern Login möglich sein.	Es haben alle Parteien die zum Projekt zugeteilt sind Zugang auf die Informationen.
Die Abhängigkeiten sind im Excel Sheet nicht ersichtlich.	Lehrkräfte sollen Abhängigkeiten erfassen können, das heisst, dass zu jedem Projekt eine oder mehrere Projekte hinzugefügt werden können.	Damit soll eine Struktur entstehen, wie Basisprojekte und weiterführende Projekte.
Die Lernenden werden per Excel Sheet zugeteilt jedoch ist es nur den Lehrpersonen ersichtlich wer zu welchem Projekt gehört.	Nur Lernende die einem Projekt zugeteilt worden sind, können auf das jeweilige zugreifen.	Die Rollenvergabe von Lernenden & Lehrpersonen macht Änderungen zwischen den Projektbeteiligten übersichtlicher. Ebenso schützt es vor unbefugten Zugriffen.
Es steht keine globale Möglichkeit zur Dokumentation zur Verfügung	Alle Lehrpersonen und nur Lernende die einem Projekt zugeteilt sind, können das Textfeld bearbeiten.	Die Dokumentation von Projekten ist ein wichtiger Bestandteil der Übersicht wie auch der Erweiterung. Da das Textfeld auf lange Sicht nicht ausreichend ist, soll es später (nach der IPA) durch ein kleines Forum ersetzt werden.

Es müssen alle Kandidaten von einem Projekt eigenhändig ergänzt werden.	Es soll ein Migration Skript für den Import der Userdaten entstehen. Das Skript migriert nur die User die sich noch nicht in der Datenbank befinden. Zudem soll auch noch eine übersichtliche Anleitung entstehen.	Mit einem Skript wird sichergestellt, dass alle User übertragen werden. Die Anleitung soll simple aufgebaut sein, dass jeder das Skript bedienen kann.
Die Statusanzeige wird vom Excel Sheet nicht unterstützt.	Zuletzt soll noch eine kleine Statusmeldung (Neu, Aktiv, Beendet) ersichtlich sein, in welchem Stadium sich das Projekt befinden.	Dies soll helfen nicht nur Projekte durchzuführen, sondern auch sie zu planen. Mit dem Status «Neu» wird das Projekt als Idee aufgenommen.

Tabelle 25 IST- SOLL Vergleich

11.4. Persönliche Vorgehensziele

Ziel	Beschreibung
Hermes 5	Das Projekt wird mittels HERMES 5 abgewickelt.
Einhaltung des Zeitplans	Kleinere Abweichungen bis zu 3 Stunden werden geduldet.
Arbeitsjournal schreiben	Jedes Mal vor dem Mittag (11:30) und vor dem Arbeitsende (16:30) wird über den Tag berichtet.
MS1	Teil 1 – Administrativer Teil beendet
MS2	Initialisierungsphase beendet
MS3	Konzeptphase beendet
MS4	Realisierungsphase beendet
MS5	Präsentation mit Fachgespräch absolviert (Nicht im Zeitplan aufgeführt)
IPA Abgabe	Das Fertige Produkt der 10 vollständigen Arbeitstage wird online auf PkOrg hochgeladen. Ebenso wird die ausgedruckte IPA per Post den Experten wie auch den VF überreicht.

11.5. Projektziele

Die Projektziele legt fest, was alles Ende des Projekts erfüllt sein muss.

Nummer	Ziel
1	Das Login erfolgt über die AD
2	Lehrkräfte können ein neues Projekt erfassen: Titel und Beschreibung werden erfasst. Der Status des Projektes ist NEU.
3	Lehrkräfte können Titel, Beschreibung und den Status ändern.
4	Lehrkräfte können Projekte löschen (in DB nur als gelöscht markiert)
5	Lernende können weder erfassen, noch ändern oder löschen.
6	Lehrkräfte können den Projekten Lernende zuordnen. Die Zuteilung ist im GUI ersichtlich.
7	Lehrkräfte können für Projekte Abhängigkeiten zu anderen Projekten erfassen. Abhängigkeiten eines Projektes sind im GUI ersichtlich. Mit einem Klick ist es möglich die Übersicht des eingetragenen Projektes zu öffnen.
8	Lehrkräfte und zugeteilte Lernende können das Textfeld zur Projektdokumentation bearbeiten.
9	Das Import-Skript importiert alle Personen aus dem Active Directory in die eigene DB.
10	Dabei werden nur Personen importiert, welche nicht schon existieren.

Tabelle 26 Projektziele



11.6. Variantenvergleich

Da Tools, Programmiersprache und die meisten Ziele bereits im Vorfeld durch die Detaillierte Aufgabenstellung vordefiniert wurden, wird der Variantenentschied weggelassen. Dies wurde so mit den Experten abgemacht.

12. Konzept

12.1. Netzwerkkonzept

Das IPA Netz ist am Labor Netz des ANEX angehängt. Das komplette IPA Netz wurde als Vorarbeit eingerichtet und dient somit zur Verfügung während der gesamten Individuellen Arbeit.

12.1.1. IP-Adresskonvention

Typ		IP-Adresse von	IP-Adresse bis
Statische IP	Netzwerkgeräte	10.10.10.1	10.10.10.19
Statische IP	Server	10.10.10.20	10.10.10.30
Statische IP	Drucker	10.10.10.31	10.10.10.49
Dynamische IP	DHCP	10.10.10.50	10.10.10.254

Tabelle 27 IP-Adresskonvention

12.1.2. Netzwerkschema

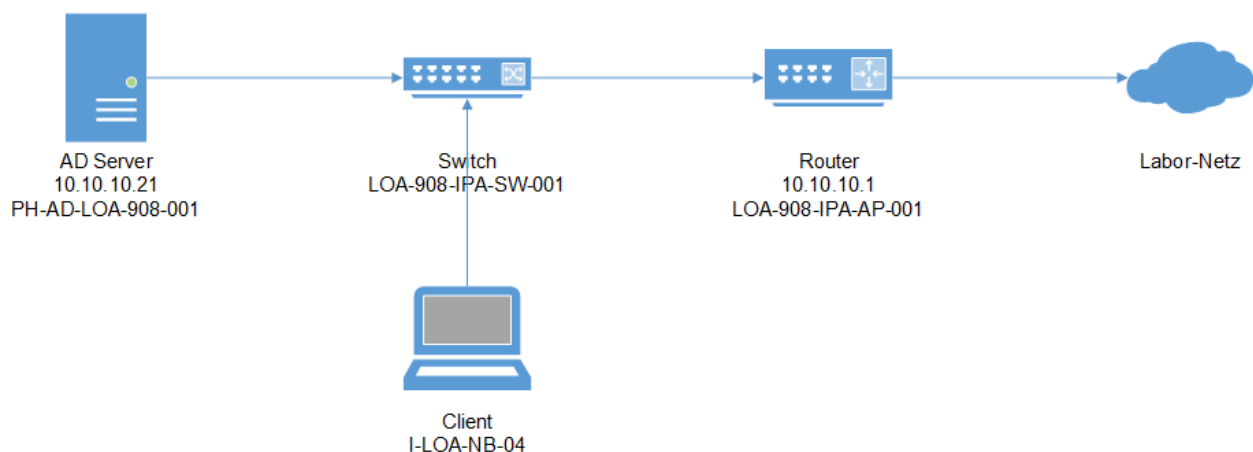


Abbildung 10 Netzwerkschema

12.2. Namenskonzept

Die Technische Fachschule Bern besitzt Zurzeit noch kein Namenskonzept für das Labor Netz. Im Rahmen dieser IPA wird ein Namenskonzept für die zur Verfügung gestellten Geräte (Switch, Server, Router) erstellt. Dieses Namenskonzept wird nur während dieser IPA verwendet.

12.2.1. Server Typ und Funktion

Der Name des Servers setzt sich aus 5 Teilen zusammen:

[Typ] [Funktion] [Standort] [Raum] [Nummer]

Beim Typ wird unterschieden ob es ein Physikalischer Server oder ein Virtueller Server ist.

Typ	Name
Physikalischer Server	PH
Virtueller Server	VM

Tabelle 28 Server Typen

Bei der Funktion wird definiert, welche Aufgabe der Server hat.

Funktion	Name
Backupserver	BACKUP
Domaincontroller	DB
Allgemeine Dienste	DC
Disaster-Recovery	DR
File Zugriff von Zuhause	FILE
Fileserver	FS
Informationssystem	IS
Lizenzserver	LIC
Logserver	LOG
Monitoring Server	MON
Orgamax (Debitoren-Software)	OORGA
Printserver	PRINT
Redmine Server (Ticketsystem)	RM
Veeam (Backup-Software)	VEEAM
Webserver	WEB
Active Directory	AD

Tabelle 29 Server Zwecke

Beim Standort wird das Gebäude bestimmt, in welchem sich das Gerät befindet.

Standort	Gebäude	Name
Lorrainestrasse 3	Hauptgebäude	LOH
Lorrainestrasse 3	Annex	LOA

Tabelle 30 Standort

Die Grossen Gebäude der TF Bern besitzen viele Räume, um da ein Überblick zu behalten in welchem Raum sich das Gerät befindet, werden die Räume nummeriert.

Die Nummer definiert das Gerät, damit mehrere Geräte im gleichen Raum Unterschieden werden können.

Beispiel:

[Typ]	[Funktion]	[Standort]	[Raum]	[Nummer]
↓	↓	↓	↓	↓
PH	AD	LOA	908	001

12.2.2. Benutzer

Bezeichnung	Name
Lehrpersonen	Der 1. & 2. Buchstabe des Nachnamens gefolgt von 1. Buchstabe des Vornamens Bsp. Özcan Altin → ALO
Lernende	1. Klassenname Danach 1. & 2. Buchstabe des Nachnamens gefolgt von 1. & 2. Buchstabe des Vornamens Bsp. Danyyil Luntovsky aus der Klasse IN16 . Danyyil Luntovsky gehört zu der Klasse IN16 . in16luda

Tabelle 31 Namenskonzept Benutzer

12.2.3. Laptops

Alle Arbeitslaptops der Betriebsinformatiker wurden vom Ressort Informatik anhand dieses Schemas benannt.

[Abteilung]	[Standort]	[Geräte Typ]	[Nummer]
↓	↓	↓	↓
Informatik	Lorraine Annex	Notebook	09
↓	↓	↓	↓
I	LOA	NB	09

12.2.4. Switches

[Standort]	[Raum]	[Zweck]	[Gerät]	[Nummer]
↓	↓	↓	↓	↓
LOA	908	IPA	SW	001

12.2.5. Access Points

Für die Access Points wird folgende Namenskonvention verwendet:

[Standort]	[Raum]	[Zweck]	[Gerät]	[Nummer]
↓	↓	↓	↓	↓
LOA	908	IPA	AP	001

12.3. ERM

Das Konzept für die Datenbank. Die Kardinalitäten sind nach der Martin-Notation gezeichnet.

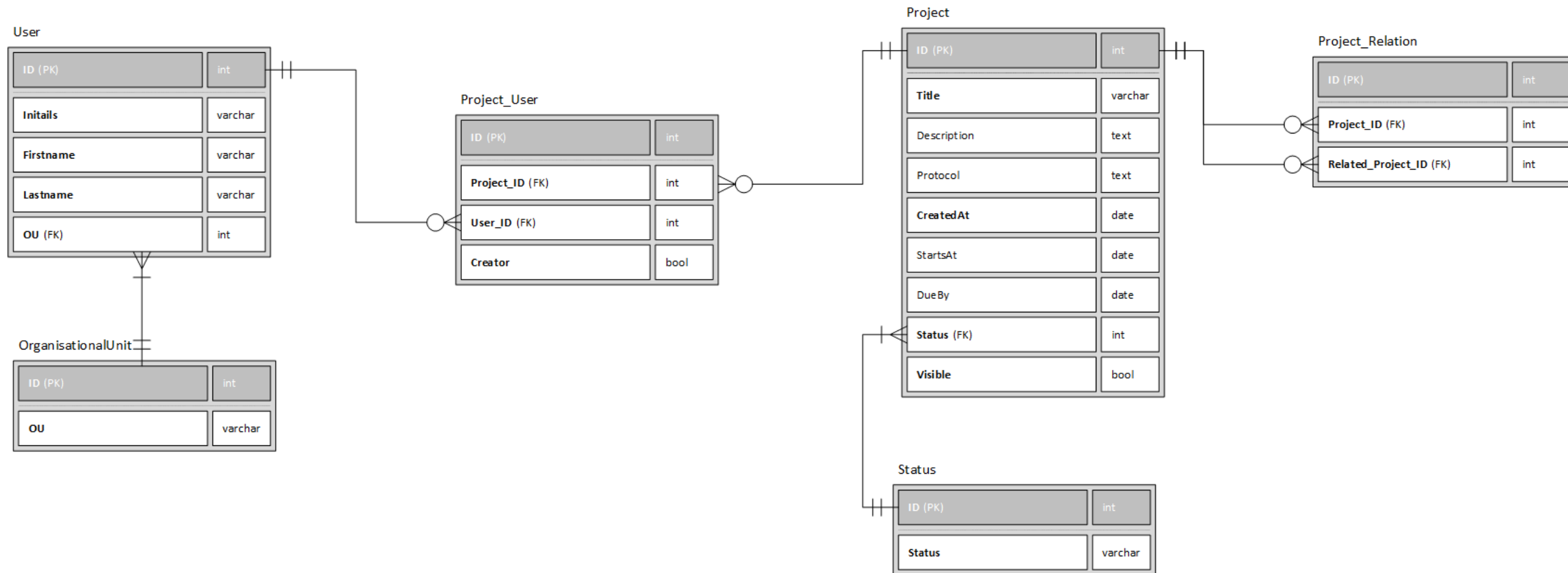


Tabelle 32 Konzept - ERM

12.4. Package Diagramm

12.4.1. Backend

Aus der folgenden Grafik kann man alle zur Verwendung geplanten Packages und Module im Backend entnehmen.

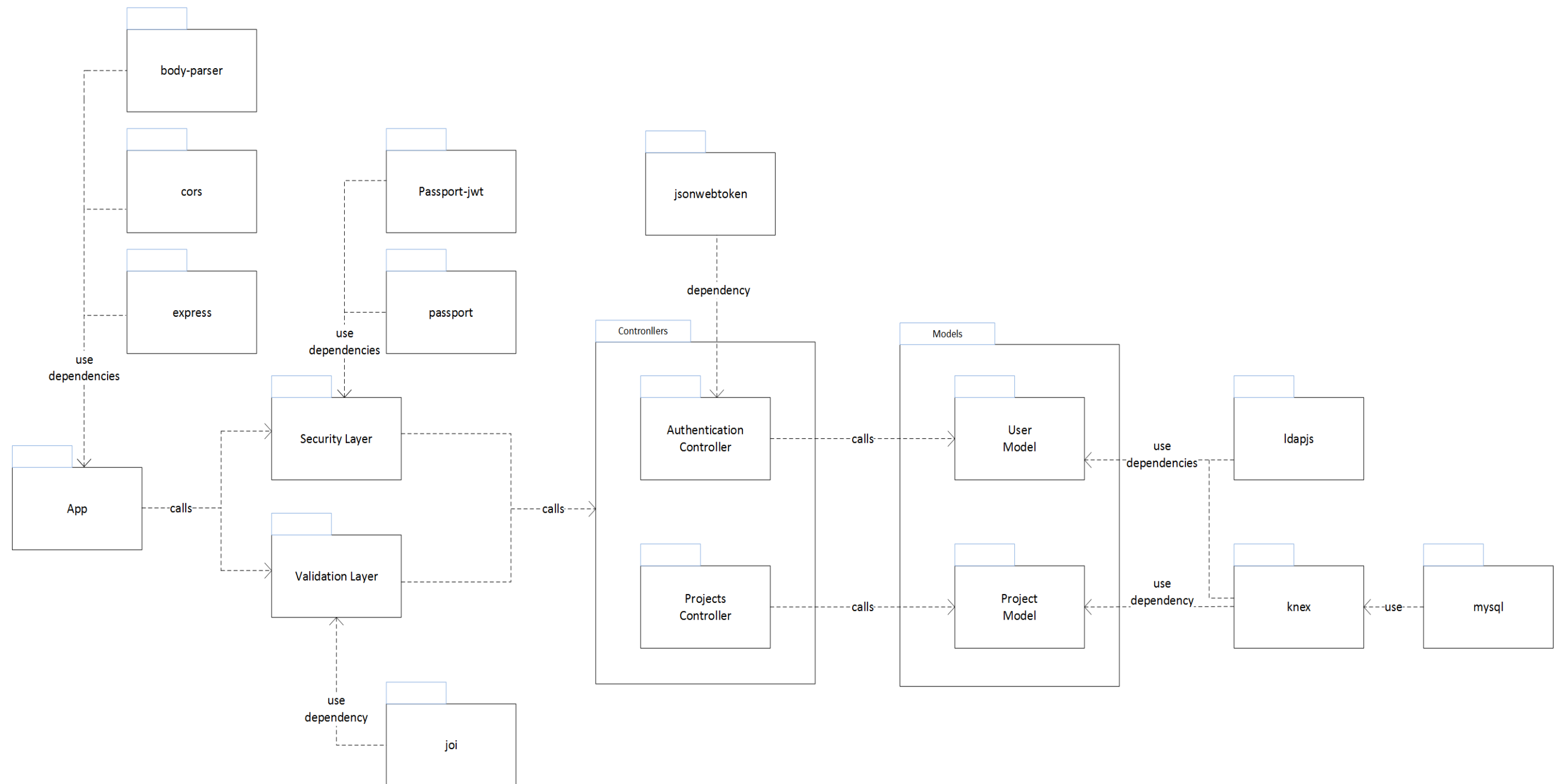


Tabelle 33 Konzept - Package Diagramm Backend

12.4.2. Frontend

Aus der folgenden Grafik kann man alle zur Verwendung geplanten Packages und Module im Frontend entnehmen.

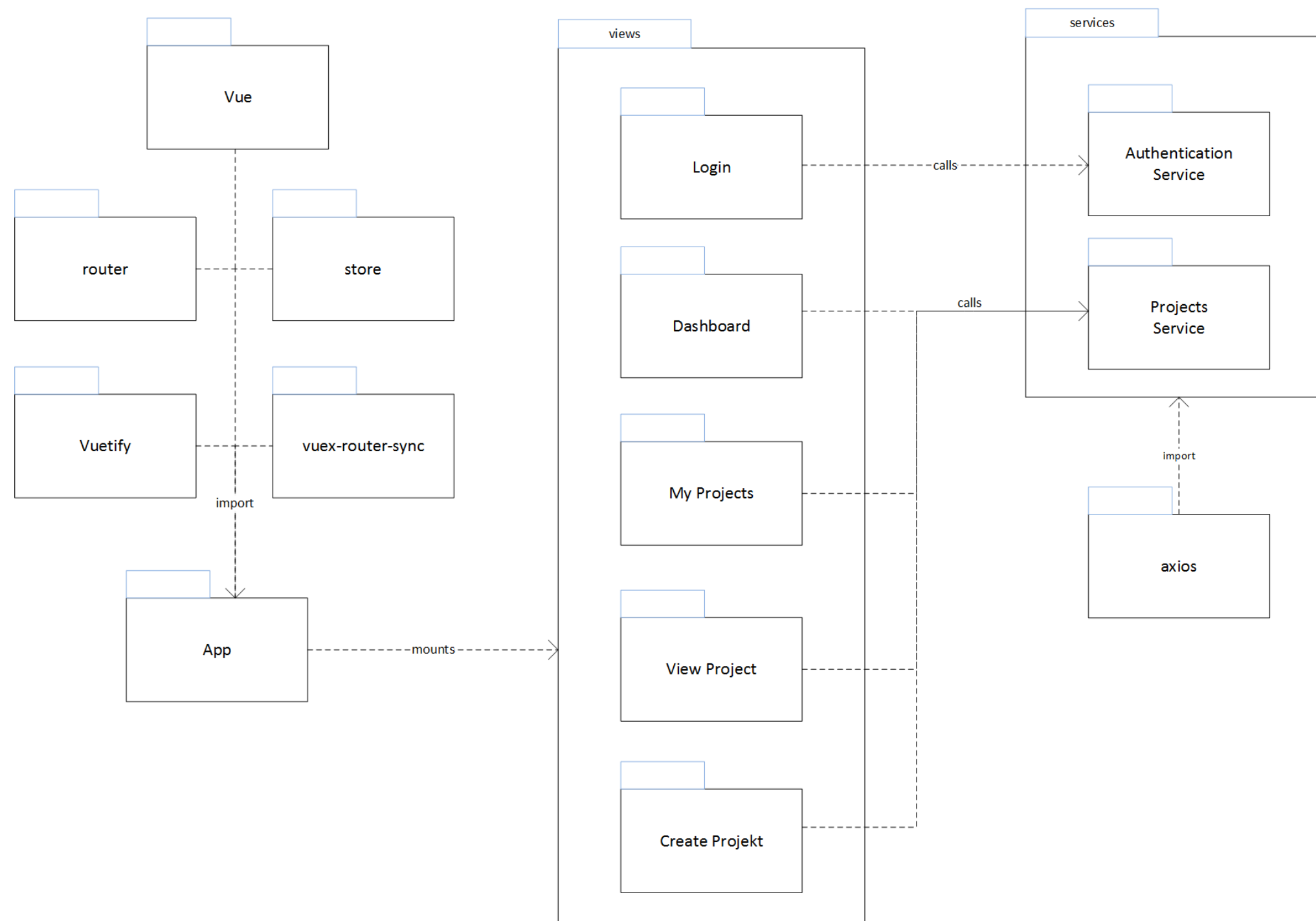


Tabelle 34 Konzept - Package Diagramm Frontend


12.5. Mockup


12.5.1. Login

Das Login ist die erste Anlaufstelle für alle User die neu auf die Applikation zugreifen.

ProjektPortfolio

LOGIN

 Kürzel

 Passwort

Anmelden

Tabelle 35 Mockup Login

Auf dem Dashboard werden alle Projekte angezeigt, die nicht gelöscht wurden. Diese Projekte sind für alle Lehrpersonen ersichtlich.



12.5.3. Projekt Ansicht

Projekt Ansicht

ProjektPortfolio

MENU ▾

PROJEKT ERSTELLEN +

ABMELDEN ↗

◀ ZURÜCK

BEARBEITEN ✎

Titel

Projekt 1

Status

Aktiv

Beschreibung

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Kandidaten

Vorname Nachname

Vorname Nachname

Vorname Nachname

Startdatum

01.01.2019

Kandidaten

Projekt 2

Projekt 3

Enddatum

01.01.2020

Protokoll

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata

Tabelle 37 Mockup Projektansicht

12.5.4. Projekt erstellen

Die Projekterstellung wird in einem Dialogfenster präsentiert, dazu muss man im Header auf «Projekt erstellen» klicken. Die Erstellung beinhaltet 3 Schritte:

1. Die Definition vom Titel, Beschreibung, Start- und Enddatum
2. Die Zuordnung von Anhängigen Projekten und den zum Projekte dazugehörigen Kandidaten.
3. Die Übersicht von den zuvor eingegebenen Daten.

The screenshot shows a web application interface for 'ProjektPortfolio'. At the top, there is a header bar with 'ProjektPortfolio', a 'MENU' dropdown, a 'PROJEKT ERSTELLEN +' button, and an 'ABMELDEN' button with an arrow icon. Below the header, a 'DASHBOARD' section is visible on the left. The main content area is partially obscured by a modal dialog box titled 'Neues Projekt erstellen'. This dialog box features a progress bar at the top with three steps: 'Initialisierung' (marked with a checkmark), 'Anpassung' (marked with a checkmark), and 'Übersicht' (marked with a checkmark). Below the progress bar, there are input fields for 'Projekt Titel', 'Projekt Beschreibung', 'Startdatum', and 'Enddatum'. A blue button labeled 'WEITER >' is located at the bottom right of the dialog box. The background of the application shows a list of projects, each with a title, creator, status, and a vertical progress bar.

Abbildung 11 Konzept Project erstellen

12.6. Testkonzept

Hier werden die Testfälle definiert und das nähere Testvorgehen beschrieben. Das Testing soll Mängel ermitteln und aufdecken sowohl als auch den ersten Kontakt mit einem Benutzer herstellen. Die Durchführung der Testfälle geschieht nach der Realisierung.

12.6.1. Testszenario

Es wird die erste fertiggestellte Version der Applikation getestet. Es müssen alle in den Vorgaben enthaltenen Funktionen vorhanden sein. Das Testverfahren wird von Özcan Altin durchgeführt.

12.6.2. Testrahmen

Zum Testen steht dem Tester der Laptop I-LOA-NB-09, die entwickelte Applikation und die eigentlichen Testfälle zur Verfügung. Von der Applikation aus wird auf das Active Directory PH-AD-LOA-908-001 zugegriffen. Der Nutzer hat ebenfalls Zugriff auf die lokale DB.

Die Tests werden auf folgenden Browsern durchgeführt:

- Firefox Quantum 65.0.2 (64-Bit)
- Google Chrome Version 72.0.3626.119 (64-Bit)

12.6.3. Testvorgehen

Die vordefinierten Testtabellen werden vom Tester ausgefüllt. Es müssen Die Zeilen Tatsächliches Resultat, Kommentar (Falls Test fehlschlägt) und die Fehlerklasse ausgefüllt werden. Um die Eingabe tätigen zu könne verwendet der Benutzer die in den Tabellen ersichtlichen Testdaten.

12.6.4. Testmethode

Die Tests werden anhand der Blackbox Testmethode durchgeführt. Dabei hat der Tester eine zu erfüllende Eingabe und ein erwartetes Resultat.

12.6.5. Testziele

Werden alle Testziele erfolgreich durchgeführt so gilt die Realisierung als abgeschlossen.

12.6.6. Re-Testing

Die Bedingung zur Einleitung des Re-Testings ist, dass ein Mangel entdeckt wird. Nachdem der Fehler vom IPA Kandidaten behoben wurde, wird der unzufrieden stellende Testfall nochmals im Re-Testing durchgeführt.

12.6.7. Fehlerklassen

Klassen ID	Fehlerklassen	Beschreibung
M0	Kein Mangel	Das erwartete sowie das Tatsächliche Resultat stimmen überein.
M1	Leichter Mangel	Grundlegend funktioniert alles, kleinere Mängel ersichtlich
M2	Schwerer Mangel	Die Anforderung wird nur teilweise erfüllt.
M3	Kritischer Mangel	Die Anforderungen werden sehr schlecht bis gar nicht erfüllt.

Tabelle 38 Fehlerklassen

12.6.8. Testtabelle

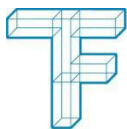
Testfall ID	Die ID ist dazu da um Testfälle zu identifizieren	
Testfall	Beschreibung des Testfalls	
Voraussetzungen	Voraussetzungen die getroffen werden müssen um den Testfall durchzuführen	
Beschreibung	Zu erfüllender Test 1	Zu erfüllender Test 2
Test Schritte	Schritte um Test 1 durchzuführen	Schritte um Test 2 durchzuführen
Test Informationen	Hier werden Angaben preisgegeben um das Testgeschehen zu steuern	
Erwartetes Resultat	Das geplante Resultat	Das geplante Resultat
Tatsächliches Resultat	Das Resultat das vom Tester eingegeben wird	Das Resultat das vom Tester eingegeben wird
Kommentar	Kommentar, falls der Testfall gescheitert ist	Kommentar, falls der Testfall gescheitert ist
Fehlerklasse	Aufgewiesenen Mängel des Testfalls in Fehlerklassen angegeben (M0)	Aufgewiesenen Mängel des Testfalls in Fehlerklassen angegeben (M0)

Tabelle 39 Testtabelle

12.6.9. Testfälle

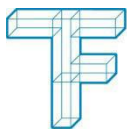
Testfall ID	1				
Testfall	Login Lernender				
Voraussetzungen	1) Webseite läuft 2) Kein User ist angemeldet				
Beschreibung	Keine Eingabe	Richtiger Kürzel & falsches Passwort	Falscher Kürzel & richtiges Passwort	Falscher Kürzel & falsches Passwort	Richtiger Kürzel & richtiges Passwort
Test Schritte		1) Kürzel eingeben 2) Passwort eingeben 3) 'Login' klicken			
Test Informationen		Kürzel: in16luda Passwort: xxxxxxxx	Kürzel: abc Passwort: gibbiX12345	Kürzel: abc Passwort: xxxxxxxx	Kürzel: in16luda Passwort: gibbiX12345
Erwartetes Resultat	Fehlermeldung: Invalide Login Informationen				
Tatsächliches Resultat					
Kommentar					
Fehlerklasse					

Tabelle 40 Testfall 1



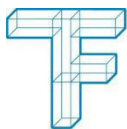
Testfall ID	2			
Testfall	Login Lehrperson			
Voraussetzungen	1) Webseite läuft 2) Kein User ist angemeldet			
Beschreibung	Richtiger Kürzel & falsches Passwort	Falscher Kürzel & richtiges Passwort	Falscher Kürzel & falsches Passwort	Richtiger Kürzel & richtiges Passwort
Test Schritte	1) Kürzel eingeben 2) Passwort eingeben 3) Login drücken			
Test Informationen	Kürzel: alo Passwort: xxxxxxxx	Kürzel: def Passwort: gibbiX12345	Kürzel: def Passwort: xxxxxxxx	Kürzel: alo Passwort: gibbiX12345
Erwartetes Resultat	Fehlermeldung: Invalide Login Informationen			
Tatsächliches Resultat				
Kommentar				
Fehlerklasse				

Tabelle 41 Testfall 2



Testfall ID	3
Testfall	Lernender Projektansicht
Voraussetzungen	1) Webseite läuft 2) Lernender ist eingeloggt 3) Pfad ist '/myProjekts'
Beschreibung	Projektansicht beim Lernenden
Test Schritte	1) Lernender klickt auf ein Projekt
Test Informationen	
Erwartetes Resultat	Der Lernende wird auf die Projektansicht Seite weitergeleitet wo er alle Informationen zum jeweiligen Projekt entnehmen kann.
Tatsächliches Resultat	
Kommentar	
Fehlerklasse	

Tabelle 42 Testfall 3



Testfall ID	4			
Testfall	Projekt Protokoll führen			
Voraussetzungen	1) Webseite läuft 2) User ist eingeloggt 3) Pfad: '/project/1'			
Beschreibung	Keine Eingabe	Eingabe von über 30'000 Zeichen	Eingabe Script Tags Injektion	Eingabe von max. 30'000 Zeichen
Test Schritte		1) Text Eingabe 2) Speichern drücken		
Test Informationen			<script> alert('Test') <script>	
Erwartetes Resultat		Fehlermeldung: Eingabe zu lang	Das Protokoll wird ohne zusätzlichen Aktionen abgespeichert. Die Seite lädt neu und die Änderungen sind ersichtlich	
Tatsächliches Resultat				
Kommentar				
Fehlerklasse				

Tabelle 43 Testfall 4



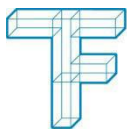
Testfall ID	5		
Testfall	Projekt Zugriffsrechte Lernende		
Voraussetzungen	1) Webseite läuft 2) Lernender ist eingeloggt 3) Pfad ist '/project/3'		
Beschreibung	Lernender versucht auf fremdes Projekt zuzugreifen.	Lernender versucht auf alle Projekte zuzugreifen.	Lernender versucht auf ein ihm zugeordnetes Projekt zuzugreifen.
Test Schritte	1) In die URL Zeile des Browsers klicken 2) Die URL zu '/project/2' ändern 3) Enter drücken	1) In die URL Zeile des Browsers klicken 2) Die URL zu '/' ändern 3) Enter drücken	1) Bei 'Abhängigen Projekten' auf das 'Projekt 1' klicken
Test Informationen	-		
Erwartetes Resultat	Dem Lernenden soll keinerlei Information über das Projekt ersichtlich sein.	Der Lernende soll nach '/myProjekts' weitergeleitet werden.	Der Lernende soll auf Projektansicht vom Projekt 1 weitergeleitet werden.
Tatsächliches Resultat			
Kommentar			
Fehlerklasse			

Tabelle 44 Testfall 5



Testfall ID	6		
Testfall	Lehrperson Zugriffsrechte		
Voraussetzungen	1) Webseite läuft 2) Lehrperson ist eingeloggt		
Beschreibung	Alle Projekte anzeigen lassen	Nur der Lehrperson ihre Projekte anzeigen lassen	Das Projekterstellungs-Dialog wird aufgerufen
Test Schritte	1) In der Navigation auf 'Dashboard' klicken	1) In der Navigation auf 'Meine Projekte' klicken	1) In der Navigation auf 'Erstellen' klicken
Test Informationen	-		
Erwartetes Resultat	Alle bislang erstellten und nicht gelöschten Projekte werden angezeigt.	Nur Projekte, wo man involviert ist, werden angezeigt.	Ein Dialog-Fenster erscheint, mit dem Titel 'Projekt erstellen'
Tatsächliches Resultat			
Kommentar			
Fehlerklasse			

Tabelle 45 Testfall 6

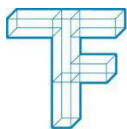


Testfall ID	7			
Testfall	Projekt erstellen Initialisierung (Teil 1)			
Voraussetzungen	1) Webseite läuft 2) Lehrperson ist eingeloggt 3) Klick auf 'Projekt erstellen' ist getätigt			
Beschreibung	Es wird kein Titel eingegeben	Es wird ein zu langer Title eingegeben	Es wird ein valider Titel eingegeben	Die Felder Beschreibung & Start- sowie Enddatum werde (nicht) ausgefüllt
Test Schritte	1) In das Feld 'Title' klicken 2) Keine Eingabe 3) Klick 'Weiter'	1) In das Feld 'Title' klicken 2) Eingabe über 100 Zeichen 3) Klick 'Weiter'	1) In das Feld 'Title' klicken 2) Eingabe weniger als 100 Zeichen 3) Klick 'Weiter'	Feld Beschreibung ausfüllen
				Feld Beschreibung nicht ausfüllen
Test Informationen	-		Titel: Projekt10	Beschreibung: Test
Erwartetes Resultat	Der zweite Schritt wird nicht eingeleitet. Fehlermeldung: 'Titel wird benötigt'	Der zweite Schritt wird nicht eingeleitet. Fehlermeldung: 'Titel zu lang'	Der zweite Schritt wird eingeleitet. Die nächste Seite der Projekt-erstellung ist zusehen.	
Tatsächliches Resultat				
Kommentar				
Fehlerklasse				

Tabelle 46 Testfall 7

Testfall ID	8			
Testfall	Projekt erstellen User & Projektzuordnung (Teil 2 / 3)			
Voraussetzungen	1) Webseite läuft 2) Lehrperson ist eingeloggt 3) Erster Schritt der Projekterstellung ist abgeschlossen			
Beschreibung	Kein Eingabe wird getätigt	Nur Kandidaten auswählen	Nur abhängige Projekte auswählen	Kandidaten & abhängige Projekte auswählen
Test Schritte	1) Weiter klicken	1) In Feld 'Kandidaten' klicken 2) Kandidaten auswählen 3) Weiter klicken	1) In Feld 'Abhängigkeiten' klicken 2) Abhängigkeiten auswählen 3) Weiter klicken	1) In Feld 'Abhängigkeiten' klicken 2) Abhängigkeiten auswählen 3) In Feld 'Kandidaten' klicken 4) Kandidaten auswählen 5) Weiter klicken
Test Informationen		Kandidat: 'Danyyil Luntovsky'	Abhängiges Projekt: 'Projekt 1'	Kandidat: 'Danyyil Luntovsky' Abhängiges Projekt: 'Projekt 1'
Erwartetes Resultat	Der dritte Schritt wird eingeleitet. Die nächste Seite der Projekterstellung ist zusehen.			
Tatsächliches Resultat				
Kommentar				
Fehlerklasse				

Tabelle 47 Testfall 8



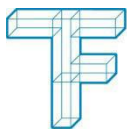
Testfall ID	9		
Testfall	Projekt erstellen, Übersicht (Teil 3 / 3)		
Voraussetzungen	1) Webseite läuft 2) Lehrperson ist eingeloggt 3) Zweiter Schritt der Projekterstellung ist abgeschlossen		
Beschreibung	Kontrollieren ob alle Daten des 2. Schrittes eingetragen sind	Kontrollieren ob alle Daten des 1. Schrittes eingetragen sind	Kontrollieren ob alle eingetragenen Daten mit dem Schritt 3 übereinstimmen
Test Schritte	1) Einmal 'Zurück' klicken	1) Zwei Mal 'Zurück' klicken	1) Sicherstellen das alle eingetragenen Informationen richtig dargestellt sind (Der Name vom Ersteller ist bei den Kandidaten ersichtlich) 2) 'Projekt erstellen' klicken
Test Informationen	-		
Erwartetes Resultat	Die zweite Seite samt allen eingegebenen Informationen wird sichtbar.	Die erste Seite samt allen eingegebenen Informationen wird sichtbar.	Das Dialog-Fenster schliesst sich. Eine Meldung erscheint, dass das Projekt erstellt wurde.
Tatsächliches Resultat			
Kommentar			
Fehlerklasse			

Tabelle 48 Testfall 9



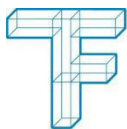
Testfall ID	10	
Testfall	Erstellung des Projekts kontrollieren (Teil 1)	
Voraussetzungen	1) Webseite läuft 2) Lehrperson ist eingeloggt 3) Projekterstellung wurde durchgeführt	
Beschreibung	Das Projekt ist von 'Meine Projekten' aus sichtbar	Das Projekt ist vom 'Dashboard' aus sichtbar
Test Schritte	1) Navigiere zu 'Meine Projekte' i) Der Ersteller & der Status, der Projektname und das Erstelldatum sind ersichtlich. 2) Klicke auf 'Projekt 10'	1) Navigiere zum 'Dashboard' i) Der Ersteller & der Status, der Projektname und das Erstelldatum sind ersichtlich. 2) Klicke auf 'Projekt 10'
Test Informationen	-	
Erwartetes Resultat	Das neu erstellte Projekt zeigt alle bei der Projekterstellung eingegebenen Informationen.	
Tatsächliches Resultat		
Kommentar		
Fehlerklasse		

Tabelle 49 Testfall 10



Testfall ID	11
Testfall	Erstellung des Projekts kontrollieren (Teil 2)
Voraussetzungen	1) Webseite läuft 2) Lernender ist eingeloggt 3) Projekterstellung wurde durchgeführt
Beschreibung	Es wird vom Account des Lernenden getestet ob das erstellte Projekt mit allen Daten ersichtlich ist
Test Schritte	1) Navigiere zu 'Meine Projekte' 2) Klicke auf 'Projekt 10'
Test Informationen	Kürzel: in16luda Passwort: gibbiX12345
Erwartetes Resultat	Das neu erstellte Projekt zeigt alle bei der Projekterstellung eingegebenen Informationen.
Tatsächliches Resultat	
Kommentar	
Fehlerklasse	

Tabelle 50 Testfall 10



Testfall ID	12	
Testfall	Projekt bearbeiten (Teil 1 / 3)	
Voraussetzungen	1) Webseite läuft 2) Lehrperson ist eingeloggt 3) Projekterstellung wurde erfolgreich durchgeführt	
Beschreibung	Es ist der Lehrperson visuell ersichtlich dass das Projekt bearbeitet wird.	Es werden alle geränderten Daten nicht übernommen.
Test Schritte	1) Klicke auf 'Projekt 10' 2) Klicke auf 'Bearbeiten'	1) Klicke auf 'Projekt 10' 2) Klicke auf 'Bearbeiten' 3) Ändere den Titel 4) Danach klicke auf 'Abbrechen'
Test Informationen		Titel: Projekt 11
Erwartetes Resultat	Alle Informationen des Projektes werden zu Feldern die man bearbeiten kann.	Es werden keine Änderungen vorgenommen. Man sieht die Projektdetails jetzt wieder in Textform.
Tatsächliches Resultat		
Kommentar		
Fehlerklasse		

Tabelle 51 Testfall 12



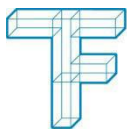
Testfall ID	13
Testfall	Projekt bearbeiten (Teil 2 / 3)
Voraussetzungen	1) Webseite läuft 2) Lehrperson ist eingeloggt 3) Projekterstellung wurde erfolgreich durchgeführt 4) «Projekt 10» ist ausgewählt 5) 'Bearbeiten' wurde geklickt
Beschreibung	Das erstellte Projekt wird bearbeitet.
Test Schritte	1) Gebe neuen Titel ein 2) Gebe neue Beschreibung ein. 3) Wähle Status 'Aktiv' 4) Ändere Protokoll 5) Klicke Speichern
Test Informationen	Titel: Projekt 10 Update Beschreibung: Test Beschreibung Status: Aktiv Protokoll: Testeintrag
Erwartetes Resultat	Die Seite wird neu geladen. Alle geänderten Informationen sind ersichtlich.
Tatsächliches Resultat	
Kommentar	
Fehlerklasse	

Tabelle 52 Testfall 13



Testfall ID	14	
Testfall	Projekt bearbeiten (Teil 3 / 4)	
Voraussetzungen	1) Webseite läuft 2) Lehrperson ist eingeloggt 3) Projekterstellung wurde erfolgreich durchgeführt 4) «Projekt 10» ist ausgewählt 5) 'Bearbeiten' wurde geklickt	
Beschreibung	Der Projekttitel wird gelöscht	Ein zu langer Projekttitel wird eingegeben
Test Schritte	1) Projekttitel im Eingabefeld löschen. 2) 'Speichern' klicken	1) Projekttitel über 100 Zeichen eingeben 2) 'Speichern' klicken
Test Informationen		
Erwartetes Resultat	Fehlermeldung: Titel zu kurz Die Seite bleibt im Bearbeitungszustand	Fehlermeldung: Titel zu lang Die Seite bleibt im Bearbeitungszustand
Tatsächliches Resultat		
Kommentar		
Fehlerklasse		

Tabelle 53 Testfall 14



Testfall ID	15	
Testfall	Projekt löschen	
Voraussetzungen	1) Webseite läuft 2) Lehrperson ist eingeloggt 3) Projekterstellung wurde erfolgreich durchgeführt 4) «Projekt 10» ist ausgewählt	
Beschreibung	Die Löschung des Projekts wird abgebrochen.	Die Löschung des Projekts wird durchgeführt.
Test Schritte	1) Klicke auf die drei Punkte (Menu) oben rechts 2) Klicke auf 'Projekt löschen' 3) Klicke auf 'Abbrechen'	1) Klicke auf die drei Punkte (Menu) oben rechts 2) Klicke auf 'Projekt löschen' 3) Klicke auf 'Projekt löschen'
Test Informationen	Kürzel: alo Passwort: gibbiX12345	
Erwartetes Resultat	Der Dialog mit der Warnung Schliesst sich. Das Projekt bleibt vorhanden.	Der Dialog mit der Warnung Schliesst sich. Das Projekt wird in der Datenbank als gelöscht markiert und ist auf der Applikation nicht mehr ersichtlich. Der User wird zum Dashboard weitergeleitet.
Tatsächliches Resultat		
Kommentar		
Fehlerklasse		

Tabelle 54 Testfall 15



Testfall ID	16	
Testfall	Sich von der Webseite abmelden	
Voraussetzungen	1) Webseite läuft 2) User ist eingeloggt	
Beschreibung	Die Lehrperson meldet sich ab.	Der Lernende melden sich ab.
Test Schritte	1) In der Navigationszeile auf 'Abmelden' klicken	
Test Informationen	Kürzel: alo Passwort: gibbiX12345	Kürzel: in16luda Passwort: gibbiX12345
Erwartetes Resultat	Der User ist abgemeldet. Er wird zum Login weitergeleitet.	
Tatsächliches Resultat		
Kommentar		
Fehlerklasse		

Tabelle 55 Testfall 16

13. Realisierung

13.1. Ordnerstruktur

Als Vorbereitung wurde die Projektumgebung eingerichtet. Bei diesem Vorgehen wurde das MVC-Pattern über die ganze Applikation angewandt. Die Ordnerstruktur wurde bis zu einem bestimmten Grad granuliert.

Befehl: vue create client

Die Grundstruktur im Frontend, wurde von Vue.js erstellt und beibehalten.

Im Rahmen der IPA wurde nur im `src` Ordner entwickelt.

<ul style="list-style-type: none"> client <ul style="list-style-type: none"> node_modules public src <ul style="list-style-type: none"> assets components plugins services store views App.vue main.js router.js .gitignore babel.config.js package-lock.json package.json README.md 	<p>assets Bilder & Medieninhalte</p> <p>components Einzelne Komponenten Bsp. Herder</p> <p>plugins Erweiterungen, in diesem Fall Vuetify</p> <p>services API Anfragen (Axios)</p> <p>store Vuex Store (Definition von App-globalen Werten)</p> <p>views Ansichten, ganze Seiten Bsp. Login</p> <p>App.vue Das Haupt-Vue-File, Zusammenführung aller Elemente</p> <p>main.js Definition der Vue Instanz und dessen verwendete Add-ons (router, store)</p> <p>Router.js Dient zur Verlinkung zwischen den Views</p>
--	--

Abbildung 12 Ordnerstruktur
Frontend

Befehl: `npm init -f`

Die Backendstruktur wurde vollständig (bis auf `node_modules`) vom Kandidaten erstellt.

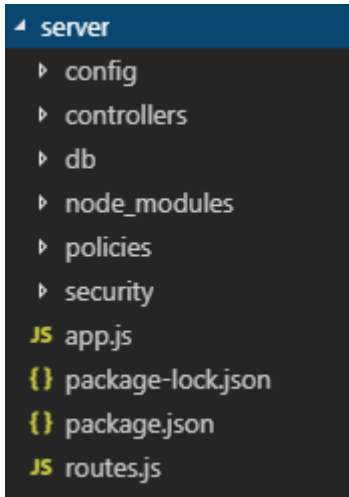


Abbildung 13 Ordnerstruktur
Backend

config

Globale Einstellungen wie der Port auf der die Applikation läuft und Verbindungsdaten (DB) werden hier definiert

controllers

Die Controller schränken die Zugriffe vom Frontend ein und unterteilen sie in verschiedene Funktionen (get, post...)

db

Hier befinden sich die Models, die auf die Datenbank und auf den AD anfragen versenden.

node_modules

Die installierten dependencies werden hier gelagert (durch node organisiert)

Policies

Regeln die, die Anfragen vom Frontend durchlaufen bevor sie zu den Controllern weitergeleitet werden

security

Hier wird der vom Frontend erhaltene Token validiert

app.js

Die Hauptdatei, sie importiert express, body-parser, longjohn und startet den Server

package.json

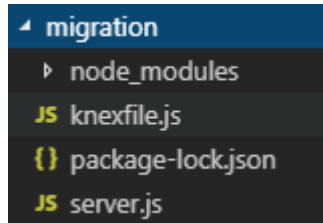
Hier sind alle installierten Pakete samt ihrer Version ersichtlich. Ebenfalls wird hier das Start File (App.js) definiert

routes.js

App.js leitet alle Anfragen an routes weiter, routes daraufhin schreibt vor welchen Zustand die Anfrage durchlaufen soll (policy oder direkt zum Controller)

Befehl: `npm init -f`

Das Migration Skript hat eine simple Ordnerstruktur und wird wie folgt definiert:



knexfile.js

Hier befindet sich die DB Konfiguration.

Server.js

Das eigentliche Skript, dass alle Funktionen tätigt.

Abbildung 14 Ordnerstruktur

Migration Skript

Wie bereits erwähnt spielen die Komponenten im MVC-Pattern zusammen um schlussendlich die gewollte Information dem End User im Frontend darzustellen.

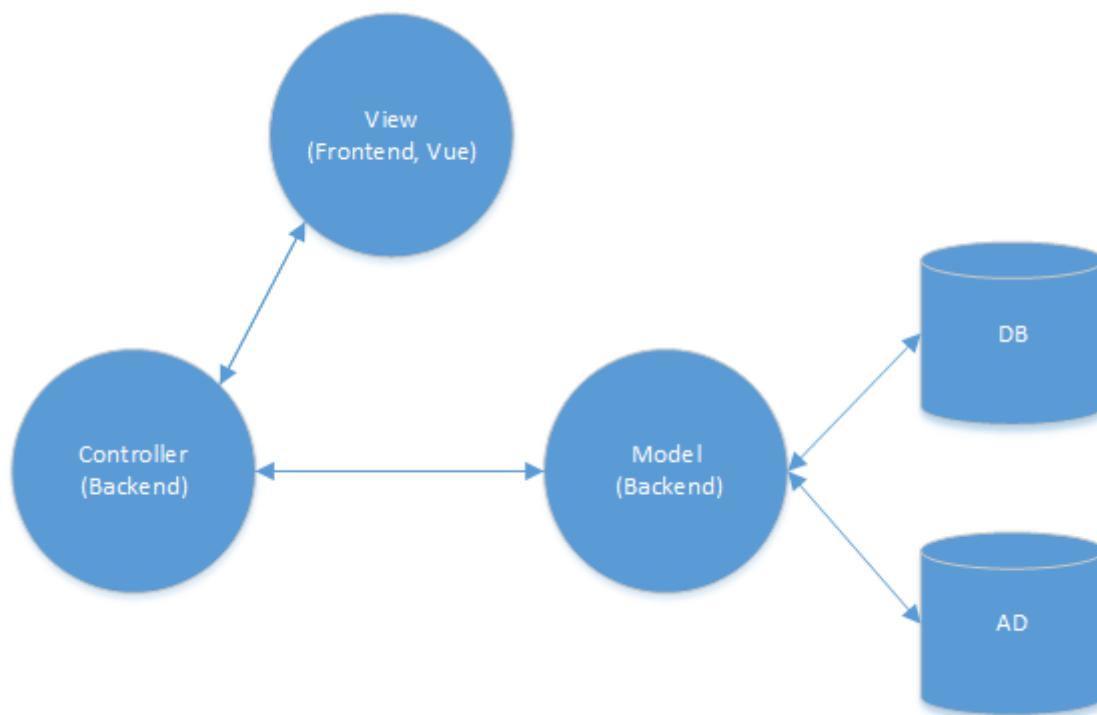


Abbildung 15 MVC-Pattern

13.2. Datenbank

Zur Datenbankerstellung werden zwei Skripte angelegt. Das *create.sql* & *insert.sql* Skript. Das *create.sql* erstellt die eigentliche Datenbank und den User, der im späteren Verlauf auf die DB zugreift. Beim *insert.sql* werden die vordefinierten Stati (Neu, Aktiv, Beendet) in die Tabelle «Status» eingelesen.

Im Vergleich zum Konzept haben sich ein paar Sachen geändert. Beim Project Table wurden die StartsAt und DueBy Attribute gelöscht, weil die Zeit nicht reichte diese Funktionalitäten einzufügen.

Ebenfalls wurde die Tabelle OrganisationalUnit durch die zwei Tabellen OUTop (Lehrperson oder Lernender) und OUSub (Klasse zB. in16) ersetzt, um die Redundanzen zu beseitigen.

Das produktive ERM nach Martin-Notation

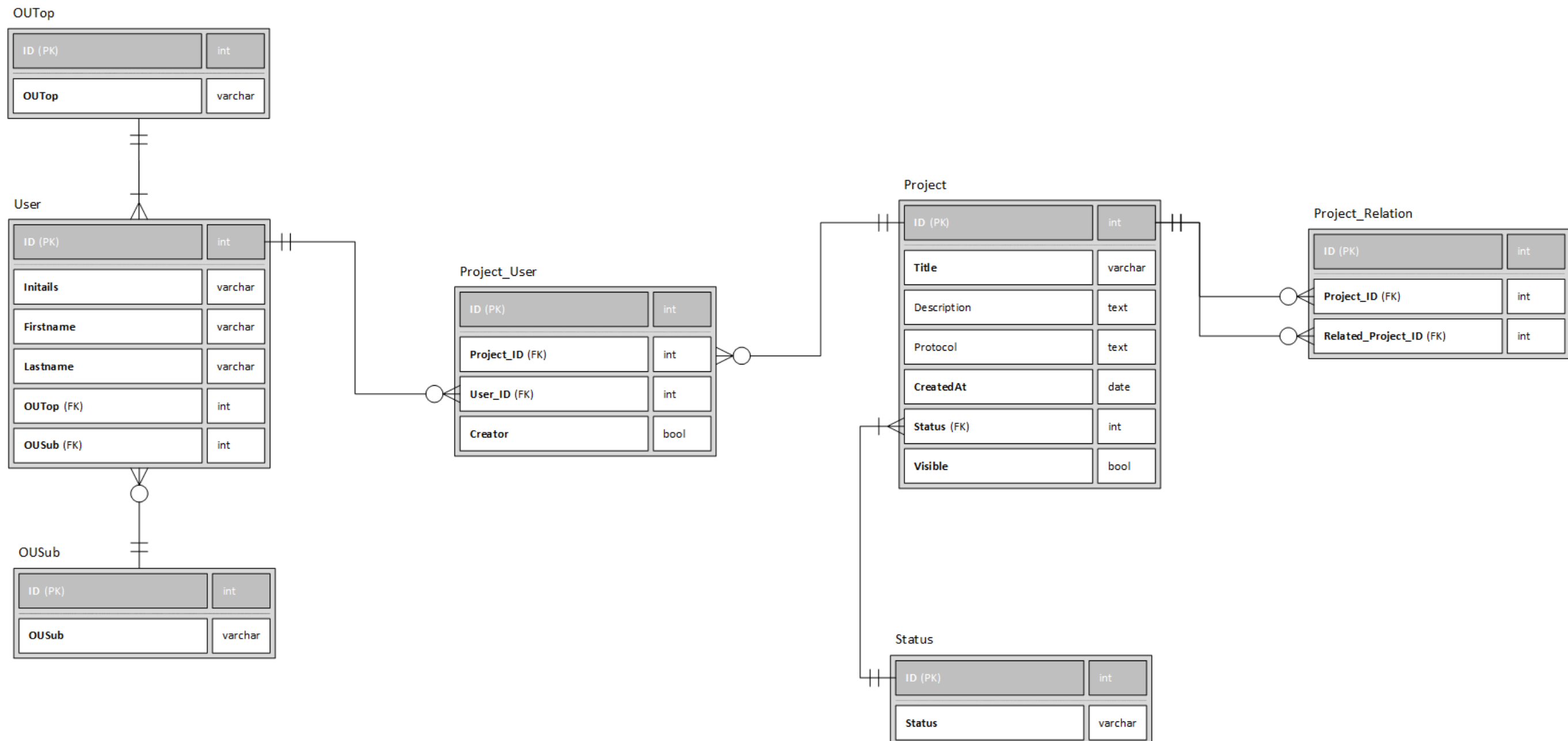


Abbildung 16 Realisierung ERM

13.3. Package Diagram

13.3.1. Backend

Das produktive Projekt verwendet folgende Dependencies im Backend

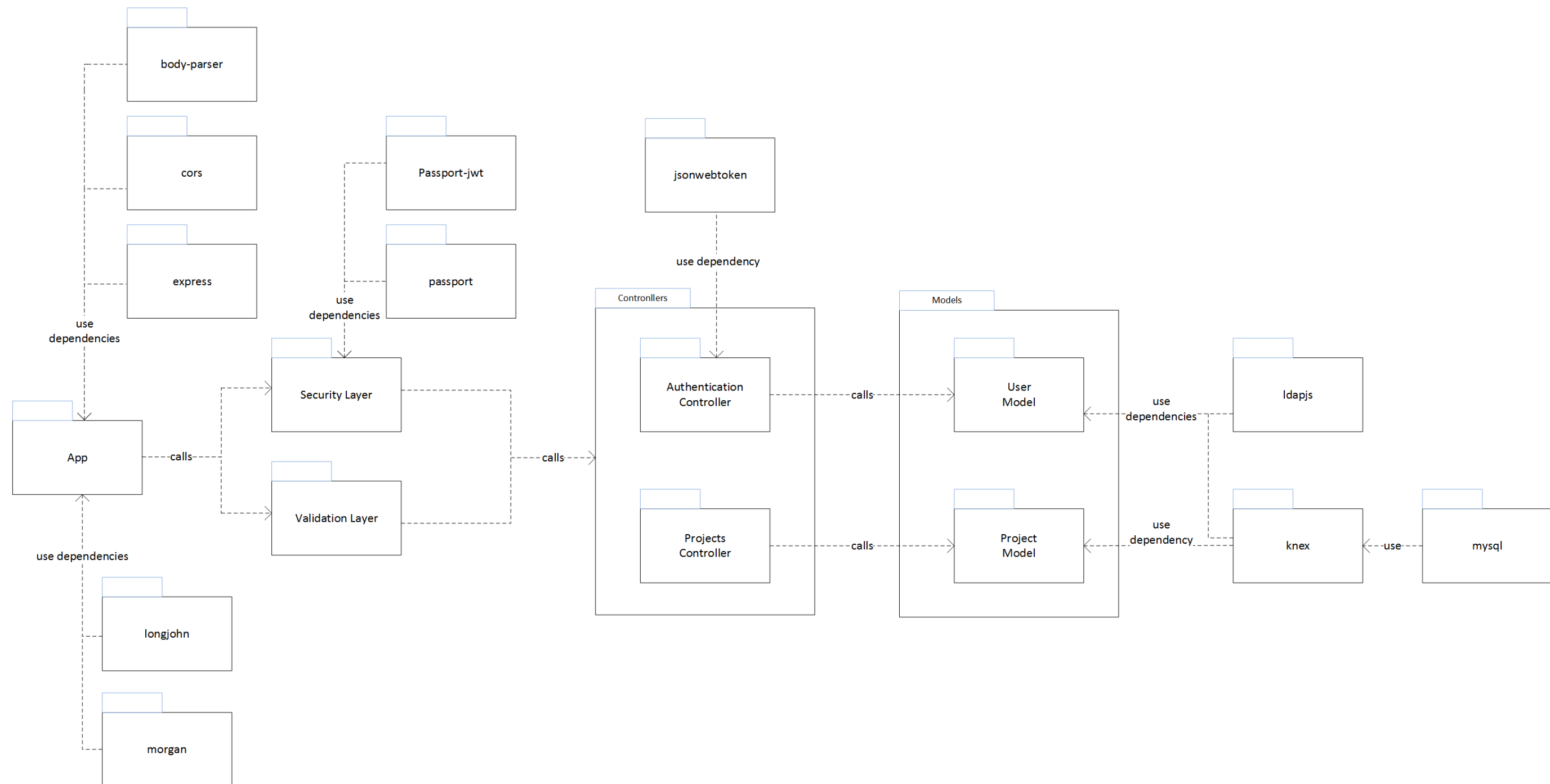


Abbildung 17 Backend Package Diagram Final

13.3.2. Frontend

Das produktive Projekt verwendet folgende Dependencies im Frontend

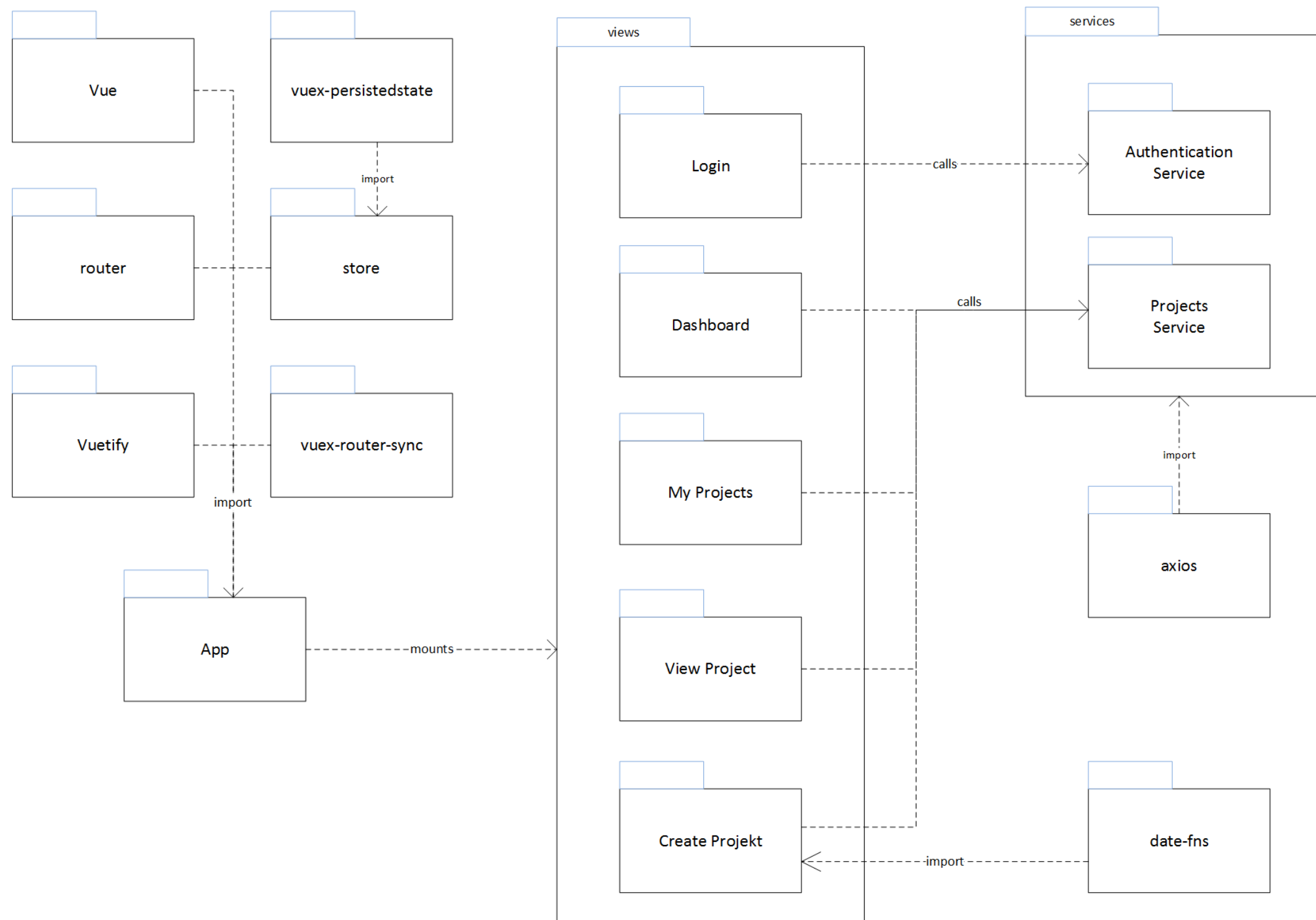


Abbildung 18 Frontend Package Diagram Final

13.4. Migration Skript

Das Migration Skript dient als eine grundlegende Basis für das Projektportfolio. Es migriert alle Benutzer in den Organisational Units: *Lernende* und *Lehrpersonen*. Im Anschluss werden die User in die (jetzt noch) lokale Datenbank abgelegt. Es werden nur User migriert, die noch nicht in der Tabelle «User» eingetragen sind.

Das Labor-Netz, in dem der AD haust, ist noch in der Entwicklungsphase. Dies gab mir den Grund das Skript flexibel zu gestalten. Dementsprechend können weitere OU's dem Skript hinzugefügt werden, um den Migrationsprozess in einem grösseren Rahmen durchzuführen. Das hinzufügen der OU's ist nur auf einen Array beschränkt und kann so nach Belieben ergänzt werden. (Es ist zu beachten, dass Administratoren Benutzer in der Applikation unter der OU *Lehrpersonen* abgelegt werden müssen).

Der Aufbau des AD's sieht wie folgt aus:

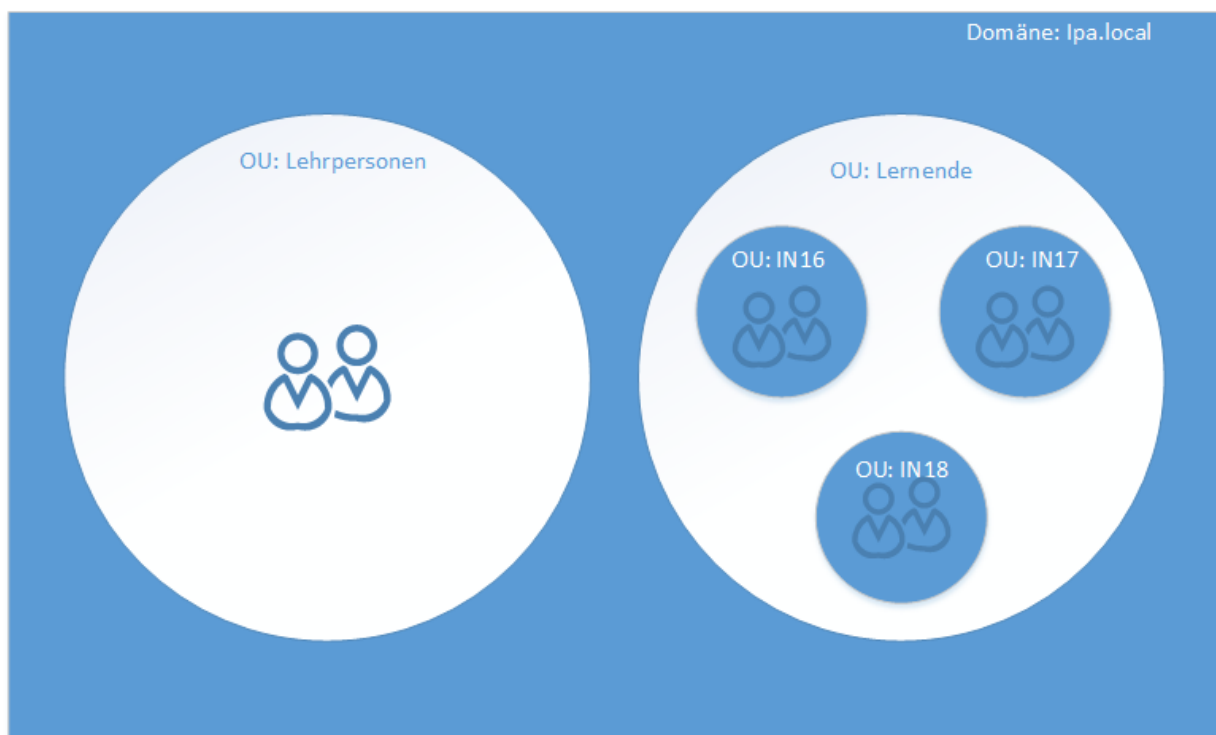


Abbildung 19 Ralisierung AD Aufbau

Diese Funktion ist zuständig für das abrufen und formatieren der User vom Active Directory. Als Parameter wird die Organisational Unit übergeben (Lehrpersonen, Lernende). Jede OU wird einzeln durchsucht und nach den Usern gefiltert. Es werden die in *OPTS* definierten Attribute zurückgegeben (Vor – Nachname, DN und die interne Emailadresse). Die zurückgegebenen Informationen werden demnach auf die DB Tabelle *User* zugeschnitten. Beim Übertragungsschluss wird der Array mit den Usern ausgegeben.

```
const OPTS = {
  filter: '(objectclass=user)',
  scope: 'sub',
  attributes: ['givenName', 'sn', 'userPrincipalName', 'dn'],
}
/**
 * @param {String}
 * @returns {Array}
 */
function getOUUsers(ou) {
  return new Promise(function(resolve, reject) {
    //define users array
    let users = []
    CLIENT.search(`OU=${ou},DC=ipa,DC=local`, OPTS, function(err, res) {
      res.on('searchEntry', function(entry) {
        //short domainlogon name until at-sign (in16test@ipa.local > in16test)
        initialsToken = entry.object.userPrincipalName.substring(0,
entry.object.userPrincipalName.indexOf('@'))
        //cut DN name to split OU information
        ouToken = entry.object.dn.split(",").slice(1, -2)
        ouTop = ouToken[ouToken.length-1].substring(3)
        ouSub = ouToken.slice(0, -1).toString().substring(3)
        //Populate array users with fetched AD data
        users.push({
          Firstname: entry.object.givenName,
          Lastname: entry.object.sn,
          Initials: initialsToken,
          OUTop: ouTop,
          OUSub: ouSub
        })
      })
      //on error callback
      res.on('error', function(err) {
        reject(`ERROR: An error occurred while fetching data from the '${ou}' organisational
unit: ${err}`)
      })
      //on last request
      res.on('end', function(result) {
        console.log(`LOG: Fetching users from OU '${ou}' successful!`)
      })
    })
  })
}
```

```

        //Promise succeeded, return users array from particular OU
        resolve(users)
        console.log('done fetching AD')
    })
})
}

```

Die AD Informationen werden wie folgt zugeschnitten:

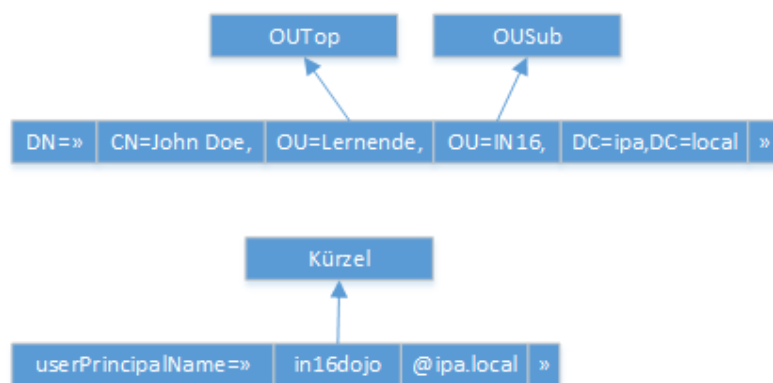


Abbildung 20 DN-Aufteilung

Die oben beschriebene Funktion wird von der folgenden aufgerufen. Dies geschieht zwei Mal, für den Wert *Lernende* und den Wert *Lehrpersonen* im Array *ORGANISATIONAL_UNITS*. Dieser Array kann nach Belieben erweitert werden. Die Benutzer, die von *getOUUsers()* empfangen werden, werden in den Array *allADUsers* gepusht. Der Array wird zu einem einzelnen geformt und ausgegeben.

```

async function getAllOUUsers() {
    //Names of the OU's on the AD server
    const ORGANISATIONAL_UNITS = ['Lernende', 'Lehrpersonen']
    let allADUsers = []
    for(let i = 0; i < ORGANISATIONAL_UNITS.length; i++){
        allADUsers.push((await getOUUsers(ORGANISATIONAL_UNITS[i])))
    }
    CLIENT.destroy()
    //kopiert von https://stackoverflow.com/questions/10865025/merge-flatten-an-array-of-arrays-in-javascript
    allADUsers = [].concat.apply([], allADUsers);
    return allADUsers
}

```

Als nächstes werden alle Benutzer mit der Funktion *getDBUsers()* der Datenbank entnommen und an die nächste Funktion verabreicht.

Die darauffolgende Funktion vergleicht die User Kürzel vom AD und der DB, sollte ein Benutzer nicht in der DB sein, wird er in den Array *comparedUsers* bewegt. Sollte die DB leer sein, werden alle AD Users ausgegeben. Sonst werden die verglichenen User von dieser Funktion ausgegeben.

Der letzte Schritt ist die eigentliche Migration der verglichenen User. Bei der Migration müssen die DB Tabellen *outop*, *ousub* und *user* befüllt werden. Die Funktion erwartet die zu migrierenden User (Array) als Parameter. In einer Schleife wird geprüft ob die Tabelle *outop* die Organisational Unit (z.B. Lernende) schon enthält, ist das der Fall, wird die ID vom jeweiligen Datensatz in *ouTopId* gespeichert, sonst wird ein leerer Array in die Variable gespeichert.

Dann wird abgefragt ob in *ouTopId* ein Wert hinterlegt ist, wenn nicht, wird die OU die nicht in der DB vorhanden ist eingefügt. Sollte die OU bereits in der DB vorhanden sein, wird deren ID in den User Array geschrieben. Der gleiche Prozess wird ebenfalls für *ousub* durchgeführt.

Im Endstand der for-Loop sollte der Array mit den Usern, die korrekten Fremdschlüssel der jeweiligen OU aufweisen. Bei den Lehrpersonen, die zu keiner Klasse gehören wird der Wert *null* eingefügt. Zuletzt werden dann die User im Array, in die DB Tabelle *user* eingefügt.

```
/**
 * @param {Array} users
 */
async function migrateUsers(users) {
  for(let i = 0; i < users.length; i++){
    let ouTopId = await KEX('OUTop').select('ID').limit(1)
      .where('OUTop', users[i].OUTop)
    let ouSubId = await KEX('OUSub').select().limit(1)
      .where('OUSub', users[i].OUSub)
    if(ouTopId <= 0){
      ouTopId = await KEX('OUTop').insert({
        OUTop: users[i].OUTop
      })
      users[i].OUTop = ouTopId[0]
    }else{
      users[i].OUTop = ouTopId[0].ID
    }
    if(ouSubId <= 0){
```



```
    if(users[i].OUSub !== ''){  
      ouSubId = await KNEW('OUSub').insert({  
        OUSub: users[i].OUSub  
      })  
      users[i].OUSub = ouSubId[0]  
    }else{  
      users[i].OUSub = null  
    }  
  }else{  
    users[i].OUSub = ouSubId[0].ID  
  }  
}  
await KNEW('user').insert(users)  
console.log('Migration successful')  
process.exit(0)  
}
```

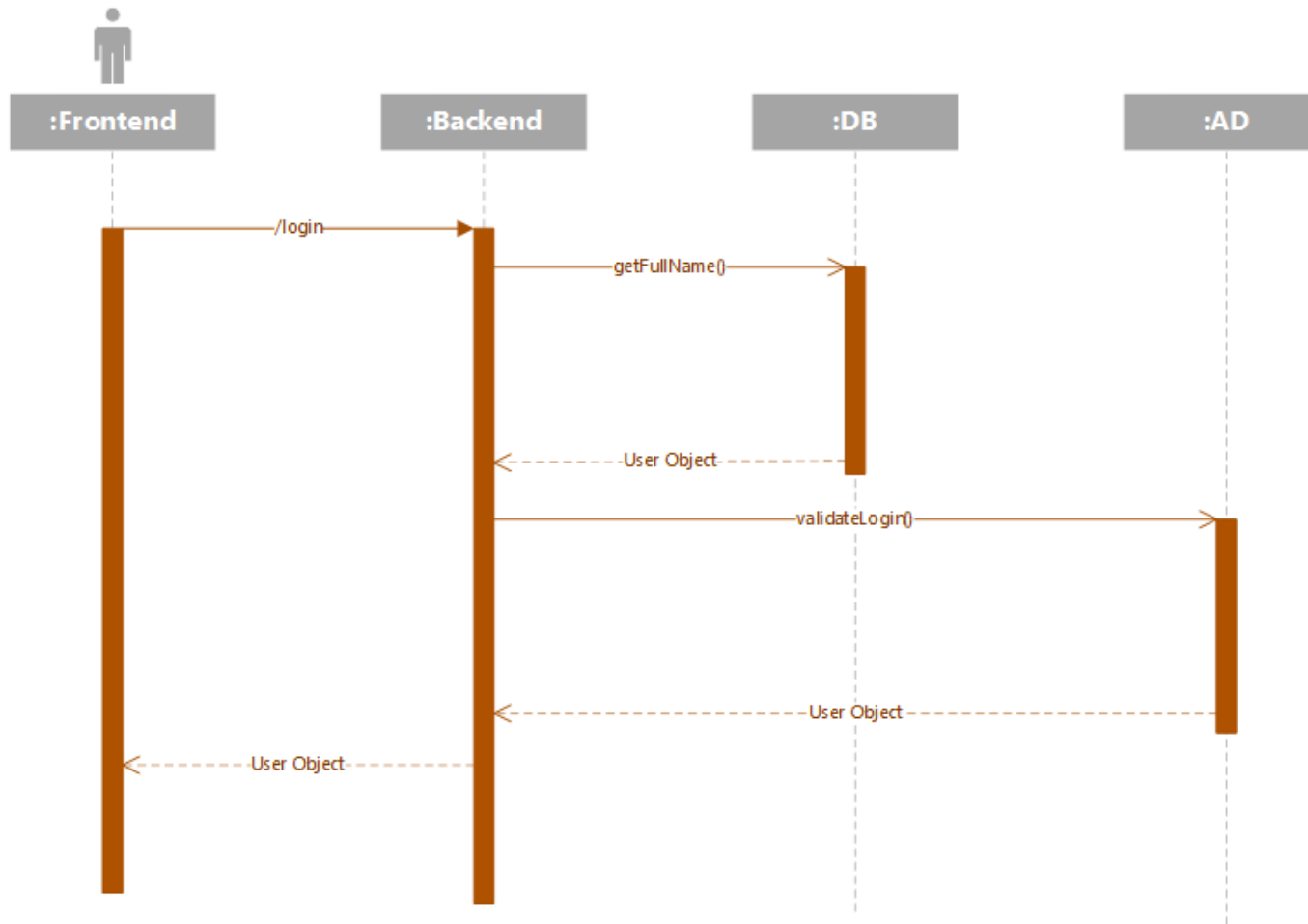
Bei der Ausführung des Skripts werden LOG-Meldungen in der Konsole ausgegeben. Sollte das Skript keine Änderungen an der DB vornehmen, so ist dies ersichtlich.

```
LOG: Fetching users from OU 'Lernende' successful!  
done fetching AD  
LOG: Fetching users from OU 'Lehrpersonen' successful!  
done fetching AD  
done fetching DB  
LOG: Everything up to date, no actions made.  
Migration successful
```

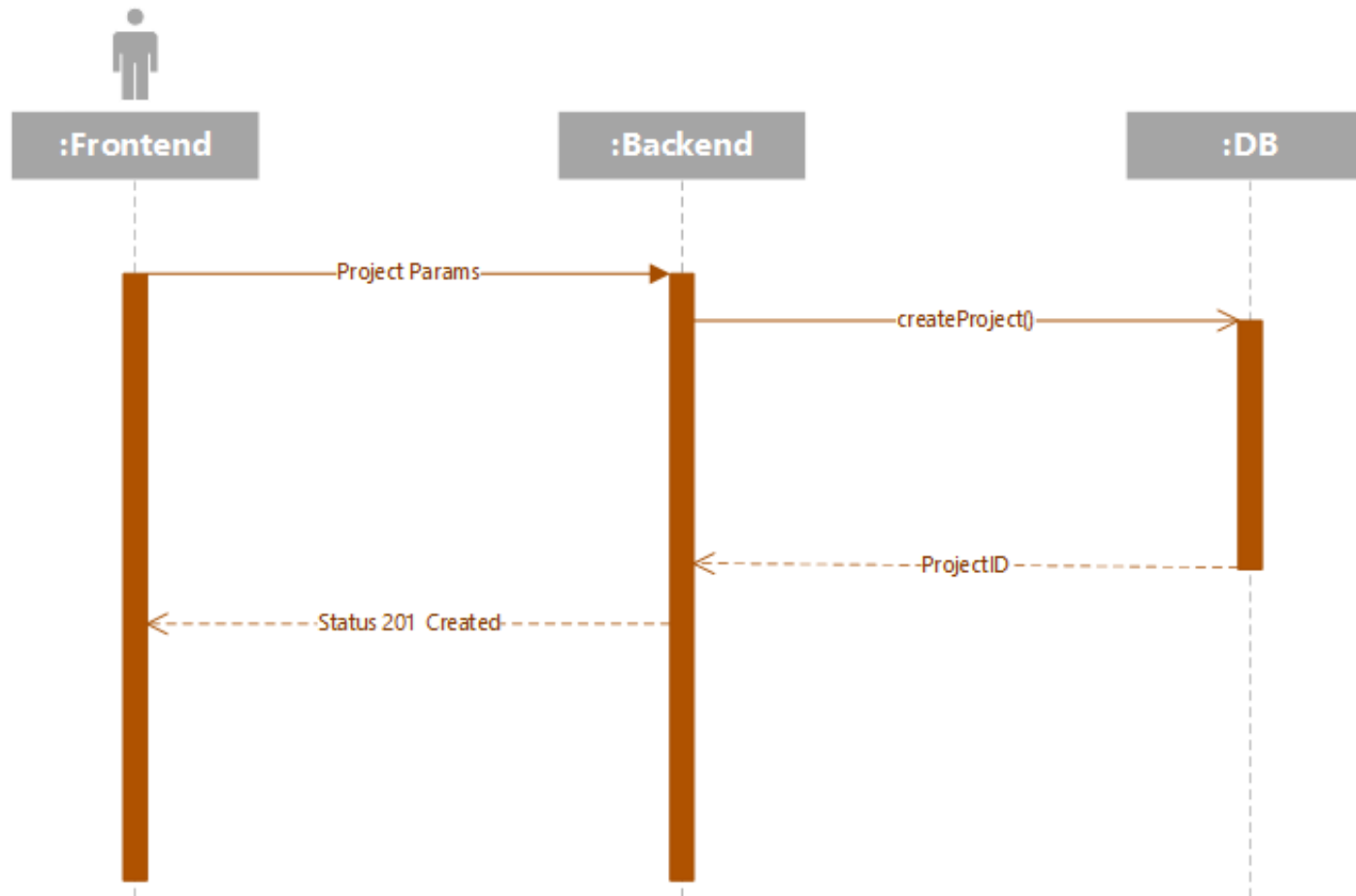
Abbildung 21 Realisierung - Migration Skript Log

13.5. Sequenzdiagramm

Das Sequenzdiagramm dient zu Veranschaulichung vom Login Prozess.



Dieses Sequenzdiagramm dient zu Veranschaulichung vom Projekt erstellen Prozess.



13.6. Vue Components (Frontend)

Die Webapplikation besteht aus verschiedenen Vue Components. In der Struktur wird zwischen Components und Views unterschieden. Die Views sind ganz dargestellte Ansichten, wobei die Components kleinere ausschnitte der Webseite wiedergeben. Rein technisch gesehen gibt es keinen Unterschied zwischen den Beiden, es wird im Sinne der Struktur dieser Unterschied gemacht.

13.6.1. Login

Beim login werden dem User der Kürzel und das Passwort der ipa.local Domäne abgefragt und mit hilfe von Axios an das Backend versandt. Wenn die Informationen nicht validiert werden konnten, wird eine fehlermeldung ausgegeben. Falls die Eingaben jedoch richtig sind, bekommt das Login vom Backend einen JSON Web Token und den vollständigen User mit zurück.

Der Token und der User werden im vuex store abgelegt und der Benutzer wird je nach Rolle entspreched weitergeleitet (Lernender: «/myProjects», Lehrperson: «/») weitergeleitet.

```
this.$store.dispatch('setToken', RESPONSE.data.token)
this.$store.dispatch('setUser', RESPONSE.data.user)
if(this.$store.state.isAuthenticated){
  this.$router.push({
    name: 'dashboard'
  })
}else{
  this.$router.push({
    name: 'myProjects'
  })
}
```

Hier ist die grafische Darstellung der JSON Objekte die zwischen dem Front – und Backend während dem erfolgreichen Login Prozess ausgetauscht werden.

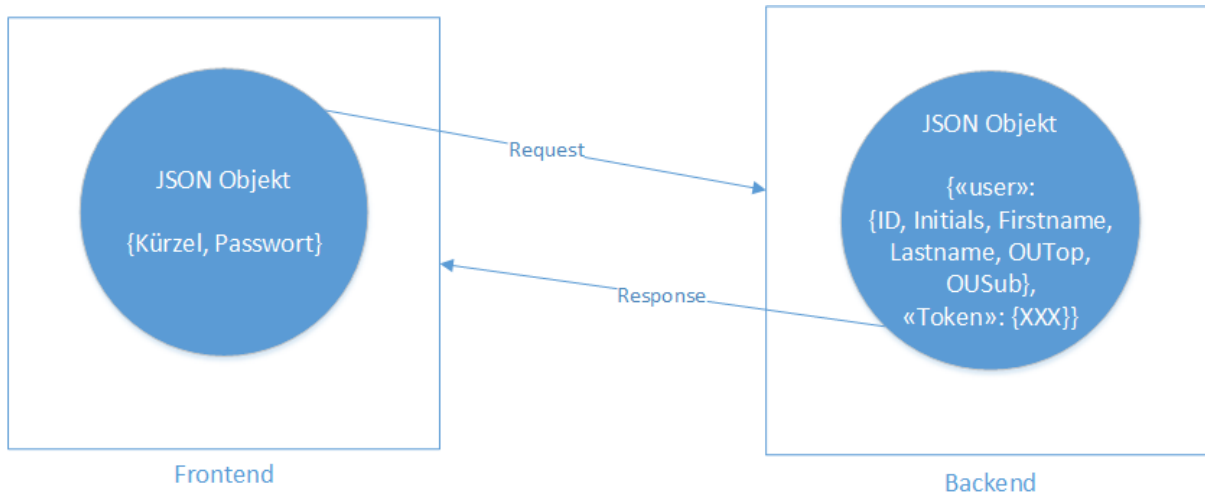


Abbildung 22 Login JSON

13.6.2. AddProject

AddProject befindet sich im components Ordner und bietet die Funktion Projekte zu erstellen. Diese Komponente wird von der Navbar aufgerufen und stellt somit im Header einen Knopf dar. Bei der Erstellung durchläuft die Lehrperson ein Formular mit Titel, Beschreibung, Kandidaten & Anhängige Projekte. Um ein Projekt zu erstellen wird nur der Titel benötigt.

Mit einem Klick auf den dargestellten Knopf, werden erstmals alle Projekte und Benutzer aus der Datenbank geholt. Danach scheitern das Formular als Popup.

```
async getProjects(){
  this.loading = true
  this.projects = []
  this.selected_projects = []
  let allProjects = (await ProjectsService.getAllProjects()).data
  allProjects.forEach(project => this.projects.push(project))
  this.loading = false
},
async getUser(){
  this.users = []
  this.selected_users = []
  this.selected_users.push(this.$store.state.user)
  let allUsers = (await ProjectsService.getAllUsers()).data
  allUsers.forEach(user => this.users.push(user))
},
```

Bei der Validierung des Formulars werden Grössen der eingegebenen Werte abgefragt. Sollte die Validierung an einem Punkt scheitern wird es Visuell dem User klargemacht. Die Funktion wird aufgerufen, wenn der «Weiter» Knopf gedrückt wird (siehe unten Abbildung Neues Projekt)

```
validateForm() {
  this.errorTitle = []
  this.errorDesc = []
  this.title = this.title.trim()
  if(!this.title){
    this.errorTitle.push('Bitte Titel eintragen!')
  }else if(this.title.length > 100){
    this.errorTitle.push('Titel zu lang!')
  }
  if(this.description.length > 2000){
    this.errorDesc.push('Beschreibung zu lang!')
  }
  if(this.errorTitle.length == 0 && this.errorDesc.length == 0){
    return this.step = 2
  }
}
```

Hier ist eine Grafik mit der Fehlerausgabe, falls nichts eingegeben wird. Es ist ebenfalls anhand '*' ersichtlich was die Pflichtfelder sind.

1 Initialisierung — 2 Konfiguration — 3 Übersicht

Titel*

Bitte Titel eintragen! 0 / 100

Beschreibung

0 / 2000

* Pflichtfelder

WEITER >

Abbildung 23 Neues Projekt

Abhängigkeiten zu anderen Projekte und Kandidaten werden von einem Vuetify Element namens *autocomplete* ausgewählt. Das Style-Element wird an eine Variable (*selected_projects*) gebunden und überträgt so alle zugewiesenen Attribute. Um alle Projekte in der Liste anzeigen zu können wird ebenfalls die Property *:items* definiert. Das Styling wird von der Erweiterung Vuetify vorgenommen.

```
<v-autocomplete
  box
  chips
  multiple
  label="Abhängige Projekte"
  item-value=""
  :items="projects"
  v-model="selected_projects"
  class="mb-0"
>
<template slot="selection" slot-scope="project">
  <v-chip
    close
    class="chip--select-multi"
    :selected="project.selected"
    @input="removeProject(project.item)"
  >
    <v-avatar>
      <v-icon>settings</v-icon>
    </v-avatar>
    {{ project.item.Title }}
  </v-chip>
</template>
<template slot="item" slot-scope="project">
  <template v-if="typeof project.item !== 'object'">
    <v-list-tile-content v-text="project.item"></v-list-tile-content>
  </template>
  <template v-else>
    <v-list-tile-avatar>
      <v-icon>settings</v-icon>
    </v-list-tile-avatar>
    <v-list-tile-content >
      <v-list-tile-title
        v-html="project.item.Title"
      >
    </v-list-tile-title>
    <v-list-tile-sub-title
      v-html="project.item.Title"
    >
  </template>
</template>
```

```
</v-list-tile-sub-title>
</v-list-tile-content>
</template>
</template>
</v-autocomplete>
```

Dieser Codeausschnitt ergibt folgendes Resultat in der Webapplikation:

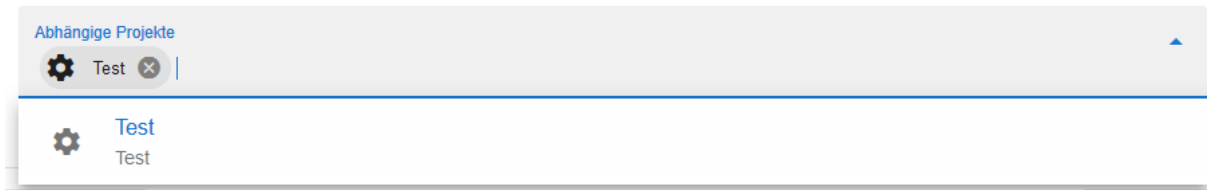


Abbildung 24 Neues Projekt Abhängigkeiten

Wenn der User alle nötigen Informationen ausgefüllt hat, kommt zuletzt eine Übersicht mit allen getätigten Eingaben. Der User kann jetzt auf «Projekt erstellen» klicken und es wird eine Anfrage mit allen ausgefüllten Feldern, über Axios an das Backend versandt.

13.6.3. Dashboard & MyProjects

Auf dem Dashboard werden alle nicht gelöschten Projekte angezeigt. Zugriff auf das Dashboard haben nur die Lehrkräfte.

Auf *MyProjects* haben beide Rollen (Lehrperson & Lernender) Zugriff. Da werden die jeweiligen Projekte abgebildet, in denen der eingeloggte User involviert ist.

Die Darstellung der beiden Views wird durch den Komponenten *showProject* gehandhabt.

Sprich *showProject* ist eine Komponente die Projekte anzeigt. Als *prop* erwartet sie ein Array mit Projekten. Die jeweiligen Projekte werden von *Dashboard* und *MyProjects* weitergeleitet.

13.6.4. ViewProject

Sollte der Benutzer auf ein Projekt draufklicken, wird er zur Komponente *ViewProject* weitergeleitet. Funktionell holt und schickt *viewProject* Aktionen an das Backend (über Axios). Beim Laden der Komponente wird die *ShowProject* Funktion aufgerufen, sie gibt das ganze Projekt mit den Abhängigkeiten zurück.

```
async mounted() {
  const PROJECT_ID = this.$store.state.route.params.projectId
  const USER_ID = this.$store.state.user.ID
  this.project = (await ProjectsService.showProject(PROJECT_ID, USER_ID)).data[0]

  this.formatted(this.project.Relations)
  this.formatted(this.project.Candidats)

  this.title = this.project.Title
  this.description = this.project.Description
  this.protocol = this.project.Protocol
  this.statusItems.map(item => {
    if(item.ID == this.project.Status){
      this.status = item.Status
    }
  })
}
```

Wurden die Daten erfolgreich empfangen, muss die *formatted()* Funktion aufgerufen werden. Die Antwort vom Backend sieht so aus:

```
Candidats: 1%Samuel Lötscher,2%Luxmithan Sukumar,3%Danyil Luntovsky,4%Usama
Guedi,15%Giulio Iannattone
```

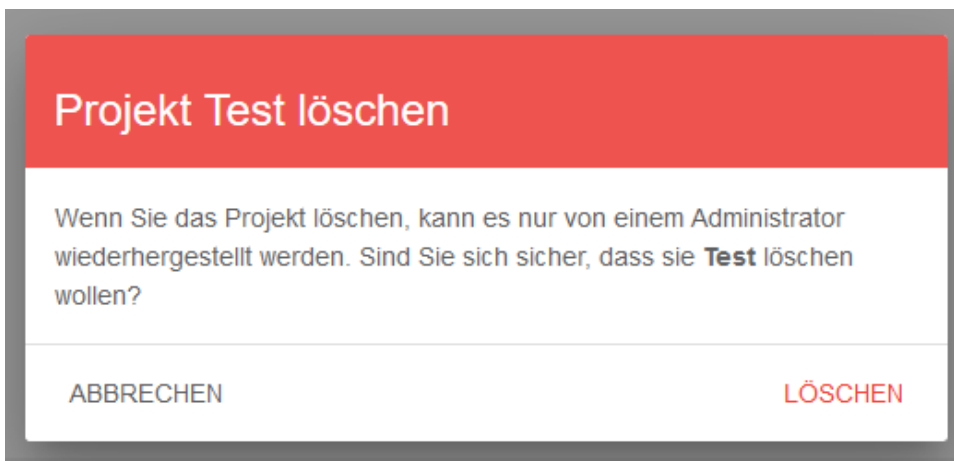
Die User sind mit einem Komma getrennt. Die UserId und der Name sind durch ein Modulo Zeichen getrennt. So muss die Funktion *formatted()* diese Rückgabe in Objekte verpacken.

```
formatted(data){
  if(data){
    this.relationsRaw = data.split(/,|%/)
    let x = 0
    let id = 0
    let container = []
    this.relationsRaw.forEach(rel => {
      if(x % 2){
        container.push({id: id, name: rel})
      }else{
        id = rel
      }
    })
  }
}
```

```
        x++  
    })  
    this.data.push([container])  
} else {  
    this.data.push([0])  
}  
}
```

Nebst dem Anzeigen hat ViewProject auch noch die Funktion die Projekte zu bearbeiten und zu löschen. Den Bearbeitungsmodus aktiviert man, indem der «Bearbeiten» Button betätigt wird. Darauf werden alle Sichtbaren Textelemente zu Eingabefeldern. Sollte der User die Bearbeitung bestätigen wollen werden die gleichen validierungsschritte durchlaufen um das Projekt letztendlich upzudaten.

Die Löschung von Projekten erfolgt durch das Menu Zeichen am oberen linken Rand. Sollte die Fuktion betätigt werden taucht folgende Meldung auf:



Sollte der User mit «Löschen» fortfahren wird er zum *Dashboard* weitergeleitet.

13.7. Vue Routing (Frontend)

Unter den Vue Dateien wird der Router verwendet um den User zu navigieren. In der *router.js* Datei wird ein Verzeichnis mit allen Komponenten erstellt auf die zugegriffen werden soll. Das *Vue.use* legt fest das die Vue Instanz den Router brauchen soll. Der *mode* wird auf *history* gesetzt um das Hashtag (#) in der URL-Zeile des Browsers auszublenden. Ebenfalls ist ersichtlich, dass die router Instanz in eine variable abgelegt und exportiert wird. Dies hat zufolge, dass der Router in den vuex Store geladen. Diese Aktion übernimmt die Dependencie *vuex-router-sync*. Es hat dann den Vorteil den Router als eine globale Variable, von jedem component aus aufrufen zu können.

```
Vue.use(Router)
```

```
export const router = new Router({
  mode: 'history',
  base: process.env.BASE_URL,
  props: true,
  routes: [
    {
      path: '/',
      name: 'dashboard',
      component: Dashboard,
    },
    {
      path: '/myProjects',
      name: 'myProjects',
      component: MyProjects,
    },
    {
      path: '/login',
      name: 'login',
      component: Login,
    },
    {
      path: '/project/:projectId',
      name: 'project',
      component: ViewProject,
    },
  ],
})
```


13.8. Vuex Store (Frontend)

Der Store befasst sich mit der Deklaration von globalen Werten oder in diesem Falle auch «States» genannt. Die States können durch die *mutations* verändert werden. Die mutations werden von *actions* aufgerufen. Möchte man jetzt eine State im Store ändern, so greift man auf eine *action* mittels der Funktion *dispatch* zu.

Im Rahmen von Projekt, muss jetzt der ganze Vuex Store im local storage abgelegt werden.

Dies hat den Vorteil, beim Nächsten Besuch auf der Webseite automatisch angemeldet zu sein.

Zur Sicherung des Stores wurde die Dependency *vuex-persistedstate* installiert und angewandt.

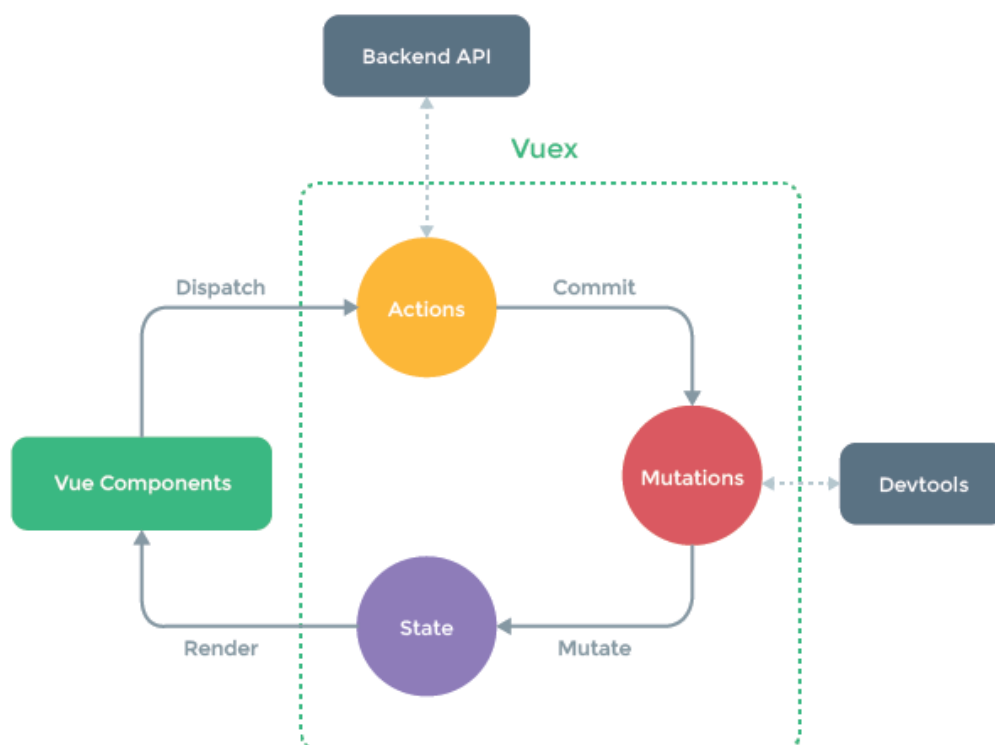


Abbildung 25 Vuex (Quelle: medium.com)

13.9. Axios (Frontend)

Als http-Client wird Axios verwendet. Es tätigt get / post Anfragen an das eigene Backend.

Bei jeder Übermittlung wird der Token, den der User beim Anmelden bekommt, mitgeschickt.

Versendet wird der Token über den Headerpunkt *Authorization*. Ebenfalls wird die Backend URL mit Port definiert:

(Datei: Api.js)

```
export default () => {  
  return axios.create({  
    baseURL: `http://localhost:1337/`,  
    headers: {  
      Authorization: `Bearer ${store.state.token}`  
    }  
  })  
}
```

Abhängig von dem Api.js File, ist das AuthenticationService.js und das ProjectsService.js. Die beiden tätigen die asynchronen Anfragen. Die Anfragen wurden in zwei verschiedene Dateien aufgeteilt, um Struktur zu schaffen. Hier ein Beispiel einer Post Anfrage, als Parameter werden der Kürzel und das vom User beim Login eingegebene Passwort über den Route *login* übermittelt.

```
login (credentials){  
  return Api().post('login', credentials);  
}
```

13.10. User Model (Backend)

Alle Userspezifischen Abfragen werden über den *AuthenticationController* an das User Model geschickt. Das Model tätigt Abfragen von der Datenbank sowie auch vom AD Server. Es verwendet die Dependencies Knex.js & Imap.js. Ein gutes Beispiel zur Verwendung des User & Project Models ist die *getProject()* abfrage, hier grafisch dargestellt:

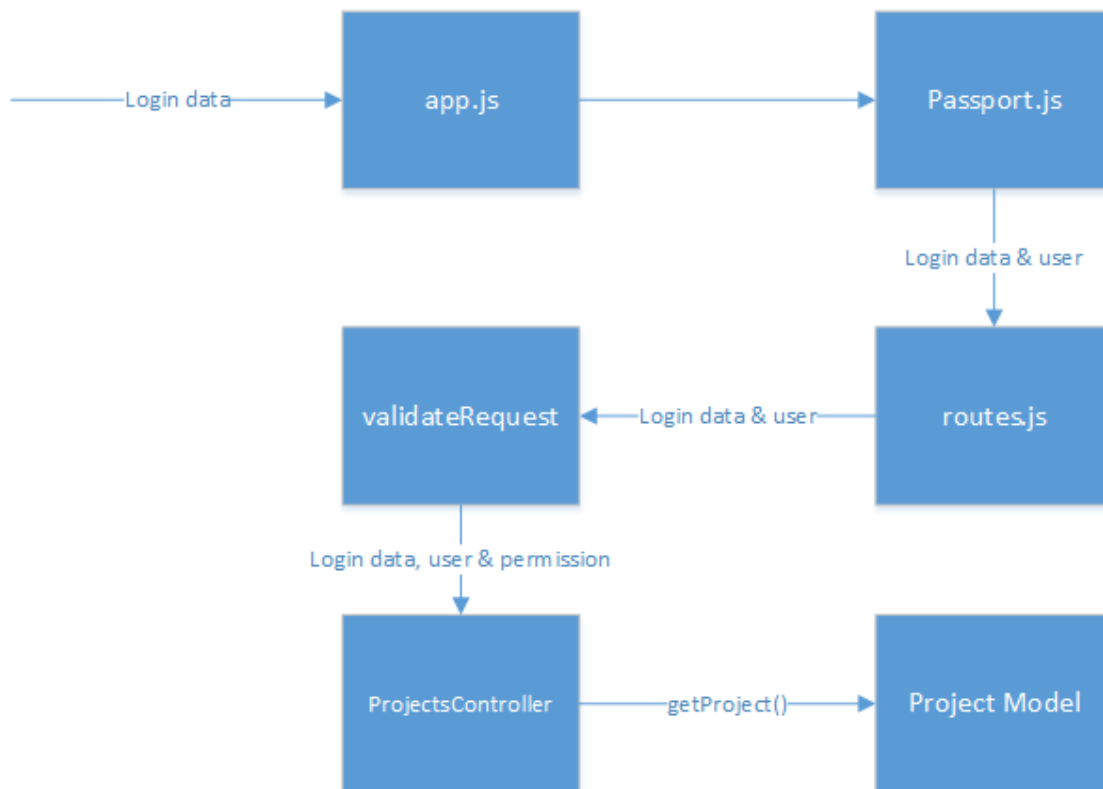


Abbildung 26 User Model Prozess

In der Grafik ist es ersichtlich, dass eine *getProject* Anfrage vom Frontend zugestellt wird. Danach wird der Token validiert. Vom Passport.js wird die Anfrage mit dem User Objekt direkt zum verantwortlichen routes.js File weitergeleitet, indem alle möglichen Anfragen definiert sind. Passt die Anfrage zur Liste, unterläuft der vom passport.js mitgesendete User ein check, ob es sich um einen Lernenden oder eine Lehrperson handelt. Darauf wird *ProjectsController* weitergeleitet, der dann die Anfragen an das Project Model stellt. Die Anfrage ist jetzt hier *getProjects()*.



Also vom AuthenticationController wird die Funktion *compareLogin()* mit den Usereingaben aus dem Frontend aufgerufen. Bei diesem Vorgang wird zuerst geprüft ob der eingegebene Kürzel bereits in der Datenbank vorhanden ist. Ist der Kürzel vorhanden wird von der DB aus das ganze Userobjekt wiedergegeben und zur Passwort Validierung in die Funktion *validateLogin()* geschickt. Erfolgen beide Funktionen, so wird an den AuthenticationController das Userobjekt zurückgeschickt.

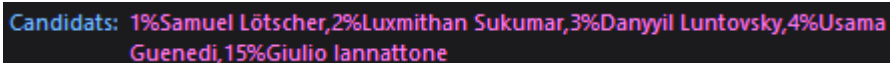
Meine Gedankengänge bei diesem Vorgang waren, dass der Active Directory mit genug Anfragen konfrontiert wird. Das heisst ich wollte so wenige Anfragen an den AD versenden wie nur möglich. Deshalb wird beim Login Vorgang zuerst der Kürzel in der DB abgeglichen und dann erst das Passwort.

```
/**
 * @param {Object} params
 * @returns {Object | boolean}
 * calls getFullName() function with initials as params, to get the full user Object
 * then calls validateLogin() with Users login information params and the full User Object
 */
async function compareLogin(params) {
  var loggedIn = false
  fullName = await getFullName(params.initials)
  try {
    return await validateLogin(params, fullName)
  } catch (err) {
    return false
  }
}
```

13.11. Projekt Model (Backend)

Auf das Projekt Model wird zugegriffen, wenn die Anfrage Projekt bezogen ist. Das Vorgehen ist gleich wie beim User Model (siehe Grafik User Model Process). Nur dass hier jetzt der ProjectsController zum Einsatz kommt. Es unterscheiden sich nur die eigentlichen Funktionen. Das Modul interagiert nur mit der DB, das heisst es wird nur die Erweiterung Knex.js benötigt.

Bei den Vue Components (Überschrift 13.5.4) wird folgende Grafik gezeigt:



Candidats: 1%Samuel Lötscher,2%Luxmithan Sukumar,3%Danyil Luntovsky,4%Usama Guenedi,15%Giulio Iannattone

Abbildung 27 Projekt Model Join Output

Diese Ausgabe kommt folgender Massen zustande:

```
async function showProject(id) {
  return await KNEX
    .select(
      'pro.ID',
      'pro.Title',
      'pro.Description',
      'pro.Protocol',
      'pro.CreatedAt',
      'sta.ID as Status',
      KNEX.raw('GROUP_CONCAT(distinct rel.Related_Project_ID, \'%\', pro_rel.Title) as Relations'),
      KNEX.raw('GROUP_CONCAT(distinct usr.ID, \'%\', usr.Firstname, \' \', usr.Lastname) as
Candidats'),
    )
    .from('project_user as pro_usr')
    .leftJoin('user as usr', 'pro_usr.User_ID', 'usr.ID')
    .leftJoin('project_relation as rel', 'pro_usr.Project_ID', 'rel.Project_ID')
    .leftJoin('project as pro', 'pro_usr.Project_ID', 'pro.ID')
    .leftJoin('project as pro_rel', 'rel.Related_Project_ID', 'pro_rel.ID')
    .leftJoin('status as sta', 'pro.Status', 'sta.ID')
    .where('pro.ID', id)
    .andWhere('pro.Visible', 1)
}
```

Bei diesem 5er Join werden alle Informationen eines Projekts aus der DB entnommen. Hier wurde probiert in einem Query, nebst allen Projekt Daten, auch noch alle Kandidaten und Abhängige Projekte rauszubekommen.

Dies war das Resultat, wie man am Oberen Codeblock erkennen kann. Dabei wurde *group_concat* mit dem Zusammenspiel von *distinct* verwendet. Die Trennung zwischen ID und dem Attributs Namen wird durch das Modulo Zeichen definiert.

Bei *getMyProjects()* wird die *userId* als Parameter erwartet. Anhand dieser werden alle Projekte, die mit dieser Id in Verbindung stehen ausgegeben. Die Zwischentabelle *project_user* wird zwei Mal gejoined, um in erster Linie den involvierten User mit den entsprechenden Projekten zu verbinden. Beim zweiten join der gleichen Tabelle, wird der Ersteller des Projektes angesprochen und ausgegeben.

```
async function getMyProjects(userId){
  return await KNEX
    .select(
      'pro.ID as ID',
      'pro.Title',
      'pro.Description',
      'pro.Protocol',
      'pro.CreatedAt',
      'sta.Status',
      'usr.ID as Usr_ID',
      'usr.Initials',
      'usr.Firstname',
      'usr.Lastname',
    ).from('project as pro')
    .leftJoin('project_user as pro_usr', 'pro.ID', 'pro_usr.Project_ID')
    .leftJoin('project_user as pro_usr1', 'pro.ID', 'pro_usr1.Project_ID')
    .leftJoin('user as usr', 'pro_usr1.User_ID', 'usr.ID')
    .leftJoin('status as sta', 'pro.Status', 'sta.ID')
    .where('pro_usr.User_ID', userId)
    .andWhere('pro.Visible', 1)
    .andWhere('pro_usr1.Creator', 1)
}
```

13.12. Sicherheit (Backend)

Die Sicherheit der Anfragen bei der Applikation wird grösstenteils durch den JSON Web Token gewährleistet. Dieser wird im AuthentikationsController erstellt und an das Frontend gesendet, wenn die Login Daten stimmen. Der Token wird mittels des User Objektes und dem Schlüssel erstellt. Der Schlüssel wurde im *config.js* File definiert. Dem Token wird eine Lebensdauer verabreicht, in diesem Falle, eine Woche.

(Datei: AuthenticationController)

```
function jwtSignUser(user) {  
  //token validation one week  
  const ONE_WEEK = 60 * 60 * 24 * 7  
  return JWT.sign({user}, CONFIG.authentication.jwtSecret, {  
    expiresIn: ONE_WEEK  
  })  
}
```

Bei einer eingehenden Anfrage, wird der Token entschlüsselt. Da bei der Token Verschlüsselung im Controller der User mitverschlüsselt wurde, wird beim entschlüsselt geprüft ob der User valide ist.

(Datei: passport.js)

```
const JWT_STRATEGY = require('passport-jwt').Strategy  
const EXTRACT_JWT = require('passport-jwt').ExtractJwt  
//this file was created with assistance from the website: http://www.passportjs.org/docs  
opts.jwtFromRequest = EXTRACT_JWT.fromAuthHeaderAsBearerToken()  
opts.secretOrKey = CONFIG.authentication.jwtSecret  
PASSPORT.use(  
  new JWT_STRATEGY(  
    opts,  
    async function(jwt_payload, done){  
      let user = await USER.getOne(jwt_payload.user.Initials)  
      if(!user){  
        return done(err, false)  
      }  
      if(user) {  
        done(null, user)  
      }else{  
        done(null, false)  
      }  
    }  
  )  
)
```



)

Der Zweite Schritt unterteilt den User in zwei Gruppen: Lernende & Lehrpersonen. Die Information (*modify: true or false*) wird nebst dem User Objekt an den entsprechenden Controller weitergeschickt.

```
module.exports = (req, res, next) => {  
  PASSPORT.authenticate('jwt', function (err, user) {  
    if (err || !user) {  
      res.status(403).send({  
        error: 'no permissions'  
      })  
    } else {  
      if (user[0].OUTop === 'Lehrpersonen') {  
        req.user = {user, modify: true}  
        next()  
      } else {  
        req.user = {user, modify: false}  
        next()  
      }  
    }  
  })  
  (req, res, next)  
}
```


13.13. Design

Das Design wurde im Vergleich zum Konzept Überarbeitet optimiert und Benutzerfreundlicher gestaltet. Dies geschah mithilfe von Vuetify. Bei der Erstellung habe ich ab und zu ein Auge in die Vuetify Dokumentation geworfen.

13.13.1. Login

Das Login wurde minimal verschönert und dekoriert.

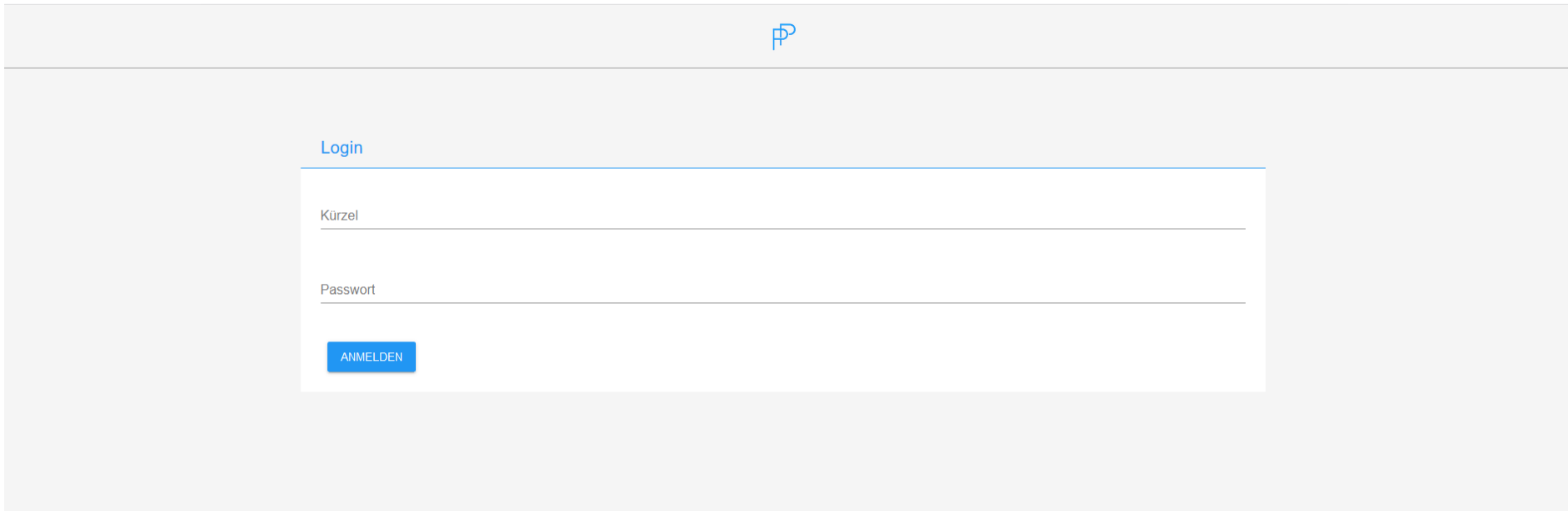
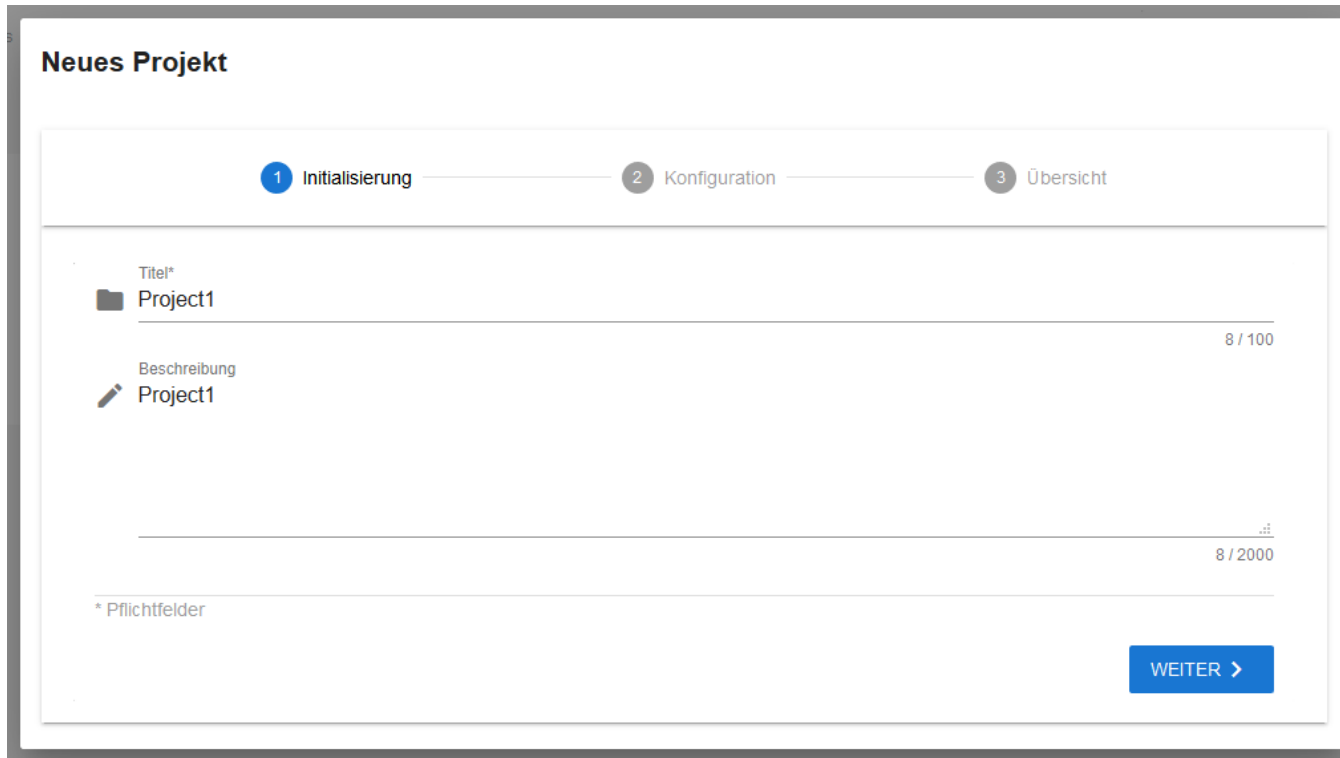
The image shows a web application login interface. At the top, there is a light gray header bar containing a small blue logo. Below the header, the main content area is a light gray rectangle. Centered within this area is a white rectangular box representing the login form. The form has a title 'Login' in blue text at the top left. Below the title are two input fields: the first is labeled 'Kürzel' and the second is labeled 'Passwort'. Both fields have thin gray borders. Below the password field is a blue button with the text 'ANMELDEN' in white capital letters.

Abbildung 28 Login

13.13.2. Project Erstellung

Erste Schritt, Projekt Titel muss ausgefüllt werden, die Beschreibung ist optional.



Neues Projekt

1 Initialisierung — 2 Konfiguration — 3 Übersicht

Titel*
Project1 8 / 100

Beschreibung
Project1 8 / 2000

* Pflichtfelder

WEITER >

Abbildung 29 Neues Projekt 1

Zweiter Schritt, die Abhängigkeiten werden definiert.


Neues Projekt


✓ Initialisierung


2 Konfiguration


3 Übersicht


Mitglieder


 Samuel Lötscher




 Luxmithan Sukumar





 Danyyil Luntovsky



Abhängige Projekte

 Test



 Test

Test

< ZURÜCK

WEITER >

Abbildung 30 Neues Projekt 2

Dritter Schritt, die Übersicht von allen zuvor eingegebenen Informationen

Neues Projekt

✓ Initialisierung

✓ Konfiguration

3 Übersicht

Projekt Titel
Project1

Beschreibung
Project1

Teilnehmer

Giulio Iannattone

Samuel Lötscher

Luxmithan Sukumar

Danyyil Luntovsky

Abhängige Projekte

Test

< ZURÜCK

PROJEKT ERSTELLEN

Abbildung 31 Neues Projekt 3

13.13.3. Dashboard & myProjects

Beide Views sehen identisch aus, bis auf die Ausgabe der Projekte unterscheidet sie nichts.

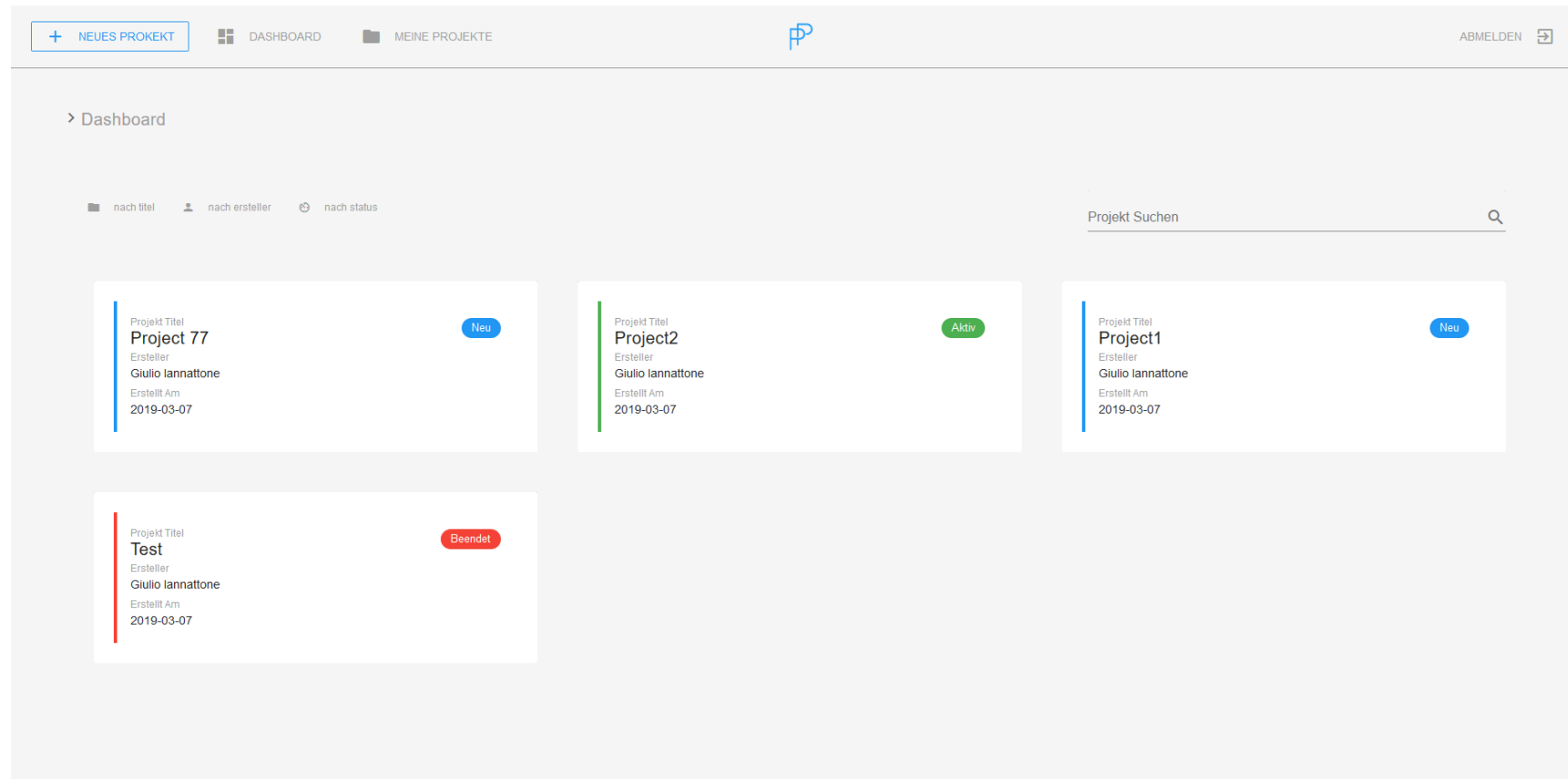


Abbildung 32 Dashboard

13.13.4. ViewProject

Alle Informationen zum Projekt können hier entnommen werden und das Protokoll kann geführt werden.

The screenshot displays a web application interface for viewing a project. At the top, there is a navigation bar with a '+ NEUES PROJEKT' button, a 'DASHBOARD' link, a 'MEINE PROJEKTE' link, a logo, and an 'ABMELDEN' button. Below the navigation bar, the main content area is titled 'ViewProject'. It features a 'ZURÜCK' button on the left and a 'BEARBEITEN' button on the right. The project details are as follows:

- Projekt Titel:** IPA
- Status:** Neu (indicated by a blue circle with the text 'Neu')
- Projekt Beschreibung:** Test Projekt
- Kandidaten:** Danyyil Luntovsky, Özcan Altin, Giulio Iannattone (each name is in a grey pill-shaped button with a person icon)
- Abhängigkeiten:** Project 77 (indicated by a gear icon in a grey pill-shaped button)
- Protokoll:** A large text area for notes, currently empty, with a '0 / 30000' character count and a 'SPEICHERN' button at the bottom right.

Abbildung 33 View Projekt

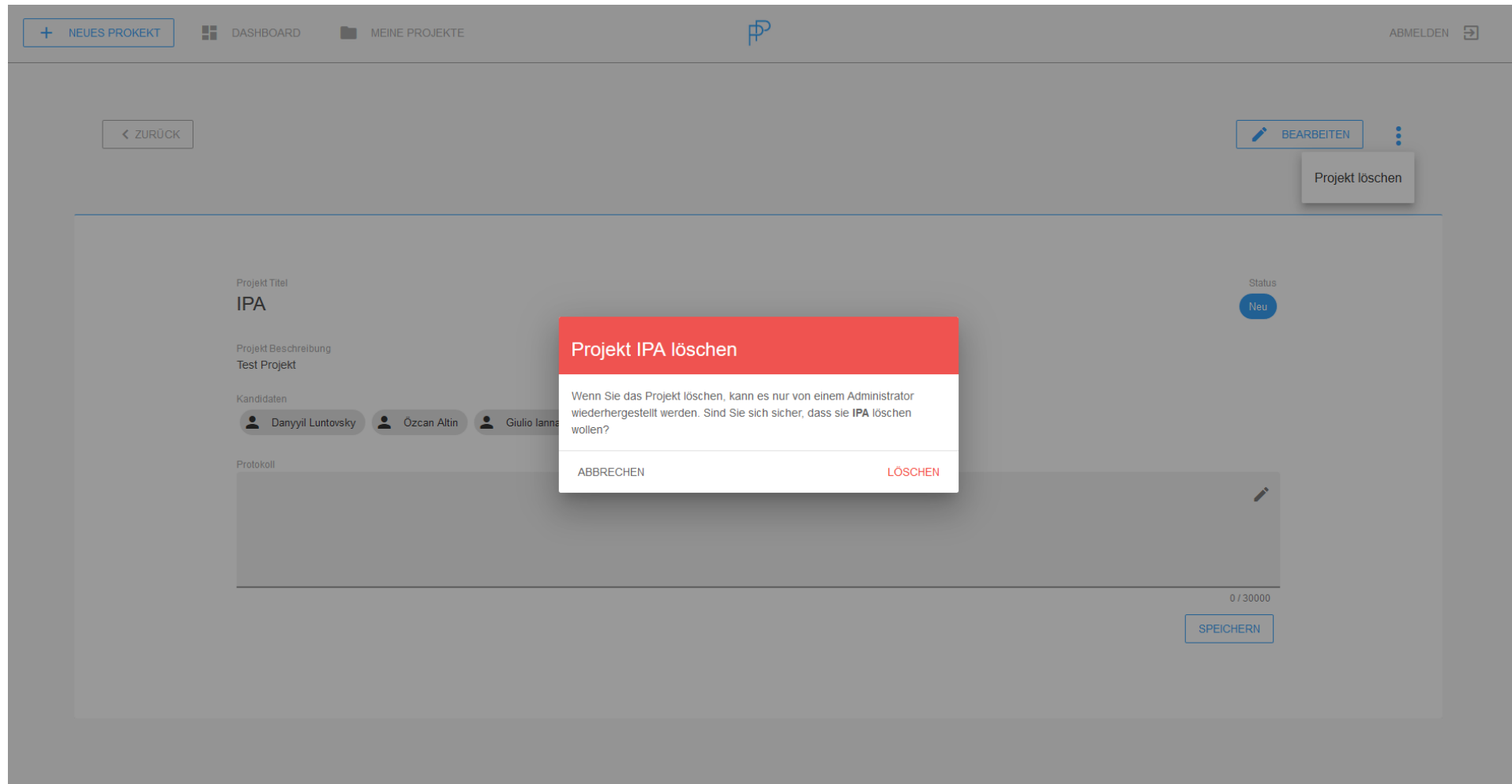
13.13.5. ViewProject bearbeiten

Klickt man auf den Bearbeiten Knopf, erscheinen Input Felder, zur Bearbeitung des Projektes.

The screenshot shows a web application interface for editing a project. At the top, there is a navigation bar with a '+ NEUES PROJEKT' button, a 'DASHBOARD' link, a 'MEINE PROJEKTE' link, a logo, and an 'ABMELDEN' button. Below the navigation bar, there is a header area with a '< ZURÜCK' button on the left and 'SAVE' and 'CANCEL' buttons on the right. The main content area contains three input fields: 'Project title' with the value 'IPA' and a character count '3 / 100', 'Status' with a dropdown menu showing 'Neu', and 'Project description' with the value 'Test Projekt' and a character count '12 / 2000'. Below these fields is a 'Protokoll' section with a large text area and a character count '0 / 30000'.

Abbildung 34 Projekt Bearbeiten

Dies ist die Meldung, wenn man das Projekt löschen will.



The screenshot shows a web application interface for managing projects. At the top, there is a navigation bar with buttons for '+ NEUES PROJEKT', 'DASHBOARD', and 'MEINE PROJEKTE'. On the right, there is an 'ABMELDEN' button. The main content area displays a project titled 'IPA' with a description 'Test Projekt'. Below the description, there is a list of candidates: Danyyil Luntovsky, Özcan Altin, and Giulio Ianna. A modal dialog box is open in the center, titled 'Projekt IPA löschen'. The dialog contains the text: 'Wenn Sie das Projekt löschen, kann es nur von einem Administrator wiederhergestellt werden. Sind Sie sich sicher, dass sie IPA löschen wollen?'. At the bottom of the dialog, there are two buttons: 'ABBRECHEN' and 'LÖSCHEN'. The background is dimmed, showing the project details and a 'SPEICHERN' button at the bottom right.



13.14. **Rechtliche Grundlagen**

Die auf der Webseite verwendeten Icons wurden von Google erstellt und heissen Material-design-icons. Die Icons sind der Apache 2.0 Lizenz untergeordnet und können somit legal verwendet werden.

Quelle: <http://google.github.io/material-design-icons/#what-are-material-icons>

13.15. Testprotokoll

Aus platzsparenden und redundanzvermeidenden Gründen werden die Testfälle mit reduzierter Information dargestellt. Die Punkte Voraussetzungen, Test Schritte und die Test Informationen werden den Tabellen entnommen. Die genauen Vorgehensschritte wurden bereits im Testkonzept (Punkt Testfälle) definiert.

Zur Visualisierung wurden die nicht erfolgreichen Testfälle mit Rot markiert.

Testfall ID	1				
Testfall	Login Lernender				
Beschreibung	Keine Eingabe	Richtiger Kürzel & falsches Passwort	Falscher Kürzel & richtiges Passwort	Falscher Kürzel & falsches Passwort	Richtiger Kürzel & richtiges Passwort
Erwartetes Resultat	Fehlermeldung: Invalide Login Informationen				Der User wird zu '/myProjects' weitergeleitet
Tatsächliches Resultat	Wie erwartet	Wie erwartet	Wie erwartet	Wie erwartet	Der User wird zu '/' weitergeleitet
Kommentar					«Meine Projekte» muss angeklickt werden, um zu '/myProjects' zu gelangen.
Fehlerklasse	M0	M0	M0	M0	M1

Tabelle 56 Testprotokoll - Testfall 1

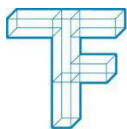


Testfall ID	2			
Testfall	Login Lehrperson			
Beschreibung	Richtiger Kürzel & falsches Passwort	Falscher Kürzel & richtiges Passwort	Falscher Kürzel & falsches Passwort	Richtiger Kürzel & richtiges Passwort
Erwartetes Resultat	Fehlermeldung: Invalide Login Informationen			Der User wird zu '/' weitergeleitet
Tatsächliches Resultat	Wie erwartet	Wie erwartet	Wie erwartet	Wie erwartet
Kommentar				
Fehlerklasse	M0	M0	M0	M0

Tabelle 57 Testprotokoll - Testfall 2

Testfall ID	3			
Testfall	Lernender Projektansicht			
Beschreibung	Projektansicht beim Lernenden			
Erwartetes Resultat	Der Lernende wird auf die Projektansicht Seite weitergeleitet wo er alle Informationen zum jeweiligen Projekt entnehmen kann.			
Tatsächliches Resultat	Wie erwartet			
Kommentar				
Fehlerklasse	M0			

Tabelle 58 Testprotokoll - Testfall 3

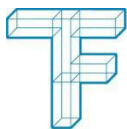


Testfall ID	4			
Testfall	Projekt Protokoll führen			
Beschreibung	Keine Eingabe	Eingabe von über 30'000 Zeichen	Eingabe Script Tags Injektion	Eingabe von max. 30'000 Zeichen
Erwartetes Resultat		Fehlermeldung: Eingabe zu lang	Das Protokoll wird ohne zusätzlichen Aktionen abgespeichert. Die Seite lädt neu und die Änderungen sind ersichtlich	
Tatsächliches Resultat	Wie erwartet	Wie erwartet	Wie erwartet	Wie erwartet
Kommentar				
Fehlerklasse	M0	M0	M0	M0

Tabelle 59 Testprotokoll - Testfall 4

Testfall ID	5		
Testfall	Projekt Zugriffsrechte Lernende		
Beschreibung	Lernender versucht auf fremdes Projekt zuzugreifen.	Lernender versucht auf alle Projekte zuzugreifen.	Lernender versucht auf ein ihm zugeordnetes Projekt zuzugreifen.
Erwartetes Resultat	Dem Lernenden soll keinerlei Information über das Projekt ersichtlich sein.	Der Lernende soll nach '/myProjekts' weitergeleitet werden.	Der Lernende soll auf Projektansicht vom Projekt 1 weitergeleitet werden.
Tatsächliches Resultat	Wie erwartet	Der Lernende wird nicht weitergeleitet	Wie erwartet
Kommentar			
Fehlerklasse	M0	M1	M0

Tabelle 60 Testprotokoll - Testfall 5

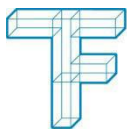


Testfall ID	6		
Testfall	Lehrperson Zugriffsrechte		
Beschreibung	Alle Projekte anzeigen lassen	Nur der Lehrperson ihre Projekte anzeigen lassen.	Das Projekterstellungs-Dialog wird aufgerufen
Erwartetes Resultat	Alle bislang erstellten Projekte werden angezeigt	Nur Projekte, wo man involviert ist, werden angezeigt.	Ein Dialog-Fenster erscheint, mit dem Titel 'Projekt erstellen'
Tatsächliches Resultat	Wie erwartet	Wie erwartet	Wie erwartet
Kommentar			
Fehlerklasse	M0	M0	M0

Tabelle 61 Testprotokoll - Testfall 6

Testfall ID	7			
Testfall	Projekt erstellen Initialisierung (Teil 1)			
Beschreibung	Es wird kein Titel eingegeben	Es wird ein zu langer Title eingegeben	Es wird ein valider Titel eingegeben	Die Felder Beschreibung & Start- sowie Enddatum werde (nicht) ausgefüllt
Erwartetes Resultat	Der zweite Schritt wird nicht eingeleitet. Fehlermeldung: 'Titel wird benötigt'	Der zweite Schritt wird nicht eingeleitet. Fehlermeldung: 'Titel zu lang'	Der zweite Schritt wird eingeleitet. Die nächste Seite der Projekterstellung ist zusehen.	
Tatsächliches Resultat	Wie erwartet	Wie erwartet	Wie erwartet	Wie erwartet
Kommentar				
Fehlerklasse	M0	M0	M0	M0

Tabelle 62 Testprotokoll - Testfall 7



Testfall ID	8			
Testfall	Projekt erstellen User & Projektzuordnung (Teil 2 / 3)			
Beschreibung	Kein Eingabe wird getätigt	Nur Kandidaten auswählen	Nur abhängige Projekte auswählen	Kandidaten & abhängige Projekte auswählen
Erwartetes Resultat	Der dritte Schritt wird eingeleitet. Die nächste Seite der Projekterstellung ist zusehen.			
Tatsächliches Resultat	Wie erwartet	Wie erwartet	Wie erwartet	Wie erwartet
Kommentar				
Fehlerklasse	M0	M0	M0	M0

Tabelle 63 Testprotokoll - Testfall 8

Testfall ID	9		
Testfall	Projekt erstellen, Übersicht (Teil 3 / 3)		
Beschreibung	Kontrollieren ob alle Daten des 2. Schrittes eingetragen sind	Kontrollieren ob alle Daten des 1. Schrittes eingetragen sind	Kontrollieren ob alle eingetragenen Daten mit dem Schritt 3 übereinstimmen
Erwartetes Resultat	Die zweite Seite samt allen eingegebenen Informationen wird sichtbar.	Die erste Seite samt allen eingegebenen Informationen wird sichtbar.	Das Dialog-Fenster schliesst sich. Eine Meldung erscheint, dass das Projekt erstellt wurde.
Tatsächliches Resultat	Wie erwartet	Wie erwartet	Wie erwartet
Kommentar			Aber nach dem Erstellen ist es nicht gleich sichtbar: Man muss z.B. zuerst Dashboard anklicken, dann wieder zurück zu Meine Projekte
Fehlerklasse	M0	M0	M1

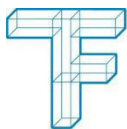


Tabelle 64 Testprotokoll - Testfall 9

Testfall ID	10	
Testfall	Erstellung des Projekts kontrollieren (Teil 1)	
Beschreibung	Das Projekt ist von 'Meine Projekten' aus sichtbar	Das Projekt ist vom 'Dashboard' aus sichtbar
Erwartetes Resultat	Das neu erstellte Projekt zeigt alle bei der Projekterstellung eingegebenen Informationen.	
Tatsächliches Resultat	Wie erwartet	Wie erwartet
Kommentar		
Fehlerklasse	M0	M0

Tabelle 65 Testprotokoll - Testfall 10

Testfall ID	11	
Testfall	Erstellung des Projekts kontrollieren (Teil 2)	
Beschreibung	Es wird vom Account des Lernenden getestet ob das erstellte Projekt mit allen Daten ersichtlich ist	
Erwartetes Resultat	Das neu erstellte Projekt zeigt alle bei der Projekterstellung eingegebenen Informationen.	
Tatsächliches Resultat	Wie erwartet	
Kommentar		
Fehlerklasse	M0	

Tabelle 66 Testprotokoll - Testfall 10



Testfall ID	12	
Testfall	Projekt bearbeiten (Teil 1 / 3)	
Beschreibung	Es ist der Lehrperson visuell ersichtlich dass das Projekt bearbeitet wird.	Es werden alle geränderten Daten nicht übernommen.
Erwartetes Resultat	Alle Informationen des Projektes werden zu Feldern die man bearbeiten kann.	Es werden keine Änderungen vorgenommen. Man sieht die Projektdetails jetzt wieder in Textform.
Tatsächliches Resultat	Wie erwartet	Wie erwartet
Kommentar		
Fehlerklasse	M0	M0

Tabelle 67 Testprotokoll - Testfall 12

Testfall ID	13	
Testfall	Projekt bearbeiten (Teil 2 / 3)	
Beschreibung	Das erstellte Projekt wird bearbeitet.	
Erwartetes Resultat	Die Seite wird neu geladen. Alle geänderten Informationen sind ersichtlich.	
Tatsächliches Resultat	Wie erwartet	
Kommentar		
Fehlerklasse	M0	

Tabelle 68 Testprotokoll - Testfall 13

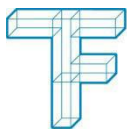


Testfall ID	14	
Testfall	Projekt bearbeiten (Teil 3 / 4)	
Beschreibung	Der Projekttitel wird gelöscht	Ein zu langer Projekttitel wird eingegeben
Erwartetes Resultat	Fehlermeldung: Titel eintragen Die Seite bleibt im Bearbeitungszustand	Fehlermeldung: Titel zu lang Die Seite bleibt im Bearbeitungszustand
Tatsächliches Resultat	Wie erwartet	Wie erwartet
Kommentar		
Fehlerklasse	M0	M0

Tabelle 69 Testprotokoll - Testfall 14

Testfall ID	15	
Testfall	Projekt bearbeiten (Teil 4 / 4)	
Beschreibung	Der zum Projekt zugeordnete Lernende kann auf das Projekt zugreifen.	
Erwartetes Resultat	Der Lernende wird auf 'Meine Projekte' weitergeleitet. Neu ist das 'Projekt 10' zu sehen. Die Informationen des Projekts sind die gleichen wie abgeändert	
Tatsächliches Resultat	Wie erwartet	
Kommentar		
Fehlerklasse	M0	

Tabelle 70 Testprotokoll - Testfall 15



Testfall ID	16	
Testfall	Projekt löschen	
Beschreibung	Die Löschung des Projekts wird abgebrochen.	Die Löschung des Projekts wird durchgeführt.
Erwartetes Resultat	Der Dialog mit der Warnung Schliesst sich. Das Projekt bleibt vorhanden.	Der Dialog mit der Warnung Schliesst sich. Das Projekt wird in der Datenbank als gelöscht markiert und ist auf der Applikation nicht mehr ersichtlich. Der User wird zum Dashboard weitergeleitet.
Tatsächliches Resultat	Wie erwartet	Wie erwartet
Kommentar		
Fehlerklasse	M0	M0

Tabelle 71 Testprotokoll - Testfall 16

Testfall ID	17	
Testfall	Sich von der Webseite abmelden	
Voraussetzungen	3) Webseite läuft 4) User ist eingeloggt	
Beschreibung	Die Lehrperson meldet sich ab.	Der Lernende melden sich ab.
Test Schritte	2) In der Navigationszeile auf 'Abmelden' klicken	
Test Informationen	Kürzel: alo Passwort: gibbiX12345	Kürzel: in16luda Passwort: gibbiX12345
Erwartetes Resultat	Der User ist abgemeldet. Er wird zum Login weitergeleitet.	
Tatsächliches Resultat	Wie erwartet	Wie erwartet
Kommentar		
Fehlerklasse	M0	M0

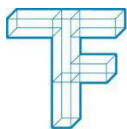
Tabelle 72 Testprotokoll - Testfall 17

13.15.1. Re-Testing

Da 3 von 16 Testfällen unzufrieden stellend abgeschlossen wurden, wird das Re-Testing durchgeführt.

Testfall ID	1
Testfall	Login Lernender
Beschreibung	Richtiger Kürzel & richtiges Passwort
Erwartetes Resultat	Der User wird zu '/myProjects' weitergeleitet
Tatsächliches Resultat	Der Lernende wird zu '/myProjects' weitergeleitet
Fehlerklasse	M0

Testfall ID	5
Testfall	Projekt Zugriffsrechte Lernende
Beschreibung	Lernender versucht auf alle Projekte zuzugreifen.
Erwartetes Resultat	Der Lernende soll nach '/myProjekts' weitergeleitet werden.
Tatsächliches Resultat	Der Lernende wird zu '/myProjects' weitergeleitet
Fehlerklasse	M0



Testfall ID	9
Testfall	Projekt erstellen, Übersicht (Teil 3 / 3)
Beschreibung	Kontrollieren ob alle eingetragenen Daten mit dem Schritt 3 übereinstimmen
Erwartetes Resultat	Das Dialog-Fenster schliesst sich. Eine Meldung erscheint, dass das Projekt erstellt wurde.
Tatsächliches Resultat	Wie erwartet
Kommentar	Die Seite wird jetzt neu geladen und das Projekt erscheint an erster Stelle.
Fehlerklasse	M0

13.15.2. Testabnahme



Datum	Name & Rolle	Unterschrift
06.02.2019	Özcan Altin, Testperson	
06.02.2019	Danyyil Luntovsky, Projektleiter	

Tabelle 73 Testabnahme

Abbildungsverzeichnis

Abbildung 1 Backup	17
Abbildung 2 Dokumentenablage	18
Abbildung 3 Projektablage	18
Abbildung 4 Dokumentenablage	19
Abbildung 5 Hauptarbeitsplatz	20
Abbildung 6 Nebenarbeitsplatz.....	21
Abbildung 7: Projektmethode	22
Abbildung 8: Projektorganisation.....	25
Abbildung 9 IST-Situation Projektportfolio	58
Abbildung 10 Netzwerkschema	63
Abbildung 11 Konzept Project erstellen.....	73
Abbildung 12 Ordnerstruktur Frontend	92
Abbildung 13 Ordnerstruktur Backend.....	93
Abbildung 14 Ordnerstruktur Migration Skript.....	94
Abbildung 15 MVC-Pattern.....	94
Abbildung 16 Realisierung ERM.....	95
Abbildung 17 Backend Package Diagram Final.....	96
Abbildung 18 Frontend Package Diagram Final	97
Abbildung 19 Ralisierung AD Aufbau	98
Abbildung 20 DN-Aufteilung	100
Abbildung 21 Realisierung - Migration Skipt Log	102
Abbildung 22 Login JSON	106
Abbildung 23 Neues Projekt.....	107
Abbildung 24 Neues Projekt Abhängigkeiten.....	109
Abbildung 25 Vuex (Quelle: medium.com)	113
Abbildung 26 User Model Prozess	115
Abbildung 27 Projekt Model Join Output	117
Abbildung 28 Login	121
Abbildung 29 Neues Projekt 1	122
Abbildung 30 Neues Projekt 2	123
Abbildung 31 Neues Projekt 3	124
Abbildung 32 Dashboard.....	125



Abbildung 33 View Projekt	126
Abbildung 34 Projekt Bearbeiten	127
Abbildung 36 Nachweis Zeitplan	156
Abbildung 37 Nachweis Dokumentation	156



Tabellenverzeichnis

Tabelle 1 Standards	15
Tabelle 2 Ordnerstruktur	19
Tabelle 3: Projektphasen.....	22
Tabelle 4 HERMES Module	24
Tabelle 5 Kontaktdaten	26
Tabelle 6: Projektrollen	27
Tabelle 7: Risikoanalyse	28
Tabelle 8: Risikoanalyse - Schadensausmass	29
Tabelle 9: Risikoanalyse – Eintrittswahrscheinlichkeit	29
Tabelle 10: Risikograph - Vor Massnahmen.....	30
Tabelle 11: Risikograph - Nach Massnahmen	31
Tabelle 12: Arbeitsjourna Tag 1	35
Tabelle 13 Arbeitsjourna Tag 2	37
Tabelle 14 Arbeitsjourna Tag 3	39
Tabelle 15 Arbeitsjourna Tag 4	41
Tabelle 16 Arbeitsjourna Tag 5	43
Tabelle 17 Arbeitsjourna Tag 6	45
Tabelle 18 Arbeitsjourna Tag 7	47
Tabelle 19 Arbeitsjourna Tag 8	49
Tabelle 20 Arbeitsjourna Tag 9	51
Tabelle 21 Arbeitsjourna Tag 10	52
Tabelle 22 Arbeitsjourna Tag 11	53
Tabelle 23 Formatierung Allgemein.....	57
Tabelle 24 Formatierung Tabelle.....	57
Tabelle 25 IST- SOLL Vergleich.....	60
Tabelle 26 Projektziele	61
Tabelle 27 IP-Adressenkonvention.....	63
Tabelle 28 Server Typen	64
Tabelle 29 Server Zwecke.....	64
Tabelle 30 Standort.....	65
Tabelle 31 Namenskonzept Benutzer	65
Tabelle 32 Konzept - ERM	67
Tabelle 33 Konzept - Package Diagramm Backend	68

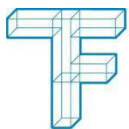


Tabelle 34 Konzept - Package Diagramm Frontend	69
Tabelle 35 Mockup Login	70
Tabelle 36 Mockup Dashboard.....	71
Tabelle 37 Mockup Projektansicht.....	72
Tabelle 39 Fehlerklassen	75
Tabelle 40 Testtabelle	75
Tabelle 41 Testfall 1	76
Tabelle 42 Testfall 2	77
Tabelle 43 Testfall 3.....	78
Tabelle 44 Testfall 4.....	79
Tabelle 45 Testfall 5.....	80
Tabelle 46 Testfall 6.....	81
Tabelle 47 Testfall 7	82
Tabelle 48 Testfall 8.....	83
Tabelle 49 Testfall 9.....	84
Tabelle 50 Testfall 10.....	85
Tabelle 51 Testfall 10.....	86
Tabelle 52 Testfall 12.....	87
Tabelle 53 Testfall 13.....	88
Tabelle 54 Testfall 14.....	89
Tabelle 55 Testfall 15.....	90
Tabelle 56 Testfall 16.....	91
Tabelle 57 Testprotokoll - Testfall 1	130
Tabelle 58 Testprotokoll - Testfall 2	131
Tabelle 59 Testprotokoll - Testfall 3	131
Tabelle 60 Testprotokoll - Testfall 4	132
Tabelle 61 Testprotokoll - Testfall 5	132
Tabelle 62 Testprotokoll - Testfall 6	133
Tabelle 63 Testprotokoll - Testfall 7	133
Tabelle 64 Testprotokoll - Testfall 8	134
Tabelle 65 Testprotokoll - Testfall 9	135
Tabelle 66 Testprotokoll - Testfall 10	135
Tabelle 67 Testprotokoll - Testfall 10	135
Tabelle 68 Testprotokoll - Testfall 12	136
Tabelle 69 Testprotokoll - Testfall 13	136



Tabelle 70 Testprotokoll - Testfall 14	137
Tabelle 71 Testprotokoll - Testfall 15	137
Tabelle 72 Testprotokoll - Testfall 16	138
Tabelle 73 Testprotokoll - Testfall 17	138
Tabelle 74 Testabnahme.....	140
Tabelle 75 Quellenverzeichnis	146
Tabelle 76: Änderungskontrolle.....	147
Tabelle 77 Freigabe der Phasen	148
Tabelle 78: Abkürzungen	149
Tabelle 79 Softwareliste	153

Quellenverzeichnis

Beschreibung	URL
HERMES Module	http://www.hermes.admin.ch/onlinepublikation/index.xhtml?element=kategorie_module.html
uml-diagrams.org	https://www.uml-diagrams.org/package-diagrams-overview.html
visual-paradigm.com	https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-package-diagram/
Ldapjs.org	http://ldapjs.org/
Node-ldap github	https://github.com/joyent/node-ldapjs/blob/master/docs/client.md
Vuetify	https://vuetifyjs.com/en/framework/a-la-carte
Vuetify autocomplete	https://vuetifyjs.com/en/components/autocomplete
Stackoverflow flatten Array	https://stackoverflow.com/questions/10865025/merge-flatten-an-array-of-arrays-in-javascript
Stackoverflow filter compare arrays	https://stackoverflow.com/questions/45599749/using-filter-to-compare-two-arrays-and-return-values-that-arent-matched
Knex insert	https://knexjs.org/
Passport-jwt	http://www.passportjs.org/packages/passport-jwt/
Github vuex-persistedstate	https://github.com/robinvdvleuten/vuex-persistedstate
Passport docs	http://www.passportjs.org/docs/
Medium vuex	https://medium.com/@sadickjunior/how-does-a-minimal-vuex-implementation-looks-like-find-out-c2c2e13619cb

Tabelle 74 Quellenverzeichnis



Dokumenteninformationen

Änderungskontrolle, Prüfung

Version	Datum	Name	Beschreibung
1.0	20.02.2019	Danyyil Luntovsky	Erste Version der Dokumentation
1.1	21.02.2019	Danyyil Luntovsky	Initialisierung wurde verfasst
1.2	22.02.2019	Danyyil Luntovsky	Das Konzept wurde angefangen
1.3	25.02.2019	Danyyil Luntovsky	Konzept wurde weitergeführt
1.4	27.02.2019	Danyyil Luntovsky	Konzept wurde beendet und Realisierung angefangen
1.5	28.02.2019	Danyyil Luntovsky	Realisierung wurde fortgeführt
1.6	01.03.2019	Danyyil Luntovsky	Realisierung wurde fortgeführt
1.7	04.03.2019	Danyyil Luntovsky	Realisierung wurde fortgeführt
1.8	06.03.2019	Danyyil Luntovsky	Realisierung wurde fortgeführt
1.9	07.03.2019	Danyyil Luntovsky	Realisierung wurde beendet
2.0	08.03.2019	Danyyil Luntovsky	Dokument wurde versandbereit gemacht.

Tabelle 75: Änderungskontrolle



Freigabe der Phasen












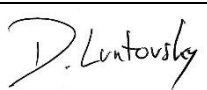
Meilenstein	Datum	Verantwortliche Fachkraft	Kandidat
Start der IPA	20.02.2019		
Initialisierung freigegeben	21.02.2019		
Konzept freigegeben	22.02.2019		
Realisierung freigegeben	25.02.2019		
Realisierung abgeschlossen	08.03.2019		
Teil 1 Administrativer Teil abgeschlossen	08.03.2019		

Tabelle 76 Freigabe der Phasen



Abkürzungsverzeichnis

Abkürzung	Bedeutung
BET	Betriebsinformatiker
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Server
EG	Erdgeschoss
GZ	Gesamtzeit
IP	Internet Protocol
IPA	Individuelle Praktische Arbeit
LAN	Local Area Network
LJ	Lehrjahr
LOA	Lorraine Annex
LOH	Lorraine Hauptgebäude
MS#	Meilenstein Nummer #
NB	Notebook
NM	Nachmittag
POE	Power Over Ethernet
R#	Risiko Nummer #
RT	Router
S#	Abwertung Skala (1-4)
SW	Switch
TF	Technische Fachschule Bern
UG	Untergeschoss
VLAN	Virtual Local Area Network
VM	Virtual Machine
VM(in Zeitplan)	Vormittag
W#	Wahrscheinlichkeit Skala (1-5)
WLAN	Wireless Local Area Network
HEX	Hauptexperte / Hauptexpertin
NEX	Nebenexperte / Nebenexpertin
VEX	Validexperte / Validexpertin
ERM	Entitätsblockdiagramm

Tabelle 77: Abkürzungen

Glossar

Bezeichnung	Definition
API	Eine Programmierschnittstelle
Active Directory	Verzeichnisdienst von Microsoft Windows Server
Axios	Axios ist eine auf Versprechen basierende asynchrone http-Clientbibliothek
Authorisationheader	Ein Attribut, dass sich im http Header befindet. Wird verwendet um den Token an das Backend zu versenden.
Body-parser	Body-parser extrahiert den gesamten Körperteil eines eingehenden Anforderungsstroms und macht ihn auf Anfrage verfügbar.
Controller	Kommuniziert mit der View und den Model, im MVC Pattern
Distinguished Name	Vom AD erhaltene Benutzerpfad
Model	Kommuniziert mit dem Controller und der DB / AD, im MVC Pattern
Dependency	Eine Node.js erweiterung zB. express
Express	Webanwendungs-Framework, mit MVC Architektur
JSON Web Token	Kurz JWT, ist ein RFC 7519 Access-Token. Wird zur Authentifikation der User verwendet.
Knex.js	Ist ein auf Node.js basierte Erweiterung, für den erleichterten Zugriff auf die MySQL DB
LDAP	Lightweight Directory Access Protocol. Ist ein Netzwerkprotokoll zur Abfrage und Änderung von Informationen verteilter Verzeichnisdienste.
Local Storage	Der lokale Speicher des Browsers

Material-Design	<i>Material Design</i> ist eine neue Designmethode. Aus einer Kombination des Flat Designs und dem realitätsnahen Skeuomorphism-Stil
Morgan	Morgan ist eine weitere HTTP-Request-Logger-Middleware für Node.js. Es vereinfacht das Protokollieren von Anforderungen in Ihrer Anwendung.
MVC	MVC ist ein Architekturmuster und steht für Model-View-Controller. MVC ermöglicht die einfache Wiederverwendung von einzelnen Komponenten und erleichtert auch das Anpassen einzelner Komponenten.
MVVM Design Pattern	Model View ViewModel ist ein Entwurfsmuster und eine Variante des Model-View-Controller-Musters.
MySQL	Eine Abkürzung für «Structured Query Language» ist eine Datenbanksprache, die verwendet wird, um Datenbanken zu erstellen, bearbeiten und Daten abzufragen.
Node_modules	In diesem Ordner werden alle Node dependencies abgelegt, die die Applikation benötigt. (Automatisch erstellter Ordner von Node.js)
Organisational Unit	Eine Userunterteilungsart auf dem AD. Es wird verwendet um Berechtigungen zu unterteilen und Ordnung zu schaffen.
OUSub	Wird als Tabelle in der Datenbank verwendet. Es Speichert alle Klassen der Lernenden.
OUTop	Wird als Tabelle in der Datenbank verwendet. Es Speichert alle Rollen der Benutzer.



Package.json	Alle installierten Pakete der Applikation werden hier aufgelistet.
Vue	Vue.js ist ein clientseitiges JavaScript-Webframework zum Erstellen von Single-Page-Webanwendungen nach dem MVVM-Pattern
Vuetify	Ist ein CSS Framework, dass zur Gestaltung des Frontends einer Website dient.
Vuex	Eine Vue Erweiterung zur Definition und Manipulation von globalen Werten
Vuex-persistedstate	Eine Dependencie um den Vuex Store im local storage zu speichern
Vuex-router-sync	Legt den Router als eine Variable im Vuex Store fest. So kann von allen Komponenten aus, auf den Router zugegriffen werden.



Softwareliste

Software-Name
Postman
Mozilla Firefox
Google Chrome
Visual Studio Code
Microsoft Visio
Microsoft Word
Microsoft Excel
MySQL Workbench
MySQL Server

Tabelle 78 Softwareliste

14. Anhang

14.1. Sitzungsprotokolle

Thematik		
Erster Expertenbesuch		
Entscheide	Teilnehmer	Ort, Datum
<ul style="list-style-type: none"> - Die HERMES Module werden aufgelistet aber nicht weiter in den Fluss der Dokumentation aufgenommen. - Die Risikoanalyse in die Initialisierung der Dokumentation verschieben. - Package Diagramm <ul style="list-style-type: none"> o Module auflisten o Welche Dependencies werden von der Applacation benötigt - Sequence Diagramm (Mit Pseudoklassen) <ul style="list-style-type: none"> o Nur Lese und Schreibe Funktionen - Backupkonzept detailliert erläutern - Glosar Alphabetisch geordnet - Aufzählung aller Abkürzungen - Bei externer Hilfe Quelle oder Person angeben (Arbeitsjournal) - Die Dokumentation korrigieren lassen - Testing Protokoll ausführlich ausbauen - Die Freigabe der Phasen Unterschreiben lassen - Der Projektablauf soll für eine Aussenstehende Person ersichtlich sein - Edgar Gmür nur PDF (nicht ausdrucken) - Den Code gezippt hochladen 	Pia Schmid, Edgar Gmür, Özcan Altin, Danyyil Luntovsky	Bern, 22.02.2019



Thematik		
Zweiter Expertenbesuch		
Entscheide	Teilnehmer	Ort, Datum
<ul style="list-style-type: none">- Die Risikoanalyse soll im Teil 1 der Dokumentation bleiben- Der Source Code soll in den Anhang der Dokumentation- Ebenfalls den Source Code bei PkOrg unter IPA-Anhänge hochladen- Die Benutzeranleitung für das Migration Skript in den Anhang der Doku, sowie auch auf PkOrg ablegen- Die Dokumentation nicht eingeschrieben per Post an Pia Schmid versenden- Die Testfälle sollen auf Chrome & Firefox durchgeführt werden- Es soll dokumentiert werden wie ich die Applikation im Frontend designt haben- Alle Unterschriften müssen eingescannt werden- Der aktuelle Zeitplan wurde den beiden Experten sowie auch dem VF zugestellt	Pia Schmid, Edgar Gmür, Özcan Altin, Danyyil Luntovsky	Bern, 04.03.2019

14.2. Nachweise der Versionierung und Datensicherung

(ohne Nachweise keine Punkte im Beurteilungskriterium „Organisation der IPA Ergebnisse“)












Name	Änderungsdatum	Typ
 Zeitplan v1.0 (Mi) 20.02.xlsx	20.02.2019 16:57	Microsoft Excel-Ar...
 Zeitplan v1.1 (Do) 21.02.xlsx	21.02.2019 16:59	Microsoft Excel-Ar...
 Zeitplan v1.2 (Fr) 22.02.xlsx	22.02.2019 16:51	Microsoft Excel-Ar...
 Zeitplan v1.3 (Mo) 25.02.xlsx	25.02.2019 16:56	Microsoft Excel-Ar...
 Zeitplan v1.4 (Mi) 27.02.xlsx	27.02.2019 16:46	Microsoft Excel-Ar...
 Zeitplan v1.5 (Do) 28.02.xlsx	28.02.2019 17:12	Microsoft Excel-Ar...
 Zeitplan v1.6 (Fr) 01.03.xlsx	01.03.2019 17:09	Microsoft Excel-Ar...
 Zeitplan v1.7 (Mo) 04.03.xlsx	06.03.2019 16:37	Microsoft Excel-Ar...
 Zeitplan v1.8 (Mi) 06.03.xlsx	07.03.2019 17:05	Microsoft Excel-Ar...
 Zeitplan v1.9 (Do) 07.03.xlsx	08.03.2019 09:46	Microsoft Excel-Ar...
 Zeitplan v2.0 (Fr) 08.03.xlsx	08.03.2019 09:46	Microsoft Excel-Ar...

Abbildung 35 Nachweis Zeitplan












Name	Änderungsdatum	Typ
 Dokumentation v1.0 (Mi) 20.02.docx	20.02.2019 16:50	Microsoft Word-D...
 Dokumentation v1.1 (Do) 21.02.docx	21.02.2019 16:58	Microsoft Word-D...
 Dokumentation v1.2 (Fr) 22.02.docx	22.02.2019 16:48	Microsoft Word-D...
 Dokumentation v1.3 (Mo) 25.02.docx	25.02.2019 17:11	Microsoft Word-D...
 Dokumentation v1.4 (Mi) 27.02.docx	27.02.2019 17:04	Microsoft Word-D...
 Dokumentation v1.5 (Do) 28.02 .docx	28.02.2019 16:55	Microsoft Word-D...
 Dokumentation v1.6 (Fr) 01.03.docx	01.03.2019 16:04	Microsoft Word-D...
 Dokumentation v1.7 (Mo) 04.03.docx	04.03.2019 17:05	Microsoft Word-D...
 Dokumentation v1.8 (Mi) 06.03.docx	06.03.2019 16:55	Microsoft Word-D...
 Dokumentation v1.9 (Do) 07.03.docx	07.03.2019 18:34	Microsoft Word-D...
 Dokumentation v2.0 (Fr) 08.03.docx	08.03.2019 10:55	Microsoft Word-D...

Abbildung 36 Nachweis Dokumentation



14.3. Sourcecode/Skripte

(ohne Bibliotheken; Eigencode muss erkennbar sein - ohne Sourcecodes/Skripte im IPA-Dokumenten-Anhang werden somit keine Punkte bei den jeweiligen Beurteilungskriterien vergeben).

Migration Skript Anleitung

Info

Dieses Skript dient zur User Migration für das TFBern Projektportfolio.

Dabei werden Benutzer vom Active Directory (PH-AD-LOA-908-001) in den Organisational Units Lernende und Lehrpersonen auf die Datenbank (*datenbankname hier einfügen*) migriert.

Es werden nur Benutzer migriert, die noch nicht in der Datenbank vorhanden sind.

Ausführung

Das Skript sollte nur von berechtigten Personen genutzt werden. Sollten Probleme bei der Ausführung auftauchen, kontaktieren sie bitte Danyyil Luntovsky (Lernender 3. Lehrjahr)

- 1) Navigieren sie in das Verzeichnis *C:/web/Projectportfolio/migration*

```
PS C:\> cd C:\web\Projectportfolio\migration
```

- 2) Geben sie folgenden Befehl ein:

```
PS C:\web\Projectportfolio\migration> node migration.js
```

Das Skript wird nun die erwähnten Aktionen durchführen.



Db-create.sql

```
/**
 * Author:      Danyyil Luntovsky
 * File name:   db-create.js
 * Version:     1.0
 * Description: Projectportfolio DB-Create Script
 */

create database if not exists projectportfolio;

use projectportfolio;

-- Cretes User with permission only granted to the Projectportfolio DB
CREATE USER 'projectportfolio_user'@'localhost' IDENTIFIED BY 'gibbiX12345';
GRANT ALL PRIVILEGES ON projectportfolio.* TO 'projectportfolio_user'@'localhost';
ALTER USER 'projectportfolio_user'@'localhost' IDENTIFIED WITH mysql_native_password BY 'gibbiX12345';

create table Status(
ID int auto_increment not null,
Status varchar(50),
primary key (ID)
);

create table Project(
ID int auto_increment not null,
Title varchar(100) not null,
Description text,
Protocol text,
CreatedAt date not null,
Status int not null,
Visible boolean not null,
primary key (ID),
foreign key (Status) references Status (ID)
);

create table Project_Relation(
ID int auto_increment not null,
Project_ID int,
Related_Project_ID int,
primary key (ID),
foreign key (Project_ID) references Project (ID),
foreign key (Related_Project_ID) references Project (ID)
);

-- Table for storing the Main OU from a user
```



```
create table OUTop(  
ID int auto_increment not null,  
OUTop varchar(255),  
primary key (ID)  
);  
  
-- Table for storing the Sub OU from a user  
create table OUSub(  
ID int auto_increment not null,  
OUSub varchar(255),  
primary key (ID)  
);  
  
create table User(  
ID int auto_increment not null,  
Initials varchar(100),  
Firstname varchar(100),  
Lastname varchar(100),  
OUTop int,  
OUSub int,  
primary key (ID),  
foreign key (OUTop) references OUTop(ID),  
foreign key (OUSub) references OUSub(ID)  
);  
  
create table Project_User(  
ID int auto_increment not null,  
Project_ID int,  
User_ID int,  
Creator boolean,  
primary key (ID),  
foreign key (Project_ID) references Project (ID),  
foreign key (User_ID) references User (ID)  
);
```




Db-insert.sql

```
/**
 * Author:      Danyyil Luntovsky
 * File name:   db-insert.js
 * Version:     1.0
 * Description: Projectportfolio DB-Insert Script
 */

use projectportfolio;

insert into Status(Status)
values ('Neu'), ('Aktiv'), ('Beendet');
```



server.js (Migration)

```
/**
 * Author:      Danyyil Luntovsky
 * File name:   server.js
 * Version:     1.0
 * Description: This script provides the functionality to migrate users from the Active Directory server
 *              into the Database. The fetched users bolng to the organisational units 'Lehrpersonen' and
 *              'Lernende'.
 *              Already existing users in the database will be remained untouched.
 */

//Loading module 'ldapjs', used to establish connection against the AD
const LDAP = require('ldapjs')
//Loading module 'mysql', adapter that is required by 'knexjs' to make requests against the DB
const MYSQL = require('mysql')
//Loading file 'knex', used for MySQL query building
const KNEX_CONFIG = require('./knexfile.js')
const KNEX = require('knex')(KNEX_CONFIG)

//Options for the search request against the AD
const OPTS = {
  filter: '(objectclass=user)',
  scope: 'sub',
  attributes: ['givenName', 'sn', 'userPrincipalName', 'dn'],
}

//Defines role as client-ldap and AD server IP
let CLIENT = LDAP.createClient({
  url: 'ldap://10.10.10.21'
})

//Establish connetcion with the AD server using DN and password, error callback
CLIENT.bind('CN=Dany,CN=Users,DC=ipa,DC=local', 'gibbiX12345', function(err) {
  err ? console.log(err) : ''
});

/**
 * @param {String} ou
 * @returns {Array}
 * On call, this function will make an request against the AD server
 * to select alle users from the OU passed through params
 */
function getOUUsers(ou) {
  return new Promise(function(resolve, reject) {

    //define users array
  })
}
```



```
let users = []

//search all users insede entered OU
CLIENT.search(`OU=${ou},DC=ipa,DC=local`, OPTS, function(err, res) {

    // each user that is found in this particular OU, will pass through this process
    res.on('searchEntry', function(entry) {

        //short domainlogon name until at-sign (in16dojo@ipa.local > in16dojo)
        initialsToken = entry.object.userPrincipalName.substring(0,
entry.object.userPrincipalName.indexOf('@'))
        //cut DN name to split OU information
        ouToken = entry.object.dn.split(",").slice(1, -2)
        ouTop = ouToken[ouToken.length-1].substring(3)
        ouSub = ouToken.slice(0, -1).toString().substring(3)

        //Populate array users with fetched AD data
        users.push({
            Firstname: entry.object.givenName,
            Lastname: entry.object.sn,
            Initials: initialsToken,
            OUTop: ouTop,
            OUSub: ouSub
        })
    })
    //on error callback
    res.on('error', function(err) {
        reject(`ERROR: An error ocured while fetching data from the '${ou}' organisational unit:
${err}`)
    })
    //on last request
    res.on('end', function(result) {
        console.log(`LOG: Fetching users from OU '${ou}' successful!`)
        //Promise succeeded, return users array from particular OU
        resolve(users)
        console.log('done fetching AD')
    })
})

})

}

/**
 * @returns {Array}
 * Calls getOUUsers
 * Does summerize recieved OU Users into an array and returns it

```



```
*/
async function getAllOUUsers() {

  //Names of the OU's on the AD server
  const ORGANISATIONAL_UNITS = ['Lernende', 'Lehrpersonen']

  let allADUsers = []

  //Makes two AD requests
  for(let i = 0; i < ORGANISATIONAL_UNITS.length; i++){
    allADUsers.push((await getOUUsers(ORGANISATIONAL_UNITS[i])))
  }

  CLIENT.destroy()

  // adopted from https://stackoverflow.com/questions/10865025/merge-flatten-an-array-of-arrays-in-javascript
  allADUsers = [].concat.apply([], allADUsers);

  return allADUsers
}

/**
 * @returns {Array}
 * gets all initials in DB table user and maps the response
 */
function getDBUsers() {
  return new Promise(function(resolve, reject) {

    let initialsDB = []

    KNEX.select('Initials').from('User')
      .then(function(entry) {
        if(entry.length === 0) {
          resolve()
        }else{
          for (let i = 0; i < entry.length; i++) {
            initialsDB.push({ Initials: entry[i].Initials })
          }
        }
        resolve(initialsDB)
        console.log('done fetching DB')
      })
  })
}
```



```
/**
 * @param {Array} usersAD
 * @param {Array | undefined} usersDB
 * @returns {Array}
 * filters already existing users on th DB based on the initials
 * returns users that arent registered in the table user
 * if there are no user to compare (undifined DB response) all AD recieved Entries will passed through
 */
function compareUsers(usersAD, usersDB) {

  if (usersDB !== undefined) {
    userInitialsDB = usersDB.map(x => x.Initials)
    comparedUsers = []

    //Hier hatte ich Schwierigkeiten die arrays zu vergleichen
    comparedUsers = usersAD.filter(val => !userInitialsDB.includes(val.Initials))
    if (comparedUsers.length <= 0) {
      console.log('LOG: Everything up to date, no actions made.')
    }

    return comparedUsers
  } else {
    return usersAD
  }
}

/**
 * @param {Array} users
 * the migration process will insert the compared users into the DB
 * first of all it checks if the OU's are already defined in the DB
 * if not the OU's will be inserted and the row ID saved and applied on the user Array
 * last follows the insert of the User array into the user table
 */
async function migrateUsers(users) {
  //
  for(let i = 0; i < users.length; i++){
    let ouTopId = await KEX('OUTop').select('ID').limit(1)
      .where('OUTop', users[i].OUTop)

    let ouSubId = await KEX('OUSub').select().limit(1)
      .where('OUSub', users[i].OUSub)

    if(ouTopId <= 0){
      ouTopId = await KEX('OUTop').insert({
        OUTop: users[i].OUTop
      })
    }
  }
}
```



```
    })
    users[i].OUTop = ouTopId[0]
  }else{
    users[i].OUTop = ouTopId[0].ID
  }

  if(ouSubId <= 0){
    if(users[i].OUSub != ''){
      ouSubId = await KNEW('OUSub').insert({
        OUSub: users[i].OUSub
      })
      users[i].OUSub = ouSubId[0]
    }else{
      users[i].OUSub = null
    }
  }else{
    users[i].OUSub = ouSubId[0].ID
  }
}

await KNEW('user').insert(users)

console.log('Migrtation successful')

process.exit(0)
}

//From here all the function are called via promise
Promise.all([getAllOUUsers(), getDBUsers()])
  .then(users => migrateUsers(compareUsers(users[0], users[1])))
  .catch(err => console.log(err))
```



Knexfile.js (Migration)

```
/**
 * Author:      Danyyil Luntovsky
 * File name:   knexfile.js
 * Version:     1.0
 * Description: configuration for connection to a mysql DB
 *              also sets DB Name, DB user name, and passwort
 */

module.exports = {
  client: 'mysql',
  connection: {
    host: '127.0.0.1',
    user: 'projectportfolio_user',
    password: 'gibbiX12345',
    database: 'Projectportfolio'
  }
}
```



Package.json (Migration)

```
/**
 * Author:      Danyyil Luntovsky
 * File name:   package.json
 * Version:     1.0
 */

{
  "name": "server",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "dependencies": {
    "knex": "^0.16.3",
    "ldapjs": "^1.0.2",
    "mysql": "^2.16.0",
  },
  "devDependencies": {
    "nodemon": "^1.18.10",
  },
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```




Config.js (Backend)

```
/**
 * Author:      Danyyil Luntovsky
 * File name:   config.js
 * Version:     1.0
 * Description: defines the Port
 *              and the jwt secret to create the Token
 */

module.exports = {
  port: process.env.PORT || 1337,
  authentication: {
    //secret key for web token
    jwtSecret: process.env.JWT_SECRET || 'secret'
  }
}
```



Knexfile.js (Backend)

```
/**
 * Author:      Danyyil Luntovsky
 * File name:   knexfile.js
 * Version:     1.0
 * Description: configuration for connection to a mysql DB
 *              also sets DB Name, DB user name, and password
 */

module.exports = {
  development: {
    client: 'mysql',
    connection: {
      host: '127.0.0.1',
      user: 'projectportfolio_user',
      password: 'gibbiX12345',
      database: 'projectportfolio',
      timezone: 'UTC',
      dateStrings: true
    },
  },
}
```



AuthenticationController.js (Backend)

```
/**
 * Author:      Danyyil Luntovsky
 * File name:   AuthenticationController.js
 * Version:     1.0
 * Description: gets the requests from the routes.js file
 *              involves logic
 *              manages request permissions
 *              also generates and sends the jwt token
 */

const USER = require('../db/models/User')
const JWT = require('jsonwebtoken')
const CONFIG = require('../config/config')

//Create Token
function jwtSignUser(user) {
  //token validation one week
  const ONE_WEEK = 60 * 60 * 24 * 7
  //encrypt the user object within the token
  return JWT.sign({user}, CONFIG.authentication.jwtSecret, {
    expiresIn: ONE_WEEK
  })
}

module.exports = {

  async login (req, res) {
    try {
      //get user input from frontend API

      var {initials, password} = req.body

      if(initials.length == 0 || password.length == 0 ){
        return res.status(403).send({
          error: 'Invalide Login Information'
        })
      }

      const user = await USER.compareLogin({
        initials: initials,
        password: password
      })

      //if compareLogin function returns false
    }
  }
}
```



```
        if(!user){
            return res.status(403).send({
                error: 'Invalid Login Information'
            })
        }else{
            //sends user object with web token
            res.send({
                user: user[0],
                token: jwtSignUser(user[0])
            })
        }
    } catch (err) {
        //if internal error
        console.log(err);

        res.status(500).send({
            error: 'An error has occurred'
        })
    }
},

async getAll(req, res) {
    try{
        if(req.user.modify == false){
            res.status(403).send({
                error: 'You have no permission'
            })
        }else{
            var users = await USER.getAll()
            res.send(users)
        }
    }catch(err){
        //if internal error
        console.log(err);
        res.status(500).send({
            error: 'An error has occurred'
        })
    }
}
}
```



ProjectsController.js (Backend)

```
/**
 * Author:      Danyyil Luntovsky
 * File name:   ProjectsController.js
 * Version:     1.0
 * Description: gets the requests from the routes.js file
 *             used to interact with the Project.js Model
 *             Stores logic and permission handler
 */

const PROJECT = require('../db/models/Project')

module.exports = {

  async getAll(req, res) {
    try {
      if(req.user.modify == false){
        res.status(403).send({
          error: 'You have no permissions'
        })
      }else{
        var projects = await PROJECT.getAll()
        res.send(projects)
      }
    }catch(err) {
      //if internal error
      console.log(err);
      res.status(500).send({
        error: 'An error has occurred'
      })
    }
  },

  async getMy(req, res) {
    try {
      var projects = await PROJECT.getMy(req.params.userId)

      res.send(projects)
    }catch(err) {
      //if internal error
      res.status(500).send({
        error: 'An error has occurred'
      })
    }
  },
}
```



```
async show(req, res) {
  try{
    let {projectId} = req.params
    let userId = req.user.user[0].ID
    let permission = await PROJECT.check(userId, projectId)

    if(permission === 1 || req.user.modify == true){
      var project = await PROJECT.show(projectId)
      res.send(project)
    }else{
      res.status(403).send({
        error: 'You have no permissions'
      })
    }
  }catch(err){
    //if internal error
    console.log(err);
    res.status(500).send({
      error: 'An error has occurred'
    })
  }
},

async post(req, res) {
  try {
    if(req.user.modify == false){
      res.status(403).send({
        error: 'You have no permissions'
      })
    }else{

      let project = req.body
      //set status to New (Neu)
      project.Status = 1
      project.Visible = 1
      let projectID = await PROJECT.post(project)

      const candidats = project.Candidats
      let projectCandidats = []
      const relations = project.Relations
      let projectRelations = []

      candidats.map(user => {
```



```
        //check if candidat is creator
        if(user == req.user.user[0].ID){
            projectCandidates.push({Project_ID: projectID, User_ID: user, Creator: true})
        }else{
            projectCandidates.push({Project_ID: projectID, User_ID: user})
        }
    })

    relations.map(project => {
        projectRelations.push({Project_ID: projectID, Related_Project_ID: project})
    })

    await PROJECT.postRef(projectCandidates, projectRelations)

    res.status(201).send({projectID})
}
}catch(err){
    console.log(err);
    res.status(500).send({
        error: 'An error has occurred'
    })
}

},

async update(req, res){
    try {
        if(req.user.modify == false){
            res.status(403).send({
                error: 'You have no permissions'
            })
        }else{
            let project = req.body
            let statusId = (await PROJECT.checkStatus(project.Status))[0].ID
            project.Status = statusId

            const candidats = project.Candidates
            const relations = project.Relations

            //implement update candidats & relations

            await PROJECT.update(project)

            res.status(201).send('Project Updated!')
        }
    }catch(err){
```



```
        res.status(500).send({
            //if internal error
            error: 'An error has occurred updating this Project'
        })
    }
},

async updateProtocol(req, res) {
    try {
        let { ProjectId, Protocol } = req.body
        let UserId = req.user.user[0].ID
        let permission = await PROJECT.check(UserId, ProjectId)

        if(permission === 1 || req.user.modify == true){

            await PROJECT.updateProtocol(ProjectId, Protocol)

            res.status(201).send('Project Updated!')
        }else{
            res.status(403).send({
                error: 'You have no permissions'
            })
        }
    }

    } catch (error) {
        console.log(error);

        //if internal error
        res.status(500).send({
            error: 'An error has occurred'
        })
    }
},

async delete(req, res){
    try {
        if(req.user.modify == false){
            res.status(403).send({
                error: 'You have no permissions'
            })
        }else{
            PROJECT.delete(req.body.projectId)
            res.status(200).send('Project deleted!')
        }
    }
    }catch(err){
        console.log(err);
    }
}
```




```
        //if internal error
        res.status(500).send({
            error: 'An error has occurred'
        })
    }
}
```



Project.js (Backend)

```
/**
 * Author:      Danyyil Luntovsky
 * File name:   Project.js
 * Version:     1.0
 * Description: This Model makes only requests to the DB
 *              it returns each request to the ProjectsController.js
 *              all project interactions are passing this files
 */

const KNEX = require('../knex')

/**
 * @returns {Array}
 * Gets all projects that are visible and where the creator is true
 */
async function getAllProjects() {
  return await KNEX
    .select(
      'pro.ID as ID',
      'pro.Title',
      'pro.Description',
      'pro.Protocol',
      'pro.CreatedAt',
      'sta.Status',
      'usr.ID as Usr_ID',
      'usr.Initials',
      'usr.Firstname',
      'usr.Lastname',
    ).from('project as pro')
    .leftJoin('project_user as pro_usr', 'pro.ID', 'pro_usr.Project_ID')
    .leftJoin('user as usr', 'pro_usr.User_ID', 'usr.ID')
    .leftJoin('status as sta', 'pro.Status', 'sta.ID')
    .where('pro_usr.Creator', 1)
    .andWhere('pro.Visible', 1)
}

/**
 *
 * @param {String} userId
 * @param {String} projectId
 * @returns {boolean}
 * Checks if the user is involved in particular project
 */
async function checkUserProject(userId, projectId){
```



```
let entry = await KNEX
  .select((
    'Project_ID',
    'User_ID'
  ))
  .from('project_user')
  .where('User_ID', userId)
  .andWhere('Project_ID', projectId)

if(entry == [] || entry.length <= 0){
  return 0
}else{
  return 1
}
}

/**
 *
 * @param {String} status
 * @returns {Array}
 * Gets particular status
 */
async function checkStatus(status){
  return await KNEX('status').select('ID')
    .where('Status', status)
}

/**
 *
 * @param {String} id
 * @returns {Array}
 * Returns full Project with candidats and relations
 */
async function showProject(id) {
  return await KNEX
    .select(
      'pro.ID',
      'pro.Title',
      'pro.Description',
      'pro.Protocol',
      'pro.CreatedAt',
      'sta.ID as Status',
      KNEX.raw('GROUP_CONCAT(distinct rel.Related_Project_ID, \'%\', pro_rel.Title) as Relations'),
      KNEX.raw('GROUP_CONCAT(distinct usr.ID, \'%\', usr.Firstname, \' \', usr.Lastname) as
Candidats'),
    )
}
```



```
.from('project_user as pro_usr')
.leftJoin('user as usr', 'pro_usr.User_ID', 'usr.ID')
.leftJoin('project_relation as rel', 'pro_usr.Project_ID', 'rel.Project_ID')
.leftJoin('project as pro', 'pro_usr.Project_ID', 'pro.ID')
.leftJoin('project as pro_rel', 'rel.Related_Project_ID', 'pro_rel.ID')
.leftJoin('status as sta', 'pro.Status', 'sta.ID')
.where('pro.ID', id)
.andWhere('pro.Visible', 1)
}
```

```
/**
 *
 * @param {String} userId
 * @returns {Array}
 * Returns all projects the user is involved in
 */
async function getMyProjects(userId){
  return await KNEX
    .select(
      'pro.ID as ID',
      'pro.Title',
      'pro.Description',
      'pro.Protocol',
      'pro.CreatedAt',
      'sta.Status',
      'usr.ID as Usr_ID',
      'usr.Initials',
      'usr.Firstname',
      'usr.Lastname',
    ).from('project as pro')
    .leftJoin('project_user as pro_usr', 'pro.ID', 'pro_usr.Project_ID')
    .leftJoin('project_user as pro_usr1', 'pro.ID', 'pro_usr1.Project_ID')
    .leftJoin('user as usr', 'pro_usr1.User_ID', 'usr.ID')
    .leftJoin('status as sta', 'pro.Status', 'sta.ID')
    .where('pro_usr.User_ID', userId)
    .andWhere('pro.Visible', 1)
    .andWhere('pro_usr1.Creator', 1)
}
```

```
/**
 *
 * @param {String} project
 * @returns {Array}
 * insters the project with the passed informations
 */
async function insertProject(project) {
```



```
let projectId = await KEX('project').insert({
  Title: project.Title,
  Description: project.Description,
  Protocol: project.Protocol,
  CreatedAt: project.CreatedAt,
  Status: project.Status,
  Visible: project.Visible,
})

return projectId
}

/**
 *
 * @param {Object} candidats
 * @param {Object} relations
 * That table populates the table project_user and project_relations
 */
async function populateProjectRelatedTables(candidates, relations){

  let users = KEX('project_user').insert(candidates)
  let projects = KEX('project_relation').insert(relations)

  await Promise.all([users, projects])
}

/**
 *
 * @param {String} project
 * Updates project main values
 */
async function updateProject(project) {

  return await KEX('project').where('ID', project.ID)
    .update({
      Title: project.Title,
      Description: project.Description,
      Protocol: project.Protocol,
      CreatedAt: project.CreatedAt,
      Status: project.Status,
    })
}

/**
 *
```



```
* @param {String} projectId
* @param {String} protocol
* Updates the protocol from an Project
*/
async function updateProtocol(projectId, protocol){
  return await KNEX('project').where('ID', projectId)
    .update({
      Protocol: protocol
    })
}

/**
 *
 * @param {String} projectId
 * Marks Project as deleted or not visible
 */
async function deleteProject(projectId) {
  return await KNEX('project').where('ID', projectId)
    .update({Visible: 0})
}

// exporting all functions listed in this file
module.exports = {
  getAll: getAllProjects,
  getMy: getMyProjects,
  show: showProject,
  post: insertProject,
  postRef: populateProjectRelatedTables,
  check: checkUserProject,
  checkStatus: checkStatus,
  update: updateProject,
  updateProtocol: updateProtocol,
  delete: deleteProject
}
```



User.js (Backend)

```
/**
 * Author:      Danyyil Luntovsky
 * File name:   User.js
 * Version:     1.0
 * Description: This Model makes Request against the AD and the DB Server
 *              and returns the packages back to the authenticationController
 *              an exception is the function getFullName() that is called from the passport.js module
 */

// Knex instance
const KNEX = require('../knex')

// Ldap instance
const LDAP = require('ldapjs')

/**
 * @param {String} userInput
 * @returns {Array}
 * takes initials, searches in DB, if entry matches, returns Array with full name and all user information
 form the DB
 */
async function getFullName(userInput) {
    let user = await KNEX.select('User.ID', 'Initials', 'Firstname', 'Lastname', 'top.OUtop',
    'sub.OUSub').from('User')
        .leftJoin('outop as top', 'User.OUtop', 'top.ID')
        .leftJoin('ousub as sub', 'User.OUSub', 'sub.ID')
        .where('Initials', userInput)

    if (user.length == 1) {
        return user
    } else if (user.length == 0) {
        return
    } else {
        return
    }
}

/**
 * @returns {Array}
 */
async function getAll() {
    return await KNEX.select().from('User')
}

/**
```



```
* @param {String} userInput
* @param {Array} user
* @returns {Array}
* takes entered user entered password, the full name and the whole user Object
* if full name and password matches the AD entry the user Object will be returned
* else not authenticated
*/
function validateLogin(userInput, user) {
  return new Promise(function(resolve, reject){

    const CLIENT = LDAP.createClient({
      url: 'ldap://10.10.10.21'
    })

    if (userInput.password === '' || userInput.password === undefined){
      reject('wrong login information')
    }else{

      function filterClass(str){
        if(str !== null){
          return `OU=${str},`
        }else{
          return ''
        }
      }

      CLIENT.bind(`CN=${user[0].Firstname}
${user[0].Lastname},${filterClass(user[0].OUSub)}OU=${user[0].OUTop},DC=ipa,DC=local`,
`${userInput.password}`, function(err) {
        CLIENT.destroy()
        if(err){
          reject('wrong login information')
        }else{
          resolve(user)
        }

        CLIENT.destroy()
      })
    }
  })
}

/**
* @param {Object} params
* @returns {Array | boolean}
```




```
* calls getFullName() function with initials as params, to get the full user Object
* then calls validateLogin() with Users login information params and the full User Object
*/
async function compareLogin(params) {

    var loggedIn = false

    fullName = await getFullName(params.initials)

    try {
        return await validateLogin(params, fullName)
    } catch (err) {
        return false
    }

}

//function that will be called from the Authentication controller and from passport.js
module.exports = {
    compareLogin: compareLogin,
    getOne: getFullName,
    getAll: getAll
}
```



Knex.js (Backend)

```
/**
 * Author:      Danyyil Luntovsky
 * File name:   knex.js
 * Version:     1.0
 * Description: The knex builder.
 *
 *             This File binds summerizes all knex config and
 *             makes ist easy to import, with only this file
 */

//gets the connection details and exports knex require
var environment = process.env.NODE_ENV || 'development';
var config = require('../config/knexfile')[environment];
module.exports = require('knex')(config);
```



validateRequest.js (Backend)

```
/**
 * Author:      Danyyil Luntovsky
 * File name:   validateRequest.js
 * Version:     1.0
 * Description: checks if user is Lehrperson or Lernender and stores information into a prop
 */
const PASSPORT = require('passport')

//If the through passing user is a Techer, he will get modify true attached on to its object
//else modify will be set to false
module.exports = (req, res, next) => {
  PASSPORT.authenticate('jwt', function (err, user) {
    if (err || !user) {
      res.status(403).send({
        error: 'no permissions'
      })
    } else {
      if (user[0].OUTop === 'Lehrpersonen') {
        req.user = {user, modify: true}
        next()
      } else {
        req.user = {user, modify: false}
        next()
      }
    }
  })(req, res, next)
}
```



Passport.js (Backend)

```
/**
 * Author:      Danyyil Luntovsky
 * File name:   passport.js
 * Version:     1.0
 * Description: Validates the Token sent via the HTTP-Authorization header from the Frontend
 */

const PASSPORT = require('passport')
const USER = require('../db/models/User')
const CONFIG = require('../config/config')

let opts = {}
const JWT_STRATEGY = require('passport-jwt').Strategy
const EXTRACT_JWT = require('passport-jwt').ExtractJwt
//this file was creater with assistance from the website:
//http://www.passportjs.org/docs/
//gets the token out of the http-header
opts.jwtFromRequest = EXTRACT_JWT.fromAuthHeaderAsBearerToken()
//gets the jwt-secret from the config file
opts.secretOrKey = CONFIG.authentication.jwtSecret
PASSPORT.use(
  //new passport strategy
  new JWT_STRATEGY(
    opts,
    async function(jwt_payload, done){
      //gets the user from the DB, with decrypted token
      let user = await USER.getOne(jwt_payload.user.Initials)
      if(!user){
        //if user matches, return user
        return done(err, false)
      }
      if(user) {
        done(null, user)
      }else{
        done(null, false)
      }
    }
  )
)
```



App.js (Backend)

```
/**
 * Author:      Danyyil Luntovsky
 * File name:   app.js
 * Version:     1.0
 * Description: The main file of the backend.
 *              instanciets express and other dependencies
 *              Redirects Requests
 */

//Import express modul
const express = require('express')
//Import body-parser
const bodyParser = require('body-parser')
//Import Cors
const cors = require('cors')
//Import morgan
const morgan = require('morgan')
//Import config file to provide port
const config = require('./config/config')

//creates an instance of express
const app = express()
//Server LOG for requests
app.use(morgan('combined'))
//Provides req.body functionality
app.use(bodyParser.json())
//Crossplatform requests
app.use(cors())
//checks the jwt-token on each request
require('./security/passport')
//routes all requursts to routes with an instance of express
require('./routes')(app)

//catches the LDAP.js error on closed connection
if (process.env.NODE_ENV !== 'production'){
  require('longjohn')
}
process.on('uncaughtException', function (err) {
  console.error(err.stack)
  console.log("Node NOT Exiting...")
})
//makes the server listen on port 1337
app.listen(config.port, () => console.log(`Server started on port ${config.port}`))
```



routes.js (Backend)

```
/**
 * Author:      Danyyil Luntovsky
 * File name:   routes.js
 * Version:     1.0
 * Description: All Accesible routes are listed in this file.
 */
// Authentication Controller for User related request
const AUTHENTICATION_CONTROLLER = require('./controllers/AuthenticationController')
// Projects Controller for Project related request
const PROJECTS_CONTROLLER = require('./controllers/ProjectsController')
// Through passing the Validate Request will return the user with modified status
const VALIDATE_REQUEST = require('./policies/validateRequest')
//Exports function with app as param
//All request paths are defined here
module.exports = (app) => {

  app.post('/login',
    AUTHENTICATION_CONTROLLER.login)
  app.get('/allUsers',
    VALIDATE_REQUEST,
    AUTHENTICATION_CONTROLLER.getAll)
  app.get('/allProjects',
    VALIDATE_REQUEST,
    PROJECTS_CONTROLLER.getAll)
  app.get('/myProjects/:userId',
    VALIDATE_REQUEST,
    PROJECTS_CONTROLLER.getMy)
  app.get('/project/:projectId/:userId',
    VALIDATE_REQUEST,
    PROJECTS_CONTROLLER.show)
  app.post('/project',
    VALIDATE_REQUEST,
    PROJECTS_CONTROLLER.post)
  app.post('/updateProject',
    VALIDATE_REQUEST,
    PROJECTS_CONTROLLER.update)
  app.post('/updateProtocol',
    VALIDATE_REQUEST,
    PROJECTS_CONTROLLER.updateProtocol)
  app.post('/delete',
    VALIDATE_REQUEST,
    PROJECTS_CONTROLLER.delete)
}
```



Package.json (Backend)

```
/**
 * Author:      Danyyil Luntovsky
 * File name:   package.json
 * Version:     1.0
 */

{
  "name": "server",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "dependencies": {
    "body-parser": "^1.18.3",
    "cors": "^2.8.5",
    "express": "^4.16.4",
    "jsonwebtoken": "^8.4.0",
    "knex": "^0.16.3",
    "ldapjs": "^1.0.2",
    "longjohn": "^0.2.12",
    "morgan": "^1.9.1",
    "mysql": "^2.16.0",
    "passport": "^0.4.0",
    "passport-jwt": "^4.0.0"
  },
  "devDependencies": {
    "nodemon": "^1.18.10",
    "ws": "3.3.2"
  },
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```



AddProject.vue (Frontend)

```
<!--
* Author:      Danyyil Luntovsky
* File name:   addProject.js
* Version:     1.0
* Description: Create project process
-->

<template>
  <v-dialog max-width="1000px" v-model="dialog">
    <!-- Opens dialog & fetches data from DB -->
    <v-btn
      flat
      outline
      color="blue"
      slot="activator"
      :loading="loading"
      @click="getUser(); getProjects()"
    >
      <v-icon left>add</v-icon>
      <span>Neues Prokekt</span>
    </v-btn>
    <v-card>
      <v-card-title>
        <h2>Neues Projekt</h2>
      </v-card-title>
      <v-card-text>
        <v-stepper v-model="step" class="mt-1 mb-1">
          <!-- Stepper header -->
          <v-stepper-header>
            <v-spacer></v-spacer>
            <v-stepper-step :complete="step > 1" step="1">Initialisierung</v-stepper-step>
            <v-divider></v-divider>
            <v-stepper-step :complete="step > 2" step="2">Konfiguration</v-stepper-step>
            <v-divider></v-divider>
            <v-stepper-step :complete="step > 3" step="3">Übersicht</v-stepper-step>
            <v-spacer></v-spacer>
          </v-stepper-header>

          <v-stepper-items class="stepper-content-height">
            <v-stepper-content step="1">
              <v-card flat>
                <v-form class="px-3">
                  <v-text-field
                    class="mb-2"

```




```
      for="title"
      label="Titel*"
      v-model="title"
      :counter="100"
      prepend-icon="folder"
      :error-messages="`${errorTitle}`"
    >
  </v-text-field>
  <v-textarea
    class="mb-4"
    :counter="2000"
    for="description"
    prepend-icon="edit"
    label="Beschreibung"
    v-model="description"
    :error-messages="`${errorDesc}`"
  >
    </v-textarea>
  <!-- Navigation Bottom -->
  <v-divider></v-divider>
  <span class="text-xs-left grey--text">
    * Pflichtfelder
  </span>

  <div class="text-xs-right">
    <v-btn
      flat
      class="primary mx-0 mt-3"
      @click="validateForm"
    >
      <span>Weiter</span>
      <v-icon>chevron_right</v-icon>
    </v-btn>
  </div>
</v-form>
</v-card>
</v-stepper-content>

<v-stepper-content step="2">
  <v-card flat light>
    <v-form class="mb-0">
      <v-container>
        <!-- User selection -->
        <v-layout wrap>
          <v-flex xs12>
            <v-autocomplete
```

```

        box
        chips
        multiple
        label="Mitglieder"
        item-value=""
        :items="users"
        v-model="selected_users"
    >
    <template slot="selection" slot-scope="user">
        <v-chip
            close
            :selected="user.selected"
            class="chip--select-multi"
            @input="removeUser(user.item)"
        >
            <v-avatar>
                <v-icon>person</v-icon>
            </v-avatar>
            {{ user.item.Firstname }} {{ user.item.Lastname }}
        </v-chip>
    </template>
    <template slot="item" slot-scope="user">
        <template v-if="typeof user.item !== 'object'">
            <v-list-tile-content v-text="user.item"></v-list-tile-content>
        </template>
        <template v-else>
            <v-list-tile-avatar>
                <v-icon>person</v-icon>
            </v-list-tile-avatar>
            <v-list-tile-content>
                <v-list-tile-title>
                    v-html="`${user.item.Firstname} ${user.item.Lastname}`"
                >
            </v-list-tile-title>
                <v-list-tile-sub-title>
                    v-html="user.item.Initials"
                >
            </v-list-tile-sub-title>
            </v-list-tile-content>
        </template>
    </template>
</v-autocomplete>
</v-flex>
</v-layout>
<!-- Project selection -->
<v-layout wrap>

```



```
<v-flex xs12>
  <v-autocomplete
    box
    chips
    multiple
    label="Abhängige Projekte"
    item-value=""
    :items="projects"
    v-model="selected_projects"
    class="mb-0"
  >
    <template slot="selection" slot-scope="project">
      <v-chip
        close
        class="chip--select-multi"
        :selected="project.selected"
        @input="removeProject(project.item)"
      >
        <v-avatar>
          <v-icon>settings</v-icon>
        </v-avatar>
        {{ project.item.Title }}
      </v-chip>
    </template>
    <template slot="item" slot-scope="project">
      <template v-if="typeof project.item !== 'object'">
        <v-list-tile-content v-text="project.item"></v-list-tile-content>
      </template>
      <template v-else>
        <v-list-tile-avatar>
          <v-icon>settings</v-icon>
        </v-list-tile-avatar>
        <v-list-tile-content >
          <v-list-tile-title
            v-html="project.item.Title"
          >
          </v-list-tile-title>
          <v-list-tile-sub-title
            v-html="project.item.Title"
          >
          </v-list-tile-sub-title>
        </v-list-tile-content>
      </template>
    </template>
  </v-autocomplete>
</v-flex>
```



```
        </v-layout>
      </v-container>
    </v-form>
    <!-- Navigation Bottom -->
    <v-divider></v-divider>
    <div class="text-xs-center">
      <v-container pa-0>
        <v-layout row justify-space-between>
          <v-btn flat class="primary mx-0 mt-3" @click="step=1">
            <v-icon>chevron_left</v-icon>
            <span>Zurück</span>
          </v-btn>
          <v-btn flat class="primary mx-0 mt-3" @click="step=3">
            <span>Weiter</span>
            <v-icon>chevron_right</v-icon>
          </v-btn>
        </v-layout>
      </v-container>
    </div>
  </v-card>
</v-stepper-content>

<v-stepper-content step="3">
  <v-container class="px-3 py-0" ref="form">
    <v-layout wrap row justify-space-center px-4 mb-4>
      <v-flex my-3 xs6>
        <div class="caption grey--text">Projekt Titel</div>
        <div>{{ title }}</div>
      </v-flex>
      <v-flex my-3 xs12>
        <div class="caption grey--text">Berschreibung</div>
        <div>{{ description }}</div>
      </v-flex>
      <v-layout row mb-5>
        <v-flex my-3>
          <div class="caption grey--text">Teilnehmer</div>
          <div
            class="left"
            :key="user.ID"
            v-for="user in selected_users"
          >
            <v-chip small class="grey white--text">
              {{ user.Firstname }} {{ user.Lastname }}
            </v-chip>
          </div>
        </v-flex>
      </v-layout>
    </v-container>
  </v-stepper-content>
```



```
<v-flex my-3>
  <div class="caption grey--text">Abhängige Projekte</div>
  <div
    class="left"
    :key="project.ID"
    v-for="project in selected_projects"
  >
    <v-chip small class="grey white--text">
      {{ project.Title }}
    </v-chip>
  </div>
</v-flex>
</v-layout>
</v-layout>
<!-- Navigation Bottom -->
<v-divider></v-divider>
<div class="text-xs-center">
  <v-container pa-0>
    <v-layout row justify-space-between>
      <v-btn flat class="primary mx-0 mt-3" @click="step=2">
        <v-icon>chevron_left</v-icon>
        <span>Zurück</span>
      </v-btn>
      <v-btn
        flat
        :loading="loading"
        @click="createProject"
        class="success mx-0 mt-3"
      >
        Projekt erstellen</v-btn>
    </v-layout>
  </v-container>
</div>
</v-container>
</v-stepper-content>
</v-stepper-items>
</v-stepper>

</v-card-text>
</v-card>
</v-dialog>
</template>

<script>
//Date formatter to
import format from 'date-fns/format'
```

```
import ProjectService from '@services/ProjectService'

export default {
  name: "addProject",
  data() {
    return {
      title: '',
      description: '',
      loading: false,
      dialog: false,
      step: 0,
      errorTitle: [],
      errorDesc: [],
      errors: [],

      selected_users: [],
      users: [],

      selected_projects: [],
      projects: [],
    }
  },
  methods: {
    removeUser (item) {
      const user_list = this.selected_users.indexOf(item)
      if (user_list >= 0) this.selected_users.splice(user_list, 1)
    },
    removeProject (item) {
      const project_list = this.selected_projects.indexOf(item)
      if (project_list >= 0) this.selected_projects.splice(project_list, 1)
    },
    async createProject(){
      try {
        await ProjectService.createProject({
          Title: this.title,
          Description: this.description,
          Protocol: null,
          CreatedAt: this.formattedCreateDate,
          Status: 1,
          Visible: 1,
          Candidats: this.selected_users.map(user => user.ID),
          Relations: this.selected_projects.map(project => project.ID)
        })
        this.$router.go()
      } catch (error) {
        this.errors.push(error.response.data.error);
      }
    }
  }
}
```



```
}

this.loading = true
this.title = ''
this.description = ''
this.loading = false
this.dialog = false
this.step = 1
},
async getProjects(){
  this.loading = true
  this.projects = []
  this.selected_projects = []
  let allProjects = (await ProjectsService.getAllProjects()).data
  allProjects.forEach(project => this.projects.push(project))
  this.loading = false
},
async getUser(){
  this.users = []
  this.selected_users = []
  this.selected_users.push(this.$store.state.user)
  let allUsers = (await ProjectsService.getAllUsers()).data
  allUsers.forEach(user => this.users.push(user))
},
validateForm() {

  this.errorTitle = []
  this.errorDesc = []

  this.title = this.title.trim()

  if(!this.title){
    this.errorTitle.push('Bitte Titel eintragen!')
  }else if(this.title.length > 100){
    this.errorTitle.push('Titel zu lang!')
  }

  if(this.description.length > 2000){
    this.errorDesc.push('Beschreibung zu lang!')
  }

  if(this.errorTitle.length == 0 && this.errorDesc.length == 0){
    return this.step = 2
  }
}
},
```



```
computed: {  
  formattedCreateDate() {  
    let date = new Date()  
    return format(date, 'YYYY-MM-DD')  
  }  
}  
}  
</script>  
  
<style scoped>  
.bottom-card-nav{  
  bottom: 0;  
}  
</style>
```




Navbar.vue (Frontend)

```
<!--
* Author:      Danyyil Luntovsky
* File name:    addProject.js
* Version:      1.0
* Description:  Navbar imports
-->

<template>
  <div>
    <!-- Header -->
    <v-toolbar class="border" height="75" flat app>
      <v-container ma-0 pa-0>
        <v-layout align-center>

          <v-flex xs5>
            <div class="text-xs-left" v-if="this.$store.state.user">
              <addProject v-if="this.$store.state.isAuthenticated" />
              <router-link
                v-if="this.$store.state.isAuthenticated"
                :to="{ name: 'dashboard' }">
                >
                  <v-btn flat color="grey">
                    <v-icon left>dashboard</v-icon>
                    <span>Dashboard</span>
                  </v-btn>
                </router-link>
                <router-link :to="{ name: 'myProjects' }">
                  <v-btn flat color="grey">
                    <v-icon left>folder</v-icon>
                    <span>Meine Projekte</span>
                  </v-btn>
                </router-link>
              </div>
            </v-flex>

            <v-flex xs2>
              <div class="text-xs-center">
                <v-btn fab flat color="blue">
                  
                </v-btn>
              </div>
            </v-flex>

          <v-flex xs5 >
```



```
<div class="text-xs-right" v-if="this.$store.state.user">
  <v-btn flat color="grey" @click="logout">
    <span>Abmelden</span>
    <v-icon right>exit_to_app</v-icon>
  </v-btn>
</div>
</v-flex>
</v-layout>
</v-container>
</v-toolbar>
</div>
</template>
```

```
<script>
import addProject from './addProject.vue'

export default {
  name: "Navbar",
  components: {
    addProject
  },
  methods: {
    logout () {
      this.$router.push({
        name: 'login'
      })
      this.$store.dispatch('setToken', null)
      this.$store.dispatch('setUser', null)
    }
  }
}
</script>
```

```
<style scoped>
.container {
  max-width: 100%;
}

.border{
  border-bottom: solid 1px #9E9E9E !important;
}

</style>
```



showProjects (Frontend)

```
<!--
* Author:      Danyyil Luntovsky
* File name:    showProjects.vue
* Version:     1.0
-->

<template>
  <div>
    <v-container
      class="my-1"
    >

      <v-layout row class="mb-3">
        <v-tooltip top>
          <v-btn small flat color="grey" @click="sortBy('Title')" slot="activator">
            <v-icon small left>folder</v-icon>
            <span class="caption text-lowercase">nach titel</span>
          </v-btn>
          <span>Projekte nach Titel sortieren</span>
        </v-tooltip>
        <v-tooltip top>
          <v-btn small flat color="grey" @click="sortBy('CreatedAt')" slot="activator">
            <v-icon small left>person</v-icon>
            <span class="caption text-lowercase">nach ersteller</span>
          </v-btn>
          <span>Projekte nach Ersteller sortieren</span>
        </v-tooltip>
        <v-tooltip top>
          <v-btn small flat color="grey" @click="sortBy('Status')" slot="activator">
            <v-icon small left>av_timer</v-icon>
            <span class="caption text-lowercase">nach status</span>
          </v-btn>
          <span>Projekte nach Status sortieren</span>
        </v-tooltip>
        <v-spacer></v-spacer>

        <v-card flat class="grey lighten-4 mr-4" min-width="500px" >
          <v-text-field v-model="search" append-icon="search" label="Projekt Suchen">
          </v-text-field>
        </v-card>
      </v-layout>

      <v-layout row wrap justify-start>
```



```
<v-flex xs4 v-for="project in filteredProjects" :key="project.ID">
  <v-card flat class="pa-4 ma-4">
    <router-link
      tag="div"
      class="project-item"
      :to="{ name: 'project', params: {projectId: project.ID}}"
    >
      <v-layout column wrap :class="`pa-3 project-${project.Status}`">
        <v-flex xs12 md5>
          <v-layout class="mb-1">
            <v-flex>
              <div class="caption grey--text">Projekt Titel</div>
              <div class="title">{{ project.Title }}</div>
            </v-flex>
            <v-flex>
              <div class="right">
                <v-chip small :class="`${project.Status} white--text`">
                  {{ project.Status }}
                </v-chip>
              </div>
            </v-flex>
          </v-layout>
          <v-flex xs6 sm4 md2 class="mb-1">
            <div class="caption grey--text">Ersteller</div>
            <div>{{ `${project.Firstname} ${project.Lastname}` }}</div>
          </v-flex>
          <v-flex xs6 sm4 md2>
            <div class="caption grey--text">Erstellt Am</div>
            <div>{{ project.CreatedAt }}</div>
          </v-flex>
        </v-flex>
      </v-layout>
    </router-link>
  </v-card>
</v-flex>
</v-layout>
</v-container>
</div>
</template>

<script>
export default {
  name: "showProjects",
  props: [
    'projects'
```



```
],
data() {
  return {
    search: '',
  }
},
methods: {
  sortBy(prop) {
    this.projects.sort((a, b) => a[prop] < b[prop] ? -1 : 1)
  },
},
computed: {
  filteredProjects: function(){
    return this.projects.filter(project => {
      return project.Title.match(new RegExp(this.search, 'i'))
    })
  }
},
}
</script>
<style scoped>
.project-item{
  cursor: pointer;
}
.project-Aktiv {
  border-left: 4px solid #4CAF50;
}
.project-Neu {
  border-left: 4px solid #2196F3;
}
.project-Beendet {
  border-left: 4px solid #F44336;
}
.v-chip.Aktiv {
  background-color: #4CAF50;
}
.v-chip.Neu {
  background-color: #2196F3;
}
.v-chip.Beendet {
  background-color: #F44336;
}
</style>
```



Api.js (Frontend)

```
/*  
 * Author:      Danyyil Luntovsky  
 * File name:   Api.js  
 * Version:     1.0  
 */  
  
import axios from 'axios';  
import store from '@store/store'  
  
export default () => {  
  return axios.create({  
    baseURL: `http://localhost:1337/`,  
    headers: {  
      Authorization: `Bearer ${store.state.token}`  
    }  
  })  
}
```



AuthenticationService.js (Frontend)

```
/*  
 * Author:      Danyyil Luntovsky  
 * File name:   AuthenticationService.js  
 * Version:     1.0  
 */  
  
import Api from '@services/Api';  
  
export default{  
  
  login (credentials){  
    return Api().post('login', credentials);  
  }  
  
}
```



ProjectsService.js (Frontend)

```
/*
 * Author:      Danyyil Luntovsky
 * File name:   ProjectsService.js
 * Version:     1.0
 */
import Api from '@services/Api';
export default{

  getAllUsers() {
    return Api().get('allUsers');
  },

  getAllProjects() {
    return Api().get('allProjects');
  },

  getMyProjects(userId) {
    return Api().get(`myProjects/${userId}`)
  },

  showProject(projectId, userId) {
    return Api().get(`project/${projectId}/${userId}`)
  },

  createProject(project) {
    return Api().post('project', project);
  },

  updateProject(project) {
    return Api().post('updateProject', project);
  },

  updateProtocol(protocol) {
    return Api().post('updateProtocol', protocol)
  },

  deleteProject(projectId) {
    return Api().post('delete', {projectId})
  }

}
```




Store.js (Frontend)

```
/*
 * Author:      Danyyil Luntovsky
 * File name:   store.js
 * Version:     1.0
 */

import Vue from 'vue'
import Vuex from 'vuex'
import createPersistedState from 'vuex-persistedstate'

Vue.use(Vuex)

export default new Vuex.Store({
  plugins: [
    createPersistedState()
  ],
  state: {
    token: null,
    user: null,
    isUserLoggedIn: false,
    isAuthenticated: false
  },
  mutations: {
    setToken (state, token) {
      state.token = token
      if (token){
        state.isUserLoggedIn = true
      }else{
        state.isUserLoggedIn = false
      }
    },
    setUser (state, user) {
      state.user = user
      if(!user){
        state.isAuthenticated = false
      }else if(user.OUTop === 'Lehrpersonen'){
        state.isAuthenticated = true
      }else{
        state.isAuthenticated = false
      }
    }
  },
  actions: {
```



```
    setToken ({commit}, token) {  
      commit('setToken', token)  
    },  
    setUser ({commit}, user) {  
      commit('setUser', user)  
    }  
  }  
})
```

Dashboard.vue (Frontend)

```
<!--
* Author:      Danyyil Luntovsky
* File name:    Dashboard.vue
* Version:      1.0
-->

<template>
<div class="dashboard">
  <v-flex class="my-5 mx-1" color="grey">
    <v-icon right>keyboard_arrow_right</v-icon>
    <span class="title grey--text text-xs-left">Dashboard</span>
  </v-flex>

  <showProjects :projects="projects"/>
</div>
</template>

<script>
import ProjectService from '@services/ProjectService'
import showProjects from '@components/showProjects'

export default {
  name: "Dashboard",
  components: {
    showProjects
  },
  data() {
    return {
      search: '',
      projects: []
    }
  },
  async mounted() {
    this.projects.push(...(await ProjectService.getAllProjects()).data.reverse())
  }
}
</script>
```

Login.vue (Frontend)

```
<!--
* Author:      Danyyil Luntovsky
* File name:    Login.vue
* Version:     1.0
-->
<template>
  <v-container class="login" mt-5>
    <v-layout row justify-center>
      <v-flex xs8 >
        <div class="white">
          <v-toolbar flat dense class="login-header blue--text">
            <v-toolbar-title>Login</v-toolbar-title>
          </v-toolbar>

          <div class="pl-4 pr-4 pt-2 pb-2 mt-3">
            <v-text-field
              label="Kürzel"
              v-model="loginName"
              color="blue"
            ></v-text-field>
            <br>
            <v-text-field
              label="Passwort"
              color="blue"
              type="password"
              v-model="password"
              @keydown.enter="login()"
            ></v-text-field>
            <br>
            <div class="danger-alert" v-html="error" />
            <v-btn
              @click="login"
              color="blue"
              class="white--text mb-3"
            >
              Anmelden
            </v-btn>
          </div>
        </div>
      </v-flex>
    </v-layout>
  </v-container>
</template>
```



```
<script>
import AuthenticationService from '@services/AuthenticationService'
export default {
  name: "Login",
  data() {
    return {
      loginName: '',
      password: '',
      error: null
    }
  },
  methods: {
    async login() {
      try {
        const RESPONSE = await AuthenticationService.login({
          initials: this.loginName,
          password: this.password
        })
        this.$store.dispatch('setToken', RESPONSE.data.token)
        this.$store.dispatch('setUser', RESPONSE.data.user)
        if(this.$store.state.isAuthenticated){
          this.$router.push({
            name: 'dashboard'
          })
        }else{
          this.$router.push({
            name: 'myProjects'
          })
        }
        this.email = '';
        this.password = '';
        this.error = null;
      } catch (err) {
        this.error = err.response.data.error;
      }
    }
  }
}
</script>
<style scoped>
.login-header{
  border-bottom: 1px solid #2196F3;
}
</style>
```

MyProjects.vue (Frontend)

```
<!--
* Author:      Danyyil Luntovsky
* File name:    MyProjects.vue
* Version:      1.0
-->
<template>
<div class="myProjects">
  <v-flex class="my-5 mx-1" color="grey">
    <v-icon right>keyboard_arrow_right</v-icon>
    <span class="title grey--text text-xs-left">Meine Projekte</span>
  </v-flex>

  <showProjects :projects="projects"/>
</div>
</template>

<script>
import ProjectService from '@services/ProjectService'
import showProjects from '@components/showProjects'

export default {
  name: "MyProjects",
  components: {
    showProjects
  },
  data() {
    return {
      search: '',

      projects: []
    }
  },
  methods: {
    sortBy(prop) {
      this.projects.sort((a, b) => a[prop] < b[prop] ? -1 : 1)
    },
  },
  computed: {
    filteredProjects: function(){
      return this.projects.filter((project) => {
        return project.Title.match(new RegExp(this.search, 'i'))
      })
    }
  },
}
```



```
    async mounted() {
      this.projects = (await ProjectsService.getMyProjects(this.$store.state.user.ID)).data.reverse()
    }
  }
</script>

<style scoped>

.project-item{
  cursor: pointer;
}

.project.complete {
  border-left: 4px solid #3cd1c2;
}

.project.ongoing {
  border-left: 4px solid orange;
}

.project.overdue {
  border-left: 4px solid firebrick;
}

.v-chip.complete {
  background-color: #3cd1c2;
}

.v-chip.ongoing {
  background-color: orange;
}

.v-chip.overdue {
  background-color: firebrick;
}

</style>
```



ViewProject.vue (Frontend)

```
<!--
* Author:      Danyyil Luntovsky
* File name:   ViewProject.vue
* Version:     1.0
-->
<template>
  <div class="viewProject">
    <v-container>
      <v-layout>
        <v-btn
          flat
          outline
          color="grey"
          class="mt-5 ml-5"
          @click="goBack"
        >
          <v-icon>keyboard_arrow_left</v-icon>
          <span>Zurück</span>
        </v-btn>
        <v-spacer></v-spacer>
        <v-btn
          flat
          outline
          color="blue"
          v-if="!showForm"
          class="mt-5 mr-3"
          v-show="this.$store.state.isAuthenticated"
          @click="showForm =! showForm"
        >
          <v-icon left>edit</v-icon>
          <span>bearbeiten</span>
        </v-btn>
        <v-menu left offset-y v-if="!showForm" close-on-content-click>
          <v-btn
            fab
            flat
            small
            color="blue"
            slot="activator"
            class="mt-5 mr-5 more-btn"
            v-show="this.$store.state.isAuthenticated"
          >
            <v-icon large>more_vert</v-icon>
          </v-btn>
        </v-menu>
      </v-layout>
    </v-container>
  </div>
</template>
```




```
<v-list offset-x >
  <v-dialog v-model="dialog" width="500px">
    <v-list-tile @click="true" slot="activator">
      <v-list-tile-title>Projekt löschen</v-list-tile-title>
    </v-list-tile>
    <v-card>
      <v-card-title class="headline red lighten-1 white--text" primary-title>
        Projekt {{project.Title}} löschen
      </v-card-title>
      <v-card-text class="grey--text text--darker-2">
        Wenn Sie das Projekt löschen, kann es nur von einem
        Administrator wiederhergestellt werden.
        Sind Sie sich sicher, dass sie <b> {{project.Title}} </b>
        löschen wollen?
      </v-card-text>
      <v-divider></v-divider>
      <v-card-actions>
        <v-btn color="grey darker-2" flat @click="dialog = false">
          Abbrechen
        </v-btn>
        <v-spacer></v-spacer>
        <v-btn color="red" flat @click="deleteProject()">
          Löschen
        </v-btn>
      </v-card-actions>
    </v-card>
  </v-dialog>
</v-list>
</v-menu>
<v-btn
  flat
  outline
  color="blue"
  class="mt-5 mr-5"
  @click="validateForm" v-if="showForm"
>
  <v-icon left>done</v-icon>
  <span>save</span>
</v-btn>
<v-btn
  flat
  outline
  color="blue"
  class="mt-5 mr-5"
  @click="showForm = ! showForm"
  v-if="showForm"
>
```



```
>
  <v-icon left>close</v-icon>
  <span>cancel</span>
</v-btn>
</v-layout>
</v-container>
<v-card flat class="pt-1 pb-3 pl-5 pr-5 ma-5 project-card">
<v-container class="my-5">
  <v-layout row wrap justify-space-around>
    <v-flex xs10>
      <v-layout wrap row>
        <!-- Project Title -->
        <v-flex>
          <div v-if="!showForm" class="caption grey--text">Projekt Titel</div>
          <v-text-field
            v-if="showForm"
            box
            for="title"
            color="blue"
            :counter="100"
            v-model="title"
            label="Project title"
            :error-messages="`${errorTitle}`"
          ></v-text-field>
          <h1 v-else class="headline grey--text text--darken-3 wordwrap">{{project.Title}}</h1>
        </v-flex>
        <!-- Project Status -->
        <v-flex text-xs-right>
          <div v-if="!showForm">
            <div class="caption grey--text mr-1">Status</div>
            <v-chip :color="`${status} white--text`">
              {{ status }}
            </v-chip>
          </div>
          <v-select
            box
            v-else
            item-value=""
            text-xs-right
            label="Status"
            v-model="status"
            item-text="Status"
            class="select-box"
            :items="statusItems"
          ></v-select>
        </v-flex>
      </v-layout>
    </v-flex>
  </v-layout>
</v-container>
</v-card>
</v-container>
</v-layout>
</v-main>
</v-app>
```



```
</v-layout>
</v-flex>
<!-- Project description -->
<v-flex xs10 mt-4>
  <div v-if="!showForm" class="caption grey--text">Projekt Beschreibung</div>
  <v-textarea
    v-if="showForm"
    box
    auto-grow
    color="blue"
    :counter="2000"
    for="description"
    v-model="description"
    label="Project description"
    :error-messages="'${errorDesc}`"
  ></v-textarea>
  <div v-else class="grey--text text--darken-3 wordwrap">{{project.Description}}</div>
</v-flex>
<v-flex xs10>
  <v-layout row justify-space-between>
    <!-- Project members -->
    <v-flex xs5 mt-4>
      <div v-if="!showForm">
        <div v-if="data[1]">
          <div class="caption grey--text">Kandidaten</div>
          <v-chip v-for="user in data[1][0]" :key="user.id">
            <v-avatar>
              <v-icon>person</v-icon>
            </v-avatar>
            {{ user.name }}
          </v-chip>
        </div>
      </div>
    </v-flex>
    <!-- Relates to Projects -->
    <v-flex xs5 mt-4 >
      <div v-if="!showForm">
        <div v-if="data[0]">
          <div class="caption grey--text">Abhängigkeiten</div>
          <v-chip v-for="relation in data[0][0]" :key="relation.id" @click="navTo(relation.id)">
            <v-avatar>
              <v-icon>settings</v-icon>
            </v-avatar>
            {{ relation.name }}
          </v-chip>
        </div>
      </div>
    </v-flex>
  </v-layout>
</v-flex>
```



```
        </div>
      </v-flex>
    </v-layout>
  </v-flex>
  <v-flex xs10 mt-4>
    <div class="caption grey--text">Protokoll</div>
    <v-textarea
      box
      auto-grow
      color="blue"
      for="protocol"
      :counter="30000"
      append-icon="edit"
      v-model="protocol"
      :error-messages="\`${errorProtocol}`"
    >
    </v-textarea>
    <v-btn
      flat
      outline
      v-if="!showForm"
      :loading="loading"
      @click="validateProtocol"
      class="blue--text right"
    >Speichern
    </v-btn>
  </v-flex>
</v-layout>
</v-container>
</v-card>
</div>
</template>

<script>
import ProjectsService from "@services/ProjectsService"

export default {
  data() {
    return {
      showForm: false,
      toggle: true,
      dialog: false,
      error: null,
      //Should be relaced afterwards with dynamic request from DB
      statusItems: [
        {ID: 1, Status: 'Neu'},

```



```
{ID: 2, Status: 'Aktiv'},
{ID: 3, Status: 'Beendet'}
],
loading: false,

errorTitle: [],
errorDesc: [],
errorProtocol: [],

data: [],
project: [],

title: '',
description: '',
protocol: '',
status: ''
}
},
methods: {
  navTo(ID){
    this.$router.push({ name: 'project', params: { projectId: ID }})
    this.$router.go()
  },
  goBack(){
    this.$router.go(-1)
    this.$router.go()
  },
  removeUser (item) {
    const user_list = this.selected_users.indexOf(item)
    if (user_list >= 0) this.selected_users.splice(user_list, 1)
  },
  removeProject (item) {
    const project_list = this.selected_projects.indexOf(item)
    if (project_list >= 0) this.selected_projects.splice(project_list, 1)
  },
  formatted(data){
    if(data){
      this.relationsRaw = data.split(/,|%/)
      let x = 0
      let id = 0
      let container = []
      this.relationsRaw.forEach(rel => {
        if(x % 2){
          container.push({id: id, name: rel})
        }else{
          id = rel
        }
      })
    }
  }
}
```



```
        }
        x++
    })
    this.data.push([container])
  }else{
    this.data.push([0])
  }
},
async updateProtocol() {
  try{
    this.loading = true
    await ProjectsService.updateProtocol({
      ProjectId: this.project.ID,
      Protocol: this.protocol
    })
    this.$router.go()
  } catch (error) {
    this.error = error.response.data.error;
  }
  this.loading = false
},
async updateProject() {
  try {
    await ProjectsService.updateProject({
      ID: this.project.ID,
      Title: this.title,
      Description: this.description,
      Protocol: this.protocol,
      CreatedAt: this.formattedCreateDate,
      Status: this.status,
    })
    this.$router.go()
  } catch (error) {
    this.error = error.response.data.error;
  }
},
async deleteProject() {
  try {
    await ProjectsService.deleteProject(this.project.ID)
    this.$router.push({
      name: 'dashboard'
    })
    this.$router.go()
  } catch (error) {
    this.error = error.response.data.error;
  }
}
```



```
    },
    validateForm() {
      this.errorTitle = []
      this.errorDesc = []
      this.errorProtocol = []

      if(!this.title){
        this.errorTitle.push('Bitte Titel eintragen!')
      }else if(this.title.length > 100){
        this.errorTitle.push('Titel zu lang!')
      }

      if(this.description.length > 2000){
        this.errorDesc.push('Beschreibung zu lang!')
      }

      if(this.protocol == null){
        this.protocol = ''
      }

      if(this.protocol.length > 30000){
        this.errorProtocol.push('Protokoll zu lang!')
      }

      if(this.errorTitle.length == 0 && this.errorDesc.length == 0 && this.errorProtocol.length == 0){
        this.updateProject()
      }
    },
    validateProtocol() {
      this.errorProtocol = []

      if(this.protocol.length > 30000){
        this.errorProtocol.push('Protokoll zu lang!')
      }else if(this.protocol == this.project.Protocol){
        return
      }else{
        this.updateProtocol()
      }
    }
  },
  async mounted() {
    const PROJECT_ID = this.$store.state.route.params.projectId
    const USER_ID = this.$store.state.user.ID
    this.project = (await ProjectsService.showProject(PROJECT_ID, USER_ID)).data[0]

    this.formatted(this.project.Relations)
```



```
this.formatted(this.project.Candidats)

this.title = this.project.Title
this.description = this.project.Description
this.protocol = this.project.Protocol
this.statusItems.map(item => {
  if(item.ID == this.project.Status){
    this.status = item.Status
  }
})
}
}
</script>

<style scoped>
.select-box{
  width: 250px !important;
  float: right;
}

.project-card{
  border-top: 1px solid #2196F3;
}

.Aktiv {
  background-color: #4CAF50;
}

.Neu {
  background-color: #2196F3;
}

.Beendet {
  background-color: #F44336;
}

.wordwrap {
  white-space: pre-wrap;
  white-space: -moz-pre-wrap;
  white-space: -pre-wrap;
  white-space: -o-pre-wrap;
  word-wrap: break-word;
}
</style>
```


App.vue (Frontend)

```
<!--
* Author:      Danyyil Luntovsky
* File name:   App.vue
* Version:     1.0
-->

<template>
  <v-app class="grey lighten-4">
    <Navbar />

    <v-content class="mx-5 mb-5">
      <router-view></router-view>
    </v-content>

  </v-app>
</template>

<script>
import Navbar from '@components/Navbar.vue'

export default {
  name: 'App',
  components: {
    Navbar
  }
}
</script>

<style>
.danger-alert{
  color: firebrick;
}
</style>
```



Main.js (Frontend)

```
/*
 * Author:      Danyyil Luntovsky
 * File name:   main.js
 * Version:     1.0
 */

import Vue from 'vue'
import './plugins/vuetify'
import App from './App.vue'
import { router } from './router'
import { sync } from 'vuex-router-sync'
import store from '@store/store'

Vue.config.productionTip = false

sync(store, router)

new Vue({
  router,
  store,
  render: h => h(App)
}).$mount('#app')
```

Router.js (Frontend)

```
/*
 * Author:      Danyyil Luntovsky
 * File name:   router.js
 * Version:     1.0
 */

import Vue from 'vue'
import Router from 'vue-router'
import Dashboard from './views/Dashboard.vue'
import MyProjects from './views/MyProjects.vue'
import Login from './views/Login.vue'
import ViewProject from './views/ViewProject.vue'
import store from '@store/store'

Vue.use(Router)

export const router = new Router({
  mode: 'history',
  base: process.env.BASE_URL,
  props: true,
  routes: [
    {
      path: '/',
      name: 'dashboard',
      component: Dashboard,
    },
    {
      path: '/myProjects',
      name: 'myProjects',
      component: MyProjects,
    },
    {
      path: '/login',
      name: 'login',
      component: Login,
    },
    {
      path: '/project/:projectId',
      name: 'project',
      component: ViewProject,
    },
  ],
})
```



```
router.beforeEach((to, from, next) => {  
  let publicPage = ['/login']  
  let authRequired = !publicPage.includes(to.path);  
  
  let loggedIn = store.state.token  
  
  if (authRequired && !loggedIn) {  
    return next('/login')  
  }  
  
  next();  
  
})
```

Package.json (Frontend)

```
/*
 * Author:      Danyyil Luntovsky
 * File name:   package.json
 * Version:     1.0
 */

{
  "name": "client",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "serve": "vue-cli-service serve",
    "build": "vue-cli-service build",
    "lint": "vue-cli-service lint"
  },
  "dependencies": {
    "axios": "^0.18.0",
    "date-fns": "^1.30.1",
    "material-design-icons": "^3.0.1",
    "vue": "^2.5.22",
    "vue-router": "^3.0.1",
    "vuetify": "^1.3.0",
    "vuex": "^3.1.0",
    "vuex-persistedstate": "^2.5.4",
    "vuex-router-sync": "^5.0.0"
  },
  "devDependencies": {
    "@vue/cli-plugin-babel": "^3.4.0",
    "@vue/cli-plugin-eslint": "^3.4.0",
    "@vue/cli-service": "^3.4.0",
    "babel-eslint": "^10.0.1",
    "eslint": "^5.8.0",
    "eslint-plugin-vue": "^5.0.0",
    "stylus": "^0.54.5",
    "stylus-loader": "^3.0.1",
    "vue-cli-plugin-vuetify": "^0.4.6",
    "vue-template-compiler": "^2.5.21",
    "vuetify-loader": "^1.0.5"
  },
  "eslintConfig": {
    "root": true,
    "env": {
      "node": true
    }
  },
}
```



```
"extends": [  
  "plugin:vue/essential",  
  "eslint:recommended"  
],  
"rules": {},  
"parserOptions": {  
  "parser": "babel-eslint"  
}  
},  
"postcss": {  
  "plugins": {  
    "autoprefixer": {}  
  }  
},  
"browserslist": [  
  "> 1%",  
  "last 2 versions",  
  "not ie <= 8"  
]  
}
```



TECHNISCHE
FACHSCHULE
BERN

Leistungssportfreundlicher
Lehrbetrieb





TECHNISCHE
FACHSCHULE
BERN

Leistungssportfreundlicher
Lehrbetrieb

