

# **Webapplikation zur Prozessdigitalisierung**

## **Bachelorarbeit**

Studiengang Informatik

OST – Ostschweizer Fachhochschule

Campus Rapperswil-Jona

Herbstsemester 2021

Autor:	Etienne Baumgartner
Betreuer:	Prof. Daniel Politze
Projektpartner:	Innovation Network, Zürich
Experte:	Ramon Schildknecht
Gegenleser:	Farhad D. Mehta

---

## Abstract

---

<b>Problem</b>	Unternehmen und Organisationen beschäftigen sich zunehmend mit der Digitalisierung und der Optimierung der internen sowie externen Prozesse. Die Anwendung von automatisierten Abläufen und die Aufbereitungen von entstandenen Daten erlaubt eine schnellere und genauere Abwicklung und Validierung der essenziellen Prozesse eines Unternehmens. Daher ist es naheliegend, dass sich Organisationen und Unternehmen um eine Automatisierung von gewissen Abläufen bemühen.
<b>Ziel</b>	Im Umfang dieser Arbeit soll der Anfrageprozess zur Projektförderung analysiert, geplant und in einer Anwendung umgesetzt werden. Dabei ist vorgesehen den bisherigen Ablauf anhand einer Webseitenergänzung abzubilden.
<b>Methode/ Vorgehen</b>	Anhand der Analyse der Abläufe und Handlungen der involvierten Parteien wird Einsicht in den zugrundeliegenden Prozess gewonnen. Aus den gesammelten Erkenntnissen geht ein Anwendungsentwurf hervor, welcher die bisherigen Prozesse ablösen bzw. optimieren soll. Abschliessend wird die Anwendung auf Basis des Entwurfs konstruiert.
<b>Wesentliche Ergebnisse</b>	Aus dieser Arbeit geht eine Konstellation aus Anwendungen hervor, welche den bestehenden Onlineauftritt erweitern und den bisherigen Anfrageprozess digitalisieren und optimieren.

## Inhalt

<b>Abstract</b> .....	<b>2</b>
<b>1 Verzeichnisse</b> .....	<b>5</b>
1.1 Abbildungen .....	5
1.2 Tabellen .....	6
1.3 Formeln.....	6
<b>2 Management Summary</b> .....	<b>7</b>
<b>3 Einleitung</b> .....	<b>8</b>
<b>4 Inception Phase</b> .....	<b>10</b>
4.1 Zusammenfassung und Definitionen .....	10
4.2 Domainanalyse .....	10
4.2.1 Domain Objekte.....	10
4.2.2 Die Domain Model.....	11
4.3 Prozessanalyse .....	12
4.3.1 Zielsetzung .....	12
4.3.2 Bedingungen .....	12
4.3.3 Einschränkungen.....	12
4.3.4 Pflichten .....	13
4.3.5 Prozess Flow.....	14
4.4 Anforderungsanalyse .....	15
4.4.1 Business Anforderungen .....	15
4.4.2 Stakeholder Anforderungen.....	15
4.4.3 Funktionale Anforderungen .....	16
4.4.4 Nicht-Funktionale Anforderungen .....	22
4.5 Architekturanalyse.....	23
4.5.1 Server.....	23
4.5.2 Datenbanken.....	23
4.5.3 Archivierung .....	24
<b>5 Elaboration Phase</b> .....	<b>25</b>
5.1 Zusammenfassung und Definitionen .....	25
5.2 Grobkonzeption .....	25
5.2.1 Technologie und Frameworks.....	25
5.2.2 Datenbanken .....	26
5.2.3 Deployment .....	26
5.3 Feinkonzeption .....	27
5.3.1 Nginx Reverse Proxy.....	28
5.3.2 Node Anwendung .....	28
5.3.3 MySQL .....	33
5.4 Testing .....	34
5.4.1 Unit Test.....	34
5.4.2 Integration Test .....	35
5.4.3 System/ End-to-End Test.....	35
5.4.4 User Acceptance Test .....	36
5.4.5 Tools .....	36

---

5.5	Entwicklungsumgebung.....	36
5.6	Production Environment .....	37
<b>6</b>	<b>Construction Phase .....</b>	<b>38</b>
6.1	Nginx.....	38
6.2	Node Anwendung.....	39
6.2.1	Allgemeiner Aufbau .....	40
6.2.2	Umsetzung und genauere Erklärungen .....	47
6.3	MySQL.....	53
6.3.1	Database.....	53
6.3.2	Auth.....	58
6.4	Verifikation .....	59
6.4.1	Unit / Integration .....	59
6.4.2	End-to-End .....	63
6.4.3	User Acceptance .....	63
<b>7</b>	<b>Transition Phase .....</b>	<b>64</b>
7.1	Docker und Testumgebung .....	64
7.2	Liveumgebung.....	65
7.3	Zusammenfassung.....	66
<b>8</b>	<b>Fazit.....</b>	<b>67</b>
<b>9</b>	<b>Literatur und Quellenverzeichnis .....</b>	<b>69</b>
<b>10</b>	<b>Erklärung zur Urheberschaft .....</b>	<b>70</b>
<b>Anhang</b>	<b>.....</b>	<b>71</b>
I	Auftrag .....	71
II	Projektplan .....	71
III	Zeitmanagement Analyse.....	76
IV	Gitlab Time Tracking Skript – gtt.py.....	84
V	MySQL Daten Skripte.....	86
VI	Prototyp Views .....	89
VII	MVP Views.....	93
VIII	UAT Skript.....	99
IX	UAT Rückmeldungen .....	100
X	Danksagung .....	100

# 1 Verzeichnisse

---

## 1.1 Abbildungen

---

Abb. 1: Domain Model .....	11
Abb. 2: Prozess Flow Diagramm.....	14
Abb. 3: Use Case Diagramm .....	18
Abb. 4: Infrastruktur Diagramm.....	27
Abb. 5: Schichten Diagramm .....	31
Abb. 6: Datenschema Entwurf .....	33
Abb. 7: Auth Datenschema Entwurf .....	33
Abb. 8: Nginx Konfiguration .....	38
Abb. 9: Schichten Diafgramm zu Ordnerstruktur.....	40
Abb. 10: package.json .....	41
Abb. 11: Node Routes .....	42
Abb. 12: Node Kontroller.....	43
Abb. 13: Node Service.....	44
Abb. 14: Node Views .....	45
Abb. 15: Node Helpers.....	45
Abb. 16: responseObject.....	47
Abb. 17: responseObject.error .....	47
Abb. 18: responseObject.info .....	48
Abb. 19: responseObject.bewerbungData.....	49
Abb. 20: Custom Error Klasse.....	50
Abb. 21: Verbindungs Pool .....	51
Abb. 22: Abfrageaufbau Konzept.....	52
Abb. 23: Database Schema .....	53
Abb. 24: init_status Trigger .....	54
Abb. 25: setup_bemerkung Trigger.....	55
Abb. 26: compute_status_update Trigger .....	56
Abb. 27: Auth Schema.....	58
Abb. 28: Jest Mock Objekte .....	61
Abb. 29: Jest Mock Funktionsaufruf.....	61
Abb. 30: Jest Datenüberprüfung .....	62
Abb. 31: Jest Code Coverage Übersicht .....	62
Abb. 32: Zeitstrahl.....	73
Abb. 33: Milestone Ansicht .....	76
Abb. 34: gtt.py Code .....	85
Abb. 35: Prototyp Bewerbung Übersicht .....	90
Abb. 36: Prototyp Anfrage Ansicht .....	90
Abb. 37: Prototyp neue Anfrage Auswahl .....	91
Abb. 38: Prototyp neue Anfrage.....	91
Abb. 39: Prototyp fortführende Anfrage.....	92
Abb. 40: Prototyp Administrations Ansicht .....	92
Abb. 41: Prototyp Prüfer Ansicht.....	93
Abb. 42: MVP Grundlayout .....	93

---

Abb. 43: MVP Bewerbung Übersicht (keine Bewerbungen) .....	94
Abb. 44: MVP Bewerbung Übersicht (Admin Ansicht) .....	94
Abb. 45: MVP neue Anfrage Auswahl .....	95
Abb. 46: MVP neue Anfrage .....	95
Abb. 47: MVP fortführende Anfrage .....	96
Abb. 48: MVP Anfrage Einsicht (Anfrager) .....	96
Abb. 49: MVP Anfrage Einsicht (Admin) .....	97
Abb. 50: MVP Anfrage Einsicht (Prüfer) .....	97
Abb. 51: MVP Login .....	98
Abb. 52: MVP Error Seite .....	98
Abb. 53: UAT Skript .....	99
Abb. 54: Rückmeldung Martin Stöckli .....	100

## 1.2 Tabellen

---

Tabelle 1: Technologien und Frameworks .....	26
Tabelle 2: Unit Test Beschreibung .....	34
Tabelle 3: Integration Test Beschreibung .....	35
Tabelle 4: End-to-End Test Beschreibung .....	35
Tabelle 5: Node Technologie Stack .....	42
Tabelle 6: Node Abhängigkeiten .....	42
Tabelle 7: Node Styling .....	42
Tabelle 8: Info Darstellungs Typen .....	48
Tabelle 9: Anfrage Daten .....	54
Tabelle 10: Bemerkung Daten .....	55
Tabelle 11: accounts Tabelle .....	58
Tabelle 12: Integrationstestbeschreibung .....	60
Tabelle 13: End-to-End Test Beschreibung .....	63
Tabelle 14: Konfiguration Zusammenfassung .....	66
Tabelle 15: Phasenaufteilung .....	73
Tabelle 16: Stündliche Phasenaufteilung .....	74
Tabelle 17: Labelbezeichnungen .....	75

## 1.3 Formeln

---

Formel 1: Stundenberechnung .....	74
-----------------------------------	----

## 2 Management Summary

---

<b>Kontext</b>	Der Anfrageprozess einer Projektförderung durch das Innovation Network verlangt im momentanen Zustand zu viel Eigenaufwand der involvierten Parteien. Das Einreichen einer Anfrage, der Entscheidungsprozess sowie die administrativen Abläufe werden über E-Mail koordiniert. Der Prozess ist für die Beteiligten nicht transparent genug und es muss mit unnötigem Mehraufwand bei der Absprache gerechnet werden.
<b>Begründung zur Handlung</b>	Die Entscheidungsträger und die Administration agieren unabhängig voneinander und haben keinerlei Einsicht über die individuellen Entscheidungen. Das Treffen einer Entscheidung muss durch eine zentrale Figur koordiniert und anschliessend umgesetzt werden. Dieser Mehraufwand kann durch die Bereitstellung einer Anwendung unterbunden werden.
<b>Ziele</b>	Das Ziel der Anwendung ist eine Plattform zu schaffen, die die individuellen Aktionen der verschiedenen Parteien im Anfrageprozess transparent und optimiert abbilden. Die Plattform erlaubt das Einsehen des Anfragebeschluss durch einen Antragsteller, die zentralisierte Entscheidungstreffung durch die Anfrageprüfer, sowie die Abwicklung der Anfrage durch die Administration. Die Anwendung soll den Mehraufwand beseitigen, der durch die unkoordinierte Kommunikation via E-Mail entsteht. Gleichzeitig soll dabei der Onlineauftritt des Innovation Networks an Professionalität gewinnen, um damit den Benutzern, Besuchern und Mitglieder des Netzwerks eine transparente und informationsgewinnende Erfahrung zu bieten.
<b>Optionen</b>	<p>In der ersten Option wird die Webseite des Innovation Networks durch eine Anfrageplattform erweitert. Der bisherige Anfrageprozess über E-Mail wird durch eine prozessorientierte Anwendung ersetzt.</p> <p>Durch die zweite Option wird der Anfrageprozess und der Onlineauftritt des Innovation Networks so belassen wie er ist.</p> <p>Durch die Wahl der ersten Option, kann das Innovation Network die Wahrnehmung ihres Onlineauftrittes verbessern. Die zusätzliche Transparenz, Einsicht in laufende Projekte und vereinfachte Anfrage generieren einen Mehrwert für die Besucher der Innovation Webseite und erhöhen somit die positive Wahrnehmung der Organisation. Ausserdem wird der Aufwand für administrative Tätigkeiten vermindert, da die erhebliche zeitliche Koordination der Entscheidungsfindung entfällt.</p>
<b>Empfehlung</b>	Die auf den beschriebenen Ausführungen und Beurteilungen gestützte Empfehlung ist: Eine Anwendung mit einem optimierten und teilsautomatisierten Anfrageprozess soll den jetzigen Bewerbungsablauf als Erweiterung der Innovation Network Webseite ablösen.

### 3 Einleitung

---

<b>Kontext</b>	<p>Die Kommunikation zwischen zwei oder mehreren Parteien in der Welt des World Wide Webs nimmt viele Gestalten an und kann unter Umständen zeitintensiv ausfallen. Besonders, wenn das Übertragungsmittel «E-Mail» heisst und das Ziel der Kommunikation eine Entscheidungsfindung zwischen mehreren Parteien ist. Ein weiterer, nicht zu vernachlässigender Umstand ist die Organisation und Aufbewahrung der Informationen, die zur Entscheidungstreffung herangezogen werden.</p> <p>Das Innovation Network greift auf diese Form der Kommunikation in ihrem Anfrageprozess zur Projektförderung zurück. Im Umfang dieser Arbeit soll eine pragmatische Lösung für einen teilsautomatisierten Bewerbungsprozess erarbeitet werden.</p>
<b>Problem</b>	<p>Ein Student will mit einer Idee und der Hoffnung auf eine finanzielle Förderung Kontakt zum Innovation Network aufnehmen. Die Webseite verweist auf eine E-Mail-Adresse und fordert zur Kontaktaufnahme via Text auf. Hinter der Verarbeitung und dem Entscheidungsprozess einer Anfrage stehen mindestens sechs Personen, welche das Anliegen des Studenten prüfen und durch das Zusammenführen der individuellen Entscheidungen bewilligen oder ablehnen. Die Abwicklung und Kommunikation fand bisher über E-Mail statt und ist im besten Fall suboptimal.</p>
<b>Ziel</b>	<p>Das Ziel dieser Arbeit ist den Bewerbungsprozess des Innovation Networks zu digitalisieren. Der Entscheidungsprozess soll in einer Webanwendung abgebildet und die E-Mail-Kommunikation zwischen den verantwortlichen Personen des Innovation Network soll durch die bereitgestellte Funktionalität der Anwendung abgelöst werden. Die notwendigen Bewerbungsdaten werden direkt verwaltet und erlaubt die Prüfung sowie die Einsicht dieser Informationen durch alle beteiligten Parteien direkt auf der Webseite des Innovation Networks.</p>
<b>Methode / Vorgehen</b>	<p>Die oben beschriebene, grundlegende Idee des Projekts beschreibt eine typische Full-Stack Anwendung. Daten werden via Formulare Bewerbungsdaten in die Anwendung eingespeist und die Anwendungslogik soll sich anhand der Eingaben der Prüfer um die Entscheidungstreffung kümmern.</p> <p>Die Anwendung setzt sich ausschliesslich aus den Vorlagen des Bewerbungsprozess und den Aussagen des Innovation Network Teams zusammen. Die Analyse des Ablaufs und der Aufgaben der involvierten Personen, sowie die Untersuchung der bestehenden Architektur bilden die Grundlage dieser Arbeit. Basierend auf diesen Informationen wird das Frontend zusammengestellt, die Abläufe im Backend definiert und die Daten abgelegt.</p> <p>Die Methodik, der das Projekt folgt, kann im Rational Unified Process zusammengefasst werden. Dieser erlaubt die agile Entwicklung und ist ebenfalls eine bewährte Art, Projekte auf der «grünen Wiese» anzugehen.</p> <p>Die Inception Phase bildet dabei den Einstiegspunkt. Es werden allgemeine Analysen bezüglich Domaine, Prozessaufbau, Anforderungen und Architektur durchgeführt. Mit dem erarbeiteten Wissen wird in der Elaboration Phase ein Entwurf der Anwendung ausgearbeitet und weitere projektspezifische Qualitätskontrollen definiert. In der Construction Phase werden die Erkenntnisse aus den vorherigen Phasen umgesetzt und anschliessen in der Transition Phase so weit fertiggestellt, dass eine Aufschaltung mit geringfügigem Zusatzaufwand möglich ist.</p>

## Überblick über die Arbeit

Die Arbeit teilt sich in folgende Abschnitte ein:

- Inception Phase bzw. Einarbeitung und Analyse
- Elaboration Phase bzw. vertiefte Analyse und Ausarbeitung des Konzepts
- Construction Phase bzw. Umsetzung des Konzepts
- Transition Phase bzw. Zurechtlegung der Artefakte und Abschluss des Codes
- Fazit

## 4 Inception Phase

---

### 4.1 Zusammenfassung und Definitionen

---

<b>Domainanalyse</b>	Untersucht die allgemein gültigen Voraussetzungen sowie den Umfang der Anwendung. Beinhaltet das Domain Model.
<b>Prozessanalyse</b>	Untersucht den Prozessablauf und die resultierenden Endzustände einer Bewerbung sowie die möglichen Handlungen der individuellen Parteien.
<b>Anforderungsanalyse</b>	Untersucht die Abhandlung einer Bewerbung aus Sicht der einzelnen Parteien, sowie deren Motive und möglichen Aktionen. Soll einen klaren Überblick über die zu entwickelnde Anwendung liefern. Wird in Zusammenarbeit mit den Akteuren/innen sowie den Bewerbungsvorlagen definiert. Beinhaltet neben einer genauen Beschreibung aller Beteiligten und deren Handlungen auch eine visuelle Darstellung in Form eines Use Case Diagramms.
<b>Qualitätsanalyse</b>	Untersucht die notwendigen Eigenschaften, die die Anwendung auf technischer Ebene aufzuweisen hat.

### 4.2 Domainanalyse

---

#### 4.2.1 Domain Objekte

---

<b>Anfrager</b>	Beschreibt den Initiator des gesamten Bewerbungsprozesses. Der Anfrager stellt eine Anfrage via der Innovation Network Webseite.  1:*      Beziehung zum zentralen <i>Anfrage</i> Objekt  Jeder Anfrager kann eine oder mehrere Anfragen einreichen.
<b>Admin</b>	Beschreibt den Administrator bzw. den Koordinator einer Anfrage.  1:*      Beziehung zum zentralen <i>Anfrage</i> Objekt
<b>Prüfer</b>	Beschreibt den Reakteur des Bewerbungsprozesses. Die Prüfer behandeln eine Anfrage und beeinflussen deren Status.  5:*      Beziehung zum zentralen <i>Anfrage</i> Objekt 5:1      Beziehung zum <i>Status</i> Objekt

**Anfrage** Beschreibt das zentrale Objekt.

- \*:1 Beziehung zum *Anfrager* Objekt
- \*:5 Beziehung zum *Prüfer* Objekt
- 1:(0..1) Beziehung zum *fortführende Anfrage* Objekt
- 1:1 Beziehung zum *Status* Objekt

**Fortführende Anfrage** Beschreibt eine Variation des zentralen Objekts.

- (0..1):1 Beziehung zum zentralen *Anfrage* Objekt

**Status** Beschreibt den Zustand des zentralen Objekts.

- 1:1 Beziehung zum zentralen *Anfrage* Objekt
- 1:5 Beziehung zum *Prüfer* Objekt

#### 4.2.2 Die Domain Model

**Diagramm**

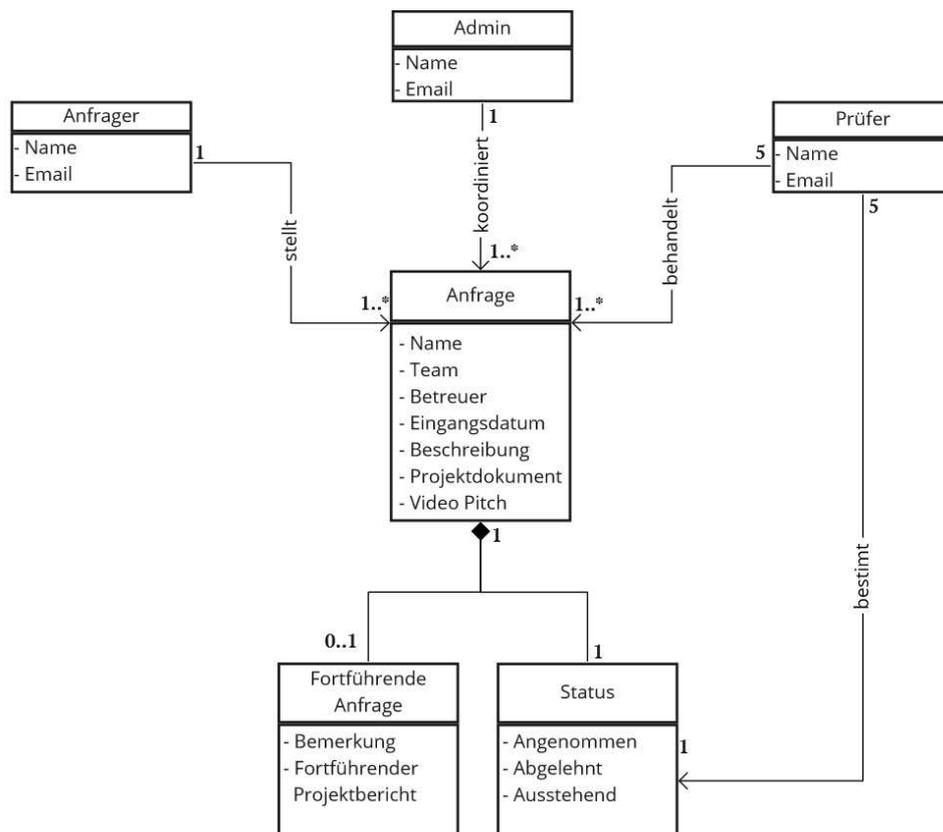


Abb. 1: Domain Model

---

## 4.3 Prozessanalyse

---

### 4.3.1 Zielsetzung

---

**Ziel des Innovation Networks** Ziel des Innovation Networks ist die finanzielle Unterstützung/Förderung beim Bau eines Prototyps.

**Ziel der Arbeit** Ziel der Arbeit ist es, den Anfrage- und Entscheidungsprozess von Projektförderungen des Innovation Networks in Form einer Webapplikation abzubilden und somit den Aufwand für alle involvierten Parteien zu vermindern.

### 4.3.2 Bedingungen

---

**Einführung** Der Anfrageprozess unterliegt gewissen Bedingungen, welche direkt vom Innovation Network vorgegeben sind. Werden diese Bedingungen vor der Bewerbung bzw. während der Ausführung des Projekts nicht erfüllt, wird die Förderung eines Projekts nicht stattgegeben bzw. nicht fortgeführt.

**Allgemeine Bedingungen** Es gelten die allgemeinen Bedingungen:

- 1) BSc- und MSc-Studierende einer Schweizer Hochschule
- 2) Produktbezogene Ideen mit einer Hardwarekomponente (keine reinen Software-, App- oder Serviceprojekte) in einem frühen Entwicklungsstadium
- 3) Projekte mit kurz- oder mittelfristiger Marktperspektive (keine langfristigen Forschungsprojekte)
- 4) Förderer in Form eines Projektpaten (akademischer Mitarbeiter) an ihrer Hochschule
- 5) Mindestens ein Teammitglied muss BSc- und MSc-Studierende an einer schweizerischen Hochschule oder ETH sein.

**Bedingungen für Anfrager der ETH** Es gelten folgende Bedingungen spezifisch für Anfragen von Studenten der ETH Zürich:

- 1) An der ETH Zürich wird die Projektförderung mit dem Student Project House koordiniert
- 2) Bewerbungen einmal pro Semester

### 4.3.3 Einschränkungen

---

**Einführung** Der Anfrageprozess unterliegt gewissen Beschränkungen, welche direkt vom Innovation Network auferlegt werden. Diese Beschränkungen beziehen sich auf die Ausschüttung von Fördergeldern sowie auf den Nutzungsbereich eines Projekts.

**Allgemeine Beschränkungen** Es gelten die allgemeinen Beschränkungen:

- 1) Bei Annahme wird ein erster Betrag von 3000 CHF ausbezahlt.
- 2) Nach Ablauf von 6 Monaten kann eine weitere Fördersumme von bis zu 2'000 CHF für nachgereicht werden.

---

**Beschränkungen des Nutzungsbereichs** Es gelten folgende Beschränkungen bezüglich des Nutzungs-/Anwendungsbereichs eines Projekts:

- 1) Materialien und Komponenten, die während der Produktentwicklung verwendet werden.
- 2) Material- und Komponenten-Kosten für Prototypen.
- 3) Benötigtes Material und Umgebung für das Testen von Prototypen.

#### **4.3.4 Pflichten**

---

**Einführung** Nach ausgeschüttetem Förderbetrag unterliegt ein Anfrager gewissen Pflichten, welche direkt vom Innovation Network bestimmt werden. Diese Pflichten beziehen sich auf die Vorstellung des erstellten Prototyps.

**Allgemeine Pflichten** Es gelten die allgemeinen Pflichten:

- 1) Spätestens ein Jahr nach Projektbeginn muss eine Präsentation über die bisherigen Arbeitsergebnisse sowie einen «One-Pager» über Ihren Projektfortschritt abgeliefert werden.
- 2) Präsentation eines Prototypen und neuer Erkenntnisse vor den Mitgliedern des Innovation Networks

**Pflichten für Anfrager der ETH** Es gelten folgende Pflichten spezifisch für Anfragen von Studenten der ETH Zürich:

- 1) Die Förderung durch das Innovation Network muss in der Projektbeschreibung auf der SPH Webseite sowie auf anderen Internetauftritten des Projekts ersichtlich sein.

### 4.3.5 Prozess Flow

#### Übersicht

Die Bedingungen, Beschränkungen und Pflichten sind im untenstehenden Prozess Diagramm abgebildet.

#### Diagramm

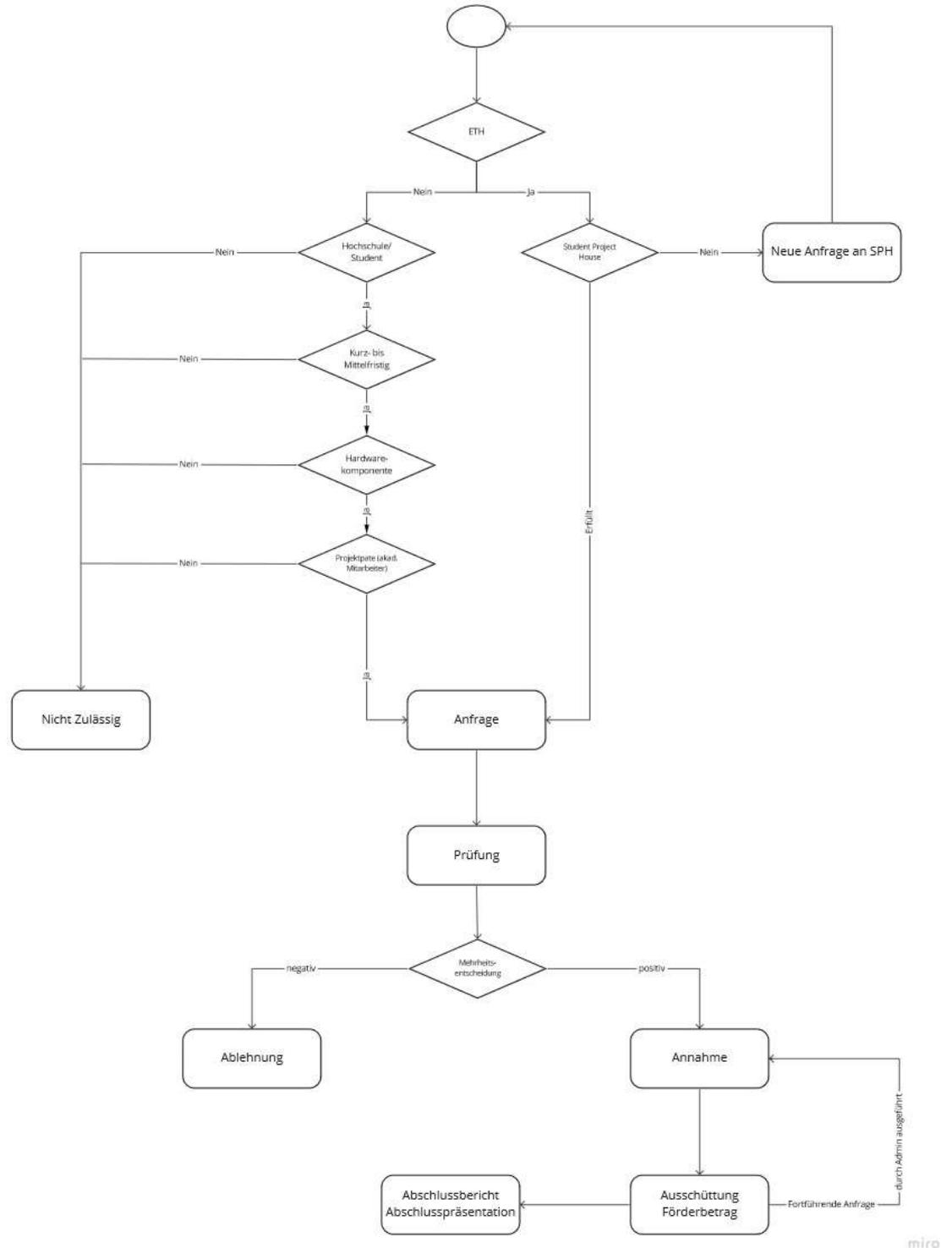


Abb. 2: Prozess Flow Diagramm

---

## 4.4 Anforderungsanalyse

---

### 4.4.1 Business Anforderungen

---

<b>Einführung</b>	Das Innovation Network stellt eine Plattform zur Beantragung von Fördergeldern zur Verfügung. Dabei kann eine Förderanfrage gestellt werden, welche bei Annahme einen ersten Förderbetrag von 3000 CHF und einen nachfolgenden Betrag von 2000 CHF ausschütten kann. Insgesamt beträgt die Fördersumme 5000 CHF.
<b>Ist Zustand</b>	Die bisherige Kommunikation und Abhandlung einer Anfrage wurde mittels E-Mail gelöst.
<b>Soll Zustand</b>	Die zukünftige Kommunikation und Abhandlung einer Anfrage soll direkt über die Website vollzogen werden.

### 4.4.2 Stakeholder Anforderungen

---

<b>Einführung</b>	Für das Projekt wurden vier primäre Stakeholder sowie ein sekundärer Stakeholder identifiziert. Alle diese Akteure vertreten ein Interesse an der Förderung eines Projekts.
<b>Innovation Network/Prüfer</b>	<p>Das Innovation Network bzw. die Prüfer sind ein Synonym für die entscheidungstragenden Vorstandsmitglieder. Sie fällen die Entscheidung bezüglich der Annahme oder der Ablehnung einer Anfrage.</p> <p>Die übergeordneten Interessen des Innovation Network/Prüfer-Stakeholders sind:</p> <ul style="list-style-type: none"><li>➤ Simple Abhandlung der Anfragen</li><li>➤ Minimale Ausgaben für Server und Datenablage</li><li>➤ Minimaler Aufwand für Änderung an bestehendem System</li></ul>
<b>Admin</b>	<p>Der Admin erfüllt die administrative Rolle im Innovation Network. Er/Sie behält die Übersicht über laufende und vergangene Projektanfragen und bearbeitet administrative Aufgaben rund um den Anfrageprozess.</p> <p>Die übergeordneten Interessen des Admin-Stakeholders sind:</p> <ul style="list-style-type: none"><li>➤ Übersicht über alle Anfragen</li><li>➤ Kontrolle über alle Anfragen</li><li>➤ Administrative Rechte und Pflichten (Ausschüttung der Förderung, organisieren der Abgabedaten und Präsentation)</li></ul>

---

<b>Anfrager</b>	<p>Der Anfrager ist der Initiator des Anfrageprozesses. Er/Sie richtet sich mit einer konkreten Idee an das Innovation Network, um eine Projektförderung anzufragen.</p> <p>Die übergeordneten Interessen des Anfrager-Stakeholders sind:</p> <ul style="list-style-type: none"><li>➤ Erhalten des Förderbetrags</li><li>➤ Transparenter Anfrageablauf und Anfrageeinsicht</li></ul>
<b>Mitglieder</b>	<p>Die Mitglieder des Innovation Networks sind passive Akteure im gesamten Anfrageprozess. Sie bestehen aus einer Gruppe von Interessenten und haben an sich keine Rechte im Förderungsprozess.</p> <p>Die übergeordneten Interessen des Mitglieder-Stakeholders sind:</p> <ul style="list-style-type: none"><li>➤ Einsicht in laufende und abgeschlossene Anfragen</li><li>➤ Zugang zu Präsentationen im 1-jährigen Networking Anlass</li></ul>
<b>Student Project House (Sekundär)</b>	<p>Das Student Project House der ETH Zürich ist internes Förderprogramm der Hochschule. Ihre Interessen liegen nicht direkt im Prozess des Innovation Network Förderung, jedoch bilden sie die Grundlage für jegliche Anfragen von Studenten der ETH Zürich.</p> <p>Die übergeordneten Interessen des Student-Project-House-Stakeholders sind:</p> <ul style="list-style-type: none"><li>➤ Grundlage einer Anfrage von Studenten der ETH Zürich</li></ul>

#### 4.4.3 Funktionale Anforderungen

---

<b>Einführung</b>	<p>Die Funktionalen Anforderungen bilden die notwendigen Funktionalitäten der Anwendung ab. Sie setzen sich aus den beteiligten Parteien (Personas) und deren Handlungen (Use Cases) im Prozess zusammen.</p>
-------------------	---

##### 4.4.3.1 Personas

<b>Einführung</b>	<p>Die Personas beschreiben die aktiven oder passiven Akteure im gesamten Anfrageprozess. Eine Persona initiiert oder reagiert auf eine Aktion im Anfrageprozess. Nachfolgend werden die Akteure und ihre Grundeigenschaften sowie ihre Motivation aufgelistet.</p>
-------------------	---

<b>Anfrager</b>	<p>Die Anfrager-Persona ist folgendermassen beschrieben:</p>
-----------------	--

- Person mit Projekt Idee
- Will einen Förderbetrag beanspruchen
- Hält ein Konzept bereit
- Hat eine Betreuungsperson im akademischen Umfeld

---

<b>Prüfer</b>	Die Prüfer-Persona ist folgendermassen beschrieben: <ul style="list-style-type: none"><li>➤ Eines von fünf Vorstandmitglieder des Innovation Networks</li><li>➤ Stellt Förderbetrag bereit</li><li>➤ Will kurz- bis mittelfristige Projekte fördern</li></ul>
<b>Admin</b>	Die Admin-Persona ist folgendermassen beschrieben: <ul style="list-style-type: none"><li>➤ Koordiniert die Anfragen</li><li>➤ Ist im Ausschüttungsprozess involviert</li><li>➤ Soll die Übersicht haben</li><li>➤ Hat Bearbeitungsrechte auf den Anfragen</li></ul>
<b>Mitglieder</b>	Die Mitglieder-Persona ist folgendermassen beschrieben: <ul style="list-style-type: none"><li>➤ Mitglied des Innovation Networks</li><li>➤ Will Einsicht in laufende und abgeschlossene Förderprojekte</li></ul>
<b>4.4.3.2 User Stories</b>	
<b>Einführung</b>	Die User Stories beschreiben im grobem Umfang den Ablauf des Anfrageprozesses mit Einbezug der verschiedenen Personas.
<b>Positive Erst-Anfrage</b>	<ol style="list-style-type: none"><li>1) Eine erste Anfrage wird vom <i>Anfrager</i> eingereicht</li><li>2) Die Anfrage wird von den fünf <i>Prüfern</i> behandelt</li><li>3) Die <i>Prüfer</i> entscheiden individuell über die Zulassung</li><li>4) Die Anfrage erhält von allen fünf <i>Prüfern</i> eine Zulassung</li><li>5) Der <i>Anfrager</i> wird über die Zulassung (positiv) informiert</li><li>6) Dem <i>Anfrager</i> werden 3000 CHF überwiesen</li><li>7) (Nach spätestens einem Jahr wird eine Projektvorstellung durchgeführt)</li></ol>
<b>Fortführende Anfrage</b>	<ol style="list-style-type: none"><li>1) Eine fortführende Anfrage mit angehängtem 1-seitigen Fortschrittsbericht wird vom <i>Anfrager</i> eingereicht</li><li>2) Die fortführende Anfrage wird von den fünf <i>Prüfern</i> und dem <i>Admin</i> eingesehen</li><li>3) Der <i>Admin</i> gibt die fortführenden Fördergelder frei</li><li>4) Der <i>Anfrager</i> wird über die fortführende Zulassung informiert</li><li>5) Dem <i>Anfrager</i> werden 2000 CHF überwiesen</li><li>6) (Nach spätestens einem Jahr wird eine Projektvorstellung durchgeführt)</li></ol>
<b>Negative Anfrage</b>	<ol style="list-style-type: none"><li>1) Eine Anfrage wird vom <i>Anfrager</i> eingereicht</li><li>2) Die Anfrage wird von den fünf <i>Prüfern</i> behandelt</li><li>3) Die <i>Prüfer</i> entscheiden individuell über die Zulassung</li><li>4) Die Anfrage erhält von den fünf <i>Prüfern</i> <b>keine</b> Zulassung</li><li>5) Der <i>Anfrager</i> wird über die verweigerzte Zulassung (negativ) informiert</li></ol>

**Einsicht der Anfragen**

- 1) *Prüfer/Anfrager/Mitglied/Admin* fragt den Status aller Anfrage an
- 2) Dem *Prüfer/Anfrager/Mitglied/Admin* wird der Status aller Anfrage angezeigt

**Ausschüttung einleiten durch Admin**

- 1) *System* meldet Zustimmung zu Anfrage
- 2) *Admin* nimmt Kontakt mit *Anfrager* bezüglich Bankdaten auf
- 3) *Admin* schüttet ersten Teil der Förderung aus
- 4) *Admin* trägt ausbezahlten Betrag in Übersicht ein

#### 4.4.3.3 Use Cases

**Einführung**

Die Use Cases beschreiben den detaillierten Umfang und den Ablauf des Anfrageprozesses mit Einbezug der verschiedenen Personas.

**Use Case Diagramm**

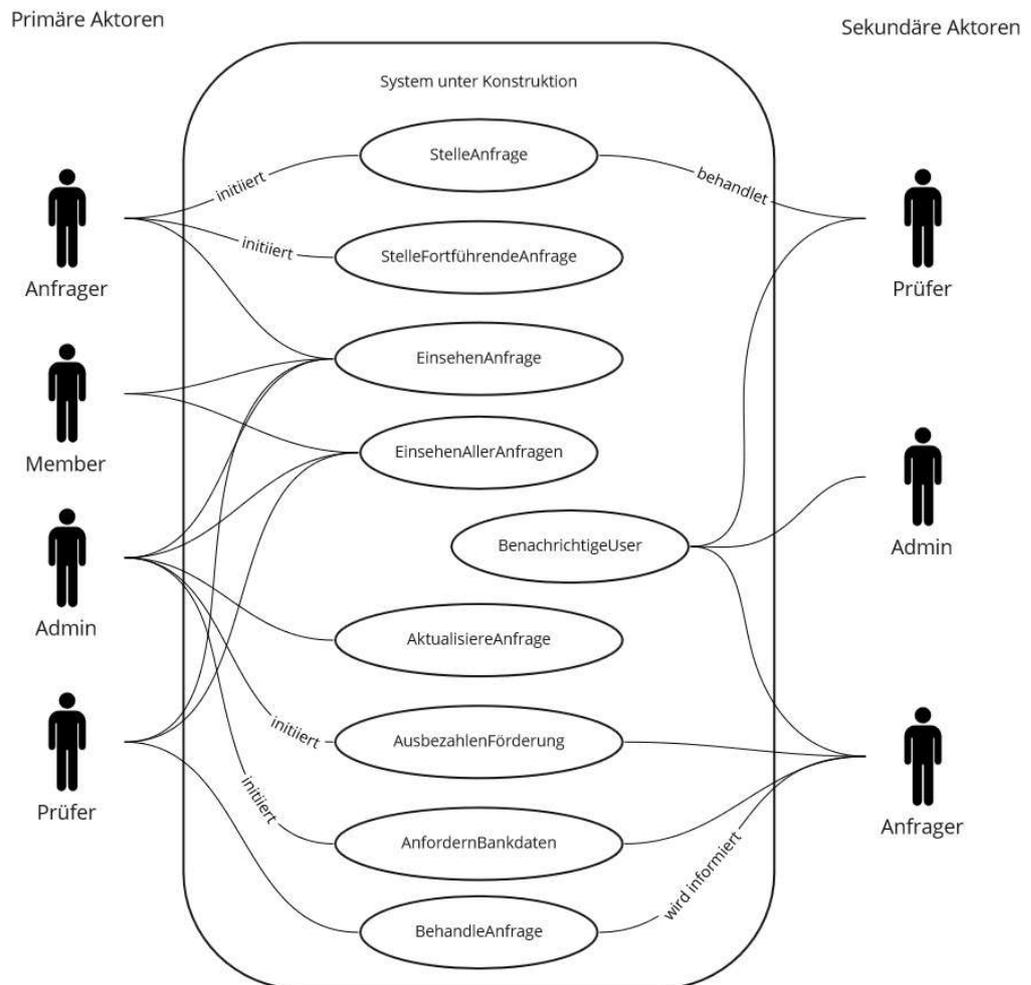


Abb. 3: Use Case Diagramm

**Stelle Anfrage**

Name	StelleAnfrage
Primary Actor	Anfrager
Stakeholders & Interessen	<p><i>Anfrager:</i> Will 3000 CHF (potenziell 5000 CHF) für ein Projekt</p> <p><i>Prüfer:</i> Will ein vielversprechendes Projekt mit 3000 CHF (potenziell 5000 CHF) unterstützen, Zulassung soll simpel abgehandelt werden</p>
Positives Szenario	<ol style="list-style-type: none"> <li>1) <i>Anfrager</i> reicht Anfrage über online Tool ein             <ol style="list-style-type: none"> <li>a. Projektvorstellung (3-5) Sätze</li> <li>b. Optionaler Video-Pitch</li> <li>c. Teamvorstellung</li> <li>d. Ideen und Vorstellungen</li> <li>e. Grobe Zeitplanung</li> <li>f. Begründung und Quantifizierung des finanziellen Betrags</li> </ol> </li> <li>2) <i>System</i> meldet dem <i>Anfrager</i> die Übermittlung seiner Anfrage</li> <li>3) <i>System</i> informiert den <i>Anfrager</i> nach der Prüfung über eine positive Zulassung</li> </ol>
Extension / Alternativer Flow	<ol style="list-style-type: none"> <li>3) <i>System</i> informiert den <i>Anfrager</i> nach der Prüfung über die Ablehnung des Antrags</li> </ol>

**Stelle fortführende Anfrage**

Name	StelleFortführendeAnfrage
Primary Actor	Anfrager
Stakeholders & Interessen	<p><i>Anfrager:</i> Will restliche 2000 CHF für ein bereits zugelassenes Projekt</p> <p><i>Prüfer:</i> Will ein vielversprechendes Projekt mit gesamthaft 5000 CHF unterstützen</p>
Positives Szenario	<ol style="list-style-type: none"> <li>1) <i>Anfrager</i> reicht eine fortführende Anfrage über online Tool ein             <ol style="list-style-type: none"> <li>a. <i>One-Pager zum Projektfortschritt</i></li> </ol> </li> <li>2) <i>System</i> meldet dem <i>Anfrager</i> die Übermittlung seiner fortführenden Anfrage</li> <li>3) <i>System</i> informiert den <i>Anfrager</i> nach der Prüfung über eine positive fortführende Zulassung</li> </ol>
Extension / Alternativer Flow	<ol style="list-style-type: none"> <li>3) <i>System</i> informiert den <i>Anfrager</i> nach der Prüfung über die Ablehnung des fortführenden Antrags</li> </ol>

<b>Behandle Anfrage</b>	Name	BehandleAnfrage
	Primary Actor	Prüfer
	Stakeholders & Interessen	<i>Anfrager:</i> Will 3000 CHF (potenziell 5000 CHF) für ein Projekt  <i>Prüfer:</i> Will ein vielversprechendes Projekt mit 3000 CHF (potenziell 5000 CHF) unterstützen
	Positives Szenario	<ol style="list-style-type: none"> <li>1) <i>System</i> meldet jedem <i>Prüfer</i> den Eingang einer Anfrage</li> <li>2) <i>Prüfer</i> behandelt die Anfrage</li> <li>3) <i>Prüfer</i> (individuell) erteilt Zulassung mit Kommentar/Bemerkung</li> <li>4) <i>System</i> wartet auf einstimmigen Entscheid</li> <li>5) <i>System</i> übermittelt Zulassung der Anfrage an <i>Anfrager</i></li> </ol>
	Extension / Alternativer Flow	<ol style="list-style-type: none"> <li>3) <i>Prüfer</i> (individuell) lehnt den Antrag ab</li> <li>4) <i>System</i> wartet auf einstimmigen Entscheid</li> <li>5) <i>System</i> übermittelt Ablehnung der Anfrage an <i>Anfrager</i></li> </ol>
<b>Einsehen aller Anfragen</b>	Name	EinsehenAllerAnfragen
	Primary Actor	Prüfer/Mitglied/Admin
	Stakeholders & Interessen	Prüfer/Mitglied/Admin: Will alle laufenden und vergangenen Anfragen abrufen
	Positives Szenario	<ol style="list-style-type: none"> <li>1) <i>Prüfer/Mitglied/Admin</i> fragt alle Anfragen an</li> <li>2) <i>System</i> liefert Liste aller Anfragen und deren Status</li> </ol>
<b>Einsehen individueller Anfrage</b>	Name	EinsehenAnfrage
	Primary Actor	Anfrager/Prüfer/Admin/Mitglied
	Stakeholders & Interessen	<i>Anfrager/Prüfer/Admin/Mitglied:</i> Will alle laufenden Anfragen abrufen und Status einsehen
	Positives Szenario	<ol style="list-style-type: none"> <li>1) <i>Anfrager/Prüfer/Admin/Member</i> fragt individuelle Anfragen an</li> <li>2) <i>System</i> liefert individuelle Anfrage und deren Status zurück</li> </ol>
	Bemerkung	Falls eine Anfrage noch nicht abgeschlossen ist, kann sie nur vom Anfrager sowie den Prüfern und dem Admin eingesehen werden.
<b>Bankdaten anfordern</b>	Name	AnfordernBankdaten
	Primary Actor	Admin
	Stakeholders & Interessen	<i>Admin:</i> Will Bankdaten des <i>Anfragers</i> <i>Anfrager:</i> Will Fördergeld
	Positives Szenario	<ol style="list-style-type: none"> <li>1) <i>Admin</i> fragt nach angenommener <i>Anfrage</i> Bankdaten des <i>Anfragers</i> via Mail an</li> <li>2) <i>Anfrager</i> liefert Bankdaten</li> <li>3) <i>Admin</i> führt Zahlungsprozess durch</li> </ol>

**Förderung ausbezahlen**

Name	AusbezahlenFörderung
Primary Actor	Admin
Stakeholders & Interessen	<i>Admin:</i> Will Fördergeld übertragen <i>Anfrager:</i> Will Fördergeld
Positives Szenario	1) <i>Admin</i> zahlt Fördergeld aus

**Anfrage aktualisieren**

Name	AktualisiereAnfrage
Primary Actor	Admin
Stakeholders & Interessen	<i>Admin:</i> Will Details einer Anfrage aktualisieren
Positives Szenario	1) <i>Admin</i> ändert getätigte Auszahlung oder Präsentationsdatum der Anfrage

**Benutzer benachrichtigen**

Name	BenachrichtigeUser									
Primary Actor	System									
Stakeholders & Interessen	<i>Admin/Prüfer/Anfrager:</i> Will Informationen über Änderungen									
Positives Szenario	1) Nach abgeschlossener Aktion benachrichtig <i>System</i> den passenden User zur Aktion									
Aktion	<table border="1"> <tr> <td>Statusänderung</td> <td>Benachrichtige</td> <td>Anfrager/Prüfer/Admin</td> </tr> <tr> <td>Eingehende Anfrage</td> <td>Benachrichtige</td> <td>Prüfer/Admin/Anfrager</td> </tr> <tr> <td>Eingehende fortführende Anfrage</td> <td>Benachrichtige</td> <td>Prüfer/Admin/Anfrager</td> </tr> </table>	Statusänderung	Benachrichtige	Anfrager/Prüfer/Admin	Eingehende Anfrage	Benachrichtige	Prüfer/Admin/Anfrager	Eingehende fortführende Anfrage	Benachrichtige	Prüfer/Admin/Anfrager
Statusänderung	Benachrichtige	Anfrager/Prüfer/Admin								
Eingehende Anfrage	Benachrichtige	Prüfer/Admin/Anfrager								
Eingehende fortführende Anfrage	Benachrichtige	Prüfer/Admin/Anfrager								

#### 4.4.4 Nicht-Funktionale Anforderungen

---

**Einführung** Die Nicht-Funktionalen Anforderungen setzen sich aus verschiedenen Überlegungen bezüglich des reibungslosen Betriebes der Anwendung zusammen.

**Rechtliche Anforderungen** Gibt es rechtliche Implikationen bezüglich der Projekte?

- Datenschutz muss eingehalten werden
- Rechtliche Implikationen müssen geklärt sein
- Liegt in der Verantwortung des Innovation Networks

**Reliability** **Wiederherstellbarkeit:**  
Die gespeicherten Daten der Anfragen und Nutzer sollen persistent gespeichert werden

**Availability** **Zugänglichkeit:**  
Bei Ausfällen muss eine alternative Kommunikation bereitgestellt sein

**Usability** **Bedienbarkeit:**  
Entwicklung für Desktop  
**Erlernbarkeit:**  
Anfrage- und Prüfungsprozess muss intuitiv gestaltet sein

**Interoperability** Die Website unterstützt die folgenden Browser und Versionen:

Chrome	96.0
Firefox	95
Edge	96
iOS Safari	13.1

Die Anwendung soll sich der bestehenden Webseite anpassen und die Funktionalität erweitern.

**Security** **Login:**  
Login, Logs nur mit ID bzw. Keine persönlichen Daten

**Anfrager:**  
Eingeschränkte Einsicht in alle Anfragen und individueller Anfrage

**Prüfer:**  
Vollumfänglicher Zugang zu allen Anfragen jedoch keine rückwirkenden Änderungen  
Kann nur seine eigenen Zustimmungen bearbeiten  
Kann alle Zustimmungen einsehen

**Admin:**  
Vollumfänglicher Zugang zu allen Anfragen

**Mitglieder:**

Eingeschränkte Einsicht in alle abgeschlossenen Anfragen

**Performance**      Laden der Daten unter normalen Load < 3s, ausser bei Übertragung von Videodaten.

## 4.5 Architekturanalyse

---

### 4.5.1 Server

---

**Hosting**            Die bisherige Anwendung ist auf einem Apache Server abgelegt. Gehostet wird der Server von Netzone (<https://www.netzone.ch/de/webhosting>)

**Weiteres**            Leider sind nur wenige Informationen bezüglich des Servers bekannt. Auf eine weitere Analyse des Ist-Zustandes wird an dieser Stelle verzichtet, da in der Elaboration Phase eine direkte Gegenüberstellung des Ist-Zustandes zum Soll-Zustand bereitgestellt wird.

### 4.5.2 Datenbanken

---

**Einführung**        Das Innovation Network muss alle übertragenen Anfragen und die damit zusammenhängenden Daten in einer Datenbank ablegen.

**Anfrage Daten**    Eine Datenbank wird für die Aufbewahrung der Anfragedaten benötigt.  
Minimal enthaltene Daten:

- Eingangsdatum
- Abschlussdatum
- Auszahlung
- Präsentation
- Status
- Bezeichnung
- Team
- Vorstellung
- Projektbericht (pdf)
- Video Pitch (mp4 o.ä.)
- Bemerkung
- Fortführender Projektbericht

**Login**                Eine Datenbank für die Authentifizierung wird benötigt.  
Enthaltene Daten:

- ID
- Name
- Rolle (Prüfer)

---

### 4.5.3 Archivierung

---

#### Einführung

Die Idee der Archivierung ist die Aufnahme bisher behandelter Anfragen in die neue Struktur. Dies erlaubt eine einfache Einsicht in die Archive des Innovation Networks.

Dieser Teil der Arbeit steht jedoch nicht im Fokus und kann nachträglich durchgeführt werden.

---

## 5 Elaboration Phase

---

### 5.1 Zusammenfassung und Definitionen

---

**Grobkonzeption** Beschreibt die allgemeine Funktionsweise der Anwendung, untersucht den Ist-Zustand und beschreibt den Soll-Zustand.

**Feinkonzeption** Beschreibt den Soll-Zustand in genauerem Detaillierungsgrad.

**Testing** Steckt den Umfang für das Testen der Anwendung.

**Entwicklungsumgebung** Beschreibt die Architektur in der Entwicklung.

**Produktionsumgebung** Beschreibt im groben, wie die Anwendung im End-Zustand aufgeschaltet werden soll.

### 5.2 Grobkonzeption

---

**Situation** Die bisherigen Inhalte laufen über einen statischen Apache Webserver.

**Zielsetzung** Am Grundkonzept der bestehenden Webseite soll nichts geändert werden. Daher wird die zusätzliche Funktionalität durch den Anfrageprozess folgendermassen ablaufen:

- 1) Rendering: Serverseitige Vorlagen
- 2) Data source: Database via MySQL
- 3) Reverse Proxy and serverseitiges Routing

Der erste und zweite Punkt beschreibt den Aufbau einer Full-stack Anwendung und um die Funktionalität der bestehenden Webseite weiterhin zu gebrauchen, wird der gesamten Anwendung ein Reverse Proxy vorgeschaltet. Dieser unterscheidet zwischen Anfragen auf den bestehenden statischen Teil und dem dynamischen neu erarbeiteten Teil.

#### 5.2.1 Technologie und Frameworks

---

**Ist-Zustand** Die bisherige Situation beschreibt einen statischen Apache Webserver mit folgenden Komponenten und Frameworks für die Erzeugung der HTML Elemente:

Technologie	Beschreibung	Version
Bootstrap	UI Framework, Front End Toolkit für Responsive Design	4.0.0-beta.2
Popper	Positionierung von UI Elementen CSS Alternative	1.12.5
JQuery	Eventhandling, DOM Traversierung, Animation, Ajax	3.2.1
Apache	Webserver	-
Font Awesome	Fonts and Icons	-
Google Font API	Fonts	-

**Tabelle 1: Technologien und Frameworks**
**Soll-Zustand**

Optimal wäre die bestehende Struktur nicht umzustellen. Das heisst, dass weiterhin die meisten Webinhalte über den Apache Web Server statische zurückgegeben werden.

Die neuen Funktionalitäten verlangen jedoch nach dynamischen Inhalten, sowie User Access Control, Datenbankzugriffe und Form Verarbeitungen. Dazu würde sich eine NodeJs Umgebung besser eignen.

Die Idee ist, dass der Teil der Website, der den Bewerbungsprozess anbieten soll, durch einen Reverse Proxy an eine Node App weitergeleitet wird. Dort sollen die Formulardaten verarbeitet und aufbewahrt werden.

Ein Nginx Reverse Proxy wird verwendet, um den Benutzer auf die richtige Anwendung weiterzuleiten.

**5.2.2 Datenbanken**
**Ist-Zustand**

Alle Dateien werden im Filesystem eines Computers aufbewahrt.

**Soll-Zustand**

Alle vorhandenen Daten sollen auf eine Datenbank gespielt werden, die auf dem Server abgelegt ist.

Bei Bedarf werden der Projektbericht sowie das Projektvideo manuell heruntergeladen und auf einem lokalen PC abgespeichert.

**5.2.3 Deployment**
**Ist-Zustand**

Die bisherige Anwendung ist auf einem Apache Server abgelegt. Gehostet wird der Server von Netzone (<https://www.netzone.ch/de/webhosting>)

**Soll-Zustand**

Das Hosting soll falls möglich nicht verändert werden. Die zusätzlichen Services sollen nebenläufig auf demselben Server betrieben werden.

### 5.3 Feinkonzeption

**Situation** Die bisherigen Inhalte laufen über denselben Apache Webserver, wobei die neuen Inhalte eingespielt werden, sobald ein Benutzer auf den Bewerbungsteil gelangt.

- 1) Rendering: Serverseitige Vorlagen
- 2) Data source: Database via MySQL
- 3) Reverse Proxy and serverseitiges Routing

**Zielsetzung** Eine eigenständige Node Anwendung liefert den Bewerbungsteil. Eine mySQL Instanz kümmert sich um die Datenhaltung. Der Reverse Proxy nimmt alle Server Anfragen entgegen. Je nach Route wird ein Benutzer auf die entsprechende Anwendung weitergeleitet. Alle Bewerbungen erreichen einen Node Applikation entwickelt mit Express und Handlebars. Alle Datenbankabfragen werden vom Node Service aus an den mySQL Service gesendet.

**Visualisierung**

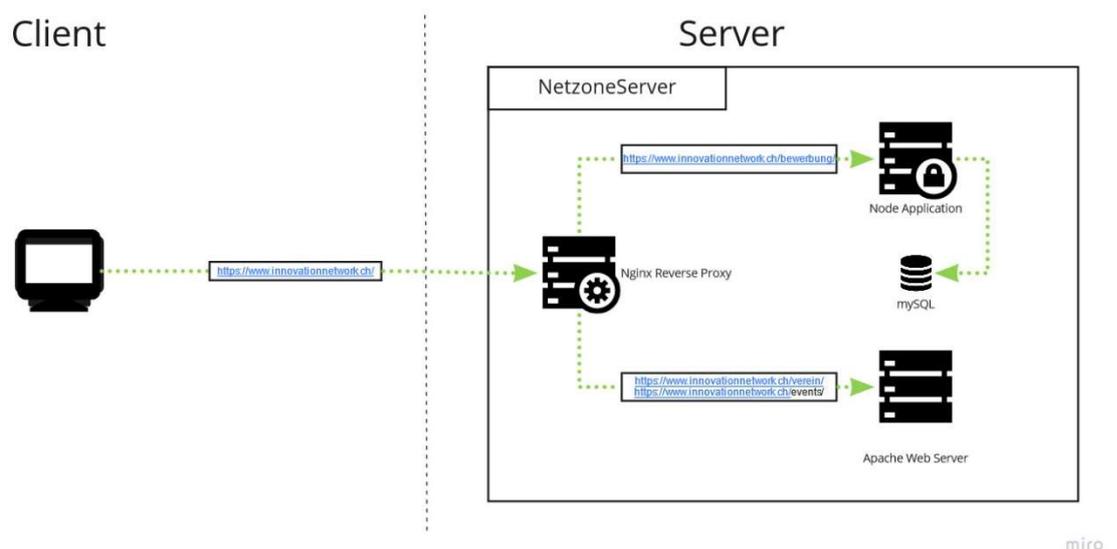


Abb. 4: Infrastruktur Diagramm

### 5.3.1 Nginx Reverse Proxy

**Allgemeine Funktionalität** Eintreffende Anfragen werden an den richtigen Service weitergeleitet. Dies hat den Vorteil, dass am bestehenden Apache Web Server keine Veränderungen durchgeführt werden müssen. Die gesamte Webseite bleibt bestehen. Die einzige Abweichung ist, dass die Bewerbungsseite nicht vom Apache Service geliefert wird, sondern von einem eigenständigen Node Service.

Service	Endpunkt	Route
Apache	Willkommens Seite	<a href="https://www.innovationnetwork.ch">https://www.innovationnetwork.ch</a>
	Vereinsübersicht	<a href="https://www.innovationnetwork.ch/verein/">https://www.innovationnetwork.ch/verein/</a>
	Veranstaltungen	<a href="https://www.innovationnetwork.ch/events/">https://www.innovationnetwork.ch/events/</a>
Node	Bewerbung	<a href="https://www.innovationnetwork.ch/bewerbung/">https://www.innovationnetwork.ch/bewerbung/</a>

Sobald eine Anfrage auf den Node Server trifft, übernimmt dieser intern das Routing.

**Bemerkung** In der Entwicklung wurde ein Sample Apache Web Server benutzt, der nicht alle Routen abdeckt. Beim Aufschalten sollte dies jedoch kein Problem sein, da alle Endpunkt der bestehenden Anwendung direkt angesteuert werden können und nur jene Anfragen, die auf dem Endpunkt *Bewerbung* landen, werden vom Node Service behandelt.

### 5.3.2 Node Anwendung

**Allgemeine Funktionalität** Die Node Anwendung nimmt alle Anfragen auf den Endpoint *Bewerbung* entgegen. Dieser Service beschreibt den eigentlichen Projektumfang und soll eine eingehende Bewerbung behandeln, Bewerbungen anzeigen sowie die Prüfer und Admin Interaktionen bereitstellen.

#### 5.3.2.1 Routing

**Routing** Die Node Applikation übernimmt den Endpoint *Bewerbung* vom Nginx Proxy Servern als Eintrittspunkt.

Entrypoint	Route
Bewerbung	<a href="https://www.innovationnetwork.ch/bewerbung/">https://www.innovationnetwork.ch/bewerbung/</a>

Ab diesem Punkt wird das Routing von der Node Applikation übernommen.

Folgend werden die internen Endpoints aufgeführt und erklärt.

**Internes Routing**

<b>Zugelassene User</b>	Anfrager
<b>Access Control</b>	Nein
<b>Typ</b>	Interaktion
<b>Use Cases</b>	StelleAnfrage, StelleFortführendeAnfrage
<b>Endpoint</b>	<b>Route</b>
Neue Anfrage Auswahl	<a href="https://www.innovationnetwork.ch/bewerbung/new/">https://www.innovationnetwork.ch/bewerbung/new/</a>
Neue Anfrage	<a href="https://www.innovationnetwork.ch/bewerbung/new?schule=eth">https://www.innovationnetwork.ch/bewerbung/new?schule=eth</a> <a href="https://www.innovationnetwork.ch/bewerbung/new?schule=ost">https://www.innovationnetwork.ch/bewerbung/new?schule=ost</a> <a href="https://www.innovationnetwork.ch/bewerbung/new?schule=allgemein">https://www.innovationnetwork.ch/bewerbung/new?schule=allgemein</a>
Anfrage Upload	<a href="https://www.innovationnetwork.ch/bewerbung/upload">https://www.innovationnetwork.ch/bewerbung/upload</a>
Fortführende Anfrage	<a href="https://www.innovationnetwork.ch/bewerbung/fortfuehrendeAnfrage/:id">https://www.innovationnetwork.ch/bewerbung/fortfuehrendeAnfrage/:id</a>
Fortführende Anfrage Upload	<a href="https://www.innovationnetwork.ch/bewerbung/uploadFortfuehrendeAnfrage">https://www.innovationnetwork.ch/bewerbung/uploadFortfuehrendeAnfrage</a>

<b>Zugelassene User</b>	Anfrager, Admin, Mitglied, Prüfer
<b>Access Control</b>	Nein
<b>Interaktion</b>	View-Only
<b>Use Cases</b>	EinsehenAllerAnfragen, EinsehenAnfrage
<b>Endpoint</b>	<b>Route</b>
Alle Anfragen	<a href="https://www.innovationnetwork.ch/bewerbung/">https://www.innovationnetwork.ch/bewerbung/</a>
Individuelle Anfrage	<a href="https://www.innovationnetwork.ch/bewerbung/anfrage/:id">https://www.innovationnetwork.ch/bewerbung/anfrage/:id</a> <a href="https://www.innovationnetwork.ch/bewerbung/anfrage/:id/video/id">https://www.innovationnetwork.ch/bewerbung/anfrage/:id/video/id</a> [Video] <a href="https://www.innovationnetwork.ch/bewerbung/anfrage/:id/uploads/id">https://www.innovationnetwork.ch/bewerbung/anfrage/:id/uploads/id</a> [PDF]

<b>Zugelassene User</b>	Admin, Prüfer
<b>Access Control</b>	Ja
<b>Interaktion</b>	Interaktion
<b>Use Cases</b>	ÄndereAnfrage, BehandleAnfrage, EinsehenAllerAnfragen, EinsehenAnfrage
<b>Endpoint</b>	<b>Route</b>
Login	<a href="https://www.innovationnetwork.ch/bewerbung/login/">https://www.innovationnetwork.ch/bewerbung/login/</a> <a href="https://www.innovationnetwork.ch/bewerbung/auth">https://www.innovationnetwork.ch/bewerbung/auth</a>
AnfragenListe	<a href="https://www.innovationnetwork.ch/bewerbung/">https://www.innovationnetwork.ch/bewerbung/</a> [Login]
Individuelle Anfrage	<a href="https://www.innovationnetwork.ch/bewerbung/anfrage/:id">https://www.innovationnetwork.ch/bewerbung/anfrage/:id</a> [Login] <a href="https://www.innovationnetwork.ch/bewerbung/einsehen">https://www.innovationnetwork.ch/bewerbung/einsehen</a> [Token]
Bearbeitung	<a href="https://www.innovationnetwork.ch/bewerbung/interaktionPruefer">https://www.innovationnetwork.ch/bewerbung/interaktionPruefer</a> [Login]

	<a href="https://www.innovationnetwork.ch/bewerbung/interaktionAdmin">https://www.innovationnetwork.ch/bewerbung/interaktionAdmin</a> [Login]
Logout	<a href="https://www.innovationnetwork.ch/bewerbung/logout">https://www.innovationnetwork.ch/bewerbung/logout</a>

### 5.3.2.2 Access Control

**Access Control** Die Zugriffskontrolle soll von einem normalen Nutzer (Anfrager/Mitglied) nicht bemerkt werden.

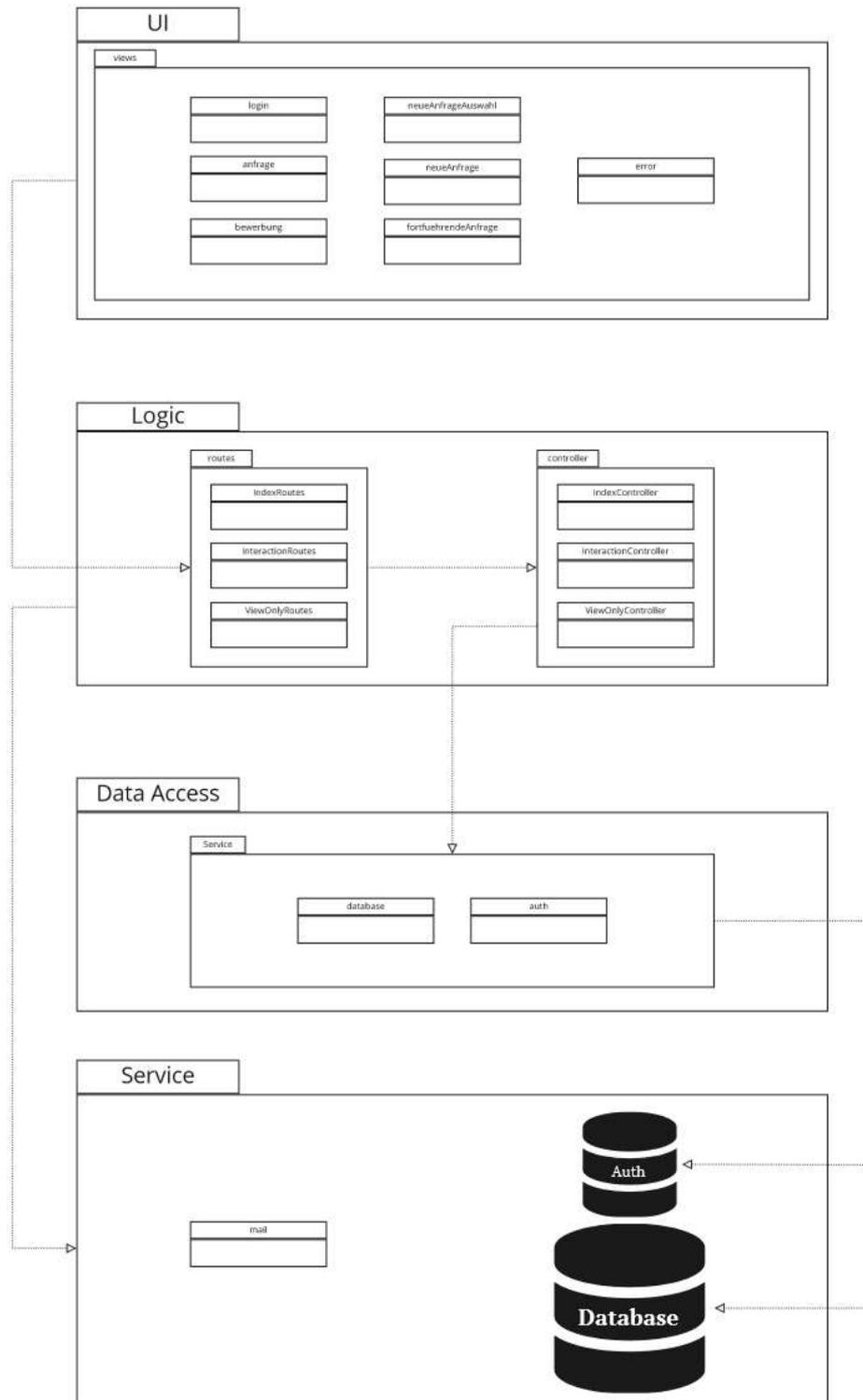
Daher wird eine Typunterscheidung zwischen View-Only und Interaktion Seiten gemacht. Die Interaktionsseite erlaubt einem zugelassenen User (Admin/Prüfer) gewisse Aktionen durchzuführen oder einen normalen User (Anfrager) neue Anfragen zu stellen. Diese Aktionen werden vom System auf die Datenbanken migriert und sind anschliessend wieder auf den View-Only Seiten durch alle User einsehbar.

Um dieses Konzept umzusetzen, wird folgendes benötigt:

Login	Admin/Prüfer sollen sich durch eine Session einloggen → Zeitliche Beschränkung auf dem Login
Admin/Prüfer Interaktionen	Die Seiten sollen Interaktion erlauben, solange ein gültiges Login vorgezeigt wird. Dafür muss die Node Applikation auf derselben Route zwischen Interaktion-Usern und View-Only-Usern unterscheiden und dementsprechend die korrespondierenden Seiten (Interaktion vs. View-Only) laden können.
Anfrager Interaktionen	Ein Anfrager braucht kein Login. Dies vereinfacht den Anfrageprozess für einen Anfrager. Nach dem Einreichen der Anfrage muss dem Anfrager ein Token übergeben werden. Dieses Token erlaubt eine vereinfachte Wiederaufnahme der Interaktion beim Fall einer Fortführenden Anfrage.

### 5.3.2.3 Schichten Diagramm

#### Schichten



miro

Abb. 5: Schichten Diagramm

### 5.3.2.4 UI

**UI** Es gibt zwei verschiedene Gruppierung für die Views.

1. View Only
2. Interaktion

Gruppe	Beschreibung	Benutzer
View Only	Betrifft die Ansichten mehrere oder individuellen Bewerbungen	Alle
Interaktion ohne Login	Betrifft den Prozess einer neuen Anfrage und der fortführenden Anfrage Jeweils nur für den mit einer Anfrage assoziierten Anfrager. Betrifft den Prozess der Behandlung der Prüfung sowie alle administrativen Handlungen	Admin Anfrager Prüfer

Die Prototypen Views können im Anhang eingesehen werden.

### 5.3.2.5 Logic

**Index Controller** Hier soll eine Unterscheidung zwischen eingeloggten und regulären Benutzer gemacht werden. Dabei wird entweder eine reguläre Session oder eine entsprechende Admin oder Prüfer Session ausgestellt. Um die drei verschiedenen Gruppen aus dem vorherigen Abschnitt abbilden zu können, muss ein Routing mit Access Control eingebaut werden. Dieses Login soll etwas versteckt werden und nur von Admin und Prüfer benutzt werden.

**Interaktion Controller** Alle Interaktionen mit und ohne Login sollen durch diesen Controller zur Verfügung gestellt werden.

**View-Only Controller** Alle Views im View Only Status sollen durch diesen Controller zur Verfügung gestellt werden.

### 5.3.2.6 Data Access

**Bewerbung** Stellt den Zugang zur Datenbank mit den Anfragen zur Verfügung.

**Auth** Stellt den Zugang zur Datenbank mit den Auth-Daten zur Verfügung.  
Die Benutzer Rollen und die Berechtigungen werden durch Mitsenden eines Client Side Session Cookies nach dem Login gewährleistet.

### 5.3.3 MySQL

#### Übersicht

Es werden zwei Datenbanken zur Verfügung gestellt, welche auf dem Server gehostet sind.

Datenbank	Beschreibung
Database	Hält alle Anfrage Daten
Auth	Hält alle Login Daten

#### Database

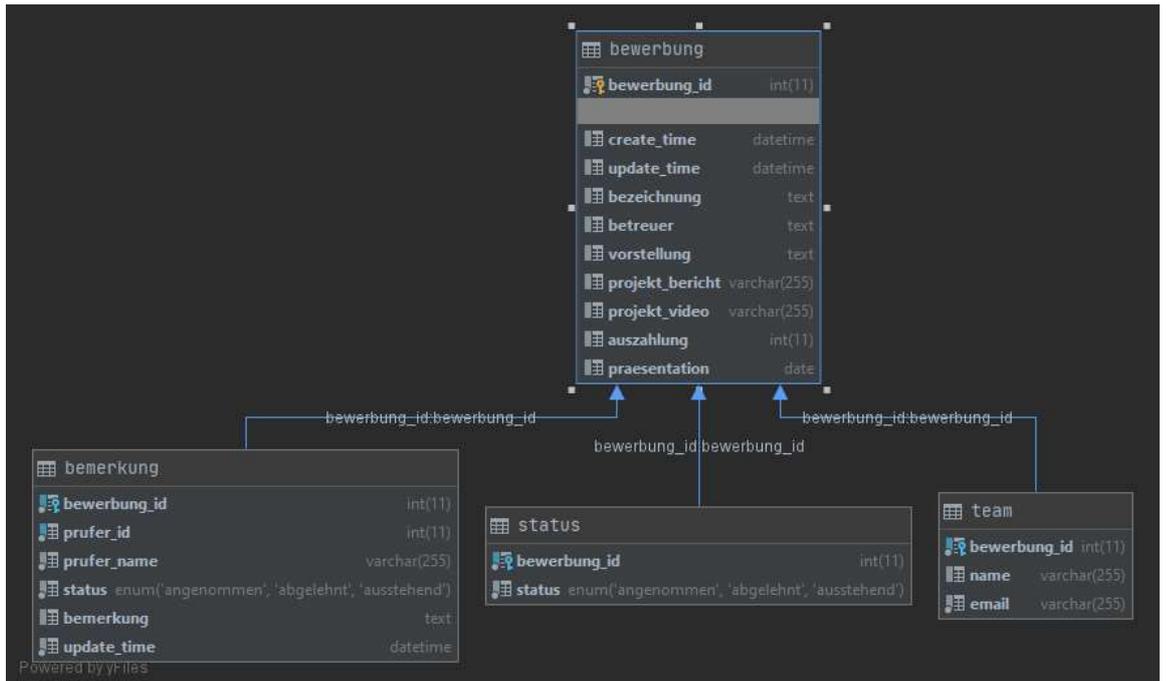


Abb. 6: Datenschema Entwurf

Der Entwurf des Bewerbungsdatenschemas ist an das Domain Model angelehnt.

#### Auth

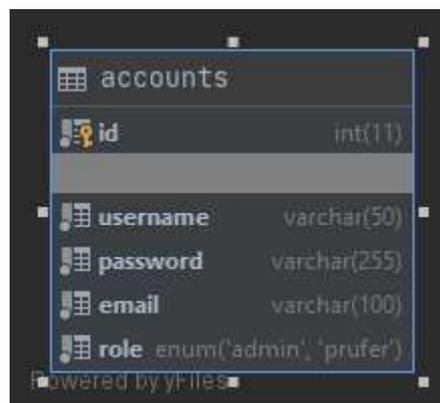


Abb. 7: Auth Datenschema Entwurf

Die Auth Datenbank soll neben den Login Daten ebenfalls die Rolle des jeweiligen Benutzers enthalten. Hierbei kann ein Enum mit den Werten «prufer» und «admin» benutzt werden.

**Bemerkung** Aus Sicht des Verfassers macht es an dieser Stelle wenig Sinn sich tiefer mit den Schemas auseinander zu setzen, da es im Verlaufe der Arbeit und der Entwicklung der Anwendung zu Änderungen kommen kann. Eine genauere Beschreibung der Schemas folgt in den Ausführungen der Construction Phase.

## 5.4 Testing

**Allgemein** Das Testen befasst sich fast ausschliesslich mit der Node Applikation. Der Apache Web Service muss nicht mit Unit Tests abgedeckt werden, da er bereits in einer Produktionsumgebung läuft. Der Nginx Reverse Proxy kann einfach im Integrations bzw. End-to-End Test geprüft werden.

### 5.4.1 Unit Test

**Unit Test** Unit Test sollen einzelne Funktionalitäten der Website testen. Dabei handelt es sich ausschliesslich um die Veränderung der Daten während dem Bewerbungsprozesses. Folgende Tests sollen dabei durchgeführt werden

<b>Aktion</b>	<b>Beschreibung</b>
Insert	Eine neue Anfrage wird in die DB geschrieben
Update	Eine fortführende Anfrage oder eine Admin Interaktion aktualisiert einen bereits existierenden Eintrag
Read	Eine Anfrage soll vollständig gelesen werde

**Tabelle 2: Unit Test Beschreibung**

### 5.4.2 Integration Test

**Integration Test** Die Integration Tests überprüfen die Korrektheit des gesamten Prozesses. Er kann also an die Use Cases angepasst werden. Dabei werden Client Aktionen simuliert und getestet, ob der Server richtig reagiert und die richtigen Resultate zurückliefert.

Use Case	Aktionen
StelleAnfrage	Datenbankeintrag insert
StelleFortführendeAnfrage	Datenbankeintrag update
BehandleAnfrage	Annahme Simulation Ablehnung Simulation Benachrichtigung Simulation Datenbankeintrag update
EinsehenAllerAnfragen/ EinsehenAnfrage	Datenbankeintrag read
AktualisiereAnfrage	Datenbankeintrag update
BenachrichtigeUser	Admin Simulation Prüfer Simulation Anfrager Simulation

Tabelle 3: Integration Test Beschreibung

### 5.4.3 System/ End-to-End Test

**System Test** Das gesamte System soll in einem Probedurchlauf durch Simulation aller Parteien getestet werden.

Personas	Aktionen
Anfrager	Neue Anfrage Fortführende Anfrage Benachrichtigung
Prüfer	Behandeln der Anfrage Benachrichtigung
Admin	Update der Anfrage Benachrichtigung
Mitglieder	Einsehen der Anfragen

Tabelle 4: End-to-End Test Beschreibung

---

#### 5.4.4 User Acceptance Test

---

**System Test**

Der *User Acceptance Test* ist die letzte Hürde im Projekt. Dabei wird die Applikations-Logik durch echte User getestet. Dabei sollen alle Parteien in einem echten Umfeld die Funktionalität der Anwendung testen. Dabei sind dem *System under Testing (SUT)* keine Beschränkungen auferlegt.

Folgende Testpersonen müssen dabei teilnehmen:

- Anfrager
- Fünf Personen in der Rolle der Prüfer oder Vorstand des Innovation Networks selbst
- Admin

---

#### 5.4.5 Tools

---

**Framework**

Als Testing Framework wird *Jest* benutzt, da alle notwendigen Testing Features unterstützt werden, wie zum Beispiel:

- Test Launchers
- Assertion Functions
- Generate and display test progress and summary
- Mocks, spies, and Stubs
- Generate code Coverage

---

### 5.5 Entwicklungsumgebung

---

**Docker**

Das Projekt ist in vier Teilapplikationen aufgeteilt. Für die Entwicklung werden daher mindestens drei Docker Container zur Verfügung gestellt, da der statische Apache Webserver nicht unbedingt dabei sein muss.

Dabei wird jeweils die gesamte Ordnerstruktur als Volume in den jeweiligen Container geladen. Die Node Anwendung wird dann durch einen internen Eingabebefehl gestartet. Der Nginx sowie der MySQL Service sind nach der Initialisierung bereits vollkommen konfiguriert und funktionsfähig.

Der MySQL Service bekommt auch Dummy Daten bereits bei einer Initialisierung eingespeist und daher kann die gesamte Anwendung zu Testzwecken auch auf einem neuen System direkt via docker-compose gestartet werden.

---

## 5.6 Production Environment

---

### Allgemein

Die Struktur in der Entwicklung kann direkt auf die Produktion erweitert werden. Dafür wird dem momentan bestehenden Server ein Nginx Service vorgeschaltet. Es wird erwartet, dass neben der Konfigurationsdatei keine weiteren Schritte benötigt werden.

Die Migration des Node Services in die Produktionsumgebung verläuft nicht viel anders. Dabei werden jegliche Dateien auf dem Server abgelegt und anschliessend gestartet. Die Funktionalität sollte ohne weitere Konfiguration gewährleistet sein.

Der Apache Web Server läuft bereits und muss im günstigsten Fall gar nicht und ansonsten nur minim konfiguriert werden.

Der MySQL Server muss etwas konfiguriert bzw. mit den bereits bestehenden Daten abgefüllt werden.

## 6 Construction Phase

### Übersicht

In der Construction Phase wird der Entwurf der vorherigen Phase umgesetzt. Anschließend werden der Aufbau sowie die rudimentären Konfigurationen der verschiedenen Anwendungen und Services genauer umschrieben.

Die Informationen sind aus dem Projektverzeichnis entnommen und können bei Bedarf hinzugezogen werden.

### 6.1 Nginx

**Zusammenfassung** Dieser Abschnitt enthält die Konfiguration des Nginx Reverse Proxy. Die Aufgabe des Proxy besteht in der korrekten Weiterleitung auf die entsprechenden Anwendungen und Services.

#### Konfigurationsdatei

```
server {
    listen 443 ssl;

    root /srv/www;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;

    client_max_body_size 200M;

    proxy_hide_header X-Powered-By;
    add_header X-Frame-Options SAMEORIGIN;
    add_header Strict-Transport-Security max-age=15768000;

    ssl_password_file /etc/nginx/ssl/password.pass;
    ssl_certificate /etc/nginx/ssl/cert.pem;
    ssl_certificate_key /etc/nginx/ssl/key.pem;

    location / {
        proxy_pass http://apache:8090/index.html;
    }

    location @apache {
        proxy_pass http://apache:8090;
    }

    location /bewerbung {
        proxy_pass http://nodejs:8080;
    }
}
```

Abb. 8: Nginx Konfiguration

---

Die meisten Angaben in der Konfiguration sind standardmässig für einen Reverse Proxy. Die wichtigsten Änderungen sind unten aufgeführt.

- HTTPS** Der Nginx Reverse Proxy läuft über den Standard HTTPS Port 443 und kümmert sich damit auch gleich um die Transport Layer Security der dahinter geschalteten Applikationen.  
In der Entwicklungsumgebung ist das Zertifikat durch die openssl CLI selbst erstellt und unter `/etc/nginx/ssl/` im Docker Container hinterlegt.
- Body Size** Die Option `client_max_body_size 200M;` wird gesetzt, um den Upload von grösseren Videodateien über den Proxy zu erlauben. Die maximale Grösse eines Files beschränkt sich auf 200MB.
- Location** Die `location` Optionen lösen die URLs auf und leiten die jeweiligen Anfragen an den richtigen Service weiter. Somit wird beispielsweise eine Anfrage auf `https://www.innovation-network.ch/bewerbung` direkt an die Node Applikation weitergeleitet.

## 6.2 Node Anwendung

---

**Zusammenfassung** In diesem Abschnitt wird die Node Anwendung genauer dargestellt und gewisse Designentscheidungen erläutert.

Die Applikation ist nach den Ausarbeitungen der Elaboration Phase zusammengesetzt. Die Absicht dahinter ist, dass eine gut verständliche Trennung der Interessen gewährleistet werden kann.

Die Basis bilden eine serverseitig gerenderte Node Express Anwendung mit der Handlebars View Engine.

## 6.2.1 Allgemeiner Aufbau

### 6.2.1.1 Struktur

#### Baumstruktur

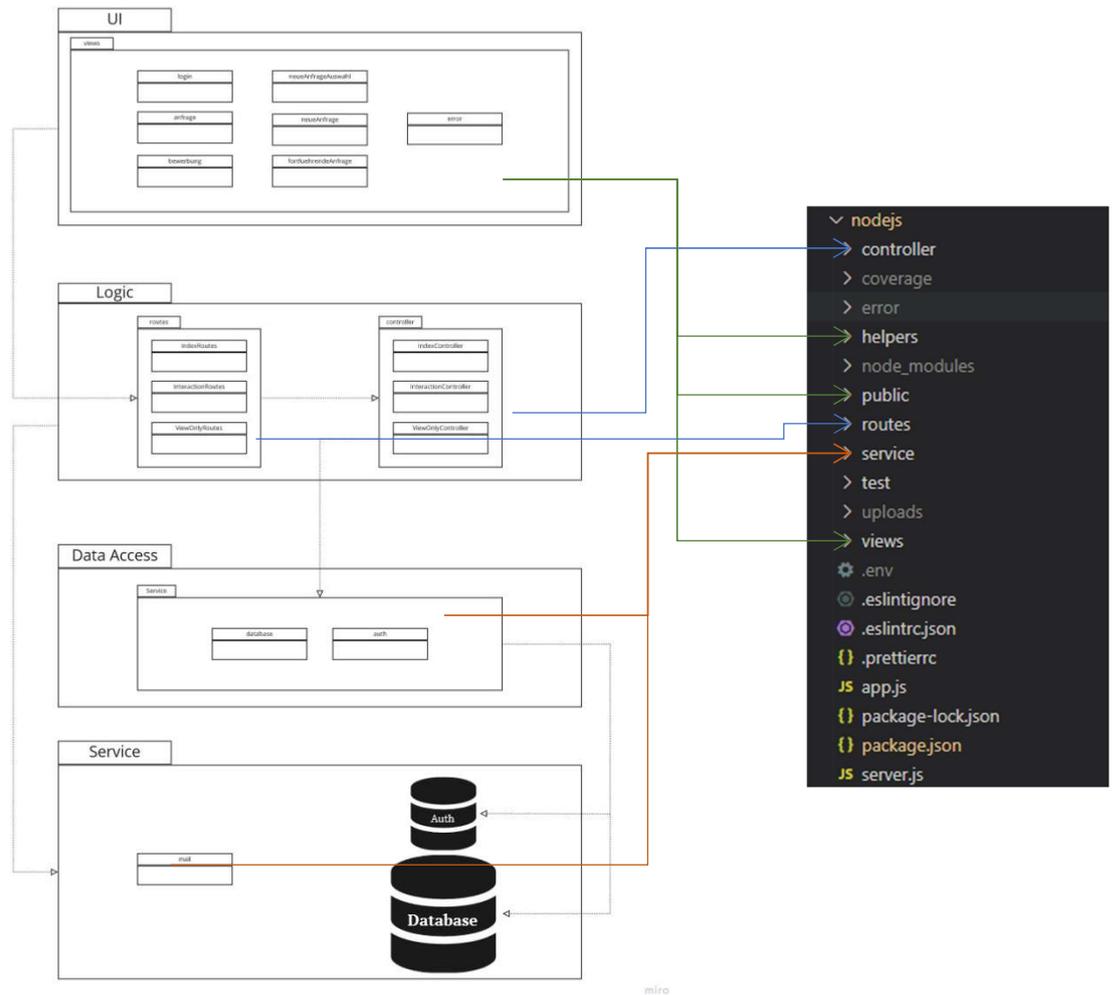


Abb. 9: Schichten Diagramm zu Ordnerstruktur

Ausgegraute Ordner sind im `.gitignore` aufgelistet.

Der Aufbau und die Funktionalität der Komponenten können den Schichten im Schichten Diagramm zugeordnet werden.

### 6.2.1.2 Abhängigkeiten

package.json

```
{
  "name": "bewerbung_express",
  "version": "1.0.0",
  "description": "Innovation Network Bewerbungsprozess.",
  "main": "index.js",
  "scripts": {
    "test": "jest --detectOpenHandles --forceExit",
    "coverage": "jest --coverage --forceExit",
    "lint": "eslint ./ ",
    "lint:fix": "npm run lint -- --fix"
  },
  "author": "Etienne Baumgartner",
  "license": "ISC",
  "dependencies": {
    "connect-flash": "^0.1.1",
    "crypto": "^1.0.1",
    "express": "^4.17.1",
    "express-handlebars": "^5.3.4",
    "express-session": "^1.17.2",
    "multer": "^1.4.3",
    "mysql": "^2.18.1",
    "nodemailer": "^6.7.2",
    "passport": "^0.5.0",
    "passport-local": "^1.0.0",
    "promise-mysql": "^5.0.4",
    "winston": "^3.3.3",
    "dotenv": "^10.0.0"
  },
  "devDependencies": {
    "eslint": "^8.4.1",
    "eslint-config-airbnb-base": "^15.0.0",
    "eslint-config-prettier": "^8.3.0",
    "eslint-plugin-import": "^2.25.3",
    "eslint-plugin-prettier": "^4.0.0",
    "jest": "^27.4.4",
    "prettier": "^2.5.1"
  }
}
```

Abb. 10: package.json

#### Scripts

Es sind zwei Skripte für die Benutzung der Jest Framework und zwei zur Sicherstellung der Coding Vorschriften definiert.

#### Abhängigkeiten

Es wird versucht die Abhängigkeiten des Projekts so minim wie möglich zu halten.

Technologie Stack	Technologie	Beschreibung	Version
	Node	Asynchrone event-driven JavaScript Runtime	alpine

Tabelle 5: Node Technologie Stack

Dependencies	Package	Beschreibung & Zweck	Version
	express	Serverseitige Web Framwork	^4.17.1
	express-handlebars	Template Engine	^5.3.4
	express-session	Session Handling	^1.17.2
	promise-mysql	Datenbank Driver mit Promises	^5.0.4
	crypto	Hashing und Salt für Passwörter	^1.0.1
	passport	Session Controller für Authentifizierung	^0.5.0
	passport-local	Session Controller mit lokalen Datenbanken	^1.0.0
	multer	Upload von Dateien	^1.4.3
	connect-flash	Übermitteln von Informationen via Session	^0.1.1
	nodemailer	Übermitteln von Notifikationen via Mail	^6.7.2
	winston	Logging von Errors	^3.3.3
	dotenv	Einlesen der .env Datei	^10.0.0

Tabelle 6: Node Abhängigkeiten

Styling und Fonts	Library	Beschreibung	Version
	Bootstrap	Frontend Toolkit	5.1.3
	Font Awsome	Fonts und Icons	-
	Google Font API	Fonts	-

Tabelle 7: Node Styling

Bootstrap ist verantwortlich für die Darstellung für das existierende Projekt. Daher wird für das Styling der Node Anwendung dieselbe Datei benutzt, um einen nahtlosen Übergang zu gewährleisten, der vom Benutzer nicht weiter bemerkt werden soll.

**devDependencies** Diese Packages werden nur in der Entwicklung benötigt. Sie ermöglichen eine vereinfachte Entwicklung, Testen und um die einheitliche Darstellung des geschriebenen Codes.

### 6.2.1.3 Routes

#### Übersicht

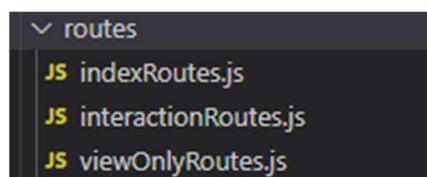


Abb. 11: Node Routes

Die Routes sind eine Abstraktion für das eigentliche Routing der Express Anwendung.

- indexRoutes.js** Nimmt die Routen für das Login, Logout und die Authentifizierung und bindet die entsprechenden Funktionen aus dem Controller.  
Hier wird auch die Einstiegsroute bzw. die Bewerbungsansicht beim ersten Aufruf eingebunden.
- interactionRoutes.js** Nimmt alle Routen entgegen, welche mit Benutzer Interaktionen zu tun haben. Ausserdem wird hier direkt mit Multer gearbeitet, um die Uploads von PDF und Video abzufangen.
- viewOnlyController.js** Nimmt die Routen aller übrigen Anfragen und bindet sie an den ViewOnly Controller.

## 6.2.1.4 Controller

### Übersicht

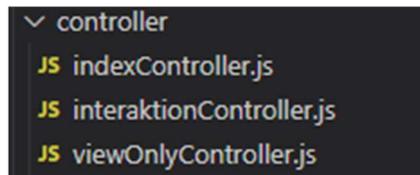


Abb. 12: Node Controller

Die Controller werden der entsprechenden Route aus dem obigen Abschnitt zugeordnet und stellen die Logik zur Verfügung.

**Index Controller** *indexController.js* kümmert sich ausschliesslich um das Login, Logout und die Authentifizierung von eingehenden User Anfragen.

**Interaktion Controller** *interaktionController.js* kümmert sich um Abhandlung von Form Eingaben bzw. User Interaktionen. Es gibt fünf Situationen, in denen eine Interaktion durch den Benutzer notwendig ist:

1. Neue Anfrage einreichen
2. Fortführende Anfrage einreichen
3. Nicht behandelte Anfrage mit Token einsehen
4. Prüfer Interaktionen
5. Admin Interaktionen

Alle diese Situationen werden durch die Funktionen in diesem Controller abgedeckt.

**View-Only Controller** *viewOnlyController.js* kümmert sich um die Darstellung aller Views, bei denen keine Interaktionen durch den Benutzer vorgesehen ist. Die folgenden Punkte beschreiben die vier Situationen, in denen dieser Controller benutzt wird:

1. Bewerbungen einsehen
2. Anfrage einsehen
  - a. Projektvideo Download
  - b. Projektbericht Download

## 6.2.1.5 Service

### Übersicht

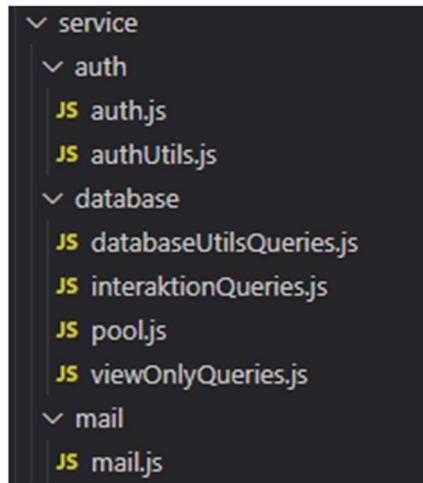


Abb. 13: Node Service

Die Anwendung hat drei Services. *auth* und *database* enthalten die Abfragen für die entsprechende Datenbank und *mail* kümmert sich um das Versenden der E-Mail-Notifikationen.

#### Auth

Der Auth-Service nimmt Login Anfragen entgegen und serialisiert und de-serialisiert den User in der Session. *authUtils.js* enthält die kryptographischen Funktionen für die Passwort Validierung.

#### Database

*interaktionQueries.js* enthält alle Datenbank Zugriffe für den Interaktion Kontroller. Es werden folgende Aktionen zur Verfügung gestellt:

1. Neue Anfrage hinzufügen
2. Update der Admin Interaktionen (Präsentationsdatum & Auszahlung)
3. Update der Prüfer-Stati und Bemerkungen
4. Einsehen des Bewerbung-Stati
5. Einsehen einer Bewerbung durch Projekt Token

*viewOnlyQueries.js* enthält alle Datenbank Zugriffe für den ViewOnly Kontroller. Es werden folgende Aktionen zur Verfügung gestellt:

1. Einsehen aller Bewerbungen
2. Einsehen aller angenommenen Bewerbungen
3. Einsehen einer Bewerbung (Prüfer/Admin)
4. Einsehen einer Bewerbung (Anfrager, Mitglied)
5. Einsehen aller Bemerkungen
6. Einsehen der Projekt Video URI

Wichtig bei *viewOnlyQueries.js* ist, dass es jeweils eine Unterscheidung zwischen den Benutzer gibt. Je nach Benutzer werden mehr oder weniger Informationen von der Datenbank abgefragt.

*pool.js* stellt eine Ansammlung von möglichen Datenbankverbindungen dar. Wenn eine Abfrage ausgeführt wird, wird eine Verbindung aus diesem «Connection Pool» genommen, um die Abfrage durchzuführen. Nach der Behandlung wird diese Verbindung wieder

in den «Connection Pool» freigelassen. Dieser Pool ist eine Singleton Instanz, damit alle Abfragen über dasselbe Verbindungsobjekt ablaufen.

## Mail

Der Mail Service basiert auf dem *nodemailer* Package. Hier sind die Funktionen zur Benachrichtigung der User definiert und werden unter gewissen Bedingungen abgefeuert. Folgende Benachrichtigungen und Bedingungen sind unterstützt:

1. Notifikation beim Eingang einer neuen Bewerbung
2. Notifikation beim Abschluss des Prüfungsprozesses
3. Notifikation beim Eingang einer fortführenden Bewerbung

Die Nachrichten enthalten die relevanten Informationen wie Status, Email und Links für die beteiligten Parteien.

Alle beteiligten Benutzer erhalten bei der jeweiligen Erfüllung der Bedingungen eine Notifikation.

## 6.2.1.6 Views

### Übersicht

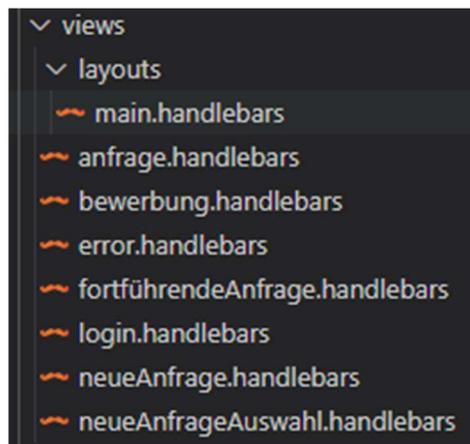


Abb. 14: Node Views

Die Views enthalten alle serverseitig gerenderten Ansichten der Anwendung.

Es ist zu beachten, dass sich die Views von den Prototyp Designs unterscheiden. Diese Änderungen sollen mehr Informationen, sowie eine verbesserte User Experience bieten.

Die Views werden im Anhang aufgeführt.

## 6.2.1.7 Helpers

### Übersicht

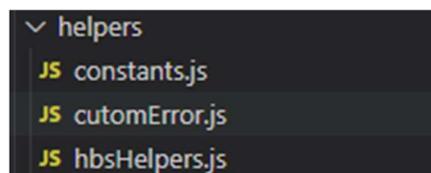


Abb. 15: Node Helpers

Die Helpers enthalten Funktionen oder Objekte, die von mehreren Komponenten benutzt werden.

**Konstanten** *constants.js* enthält gewisse konstante Texte, die bei den Views im Einsatz sind.

**Custom Error** *customError.js* enthält einen Konstruktor für einen Custom Error. Das Error Handling der Anwendung wird mehrheitlich über diesen Custom Error abgehandelt.

Es wird in den Designentscheiden aufgezeigt, wie sich der Aufbau dieser Custom Errors zusammensetzt.

**Helpers** *hbsHelpers.js* enthält alle Helper Funktionen für die Handlebar Views und hilft bei den verschiedenen Darstellungen der unterschiedlichen Benutzer.

### 6.2.1.8 Sonstiges

**Error Log** Der *error* Verzeichnis enthält den Error Log. Hier werden alle Fehlermeldungen aufbewahrt, um bei der Instandhaltung der Applikation behilflich zu sein.

**Tests** Im *test* Verzeichnis befinden sich alle definierten Unit und Integrationstests. Es werden hierbei vor allem die verschiedenen Abfragen auf die Datenbank getestet.

**Code Coverage** Im *coverage* Verzeichnis werden die automatisch generierten Dateien von der Jest Coverage Funktion aufbewahrt.

**Node Modules Abhängigkeiten** Im *node\_modules* Verzeichnis werden alle Abhängigkeiten und Node Module gesammelt.

**Upload** Im *uploads* Verzeichnis werden die Dateien abgelegt, welche bei einer neuen Bewerbung mit hochgeladen werden. Es handelt sich hierbei ausschliesslich um PDFs und mp4 Dateien.

**Serverkonfiguration** Die *server.js* Datei enthält die Konfiguration des Express Servers. Hier werden auch importieren und selbst geschriebenen Middlewares instanziiert.

**Startpunkt** Die *app.js* Datei startet den Server auf dem zugewiesenen Port.

**Umgebungsvariablen** Im *.env* werden sensitive Umgebungsvariablen aufbewahrt, welche nur in der Zielumgebung benutzt werden sollen.

**Eslint Konfiguration** *eslint\*/.prettierrc* enthalten alle Konfigurationen bezüglich *eslint*.

## 6.2.2 Umsetzung und genauere Erklärungen

### 6.2.2.1 Daten Verarbeitung

#### Übersicht

Um die Datenweiterleitung konsistent zu halten, sind alle Rückgabeobjekte der Datenbankabfragen gleich aufgebaut.

Die Struktur des Objekts ist die folgende:

```
const responseObject = { error: null, info: null, data: null }
```

Abb. 16: responseObject

**Error Informationen** Das *error*-Objekt sammelt genauere Informationen bezüglich der Custom Error Propagation. Die enthaltenen Informationen werden einer eigenen Error Klasse übergeben und dort mit zusätzlicher Deklaration angereichert. Dieses Vorgehen erlaubt die Darstellung eigener Error Beschreibungen, um einem User ein besseres Verständnis des vorgefallenen Fehlers zu vermitteln.

Dieses Objekt wird für fast allen geworfenen Fehler bei den Datenabfragen erstellt.

Beispielsweise würde eine Abfrage eines nicht eingeloggten Nutzers auf eine Bewerbung im Status «*Ausstehend*» folgendes Error Objekt generieren:

```
responseObject.error = {  
  message: 'Bewerbung ist nicht zugänglich.',  
  handling: 'redirect',  
  info: 'Berechtigung',  
}
```

Abb. 17: responseObject.error

Das *message* Feld enthält die genauere Beschreibung des aufgetretenen Fehlers. Es wird unter Umständen auf der Error Seite angezeigt. Das *handling* Feld informiert die Error-handling Middleware über die notwendige Aktion. Es sind Fehler vorhanden, welche nicht auf die Error Seite weitergeleitet werden sollen und diese haben dementsprechend anstelle des «*redirect*» eine «*none*» vermerkt. Die *info* Angabe erlaubt der Custom Error Klasse eine Unterscheidung auf die Art des Fehlers. Im Falle von «*Berechtigung*» wird ein 403 Error erzeugt.

**Benutzer Informationen** Das *info*-Objekt soll Informationen zu Zuständen liefern, die vom Nutzer als Fehler verstanden werden könnten. Falls eine *info* auf dem *responseObject* vorhanden ist, wird diese durch Handlebars aufgelöst und dem Nutzer in Form einer Information vorgelegt.

Beispielsweise würde im Falle einer leeren Datenbank bzw. falls alle Bewerbungen im Status «*Ausstehend*» wären, das folgende *info*-Objekt generiert:

```
responseObject.info = {
  message: 'Momentan keine Bewerbungen vorhanden.',
  type: 'info',
}
```

Abb. 18: responseObject.info

Es werden drei verschiedene Infotypen unterstützt und ihre Darstellung ist unten dargestellt:

Typ	Beispieldarstellung
info	 Momentan keine Bewerbungen vorhanden.
error	 Bewerbung nicht vorhanden.
success	 Projekt Token aufbewahren für zukünftigen Zugriff und Bearbeitung der Projektanfrage. 51dbae3e904325cc0dd03293

Tabelle 8: Info Darstellungs Typen

**Daten** Das *data*-Objekt enthält sämtliche Informationen einer Bewerbung, die auf der entsprechenden View benutzt werden.

Eine vollständig bearbeitete Bewerbung, also mit zusätzlicher fortführender Bewerbung, würde folgendes Objekt generieren, wobei *results[0]* die Rückgabe aus der Datenbankabfrage darstellt:

```
const bewerbungData = {
  anfrage: {
    bewerbung_id: results[0].bewerbung_id,
    create_time: results[0].create_time,
    update_time: results[0].update_time,
    bezeichnung: results[0].bezeichnung,
    betreuer: results[0].betreuer,
    status: results[0].status,
    team: results[0].team,
    vorstellung: results[0].vorstellung,
    projekt_bericht: results[0].projekt_bericht,
    projekt_video: results[0].projekt_video,
    auszahlung: results[0].auszahlung,
    praesentation: results[0].praesentation,
  },
  fortfuehrendeAnfrage:
    results[0].fortfuehrende_bemerkung &&
    results[0].fortfuehrender_projekt_bericht
    ? {
      fortfuehrende_bemerkung: results[0].fortfuehrende_bemerkung,
      fortfuehrender_projekt_bericht:
        results[0].fortfuehrender_projekt_bericht,
    }
    : null,
}
returnObject.data = bewerbungData
```

Abb. 19: returnObject.bewerbungData

Das *data*-Objekt enthält alle Daten, die zur Popularisierung einer Bewerbungsanzeige nötig sind.

### 6.2.2.2 Fehler Verarbeitung

#### Custom Error

```
class CustomError extends Error {
  constructor(error) {
    super()
    this.name = this.constructor.name

    this.handling = error.handling
    this.message = error.message
    this.info = error.info

    switch (error.info) {
      case 'Berechtigung':
        this.statusCode = 403
        this.name = 'Zugang eingeschränkt.'
        break
      default:
        this.statusCode = 500
    }
  }
}
```

Abb. 20: Custom Error Klasse

Die Custom Error Klasse kümmert sich um den rudimentären Aufbau eines Fehlers. Tritt ein Fehler auf, werden alle systemgenerierten Informationen, sowie zusätzliche Informationen gebündelt und an diese Klasse weitergegeben. Ein eigenständiger Error wird erzeugt und kann mit den zusätzlichen Informationen bei Bedarf geworfen werden.

Express leitet den Fehler an die entsprechende Error-Handling-Middleware weiter.

Bevor der Nutzer eine Fehler Seite mit den zusätzlichen Informationen angezeigt bekommt, wird der gesamte Fehler mit Hilfe des Nodepackage *winston* in einem log-File abgelegt und zur späteren Betrachtung durch einen Maintainer aufbewahrt.

### 6.2.2.3 Datenzugriff

**promise-mysql** Für die Datenbankzugriffe wird das Package *promise-mysql* benutzt. Es baut auf dem standardmässigen *mysql* Package auf und unterstützt den zusätzlichen Gebrauch von Promisses.

Diese Variante wird bevorzugt, weil die ersten Implementationsversuche mit der Hilfe von Callbacks zu unübersichtlich waren.

Ebenfalls kann durch die Verwendung von Promisses auf wichtige asynchrone Operationen gewartet werden und die Fehlerbehandlung wird erleichtert.

**Connection Pool** Der Verbindungsaufbau wird über einen «Connection Pool» *pool.js* gewährleistet. Dieses Singleton Objekt gibt auf Anfrage eine Verbindung zurück und gibt sie nach erfüllter Abfrage wieder an den Pool zurück. Es werden 20 gleichzeitige Verbindungen erlaubt.

```
const mysql = require('promise-mysql')

const connectionConfig = {
  connectionLimit: 20,
  host: process.env.MYSQL_DATA_HOST,
  user: process.env.MYSQL_DATA_USER,
  password: process.env.MYSQL_DATA_PASSWORD,
  database: process.env.MYSQL_DATA_DATABASE,
  port: process.env.MYSQL_DATA_PORT,
  multipleStatements: true,
}

let tempPool
const pool = getPool()

function getPool() {
  if (tempPool) return tempPool
  tempPool = mysql.createPool(connectionConfig)
  return tempPool
}
```

Abb. 21: Verbindungs Pool

#### Aufbau einer Abfrage

Alle Abfragefunktionen sind alle gleich aufgebaut.

1. Data Objekt
2. Verbindung zur Datenbank via Pool
3. Abfrage String
4. Abfrage ausführen
5. Nutzer Anzeige erstellen (falls nötig)
6. Custom Error Propagation
7. Verarbeitung Abfrage Resultate
8. Abfrage Error Propagation
9. Verbindung Error Propagation
10. Rückgabe des Data Objekts

Das Konzept ist unten durch eine Dummy Abfrage dargestellt:

```
async function Abfrage () {
  // Data Objekt
  const returnObject = { error: null, info: null, data: null }
  try {
    // Verbindung zur Datenbank via Pool
    const connection = await ppool
    // Abfrage String
    const query = 'Abfrage'

    try {
      // Abfrage ausführen
      await connection.query(query).then((results) => {
        // Benutzer Informationsanzeige erstellen

        /*
        returnObject.info = {
          message: 'Wichtige Informationen für den Nutzer',
          type: 'info',
        }
        */

        // Custom Error Propagation

        /*
        returnObject.error = {
          message: 'Bewerbung ist nicht zugänglich.',
          handling: 'redirect',
          info: 'Berechtigung',
        }
        */

        // Verarbeitung Resultat
      })
    } catch (e) {
      // Abfrage Error Propagation
      returnObject.error = {
        message: 'SQL Error',
        handling: 'redirect',
        info: e,
      }
    }
  } catch (e) {
    // Verbindung Error Propagation
    returnObject.error = {
      message: 'Connection Error',
      handling: 'redirect',
      info: e,
    }
  }
}
```

Abb. 22: Abfrageaufbau Konzept

## 6.3 MySQL

### Übersicht

Für die Entwicklung wurden zwei separate mysql Services aufgebaut. Die Begründung dafür ist die einfachere und unabhängige Entwicklung der beiden Datenbanken, ohne dass sich Daten und Authentifizierung in den Weg kommen.

Bei einer Produktionsumgebung kann man diese beiden Services einfach zu einem einzelnen zusammenziehen.

### 6.3.1 Database

#### Schema

Die Data Datenbank wurde angelehnt an das Domain Model entwickelt.

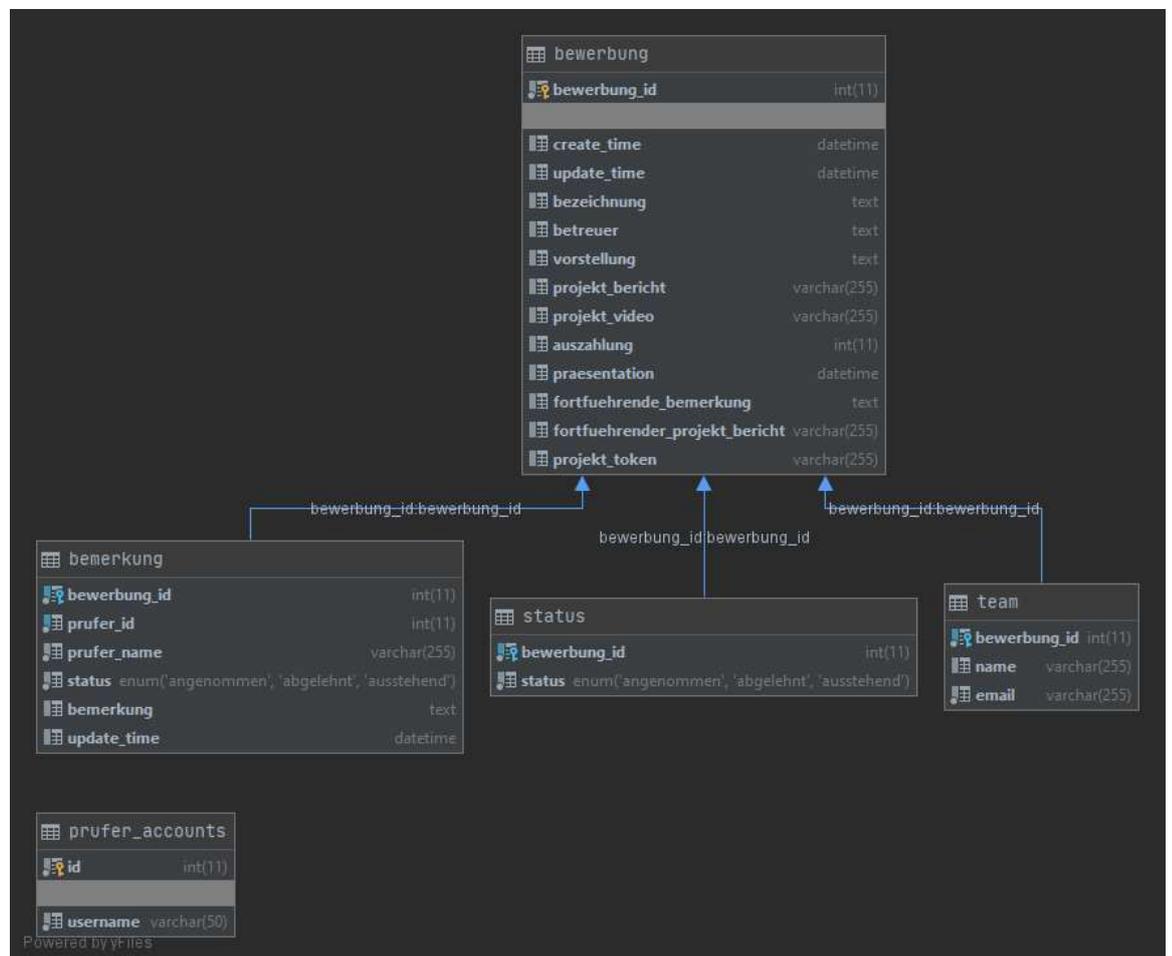


Abb. 23: Database Schema

Es sind fünf Tabellen vorhanden, wobei die Daten der *pruefer\_accounts* Tabelle lediglich als «Variablen» zur Generierung der Einträge in der *bemerkung* Tabelle benötigt werden.

### Überlegungen

Die oben präsentierte Konstellation erlaubt eine klare Abtrennung der Prüfer Interaktionen auf den Daten. Die Bemerkungen und Entscheide der Prüfer werden in der *bemerkung* Tabelle gehalten. Der Status einer Anfrage wird in einer separaten *status* Tabelle aufbewahrt, welche durch einen Trigger auf der *bemerkung* Tabelle bei vollendetem Entscheidungsprozess automatisch angepasst wird.

Die *team* Tabelle enthält die Daten des Anfragers. Diese Tabelle hätte in der *bewerbung* Tabelle integriert werden können. Die Idee einer separaten Haltung ist, dass mehrere Anfrager definiert werden können. Dieses Feature wird jedoch in der Anwendung nicht weiterverfolgt. Die eigenständige Tabelle wird trotzdem so belassen.

Die Generierung Skripte sind im Anhang beigelegt.

**Anfragedaten**

Die *bewerbung* Tabelle enthält alle mit einer Anfrage assoziierten Daten.

Kolonne	Bezeichnung
<b>bewerbung_id</b>	Primärer Schlüssel
<b>create_time</b>	Erstellungszeit
<b>update_time</b>	Aktualisierungszeit
<b>bezeichnung</b>	Anfragebezeichnung / Titel
<b>betreuer</b>	Anfragebetreuer
<b>vorstellung</b>	Anfragebeschreibung
<b>projekt_bericht</b>	PDF-Speicherort
<b>projekt_video</b>	Video-Speicherort
<b>auszahlung</b>	Bisherige Auszahlung (0/3000/5000)
<b>praesentation</b>	Präsentationsdatum
<b>fortfuehrende_bemerkung</b>	Fortführende Anfrage Bemerkung
<b>fortfuehrender_projekt_bericht</b>	PDF-Speicherort der fortführenden Anfrage
<b>projekt_token</b>	Projekt Token, Zugangscode zu laufender Anfrage

Tabelle 9: Anfrage Daten

Neben den Daten werden zwei Triggers auf dieser Tabelle definiert, welche einen mehrere Einträge in die *status* sowie in die *bemerkung* Tabelle einfügen.

<b>Trigger Bezeichnung</b>	init_status
<b>Auslösende Tabelle</b>	<i>bewerbung</i>
<b>Auslöser Typ</b>	Insert
<b>Auslöser Zeitpunkt</b>	After
<b>Zweck</b>	Fügt den default Status «Ausstehen» für die eingereichte Anfrage in die <i>status</i> Tabelle ein.
<pre> create trigger init_status after insert on bewerbung for each row insert into status (bewerbung_id) VALUES (NEW.bewerbung_id); </pre>	

Abb. 24: init\_status Trigger

<b>Trigger Bezeichnung</b>	setup_bemerkung
<b>Auslösende Tabelle</b>	<i>bewerbung</i>
<b>Beteiligte Tabelle</b>	<i>prufer_accounts</i>
<b>Auslöser Typ</b>	Insert
<b>Auslöser Zeitpunkt</b>	After
<b>Aktion</b>	Fügt neue Bemerkungen in die Tabelle <i>bemerkung</i> anhand der vorhandenen Einträge in der <i>prufer_accounts</i> Tabelle ein.

```

create trigger setup_bemerkung
  after insert
  on bewerbung
  for each row
  insert into
  bemerkung (bewerbung_id, prufer_id, prufer_name)
select
  NEW.bewerbung_id as bewerbung_id,
  id,
  username
from
  prufer_accounts;

```

Abb. 25: setup\_bemerkung Trigger

**Prüfer Bemerkungen**

Die *bemerkung* Tabelle enthält alle mit einer Anfrage assoziierten Daten, welche durch die Prüfer gesetzt werden.

Kolonne	Bezeichnung
<b>bewerbung_id</b>	Teil des zusammengesetzten Schlüssels Fremdschlüssel zur <i>bewerbung</i> Tabelle
<b>prufer_id</b>	Teil des zusammengesetzten Schlüssels Prüferidentifikation
<b>prufer_name</b>	Prüfer Benutzername
<b>status</b>	Der vom Prüfer gesetzten Status
<b>bemerkung</b>	Die vom Prüfer mitgegebene Bemerkung
<b>update_time</b>	Aktualisierungszeit

Tabelle 10: Bemerkung Daten

Die Tabelle *bemerkung* enthält eine der wichtigsten Funktionen der gesamten Anwendung. Diese Funktion kümmert sich um die Aktualisierung des Anfragestatus, sobald die Entscheidungsfindung abgeschlossen ist.

<b>Trigger Bezeichnung</b>	compute_status_update
<b>Auslösende Tabelle</b>	bemerkung
<b>Beteiligte Tabelle</b>	status
<b>Auslöser Typ</b>	Update
<b>Auslöser Zeitpunkt</b>	After
<b>Aktion</b>	Sobald es eine einheitliche Entscheidung der Prüfer gibt, also fünf Annahmen oder fünf Ablehnungen, aktualisiert dieser Trigger den Status dementsprechend.

```

create trigger compute_status_update
  after update
  on bemerkung
  for each row
Begin if (
  (
    select
      count(*)
    from
      db.bemerkung
    where
      status like 'Angenommen'
      and db.bemerkung.bewerbung_id = NEW.bewerbung_id
    ) >= 5
  ) Then
update
  db.status
set
  status = 'Angenommen'
where
  db.status.bewerbung_id = NEW.bewerbung_id;

elseif (
  (
    select
      count(*)
    from
      db.bemerkung
    where
      status like 'Abgelehnt'
      and db.bemerkung.bewerbung_id = NEW.bewerbung_id
    ) >= 5
  ) then
update
  db.status
set
  status = 'Abgelehnt'
where
  db.status.bewerbung_id = NEW.bewerbung_id;

else
update
  db.status
set
  status = 'Ausstehend'
where
  db.status.bewerbung_id = NEW.bewerbung_id;

end if;
end;

```

Abb. 26: compute\_status\_update Trigger

Das Setzen des Anfragestatus benötigt die Ablehnung oder Zustimmung aller Prüfer. Diese Logik wird durch den Vorstand des Innovation Network definiert und kann bei Bedarf zum Beispiel zu einem Mehrheitsentscheid umgeschrieben werden.

**Status**

Die *status* Tabelle enthält den Anfragestatus und wird bei Eingang einer neuen Anfrage durch den Trigger der *bewerbung* Tabelle initial gesetzt und später durch den Trigger auf der *bemerkung* Tabelle aktualisiert.

Kolonne	Bezeichnung
<b>bewerbung_id</b>	Primärere und Fremdschlüssel
<b>status</b>	Status der Anfrage (Ausstehend/Angenommen/Abgelehnt)

**Team**

Die *team* Tabelle enthält die notwendigen Informationen des Anfragers.

Hier könnte man eine Erweiterung der Anwendung ansetzen, indem man die Aufnahme aller Teammitglieder eines Anfrageprojekts erlaubt.

Kolonne	Bezeichnung
<b>bewerbung_id</b>	Primärere und Fremdschlüssel
<b>name</b>	Name des Anfragers
<b>email</b>	E-Mail des Anfragers

**Prüfer Accounts**

Die Daten der *pruefer\_accounts* Tabelle werden lediglich als «Variablen» zur Generierung der Einträge in der *bemerkung* Tabelle benötigt. Sie kommen im oben beschriebenen *setup\_bemerkung* Trigger der *bewerbung* Tabelle vor und werden nur nach Einreichung einer neuen Anfrage benötigt.

Kolonne	Bezeichnung
<b>id</b>	Primärer Schlüssel
<b>username</b>	Prüfer Benutzername

### 6.3.2 Auth

#### Auth

Die Auth Datenbank besteht aus den jeweiligen Logins. Die Passwörter sind eine Kombination aus dem «hashed» Passwort und einer Salt Kodierungen, damit keine Plain Text Passwörter gespeichert werden.

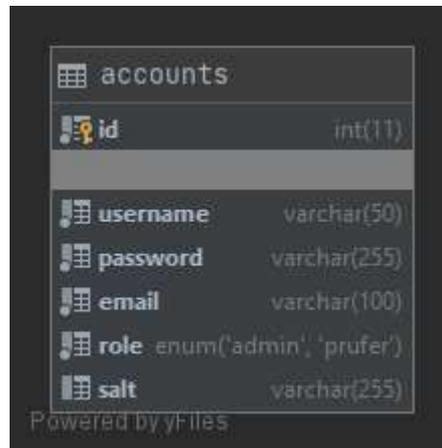


Abb. 27: Auth Schema

#### accounts

Die *accounts* Tabelle enthält die Login Daten des Admins und der fünf Prüfer.

Kolonne	Bezeichnung
<b>id</b>	Primärer Schlüssel
<b>username</b>	Login Benutzername
<b>password</b>	«Gehashtes» Passwort
<b>email</b>	E-Mail des Benutzers
<b>role</b>	Rolle des Benutzer (Admin/Prüfer)
<b>salt</b>	Salt, welcher zur Validierung des Passwortes benötigt wird.

Tabelle 11: accounts Tabelle

Beim Login wird das *password* und der *salt* an die Node Anwendung zurückgeliefert. Dort wird durch eine Umkehrfunktion das eingegebene Passwort mit den beiden Kodierungen verglichen und entweder bestätigt oder verworfen.

---

## 6.4 Verifikation

---

**Übersicht** Allgemein ist die Zusammensetzung der Views durch die Handlebars Vorlagen schwer zu testen. Jedoch kann dieser Umstand etwas umgangen werden, da durch die Custom Error Propagation sowie die Infopropagation alle wichtigen Informationen übermittelt werden. Diese Objekte sind jeweils im *returnObject* beim Datenzugriff enthalten. Somit können die Tests auf den vorhandenen Abfrage Funktionen aufgebaut werden.

In der Elaborationsphase wurden vier Testkategorien definiert. Unit, Integration End-to-End und User Acceptance Tests. Im Endeffekt wurde jedoch auf separate Unit Tests verzichtet, da jeder einzelne Integration Test eine Iteration von Insert, Update, Read und Delete durchführt. Die übrigen Tests werden nach den Überlegungen aus der Elaboration Phase aufgebaut.

### 6.4.1 Unit / Integration

---

**Administrative Interaktionen** *adminInteractionTests* überprüft die Funktion der zwei Admin Interaktionen zur Änderung der Auszahlung und das Setzen des Präsentationsdatum.

**Prüfer Interaktionen** *pruferInteractionTests* überprüft die Funktion der Prüfer Interaktionen auf einer Bewerbung.

**Benutzer/Anfrager Interaktionen** *userInteractionTests* überprüft die Funktion der Anfrager Interaktion.

**Keine Interaktionen** *viewOnlyTests* überprüft die Funktionen aller Nutzer, welche keine Interaktion benötigen.

**Ablauf**

Für die oben gelisteten Files werden folgende Tests durchlaufen:

<b>File</b>	<b>Tests</b>	<b>Use Cases</b>
<b>adminInteractionTests</b>	<ol style="list-style-type: none"> <li>1. Bewerbung erstellen</li> <li>2. Auszahlung aktualisieren</li> <li>3. Präsentation aktualisieren</li> <li>4. Bewerbung löschen</li> </ol>	AktualisiereAnfrage
<b>pruferInteractionTests</b>	<ol style="list-style-type: none"> <li>1. Bewerbung erstellen</li> <li>2. Status und Bemerkung aktualisieren</li> <li>3. Bemerkung aktualisieren</li> <li>4. Status aktualisieren</li> <li>5. Neue Bemerkungen überprüfen</li> <li>6. Bewerbung löschen</li> </ol>	BehandleAnfrage EinsehenAnfrage
<b>userInteractionTests</b>	<ol style="list-style-type: none"> <li>1. Bewerbung erstellen</li> <li>2. Bewerbung per Token anzeigen</li> <li>3. Bewerbung per invalidem Token anzeigen</li> <li>4. Fortführende Bewerbung erstellen</li> <li>5. Invalide fortführende Bewerbung erstellen</li> <li>6. Bewerbung löschen</li> </ol>	StelleAnfrage StelleFortführendeAnfrage EinsehenAnfrage
<b>viewOnlyTests</b>	<ol style="list-style-type: none"> <li>1. Bewerbung erstellen</li> <li>2. Alle Anfragen anzeigen</li> <li>3. Alle angenommenen Anfragen anzeigen</li> <li>4. Individuelle Interaktion Anfrage anzeigen</li> <li>5. Individuelle nicht zugängliche Anfrage anzeigen</li> <li>6. Individuelle nicht zugängliche Anfrage Projekt Video URI abfragen</li> <li>7. Individuelle Anfrage nach Statusänderung anzeigen</li> <li>8. Bemerkungen der individuellen Anfrage anzeigen</li> <li>9. Individuelle Anfrage Projekt Video URI abfragen</li> <li>10. Bewerbung löschen</li> </ol>	EinsehenAnfrage EinsehenAllerAnfragen

**Tabelle 12: Integrationstestbeschreibung**

## Kontroller Tests

Überprüft die Funktionen der Controller. Es werden zur Funktion passende Mocks für das *Request*, *Response* und das *Next* Objekt erstellt.

Beispiel für die *uploadNeueAnfrage* Funktion des *interaktionController.js* Datei:

```
const mockResponse = {
  render: jest.fn(),
}

const mockNext = jest.fn()

const mockRequest = {
  body: {
    projektbezeichnung: null,
    betreuer: 'Jest',
    team: 'Jest Team',
    projektvorstellung:
      'Das ist ein Test für das Hinzufügen einer neuen Anfrage.',
    projekt_video: '/uploads/jest_video.mp4',
    email: 'jest@test.com',
  },
  files: {
    projektbericht: [{ path: '/uploads/jest_bericht.pdf' }],
    projektvideo: [{ path: '/uploads/jest_video.mp4' }],
  },
}
```

Abb. 28: Jest Mock Objekte

Eine Funktion wird mit diesen drei Mocks aufgerufen:

```
await interactionController.uploadNeueAnfrage(
  mockRequest,
  mockResponse,
  mockNext
)
```

Abb. 29: Jest Mock Funktionsaufruf

Anschließend werden separate Datenbankabfragen gemacht, um die Resultate der aufgerufenen Funktion zu überprüfen:

```
const connection = await ppool
  await connection
    .query(
      "Select bewerbung_id, projekt_token from bewerbung where betreuer = 'Jest'"
    )
    .then((res) => {
      id = res[0].bewerbung_id
      token = res[0].projekt_token
    })

expect(id).not.toBeNull()
expect(token).not.toBeNull()
```

Abb. 30: Jest Datenüberprüfung

## Code Coverage



Abb. 31: Jest Code Coverage Übersicht

Die Code Coverage zeigt, wie gründlich der geschriebene Code getestet wird.

Auffallend ist, dass bei den Kontrollern ein geringer Wert beim *Branch* Indikator erreicht wird. Dies ist auf das Custom Error Handlin zurückzuführen, da bei den Tests gewisse Branches, die in geworfenen Fehlern enden, nicht erreicht werden.

Es stehen bei Bedarf genauere Informationen im Abgabeordner unter *ba-innonet/no-dejs/coverage* zur Verfügung.

## Ausgelassene Dateien

Folgende Kategorien werden nicht automatisch getestet oder die Testresultate sind nicht speziell aufgezeichnet:

- Views und Handlebar Helpers
- Routes
- Mail

Die Views sind allgemein schwierig automatisch zu testen und die Routes sind trivial.

Der Mail Service wurde mit der Hilfe von einem Testing SMTP E-Mail Account *ethereal.email* auf Form, Inhalt und Status der Übermittlung überprüft und für befriedigend empfunden.

## 6.4.2 End-to-End

### Übersicht

Der End-to-End Test beschreibt die Behandlung einer Bewerbung durch alle beteiligten Nutzer und Systeme.

Dieser Test wurde vor dem User Acceptance Test mehrmals durch den Verfasser dieser Arbeit, sowie weiteren Testpersonen durchgeführt.

Der End-to-End Test beinhaltet folgende Schritte (Weiterleitungen und Ansichten werden nicht genannt):

Schritt	Erwartungen	Status
Neue Anfrage erstellen	<ul style="list-style-type: none"> <li>➤ Übertragung der Formulardaten</li> <li>➤ Übergabe des Projekt Tokens auf der Website</li> <li>➤ Notifikation an Prüfer, Admin und Anfrager mit den richtigen Informationen</li> </ul>	Erfüllt
Prüfer Interaktion	<ul style="list-style-type: none"> <li>➤ Bewerbung im Status «<i>Angenommen</i>»               <ul style="list-style-type: none"> <li>a. Fortführende Anfrage Option offen</li> </ul> </li> <li>➤ Notifikation an Prüfer, Admin und Anfrager bezüglich Statuswechsel               <ul style="list-style-type: none"> <li>a. Admin erhält E-Mail-Adresse des Anfragers in der Notifikation</li> </ul> </li> </ul>	Erfüllt
Fortführende Anfrage erstellen	<ul style="list-style-type: none"> <li>➤ Notifikation an Prüfer, Admin und Anfrager bezüglich fortführender Bewerbung</li> </ul>	Erfüllt
Admin Interaktion	<ul style="list-style-type: none"> <li>➤ Auszahlung und Präsentationsdatum gesetzt und einsehbar</li> </ul>	Erfüllt

Tabelle 13: End-to-End Test Beschreibung

## 6.4.3 User Acceptance

### Übersicht

Der User Acceptance Test läuft zwischen dem 28.12.2021 und dem 09.01.2022 auf einem Server des Instituts für Software der OST Rapperswil.

Es sind alle Logins der Vorstandsmitglieder des Innovation Networks, sowie ein Admin Account vorhanden.

Der Test soll dem Auftraggeber als Proof-of-Concept eine Einsicht in die geleistete Arbeit ermöglichen. Anschliessend werden die Befunde und Rückmeldungen im Anhang aufgenommen und vor dem potenziellen Gebrauch aufbereitet.

### Skripte

Es werden Skripte für den Prüfer und den Admin Benutzer zur Verfügung gestellt und sollen als Guideline dienen.

Das Skript sowie sämtliche Rückmeldungen sind im Anhang bereitgestellt.

## 7 Transition Phase

---

### Übersicht

Es wird zwischen der Testumgebung und der Liveumgebung unterschieden.

Der Zugang zur Liveumgebung ist eingeschränkt und von einer dritten Partei betreut. Aus diesem Grund wurde die Anwendung auch auf der Basis von Docker aufgebaut und soll im Falle einer Liveschaltung auf die genaueren Konfigurationen der Liveumgebung angepasst werden.

Der modulare Aufbau der Anwendung und die gute Handhabung der Konfigurationen durch die Docker Umgebung, erlauben eine fließende Übertragung auf die Liveumgebung.

### 7.1 Docker und Testumgebung

---

#### Übersicht

Die verschiedenen Komponenten der Arbeit laufen in separaten Docker Container und teilen sich ein gemeinsames Netzwerk. Dies erlaubt eine schnelle Entwicklung, da viele der Konfigurationen, wie beispielsweise die Netzwerkkommunikation zwischen den Komponenten und die Portzuteilung bereits durch *docker-compose* übernommen werden.

Das Deployment fällt dementsprechend simpel aus, falls eine Infrastruktur wie Portainer (siehe <https://www.portainer.io/>) zur Verfügung steht. Dieser Umstand wird durch das einfache Aufschalten der Anwendung in der Testumgebung gezeigt.

Alle wichtigen Konfigurationen können aus den Dateien von Docker bzw. den eigenständigen Konfigurationsdateien der einzelnen Komponenten entnommen werden.

#### Paketierung

Die Paketierung der Komponenten greift auf die Image Generierung von Docker zurück. Die Images der verschiedenen Anwendungskomponenten werden direkt über die Gitlab CD/CI gebaut und stehen anschliessend im Container Registry zur Verfügung. Dabei werden jeweils nur die komponentenspezifischen Dateien einbezogen und erlauben eine optimale Abtrennung der Komponenten. Dies erleichtert die Pflege und die Erweiterbarkeit, da diese Zusammenstellung der Arbeit auf einzelnen Komponenten aufbaut und somit deren separater Aufschaltung ermöglicht.

#### Rollout

Im Rahmen der Testumgebung kann ein kontinuierliches Rollout angestrebt werden. Änderungen werden beim «Merge» auf die Masterbranch direkt in den jeweiligen Images abgebildet und müssen von Portainer nur noch frisch «gepullt» werden, um in Effekt zu treten.

Dieses Verfahren ist im *.gitlab-ci.yml* definiert und kann in den Projektdateien eingesehen werden.

#### Datenhandhabung

Konsistente Daten wie die Projektberichte und die Projektvideos werden direkt via Volume in den jeweiligen Container aufbewahrt und können bei einem Neustart direkt zurück in die Anwendung gespeist werden.

## 7.2 Liveumgebung

---

<b>Übersicht</b>	<p>Es sind nicht viele Informationen über den Server bekannt, auf der die momentane Webseite des Innovation Networks gehostet ist. Die Idee dieser Arbeit ist dennoch die bestehende Infrastruktur des Webservers nicht zu verändern.</p> <p>Es wird keine genaue Anleitung zu Aufschaltung des Projekts zur Verfügung gestellt, denn die verschiedenen Komponenten sollten in der individuellen Aufschaltung keine Probleme bereiten. Das Wichtige ist, dass die Services korrekte Weiterleitungen und Abfragen zueinander stellen können. Daher wird der grösste Aufwand bei der korrekten Konfiguration der Schnittstellen erwartet.</p>
<b>Paketierung</b>	<p>Die verschiedenen Komponenten sind bereits durch die Struktur von Docker aufbereitet. Es wird jedoch die richtige Konfiguration der einzelnen Services analog zu deren Docker Konfigurationen vorausgesetzt.</p>
<b>Nginx</b>	<p>Der Reverse Proxy hat die Routenweiterleitung bereits in der Konfiguration vermerkt. Die Konfiguration, dass der nginx alle Abfragen auf den Server als erstes erhält, muss jedoch noch manuell eingestellt werden.</p> <p>Anschliessend werden die Anfragen zu den korrespondierenden Anwendungen weitergeleitet.</p>
<b>Node</b>	<p>Das Aufschalten der Node Anwendung lässt sich durch folgende Schritte zusammenfassen:</p> <ol style="list-style-type: none"><li>1. Lade das Verzeichnis der Anwendung auf den Server</li><li>2. Installiere alle Node Abhängigkeiten</li><li>3. Starte den Service</li></ol> <p>Die einzigen manuellen Konfigurationen sind die Verbindungen zu den Datenbanken. Diese müssen der Deployment Konfiguration der mysql Services angepasst werden und sind in der <code>.env</code> Datei der Node Anwendung zu finden.</p>
<b>Datenbanken</b>	<p>Die Datenbankservices können auf dem Server via der Eingabeaufforderung heruntergeladen und installiert werden. Dabei muss nur auf die Wahl des Benutzers und des Passwortes geachtet werden, damit die Anfragen der Node Anwendung authentifiziert werden können.</p> <p>Die Datenbanken, Tabellen und Triggers werden via dem zur Verfügung gestellten Init-Skript erstellt, welches manuell in die mysql-Eingabe kopiert und ausgeführt wird.</p> <p>Die Verbindung zur Datenbank kann auf dem Server über den <code>localhost</code> abgehandelt werden, da die Node Anwendung auf demselben Server läuft. Somit fallen für den Datenbank Service keine weiteren Konfigurationen bezüglich des Netzwerks an.</p>

## 7.3 Zusammenfassung

**Checkliste** Anbei eine kurze Zusammenfassung der vorzunehmenden Konfigurationen:

Service	Konfigurationen
Ngnix	<ul style="list-style-type: none"><li>➤ Abfangen aller Anfragen auf den Server</li><li>➤ Korrekte Weiterleitung (Ports)</li></ul>
Node	<ul style="list-style-type: none"><li>➤ Anpassung der .env Datei bezüglich der Datenbank Logins</li><li>➤ Kopieren des Verzeichnisses</li><li>➤ Laden der Abhängigkeiten</li><li>➤ Anpassen der Ports</li></ul>
MySQL	<ul style="list-style-type: none"><li>➤ Herunterladen des Services analog zum Dockerfile</li><li>➤ Anpassung der Logindaten</li><li>➤ Anpassung des Netzwerkzugangs und Ports</li></ul>

Tabelle 14: Konfiguration Zusammenfassung

## 8 Fazit

---

### Ergebnisse

Der Umfang der Arbeit wurde bewusst locker gesteckt. Diese Tatsache ermöglichte diese Arbeit als eine Art «Grüne Wiese» Projekt anzugehen. Allgemeine Erwartungen wurden nur durch die allgemeine Prozessbeschreibung einer Förderungsbewerbung beim Innovation Network vorgegeben. Die Aufgabe bestand darin, diesen Prozess in einer Anwendung abzubilden und mit möglichst wenig zusätzlichem Aufwand den bestehenden Onlineauftritt des Innovation Networks zu erweitern.

Dieses Vorhaben konnte durch das Erstellen einer Full-Stack Anwendung mit Frontend, Backend und Datenspeicherung erfüllt werden. Um die Erweiterung möglichst ohne weiteren Aufwand einzubinden, wurde ein Reverse Proxy zur Anfrageunterscheidung eingeplant. Dieser Proxy kümmert sich um die Unterscheidung zwischen dem Routing zur bisherigen statischen Lösung und der neu konzipierten Full-Stack Anwendung.

Schlussendlich entstand eine Webapplikation, welche einen gut nachvollziehbaren Prozess möglichst simpel und ohne unnötige zusätzliche Features in einer Konstellation von mehreren Anwendungen abbildet.

### Probleme

Das Umschalten der Testumgebung hat zeitlich etwas Mühe bereitet. Der User Acceptance Tests mit den echten Akteuren des Innovation Networks konnte wegen privaten und arbeitsbedingten Gründen erst in den Weihnachtsferien losgetreten werden. Dementsprechend wurden die designtechnischen Mängel aus Sicht der eigentlichen Benutzer der Anwendung zu spät gesammelt.

Zum Beispiel bietet die Möglichkeiten des Verfassens eines Kommentares auf einer Bewerbung nach der Meinung des Verfassers zu wenig. Neue Kommentare überschreiben die alten Kommentare und es gibt keine Interaktionsmöglichkeiten, wie sich das ein Benutzer vielleicht von anderen Anwendungen gewöhnt ist. Es wäre ein nützliches Feature, wenn Prüfer sich gegenseitig in Kommentaren referenzieren oder auf diese antworten könnten. Diese einfache, aber wertvolle Erweiterung war jedoch im Umfang dieser Arbeit nicht lösbar, da sie erstens konzeptuell vor allem in der Datenhandhabung bzw. der Historie zu aufwändig gewesen wäre und zweitens die Idee etwas zu spät aufkam.

### Offene Punkte

Wegen der Ungewissheit bezüglich des Aufbaus des endgültigen Servers könnte bei der Liveschaltung zusätzliche Arbeit anfallen. Dieser Umstand ist bei Erweiterungen grundsätzlich normal, doch im Falle dieser Arbeit ist der zusätzliche Aufwand schwer einzuschätzen. Unter Umständen muss der Gebrauch eines besser ausgerüsteten Servers in Erwägung gezogen werden. Dies ist vor allem der Speicherung der Projektvideos zuzuschreiben, da diese am meisten Speichervolumen benötigen.

Ein weiterer Punkt bezüglich der Speicherung ist die Replikation der Datensätze. Die Arbeit befasst sich nur wenig mit der Möglichkeit eines Datenbackups. In der Testumgebung hat ein Neustart bzw. das Pulling der neusten Version des MySQL Images den Verlust der bisherigen Daten zur Folge. Dies ist im Falle der Testumgebung nicht weiter schlimm, doch sollte eine Container Lösung in der Liveumgebung anbieten, müsste eine konsistente Datenorganisation bei Container Updates sichergestellt werden.

So wie die Liveumgebung momentan angesetzt ist, gibt es dieses Problem mit dem Datenverlust bei Updates nicht. Jedoch sind die Daten nur einmalig in einer Datenbank abgelegt. Diesem Umstand kann entgegengewirkt werden, indem bei der Betrachtung einer Bewerbung darauf geachtet wird, dass die Projektberichte, sowie die Projektvideos heruntergeladen und lokal auf einem Computer des Innovation Networks platziert werden.

Dies ist nicht die eleganteste Lösung, dennoch aus Sicht des Verfassers zufriedenstellend, da dies auch der bisherigen Lösung entspricht.

**Erfahrungen**

Die losen Erwartungen und Vorschriften bezüglich des Aufbaus der finalen Anwendung waren ein grosser Vorteil. Das Vertrauen des Betreuers und des Auftraggebers erlaubten eine eigenständige Analyse, Interpretation und Aufbereitung der Aufgabenstellung und Vorgaben. Lösungen für die individuellen Bedürfnisse der verschiedenen Parteien im Bewerbungsprozess zu konzipieren, war sehr abwechslungs- und lehrreich. Es war interessant, einen Prozess aus der Sicht von mehreren Benutzern zu definieren, zu verstehen und umzusetzen.

Der Kontakt mit dem Betreuer und den echten Akteuren des Innovation Networks waren ebenfalls eine grosse Hilfe und eine persönliche Bereicherung. Eine weitere Meinung und die Möglichkeit, gewisse Lösungen zu diskutieren, halfen in der Identifizierung wichtiger Erwartungen an die Anwendung. Als alleiniger Verfasser dieser Arbeit waren dies Bemerkungen und Inputs höchst willkommen.

**Reflektion und Aus-sichten**

Ich konnte diese Arbeit als Einzelperson antreten. Dies kam mir als Teilzeitstudent sehr entgegen, da die freie Arbeitseinteilung für mich einer der wichtigsten Punkte darstellte. Gegen Ende des Semesters hat sich dieser Umstand jedoch im negativen Sinn bemerkbar gemacht. Ich habe während der Planung die Festtage und die Jahresabschlüsse ausser Acht gelassen und dies resultierte in der etwas verschobenen Behandlung der User Acceptance Tests. Ich habe daraus gelernt, dass bei einem zukünftigen Projekt die Festtage und Abgabetermine im Arbeitsumfeld mit eingeplant werden müssen, um dem zeitlichen Druck aus dem Weg gehen zu können.

Ansonsten bin ich mit der Umsetzung zufrieden und die Arbeit hat mir sehr viel Freude bereitet. Ich habe die Motivation nie verloren und war dankbar für das Vertrauen, dass Prof. Politze in mich gesetzt hat. Ich arbeite als Person gerne selbständig und die Betreuung entsprach genau meinen Vorstellungen. Es gefiel mir das Projekt ohne grössere Vorschriften anzugehen und nach meinem besten Gewissen und besten Fähigkeiten zu arbeiten.

Es wäre in meinem Sinne, wenn sich das Innovation Network meiner Arbeit annehmen würde und überprüft, ob meine Leistung zu einer möglichen Erweiterung ihres Onlineauftritts vollendet werden kann.

## 9 Literatur und Quellenverzeichnis

---

### Herkunft der Vorlage

Das Dokument wurde auf der Basis einer Vorlage für Technische Berichte erstellt. Die Vorlage ist ein Element des „Werkzeugkastens Technische Berichte“ der Hochschule für Technik Rapperswil. Sie orientiert sich an Prinzipien des Strukturierten Schreibens.

### Quellen

Der Grossteil der Arbeit beruht auf dem vermittelten Wissen der Dozenten und Dozentinnen der OST Standort Rapperswil. Dennoch werden nachfolgend einige Webseiten gelistet, die bei der Lösungsfindung wichtige Einblicke verschaffen konnten. Diese Quellen sind nicht direkt im Bericht referenziert, da sie eher konzeptuelle Einsichten mit sich brachten.

François Romain, 14. Juli 2017: <https://francoisromain.medium.com/setup-node-js-apache-nginx-reverse-proxy-with-docker-1f5a5cb3e71e>

Learninjava, 7. Juli 2020: <https://www.learninjava.com/react-proxying-api-backend-apache-nginx-tomcat/>

## 10 Erklärung zur Urheberschaft

---

**Erklärung** Ich erkläre hiermit, dass ich die vorliegende Arbeit ohne Hilfe Dritter angefertigt habe. Ich habe nur die Hilfsmittel benutzt, die ich angegeben habe. Gedanken, die ich aus fremden Quellen direkt oder indirekt übernommen habe, sind kenntlich gemacht. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

**Ort/Datum** Rapperswil, 14. Januar 2022

**Unterschrift** 

Etienne Baumgartner

---

## Anhang

---

### I Auftrag

**Titel** Webapplikation zur Prozessdigitalisierung

**Themengebiet** Application Design

**Betreuer** Daniel Patrick Politze

**Beschreibung** Der Verein "Innovationnetwork.ch" bildet eine Plattform für den Austausch von Wissen und Methoden zwischen Industrie und Forschung. Der Verein fördert innovative und produktorientierte Studentenprojekte und unterstützt die Prototypenentwicklung vielversprechender Projekte mit einem Betrag bis zu insgesamt 5'000 CHF. Für die Bewerbung um die Förderung ist ein Antrag einzureichen - heute jedoch in beliebiger Form. Im Rahmen der Arbeit wird nun dieser Bewerbungsprozess digitalisiert und ist dann neu auch direkt über die Webseite des Vereins startbar.

### II Projektplan

#### Methodik & Vorgehen

##### Rational Unified Process

**RUP** Die Bearbeitung der Bachelorarbeit orientiert sich am *Rational Unified Process (RUP)*. Dabei wird eine klare Strukturierung in 4 Phasen vorgegeben.

**Inception** Analyse des Problems und Aufstellung des Konzepts, sowie allgemeine Planung

**Elaboration** Genauere Analyse und granulare Ausarbeitung des Konzepts, sowie Modellierung.

**Construction** Realisierung des Konzepts

**Transition** Abschluss, Paketierung, sowie Rollout.

**Planung & Dokumentation** Zusätzlich zu diesen vier Phasen werden dieser Arbeit zwei weitere Phasen hinzugefügt. Eine Planungsphase wird am Anfang separat neben der Inception Phase eingeräumt, sowie eine Dokumentationsphase am Ende.

**Planung & Setup** Umfasst die allgemeine Planung rund um das Projekt, fließt in die Inception Phase mit ein

**Dokumentation** Umfasst die Ausarbeitung der Dokumentation.

## Sprints

**Sprintdauer** Die Bearbeitung der Themen in den verschiedenen Phasen wird in einwöchigen Sprints durchgeführt.

**Phasen** Nicht jede Phase des Projekts fällt in dieses agile Verfahren. Die Sprints werden vor allem als Verwaltungstechnik angewendet, um den Aufwand des RUP besser einsehbar zu gestalten und finden ihre Anwendung überwiegend in den vier Phasen des RUP.

## Zeit Management

### Grobgranulare Betrachtung

#### Zeitstrahl

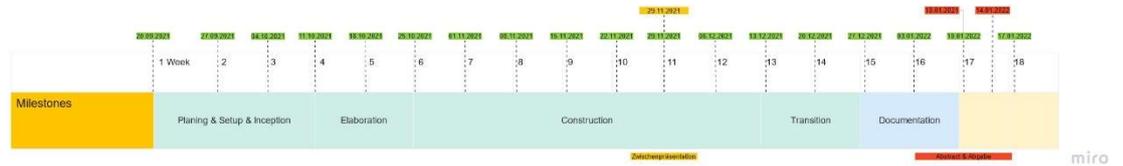


Abb. 32: Zeitstrahl

Obenstehend wird eine Zeitachse aufgeführt, welche die zeitliche Strukturierung der Arbeit visuell darstellt.

#### Aufspaltung

Untenstehend sind die verschiedenen Phasen nach Wochen und Daten aufgelistet.

Phase	Wochen (von-bis&mit)	Daten (von-bis)
Planung & Setup	1-2	20.09.2021-11.10.2021
Inception	2-3	27.09.2021-11.10.2021
Elaboration	4-5	11.10.2021-25.10.2021
Construction	6-12	25.10.2021-13.12.2021
Transition	13-14	13.12.2021-27.12.2021
Dokumentation	15-16	27.12.2021-10.01.2022

Tabelle 15: Phasenaufteilung

Hierbei gilt zu beachten, dass die ersten beiden Phasen in einander übergehen.

#### Milestones

Das Ende einer Phase wird zugleich als Projekt Milestones definiert. Dadurch können Arbeitspakete wohldefinierten Abgabeterminen zugeordnet werden.

**Stundenberechnung**

Im Bachelorstudiums an der OST Standort Rapperswil repräsentiert diese Arbeit 12 ETC. Mit der Zuordnung von 30 Stunden pro ETC und dem zeitlichen Umfang von ungefähr 16 Wochen, folgt daraus die optimale Aufteilung von 22.5 Stunden Aufwand pro Woche.

$$\frac{360 \text{ [Stunden]}}{16 \text{ [Wochen]}} = 22.5 \frac{\text{[Stunden]}}{\text{[Woche]}}$$

**Formel 1: Stundenberechnung**

Untenstehend werden die Stunden pro Projektphase aufgeteilt dargestellt. Die Planung & Setup sowie die Inception Phase wurden wegen ihrer Überschneidung auf eine Phase zusammengeführt.

Phase	Wochen (von-bis&mit)	Stunden pro Woche
Planung & Setup + Inception	1-3	67.5
Elaboration	4-5	45
Construction	6-12	157.5
Transition	13-14	45
Dokumentation	15-16	45

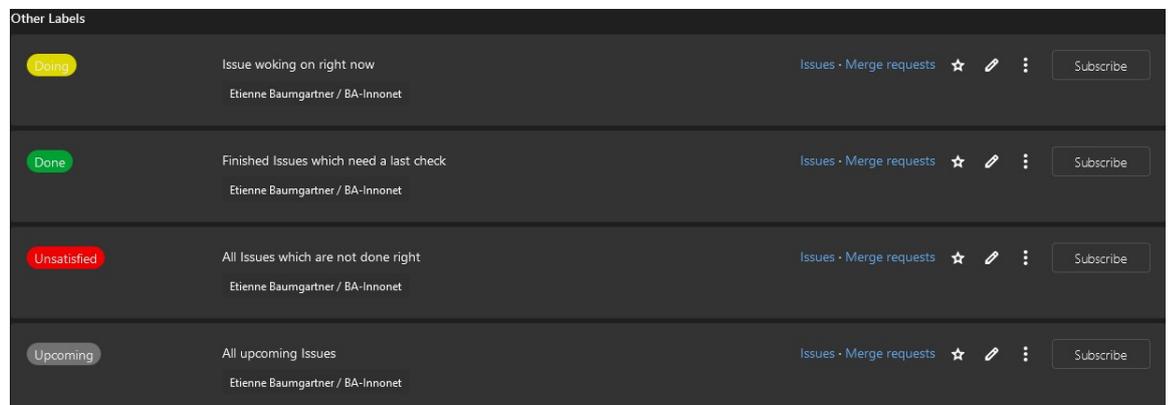
**Tabelle 16: Stündliche Phasenaufteilung**

**Feingranulare Betrachtung**

**Arbeitspakete**

Für die feinere Planung werden Arbeitspakete in Form von Gitlab Issues benutzt. Gitlab Issues sind für die zentrale Organisation genügend flexibel und ermöglichen das effiziente Zusammenführen von Arbeitsstunden sowie eine simple Form des Reporting. Auf diesen Paketen können vorzu Stunden gebucht werden, um den Zeitaufwand quantitativen nachvollziehen zu können.

Um die zeitgerechte Abarbeitung der Arbeitspakete gewährleisten zu können, wird vom Label Feature von Gitlab Gebrauch gemacht. Die Labels ermöglichen die Fortschrittskontrolle und eine Übersicht über die noch zu verrichtende, sowie die unbefriedigte Arbeit.



**Abb. 10.2: Label Definition**

Label	Bedeutung	Nächstes Label
<i>Upcoming</i>	Enthält alle in einem Milestone zu bearbeitenden Arbeitspakete, Synonym zu Backlog	<i>Doing</i>
<i>Doing</i>	Enthält die Arbeitspakete, die im momentanen Sprint bearbeitet werden	<i>Done</i> <i>Unsatisfied</i>
<i>Unsatisfied</i>	Enthält die zu überarbeitenden Arbeitspakete, die während einem Sprint nicht vollständig abgeschlossen wurden	<i>Done</i>
<i>Done</i>	Enthält die abgeschlossenen Arbeitspakete, die einer letzten Überprüfung standhalten müssen, bevor sie geschlossen werden	-

**Tabelle 17: Labelbezeichnungen**

Diese Konstellation hilft bei der Sicherstellung, dass die Features korrekt abgearbeitet und ihre vollumfängliche Funktionalität gewährleistet ist.

Neben den oben beschriebenen Labels können die Arbeitspakete im Status *Open* oder *Closed* sein. Dies ist eine Gitlab Bezeichnung für vollkommen neue Issues oder vollkommen abgeschlossene Issues.

Es ist zu beachten, dass die Arbeitspakete nicht immer vollständig im Voraus bekannt sind. Deshalb wird an dieser Stelle auf die Darstellung einer Zuteilung der Arbeitspakete auf die jeweiligen Milestones verzichtet. Die genaue Aufteilung kann im Projektbericht unter der Analyse des Zeitmanagements eingesehen werden.

#### Milestones

Jedes Arbeitspaket wird zur Aufwandskontrolle einem der Milestones am Ende jeder Phase zugeordnet. Während die Arbeitspakete den Aufwand eines umgesetzten Features messen, dienen die Milestones zur Kontrolle des allgemeinen Projektfortschritts. Sie erlauben eine gute Einsicht in den momentanen Zustand der Arbeit und ermöglichen eine schnelle Reaktion, falls eine Phase nicht wie geplant verläuft. Des Weiteren stellen die Milestones die aufsummierte, aufgewendete Zeit pro Phase zur Verfügung, welche am Ende der Arbeit in der Analyse des Zeitmanagement Aufschlüsse auf Defizite bei der Planung und Umsetzung liefern.

## Umgebung

#### Gitlab

Die Arbeit wird auf gitlab unter <https://gitlab.ost.ch/etienne.baumgartner/ba-innonet> geführt.

### III Zeitmanagement Analyse

#### Übersicht

In diesem Abschnitt wird eine genauere Betrachtung der Planungsumsetzung bereitgestellt. Dies beinhaltet die Zuordnung der Arbeitspakete zu den jeweiligen Milestones bzw. der verschiedenen Projektphasen, sowie die Darlegung des zeitlichen Aufwands pro Phase.

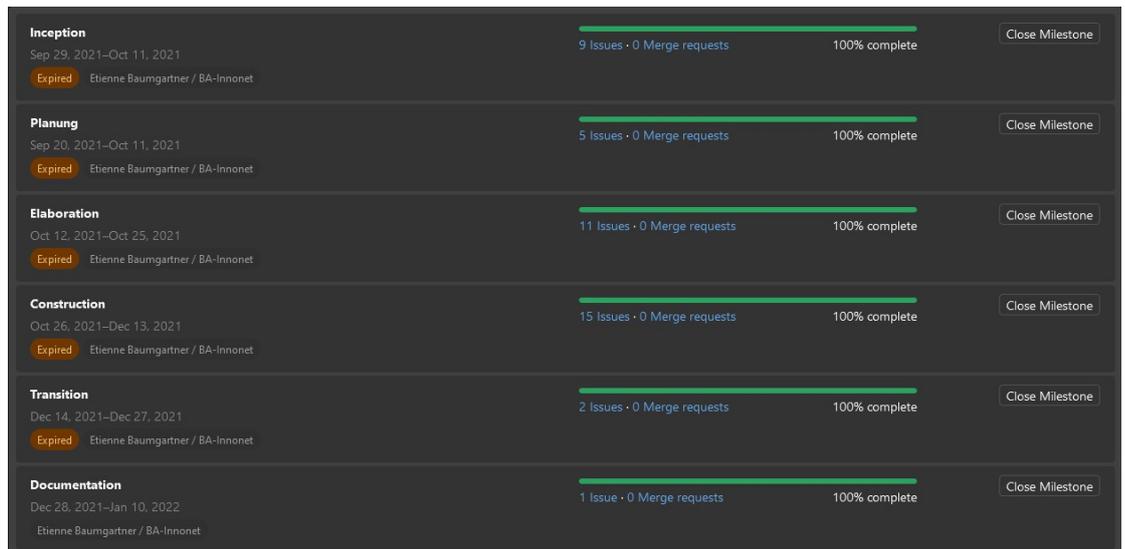
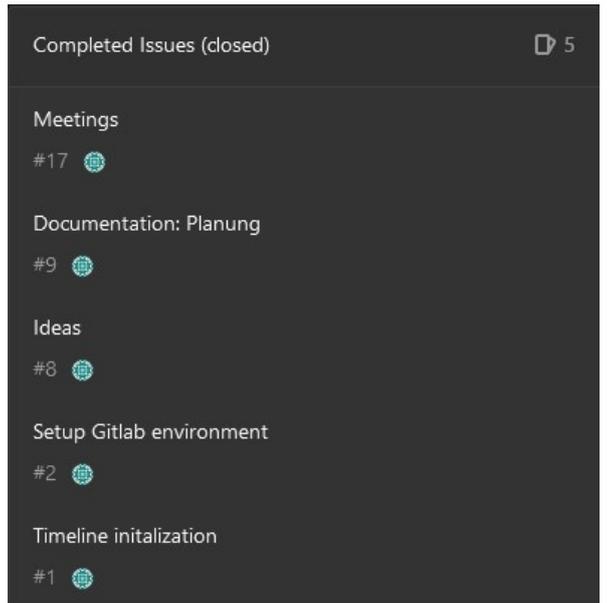


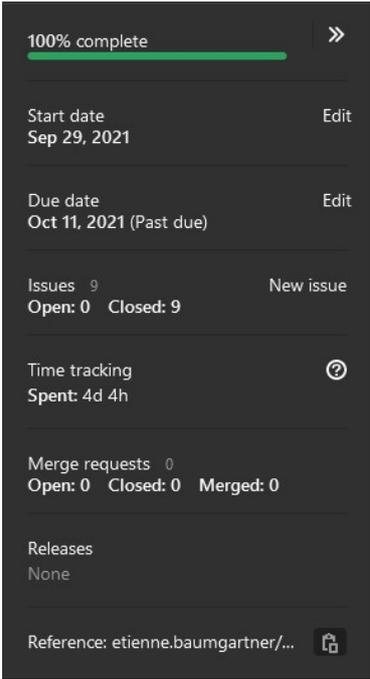
Abb. 33: Milestone Ansicht

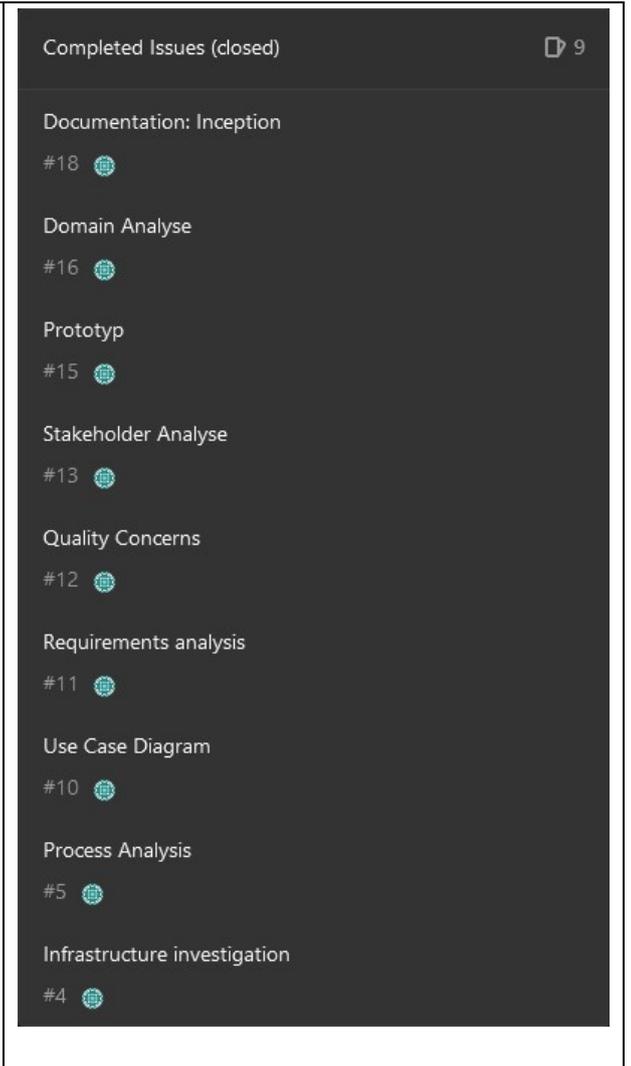
#### Planung

Übersicht	
	<div style="background-color: #333; color: white; padding: 10px;"> <p>100% complete &gt;&gt;</p> <hr/> <p>Start date Sep 20, 2021 Edit</p> <hr/> <p>Due date Oct 11, 2021 (Past due) Edit</p> <hr/> <p>Issues 5 <span style="float: right;">New issue</span> Open: 0 Closed: 5</p> <hr/> <p>Time tracking ⓘ Spent 1d 3h 55m Est 6h</p> <hr/> <p>Merge requests 0 Open: 0 Closed: 0 Merged: 0</p> <hr/> <p>Releases None</p> <hr/> <p>Reference: etienne.baumgartner/... 📄</p> </div>
Arbeitspakete	5
Ist-Zeit	11h 55m
Soll-Zeit	Keine Angabe, da kombiniert mit Inception Phase

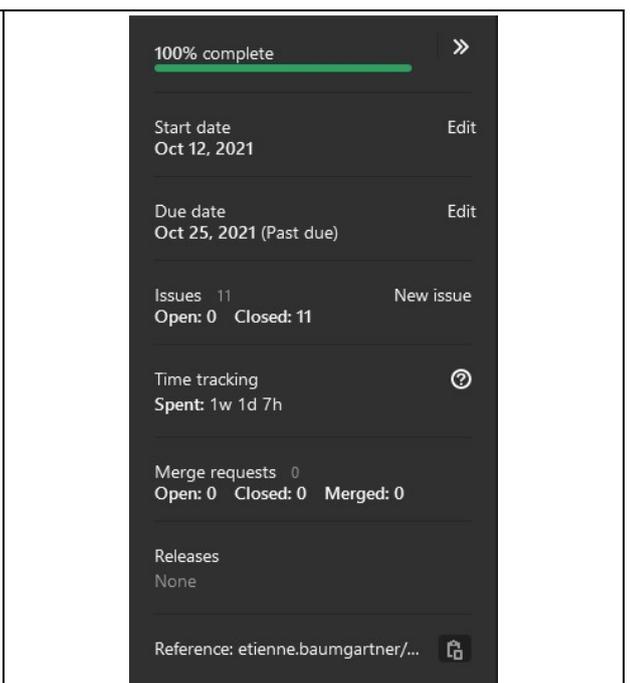
Arbeitspakete	
---------------	--

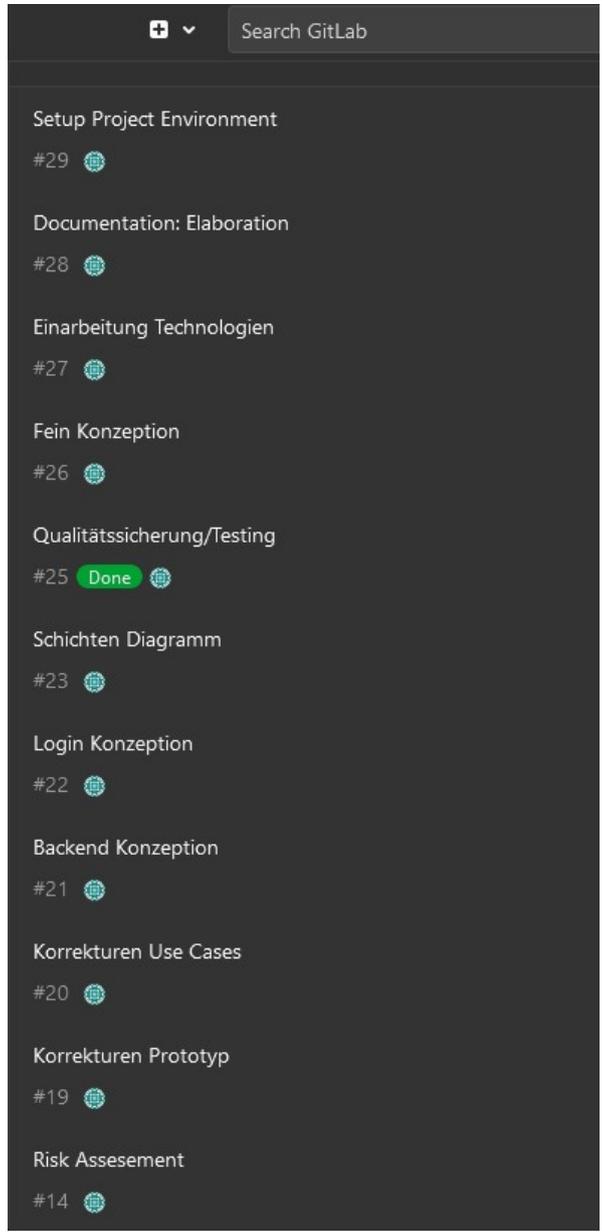
**Inception**

Übersicht	
Arbeitspakete	9
Ist-Zeit	36h
Soll-Zeit	Keine Angabe, da kombiniert mit Inception Phase

<p>Arbeitspakete</p>	 <p>Completed Issues (closed) <span style="float: right;">9</span></p> <ul style="list-style-type: none"> <li>Documentation: Inception #18</li> <li>Domain Analyse #16</li> <li>Prototyp #15</li> <li>Stakeholder Analyse #13</li> <li>Quality Concerns #12</li> <li>Requirements analysis #11</li> <li>Use Case Diagram #10</li> <li>Process Analysis #5</li> <li>Infrastructure investigation #4</li> </ul>
----------------------	--

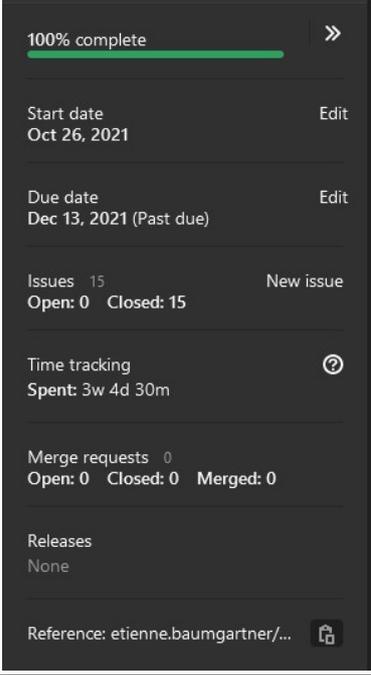
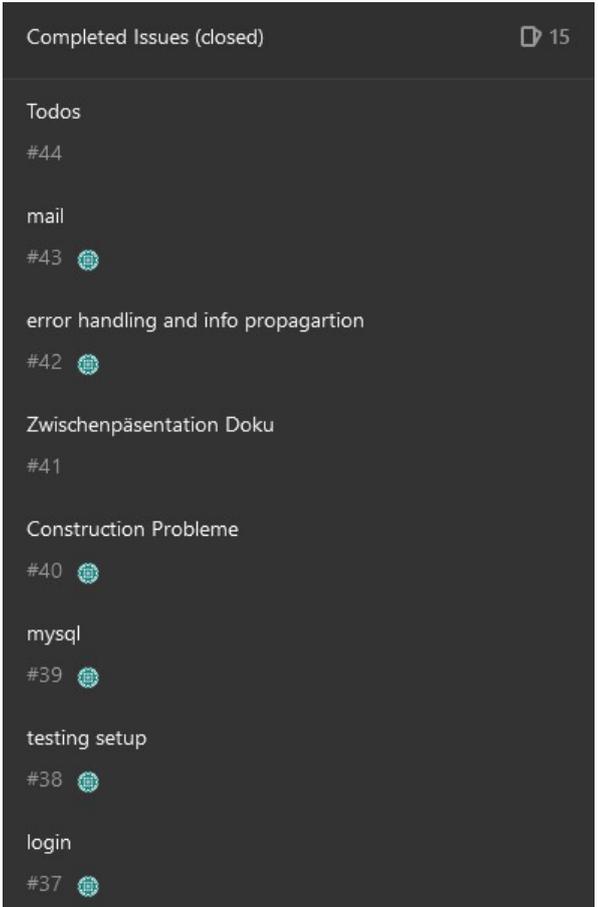
**Elaboration**

<p>Übersicht</p>	 <p>100% complete <span style="float: right;">»</span></p> <hr/> <p>Start date Oct 12, 2021 <span style="float: right;">Edit</span></p> <hr/> <p>Due date Oct 25, 2021 (Past due) <span style="float: right;">Edit</span></p> <hr/> <p>Issues 11 <span style="float: right;">New issue</span> Open: 0 Closed: 11</p> <hr/> <p>Time tracking <span style="float: right;">🕒</span> Spent: 1w 1d 7h</p> <hr/> <p>Merge requests 0 Open: 0 Closed: 0 Merged: 0</p> <hr/> <p>Releases None</p> <hr/> <p>Reference: etienne.baumgartner/... <span style="float: right;">🔖</span></p>
------------------	--

Arbeitspakete	11
Ist-Zeit	55h
Soll-Zeit	45h
Arbeitspakete	 <p>The screenshot shows a GitLab issue list with the following items:</p> <ul style="list-style-type: none"> <li>Setup Project Environment #29</li> <li>Documentation: Elaboration #28</li> <li>Einarbeitung Technologien #27</li> <li>Fein Konzeption #26</li> <li>Qualitätssicherung/Testing #25 (Done)</li> <li>Schichten Diagramm #23</li> <li>Login Konzeption #22</li> <li>Backend Konzeption #21</li> <li>Korrekturen Use Cases #20</li> <li>Korrekturen Prototyp #19</li> <li>Risk Assesement #14</li> </ul>

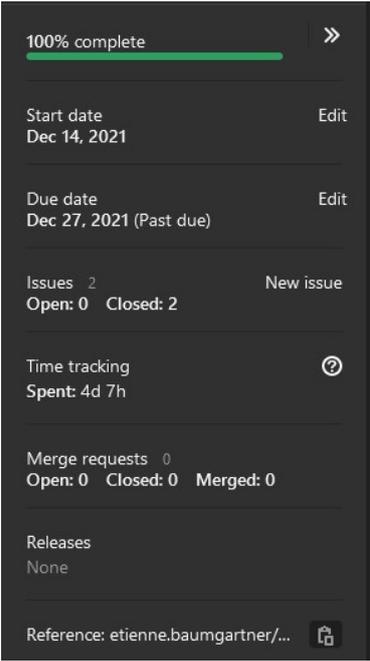
Construction

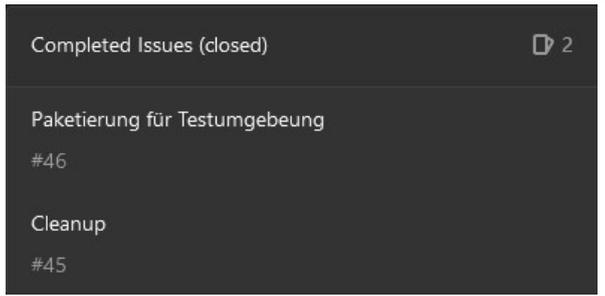
Übersicht	
-----------	--

	
Arbeitspakete	15
Ist-Zeit	152h 30m
Soll-Zeit	157h 30m
Arbeitspakete	

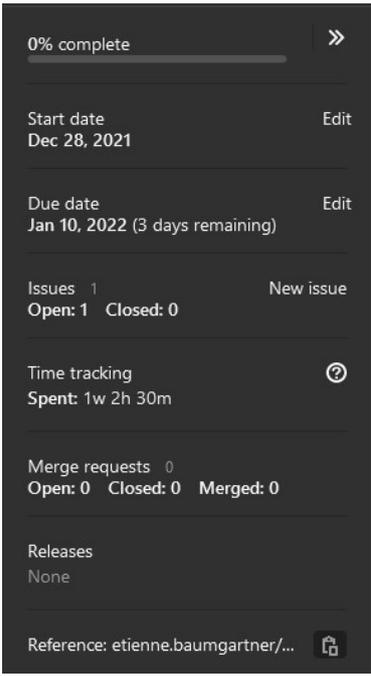
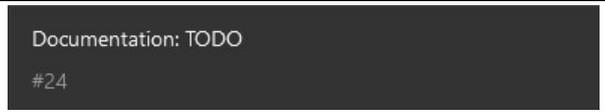
	<p>Prüfer: Anfrage #36 </p> <p>Admin: Anfrage #35 </p> <p>Anfrager: fortführende Anfrage #34 </p> <p>Anfrager: neue Anfrage #33 </p> <p>View Only: Anfrage #32 </p> <p>View Only: Anfrage Liste #31 </p> <p>Grundlayout #30 </p>
--	--

Transition

Übersicht	
Arbeitspakete	2
Ist-Zeit	39h
Soll-Zeit	45h

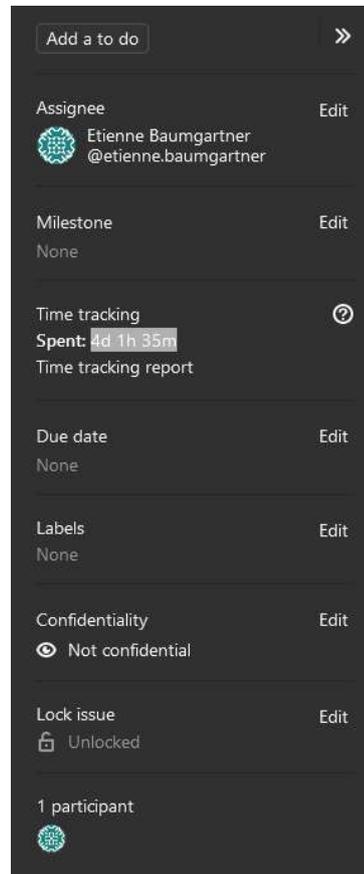
Arbeitspakete	
---------------	--

**Dokumentation**

Übersicht	
Arbeitspakete	1
Ist-Zeit	42h 30m
Soll-Zeit	45
Arbeitspakete	

**Not classiefied**

Ein Issue wurde nur zum Zweck der nicht klassifizierten Tätigkeiten erstellt. Darauf sind Arbeitsstunden eingetragen, welche keinem anderen Issue zugewiesen werden konnten. Die Tätigkeiten sind jeweils in den Kommentaren direkt auf dem Issue vermerkt. Das Arbeitspaket trägt den Namen *Not classified time consumption* und es wurden 33h 35m darauf verwendet



**Gesamt**

Die gesamthaft aufgewendete Zeit ist in der untenstehenden Tabelle verzeichnet:

Milestone	Ist-Zeit	Soll-Zeit	Differenz
Planung	11h 55m	67h 30m	19h 35m
Inception	36h		
Elaboration	55h	45h	10h
Construction	152h 30m	157h 30m	5h
Transition	39h	45h	6h
Documentation	42h 30m	45h	2h 30m
Keine Kategorie	33h 35m	-	33h 35m
<b>Total</b>	<b>369h 30m</b>	<b>360h</b>	<b>9h 30m</b>

In der Gesamtbetrachtung wurden die aufzubringenden Stunden erreicht bzw. 9 Stunden und 30 Minuten zusätzliche Aufwand erbracht.

Die Planung & Inception zusammen mit der Elaboration Phase weisen die grösste Abweichung zwischen Plan- und effektiv Arbeitszeit. Es wäre sinnvoller gewesen, für die Ausarbeitung des Entwurfs mehr und für die Analyse weniger Zeit einzuplanen.

Es wird darauf aufmerksam gemacht, dass 33 Stunden und 35 Minuten keiner der Phasen zugeordnet wurde.

Die anderen Phasen befinden sich im erwarteten Verhältnis der Plan- und Ausführungszeit. Eine Abweichung zur Planung von 10-15% liegt im akzeptablen Bereich. Die jeweilige Abweichung können in der untenstehenden Abbildung eingesehen werden.

<b>Milestone</b>	<b>Abweichung zu Soll-Zeit</b>
<b>Planung &amp; Inception</b>	-29.01%
<b>Elaboration</b>	+22.22%
<b>Construction</b>	-3.17%
<b>Transition</b>	-13.33%
<b>Documentation</b>	-5.55%

Die verbrachten Stunden auf den einzelnen Issues wird nicht bereitgestellt. Falls diese jedoch ebenfalls eingesehen werden möchten, kann dies auf dem Git Repository jederzeit gemacht werden.

**gtt.py**

Zu Beginn der Arbeit wurde ein python Tool zur Berechnung der aufgewendeten Stunden geschrieben. Dieses Tool erlaubt eine CLI Ansicht der auf den jeweiligen Issues verbrachte Zeit pro Milestone. Der Code des Tools wird dem Anhang ebenfalls angefügt.

## **IV Gitlab Time Tracking Skript – gtt.py**

**Übersicht**

Dieses Skript wurde zur Sammlung der Arbeitspakete geschrieben. Es gibt eine tabellarisierte Ansicht der Arbeitspaketzeiten pro Milestone zurück.

**Code**

```

import requests
import pandas as pd
from tabulate import tabulate
import matplotlib.pyplot as plt
import datetime

class gtt:

    def __init__(self):
        self.API_KEY = "?access_token=MUf6DP_PRU1uEfzxxp9_" # Enter your API key here
        self.BASE_URL = "https://gitlab.ost.ch/api/v4/projects/2576/"

        self.issues = self.getData()

    def __str__(self) -> str:
        sum = 0;
        for key, value in self.issues.items():
            print(
                f"Issues and Times for Milestone {key}: \n {tabulate(value, headers=value.columns,
                    tablefmt='pretty')}")
            sum += value['time'].sum()
            print(f"Total time on Milestone: {value['time'].sum()/3600}")

        return f"\n \n \n Total time spent on project: {datetime.timedelta(seconds=sum)}"

    def getData(self):
        milestones_id_dict = dict()
        milestones_issue_dict = dict()
        milestones = requests.get(f"{self.BASE_URL}/milestones{self.API_KEY}")
        for ele in milestones.json():
            milestones_id_dict[ele["title"]] = ele["id"]

        for milestone_name, milestone_id in milestones_id_dict.items():
            issues = requests.get(
                f"{self.BASE_URL}/milestones/{milestone_id}/issues{self.API_KEY}").json()
            issues_times = {"issue": [], "time": []}

            if not issues == []:
                issues_times = {
                    "issue": [issue["title"] for issue in issues],
                    "time": [issue["time_stats"]["total_time_spent"] for issue in issues]}

            milestones_issue_dict[milestone_name] = pd.DataFrame.from_dict(
                issues_times)

        return milestones_issue_dict

    # Showing pie of milestone issues

    def showplt(self, milestone):
        values = self.issues[milestone]["time"]
        labels = self.issues[milestone]["issue"]

```

**Abb. 34: gtt.py Code**

## V MySQL Daten Skripte

### Database Init

```
CREATE DATABASE IF NOT EXISTS db DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;

USE db;

-- auto-generated definition
create table bewerbung (
  bewerbung_id int auto_increment primary key,
  create_time datetime null,
  update_time datetime null default null,
  bezeichnung text null,
  betreuer text null,
  vorstellung text null,
  projekt_bericht varchar(255) null,
  projekt_video varchar(255) null,
  auszahlung int default 0 null,
  praesentation datetime null,
  fortfuehrende_bemerkung text null,
  fortfuehrender_projekt_bericht varchar(255) null,
  projekt_token varchar(255) null
);

create table bemerkung (
  bewerbung_id int not null,
  prufer_id int not null,
  prufer_name varchar(255) not null,
  status enum ('Angenommen', 'Abgelehnt', 'Ausstehend') default 'Ausstehend' not null,
  bemerkung text null,
  update_time datetime default null,
  constraint bemerkung_pk unique (bewerbung_id, prufer_id),
  constraint bemerkung_bewerbung_bewerbung_id_fk foreign key (bewerbung_id) references
bewerbung (bewerbung_id) on update cascade on delete cascade
);

create table status (
  bewerbung_id int not null,
  status enum ('Angenommen', 'Abgelehnt', 'Ausstehend') default 'Ausstehend' not null,
  constraint status_bewerbung_bewerbung_id_fk foreign key (bewerbung_id) references
bewerbung (bewerbung_id) on update cascade on delete cascade
);

create table team (
  bewerbung_id int not null,
  name varchar(255) null,
  email varchar(255) not null,
  constraint team_bewerbung_bewerbung_id_fk foreign key (bewerbung_id) references bew-
erbung (bewerbung_id) on update cascade on delete cascade
```

```
);

create table prufer_accounts (
  id int auto_increment primary key,
  username varchar(50) not null
);

Insert into
  prufer_accounts(id, username)
values
  (7, 'danpol');

Insert into
  prufer_accounts(id, username)
values
  (3, 'mirmeb');

Insert into
  prufer_accounts(id, username)
values
  (4, 'gabsch');

Insert into
  prufer_accounts(id, username)
values
  (5, 'uweleo');

Insert into
  prufer_accounts(id, username)
values
  (6, 'marsto');

create trigger init_status
after
insert
  on bewerbung for each row
insert into
  status (bewerbung_id)
VALUES
  (NEW.bewerbung_id);

create trigger setup_bemerkung
after
insert
  on bewerbung for each row
Insert into
  bemerkung (bewerbung_id, prufer_id, prufer_name)
select
```

```
NEW.bewerbung_id as bewerbung_id,
id,
username
from
  prufer_accounts;

-- make sure that one is part of the db, else add manually

create trigger compute_status_update
  after update
  on db.bemerkung
  for each row
Begin if(
  (
    select
      count(*)
    from
      db.bemerkung
    where
      status like 'Angenommen'
      and db.bemerkung.bewerbung_id = NEW.bewerbung_id
    ) >= 5
  ) Then
update
  db.status
set
  status = 'Angenommen'
where
  db.status.bewerbung_id = NEW.bewerbung_id;

else if (
  (
    select
      count(*)
    from
      db.bemerkung
    where
      status like 'Abgelehnt'
      and db.bemerkung.bewerbung_id = NEW.bewerbung_id
    ) >= 5
  ) then
update
  db.status
set
  status = 'Abgelehnt'
where
  db.status.bewerbung_id = NEW.bewerbung_id;
```

```
else
update
    db.status
set
    status = 'Ausstehend'
where
    db.status.bewerbung_id = NEW.bewerbung_id;

end if;
end if;
end;
```

**Auth Init**

```
CREATE DATABASE IF NOT EXISTS innoNetAuth DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;

USE innoNetAuth;

create table accounts (
    id int auto_increment primary key,
    username varchar(50) not null,
    password varchar(255) not null,
    email varchar(100) not null,
    role enum ('Admin', 'Prufer') not null,
    salt varchar(255) null
);
```

## VI Prototyp Views

**Übersicht** Beachte, dass es sich bei den folgenden Ansichten um Prototypendesigns handelt und nicht um die echte Darstellung.

**View Only** <https://www.innovationnetwork.ch/bewerbung/>

innovation  
Network

Verein · Veranstaltungen · Projekte · Bewerbung

Neue Bewerbung

Eingang	Projekt	Team	Betreuer	Status	Ausbezahlt	Präsentation
04.10.2021	<a href="#">BA-InnoNetwork</a>	Etienne Baumgartner	Prof. Daniel Politze	Angenommen	3000 CHF	14.01.2022
01.01.2000	<a href="#">Blah Blah</a>	Hans Muster	Peter Meier	Ausstehend	-	-
02.02.2020	<a href="#">Hee Hee</a>	Hans Peter	Muster Meier	Abgelehnt	-	-
03.05.2010	<a href="#">Sample</a>	Muster Peter	Hans Meier	Abgeschlossen	5000 CHF	01.02.2011

miro

Abb. 35: Prototyp Bewerbung Übersicht

<https://www.innovationnetwork.ch/bewerbung/anfrage/:id>

innovation  
Network

Verein · Veranstaltungen · Projekte · Bewerbung

**Bewerbung**

**Projekt:** BA-InnovationNetwork  
**Eingang:** 04.10.2021  
**Team:** Etienne Baumgartner  
**Betreuer:** Prof. Daniel Politze

**Projektvorstellung:** 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.

**Projektbericht:** Projekt Download (pdf)

**Video Pitch:** 

**Fortführende Anfrage:** Fortführende Anfrage

**Status:** ✔ Angenommen

**Präsentation:** 14.01.2022

**Förderbetrag:** 3000 CHF

miro

Abb. 36: Prototyp Anfrage Ansicht

Interaktion ohne Login <https://www.innovationnetwork.ch/bewerbung/new/>

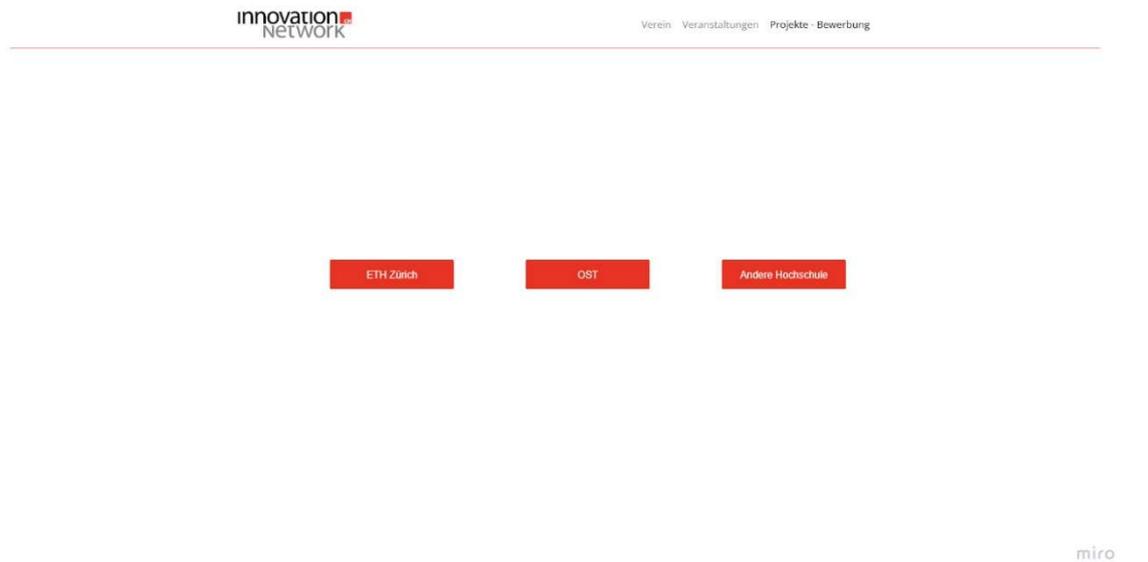


Abb. 37: Prototyp neue Anfrage Auswahl

<https://www.innovationnetwork.ch/bewerbung/new?schule=ost>

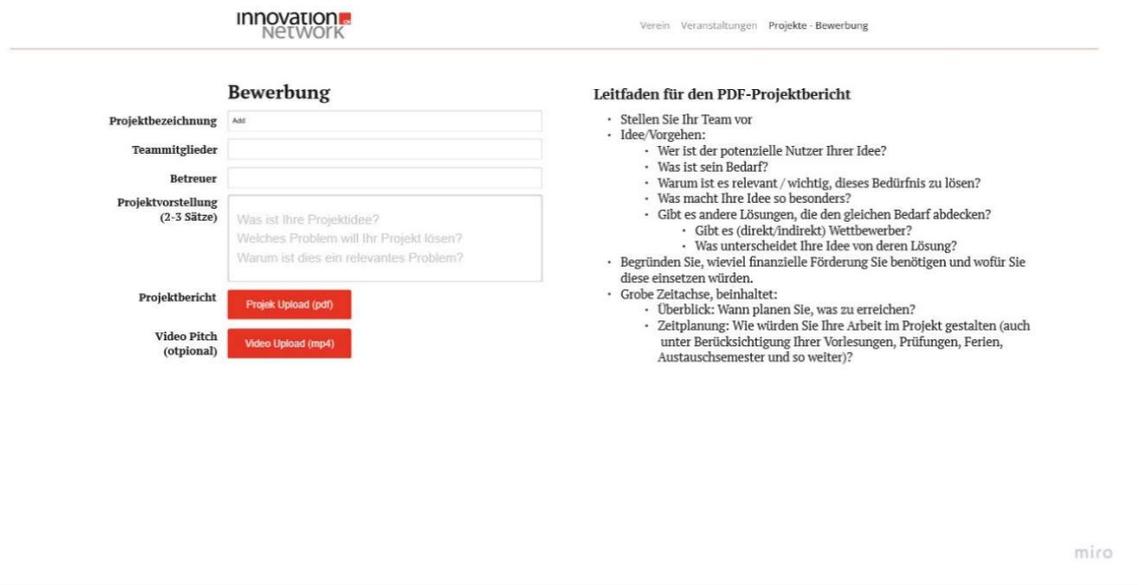


Abb. 38: Prototyp neue Anfrage

<https://www.innovationnetwork.ch/bewerbung/fortfuehrendeAnfrage/:id>

**innovation NETWORK** Verein | Veranstaltungen | Projekte - Bewerbung

### Fortführende Bewerbung

**Bemerkungen**

**Fortführender Projektbericht** [Projekt Download \(pdf\)](#)

**Leitfaden für fortführende PDF-Bewerbung**

Bitte reichen Sie eine Begründung für die gewünschte zusätzliche Förderung auf einer Seite (pdf) ein.

Folgende Punkte sollten enthalten sein:

- Beschreiben Sie den Projektfortschritt aus der initialen Förderungsphase
- Erwähnen Sie Resultate aus dem vergangenen halben Jahr (Absolvierte Tests, gefertigte Prototypen, durchgeführte Nutzerstudien,...)
- Begründen Sie, wieviel weitere finanzielle Förderung Sie benötigen und wofür Sie diese einsetzen würden.
- Ihre Kontaktdaten

[Fortführende Anfrage](#)

miro

Abb. 39: Prototyp fortführende Anfrage

Interaktion mit  
Login

<https://www.innovationnetwork.ch/bewerbung/anfrage/:id> [Admin]

**innovation NETWORK** Verein | Veranstaltungen | Projekte - Bewerbung

### Bewerbung

**Projekt** BA-InnovationNetwork  
**Eingang** 04.10.2021  
**Team** Etienne Baumgartner  
**Betreuer** Prof. Daniel Politze  
**Projektvorstellung** 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.

**Projektbericht** [Projekt Download \(pdf\)](#)

**Video Pitch**

**Fortführende Anfrage** [Fortführende Anfrage](#)

**Status: Angenommen** Deadline: 18.10.2021

- Prof. Dr. Daniel Politze
- Prof. Dr. Mirko Meboldt
- Dr. Gabriele Schwarz
- Dr. Uwe Leonhardt
- Dr. Martin Stöckli

**Präsentation**

June 2016						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

**Auszahlung 1. Batch**

**Auszahlung 2. Batch**

**Auszahlung Gesamt:**  miro

Abb. 40: Prototyp Administrations Ansicht

<https://www.innovationnetwork.ch/bewerbung/anfrage/:id> [Prüfer]

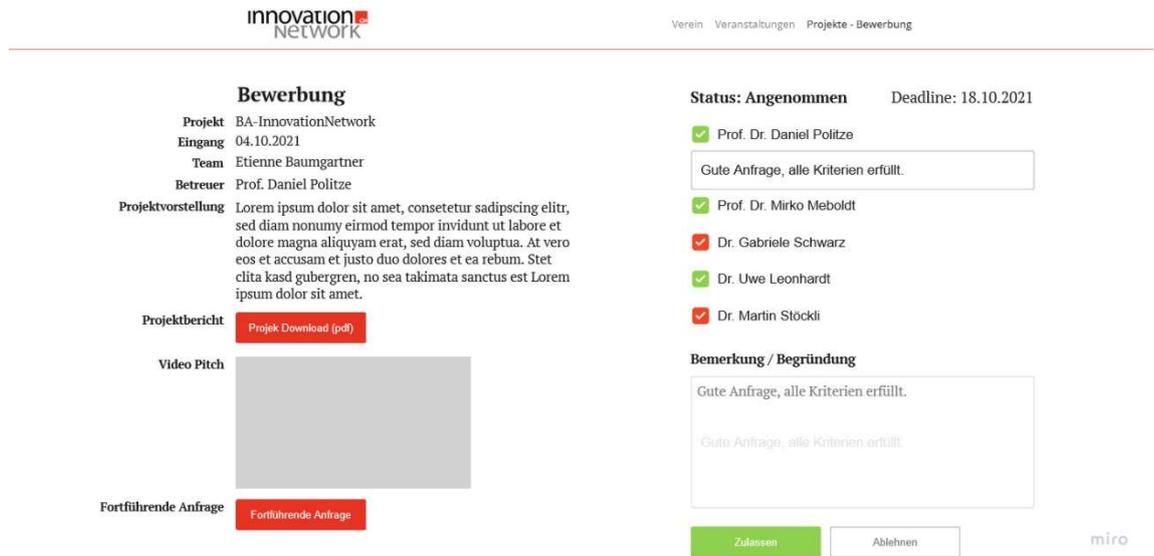


Abb. 41: Prototyp Prüfer Ansicht

## VII MVP Views

**main.handlebars** Das Grundlayout (*main.handlebars*) ist bei allen Views gleich. Die jeweiligen Inhalte werden injiziert.

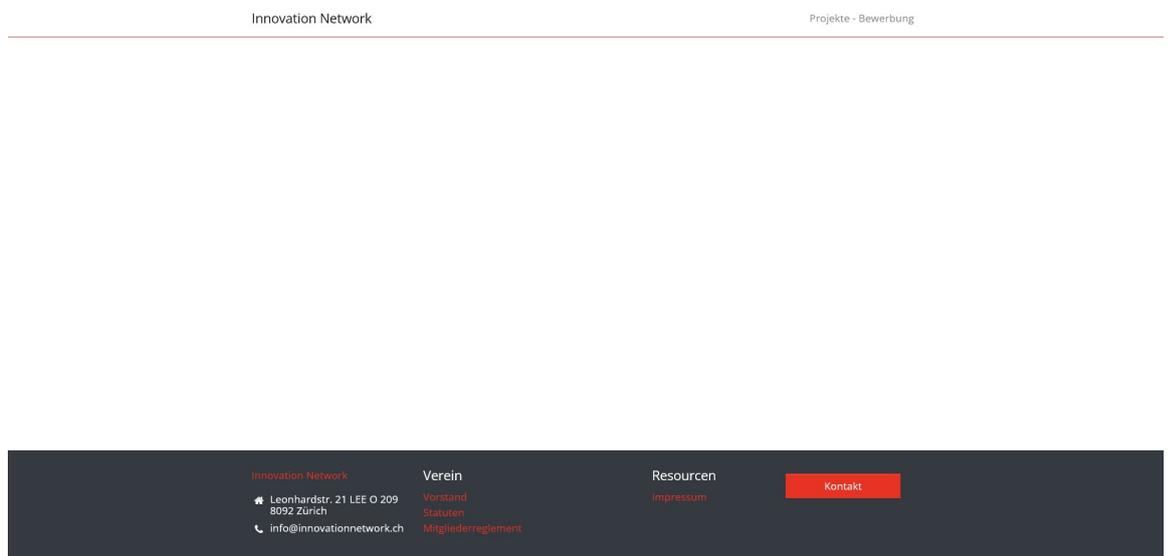


Abb. 42: MVP Grundlayout

**bewerbung.handlebars** Stellt den Eingangspunkt der Node Anwendung dar. Listet alle vorhandenen Bewerbungen, welche für den momentanen Benutzer einsehbar sind.

Szenario: Keine angenommenen Bewerbungen:

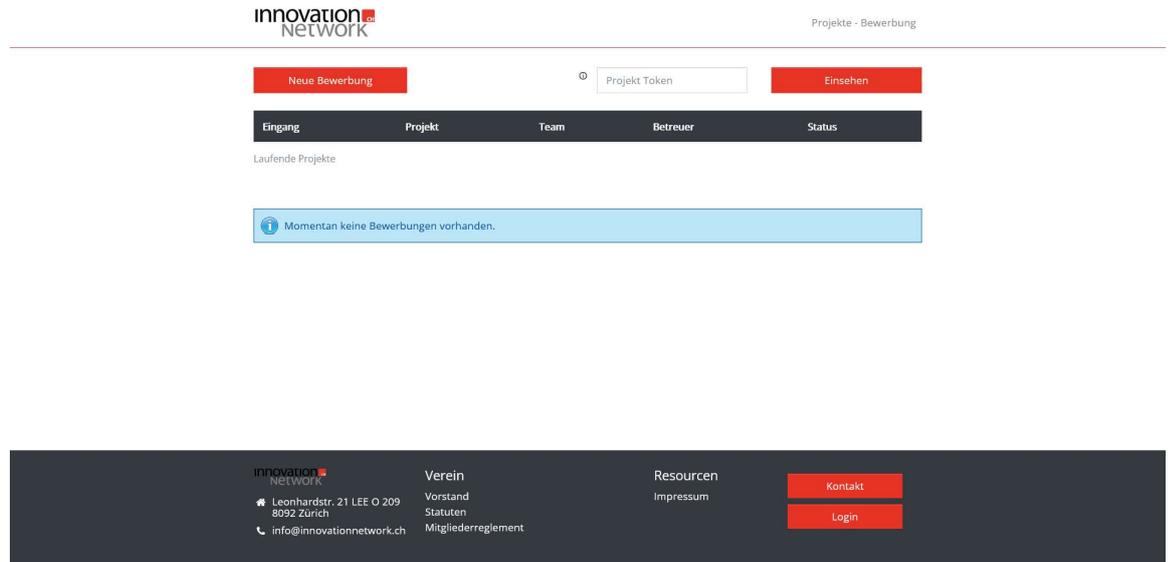


Abb. 43: MVP Bewerbung Übersicht (keine Bewerbungen)

Szenario: Eingeloggt als Admin:

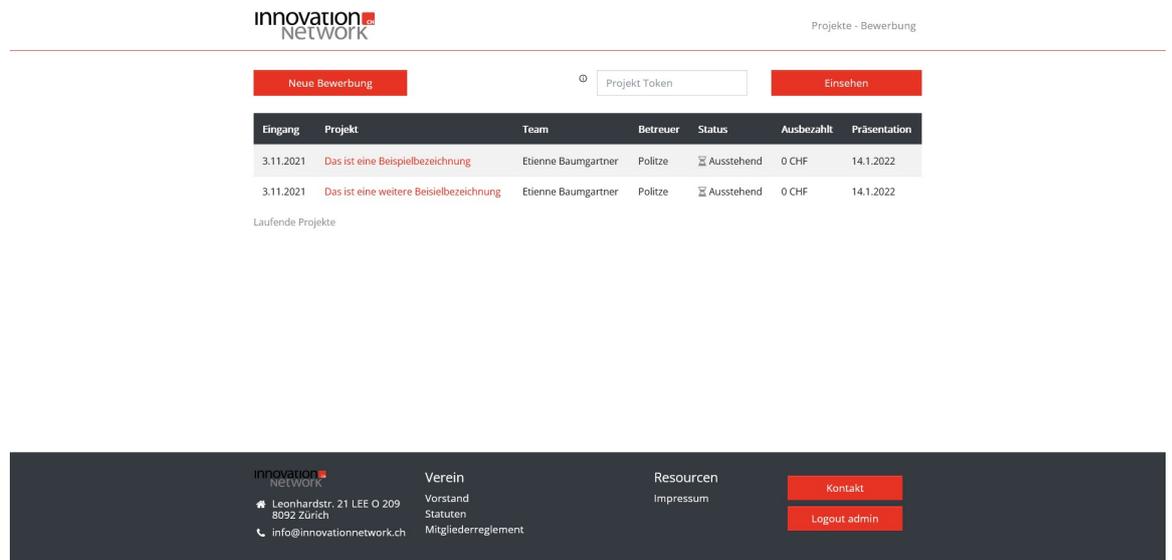


Abb. 44: MVP Bewerbung Übersicht (Admin Ansicht)

neueAnfrage/neue-AnfrageAuswahl.handlebars - Behandelt die Eingabe einer neuen Bewerbung.

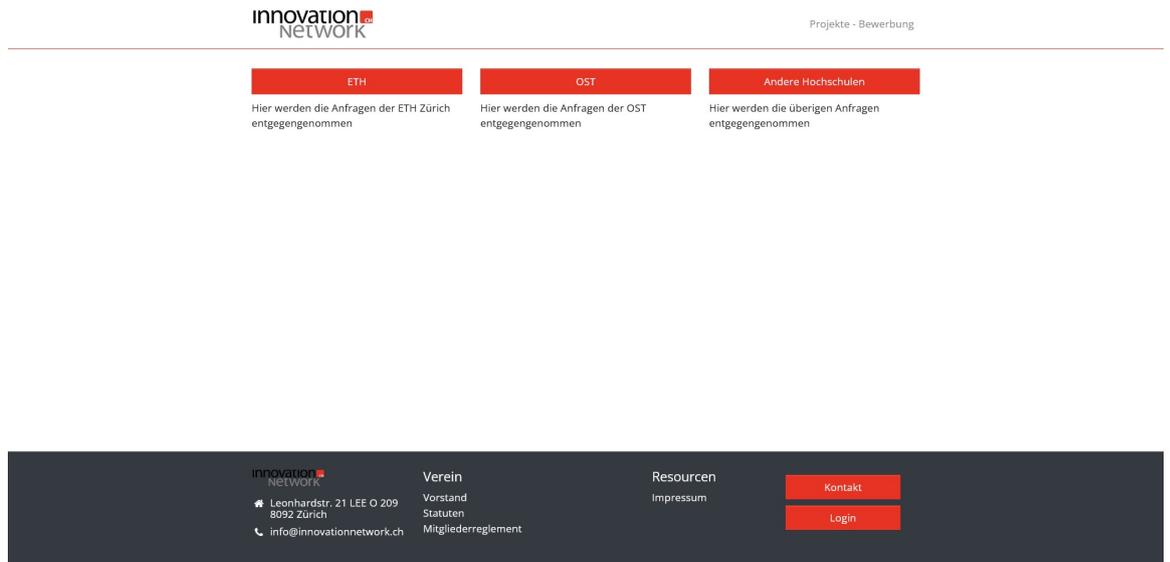


Abb. 45: MVP neue Anfrage Auswahl

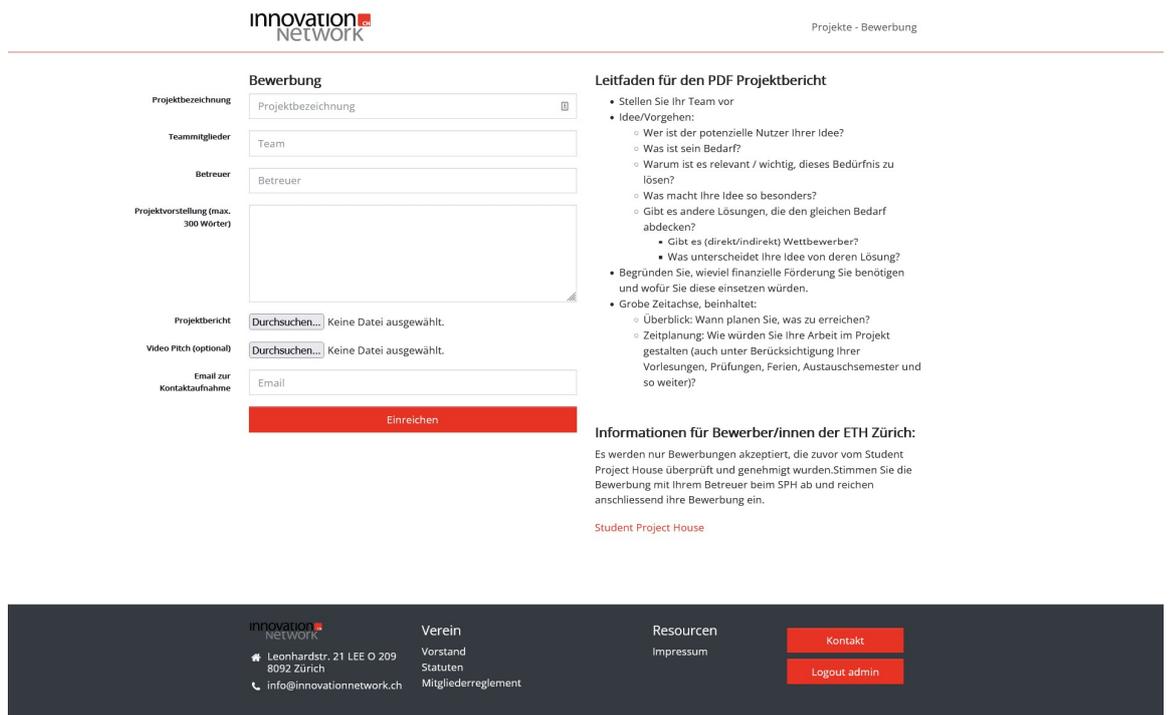


Abb. 46: MVP neue Anfrage

fortfuehrendeAnfrage.handlebars

Behandelt die Eingabe einer fortfuehrenden Anfrage.

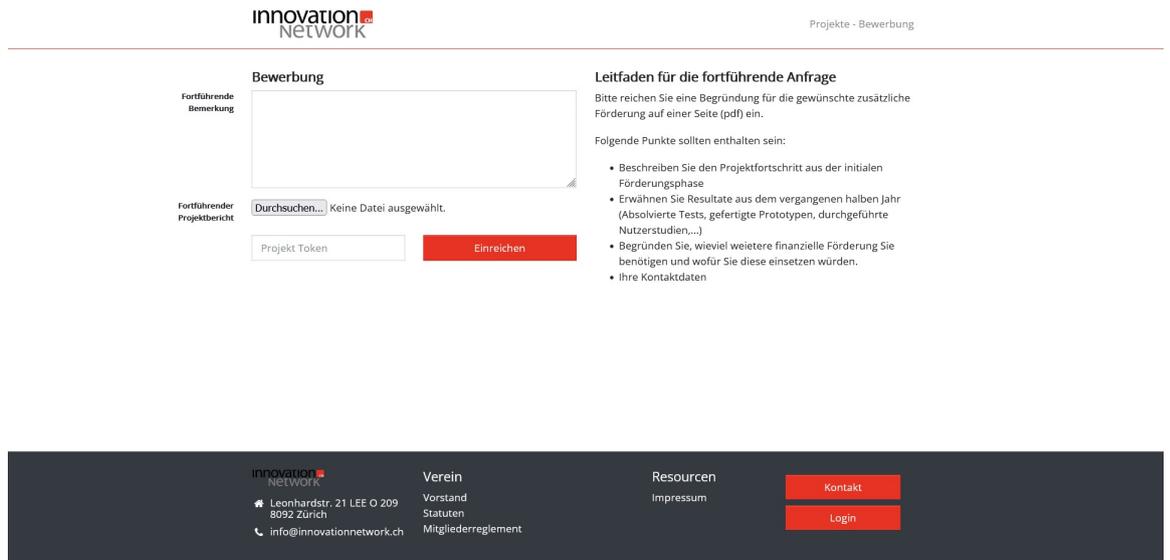


Abb. 47: MVP fortführende Anfrage

`anfrage.handlebars` Gibt alle Informationen für den jeweiligen Benutzer heraus.

**Szenario Anfrager oder Mitglied Zugriff:**

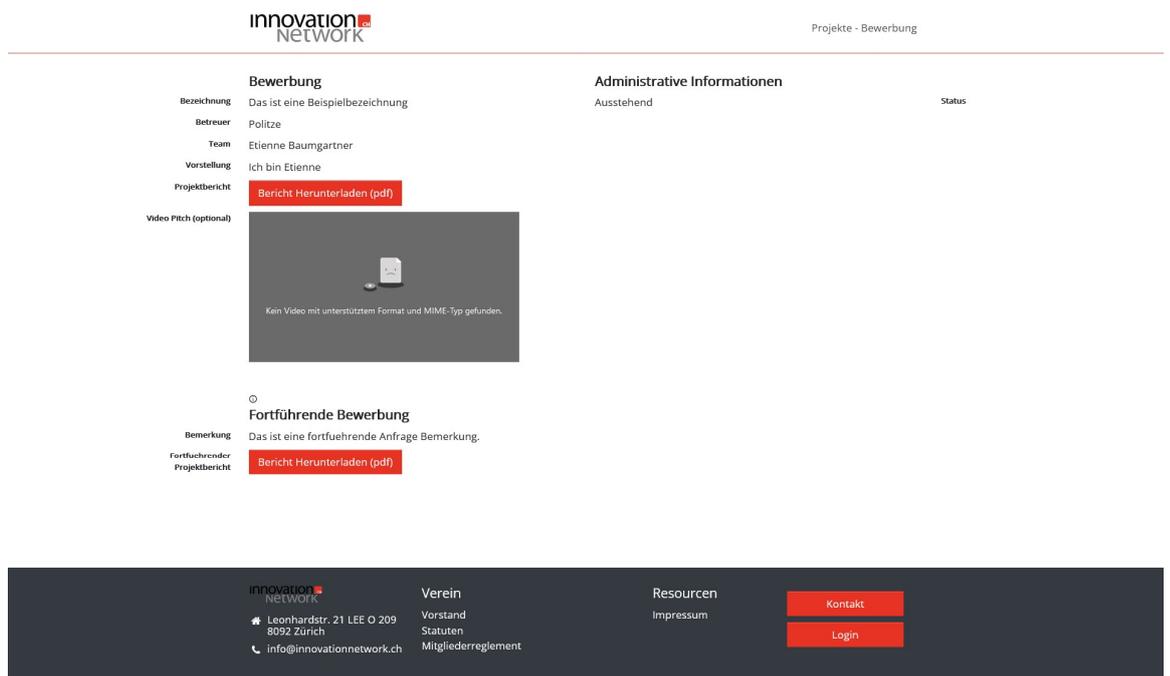


Abb. 48: MVP Anfrage Einsicht (Anfrager)

**Szenario Admin Zugriff:**

The screenshot shows the 'Bewerbung' (Application) view for an administrator. The page is divided into several sections:

- Bewerbung (Application Details):**
  - Bezeichnung: Das ist eine Beispielbezeichnung
  - Betreuer: Politze
  - Team: Etienne Baumgartner
  - Vorstellung: Ich bin Etienne
  - Projektbericht: Bericht Herunterladen (pdf)
  - Video Pitch (optional): Kein Video mit unterstütztem Format und MIME-Typ gefunden.
  - Fortführende Bewerbung: Das ist eine fortführende Anfrage Bemerkung. Bericht Herunterladen (pdf)
- Administrative Informationen (Administrative Information):**
  - Ausstehend: 0
  - Dates: 3.11.2021, 14.1.2022
  - Prüfer: prufer, prufer2, prufer3, prufer4, prufer5
  - Status: Auszahlung
  - Erstellungsdatum: 3.11.2021
  - Präsentation: Präsentation
  - Bemerkungen: Bemerkungen:
- Administrative Interaktion (Administrative Interaction):**
  - Update Präsentation (button)
  - Update Auszahlung (button)
  - Präsentation: Präsentation
  - Auszahlung Total: Auszahlung Total

The footer contains contact information for the Verein (Association), including address (Leonhardstr. 21 LEE O 209, 8092 Zürich) and email (info@innovationnetwork.ch). It also includes links for Ressourcen (Resources), Impressum, Kontakt, and Logout admin.

Abb. 49: MVP Anfrage Einsicht (Admin)

### Szenario Prüfer Zugriff:

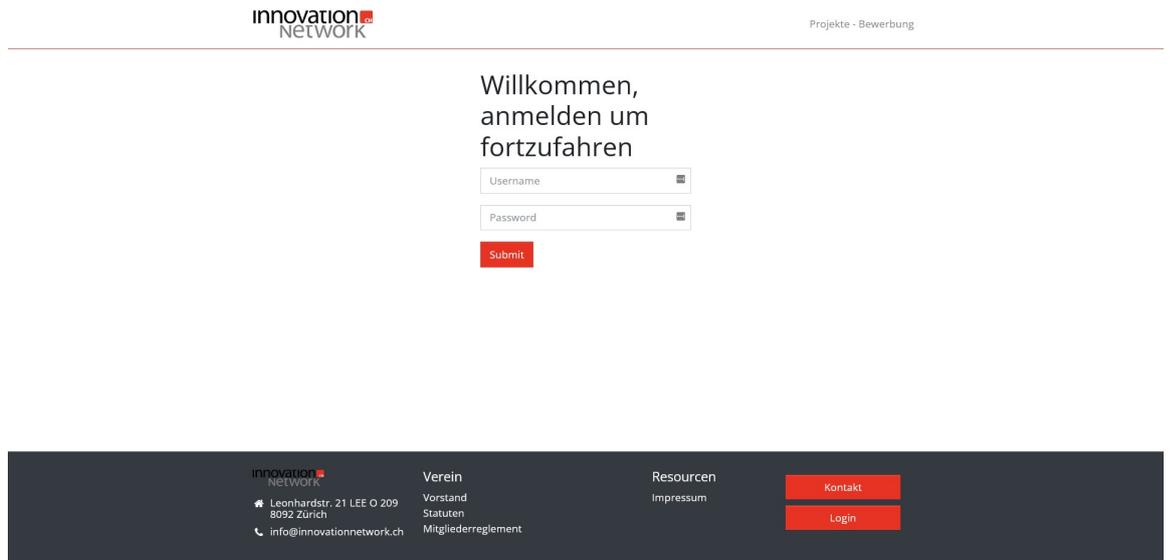
The screenshot shows the 'Bewerbung' (Application) view for a reviewer. The page is divided into several sections:

- Bewerbung (Application Details):**
  - Bezeichnung: Das ist eine Beispielbezeichnung
  - Betreuer: Politze
  - Team: Etienne Baumgartner
  - Vorstellung: Ich bin Etienne
  - Projektbericht: Bericht Herunterladen (pdf)
  - Video Pitch (optional): Kein Video mit unterstütztem Format und MIME-Typ gefunden.
  - Fortführende Bewerbung: Das ist eine fortführende Anfrage Bemerkung. Bericht Herunterladen (pdf)
- Administrative Informationen (Administrative Information):**
  - Ausstehend: 0
  - Dates: 3.11.2021, 14.1.2022
  - Prüfer: prufer, prufer2, prufer3, prufer4, prufer5
  - Status: Auszahlung
  - Erstellungsdatum: 3.11.2021
  - Präsentation: Präsentation
  - Bemerkungen: Bemerkungen:
- Prüfer Interaktion (Reviewer Interaction):**
  - Formular zum Verfassen eines Kommentars (Kommentar verfassen)
  - Zulassen (button)
  - Ablehnen (button)
  - Kommentieren (button)

The footer contains contact information for the Verein (Association), including address (Leonhardstr. 21 LEE O 209, 8092 Zürich) and email (info@innovationnetwork.ch). It also includes links for Ressourcen (Resources), Impressum, Kontakt, and Logout pruefer.

Abb. 50: MVP Anfrage Einsicht (Prüfer)

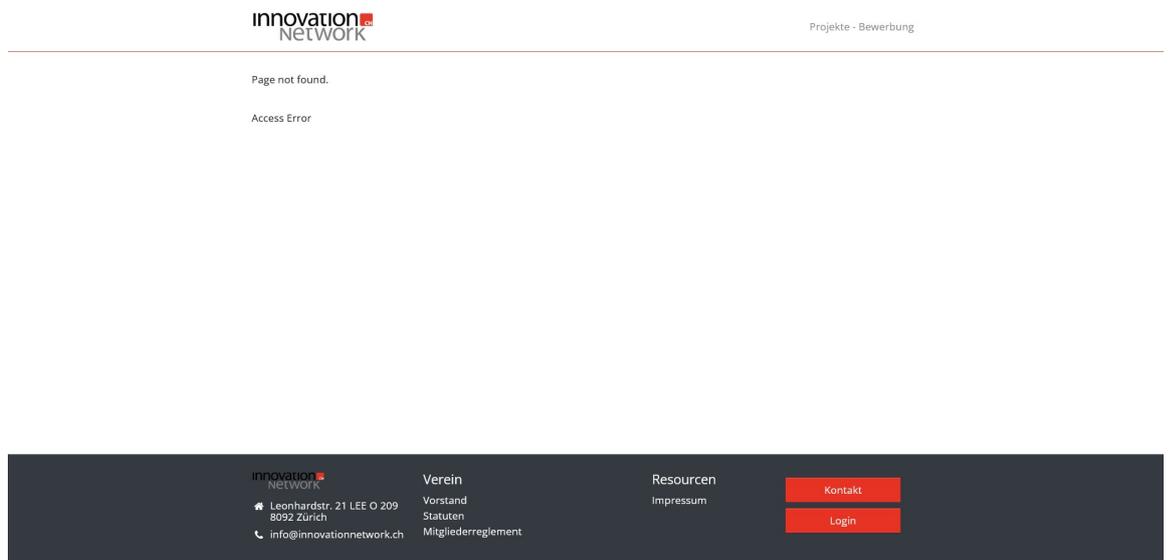
login.handlebars Erlaubt das Login der Prüfer und des Admins. Ist stets unten rechts aufgelistet.



**Abb. 51: MVP Login**

**error.handlebars** Ein Fehler wird auf dieser View dargestellt und kann weitere Informationen enthalten.

**Szenario Zugriff auf Bewerbung ohne Zugriffsrechte:**



**Abb. 52: MVP Error Seite**

## VIII UAT Skript

### Skript

# User Acceptance Test

## Informationen

- Leider habe ich noch nicht alle E-Mail Adressen der User, daher habe ich die Email Notifikationen ausgeschaltet.
- Bitte vermerken Sie alle Dinge, die Ihnen aufgefallen sind. Ich würde diese gerne in meiner Arbeit als Verbesserungsvorschläge im Anhang aufnehmen.

## Logins

User	Login	Passwort
Mirko Meboldt	mirmeb	mirmeb
Gabriel Schwarz	gabsch	gabsch
Uwe Leonhardt	uweleo	uweleo
Martin Stöckli	marsto	marsto
Daniel Politze	danpol	danpol
Jasmin Jansen	jasjan	jasjan

## Anfrager (Bereits durchgeführt)

1. Gehe auf <https://innonet.sifs0003.infs.ch/bewerbung/>
2. Klicke auf Neue Bewerbung
3. Fülle das Formular aus
4. Klicke auf Einreichen
5. Warte auf die Weiterleitung
6. Einsehen der Bewerbung
7. (Hier würde eine Email Notifikation rausgehen)

## Prüfer

1. Gehe auf <https://innonet.sifs0003.infs.ch/bewerbung/>
2. Klicke den Login Button (unten rechts)
3. Nach erfolgreichem Login werden alle vorhandenen Bewerbungen angezeigt
4. Klicken Sie auf eine vorhandene Bewerbung
5. Laden Sie den Projektbericht herunter (Dummy Bericht)
6. Schauen Sie sich, falls vorhanden, das Projekt Video an (Dummy Video)
7. Schreiben Sie einen Kommentar und bestätigen Sie eine der beiden Optionen (Annehmen oder Ablehnen) Kommentieren können Sie immer. **Achtung:** Kommentieren überschreibt den vorherigen Kommentar)

Sobald fünf Annehmen vorhanden sind, wird der Status zu *Angenommen* wechseln.

Sobald fünf Ablehnungen vorhanden sind, wird der Status zu *Abgelehnt* wechseln.

➔ Es ist momentan also eine einstimmige Prüfung nötig, um den Status einer Bewerbung zu setzen. Ansonsten bleibt die Bewerbung im Status *Ausstehend*.

(Hier würde eine Email Notifikation rausgehen und ab diesem Moment kann ein Anfrager eine fortführende Bewerbung starten)

### Abb. 53: UAT Skript

## IX UAT Rückmeldungen

**Übersicht** Die Rückmeldungen werden aus zeitlichen Gründen nicht mehr speziell aufbereitet, sondern lediglich als zusätzliches Material zur Verfügung gestellt.

### Martin Stöckli

Liebe Kolleginnen und Kollegen

Zuerst einmal ein Gutes Neues Jahr allen!

Ich habe das Tool bereits getestet.

Feedback: Soweit gut und klar! Sehr schön ist die Einfachheit.

Zwei Bemerkungen:

- Was heisst "Fortführende Bewerbung" → Neueinreichung (2. Versuch) oder Anschlussfinanzierung (2. Teil)?
- Zum Logout kommt bei mir ein Error. Es fehlt wohl eine Logoutbestätigungssseite.

**Abb. 54: Rückmeldung Martin Stöckli**

## X Danksagung

An dieser Stelle möchte ich mich beim Institut für Software und vor allem bei Nicola Jordan bedanken, der die Aufschaltung der Testumgebung über Weihnachten ermöglichte und mir trotz Ferien bei der reibungslosen Umsetzung zur Seite stand.