

# Projekt Softwaretechnik / Medieninformatik

## Dokumentation

25. Juni 2022



Projekt  
**Q-CV**

Team: QUANTO Solutions

vorgelegt von

Christoph Merck	765460
Leto Ziegler	765000
Marc Starzmann	765689
Navamin Leelarburanathanakoon	768734

# Kurzfassung

---

Diese Dokumentation entstand im Zuge des Moduls Projekt Softwaretechnik und Medieninformatik und der daraus entstandenen Software. Zusätzlich zur Dokumentation wird in diesem Dokument auf den Entwicklungsprozess und das Projektmanagement eingegangen.

# Inhaltsverzeichnis

---

<b>Kurzfassung</b>	<b>1</b>
<b>Inhaltsverzeichnis</b>	<b>2</b>
<b>Abbildungsverzeichnis</b>	<b>4</b>
<b>1. Allgemeines</b>	<b>5</b>
1.1 Zielgruppe	5
1.2 Personas	6
1.3 Probleme	7
1.4 Eigenschaften	7
1.5 Alleinstellungsmerkmal	7
<b>2. User Stories</b>	<b>8</b>
2.1 Einloggen bei Q-CV	8
2.2 Skills verwalten	8
2.3 Projekte verwalten	9
2.4 Erstellen von verschiedenen CV's	9
2.5 Downloaden/Exportieren von CV's	9
2.6 Stammdaten aktualisieren	10
2.7 Bildung verwalten	10
2.8 Karriere verwalten	10
2.9 Benutzer erstellen	11
<b>3. User Interface</b>	<b>12</b>
3.1 Prototyp	12
3.2 Usability	16
<b>4. Technisches Konzept</b>	<b>17</b>
4.1 Frameworks	17
4.1.1 React	17
4.1.2 Express	17
4.1.3 NodeJS	17
4.1.4 Mongoose	17
4.2 Softwarearchitektur	18
4.3 Datenbank	20
<b>5. Projektumsetzung</b>	<b>21</b>
5.1 Arbeitsaufteilung	21
5.2 Management	22

<b>6. Reflektion</b>	<b>23</b>
6.1 Teamarbeit	23
6.2 Projektmanagement	23
6.3 Lernfortschritt	24
<b>7. Lizenzen</b>	<b>25</b>
<b>8. Anleitungen</b>	<b>26</b>
8.1 Installationshandbuch	26
8.2 Administrationshandbuch	28
8.2.1 Login	28
8.2.2 Dashboard	28
8.2.3 User Data	29
8.2.4 Education	29
8.2.5 Career	29
8.2.6 IT-Skills	30
8.2.7 Projects	30
8.2.8 Create User	31
<b>9. Fazit</b>	<b>32</b>

# Abbildungsverzeichnis

---

<b>Abb.1:</b> Persona 1: Erik Müller	6
<b>Abb.2:</b> Persona 2: <i>Lena Schmidt</i>	6
<b>Abb.3:</b> Login Bildschirm	12
<b>Abb.4:</b> Dashboard	13
<b>Abb.5:</b> User Data	13
<b>Abb.6:</b> Education	14
<b>Abb.7:</b> Career	14
<b>Abb.8:</b> IT-Skills	15
<b>Abb.9:</b> Projects	15
<b>Abb.10:</b> MERN Struktur	18
<b>Abb.11:</b> Client Diagramm	19
<b>Abb.12:</b> Server Diagramm	19
<b>Abb.13:</b> ERM Diagramm	20

# 1. Allgemeines

---

Greifen wir zum Einstieg noch einmal auf, worum es in der Software geht. Wir bauen für QUANTO Solutions eine Software für effizientere und einheitlichere Lebenslaufferstellung. Dabei soll alles per Datenbank verwaltet werden und es ist möglich, mit einfachem Drag-and-drop verfahren, schnell einen Lebenslauf nach Bedarf des zukünftigen Projekts zu erstellen.

Um dieses Projekt nun zu manifestieren und besser zu beschreiben, gehen wir in dieser Dokumentation auf die wichtigsten Punkte ein, bzw. zeigen auf, wie die Software zukünftig laufen soll.

## 1.1 Zielgruppe


Als Zielgruppe der Software wird eindeutig die komplette Firma QUANTO Solutions gesehen. Letztendlich wird die Software von allen Mitarbeitern genutzt und dient ebenso als gut nutzbare Datenbank für bereits geleistete Projekte. Generell gesagt, dient sie als Effizienzsteigerung der Arbeit in QUANTO Solutions.

Ziel sollte es dabei sein, sobald ein Mitarbeiter die Anweisung bekommt, für ein neues Projekt einen Lebenslauf zu erstellen, kann er sich aus den Punkten Projekte, Abschlüsse, Karriere und IT-Skills das Passende nehmen. So entsteht dann der passende Lebenslauf.

Um nochmal ein besseres Vorstellungsvermögen von den Nutzern und Zielgruppen zu bekommen, haben wir zwei Personas erstellt.

## 1.2 Personas

### Erik Müller



"Die Zufriedenheit der Kunden ist das oberste Gebot."

Alter: 32  
Beruf: Berater  
Familie: ledig  
Wohnort: Esslingen

#### Persönlichkeit

Introvertiert	Extrovertiert
Überdenker	Gefühl
Gespür	Intuition
Verurteilend	Wahrnehmung

#### Ziele

- In der Firma Aufsteigen und einen guten Ruf besitzen.
- Alle Projekte mit 100% Zufriedenheit durchführen.
- Viele Kundinnen und Kunden wie möglich mit seinen Fähigkeiten beeindrucken.

#### Frustrationen

- Anstrengende Erstellung von Kundenspezifischen Lebensläufen.
- Keine einheitliche Designs
- Fehlende Projektverwaltung.

#### Profil

Erik, 32 Jahre alt, von Beruf Berater, arbeitet bereits seit einigen Jahren als Berater im IT-Bereich, und kennt die aktuelle Situation mit der Verwaltung der Lebensläufe innerhalb der Firma gut. In vielen Firmen wird das Gestalten der Lebensläufe der Person selbst überlassen und die Firma hat somit kein einheitliches Erscheinungsbild. Verschiedene Skills sind unterschiedlich formatiert und es gibt kein Design.


Ich würde mir wünschen, dass es möglich wäre, das Erstellen der Lebensläufe zu vereinfachen und einheitlich zu gestalten, um einen besseren Eindruck auf Kunden zu erzeugen

#### Motivationen

Initiative	80%
Angst	20%
Reife	60%
Energie	85%
Sozial	40%

Abb. 1 Persona: Erik Müller, 32 Jahre, Berater

### Lena Schmidt



"Ordnung und Einheitlichkeit erzeugt Zufriedenheit."

Alter: 36  
Beruf: Beraterin  
Familie: verheiratet mit Kind  
Wohnort: Stuttgart

#### Persönlichkeit

Introvertiert	Extrovertiert
Überdenker	Gefühl
Gespür	Intuition
Verurteilend	Wahrnehmung

#### Ziele

- Mit ihrem Job genügend Geld für ihre Familie zu verdienen.
- Gemeinsam mit der Firma voranschreiten und wachsen.
- Ein einheitliches Auftreten der verschiedenen Mitarbeiter:innen zu erschaffen.

#### Frustrationen

- Das nicht einheitliche erstellen der Lebensläufe.
- Ein dedizierter Speicher für alle Lebensläufe der Mitarbeiter:innen

#### Profil

Lena, 36 Jahre alt, von Beruf Beraterin, hat durch ihr Kind gelernt, wie wichtig es ist, strukturiert an eine Sache heranzugehen.

In ihrer Firma fehlt ihr leider jegliche Einheitlichkeit oder Struktur bei einem Lebenslauf. Sie wünsche sich, dass es eine firmeninterne Software für dieses Problem geben würde. Ebenfalls würde sie die Funktion, alle Lebensläufe von allen Mitarbeiter:innen zu sehen sehr freuen, um Inspiration für ihre Lebensläufe zu sammeln.

#### Motivationen

Initiative	70%
Angst	10%
Reife	80%
Energie	30%
Sozial	60%

Abb. 2 Persona: Lena Schmidt, 36 Jahre, Beraterin

## 1.3 Probleme

Natürlich kommen bei neuen Herausforderungen neue Probleme auf. Erst recht, wenn man mit neuen Technologien arbeiten möchte, um sich selbst auszuprobieren. Aber auch Produktanforderungen können zu neuen Hindernissen führen, weil man sich mit Teilgebieten noch nicht auseinandergesetzt hat.

Folgende Probleme sind Herausforderungen für unsere Entwicklung:

### **Login:**

Erstellung eines kompatiblen Logins mit der KeyCloak open source Software.

### **CV Erstellung:**

Die größte Herausforderung, welcher wir uns noch stellen müssen, ist die Erstellung eines CV, welche dann auch als PDF downloadbar ist.

### **Die Technologie MERN Stack:**

Die von uns gewählte Technologie stellt sich ebenso als Herausforderung heraus. Das Team muss sich erst hiermit zurechtfinden und lernen, wie man damit umgeht. Hierzu gehört auch der Umgang mit einem neuem Datenbank Design, da MongoDB eine Datenbank ohne SQL ist und wir dies nicht in einer Vorlesung behandelt haben. Jedoch bringen wir Ehrgeiz und ein wenig Erfahrung mit, wodurch diese Hürde mit Sicherheit gemeistert werden kann.

## 1.4 Eigenschaften

Bei der Software handelt es sich nicht um eine Desktop-App, sondern um eine Webapplikation, welche über das Internet aufgerufen wird. Hierbei ist das Mobile First Prinzip zu beachten, sodass die Seite vom Computer sowie vom Handy unterwegs aufgerufen werden kann.

Durch das MERN Stack nehmen wir die MongoDB Datenbank, worüber wir alle unsere Daten speichern und bequem über Web Requests verwalten können. Es gibt als „Überwachungs-Rolle“ den Admin bzw. mehrere Administratoren. Diese besitzen mehrere Features, wie die Erstellung neuer Accounts oder auch die Verwaltung und Bearbeitungsmöglichkeit von allen Projekten, Abschlüsse usw..

## 1.5 Alleinstellungsmerkmal

Die Q-CV Software zeichnet sich in erster Linie an den gewünschten Merkmalen von QUANTO Solutions aus. Hierbei fängt es bei dem Logo und der Farbwahl des Designs an und hört bei den extra Features, wie die integrierte Projektverwaltung, auf.



## 2. User Stories

---

Für die User Stories haben wir ein Konzept überlegt, um sie richtig in das Kanban Board einzubinden und produktiv nutzen zu können. Hierbei gibt es dann eine User-Story, welche ein Problem etc. allgemein beschreibt. Darunter liegen weitere User Stories, mit welchen man letztendlich arbeitet. In den folgenden Unterkapiteln listen wir diese auf.

### 2.1 Einloggen bei Q-CV

Als ein Mitarbeiter oder Administrator möchte ich mich in die Software einloggen zu können um Zugriff auf die Funktionen passend zu meiner Rolle zu bekommen.

- Als ein Mitarbeiter oder Admin möchte ich mich via KeyCloak in der Software anmelden.
- Als ein Mitarbeiter oder Admin möchte ich eine grafische Oberfläche für den Login.

### 2.2 Skills verwalten

Als ein Mitarbeiter oder Admin möchte ich die Möglichkeit haben Skills in einem öffentlichen Speicher zu erstellen, bearbeiten oder auch zu löschen.

- Als ein Mitarbeiter oder Admin möchte ich eine API, welche mir neue Skills in der Datenbank speichert.
- Als ein Mitarbeiter oder Admin möchte ich eine grafische Oberfläche für die Erstellung neuer Skills.
- Als ein Admin möchte ich eine API, welche Skills auf der Datenbank aktualisiert.
- Als ein Admin möchte ich eine grafische Oberfläche zum Updaten der Skills.
- Als ein Admin möchte ich eine API welche Skills auf der Datenbank löscht.
- Als ein Admin möchte ich eine grafische Oberfläche für das Löschen von Skills.

## 2.3 Projekte verwalten

Als ein Mitarbeiter oder Administrator möchte ich in einem öffentlichen Speicher Projekte anlegen, bearbeiten oder löschen

- Als ein Mitarbeiter oder Admin möchte ich eine API, welche neue Projekte in der Datenbank anlegt
- Als ein Mitarbeiter oder Admin möchte ich eine grafische Oberfläche für das Erstellen neuer Projekte.
- Als ein Admin möchte ich eine API für das Updaten von Projekten auf der Datenbank.
- Als ein Admin möchte ich eine grafische Oberfläche für das Updaten von Projekten.
- Als ein Admin möchte ich eine API welche Projekte auf der Datenbank löscht.
- Als ein Admin möchte ich eine grafische Oberfläche für das Löschen von Projekten
- Als ein Admin möchte ich Projekte verwalten, Mitarbeiter und Skills hinzufügen.

## 2.4 Erstellen von verschiedenen CV's

Als ein Mitarbeiter oder Administrator möchte ich mehrere CV's erstellen, um verschiedene Kundenwünsche zu erfüllen.

- Als ein Mitarbeiter oder Admin möchte ich eine API, zum Erstellen von einem CV in der Datenbank.
- Als ein Mitarbeiter oder Admin möchte ich eine grafische Oberfläche für das Erstellen von CV's.
- Als ein Mitarbeiter oder Admin möchte ich eine API, um einen CV zu bearbeiten und auf der Datenbank zu updaten.
- Als ein Mitarbeiter oder Admin möchte ich eine grafische Oberfläche zum Aktualisieren des CV.

## 2.5 Downloaden/Exportieren von CV's

Als ein Mitarbeiter oder Admin möchte ich einen CV als PDF oder docx Dokument herunterladen/exportieren.

- Als ein Mitarbeiter oder Admin möchte ich eine API, mit welcher ich alle Daten erhalten und herunterladen kann.
- Als ein Mitarbeiter oder Admin möchte ich die Möglichkeit haben, eine Datei zu erstellen.

## 2.6 Stammdaten aktualisieren

Als ein Mitarbeiter oder Admin möchte ich meine Stammdaten aktualisieren können, um meine CV-Daten zu vervollständigen.

- Als ein Mitarbeiter oder Admin möchte ich eine API, welche die neuen Daten in der Datenbank anlegen.
- Als ein Mitarbeiter oder Admin möchte ich eine API, welche Updates auf die Datenbank schreiben.
- Als ein Mitarbeiter oder Admin möchte ich eine grafische Oberfläche für das Verwalten meiner Stammdaten.

## 2.7 Bildung verwalten

Als ein Mitarbeiter oder Administrator möchte ich meine Daten zu meinem Bildungsweg aktuell halten, um meine CV-Daten aktuell zu halten.

- Als ein Mitarbeiter oder Administrator möchte ich eine API, um neue Einträge zu meiner Karriere in der Datenbank zu erstellen.
- Als ein Mitarbeiter oder Administrator möchte ich eine API, um Einträge zu meiner Karriere in der Datenbank zu aktualisieren.
- Als ein Mitarbeiter oder Administrator möchte ich eine GUI, um meine Daten zu managen.

## 2.8 Karriere verwalten

Als ein Mitarbeiter oder Admin möchte ich meine Karrieredaten aktuell halten, um meine CV-Daten vollständig zu haben.

- Als ein Mitarbeiter oder Admin möchte ich eine API, um meine Karriere-Daten in der Datenbank abzuspeichern.
- Als ein Mitarbeiter oder Admin möchte ich eine API, um meine Karriere-Daten zu aktualisieren.
- Als ein Mitarbeiter oder Administrator möchte ich eine GUI, um meine Karriere-Daten zu bearbeiten

## 2.9 Benutzer erstellen

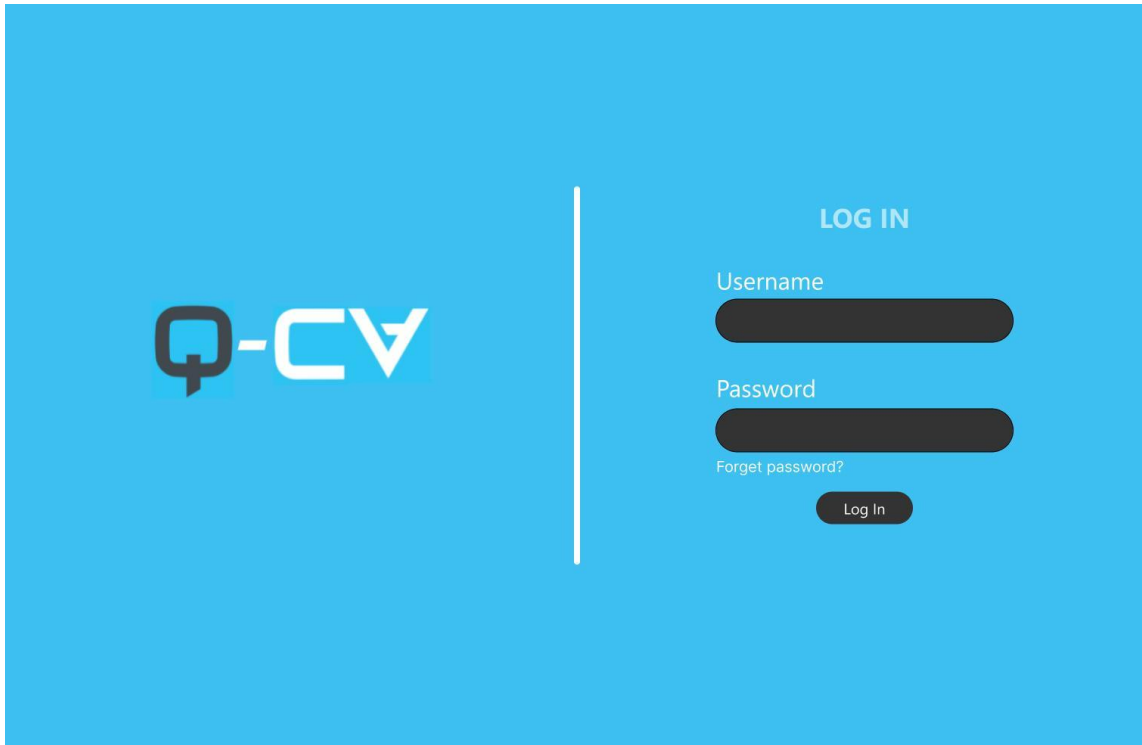
Als ein Administrator möchte ich Mitarbeiter oder Admins anlegen, verwalten und löschen können.

- Als ein Admin möchte ich eine API zum Erstellen von User.
- Als ein Admin möchte ich eine grafische Oberfläche für das Erstellen von User.
- Als ein Admin möchte ich eine API für das Löschen von User.
- Als ein Admin möchte ich eine grafische Oberfläche zum Löschen von User
- Als ein Admin möchte ich eine API zum Aktualisieren von User.
- Als ein Admin möchte ich eine grafische Oberfläche zum Aktualisieren der User.
- Als ein Admin möchte ich eine grafische Oberfläche für einen Überblick der User.

## 3. User Interface

---

### 3.1 Prototyp



*Abb. 3 Login Bildschirm: Benutzer und Administratoren können sich über diese Seite in der Software anmelden.*

- Dashboard
- User Data
- Education
- Career
- IT-Skills
- Projects
- + Create CV
- + Create User

Logged in as Jhon Doe  
Log out

Name	Date	Title	Manage	Tags
Christoph Merck	12.04.22	Computer Engineering		SAP, SAP-ERP, S/4HANA
Leto Ziegler	10.04.22	SAP Neuerungen		SAP-ERP
Marc Starzmann	09.03.22	SAP Erfolge		S/4HANA
Kevin Lee	24.12.21	QUANTO-Project		SAP, S/4HANA
Christoph Merck	01.01.01	Extern Project		SAP, SAP-ERP

**Abb. 4 Dashboard:** Hier kann ein Admin oder Benutzer alle erstellten Lebensläufe einsehen.

- Dashboard
- User Data
- Education
- Career
- IT-Skills
- Projects
- + Create CV
- + Create User

Logged in as Jhon Doe  
Log out

First Name

Last Name

Email

Linguistic

Qualifications

Short Profile

**Abb. 5 User Data:** Benutzer können ihre Profilinformationen hier ändern.

- Dashboard
- User Data
- Education
- Career
- IT-Skills
- Projects
- + Create CV
- + Create User

Logged in as  
 Jhon Doe  
 Log out

## Education



Hochschule Esslingen - University of Applied Sciences  
 Informatik (Computer Science)  
 Mar 2019 - July 2023



The University of Alabama  
 Bachelor's degree, Computer Science  
 2017 - 2018

New education:

School

Degree

Field of study

Start date



End date



Abb. 6 Education: Benutzer können Karten zu ihrer Ausbildung hinzufügen oder entfernen.

- Dashboard
- User Data
- Education
- Career
- IT-Skills
- Projects
- + Create CV
- + Create User

Logged in as  
 Jhon Doe  
 Log out

## Career



Software Developer  
 TeamViewer - Internship  
 August 2022 - December 2022

- Revamped the user authentication process of company's chat app's using API "Hyperintegration" software, improving overall efficiency and improving codebase unification.
- Prototyped API and redirected routes between former and current backends using Kong APIM.
- Identified and prioritized API calls of company's chat application.
- Completed other non-technical tasks as per instructions, eg. interview/document translation, product guide reviewing.

New experience:

Title

Employment type

Company

Location

Start date



End date



Abb. 7 Career: Benutzer können Karten zu ihrer bisherigen Arbeitserfahrung hinzufügen oder entfernen.

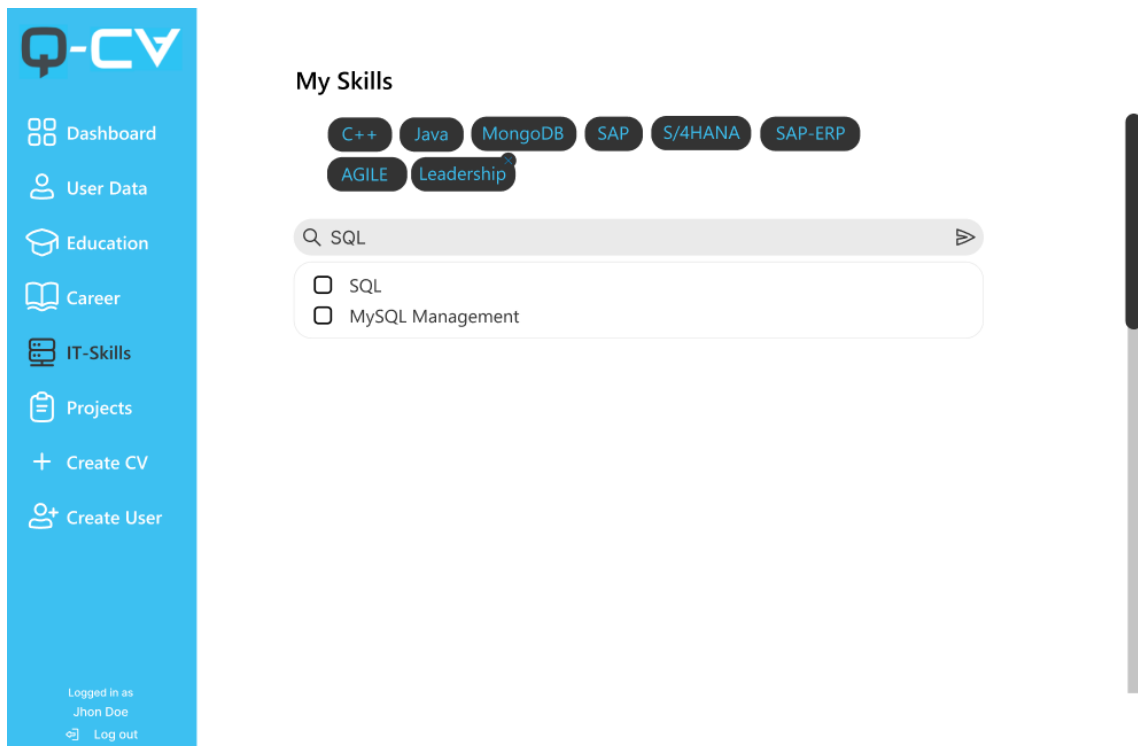


Abb. 8 IT-Skills: Benutzer können Fähigkeiten einer Bibliothek von Tags zu ihrem Profil hinzufügen oder daraus entfernen.

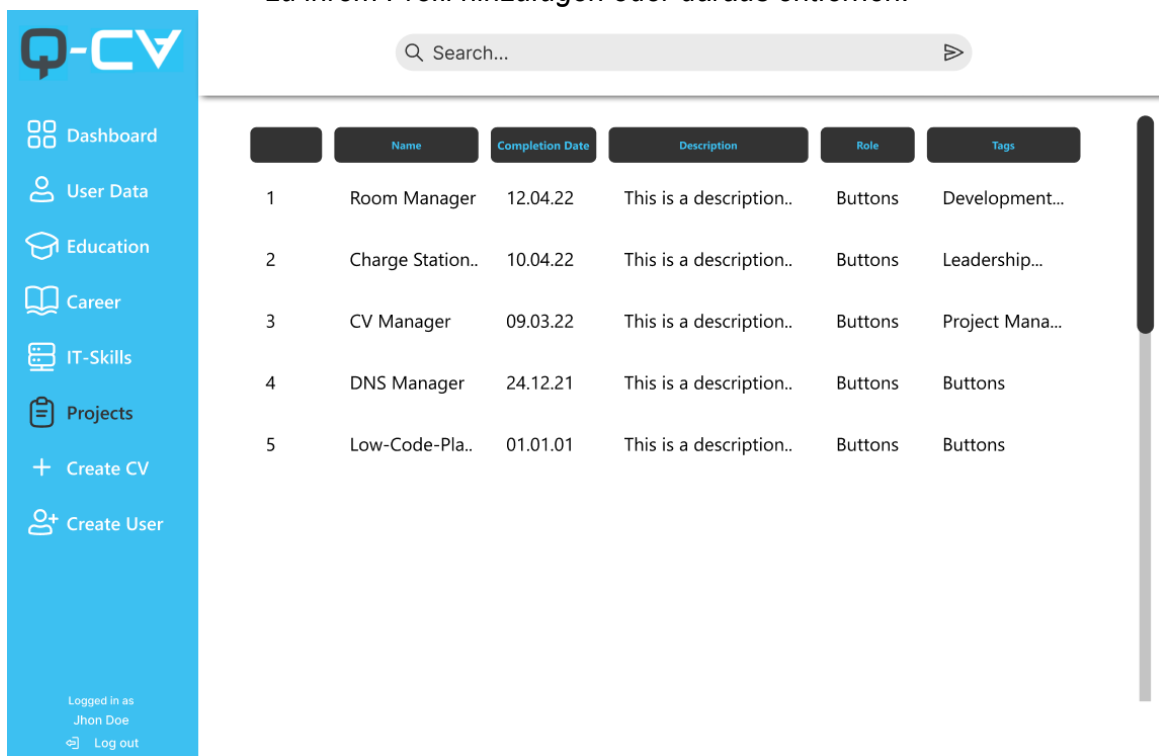


Abb. 9 Projects: Benutzer können eine Liste von Projekten anzeigen, die sie ihrem CV hinzufügen können.



## 3.2 Usability

Um unsere Applikation für jegliche Benutzer zugänglich zu machen und keine beeinträchtigten Personen auszuschließen, haben wir diese Kriterien ausgearbeitet:

### **Einfache Navigation:**

Mit einer statischen Seitenleiste, über die Benutzer/Administratoren einfach zu wichtigen Seiten navigieren können.

### **Ästhetik:**

Helle und thematisch einheitliche Farben mit einem hohen Kontrast, um die Anwendung einfach und beruhigend zu gestalten.

### **Dargestellte Informationen (Hick's Law):**

Je mehr Optionen präsentiert werden, desto länger dauert es, Entscheidungen zu treffen. Die Benutzer werden nicht mit viel Text überlastet, die Funktionalität der Symbole ist selbsterklärend und wird häufig in anderen bestehenden Webseiten verwendet (z. B. Plus- und Stift-Symbole zum Hinzufügen bzw. Bearbeiten).

### **Feedback:**

Benachrichtigungen werden verwendet, um dem Benutzer häufig Änderungen mitzuteilen (z. B. Änderungen speichern, Sortierkriterien ändern).

## 4. Technisches Konzept

---

### 4.1 Frameworks

In den folgenden Unterkapiteln werden ein paar Frameworks, die wir besonders häufig verwenden und essenziell wichtig für unser Projekt sind erklärt. Eine genaue Auflistung der verwendeten Frameworks und Libraries ist in Kapitel 7. zu sehen.

#### 4.1.1 React

Für das Frontend verwenden wir das Open Source React Framework. Mit diesem JavaScript Framework werden dynamische Client-Side Anwendungen in JavaScript geschrieben und in HTML übersetzt. Mit React kann man komplexe Interfaces mit simplen Komponenten erstellen, diese auch wieder verwenden und mit Daten auf dem Backend-Server verbinden. Das bedeutet, dass man mit wenig Code viel erreichen kann, was dabei hilft einen guten Überblick über größere Projekte zu behalten.

#### 4.1.2 Express

Für das Backend verwenden wir das Express-Framework, welches wir auf einem NodeJS Server laufen lassen. Express ist sehr effektiv, um URL Routing zu betreiben. Es leitet die HTTP Anfragen wie, GET's oder POST's von dem React Frontend zu dem NodeJS Backend, um da die Daten zu verarbeiten und auf die MongoDB Datenbank, wenn nötig, zuzugreifen.

#### 4.1.3 NodeJS

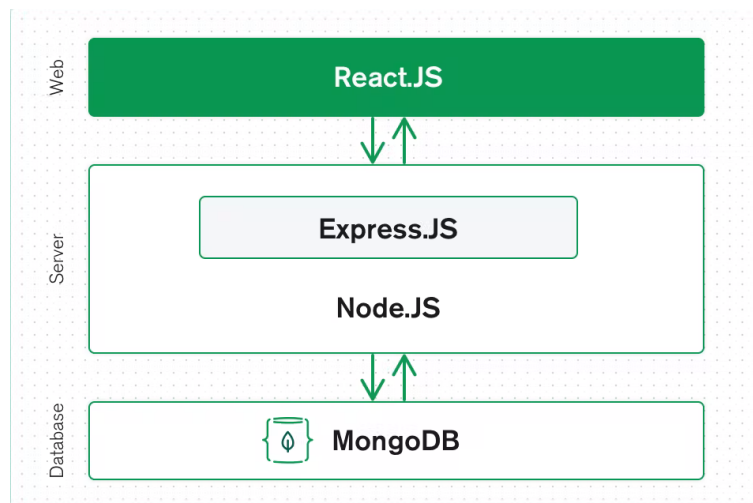
NodeJS ist eine moderne, weit verbreitete und schnelle Open-Source JavaScript Runtime-Umgebung. Wir verwenden NodeJS als Backend, da es uns ermöglicht JavaScript auf unserem Server, ohne Browser laufen zu lassen und so unsere Kommunikation mit unserer Datenbank ermöglicht.

#### 4.1.4 Mongoose

Um mit MongoDB zu kommunizieren, verwenden wir die Mongoose Bibliothek, um die Daten für die MongoDB Datenbank zu modellieren. Diese Modellierung dient uns dazu, dass wir Daten in einem bestimmten Format in der Datenbank ablegen und diese auch wieder interpretieren können, wenn wir sie uns wiederholen

## 4.2 Softwarearchitektur

Für unsere Architektur orientieren wir uns an dem sogenannten MERN-Stack. MERN steht dabei für MongoDB, Express, React und NodeJS. Der MERN-Stack ist eine 3 Stufen Architektur, welche aus dem Frontend, Backend und der Datenbank besteht. React ist für das Frontend konzipiert, NodeJS und Express für das Backend und MongoDB ist die Datenbank. Express ist für das Routing zwischen React und NodeJS zuständig, damit NodeJS die Datenbankzugriffe machen kann.



*Abb. 10 MERN Struktur: Web (Frontend), Server (Backend)  
Database (Datenbank)*

## Client Diagramm

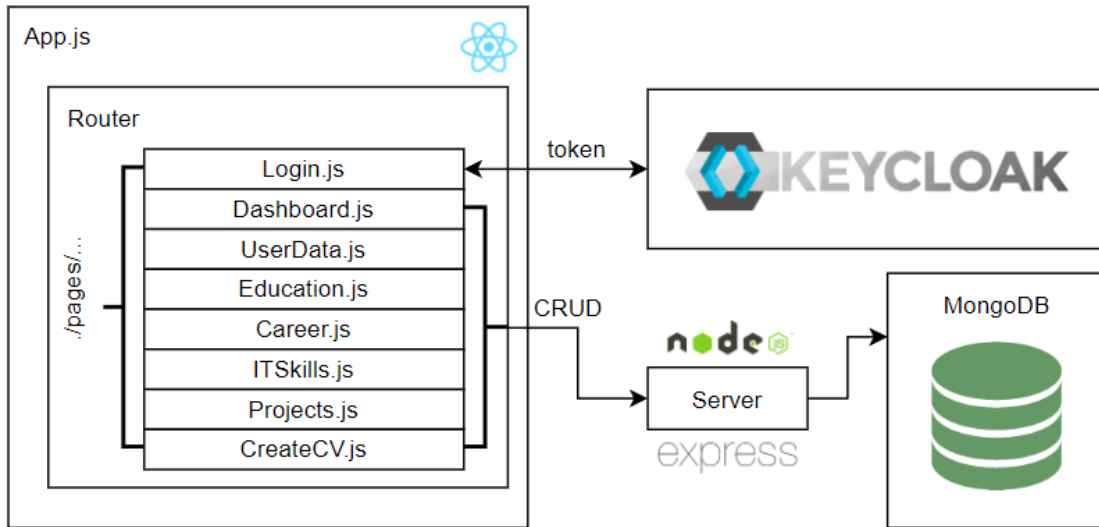


Abb. 11 Client Diagramm: Beschreibt den Aufbau unseres Frontends

## Server Diagramm

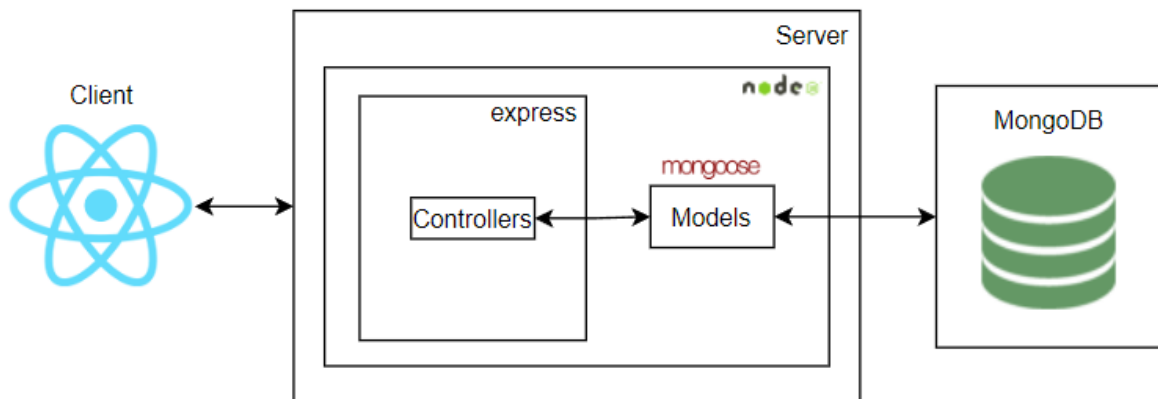


Abb. 12 Server Diagramm: Beschreibt den Aufbau unseres Backends.

## 4.3 Datenbank

Als Datenbank verwenden wir eine MongoDB Datenbank. MongoDB speichert Daten im JSON-Format. MongoDB ist gut darin, sich für große Datenmengen skalieren zu lassen. Auch ist MongoDB für die Benutzung mit JavaScript optimiert. Das Aufsetzen der Datenbank ist unkompliziert und das Verbinden dieser mit unserem Backend ist auch einfach. Diese Eigenschaften machen die MongoDB-Datenbank zu einem hervorragenden Kandidaten für unser Projekt.

Im Folgenden haben wir ein ER Diagramm zur Veranschaulichung unserer genutzten Daten in der MongoDB Datenbank.

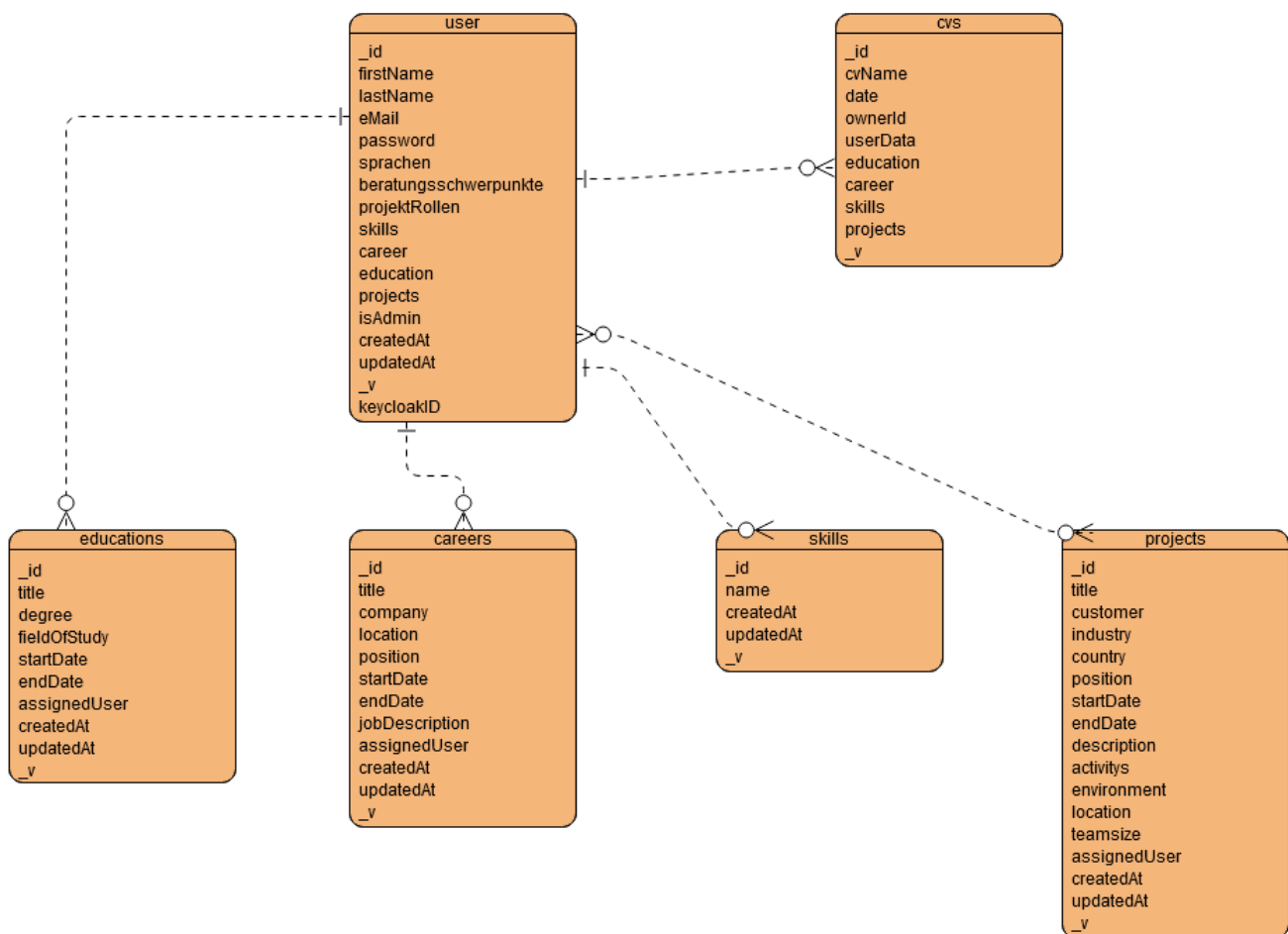


Abb. 13 ERM Diagramm: Unsere MongoDB Datenbank Struktur.

## 5. Projektumsetzung

---

### 5.1 Arbeitsaufteilung

Die übliche Arbeitsteilung liegt bei solchen Projekten zwischen Frontend und Backend, bei welchen man sich entweder um das GUI kümmert oder wie bei uns um die serverseitigen Funktionen. Wir haben uns jedoch dafür entschieden, das Projekt featurebasiert abzuarbeiten. Das bedeutet, dass die Aufgaben als User Storys so verfasst sind, dass sie Frontend und Backend zugleich abdecken. Dies hielten wir für sinnvoll, da so jedes Teammitglied beim MERN-Stack die Möglichkeit hatte Client und Server Funktionen zu lernen. Ein weiterer Vorteil, welcher hierdurch geboten wird, ist, dass niemand auf den anderen großartig warten muss, sondern jeder mit einer individuellen Stelle anfangen kann.

Folgende Features wurden von den jeweiligen Teammitgliedern erarbeitet:

- **Christoph Merck**
  - Die grundlegende Projektumgebung der Software (Docker, GitHub, usw.).
  - Die Navigation der Software.
  - Die Erstellung und Verwaltung von Projekten.
  - Die Erstellung und Verwaltung von Educations.
  - Die Software in der Cloud anzulegen.
  - Eine Keycloak API zur Erstellung und Verwaltung von Nutzern.
- **Leto Ziegler**
  - Die Erstellung und Verwaltung von IT-Skills.
  - Die Erstellung und Verwaltung von Careers.
  - Die Erstellung und Verwaltung des Dashboards.
  - Die Erstellung und Verwaltung von CV's.
- **Marc Starzmann**
  - Die Verwaltung der User Data.
  - Die Erstellung und Verwaltung von Benutzern.
- **Navamin Leelarburanathanakoon**
  - Der Login mittels Keycloak.
  - Die Erstellung und Verwaltung des Dashboards.
  - Die Erstellung und Verwaltung von CV's.

## 5.2 Management

Wir hatten jeden Dienstag Meetings mit unseren Projektleitern, wo wir entweder über die Arbeit vergangener Woche, oder auch nach einem Sprint unser Sprint-Review durchgesprochen haben. Dabei haben wir meist zu Beginn des Meetings unseren aktuellen Arbeitsstand präsentiert, wo jeder seine Arbeit einzeln präsentieren und auch dazu Fragen stellen oder Feedback erhalten konnte. Die Meetings wurden von uns über Google Docs dokumentiert, sodass wir immer einen Überblick haben, welche Punkte für die kommende Woche wichtig sind, oder was geändert werden muss.

Die interne Organisation in der Gruppe lief größtenteils über WhatsApp, worüber wir uns spontan austauschen konnten und Termine vereinbart haben. Ebenfalls besaßen wir einen Discord-Server, auf welchem wir uns getroffen haben, um gemeinsam an Aufgaben zu arbeiten. Festgelegte interne Meetings gab es nur je nach Bedarf, da wir, falls es an einem Tag nur eine Frage gab, über WhatsApp oder Discord einzeln kommuniziert haben.

Die Aufgabenverteilung haben wir über Github geregelt, worüber wir unseren Code gemanagt, und zusätzlich über ein Projekt Board die Aufgaben und deren Fortschritt verwaltet haben. So hatten wir einen Überblick über die Arbeit und die Möglichkeit, den geschriebenen Code mit dem Hauptcode zu verbinden. Der Hauptcode ist im Grunde auf zwei Zweige (sogenannte Branches) aufgeteilt. Dabei haben wir einmal die Main Branch worauf der fertige Code abgelegt wird, welcher nicht mehr geändert wird und die Dev Branch auf welcher noch gearbeitet werden darf, falls es noch kleine Probleme gibt. Für Merge-Anfragen musste noch ein weiteres Teammitglied hinzugezogen werden, welches den Code erneut überfliegt und die Anfrage bestätigen musste. So haben wir verhindert, dass Fehler nicht einfach in den Hauptcode gelangen.

## 6. Reflektion

---

### 6.1 Teamarbeit

Das Team allgemein hat effektiv zusammengearbeitet. Es gab niemanden welcher sich aus Entscheidungen und Meetings rausgehalten hat und jeder hat seinen Teil beigetragen. Auch bei Fragen gab es stetig Antworten und falls sich jemand schon besser auskannte haben wir untereinander geholfen durch Tipps, Code Review oder auch durch Pair programming. Zusätzlich zur Teamaufstellung kann man rückblickend sagen, dass die featurebasierte Aufteilung gut war, da es so auch erst recht zwischen Back- und Frontend keine Missverständnisse gab. Diese wurden aber auch durch viel Einheitlichkeit zwischen unseren Features verhindert.

### 6.2 Projektmanagement

Natürlich ist die erste Planung eines Projekts auch bezüglich Prototypen nie 100 % identisch mit dem Endergebnis. Wir haben zum Anfang einen grundlegenden Prototyp erstellt und auch im allgemeinen Punkte wie Requirements und User Storys aufgeschrieben. An diesen haben wir relativ viel Zeit verbracht und lange überarbeitet, bis es dann an die Programmierung ging. Dabei wurden wir von Meeting zu Meeting immer genauer und spezifischer, was die einzelnen Features anging. Somit wurde das simple Anfangsmodell in ein ausführlicheres, komplizierteres und umfangreicheres Modell entwickelt. Dadurch, dass wir wenig Erfahrung mit solchen kundenbezogenen Projekten hatten und wir nach jedem Meeting in der Planung etwas verändern mussten, hatten wir eine etwas längere Planungsphase. Das bedeutet, wir haben relativ spät mit Programmieren angefangen. Der Aufwand, welcher als Gruppe zu Beginn geschätzt wurde, ist damit eindeutig übertroffen worden.



## 6.3 Lernfortschritt

Dadurch, dass wir uns in diesem Projekt an eine neue Technologie, dem MERN-Stack, gewagt haben, hat jeder aus dem Team etwas daraus lernen können. Zum einen die Full Stack Entwicklung, da Front- und Backend von jedem Mitglied bearbeitet wurden. Aber auch einzelne Frameworks, wie Material-UI, was als gute Ergänzung zu dem vorherigen Vorlesungsstoff diente, haben wir kennen und schätzen gelernt. Die größte Erfahrung, welche jedoch gemacht wurde, ist die reale Umsetzung eines Kundenprojekts. Alles musste hierbei gut mit dem Kunden durchgeplant und für den Kunden realisiert werden. Unter anderem konnte man dabei auch in neue Stationen wie dem Code Review eintauchen und auch einiges an Erfahrung mitnehmen. Dabei wichtig war, dass es immer konstruktive Kritik und Ratschläge von den Kunden/Coaches gab und so einiges an den bisher erlernten Fähigkeiten verbessert werden konnte.

## 7. Lizenzen

---

Framework / Library	Lizenz	Version	Verwendung
fortawesome	MIT	6.1.1	Client
material-ui	MIT	4.12.4	Client
react-keycloak	MIT	3.4.0	Client
axios	MIT	0.26.1	Client
keycloak-js	Apache-2.0	18.0.0	Client
bootstrap	MIT	5.1.3	Client
react	MIT	18.0.0	Client
react-dom	MIT	18.0.0	Client
react-router-dom	MIT	6.3.0	Client
react-scripts	MIT	4.0.3	Client
react-bootstrap	MIT	2.4.0	Client
uuid	MIT	8.3.2	Client
react-beautiful-dnd	Apache-2.0	13.1.0	Client
body-parser	MIT	1.20.0	Server
cors	MIT	2.8.5	Server
dotenv	BSD-2-Clause	16.0.0	Server
express	MIT	4.17.3	Server
express-async-handler	MIT	1.2.0	Server
mongoose	MIT	6.2.10	Server
node-fetch	MIT	3.2.6	Server
nodemon	MIT	2.0.15	Server

## 8. Anleitungen

---

### 8.1 Installationshandbuch

1. Klonen Sie das GitHub Repository.

```
foo@bar:~$ git clone https://github.com/FreakyPlays/Q-CV.git
```

2. Installieren Sie Docker, Docker-Compose und WSL2.

<https://www.docker.com/products/docker-desktop/>

3. Falls ihr Betriebssystem kein make Befehl mitliefert, installieren Sie ein Drittanbieter make wie GNUmake.

<https://www.gnu.org/software/make/>

4. Falls Sie die vorgefertigten .env und .json Dateien nicht besitzen, fahren Sie mit Schritt 8. fort.

5. Füllen Sie das Projekt mit den vorgefertigten Dateien

- a. Legen Sie die .env Dateien in den richtigen Ordnern ab (/config, /client/config und /server/config). Um herauszufinden welche .env Datei in welchen Ordner gehört, vergleichen Sie die Variablen in den .env\_example Dateien mit den .env Dateien in den entsprechenden Ordnern.
- b. Überschreiben Sie die master-realm.json und die Q-CV-realm.json Dateien in dem /realm-config Ordner.

6. Vergewissern Sie sich, dass Sie sich im Wurzelverzeichnis des Repositorys befinden. Anschließend bauen Sie den Container und führen diesen aus.

```
foo@bar:~/Q-CV$ make build-[dev|production]
```

```
foo@bar:~/Q-CV$ make run-[dev|production]
```

7. Fahren Sie mit Schritt 14. Fort.

8. Erstellen Sie in den config Ordnern die [dev|production].env Dateien mit den Daten beschrieben in den .env\_example Dateien.

- a. Die KC\_CLIENT\_SECRET Variable in der /server/config/[dev|production].env wird später zugewiesen.

9. Wechseln Sie in den Q-CV Ordner und bauen Sie den Develope oder Production Container.

```
foo@bar:~$ cd Q-CV/
```

```
foo@bar:~/Q-CV$ make build-[dev|production]
```

10. Nach dem Bauen des Containers führen Sie diesen Container aus.

```
foo@bar:~/Q-CV$ make run-[dev|production]
```

11. Rufen Sie den Keycloak Server auf, indem Sie die Adresse und Port aus den .env Dateien in ihrem Browser eingeben.

Im Normalfall: <http://127.0.0.1:8080>

12. Speichern Sie das neue Client-Secret in ihren .env Dateien.
  - a. Loggen Sie sich mit den in der .env Datei festgelegten Benutzerdaten in der Admin-Konsole ein. ( `/server/config/[dev|production].env` )
  - b. Versichern Sie sich, dass Sie sich im `Master` Realm befinden, indem Sie in der linken oberen Ecke den Namen abgleichen. Falls sich hier ein anderer Name befindet, können Sie durch klicken auf den Namen das Kontextmenü aufrufen und den Realm wechseln.
  - c. Gehen Sie zur `Clients-Sektion` mittels der Navigation auf der linken Seite.
  - d. Wählen Sie in der Liste die `admin-cli` aus.
  - e. Navigieren Sie zu dem Tab `Credentials` der `admin-cli`.
  - f. Klicken Sie auf `Regenerate Password`.
  - g. Kopieren Sie den neu generierten Code aus dem Textfeld.
  - h. Speichern Sie das neue Client-Secret in der `/server/config/[dev|production].env` Datei in der `KC_CLIENT_SECRET` Variable.
13. Nachdem Sie das neue Client-Secret in ihrer .env Datei gespeichert haben, starten Sie ihre Container neu.  
`foo@bar:~/Q-CV$ make stop-[dev|production]`  
`foo@bar:~/Q-CV$ make run-[dev|production]`
14. Rufen Sie erneut den Keycloak Server mit ihren in den .env Datei definierten Adresse und Port auf.  
Im Normalfall: `http://127.0.0.1:8080`
15. Erstellen Sie einen neuen temporären Administrator.
  - a. Loggen Sie sich mit dem Username und Passwort aus der `/server/config/[dev|production].env` Datei in der Admin-Konsole ein.
  - b. Versichern Sie sich, dass Sie sich im `Q-CV` Realm befinden, indem Sie in der linken oberen Ecke den Namen abgleichen. Falls sich hier ein anderer Name befindet, können Sie durch klicken auf den Namen das Kontextmenü aufrufen und den Realm wechseln.
  - c. Navigieren Sie zur User-Sektion über die seitliche Navigation.
  - d. Klicken Sie den `Add User` Knopf.
  - e. Füllen Sie die verschiedenen Felder aus und fügen Sie den Nutzer zur `/Admins` Gruppe hinzu.
  - f. Wechseln Sie zu dem `Credentials` Tab und geben Sie ein Passwort an.
16. Rufen Sie die Software mittels der von Ihnen in der .env Datei definierten Adresse und Port auf.  
Im Normalfall: `http://127.0.0.1:3000`
17. Loggen Sie sich in die Software mit dem temporären Administrator ein.
18. Erstellen Sie auf der Users-Seite einen oder mehrere neue Nutzer für die Software. Diese neu angelegten Nutzer können die Software im vollen Umfang nutzen. Ihr temporärer Administrator kann nun in der Keycloak Administrator Konsole gelöscht werden.

## 8.2 Administrationshandbuch

In den folgenden Textabschnitten sind die verschiedenen Funktionen der Software erklärt. Hier gibt es meist zwei Nutzungsfälle. Zum einen gibt es den Fall mit der Rolle des Administrators oder mit der Rolle des Benutzers. Da beide Rollen dieselbe Grundfunktion besitzen, wird hier die Funktion des Benutzers beschrieben und im Anschluss die Funktion des Administrators.

### 8.2.1 Login

Auf dieser Seite kann man sich mittels der E-Mail und des zugehörigen Passworts eines Benutzers einloggen. Durch das Setzen der Checkbox am linken unteren Rand mit dem Label „Remember me“, bleibt der Benutzer für eine längere Zeit eingeloggt. Die Benutzer, die sich hier einloggen können, werden innerhalb der Software erstellt (Falls dies der erste Login ist, folgen Sie den Anweisungen des Installationshandbuchs).

### 8.2.2 Dashboard

In diesem Fenster werden die erstellten CV's aufgelistet. Man kann entweder nur seine eigenen CV's auflisten, oder alle CV's im System. Um diese Ansichten zu wechseln, klicken Sie auf das Personen-Symbol oben rechts neben dem Plus-Button. Ihre eigenen CV's können Sie bearbeiten, löschen oder neue erstellen. Das Erstellen können Sie über den Plus-Button oben rechts machen, dort werden Sie zu der Create-Page weitergeleitet. Hier können Sie dann alle Informationen in die Textfelder eintragen. Man kann in dem Bereich Education die vorher eingetragenen Daten verwalten, indem man z. B. die Textfelder bearbeitet, die Karten per Drag-and-drop verschiebt, oder irrelevante Karten über das Mülleimer-Symbol aus dem CV löscht. Über das kleine Plus-Symbol an den Karten kann man eine weitere Karte nach der Karte, an welcher man das Plus-Symbol angeklickt hat, einfügen. Dies funktioniert genau gleich mit den Career-Karten und den Projektkarten. Unten auf der Seite können Sie, mit Anklicken der Checkboxes, aus Ihren Skills die auswählen, die Sie in dem CV haben möchten. Wenn Sie ein CV bearbeiten wollen, können Sie das über das Stift-Symbol tun. Sie werden dann zu der Edit-Page weitergeleitet, die genau wie die Create Page funktioniert, nur mit dem Unterschied, dass die Seite entsprechend der Informationen im CV gefüllt ist.

### 8.2.3 User Data

In diesem Fenster werden die Daten des angemeldeten Nutzers in Textfeldern angezeigt. Dabei handelt es sich um die reinen Grunddaten und haben mit den restlichen Punkten wie Projekte etc. nichts zu tun.

Die Daten im Textfeld lassen sich verändern und durch den unteren blauen Button aktualisieren. Beachten Sie hierbei die richtige Eingabe, wenn es mehrere in einer Zeile sind.

### 8.2.4 Education

Auf der Seite Educations werden alle Bildung/Ausbildung/Studium Einträge aufgelistet. Hierbei werden alle Daten auf einer Karte visualisiert. Um einen neuen Eintrag zu erstellen, kann man in der rechten oberen Ecke auf das Plus-Symbol klicken. Dies öffnet ein Pop-up, in dem man Daten in die Textfelder eintragen kann. Hierbei ist das Feld Field of Study ein optionales Feld und muss nicht ausgefüllt werden. Sind alle Felder ausgefüllt, kann man auf den Submit Knopf am linken unteren Rand des Pop-ups klicken. Anschließend wird der neue Eintrag auf der Seite angezeigt. Will man die Informationen in dieser Karte verändern, kann man auf das Stift-Symbol am rechten oberen Rand der Karte klicken. Dies öffnet erneut ein Pop-up, indem die Daten der Karte bereits geladen sind. Nun kann man diese Daten nach Belieben bearbeiten oder auch die Field of Study hinzufügen oder entfernen. Mittels des Submit Knopfs werden diese Daten anschließend gespeichert. Möchte man jetzt eine Education löschen, gibt es ebenfalls an der rechten oberen Ecke ein Mülleimer-Symbol. Drückt man auf dieses Symbol, erscheint ein Pop-up, auf dem man den Löschvorgang bestätigen muss.

### 8.2.5 Career

In diesem Fenster wird die Karriere aufgelistet. Diese ist dabei nach dem Datum sortiert, an welchen die einzelnen Bestandteile stattgefunden haben. Dabei wird das oberste Element als das aktuellste genommen. Man kann diese „Karriere-Karten“ auch bearbeiten, indem man auf das Stift-Symbol klickt. Ein neues Item kann man durch das Plus-Symbol in der rechten oberen Ecke erstellen. Wenn man ein neues Item erstellt oder eines bearbeiten möchte, erscheint ein Pop-up, in welchem man die gewünschten Informationen per Texteingabe eingeben kann. Indem man bei dem Pop-up dann Enter oder auf den Submit-Button drückt, werden die entsprechenden Änderungen vorgenommen.

### 8.2.6 IT-Skills

Auf diese Seite können Sie zwei Felder sehen. In dem oberen Feld können Sie ihre eigenen Skills sehen und, wenn Sie welche haben, über das Minus-Symbol entfernen. In dem unteren Feld sehen Sie alle Skills, die in der Datenbank existieren. Über das Plus-Symbol bei den Skills können Sie sich diese zuweisen. Wenn Sie einen Skill suchen, welcher aber nicht existiert, können Sie diesen über das Plus-Symbol oben rechts auf der Seite hinzufügen. Wenn Sie da draufklicken kommt ein Pop-up, in welchem Sie dann in eine Texteingabe den Namen des Skills hinzufügen und über den Submit-Button in die Datenbank hinzufügen. Wenn Sie Admin-Rechte haben, dann können Sie die Skills über das Stift-Symbol, genau wie beim Erstellen, über ein Pop-up bearbeiten. Skills aus der Datenbank löschen können Sie auch nur als Admin über das Mülleimer-Symbol.

### 8.2.7 Projects

Wie bereits in den vorherigen Funktionen ist es möglich auf dieser Seite die eigenen Projekte zu sehen. Jedes Projekt stellt die Informationen auf einer Karte dar. Um ein neues Projekt zu erstellen, klickt man auf das Plus-Symbol in der rechten oberen Ecke. Im Anschluss gelangt man auf die Create Project Seite. Hier kann man die Daten in die vorgegebenen Textfelder einfügen. Diese werden sofort validiert, ob eine korrekte Eingabe stattgefunden hat. Hat man nun alle notwendigen Felder ausgefüllt (Die optionalen Felder sind mit einem \* vermerkt), so kann man unterhalb auf den Submit Button drücken. Nach dem Absenden der neuen Daten gelangt man wieder zu der Projects Seite. Hier werden die neuen Projekte oben an der Liste angezeigt. Auf jeder Karte findet man oben rechts weitere Funktionen. So gibt es zu einem die Kopie Funktion. Durch Drücken des Kopier-Symbols gelangt man zu der Copy Projects Seite. Hier kann man die Daten noch einmal genauer an das neue Projekt anpassen. Durch Drücken des Stift-Symbols gelangt man auf die Edit Project Seite. Hier kann man die Daten innerhalb des gespeicherten Projektes bearbeiten und speichern. Mittels des Mülleimer-Symbols kann man das ausgewählte Projekt löschen. Damit man dies nicht aus Versehen macht, erscheint ein Pop-up, auf dem eine erneute Bestätigung notwendig ist. Möchte man sich von anderen Projekten inspirieren lassen oder von anderen das Projekt kopieren, so kann man mittels des User-/Group-Symbols im rechten oberen Bereich die Ansicht zwischen den eigenen und allen Projekten wechseln. Um mit einer großen Menge von Projekten besser umzugehen, gibt es im oberen Bereich eine Suchleiste. Mittels eingegebener Suchbegriffe in der Suchleiste wird in der Projektbeschreibung und in dem Projekttitel nach gültigen Projekten gesucht und diese angezeigt. Bei fremden Projekten kann ein normaler Nutzer die Projekte nur kopieren. Ein Administrator hingegen kann diese Projekte durch die Symbole rechts oben auch bearbeiten und löschen.

### 8.2.8 Create User

Dieses Fenster ist nur für einen eingeloggten Administrator sichtbar. Es dient zur reinen Benutzerverwaltung, worunter auch die Erstellung, Bearbeitung und das Zurücksetzen eines Passwortes gehört. Dabei werden die User in Karten dargestellt. Angezeigt werden hier der Vor- und Nachname, die Datenbank ObjektID, die E-Mail und die aktuelle Benutzerrolle. Zusätzlich sind drei Knöpfe auf der jeweiligen Karte zu finden. Der größte offensichtlichste ist unter den Daten angereichert der Passwort-Knopf. Wenn dieser betätigt wird, öffnet sich ein Pop-up, wo der Administrator die Möglichkeit hat, ein neues Passwort für den Nutzer einzustellen, welches dieser nutzen bzw. dann wieder neu verändern kann. Die anderen zwei Knöpfe sind rechts oben auf der User-Karte zu finden und sind über die Icons ansprechbar. Der Stift öffnet ein Pop-up, in dem der Admin die wichtigsten Daten des Users ändern kann, darunter Name, Vorname und E-Mail. Auf dem Mülleimer kann der Admin einen User löschen, wobei es hier eine Bestätigungsanforderung gibt. Als Letztes gibt es noch auf der Seite oben rechts einen blauen Knopf, der es ermöglicht einen User mit Namen, Vorname, E-Mail, Passwort und der Rolle zu erstellen.



## 9. Fazit

---

Das Ziel des Projekts bestand daraus, eine Software nach den Wünschen einer Firma zu entwickeln. Hierbei wurde uns die Software der Firma QUANTO Solutions aufgetragen. Diese wollten eine Software erhalten, in der Sie ihre CV Erstellung erleichtern. Im Verlauf des Projektes haben wir stetig und fleißig an dem Projekt gearbeitet und können stolz sagen, dass uns die Grundfunktionen der Software gelungen sind. So wurden alle wichtigen Bedingungen der Firma erfüllt und in der Software implementiert. Natürlich ist dies noch nicht die eine vollkommene ausgereifte Software. So sind zum Beispiel noch ein paar wenige Ideen der Firma offen und könnten implementiert werden. Hierzu zählen Features wie einen Versionsverlauf der CV's, verschiedene Word Templates zur Auswahl stellen und rollenbasierte und auf den Nutzer spezifizierte Vorschläge in der Suchleiste. Ebenfalls haben wir als Team zusätzliche Ideen für die Software bei der Entwicklung bekommen. So kam uns die Idee zu einer Einstellungsseite in der Einstellungen getroffen werden können wie Design der Software, eine Art Flagging von Projekten oder sonstigen um Admins auf Fehler innerhalb Datensätzen auffällig zu machen oder Projekte für verschiedene Nutzer gleichzeitig anzulegen. Somit kann man sagen, dass die Software definitiv noch ausbaufähig ist und noch nicht in ihrem finalen Stadium angekommen ist.