



HTL - Perg

Abteilung: Höhere Lehranstalt für Informatik

Diplomarbeit

Trizebs Reloaded - Webapplikation zur Projekt- und Zeitverwaltung

Projektteam:

Elena Friedl

Hilal Korkmaz

Betreuer:

Prof. Maria Inreiter, MSc

In Zusammenarbeit mit:

MAIT Austria GmbH

Gerald Birnzain, MSc

Eidesstattliche Erklärung

Die unterfertigten Kandidaten / Kandidatinnen haben gemäß § 34 (3) SchUG in Verbindung mit § 22 (1) Zi. 3 lit. b der Verordnung über die abschließenden Prüfungen in den berufsbildenden mittleren und höheren Schulen, BGBl. II Nr. 70 vom 24.02.2000 (Prüfungsordnung BMHS), die Ausarbeitung einer Diplomarbeit mit der umseitig angeführten Aufgabenstellung gewählt.

Die Kandidaten / Kandidatinnen nehmen zur Kenntnis, dass die Diplomarbeit in eigenständiger Weise und außerhalb des Unterrichtes zu bearbeiten und anzufertigen ist, wobei Ergebnisse des Unterrichtes mit einbezogen werden können.

Die Abgabe der Diplomarbeit hat bis spätestens 03.04.2024 beim zuständigen Betreuer / der zuständigen Betreuerin zu erfolgen.

Die Kandidaten / Kandidatinnen nehmen weiters zur Kenntnis, dass gemäß § 9 (6) der Prüfungsordnung BMHS nur der Schulleiter bis spätestens Ende des vorletzten Semesters den Abbruch einer Diplomarbeit anordnen kann, wenn diese aus nicht beim Prüfungskandidaten (bei den Prüfungskandidaten) gelegenen Gründen nicht fertiggestellt werden kann.

Kandidaten / Kandidatinnen inkl. Unterschrift:

Perg, 02. April 2024

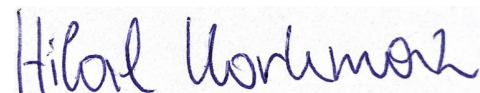
Ort, Datum



Elena Friedl

Perg, 02. April 2024

Ort, Datum



Hilal Korkmaz

Gender Erklärung

Für die bessere Lesbarkeit werden in dieser Diplomarbeit personenbezogene Bezeichnungen, die sich zugleich auf Frauen und Männer beziehen, generell nur in der Deutschen üblichen maskulinen Form angeführt.

Dies soll jedoch keinesfalls eine Geschlechtsdiskriminierung oder eine Verletzung des Gleichheitsgrundsatzes zum Ausdruck bringen.

Perg, 02. April 2024

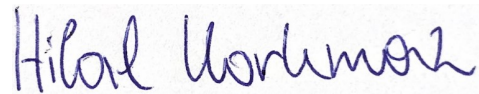
Ort, Datum



Elena Friedl

Perg, 02. April 2024

Ort, Datum



Hilal Korkmaz

Danksagung

Wir möchten uns herzlich bei allen Personen bedanken, welche diese Diplomarbeit möglich gemacht haben.

Vor allem gilt unser Dank unserer Diplomarbeitsbetreuerin Frau Prof. Maria Inreiter, MSc. Sie hat uns in allen Aspekten der Diplomarbeit stets unterstützt und hat jegliche Fragen beantwortet.

Ein weiterer Dank geht an die MAIT Austria, besonders an unseren Praktikums- und Projektbetreuer Gerald Birnzain, MSc. Er ist uns während unseres Praktikums stets Beiseite gestanden. Ohne ihn wäre das Projekt so nicht möglich gewesen.

Abschließend möchten wir uns auch bei dem Projektteam aus der HTL-Leonding, bestehend aus Florian Ster, Christoph Moosbrugger und Danijel Pejic für die Zusammenarbeit bedanken.

Kurzfassung

Trizebs Reloaded ist eine Webanwendung, welche die Anwendung zur Projektverwaltung *Trizebs* und Anwendung zur Zeit- und Aufgabenverwaltung *Task* vereinigt. Das Ziel ist, eine übersichtlichere Benutzeroberfläche zu schaffen und so die Arbeitseffizienz der Mitarbeiter zu erhöhen. Zeitgleich wird auch die Rollenverwaltung sichergestellt. Die neue Anwendung besteht aus drei Grundfunktionalitäten:

- **Projektverwaltung**
- **Aufgabenverwaltung**
- **Zeitverwaltung**

Um diese Funktionen effizient in die neue Anwendung einzubauen, wurde vor der Realisierung ein Konzept entwickelt. Dieses skizziert wie die Funktionen in Trizebs Reloaded integriert und angeordnet werden, um die Arbeitseffizienz zu steigern. Außerdem wird beschrieben, wie man bei der Vereinigung der bisherigen Anwendungen vorgeht. Die Implementierung folgt diesem Ansatz.

Das Backend wird mit Java Spring Boot entwickelt, das Frontend mit Vue.js. Die bisher genutzte Oracle Datenbank wird durch eine SQL Server Datenbank ersetzt. Die Rollen- und Rechteverwaltung, welche wichtige Bestandteile des Projektes sind, werden mit Azure Active Directory realisiert. Die Datenbankzugriffe vom Backend werden mithilfe von JPA vorgenommen. Damit man die Daten in das Frontend übermitteln kann, ist im Java Projekt eine Rest API implementiert.

Abstract

Trizebs Reloaded is a web application which combines the application for project management *Trizebs* and the application for time and task management *Task*. The goal is to create a clear user interface in order to increase the work efficiency of the employees, while simultaneously ensuring the role management. The new application consists of three main functionalities:

- **Project management**
- **Task management**
- **Time management**

To incorporate these functionalities into the new application, a concept has been created before the implementation. This concept shows how the functionalities should be integrated into Trizebs Reloaded and how they have to be arranged for the purpose of increasing the work efficiency. Additionally, it describes how the procedure of combining the two previous applications works. The implementation follows this approach.

The backend is implemented with Java Spring Boot and the frontend with Vue.js. The Oracle database, which has been used until now, is replaced by a SQL Server database. The roles and rights management, which is an important part of this project, is carried out with Azure Active Directory. The access to the database from the backend is performed with JPA. To transfer the data from the backend into the frontend, a Rest API is implemented in the Java project.

Inhaltsverzeichnis

1	Einleitung	9
1.1	Ausgangssituation	9
1.2	Zielsetzung	9
1.3	Projekthalt	9
1.3.1	Anforderungen	10
1.3.2	Strategien zur Zielerreichung	10
1.4	Projektumfeld	11
1.4.1	Projektteam	11
1.4.2	Auftraggeber	12
2	Theoretische Grundlagen und Methoden	13
2.1	Verwendete Technologien	13
2.1.1	Visualisierung	13
2.1.2	Webanwendung	17
2.1.3	Datenhaltung	20
2.1.4	Zugriffskontrolle	25
2.2	Versionsverwaltung	25
2.2.1	Azure DevOps	25
3	Konzept	27
3.1	Funktionalität	27
3.1.1	Funktionsanalyse	27
3.1.2	Use Cases	31
3.2	Vereinigung der Anwendungen	34
3.3	Design Science	34
3.4	Datenmodell	35
3.5	Architektur	36
3.6	Informationsfluss	37

4 Implementierung	38
4.1 Datenbank	38
4.1.1 Beschreibung	38
4.1.2 Datenmodell	38
4.2 Webanwendung	43
4.2.1 Rechteverwaltung	43
4.2.2 Rest-API	45
4.3 Benutzerschnittstelle	59
4.3.1 Verwaltung der Projekte	59
4.3.2 Verwaltung der Tasks	65
4.3.3 Verwaltung der Zeiterfassungen	68
5 Ergebnis	72
5.1 Integration von Trizebs	72
5.1.1 Projekte anlegen	72
5.1.2 Projekte ansehen	72
5.1.3 Unterprojekte anlegen	75
5.1.4 Unterprojekte ansehen	75
5.2 Integration von Task	76
5.2.1 Aufgabe anlegen	76
5.2.2 Aufgabe ansehen	77
5.3 Zeiterfassungen	79
5.3.1 Zeiterfassungen ansehen	79
5.3.2 Zeiterfassung starten und stoppen	80
6 Aufgabenverteilung	81
6.1 Hilal Korkmaz	81
6.2 Elena Friedl	82
7 Resümee	84
8 Ausblick	85

1 Einleitung

1.1 Ausgangssituation

Die Firma MAIT Austria nutzt für ihr Projektmanagement die beiden Anwendungen Trizebs und Task. Trizebs wird verwendet um die Projekte und deren Unterprojekte zu verwalten. Task verwaltet die Aufgaben zu einem Unterprojekt, sowie die Aufgaben der Mitarbeiter. Weiteres werden in Task noch die Zeitbuchungen der Mitarbeiter zu einer bestimmten Aufgabe verwaltet. Die Trennung der beiden Programme führt zu einem Mehraufwand bei der Synchronisation der Datenbestände. Außerdem ist das System sehr veraltet und optisch nicht sehr ansprechend. Deshalb wünschen sich die Mitarbeiter der MAIT Austria eine neue Anwendung mit einer einfach zu bedienenden und übersichtlichen Benutzeroberfläche.

1.2 Zielsetzung

Das Ziel ist die Vereinigung der Funktionalitäten beider derzeit eingesetzten Programme Trizebs und Task zu einer neuen und modernen Webanwendung. Eine zentrale Anforderung an die neue Webanwendung ist die übersichtliche Darstellung der Inhalte, die Erhöhung der Benutzerfreundlichkeit und somit die Verbesserung der Bedienbarkeit für den Benutzer. Außerdem wird eine andere Datenbanktechnologie verwendet. Der Auftraggeber fordert einen Wechsel von Oracle zu Microsoft SQL Server. Durch diese Neuentwicklung erhofft sich der Auftraggeber Zeit zu sparen und somit den Arbeitsalltag zu erleichtern.

1.3 Projektinhalt

Damit die Neuentwicklung den Bedürfnissen der Benutzer entspricht und die Effizienz im Projektmanagement verbessert wird, wurden präzise Anforderungen definiert, welche sich auf die Datenmigration, die Sicherheit, die Funktionalität und auf das Design beziehen.

1.3.1 Anforderungen

- **Benutzeroberfläche und Design:**

Die neue Benutzeroberfläche muss einfach zu verstehen und übersichtlich gestaltet sein. Um ein einheitliches Design im Sinne der Corporate Identity des Unternehmens zu gewährleisten, werden die Farben von der MAIT verwendet. Eine weitere Anforderung ist eine ansprechende Benutzeroberfläche, damit eine angenehme Benutzererfahrung sichergestellt wird.

- **Funktionalität:**

Die Funktionen von Trizebs und Task werden in der neuen Anwendung, Trizebs Reloaded, übernommen. Dies beinhaltet das Verwalten der Projekte, Unterprojekte, Aufgaben und Zeitbuchungen.

Außerdem wünscht sich der Auftraggeber die Durchführung einer Migration von einer Oracle Datenbank zu Microsoft SQL Server.

- **Sicherheit und Zugriffsverwaltung:**

Ein sehr wichtiger Aspekt in unserer neuen Anwendung ist auch die Zugriffs- und Rechteverwaltung, welche mittels Azure Active Directory realisiert wird. Diese stellt sicher, dass die einzelnen Funktionalitäten nur von Benutzern ausgeführt werden können, welche auch die notwendigen Berechtigungen besitzen.

1.3.2 Strategien zur Zielerreichung

Die erforderlichen Daten für die neue Webanwendung werden aus den beiden bestehenden Programmen Trizebs und Task abgeleitet. Darauf aufbauend wird ein relationales Datenmodell erstellt, das mithilfe der Datenbanktechnologie Microsoft SQL Server umgesetzt wird.

Die Zugriffe für die einzelnen Benutzer müssen eingeschränkt werden, da nicht

jeder auf alle Informationen Einsicht haben darf und auch keinen unbeschränkten Zugriff auf alle Funktionen besitzen darf. Auf Grund dessen besteht eine Zugriffs- und Rechteverwaltung im Hintergrund der Anwendung. Die Rollen und Rechte werden im Azure Active Directory hinterlegt.

Die Benutzeroberfläche zeigt in einem einfachen und übersichtlichen Stil die unterschiedlichen Seiten an. Zum einen gibt es eine Ansicht für alle Projekte sowie eine Detailsansicht mit allen Unterprojekten. Weiters werden alle Aufgaben eines Benutzers und alle Aufgaben eines Unterprojekts übersichtlich dargestellt. Außerdem gibt es auch eine Seite mit allen Zeitbuchungen eines Benutzers. Die Ansicht nach dem Login ist abhängig von der Rolle des Benutzers.

1.4 Projektumfeld

Neben dem Projektteam spielen weitere Faktoren eine wichtige Rolle im Projekt. Dazu gehört die Unterstützung von den Betreuern, die zur Verfügung gestellten Ressourcen und die Infrastruktur.

1.4.1 Projektteam

Das gesamte Projekt wurde unter der Leitung von Gerald Birnzain mit zwei Projektteams aus der HTL Perg und der HTL Leonding umgesetzt. Die vorliegende Arbeit beinhaltet die Arbeit von den Schülerinnen Hilal Korkmaz und Elena Friedl aus der HTL Perg. Die Aufteilung der Aufgaben im Projekt wurde wie folgt festgelegt: Hilal Korkmaz widmet sich der Datenmodellierung sowie der Neuentwicklung von Task. Elena Friedl ist verantwortlich für die Integration von Trizebs.

Das zweite Projektteam besteht aus drei Schülern der HTL Leonding. Ihre Aufgabe ist es mittels der Zugriffs- und Rechteverwaltung eine Grundlage für die weitere Entwicklung zu schaffen. Gerald Birnzain ist während des fünfwöchigen Praktikums den Schülerinnen zur Seite gestanden. Durch seine Erfahrungen

als Projektleiter und Softwareentwickler hat er den beiden Projektteams nicht nur bei Fragen zur Anwendung, sondern auch bei technischen Herausforderungen wertvolle Unterstützung geboten. Die Diplomarbeitbetreuerin ist Frau Prof. Maria Inreiter. Sie steht den Schülerinnen zur Seite beim Schreiben der Diplomarbeit und unterstützt sie bei der Organisation.

1.4.2 Auftraggeber

Der Auftraggeber ist das Softwareunternehmen Mait Austria¹. Das Unternehmen umfasst 24 Standorte in Österreich, Deutschland und der Schweiz und beschäftigt 100 Mitarbeiter in Österreich. Die Mait ist im Bereich der Digitalisierung tätig.

Das Unternehmen konzipiert Softwarelösungen für die Fertigungsbranche und vereinfacht somit das Management des gesamten Produktlebenszyklus. Außerdem entwickelt das Unternehmen auch Lösungen für das Management der Kerngeschäftsprozesse, hierzu zählen auch Vertrieb, Einkauf, Lager, Produktionsplanung und Finanzen. Ein weiteres Tätigkeitsfeld ist die Erstellung von IT-Lösungen für Managed Services, Cloud, Hardware und Infrastruktur.

¹<https://mait.at/>

2 Theoretische Grundlagen und Methoden

2.1 Verwendete Technologien

In diesem Abschnitt werden die Technologien vorgestellt, welche bei der Entwicklung des Projektes verwendet wurden.

2.1.1 Visualisierung

Vue.js ist eines der bekanntesten Java-Script Frameworks, welches 2014 von dem Singapurer Evan You entwickelt worden ist. Das Vue.js Logo ist in Abbildung 1 dargestellt. Das Wort Vue ist französisch und bedeutet Blick oder Ansicht. Diese Bezeichnung spiegelt die Funktion des Frameworks wieder, nämlich das Erstellen von Single-Page-Anwendung. [9]



Abbildung 1: Vue.js Logo, Quelle: [18]

Eine Single-Page-Anwendung ist eine Art von Webanwendung, welche sich durch das Laden von nur einer einzigen HTML-Seite auszeichnet. Die benötigten Seiten werden dann dynamisch nach Wunsch des Anwenders ausgetauscht. Das bringt, unter anderem, den Vorteil, dass die Seiten flüssiger und schneller funktionieren, da nur die benötigten Inhalte ausgetauscht werden und nicht eine ganze Seite neu laden muss.

Um eine strukturierte Entwicklung von Webanwendungen zu erstellen, besteht die Vue.js Architektur aus verschiedenen Komponenten. Die wichtigsten Komponenten der Architektur sind folgende: [27]

- **Komponenten:** Komponenten sind eigenständige und wiederverwendbare Teile einer Anwendung, welche spezielle Funktionen und Darstellungen besitzen. Sie können einfache Elemente wie Buttons oder Tabellen sein oder auch komplexere Bereiche, wie ganze Seitenlayouts oder Navigationsleisten. Eine Komponente besteht immer aus einem Template (HTML), einem Script (JavaScript) und einem Style (css) Teil.

- **Datenbindung (Data Binding):** Vue.js bietet eine bidirektionale Datenbindung zwischen der Benutzeroberfläche und den Daten. Das bedeutet, dass bei einer Änderung der Benutzeroberfläche die Änderungen auch automatisch in den Daten vorgenommen werden. Dasselbe funktioniert im Umkehrschluss. Wenn sich die Daten ändern, wird auch automatisch die Benutzeroberfläche aktualisiert.
- **Event-Handling:** Event-Handling ermöglicht es Benutzern auf Ereignisse zu reagieren und diese zu behandeln. Einige Beispiele für solche Ereignisse sind Klicks, Eingaben oder Mausbewegungen.
- **Routing:** Routing ermöglicht die Navigation zwischen den verschiedenen Ansichten der Anwendung. Dafür gibt es ein JavaScript File, in dem man alle Routen definiert. Beim Routing werden die benötigten Ansichten nur ausgetauscht und nicht vollständig neu geladen. Dies verbessert die Leistung der Anwendung.
- **Services:** Services sind in Vue.js einzelne JavaScript Files in denen Funktionen oder Objekte definiert werden, die eine spezifische Funktionalität bereitstellen und in verschiedenen Teilen einer Anwendung wiederverwendet werden können. Sie werden oft verwendet, um Daten von einer externen Quelle zu holen, die von verschiedenen Komponenten genutzt werden können.

Design Science (Entwurfswissenschaft), ist ein Forschungsansatz, bei dem wissenschaftliche und praktische Artefakte erstellt werden, um Probleme simpler lösen zu können. Ein Artefakt ist jedes Ergebnis, welches bei einem Forschungsprozess geschaffen wird. In der Software Entwicklung können verschiedene Dinge als Artefakte bezeichnet werden. [10] [13]

- Eine Softwarelösung, die ein bestimmtes Problem löst oder eine spezifische Aufgabe erleichtert.
- Ein Prototyp oder ein Mock-up, der einen Designvorschlag demonstriert oder dazu dient sie zu testen.
- Ein Framework oder eine Bibliothek, die die Entwickler nutzen können, um die Entwicklung eines Produktes zu erleichtern.

Weitere Aspekte von Design Science:

- **Einsatzbereiche:**

Design Science wird nicht nur in der Softwareentwicklung eingesetzt, sondern auch in Bereichen wie Wirtschaftsinformatik, Angewandte Informatik, Management, Ingenieurwissenschaften oder Medizin implementiert.

- **Methoden und Prinzipien:**

Einige der grundlegenden Methoden und Prinzipien sind die Entwicklung von Artefakten, die Nutzerzentrierung und die Integration von Theorie und Praxis.

- **Beispielhafte Anwendungen:**

Die Entwicklung einer intuitiven Benutzeroberfläche, um die Benutzererfahrung zu verbessern und die Effizienz zu erhöhen. Weiters könnte die Gestaltung eines medizinischen Geräts, welches neue Behandlungsmöglichkeiten oder genauere Diagnosen ermöglicht, ein gutes Beispiel sein. Außerdem könnte auch die Entwicklung eines Lehrplans für Schulen, der moderne und pädagogische Ansätze integriert, um den Erfolg von Schülern zu steigern, als Beispiel fungieren. Diese Beispiele zeigen, wie Design Science dazu beiträgt, Lösungen für Probleme zu entwickeln.

- **Beitrag zur Forschung und Praxis:**

Design Science trägt sowohl zur wissenschaftlichen Forschung als auch zur Lösung praktischer Probleme bei.

In der Forschung ist es dafür da, das Wissen zu erweitern, indem neue Designprinzipien entwickelt, bestehende Modelle verbessert und neue Erkenntnisse gewonnen werden.

In der Praxis findet man mit der Hilfe von Design Science Lösungen für reale Probleme. Die Entwicklung neuer Produkte, Dienstleistungen oder Systeme, könnte das beinhalten.

Bootstrap ist ein Open-Source CSS Framework, welches Entwicklern ermöglicht einfach und effektiv responsive Webanwendungen zu erstellen. Das Logo von Bootstrap ist in Abbildung 2 ersichtlich. Außerdem besitzt es bereits eine Sammlung an CSS-Styles für Tabellen, Icons, Buttons und vieles mehr, sowie vorgefertigte JavaScript Erweiterungen. Dies bietet eine große Unterstützung, um Anwendungen ansehnlich zu gestalten.[11] Das sind die Hauptfunktionen von Bootstrap:



Abbildung 2: Bootstrap Logo, Quelle: [2]

- **Responsives Grid-System:** Bootstrap besitzt ein Grid-System, welches aus zwölf Spalten besteht. Dieses soll die Entwickler dabei unterstützen responsive Webseiten zu erstellen, da sich dieses Grid-System automatisch an die verschiedenen Bildschirmgrößen anpasst.
- **Vorgefertigte CSS-Komponenten:** Bootstrap stellt außerdem eine große Menge an vorgefertigten CSS-Klassen für Buttons, Dropdowns, Navigationen, Formulare, Modals und vieles mehr zur Verfügung. Dies erleichtert die Arbeit, da die Entwickler die Styles der Komponenten nicht immer von Grund auf neugestalten müssen.
- **Browser Kompatibilität:** Bootstrap funktioniert auf den meisten modernen Webbrowsern problemlos. So werden Designs und Funktionen vereinfacht, die über verschiedene Browser hinweg verwendet werden.

2.1.2 Webanwendung

Webanwendungen sind Anwendungsprogramme, die nicht lokal auf dem Rechner laufen, sondern auf einem Webserver, der meist weiter entfernt liegt. Sie basieren auf dem Client-Server Prinzip. Der Client ist in den meisten Fällen der Webbrowser.[20]

Durch die Eingabe des Url in den Browser schickt man eine Anfrage an den Server. Dieser gibt die Anfrage an die Webanwendung weiter. Der Quell-Code wird geladen und zurück zum Browser geleitet. Die erzeugte Webseite ist die Benutzeroberfläche der Anwendung.

Die Architektur einer Webanwendung besteht aus der Präsentationsschicht, Logikschicht und Datenhaltungsschicht. Im Webbrowser wird die Präsentationsschicht ausgeführt. Die anderen beiden Schichten werden im Webserver durchgeführt. [21]

Client-Server-Prinzip Die Kommunikation zwischen Client und Server funktioniert grundlegend wie in Abbildung 3 ersichtlich:

1. Der Client stellt eine Verbindung zum Server her und sendet eine Anfrage. Zum Beispiel braucht der Client bestimmte Informationen.
2. Der Server empfängt diese Anfrage und bearbeitet sie, indem er etwa auf eine Datenbank zugreift.
3. Der Server sendet die Antwort auf die Anfrage zurück an den Client.[23]

Das Client-Server Prinzip wird von vielen Diensten verwendet. Beispiele sind Outlook, Drucker, Online Spiele aber auch DNS (Domain Name System) Server.

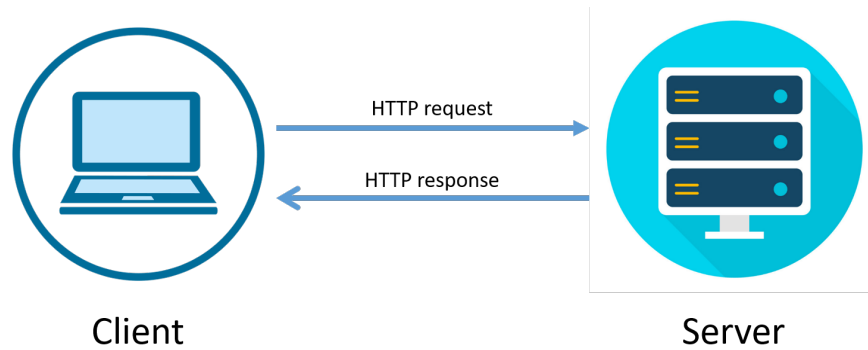


Abbildung 3: Grafische Darstellung Client-Server Prinzip, Quelle: [15]

HTTP/HTTPS HTTP ist ein Akronym für Hypertext Transfer Protocol. Das S bei HTTPS steht für Secure. Dies ist auch der einzig wesentliche Unterschied zwischen HTTP und HTTPS. HTTPS ist verschlüsselt und die Verschlüsselung wird durch Technologien wie SSL (Secure Socket Layer) und TLS (Transport Layer Secure) hergestellt. TLS ist der Nachfolger von SSL und wird heutzutage hauptsächlich für Verschlüsselung von HTTPS verwendet. HTTP und HTTPS verwenden das Client-Server Prinzip. Sie sind das Haupt-Kommunikationsprotokoll im Web. Mit einer HTTP oder HTTPS Anfrage an den Server ist es möglich Webseiten aufzurufen.[22]

Java Spring Boot ist ein Framework für Java Anwendungen. Es wurde entwickelt, um den Entwicklungsprozess für Programmierer zu vereinfachen und zu beschleunigen. Das Framework stellt Konfigurationen für eine Anwendung automatisch zu Verfügung. Wenn man eine Datenbank-Abhängigkeit in das eigene Programm einfügt, wird dies von Java Spring Boot erkannt und alle benötigten Konfigurationen werden vorgenommen. Java Spring Boot ist eine Weiterentwicklung des Spring Frameworks. Das Spring Framework besitzt eine Vielzahl an Werkzeugen und Bibliotheken. Außerdem bietet das Framework einige Funktionen an, wie zum Beispiel Dependency Injection oder auch Datenbankzugriffe. [26]

Restful APIs REST (Representational State Transfer) ist eine Architektur, die Regeln für den Datenaustausch von Webdiensten festlegt. Außerdem definiert es, wie die Webdienste aufgebaut sein müssen, um eine Kommunikation zu ermöglichen. REST baut auf den HTTP- Protokollen auf und kann somit die Ressourcen durch die Methoden GET, POST, DELETE und PUT abfragen. Mit einem Get-Request kann man beispielsweise Daten von einer Datenbank holen, und mit einem POST-Request werden Daten an den Server gesendet, wie in Abbildung 4 ersichtlich. Mithilfe von einem PUT-Request ist es möglich, bestehende Daten zu ändern und zu aktualisieren. Mit DELETE kann man Daten löschen. API (Application Programming Interface) ist die Implementierung von REST und definiert welche Ressourcen man abfragen kann. Sie ist die Schnittstelle zwischen den Softwarekomponenten. Der Unterschied liegt darin, dass REST die Regeln definiert, beispielsweise wie der Webdienst aufgebaut sein muss, und die API ermöglicht dann die Kommunikation der Komponenten. Eine Anwendung, oder auch API, die die REST-Architektur korrekt implementiert, wird als Restful bezeichnet. [25][8]

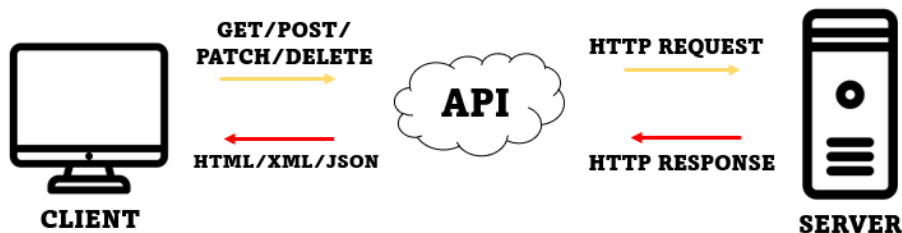


Abbildung 4: Grafische Darstellung von Rest API, Quelle: [1]

2.1.3 Datenhaltung

Datenhaltung ist die Speicherung von Daten. Unternehmen und Privatpersonen speichern ihre Daten aus unterschiedlichen Gründen, unter anderem für eine gute Verwaltung und Organisation ihrer Daten. Bei der Datenhaltung ist immer die Datenintegrität zu gewährleisten. Das bedeutet, dass die Daten korrekt und konsistent sein müssen. [3]

Ein weiterer Aspekt bezüglich der Speicherung von Daten, könnte eine Sicherung der Daten sein, damit, zum Beispiel, bei einem plötzlichen Stromausfall nicht alle Daten verloren gehen. Für den Fall, dass die Sicherung schiefgeht, braucht man einen Wiederherstellungsplan. Oft werden Backups erstellt, unter anderem auch an verschiedenen Orten, um im Notfall eine Wiederherstellung der Daten zu ermöglichen. Es ist zu beachten, dass man die eigenen Daten vor externen Gefährdungen schützen muss. Dies könnte etwa ein Hacking Angriff sein. Deshalb ist es wichtig, die Daten möglichst immer zu verschlüsseln. Daten kann man auf verschiedene Weisen speichern. Je nach Anforderungen ist es möglich, die eigenen Daten in einer Cloud, einem Dateisystem, einer relationalen Datenbank oder auch in einer No-Sql Datenbank zu speichern. [6]

Relationales Datenmodell In einem relationalen Datenmodell werden die Daten in Tabellen, Zeilen und Spalten organisiert. Die Spalten haben jeweils einen fixen Datentyp und die Attribute, die in den Spalten stehen, sind Werte des festgelegten Datentyps.[5]

In Abbildung 5 wird diese Architektur grafisch dargestellt. Um jede Zeile, also jeden Datensatz, eindeutig zu machen, gibt es einen Primärschlüssel. In den meisten Fällen ist das eine eindeutige ID.

Beziehungen zwischen den Tabellen kann man mithilfe von Fremdschlüsseln darstellen. Wenn man einen Verweis auf den Primärschlüssel von einer Tabelle erzeugt, kann man einen Fremdschlüssel und somit auch eine Beziehung herstellen.

Einige bekannte relationale Datenbanken sind MySQL, Microsoft SQL Server, Maria DB, PostgreSQL und Oracle.

Vorteile von relationalen Datenbanken sind:

- **Struktur**

Klare Struktur der Daten

- **Flexibilität**

Komplexe Abfragen können durch Verwendung der SQL-Abfragesprache ausgeführt werden.

- **Datenintegrität**

Durch Regeln wie Primär- und Fremdschlüssel, ist es einfach die Datenkonsistenz zu gewährleisten.

- **Datenunabhängigkeit**

Änderungen in der Anwendung haben keinen Einfluss auf die Datenbank.

- **Skalierbarkeit**

Bei zunehmenden Datenmengen ist es möglich eine hohe Leistung beizubehalten.

- **Transaktionen**

Transaktionen sind zusammenhängende Operationen, die gemeinsam ausgeführt werden.

Nachteile von relationalen Datenbanken:

- **Datenstruktur**

Änderungen an der Datenstruktur können komplex und aufwändig sein, vor allem bei großen Datenbanken.

- **Einschränkung bei unstrukturierten Daten**

Sie sind weniger geeignet für Speicherung von semi-strukturierten Daten, wie Textdokumente oder Videodateien.

- **Kosten**

Die Wartung der Datenbank kann hohe Kosten beanspruchen.

- **Datenmigration**

Das Bewegen der Daten von einer relationalen Datenbank auf eine Andere kann komplex sein und erfordert manuelle Anpassungen. [24]

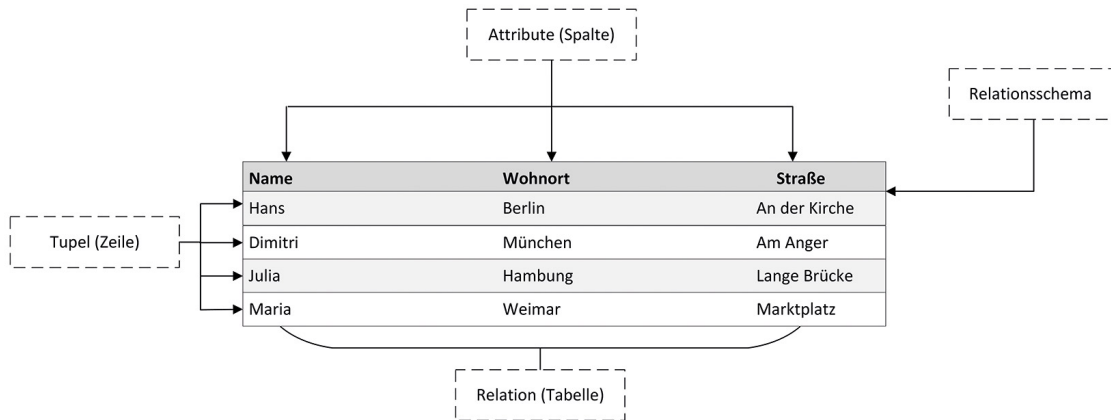


Abbildung 5: Architektur einer relationalen Datenbank, Quelle: [7]

Microsoft SQL Server ist eine von Microsoft entwickelte relationales Datenbankmanagementsystem. [4]

Der SQL Server bringt einige Vorteile mit sich, unter anderem die einfache Verwaltung mithilfe von Tools. In diesem Fall ist es SQL Server Management Studio (SSMS). Außerdem werden, abgesehen von Java, noch Programmiersprachen wie Python, .NET oder auch R unterstützt. Des Weiteren bringt die Datenbank den Vorteil, dass man Transaktionen durchführen kann.

Die Architektur von SQL-Server, wie auch in Abbildung 6 dargestellt, besteht grundlegend aus drei Komponenten:

- **Protokollschicht**
- **Relational Engine**
- **Speicher Engine**

Die Protokollschicht stellt die Kommunikation von Client und Server sicher. Dabei unterstützt es das TCP/IP und das Named Pipes Protokoll. [16]

Die Relational Engine ist im Grundlegenden für die Verarbeitung von Transaktionen und Abfragen zuständig. Die Engine besteht aus drei weiteren Komponenten:

- **CMD Parser**

Der SQL Befehl wird in ein Format umgewandelt, damit es die Engine versteht.

- **Optimierer**

Die Abfrage wird analysiert und die effizienteste Methode wird zum Ausführen ausgewählt.

- **Abfrageausführer**

Die effizienteste Methode wird ausgeführt und die Ergebnisse der Abfrage werden generiert.

Die Speicher Engine verwaltet die Zugriffe auf die Daten. Hier gibt es auch wieder drei weitere Komponenten:

- **Zugriffsmethode**

Hier wird die Entscheidung darüber getroffen, wie auf die Daten zugegriffen wird.

- **Puffermanager**

Hier wird der Zwischenspeicher (Cache) verwaltet. Dieser speichert Daten temporär im Arbeitsspeicher. Außerdem wird das Laden und Speichern zwischen Cache und Festplatte gesteuert, um die Leistung zu verbessern und die Zugriffsgeschwindigkeit zu erhöhen.

- **Transaktionsmanager**

Hier werden die Transaktionen verwaltet[19]

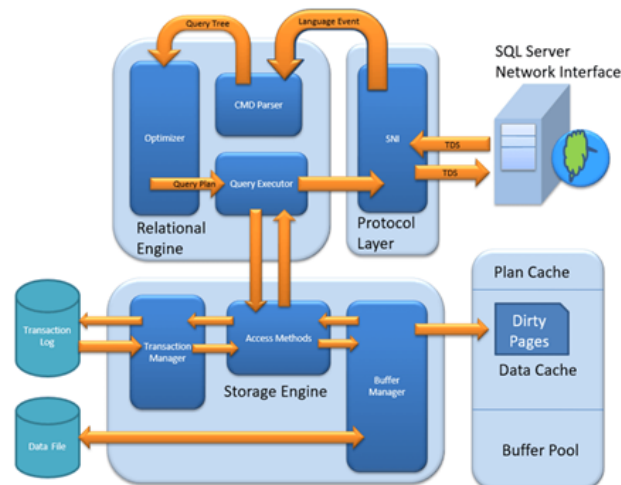


Abbildung 6: Architektur von SQL-Server, Quelle: [17]

2.1.4 Zugriffskontrolle

Zugriffskontrollen werden von vielen Unternehmen verwendet, um die Zugriffe auf unterschiedliche Daten zu kontrollieren und regulieren. Dies ist vor allem wichtig für sensible und vertrauliche Daten.

Azure Active Directory kurz Azure AD, ist ein von Microsoft zur Verfügung gestellter Dienst zur Verwaltung von Zugriffsrechten. Azure AD ist die cloudbasierte Variante zu Active Directory. Azure AD ermöglicht, unter anderem, einen Single Sign On (SSO) für verschiedene Dienste. Das bedeutet, dass man die jeweiligen Zugangsdaten nur einmal eingeben kann, um auf unterschiedliche Dienste zugreifen. Beispiele dafür wären Microsoft 365, Azure DevOps oder auch Anwendungen, die in Azure AD integriert sind. Administratoren können im Azure AD Rollen festlegen oder vordefinierte Rollen Benutzern zuweisen. Das können einzelne Benutzer oder auch Gruppen sein. Die Benutzer haben die Berechtigungen, die der Rolle zugewiesen sind. Ein Benutzer mit der Rolle Default User kann beispielsweise eine Liste von Autos ansehen. Hingegen kann ein Admin zusätzlich neue Autos anlegen.

2.2 Versionsverwaltung

Die Versionsverwaltung ist ein System, welches Entwickler bei der gemeinsamen Arbeit an Codes unterstützt, und auch dabei, Änderungen rückgängig zu machen und den Verlauf der Entwicklung nachzuverfolgen.

2.2.1 Azure DevOps

Azure DevOps ist eine Cloud-basierte, von Microsoft entwickelte Sammlung an Diensten, welche Entwicklerteams dabei unterstützt ihre Arbeit zu organisieren, gemeinsam an Codes zu arbeiten, automatisch zu testen und fertige Programme zu veröffentlichen. [14] [12]

Hier sind einige Funktionen von Azure DevOps:

- **Azure Boards:**

Unterstützt Teams bei der agilen Planung und Verfolgung ihrer Projekte, indem es die Möglichkeit bietet, Arbeitsaufgaben, User Stories, Bugs und Features zu erstellen, priorisieren und zu verwalten.

- **Azure Repos:**

Ist ein Git-basiertes Versionsverwaltungs-Tool, um Code sicherzustellen und es erleichtert Teams die Arbeit an gleichen Codes. Dazu stellt es Funktionen wie Pull-Requests, Branching, Code-Reviews und vieles mehr zur Verfügung.

- **Azure Pipelines:**

Ist ein CI/CD (Continuous Integration/Continuous Delivery) System, welches ermöglicht Build-, Test- und Release Pipelines zu erstellen, die für eine automatische Auslieferung von Software sorgen.

- **Azure Test Plans:**

Bietet Tools für das Testmanagement, mit dem Tests erstellt, geplant, ausgeführt und verfolgt werden können. Um sicherzustellen, dass die Anwendung bestmöglich funktioniert, ist es mit Azure Test Plans möglich Tests manuell durchzuführen, automatische Tests zu integrieren und Testabdeckungsberichte zu generieren.

- **Azure Artifacts:**

Mit Azure Artifacts können Pakete wie NuGet, npm und Maven in privaten oder öffentlichen Registern gehostet werden.

Neben all den Hauptfunktionen ist es auch möglich andere Tools von Microsoft und Drittanbietern zu integrieren und somit die Entwicklungsprozesse zu verbessern. Außerdem ermöglicht es die Erstellung von benutzerdefinierten Skripten und Workflows durch die Nutzung von YAML basierten Dateien.

3 Konzept

3.1 Funktionalität

Die Funktionalität wird in der Funktionsanalyse beschrieben. Mithilfe von Abbildungen werden die Verbesserungen und die Änderungen in der neuen Anwendung Trizebs Reloaded im Vergleich zu den früheren Anwendungen dargestellt und erklärt.

3.1.1 Funktionsanalyse

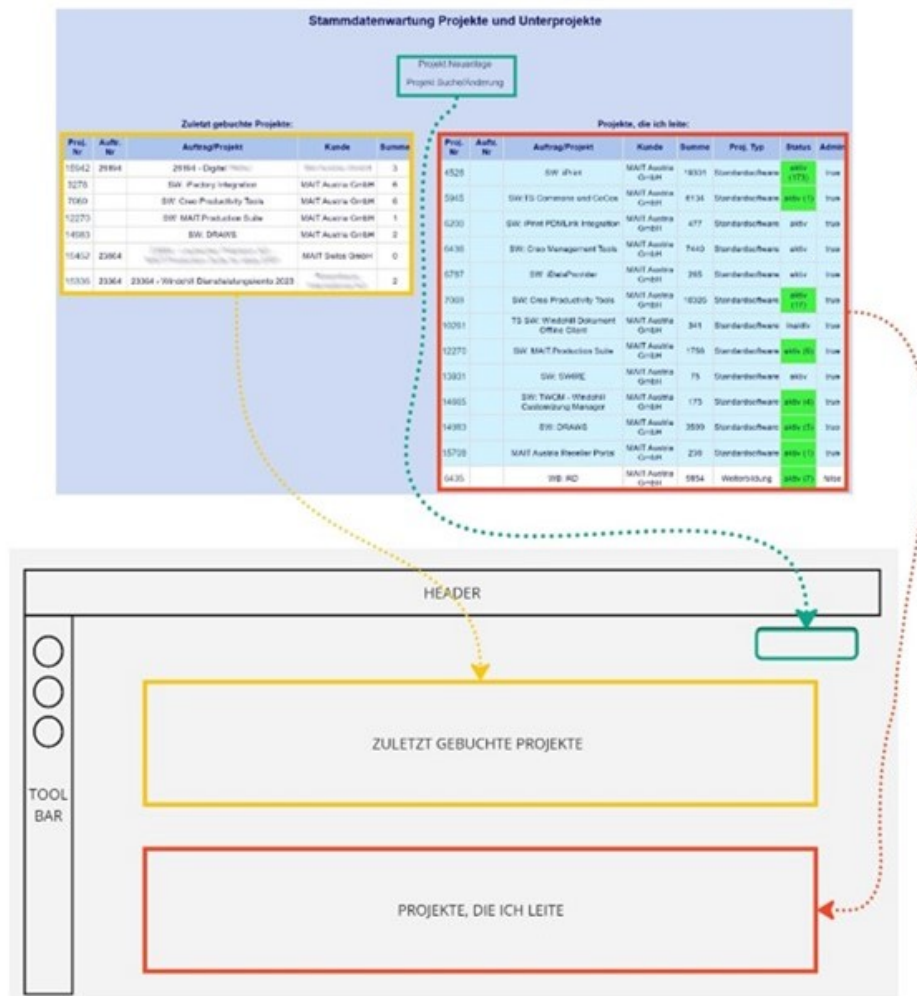


Abbildung 7: Anwendung Trizebs (oben), Konzept Trizebs Reloaded (unten)

In der Abbildung 7 ist die Oberfläche der Anwendung Trizebs zu sehen. Die Seite zeigt einen Überblick über die zuletzt gebuchten Projekte und den Projekten, die man jeweils leitet. Es ist zu erkennen, dass die beiden Auflistungen nebeneinander angeordnet sind. In der neuen Anwendung befinden sich die zwei Listen untereinander und sind jeweils scrollbar. Dies soll sicherstellen, dass die Seite übersichtlich bleibt, für den Fall, dass die Auflistungen sehr umfangreich werden. Die grundlegenden Konzepte kann man in der unteren Darstellung erkennen. Über den Auflistungen ist ein Button platziert, mit dem man neue Projekte erstellen kann. Wenn man auf eines dieser Projekte in der Liste klickt, kommt man auf eine Detail Ansicht, in der man mehr Informationen zu einem Projekt einsehen kann. Außerdem werden alle Unterprojekte zu diesem Projekt aufgelistet und es ist möglich, ein neues Unterprojekt zu einem Projekt erstellen. Diese Erstellfunktionen werden allerdings nur Projektleitern gewährt. Neben dem Button, der das hinzufügen ermöglicht, befindet sich ein weiterer Button, mit dem man nach bestimmten Projekten suchen kann. Die Liste mit den zuletzt gebuchten Projekten ist für die Rolle des Standardbenutzers nicht vorgesehen und deshalb nicht sichtbar. Jeder Projektleiter kommt standardmäßig auf die Projekt-Ansicht nach dem Login. Das ist auch eine Abänderung zur alten Applikation, wo jeder Benutzer im Anschluss zum Login auf die Projekt-Ansicht kommt.



Abbildung 8: Anwendung Task (oben), Konzept Trizebs Reloaded(unten)

Die Abbildung 8 zeigt die frühere Anwendung Task. Die Ansicht bietet eine Liste aller Aufgaben eines Benutzers. In der neuen Anwendung wird ebenso eine Liste mit allen eigenen Aufgaben angezeigt, wie man in der unteren Abbildung erkennen kann. Diese ist auch wieder scrollbar, damit sich der Benutzer gut zurechtfindet. Weiters kann man hier direkt eine Zeiterfassung zu einer Aufgabe starten und stoppen. Um eine neue Aufgabe hinzuzufügen, muss man in der Detail-Ansicht eines Projektes ein Unterprojekt auswählen. Man kommt auf eine Detail-Ansicht des Unterprojektes, wo alle Aufgaben des Unterprojektes aufgelistet sind. Über dieser Liste befindet sich ein Button, mit dem man eine Aufgabe erstellen kann. Wenn man ein normaler Benutzer ist, kommt man nach dem Login standardmäßig auf die Aufgaben-Ansicht. Der Grund hierfür ist, dass so die Effizienz der neuen Anwendung steigt. Die Teammitglieder können somit direkt nach dem Einloggen ihre Zeiterfassung starten.

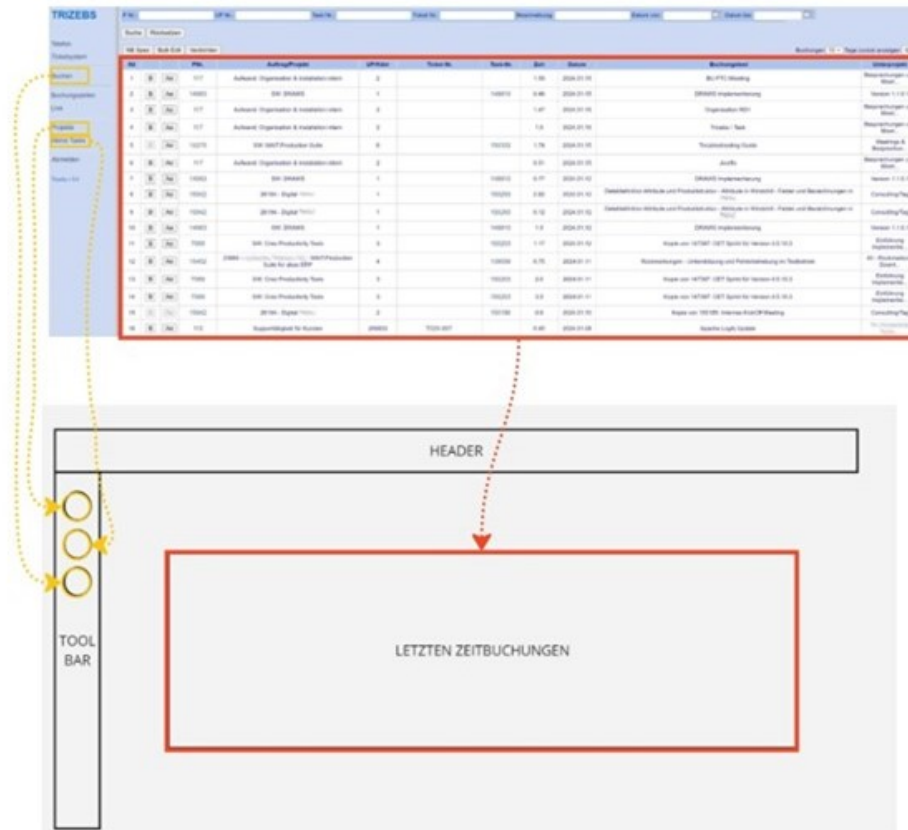


Abbildung 9: Anwendung Task (oben), Konzept Trizebs Reloaded(unten)

In Abbildung 9 ist eine weitere Ansicht von der alten Anwendung Task. Diese Seite, die einem die eigenen Zeiterfassungen anzeigt, kann man über die Toolbar erreichen. Zusätzlich ist es in dieser Auflistung möglich weitere Zeitbuchungen zu starten und zu stoppen. Außerdem kann man die bereits gebuchten Zeiten auch wieder bearbeiten. In der neuen Version kann man ebenfalls über die Toolbar auf diese Seite navigieren, jedoch werden die Wörter in dieser Ansicht durch Icons ersetzt. Die Icons sind platzsparend und optisch ansprechend, da die Toolbar schmaler ist. Über diese Icons kann man auf die Aufgaben-Ansicht und auch auf die Projekt-Ansicht gelangen. Im Header wird das neue Trizebs Reloaded Logo angezeigt, bei einem Klick auf die Abbildung wird der Benutzer auf die Startseite weitergeleitet. Weiters gibt es ein Icon, das es ermöglicht, sich wieder auszuloggen.

3.1.2 Use Cases

In Trizebs Reloaded unterscheidet man zwischen den beiden Benutzern **Standardbenutzer** und **Projektleiter**. Diese beiden Rollen besitzen unterschiedliche Rechte in der Anwendung.

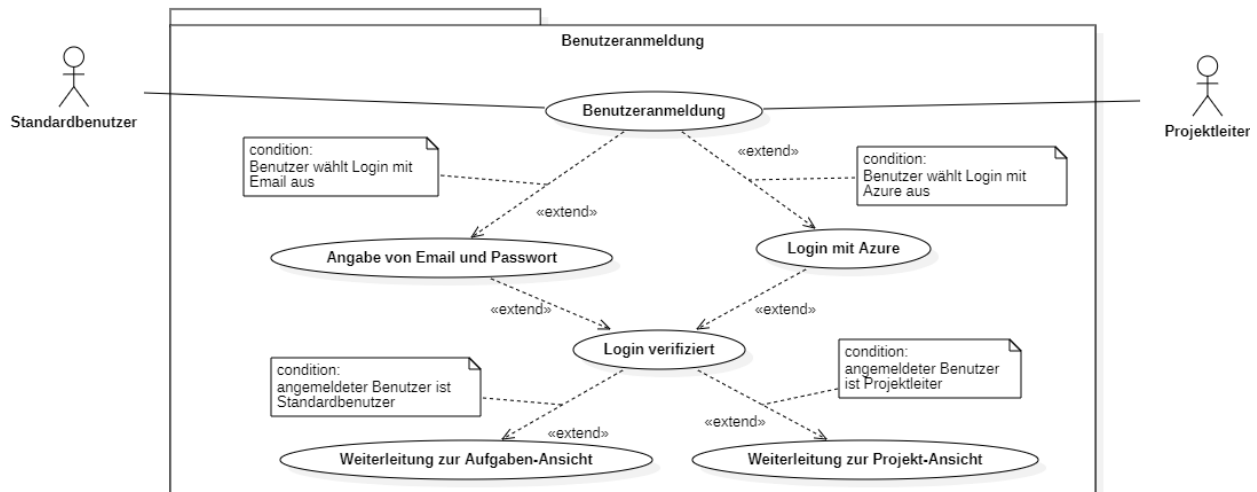


Abbildung 10: Use Case Diagramm zur Benutzeranmeldung

Die Abbildung 10 ist das Use Case Diagramm zur Benutzeranmeldung. Hier kann sich ein Benutzer entweder mit E-Mail und Passwort oder mit dem Azure-Login einloggen. Nach einem erfolgreichen Login wird ein Standardbenutzer zur Aufgaben-Ansicht weitergeleitet. Ein Projektleiter wird zur Projekt-Ansicht geleitet.

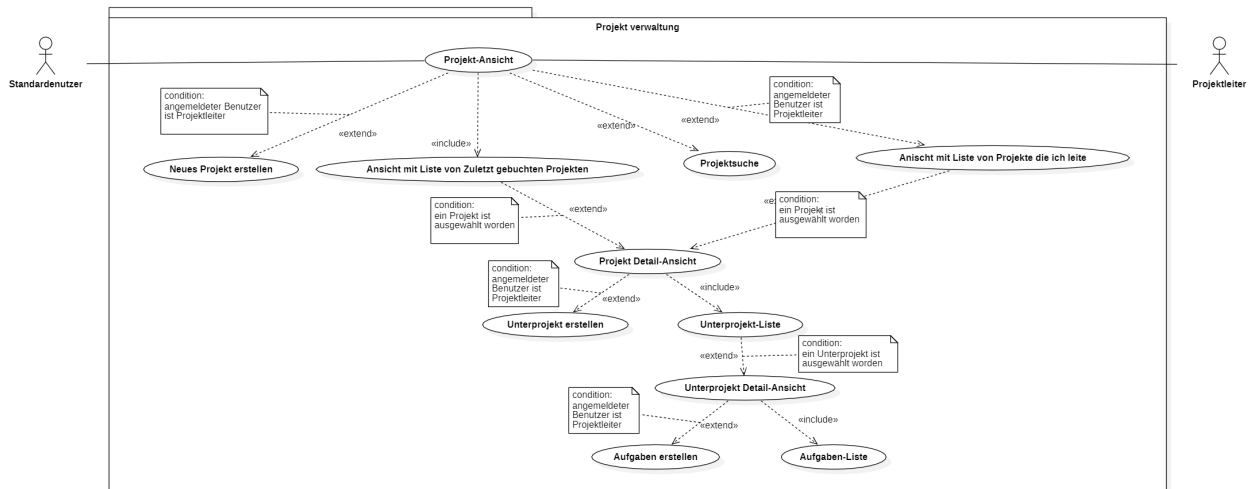


Abbildung 11: Use Case Diagramm zur Projektverwaltung

Die Abbildung 11 zeigt das Use Case Diagramm zur Projektverwaltung. In der Projekt-Ansicht haben die Benutzer eine Liste mit ihren zuletzt gebuchten Projekten. Projektleiter haben zusätzlich eine Auflistung von den Projekten, die sie leiten. Außerdem ist es ihnen möglich ein neues Projekt zu erstellen. Nach einem bestimmten Projekt suchen können sowohl ein Standardbenutzer als auch ein Projektleiter. Wenn die Benutzer ein Projekt aus den Auflistungen auswählen, werden sie auf eine Detail-Ansicht geleitet. Hier gibt es zusätzlich zu den Details von dem Projekt eine Liste mit allen Unterprojekten aus diesem Projekt. Ein Projektleiter besitzt wieder die Möglichkeit ein weiteres Unterprojekt zu erstellen. Die Benutzer können erneut ein Unterprojekt aus der Liste auswählen, dann kommen sie auf eine Seite, wo die Aufgaben dieses Unterprojekts aufgelistet sind. Einem Projektleiter ist es in diesem Fall wieder möglich, eine neue Aufgabe zu erstellen.

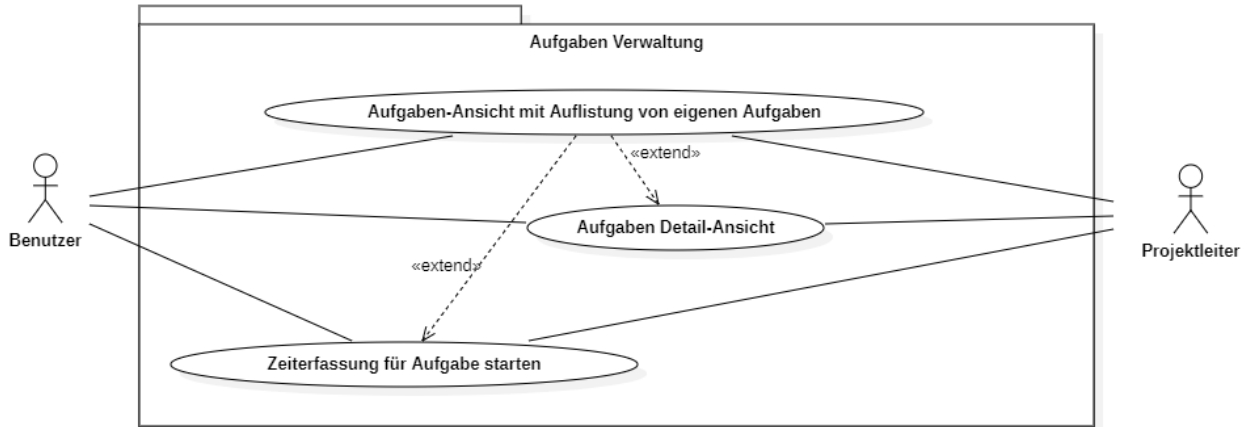


Abbildung 12: Use Case Diagramm zur Aufgabenverwaltung

In Abbildung 12 ist das Use Case Diagramm zur Aufgabenverwaltung ersichtlich. In der Aufgaben-Ansicht haben die Benutzer die Möglichkeit neue Zeiterfassungen zu bestimmten Aufgaben zu starten. Wenn die Benutzer eine Aufgabe auswählen, werden sie auf eine Detail-Ansicht dieser Aufgabe geleitet.

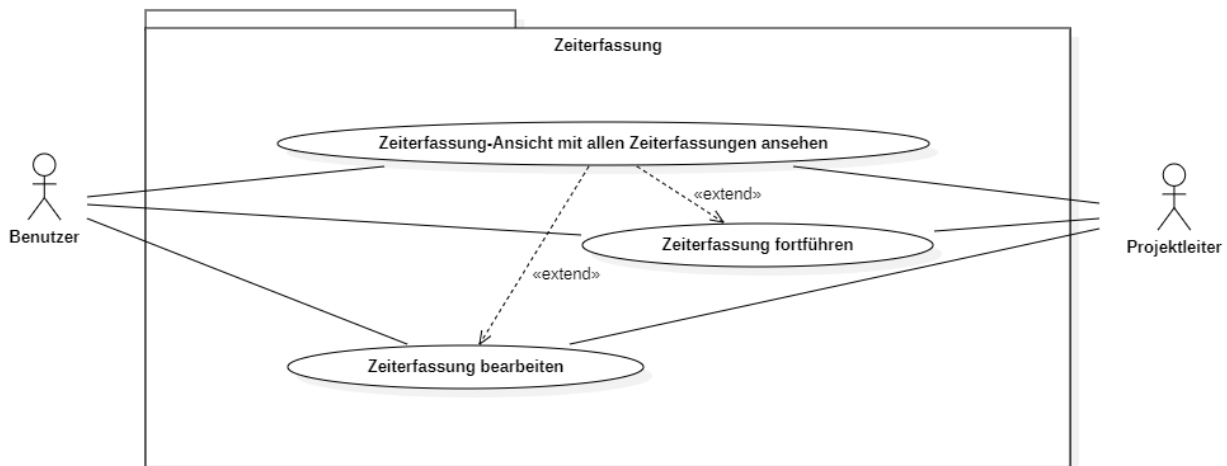


Abbildung 13: Use Case Diagramm zur Zeiterfassung

In Abbildung 13 kann man das Use Case Diagramm zur Zeiterfassung erkennen. Die Benutzer können eine Ansicht aufrufen, wo alle bisherigen und aktuell laufende Zeiterfassungen aufgelistet sind. In dieser Liste ist es ihnen möglich eine von den Zeiterfassung fortzuführen und zu bearbeiten.

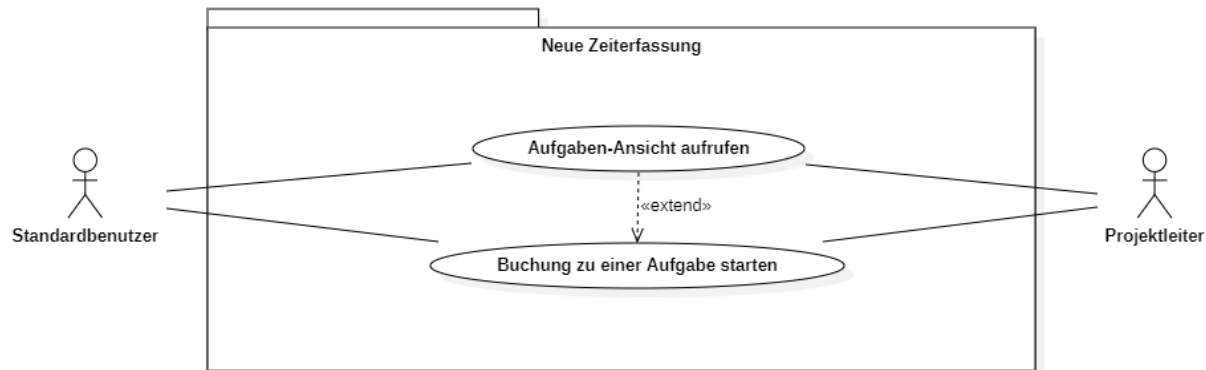


Abbildung 14: Use Case Diagramm zur Erstellung einer neuen Zeiterfassung

In Abbildung 14 ist das Use Case Diagramm zur Erstellung einer neuen Zeiterfassung dargestellt. Um eine neue Zeiterfassung zu erstellen, müssen die Benutzer die Aufgaben-Ansicht aufrufen. In der Aufgabenliste können sie zu einer bestimmten Aufgabe eine neue Zeiterfassung starten, indem sie auf den Start Button klicken.

3.2 Vereinigung der Anwendungen

Durch das Zusammenführen der beiden Applikationen Trizebs und Task bietet die neue Version den Benutzern eine verbesserte Benutzererfahrung. Dadurch, dass man nun eine einzige Anwendung hat, kann der Datenaustausch effizienter abgewickelt werden. So kann man die Leistung der Anwendung erheblich steigern. Die Funktionalitäten in der neuen Applikation sind so angeordnet, dass es klar ersichtlich ist, wo sich welche Funktionen befinden und wie sie zu bedienen sind. So wird die Eingewöhnungsphase verkürzt und den Nutzern eine unkomplizierte Nutzung der Anwendung ermöglicht.

3.3 Design Science

Das Design der neuen Anwendung, sowie die Anordnung der einzelnen Komponenten wird durch Verwendung von Design Science gestaltet. Dabei werden zuerst die Funktionalitäten und Komponenten der früheren Anwendungen, sowie die Schwierigkeiten, welche damit entstanden sind, analysiert, um an-

schließlich Ziele für die neue Anwendung zu schaffen.

Ziel dabei soll sein, die Benutzererfahrung und die Effizienz der Arbeit zu steigern. Ein Schwerpunkt liegt auf der Gestaltung der neuen Benutzeroberfläche, um eine einfache Bedienbarkeit für die Nutzer zu ermöglichen.

3.4 Datenmodell

Das Datenmodell der neuen Anwendung legt die Struktur und Organisation der Daten fest. Die Entitäten, Attribute und die Beziehung der Entitäten werden klar definiert, um ein einheitliches Verständnis aller Projektmitglieder sicherzustellen. Das Datenmodell ermöglicht die effiziente Verwaltung, Speicherung und Abfrage der Daten, wodurch die Applikation fließend funktionieren kann. In unserem Datenmodell werden die Projekte und Unterprojekte gespeichert, welche miteinander verbunden sind. Die Unterprojekte besitzen verschiedene Aufgaben, diese sind auch in der Datenbank gesichert. Außerdem kann man in der Datenbank Zeiterfassungen speichern, und jede Zeiterfassung gehört zu einer bestimmten Aufgabe. Das konzeptuelle ER-Diagramm ist in Abbildung 15 dargestellt.

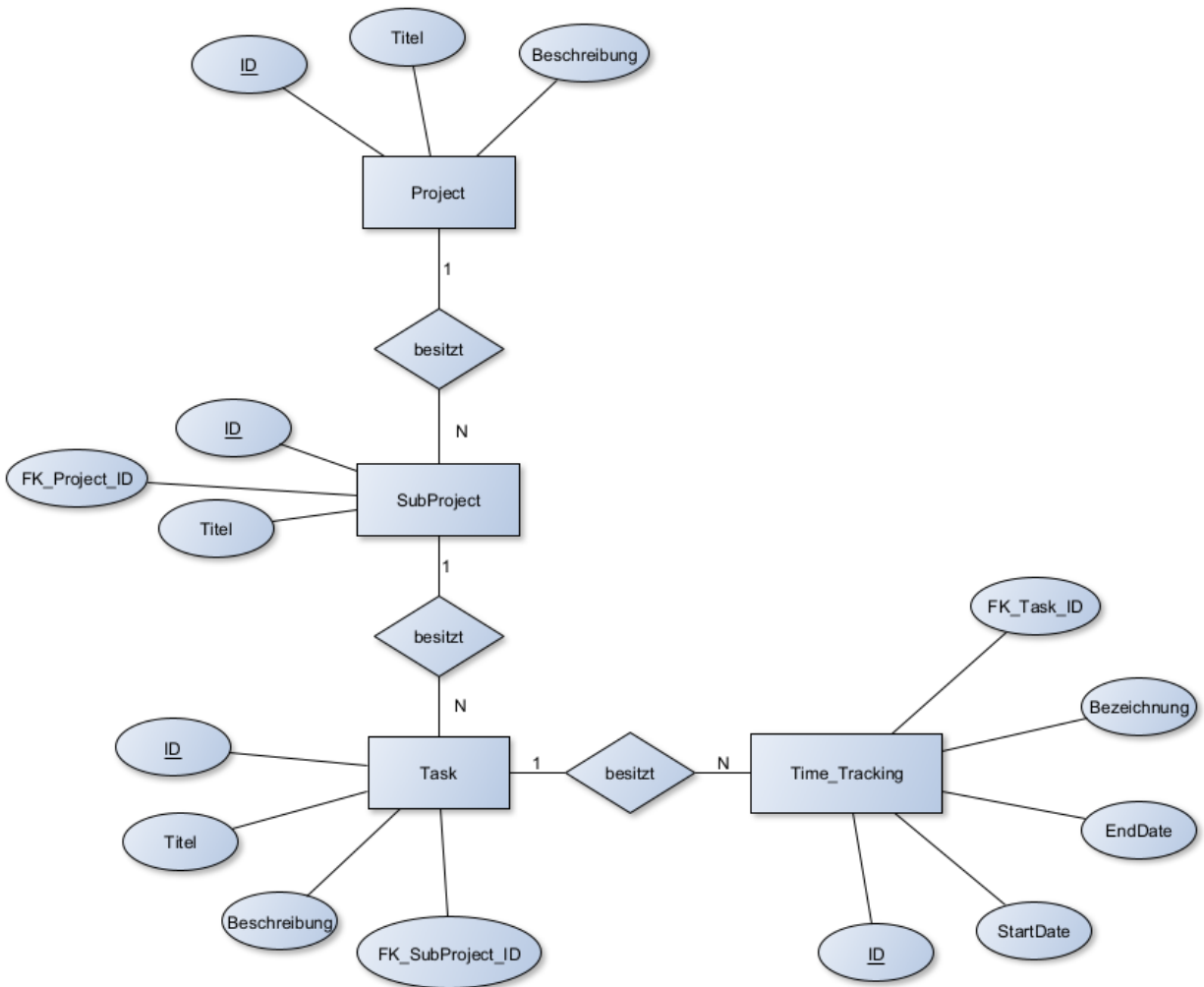


Abbildung 15: Konzept Datenmodell

3.5 Architektur

Die Anwendung besteht aus den Komponenten Frontend, Backend und der Datenspeicherung. Das Frontend ist mit Vue.js implementiert und das Backend mit Java Spring Boot. Die Daten, welche in der Applikation angezeigt und eingelesen werden, sind in einer Microsoft SQL Server Datenbank gesichert. Datenbank und Backend interagieren mittels JPA (Java Persistence API), die Kommunikation zwischen Frontend und Backend erfolgt über RESTful-APIs, wie in Abbildung 16 ersichtlich. Die Authentifizierung findet über einen JWT-Token statt.

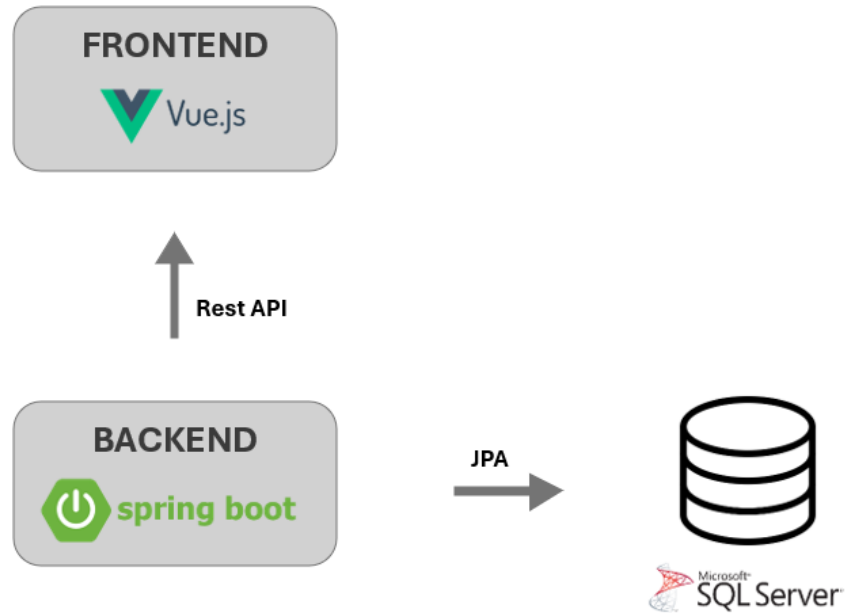


Abbildung 16: Grafische Abbildung der Architektur

3.6 Informationsfluss

In dem Projekt werden die Daten im Backend mittels JPA aus der Datenbank gelesen. Die extrahierten Datensätze sind auf die Anzeigen in der Benutzeroberfläche abgestimmt. Diese Datensätze werden dem Frontend mittels einer REST API übermittelt, und hier werden sie schließlich mithilfe von Services für die Benutzer übersichtlich und strukturiert präsentiert.

4 Implementierung

4.1 Datenbank

In diesem Kapitel werden die Schritte der Datenmodellierung, angefangen von der ersten ER-Diagramm Version bis hin zur finalen Version, geschildert. Außerdem wird die Erstellung der Datenbank genauer erklärt.

4.1.1 Beschreibung

Trizebs Reloaded hat im Hintergrund des Projektes eine SQL Server Datenbank. Eines der Gründe hierfür ist die einfache Verwaltung durch das Tool SQL Server Management Studio (SSMS). Darüber hinaus kann die Installation von SSMS einfach und schnell erfolgen. Außerdem ist eine relationale Datenbank in diesem Fall die beste Lösung, weil viele Daten abgespeichert werden müssen. Durch die Architektur von relationalen Datenbanken sind die Daten strukturiert und übersichtlich gesichert.

4.1.2 Datenmodell

Das Datenmodell hat mehrere Versionen, da wir dieses während der Entwicklung immer wieder angepasst haben. Der Großteil des Datenmodells ist von den bisherigen Anwendungen abgeleitet. Jedoch sind ein paar Meetings durchgeführt worden, um mit den Mitarbeitern das vorhandene Datenmodell zu besprechen. Mit dem Feedback ist das Datenmodell erweitert und verbessert worden.

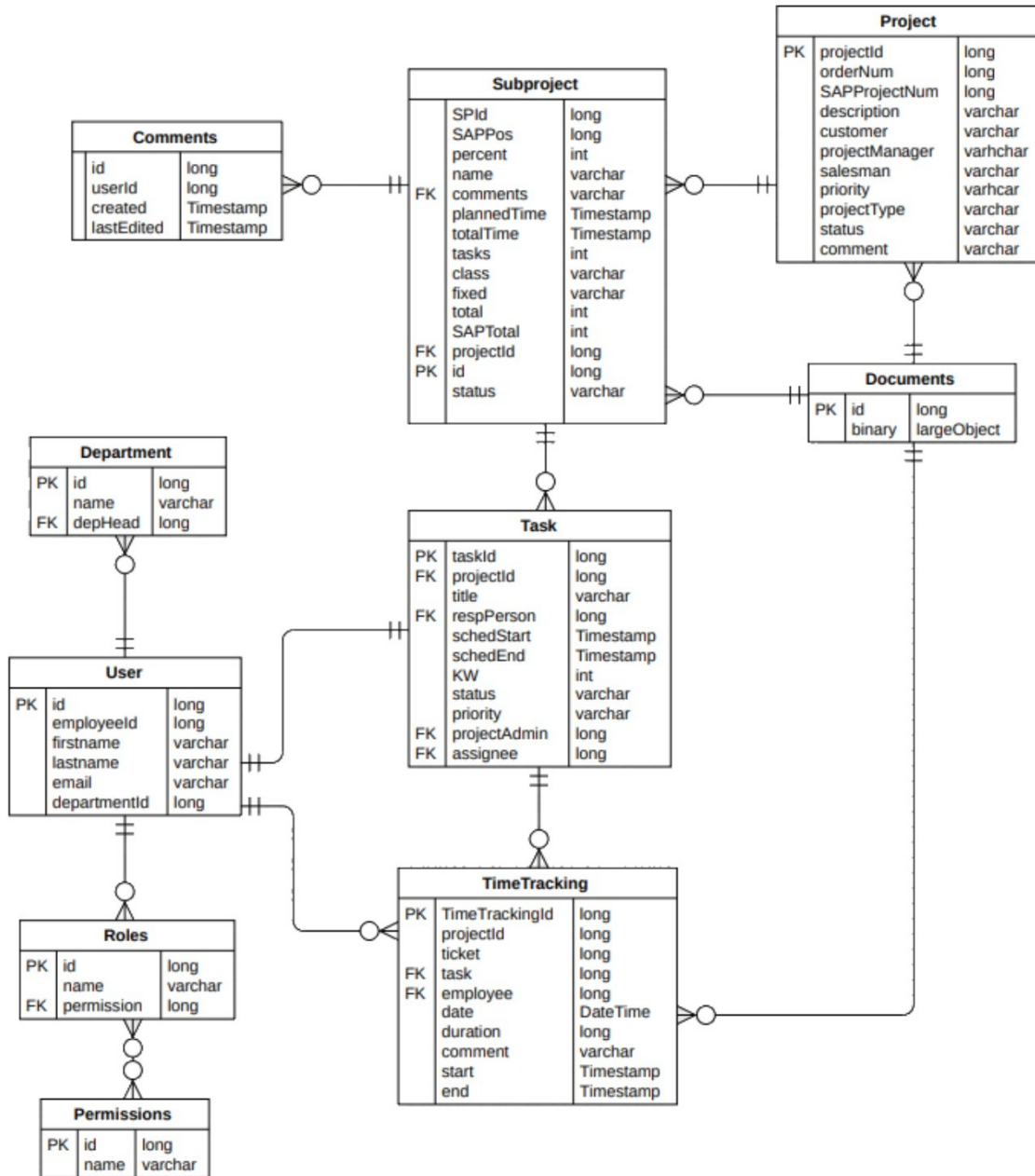


Abbildung 17: ERD Version 1

In Abbildung 17, der ersten Version des ER-Diagramms, kann man sehen, dass es im Vergleich zu Abbildung 18 und 19 noch nicht sehr detailliert ist. Die Grundkomponenten sind im Diagramm enthalten und es existieren bereits Beziehungen. Es fehlen jedoch einige Primär- und Fremdschlüssel, und auch viele Entitäten. Das Meeting mit den Mitarbeitern hat über dieses ERD stattgefunden.

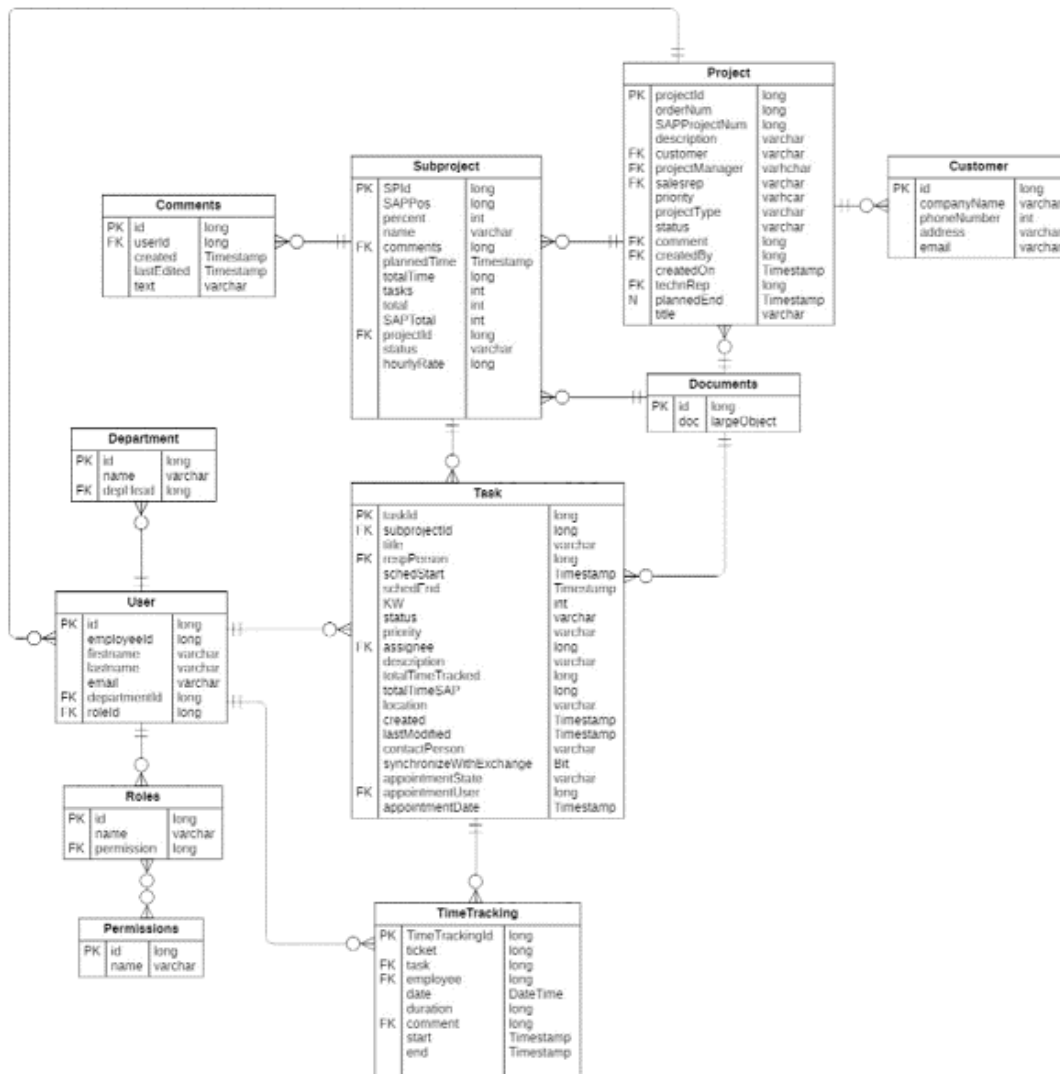


Abbildung 18: ERD Version 2

In der Abbildung 18, die nächste Version der ER-Diagramms, ist zu erkennen, dass es eine neue Entität *Customer* gibt, und diese hat eine 1:N Beziehung zu *Project*. Außerdem gibt es eine neue Beziehung zwischen *User* und *Project*. Die Beziehung *Documents* und *Timetracking* ist aufgelöst worden, stattdessen gibt es eine 1:N Beziehung zwischen *Documents* und *Task*.

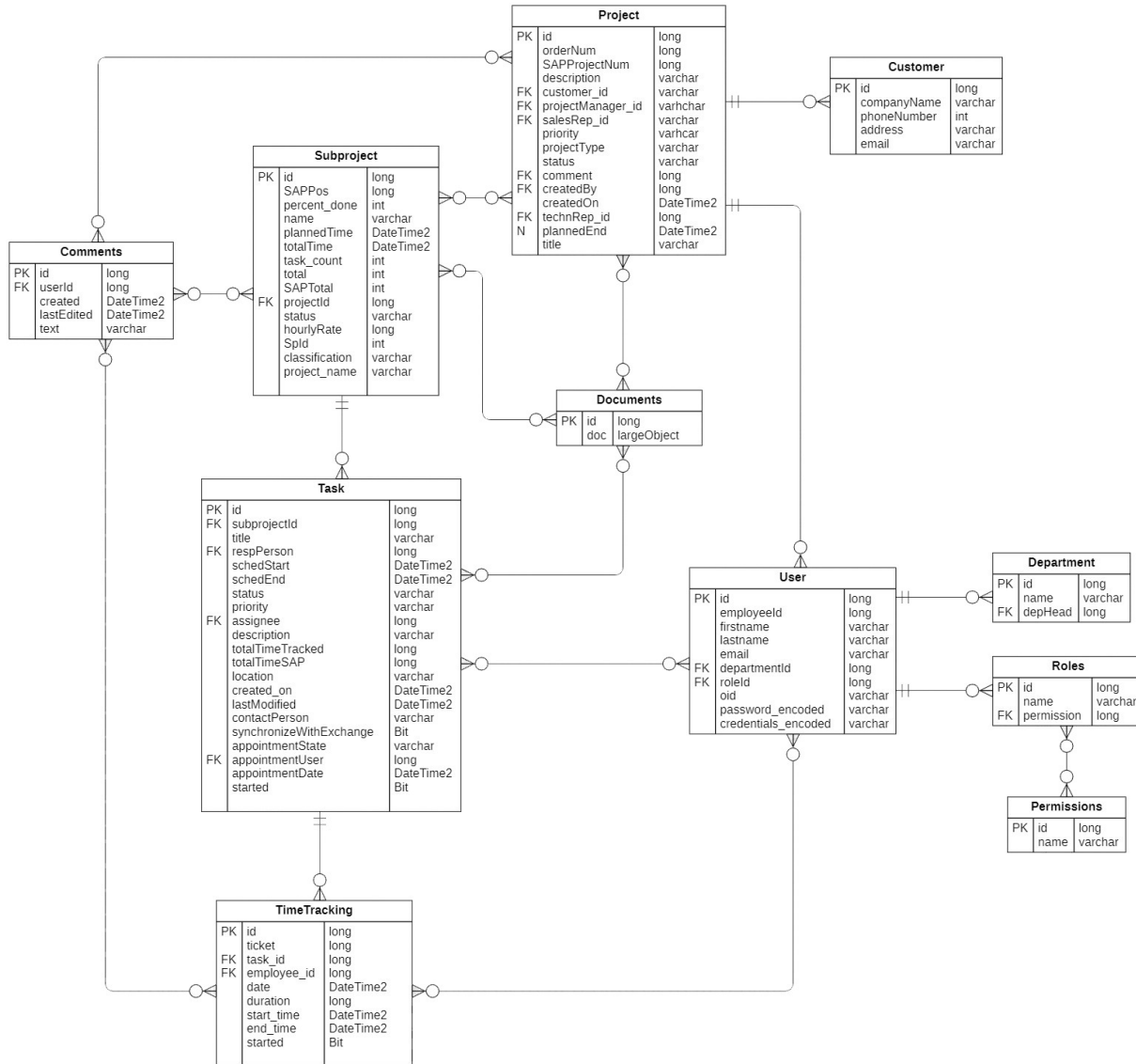


Abbildung 19: ERD finale Version

In der finalen Version gibt es hauptsächlich Änderungen in den Beziehungen zwischen den Entitäten, wie in Abbildung 19 erkennbar. Die Beziehungen von *Comments-Subproject*, *Comments-Project*, *Project-Documents*, *Subproject-Documents*, *Task-Documents*, *TimeTracking-User*, *Task-User* und *Subproject-Project* haben sich alle von 1:N zu N:M geändert. Weiters gibt es nun eine N:M Beziehung zwischen *Comments* und *TimeTracking*.

Nachdem die finale Version des ER-Diagramms erstellt worden ist und sichergestellt wurde, dass es allen Anforderungen entspricht, wird mit dem Aufbau der Datenbank mittels Code-First Ansatz begonnen. Dazu werden im Java Backend die Model Klassen definiert. Durch die Verwendung von JPA werden die Datenbank-Tabellen automatisch erstellt und können somit sehr simpel an Änderungen angepasst werden.

```
1 @Entity
2 @Getter
3 @Setter
4 @NoArgsConstructor
5 public class Project {
6     @Id
7     @GeneratedValue(strategy = GenerationType.AUTO)
8     private Long id;
9     private String title;
10
11     @Column(nullable = true)
12     private Date plannedEnd;
13
14     @OneToMany(fetch = FetchType.EAGER, mappedBy = "project", cascade =
15     CascadeType.ALL)
16     private List<SubProject> subProjects;
17
18     @ManyToOne(cascade = {CascadeType.DETACH, CascadeType.PERSIST})
19     private AppUser projectManager;
20
21     //Weitere Attribute...
```

Listing 1: Model Klasse von Project

In Listing 1 ist eine Model Klasse zu sehen. Mit der Annotation @Entity wird die Modelklasse als Entitätsklasse definiert. Dadurch erkennt JPA, dass diese Klasse für eine Tabelle in der Datenbank steht.

4.2 Webanwendung

Um auf die Daten aus der Datenbank zugreifen zu können wird in Trizebs Reloaded JPA verwendet. Dazu wurden einige Endpoints, Services und Repositories erstellt. In den Endpoints werden die jeweiligen Service Methoden aufgerufen und im Service wiederum die Methoden aus dem RepositoryInterface. Die Repository Interfaces erben vom JPARepository Interface, welches bereits viele Methoden zur Verfügung stellt.

4.2.1 Rechteverwaltung

Nicht jeder User darf in TrizebsReloaded alle Funktionen bedienen. Deswegen sind im Azure Active Directory verschiedene Rollen definiert.

Role	Project								SubProjects							
	create		edit		delete		view		create		edit		delete		view	
	own	all	own	all	own	all	own	all	own	all	own	all	own	all	own	all
CloudTask.ProjectCollectionAdmin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CloudTask.ProjectManager	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CloudTask.TaskManager							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DefaultAccess																

Abbildung 20: Rollen-Tabelle Teil 1

Role	Tasks								TimeBookings							
	create		edit		delete		view		create		edit		delete		view	
	own	all	own	all	own	all	own	all	own	all	own	all	own	all	own	all
CloudTask.ProjectCollectionAdmin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CloudTask.ProjectManager	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CloudTask.TaskManager							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DefaultAccess			<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

Abbildung 21: Rollen-Tabelle Teil 2

Man kann in den Abbildungen 20 und 21 erkennen, dass vier Rollen definiert sind. Die Rechte sind auf vier Gruppen Unterteilt:

Rollen

- ProjectCollectionAdmin
- ProjectManager
- TaskManager

- Default Access

Rechte-Gruppen

- Project
- SubProject
- Task
- TimeBookings

Ein **Admin** besitzt alle Rechte. Das bedeutet, dass man als Admin Projekte, Unterprojekte, Aufgaben und Zeiterfassungen erstellen darf und, zusätzlich zu den eigenen, die von anderen Mitarbeitern bearbeiten, löschen und ansehen darf.

Als **Project Manager** darf man Projekte erstellen und die eigenen Projekte editieren. Außerdem hat man das Recht, alle Projekte zu löschen, aber nur in die eigenen Projekte einzusehen. Bei den Unterprojekten gilt, dass man diese erstellen und auch bearbeiten kann. Es ist möglich alle Unterprojekte anzusehen und diese zu löschen. Weiters kann man mit dieser Rolle Aufgaben erstellen, bearbeiten und alle Aufgaben einsehen. Man darf diese jedoch nicht löschen. Das gleiche gilt für Zeiterfassungen.

Für einen **Task Manager** gilt, dass man sowohl die eigenen Projekte als auch die Projekte von anderen ansehen kann. Dasselbe gilt für die Unterprojekte. Dafür hat man für Tasks, also für Aufgaben, alle Rechte und das gleiche gilt auch für die Zeiterfassungen.

Default Access besitzt die wenigsten Rechte. Man darf alle Aufgaben ansehen und die eigenen bearbeiten. Außerdem ist es mit dieser Rolle möglich eigene Zeiterfassungen zu erstellen, diese zu editieren, zu löschen und anzusehen.

4.2.2 Rest-API

- **Services:**

- **UserService:** Im UserService werden alle Datenbankoperationen bearbeitet, welche mit einem Benutzer zu tun haben. Dazu gibt es folgende Methoden:

- * **Create User:** Um einen neuen Benutzer zu erstellen, gibt es die Methode createUser:

```
1 public AppUser createUser(AppUser user){
2     if(user != null){
3         return userRepository.save(user)
4     }
5     return null;
6 }
```

Listing 2: createUser Methode

- * **Find User:** Um einen bestimmten Benutzer basierend auf seiner ID zu finden, gibt es die Methode findUser:

```
1 public AppUser findUser(Long id){
2     Optional<AppUser> user = userRepository.findById(id);
3     return user.orElse(null);
4 }
```

Listing 3: findUser Methode

- * **Update User:** Um einen Benutzer zu ändern, gibt es die Methode updateUser:

```
1 public AppUser updateUser(AppUser user){
2     if(user != null){
3         return userRepository.save(user)
4     }
5     return null;
6 }
```

Listing 4: updateUser Methode

- * **Delete User:** Um einen Benutzer zu löschen, gibt es die Methode `deleteUser`:

```
1 public boolean deleteUser(Long id){
2     Optional<AppUser> user = userRepository.findById(id);
3     if(user.isPresent()){
4         userRepository.delete(user.get());
5         return true;
6     }
7     return false;
8 }
```

Listing 5: deleteUser Methode

- **ProjectService:** Im ProjectService werden alle Datenbankoperationen bearbeitet, welche mit einem Projekt zu tun haben. Dazu gibt es folgende Methoden:

- * **Create Project:** Um ein neues Projekt zu erstellen, gibt es die Methode `createProject`:

```
1 public Project createProject(Project project){
2     if(project != null){
3         return projectRepository.save(project);
4     }
5     return null;
6 }
```

Listing 6: createProject Methode

- * **Find Project:** Um ein bestimmtes Projekt basierend auf seiner ID zu finden, gibt es die Methode findProject:

```
1 public Project findProject(Long id){
2     Optional<Project> project = projectRepository.findById(id);
3     if(project.isPresent()){
4         return project.get();
5     }else{
6         return null;
7     }
8 }
```

Listing 7: findProject Methode

- * **Update Project:** Um ein Projekt zu ändern, gibt es die Methode updateProject:

```
1 public Project updateProject(Long id){
2     if(project != null){
3         return projectRepository.save(project)
4     }
5     return null;
6 }
```

Listing 8: updateProject Methode

- * **List all Projects:** Um alle Projekt aufzulisten, gibt es die Methode listAllProjects:

```
1 public List<Project> listAllProjects(){
2     List<Project> projects = projectRepository.findAll();
3     return projects;
4 }
```

Listing 9: listAllProjects Methode

- * **List Projects Managed by User:** Um alle Projekte des eingeloggten Benutzers anzuzeigen, gibt es die Methode `listAllProjectsManagedByUser`:

```
1 public List<Project> listAllProjectsManagedByUser(AppUser user){
2     return projectRepository.findAllProjectsByProjectManager(user
3     );
3 }
```

Listing 10: `listAllProjectsManagedBy` Methode

- * **Delete Project:** Um ein Projekt zu löschen, gibt es die Methode `deleteProject`:

```
1 public boolean deleteProject(Long id){
2     Optional<Project> project = projectRepository.findById(id);
3     if(project.isPresent()){
4         projectRepository.delete(project.get());
5         return true;
6     }
7     return false;
8 }
```

Listing 11: `deleteProject` Methode

- **SubProjectService:** Im `SubProjectService` werden alle Datenbankoperationen bearbeitet, welche mit einem Unterprojekt zu tun haben. Dazu gibt es folgende Methoden:

- * **Create SubProject:** Um ein neues Unterprojekt zu erstellen, gibt es die Methode `createSubProject`:

```
1 public SubProject createSubProject(SubProject subProject){
2     if(subProject != null){
3         if(subProject != null){
4             return subProjectRepository.save(subProjects);
5         }
6     }
6 }
```

Listing 12: `createSubProject` Methode

- * **Find SubProject:** Um ein bestimmtes Unterprojekt basierend auf der ID zu finden, gibt es die Methode findSubProject:

```
1 public SubProject findSubProject(Long id){
2     Optional<SubProject> subProject = subProjectRepository.
        findById(id);
3     if(subProject.isPresent()){
4         return subProject.get();
5     }else{
6         return null;
7     }
8 }
```

Listing 13: findSubProject Methode

- * **Update SubProject:** Um ein Unterprojekt zu ändern, gibt es die Methode updateSubProject:

```
1 public SubProject updateSubProject(SubProject subProject){
2     if(subProject!=null){
3         return subProjectRepository.save(subProject);
4     }
5     return null;
6 }
```

Listing 14: updateSubProject Methode

- * **Delete SubProject:** Um ein Unterprojekt zu löschen, gibt es die Methode deleteSubProject:

```
1 public boolean deleteSubProject(Long id){
2     Optional<SubProject> subProject = subProjectRepository.
        findById(id);
3     if(subProject.isPresent()){
4         subProjectRepository.delete(subProject.get());
5         return true;
6     }
7     return false;
8 }
```

Listing 15: deleteSubProject Methode

- **TaskService:** Im TaskService werden alle Datenbankoperationen bearbeitet, welche mit einer Aufgabe zu tun haben. Dazu gibt es folgende Methoden:

- * **Create Task:** Um eine neue Aufgabe zu erstellen, gibt es die Methode createTask:

```
1 public Task createTask(Task task){
2     if(task!=null){
3         return taskRepository.save(task);
4     }
5     return null;
6 }
```

Listing 16: createTask Methode

- * **Find Task:** Um ein bestimmtes Unterprojekt basierend auf der ID zu finden, gibt es die Methode findTask:

```
1 public Task findTask(Long id){
2     Optional<Task> task = taskRepository.findById(id);
3     if(task.isPresent()){
4         return task.get();
5     }else{
6         return null;
7     }
8 }
```

Listing 17: findTask Methode

- * **Get all Tasks for responsible Person:** Um alle Aufgaben des eingeloggten Benutzers aufzulisten, gibt es die Methode getAllTasksForRespPerson:

```
1 public List<Task> getAllTasksForRespPerson(AppUser user) {
2     return taskRepository.findAllTasksByRespPerson(user);
3 }
```

Listing 18: getAllTasksForRespPerson Methode

- * **Update Task:** Um eine Aufgabe zu ändern, gibt es die Methode `updateTask`:

```
1 public Task updateTask(Long id, Task task) {
2     if (taskRepository.existsById(id)) {
3         task.setId(id);
4         return taskRepository.save(task);
5     } else {
6         return null;
7     }
8 }
```

Listing 19: `updateTask` Methode

- * **Delete Task:** Um eine Aufgabe zu löschen, gibt es die Methode `deleteTask`:

```
1 public boolean deleteTask(Long id){
2     Optional<Task> task = taskRepository.findById(id);
3     if(task.isPresent()){
4         taskRepository.delete(task.get());
5         return true;
6     }
7     return false;
8 }
```

Listing 20: `deleteTask` Methode

- **TimeTrackingService:** Im `TimeTrackingService` werden alle Datenbankoperationen bearbeitet, welche mit einer Zeitbuchung zu tun haben. Dazu gibt es folgende Methoden:

- * **Create TimeTracking:** Um eine neue Zeitbuchung zu erstellen, gibt es die Methode createTimeTracking:

```
1 public TimeTracking createTimeTracking(TimeTracking timeTracking
2     ) {
3     if (timeTracking != null) {
4         return timeTrackingRepository.save(timeTracking);
5     }
6     return null;
7 }
```

Listing 21: createTimeTracking Methode

- * **Find TimeTracking:** Um eine bestimmte Zeitbuchung basierend auf der ID zu finden, gibt es die Methode findTimeTracking:

```
1 public TimeTracking findTimeTracking(Long id){
2     TimeTracking timeTracking = findTimeTracking(id);
3     if(timeTracking!=null){
4         return timeTracking;
5     }else{
6         return null;
7     }
8 }
```

Listing 22: findTimeTracking Methode

- * **List all TimeTrackings of User:** Um alle Zeitbuchungen des eingeloggtten Benutzers aufzulisten, gibt es die Methode listAllTrackingsOfUser:

```
1 public List<TimeTracking> listAllTimeTrackings(AppUser user){
2     return timeTrackingRepository.findAllTrackingsByEmployee(user
3     );
4 }
```

Listing 23: listAllTimeTrackings Methode

- * **Update TimeTracking:** Um eine Zeitbuchung zu ändern, gibt es die Methode updateTimeTracking:

```
1 public TimeTracking updateTimeTracking(Long id, TimeTracking
   updatedTimeTracking) {
2     if (timeTrackingRepository.existsById(id)) {
3         updatedTimeTracking.setId(id);
4         return timeTrackingRepository.save(updatedTimeTracking);
5     } else {
6         return null;
7     }
8 }
```

Listing 24: updateTimeTracking Methode

- * **Delete TimeTracking:** Um eine Zeitbuchung zu löschen, gibt es die Methode deleteTimeTracking:

```
1 public boolean deleteTimeTracking(Long id){
2     Optional<TimeTracking> timeTracking = timeTrackingRepository.
   findById(id);
3     if(timeTracking.isPresent()){
4         timeTrackingRepository.delete(timeTracking.get());
5         return true;
6     }
7     return false;
8 }
```

Listing 25: deleteTimeTracking Methode

- **Endpoints:** Die Service Methoden werden nun in den Endpoints aufgerufen.

– **UserEndpoint:**

In Tabelle 1 sind alle Endpoints definiert, welche für die Verwaltung der Benutzer verwendet werden.

Methode	Pfad	Permission	Erklärung
POST	/users	Project.Create	Legt einen neuen Benutzer in der Datenbank an. Der neue Benutzer wird als requestBody übergeben
PATCH	/users/{id}	Project.Create	Aktualisiert den Benutzer mit der angegebenen ID. Das aktualisierte Benutzer Objekt wird als requestBody übergeben.
GET	/users/{id}	Project.Create	Liefert den Benutzer mit der angegebenen ID
DELETE	/users/{id}	Project.Create	Löscht den Benutzer mit der angegebenen ID

Tabelle 1: User-Endpoint Tabelle

– ProjectEndpoint:

In Tabelle 2 sind alle Endpoints definiert, welche für die Verwaltung der Projekte verwendet werden.

Method	Path	Permission	Erklärung
POST	/project	Project.Create	Legt ein neues Projekt in der Datenbank an. Das neue Projekt wird als RequestBody übergeben
PATCH	/project/{id}	Project.Edit	Aktualisiert das Projekt mit der angegebenen ID. Das aktualisierte Project Objekt wird als RequestBody übergeben.
GET	/project/{id}	Project.View	Liefert das Projekt mit der angegebenen ID
GET	/project/listAll-ProjectsManagedByUser/{id}	Project.View	Liefert alle Projekte des Benutzers mit der angegebenen ID
DELETE	/project/{id}	Project.Delete	Löscht das Projekt mit der angegebenen ID

Tabelle 2: Project-Endpoint Tabelle

– SubProjectEndpoint:

In Tabelle 3 sind alle Endpoints definiert, welche für die Verwaltung der Unterprojekte verwendet werden.

Method	Path	Permission	Erklärung
POST	/subprojects	SubProject.Create	Legt ein neues Unterprojekt in der Datenbank an. Das neue Unterprojekt wird als RequestBody übergeben
PATCH	/subprojects/{id}	SubProject.Edit	Aktualisiert das Unterprojekt mit der angegebenen ID. Das aktualisierte Unterprojekt Objekt wird als RequestBody übergeben.
GET	/subprojects/{id}	SubProject.View	Liefert das Unterprojekt mit der angegebenen ID
DELETE	/subprojects/{id}	SubProject.Delete	Löscht das Unterprojekt mit der angegebenen ID

Tabelle 3: Subproject-Endpoint Tabelle

– TaskEndpoint:

In Tabelle 4 sind alle Endpoints definiert, welche für die Verwaltung der Aufgaben verwendet werden.

Method	Path	Permission	Erklärung
POST	/tasks	Task.Create	Legt eine neue Aufgabe in der Datenbank an. Die neue Aufgabe wird als RequestBody übergeben
PATCH	/tasks/{id}	Task.Edit	Aktualisiert die Aufgabe mit der angegebenen ID. Das aktualisierte Aufgaben Objekt wird als RequestBody übergeben.
GET	/tasks/{id}	Task.View	Liefert die Aufgabe mit der angegebenen ID
GET	/tasks/tasksOfUser	Task.View	Liefert alle Aufgaben des Benutzers mit der angegebenen ID
DELETE	/tasks/{id}	Task.Delete	Löscht die Aufgabe mit der angegebenen ID

Tabelle 4: Task-Endpoint Tabelle

– TimeTrackingEndpoint:

In Tabelle 5 sind alle Endpoints definiert, welche für die Verwaltung der Zeitbuchungen verwendet werden.

Methode	Pfad	Permission	Erklärung
POST	/time_tracking	TimeTracking.Create	Legt eine neue Zeitbuchung in der Datenbank an. Die neue Zeitbuchung wird als Request-Body übergeben
PATCH	/time_tracking/{id}	TimeTracking.Edit	Aktualisiert die Zeitbuchung mit der angegebenen ID. Das aktualisierte Zeitbuchungs Objekt wird als RequestBody übergeben.
GET	/time_tracking/{id}	TimeTracking.View	Liefert die Zeitbuchung mit der angegebenen ID
GET	/time_tracking/time_trackingOfUser/{id}	TimeTracking.View	Liefert alle Zeitbuchung des Benutzers mit der angegebenen ID
DELETE	/time_tracking/{id}	TimeTracking.Delete	Löscht die Zeitbuchung mit der angegebenen ID

Tabelle 5: Task-Endpoint Tabelle

4.3 Benutzerschnittstelle

In dem Vue.js Projekt gibt es verschiedene Komponenten, um die Anwendung übersichtlich und wiederverwendbar zu halten. Diese Komponenten befinden sich im components Ordner. Mit einem Rechtsklick auf den Ordner kann man unter New eine neue Vue-Komponente anlegen. Neben dem Komponenten Ordner gibt es einen Ordner für alle Services. Diese können ebenfalls ganz einfach mit einem Rechtsklick auf den Ordner unter New und JavaScriptFile erstellt werden. Weiters gibt es noch einen Ordner mit dem Namen assets. In diesem befinden sich Bilder, die in der Anwendung verwendet werden. Die Ordner Struktur ist in Abbildung 22 grafisch ersichtlich.

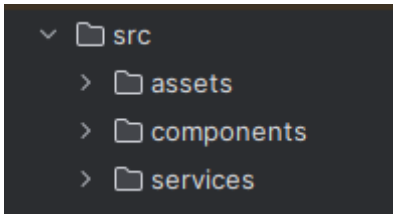


Abbildung 22: Frontend Ordner Struktur

4.3.1 Verwaltung der Projekte

In der Anwendung beruht die Verwaltung der Projekte auf dem Anzeigen und dem Erstellen von neuen Projekten. Die Unterprojekte zu einem Projekt kann man ebenfalls einsehen und erstellen. Für die Umsetzung existieren im Vue.js Projekt folgende Komponenten:

- **ProjectsComponent** In der Projects Komponente werden alle zuletzt gebuchten Projekte, sowie alle Projekte, welche man leitet, angezeigt. Dazu gibt es zwei Tabellen.

```
1 <div class="pad-top-48">
2   <h1>Zuletzt gebuchte Projekte</h1>
3   <div class="tableContainer">
4     <table class="projectTable mar-top-48">
5       <thead>
6         <tr>
7           <th>Projektnummer</th>
8           <th>Auftragsnummer</th>
9           <th>Projekt</th>
10          <th>Kunde</th>
11        </tr>
12      </thead>
13      <tbody>
14        <tr v-for="project in projects" :key="project.id" class="hover"
15          @click="navigateToOtherComponent(project.id)">
16          <td>{{ project.id }}</td>
17          <td>{{ project.orderNum }}</td>
18          <td>{{ project.title }}</td>
19          <td>{{ project.customer.companyName }}</td>
20        </tr>
21      </tbody>
22    </table>
23  </div>
</div>
```

Listing 26: Liste der Zuletzt gebuchten Projekte

In Listing 26 ist die Tabelle der zuletzt gebuchten Projekte zu sehen. Die th in Zeile 7-10 definieren die Spaltenüberschriften. In Zeile 14 wird mittels foreach-Schleife die Liste von Projekten durchgegangen und von jedem Projekt wird dann die id, die Auftragsnummer, der Titel und der Kunde angezeigt. Auf diese Attribute wird, wie gewohnt, mit der Punktnotation zugegriffen.

```
1 <div class="mar-top-48">
2   <h1>Projekte, die ich leite </h1>
3   <table class="projectTable mar-top-48">
4     <thead>
5     <tr>
6       <th>Projektnummer</th>
7       <th>Auftragsnummer</th>
8       <th>Projekt</th>
9       <th>Kunde</th>
10    </tr>
11    </thead>
12    <tbody>
13     <tr v-for="project in projectManagedBy" :key="project.id" class="
14     hover" @click="navigateToOtherComponent(project.id)">
15       <td>{{ project.id }}</td>
16       <td>{{ project.orderNum }}</td>
17       <td>{{ project.title }}</td>
18       <td>{{ project.customer.companyName }}</td>
19     </tr>
20    </tbody>
21  </table>
22 </div>
```

Listing 27: Liste der Projekte die man leitet

In Listing 27 ist die Tabelle der Projekte, die man leitet, zu sehen. Die th in Zeile 6-9 definieren wieder die Spaltenüberschriften. In Zeile 13 wird mittels foreach-Schleife die Liste von Projekten durchgegangen und von jedem Projekt wird dann die id, die Auftragsnummer, der Titel und der Kunde angezeigt.

Auf diese Attribute wird, wie gewohnt, mit der Punktnotation zugegriffen. Die Projekte werden mittels zwei Service Methoden geladen.

```
1 created() {
2   allProjectsService.getProjects().then(project => {
3     this.projects = project;
4     console.dir(this.projects);
5   });
6
7   projectManagedByService.getProjects().then(project => {
8     this.projectManagedBy = project;
9     console.dir(this.projectManagedBy);
10  });
11 },
```

Listing 28: Methoden der Projekt listen

Diese Methoden aus Listing 28 befinden sich im JavaScript Teil der Projects Komponente. Durch die `getProjects` Methode aus dem `projectManagedBy Service`, bekommt man alle Projekte, die der eingeloggte User leitet. Die `getProjects` Methode aus dem `allProjects Service`, liefert alle zuletzt gebuchten Projekte.

Wenn man auf eine Zeile der Tabelle klickt, wird der Event-Handler `@click` aufgerufen, welcher die Methode `NavigateToOtherComponent` aufruft. Diese Methode sorgt dafür, dass man auf die Detailansicht dieses Projektes gelangt.

- **SubProjectsComponent**

In dieser Detailansicht werden mehrere Informationen zum Projekt angezeigt. Außerdem werden hier alle Unterprojekte aufgelistet, die zu diesem Projekt gehören.

```
1 created() {
2     this.projectId = this.$route.params.projectId;
3
4     subprojectsOfProjectService.getSubProjects(this.projectId).then(
5     subProjects => {
6         this.subProjects = subProjects;
7     });
8
9     oneProjectService.getProject(this.projectId).then(project => {
10        this.project = project;
11    });
12 }
```

Listing 29: Methoden für ein Projekt und Unterprojekte

In Zeile 4 wird die Service Methode aufgerufen, um das jeweilige Projekt zu bekommen, dieses wird in einem Projekt Objekt abgespeichert und anschließend angezeigt. In Zeile 8 wird die Methode aufgerufen, um alle Unterprojekte des Projektes zu erhalten, diese werden in einer Liste abgespeichert und angezeigt.

- **CreateProjectComponent** In der Projects Komponente befindet sich ein Button, welcher auf die "CreateProjectComponent" leitet. In dieser Komponente kann man ein neues Projekt anlegen.

Dazu gibt es eine Tabelle mit Input Feldern um alle Parameter eines Projektes angeben zu können. Beim Klick auf den Button "Projekt erstellen" wird die Methode submitProject aufgerufen. In dieser Methode wird das Projekt Objekt erstellt und anschließend wird die Service Methode createProject aufgerufen.

```
1 async submitProject() {
2     try {
3         const projectData = {
4             title: this.title ,
5             orderNum: this.orderNum,
6             sapproject_num: this.SAPProjectNum,
7             priority: this.priority ,
8             createdOn: Date.now(),
9             createdBy: {
10                id: userService.getUserId(),
11            },
12            //Andere Attribute...
13        };
14        const response = await createProjectService.createProject(
15            projectData);
16        this.showAlert();
17    } catch (error) {
18        console.error("Fehler beim Erstellen des Projekts:", error);
19    },
```

Listing 30: submitProject Methode

In Listing 30 ist die Methode zu sehen, welche ein Projekt Objekt erstellt und die Methode aus dem Service aufruft, welche das neue Projekt dann an das Backend weiterleitet.

- **CreateSubProjectComponent** In der Subprojects Komponente befindet sich über der Liste an Unterprojekten ein Button, welcher auf eine weitere Komponente führt. In dieser Komponente können neue Unterprojekte erstellt werden.

Dazu gibt es ebenfalls eine Tabelle mit Input Feldern, um alle Parameter eines Unterprojektes angeben zu können. Beim Klick auf den Button Unterprojekt erstellen wird die Methode submitSubProject aufgerufen. In dieser Methode wird das Unterprojekt Objekt erstellt und anschließend wird die Service Methode createSubProject aufgerufen.

```
1 async submitSubProject() {
2     try {
3         const subProjectData = {
4             name: this.name,
5             hourlyRate: this.hourly_rate,
6             SAPTotal: this.saptotal,
7             SAPPos: this.sappos,
8             //Andere Attribute...
9         };
10        const response = await createSubProjectService.createSubProject(
11            subProjectData);
12        this.showAlert();
13    } catch (error) {
14        console.error("Fehler beim Erstellen des Unterprojekts:", error);
15    },
```

Listing 31: submit SubProject Methode

In Listing 31 ist die Methode zu sehen, welche ein Unterprojekt Objekt erstellt und die Methode aus dem Service aufruft, welche das neue Unterprojekt dann an das Backend weiterleitet.

4.3.2 Verwaltung der Tasks

Die Verwaltung der Tasks beziehen sich in der Anwendung auf die Anzeige der Tasks zu einem Unterprojekt, auf die Einsehung in die eigenen Tasks und auf die Erstellung neuer Tasks.

Für die Umsetzung existieren im Vue.js Projekt folgende Komponenten:

- **TasksComponent:** In dieser Komponente werden einem Benutzer alle eigenen Aufgaben angezeigt. Dazu gibt es, wie auch bei den Projekten, eine Liste der Aufgaben.

- **TasksOfSubprojectComponent:** Um auf diese Seite zu gelangen, muss man in der SubProjectsComponent auf eines der Unterprojekte klicken. Zu diesem Unterprojekt werden dann alle dazugehörigen Tasks aufgelistet. Hier ist die Auflistung aller Aufgaben eines Unterprojektes zu sehen.

```

1 <tbody v-for="(task, index) in tasks" :key="index">
2   <tr class="hover" :class="{ 'runningTask': highlightedRowIndex ===
   index }" @click="navigateToOtherComponent(task.id)">
3     <td>{{ task.id }}</td>
4     <td>{{ task.projectNr }}</td>
5     <td>{{ task.title }}</td>
6     <td>{{ task.respPerson.lastName }}</td>
7     <td>{{ task.totalTimeSAP }} Stunden</td>
8     <td>{{ task.totalTimeTracked }} Stunden</td>
9     <td>{{ task.totalTimeSAP - task.totalTimeTracked }} Stunden</td>
10    <td>{{ formatDate(task.schedStart) }}</td>
11    <td>{{ formatDate(task.schedEnd) }}</td>
12    <td>{{ task.kw }}</td>
13    <td>{{ task.status }}</td>
14  </tr>
15 </tbody>

```

Listing 32: Liste aller Aufgaben

Die Aufgaben werden mit der Methode `getTasksOfSubproject` aus dem Service geholt und in einer Liste gespeichert. Diese Liste wird dann als Tabelle dargestellt.

```

1 created() {
2   this.subProjectNum = this.$route.params.subProjectNum;
3
4   tasksOfSubprojectService.getTasksOfSubproject(this.subProjectNum).
   then(tasks => {
5     this.tasks = tasks;
6     console.dir(this.tasks);
7   });
8 },

```

Listing 33: Methode der Aufgaben eines Unterprojektes

In Listing 33 wird die ID des jeweiligen Unterprojektes aus dem Pfad geholt und in die Variable `subProjectNum` gespeichert. Danach wird die Service Methode aufgerufen, um alle Aufgaben des Unterprojektes zu bekommen und in einer Liste zu speichern.

- **TaskDetailComponent:** Bei einem Klick auf einen Task in der Liste gelangt man auf eine Detail-Ansicht. In dieser werden, wie auch bei den Projekten, mehr Details zu einem Task angezeigt. Dazu werden alle Attribute der Aufgabe übersichtlich in einer tabellarischen Form ausgegeben.
- **CreateTaskComponent:** In der `TasksOfSubProject` Komponente gibt es einen Button, welcher es ermöglicht, eine neue Aufgabe anzulegen. Beim Klick auf den Button gelangt man auf eine Weitere Komponente. Auch hier gibt es eine Tabelle mit Input Feldern, um alle Parameter der Aufgabe angeben zu können. Beim Klick auf den Button "Task erstellen" wird die Methode `submitTask` aufgerufen. In dieser Methode wird das Task Objekt erstellt und anschließend wird die Service Methode `createTask` aufgerufen.

```
1 async submitTask() {
2   try {
3     const taskData = {
4       title: this.title ,
5       description: this.description ,
6       priority: this.priority ,
7       status: "InProgress" ,
8       respPerson: {
9         id: this.selectedRespPerson.id
10      },
11      createdOn: Date.now() ,
12      //Weitere Attribute...
13    };
14    const response = await createTaskService.createTask(taskData);
15    this.showAlert();
16  } catch (error) {
17    console.error("Fehler beim Erstellen des Tasks:", error);
```

Listing 34: submitTask Methode

4.3.3 Verwaltung der Zeiterfassungen

In der Anwendung können Zeitbuchungen gestartet und gestoppt werden. Eine Zeitbuchung wird immer einem Task zugewiesen, daher ist das Starten und Stoppen der Zeitbuchungen in der Ansicht aller eigenen Aufgaben möglich. In der Tabelle der Tasks befindet sich ein Start-Button, klickt man auf diesen, wird eine Zeitbuchung zu dem jeweiligen Task gestartet. Wenn ein Task gestartet wurde, wird die Zeile grün hinterlegt. Außerdem ändert sich der Start-Button zu einem Stop-Button. Die erstellte Zeitbuchung wird dann in der "TimeTracking" Komponente angezeigt. In dieser Komponente wird die Zeitbuchung als gestartet markiert, sie ist ebenfalls grün hinterlegt und es befindet sich hier ein Stop-Button, um die Buchung wieder zu stoppen. Man kann die Buchung nun entweder in der Tasks Komponente oder in der „TimeTracking“ Komponente stoppen. Beide dieser Ansichten sind über die Toolbar zu erreichen.

Wenn man in hier in Zeile 5 auf das PlayIcon klickt, wird die Methode startTask aufgerufen. Diese sorgt dafür, dass eine Zeitbuchung zu dieser Aufgabe gestartet wird. Klickt man in Zeile 3 auf das StopIcon, wird diese Zeitbuchung wieder gestoppt, indem die Methode stopTask aufgerufen wird.

```
1 <div class="div-size-20 icon_hover mar-left-8">
2   <template v-if="task.started || (isStarted && highlightedRowIndex ===
3     index)">
4     <StopIcon @click="stopTask(task)"/> </template>
5   <template v-else>
6     <PlayIcon @click="startTask(index, task)"/></template>
7 </div>
```

Listing 35: start und stop Task

In Listing 36 ist die startTask Methode zu sehen.

```
1 startTask(rowIndex, task) {
2     this.highlightedRowIndex = rowIndex;
3     this.isStarted = true;
4     const current = new Date();
5     try {
6         const timeTracking = {
7             duration: 0,
8             task: task,
9             employee: {
10                id: userService.getUserId(),
11            },
12            date: Date.now(),
13            startTime: Date.now(),
14            started: 1,
15        };
16        const response = timeTrackingsOfUserService.startTimeTracking(
timeTracking);
17    } catch (error) {
18        console.error("Fehler beim Tracken der Zeit:", error);
19    }
20 },
```

Listing 36: startTask Methode

In Listing 37 ist die stopTask Methode zu sehen.

```
1 stopTask(task) {
2   this.highlightedRowIndex = -1;
3   this.isStarted = false;
4   timeTrackingsOfUserService.getLastTracking().then(lastTracking => {
5     this.lastTracked = lastTracking;
6     console.dir(this.lastTracked);
7     console.dir(this.lastTracked.id);
8     const timeTrackingToUpdate = {
9       id: this.lastTracked.id,
10      endTime: Date.now(),
11      date: this.lastTracked.date,
12      startTime: this.lastTracked.startTime,
13      task: this.lastTracked.task,
14      employee: {
15        id: userService.getUserId(),
16      },
17      duration: Date.now() - this.lastTracked.startTime,
18      started: 0,
19    };
20    try {
21      const response = timeTrackingsOfUserService.stopTimeTracking(this.
lastTracked.id, timeTrackingToUpdate);
22    } catch (error) {
23      console.error("Fehler beim Tracken der Zeit:", error);
24    }
25  }
26 });
```

Listing 37: stopTask Methode

In der TimeTracking Ansicht ist, neben dem Starten und Stoppen eines Eintrages, auch das Editieren einer Zeitbuchung möglich. Neben dem Start-Button befindet sich ein Edit-Button, klickt man auf diesen, ändern sich die Startzeit und Endzeit Felder zu Input Feldern und man kann diese beiden Werte bearbeiten. Bei anschließendem Klick auf einen Bestätigungs-Button, werden die Änderungen gespeichert.

```
1 <template v-else>
2   <td>
3     <div class="div-row actions">
4       <div class="div-size-20 icon_hover mar-left-8">
5         <CheckIcon class="icons" @click="saveEdit(tracking)"/>
6       </div>
7     </div>
8   </td>
9   <td>{{ tracking.id }}</td>
10  <td><input type="datetime-local" v-model="tracking.startTime"></td>
11  <td><input type="datetime-local" v-model="tracking.endTime"></td>
12  <td>{{ formatDuration(tracking.duration) }}</td>
13  <td>{{ formatDate(tracking.date) }}</td>
14  <td>{{ tracking.task.title }}</td>
15 </template>
```

Listing 38: TimeTracking bearbeiten

In Zeile 10 und 11 sind die beiden Input Felder zu sehen, um die Startzeit und Endzeit zu bearbeiten. Klickt man, wie in Zeile 5 zu sehen, auf den Bestätigungs-Button, wird die saveEdit Methode aufgerufen und die Änderungen werden gespeichert.

5 Ergebnis

5.1 Integration von Trizebs

Die Funktionalitäten von der früheren Anwendung Trizebs werden in Trizebs Reloaded übernommen.

5.1.1 Projekte anlegen

Das Erstellen von Projekten ist nur Projektleitern gestattet. In der Ansicht der Projekte gelangt man über den Plus-Button zu der Ansicht, um ein neues Projekt anzulegen.



The screenshot shows the 'Projekt anlegen' (Create Project) form in the Trizebs Reloaded application. The form is titled 'Projekt anlegen' and is located in the center of the page. It contains several input fields for project details:

- Auftragsnummer
- Priorität *
- Projekttyp *
- Beschreibung
- Projektname *
- Verkäufer *
- Technischer Ansprechpart
- TT.mm.jjjj
- SAP Nr
- Kunde *
- Projektleiter *

At the bottom left of the form, there is a blue button labeled 'Erstellen'.

Abbildung 23: Projekt anlegen Ansicht

Wie man in Abbildung 23 erkennen kann, befinden sich auf dieser Seite einige Input Felder, um alle Informationen eines neuen Projektes zu definieren. Beim Klick auf den Erstellen-Button, wird das Projekt in der Datenbank angelegt.

5.1.2 Projekte ansehen

Wie in Abbildung 24 ersichtlich, befinden sich, wenn man als Projektleiter eingeloggt ist, auf der Seite aller Projekte die Liste aller Projekte die man selber leitet und aller zuletzt gebuchten Projekte.



Projektnummer	Auftragsnummer	Projekt	Kunde
1	1	Website Redesign Project	KreativDesign GmbH
2	2	Mobile App Launch Project	KreativDesign GmbH
3	3	Database Migration Project	BioGenuss GmbH
4	4	E-Commerce Platform Development	NaturShop OG
5	5	UI/UX Improvement Project	TechExperts AG
6	6	Android App Upgrade Project	TechWorld AG
7	7	Data Analysis Report Project	ModeTrends e.U.
8	8	Backend API Integration Project	ModeTrends e.U.
9	9	iOS App Development Project	SportProfi AG
10	10	Data Security Enhancement Project	Backzauber e.U.

Projektnummer	Auftragsnummer	Projekt	Kunde
1	1	Website Redesign Project	KreativDesign GmbH
2	2	Mobile App Launch Project	KreativDesign GmbH

Abbildung 24: Liste aller zuletzt gebuchten und aller Projekte die man leitet

Als Standardbenutzer ist, wie in Abbildung 25 ersichtlich, nur die Liste der zuletzt gebuchten Projekte zu sehen.



Projektnummer	Auftragsnummer	Projekt	Kunde
1	1	Website Redesign Project	KreativDesign GmbH
2	2	Mobile App Launch Project	KreativDesign GmbH
3	3	Database Migration Project	BioGenuss GmbH
4	4	E-Commerce Platform Development	NaturShop OG
5	5	UI/UX Improvement Project	TechExperts AG
6	6	Android App Upgrade Project	TechWorld AG
7	7	Data Analysis Report Project	ModeTrends e.U.
8	8	Backend API Integration Project	ModeTrends e.U.
9	9	iOS App Development Project	SportProfi AG
10	10	Data Security Enhancement Project	Backzauber e.U.

Abbildung 25: Liste aller zuletzt gebuchten Projekte

In Abbildung 26 ist zu erkennen, dass sich beim Klick auf ein Projekt eine Detailansicht, in der man alle Details zu diesem Projekt sieht, öffnet.

Details

Projektnummer	1	Projektname	Website Redesign Project	Auftragsnummer	1
Beschreibung	Website Redesign	Kunde	KreativDesign GmbH	Status	In Progress
Technischer Kontakt	Friedl	Verkäufer	Friedl	Projektleiter	Friedl
Priorität	A				

Unterprojekte

UnterProjekt Nr	UnterProjekt Name	SAP Position	Tasks
1	Website Redesign - UX Research	120	10
2	Website Redesign - UI Design	80	8
3	Website Redesign - Frontend Development	90	12
4	Website Redesign - Backend Development	110	5
5	Website Redesign - Testing and QA	70	15
6	Test	130	4

Abbildung 26: Detail Ansicht eines Projektes und Liste der Unterprojekte dazu

Weiters gibt es eine eigene Seite, um nach dem Namen eines bestimmten Projektes zu suchen. In Abbildung 27 ist die Suchfunktion zu sehen.

Projektsuche

Projektname Suchen

Projekt Nr	Auftrags Nr	Projektname	Kunde	SAP Nr	Beschreibung	Priorität	Tech. Ansprechperson	Status
1	1	Website Redesign Project	KreativDesign GmbH	1	Website Redesign	A	Friedl	In Progress
2	2	Mobile App Launch Project	KreativDesign GmbH	2	Mobile App Launch	C	Friedl	Not Started
3	3	Database Migration Project	BioGenuss GmbH	3	Database Migration	A	Friedl	Completed
4	4	E-Commerce Platform Development	NaturShop OG	4	E-Commerce Platform	A	Korkmaz	In Progress
5	5	UI/UX Improvement Project	TechExperts AG	5	UI/UX Improvement	C	Birnzain	Not Started
6	6	Android App Upgrade Project	TechWorld AG	6	Android App Upgrade	B	Birnzain	In Progress
7	7	Data Analysis Report Project	ModeTrends e.U.	7	Data Analysis Report	A	Ster	Not Started
8	8	Backend API Integration Project	ModeTrends e.U.	8	Backend API Integration	C	Ster	Completed
9	9	iOS App Development Project	SportProfi AG	9	iOS App Development	C	Moosbrugger	Not Started
10	10	Data Security Enhancement Project	Backzauber e.U.	10	Data Security Enhancement	B	Pejic	In Progress
2402	12	TestProject	Gartenparadies OG		Das ist eine Test Beschreibung	A	Friedl	In Progress

Abbildung 27: Projekt suchen.

5.1.3 Unterprojekte anlegen

Als Projektleiter sieht man in der Detailansicht eines Projektes einen Plus-Button, mit dem man ein neues Unterprojekt anlegen kann. Klickt man auf diesen öffnet sich, wie auch bei den Projekten, eine neue Ansicht, in der man ein Unterprojekt erstellen kann. In Abbildung 28 sieht man, dass sich wieder einige Input Felder auf dieser Seite befinden, um alle Informationen eines Unterprojektes zu definieren. Beim Klick auf Unterprojekt erstellen, wird das neue Unterprojekt in der Datenbank angelegt.



The screenshot shows a web interface for creating a sub-project. The header is blue with the 'rizebs' logo and a user profile icon. A sidebar on the left contains navigation icons. The main content area is titled 'Unterprojekt anlegen' and contains several input fields: 'Name', 'Klassifizierung *', 'SAP Position *', 'Hourly rate *', and 'SAP Total'. A date field is labeled 'TT.mm.jjjj'. A blue 'Erstellen' button is located at the bottom of the form.

Abbildung 28: Unterprojekt anlegen Ansicht

5.1.4 Unterprojekte ansehen

Ebenfalls in der Detailansicht der Projekte befindet sich die Liste aller Unterprojekte des jeweiligen Projektes. Beim Klick auf ein bestimmtes Unterprojekt öffnet sich eine Seite mit allen Aufgaben zu diesem Unterprojekt.

5.2 Integration von Task

Die frühere Anwendung Task, welche der Mait Austria zur Aufgaben- und Zeit-erfassung gedient hat, wird nun in Trizebs Reloaded integriert.

5.2.1 Aufgabe anlegen

Das Erstellen von Aufgaben kann nur von einem Projektleiter durchgeführt werden. Jedoch ist dies ein wichtiger Schritt für alle Mitarbeiter, damit sie immer wieder neue Aufgaben zu erledigen haben und Zeitbuchungen starten können.



The screenshot shows the 'Tasks 1' view in the Trizebs Reloaded application. It features a table with the following columns: Task Nr, Titel, Verantwortliche Person, Geplante Zeit, Tatsächliche Zeit, Zeit verbleibend, Start, Ende, and Status. The table contains four rows of task data.

Task Nr	Titel	Verantwortliche Person	Geplante Zeit	Tatsächliche Zeit	Zeit verbleibend	Start	Ende	Status
6	Research Data Analysis	Friedl	25 Stunden	15 Stunden	10 Stunden	05.09.2023	05.09.2023	Planned
7	Stakeholder Presentation	Friedl	20 Stunden	10 Stunden	10 Stunden	10.09.2023	10.09.2023	In Progress
8	Revised Wireframes	Friedl	25 Stunden	15 Stunden	10 Stunden	15.09.2023	15.09.2023	Planned
9	UI Implementation	Friedl	30 Stunden	20 Stunden	10 Stunden	20.09.2023	20.09.2023	In Progress

Abbildung 29: Aufgaben eines Unterprojektes

Die Abbildung 29 zeigt die Auflistung der Aufgaben von Unterprojekt 1. Man gelangt auf die Seite, indem man auf ein Unterprojekt in der Projekt-Detail-Ansicht klickt.

Rechts oben ist ein Button, mit diesem können Projektleiter neue Aufgaben erstellen. Der Grund dafür ist, dass eine Aufgabe immer zu einem Unterprojekt gehört. In dieser Ansicht kann man auf einem Blick erkennen, welche Aufgaben bereits existieren und welche noch fehlen.

The screenshot shows the 'Task anlegen' form with the following fields:

- Neuer Task:** Title 'A', Person 'Friedl'.
- Das ist ein neuer Task:** Date '03.03.2024', Person 'Korkmaz'.
- Büro A:** Date '17.03.2024', Person 'Friedl'.

A blue button labeled 'Erstellen' is located at the bottom left of the form.

Abbildung 30: Aufgabe erstellen

Wenn man auf das Plus klickt, wird man auf die Seite von Abbildung 30 geleitet. Hier muss man die angegebenen Felder ausfüllen. Nachdem man auf Erstellen klickt, wird die Aufgabe in die Aufgaben-Liste eingefügt, wie man in Abbildung 31 erkennen kann.

Task Nr	Titel	Verantwortliche Person	Geplante Zeit	Tatsächliche Zeit	Zeit verbleibend	Start	Ende	Status
6	Research Data Analysis	Friedl	25 Stunden	15 Stunden	10 Stunden	05.09.2023	05.09.2023	Planned
7	Stakeholder Presentation	Friedl	20 Stunden	10 Stunden	10 Stunden	10.09.2023	10.09.2023	In Progress
8	Revised Wireframes	Friedl	25 Stunden	15 Stunden	10 Stunden	15.09.2023	15.09.2023	Planned
9	UI Implementation	Friedl	30 Stunden	20 Stunden	10 Stunden	20.09.2023	20.09.2023	In Progress
202	Neuer Task	Korkmaz	0 Stunden	0 Stunden	0 Stunden	03.03.2024	17.03.2024	InProgress

Abbildung 31: neue Aufgabe angelegt

5.2.2 Aufgabe ansehen

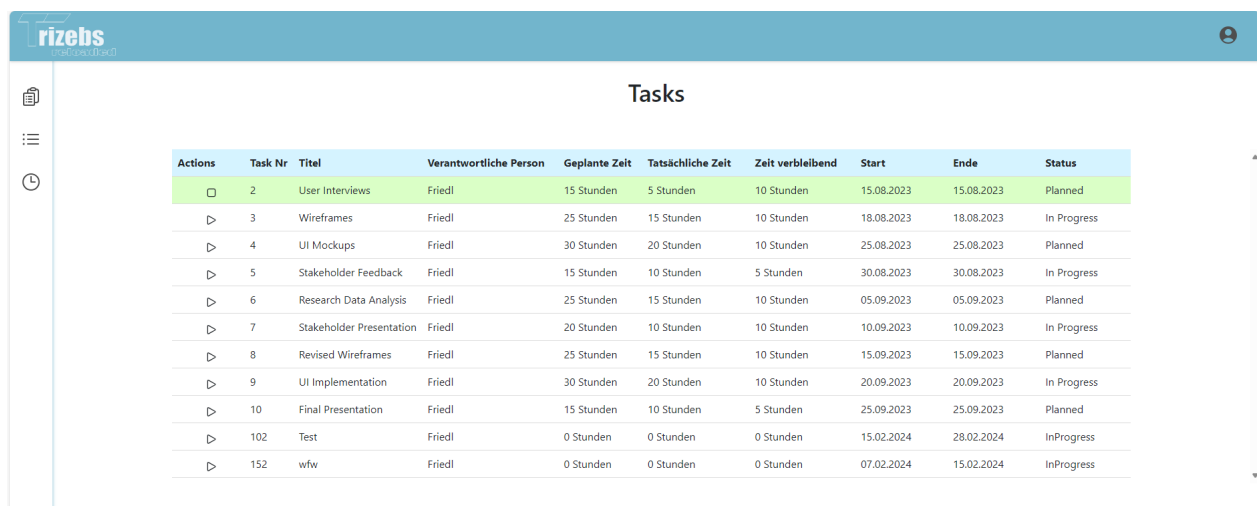
Jeder Benutzer kann über die Toolbar auf die Aufgaben-Ansicht gelangen. Wenn man kein Projektleiter ist, ist das die Standardansicht nach dem Login.



Actions	Task Nr	Titel	Verantwortliche Person	Geplante Zeit	Tatsächliche Zeit	Zeit verbleibend	Start	Ende	Status
▶	2	User Interviews	Friedl	15 Stunden	5 Stunden	10 Stunden	15.08.2023	15.08.2023	Planned
▶	3	Wireframes	Friedl	25 Stunden	15 Stunden	10 Stunden	18.08.2023	18.08.2023	In Progress
▶	4	UI Mockups	Friedl	30 Stunden	20 Stunden	10 Stunden	25.08.2023	25.08.2023	Planned
▶	5	Stakeholder Feedback	Friedl	15 Stunden	10 Stunden	5 Stunden	30.08.2023	30.08.2023	In Progress
▶	6	Research Data Analysis	Friedl	25 Stunden	15 Stunden	10 Stunden	05.09.2023	05.09.2023	Planned
▶	7	Stakeholder Presentation	Friedl	20 Stunden	10 Stunden	10 Stunden	10.09.2023	10.09.2023	In Progress
▶	8	Revised Wireframes	Friedl	25 Stunden	15 Stunden	10 Stunden	15.09.2023	15.09.2023	Planned
▶	9	UI Implementation	Friedl	30 Stunden	20 Stunden	10 Stunden	20.09.2023	20.09.2023	In Progress
▶	10	Final Presentation	Friedl	15 Stunden	10 Stunden	5 Stunden	25.09.2023	25.09.2023	Planned

Abbildung 32: Aufgaben Ansicht

In Abbildung 32 werden jedem Nutzer die eigenen Aufgaben angezeigt. So bekommt man einen guten Überblick über die jeweiligen Aufgaben und weiß, welche man noch zu erledigen hat. In der Liste befindet sich ganz links ein Start-Button, damit man zu einer Aufgabe direkt eine Zeitbuchung starten kann. Sobald man den Start-Button klickt, wird die Aufgabe, zu der man eine Zeitbuchung startet, grün markiert, wie in Abbildung 33 ersichtlich.



Actions	Task Nr	Titel	Verantwortliche Person	Geplante Zeit	Tatsächliche Zeit	Zeit verbleibend	Start	Ende	Status
☐	2	User Interviews	Friedl	15 Stunden	5 Stunden	10 Stunden	15.08.2023	15.08.2023	Planned
▶	3	Wireframes	Friedl	25 Stunden	15 Stunden	10 Stunden	18.08.2023	18.08.2023	In Progress
▶	4	UI Mockups	Friedl	30 Stunden	20 Stunden	10 Stunden	25.08.2023	25.08.2023	Planned
▶	5	Stakeholder Feedback	Friedl	15 Stunden	10 Stunden	5 Stunden	30.08.2023	30.08.2023	In Progress
▶	6	Research Data Analysis	Friedl	25 Stunden	15 Stunden	10 Stunden	05.09.2023	05.09.2023	Planned
▶	7	Stakeholder Presentation	Friedl	20 Stunden	10 Stunden	10 Stunden	10.09.2023	10.09.2023	In Progress
▶	8	Revised Wireframes	Friedl	25 Stunden	15 Stunden	10 Stunden	15.09.2023	15.09.2023	Planned
▶	9	UI Implementation	Friedl	30 Stunden	20 Stunden	10 Stunden	20.09.2023	20.09.2023	In Progress
▶	10	Final Presentation	Friedl	15 Stunden	10 Stunden	5 Stunden	25.09.2023	25.09.2023	Planned
▶	102	Test	Friedl	0 Stunden	0 Stunden	0 Stunden	15.02.2024	28.02.2024	InProgress
▶	152	wfw	Friedl	0 Stunden	0 Stunden	0 Stunden	07.02.2024	15.02.2024	InProgress

Abbildung 33: Aufgabe gestartet

5.3 Zeiterfassungen

Zeiterfassungen dienen als Übersicht der geleisteten Arbeit von den Mitarbeitern. Die Mitarbeiter und auch die Projektleiter können so erkennen, wer an welcher Aufgabe wie viel gearbeitet hat.

5.3.1 Zeiterfassungen ansehen

Um alle bisher erfassten Zeiten einzusehen, gibt es eine separate Ansicht. Diese ist über die Toolbar erreichbar und in Abbildung 34 dargestellt.

Actions	ID	Start	Ende	Dauer	Datum	Task
	3253	27.2.2024 21:54	27.2.2024 21:54	0 Minuten 14 Sekunden	27.2.2024	Stakeholder Feedback
	3254	27.2.2024 21:54	27.2.2024 21:54	0 Minuten 07 Sekunden	27.2.2024	Final Presentation
	3302	3.3.2024 18:45	3.3.2024 18:45	0 Minuten 19 Sekunden	3.3.2024	User Interviews
	3352	4.3.2024 15:14	4.3.2024 15:15	0 Minuten 21 Sekunden	4.3.2024	User Interviews
	3402	9.3.2024 13:45	9.3.2024 13:45	0 Minuten 12 Sekunden	9.3.2024	User Interviews

Abbildung 34: Zeitbuchungen - Ansicht

In der Abbildung 35 kann man erkennen, dass es ganz links einen bearbeiten-Button gibt. Mit diesem können alle Benutzer vergangene Buchungen bearbeiten.

Actions	ID	Start	Ende	Dauer	Datum	Task
<input checked="" type="checkbox"/>	3253	<input type="text" value="TT.mm.jjjj --:--"/>	<input type="text" value="TT.mm.jjjj --:--"/>	0 Minuten 14 Sekunden	27.2.2024	Stakeholder Feedback
	3254	27.2.2024 21:54	27.2.2024 21:54	0 Minuten 07 Sekunden	27.2.2024	Final Presentation
	3302	3.3.2024 18:45	3.3.2024 18:45	0 Minuten 19 Sekunden	3.3.2024	User Interviews
	3352	4.3.2024 15:14	4.3.2024 15:15	0 Minuten 21 Sekunden	4.3.2024	User Interviews
	3402	9.3.2024 13:45	9.3.2024 13:45	0 Minuten 12 Sekunden	9.3.2024	User Interviews

Abbildung 35: Zeitbuchung bearbeiten

5.3.2 Zeiterfassung starten und stoppen

In der Zeiterfassungsliste ist es möglich eine bereits vorhandene Buchung weiterzuführen. Wenn man den Start Button tätigt, läuft die Zeiterfassung weiter.









Actions	ID	Start	Ende	Dauer	Datum	Task
 	3252	27.2.2024 21:53	27.2.2024 21:55	1 Minuten 08 Sekunden	27.2.2024	User Interviews
 	3253	27.2.2024 21:54	27.2.2024 21:54	0 Minuten 14 Sekunden	27.2.2024	Stakeholder Feedback
 	3254	27.2.2024 21:54	27.2.2024 21:54	0 Minuten 07 Sekunden	27.2.2024	Final Presentation
 	3302	3.3.2024 18:45	3.3.2024 18:45	0 Minuten 19 Sekunden	3.3.2024	User Interviews

Abbildung 36: Zeitbuchung gestartet

In Abbildung 36 kann man erkennen, dass die laufende Zeiterfassung grün hinterlegt wird. Die Aufgabe, zu der die Zeiterfassung gehört, wird in der Aufgaben-Ansicht ebenfalls mit grün markiert, wie in Abbildung 37.





















Actions	Task Nr	Titel	Verantwortliche Person	Geplante Zeit	Tatsächliche Zeit	Zeit verbleibend	Start	Ende	Status
 	2	User Interviews	Friedl	15 Stunden	5 Stunden	10 Stunden	15.08.2023	15.08.2023	Planned
 	3	Wireframes	Friedl	25 Stunden	15 Stunden	10 Stunden	18.08.2023	18.08.2023	In Progress
 	4	UI Mockups	Friedl	30 Stunden	20 Stunden	10 Stunden	25.08.2023	25.08.2023	Planned
 	5	Stakeholder Feedback	Friedl	15 Stunden	10 Stunden	5 Stunden	30.08.2023	30.08.2023	In Progress
 	6	Research Data Analysis	Friedl	25 Stunden	15 Stunden	10 Stunden	05.09.2023	05.09.2023	Planned
 	7	Stakeholder Presentation	Friedl	20 Stunden	10 Stunden	10 Stunden	10.09.2023	10.09.2023	In Progress
 	8	Revised Wireframes	Friedl	25 Stunden	15 Stunden	10 Stunden	15.09.2023	15.09.2023	Planned
 	9	UI Implementation	Friedl	30 Stunden	20 Stunden	10 Stunden	20.09.2023	20.09.2023	In Progress
 	10	Final Presentation	Friedl	15 Stunden	10 Stunden	5 Stunden	25.09.2023	25.09.2023	Planned
 	102	Test	Friedl	0 Stunden	0 Stunden	0 Stunden	15.02.2024	28.02.2024	InProgress

Abbildung 37: Zeitbuchung von einer Aufgabe gestartet

6 Aufgabenverteilung

In diesem Kapitel wird festgehalten, wer welchen Teil der Diplomarbeit geschrieben hat. Einige Teile wurden von beiden Teammitgliedern gemeinsam verfasst, da es in den Aufgabenbereich beider Mitglieder fällt.

6.1 Hilal Korkmaz

- **Kurzfassung und Abstract**
- **Einleitung**
 - Ausgangssituation
 - Zielsetzung
 - * **Projekthalt**
 - Strategien zur Zielerreichung
 - * **Projektumfeld**
 - Projektteam
 - Auftraggeber
- **Theoretische Grundlagen und Methoden**
 - **Webanwendung**
 - * Client-Server-Prinzip
 - * HTTP/HTTPS
 - * Java Spring Boot
 - * Restful APIs
 - **Datenhaltung**
 - * Relationales Datenmodell
 - * Microsoft SQL Server

- **Konzept**
- **Implimentierung**
 - **Datenbank**
 - * Beschreibung
 - * Datenmodell
 - **Webanwendung**
 - * Rechte-Verwaltung
- **Ergebnis**
 - **Integration von Task**
 - * Aufgabe anlegen
 - * Aufgabe ansehen
 - **Zeiterfassungen**
 - * Zeiterfassung ansehen
 - * Zeiterfassung starten und stoppen

6.2 Elena Friedl

- **Einleitung**
 - **Projekinhalt**
 - * Anforderungen
 - * Strategien zur Zielerreichung
- **Theoretische Grundlagen und Methoden**
 - **Verwendete Technologien**
 - * Vue.js
 - * Design Science
 - * Bootstrap

- **Versionsverwaltung**

- * Azure DevOps

- **Implementierung**

- **Datenbank**

- * Datenmodell

- **Webanwendung**

- * Rest-API

- **Benutzerschnittstelle**

- * Verwaltung der Projekte
- * Verwaltung der Tasks
- * Verwaltung der Zeiterfassung

- **Ergebnis**

- **Integration von Trizebs**

- * Projekte anlegen
- * Projekte ansehen
- * Unterprojekte anlegen
- * Unterprojekte ansehen

7 Resümee

Diese Diplomarbeit ist die erste Erfahrung im Bereich vom Schreiben technischer und wirtschaftlicher Arbeiten.

Die Planung hat bereits vor den Sommerferien 2023 begonnen. Das war ein großer Vorteil für das Praktikum, welches direkt in der Woche nach Schulschluss begonnen hat. Nach ein paar Absprachen mit der Firma und dem Betreuer konnte direkt mit dem Entwickeln des technischen Teiles begonnen werden. So konnte ein großer Teil innerhalb von den fünf Wochen Praktikum erledigt werden.

Eine wichtige Erfahrung für das spätere Arbeitsleben ist die Zusammenarbeit mit dem Projektteam aus Leonding. Diese hat gezeigt, wie wichtig eine gute Projektorganisation ist, wenn mehrere Teams an verschiedenen Teilen eines Projektes arbeiten. Die größte Hürde war die unterschiedlichen Zeiträume des Praktikums. Das Projektteam aus Leonding hat eine Woche früher gestartet und eine Woche früher aufgehört und so ist es zu großen Kommunikationsschwierigkeiten gekommen.

Da der technische Teil im Praktikum nicht gänzlich fertig gestellt werden konnte, sind wir mit der Mait Austria in Kontakt getreten, um an dem Projekt weiterarbeiten zu können. Aus rechtlichen Gründen ist das ein großer Aufwand gewesen. Doch, trotz einigen Verzögerungen, hat uns das Unternehmen ein Notebook zur Verfügung gestellt, auf dem wir weiterarbeiten konnten.

Durch die Verzögerung kam es zu einer Zeitknappheit, da wir nur ein Notebook erhalten haben. So hat sich Elena vorerst auf die Fertigstellung des technischen Teiles fokussiert und Hilal auf das Schreiben der Diplomarbeit. Aufgrund dessen, dass der technische Teil vor den Semesterferien finalisiert wurde, konnten sich nun beide Teammitglieder auf das Schreiben der Arbeit konzentrieren.

8 Ausblick

Für das Projektteam ist das Projekt abgeschlossen. Die Firma Mait Austria wird jedoch noch weitere Features dazu entwickeln, wie beispielsweise einen Dokumenten-Upload und das Einfügen von Kommentaren zu Projekten, Unterprojekten und Zeiterfassungen. Nach diesen Erweiterungen wird die Firma, wie besprochen, die neue Anwendung Trizebs Reloaded statt den vorherigen Tools Trizebs und Task für ihre Projekt- und Aufgabenverwaltung sowie für die Zeitaufzeichnung produktiv im Unternehmen einsetzen.

Literaturverzeichnis

(TIL14)REST API ☐ ☐☐☐☐? URL: <https://velog.io/@wlgus2134/TIL14REST-API%EB%9E%80-%EB%AC%B4%EC%97%87%EC%9D%B8%EA%B0%80>.

Bootstrap, plain, Wortmarke, logo Symbol. URL: <https://icon-icons.com/de/symbol/bootstrap-plain-Wortmarke-logo/146620>.

Redaktion ComputerWeekly.de. *Datenhaltung*. Accessed: 13.02.2024. 2016. URL: <https://www.computerweekly.com/de/definition/Datenhaltung-Data-Retention#:~:text=Unter%20der%20Bezeichnung%20Datenhaltung%20oder,Daten%20aus%20unterschiedlichsten%20Gr%C3%BCnden%20auf>.

Redaktion ComputerWeekly.de. *Microsoft SQL Server*. Accessed: 19.02.2024. 2020. URL: <https://www.computerweekly.com/de/definition/Microsoft-SQL-Server>.

Das relationale Datenmodell - Theoretische Grundlagen I. Accessed: 13.02.2024. URL: https://dhvlab.gwi.uni-muenchen.de/wiki/Das_relationale_Datenmodell_-_Theoretische_Grundlagen_I#:~:text=Das%20relationale%20Datenmodell%20organisiert%20Informationen,k%C3%B6nnen%20auf%20Tabellen%20heruntergebrochen%20werden.

Datenhaltung. Accessed: 13.02.2024. 2015. URL: <https://www.e-teaching.org/technik/datenhaltung>.

Veikko Krypczyk Elena Bochkor. *Grundlagen des relationalen Datenbankdesigns - kompakt und klar Die Basics beherrschen*. 2019. URL: <https://entwickler.de/php/die-basics-beherrschen>.

Alexander S. Gillis. *REST API (RESTful API)*. Accessed: 19.02.2024. 2021. URL: <https://www.computerweekly.com/de/definition/RESTful-API>.

Ambient Innovation GmbH. *Vue.js*. Accessed: 08.02.2024. URL: <https://ambient.digital/wissen/ambipedia/vue-js/>.

Dr. Matthias Gottlieb. *Design Science*. Accessed: 11.02.2024. URL: <https://goitsystems.de/de/forschungsmethoden/qualitative-methoden/design-science>.

Jan-Dirk Kranz. *Was ist Bootstrap?* Accessed: 07.03.2024. 2019. URL: <https://it-talents.de/it-wissen/bootstrap/>.

Dipl.-Ing. (FH) Stefan Luber. *Was ist Azure DevOps?* Accessed: 14.02.2024. 2021. URL: <https://azure.microsoft.com/de-de/resources/cloud-computing-dictionary/what-is-devops>.

Julian Philip Marx. *Design Science Research Methodologie*. Accessed: 13.02.2024. URL: <https://shribe.de/design-science-research-methodologie/>.

Microsoft. *Was ist DevOps?* Accessed: 14.02.2024. URL: <https://azure.microsoft.com/de-de/resources/cloud-computing-dictionary/what-is-devops>.

Roland Pelayo. *What is RESTful API*. 2020. URL: <https://www.teachmicro.com/esp32-restful-api/>.

Richard Peterson. *SQL Server Architektur (Erklärt)*. Accessed: 23.02.2024. 2024. URL: <https://www.guru99.com/de/sql-server-architecture.html>.

Richard Peterson. *SQL Server Architektur (Erklärt)*. 2024. URL: <https://www.guru99.com/de/sql-server-architecture.html>.

Stick PNG. *Download Vue.js full logo transparent PNG*. URL: <https://www.stickpng.com/img/icons-logos-emojis/tech-companies/vuejs-full-logo>.

John Terra. *SQL Server Architecture Explained*. Accessed: 23.02.2024. 2023. URL: <https://www.simplilearn.com/what-is-microsoft-sql-server-architecture-article>.

Togs. *Webanwendung*. Accessed: 19.02.2024. 2023. URL: <https://de.wikipedia.org/wiki/Webanwendung>.

Übersicht über die Drei-Schichten-Architektur. Accessed: 19.02.2024. URL: https://docs.aws.amazon.com/de_de/whitepapers/latest/serverless-multi-tier-architectures-api-gateway-lambda/three-tier-architecture-overview.html.

Duke Vukadinovic. *Was ist der Unterschied zwischen HTTP und HTTPS?* Accessed: 13.02.2024. 2018. URL: <https://www.globalsign.com/de-de/blog/unterschied-zwischen-http-und-https>.

Was ist das Client-Server-Modell? Accessed: 13.02.2024. 2022. URL: <https://www.ionos.at/digitalguide/server/knowhow/client-server-modell/>.

Was ist eine relationale Datenbank? Accessed: 13.02.2024. URL: <https://cloud.google.com/learn/what-is-a-relational-database?hl=de#:~:text=Das%20relationale%20Datenbankmodell%2C%20das%20in,einer%20anderen%20Tabelle%20zu%20verkn%C3%BCpfen.>

Was ist eine REST-API? Accessed: 19.02.2024. 2024. URL: <https://www.redhat.com/de/topics/api/what-is-a-rest-api>.

Was ist Java Spring Boot? Accessed: 14.02.2024. URL: <https://www.ibm.com/de-de/topics/java-spring-boot>.

Evan You. *My Name is Evan You*. Accessed: 08.02.2024. URL: <https://evanyou.me/>.

Abbildungsverzeichnis

1	Vue.js Logo, Quelle: [18]	13
2	Bootstrap Logo, Quelle: [2]	16
3	Grafische Darstellung Client-Server Prinzip, Quelle: [15]	18
4	Grafische Darstellung von Rest API, Quelle: [1]	19
5	Architektur einer relationalen Datenbank, Quelle: [7]	22
6	Architektur von SQL-Server, Quelle: [17]	24
7	Anwendung Trizebs (oben), Konzept Trizebs Reloaded (unten)	27
8	Anwendung Task (oben), Konzept Trizebs Reloaded(unten)	29
9	Anwendung Task (oben), Konzept Trizebs Reloaded(unten)	30
10	Use Case Diagramm zur Benutzeranmeldung	31
11	Use Case Diagramm zur Projektverwaltung	32
12	Use Case Diagramm zur Aufgabenverwaltung	33
13	Use Case Diagramm zur Zeiterfassung	33
14	Use Case Diagramm zur Erstellung einer neuen Zeiterfassung	34
15	Konzept Datenmodell	36
16	Grafische Abbildung der Architektur	37
17	ERD Version 1	39
18	ERD Version 2	40
19	ERD finale Version	41
20	Rollen-Tabelle Teil 1	43
21	Rollen-Tabelle Teil 2	43
22	Frontend Ordner Struktur	59
23	Projekt anlegen Ansicht	72
24	Liste aller zuletzt gebuchten und aller Projekte die man leitet	73
25	Liste aller zuletzt gebuchten Projekte	73
26	Detail Ansicht eines Projektes und Liste der Unterprojekte dazu	74
27	Projekt suchen.	74
28	Unterprojekt anlegen Ansicht	75
29	Aufgaben eines Unterprojektes	76

30	Aufgabe erstellen	77
31	neue Aufgabe angelegt	77
32	Aufgaben Ansicht	78
33	Aufgabe gestartet	78
34	Zeitbuchungen - Ansicht	79
35	Zeitbuchung bearbeiten	79
36	Zeitbuchung gestartet	80
37	Zeitbuchung von einer Aufgabe gestartet	80
38	Diplomarbeitsplakat	94

Code Listings

1	Model Klasse von Project	42
2	createUser Methode	45
3	findUser Methode	45
4	updateUser Methode	45
5	deleteUser Methode	46
6	createProject Methode	46
7	findProject Methode	47
8	updateProject Methode	47
9	listAllProjects Methode	47
10	listAllProjectsManagedBy Methode	48
11	deleteProject Methode	48
12	createSubProject Methode	48
13	findSubProject Methode	49
14	updateSubProject Methode	49
15	deleteSubProject Methode	49
16	createTask Methode	50
17	findTask Methode	50
18	getAllTasksForRespPerson Methode	50
19	updateTask Methode	51
20	deleteTask Methode	51
21	createTimeTracking Methode	52
22	findTimeTracking Methode	52
23	listAllTimeTrackings Methode	52
24	updateTimeTracking Methode	53
25	deleteTimeTracking Methode	53
26	Liste der Zuletzt gebuchten Projekte	60
27	Liste der Projekte die man leitet	61
28	Methoden der Projekt listen	62
29	Methoden für ein Projekt und Unterprojekte	63

30	submitProject Methode	64
31	submit SubProject Methode	65
32	Liste aller Aufgaben	66
33	Methode der Aufgaben eines Unterprojektes	66
34	submitTask Methode	67
35	start und stop Task	68
36	startTask Methode	69
37	stopTask Methode	70
38	TimeTracking bearbeiten	71

Tabellenverzeichnis

1	User-Endpoint Tabelle	54
2	Project-Endpoint Tabelle	55
3	Subproject-Endpoint Tabelle	56
4	Task-Endpoint Tabelle	57
5	Task-Endpoint Tabelle	58

Anhang



Abbildung 38: Diplomarbeitplakat