

# **Project.ONE**

**IT-Projekt Sommersemester 2022**

***Zeiterfassungssystem***

**Dokumentation und Handbuch**

Hochschule der Medien Stuttgart, Fakultät Information und Kommunikation

Wirtschaftsinformatik und Digitale Medien

## **Betreuer**

Prof. Dr. Peter Thies

Prof. Dr. Christoph Kunz

## **Gruppe 10**

Elif Demir

Malek Bouaziz

Quang Cuong Dong

Harbin Tairi

Abdul Aziz Muslim Zai

Murad Zadran

<b>Inhaltsverzeichnis</b>	2
<b>Abbildungsverzeichnis</b>	4
<b>1. Anforderungsspezifikation</b>	
1.1. Rahmenbedingungen	5
1.2. Zielsystem	5
<b>2. Vorgehensweise und Erläuterung der Meilensteine</b>	
2.1. Meilenstein II	6
2.2. Meilenstein III	6
2.3. Meilenstein IV	6
2.4. Meilenstein V	6
<b>3. Meilenstein II: Use-Cases</b>	
3.1. Anmeldung	7
3.2. Profilerstellung und -bearbeitung	8
3.3. Zeiterfassung	9
3.4. Abwesenheit	10
3.5. Stundenübersicht	11
3.6. Projektkontrolle	12
<b>4. Mock-Ups</b>	13
<b>5. Meilenstein III: ERM- und UML-Klassendiagramme</b>	
5.1. ERM- und UML-Klassendiagramm	15
5.2. Entitäten und Attribute	16
5.2.1. Entität Benutzer	16
5.2.2. Entität Arbeitszeitkonto	16
5.2.3. Entität Projekt	16
5.2.4. Entität Zeitintervall	16
5.2.5. Entität Zeitintervallbuchung	17
5.2.6. Entität Ereignis	17
5.2.7. Entität Ereignisbuchung	17
5.2.8. Entität Buchung	17
5.2.9. Entität Projektarbeit	17
5.3. Entität Pause	17
5.3.1. Entität Abwesenheit	17
5.3.2. Entität Kommen	17

5.3.3. Entität Gehen	17
5.3.4. Entität Aktivitäten	18
5.3.5. Entität Membership	18
5.4. Datenbank	18
5.4.1. UML-Klassendiagramm	18
5.3. Datenbank	18
5.4. UML-Klassendiagramm	18
5.5. Überblick Klassendiagramme	19
<b>6. Installationsanleitung</b>	
6.1. Vorausgesetzte Programme	23
6.2. GitHub Repository	23
6.3. Applikation Online zur Verfügung stellen	22
6.4. Anlegen einer Datenbank mit Google Cloud	22
<b>7. Benutzerhandbuch</b>	
7.1. Account erstellen und löschen	24
7.2. Erstellen und Bearbeiten des Profils	25
7.3. Erstellen, Bearbeiten und Löschen von Projekten und den verknüpften Inhalten	25
7.3.1. Mitarbeiter einem Projekt zuweisen und entfernen	27
7.3.2. Aktivitäten erstellen, bearbeiten und löschen	27
7.3.3. Projektarbeiten erstellen, bearbeiten und löschen	28
7.4. Buchungen durchführen, bearbeiten und löschen	28
7.5. Arbeitszeitkonto Funktionen	30
<b>8. Literaturverzeichnis</b>	31

## **Abbildungsverzeichnis**

Figure 1: Use Case Anmeldung.....	7
Figure 2: Beschreibung Use Case Profilverwaltung .....	7
Figure 3: Use Case Profilerstellung und - Bearbeitung .....	8
Figure 4: Beschreibung Use Case Profilerstellung und - Bearbeitung .....	8
Figure 5: Use Case Zeiterfassung .....	9
Figure 6: Beschreibung Use Case Zeiterfassung .....	9
Figure 7: Use Case Abwesenheit .....	10
Figure 8: Beschreibung Use Case Abwesenheit .....	10
Figure 9: Use Case Stundenübersicht.....	11
Figure 10: Beschreibung Use Case Zeiterfassung .....	11
Figure 11: Use Case Projektkontrolle .....	12
Figure 12: Beschreibung Use Case Projektkontrolle .....	12
Figure 13: Mockup Projektansicht/Arbeitszeiten.....	13
Figure 14: Mockup Projektdetailansicht/Aktivitäten .....	13
Figure 15: Mockup Abwesenheit Historie und Abwesenheit eintragen.....	14
Figure 16: Mockup Profilansicht .....	14
Figure 17: ERM Diagramm V1.....	15
Figure 18: ERM Diagramm V2.....	16
Figure 19: UML Klassendiagramm Übersicht.....	19
Figure 20: Klassendiagramm Zeitkonto .....	19
Figure 21: Klassendiagramm Projektbeteiligung .....	20
Figure 22: Klassendiagramm Projektstruktur.....	20
Figure 23: Klassendiagramm Zeitereignis .....	21
Figure 24: Klassendiagramm Buchungen.....	21
Figure 25: Mapperklassen .....	22
Figure 26: Account erstellen und löschen.....	24
Figure 27: Anmeldung google Account.....	24
Figure 28: Erfolgreiche anmeldung mit google Account .....	25
Figure 29: Account Funktionen.....	25
Figure 30: Projekt erstellen.....	26
Figure 31: Projekt anlegen.....	26
Figure 32: Projekt bearbeiten .....	26
Figure 33: Mögliche Funktionen innerhalb des Projekts .....	27
Figure 34: Mitarbeiter zum Projekt hinzufügen .....	27
Figure 35: Aktivitäten erstellen, bearbeiten und löschen .....	27
Figure 36: Projektarbeiten erstellen, bearbeiten und löschen .....	28
Figure 37: Buchung erfassen.....	28
Figure 38: Arbeitsbeginn erfassen.....	29
Figure 39: Arbeitsende erfassen.....	29
Figure 40: Pause erfassen.....	29
Figure 41: Funktionen des Arbeitszeitkontos.....	30

## **1. Anforderungsspezifikation**

### **1.1. Rahmenbedingungen**

Im Rahmen der Lehrveranstaltung “Software-Praktikum”, sollten sich für das Projekt maximal 6 Personen zusammenschließen. Die Umsetzung wurde als kollektive Aufgabenstellung formuliert, d.h. dass über die bereits gesammelten Erkenntnisse im Verlauf des Studiums und die gelernten Inhalte in Modulen wie unter anderem Programmieren, Algorithmen und Datenstrukturen, Datenbank Grundlagen, Software-Engineering, Web-Technologie oder auch die Schlüsselkompetenz-Module wie beispielsweise Projektmanagement, Führung und Teamarbeit o.ä., welche als Grundgerüst verwendet werden sollten, wurde die Vorgabe an die Gruppen adressiert, ein System zur kollaborativen Zeiterfassung und Auswertung von Projektarbeiten in Form einer Webapplikation zu realisieren. Das System soll die Arbeitszeiterfassung durch Projektbeteiligte sowie deren Auswertung durch Benutzer ermöglichen. Die Applikation soll Personen unterstützen, ihre Arbeitsleistung zu Projekten und deren Aktivitäten zu verwalten.

Die Phasen der Planung und Organisation orientierten sich an sieben Meilensteine, die man im Verlauf des Semesters fortlaufend abzugeben hat. Die ersten Meilensteine, die in Form von Use Cases, Klassendiagramme und ER-Diagramme durchzuführen sind, dienen dazu, eine intensive Anforderungsanalyse durchzuführen und so ein Grundgerüst für den Source Code und folglich auch für die Systempräsentation in dieser Dokumentation zu erhalten.

Für die Ausführung der praktischen Umsetzung wurde Python und JavaScript sowie MySQL, HTML, CSS, React, Github, Firebase und Frameworks (Flask, Flask-RestX, React) vorausgesetzt.

Im Laufe des Projektes stehen zwei Lehrbeauftragte für Fragen sowie Unterstützung bereit. Die Unklarheiten oder der Fortschritt sollen anhand von Coaching-Termine in Form von Feedbacks bekannt gegeben werden, um mögliche Risiken sowie Unstimmigkeiten unter den Gruppen und im Projekt zu vermeiden.

### **1.2 Zielsystem**

Für eine solche Funktionsweise müssen dem System die Begriffe Person, Buchung, Arbeitszeitkonto, Projekt, Aktivität, Ereignis, Ereignisbuchung, Zeitintervall, Zeitintervallbuchung sowie deren Verknüpfungen bekannt sein. Es sind spezielle Subklassen von Ereignis wie Kommen und Gehen und spezielle Subklassen von Zeitintervall wie Pause und Projektarbeit zu berücksichtigen. Zudem sind geeignete Superklassen zu finden, die die Gemeinsamkeiten der oben genannten Begriffe zusammenfassen.

Objekte der vorgenannten Begriffe (Klassen) sollen in dem Zielsystem jederzeit durch dazu jeweils Berechtigte angelegt, gelesen, editiert und gelöscht werden können.

Für jedes Element kann eine Reihe von Attributen festgelegt werden. Dies sind für alle: Datum der letzten Änderung und ID; für Person zusätzlich der Vor- und Nachname (Text), Mail-Adresse und Benutzername; bei Projekt zusätzlich eine Bezeichnung und der Auftraggeber (jeweils als Zeichenkette); bei Aktivität ebenfalls eine Bezeichnung (Zeichenkette) und die Kapazität in Personentagen

(Fließkommazahl); bei Ereignis ein Zeitpunkt, an dem jenes Ereignis integriert). Gemeinsame Attribute werden dabei vorteilhaft in einer gemeinsamen Basisklasse definiert. Hieraus ergeben sich mithin miteinander verbundene, abstrakte Basisklassen wie etwa BusinessObject.

## 2. Vorgehensweise und Erläuterung der Meilensteine

### 2.1 Meilenstein I

Als Bedingung der Projektdurchführung wurde vorausgesetzt, dass sich die Studierende in eine **Gruppe eintragen**. Die Gruppen müssen ihre Zusammensetzung spätestens eine Woche nach Bekanntgabe der Aufgabenstellung den Lehrbeauftragten verbindlich über das Moodle-System mitteilen. Die Gruppengröße sollte nicht die Anzahl an sechs Personen übertreffen. Die Aufgabe ist so dimensioniert, dass sie auch bei Ausscheiden von Teammitgliedern durch vier Personen erfolgreich bearbeitet werden kann.

### 2.2 Meilenstein II

Damit die Funktionen des Systems aus der Sicht des Anwenders klar dargestellt werden, visualisiert man verschiedene Szenarien der Systemanwendung in Form von **Use-Case-Diagrammen**. Die Diagramme sind typischerweise sehr repräsentativ für die klassische und spezifische Situationen, in denen Benutzer erleben, wann und wie die betreffende Anwendung zu verwendet ist. Die Diagramme, sollten mit Erstellungs-Tools wie Star-UML erstellt werden.

### 2.3 Meilenstein III

Die Perspektive aus Sicht des Entwicklers sollte in diesem Meilenstein durch **Klassen- sowie ER-Diagramme** modelliert werden. Die Vorgehensweise um ein Klassendiagramm zu entwerfen wurde im Modul “Software-Engineering” intensiv behandelt. In Bezug der einzelnen Beziehungstypen, die sich aus den Entitäten der Webapplikation ergeben soll, ist ein Entity-Relationship-Diagramm zu entwerfen. Dadurch soll eine Typisierung von Objekten und ihre relationalen Beziehungen zueinander stattfinden.

### 2.4 Meilenstein IV

Der **Source-Code** mit all seinen Pfaden und Ordnerstrukturen sollte als Zip-Datei zur Verfügung gestellt werden. Dabei sollten alle Bestandteile realisiert sein und ein vollständiger Funktionsumfang gegeben sein. Mithilfe des Git-hub Accounts wurde die Umsetzung durchgeführt. Von dort aus wurde das entsprechende Repository als Code gespeichert und den Dozenten zur Verfügung gestellt.

### 2.5 Meilenstein V

Ein Überblick der eingereichten Ergebnisse sollte mithilfe einer **Präsentation** wiedergegeben werden und alle geforderten Funktionen zu Demonstrationszwecken den Lehrbeauftragten vorgestellt werden. Die Teilnahme aller Gruppenmitglieder wurde hierfür festgelegt. Ausschlaggebend für die erfolgreiche Präsentation sollte die Tiefe des Projektes und der vollständige Funktionsumfang sein.

### 3. Meilenstein II: Use-Cases

#### 3.1. Anmeldung

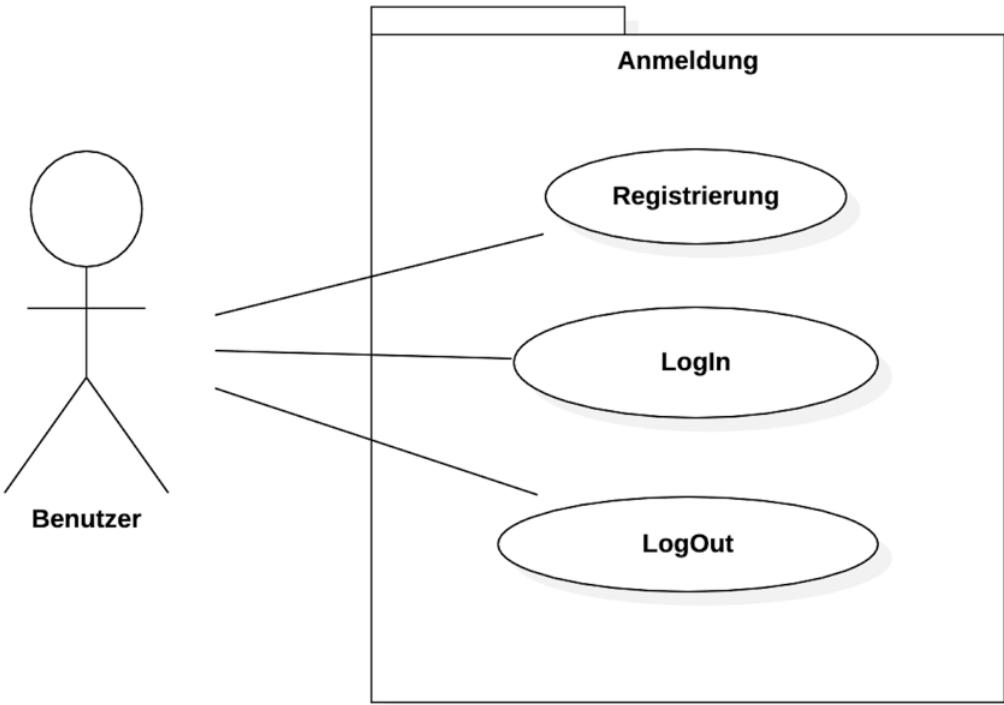


Figure 1: Use Case Anmeldung

Name	Registrierung	Log In	Log Out
Ziel	Benutzer hat erfolgreich einen Account erstellt und ist angemeldet.	Der Benutzer loggt sich mit seinem Profil ein	Der Benutzer meldet sich aus der App ab
Vorbedingung	Der Benutzer ist ein Projektmitglied und besitzt einen Google Account.	Der Benutzer hat sich bereits registriert	Der Benutzer hat sich registriert und ist angemeldet
Nachbedingung	Der Benutzer hat einen Account angelegt und ist angemeldet.	Der Benutzer hat Zugriff auf sein Profil und alle damit verbundenen Funktionen	Der Benutzer ist nicht mehr mit seinem Profil angemeldet
Nachbedingung Sonderfall	-	1. Account existiert noch nicht 2. Benutzer registriert sich 3. Benutzer trägt Vorname, Nachname, Benutzername ein	-
Akteure	Benutzer		
Normalablauf	1. Benutzer registriert sich 2. Benutzer trägt Vorname, Nachname, Benutzername ein.	1. Der Benutzer loggt sich mit seiner E-Mail und dem Passwort ein 2. Der Benutzer ist in seinem Account eingeloggt	1. Benutzer hat sich erfolgreich mit seinem Profil angemeldet 2. Benutzer drückt auf Log Out Button 3. Benutzer ist nicht mehr angemeldet
Ablauf im Sonderfall	1. E-Mail ist ungültig 2. Google Fehlermeldung tritt auf	Der Benutzer erstellt einen neuen Account	--

Figure 2: Beschreibung Use Case Profilverwaltung

### 3.2. Profilerstellung und -bearbeitung

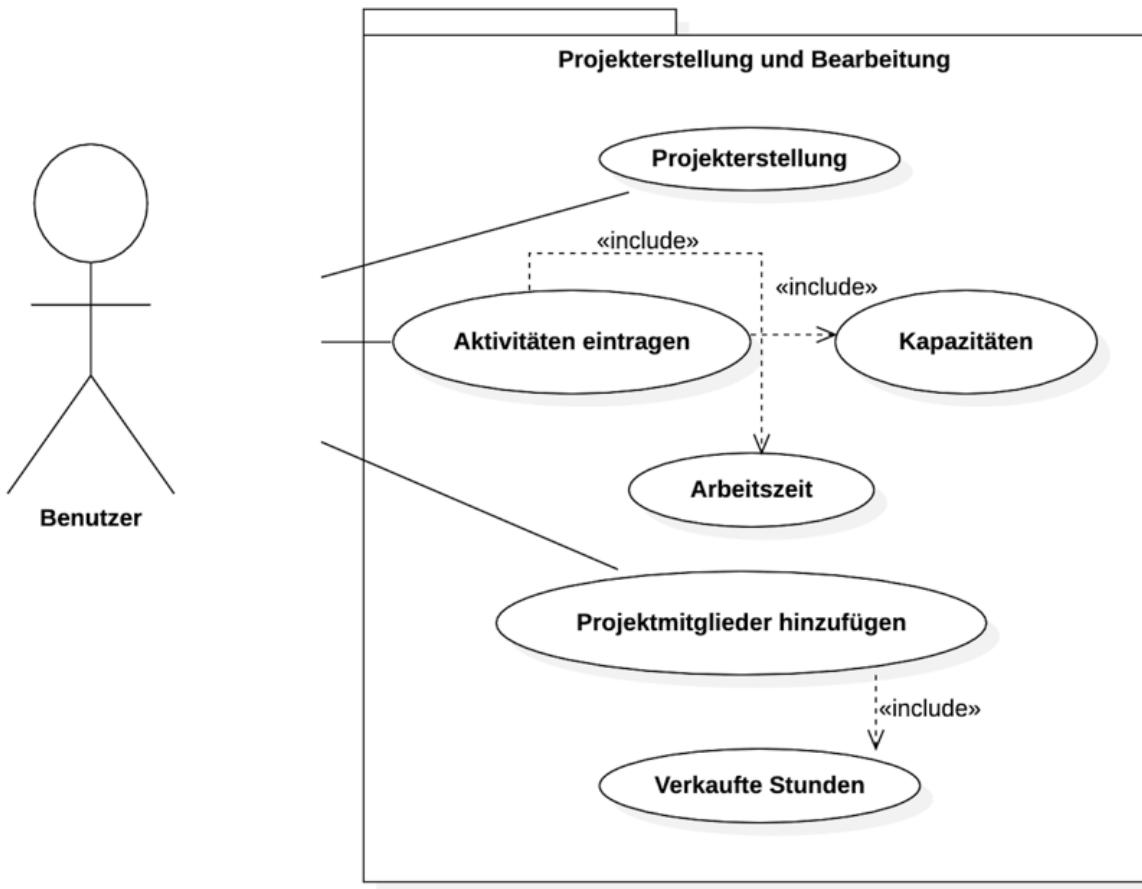


Figure 3: Use Case Profilerstellung und - Bearbeitung

Name	Projekterstellung	Aktivitäten eintragen	Kapazitäten	Arbeitszeit	Projektmitglieder hinzufügen	Verkaufte Stunden
Ziel	Benutzer hat erfolgreich ein Projekt erstellt	Projektleiter trägt seine Aktivitäten ein	Projektleiter trägt seine Kapazität ein	Projektleiter trägt die vorgesehene Arbeitszeit für die Aktivität ein	Projektmitglieder wurden dem Projekt mit den verkauften Stunden hinzugefügt	Projektleiter teilt den Projektmitgliedern ihre verkauften Stunden zu
Vorbedingung	Benutzer hat einen Account und ist eingeloggt	Projektleiter ist in seinem Account eingeloggt und hat zu der Aktivität die Kapazität und Duration eingetragen.	Benutzer ist Projektleiter in dem Projekt	Benutzer ist Projektleiter in dem Projekt	Benutzer ist Projektleiter in dem Projekt und hat die Projektmitglieder mit den jeweiligen Stunden eingetragen.	Benutzer ist Projektleiter in dem Projekt
Nachbedingung	Benutzer erstellt ein Projekt mit Namen und ist der Projektleiter	Projektleiter hat erfolgreich seine Aktivitäten eingetragen	Projektleiter hat die Kapazität der Aktivität bearbeitet	Projektleiter hat die vorgesehenen Arbeitszeit in dem Projekt bearbeitet	Benutzer wurden mit den verkauften Stunden als Projektmitglieder dem Projekt hinzugefügt	Zu jedem eingetragenen Projektmitglied wurden die verkauften Stunden eingetragen
Nachbedingung Sonderfall						
Akteure				Projektleiter		
Normalablauf	1. Benutzer loggt sich ein 2. Benutzer erstellt ein Projekt und vergibt einen Namen 3. Benutzer wird Projektleiter	1. Projektleiter trägt die Kapazität und Dauer zu der Aktivität ein 2. Projektleiter fügt die Aktivität dem Projekt hinzu	1. Projektleiter bearbeitet die Aktivität des Projekts und speichert.	1. Projektleiter bearbeitet die vorgesehene Arbeitszeit des Projekts und speichert.	1. Projektleiter fügt Projektmitglieder und die verkauften Stunden dem Projekt zu 2. Projektleiter speichert die Änderung	1. Projektleiter fügt Projektmitglieder und die verkauften Stunden dem Projekt zu
Ablauf im Sonderfall						

Figure 4: Beschreibung Use Case Profilerstellung und - Bearbeitung

### 3.3. Zeiterfassung

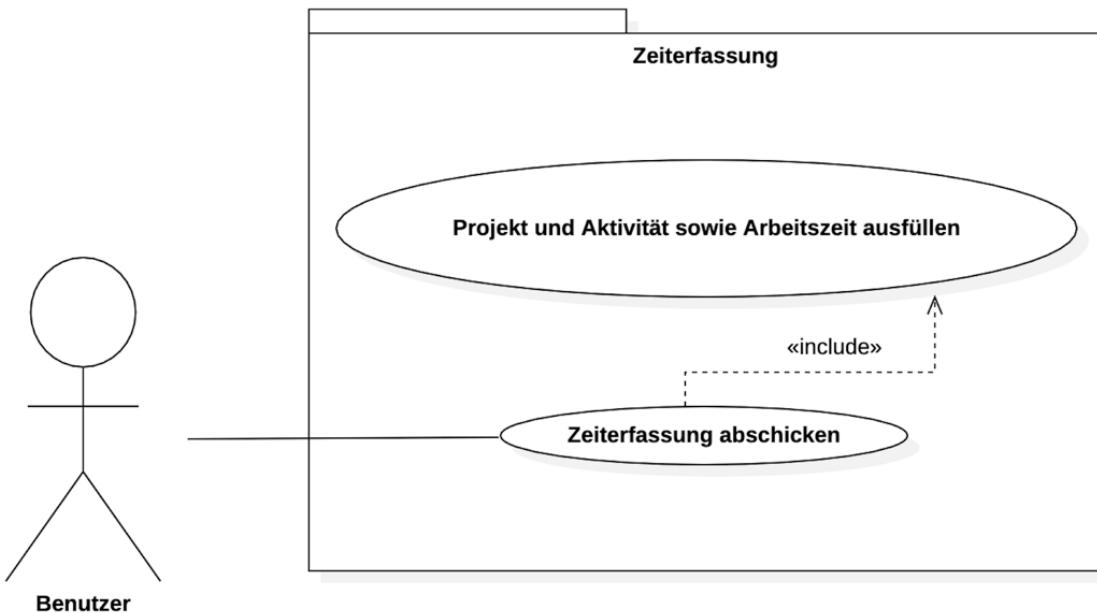


Figure 5: Use Case Zeiterfassung

Name	Projekt und Aktivität sowie Arbeitszeit ausfüllen	Zeiterfassung abschicken
Ziel	Projektmitglied hat seine Brutto Arbeitszeit zu der Aktivität eingetragen	Die eingegebene Zeit wird der Aktivität zugeordnet und gespeichert
Vorbedingung	Benutzer ist Projektmitglied in dem Projekt	Benutzer ist Projektmitglied in dem Projekt und hat seine Brutto Arbeitszeit zu der Aktivität eingetragen.
Nachbedingung	Brutto Arbeitszeit wurde zu der Aktivität eingetragen	-
Nachbedingung Sonderfall		-
Akteure	Projektmitglied	
Normalablauf	1. Projektmitglied wählt sein Projekt und seine Aktivität aus zu der er eine Zeit erfassen möchte 2. Projektmitglied trägt seine eingetragene Zeit ein	1. Projektmitglied speichert die Eingaben
Ablauf im Sonderfall		-

Figure 6: Beschreibung Use Case Zeiterfassung

### 3.4. Abwesenheit

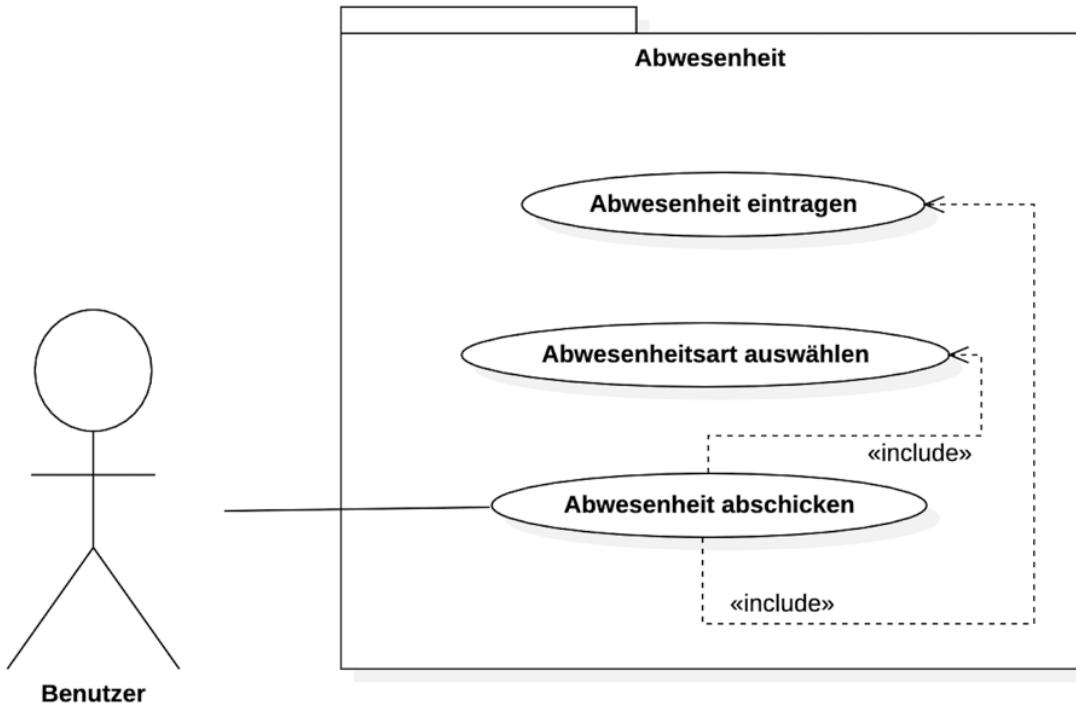


Figure 7: Use Case Abwesenheit

Name	Abwesenheit abschicken	Abwesenheitsart auswählen	Abwesenheit eintragen
Ziel	Projektmitglied trägt seine Abwesenheitsdauer und Abwesenheitsart ein	Projektmitglied trägt seine Abwesenheitsart ein	Projektmitglied trägt seine Abwesenheit ein
Vorbedingung	Projektmitglied plant Abwesenheit	Projektmitglied plant Abwesenheit	Projektmitglied plant Abwesenheit
Nachbedingung	Abwesenheitsdauer und Abwesenheitsart wurden eingetragen	Abwesenheitsart wurde eingetragen	Abwesenheitsdauer wurde eingetragen
Nachbedingung Sonderfall	-		
Akteure	Projektmitglied		
Normalablauf	1. Projektmitglied trägt Abwesenheit ein 2. Projektmitglied wählt Abwesenheitsart 3. Projektmitglied schickt Abwesenheit ab	1. Projektmitglied wählt Abwesenheitsart	1. Projektmitglied trägt Abwesenheitsdauer ein
Ablauf im Sonderfall	-		

Figure 8: Beschreibung Use Case Abwesenheit

### 3.5. Stundenübersicht

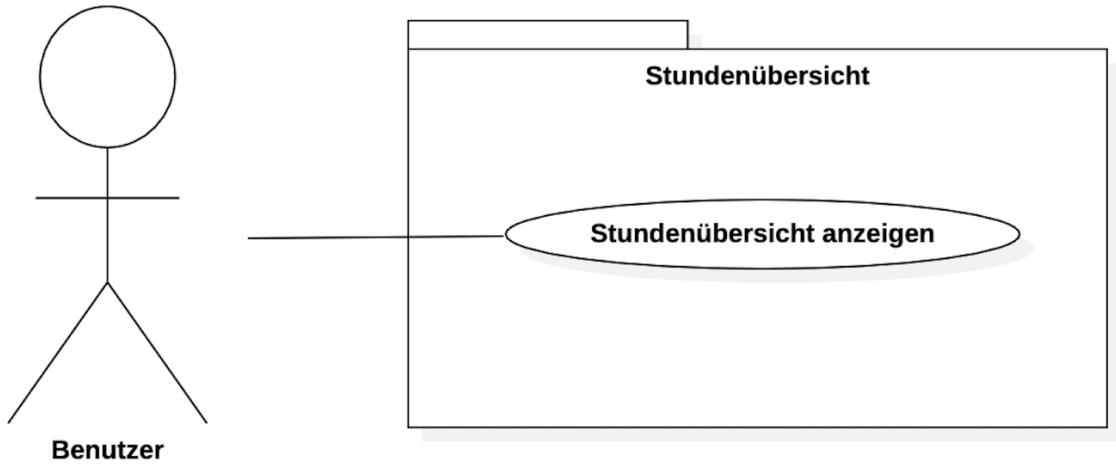


Figure 9: Use Case Stundenübersicht

Name	Stundenübersicht anzeigen
Ziel	Stundenübersicht zu dem eingeloggten Benutzer wird angezeigt
Vorbedingung	Benutzer ist eingeloggt und hat auf die Stundenübersicht gedrückt
Nachbedingung	-
Nachbedingung Sonderfall	-
Akteure	Benutzer
Normalablauf	1. Benutzer klickt auf "Stundenübersicht" 2. Stundenübersicht zu jedem Projekt wird mit Aktivität angezeigt
Ablauf im Sonderfall	-

Figure 10: Beschreibung Use Case Zeiterfassung

### 3.6. Projektkontrolle

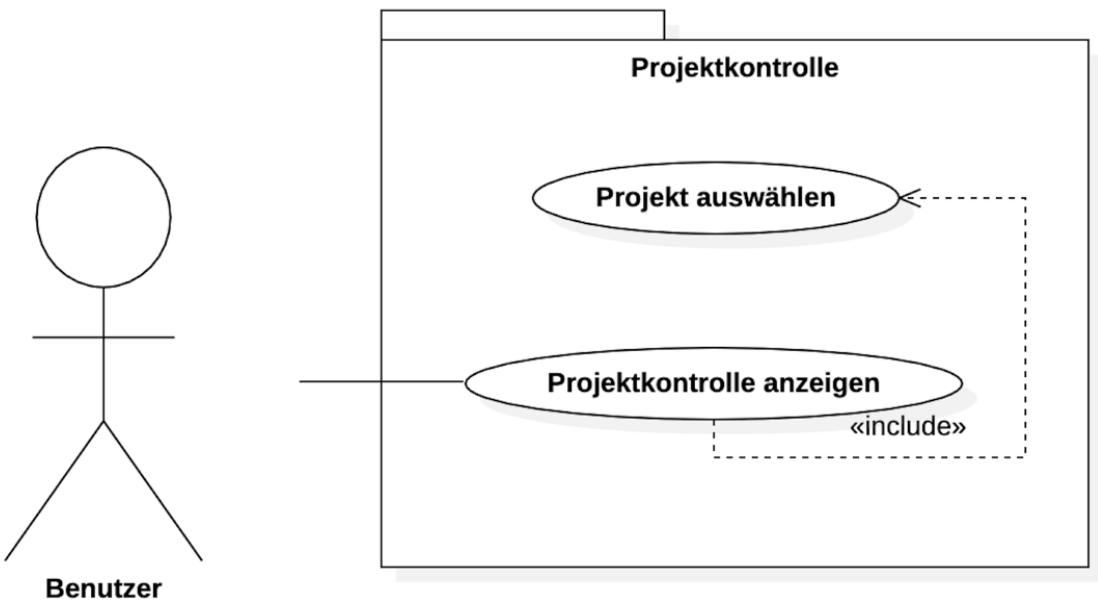


Figure 11: Use Case Projektkontrolle

Name	Projektkontrolle anzeigen	Projekt auswählen
Ziel	Projektkontrolle zu dem gewünschten Projekt wird angezeigt	Gewünschtes Projekt wird ausgewählt
Vorbedingung	Benutzer ist eingeloggt und hat sein gewünschtes Projekt ausgewählt	Benutzer ist eingeloggt
Nachbedingung	-	-
Nachbedingung Sonderfall	-	-
Akteure	Benutzer	
Normalablauf	1. Benutzer hat sein gewünschtes Projekt ausgewählt 2. Dem Benutzer wird die Projektkontrolle zu dem ausgewählten Projekt angezeigt	1. Benutzer wählt per Drop-Down-Menü ein Projekt aus
Ablauf im Sonderfall	-	-

Figure 12: Beschreibung Use Case Projektkontrolle

#### 4. Mock-Ups

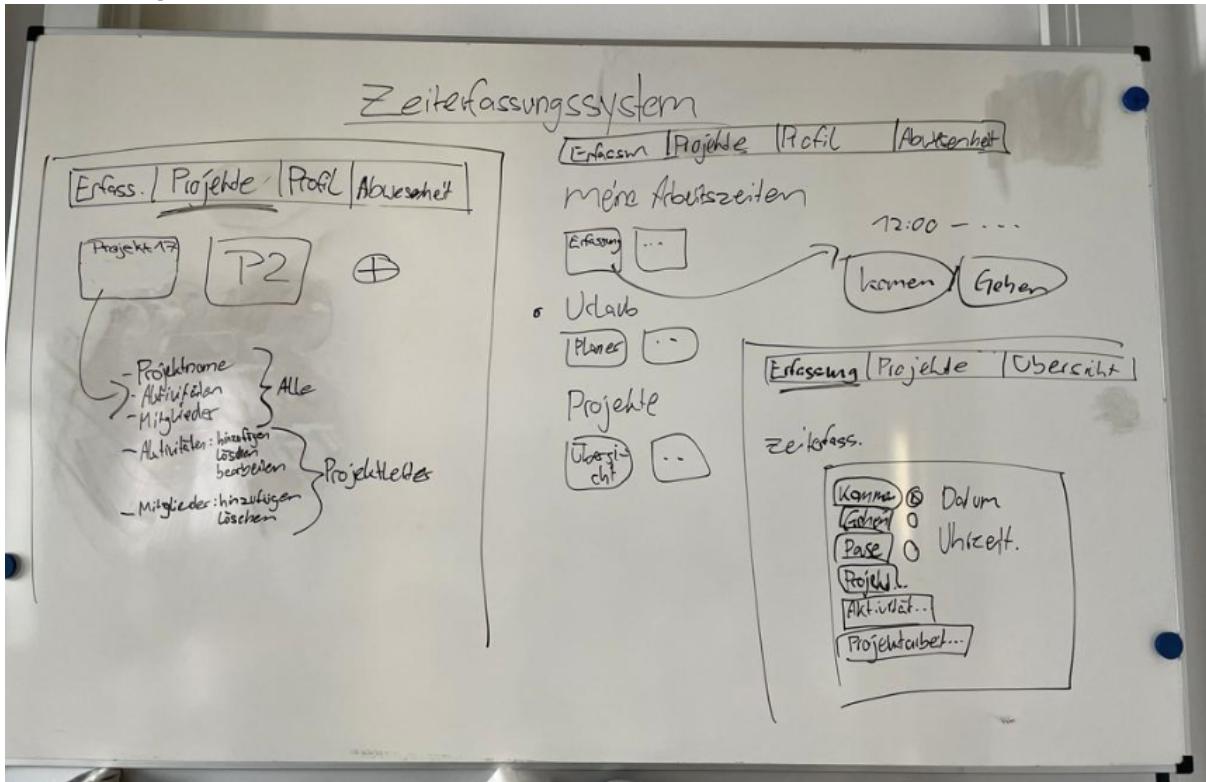


Figure 13: Mockup Projektansicht/Arbeitszeiten

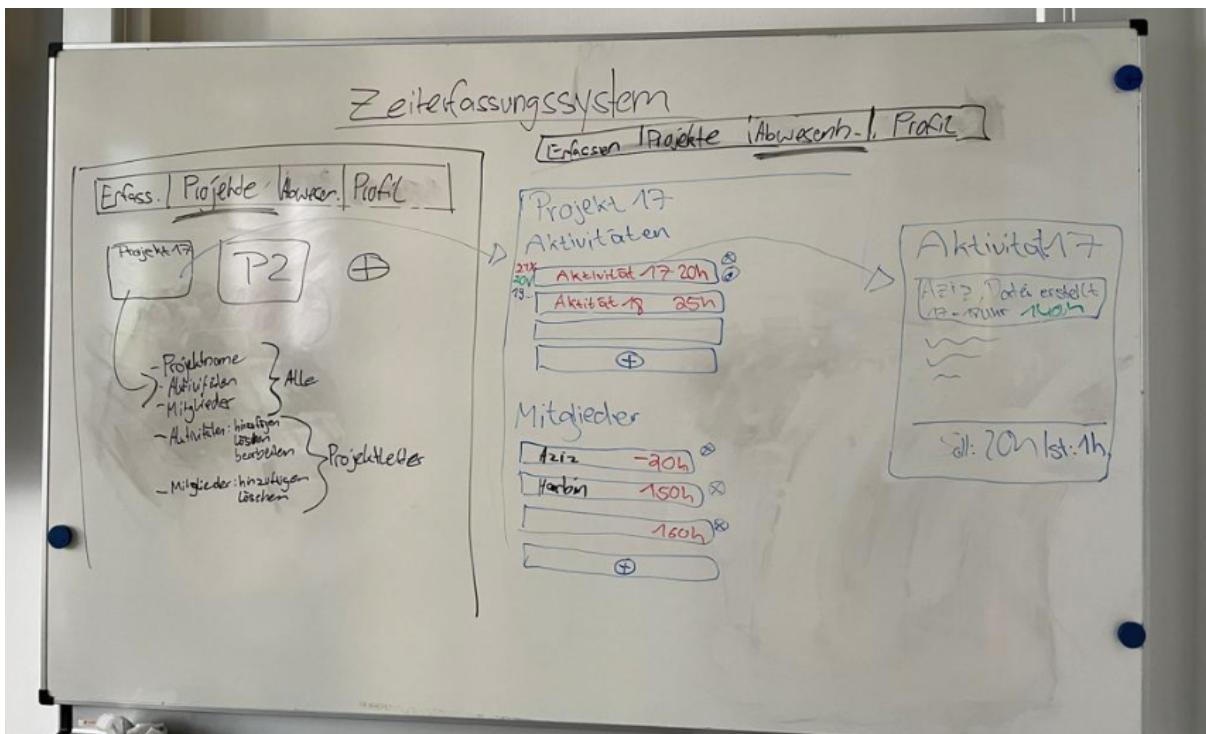


Figure 14: Mockup Projektdetailansicht/Aktivitäten

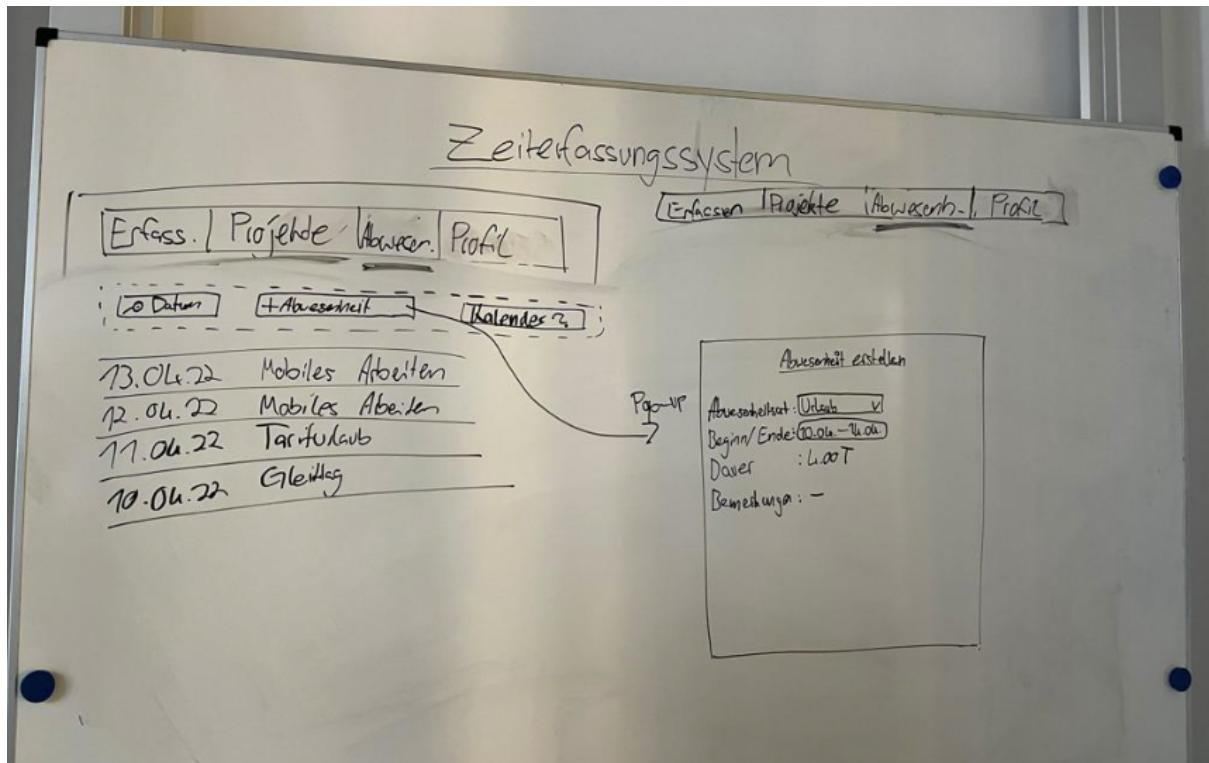


Figure 15: Mockup Abwesenheit Historie und Abwesenheit eintragen

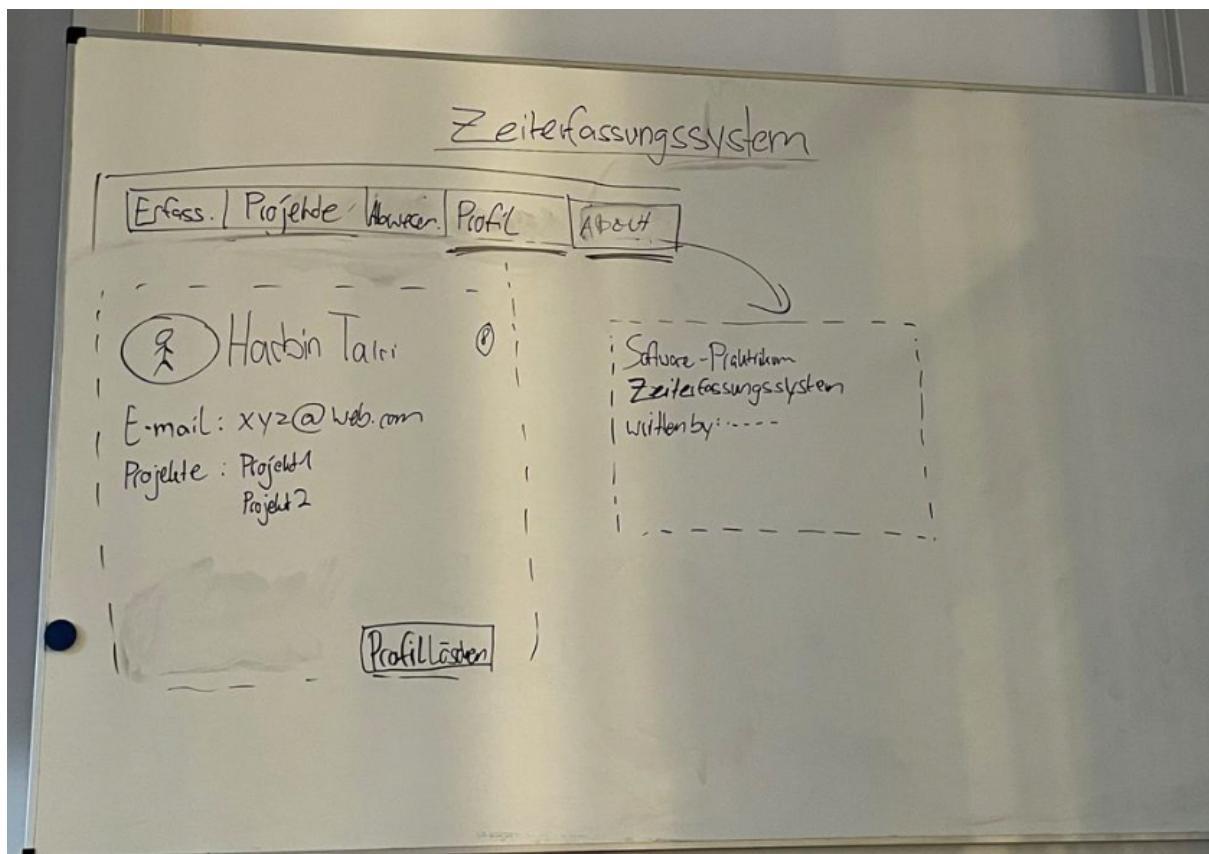


Figure 16: Mockup Profilansicht

## 5. Meilenstein III: ERM und UML-Klassendiagramme

### 5.1. ERM und UML-Klassendiagramm

In Meilenstein III mussten wir sowohl ER-Modelle als auch Klassendiagramme für unser Projekt erstellen. Darin wurden alle Entitäten, Beziehungen, Kardinalitäten und Attribute (enthält Primär – und Fremdschlüsse) abgebildet, welche für unser Projekt notwendig waren. Mithilfe des ERM wird der logische Aufbau einer Datenbank dargestellt. Dabei war zu beachten, dass im ER-Modell die Chen-Notation angewendet wird.

Nachfolgend wurde das UML-Klassendiagramm erstellt, bei dem sich die Klassen durch die Entitäten ergeben. Dieses besteht aus Klassen, Assoziationen, Kardinalitäten und Rollenbezeichnungen. Das Klassendiagramm bietet uns eine schnelle Übersicht über das Zusammenspiel der einzelnen Systemelemente und deren Eigenschaften und Beziehungen untereinander. Das Unified Modeling Language Diagramm wurde mithilfe von StarUML generiert. Anschließend wurde das ER-Modell erläutert und die einzelnen Entitäten mit ihren Attributen aufgeführt. Daraufhin wurde die verwendete Datenbank vorgestellt. Schlussendlich wurde der Meilenstein III mit einer ausführlichen Auflistung und Ausarbeitung der UML-Klassendiagramme abgeschlossen.

Zu Beginn des Projekts wurde ein ER-Diagramm erstellt, welches unter Version 1 zu sehen ist. Im Laufe des Projekts sind jedoch Änderungen aufgetreten. Version 1 wurde daher angepasst und das finale Modell ist nun unter Version 2 zusehen.

#### Version 1:

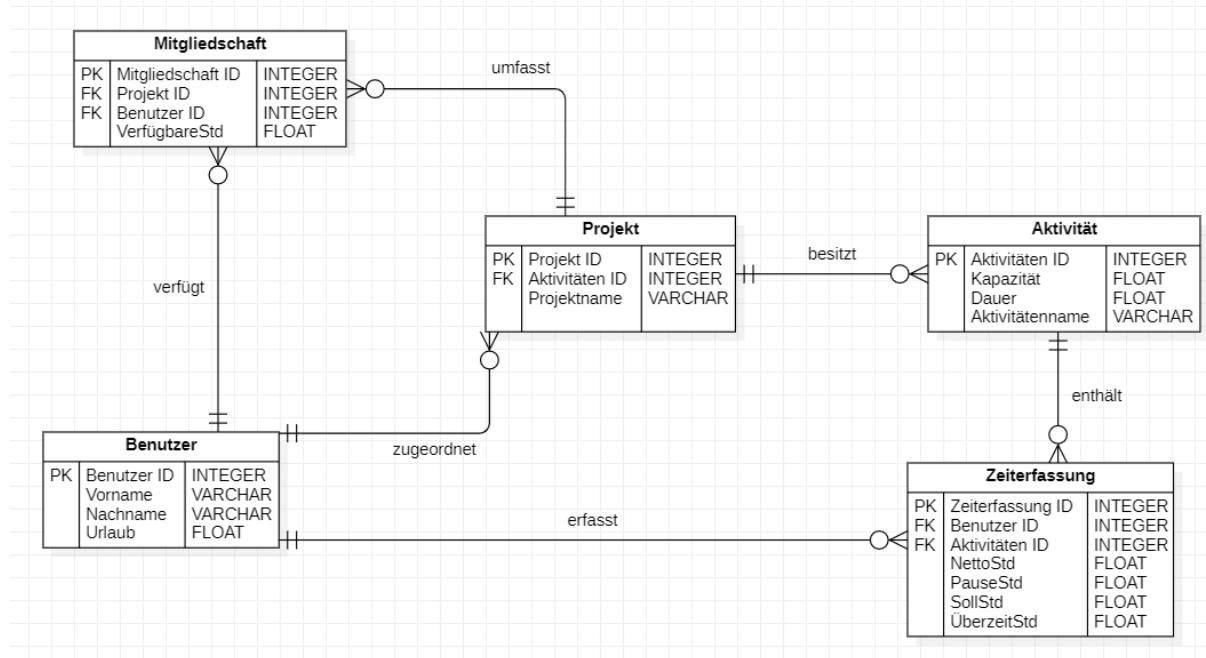


Figure 17: ERM Diagramm V1

## Version 2:

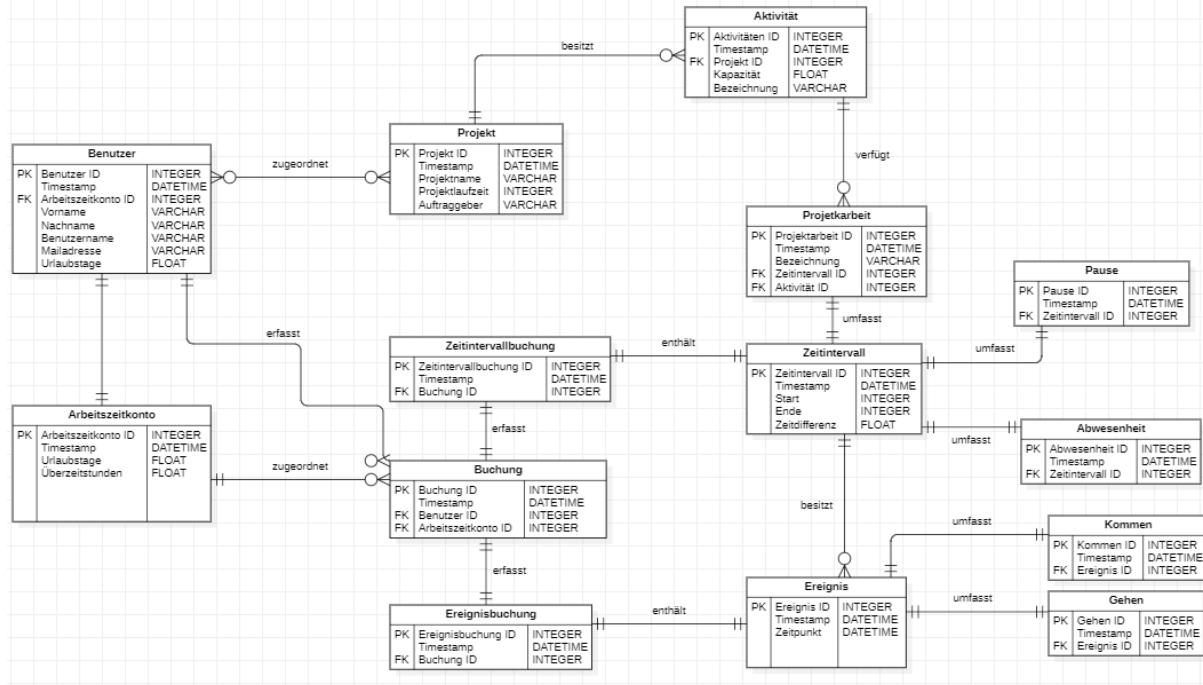


Figure 18: ERM Diagramm V2

### 5.2. Entitäten und Attribute

Im Folgenden werden die jeweiligen Entitäten Benutzer, Arbeitszeitkonto, Projekt, Zeitintervallbuchung, Zeitintervall, Ereignisbuchung, Buchung, Ereignis, Projektarbeit, Pause, Abwesenheit, Kommen, Gehen unseres Er-Modells näher beschrieben und die dazugehörigen Attribute erläutert.

#### 5.2.1. Entität Benutzer

Die Entität Benutzer besitzt die Attribute Benutzer ID, Timestamp, Vorname, Nachname, E-Mail, Urlaubstage und Google\_user\_id. Ein Benutzer kann Projekte erstellen und ebenso als Mitarbeiter in Projekten hinzugefügt werden.

#### 5.2.2. Entität Arbeitszeitkonto

Die Entität Arbeitszeitkonto besitzt die Attribute Benutzer ID, Timestamp, Urlaubskonto, User, Arbeitsleistung und Gleitzeit. Jedem Benutzer ist genau ein Arbeitszeitkonto zugeordnet. Darin wird die aktuelle Arbeitsleistung und Urlaubstage angezeigt.

#### 5.2.3. Entität Projekt

Die Entität Projekt besitzt die Attribute Benutzer ID, Timestamp, Projektname, Laufzeit, Auftraggeber und verfügbare Stunden. Darin werden Informationen zum Projekt gespeichert.

#### 5.2.4. Entität Zeitintervall

Die Entität Zeitintervall besitzt die Attribute Benutzer ID, Timestamp, Start, Ende und Bezeichnung. Das Zeitintervall hält die Zeitlich wichtigen Informationen bzgl. einer durchgeführten Projektarbeit eines Benutzers fest und natürlich auch die Bezeichnung dessen. Ebenfalls fließen in das Zeitintervall die Pausenzeit sowie die Abwesenheit mit ein.

### **5.2.5. Entität Zeitintervallbuchung**

Die Entität Zeitintervallbuchung besitzt die Attribute Benutzer ID, Timestamp, Zeitintervall, Zeitdifferenz und Bezeichnung. Diese besteht aus IST- und SOLL Buchungen von Zeitintervallen. Die Ist Buchung wird vom Projektleiter durchgeführt. Die Sollbuchung wird vom Mitarbeiter, der diese Projektarbeit erhalten hat, ausgeführt.

### **5.2.6. Entität Ereignis**

Die Entität Zeitintervall besitzt die Attribute Benutzer ID, Timestamp, Zeitpunkt und Bezeichnung. Das Ereignis spiegelt die Arbeitszeit des Benutzers wider. Die Zeitlichen Informationen bzgl. des Arbeitsantritts und Arbeitsende fließen ebenso mit in das Ereignis ein. Ein Ereignis kann also ein Arbeitstag sein.

### **5.2.7 Entität Ereignisbuchung**

Die Entität Ereignisbuchung besitzt die Attribute Benutzer ID, Timestamp und Ereignis. Diese besteht aus Ist- und Sollbuchungen von Ereignissen. Die Ist Buchung wird vom Projektleiter durchgeführt. Die Sollbuchung wird vom Mitarbeiter, dem das Ereignis zugeordnet ist, ausgeführt.

### **5.2.8 Entität Buchung**

Die Entität Buchung besitzt die Attribute Benutzer ID, Timestamp, erstellt von, erstellt für, ist Buchung und Bezeichnung. Diese werden von den beiden Klassen Zeitintervallbuchung und Ereignisbuchung geerbt.

### **5.2.9 Entität Projektarbeit**

Die Entität Projektarbeit besitzt die Attribute Benutzer ID, Timestamp, Beschreibung und Aktivität. Die Projektarbeiten sind die "Aufgaben", die ein Benutzer zugeteilt bekommt, die er in einer bestimmten Zeit zu erledigen hat.

### **5.3.0 Entität Pause**

Die Entität Pause besitzt die Attribute Benutzer ID, Timestamp, Start, Ende und Bezeichnung. Da jeder Benutzer eine Pause während dem Arbeiten einlegen muss, wird die Pausenzeit mit in das o.g. Zeitintervall mit einbezogen.

### **5.3.1 Entität Abwesenheit**

Die Entität Abwesenheit besitzt die Attribute Benutzer ID, Timestamp, Abwesenheitsart und Bezeichnung. Da jeder Benutzer auch einen Urlaubsanspruch hat, Krankheitsbedingt oder aus anderen Gründen nicht arbeiten kann, werden diesen Informationen in der Entität Abwesenheit gespeichert. Alle Informationen fließen auch in das Zeitintervall mit ein.

### **5.3.2 Entität Kommen**

Die Entität Kommen besitzt die Attribute Benutzer ID, Timestamp, Zeitpunkt und Bezeichnung. Wird ein Arbeitstag von einem Benutzer angetreten, so wird die Zeit am Anfang des Tages sowie die Bezeichnung der Aufgabe festgehalten und in der Entität Kommen gespeichert. Diese Informationen fließen in die Entität Ereignis mit ein.

### **5.3.3 Entität Gehen**

Die Entität Gehen besitzt die Attribute Benutzer ID, Timestamp, Zeitpunkt und Bezeichnung. Wird ein Arbeitstag von einem Benutzer nach Antritt beendet, so wird die Zeit am Ende des Tages sowie die Bezeichnung der Aufgabe festgehalten und in der Entität Kommen gespeichert.

### **5.3.4 Entität Aktivitäten**

Die Entität Aktivitäten besitzt die Attribute Benutzer ID, Timestamp, Bezeichnung, Dauer, Kapazität und Projekt. In einem Projekt kann der Projektleiter Aktivitäten erstellen und in diesen ebenfalls unterschiedliche Projektarbeiten (Aufgaben) erstellen.

### **5.3.5 Entität Membership**

Die Entität Membership besitzt die Attribute Benutzer ID, Timestamp, Benutzer, Projekt und Projektleiter. Im Membership wird lediglich nur festgelegt, wer der Projektleiter eines Projektes ist. Derjenige der ein Projekt erstellt wird automatisch Projektleiter. Alle anderen Personen im Projekt sind dann Mitarbeiter.

### **5.3.4 Datenbank**

Zur Ablage der Daten haben wir MySQL-Workbench verwendet. Darin wurden mithilfe von SQL-Befehlen Abfragen erstellt, wodurch unsere Entitäten in die Tabellen übernommen wurden. Hierbei lassen sich die einzelnen Eigenschaften der Entitäten anzeigen. Darüber hinaus lassen sich in der MySQL-Workbench Daten anzeigen, welche vom Nutzer in unser Programm geladen wurden. Es werden Informationen zur Einrichtung und zum Anlegen der Datenbank in Kapitel 6.4 „Anlegen einer Datenbank mit Google Cloud SQL“ gegeben.

### **5.3.4 UML-Klassendiagramm**

Auf Basis der Klassen-Struktur wurden die UML-Klassendiagramme erstellt. Da das UML-Klassendiagramm mit seiner Strukturierung von Klassen und deren Beziehungen stets zur Übersicht diente, war es ein essentieller Bestandteil unserer Projektarbeit. Klassen wurden im Laufe des Projekts ergänzt, da diese zu Beginn des Projekts nicht bekannt waren. Dadurch ergaben sich über unser Projekt hinweg zahlreiche Veränderungen im Klassendiagramm.

## 5.5. Überblick Klassendiagramme

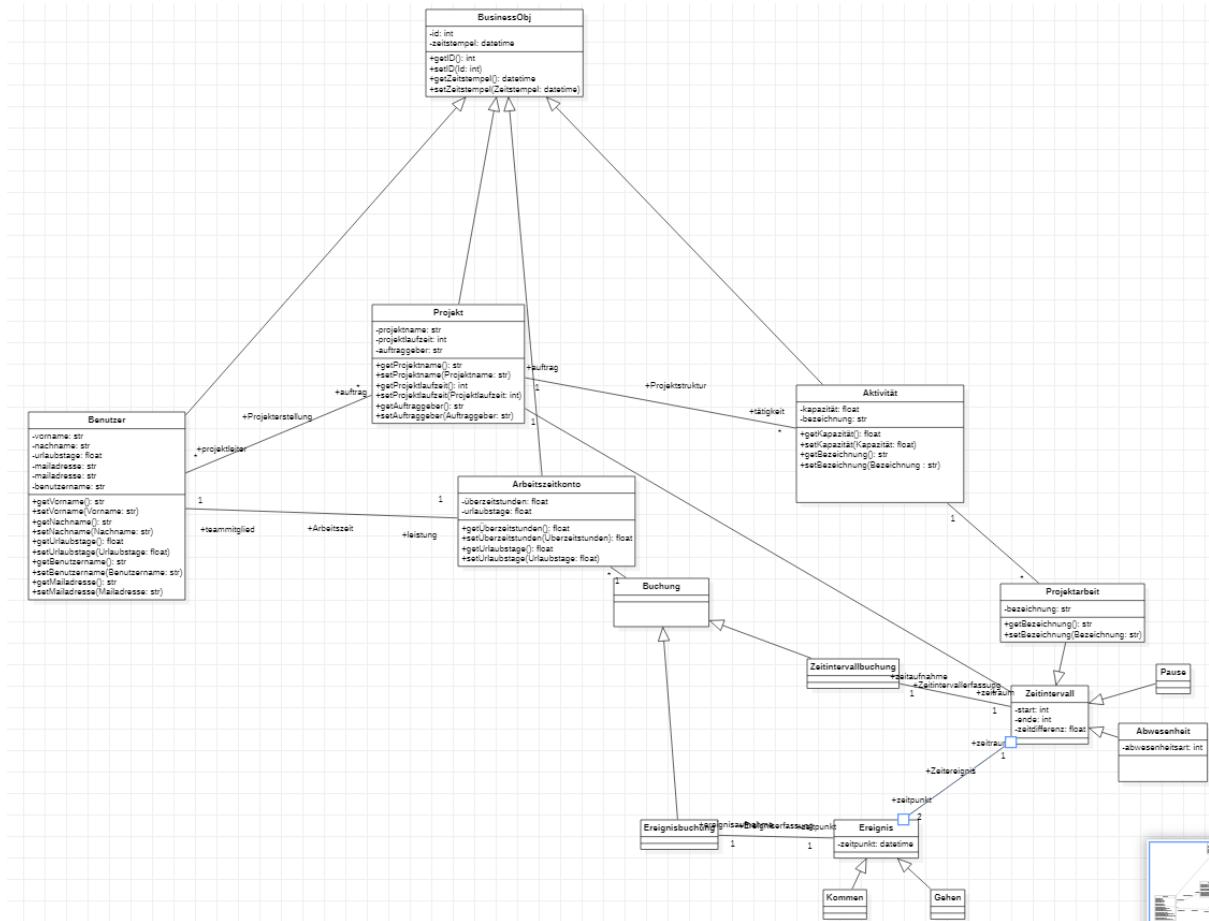


Figure 19: UML Klassendiagramm Übersicht

## Klassendiagramme mit Beziehungen im Detail:

### Zeitkonto:

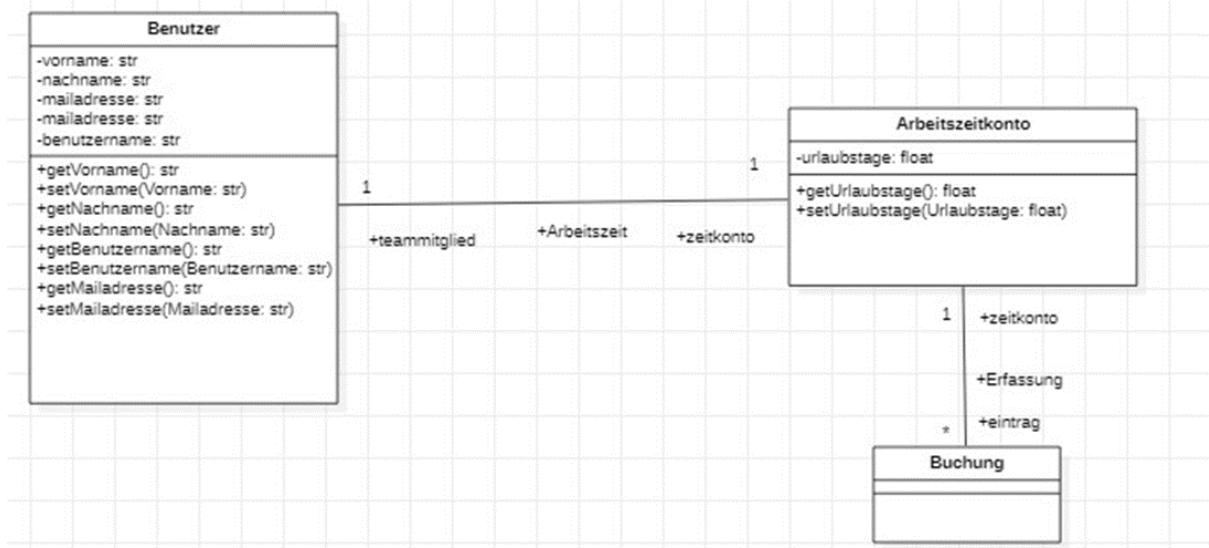


Figure 20: Klassendiagramm Zeitkonto

## Projektbeteiligung:



Figure 21: Klassendiagramm Projektbeteiligung

## Projektstruktur:

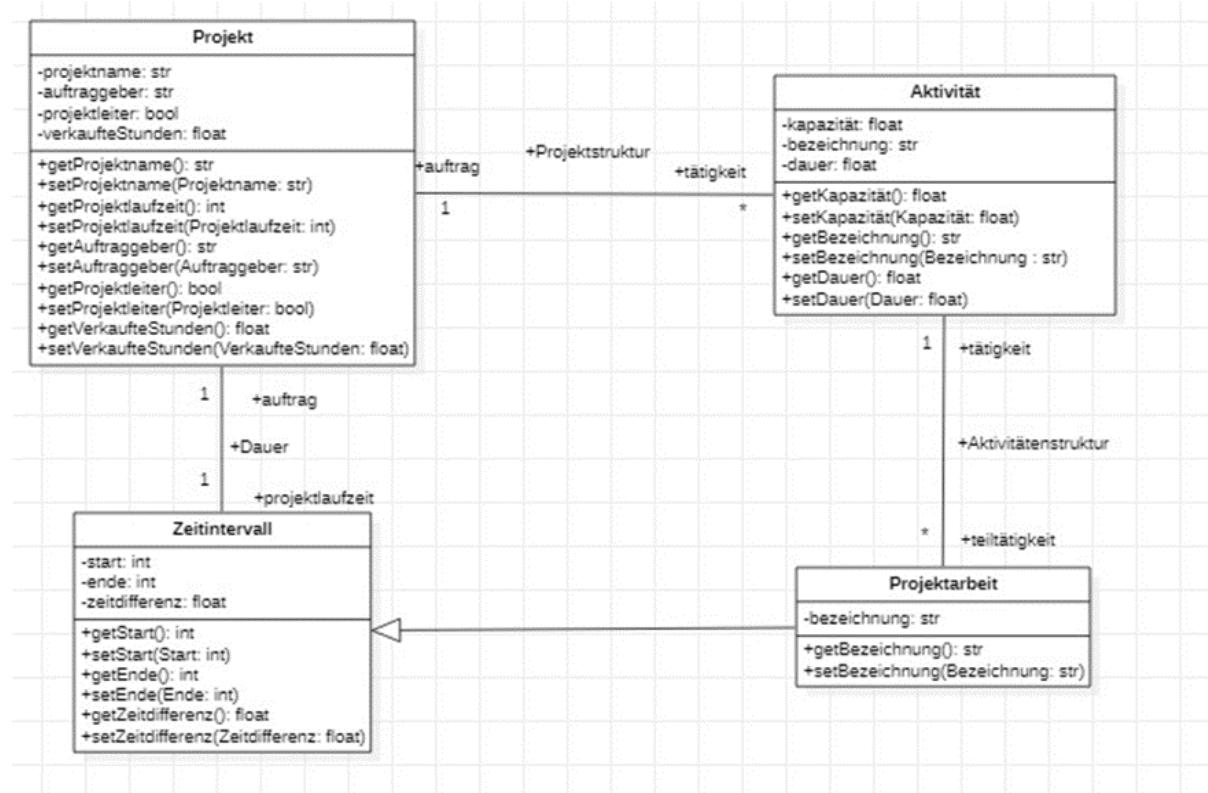


Figure 22: Klassendiagramm Projektstruktur

## Zeitereignis:

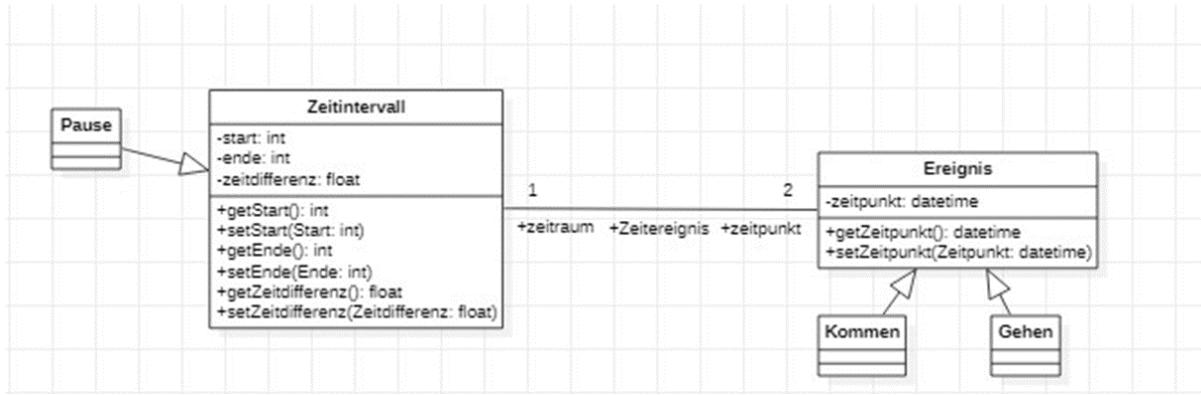


Figure 23: Klassendiagramm Zeitereignis

## Buchungen:

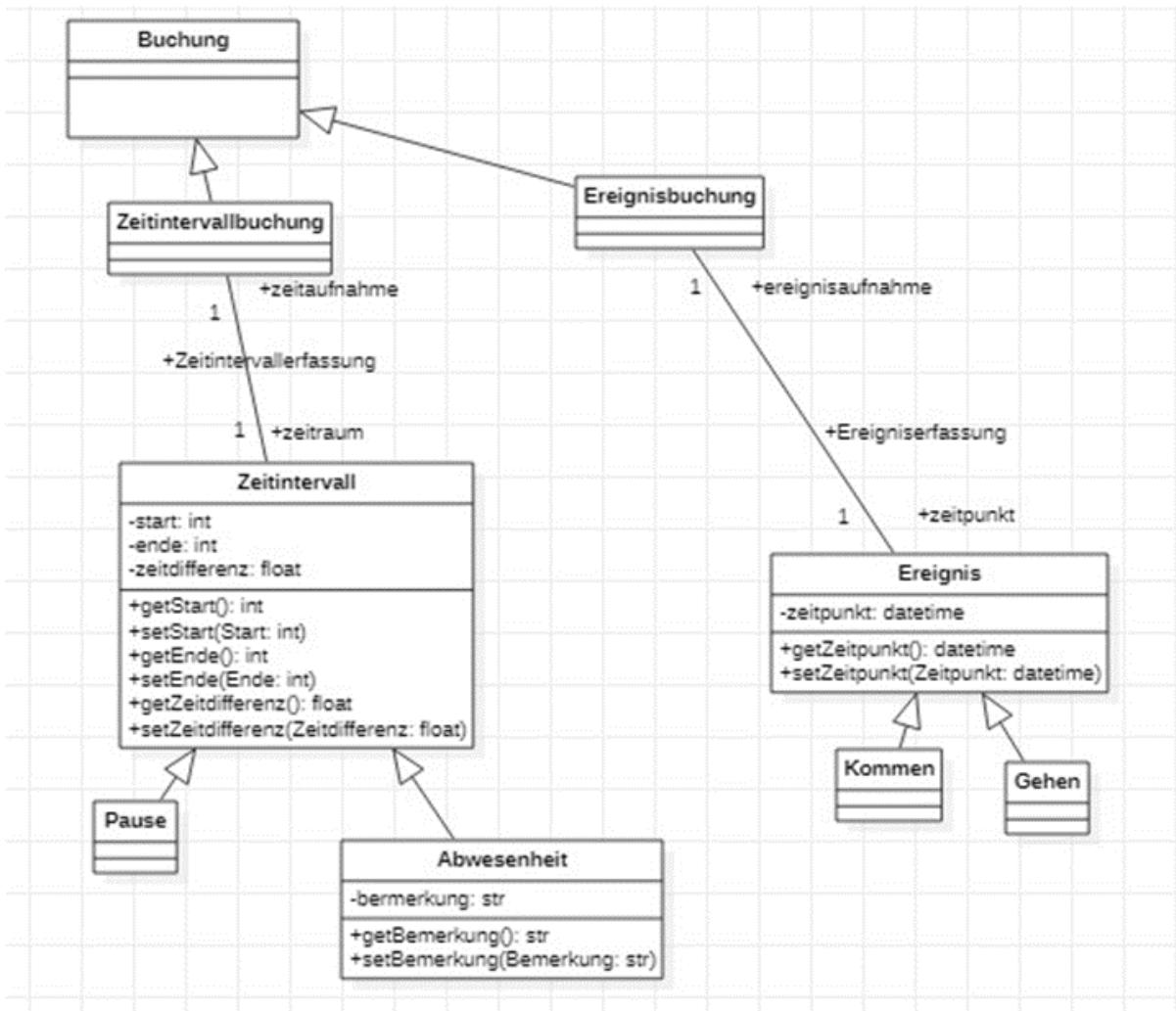


Figure 24: Klassendiagramm Buchungen

## Mapperklassen:

Hier wurden alle Mapper aufgeführt, welche für das Projekt benötigt werden. Die einzelnen Klassen besitzen jegliche Methoden, welche zur Entwicklung der Mapper dienen. Die Aufgabe der Mapperklassen besteht in der Abbildung der Objekte in der Datenbank. Sie ermöglichen mit ihren Methoden einen Zugriff auf die Datensätze der MySQL-Datenbank. Die Methoden dienen der Ausführung von CRUD-Befehlen (Create, Read, Update, Delete). Es ist ebenfalls zu erkennen, dass alle Klassen von der Superklasse „Mapper“ erben.

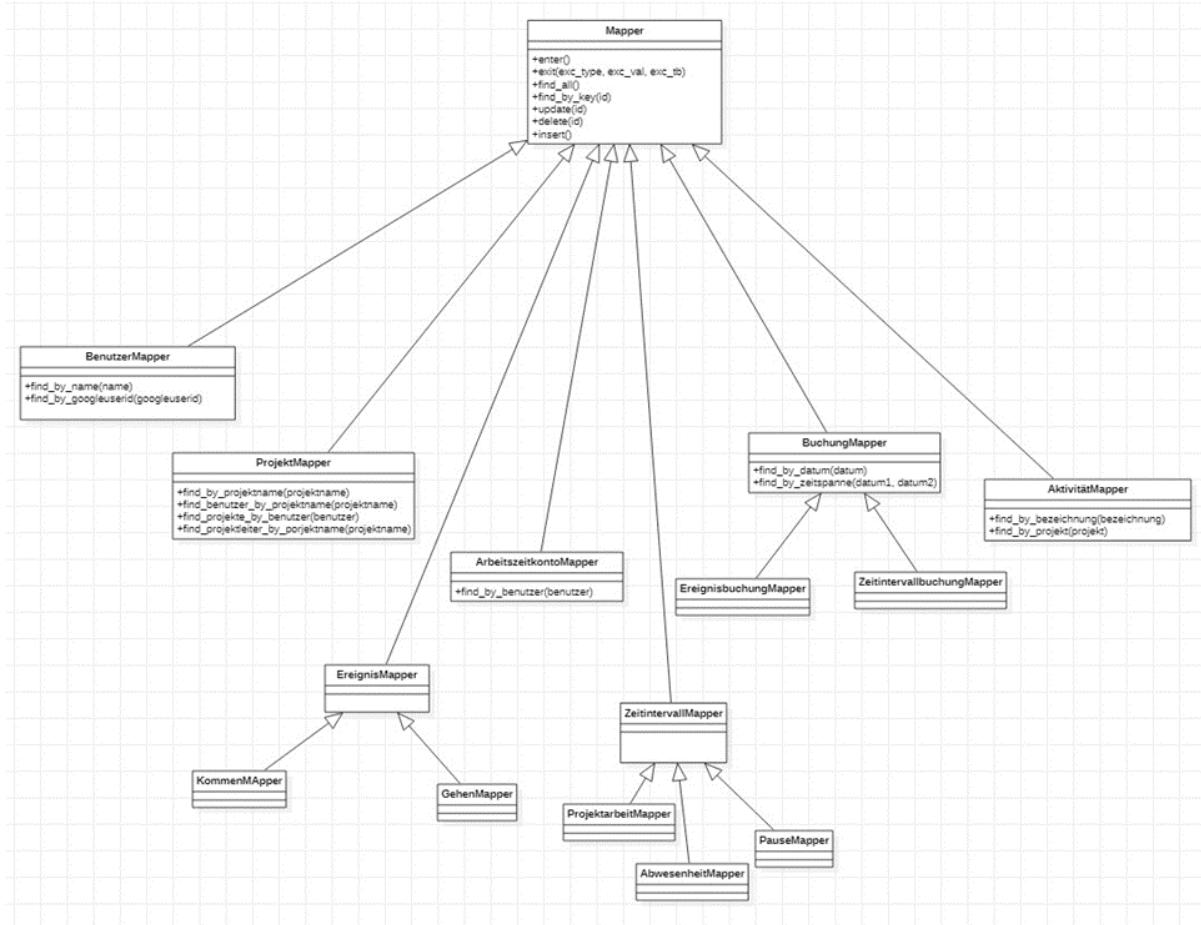


Figure 25: Mapperklassen

## **6 Installationsanleitung**

### **6.1. Vorausgesetzte Programme**

Für die Entwicklung des Zeiterfassungssystems werden folgende Programme benötigt:

- Pycharm oder VS Code
- Anaconda Environment
- Google App Engine (Version 1.9.25)
- GitHub Account
- Google Cloud SQL
- MySQL Connector, MySQLWorkbench

Diese Programme können kostenlos erworben und installiert werden. Benötigte Plugins werden in Pycharm/VS Code direkt installiert. Die Versionen sollten bei allen Mitgliedern dieselben sein, um Konflikte zu vermeiden. Es wird im Backend mit Python und dem Webframework Flask programmiert. Das Frontend wird mit JavaScript und der Softwarebibliothek React entwickelt. Die Datenbank wird mit MySQL und dem Tool MySQL Workbench angelegt - Die Verbindung dazu stellt der MySQL Connector von Python her.

### **6.2. Github Repository**

Es wird ein Account mit einem Repository erstellt, welches für die Entwicklung der App genutzt wird. Zu diesem Repository werden alle Gruppenmitglieder eingefügt mit den entsprechenden Rechten um Code durch einen Commit auf das Repo hochzuladen, Commits zu beurteilen, freizugeben, Branches zu erstellen und löschen und weiteres.

Jedes Mitglied benötigt einen Github Account um zum Repository hinzugefügt werden zu können. Das Repository muss lokal mit Pycharm/VS Code verbunden werden um es anzuzeigen, bearbeiten und Änderungen veröffentlichen zu können. Branches werden erstellt, um unabhängig von den anderen Teammitgliedern an dem Projekt arbeiten zu können. Eine Art der Nutzung von Branches, wäre das Erstellen von Feature Branches, welche einem Mitglied zugeteilt werden. Jedes Mitglied kann eine Branch erstellen und diese veröffentlichen, damit jedes Mitglied darauf zugreifen kann. Es sollten nicht mehrere an einer Datei in einem Branch arbeiten, da sonst Merge-Konflikte entstehen können.

### **6.3. Applikation Online zur Verfügung stellen**

Um die Applikation online verwenden zu können, wird für die Google App Engine und Google Cloud Storage ein Google Konto benötigt. Dieses muss vorher eingerichtet werden. Das Deployen der lokalen App auf den Google Server wurde nach der Anleitung für das Deployment im Moodle Kurs Software-Praktikum durchgeführt.

### **6.4. Anlegen einer Datenbank mit Google Cloud**

Um Daten online abzuspeichern wird die Google Cloud SQL-Datenbank verwendet. Voraussetzung für das Anlegen einer Datenbank in Google ist ein Google Account. Ein Projekt wird vom User angelegt und eine Projekt-ID erstellt. Dazu benötigt Google die Bankdaten des Users, da nach einer 90-tägigen Probephase die Nutzungsgebühr anfällt. Anschließend wird unter dem Menüpunkt „Storage“ ein Bucket erstellt. In diesen Bucket kommt die SQL-Datei MySQL.sql hinein. Die Instanz wird anschließend erstellt. Die lokale Datenbank wird auf die Google Cloud als MySQL Datenbank hochgeladen. Diese wird anschließend durch SQL-Befehle auf Vollständigkeit und Funktionalität getestet. In der mysql\_connecter.py Datei

muss die Google-URL eingetragen werden mit der Applikations-ID und Versionsnummer eingetragen werden, damit es deployed werden kann.

## 7. Benutzerhandbuch

### 7.1. Account erstellen und löschen

Um sich auf unserer Webapp anzumelden, wird ein Google Account vorausgesetzt. Die Anmeldung erfolgt per Google Firebase und wird damit authentifiziert.

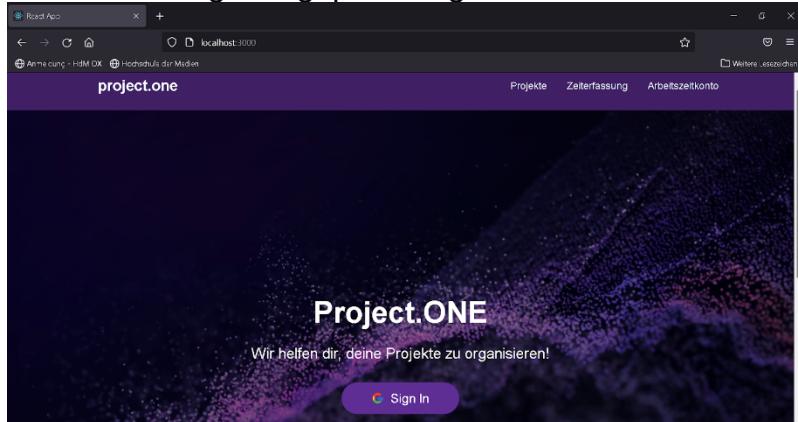


Figure 26: Account erstellen und löschen

Beim erstmaligen Login wird ein Project.One Account mit dem Benutzernamen des Google User Accounts erstellt.

A screenshot of a Google sign-in page. At the top left is a "Über Google anmelden" button with a "G" icon. The main heading is "Anmeldung". Below it is a link "Weiter zu projectone-93c6e.firebaseioapp.com". There is a text input field labeled "E-Mail oder Telefonnummer". Below the input field is a link "E-Mail-Adresse vergessen?". A note in small text states: "Wenn Sie fortfahren möchten, müssen Sie zustimmen, dass Google Ihren Namen, Ihre E-Mail-Adresse, Ihre Spracheinstellung und Ihr Profilbild an projectone-93c6e.firebaseioapp.com weiter gibt." At the bottom left is a "Konto erstellen" button, and at the bottom right is a blue "Weiter" button.

Figure 27: Anmeldung Google Account

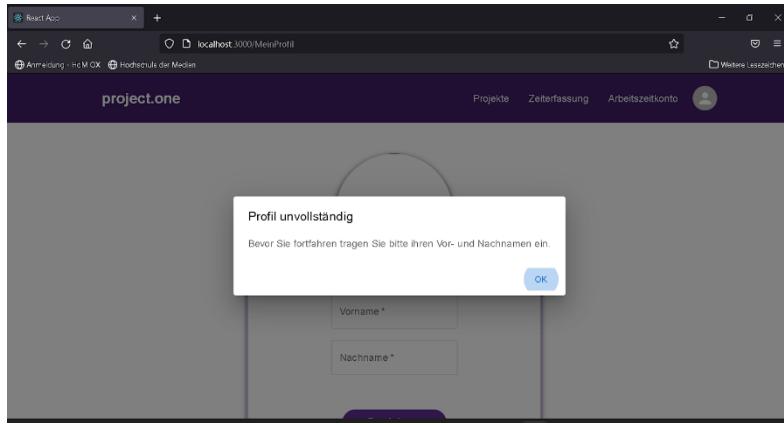


Figure 28: Erfolgreiche Anmeldung mit Google Account

Es erfolgt ein automatischer Login mit dem letzten angemeldeten Google User auf der Webapp. Um einen neuen Project.One Account zu erstellen bzw. den Account zu wechseln, loggt man sich oben rechts über den Logout Button aus und kann sich anschließend erneut mit einem anderen Google Account anmelden. Der Account kann über das “Account Löschen” Icon oben rechts dauerhaft gelöscht werden.

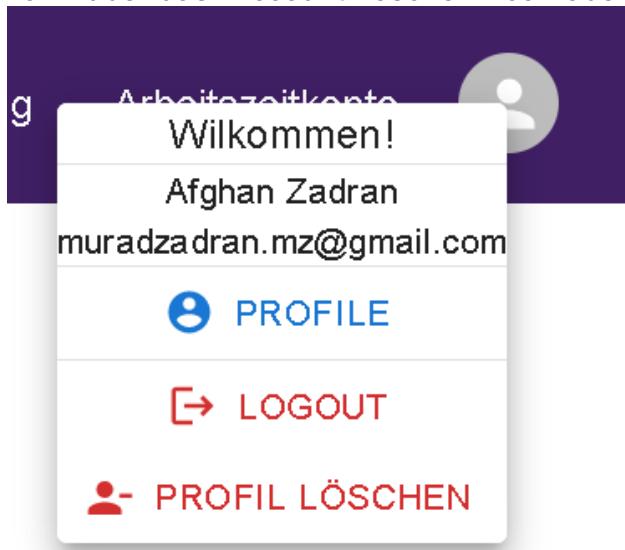


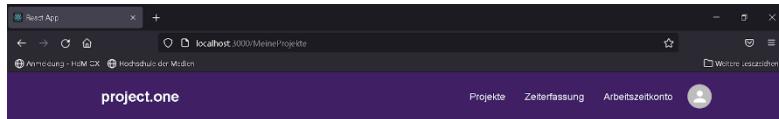
Figure 29: Account Funktionen

## 7.2. Erstellen und Bearbeiten des Profils

Nach erfolgreicher erstmaliger Anmeldung wird der Nutzer dazu aufgerufen seinen Vor- und Nachnamen einzutragen. Falls der eigene Name geändert werden soll, kann dies durch Klick auf den Account Button oben rechts getan werden, indem die Profilansicht mit Vornamen und Nachnamen wieder angezeigt wird. Dort kann der Vor- und Nachname geändert und durch Klick auf “Speichern” übernommen werden.

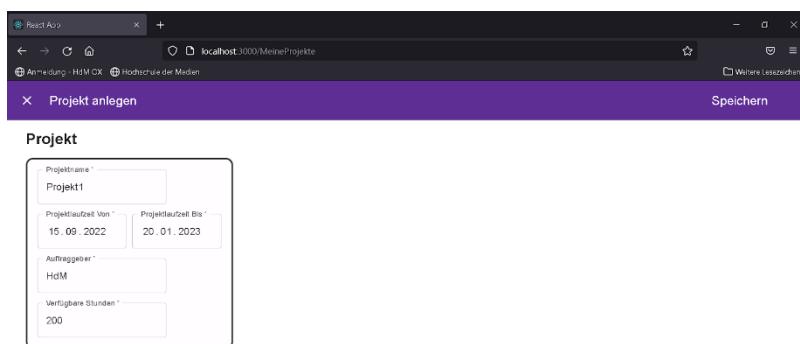
## 7.3. Erstellen, Bearbeiten und Löschen von Projekten und den verknüpften Inhalten

Ein Projekt kann von jedem erstellt werden, sobald ein User ein Projekt erstellt, übernimmt dieser die Rolle des Projektleiters für dieses Projekt. Ein Projekt kann über die Registerkarte “Projekte” durch Klicken auf der Karte mit dem Pluszeichen erstellt werden.



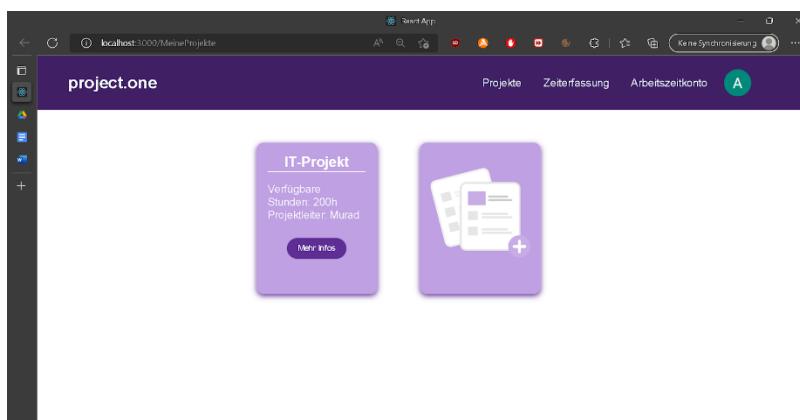
*Figure 30: Projekt erstellen*

Dabei werden die Felder “Projektname”, “Projektaufzeit (Beginn und Ende)”, “Auftraggeber” und die “verfügaren Stunden” ausgefüllt und abgespeichert.



*Figure 31: Projekt anlegen*

Um ein Projekt zu bearbeiten klickt man auf das Projekt, welches als Karte bei der Registerkarte “Projekte” angezeigt wird.



*Figure 32: Projekt bearbeiten*

Auf der detaillierten Übersicht des Projekts, können anschließend Mitarbeiter, Aktivitäten und Projektarbeiten angezeigt, erstellt, bearbeitet und gelöscht werden. Das ganze Projekt kann ebenfalls gelöscht werden indem man oben rechts auf in der AppBar auf das Löschen Icon klickt.

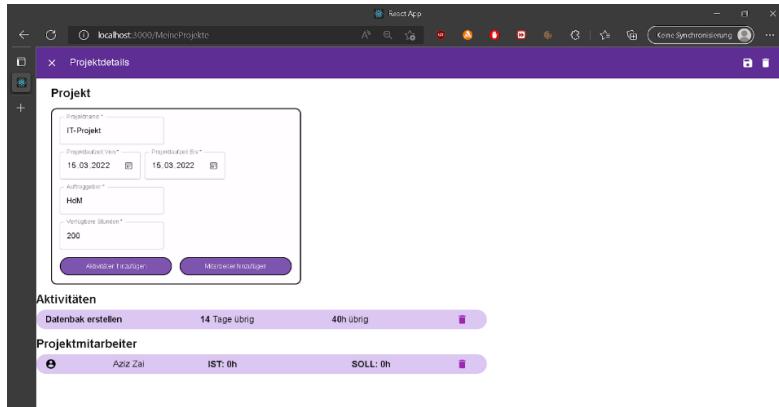


Figure 33: Mögliche Funktionen innerhalb des Projekts

### 7.3.1 Mitarbeiter einem Projekt zuweisen und entfernen

Bei der Projekt Detailübersicht können vom Projektleiter Mitglieder dem Projekt durch den Button “Mitglieder hinzufügen” zugewiesen werden. Die Mitglieder werden anschließend unten als Liste angezeigt und können dort ebenfalls durch Klicken auf den Delete Button aus dem Projekt entfernt werden.

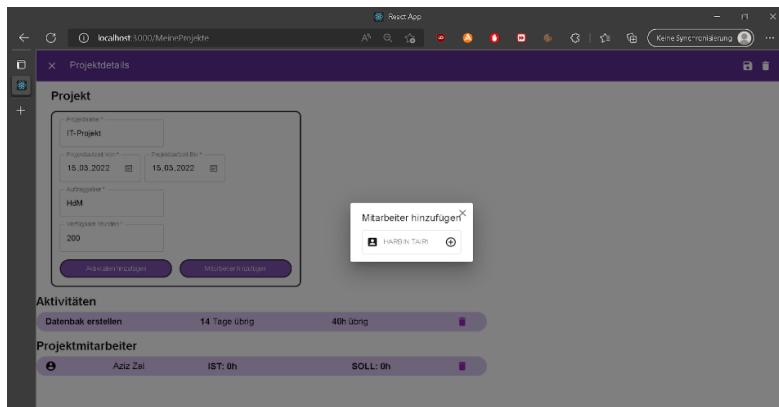


Figure 34: Mitarbeiter zum Projekt hinzufügen

### 7.3.2 Aktivitäten erstellen, bearbeiten und löschen

Eine Aktivität kann ebenfalls in der Projekt Detailübersicht vom Projektleiter durch Klicken des Buttons “Aktivität hinzufügen” erstellt werden. Bei der Erstellung einer Aktivität sind folgende Felder auszufüllen: “Bezeichnung”, “Dauer (Anzahl Tage)” und “Kapazität” (verfügbare Stunden für diese Aktivität innerhalb des Projekts).

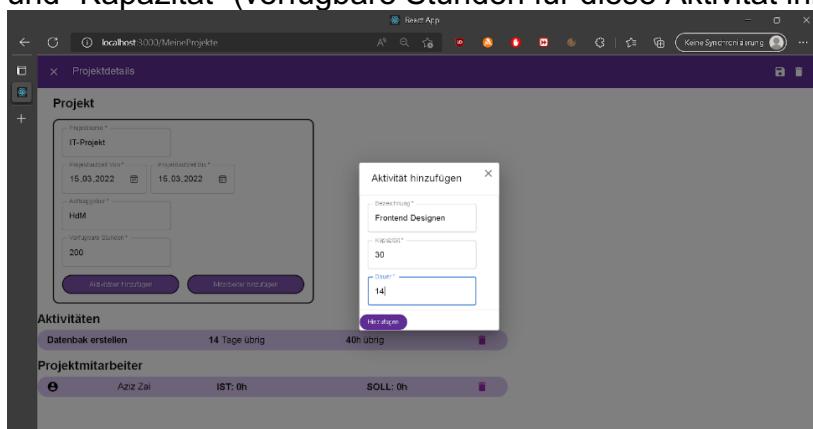


Figure 35: Aktivitäten erstellen, bearbeiten und löschen

Aktivitäten können durch Anklicken dieser in der aufgeführten Liste durch öffnen des PopUps wieder bearbeitet und anschließend abgespeichert werden. Ebenfalls können diese durch das Delete Icon am Ende der Aktivität gelöscht werden.

### 7.3.3 Projektarbeiten erstellen, bearbeiten und löschen

Projektarbeiten sind Aufgaben, die innerhalb einer Aktivität durchzuführen sind. Projektarbeiten können nur vom Projektleiter erstellt werden und Mitgliedern zugewiesen werden. Dabei werden spezielle Sollbuchungen (Projektarbeiten) mit Bezeichnung und der Projektdauer festgelegt und einem Teammitglied zugewiesen.

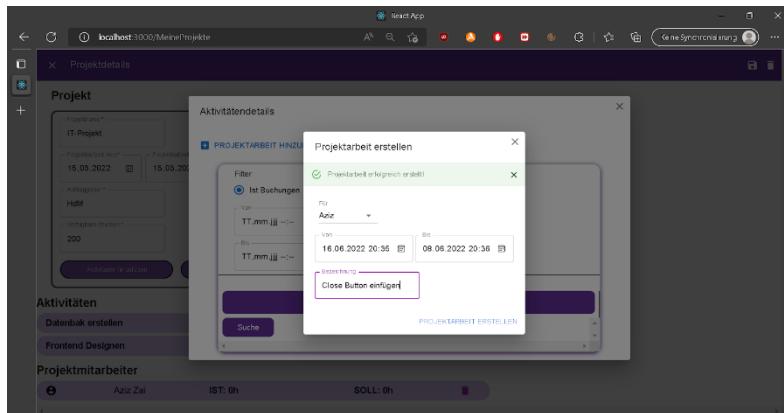


Figure 36: Projektarbeiten erstellen, bearbeiten und löschen

### 7.4 Buchungen durchführen, bearbeiten und löschen

Wenn ein Nutzer eine Buchung für eine Tätigkeit erstellen möchte, geht er dazu auf die Registerkarte "Zeiterfassung". Hier wählt er zuerst das Projekt aus. Als zweiten Schritt wählt er die Aktivität aus. Im dritten Schritt wählt er die Projektarbeit aus. Diese Auswahl ist notwendig, damit die Buchung für das richtige Projekt, Aktivität und Projektarbeit durchgeführt wird. Nachdem alle 3 Schritte ausgeführt worden sind, trägt der Nutzer die Arbeitszeit ein indem er den Beginn und das Ende auswählt und anschließend die Pausenzeit einträgt.

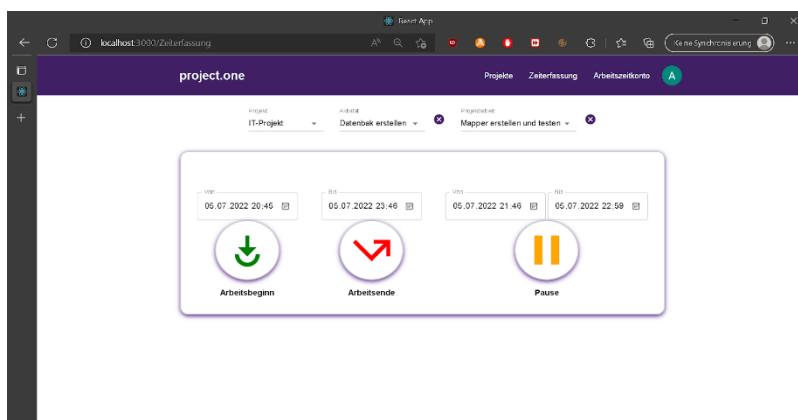


Figure 37: Buchung erfassen

Zuerst muss der Nutzer den Arbeitsbeginn buchen, indem er auf den Button Arbeitsbeginn klickt.

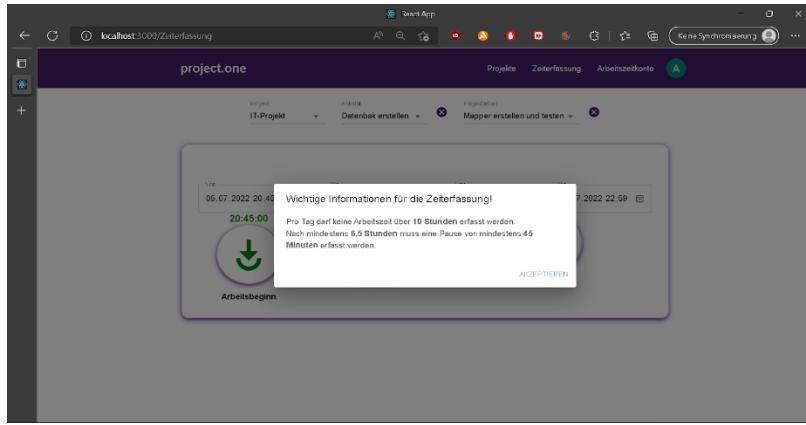


Figure 38: Arbeitsbeginn erfassen

Anschließend bucht er das Arbeitsende ebenfalls durch Anklicken des Arbeitsende Buttons. Dort wird er nach einer Tätigkeitsbeschreibung für den Tag gefragt. Diese muss er einfügen und dann abschicken.

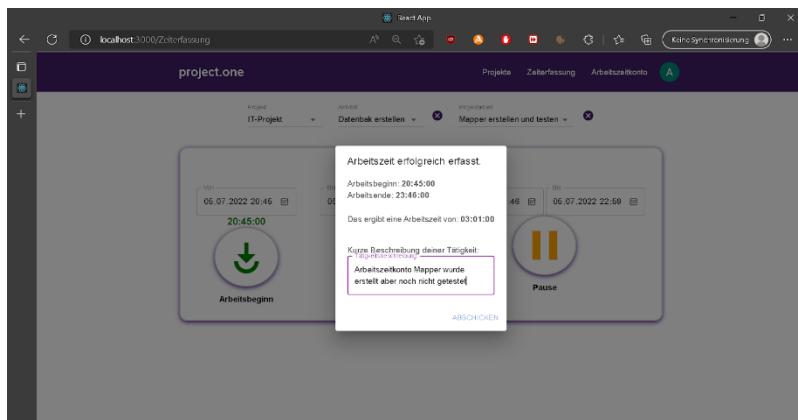


Figure 39: Arbeitsende erfassen

Für die Pausenbuchung muss auch auf den Button Pause geklickt werden, damit die Pause gebucht wird.

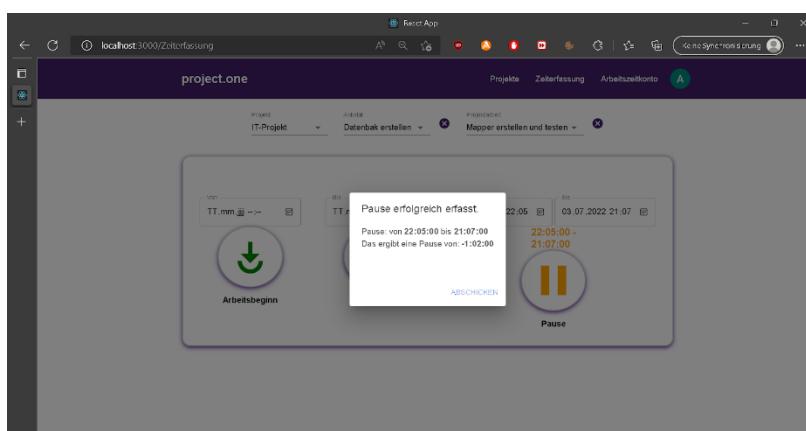


Figure 40: Pause erfassen

Die Buchungen mit den ausgewählten Daten wird erstellt und im Arbeitszeitkonto sowie in der Projekt Detailübersicht für den Projektleiter beim Anklicken der jeweiligen Aktivität sichtbar. Die Buchungen können im Arbeitszeitkonto bearbeitet und gelöscht werden durch die entsprechenden Buttons.

## 7.5 Arbeitszeitkonto Funktionen

Das Arbeitszeitkonto wird durch klicken auf der Registerkarte “Arbeitszeitkonto” angezeigt. Darin werden die Urlaubstage, die aktuelle Arbeitsleistung und die Gleitzeit sowie die Zeitintervallbuchungen und Ereignisbuchungen angezeigt.

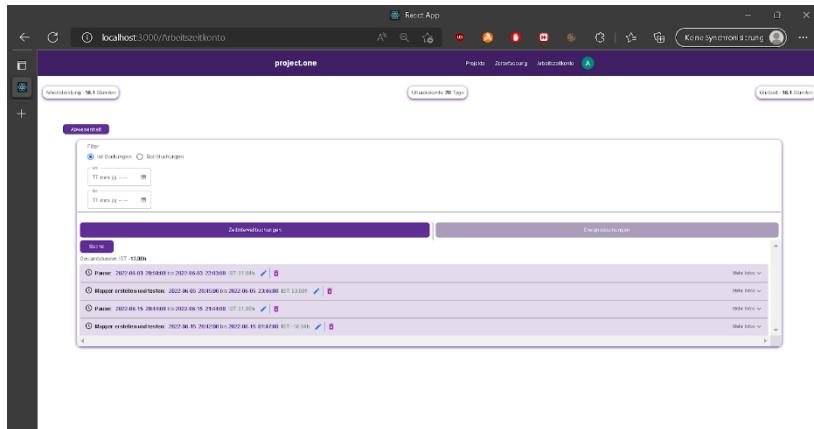


Figure 41: Funktionen des Arbeitszeitkontos

Im Arbeitszeitkonto gibt es einen Filter, der die Buchungen nach der gesuchten Zeitspanne anzeigen. Die Buchungen können hier bearbeitet und gelöscht werden. Bei Abwesenheit z.B. aufgrund von Krankheit oder Urlaub kann eine Abwesenheit durch Klicken des Buttons “Abwesenheit” für einen Zeitraum gebucht werden.

## **Literaturverzeichnis**

Oestereich, B., Scheithauer, A. & Bremer, S. (2013). *Analyse und Design mit der UML 2.5: Objektorientierte Softwareentwicklung* (11., umfassend überarbeitete und aktualisierte Auflage). Oldenbourg Verlag.