|  |  |
| --- | --- |
|  | Praxisarbeit «Aufgabenverwaltung» Webtechnologie 1  Neven Kljajić |

Inhaltsverzeichnis

[1 Einleitung 3](#_Toc127803260)

[2 Planung 3](#_Toc127803261)

[3 Architektur 4](#_Toc127803262)

[4 Use-Case-Diagramm 5](#_Toc127803263)

[5 ERM (Entity-Relationship-Modell) 5](#_Toc127803264)

[6 Testcases und Testplan 7](#_Toc127803265)

[7 Anleitung zur Inbetriebnahme der Webseite 7](#_Toc127803266)

[8 Lessons Learned 7](#_Toc127803267)

[9 Selbstständigkeitserklärung 7](#_Toc127803268)

# Einleitung

Diese Dokumentation dient als umfassender Leitfaden für das Praxisprojekt «Aufgabenverwaltung», eine Webseite, die zur Verwaltung von eigenen Aufgaben entwickelt wurde.

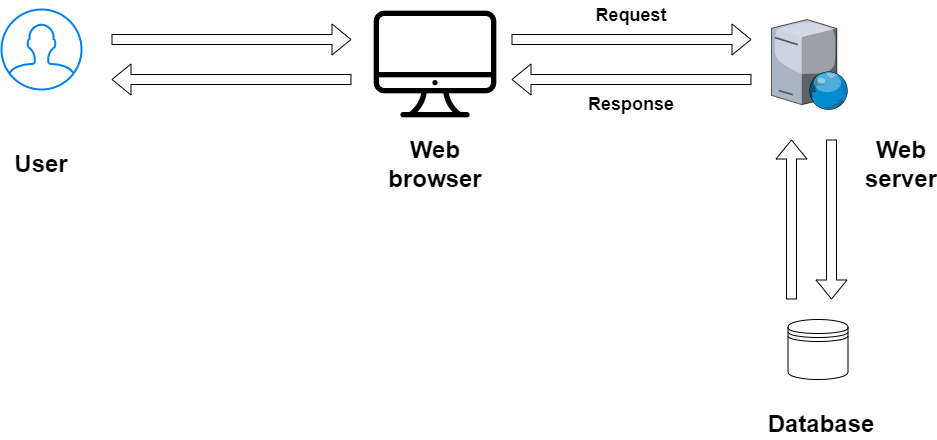
Ziel dieses Projekts war es, ein einfach zu bedienendes Aufgabenverwaltungssystem, um Aufgaben effizienter zu verwalten. Die Anwendung besteht aus zwei Teilen:

* Tasks: Übersicht aller Aufgaben, sortiert nach Fälligkeit. Es kann nach Aufgabentypen gefiltert werden. Zudem können Aufgaben erfasst, editiert und gelöscht werden.
* Admin-Bereich: Verwaltung von Aufgabentypen (erfassen, editieren, löschen) sowie Personen, für welche Aufgaben zugewiesen werden können.

Diese Dokumentation beschreibt die verschiedenen Komponenten des Projekts, einschließlich der Architektur, der verwendeten Technologien, das Datenbankdesigns des Systems, die Planung, Testcases etc. Darüber hinaus bietet sie eine Anleitung zur Inbetriebnahme der Webseite, damit dies der Dozent testen und bewerten kann.

# Planung

# Architektur



Web-Browser:

Der Webbrowser ist die wichtigste Komponente für den Zugriff auf die webbasierte Anwendung. Er stellt die Benutzeroberfläche bereit, die den Inhalt der Anwendung anzeigt und den Benutzern die Interaktion mit dem System ermöglicht. Der Browser ist für das Rendern von HTML und CSS sowie für die Verarbeitung von Benutzereingaben zuständig. Die Benutzeroberfläche der Anwendung wird mit HTML und CSS erstellt, wobei Bootstrap zusätzliche Styling- und Layout-Optionen bietet. Für Bootstrap hat sich der Verfasser deshalb entschieden, weil es eine moderne und einfache Alternative zu einfachem CSS darstellt. Es besteht die Möglichkeit einfach und zeitentsprechende Vorlagen zu entnehmen oder die vorkonfigurierten Klassen zu benutzen.

Die Aufgabe des Browsers besteht darin, HTTP-Anfragen an den Server zu senden und die Antwort für den Benutzer anzuzeigen. Die PHP-Skripte der Anwendung generieren auf der Serverseite dynamisch HTML, das dann als Antwort auf eine Benutzeranfrage an den Browser gesendet wird.

Insgesamt ist der Webbrowser eine Schlüsselkomponente in der Client-Server-Architektur der Anwendung, die es den Benutzern ermöglicht, über eine moderne, reaktionsschnelle und benutzerfreundliche Schnittstelle auf die Funktionen der Anwendung zuzugreifen und mit ihnen zu interagieren.

Webserver:

Der Webserver ist eine entscheidende Komponente der Systemarchitektur, da er als Gateway für den Zugang der Benutzer zu den Systemfunktionen dient. Der Webserver ist für den Empfang von Benutzeranfragen, deren Verarbeitung und die Rückgabe der entsprechenden Antwort verantwortlich. Die in diesem Projekt verwendete Webserver-Software ist Apache, ein weit verbreiteter Open-Source-Webserver. Apache wurde aufgrund seiner Stabilität, Leistung und Sicherheitsfunktionen ausgewählt, jedoch auch weil dies im Unterricht vorgestellt wurde und die Arbeit damit sehr gut funktioniert.

Der Webserver ist so konfiguriert, dass er mit PHP arbeitet, einer serverseitigen Skriptsprache, die zur Erstellung dynamischer Webseiten verwendet wird. Die serverseitigen Skripte sind für die Interaktion mit der Datenbank, die Verarbeitung von Benutzereingaben und die Generierung dynamischer Inhalte verantwortlich, die dem Benutzer über den Webserver zur Verfügung gestellt werden. Zusätzlich zu Apache und PHP verwendet das System MySQL als Datenbankmanagementsystem, um Daten zu speichern und zu verwalten.

Insgesamt ist die Webserver-Komponente ein wichtiger Bestandteil der Systemarchitektur.

Datenbank:

Die Datenbankkomponente bildet die Grundlage des Systems und ist für die Speicherung, Verwaltung und den Zugriff auf die Anwendungsdaten zuständig. Die für dieses Projekt gewählte Datenbanktechnologie ist MySQL, ein open source relationales Datenbankmanagementsystem. Die Wahl fiel auf MySQL, weil uns dies im Unterricht vorgestellt wurde, jedoch punktet MySQL auch wegen seiner Zuverlässigkeit, Skalierbarkeit und Benutzerfreundlichkeit sowie wegen seiner weiten Verbreitung und der großen Nutzergemeinde.

Das Datenbankschema wurde entwickelt, um die für das System erforderlichen Daten zu speichern, darunter Aufgaben, Aufgabentypen und Personen. Jede Tabelle in der Datenbank verfügt über einen Primärschlüssel und geeignete Indizes, um einen effizienten Datenabruf und eine effiziente Datenmanipulation zu gewährleisten.

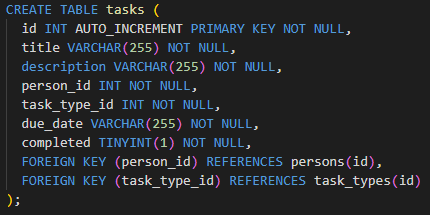
Insgesamt bietet die Wahl von MySQL als Datenbanktechnologie für dieses Projekt eine zuverlässige, skalierbare und sichere Grundlage für das System, und das Datenbankdesign gewährleistet eine effiziente und konsistente Datenspeicherung und -verwaltung.

# Use-Case-Diagramm

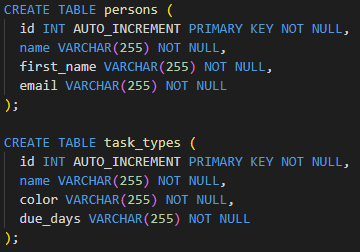
# ERM (Entity-Relationship-Modell)

Die Implementierung des ERM erfolgte mit einer MySQL-Datenbank. Der Verfasser erstellte zunächst die Tabellen mit Hilfe der CREATE TABLE-Anweisungen und definierten dann die Beziehungen zwischen ihnen auf der Grundlage der Projektanforderungen.

Zuerst wurde die Tabelle «tasks» erstellt:

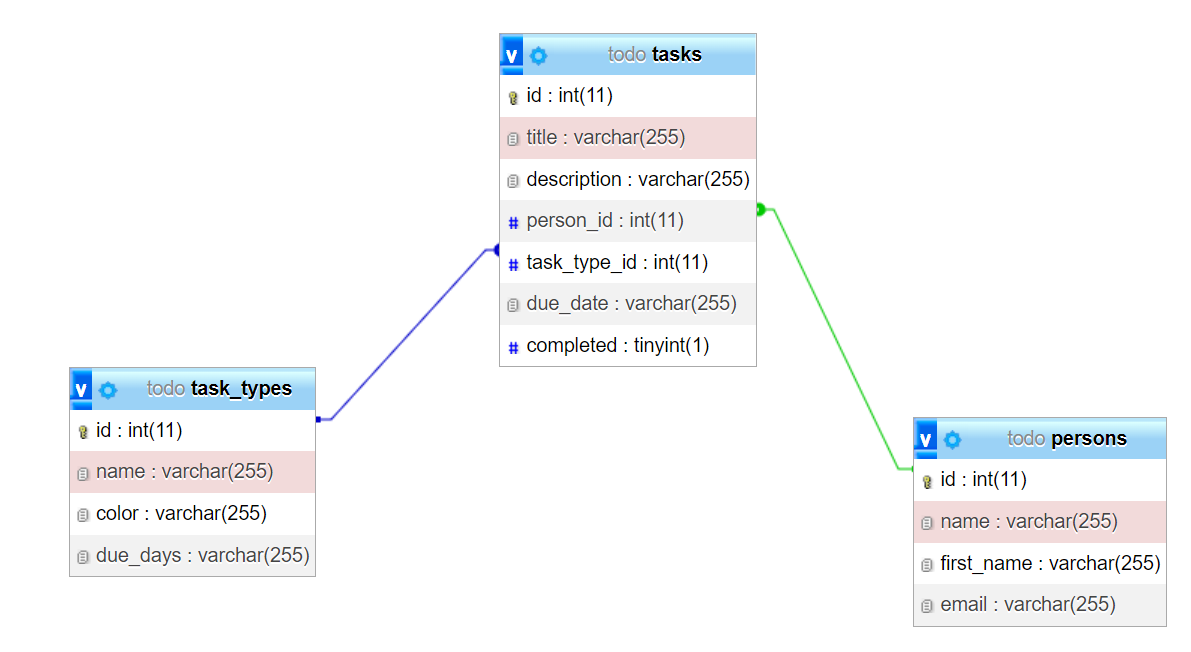


Wie in der Projektanforderungen beschrieben, wurde ein Titel, eine Beschreibung, eine Person, ein Aufgabentyp sowie ein Fälligkeitsdatum angegeben. Zusätzlich wurde auch eine «completed» Spalte erstellt, damit man später einen Task als abgeschlossen markieren kann. Für die Person und den Aufgabentyp wurde jedoch eine separate Tabelle erschaffen:



Als Primärschlüssel wurde für alle 3 Tabellen, jeweils die ID gewählt. Mit den Fremdschlüsseln «person\_id» und «task\_type\_id» wurde in der Tabelle «tasks» die Beziehung sichergestellt.

Die Tabellen wurden so angelegt, dass sie die für unsere Webanwendung erforderlichen Informationen, wie Benutzerdaten und Aufgaben, speichern. Daraufhin konnte ganz einfach das ERM-Diagramm erstellt werden, dass die Beziehungen zwischen den Tabellen grafisch darstellen:



# Testcases und Testplan

# Anleitung zur Inbetriebnahme der Webseite

# Lessons Learned

Die Aufgabe gestaltete sich für die Autoren als anspruchsvoller als anfänglich gedacht. Die Aufgabenstellung haben die Verfasser auch anfänglich nicht richtig verstanden. Die beiden Varianten, welche erstellt werden sollten, wurden fehlinterpretiert und die Verfasser dachten, es bräuchte ein Konzept, wie die Anbindung von den Tankstellen zu der Datenbank stattfinden soll und haben nur ein Design erstellt. Dies kostete die Verfasser einige Stunden, da die Arbeit mehrfach gemacht werden musste.

Zusätzlich kam noch dazu, dass die Aufgabe zuerst mit einem dritten Klassenkameraden gemacht werden sollte. Jedoch stellte sich heraus, dass dieser keine grosse Ambitionen bezüglich des Abschlusses der Gruppenarbeit hatte und somit am Ende der ganze Plan sowie die Arbeiten, welche dieser Mitschüler hätte erledigen sollen, am Ende in Eile von den beiden Verfassern erledigt werden musste.

Nichtsdestotrotz konnten die Verfasser, besonders aufgrund der nach hinten verschobener Deadline, die Arbeit rechtzeitig abgeben, jedoch mit einem mulmigen Gefühl, da aufgrund des Zeitdrucks am Ende, nicht alle Arbeiten, nach dem sich selbst auferlegtem Standard, erfüllt werden könnten.

Weiter war es für die Verfasser schwierig, z.B. die Systemgrenzen für eine Datenbank zu definieren, da sie es noch nie gemacht haben. Auch war es für die Verfasser lange nicht klar, wie das Use-Case-Diagramm zu erstellen ist. Jedoch ergab sich hier eine sinnvolle Einsatzmöglichkeit, nämlich um die einzelnen Benutzer für die Benutzerverwaltung zu erkennen und diese darzustellen.

Als das RM-Diagramm dann bereit war, konnten die Verfasser dann mit der Implementierung beginnen. Diese stellte sich nicht allzu schwierig da, da das RM-Diagramm gut erarbeitet wurde. Jedoch passierten häufig Flüchtigkeitsfehler, die dann zu ständigen Korrekturen führten. Nichts destotrotz funktionierte alles gut und auch die Befüllung der Tabellen mit mind. 3 Initialdaten gelang humorvoll gut. Das Benutzerberechtigungskonzept erwies sich zuerst komplizierter als gedacht. Nach einigen Recherchen der Unterrichtsmaterialien, machte es sehr schnell Klick und die Implementierung gelang problemlos.

# Selbstständigkeitserklärung

Hiermit bestätige ich, Neven Kljajić, die Arbeit selbstständig und ohne Hilfe von anderen Personen durchgeführt zu haben. Des Weiteren bestätigt sie hiermit, dass er keine weiteren Quellen, ausser den angegebenen, verwendet zu haben.

Neven Kljajic   
Zürich, 14. September 2022