

Abschlussprüfung 2022

Fachinformatiker für Anwendungsentwicklung

Dokumentation zur betrieblichen Projektarbeit

**Online- Fehlzeitenerfassen**

Durchführungszeitraum: 05.10.2022 - 09.11.2022

**Prüfungsbewerber**

Mohannad Hafez

Hans-Böckler-Allee 56

52074 Aachen

Logo

Description automatically generated

**Ausbildungsbetrieb**

Wilke Technology

Heider-Hof-Weg 23D

52080 Aachen

**Ausbilder**

Jürgen Wilke

**Projektunterstützer**

Tobias Esser

**Inhalt**

[1 Einleitung 3](#_Toc100836549)

[1.1 Projektbeschreibung 3](#_Toc100836550)

[1.2 Projektziel 3](#_Toc100836551)

[1.3 Projektanforderung 3](#_Toc100836552)

[2 Projektplanung 4](#_Toc100836554)

[2.1 Projektphasen 4](#_Toc100836555)

[2.2 Ressourcenplanung 4](#_Toc100836556)

[2.3 Entwicklungsprozess 4](#_Toc100836557)

[3 Analysephase 5](#_Toc100836558)

[3.1 Ist-Analyse 5](#_Toc100836559)

[3.2 Wirtschaftlichkeitsanalyse 5](#_Toc100836560)

[3.3 Make or Buy-Entscheidung 6](#_Toc100836561)

[3.4 Projektkosten 6](#_Toc100836562)

[4 Entwurfsphase 6](#_Toc100836564)

[4.1 Zielplattform 6](#_Toc100836565)

[5 Implementierungsphase 7](#_Toc100836567)

[5.1 Implementierung der Oberfläche 7](#_Toc100836568)

[5.2 Implementierung der Funktionalität 7](#_Toc100836569)

[6 Dokumentation 8](#_Toc100836570)

[6.1 Benutzerhandbuch 8](#_Toc100836571)

[6.2 Entwicklerdokumentation 14](#_Toc100836572)

[6.2.1 Backend 14](#_Toc100836573)

[6.2.2 Front-End 17](#_Toc100836574)

[6.2.3 Datenbank 17](#_Toc100836575)

[7 Fazit 18](#_Toc100836576)

[7.1 Soll/ Ist-Vergleich 18](#_Toc100836577)

[8 Anhang 19](#_Toc100836578)

[8.1 Abkürzungsverzeichnis 19](#_Toc100836579)

[8.2 Quellenverzeichnis 19](#_Toc100836580)

[8.3 Code-Anhang 19](#_Toc100836581)

# Einleitung

Die folgende Projektdokumentation beinhaltet den Ablauf des IHK-Abschlussprojektes, welches ich zum Ende meiner Ausbildung zum Fachinformatiker der Fachrichtung Anwendungsentwicklung

durchgeführt habe.

## Projektbeschreibung

Entwicklung einer webbasierten Projektzeitverfassung mit zentralen Datenbanken.  
Zur Realisierung sollen die Webtechnologien und Programmiersprachen PHP, MySQL, HTML, CSS, JavaScript zum Einsatz kommen.

# Ausgangssituation (IST-ANALYSE)

# Die Mitarbeiter unseres Unternehmens melden sich über eine passwortgeschützte Webseite als abwesend. Hierbei gibt es nur einen Satz Login-Daten, die allen Mitarbeiter bekannt sind. Es handelt sich um ein simples Formular, bei dem Vorname, Nachname, Grund der Abwesenheit (Krankheit, Arztbesuch, Sonstiges), sowie Start- und Endzeitpunkt angegeben werden. Durch einen Button wird eine E-Mail mit den eingegebenen Daten erzeugt und an eine E-Mail-Adresse geschickt, die nur von berechtigten Personen eingesehen werden kann.

## Projektziel

## Alle Angestellten sollen Ihre Fehlzeiten weiterhin per Web-Browser in ein zentrales System eintragen. Ein Administrator soll hier die Möglichkeit haben neue Benutzer anzulegen, zu ändern oder zu löschen.

Die Fehlzeitendaten sollen in einer Datenbank gespeichert werden.  
Die in der Datenbank gespeicherten Fehlzeiten sollen über eine interne API abrufbar sein.

Diese Schnittstelle dient der automatischen Eintragung in die Stundenzettel mithilfe einer anderen Applikation und Organisation der Vertretung eines abwesenden Mitarbeiters.

Die zeitnahe Löschung nicht mehr benötigter Datenbankeinträge soll ebenfalls über einen API-Aufruf ausgelöst werden. Hier werden automatisch alle Fehlzeiten deren Endzeitpunkt in der Vergangenheit liegen entfernt.

Da die Applikation bei unserem Webhoster in Betrieb genommen wird muss sie mit PHP 8.0 und MariaDB 10 funktionieren.

Zudem soll ein einwandfreier Betrieb mit den aktuellen Versionen von Google Chrome und Mozilla Firefox sichergestellt werden.

## Kundenwünsche

* Login mit mitarbeiterspezifischen Benutzername & Passwort
* Erstellung neuer Benutzer durch berechtigte Benutzer
* Fehlzeiterfassung der Benutzer
* API zum lesen aktueller Fehlzeiten und Löschung veralteter
* Einfache Bedienung

## Abweichen zum Antrag

Nach einer Besprechung mit dem Projektunterstützer und Genehmigung von Herrn Timo Beyel habe ich die Aufgabe: ein API zum Lesen aktueller Fehlzeiten und Löschung veralteter, für spätere Zeit verschoben, dadurch ist diese Aufgabe nicht ein Teil meines Projektes.

Das hat auch die Folge, dass es kein Trigger mehr zur Verfügung steht (Aufruf), deswegen kann ich nicht die alten Anträge, die ihre (bis Datum) überschritten sind, nicht mehr gelöscht werden können.

## Projektumfeld

## Prozessschnittstellen

# Ressourcen und Ablaufplanung

In der Projektplanung soll die notwendige Zeit und die benötigten Ressourcen in eine für die Entwicklung sinnvolle Struktur gebracht werden. Dazu wird der Ablauf des Projektes geplant und erfasst.

## Ressourcenplanung

Die Ressourcenplanung zeigt alle für das Projekt eingesetzten Ressourcen auf.

In meinem Fall sind die:

* Arbeitszeit
* Entwicklungsumgebung
* Programmierfähigkeiten
* Softwarelizenzen

Die im Projekt eingesetzten Softwarebibliotheken unterliegen der GPL V3. Die zur Entwicklung notwendigen Hard- und Softwareprodukte wurden vom Ausbildungsbetrieb zur Verfügung gestellt.

## Projektkosten

Im Folgenden werden die anfallenden Kosten für dieses Projekt kalkuliert. Der Kostensatz eines Auszubildenden liegt bei ca. 11,25 € und der eines Mitarbeiters 20 €. Für die Ressourcennutzung.

In meinem Fall, hat meine Stunde nur 0,625 die Firma gekostet, da ich nur 100€ pro Monat verdiene.

160 Std monatlich:100€ | 100€:160Std = 0,625€ pro Std.

Aus diesen Daten entsteht folgende Kalkulation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Aufgabe | Mitarbeiter | Zeit | Personalkosten | Ressourcenkosten |
| Entwicklung | Umschüler | 79h | 49,375€ | Kostenlos (Open-Source) |

## Ablaufplanung

Für das Projekt standen mir 79 Stunden zur Verfügung, welche von mir auf einzelne Phasen verteilt wurden. Diese Phasen lassen sich grob auf Analyse, Entwurf, Implementierung, Deployment und Dokumentation aufbrechen. Dadurch entstehen fünf Hauptphasen, welche wiederum in kleinere Unterphasen zerlegt werden, wodurch eine detaillierte Ansicht aller Aufgaben entsteht.

|  |  |
| --- | --- |
| Projektphase | Dauer in Stunden |
| Analyse | 4 |
| Entwurf | 13 |
| Implementierung | 40 |
| Deployment & Abnahme | 2 |
| Dokumentation | 20 |
| Summe | 79 Stunden |

# Durchführung und Auftragsbearbeitung

## Prozessschritte

Analyse

* Durchführung einer Ist-Analyse
* Erstellung eines Soll-Konzepts

Entwurf

* Erstellung einer GUI
* Datenbank Struktur erstellen

Implementierung

* Code erstellen
* Code cleanup

Abnahme

* Code-Review mit dem Ausbilder
* Funktionstest intern
* Gegebenenfalls Fehlerkorrekturen

Dokumentation

* Erstellung der Projektdokumentation
* Erstellung des Benutzerhandbuch

## Vorgehensweise

Um eine flexible Umsetzung der Anforderungen zu ermöglichen, habe ich mich für den

SCRUM Entwicklungsprozess entschieden, SCRUM ist für Projekte geeignet, in denen viel Unklarheit herrscht. Häufig sind dies Vorhaben, die nach neuen Lösungen suchen und deren Kombination aus unvorhersehbaren Risiken der Implementierung und noch unklaren Anforderungen bestehen.

## Arbeitsschritte

Bevor ich mit der Arbeit angefangen habe, der Projektunterstützer bat mich, ein neues Repository auf GitHub einzurichten, die ihm die Möglichkeit geben, die Änderungen zu beobachten und einfach die Arbeitsschritte zu beobachten und das Code zu speichern und sichern.

Ich habe ein Framework aus Bootstrap ausgesucht, Bootstrap ist ein Frontend-Framework, mit dessen Hilfe Webentwickler geräteübergreifende Websites verschiedenster Art erstellen können. Zu diesem Zweck bietet das Open-Source-Projekt diverse Gestaltungsvorlagen, die auf HTML und CSS basieren, sowie optionale JavaScript-Erweiterungen. Und dann habe ich das Framework bearbeitet und auf meine Anforderungen umgestellt.

Ich habe zunächst mein DatenBank (fehlzeitdb) hergestellt, die aus 2 Tabellen besteht.

1. tbl\_users: [id(int)PK, vorname(varchar), nachname(varchar), b\_name(varchar), pass(varchar), role(int)].
2. tbl\_grund: [gr\_id(int)PK, u\_id(int)FK, grund(int), note(varchar), von\_datum(datetime), bis\_datum(datetime), reg\_datum(datetime)].

Das Bild zeigt der Primary Schlüssel und der Sekundär Schlüssel mit einer Beziehung von n zu m.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Dann habe ich CodeIgniter heruntergeladen und mein Projekt durch Visual Studio code erstellt, einige Dateien sollen bearbeitet werden:  
application-> config-> config.php

$config['base\_url'] = 'http://localhost/FehlzeitPro/';//der Name meines Projektverzeichnis.

$config['log\_threshold'] = 4;

Aktivieren die Fehlerprotokollierung, indem einen Schwellenwert über Null festlegen. Der Schwellenwert bestimmt, was protokolliert wird. 4 = All Messages.

Diese Fehlermeldungen werden in dem Pfad: application-> config-> gespeichert.

$config['encryption\_key'] = 'UVvM3gosO8yK9JeijFwasEXD8kHs2G1y';

Wenn ich die Encryption-Klasse verwenden, muss ich einen Verschlüsselungsschlüssel festlegen.

application-> config-> autoload.php

$autoload['libraries'] = array('database','session','encryption');

Klassen initialisieren. Dies sind die Klassen, die sich in system/libraries/ oder Ihrem application/libraries/-Verzeichnis befinden, mit dem Zusatz der 'Datenbank'-Bibliothek, was ein Sonderfall ist.

$autoload['model'] = array('dbmodel');

Rufe das Datenbankmodell auf, wodurch es mit meiner Datenbank kommunizieren kann.

application-> config-> database.php

$db['default'] = array(‘database' => 'fehlzeitdb'); Festlegen des Namens meiner Datenbank.

Dann habe ich die Anmeldeansicht erstellt, die ein Textfeld für den Benutzernamen und ein weiteres für das Passwort enthält, das der Benutzer ausfüllen muss, um sich anzumelden.

Als nächstes erstellte ich mein Benutzerformular, das der Benutzer ausfüllen sollte, um seine „Abwesend Zeit“ hinzuzufügen. es enthält einige Überschriften, Textfelder, Drop-down und ein Optionsfeld.

Ich habe JavaScript verwendet, um Notizen und ein Textfeld ein- und auszublenden. Wenn der Arbeitnehmer bei einem Arztbesuch oder krank ist, muss er eine AU mitbringen. und wenn es sonstiges ist, muss er eine kurze Meldung darüberschreiben.

So ich habe eine neue Datei „info\_helper“ innerhalb das Helper Verzeichnis erstellt, und weil dieses Verzeichnis außerhalb der Scope ist, muss ich eine Instanz von der Klasse CI\_Controller erzeugen, dadurch ich die Funktion in dbmodel.php erreichen kann.

   function get\_autor\_name\_from\_DB($id)

    {

        $this->db->select('\*');

        $this->db->from('tbl\_users');

        $this->db->where('id', $id);

        $query = $this->db->get();

        $row\_array = $query->row\_array();

        return $row\_array;

    }

Formular ausfüllen: nach der Ausfüllung des Formulars muss geprüft werden, ob alle Felder ausgefüllt sind, und einen Grund von die drei Optionen ausgesucht ist.

Beim „Mitarbeiter bearbeiten“ und „Mitarbeiter addieren“ muss geprüft werden, ob der neue Benutzername schon in der DB existiert ist, deswegen habe ich eine Funktion in dbmodel.php geschrieben, um das zu überprüfen.

    function check\_is\_Bname\_existiert($mit\_benuzter)

    {

        $this->db->select('\*');

        $this->db->where('b\_name', $mit\_benuzter);

        $this->db->from('tbl\_users');

        $query = $this->db->get();

        $result\_array = $query->result\_array();

        if (!empty($result\_array)) {

            return 1;

        }

        else

        {

            return 0;

        }

    }

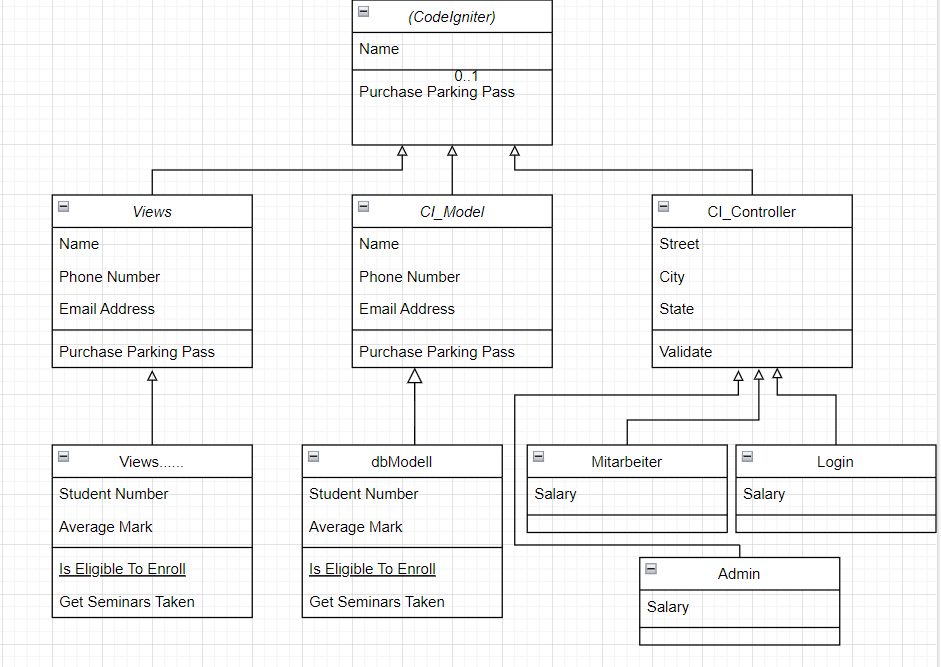
Auch wollten wir wissen, ob denselben Benutzer sein daten bearbeiten will, und nicht jedes Mal neues Passwort einzugeben. deswegen habe ich eine Funktion in dbmodel.php geschrieben, um das zu überprüfen.

function check\_is\_same\_Bname($mit\_id, $mit\_benuzter)

Für die Verschlüsselung habe ich die Verschlüsselungsbibliothek des CodeIgniter benutzt, die Verschlüsselungsbibliothek bietet eine bidirektionale Datenverschlüsselung. Um dies kryptografisch sicher zu tun, nutzt es PHP-Erweiterungen, die leider nicht immer auf allen Systemen verfügbar sind.

$key = $this->encryption->encrypt($mit\_password); // Passwort verschlüsseln

$inc\_password = $this->encryption->decrypt($result\_row\_array['kennwort']); // decrypt the DB password and the users password



## Abweichungen, Anpassungen, Entscheidungen

Das Projekt wurde mithilfe von PHP, JavaScript sowie dem CodeIgniter- und Bootstrap Framework entwickelt.

Für mein Localhost webserver habe ich mich für XAMPP entschieden, XAMPP ist die Kombination des Apache Webservers, der Datenbank MySQL und der Skriptsprachen Perl und PHP.

Mit XAMPP ist es möglich, in wenigen Schritten einen eigenen Webserver mit all seinen Komponenten auf meinem lokalen Rechner zu installieren.  
  
In XAMPP ist der Ordner htdocs, in dem alle Projekte, die auf „localhost“ sich befinden, gespeichert werden müssen. Deswegen habe ich mein Repository in dem Ordner gemacht, und es befindet sich in demselben Ordner, in dem sich mein Projekt befindet.

Für die Implementierung und die Bearbeitung von mein PHP code habe ich Visual Studio code benutzt, Visual Studio Code ist ein kostenloser Quelltext-Editor von Microsoft zur Erstellung von Cross-Plattform-Applikationen mit JavaScript, HTML und CSS entwickelt.

Ich habe mich für CodeIgniter entschieden, CodeIgniter ist eines der besten PHP-Frameworks, das eine Open-Source-Entwicklungsplattform bietet. Es unterstützt PHP-Entwickler, indem es die einfachste Möglichkeit bietet, modulare Methoden zu verwenden und jede explizite Funktion zu erhalten. Aufgrund seiner benutzerfreundlichen URLs ist es einfach, große dynamische und sichere Webanwendungen zu erstellen.

In der Datenbank habe ich einen Standardbenutzer mit dem Benutzernamen (admin) und einem Passwort (admin) für die erste Verwendung der Software erstellt und ich werde in der Bedienungsanleitung deutlich machen, dass dieses Passwort nach der ersten Anmeldung ersetzt werden sollte.

Wenn sich der Benutzer als Administrator anmeldet, kann er alle Benutzer sehen, die bearbeitet oder entfernt werden können, und durch Klicken auf Benutzer hinzufügen kann er einen neuen Benutzer oder einen neuen Administrator hinzufügen. Der Administrator kann den Standardadministrator nicht entfernen.

Nach einer Diskussion mit dem Projektunterstützer haben wir uns entschieden, das Namensfeld zu einer Dropdown-Liste zu machen, in der Sie alle Benutzer sehen können, deren Namen bereits in der DB gespeichert sind, und dass der Benutzer einen anderen Benutzer als krank markieren kann. Und die Möglichkeit, dass ein Mitarbeiter einen anderen Mitarbeiter als anwesend registrieren kann.

Deshalb müssen wir eine weitere Spalte in der Datenbank zur „tbl\_grund“ Tabelle hinzufügen, um den Benutzer zu speichern, der diesen Eintrag hinzugefügt hat. Der Name dieser Spalte ist autor\_id.

Auch dazu, wir brauchen den Benutzername des Autors von der DB, deswegen brauchen wir einen „Helper“ zu benutzen, deren Aufgabe ist: von der AutorID einen Autorname zu machen, durch einen DB Query.

Nach einer Diskussion mit dem Projektunterstützer haben wir uns entschieden, dass wenn einen Benutzer gelöscht werden, werden alle seine Fehlzeiten von der DB auch gelöscht.

***CodeIgniter***

CodeIgniter ist ein seit 2014 erhältliches, in PHP geschriebenes, quelloffenes Webframework.

CodeIgniter ist schlank gehalten, wodurch eine hohe Performance erreicht wird und die Einarbeitungszeit im Vergleich zu anderen Frameworks kurz ist. Zentraler Bestandteil ist die Model View-Controller-Architektur (MVC). CodeIgniter enthält eine große Zahl an Bibliotheksklassen und Hilfsfunktionen, die mir Standardaufgaben abnehmen.

***Bootstrap***

Bootstrap ist ein kostenloses und quelloffenes Frontend-Web-Framework zur Erstellung von Webseiten. Es dient nur der Front-End-Entwicklung und enthält HTML- und CSS-basierte Designvorlagen für Typografie, Formulare, Schaltflächen und Navigation.

# Projektergebnisse

## Soll-Ist-Vergleich

Der tatsächliche Aufwand lag insgesamt bei 76 Arbeitsstunden.

Die gestellten Anforderungen wurden erfüllt. Die Software entspricht seinen Erwartungen und macht genau das, was sie soll, und der Kunde ist mit dem Ergebnis zufrieden.

## Qualitätskontrolle

Es sollte sein, dass nach der Anmeldung als ein User, das er nicht einfach ein Url aufrufen, die ein Admin View gibt, aber es ist nicht darüber gedacht.

Lösung: muss bei jeder Admin View Aufruf zu überprüfen, ob $user\_role immer noch 1 (admin) ist.

Das kann man von den Session Daten einfach erreichen.

## Abweichungen, Anpassungen, Entwicklungsmöglichkeiten

Mit Blick auf die Zukunft können wir dieses Projekt in Verbindung mit unserem PSI-Programm entwickeln. PSI ist eine kostenlose Instant-Messaging-Anwendung, die für das XMPP-Netzwerk entwickelt wurde.  
In diesem Fall können wir die Ansicht einsehen, dass dieser Mitarbeiter heute krank oder aus irgendeinem Grund nicht im Unternehmen ist.

In einem anderen Fall gibt es in unserem Unternehmen einen Küchendienst, den jeder Mitarbeiter in einen jeweiligen Abstand erledigen sollte. und wenn der Mitarbeiter, der diesen Dienst leisten sollte, krank ist oder aus irgendeinem Grund nicht im Unternehmen ist, sollte sein Stellvertreter diese Verantwortung übernehmen. Dann kann der Stellvertreter über die Abwesenheit des Verantwortlichen informiert werden, um seine Aufgabe zu übernehmen.

Wir können auch die Mitarbeiter als Admin addieren, aber hier steht die Frage: kann ein Admin anderen Admin löschen? Ich behaupte: Ja, aber muss geachtet werden, dass der Standardbenutzer nicht gelöscht werden darf, oder dass mindestens einen Admin muss immer in der DB bleibt.  
Das kann man erreichen entweder: Der erste Antrag in DB „Admin“ immer außerhalb die Löschung Querreys, oder immer zu überprüfen: ist dieser Admin der letzte Admin ist? Bevor löschen.