

# Entwicklung eines webbasierten ePortfolios

Moritz Mandler & Didier Zielke

Juni 2020

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>3</b>
<b>2</b>	<b>Projektdefinition</b>	<b>3</b>
2.1	Ist-Analyse . . . . .	3
2.2	Anforderungsdefinition (Soll-Konzept) . . . . .	3
<b>3</b>	<b>Projektplanung</b>	<b>4</b>
3.1	Ressourcenplanung . . . . .	4
3.2	Kosten . . . . .	5
<b>4</b>	<b>Projektdurchführung</b>	<b>5</b>
4.1	Softwareentwurf . . . . .	5
4.2	Daten- und Klassenentwurf . . . . .	7
4.3	Realisierung . . . . .	8
4.3.1	Übersicht . . . . .	8
4.3.2	Programmierung der Geschäftslogik und der Datenzugriffs- klassen . . . . .	8
4.3.3	Programmierung der Controllerklassen . . . . .	8
4.3.4	Programmierung der Viewklassen . . . . .	8
4.4	Tests . . . . .	8
<b>5</b>	<b>Soll-/Ist-Vergleich</b>	<b>8</b>
<b>6</b>	<b>Fazit</b>	<b>8</b>
<b>7</b>	<b>Anhang</b>	<b>8</b>
7.1	index.php . . . . .	8

# 1 Einführung

Die IT-Solution & Design GmbH ist eine Lernfirma innerhalb des Berufsförderungswerks Hamburg GmbH (BFW). Als norddeutsches Zentrum für berufliche Rehabilitation und Integration ist das BFW Hamburg kompetenter Partner für Unternehmen, Träger der beruflichen Rehabilitation und Versicherungen und vor allem für Menschen, die aus gesundheitlichen Gründen ihre bisherige Tätigkeit nicht mehr ausüben können. Es soll eine einfache Plattform zur Veröffentlichung einer 'Mappe' der eigenen Arbeitsergebnisse geboten werden. Benutzer müssen sich authentifizieren und können dann die eigene Mappe bearbeiten. Innerhalb dieser Mappe, die hier nur eine Webseite darstellt, können beliebige Download-Dateien, Texte und Bilder eingefügt werden. So soll den Benutzern auch die Möglichkeit gegeben werden Lebensläufe, Arbeitsproben oder Zertifikate auf der eigenen Seite abzulegen. Die IT-Solutions & Design hat vom BFW Hamburg den Auftrag erhalten diese Webanwendung zu erstellen.

## 2 Projektdefinition

### 2.1 Ist-Analyse

Die Teilnehmer\*innen des BFW müssen sich während ihrer Ausbildung auf einen Praktikums- und einen Arbeitsplatz bewerben. Viele Bewerbungen werden dabei per E-Mail versendet. Um eine E-Mail nicht zu groß werden zu lassen, werden dabei manchmal Arbeitsergebnisse, Zertifikate o. ä. nicht mit gesendet. Das könnte den Gesamteindruck im Bewerbungsprozess verschlechtern. Außerdem müssen die Teilnehmer\*innen entscheiden, auf welche Unterlagen, Arbeitsproben und Bilder sie verzichten wollen.

### 2.2 Anforderungsdefinition (Soll-Konzept)

Wie im Pflichtenheft angegeben (siehe Anhang) sollen folgende Anforderungen erfüllt werden:

#### i. Musskriterien

- Neuanlegen, Ändern und Löschen von Benutzern durch den Admin
- Jedem Benutzer ist eine Benutzerseite zugeordnet, die nur von dem jeweiligen Benutzer bearbeitet werden darf
- Neuanlegen, Ändern und Löschen von Texten (auch Links) auf der jeweiligen Benutzerseite

- Hochladen von Bildern und Pdf-Dateien innerhalb der Benutzerseite
- Berücksichtigung verschiedener Berechtigungsstufen (Admin, Benutzer, Gast)
- Nur registrierte Gäste dürfen die Inhalte der Benutzer einsehen. Anlegen eines Gasts über Formular mit Überprüfung der E-Mailadresse (Freischaltung des Gastaccounts über E-Mail-Link) und Versenden eines generierten Kennworts

## **ii. Wunschkriterien**

- Gäste dürfen nur die Portfolios sehen, welche ihnen über eine Freigabe zugeordnet wurden.

## **iii. Abgrenzungskriterien**

- Keine Überprüfung auf Verletzung von Urheberrechten oder Verstöße gegen das Datenschutzgesetz

# **3 Projektplanung**

## **3.1 Ressourcenplanung**

Als Ressource sollen ein PC mit der im Pflichtenheft angegebenen technischen Produktumgebung zur Verfügung gestellt werden. Für die Kalkulation wurde eine Stundenplanung gemacht. Zwei Personen sollen dem Projekt für die folgenden Arbeitsschritte zur Verfügung stehen.

<b>Software-Entwurf</b>	
Use-Case-Diagramm	4h
Klassenmodell	4h
Datenmodell	4h
Summe:	12h
<b>Realisierung</b>	
Entwurf der Testfälle	4h
Programmierung Geschäftslogik	24h
Programmierung Datenzugriffsklassen	10h
Programmierung Controller-Klassen	24h
Programmierung View-Klassen	20h
Summe:	82h
<b>Tests</b>	
Testfälle programmieren und durchführen	10h
Eventuelle Fehlerbeseitigung	4h
Summe:	
<b>Abschluss</b>	
Soll-Ist-Vergleich	6h
Dokumentation	25h
Übergabe	1h
Summe:	32h
<b>Gesamtsumme:</b>	<b>140h</b>

## 3.2 Kosten

Die Kosten für dieses Projekt belaufen sich bei einem Stundensatz von 35,00€ und einem Gesamteinsatz von 140Std. pro Person, wie aus der Ressourcenplanung hervorgeht, auf:

$$2 \text{ Personen} * 140\text{Std} * 35,00\text{€} = 9800\text{€}$$

# 4 Projektdurchführung

## 4.1 Softwareentwurf

Zuerst werden die Anwendungsfälle ermittelt. Es wird drei Hauptanwendungsfälle geben. Die Nutzung als Administrator, als Benutzer und als Gast. Wir entscheiden uns für eine zentrale Login-Oberfläche und danach zu einer strikten Trennung

der Hauptanwendungsfälle.

Für die Umsetzung der Wunschkriterien wird die Benutzeroberfläche unterteilt in:

- Ansichten aller Seiten des eigenen Portfolios mit Bearbeitungsmöglichkeit
- Möglichkeit zum erstellen und löschen neuer Seiten innerhalb des eigenen Portfolios
- Gästeverwaltung zum erstellen und löschen von Gästen des eigenen Portfolios und setzen derer Berechtigungen

Als Vorgehensweise bei der Softwareentwicklung kommt der Ansatz der strukturierten Programmierung oder die objektorientiert Programmierung in Betracht. Die objektorientierte Programmierung bietet den Vorteil, dass das Model-View-Controller-Modell (MVC-Modell) zur Anwendung kommen könnte. Dadurch gibt es ein einheitliches Vorgehen, welches leicht erweitert werden kann. Bei der strukturierten Programmierung besteht die Gefahr, dass ein Wust von ineinander geschachtelten Dateien die Übersicht erschwert.

Als weiterer Vorteil von objektorientierter Programmierung kann man die Datenkapselung anführen. In dem der Zugriff auf Instanzvariablen als private gekennzeichnet wird, können diese Variablen von außerhalb der Klasse nicht verändert werden, bzw. nur durch geeignete Methoden, wenn es sinnvoll erscheint. Dadurch wird erreicht, dass die Instanzvariablen immer in einem fehlerfreien Zustand vorliegen.

Aus diesen Gründen haben wir entschieden die Anwendung objektorientiert zu entwickeln.

Für die Verwendung der programmierten Klassen sollen Namespaces zum Einsatz kommen, weil dadurch kein Import für jede benutzte Klasse nötig ist.

Wenn bei dem Prinzip der Objektorientierung eine strikte Trennung der Zuständigkeiten von Klassen und Methoden beachtet wird, ist die Wartbarkeit des Codes einfacher. Hier ist ein einfaches MVCModell geplant, bei welchem die Model-Klassen die Geschäftslogik implementieren, die View-Klassen die Ein- und Ausgaben darstellen und die Controller-Klassen für die Verbindung dazwischen eingesetzt werden sollen.

Für jeden Anwendungsfall soll es mindestens eine Controller-Klasse geben, die über den Frontcontroller aufgerufen wird. Außerdem soll es Klassen geben, die für die Schnittstelle der ankommenden HTTP-Anfragen zuständig sind (Request) und Klassen, die für die Antwort benutzt werden (Response). Neben den reinen Modelklassen wird geplant, Datenzugriffsklassen zu implementieren, die für die Datenbankzugriffe (CRUD) verwendet werden sollen. Durch die Verwendung eines DAO-Interfaces könnte man schnell die Datenzugriffsklassen für andere Datenbanksysteme erweitert werden. Aus Zeitgründen wurde aber hierauf verzichtet.

## 4.2 Daten- und Klassenentwurf

Das Klassenmodell und das Datenmodell befindet sich im Anhang: Da ein Benutzer mehrere Seiten in seinem Portfolio haben kann, wurde hier eine 1:n Beziehung im Datenmodell genutzt. Da eine Seite sich aus mehreren Inhalten zusammensetzen kann, wird auch hier eine 1:n Beziehung im Datenmodell genutzt. Da ein Gast keine eigenen Seiten hat, wird eine Berechtigungs-Tabelle für Seiten der Benutzer genutzt. Ein Gast kann mehrere Berechtigungen haben, aber jede Berechtigung muss eindeutig einer Person zugeordnet sein. Daher wird hier eine 1:n Beziehung im Datenmodell genutzt.

Im Klassenmodell wird festgelegt, dass jeder User eine Liste von Seiten beinhaltet. Jede Seite beinhaltet wiederum eine Liste von Inhalten.

Aus Gründen der Einfachheit wird im Klassenmodell und im Datenmodell zwischen den Benutzergruppen Administrator, Benutzer und Gast über eine Eigenschaft „status“ unterschieden, ansonsten jedoch das gleiche Model genutzt.

## 4.3 Realisierung

### 4.3.1 Übersicht

### 4.3.2 Programmierung der Geschäftslogik und der Datenzugriffsklassen

### 4.3.3 Programmierung der Controllerklassen

### 4.3.4 Programmierung der Viewklassen

## 4.4 Tests

## 5 Soll-/Ist-Vergleich

## 6 Fazit

## 7 Anhang