Game Artist und Game Programmer in spe

Dokumentation zum zweiten Prototyping

# Two-Button Game

**2D-basiertes Two-Button Game zur Ermittlung von Arbeiten in der Unity Engine**

Abgabetermin: Köln, den 23.01.2025

**Präsentanten:**

Frank-Peter Scheuer

Erik Gattermann

Eleftherios Kotsaridis

Tony Kräuter

Niran Hasilik



# Ausbildungsbetrieb:

School of Games

Sachsenring 79

50677 Köln-Neustadt-Süd

## Inhaltsverzeichnis

|  |  |  |
| --- | --- | --- |
| **A** | **Inhaltsverzeichnis** |  |
| *A* | *Inhaltsverzeichnis* | *II* |
| *B* | *Abkürzungsverzeichnis* | *V* |
| *C* | *Tabellenverzeichnis* | *V* |
| *D* | *Bilderverzeichnis* | *V* |
| *E* | *Literaturverzeichnis* | *VII* |
| *F* | *Anhangsverzeichnis* | *VIII* |
| [*1*](#_bookmark0) | [*Einleitung*](#_bookmark0) | [*1*](#_bookmark0) |

* 1. [Einleitung 1](#_bookmark0)
  2. [Projektbeschreibung 1](#_bookmark1)
  3. [Projektziel 2](#_bookmark2)
  4. [Projektbegründung 2](#_bookmark3)
  5. [Projektschnittstellen 2](#_bookmark4)
  6. [Projektabgrenzung 2](#_bookmark5)
  7. [Projektphasen 3](#_bookmark6)

1. [*Planungs- und Analysephase 3*](#_bookmark7)
   1. [Ist-Analyse 3](#_bookmark7)
   2. [Soll-Konzept 4](#_bookmark8)
   3. [Wirtschaftlichkeitsanalyse 4](#_bookmark9)
      1. [„Make or Buy“-Entscheidung 4](file:///D:\Dokumente\6.%20Momentane%20Projekte%202024\2024%20School%20of%20Games\1.%20Game%20Artist\Make_or#_2.3.1_)
      2. [Projektkosten 5](#_bookmark10)
      3. [Amortisationskosten (Seite 1) 6](#_bookmark11)
      4. [Amortisationskosten (Seite 2) 7](#_bookmark12)
   4. [Pflichtenheftanalyse 7](#_bookmark13)
2. [*Definitionsphase 8*](#_bookmark14)
   1. [Zeitplanungsdefinition 8](#_bookmark14)
   2. [Risikodefinition 8](#_bookmark15)
   3. [Definition des Qualitätssicherungskonzept 8](#_bookmark16)
3. [Entwurfsphase 9](#_bookmark17)
   1. [Zielplattform 9](#_bookmark17)
   2. [Entwurf der Testfertigstellung 9](#_bookmark18)
   3. [Entwurf des Zertifikatserstellung 9](#_bookmark19)
   4. [Entwurf der Zertifikatsübersicht 9](#_bookmark20)
   5. [Entwurf der Testszenarien 10](#_bookmark21)
   6. [Erstellungsvorgang der kompletten DV-Entwurfsdokumente 10](#_bookmark22)
4. [Implementierungsphase 11](#_bookmark23)
   1. [Implementierung der Projekthierachie 11](#_bookmark23)
   2. [Implementierung der Testfertigstellung 11](#_bookmark24)
   3. [Implementierung des Zertifikatsbereiches 11](#_bookmark25)
   4. [Implementierung der Zertifikatsübersicht 12](#_bookmark26)
   5. [Testprotokolle des ersten Testlaufs zur Qualitätssicherung 12](#_bookmark27)
   6. [Implementierung der Fehlerbeseitigung 12](#_bookmark28)
5. [Qualitätssicherungsphase 12](#_bookmark29)
   1. [Protokollierung der Softwarepräsentation des Teilprojektes 12](#_bookmark29)
   2. Protokolle der Code-Reviews 12
   3. [Protokollierte Abnahme des Projektes 13](#_bookmark31)
   4. [Entwicklerdokumentation 13](#_bookmark32)
   5. [Qualitätsmanagementbericht 13](#_bookmark33)
   6. [Benutzerhandbuch 13](#_bookmark34)
   7. [Prozessorientierter Projektbericht 13](#_bookmark35)
6. [Projektbewertungsphase 14](#_bookmark36)
   1. [Soll-/Ist-Vergleich 14](#_bookmark36)
   2. [Gewonnene Erkenntnisse 15](#_bookmark37)
   3. [Abweichung vom Projektantrag 15](#_bookmark38)
   4. [Ausblick 15](#_bookmark39)

**B Abkürzungsverzeichnis**

SoG School of Games

MySQL My Structured Query Language

IHK Industrie und Handelskammer

DEV Development (Entwicklung)

PHP Personal Home Page

HTTP Hypertext Transfer Protocol

UML Unified Modeling Language

DV Datenverarbeitung

CSS Cascading Style Sheets

IDE Integrated Development Environment

**C Tabellenverzeichnis**

[Tabelle 1 Zeitplanung S.3](#_bookmark6)

[Tabelle 2 Kostenaufstellung S.5](#_bookmark10)

[Tabelle 3 Zeiteinsparung S.6](#_bookmark11)

[Tabelle 4 Soll-/Ist-Vergleich S.14](#_bookmark36)

|  |  |  |
| --- | --- | --- |
| **D** | **Bilderverzeichnis** |  |
| [Bild 1](#_bookmark40) | [Projektantrag](#_bookmark40) | [S.16-20](#_bookmark40) |
| [Bild 2](#_bookmark41) | [Bewilligter Projektantrag](#_bookmark41) | [S.21](#_bookmark41) |
| [Bild 3](#_bookmark42) | [Ist-Analyse](#_bookmark42) | [S.22](#_bookmark42) |
| [Bild 4](#_bookmark43) | [Wirtschaftlichkeitsanalyse](#_bookmark43) | [S.23](#_bookmark43) |
| [Bild 5](#_bookmark44) | [Amortisationsanalyse](#_bookmark44) | [S.24](#_bookmark44) |
| [Bild 6](#_bookmark46) | [Anwendungsfall](#_bookmark46) | [S.29](#_bookmark46) |
| [Bild 7](#_bookmark47) | [Projektablaufplan](#_bookmark47) | [S.36](#_bookmark47) |
| [Bild 8](#_bookmark48) | [Risikoanalyse](#_bookmark48) | [S.37](#_bookmark48) |
| [Bild 9](#_bookmark49) | [Qualitätssicherungsnormen](#_bookmark49) | [S.40](#_bookmark49) |
| [Bild 10](#_bookmark50) | [Grafische Entwürfe der Teilbereiche](#_bookmark50) | [S.41-43](#_bookmark50) |
| [Bild 11](#_bookmark53) | [Testentwürfe](#_bookmark53) | [S.44](#_bookmark53) |
| [Bild 12](#_bookmark54) | [Klassendiagramm](#_bookmark54) | [S.45](#_bookmark54) |
| [Bild 13](#_bookmark55) | [Datenflussdiagramm](#_bookmark55) | [S.46](#_bookmark55) |
| [Bild 14](#_bookmark56) | [ER-Modell](#_bookmark56) | [S.47](#_bookmark56) |
| [Bild 15](#_bookmark57) | [Layout und Codebeispiele der Testfertigstellung](#_bookmark57) | [S.48-49](#_bookmark57) |
| [Bild 16](#_bookmark59) | [Layout und Codebeispiele der Zertifikatserstellung](#_bookmark59) | [S.50-51](#_bookmark59) |
| [Bild 17](#_bookmark60) | [Layout und Codebeispiele der Zertifiksatsübersicht](#_bookmark60) | [S.52-53](#_bookmark60) |
| [Bild 18](#_bookmark65) | [Auszug aus der Entwicklerdokumentation](#_bookmark65) | [S.58](#_bookmark65) |
| [Bild 19](#_bookmark66) | [Abschließendes Qualitätssicherungskonzept](#_bookmark66) | [S.59-64](#_bookmark66) |
| [Bild 20](#_bookmark67) | [Benutzerhandbuch](#_bookmark67) | [S.65-74](#_bookmark67) |

**E Literaturverzeichnis**

1. Informationen zur TRAIN & EDUCATION Ltd.: https:/[/www.trainandeducation.de/1-0-Startseite.html](http://www.trainandeducation.de/1-0-Startseite.html)
2. Grundlagen in das Framework Laravel https://laracasts.com/
3. Grundlagen der Programmiersprache PHP https:/[/www.php.de/](http://www.php.de/)
4. Grundlagen in das Framework Bootstrap https://getbootstrap.com/
5. Grundlagen der Versionsverwaltung https://gitlab.com/gitlab-org/gitlab

**F Anhangsverzeichnis**

* 1. [Projektantrag mit Bewilligung 16](#_bookmark40)
  2. [Strukturierte Ist-Analyse 22](#_bookmark42)
  3. [Strukturierte Wirtschaftlichkeitsanalyse (Darstellung des Teilprojektes) 23](#_bookmark43)
  4. [Strukturierte Amortisationsanalyse (Darstellung des Teilprojektes) 24](#_bookmark44)
  5. [Strukturierte Pflichtenheftanalyse (Darstellung des Teilprojektes) 25](#_bookmark45)
  6. [Projektablaufplan (Netzplan) 36](#_bookmark47)
  7. [Risikoanalyse 37](#_bookmark48)
  8. [Strukturierte Qualitätssicherungsnormen 38](#_bookmark49)
  9. [Entwürfe der Teilbereiche 41](#_bookmark50)
  10. [Testentwürfe 44](#_bookmark53)
  11. [Klassendiagramm 45](#_bookmark54)
  12. [Datenflussdiagramm 46](#_bookmark55)
  13. [ER-Modell 47](#_bookmark56)
  14. [Layout und Codebeispiele der Testfertigstellung 48](#_bookmark57)
  15. [Layout und Codebeispiele der Zertifikatserstellung 50](#_bookmark58)
  16. [Layout und Codebeispiele der Zertifikatsübersicht 52](#_bookmark60)
  17. [Protokoll des ersten Testlaufs des Teilprojektes 54](#_bookmark61)
  18. [Abschlussbericht des Projektes 55](#_bookmark62)
  19. [Protokoll des Code-Review 56](#_bookmark63)
  20. [Unterzeichnetes Abnahmeprotokoll 57](#_bookmark64)
  21. [Auszug aus der Entwicklerdokumentation 58](#_bookmark65)
  22. [Abschließendes Qualitätssicherungskonzept 59](#_bookmark66)
  23. [Benutzerhandbuch 66](#_bookmark67)
  24. [Glossar des Abschlussprojektes 76](#_bookmark68)

1. ***Einleitung***

**1 Einleitung**

# Einleitung

In der folgenden Projektdokumentation kommt es zur Erläuterung des Projektes des Teams das innerhalb seines Prototyping als Game-Artist erfolgte. Das Projekt wird in den Räumen der Berufsbildenden Schule der School of Games (**SoG**) durchgeführt. Die **SoG** ist ein privates Schulungsunternehmen mit Sitz in Köln, die zurzeit 50 Angestellte beschäftigt. Zu den Produkten der SoG gehören Beratungen für Weiterbildungen und Seminare in den Bereichen der Spieleentwicklung in künstlerischen und programmiertechnischen Bereich. Dabei werden aber auch Stoffe gelehrt die in Teams und Firmen, wie aber auch anderen Medienunternehmen genutzt werden können. Außerdem bietet die **SoG** Zertifizierungen für die Medienbranche an, die zum Abschluss eines Gesamtkurses ausgegeben wird. Zielstellung dieser Dokumentation ist, die Entwicklung des Projektes von der Planung bis zur Implementierung grafisch, nachvollziehbar und verständlich darzustellen und dies mit geeigneten Diagrammen und weiteren Dokumenten zu unterstützen.

# Projektbeschreibung

Im Rahmen dieses Gesamtprojektes soll der Prozess der Entwicklung eines Two-Button Game im theoretischen und praktischen grafischen Aufbau dargestellt werden. In einzelnen Teilprojekt wird der Prozess der abschließenden Arbeit an Gametests und der darauffolgenden Levelerstellung beschäftigen.

Eine schriftliche Auftragsvergabe erfolgte in dem den Unterricht unterstützenden Online Dienst Discord und wurde von den Dozenten der SoG verfasst. Diese wurde in einer Kopie des Bildes im Anhang [**A.1:**](#_bookmark40)[**Bewilligter Projektantrag**](#_bookmark40)auf Seite **16** hinterlegt.

***1 Einleitung***

# Projektziel

Ziel des Teamprojektes ist es ein 2D-Game zu entwickeln, in dem man die Spielfigur mit nur 2 Tasten durch das Spielfeld und Missionen führt. Dadurch soll das nötigen Wissen bei den Schülerin vermittelt werden und Bewerbungsartikel entstehen.

# Projektbegründung

Die Erstellung der vom Auftraggeber benötigten und durch das vorgegebene Thema erläuterten Aufgaben, soll das Produkt in Mobil aber auch Arcade Geräten stattfinden können.

# Projektschnittstellen

Um an die Daten zu gelangen, muss die Anwendung mit den Eingaben des Spielers kommunizieren und Werte abfragen können.

Aufgrund der Zeitbegrenzung für das Gesamtprojekt und der Teilprojekte der anderen Teammitglieder wurden Funktionen zur Weiterleitung von Daten in das Spiel erstellt. Deswegen ist eine Abstimmung mit den Projektbeteiligten notwendig.

Problematiken hierbei wurden aus den Mitgliedern des Teams teilweise übernommen, aber auch zum Teil gelöscht.

# Projektabgrenzung

Da der Projektumfang aufgrund der **SoG**-Vorgaben begrenzt ist, ist die Weiterführung des Projektes zu einen größeren Umfang nicht zielführend und es wird deswegen draufverzichtet.

***1 Einleitung***

# Projektphasen

Eine grobe Zeitplanung mit den Hauptphasen lässt sich aus der untenstehenden **Tabelle 1 (Zeitplanung)** lässt sich entnehmen.

|  |  |  |
| --- | --- | --- |
| **Projektphase** | **Geplante Zeit** | **Zeitpunkt** |
| **Analysephase** | 6h | 06.01.2025 |
| **Definitionsphase** | 6h | 07.01.2025 |
| **Entwurfsphase** | 25h | 07.01.-15.01.2025 |
| **Implementierungsphase** | 20h | 08.01.-15.01.2025 |
| **Projektabschlussphase** | 15h | In Arbeit |
| ***Puffer*** | *Geplant: 18h* |  |
| **Gesamt** | **100h** |  |

Tabelle 1: Zeitplanung

**2 Planungs- und Analysephase**

# Ist-Analyse

Wie unter **1.1** (**Projektbeschreibung**) beschrieben wird in meinem Teilprojekt der Anteil der Bereitstellung der Tests, deren Auswertung und der darauffolgenden Erstellung der Zertifikate vollzogen. In der Ist-Analyse wird herausgearbeitet, wie die Daten aktuell ermittelt und verarbeitet werden.

Wie bereits erläutert in 1.3 (Projektbegründung) entsteht ein großes Potential von Fehlern in der bisherigen Erstellung. Die Problematiken entstehen hierbei bei einer Fehlerhaften Lesung der Daten bzw. Bewertungen, verpasste Terminierte Abgaben und fehlerhaftes pflegen von Ergebnissen.

Im Bereich Zertifikatserstellung kann es bei der Sichtung der Daten zu Fehlern kommen. In allen Bereichen fällt sehr viel überflüssige manuelle Arbeit an, was zu einem sehr großen Zeitaufwand führt. Außerdem ist an keiner Stelle eine automatisierte fachliche Prüfung anhand von fest definierten Plausibilitäten vorgesehen, sodass Fehler nicht direkt erkannt werden können.

Dies wurde unter [**A.2 Strukturierte Ist-Analyse**](#_bookmark42)auf Seite **22** in einem Beispielfall visuell in einem Schaubild erläutert, bei dem es bei jedem dargestellten Pfeil bisher zu Fehlern kommen kann.

1. ***Analysephase***

# Soll-Konzept

Für die in der Ist-Analyse festgestellten und grafisch erläuterten Probleme soll nun in diesem Teilprojekt Abhilfe geschaffen werden, um auch eine Zeitersparnis zu erreichen. Dies erfolgte mithilfe der **IDE** *Microsoft Visual Studio Code* und der *Versionsverwaltung* **GitLAB** um im Gesamtprojekt mitzuarbeiten. Zur Lösung sollen Dateien im *Frontend*- wie im *Backend* entwickelt und bereitgestellt werden.

Um die Frontendanforderungen zu erfüllen wurde in der Entwicklungsumgebung im Zusammenspiel mit dem *Framework* **Bootstrap** eine grafische Oberfläche entwickelt, die die geforderten Funktionen einbringen werden. Hierdurch wird die Problematik der fehlerhaften Eintragung von Bewertungen oder anderen Teilnehmerdaten abgegriffen da Sie alle im selben Format eingetragen werden.

Zu Erfüllung der Backendanforderungen wurden außerdem in der Entwicklungsumgebung die Arbeit mit dem *Framework* **Laravel** die Möglichkeit geschaffen die Funktionen der Ausstattung von bestehenden Tests, Berechnung der Testergebnisse und deren Speicherung, sowie der Erstellung von Zertifikaten, deren Speicherung sowie deren Ausdruck eingebracht. Durch die Möglichkeit der Beschränkung auf bestimmte Dozentengruppe kann außerdem ein Grad des Datenschutzes erstellt werden.

# Wirtschaftlichkeitsanalyse

# „Make or Buy“-Entscheidung

Die im Projekt verwendeten Daten betreffen das Kerngeschäft des Schulungsunternehmens der **AFP.** Insbesondere die personenbezogenen Daten sollen nicht an externe Anwendungen weitergegeben werden, sondern im eigenen Unternehmen bearbeitet werden. Zudem soll die Anwendung in einem in dem Gesamtprojekt entstehenden Datenbanksystem entstehen und mit diesem zusammenarbeiten und kommunizieren, welches proprietäre Schnittstellen bereitstellt. Als strategische Entscheidung, das Kerngeschäft in PHP umzusetzen und dies nicht durch externe Dienstleister machen zu lassen und betriebsinterne Daten in externe Anwendungen zu exportieren, wurde sich bei diesem Projekt für eine Eigenentwicklung entschieden.

***2 Analysephase***

# Projektkosten

Im Folgenden werden die Kosten, die im Laufe des Teilprojektes anfallen, kalkulieren. Neben den anfallenden Personalkosten der Entwicklung und weiterer an dem Teilprojektbeteiligter mussten auch die Aufwendungen für die verwendeten Ressourcen, die unter **2.3** (**Machbarkeitsstudie**) aufgeführt wurden, eingeplant werden.

Die Personalkosten werden auf Basis von dem Auftraggeber angegebenen Stundensätze kalkuliert. Die aufgeführten Stundensätze werden aufgrund der kalkulatorischen Kosten die für Entwickler und Praktikanten in der Anwendungsentwicklung sowie die Sozialaufwendungen der Train & Education entstehen. Hinzu kommen die oben erwähnten Kosten, die für die Nutzung der Ressourcen anfallen. Für einen Mitarbeiter der Anwendungsentwicklung wird ein Stundensatz von **83,00€** angegeben.

Der Stundensatz für einen Auszubildenden ist auf **63,00€** festgelegt. Als Satz für die Ressourcennutzung werden pauschal **5,00€** angenommen. Die Durchführungszeit des Projektes beträgt 70 Stunden. In Tabelle **2 (Kostenaufstellung)** sind die Kosten unterteilt nach den einzelnen Projektvorgängen aufgelistet, sowie summiert dargestellt, um die Gesamtkosten, die während des Teilprojektes anfallen, zu erhalten. Diese belaufen sich auf **5.376,00€.**

Unter [**A.3 Strukturierte Wirtschaftlichkeitsanalyse**](#_bookmark43)auf Seite **23** wird das in einem hinterlegten Schaubild visuell erläutert.

|  |  |  |  |
| --- | --- | --- | --- |
| **Vorgang** | **Zeit** | **Kosten pro Stunde** | **Kosten** |
| **Entwicklungskosten** | 70h | 63,00 € + 5,00€ = 68,00€ | 4.760,00 € |
| **Fachgespräch** | 4h | 83,00 € + 5,00€ = 88,00€ | 352,00 € |
| **Abnahmetest** | 1h | 83,00 € + 5,00€ = 88,00€ | 88,00 € |
| **Code-Review** | 2h | 83,00 € + 5,00€ = 88,00€ | 176,00 € |
|  |  |  | **5.376,00 €** |

Tabelle 2: Kostenaufstellung

***2 Analysephase***

# Amortisationskosten (Seite 1)

Im Folgenden habe ich die Amortisationsdauer betrachtet, also ab welchen Zeitpunkt sich die Entwicklung des Teilprojektes gewinnbringend lohnt bzw. amortisiert. Die Anschaﬀungskosten habe ich bereits in der obenstehenden Tabelle **2 (Kostenaufstellung)** bestimmt. Durch die Automatisierung der Erstellung der Excel-Dateien lässt sich die Zeit dieser manuellen Arbeit einsparen. Außerdem entfallen die Versionierung und Nachbearbeitung sowie Kontrolle der Dateien wie unter **2.1** (**Ist-Analyse**) beschrieben ist. Anhand von Erfahrungen aus den letzten Durchläufen der Test- und Zertifikatserstellung und eigenen bisher erwirkten Erfahrungen können die Vorgangszeiten ermittelt werden. In der neuen Anwendung muss durch einen Mitarbeiter die Erstellung der neuen Dateien einmalig nach einer **DEV** angestoßen werden, wenn sichergestellt wurde, dass alle Daten in der Datenbank hinterlegt worden sind. Für diesen Arbeitsschritt werden 20 Minuten kalkuliert. Grundsätzlich wird wie unter **3.2.2** (**Projektkosten**) ein Personentag (PT) mit 7,6 Stunden angenommen, was 456 Minuten entspricht. Durch Gespräche mit den Mitarbeitern werden die genannten Schritte im Durchschnitt 4-mal pro Monat ausgeführt, da wie unter 1.1 (Projektbeschreibung) erwähnt ein Modultest durchgeführt wird und dies mehrmals erfolgen muss.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vorgang** | **Anzahl**  **pro Jahr** | **Zeit alt**  **pro Vorgang/Jahr** | **Zeit neu**  **pro Vorgang/Jahr** | **Einsparung**  **pro Jahr** |
| **Durchführung und Bewertung** | 40 | 2 **PT** ≙ 912 min/36.480 min | 20 min | 36.460 min |
| **Zertifikatserstellung** | 4 | 3 **PT** ≙ 1.368 min/5.472 min | 5 min | 5.467 min |
| **Versionierung der Dateien** | 4 | 0,5 **PT** ≙ 228 min/912 min | - | 912 min |
| **Aufbereitung der Dateien** | 2 | 4 **PT** ≙ 1.824 min/3.648 min | - | 3.648 min |
| **Kontrolle der Daten** | 4 | 1 **PT** ≙ 456 min/1.824 min | - | 1.824 min |
| **Betrieb der Anwendung** | 4 | - | 1 **PT** ≙ 456 min | -456 min |
| **Kompletter Vorgang** |  | **4.788 min/48.336 min** | **481 min/1.924 min** | **46.412 min** |

Tabelle 3: Zeiteinsparung

Dadurch ergibt sich eine jährliche Einsparung von 46.412 𝑚𝑖𝑛 \*(25+5) €/h =23.206,00€.

60 𝑚𝑖𝑛/ℎ

Die Amortisationszeit beträgt also 5.376,00 €

23.206 €/𝐽𝑎ℎ𝑟

≈ 0,23 Jahre ≈ 84 Tage.

1. ***Analysephase***

# Amortisationskosten (Seite 2)

Unter [**A.4 Strukturierte Amortisationsanalyse**](#_bookmark44)auf Seite **24** wird die Zeiteinsparung in einem Schaubild visuell erläutert. Wie bereits beschrieben, findet dieser Prozess etwa viermal im Monat statt. Aus der ermittelten Zeit lässt sich ableiten, dass sich aufgrund der großen Einsparung an manueller Arbeit das Projekt schon nach der ersten Ausführung amortisiert hat, da der Prozess circa durchschnittlich viermal im Monat stattfindet. Daher lässt sich das Projekt als sehr wirtschaftlich einordnen.

# Pflichtenheftanalyse

Im Laufe des Gesamtprojektes soll die Erstellung von Fragen, Tests und Zertifikaten automatisiert werden und damit die bestehende manuelle Arbeit ersetzen.

Für das Gesamtprojekt wurde vom Auftraggeber daher ein Pflichtenheft erstellt wofür ich in eine zu meinem Teilprojekt betreffende Analyse erstellte. Diese ist im Anhang unter [**A.5 Strukturierte**](#_bookmark45)[**Pflichtenheftanalyse**](#_bookmark45)auf **Seite 25 - 35** aufgeführt.

1. ***Definitionsphase***

**3 Definitionsphase**

# Zeitplanungsdefinition

Auf Grundlage der in der Planungs- und Analysephase erstellten Unterlagen habe ich in der Definitionsphase den Ablaufplan in Form eines Netzplans hinterlegt. Dieser liegt auf dem Grundkonzept der im Antrag angegebenen Anlagen auf und erfasst die einzelnen Prozesse zur Entwicklung der Anwendung. Es dient dazu am Ende der Projekterstellung als Leitfaden zudienen, sowie den zeitlichen Ablauf der Erstellung zu erkennen und wie das Projekt unter Zeitdruck verwirklicht werden konnte. Abgebildet ist dies Verdeutlichung der Zeitplanung diese in [**A.6:**](#_bookmark47)[**Projektablaufplan (Netzplan)**](#_bookmark47)und Seite **36** im Anhang.

# Risikodefinition

Auf Grundlage der zuvor erstellten Unterlagen wurde in der Definierungsphase eine Gegenüberstellung der identifizierten Risiken und Maßnahmen erstellt, die in der Form eines verkürzten weiteren Ablaufplans unter [**A.7: Risikoanalyse der Zertifikatserstellung**](#_bookmark48)auf Seite **37** hinterlegt wird. In dieser sind die Maßnahmen zur Umsetzung detailliert auf die Entwicklungsbereiche ausgelegt und die möglichen Risiken genannt. Des Weiteren werden die Möglichkeiten genannt diese Risiken zu vermindern bzw. zu beseitigen. Im späteren Verlauf des Projektberichtes wird unter **6 Qualitätssicherungsphase** auf Grundlage der Risikoanalyse die Testdokumentation erstellt.

# Definition des Qualitätssicherungskonzept

Zuletzt wurde in der Definierungsphase ein Konzept zur Qualitätssicherung erstellt. In diesem sind die Regelstellung der Softwareentwicklung aufgeführt, um mögliche Problematiken zu vermeiden. Um einen besseren Überblick zu verschaffen habe ich diese zusammengefasst und in einem vorhererstellten Dokument untergebracht was unter [**A.8: Qualitätssicherungsnormen**](#_bookmark49)auf **Seite 38**

**- 40** dargestellt ist.

## Entwurfsphase

**4 Entwurfsphase**

# Zielplattform

Wie bereits unter **1.1 Projektbeschreibung** erwähnt soll mein Abschlussprojekt in der letzten Phase eines Gesamtprojektes entstehen und daher mit den erstellten Anwendungen webbasiert funktionieren. Dies machen das Deployment und die Aktualisierung auf eine neue Version einfacher. Die Daten, auf die zugegriffen werden soll, werden nach der Erstellung des Projektes in diesem eingebracht. Als Programmiersprache soll laut der Teamentscheidung *PHP* verwendet werden. Laufen wird die Anwendung auf dem hausinternen Server, auf dem eine MySQL bzw.

Maria Datenbank in Verbindung mit dem Apache Server laufen wird.

# Entwurf der Testfertigstellung

Auf Grundlage der im vorherigen Teilprojekt, eingebrachten Testerstellung ist in diesem Schritt deren Auswertung mit Funktionen und eine Ansicht der erstellten Tests unterzubringen. Das Layout zur Ausstattung und Benotung ist unter [**A.9 Grafische Entwürfe der Teilbereiche**](#_bookmark50)auf Seite **41** eingebracht.

# Entwurf des Zertifikatserstellung

Auf Basis der in der Benutzerverwaltung hinterlegten Daten sollen in dieser Oberfläche die Funktionen der Aufrufung der Informationen und deren Einbringung in ein hinterlegtes Formular für ein Zertifikat erstellt werden. Das Layout zur Erstellung von Zertifikaten ist unter [**A.9**](#_bookmark51)[**Grafische Entwürfe der Teilbereiche**](#_bookmark51)auf Seite **42** angezeigt.

# Entwurf der Zertifikatsübersicht

Die benoteten Tests und Zertifikate sollen in einer weiteren Ansicht für bestimmte Benutzergruppen (Externe Dozenten) gesperrt werden. Das Layout zur Beschränkung von Zertifikaten ist unter [**A.9**](#_bookmark52)[**Entwürfe der Grafische Teilbereiche**](#_bookmark52)auf Seite **43** angezeigt.

***4 Entwurfsphase***

# Entwurf der Testszenarien

Um die Qualität in meinem Projekt zu sichern wurden von mir vor der Erstellung der DV-Entwurfs- dokumenten konstruktive Ideen gesammelt und daraus unterschiedliche Testszenarien entwickelt.

Diese Entwürfe wurden vom Entwickler unter [**A.10 Testentwürfe**](#_bookmark53)auf Seite **44** visuell abgespeichert.

# Erstellungsvorgang der kompletten DV-Entwurfsdokumente

In diesem Erstellungsvorgang wurden auf Basis der Daten aus den vorherstehenden Teilprojekten verschiedenen Diagramme und Modelle entwickelt.

Eine Darstellung der Beziehung der Klassen über verschiedene Schnittstellen ist unter

[**A.11 Klassendiagramm**](#_bookmark54)auf Seite **45** angezeigt.

Eine Darstellung der Art der Verwendung, die Bereitstellung und Veränderung von Daten innerhalb der Anwendung ist unter [**A.12 Datenflussdiagramm**](#_bookmark55)auf Seite **46** dargestellt.

Eine Visualisierung der Datenmodellierung des Informationssystems im Teilprojekt ist unter

* 1. [**ER-Modell**](#_bookmark56)auf Seite **47** ausgearbeitet.

## Implementierungsphase

**5 Implementierungsphase**

# Implementierung der Projekthierachie

Zur Implementierung der Projekthierachie wurde im Team des Gesamtprojektes eine Übereinkunft zum Framework ***Laravel*** eingebracht, was die Skriptsprache **PHP** unterstützt, in die das System erstellt werden soll. Gemeinsam mit der freien Software wurde mit der verteilten Versionsverwaltung eine Basis zur Arbeit auf der gemeinsamen Oberfläche mit einbringen von Dateien eingebracht. Durch das Framework wurde dabei ein Basisprojekt zur Bearbeitung angelegt von den verschiedenen Pfaden zu den unterschiedlichen Funktionen (Controllern) führen sollen. In dem Frontendbereich wurde das Framework ***Bootstrap*** eingebracht um die Daten zu dem unterschiedlichen Teilbereich den Entwürfen entsprechend anzulegen und diese visuell im Webbrowser darzustellen.

# Implementierung der Testfertigstellung

In diesem Schritt soll die Ansicht für die freigegebenen Tests und deren Benotung eingebracht und mit den **CSS**-Vorgaben grafisch aufgearbeitet werden. Des Weiteren wurden die Funktionen für die entstanden grafische Oberfläche eingebracht. Jeder Kus erhält dafür einen Aussagekräftigen Titel und kann über diesen einer Klasse zugeordnet werden. Um die einzelnen Attribute mit den passenden Datenbankspalten zu verbinden, wurden diese in der Controllerfunktion untergebracht.

Diese werden in Anhang [**A.14 Layout und Codebeispiele der Testfertigstellung**](#_bookmark57)unter **S. 48 - 49**

mit im Code befindlichen Kommentaren hinterlegt.

# Implementierung des Zertifikatsbereiches

In dieser Maßnahme habe ich die Ansicht der Zertifikatserstellung deponiert und mit den **CSS**- Vorgaben grafisch aufgearbeitet. Zudem wurden Aufgaben der Erstellungsoberfläche eingefügt. Die Klassen erhalten hierbei einen Aussagekräftigen Aufgabentitel und können über dem Controller diesen zugeordnet werden. Um die einzelnen Attribute mit den passenden Datenbankspalten zu verbinden, wurden diese mit der echo-Methode abgerufen. Diese werden in Anhang [**A.15 Layout**](#_bookmark58)[**und Codebeispiele der Zertifikatserstellung**](#_bookmark58)unter **S. 50 - 51** mit im Code befindlichen Kommentaren dargelegt.

***5 Implementierungsphase***

# Implementierung der Zertifikatsübersicht

In dieser Aktion wurde die Zertifikatsübersichtsdarstellung eingebettet, um Sie mit den **CSS**- Vorgaben grafisch aufzuarbeiten. Außerdem wurden die Aktionen der Darstellung in einem Controller angebunden. Diese werden in Anhang [**A.16 Layout und Codebeispiele der**](#_bookmark60)[**Zertifikatsübersicht**](#_bookmark60)unter **S. 52 - 53** mit im Code befindlichen Kommentaren gezeigt.

# Testprotokolle des ersten Testlaufs zur Qualitätssicherung

Auf Grund der entworfenen Testentwürfe werden zur Qualitätssicherung nach der Implementierung der Oberflächen und Ihrer Funktionen verschiedene Ersttests durchgeführt und in Protokollen festgehalten.

Diese werden im Anhang [**A.17 Protokoll des ersten Testlaufs**](#_bookmark61)auf **S. 54** mit Ihrem Ergebnis festgehalten.

# Implementierung der Fehlerbeseitigung

Auf Grundlage der entstandenen Tests wurde die Fehlerbeseitigung durchgeführt und in die Anwendungen eingebracht.

**6 Projektabschlussphase**

# Protokolle der Code-Reviews

Zum Beginn der Projektabschlussphase erfolgte eine detaillierte Durchführung eines Code- Reviews, zur Untersuchung der erstellten Gesamtcodes des Projektes. Zudem erstellte ich in diesen Vorgang zu deren Abschluss meinen Abschlussbericht des Projektes. Diese Ergebnisse habe ich unter [**A.19 Protokoll des Code-Review**](#_bookmark63)auf Seite **56** hinterlegt.

# Protokollierung der Softwarepräsentation des Teilprojektes

In der darauffolgenden Phase wurde eine Präsentation vor dem Auftraggeber des Gesamtprojektes und dem Koordinator durchgeführt. Die dadurch erlernten Erfahrungen sind von mir in einem Protokoll aufgenommen worden. Diese habe ich im Anhang unter [**A.18 Abschlussbericht des**](#_bookmark62)[**Projektes**](#_bookmark62)auf **Seite 55** hinterlegt.

## Qualitätssicherungsphase

# Protokollierte Abnahme des Projektes

Zum Abschluss der Abnahmephase erfolgte eine Erstellung des IHK-konformen Abnahmeprotokoll des Teilprojektes. Dieses habe ich unter [**A.20 Unterzeichnetes Abnahmeprotokoll**](#_bookmark64)auf **Seite 57** hinterlegt.

# Entwicklerdokumentation

Um andere Entwickler mit meiner Arbeit in der Softwareentwicklung vertraut zu machen erstellte ich während der Projektdokumentation mehrere Notizen, die ich in einer Entwicklerdokumentation festhielt. Einen Auszug mit dem Eigenanteil der Teilprojektdokumentation habe ich unter

* 1. [**Auszug aus der Entwicklerdokumentation**](#_bookmark65)auf **Seite 58** hinterlegt.

# Qualitätsmanagementbericht

Bei der Erstellung des Abschließenden Qualitätsmanagementberichtes erstellte ich ein Dokument mit verschiedenen wichtigen Protokollen der Meetings der Teilentwickler des Gesamtprojektes und dem Projektkoordinator. Diese Protokolle habe ich unter [**A.22 Abschließendes**](#_bookmark66)[**Qualitätssicherungskonzept mit Bericht**](#_bookmark66)auf **Seite 59-64** hinterlegt.

# Benutzerhandbuch

Um Benutzer beim Umgang mit meiner Teilprojektoberfläche und deren Funktionen zu unterstützen hat unser Team mit ihren jeweiligen Teilprojekten ein Benutzerhandbuch erstellt. Im Anhang sind daher unter [**A.23 Benutzerhandbuch**](#_bookmark67)auf **Seite 65-74** mein Anteil mit den betreffenden Funktionalitäten hinterlegt.

# Prozessorientierter Projektbericht

Aufgrund der Anforderungen der IHK wurden zum Abschluss der Qualitätssicherungsphase der Prozessorientierte Projektbericht erstellt indem alle Schritte meiner Arbeit festgehalten sind. Diesen habe ich nach dem Inhaltsverzeichnis in diesem Dokument auf den **Seite 1-15** hinterlegt. Im Zuge der Erstellung des Berichtes wurde ein dazugehörendes Glossar erstellt das im Anhang unter [**A.24**](#_bookmark68)[**Glossar des Gesamtprojektes**](#_bookmark68)auf **Seite 75-78** hinterlegt ist.

***6 Qualitätssicherungsphase***

**7 Projektbewertungsphase**

# Soll-/Ist-Vergleich

Abschließend wird die Planung des Projektes vom Entwickler abschließend bewertet. Dazu soll die benötigte mit der im Vorfeld in der Definitionsphase kalkulierten Zeit verglichen werden.

In Tabelle **4 (Soll-/Ist-Vergleich)** ist dem entsprechend eins Ergebnisses zu erkennen.

|  |  |  |  |
| --- | --- | --- | --- |
| **Projektphase** | **Geplante Zeit** | **Tatsächlich** | **Differenz** |
| **Analysephase** | 6h | 6h | 0h |
| **Definitionsphase** | 6h | 6h | 0h |
| **Entwurfsphase** | 16h | 14h | -2h |
| **Implementierungsphase** | 22h | 27h | +5h |
| **Projektabschlussphase** | 17h | 17h | 0h |
| **Gesamt** | 67h | 70h | +3h |

Tabelle 4: Soll-/Ist-Vergleich

Aus **Tabelle 4(Soll-/Ist-Vergleich)** lässt sich entnehmen das in der **Implementierungsphase** mehr Zeit als geplant war, verwendet werden musste.

Für die Einbringung und Testung der zu erstellenden Funktionen musste die zusätzliche Zeit verwendet werden.

In der **Entwurfsphase** benötigte der Ersteller 2 Stunden weniger, da er durch Erfahrung die Entwürfe sich besser visuell erarbeiten konnte.

Durch den geplanten Puffer von 3 Stunden konnte das Projekt erfolgreich in 70 Stunden, die der Vorgabe der IHK Köln entsprechen, erstellt werden.

Die vorgenommenen Meetings des Gesamtprojektes nahmen zudem auch Zeit in Anspruch, dies aber nur im leichten Umfang, so dass Sie in der Zeitplanung keinen Einfluss haben.

So konnte der Projektabschluss wie geplant stattfinden.

## 7 Projektbewertungsphase

# Gewonnene Erkenntnisse

Anhand der Umsetzung dieses Abschlussprojektes konnte ich wertvolle Erfahrungen über die Arbeit an einem Projekt mit allen dazugehörigen Arbeitsschritten sammeln. Vor allem in der Analysephase konnte mit dem Ansprechpartner neue Erkenntnisse gewonnen werden. Spannend war vor allem der Einsatz von verschiedenen Frameworks, um alle Komponenten und Funktionen der HTML/MySQL/PHP-Anwendung implementieren zu können. Zudem konnte der Autor auch in Bezug auf die Programmierung im Frontend Neues dazu lernen.

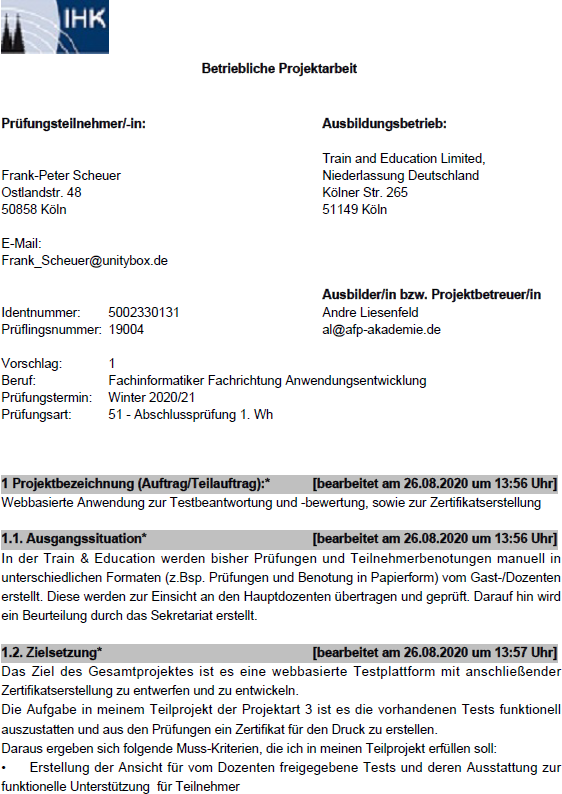
# Abweichung vom Projektantrag

Wie bereits erwähnt erfüllt das Testbewertungstool nicht die vollständige Funktionalität der Anzeigenlogik um die berechnete Note anzuzeigen. Aufgrund dessen können die Muss-Kriterien nicht vollständig eingebracht werden.

# Ausblick

Es besteht die Möglichkeit die Funktion der Testfertigstellung mit dem anderen Teilprojekten zu verbinden und dadurch die Tests den Teilnehmern aus einer vorhandenen Übersicht zuzusenden. Zudem kann man die in der Testfertigstellung erhalten Ergebnisse für den individuellen Teilnehmer zur Verfügung stellen, damit dieser wo seine Schwächen erfährt und mit Ihnen arbeiten kann.

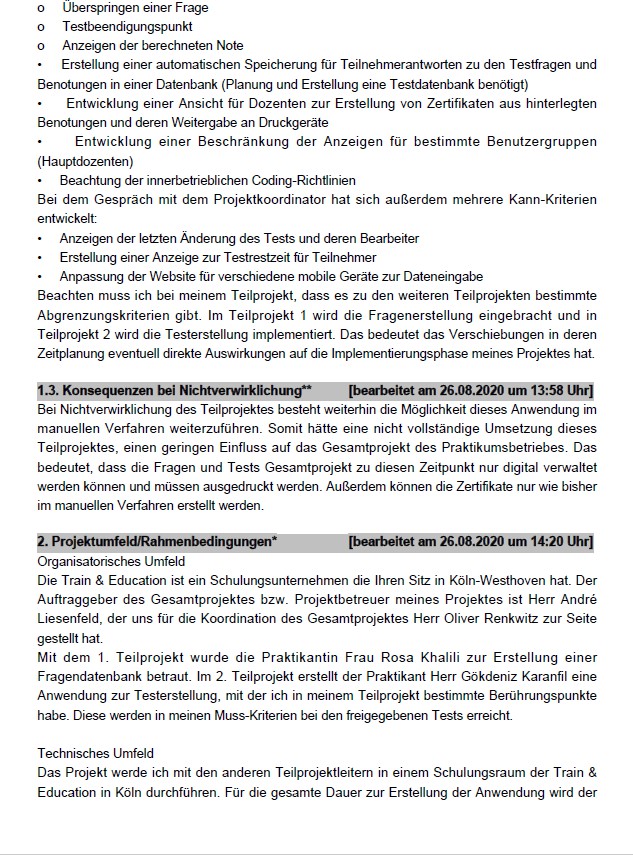
Diese können dann sehr einfach in der Anwendung untergebracht werden, da der Aufbau modular ist.



*A Anhang*

# Projektantrag (Seite 1)

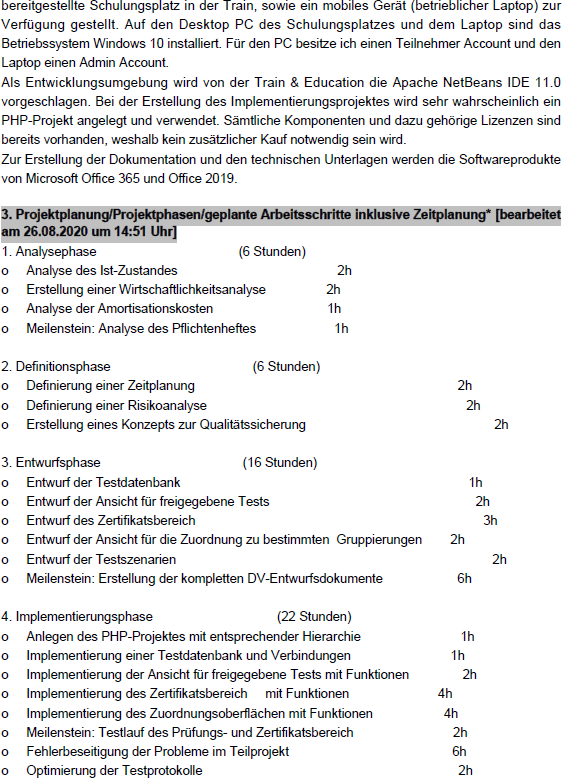
[**Einleitung**](#_bookmark1)



*A Anhang*

# Projektantrag (Seite 2)

[**Einleitung**](#_bookmark1)



*A Anhang*

# Projektantrag (Seite 3)

[**Einleitung**](#_bookmark1)

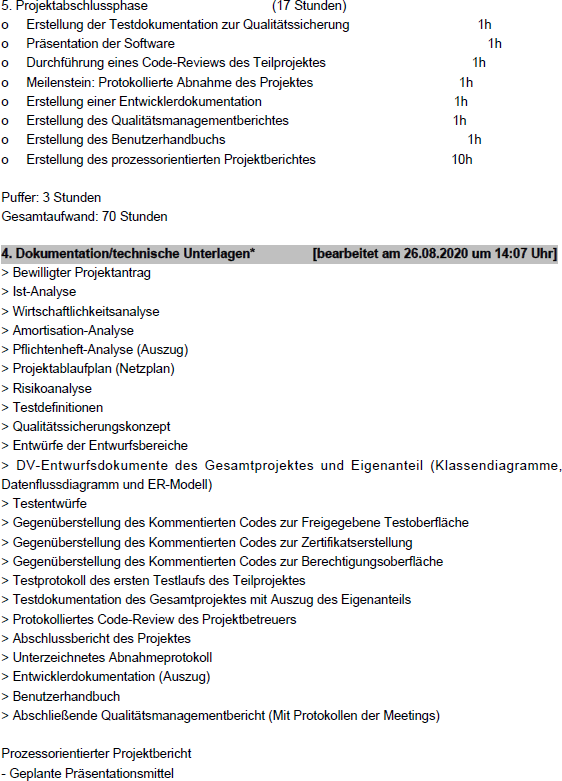
*A Anhang*



**A.1**

**Projektantrag (Seite 4)**

[**Einleitung**](#_bookmark1)



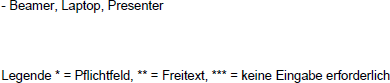
*A Anhang*



A.1

**Projektantrag (Seite 5)**

[**Einleitung**](#_bookmark1)



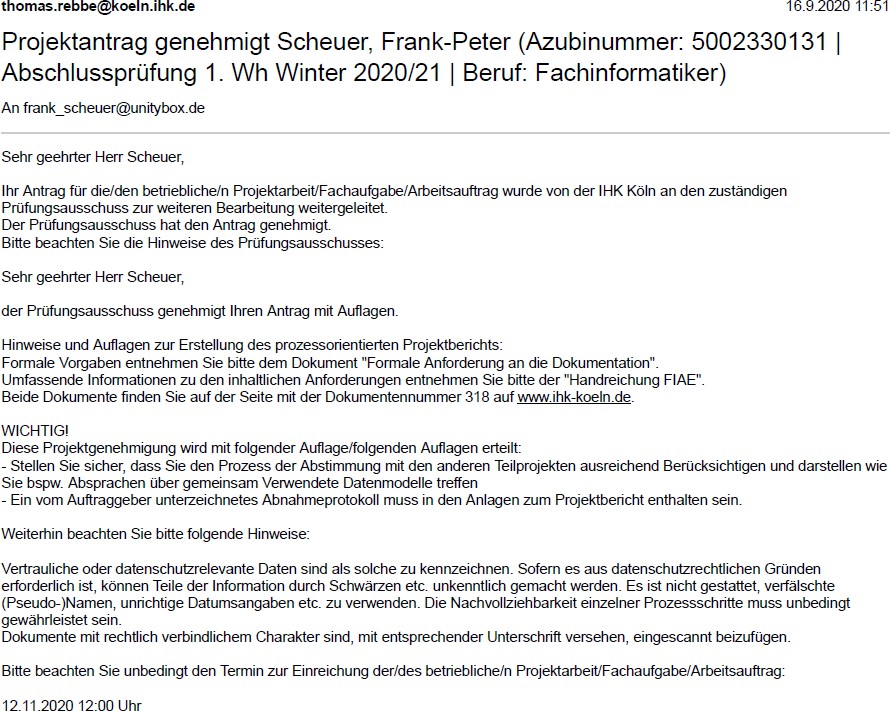
*A Anhang*

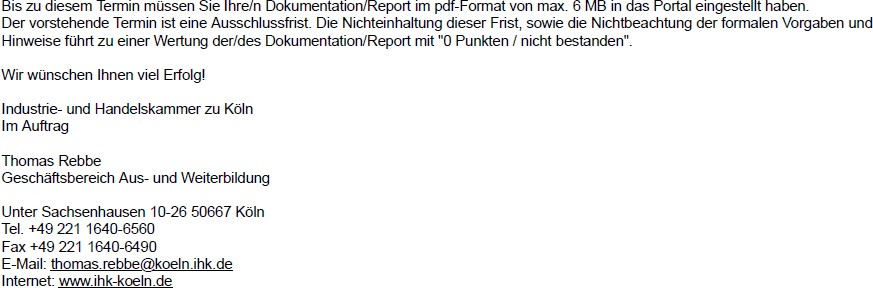


**A.1**

**Bewilligter Projektantrag**

[**Einleitung**](#_bookmark1)

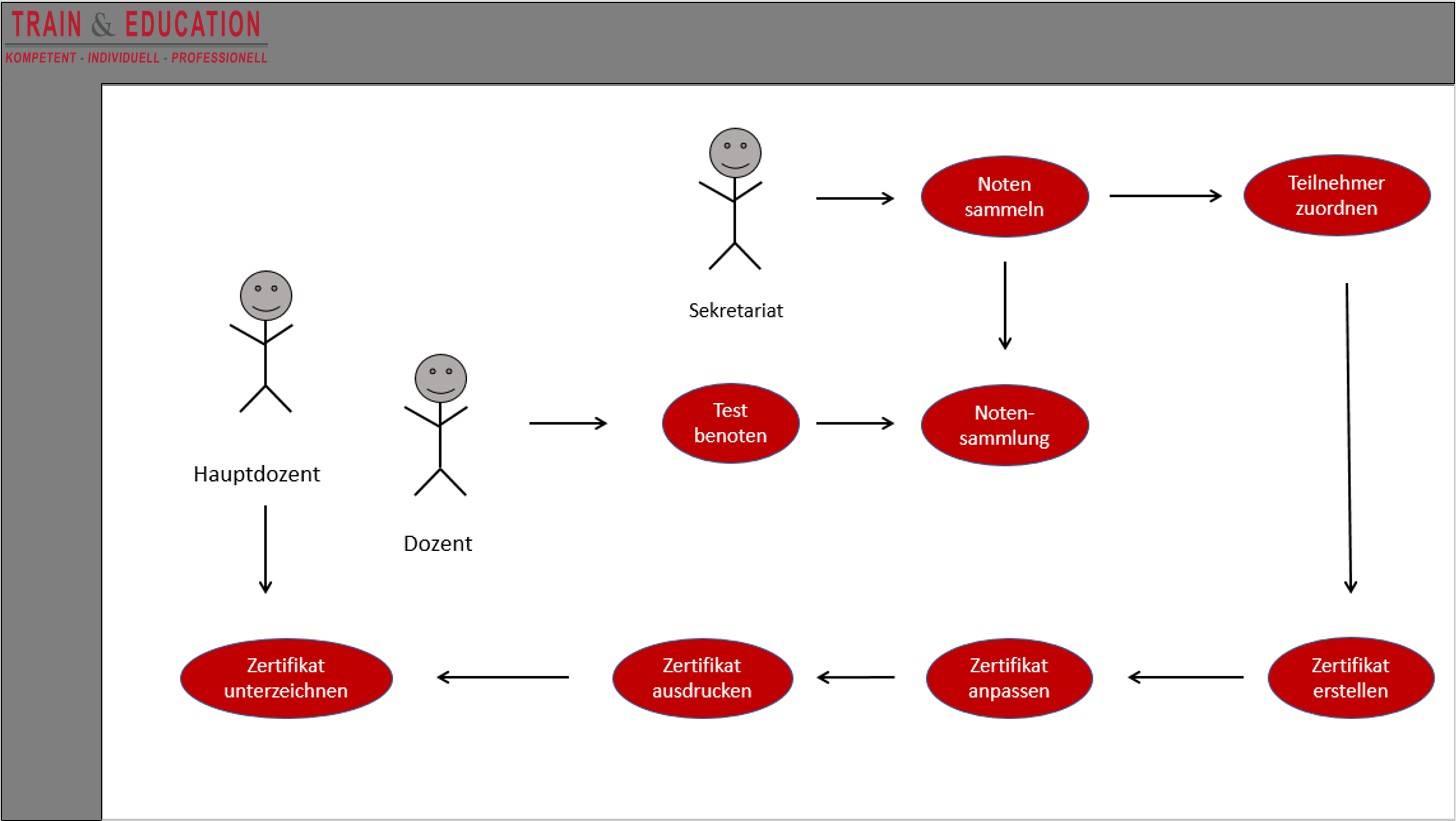




*A Anhang*

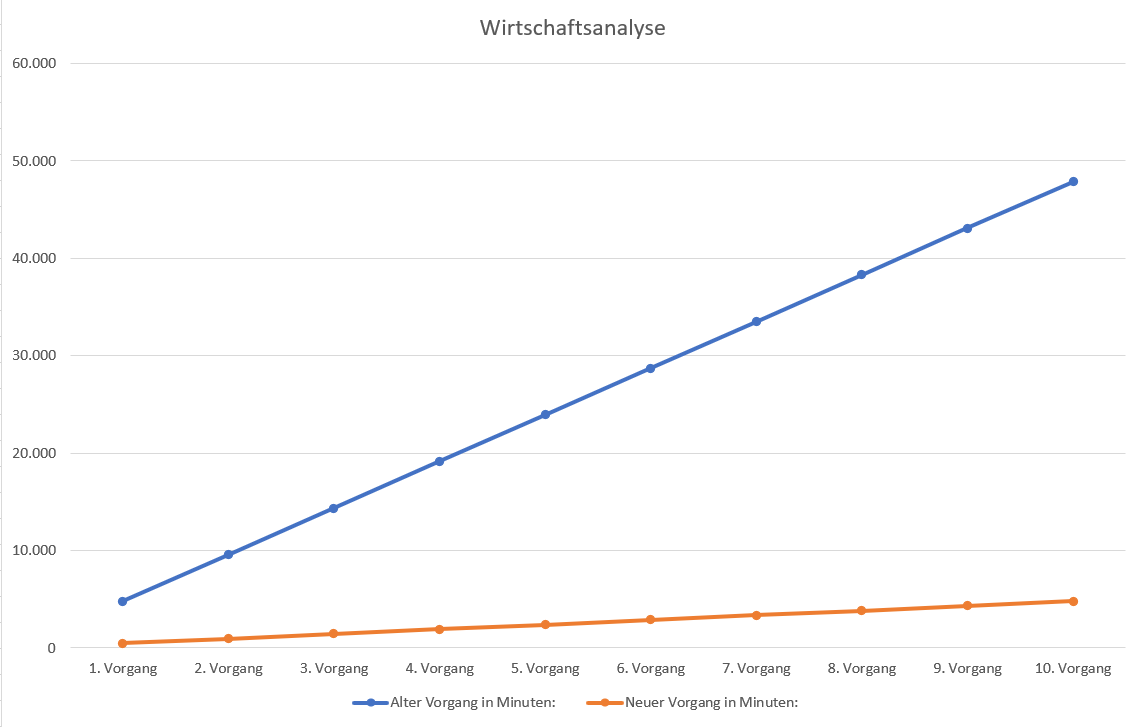
[**Analyse**](#_bookmark7)

# Strukturierte Ist-Analyse



In diesem Schaubild sieht man wie der momentane Ablauf der Testbenotung und Zertifikatserstellung ist.

*A Anhang*



[**Analyse**](#_bookmark10)

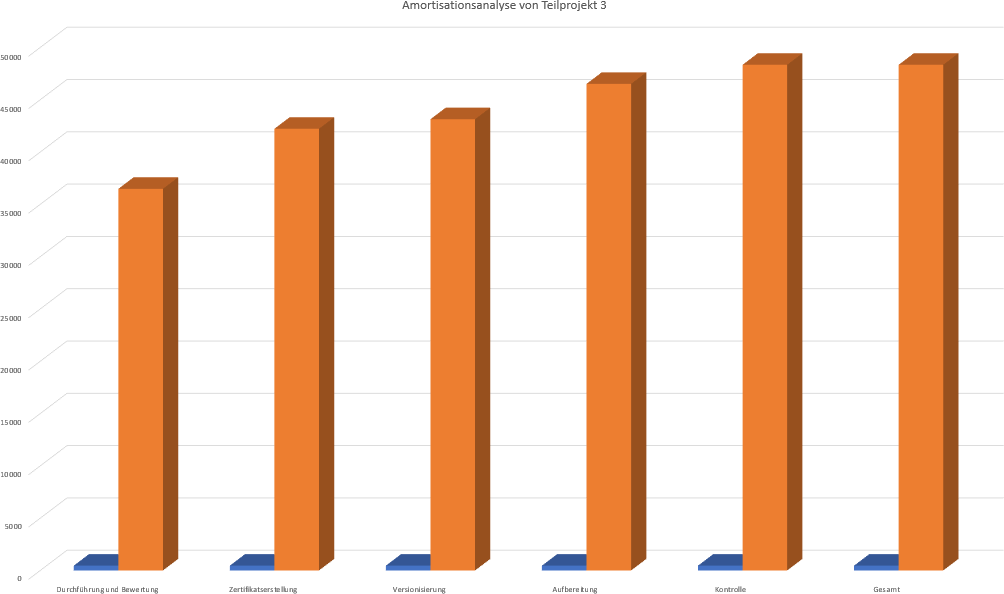
# Strukturierte Wirtschaftlichkeitsanalyse

Hier erkennt man ob sich eine Entwicklung befürworten lässt und dass die meiste Zeit in der Testprüfung und -bewertung benötigt wird, es ergibt ein zweiter beachtenswerter Punkt in der Zertifikatserstellung.

*A Anhang*

[**Analyse**](#_bookmark12)

# Strukturierte Amortisationsanalyse



In dieser Grafik sieht man im Orange Balken die Zeit die bisher benötigt wird für die Anforderung am Gesamtprojekt im Vergleich zu dem im Blauen Balken dargestellten nach der Fertigstellung der Anwendung benötigten Zeit.

*A Anhang*

[**Analyse**](#_bookmark13)

# Strukturierte Pflichtenheftanalyse (Seite 1)

1. **Ausgangssituation und Zielsetzung**
   1. Ausgangssituation
   2. Zielsetzung
2. **Funktionale Anforderungen**
   1. Softwarearchitektur, Leistungen der Komponenten
   2. Anwendungsfälle, Use Cases
   3. Datenmodell
   4. Abgrenzungskriterien
3. **Nicht-funktionale Anforderungen**
   1. Einsatzumgebung Hardware
   2. Einsatzumgebung Software
   3. Einsatzumgebung Orgware
   4. Leistungsparameter Datenvolumen
   5. Leistungsbereich Benutzerführung
   6. Leistungsparameter Verarbeitungsgeschwindigkeit
4. **Lebenszyklus des Gesamtsystems**
5. **Schnittstellenübersicht**
6. **Lieferumfang**
   1. Zu übergebende Dokumente
   2. Software-Bereitstellung
   3. Installation und Datentransfer

*A Anhang*

# A.5 Strukturierte Pflichtenheftanalyse (Seite 2)

[**Analyse**](#_bookmark13)

* 1. Schulungsleistungen
  2. Supportleistungen

# Abnahmekriterien

* 1. Qualitätsziele
  2. Testszenarien

# Ausgangssituation und Zielsetzung

* 1. **Ausgangssituation**

Anlass zur Durchführung des Teilprojektes ist ein Auftrag zur Erstellung einer Anwendung im Auftrag des Unternehmens der Akademie für Fortbildung am Standort Köln. Die Arbeit bei der Erstellung der Teilnehmerbenotung soll durch die Anwendung automatisiert und erleichtert werden. Bisher erfolgt dieses Verfahren der Erstellung per einer Dokumentation Via Excel und der weiterführenden manuellen Arbeit.

Das Projekt soll die automatische Errechnung und Teilnehmerbezogene Speicherung von Testnoten mit anschließender Möglichkeit zum Papier- oder PDF-Druck.

In einem ersten Meeting vor der Aufteilung auf die Teilprojekte wurde beschlossen die Software auf Basis der Programmiersprache PHP mit einer bestehenden MySQL Datenbank zu erstellen. Bei diesem Treffen wurde ebenfalls beschlossen die gesamte Anwendung mit dem Framework Laravel zu implementieren. Diese Entscheidung bilden die Ausgangssituation dieses Entwicklerprojektes.

Während der Erstellung werden die Muss-Kriterien aus dem Pflichtenheft des Projektkoordinators analysiert und in das Pflichtenheft des Teilprojektes übernommen.

Spezielle Software soll nicht installiert, sondern die Anwendung individuell und gegebenenfalls anpassbar erstellt werden.

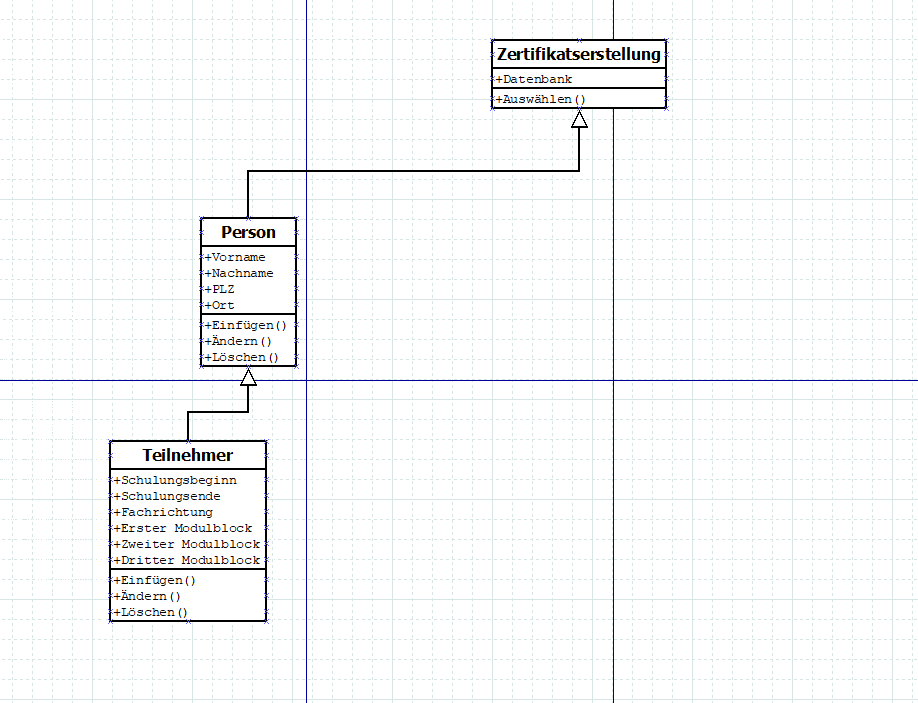
*A Anhang*

# A.5 Strukturierte Pflichtenheftanalyse (Seite 3)

[**Analyse**](#_bookmark13)

* 1. **Zielbestimmung**

Ziel des Projektes ist die Entwicklung einer Anwendung zur Bewertung von Tests und anschließender Zertifikatsausgabe. In erster Linie muss die Software leicht bedienbar, anpassbar



und erweiterbar und deren Programmieraufbau nachvollziehbar sein.

Der **Ort für die Ausführung** der Softwarekomponenten soll erstmal gebunden an einen Server der Train & Education Köln sein. Änderungen der Software müssen jederzeit möglich sein. Die **Kommunikation** der Softwarekomponenten soll über standardisierte Netzwerkprotokolle stattfinden.

*A Anhang*

[**Analyse**](#_bookmark13)

A.5 **Strukturierte Pflichtenheftanalyse (Seite 4)**

1. **Funktionale Anforderungen**
   1. **Softwarearchitektur, Leistungen der Komponenten**

Die Software setzt sich aus zwei Komponenten zusammen:

* + - PHP als Programmiersprache zum Aufbau der Datenbank
    - MySQL als Client Oberfläche für die erstellte Datenbank

PHP wurde im ersten Meeting als am idealsten zur Erstellung erschlossen. Im weiteren Verlauf wurde sich für das Laravel Framework entschieden.

Der MySQL Client soll die Testdaten auslesen und in die Datenbank einfügen.

Der Power Shell Client wurde vom Ansprechpartner gewünscht, da MySQL die geforderte Aufgabe nicht ausführen kann.

Dem Pflichtenheft des Gesamtprojektes sind folgende Punkte zu entnehmen.

1. Die Software muss freigegebene Tests anzeigen und die Funktionen **Fragen überspringen**, **Testtimer setzen** und **Benotung erzeugen**.
2. Die Software muss im **Internet** über eine **browserabhängige Benutzeroberfläche** mit dynamischen Objektgrößen erfolgen, dies soll auf **Stationären Geräten** möglich sein.
3. Die Softwareoberfläche soll die **Testausstattung** ermöglichen und diese in einer Datenbank abspeichern. Dabei sollen die Funktionen **Fragen überspringen**, **Testtimer setzen** und **Benotung** eingebracht werden
4. Die Softwareoberfläche soll die **Testbewertung** ermöglichen und diese in einer Datenbank abspeichern.
5. Bei einer **Testbewertung** muss die Verfügbarkeit der ***Teilnehmer*** im bisherigen Datenbestand überprüft werden. Ist dieser nicht vorhanden, wird ein neuer erstellt. Dies soll über ein eigenes Formular erstellt werden.
6. Die Softwareoberfläche soll die **Zertifikatserstellung** ermöglichen und diese anschließend auszudrucken.
7. Die Softwareoberfläche soll die **Zertifikatsbeschränkung** ermöglichen und dadurch Ansichten zu beschränken.
8. Die Software muss eine Weitergabe von Daten an ein angeschlossenes **Druckgerät**

ermöglichen.

1. Die Software muss eine **Kontinuierliche Integration** bereitstellen.
2. Die Software muss die **innerbetrieblichen Coding-Richtlinien** einhalten.
3. Dem System werden aufgrund der anderen Teilprojekte eine **Schnittstelle** in der Testübersicht zur Verfügung gestellt.

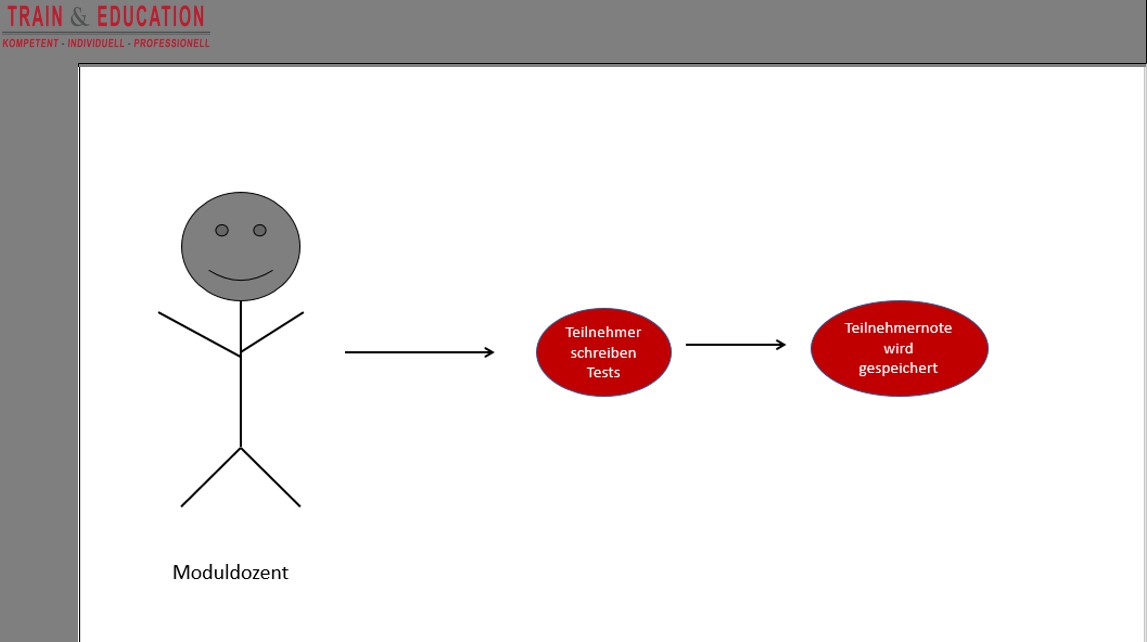
*A Anhang*

[**Analyse**](#_bookmark13)

A.5 **Strukturierte Pflichtenheftanalyse (Seite 5)** Aufgrund der Auftragsanforderungen sind folgenden Kann-Kriterien für dieses Teilprojekt entstanden.

1. Die Software soll bei einer **Änderung von Tests den Änderungszeitpunkt** und den Bearbeiter hinterlegen.
2. Der Testtimer soll eine **Anzeige der Testrestzeit** für digitale Teilnehmer ermöglichen.
3. Die Software der **browserabhängige Benutzeroberfläche** kann mit Mobilen Geräten bearbeitet werden.

# Anwendungsfälle (Use Cases)



*A Anhang*

**Nummer Use Case 1**

[**Analyse**](#_bookmark13)

# A.5 Strukturierte Pflichtenheftanalyse (Seite 6)

|  |  |
| --- | --- |
| **Kurzbeschreibung** | Anmelden in der Datenbank |
| **Akteure** | * Dozent der Train & Education |
| **Vorbedingungen** | Verbindung der Rechner damit eine zentrale Stelle erstellt werden kann |
| **Essenzieller Ablauf** | 1. Der Benutzer öffnet die Datenbank „Testplattform“. 2. Das System lädt das Formular „Schulungslogin“. 3. Der Benutzer klickt auf die Schaltfläche „Testausstattung und Benotung“. 4. Der Benutzer lässt einen Test von den Teilnehmern schreiben. 5. Der Benutzer lässt den Test benoten und die Daten daraufhin speichern. 6. Das System schließt die Speicherung und öffnet die Schaltfläche   „Testausstattung und Benotung“. |
| **Nachbedingung** |  |
| **Ergebnis** | Der Benutzer hat einen Teilnehmer bewertet und die Daten gespeichert. |

* 1. **Datenmodell**

Das Datenmodell orientiert sich an den Vorgaben des Schulungsmodulsaufbau der Schulungen im Unternehmen.

1. Tests Enthält die Testdaten des Modulumfelds mit seinen Fragen.
2. Teilnehmer Enthält die persönlichen Daten des Teilnehmers mit seinem/seinen Status.

# Abgrenzungskriterien

Es gehen folgende Abgrenzungskriterien gehen aus dem Pflichtenheftmeeting hervor.

* + - Es sind Maßnahmen zur **Datensicherung** in einer bestehenden MySQL Datenbank zutreffen.
    - Es sind Maßnahmen zur **Datenarchivierung** in einer bestehenden MySQL Datenbank zutreffen.
    - Es sind Maßnahmen zur **Zugriffssicherheit** über die bestehende Benutzerverwaltung zutreffen.

*A Anhang*

[**Analyse**](#_bookmark13)

A.5 **Strukturierte Pflichtenheftanalyse (Seite 7)**

1. **Nicht-funktionale Anforderungen**
   1. **Einsatzumgebung Hardware**

Die Software soll lauffähig auf den **vorhandenen Hardwaresystem** in den Schulungsräumen der Train & Education sein. Zusätzliche Investitionen für Hardware werden ausgeschlossen.

# Einsatzumgebung Software

Die Software soll lauffähig auf den **vorhandenen Betriebssystemen** ab Windows 10 in den Schulungsräumen der Train & Education. Zusätzliche Investitionen für Software werden ausgeschlossen. Es ist die Software des Entwicklers zu verwenden. Gearbeitet wird mit folgenden Produkten:

* + - MySQL/MariaDB als Datenbankmanagementsystem
    - Visual Studio Code als Entwicklungsumgebung
    - PHP als Programmiersprache
    - CSS und JavaScript als Skriptsprachen
    - Laravel als unterstützendes Framework
    - Apache als Server zum Betrieb im Netzwerk

# Einsatzumgebung Orgware

Der Betrieb der Software setzt möglichst **keine zusätzlichen organisatorischen Regelungen**

voraus.

# Leistungsparameter Datenvolumen

Laut dem vorhandenen Pflichtenheft erstellt der Entwickler eine dynamische Funktion zur Erstellung der Einträge für das Datenbanksystem erstellen.

Pro Fachrichtung werden bei der AFP 10-20 Teilnehmer ausgebildet. Dozenten sind in der Aufzählung nicht mit aufzunehmen. Die Größe des Datenbanksystems sind laut Annahme 80 Teilnehmer, wenn man von vollbesetzen Klassen für 2 Jahre ausgeht. Da die Daten aber bestenfalls archiviert werden sollen ist von mehr Datenplatz auszugehen.

Der Speicherbedarf der Gesamtsoftware wird nach Fertigstellung dargestellt und gegebenfalls an dieser Stelle eingefügt.

*A Anhang*

[**Analyse**](#_bookmark13)

# A.5 Strukturierte Pflichtenheftanalyse (Seite 8)

* 1. **Leistungsbereich Benutzerführung**

Aufgrund der erstellten Anforderungen und im Meeting besprochenen Informationen geht der Entwickler von den hinterlegten Wünschen zum Design des Managementsystems der Datenbanken aus, um diese zu entwerfen und implementieren.

# Leistungsparameter Verarbeitungsgeschwindigkeit

Aufgrund der Erfahrungen meiner belegte Annahme für die **Antwortzeit** von Datenbankabfragen kann ich diese erstellen. Der Entwickler legt eine **Antwortzeit** von unter 5 Sekunden an.

# Lebenszyklus des Gesamtsystems

Aufgrund der Anforderungen der AFP Köln geht dieser von einer Annahme für die Anforderungen des Lebenszyklus aus.

# Zeitpunkt des Einsatzes

Das Datenbankmanagementsystem soll zum Ende des Praktikums am 23.10.2020 fertiggestellt und einsatzbereit sein.

# Einsatzumgebung Software

Es gilt bezüglich der Datenbank-Funktionalität das **vorhandene System der Excelspeicherung**

abzulösen.

# Zu erwartende wachsende Anforderungen

Die Anzahl der Teilnehmer und deren Fachrichtungen können sich langfristig ausdehnen. Aufgrund dessen muss die Anzahl der Benotungen und hinterlegten PDF-Zertifikate anpassbar und erweiterbar sein.

*A Anhang*

[**Analyse**](#_bookmark13)

A.5 **Strukturierte Pflichtenheftanalyse (Seite 9)**

1. **Schnittstellenübersicht**

|  |  |
| --- | --- |
| **Schnittstelle** | **Kommunikationsstandard** |
| Application – Datenbank | JDBC mit eingebetteten SQL |
| Datenbank – Client | Selbsterstellt Anwendung auf Basis von PHP |

1. **Lieferumfang**
   1. **Zu übergebende Dokumente**
      * Systemdokumentation
      * Benutzerhandbuch für den Betrieb in der Train & Education
      * Administrationshinweise

# Software-Bereitstellung

Die gesamte Software soll mit der Webanwendung zur Versionsverwaltung für Softwareprojekte GitLab des Gesamtprojektes übergeben werden. Die clientseitigen Zugriffskomponenten sollen über die GitLab Anwendung als Download aus einem geschützten Bereich angeboten und dort gegebenfalls auch aktualisiert werden. Die Bereitstellungsdatei kann zudem per Arbeitsgruppe des Entwicklersystems angeboten und den Auftraggeber zur Verfügung gestellt werden. Dies erfolgt, um einen hohen Sicherheitsstandard zu gewähren.

# Installation und Datentransfer

Die Software ist für Administratoren in einer komprimierten Form bereitzustellen, die sich selbst entpackt und selbstständig die Installation startet. Ein Datentransfer aus Vorgängerlösungen ist nicht notwendig.

# Schulungsleistungen

Entfallen

# Supportleistungen

Der Support wird durch die **Mitarbeiter der Train & Education** gewährleistet. Während seiner Umschulung steht das Entwicklerteam für Fragen zur Verfügung.

*A Anhang*

[**Analyse**](#_bookmark13)

* 1. **Strukturierte Pflichtenheftanalyse (Seite 10)**

1. **Abnahmekriterien**
   1. **Qualitätsziele**

Aufgrund des vorhandenen Pflichtenheftes muss der Entwickler des Teilprojektes die Ziele der Funktionalität, Zuverlässigkeit, Benutzbarkeit, Effizienz, Änderbarkeit und Übertragbarkeit der Software/Anwendung beachten.

# Testszenarien

Aufgrund des im vorhandenen Pflichtenheftes erstellten Vorgabe der Qualitätskontrolle kann der Entwickler sechs Einträge Testszenarien erstellen.

|  |  |
| --- | --- |
| **Testfall:** | **1/ „Freigegebenen Test ausstatten“** |
| **Anforderungen an das System** | Das Anmelden an das System erfolgt als Dozent, der Zugang existiert bereits im System der Benutzerverwaltung, die  Eingabe von Benutzername und Passwort ist korrekt. |
| **Prüfkriterien** | Der Moduldozent erhält Zugang zur **Ansicht der Testausstattung** mit der funktionellen Unterstützung der  bereitgestellten Tests. |
| **Prüfmethode** | Funktionsüberprüfung (Ob diese Funktionieren) |

|  |  |
| --- | --- |
| **Testfall:** | **2/ „Teilnehmerantworten speichern“** |
| **Anforderungen an das System** | Das Anmelden an das System erfolgt als Dozent, der Zugang  existiert bereits im System der Benutzerverwaltung, die Eingabe von Benutzername und Passwort ist korrekt. |
| **Prüfkriterien** | Der Benutzer erhält Zugang zur **Ansicht der Testantworten**  und kann die mit dem Teilnehmerdaten als Ausdruck speichern z.Bsp. „Ergebnis: 1.a“. |
| **Prüfmethode** | Funktionsüberprüfung |

*A Anhang*

|  |  |  |
| --- | --- | --- |
|  | |  |
| A.5 **Strukturierte Pflichtenheftanalyse (Seite 11)** | | [**Analyse**](#_bookmark13) |
| **Testfall:** | **3/ „Teilnehmerbenotung speichern“** | |
| **Anforderungen an das System** | Das Anmelden an das System erfolgt als Dozent, der Zugang existiert bereits im System der Benutzerverwaltung, die  Eingabe von Benutzername und Passwort ist korrekt. | |
| **Prüfkriterien** | Der Benutzer erhält Zugang zur **Ansicht der Testantworten** und kann diese mit einer Funktion bewerten lassen. Diese kann er mit den Teilnehmerdaten den berechneten Ergebnis  speichern z.Bsp. „Note: 1“. | |
| **Prüfmethode** | Funktionsüberprüfung |  |

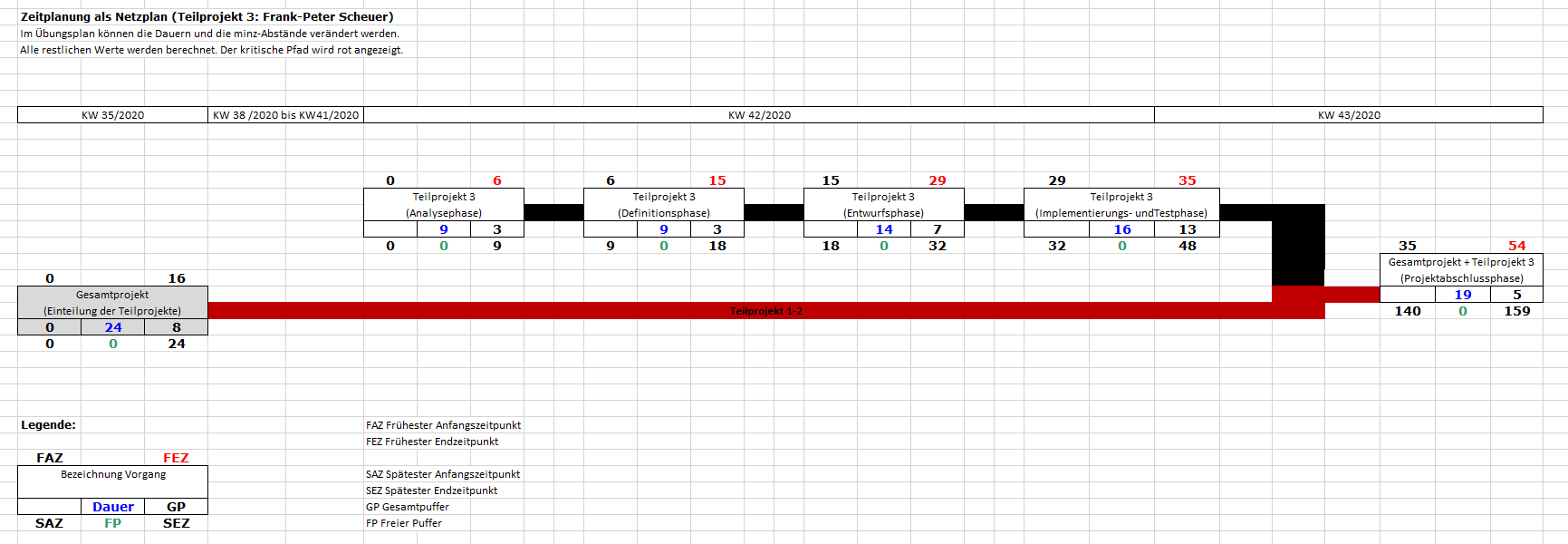
|  |  |
| --- | --- |
| **Testfall:** | **4/ „Teilnehmerzertifikat erstellen“** |
| **Anforderungen an das System** | Das Anmelden an das System erfolgt als Dozent, der Zugang existiert bereits im System der Benutzerverwaltung, die  Eingabe von Benutzername und Passwort ist korrekt. |
| **Prüfkriterien** | Der Benutzer erhält Zugang zur **Ansicht der Zertifikatserstellung** und kann per Funktionen diesen mit dem  hinterlegten Formular und den Teilnehmerdaten ausstatten. |
| **Prüfmethode** | Funktionsüberprüfung |

|  |  |
| --- | --- |
| **Testfall:** | **5/ „Zertifikatsansichtsbeschränkung“** |
| **Anforderungen an das System** | Das Anmelden an das System erfolgt als Dozent oder Sekretariats-mitarbeiter, der Zugang existiert bereits im System, die Eingabe von Benutzername und Passwort ist  korrekt. |
| **Prüfkriterien** | Der Benutzer erhält Zugang zur **Ansicht der Zertifikate** und kann per Funktionen diese für bestimmte Benutzergruppen in  Verbindung mit der Benutzerverwaltung berechtigen. |
| **Prüfmethode** | Funktionsüberprüfung |

*A Anhang*

[**Definition**](#_bookmark14)

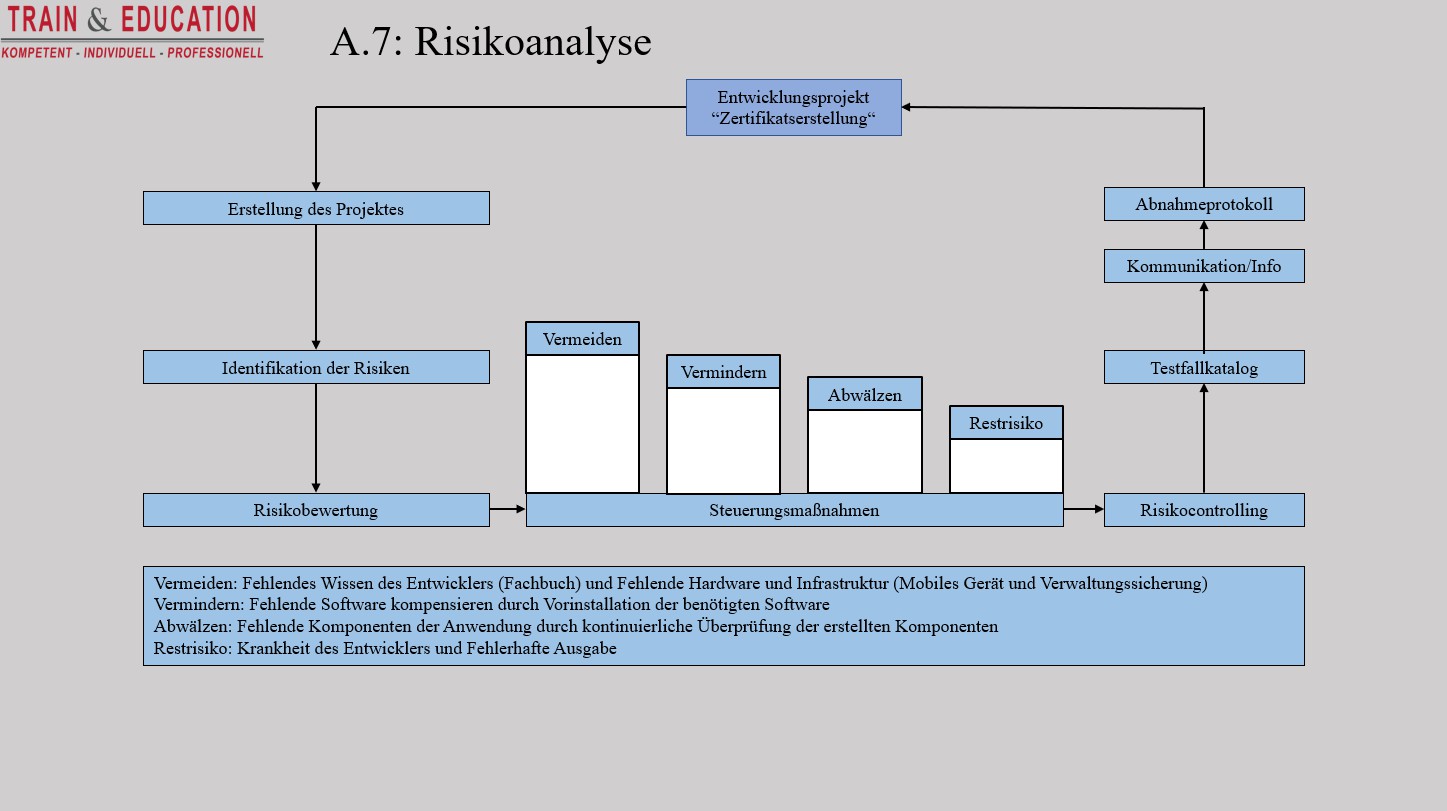
# Projektablaufplan (Netzplan)



*A Anhang*

[**Definition**](#_bookmark15)

# Risikoanalyse der Zertifikatserstellung



*A Anhang*

[**Definition**](#_bookmark16)

# Qualitätssicherungsnormen (Seite 1)

* 1. **Einleitung des Normenkatalogs**

Aufgrund der Kriterien des Auftraggebers, sowie der weiteren Qualitätsmerkmale der **IT** in Formen von Leitlinien die zu diesem Projekt passen, kommt es zur Erstellung einer entsprechenden Sammlung. Dieser liegt in verschiedenen Normen zugrunde und stellt die Softwarequalität sicher so dass diese eingehalten und befolgt werden kann. Im Folgenden werde ich die verwendeten Normen spezifizieren und auf das Projekt in den Testszenarien ableiten.

# Verwendete Normen für die Softwarequalität

Zusätzlich zu den ISO 9000-Reihe verwendete der Entwickler weitere Normen zu Beachtung der Qualitätsmerkmale. Im Testfall Katalog werden diese Normen und deren Inhalte genauer detailliert und im Vergleich im Fazit auf das Projekt projiziert.

# ISO-9000-Standards zur Erfüllung der Anforderungen im Qualitätsmanagement

* + - * **ISO/IEC** 9000/9001 definiert die Einhaltung der Vertragsinhalte des Entwicklers bei der Erstellung des Designs, der zu erstellenden Dokumente, Prüfung durch etwaige Ansprechpartner und Fehlervorbeugung. Des Weiteren werden die „Normen zum QM und zur QS -Teil 3- Leitfaden für die Anwendung von ISO 9001 auf die Entwicklung, Lieferung und Wartung von Software“ beachtet.
      * **ISO/IEC** 9126 definiert die Einhaltung der Qualitätsmerkmale bei der Erstellung der Softwareprodukten. Sie bezieht sich dabei mit ihren Kriterien ausschließlich auf die Qualität des Produktes (Produktqualität).
      * **ISO/IEC** 25000 definiert die Qualitäts- und Bewertungskriterien bei der Erstellung der Softwareprodukten. Sie bezieht sich mit ihren Kriterien ausschließlich auf die Bewertung des Produktes (Produktbewertung).

# ANSI/IEEE-Standards zur Dokumentation von Prozess und Produkt

* + - * **ANSI/IEEE**-Std 828-1983 (Software Configuration Management Plans) liefert konkrete Verfahrensnormen zur Verwaltung von Softwarekonfigurationen
      * **ANSI/IEEE**-Std 829-1983 (Software Test Documentation) nennt Forderungen an den Aufbau und Inhalt von Testdokumenten
      * **ANSI/IEEE**-Std 830-1984 (Software Requirements Specification) nennt einen Leitfaden zur methodischen Erstellung von Softwareanforderungen
      * **ANSI/IEEE**-Std 730-1984 (Software Quality Assurance Plans) definiert Anforderungen an die Verantwortlichkeiten sowie eine Gliederung zum Inhalt eines Qualitätssicherungsplans
      * **ANSI/IEEE**-Std 1012-1986 (Software Verification and Validation Plans) liefert eine Systematik zur Durchführung von Aktivitäten zur Verifizierung und Validierung im gesamten Softwarelebenszyklus
      * **ANSI/IEEE**-Std 1058.1-1987 (Standard for Software Project Management Plans) definiert Forderungen an Aufbau und Inhalt des Projektplans.

*A Anhang*

[**Definition**](#_bookmark16)

# Qualitätssicherungsnormen (Seite 2)

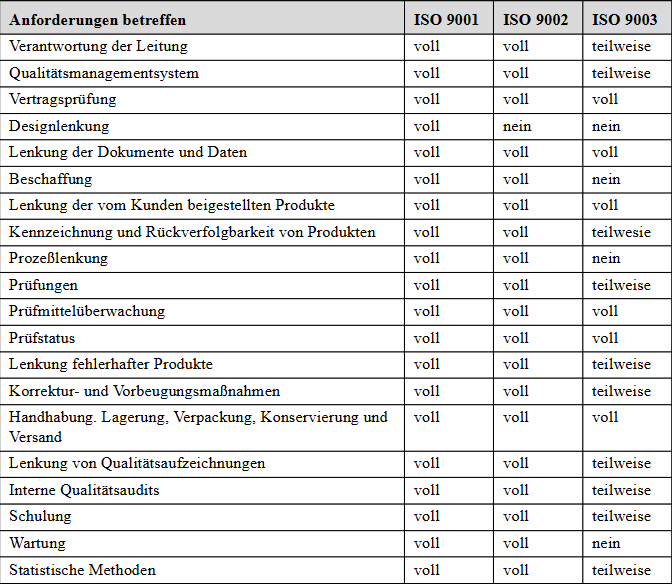
* + 1. **DIN-Standards zur Erfüllung der Gütebedingungen und Prüfbestimmungen**
       - **DIN** 66272 definiert die Einhaltung der Standards für Prüfbedingungen des Projektes. Die Qualität eines Softwareprodukts wird als gut bezeichnet, wenn es die beschriebenen Eigenschaften in solchem Maße aufweist, dass es damit den Erfordernissen seiner Anwendung gerecht wird. Mit dieser Norm wurde gewissermaßen Abschied von der(unrealistischen) Forderung genommen, Softwarequalität nur dann beurteilen zu können, wenn quantitative Messgrößen für die einzelnen Merkmale vorhanden sind. „Der Stand der Technik gestattet es noch nicht, quantitative Messgrößen für alle Merkmale zu normen. Anforderungen und Beschreibungen können auch qualitativ sein.“
       - **DIN** 66285 definiert die Einhaltung der Mindeststandards für Gütebedingungen des Projektes. Die Gütebedingungen betreffen dabei die Produktbeschreibung, die Produktdokumentation sowie Programme und Daten. Die Norm richtet sich somit an das fertige Softwareprodukt und enthält keinerlei Vorschriften für den Erstellungsprozess. Als Anwendungsbereich der DIN 66 285 werden ausschließlich Anwendungssoftware und Dienstprogramme (z.B. für Datenverwaltung) genannt. Software für Sicherheitstechnik oder hardware-integrierte Steuerungssoftware (sog. "embedded software") sind von der Norm ausdrücklich ausgenommen. Die Güteprüfung von Software nach DIN 66 285 ist eine typische Produktprüfung. Es wird lediglich das Endprodukt, nicht jedoch der Entwicklungsprozess geprüft. Dies ist aus Sicht des Anwenders sicherlich völlig ausreichend, da er nur an der Qualität des Endprodukts interessiert ist. Für das Qualitätsmanagement reicht die Norm jedoch nicht aus, da sie nicht während des Entwicklungsprozesses eingesetzt werden kann. Die Norm bietet keine inhaltlich relevanten Informationen für qualitätsbezogene Managemententscheidungen im Prozessablauf.

*A Anhang*

[**Definition**](#_bookmark16)

# A.8 Qualitätssicherungsnormen (Seite 3)

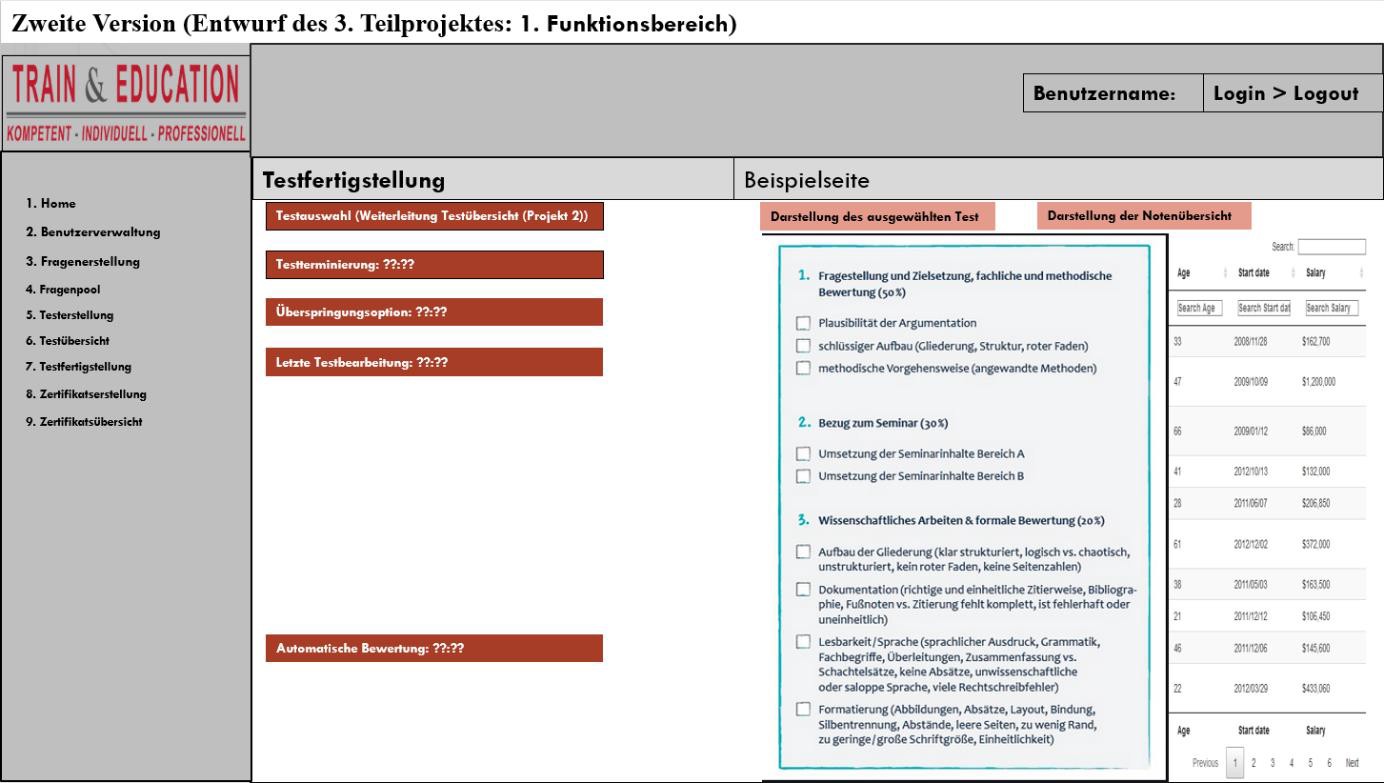
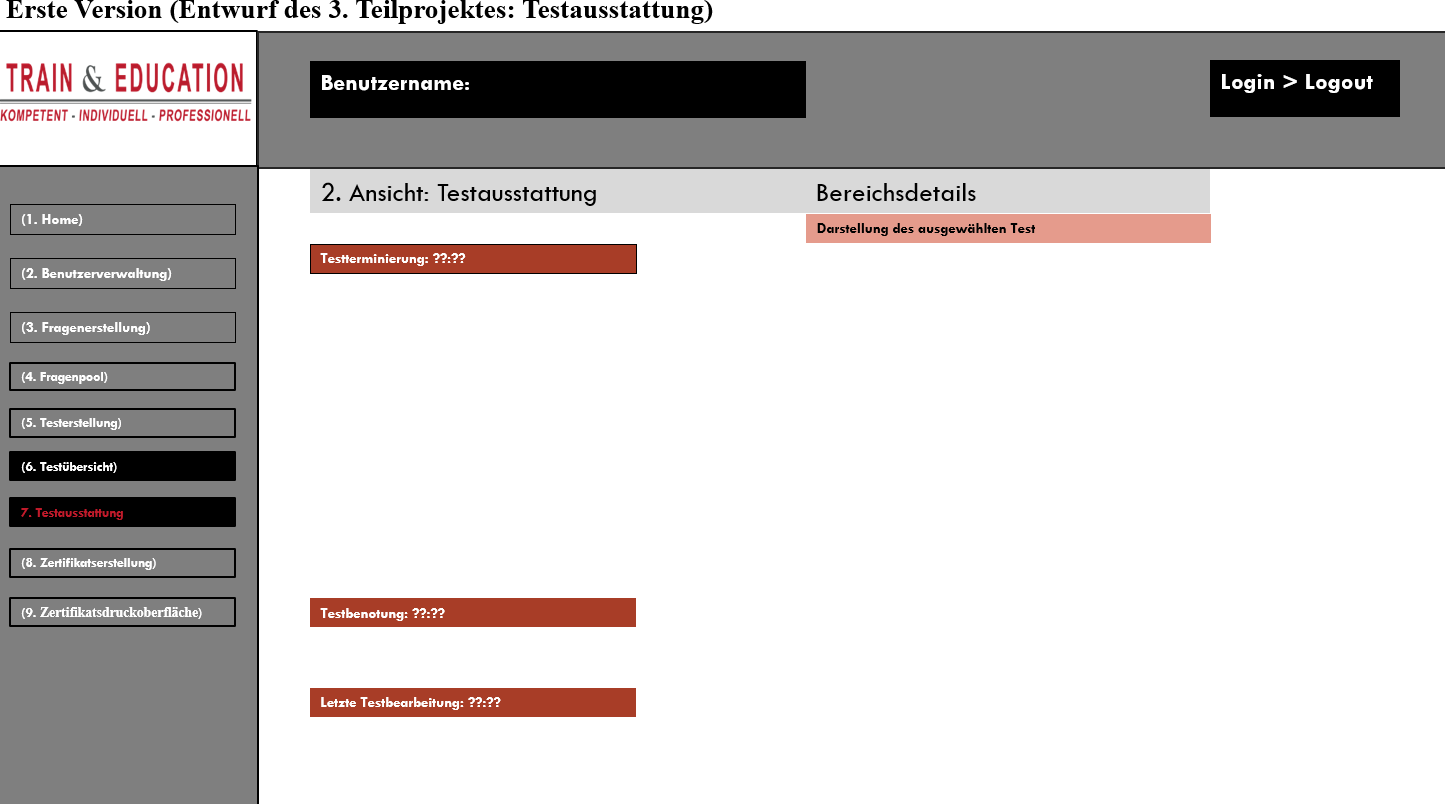
* 1. **Vergleich von DIN EN ISO 9001, 9002 und 9003**



*A Anhang*

[**Entwurf**](#_bookmark18)

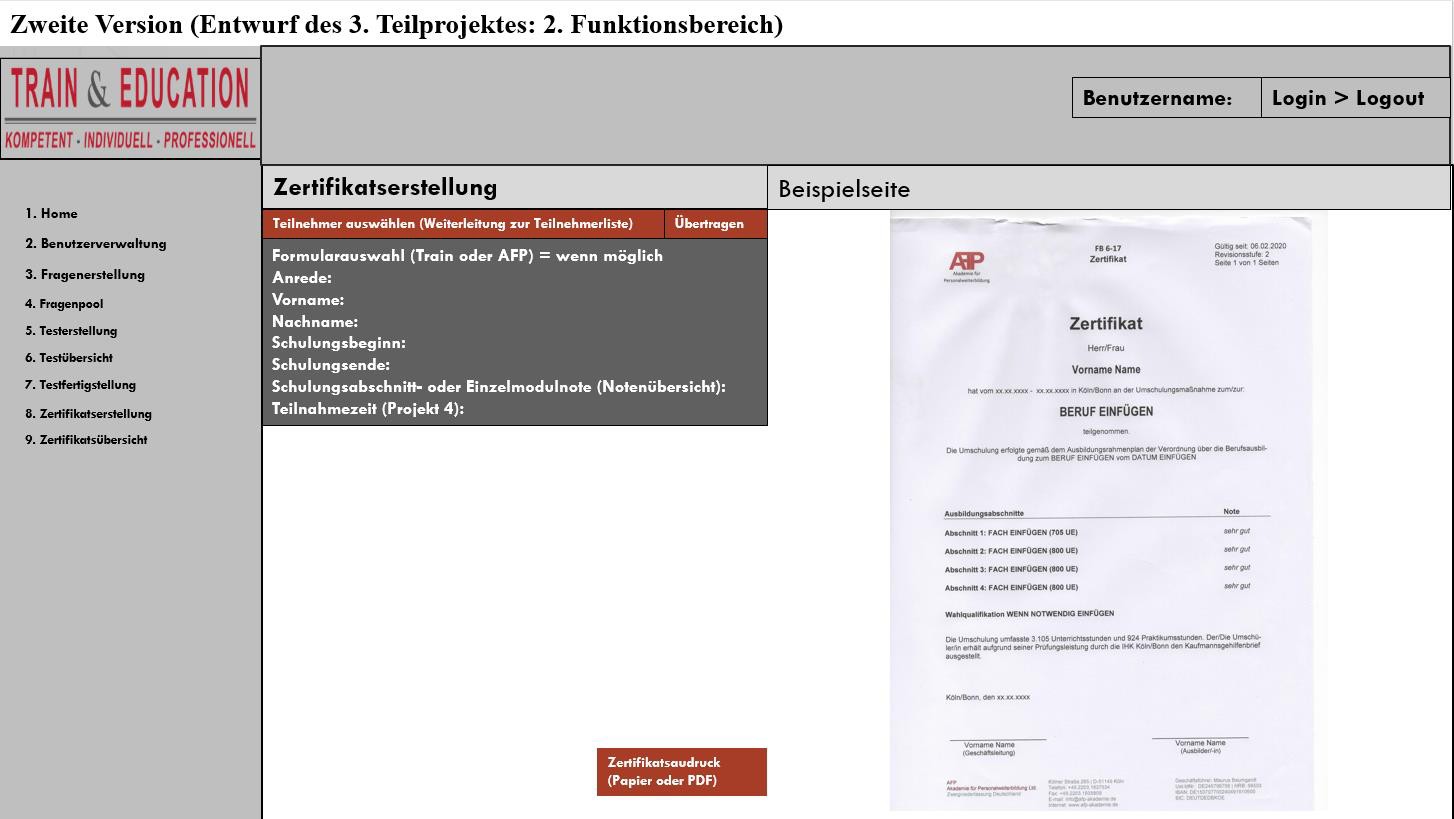
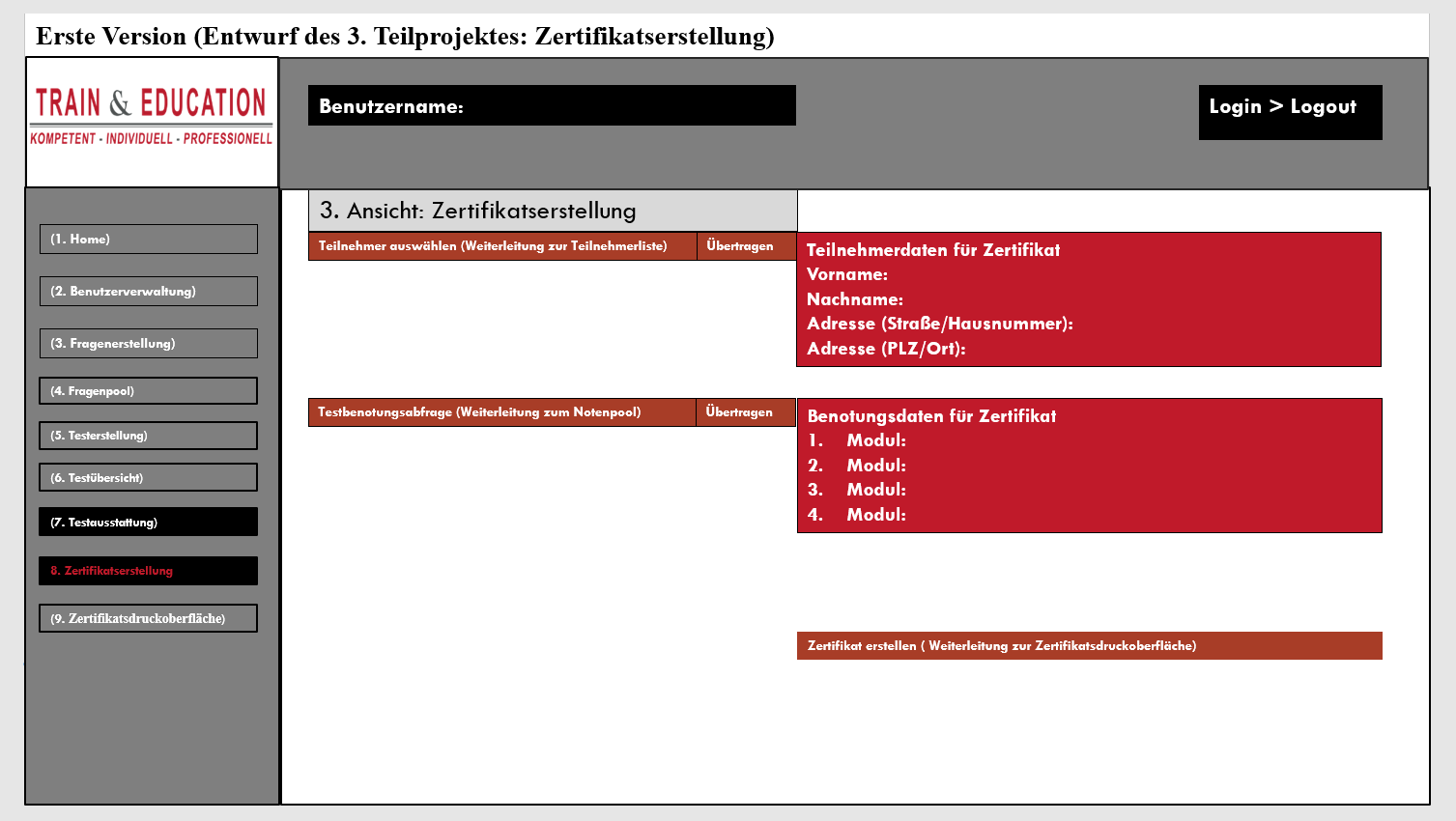
# A.9 Grafische Entwürfe der Teilprojekte (Seite 1)



*A Anhang*

[**Entwurf**](#_bookmark19)

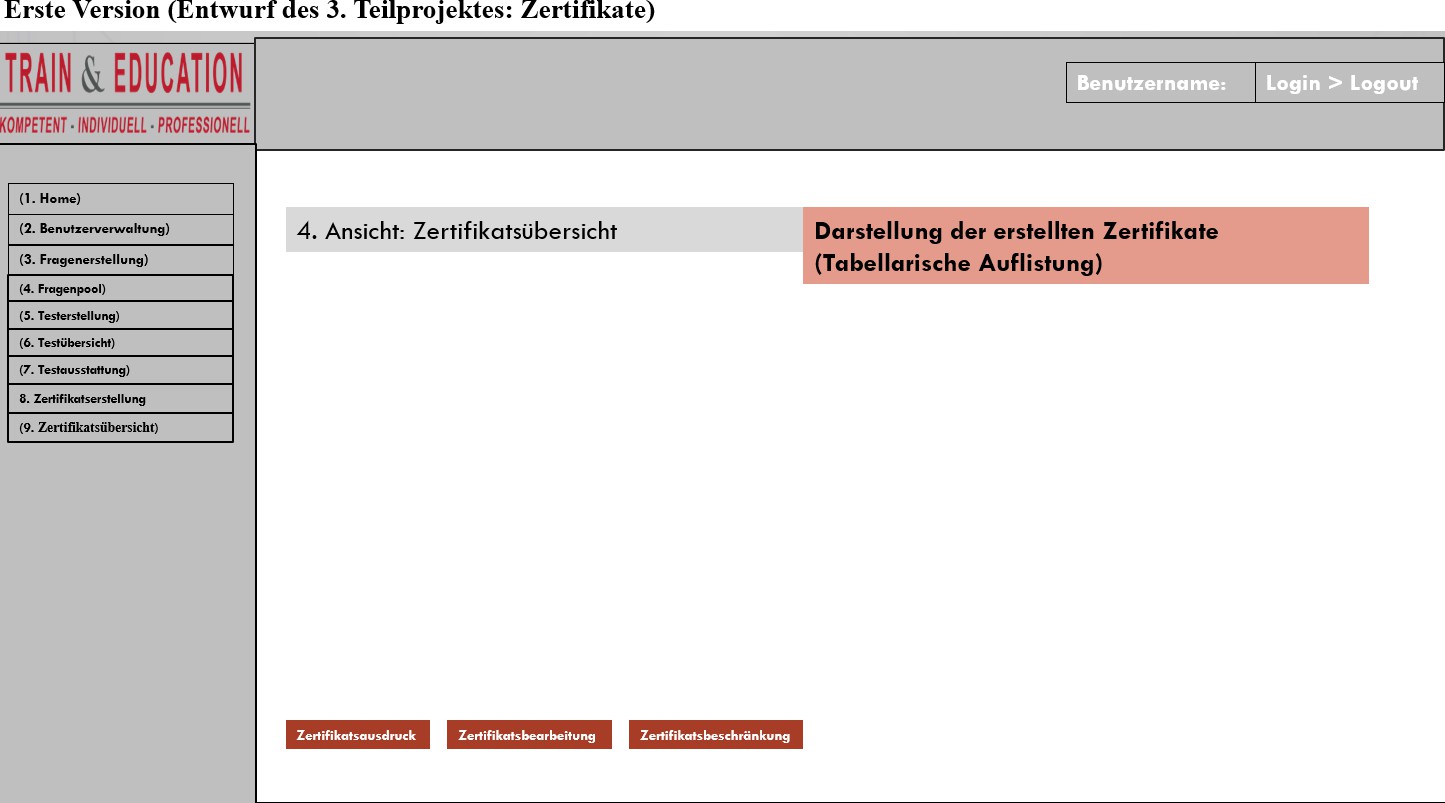
# Grafische Entwürfe der Teilprojekte (Seite 2)

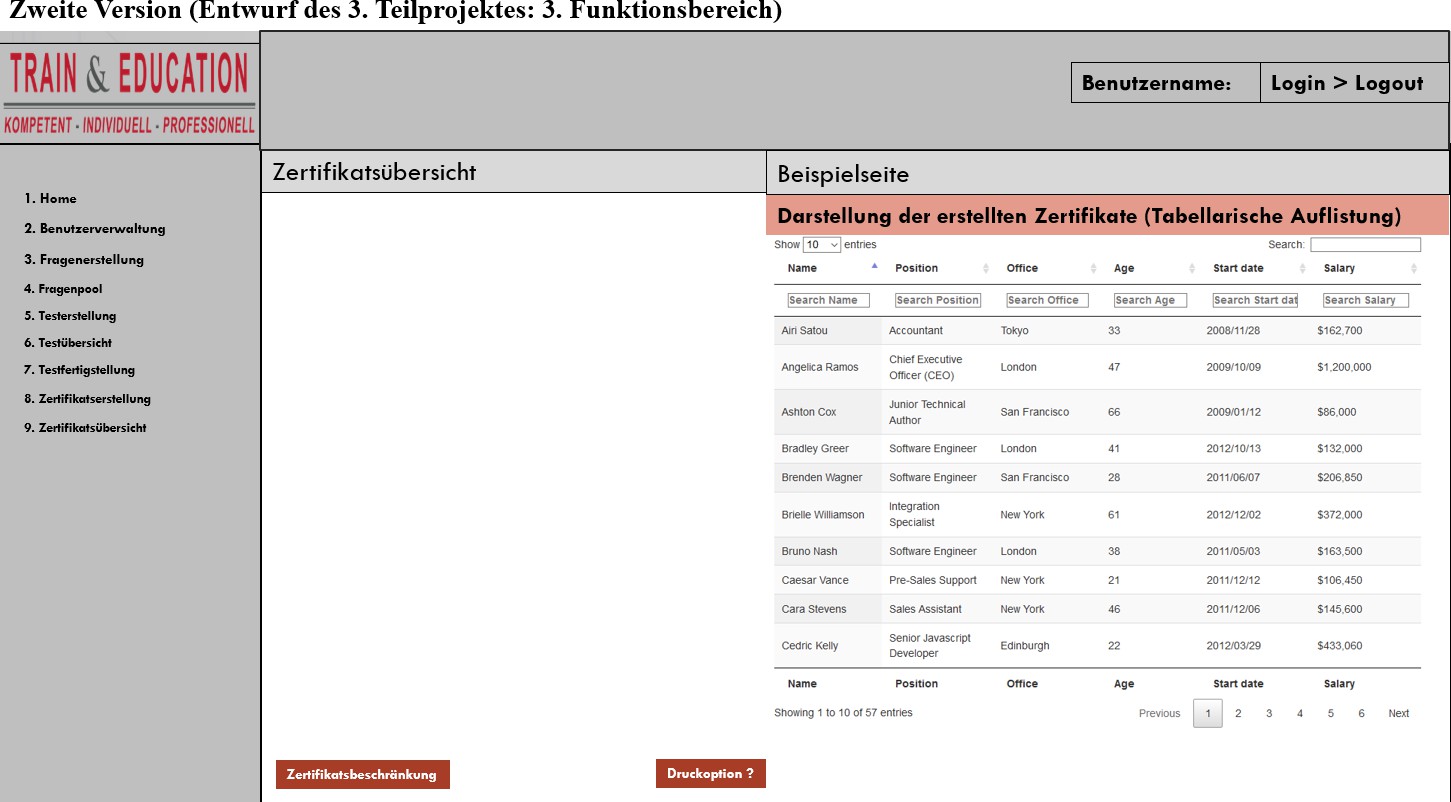


*A Anhang*

[**Entwurf**](#_bookmark20)

# Grafische Entwürfe der Teilprojekte (Seite 3)

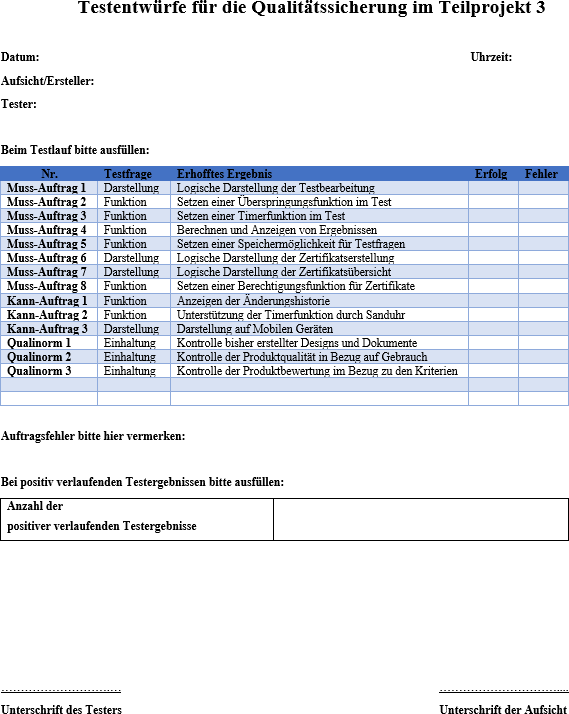




*A Anhang*

[**Entwurf**](#_bookmark21)

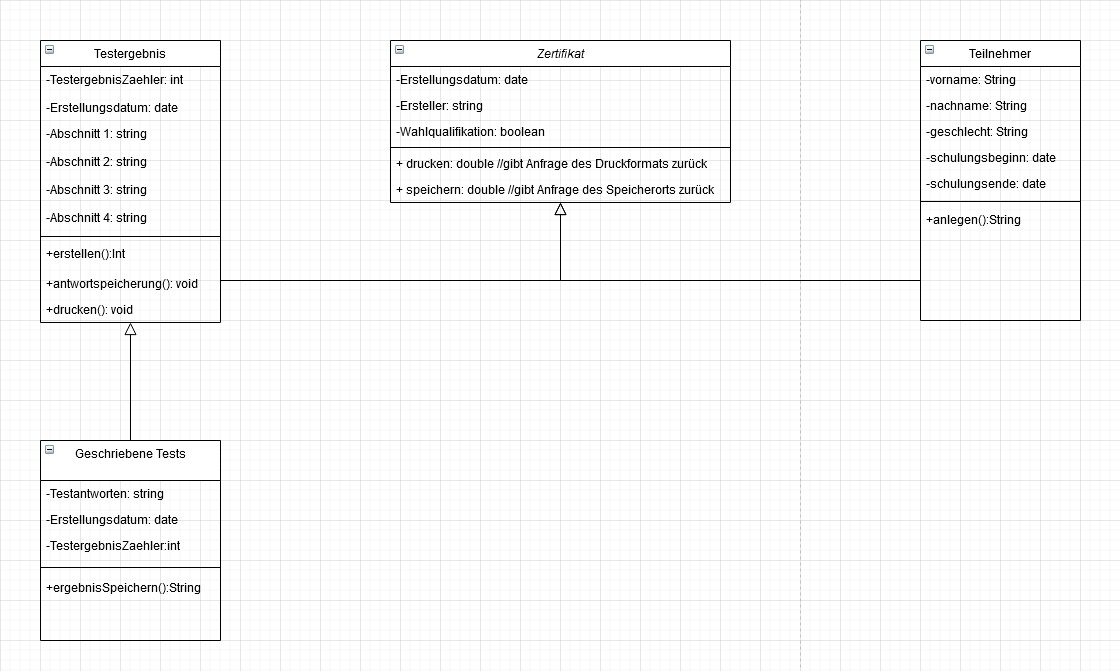
# Testentwürfe



*A Anhang*

[**Entwurf**](#_bookmark22)

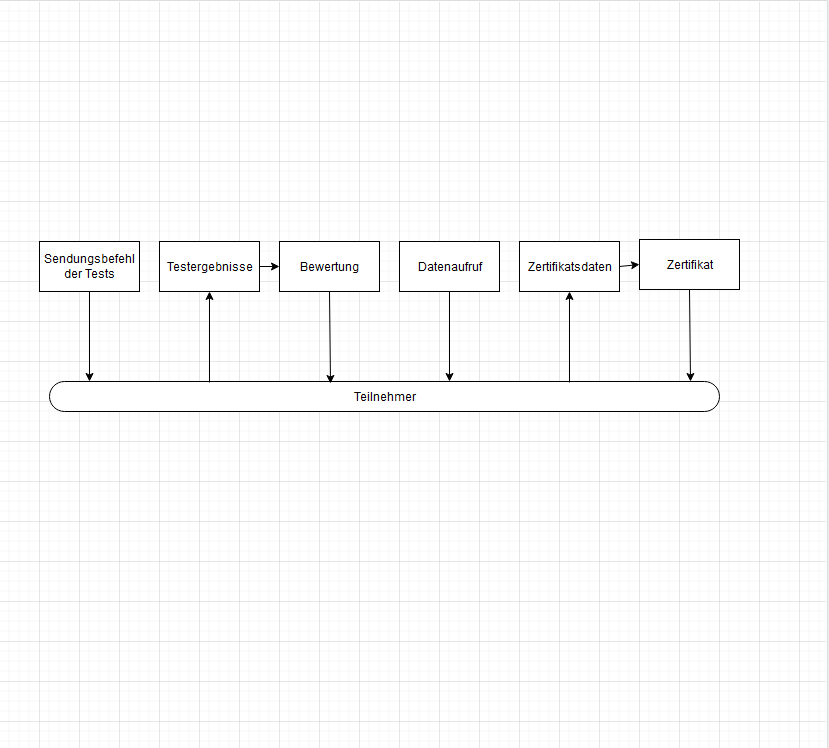
# Klassendiagramm



*A Anhang*

[**Entwurf**](#_bookmark22)

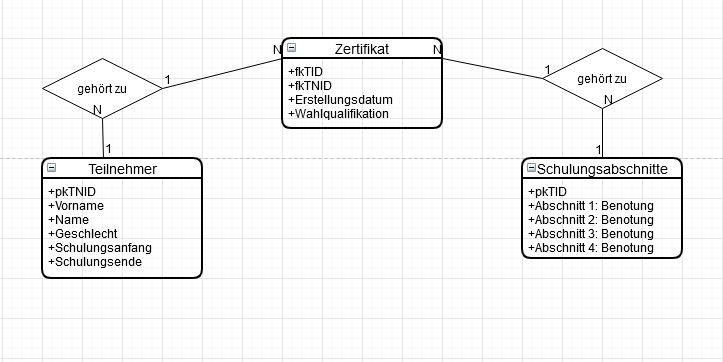
# Datenflussdiagramm



*A Anhang*

[**Entwurf**](#_bookmark22)

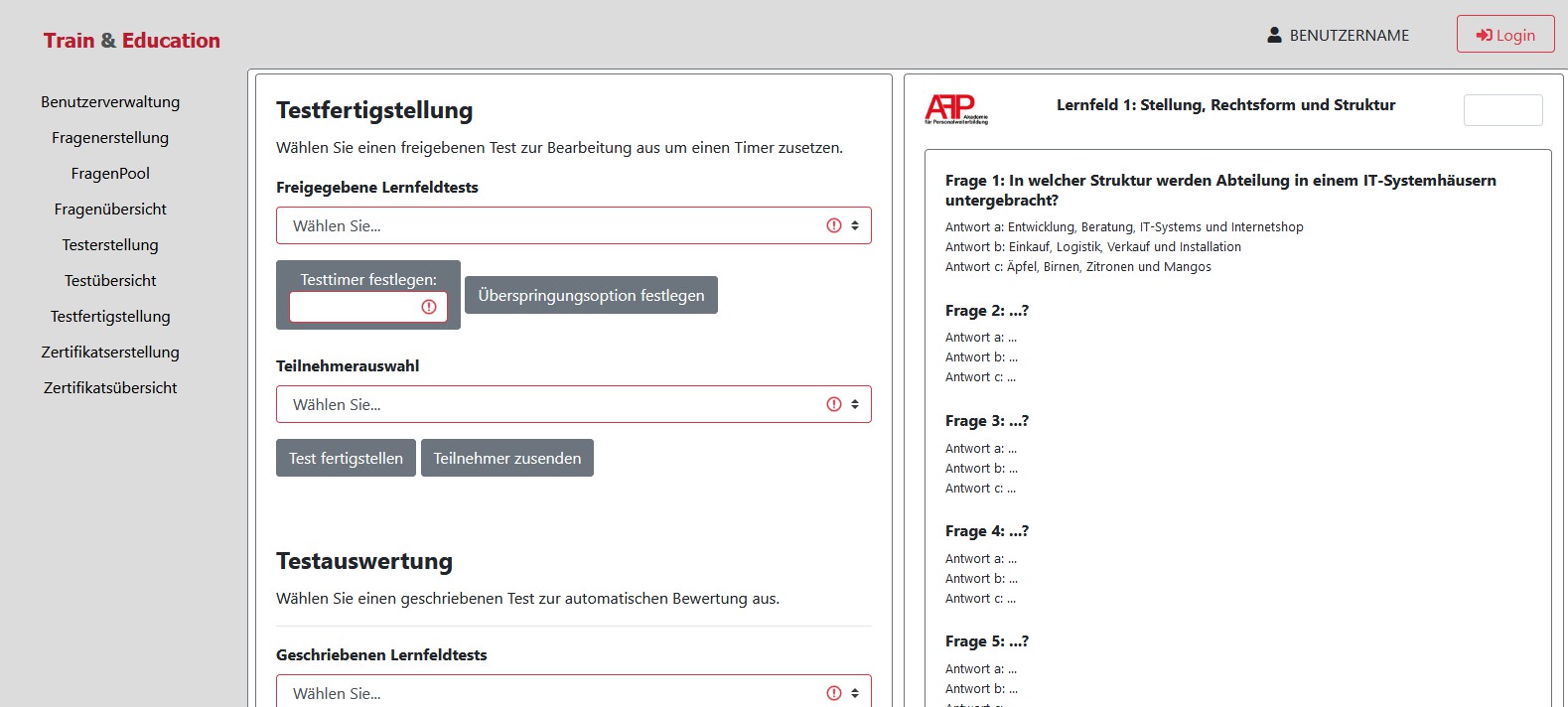
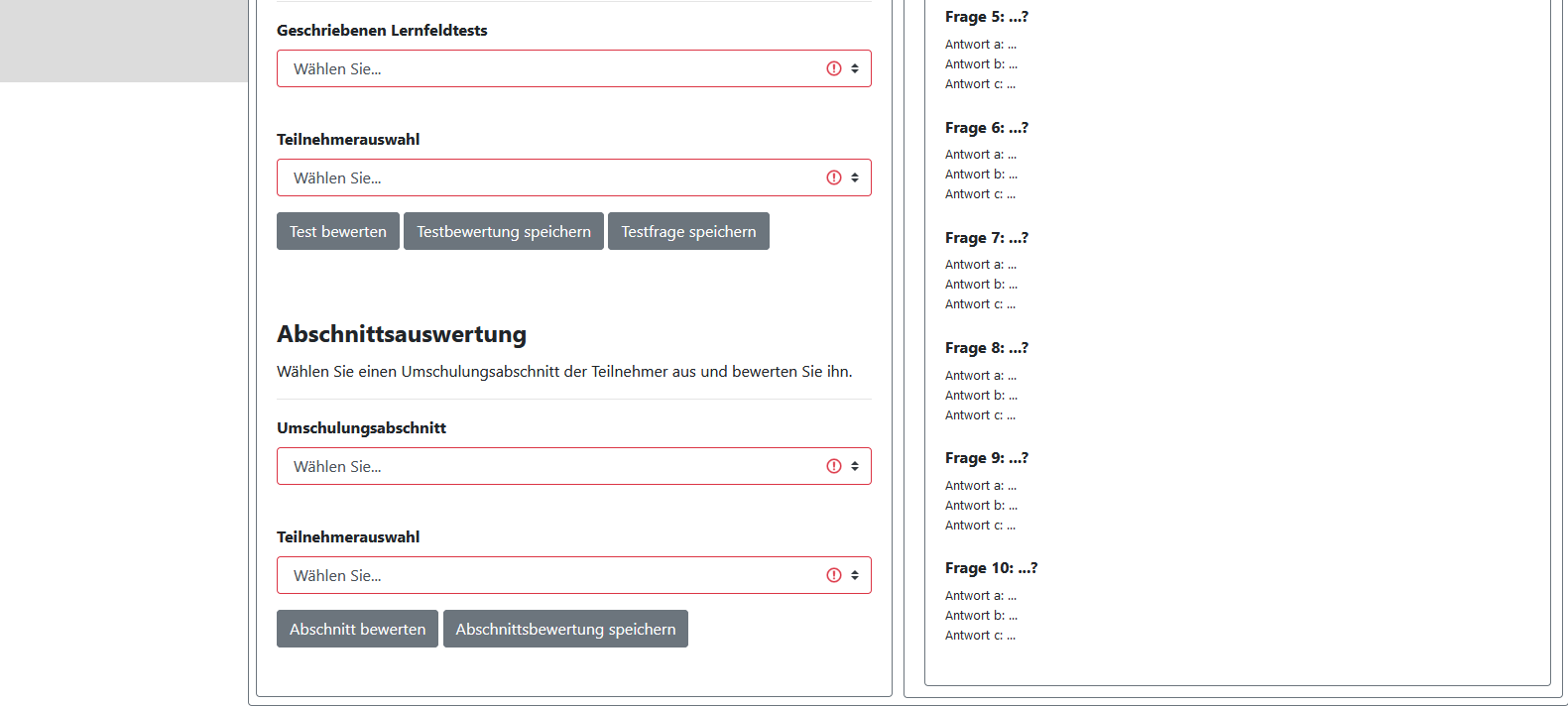
# ER-Modell



*A Anhang*

[**Implementierung**](#_bookmark24)

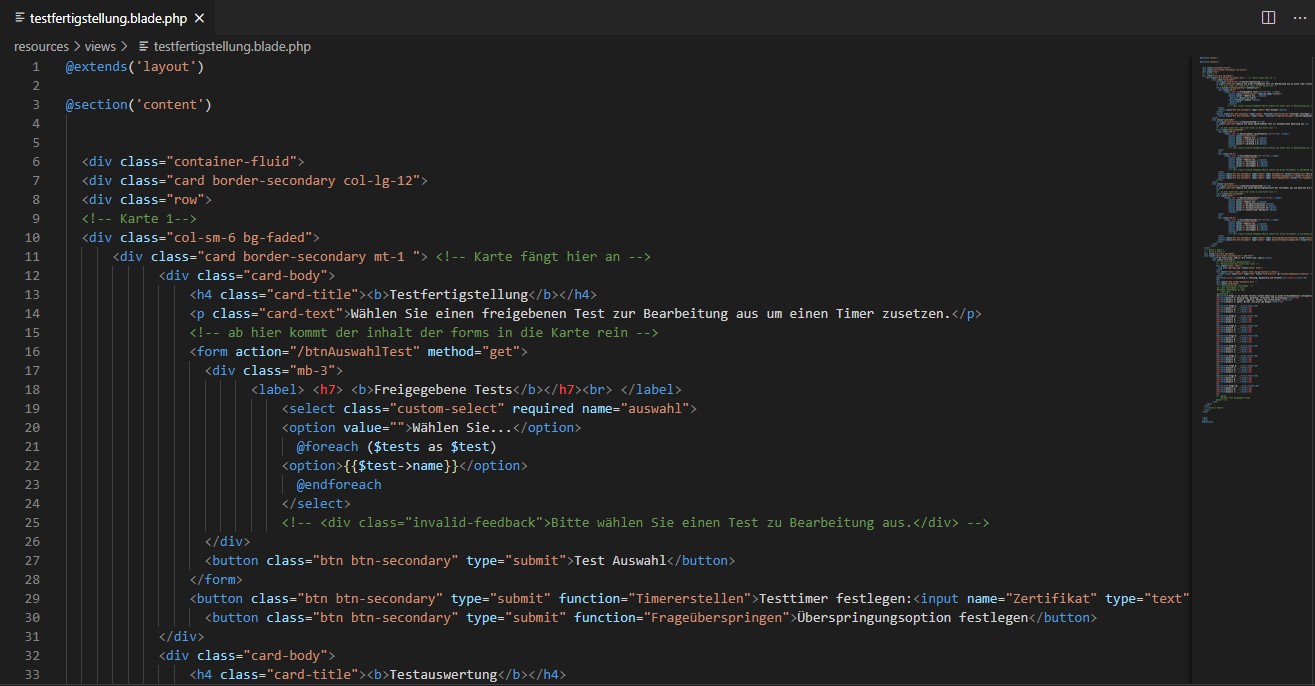
# Layout der Testfertigstellung



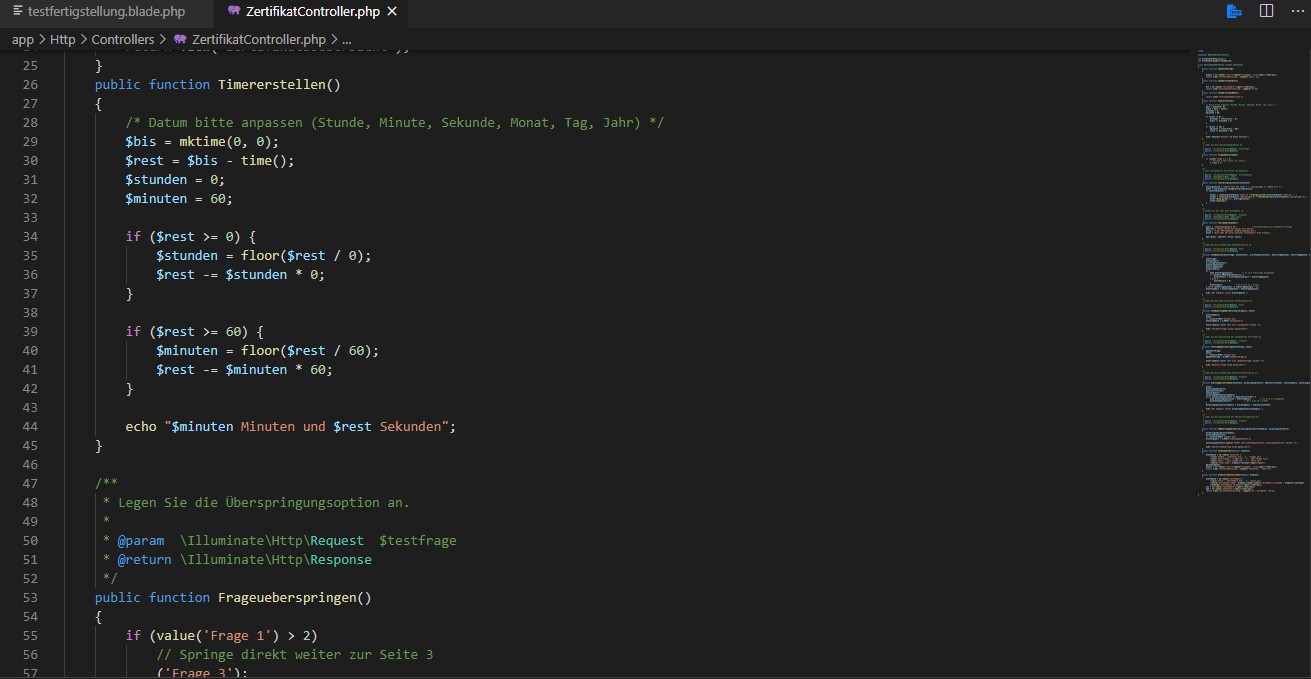
*A Anhang*

[**Implementierung**](#_bookmark24)

# Codebeispiele der Testfertigstellung



1. In der oben stehenden Grafik sieht man die Codes der Weboberfläche der Testfertigstellung.

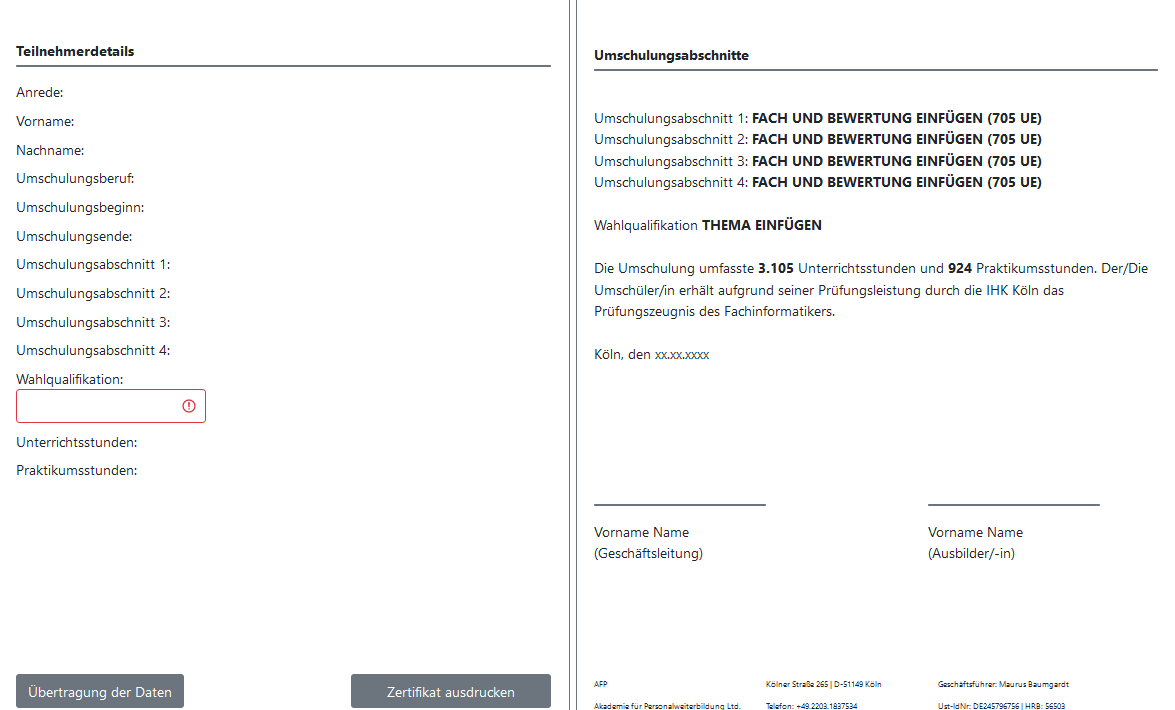


1. In dieser Abbildung erkennt man wie die Funktionen der Testfertigstellung untergebracht wurden.

*A Anhang*

[**Implementierung**](#_bookmark25)

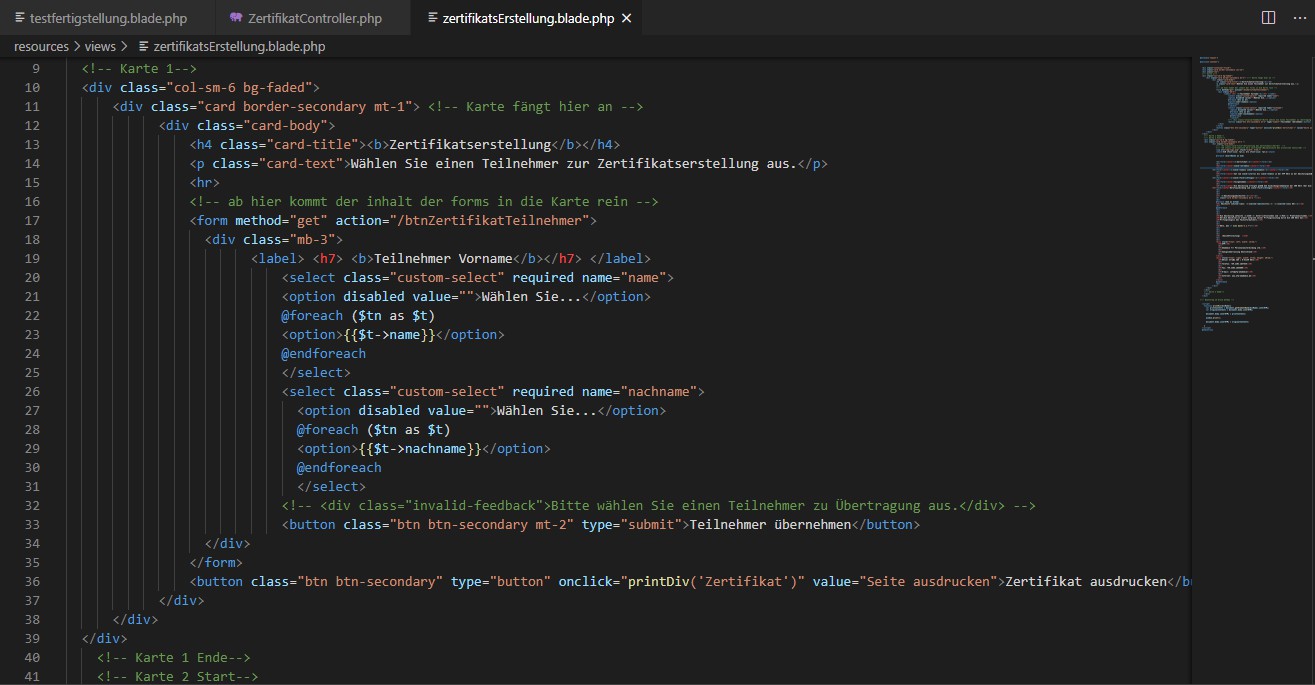
# Layout der Zertifikatserstellung



*A Anhang*

[**Implementierung**](#_bookmark25)

# Codebeispiele der Zertifikatserstellung



1. In dieser Grafik erkennt man die Programmierung der Oberfläche der Zertifikatserstellung.

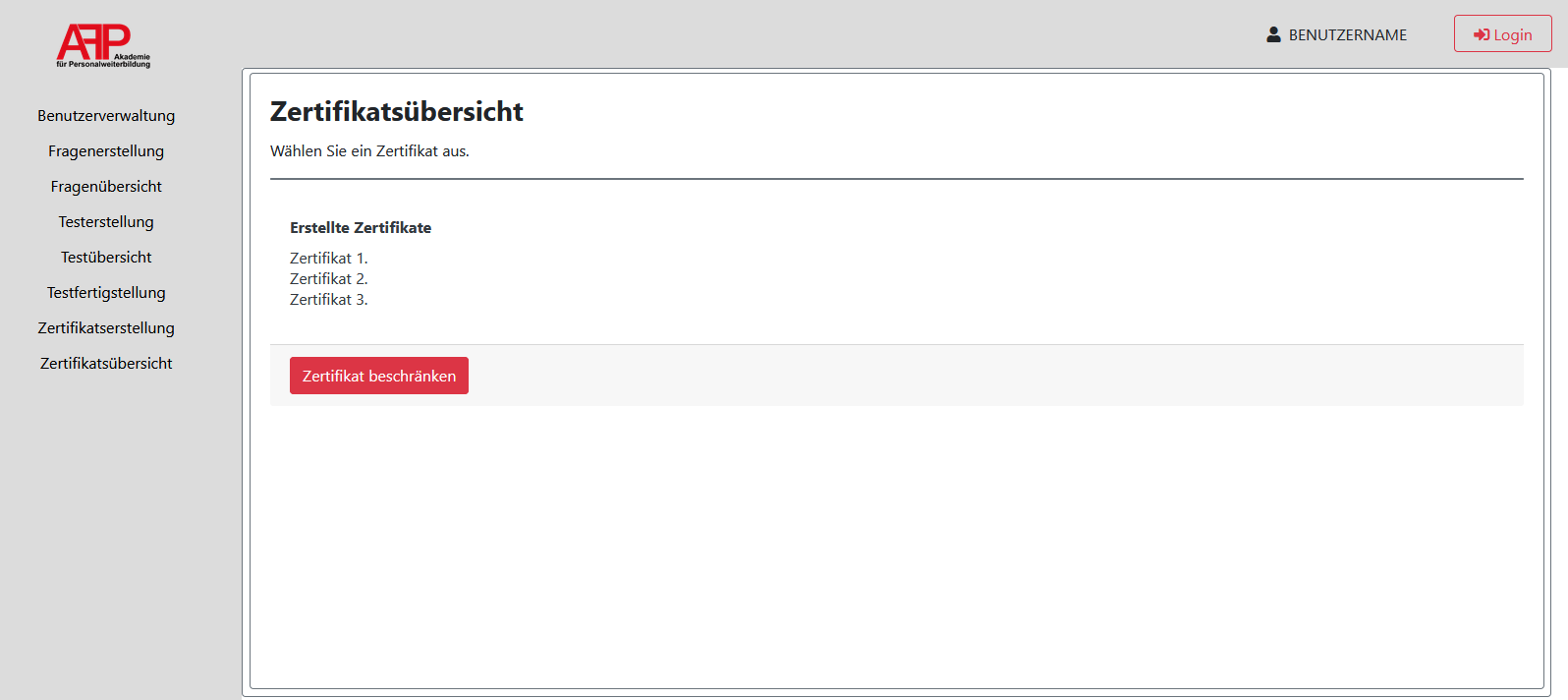


1. In dieser Abbildung erkennt man wie die Funktion der Zertifikatserstellung programmiert ist.

*A Anhang*

[**Implementierung**](#_bookmark26)

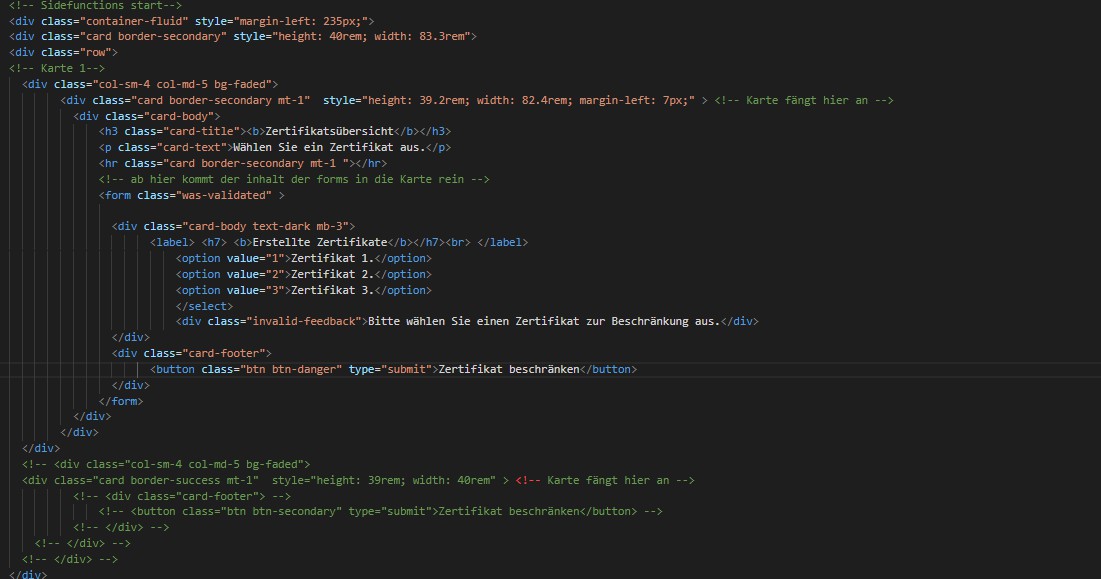
# Layout der Zertifikatsübersicht



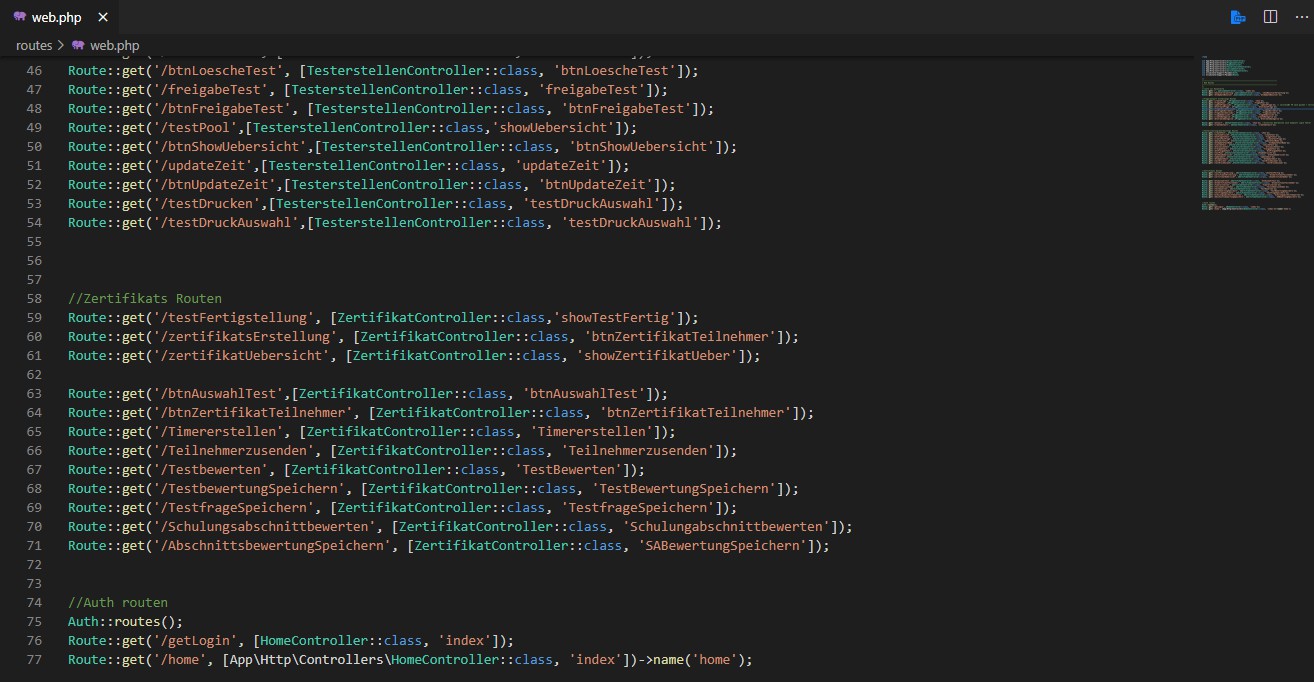
*A Anhang*

[**Implementierung**](#_bookmark26)

# Codebeispiele der Zertifikatsübersicht



In dieser Darstellung sieht man die Programmierung der Zertifikatsübersicht.

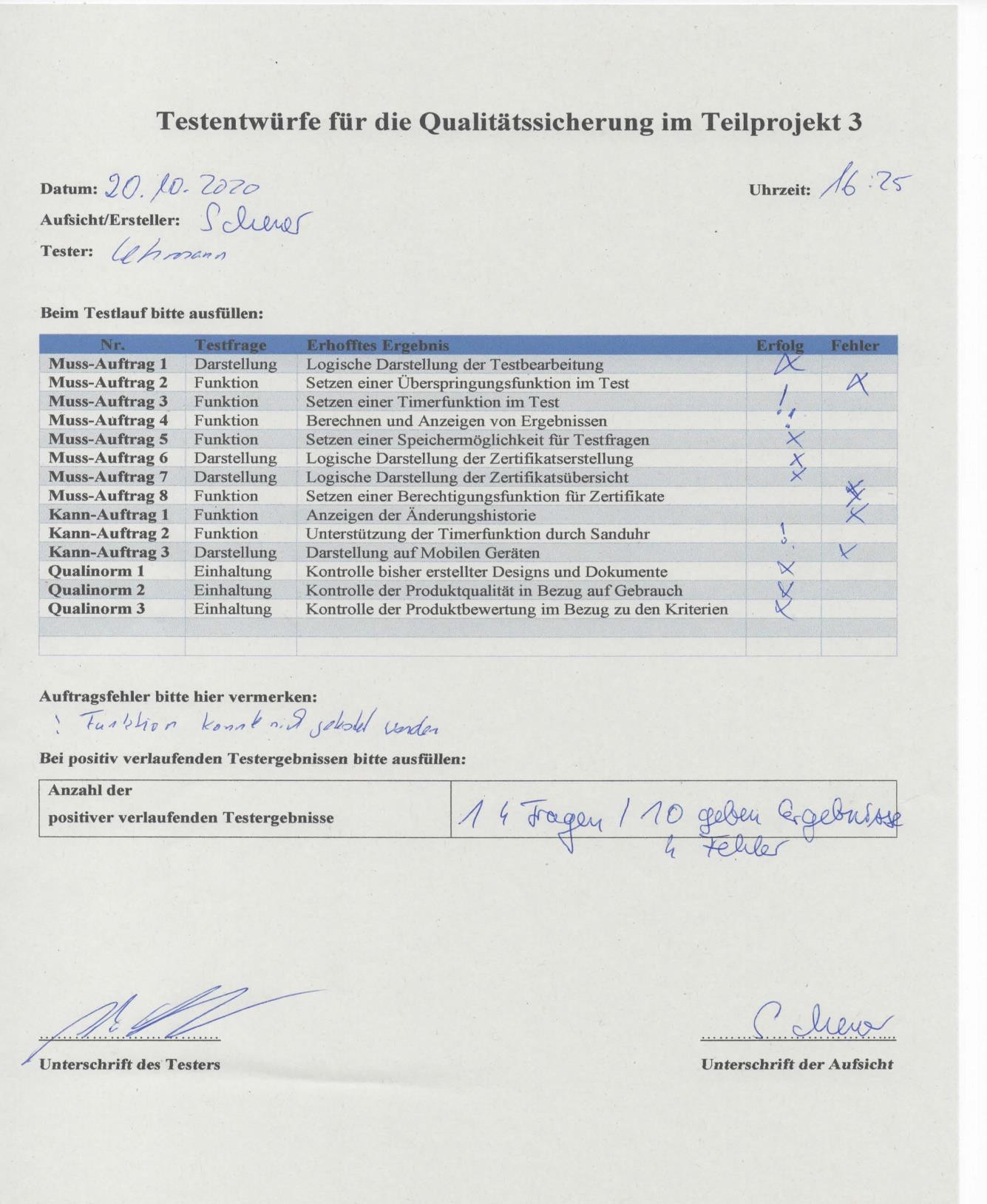


In dieser Grafik sieht man wie die Verbindung der Funktionen zu den Buttons auf der Website erstellt werden.

*A Anhang*

[**Implementierung**](#_bookmark27)

# Protokoll des ersten Testlaufs



*A Anhang*

[**Abschluss**](#_bookmark30)

# Abschlussbericht des Projektes

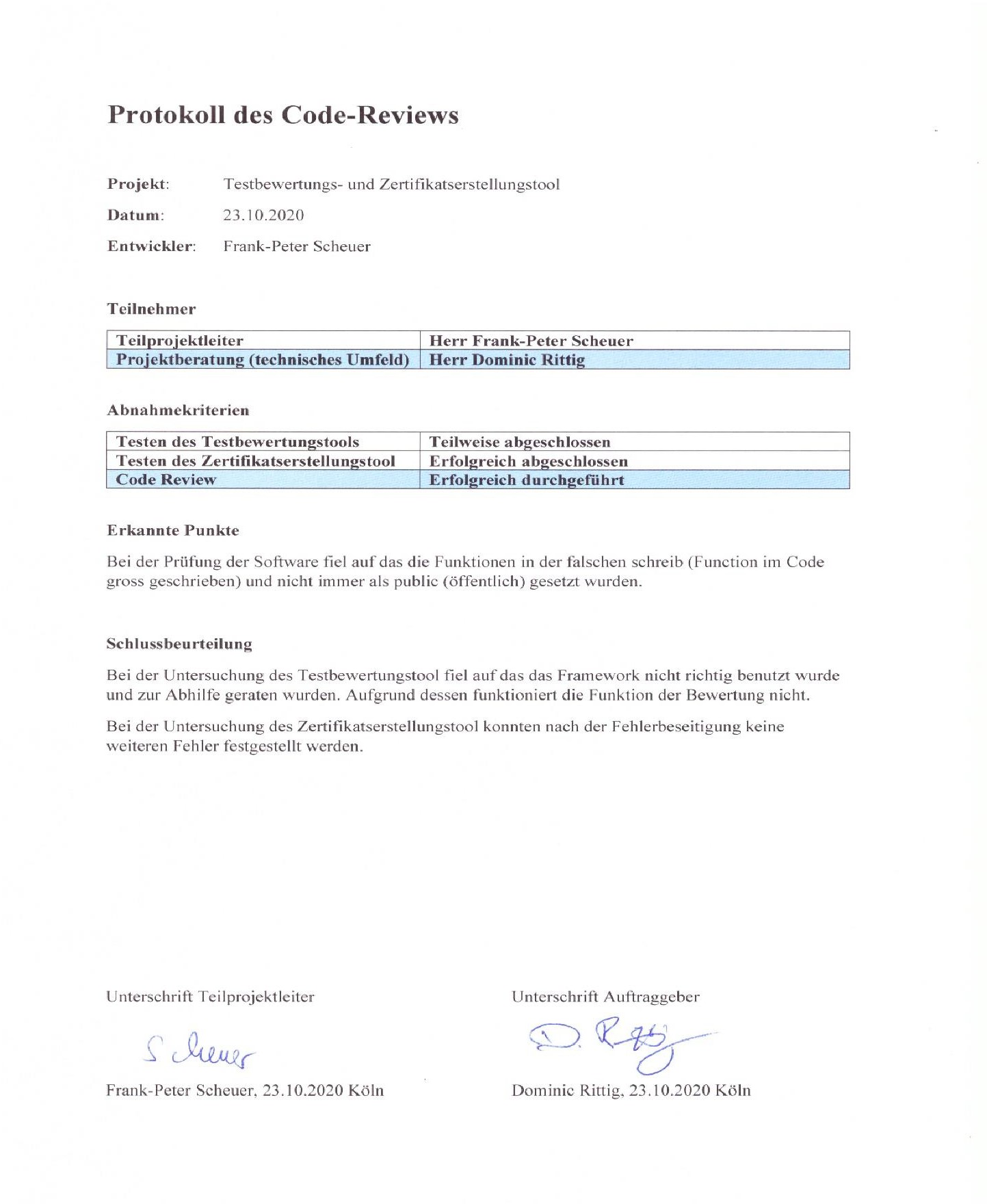
|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Auftraggeber**: Herr Andre Liesenfeld | |
| **Projektbeginn**: 12.10.2020 | **Projektende**: 23.10.2020 | **Projektkoordinator**: Herr Oliver Renkwitz | |
| **Projektziele und Aufgabenstellung:**   * **Ansichtserstellung zur Testfertigstellung**   + **mit den Funktionen:** Überspringen, Timer, Bewerten (Test, Lernfeld, Abschnitt) * **Ansichtserstellung zur Zertifikatserstellung**   + **mit den Funktionen:** Übertragen, manuelle Eingabe für Wahlqualifikation * **Ansichtserstellung zur Zertifikatsbeschränkung**   + **mit der Funktion:** Beschränkung der Übersichtsansicht | | | |
| **Projektorganisation:**   * Herr Andre Liesenfeld (als Vertretung von Herrn Baumgardt) * Herr Oliver Renkwitz (zur Koordination des Gesamtprojektes) * Herr Dominic Rittig (zur Beratung bei Projektfragen) * Herr Gökdeniz Karanfil (Bearbeiter des vorhergehenden Teilprojektes) | | | |
| **Leistungen und Projektergebnisse:**   * Das Frontend konnte komplett erstellt und im Backend die Funktionen vorbereitet werden. * Konzeptionelle wurde das Projekt so aufgebaut das die Anlagenbearbeitung der Dokumentation verstärkt in der Zeit stattfand und die Funktionserstellung. * Die technische Umsetzung fand auf dem bereitgestellten Laptop statt um die Arbeit in der Corona Situation stattfinden zu lassen. * Zur Sicherung der Qualitätsmerkmale wurde in der laufenden Entwicklung der Projektantrag eingehend beachtet und der in einem vorherigen Projekt erstellte Normenkatalog zu Rate gezogen. | | | |
| **Ereignisse der Durchführung:**   * Durch Anpassungen wurden im Projekt zum Großteil alle Kriterien eingebracht, bis auf die Muss- Funktion der Notenberechnung. Dadurch kam es zu kritischen Problemen in der Erstellung des Zertifikats, die schlussendlich durch die Erfüllung einiger Kann-Kriterien kompensiert werden   konnten. | | | |
| **Nachkalkulation:**   * Es fielen keine zusätzlichen Kosten für das Projekt an. | | | |
| **Ort, Datum**  ***Unterschriften*** | *Auftraggeber* | | *Projektkoordination* |

Dies war der Entwurf für einen eigenen Abschlussbericht, der zum besseren Projektverständnis dem Design der anderen Teilprojekte angepasst wurde.

*A Anhang*

[**Abschluss**](#_bookmark29)

# Protokoll des Code-Review



*A Anhang*

[**Abschluss**](#_bookmark31)

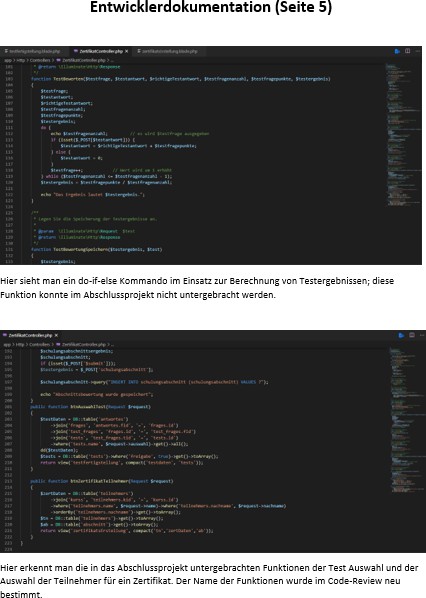
# Unterzeichnetes Abnahmeprotokoll



*A Anhang*

[**Abschluss**](#_bookmark32)

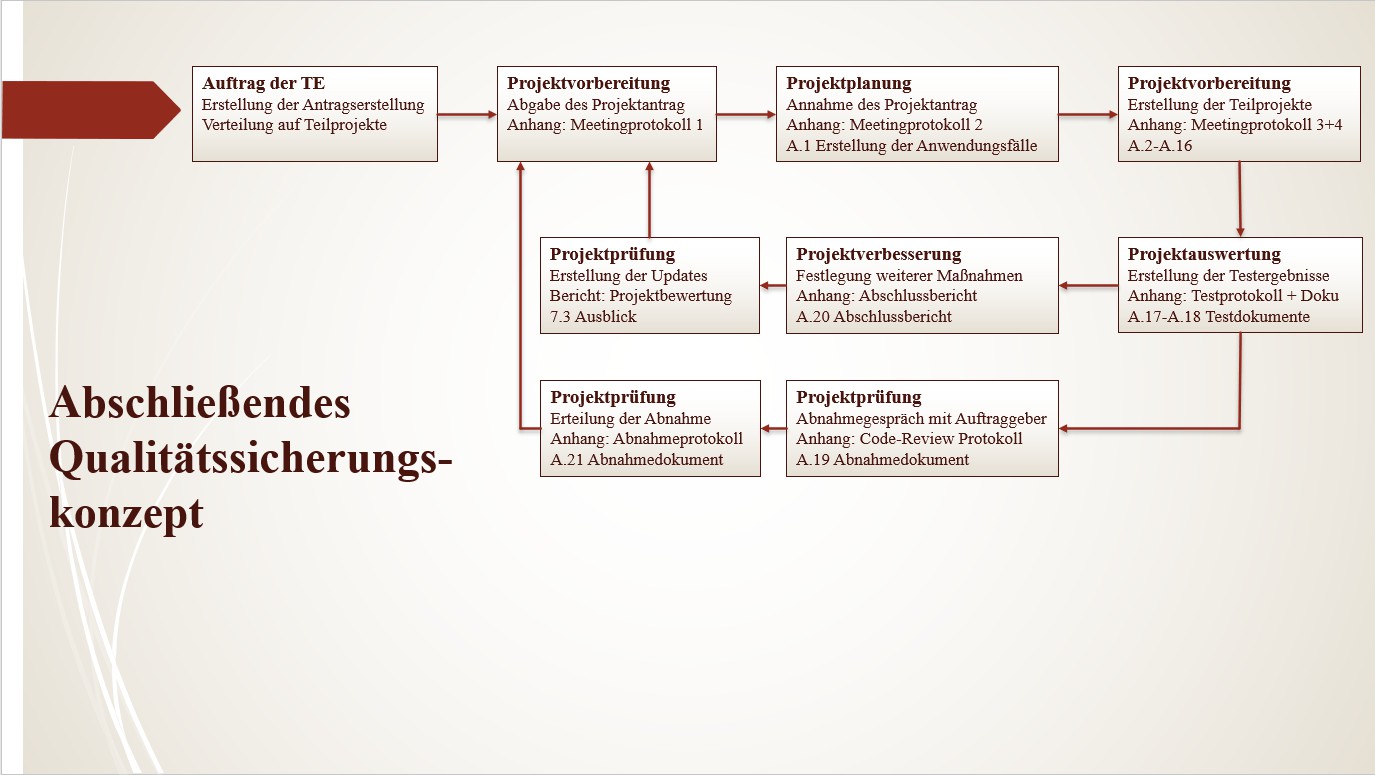
# Auszug aus der Entwicklerdokumentation



*A Anhang*

[**Abschluss**](#_bookmark33)

# Abschließendes Qualitätssicherungskonzept mit Bericht

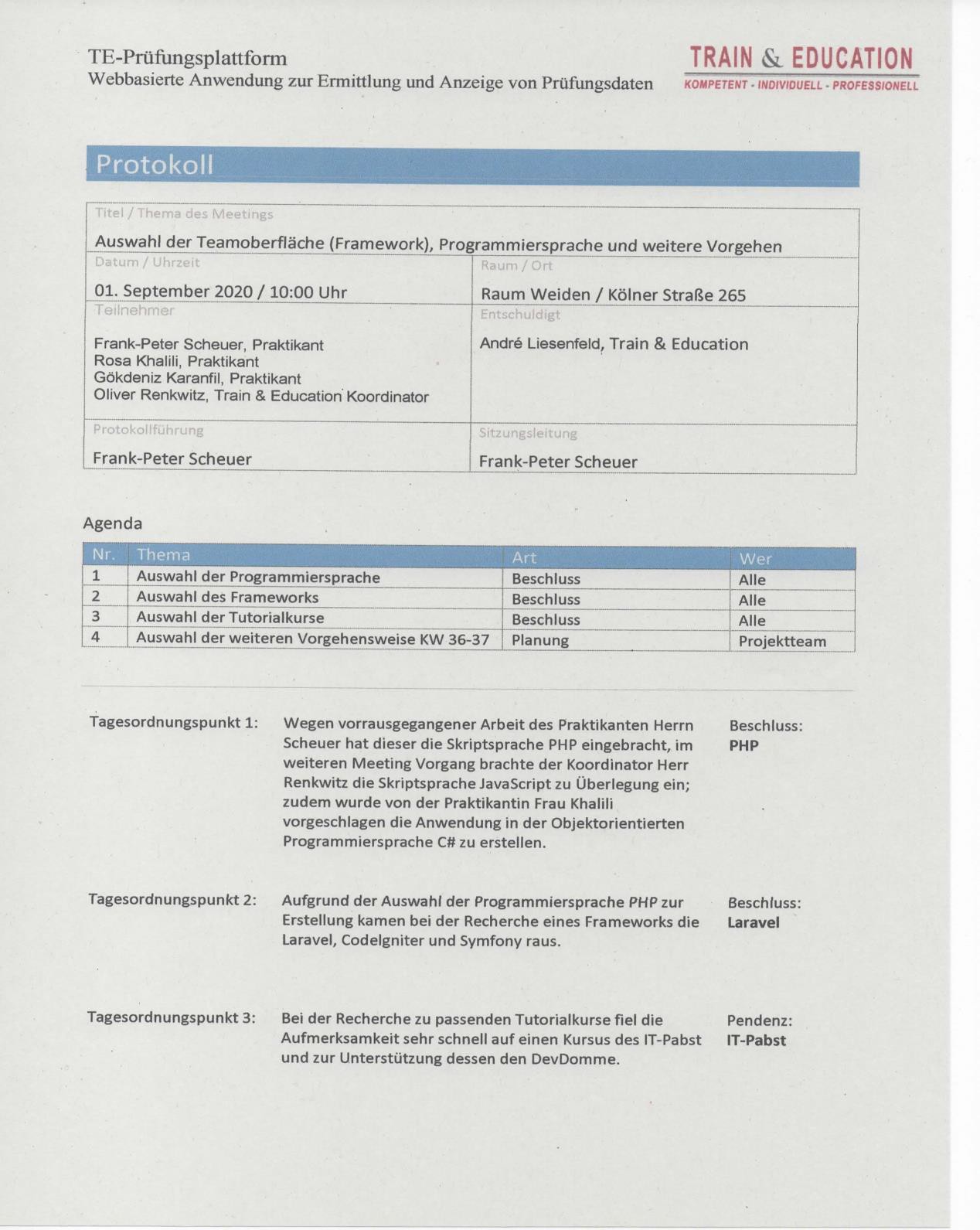


Anhand der oben dargestellten Grafik erkennt man das die Qualität der Anwendung im Laufe der Entwicklung durch mehrere Maßnahmen unterstützt wurde. Abschließend des Teilprojektes steht der Abschlussbericht mit Abnahmeprotokoll was zu einem weiteren Ausbau der Funktionen führt.

*A Anhang*

[**Abschluss**](#_bookmark33)

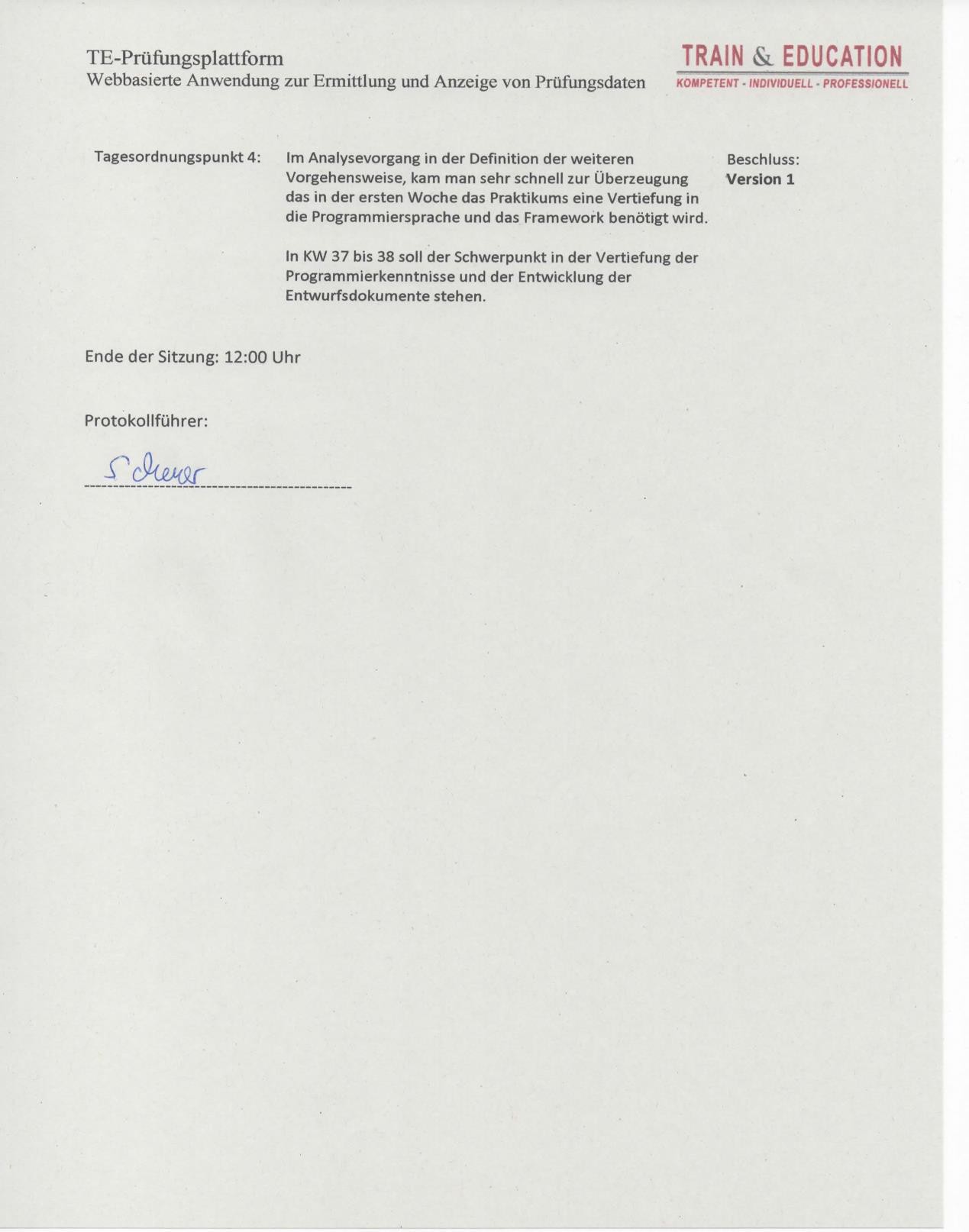
* 1. **Abschließendes Qualitätssicherungskonzept (Meeting 1.1)**



*A Anhang*

[**Abschluss**](#_bookmark33)

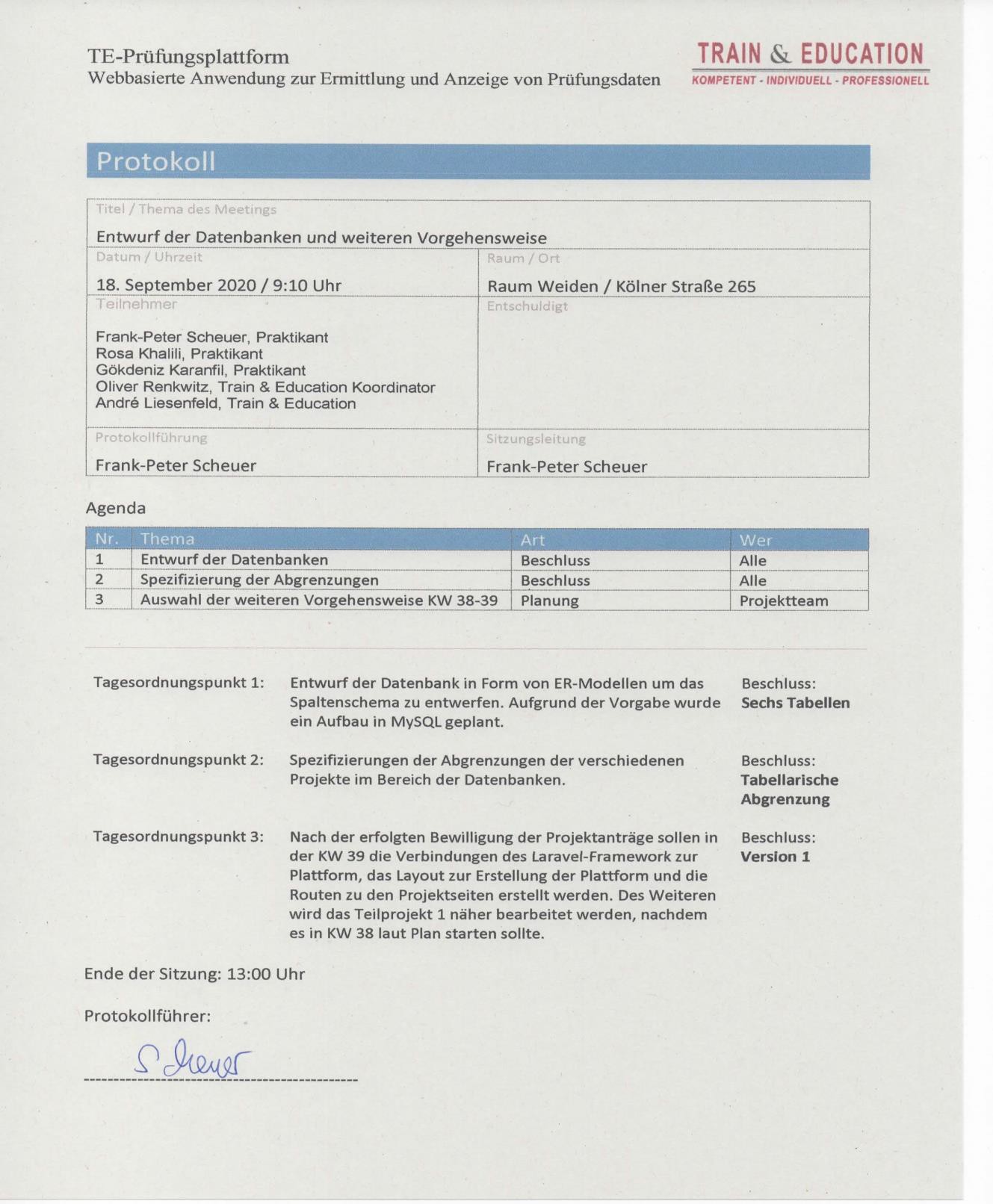
* 1. **Abschließendes Qualitätssicherungskonzept (Meeting 1.2)**



*A Anhang*

[**Abschluss**](#_bookmark33)

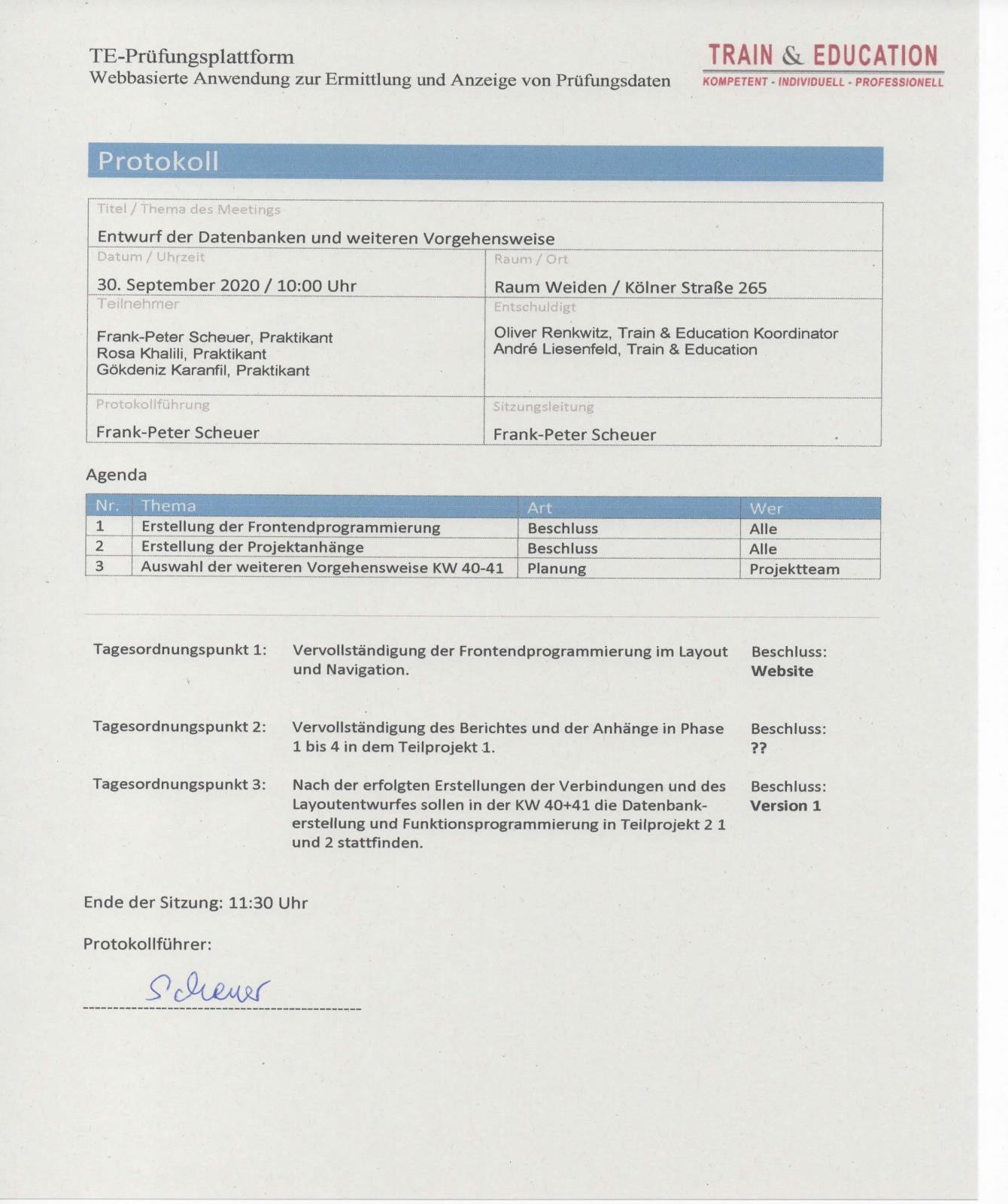
* 1. **Abschließendes Qualitätssicherungskonzept (Meeting 2)**



*A Anhang*

[**Abschluss**](#_bookmark33)

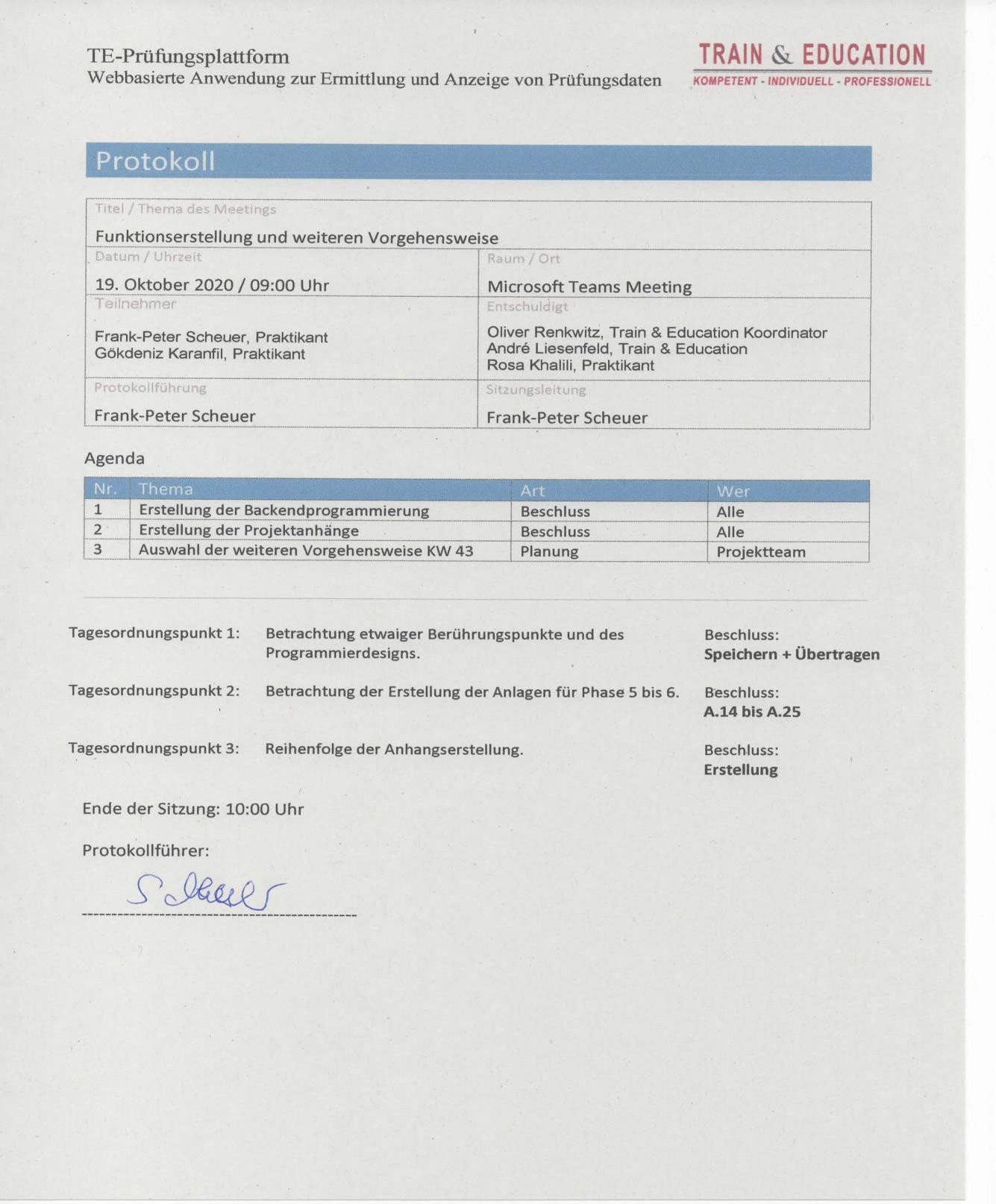
* 1. **Abschließendes Qualitätssicherungskonzept (Meeting 3)**



*A Anhang*

[**Abschluss**](#_bookmark33)

* 1. **Abschließendes Qualitätssicherungskonzept (Meeting 4)**



*A Anhang*

[**Abschluss**](#_bookmark34)

# Benutzerhandbuch (Seite 1)

1. **Die Testfertigstellung und Zertifikatserstellung**
   1. Wie bekomme ich die Tests?
   2. Wofür kann ich die Software benutzen?
   3. Was muss ich vorher für die Nutzung können?
2. **Nutzung des Fertigstellungs- und Erstellungstool**
   1. Die Testfertigstellung und ihre Funktionen
   2. Die Zertifikatserstellung und ihre Funktionen
   3. Die Zertifikatsübersicht und ihre Funktionen
3. **Beispiele zur Anwendung**
   1. Testfertigstellen und versenden
   2. Testergebnisse auswerten und abspeichern
   3. Schulungsabschnitte bewerten und abspeichern
   4. Zertifikate erstellen und ausdrucken
   5. Zertifikatsansicht beschränken
4. **Glossar**

*A Anhang*

[**Abschluss**](#_bookmark34)

# Benutzerhandbuch (Seite 2)

1. **Die Testfertigstellung und Zertifikatserstellung**
   1. **Wie bekomme ich die Tests?**

Um Zugriff zu der Testfertigstellung zu erhalten, in der die Testübersicht über die MySQL-Datenbank mit dieser Oberfläche verbunden. Die notwendige Verbindung muss von Ihrem zuständigen Administrations- ansprechpartner erstellt werden.

* 1. **Wofür kann ich die Software benutzen?**

Diese Anwendung ist zur Nutzung durch Dozenten bei der Notenerstellung und Zertifikatserstellung ausgelegt. Es ist aber möglich die Software an anderen Unternehmungsbereiche nach einer Anpassung weiterzugeben.

* 1. **Was muss ich vorher für die Nutzung können?**

Zur Nutzung der Anwendung ist es von Vorteil eine Grundschulung in der Nutzung von Bürosoftware erhalten zu haben.

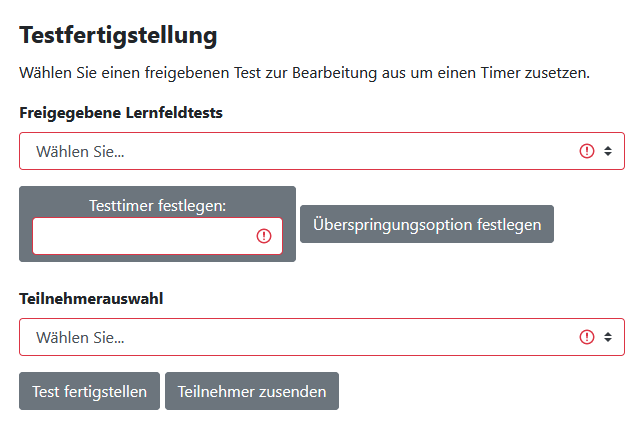
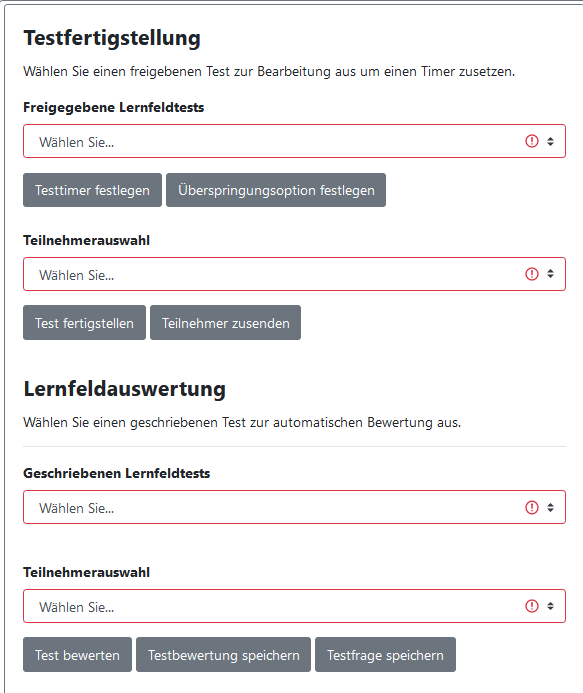
*A Anhang*

[**Abschluss**](#_bookmark34)

# Benutzerhandbuch (Seite 3)

1. **Nutzung des Fertigstellungs- und Erstellungstool**
   1. **Die Testfertigstellung und ihre Funktionen**

Nachdem Aufruf der **Testfertigstellung** wird dieser in zwei Bereiche eingeteilt: **Funktionen** und **Ansicht**.



Im ersten Funktionsbereich erhalten Sie den Zugriff auf freigegebene Tests in der MySQL- Datenbank. Der ausgewählte Test wird nun in der Ansicht angezeigt.

Über Schaltflächen haben Sie nun die Möglichkeit den Test mit einem „**Timer**“ einer

„**Überspringungs-option**“ auszustatten. Nachdem Sie im Folgenden einen Teilnehmer ausgewählt haben können Sie den Test fertigstellen und diesem zusenden.

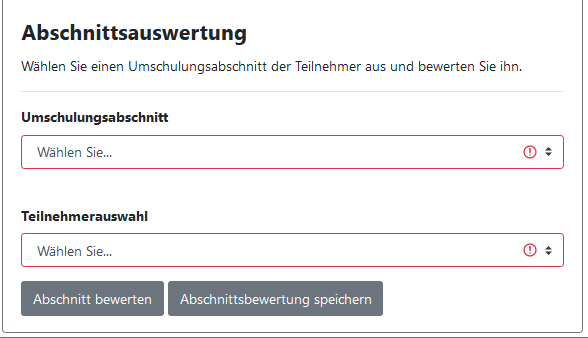
Im zweiten Funktionsbereich können die geschriebenen Lernfeldtests aus der MySQL- Datenbank entnommen und einem Teilnehmer zugewiesen werden. Daraufhin wird dieser ausgewählt und einem Teilnehmer zugeordnet. Nun kann über die Punktevergaben im Fragenpool über den Button “**Test bewerten**“ das Ergebnis automatisch erstellt und über

„**Testbewertung speichern**“ in der Datenbank hinterlegt werden. Zudem besteht die Möglichkeit über Testfrage speichern eine bestimmte zur Beachtung gesondert zu speichern.

*A Anhang*

[**Abschluss**](#_bookmark34)

# Benutzerhandbuch (Seite 4)

Im dritten Funktionsbereich ist die Funktion der Bewertung des Umschulungsabschnittes untergebracht. Zuerst wird über das Dropdown Menü der Umschulungsabschnitt dieser ausgewählt und die qualifizierte Teilnehmerauswahl zur Verfügung gestellt.

Dieser kann über den Button “**Abschnitt bewerten**“ automatisch erstellt und über

„**Abschnittsbewertung speichern**“ in der Datenbank hinterlegt werden.

* 1. **Die Zertifikatserstellung und ihre Funktionen**

Nachdem Aufruf der **Zertifikatserstellung** wird dieser ebenfalls in zwei Bereiche eingeteilt: **Funktionen** und **Ansicht**.

Im ersten Funktionsbereich erhalten Sie als erstes den Zugriff auf Bewertete Teilnehmer in der MySQL-Datenbank. Der ausgewählte Teilnehmer wird nun mit seinen hinterlegten, für ein Zertifikatbenötigten Daten in der Ansicht angezeigt.

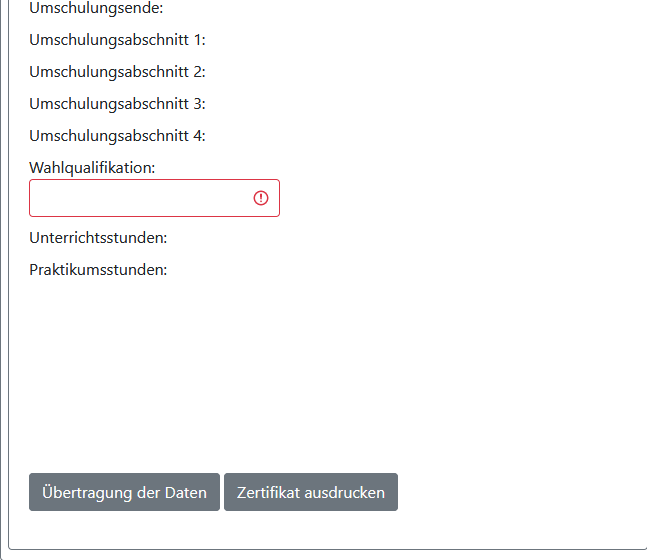
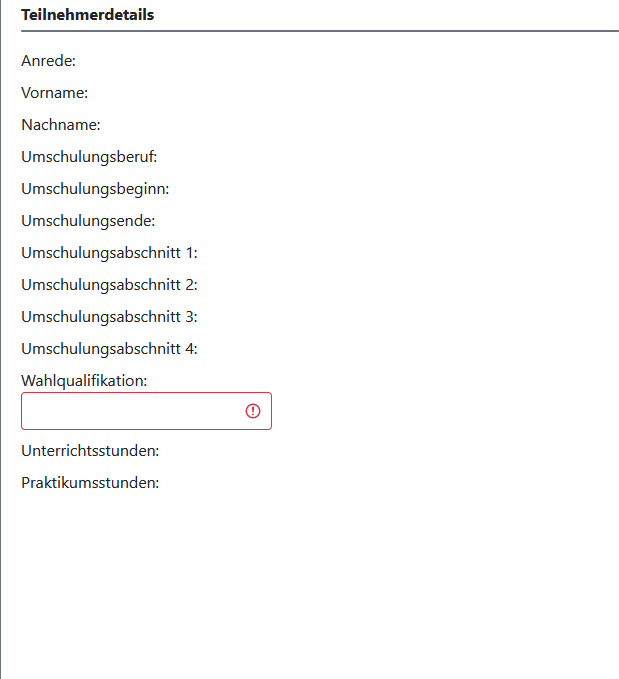
Bei den Teilnehmerdetails wird angeboten eine Wahlqualifikation einzutragen und mit dem Button „**Übertragen der Daten**“ werden diese im in der Ansicht gezeigten „**Formular**“ angezeigt.

*A Anhang*

[**Abschluss**](#_bookmark34)

# Benutzerhandbuch (Seite 5)

Das fertige Formular kann nun vom Dozenten ausgedruckt werden.

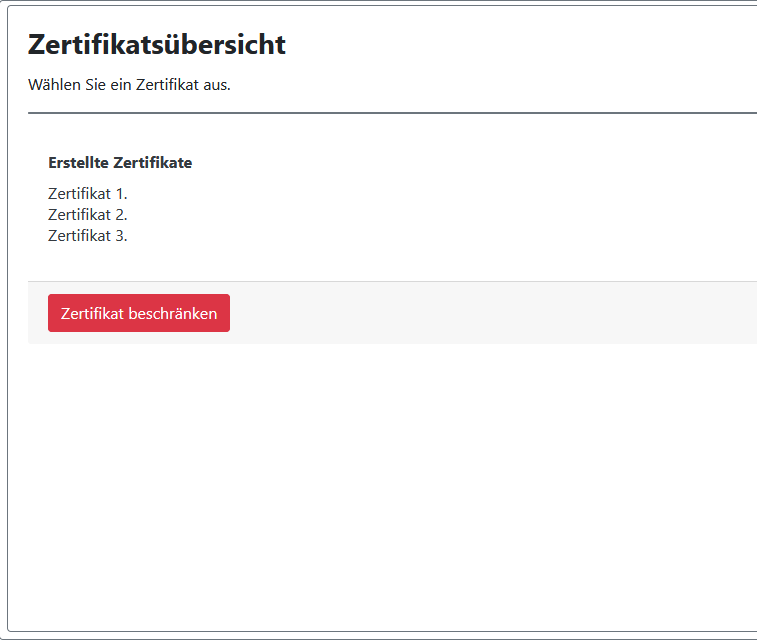


*A Anhang*

[**Abschluss**](#_bookmark34)

# Benutzerhandbuch (Seite 6)

* 1. **Die Zertifikatsübersicht und ihre Funktionen**

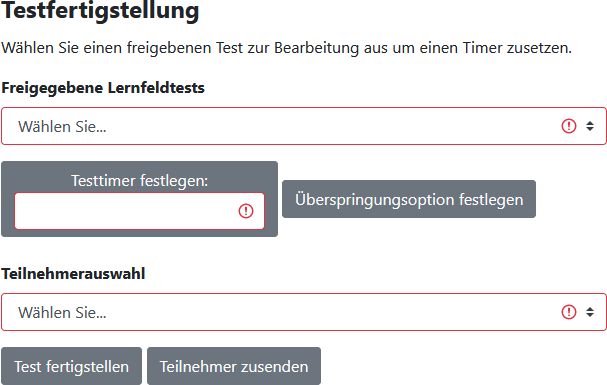
Nachdem Aufruf der Zertifikatsübersicht wird Ihnen die in der MySQL-Datenbank hinterlegten erstellten Zertifikate angezeigt. Sie haben nun die Möglichkeit über den Button „**Zertifikat beschränken**“ diesen für bestimmte Benutzergruppen zu sperren.

*A Anhang*

[**Abschluss**](#_bookmark34)

# Benutzerhandbuch (Seite 7)

1. **Beispiele zur Anwendung**
   1. **Testfertigstellen und versenden**

Zur Fertigstellung eines Testes öffnen Sie über Freigegebene Lernfeldtests diese MySQL Datenbank damit dieser im Ansichtsbereich angezeigt wird. Geben Sie nun den

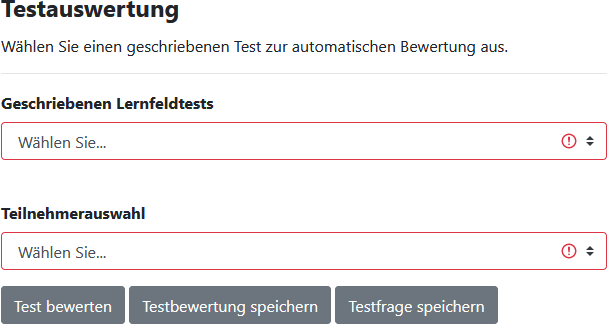
**„Testtimer“** an und ob die **„Überspringen“**

Option eingeschaltet werden soll.

Der **Timer** sollte nun im auf den Test angezeigt werden.

Durch das Auswählen des Teilnehmers von dem nun die E-Mail-Adresse aus der MySQL Datenbank hinterlegt wird kann der Test fertiggestellt und dem Teilnehmer zugesendet werden.

* 1. **Testergebnisse auswerten und abspeichern**



Betreffenden ausgewählt haben, diesem hinterlegt.

Zur Auswertung eines Testes öffnen Sie über Geschriebene Lernfeldtests diese MySQL Datenbank damit dieser im Ansichtsbereich angezeigt wird. Geben Sie nun den **„Test**

**bewerten“** an um diesen automatisch bewerten zu können (diese werden auf dem Test angezeigt) und mit der **„Testbewertung**

**speichern“** Option wird das Ergebnis nachdem Sie in der Teilnehmer auswahl einen

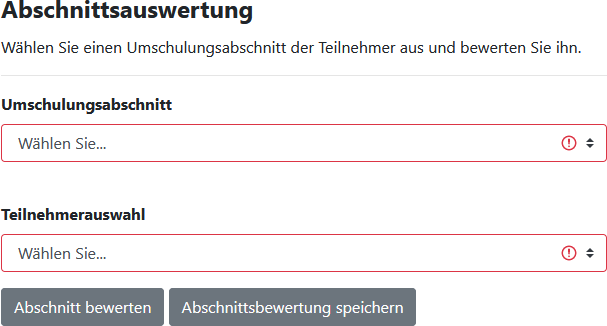
Wen sie eine bestimmte Frage speichern möchten können Sie dies mit „**Testfrage speichern**“ tun.

*A Anhang*

[**Abschluss**](#_bookmark34)

# Benutzerhandbuch (Seite 8)

* 1. **Lernfelder bewerten und abspeichern**

Zur Auswertung eines Umschulungsabschnittes öffnen Sie über Umschulungabschnitt die MySQL Datenbank und wählen einen aus und über Teilnehmerauswahl einen Teilnehmer aus dem Umschulungsabschnitt. Durch die im Abschnitt erstellten Testbewertung kann nun über den Button „Abschnitt bewerten“ ein Ergebnis erzielt werden.

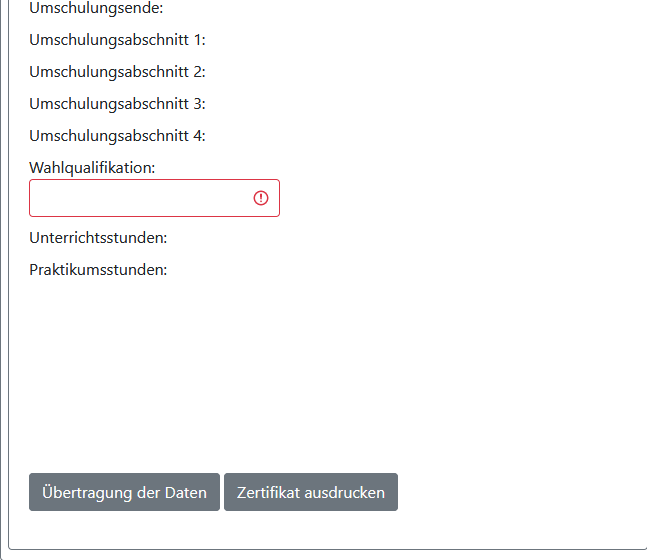
*A Anhang*

[**Abschluss**](#_bookmark34)

# Benutzerhandbuch (Seite 9)

* 1. **Zertifikate erstellen und ausdrucken**

Zur Erstellung eines Zertifikats wählen Sie den entsprechenden Bewertenden Teilnehmer aus und lassen Sich die Teilnehmerdetails anzeigen.

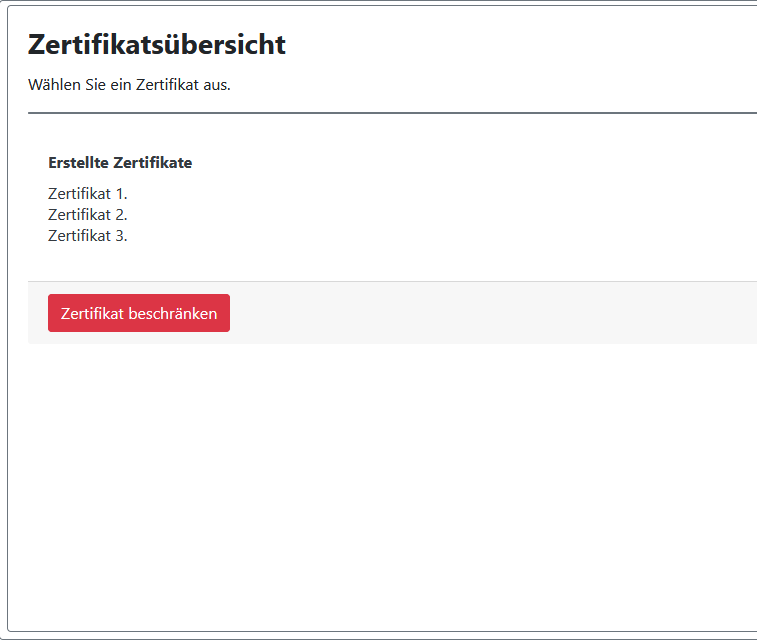


Nach der Auswahl eine bestanden Wahlqualifikation übertragen können Sie die Daten in das nebenstehende Zertifikat übertragen und das Produkt schließlich ausdrucken.

*A Anhang*

[**Abschluss**](#_bookmark34)

# Benutzerhandbuch (Seite 10)

* 1. **Zertifikatsansicht beschränken**

Zur Beschränkung eines Zertifikats wählen Sie diese einen oder mehrere erstellte Zertifikate aus und drücken den Button „Zertifikat beschränken“ um diese für einen bestimmte Fachrichtung zu kategorisieren.

**5. Glossar des Benutzerhandbuches**

|  |  |
| --- | --- |
| **Begriff** | **Erläuterung** |
| **MySQL** | **MySQL** ist das am verbreitetste Datenbankverwaltungs- system. Es ist eine kostenlose Software und für verschiedene Betriebssysteme verfügbar und bildet damit die Grundlage für  viele dynamische Webauftritte (siehe dieses Projekt). |

*A Anhang*

[**Abschluss**](#_bookmark35)

# Glossar des Abschlussprojektes (Seite 1)

|  |  |
| --- | --- |
| **Begriff** | **Erläuterung** |
| **AFP** | Die **AFP** oder auch *Akademie für Personalweiterbildung* ist ein 2018 in Köln eröffnetes Unternehmen zur Weiterbildung und Umschulung zum Fachinformatiker. Sie bietet zudem auch  Schulungen im weiteren IT Umfeld an. |
| **Backend** | Das **Backend** beschäftigt sich in diesem Projekt mit der  Datenverarbeitung und weiteren Funktionalität im Hintergrund. |
| **Bootstrap** | Das **Bootstrap**-Programmiergerüst enthält Vorlagen zur Darstellung im von Formularen, Buttons, Tabellen, Navigations- und weiteren Gestaltungselementen die im Projekt benutzt  werden. Es ist der derzeit beliebteste Bausatz für Websites. |
| **CSS** | Die **CSS**-Gestaltungssprache steht für die Erstellung von Websites und steht für *Cascading Style Sheet*. Mit ihr kann man Gestaltungsanweisungen erstellen die mit Auszeichnungssprachen  wie **HTML** in **Bootstrap** benutzt werden. |
| **Framework** | Das **Framework** ist ein Programmiergerüst, dass in der Softwaretechnik und in diesem Projekt zur Unterstützung der Teamarbeit verwendet wird. Es entwirft dabei eine bessere Entwicklung und Erstellung des Datenmodells was über einen Controller gesteuert wird und unteranderem die entwickelte  Website präsentiert. |
| **Frontend** | Das **Frontend** präsentiert in diesem Projekt die Verarbeitung  visuell auf einer Website zur Benutzung. |

*A Anhang*

[**Abschluss**](#_bookmark35)

# Glossar des Abschlussprojektes (Seite 2)

|  |  |
| --- | --- |
| **Begriff** | **Erläuterung** |
| **GitLAB** | Die **GitLAB**-Verwaltung wird bei Softwareprojekten zur Sicherung der Qualität benutzt. In diesem Projekt wurden dabei  die kontinuierliche Integration und Nutzerverwaltung benutzt. |
| **HTML** | Die **HTML**-Auszeichnungssprache steht für die Auszeichnung und Vernetzung von Websites und steht für *Hypertext Markup Language*. Mit ihr kann man Inhalte verknüpfen die mit  Gestaltungssprachen wie **CSS** in **Bootstrap** benutzt werden. |
| **IDE** | Die **IDE** steht für eine Entwicklungsumgebung in der Softwareprodukte entwickelt werden. Sie stellen bestimmte Werkzeuge zur Verfügung die dem Anwendungsentwickler zuarbeiten und auf Fehler aufmerksam machen. Zudem übernimmt es bestimmte Aufgaben und stellt Erweiterungen zur Verfügung. **IDEs** gibt es für nahezu alle Programmiersprachen  und Plattformen. |
| **Laravel** | Das **Laravel**-Programmiergerüst enthält Vorlagen zur **MVC** Erstellung unter **PHP** im **Backend**. Das bedeutet das mit diesem Gerüst ein Aufbau einer Website mit Funktionalität als Bausteine entworfen werden. Außerdem zählte es 2020 zu den Top10 der  meistbenutzten Frameworks (Platz 3). |

*A Anhang*

[**Abschluss**](#_bookmark35)

# Glossar des Abschlussprojektes (Seite 3)

|  |  |
| --- | --- |
| **Begriff** | **Erläuterung** |
| **Microsoft Visual Studio Code** | Der **Microsoft Visual Studio Code** ist ein Quelltext-Editor von Microsoft und stellt Funktionen wie Syntaxhervorhebung, Code- Faltung, Debugging, Autovervollständigung und lokale Versionsverwaltung. Es wird in diesem Projekt als  Entwicklungsumgebung benutzt. |
| **MVC** | **MVC** ist ein strukturiertes Software-Architekturmuster und wird für eine flexible Entwerfung eines Programms genutzt, das spätere Änderung oder Erweiterung erleichtert und eine Wiederverwendbarkeit der einzelnen Komponenten ermöglicht.  Es ist dann zum Beispiel möglich, eine Anwendung zu schreiben, die dasselbe Modell nutzt und es dann für Windows, Mac, Linux oder für das Internet zugänglich macht. Die Umsetzungen nutzen dasselbe Modell, nur Controller und View müssen dabei jeweils  neu implementiert werden. |
| **MySQL** | **MySQL** ist das am verbreitetste Datenbankverwaltungssystem. Es ist eine kostenlose Software und für verschiedene Betriebssysteme verfügbar und bildet damit die Grundlage für viele dynamische  Webauftritte. |
| **PHP** | Die **PHP**-Skriptsprache wird für die Erstellung dynamischer Webseiten oder Webanwendungen verwendet und bedeutet komplett *Personal Hypertext Preprocessor*. Sie zeichnet sich durch breite Datenbankunterstützung und Internet- Protokolleinbindung sowie die Verfügbarkeit zahlreicher  Funktionsbibliotheken aus. |

*A Anhang*

[**Abschluss**](#_bookmark35)

* 1. **Glossar des Abschlussprojektes (Seite 4)**

|  |  |
| --- | --- |
| **Begriff** | **Erläuterung** |
| **TE** | Die **TE** oder auch **Train & Education** ist ein 2014 in Köln eröffnetes Unternehmen zur Personalweiterbildung mit mehreren Seminaren im IT-Bereich zur Qualifizierung. Sie bietet unteranderem auch Schulungen im kaufmännischen und IT-  technischen Bereich an. |
| **Versionsverwaltung** | Die **Versionsverwaltung** ist ein System, das zur Erfassung von Änderungen an Dokumenten oder Dateien verwendet wird. Alle Versionen werden in einem Archiv mit Zeitstempel und Benutzerkennung gesichert und können später wiederhergestellt werden. **Versionsverwaltungssysteme** werden typischerweise in der Softwareentwicklung eingesetzt, um Quelltexte zu verwalten. Versionsverwaltung kommt auch bei Büroanwendungen zum  Einsatz. |