Internetserver-programmierung

Projektbericht im Rahmen der Portfolio-Prüfung SoSe 2020

**. Zeitraum: 20.07.2020 – 16.08.2020**

**Gruppe 3**

**Gruppenmitglied: Abdessamad Aouam**

**Gruppenmitglied: Tonio Anantharaj**

Inhaltsverzeichnis

1. [Spezifikation 3](#_Toc47095417)

[1-1 Benutzergruppen 3](#_Toc47095418)

[1-2 Funktionale Anforderungen 4](#_Toc47095419)

[1-3 Nicht Funktionale Anforderungen 5](#_Toc47095420)

1. [Entwurf 6](#_Toc47095421)

[2-1 Klassendiagramm 6](#_Toc47095422)

[2-2 Entwurfmustereinsatz 8](#_Toc47095422)

1. [Implementierung 9](#_Toc47095421)

[3-1 Screenshots 9](#_Toc47095422)

[3-2 Frameworks 12](#_Toc47095422)

1. [Zusammenfassung 13](#_Toc47095421)

[4-1 Aufgabenteilung 13](#_Toc47095422)

[4-2 Ausblick 13](#_Toc47095422)

[Abbildungsverzeichnis 14](#_Toc47095468)

1. **Spezifikation**

Eine Planung von Kursen und Prüfungen ist eine große Herausforderung: ab 200 Kursen im Jahr führen unübersichtliche Arbeitsabläufe nicht nur zu einem enormen Zeitaufwand, sondern letztlich auch zu suboptimaler Planung und Durchführung der Kurse.

* 1. **Benutzergruppen**

**Teilnehmer**:

Sind Personen, die sich über das Internet an einem Kurs oder an einer Prüfung anmelden und eine intuitiv bedienbare Webanwendung sowie Information zu den angebotenen Kursen/Prüfungen haben wollen. Unsere Ziele an den Benutzer: einfache Bedienung, schnelle Zielerreichung und Zufriedenstellung.

**Verwalter:**

Verwalter sind Personen, die die Kurse verwalten. Das Ziel der Verwalter ist das unkomplizierte Einpflegen/ Verwalten von Daten in die vorhandene Datenbank.

**Kunde**:

Der Auftraggeber ist der Kunde. Das Ziel des Kunden ist eine funktionierende Webanwendung, die den Kunden hilft, einen Kurs/Prüfung zu verwalten. Der Kunde möchte einen Überblick über Nutzerstatistiken haben, sodass ein reibungsloser Ablauf, hohe Qualität und eine optimale Wirtschaftlichkeit des Kurses gewährleistet ist.

* 1. **Funktionale und nicht-funktionale Anforderungen**

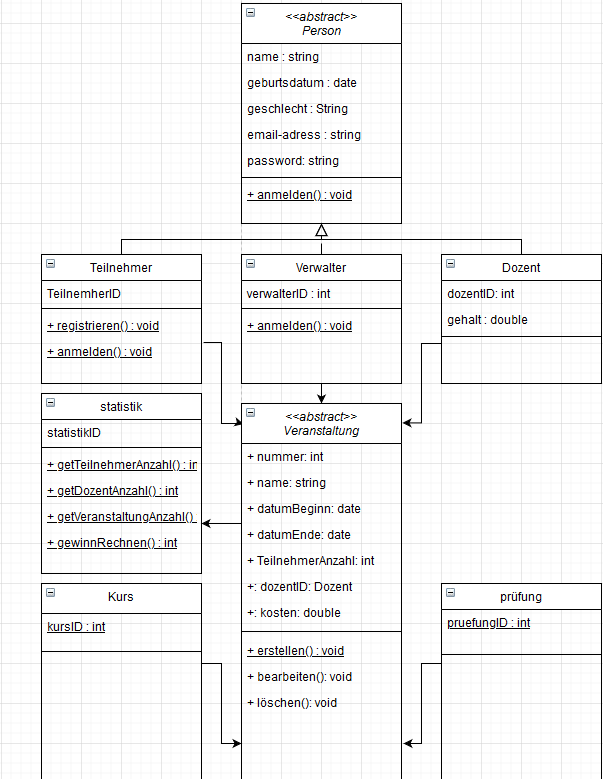
**Teilnehmer**:

* Kann sich eine Übersicht aller Kurse und Prüfungen anzeigenlassen und nach Auswahl eines Kurses/Prüfung mit detaillierter Ansicht inklusive Informationen (z.B. zu Gebühren) zur Anmeldung übergehen.
* Kann sich für einen Kurs/eine Prüfung anmelden, solange es keine zeitlichen Überschneidungen mit anderen Kursen oder Prüfungen gibt.
* Kann seine angemeldeten Kurse und Prüfungen einsehen (Übersicht).

**Verwaltung**:

* + Kann Kurse und Prüfungen anlegen/editieren/löschen.
  + Kann Dozenten Kurse zuweisen. Dozenten sind feste Datensätze, die aus der Datenbank gelesen werden können und haben feste Stundensätze/Lohn.
  + Hat eine Übersicht mit verschiedenen Statistiken zu:
  + Wie viele Teilnehmer gibt es insgesamt
  + Wie viele Dozenten unterrichten
  + Wie viele Kurse werden angeboten
  + Wie viele Prüfungen werden angeboten
  + Top/Flop Kurse mit den meisten/wenigsten Teilnehmern und Umsatz (Gebühren der Teilnehmer –Lohn der Dozenten)
  + Top/Flop Prüfungen mit den meisten/wenigsten Teilnehmern und Umsatz
  + Gesamtumsatz
  + Nicht rentable Dozenten (Umsatz < Teilnehmergebühren –Dozentenlohn)
  1. **nicht-funktionale Anforderungen**

1. **Entwurf**
   1. **Klassendiagramm**

****

**Person**:

Die abstrakte Klasse Person beschreibt die Eigenschaften einer Person. Die Attribute und Methoden, die allen Personen gemeinsam sind, werden in der Oberklasse Person gebündelt. Die Abstrakte Klasse wird in der Konkreten Unterklassen implementiert.

**Teilnehmer**:

Die konkrete Klasse Teilnehmer erbt von der abstrakten Superklasse Person Attribute und Methoden und implementiert die abstrakte Methode. Ein Teilnehmer kann eine oder mehreren Veranstaltungen buchen, solange keine zeitlichen Überschneidungen mit anderen Veranstaltungen gibt.

**Verwalter**:

Die konkrete Klasse Teilnehmer erbt von der abstrakten Superklasse Person Attribute und Methoden und implementiert die abstrakte Methode.

**Dozent**:

Die konkrete Klasse Teilnehmer erbt von der abstrakten Superklasse Person Attribute und Methoden und implementiert die abstrakte Methode.

**Verwaltung**:

Die abstrakte Klasse Veranstaltung beschreibt die Eigenschaften einer Veranstaltung. Die Attribute und Methoden, die allen Veranstaltungen gemeinsam sind, werden in der Oberklasse Person gebündelt.

**Kurs**:

Die konkrete Klasse Kurs erbt von der abstrakten Superklasse Veranstaltung Attribute und Methoden und implementiert die abstrakte Methode.

**Prüfung**:

Die konkrete Klasse Prüfung erbt von der abstrakten Superklasse Veranstaltung Attribute und Methoden und implementiert die abstrakte Methode.

**Statistik**:

Die Klasse Statistik berechnet anhand von Methoden verschiedene Statistiken. Die Statistiken werden von dem Verwalter angezeigt werden.

* 1. **Entwurfsmustereinsatz**

**Verhaltungsmuster – Schablone**

Die Verwendung der Schablonemethode ermöglicht es Unterklassen Kurs/Prüfung, sowie Teilnehmer/Dozent/Verwalter, bestimmte Schritte des Algorithmus zu implementieren, ohne die Struktur zu verändern. Die abstrakten Klassen werden sowohl konkrete als auch abstrakte Methoden definiert. In die Konkreten Methoden werden die Gemeinsamkeiten der Algorithmen implementiert. die Abstrakten Methoden dienen dazu, um unterschied der Algorithmen zu definieren, welche in die Subklassen implementiert wird.

Diese ermöglicht uns zwei wichtige Punkte:

* Die Verdopplung von Code wird vermieden.
* Die Kontrolle um Erweiterungen durch Unterklassen.

1. **Implementierung**
   1. **Screenshots**

**Ein Bild, das Screenshot, Computer, computer, sitzend enthält.

Automatisch generierte Beschreibung**

Abbildung 1: Kurs erstellen

Ein Bild, das Screenshot, sitzend, Computer, Monitor enthält.

Automatisch generierte Beschreibung

Abbildung 2: Prüfung erstellen

Ein Bild, das Screenshot, Computer, Monitor, computer enthält.

Automatisch generierte Beschreibung

Abbildung 3: Liste der Kurse

Ein Bild, das Screenshot, Computer, Monitor, computer enthält.

Automatisch generierte Beschreibung

Abbildung 4: Liste der Prüfungen

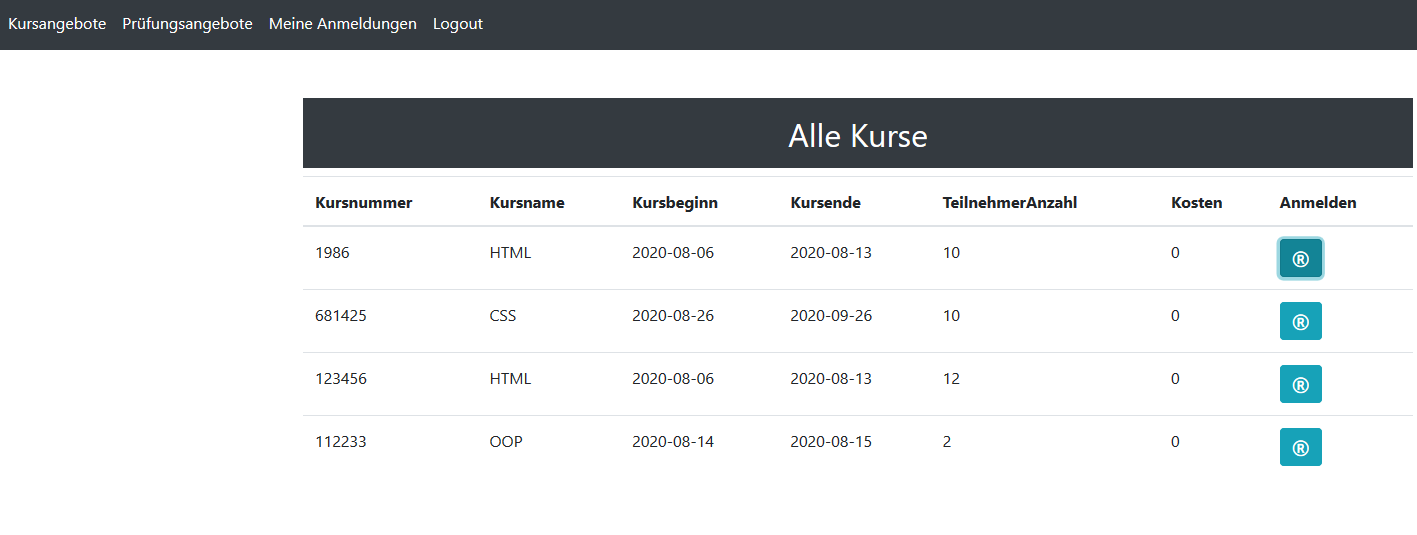


Abbildung 5: User Menü

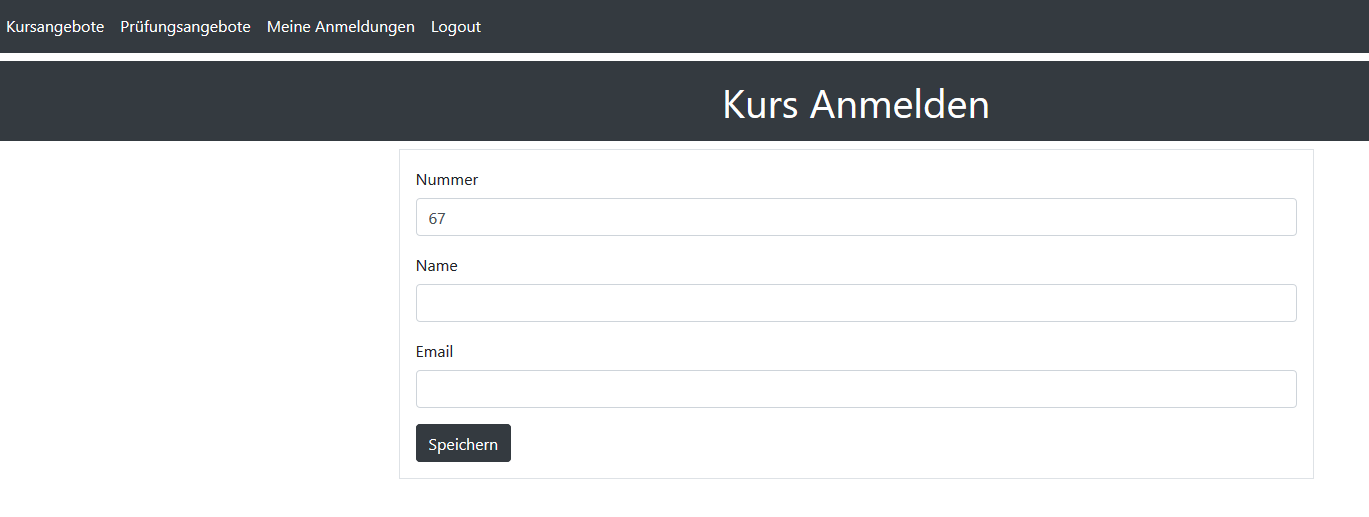


Abbildung 6: Kurs anmelden

* 1. **Genutzte Frameworks: Bootstrap**

Wir wollten uns intensiv mit der Programmierung der Anwendung beschäftigen, sodass wir für das Anwendungsdesign ein kostenfreies Framework (Bootstrap) verwendet haben. Bootstrap beinhält, im Gegensatz zu HTML, CSS und JavaScript, bereits viele Anwendungen

(Icons etc.). Die Entwicklungszeit wird durch dieses Framework erheblich verkürzt. Des Weiteren ist dieses Framework einfach auszuführen. Bei der Zusammenarbeit an einem Projekt durch mehrere Entwickler, ist Bootstrap das passende Framework.

1. **Zusammenfassung**
   1. **Aufgabenteilung**

|  |  |
| --- | --- |
| **Abdessamad Aouam** | **Tonio** |
| * Kurs/Prüfung erstellen * Kurs/Prüfung ändern * Kurs/Prüfung löschen * Liste der Kurse anzeigen * Liste der Prüfungen anzeigen * Für einen Kurs anmelden * Für eine Prüfung anmelden * Webseite designen * Entwurfmuster erstellen * Doku schreiben |  |

* 1. **Ausblick**

# Abbildungsverzeichnis

[Abbildung 1: Kurs erstellen 9](#_Toc47095497)

[Abbildung 2: Prüfung erstellen 10](#_Toc47095498)

[Abbildung 3: Liste der Kurse 10](#_Toc47095499)

[Abbildung 4: Liste der Prüfungen 11](#_Toc47095500)

[Abbildung 5: User Menü 11](#_Toc47095501)

[Abbildung 6: Kurs anmelden 12](#_Toc47095502)