

Inhaltsverzeichnis

[Abbildungsverzeichnis 3](#_Toc88668039)

[Tabellenverzeichnis 3](#_Toc88668040)

[Abkürzungsverzeichnis 3](#_Toc88668041)

[1. Einleitung 4](#_Toc88668042)

[1.1 Projektbeschreibung 4](#_Toc88668043)

[1.2 Projektziel 4](#_Toc88668044)

[1.3 Projekbegründung 4](#_Toc88668045)

[1.4 Projektschnitstellen 4](#_Toc88668046)

[1.5 Projektumfeld 4](#_Toc88668047)

[2. Projektplanung 4](#_Toc88668048)

[2.1 Projektphasen 4](#_Toc88668049)

[2.2 Vorgehensmodell 5](#_Toc88668050)

[2.3 Zeitplanung 5](#_Toc88668051)

[2.4 Budgetplanung 5](#_Toc88668052)

[3. Projektvorbereitung 6](#_Toc88668053)

[3.1 Ist-Analyse 6](#_Toc88668054)

[3.2 Anwendungsfälle 6](#_Toc88668055)

[3.3 Entwicklungsplattformen 7](#_Toc88668056)

[3.4 Entwurf der Datenstruktur 7](#_Toc88668057)

[3.5 Entwurf der Benutzeroberfläche 7](#_Toc88668058)

[3.6 Lastenheft 7](#_Toc88668059)

[3.7 Pflichtenheft 8](#_Toc88668060)

[4. Implementierung 8](#_Toc88668061)

[4.1 Implementierung der Benutzeroberfläche 8](#_Toc88668062)

[4.2 Implementierung der Datenstrukturen 8](#_Toc88668063)

[4.3 Implementierung der Programmierlogik 8](#_Toc88668064)

[4.4 Testen der Anwendung 8](#_Toc88668065)

[5. Fazit 9](#_Toc88668066)

[5.1 Soll-Ist-Vergleich 9](#_Toc88668067)

[5.2 Lern-Erfolg 9](#_Toc88668068)

[5.3 Ausblick 9](#_Toc88668069)

[6. Anhang 10](#_Toc88668070)

[6.1 ERM-Datenmodell ² 10](#_Toc88668071)

[6.2 Mockups 1,2 11](#_Toc88668072)

[6.3 Verwendete Ressourcen 15](#_Toc88668073)

[6.4 Use-Case-Diagramm ³ 16](#_Toc88668074)

[6.5 Screenshots der Anwendung 1 17](#_Toc88668075)

[6.6 Projektantrag 21](#_Toc88668076)

# Abbildungsverzeichnis

[Abbildung 1 ERM-Datenmodell 7](#_Toc88325346)

[Abbildung 2 Mockup Schüler-Login 8](#_Toc88325347)

[Abbildung 3 Mockup Lehrer-Login 9](#_Toc88325348)

[Abbildung 4 Mockup Lehrer Verwaltung Klausur 10](#_Toc88325349)

[Abbildung 5 Mockup Lehrer Verwaltung Klasse 10](#_Toc88325350)

[Abbildung 6 Mockup Lehrer Verwaltung Mein Konto 11](#_Toc88325351)

[Abbildung 7 Mockup Klausurtermine 11](#_Toc88325352)

[Abbildung 8 Mockup Admin 12](#_Toc88325353)

[Abbildung 9 Use-Case-Diagramm 13](#_Toc88325354)

[Abbildung 10 Anwendung-Screenshot Schüler Login 14](#_Toc88325355)

[Abbildung 11 Anwendung-Screenshot Lehrer-Login 14](#_Toc88325356)

[Abbildung 12 Anwendung-Screenshot Lehrer Verwaltung Klausur 15](#_Toc88325357)

[Abbildung 13 Anwendung-Screenshot Lehrer Verwaltung Klasse 15](#_Toc88325358)

[Abbildung 14 Anwendung-Screenshot Lehrer Verwaltung Mein Konto 16](#_Toc88325359)

[Abbildung 15 Anwendung-Screenshot Klausurtermine 16](#_Toc88325360)

[Abbildung 16 Anwendung-Screenshot Admin 17](#_Toc88325361)

# Tabellenverzeichnis

[Tabelle 0‑I Zeitplanung 3](#_Toc88325000)

[Tabelle 0‑I Projektvorbereitung 4](#_Toc88325001)

# Abkürzungsverzeichnis

|  |  |
| --- | --- |
| **CSS** | Cascading Style Sheets |
| **ERM** | Entity-Relationship-Modell |
| **GUI** | Graphical User Interface |
| **HTML** | Hypertext Markup Language |
| **JS** | JavaScript |
| **MVC** | Model View Controller |
| **SQL** | Structured Query Language |
| **UI** | User Interface |
| **URL** | Uniform Resource Locator |

4

# 1. Einleitung

Dieses Dokument simuliert eine Dokumentation für ein IHK-Abschlussprojekt. Es wurde im Rahmen einer Ausbildung zum Fachinformatiker für Anwendungsentwicklung als Vorbereitung auf die wirkliche Abschlussprüfung der IHK erstellt.2

## 1.1 Projektbeschreibung

Die Schüler der Brühlwiesenschule Hofheim Jason Preziuso, Joshua Zacherl, Kadri Kalesi und Jaime Gamero möchten einen Klausurenplaner für die genannte Schule erstellen. Bei dem Klausurenplaner handelt es sich um eine digitale Web-Applikation zum Verwalten und Planen der bevorstehenden Klausuren für die einzelnen Schulklassen. Die Lehrer haben die Möglichkeit, Klausuren dem Planer hinzuzufügen und den Schülern Zugriff darauf zu gewähren, damit die Schüler sehen, wann die Klausuren stattfinden. 2

## 1.2 Projektziel

Das Ziel des Projekts besteht darin, eine digitale Web-Applikation für die Brühlwiesenschule in Hofheim zu entwickeln, welche die Verwaltung von bevorstehenden Klausuren so unkompliziert wie möglich macht. Wichtig ist, dass Schüler und Lehrer von jedem Ort, der über eine Internetverbindung verfügt, darauf zugreifen können. 2

## 1.3 Projekbegründung

Aktuell werden die Klausuren in der Klasse 12ITa in Moodle angepinnt. Andere Klassen verwenden einen Zettel mit Datum und Fach, welcher an die Wand des Klassenzimmers geheftet wird. Eine gute Lösung ist dies nicht, weil es sehr unübersichtlich ist. Ebenso denkt nicht immer jeder Lehrer daran, diese Liste zu aktualisieren. Außerdem ist der Zettel an der Wand nur vor Ort in der Brühlwiesenschule einsehbar. 2

## 1.4 Projektschnitstellen

Die Schnitstelle wurde zwischen der Datenbank und dem Back-End mit Node.js hergestellt. Das Front-End wurde mit dem Back-End in Javascript verknüpft.3

## 1.5 Projektumfeld

Das Projekt werden die vier angegebenen Schüler gemeinsam umsetzen. Die Umsetzung des Projekts erfolgt größtenteils in der Brühlwiesenschule vor Ort. Teilaufgaben werden aber auch nach der Schule zu Hause oder während der Arbeitszeit im jeweiligen Betrieb erledigt. 2

# 2. Projektplanung

## 2.1 Projektphasen

Für die Umsetzung des Projektes hat Herr Grüning 72 Stunden vorgeschrieben. Zu Anfang des Projekts, wurde eine Aufteilung der verschiedenen Phasen des Prozesses der gesamten Entwicklung niedergeschrieben. Eine grobe Zeitplanung mit jeweiligen Prozessschritten befindet sich unter dem Punkt „Zeitplanung“ 1

## 2.2 Vorgehensmodell

Für die Entwicklung des Projekts wurde die agile Methodik Scrum gewählt. Die benötigten Arbeitspakete für diese Methode wurden auf einem Kanban Board angelegt. Es wurden Verantwortlichkeiten für die einzelnen Arbeitspakete festgelegt und zwei Teams gebildet. Das Front-End-Team hat mithilfe von HTML, CSS und JavaScript die Weboberfläche und einzelne Funktionalitäten umgesetzt. Das Back-End-Team hat für die Entwicklung die Technologien MySQL und Node.js genutzt. 3

## 2.3 Zeitplanung

Die untenstehende Zeitplanung beläuft sich auf insgesamt 72 Stunden.

Tabelle 0‑I Zeitplanung

|  |  |  |
| --- | --- | --- |
|  | **Verantwortlichkeit(en)** | **Dauer (in Stunden)** |
| **Dokumentation** |  | **18** |
| Projektbeschreibung | Team | 5 |
| Lastenheft | Jason | 4 |
| Installationsanleitung für Admins | Kadri | 3 |
| Benutzerhandbuch für Benutzer | Joshua | 4 |
| Team-Protokolle | Jason | 2 |
| **Front-End** |  | **20** |
| Grafische Oberfläche aller Unterseiten | Joshua | 10 |
| Javascript Funktionalitäten | Jason | 10 |
| **Back-End** |  | **20** |
| Implementierung der API | Jaime | 10 |
| Implementierung der Datenbank und Tabellen | Kadri | 5 |
| MySQL-Statements schreiben | Kadri | 5 |
| **Testen** |  | **14** |
| Test und Qualitätssicherung | Team | Je 3 |
| Test-Protokolle | Jaime | 2 |

1

## 2.4 Budgetplanung

Eventuell anfallende Kosten entstehen für das Programm Balsamiq Mockups 3. Damit kann man sich Mockups erstellen. Die Web-App würde 9 $ im Monat oder 90 $ im Jahr kosten. Die Kosten entfallen, da Kadri Kalesi eine Lizenz dafür besitzt. Weitere Kosten, die entstehen, sind die Serverkosten, welche aber die Brühlwiesenschule übernimmt, da die Server bereits in der Schule vorhanden sind. 4

# 3. Projektvorbereitung

In der folgenden Tabelle sind die verschiedenen Phasen der Projektausführung aufgeführt.

Tabelle 0‑I Projektvorbereitung

|  |  |
| --- | --- |
| **Nr.** | **Punkte** |
| **1** | Ist-Analyse |
| **2** | Design Anwendungsfälle |
| **3** | Auswahl der Entwicklungsplattform |
| **4** | Entwerfen des Architekturdesigns |
| **5** | Modellierung der Datenstruktur |
| **6** | Entwerfen der Benutzeroberfläche |
| **7** | Implementieren der Datenstruktur |
| **8** | Implementieren der Benutzeroberfläche |
| **9** | Implementierung der Geschäftslogik |
| **10** | Durchführen von Tests |
| **11** | Deployment der Anwendung |
| **12** | Abnahme der Anwendung |
| **13** | Erstellen der Projektdokumentation |

4

## 3.1 Ist-Analyse

Wie in der Projektbeschreibung erwähnt, wird das Projekt eine Webanwendung. Webanwendungen sind sehr beliebt und optimal für unser Projekt.

Die aktuelle Lage ist, dass man ein Ausgedrucktes Papier im Klassenraum der entsprechenden Klassen an die Wand heftet. Das Papier beinhaltet eine Tabelle mit den Spalten: Fach und Datum. Die Lehrer tragen händisch die Daten für die Klausur auf den Zettel ein. Man sieht anhand des Zettels an der Wand, zu welchem Datum man welche Klausur schreibt. Die Schwächen darin liegen leider, dass man nicht von überall darauf zugreifen kann und die allgemeine Verwaltung nicht Nutzerfreundlich ist. Man muss immer im Klassenraum sein, um die Liste zu aktualisieren oder die Informationen abzulesen. Es kann vorkommen, dass sich z.B. eine Klausur verschiebt und die Schüler es zu spät bemerken durch den Zettel an der Wand. Das ist alles Suboptimal, am besten müsste alles direkt abrufbar sein und aktualisiert werden können, egal von welchem Standort aus. In dem Fall ist eine Webanwendung am besten geeignet. Lehrer könnten von überall, ob Unterwegs oder von Zuhause aus der Liste aktualisieren oder editieren. Schüler könnten immer darauf zugreifen, ohne darauf angewiesen zu sein, im Klassenraum sein zu müssen. Das Beste Beispiel ist dafür die Klasse 12Ita, die nur im Blockunterricht anwesend ist und ansonsten nie Zugriff auf den Klassenraum haben und teilweise bis zu zwei Monaten im Betrieb arbeiten (Auszubildende). 4

## 3.2 Anwendungsfälle

Für die Übersicht, wie die Anwenderinnen und Anwender mit der Anwendung arbeiten und welche Fälle aus Endanwendersicht abgedeckt werden müssen, wurden ein Use-Case-Diagramm erstellt. Im Fall des Projektes gehören lediglich Administrator\*innen, Studenten\*innen und Lehrer\*innen zu den Akteuren. ³

## 3.3 Entwicklungsplattformen

* Apache NetBeans IDE 12.1 (<https://netbeans.apache.org/>)
* **Visual Studio Code** <https://code.visualstudio.com/>
* **Insomnia** <https://insomnia.rest/>
* **Atom** <https://atom.io/>
* **Mamp** <https://www.mamp.info/en/mac/>

**1**

## 3.4 Entwurf der Datenstruktur

Bevor die Datenbank angelegt werden kann wurde ein ERM erstellt, welches im Anhang sichtbar ist. Dieses gibt Auskunft über alle benötigten Datenbanktabellen mit deren Attributen und den Beziehungen untereinander. 2

## 3.5 Entwurf der Benutzeroberfläche

Die Mockups für die Weboberfläche wurden als Erstes in Form von Skizzen auf Papier erstellt. Beim Erstellen der Skizzen musste beachtet werden, dass alle geforderten Daten aus der Projektbeschreibung, die in der Datenbank gespeichert werden sollen, auch auf den verschiedenen Unterseiten der Weboberfläche in den entsprechenden Tabellen angezeigt werden. Außerdem müssen ausreichend Knöpfe vorhanden sein, um alle Funktionalitäten abdecken zu können. Diese Skizzen wurden dann in das Programm Balsamiq Mockups 3 übertragen, um die Mockups in digitaler Form dem Projektantrag beifügen zu können. Nach Besprechung des Projektantrags mit dem Auftraggeber wurden noch Anpassungen am Mockup vorgenommen, um allen Anforderungen gerecht zu werden. 1,2

## 3.6 Lastenheft

Der Klausurenplaner soll die Möglichkeiten bieten, Klausurtermine erstellen, ändern und löschen zu können. Dabei sollen folgende Attribute enthalten sein:

- Datum & Uhrzeit bzw. Schulstunde(n)

- Klassendaten

- Fach

- Raum

- Klausurthemen (Kann-Feld)

Klassendaten können eingepflegt werden mit einem Lehrer-Account. Dieser Account soll folgende Attribute besitzen:

- Klasse

- Passwort

- Import-Funktion für Klassensätze (Klasse mit Passwort)

- Reset-Funktion für alle Termine eines wählbaren Zeitraums

Die User-Verwaltung sollte folgende Dinge enthalten:

- Lehrer und/oder Admin mit Namen, Mail, Passwort (Lehrer können auch Admin sein)

- Import-Funktion für Lehrerdaten mit Passwort

Außerdem sollten folgende allgemeine Anforderungen erfüllt werden:

- Browser-unabhängig aufrufbar

- Model und View müssen trennbar sein, sodass die View austauschbar ist

- Responsive Design für Smartphone, Tablet und PC

- Hilfe-Bereich soll aufrufbar sein, der Funktionalität und Bedienung erklärt

- Logo soll stets sichtbar sein 2

## 3.7 Pflichtenheft

Wir verpflichten uns, die angegebenen Forderungen aus dem Lastenheft entsprechend umzusetzen. 3

# 4. Implementierung

## 4.1 Implementierung der Benutzeroberfläche

Die Benutzeroberfläche konnte anhand der Mockups, welche unter Entwurf der Benutzeroberfläche beschrieben sind, mit kleineren Abweichungen implementiert werden. Dem Textfeld „Klasse“ auf der Seite Schüler-Login wurde noch ein Platzhalter-Text hinzugefügt, um den Benutzern eine Beispieleingabe zu zeigen. Die eingebundenen Tabellen wurden noch durch Angabe der dargestellten Zeilen erweitert und die Auswahl der Anzahl der Tabellenreihen wurde ermöglicht. Speziell bei der Tabelle auf der Seite Lehrer-Verwaltung wurde noch die Spalte „Lehrer“ hinzugefügt. Auf derselben Seite wurde ein Eingabefeld entfernt, da zur Eingabe des Datums ein Feld ausreichend ist.

Damit die Seiten dem Responsive Webdesign unterliegen, wurde das Framework Bootstrap von Twitter verwendet. Kleinere Anpassungen wurden mithilfe von CSS durchgeführt. Die Funktionalitäten für alle Knöpfe und die Anzeige der Daten in den Tabellen aus der Datenbank wurden mit JavaScript umgesetzt. ²

## 4.2 Implementierung der Datenstrukturen

Bei der Implementierung der Datenbankstruktur, wurde sich an dem ERM (Abbildung 1 ERM-Datenmodell) orientiert. Alle Entitäten mit ihren Attributen wurden über PhpMyAdmin in die Datenbank eingefügt. 3

## 4.3 Implementierung der Programmierlogik

Für die Implementierung der Logik wurde ein kontrolliertes Ansichtsmodell (MVC) erstellt, in dem die Schichten getrennt sind. Es wurde eine Middleware-Schicht (API) implementiert, die für die Verbindung des Front-Ends mit dem Backend zuständig ist. Wir haben uns für eine zweischichtige API-Architektur entschieden, bei der eine Schicht die Endpunkte (Rest-Methoden) und die andere die zur Datenbank aufgerufenen Funktionen enthält.3

## 4.4 Testen der Anwendung

Während der Implementierungsphase, wurden auch Testfälle in geschrieben, um die Implementierten Methode auf Fehler zu untersuchen. Alle Methoden wurden auf Funktionalität und Sicherheit geprüft.4

# 5. Fazit

## 5.1 Soll-Ist-Vergleich

Die meisten geforderten Funktionalitäten des Projekts wurden umgesetzt. Aufgrund Krankheit und Doppelbelastung von zwei Teammitgliedern durch die vorgezogene Abschlussprüfung wurden zwei Funktionen nicht umgesetzt. Auf der Seite Lehrer-Verwaltung ist es so nicht möglich Klausuren in einem bestimmten Zeitraum zu löschen. Außerdem fehlt die Funktionalität zum Hochladen einer CSV-Datei, um Klassen- und Lehrerdaten zu importieren. Das erwähnte Kann-Feld Klausurthemen im Lastenheft wurde implementiert. So ist es möglich, beim Eintragen einer Klausur die Themen anzugeben. Diese sehen dann die Schüler auf der Seite Klausurtermine. 4

## 5.2 Lern-Erfolg

Die Umsetzung des Abschlussprojektes war für die Autoren eine bereichernde Erfahrung. Durch die intensive Auseinandersetzung mit vielen komplexen technischen Mitteln konnten die bereits bestehenden Kenntnisse der Softwareentwicklung eingesetzt, aber auch erweitert werden. Besonders im Bereich der eigenständigen Planung und Herangehensweise wurden viele neue Kenntnisse erworben. In diesem Projekt sind den Autoren einmal mehr deutlich geworden, wie wichtig die Kommunikation mit dem Projektleiter ist, um die Akzeptanzkriterien der Kundenansprüche ordnungsgemäß zu erfüllen. Nach der Ansicht der Autoren hatte das Projekt in vielerlei Hinsicht einen großen Mehrwert in Bezug auf den Ausbau seiner praktischen Kenntnisse der Softwareentwicklung. 1

## 5.3 Ausblick

In der vorgegebenen Zeit konnte eine lauffähige Erweiterung der Anwendung implementiert werden. In der nächsten Version können die fehlenden Funktionalitäten ergänz werden:

* Löschen von Klausuren in einem bestimmten Zeitraum
* Import einer CSV-Datei mit Lehrer und Schülerdaten 4

# 6. Anhang

## 6.1 ERM-Datenmodell ²

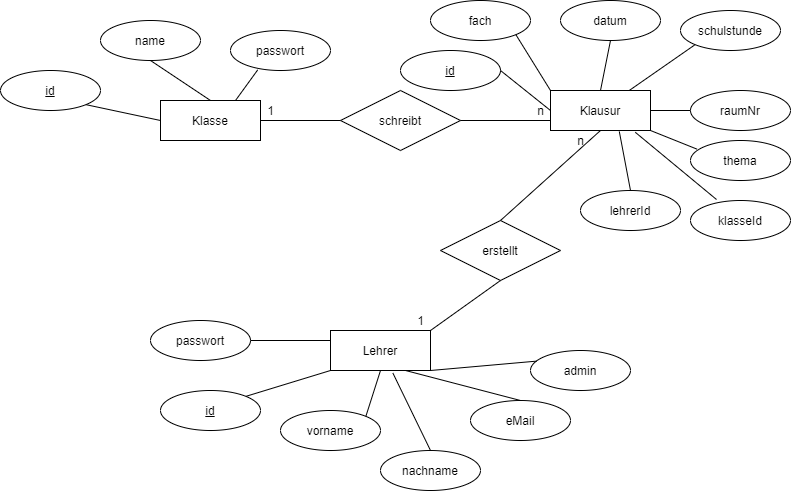


Abbildung 1 ERM-Datenmodell

## 6.2 Mockups 1,2

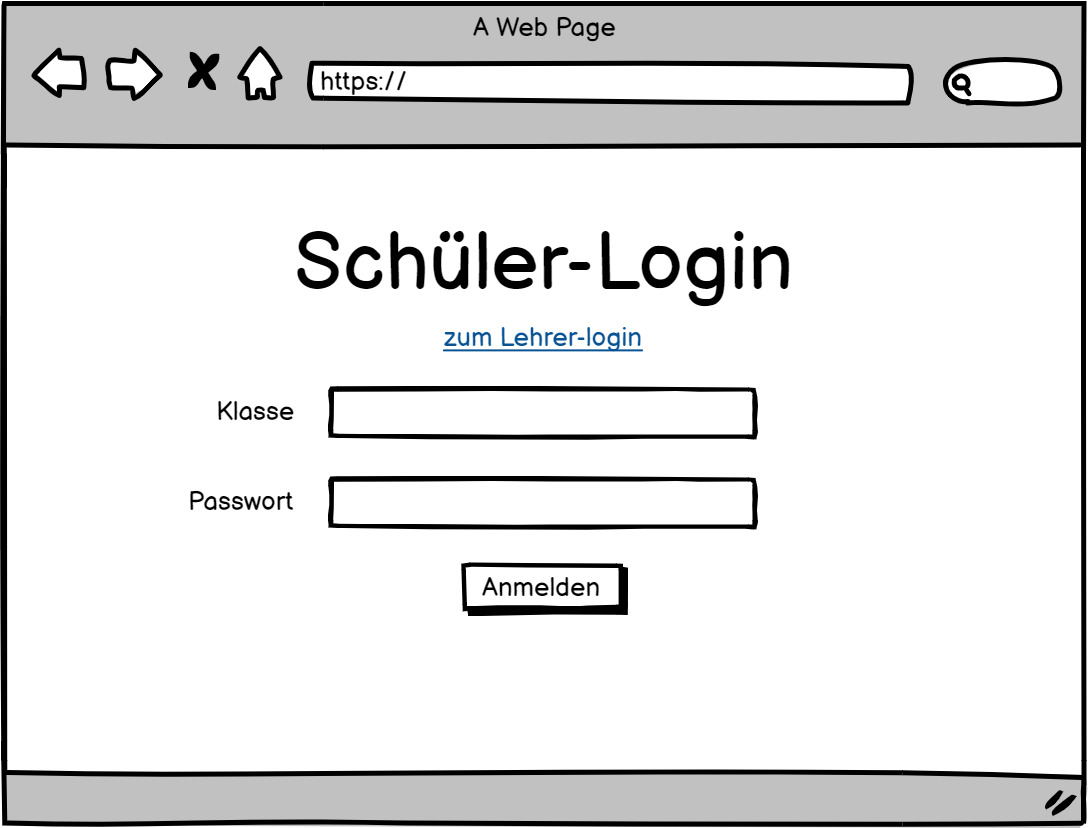


Abbildung 2 Mockup Schüler-Login

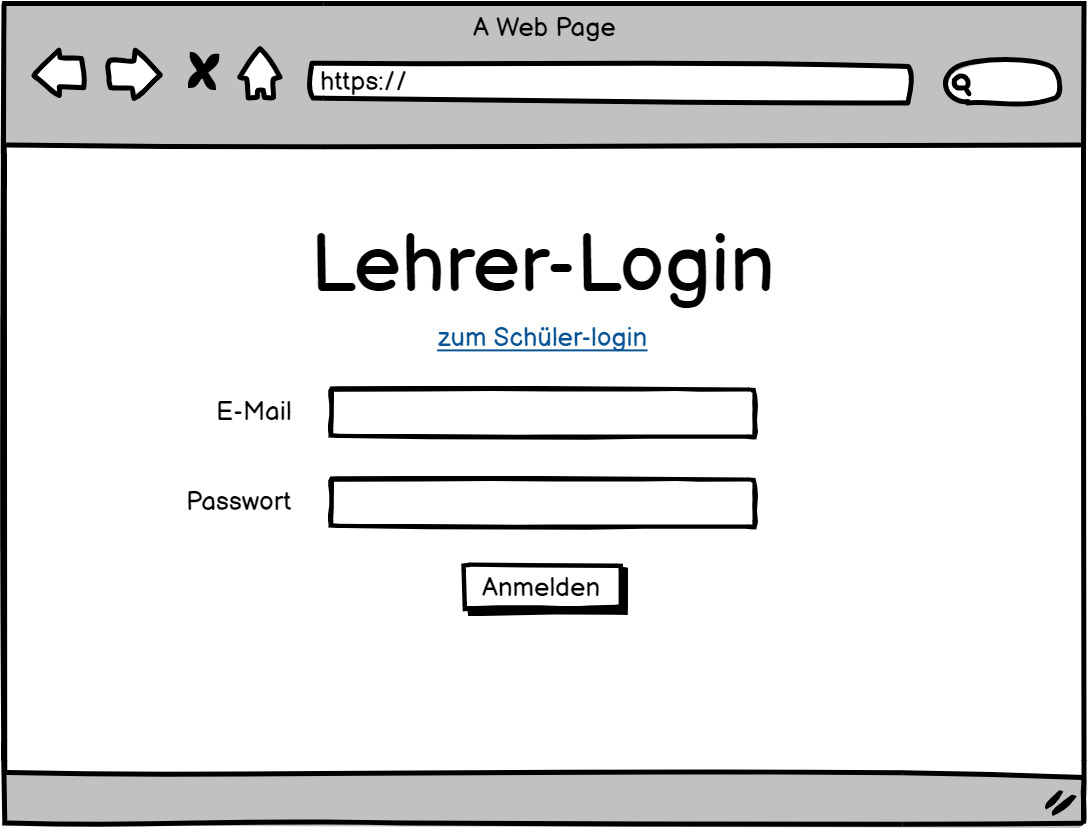


Abbildung 3 Mockup Lehrer-Login

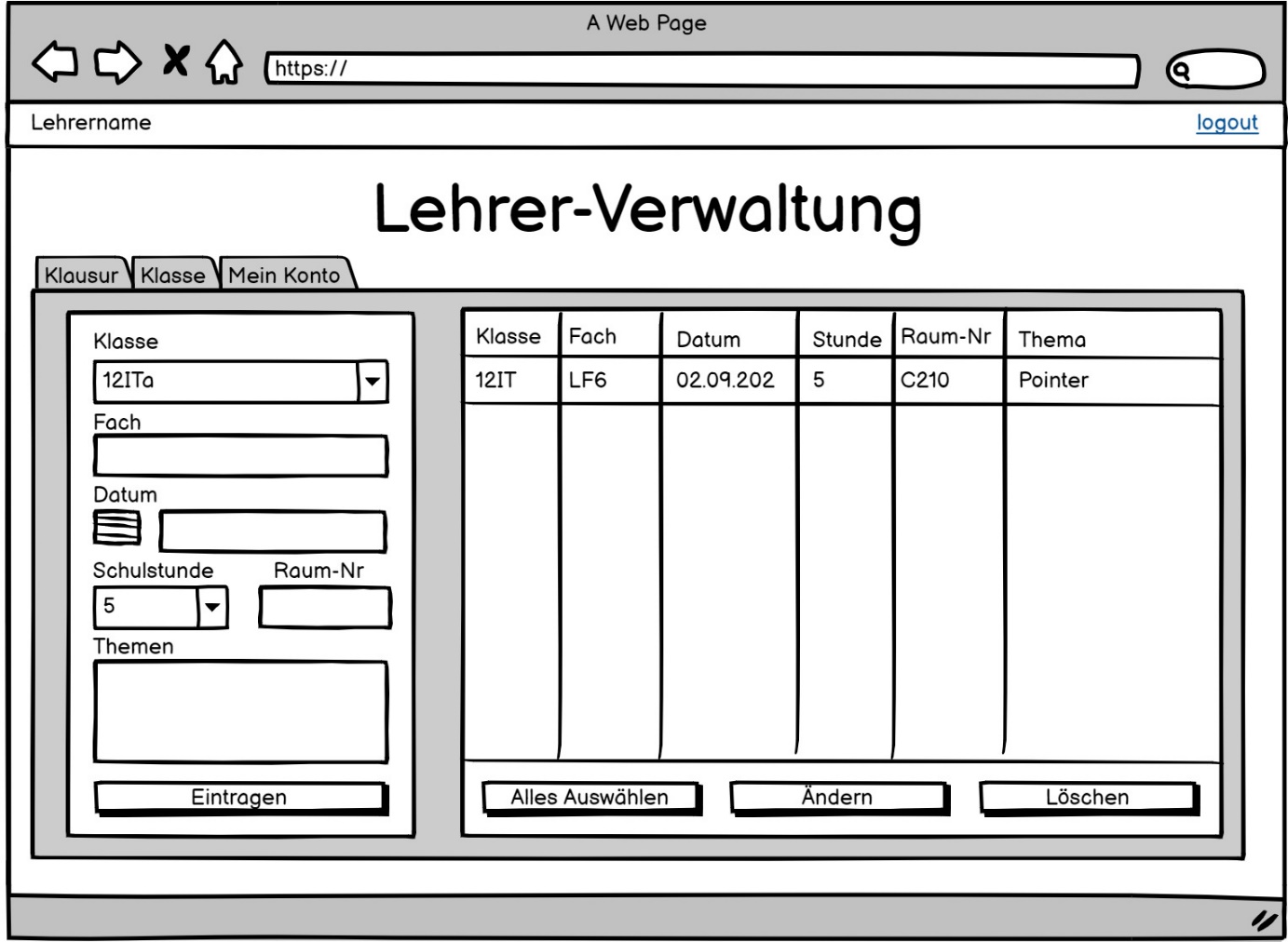


Abbildung 4 Mockup Lehrer Verwaltung Klausur

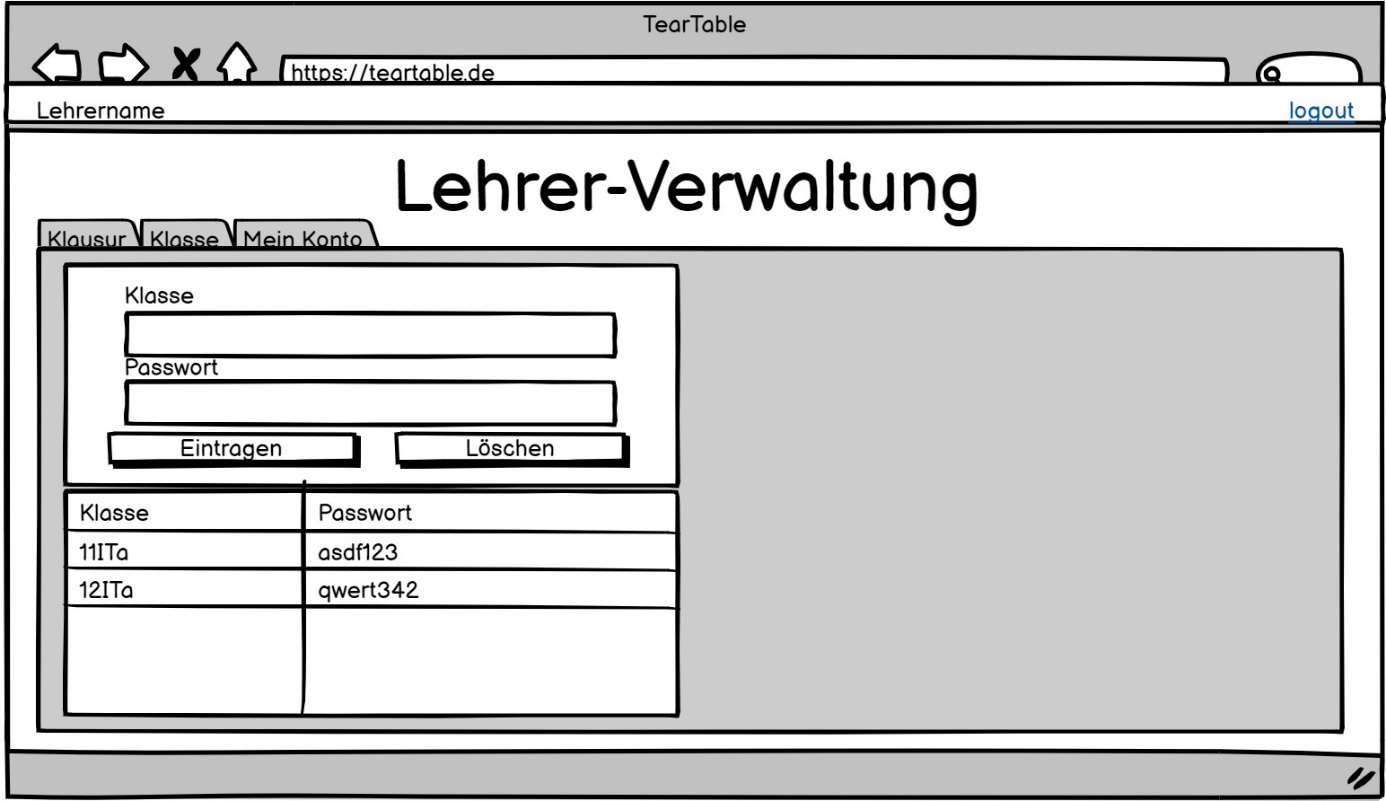


Abbildung 5 Mockup Lehrer Verwaltung Klasse

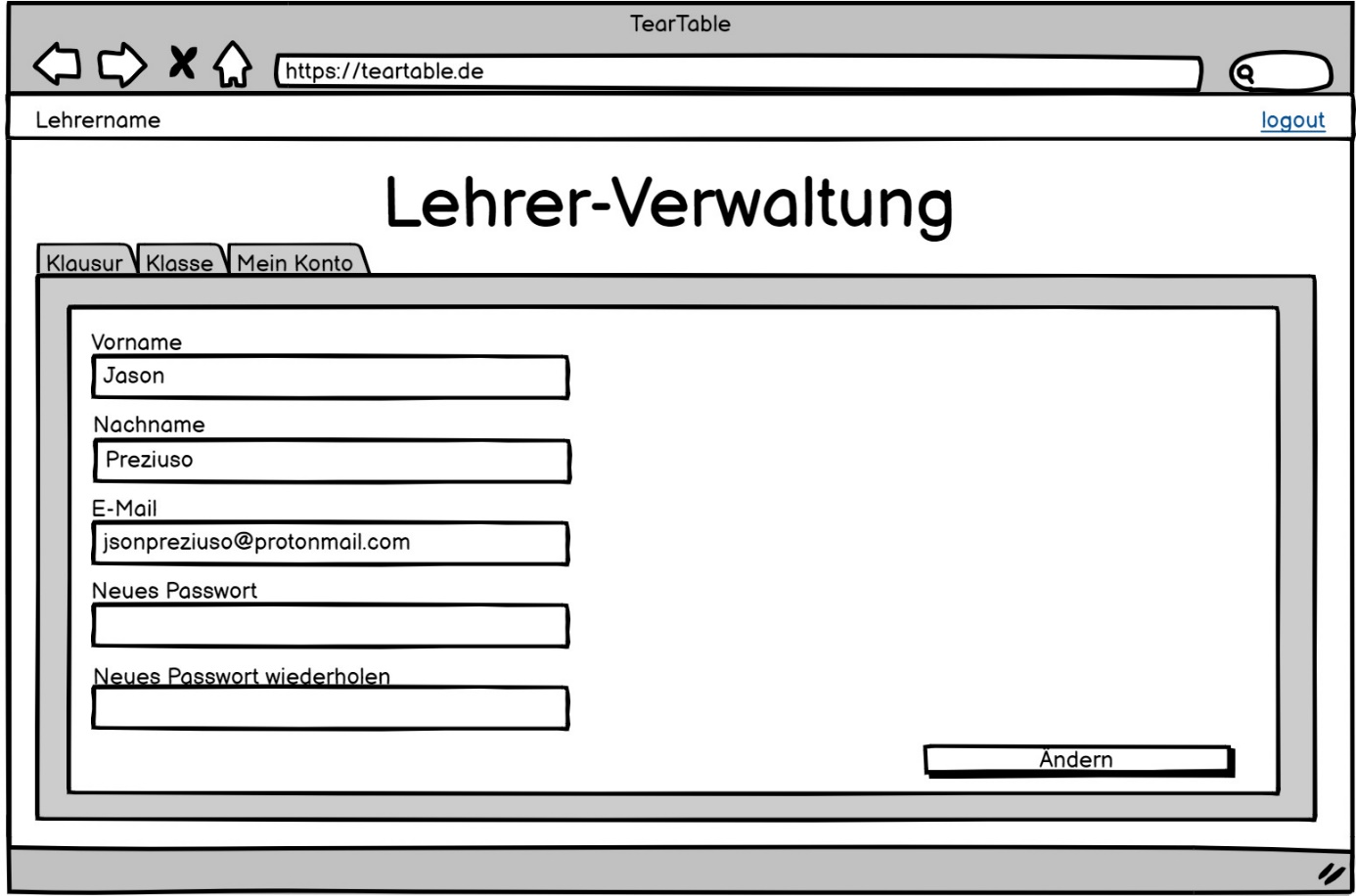


Abbildung 6 Mockup Lehrer Verwaltung Mein Konto

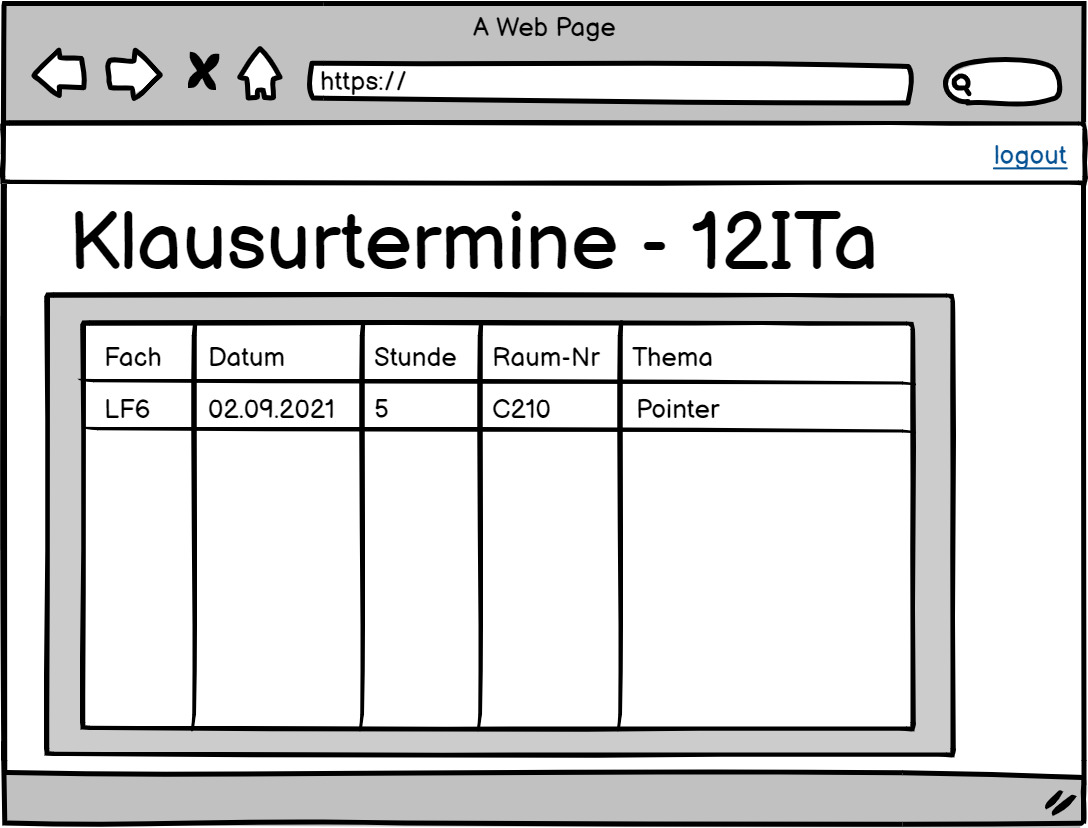


Abbildung 7 Mockup Klausurtermine

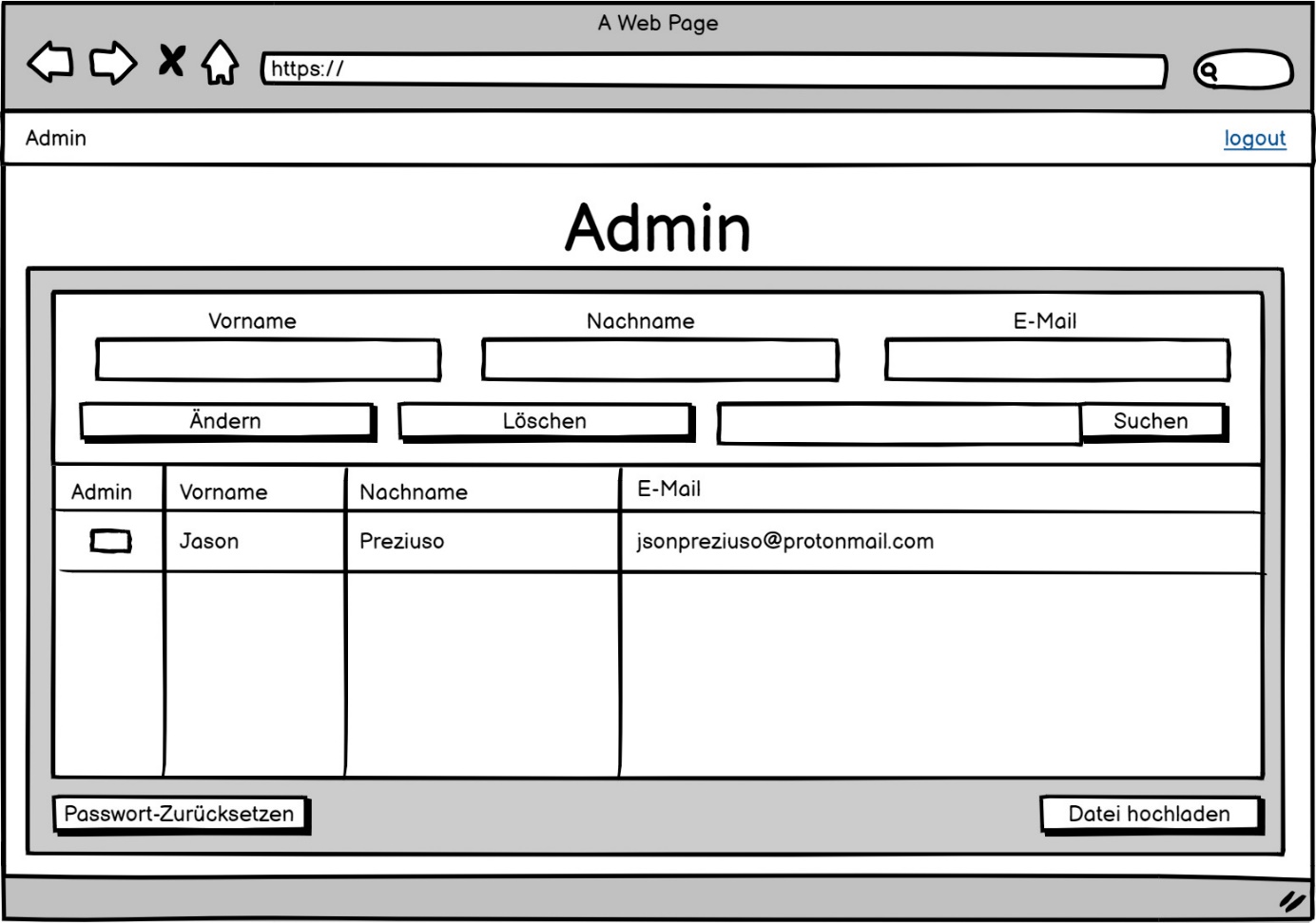


Abbildung 8 Mockup Admin

## 6.3 Verwendete Ressourcen

* Laptop Fujitsu LIFEBOOK E Series - Windows 10
* Laptop Lenovo LT2253
* Laptop Dell Latitude 7480
* MacBook Pro Retina, 15-inch
* <https://getbootstrap.com/>
* <https://color.adobe.com/de/create/color-wheel>
* https://www.w3schools.com/jquERy/default.asp
* https://www.w3schools.com/js/js\_ajax\_intro.asp
* https://www.w3schools.com/mySQl/default.asp
* https://www.w3schools.com/nodejs/
* <https://stackoverflow.com/>
* <https://www.mysql.com/>
* <https://docs.microsoft.com/en-us/learn/modules/build-web-api-nodejs-express/>
* <https://dev.to/sachinsarawgi/rest-api-with-nodejs-using-sqlite3-and-express-l3c>
* <https://expressjs.com/es/guide/routing.html>
* <https://devdotcode.com/how-to-connect-a-nodejs-api-to-mysql-database-3/>
* <https://getbootstrap.com/>
* <https://color.adobe.com/de/create/color-wheel> 1

## 6.4 Use-Case-Diagramm ³

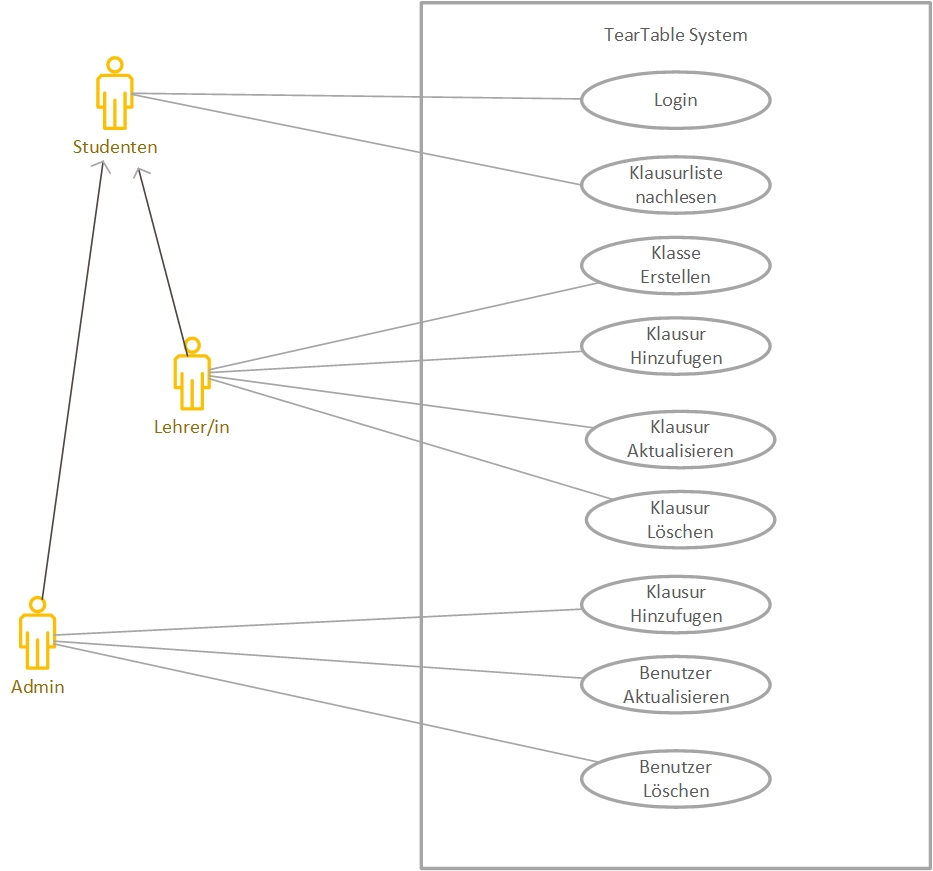


Abbildung 9 Use-Case-Diagramm

## 6.5 Screenshots der Anwendung 1

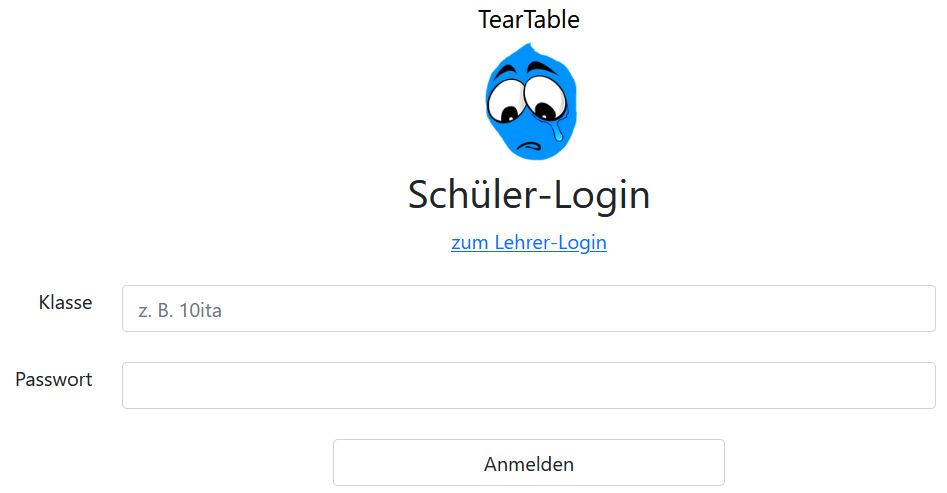


Abbildung 10 Anwendung-Screenshot Schüler Login

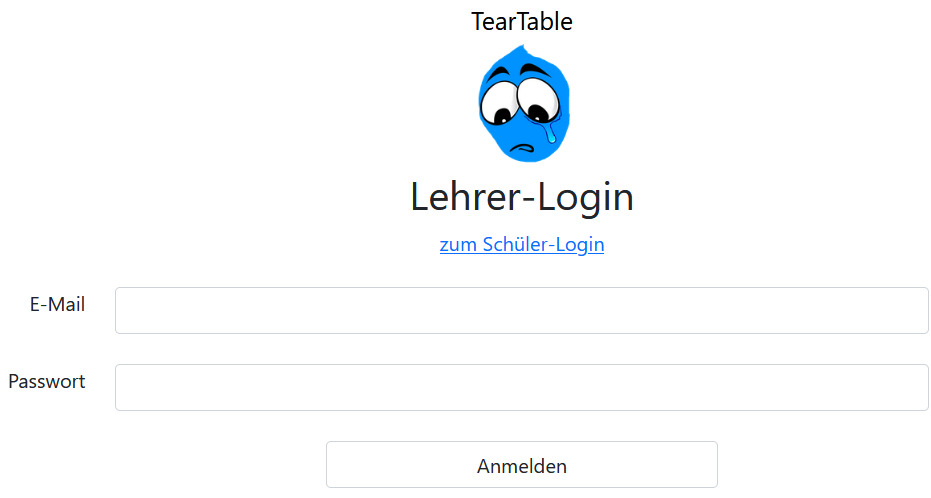


Abbildung 11 Anwendung-Screenshot Lehrer-Login

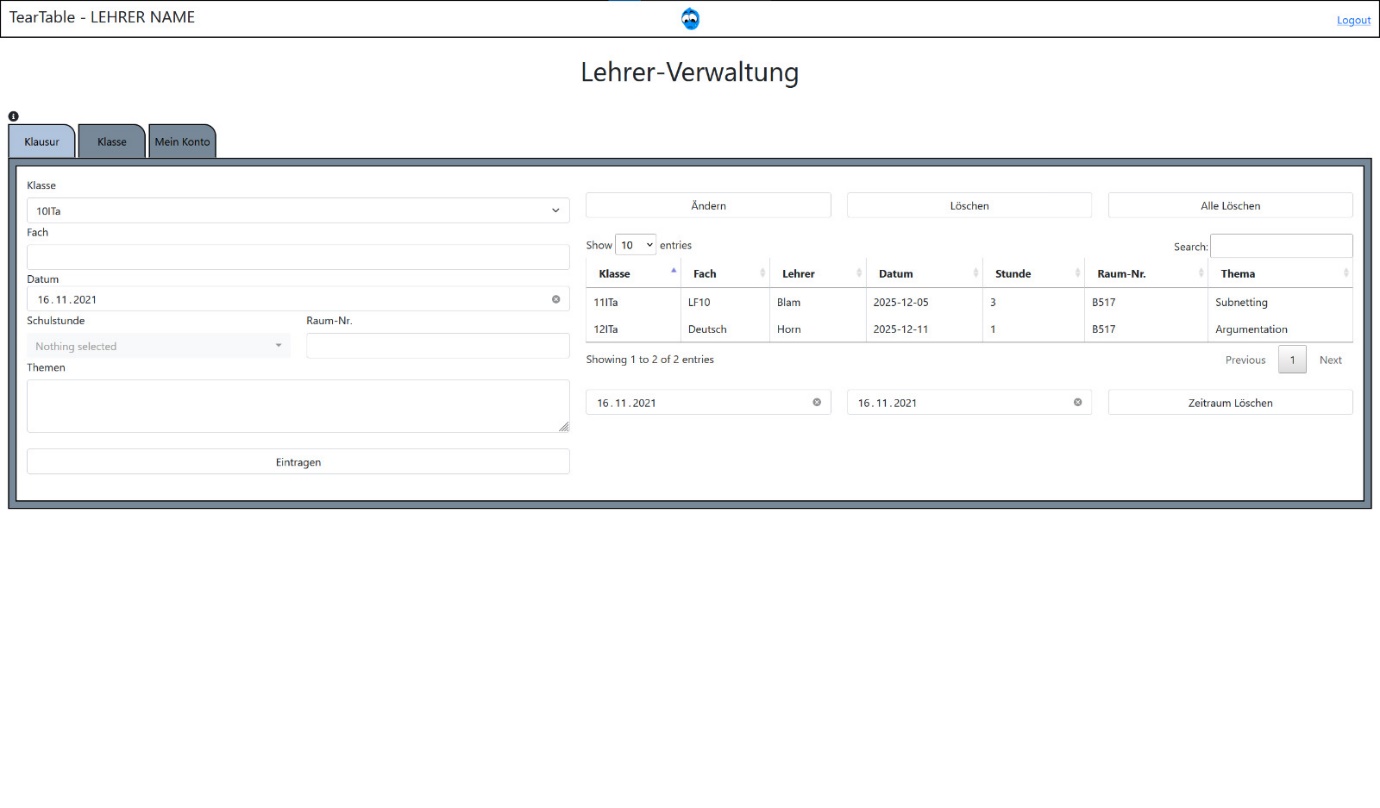


Abbildung 12 Anwendung-Screenshot Lehrer Verwaltung Klausur

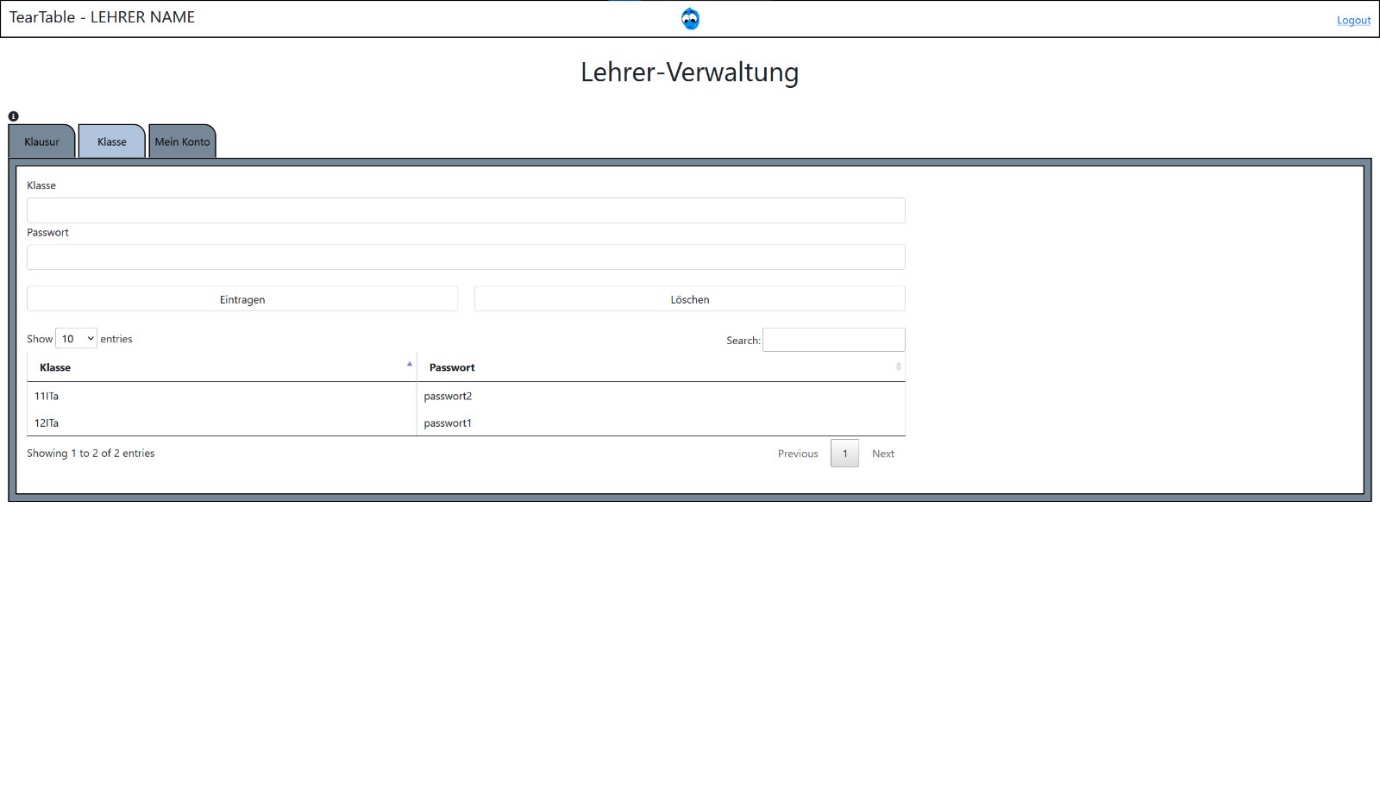


Abbildung 13 Anwendung-Screenshot Lehrer Verwaltung Klasse

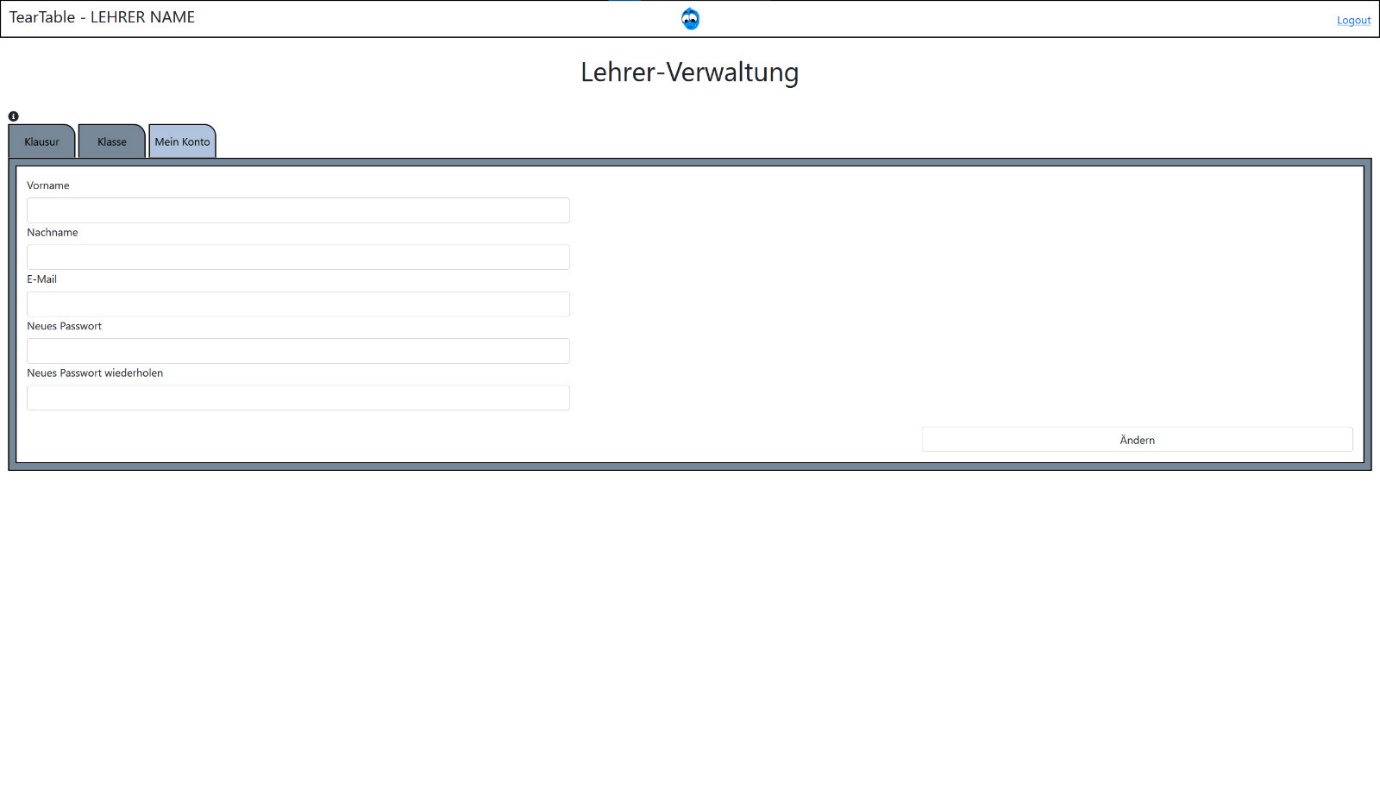


Abbildung 14 Anwendung-Screenshot Lehrer Verwaltung Mein Konto

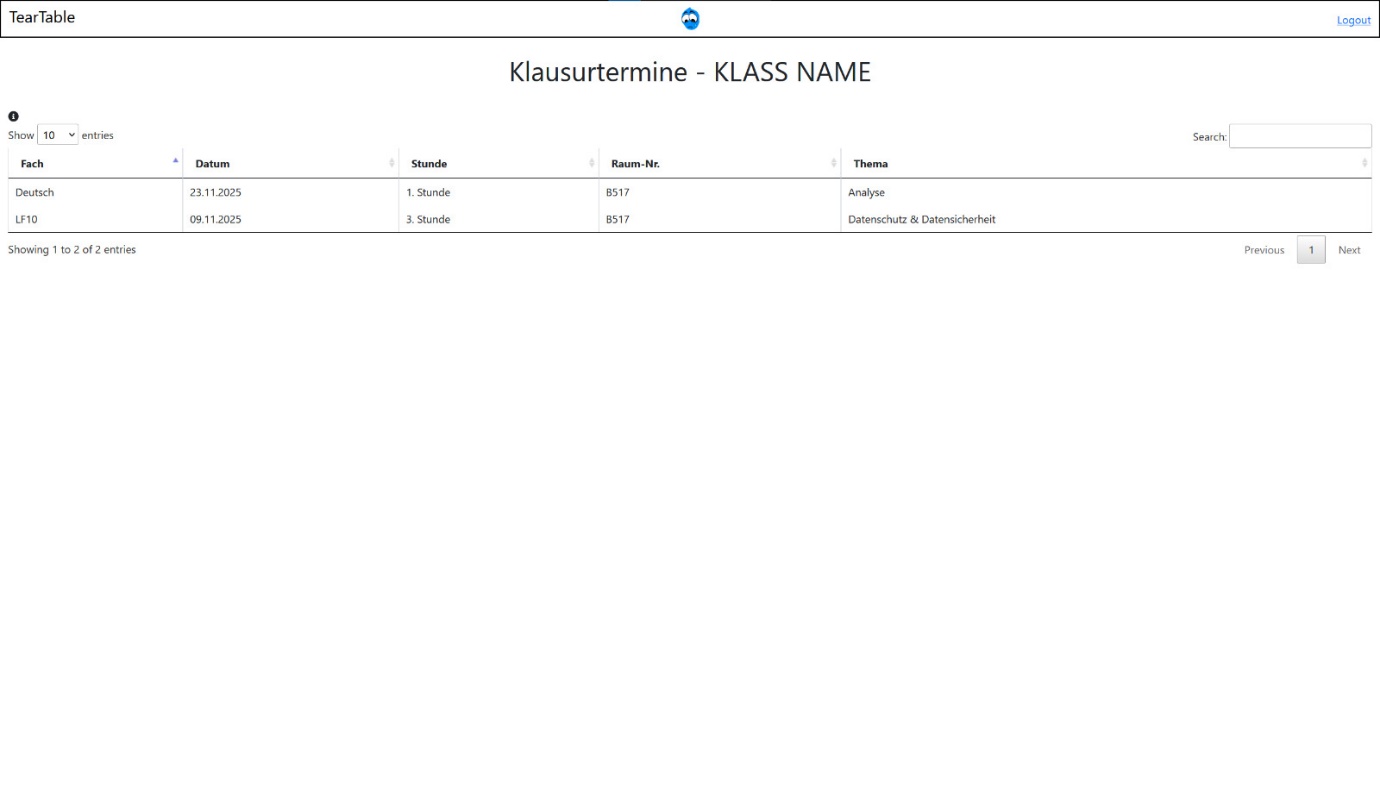


Abbildung 15 Anwendung-Screenshot Klausurtermine

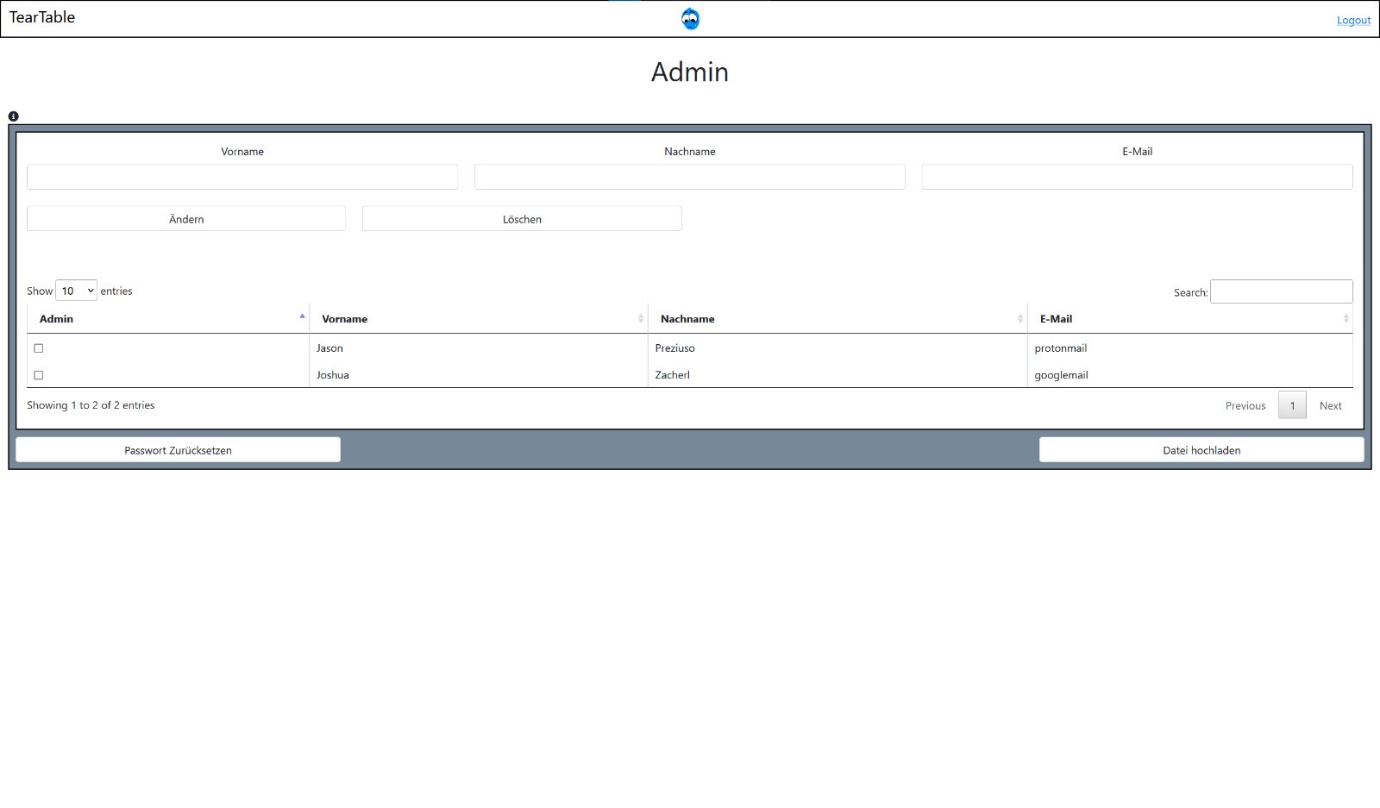


Abbildung 16 Anwendung-Screenshot Admin

## 6.6 Projektantrag

**Klausurenplaner**

Projektname: TearTable

Auftraggeber: Brühlwiesenschule Hofheim

Jason Preziuso, Joshua Zacherl, Jaime Gamero, Kadri Kalesi

Projektbeschreibung

|  |  |
| --- | --- |
| **Name des Projekts:**  Klausurplaner (Web-Applikation) “TearTable” | **Starttermin:** 01.09.2021  (4)  **Endtermin:** laufend |

|  |  |
| --- | --- |
| **Projektträger/Ansprechpartner:**  Jason Preziuso | **Adresse:**  (4)  Brühlwiesenschule Hofheim  06192 2904-0  office@bws-hofheim.de  Gartenstraße 28  65719 Hofheim |

**Ausgangslage:** (1)

Aktuell werden die Klausuren in der Klasse 12ITa in Moodle angepinnt. Eine gute Lösung ist dies nicht, weil es unübersichtlich ist und die Lehrer nicht immer daran denken, alles händisch einzutragen. In manchen Klassen werden Zettel mit Datum und Fach an die Wand des Klassenzimmers geheftet. Die Termine sind nicht immer einsehbar oder erreichbar für die Schüler, außer sie sind im Klassenraum anwesend.

**Kurzbeschreibung:** (3)

Die Schüler der Brühlwiesenschule Hofheim *Jason Preziuso*, *Joshua Zacherl*, *Kadri Kalesi* und *Jaime Gamero* möchten einen Klausurenplaner für die genannte Schule erstellen. Bei dem Klausurenplaner handelt es sich um eine digitale Web-Applikation zum Verwalten und Planen der Klausuren für die Schulklassen. Die Lehrer haben die Möglichkeit, Klausuren in den Planer hinzuzufügen und den Schülern Zugriff darauf zu gewähren, um zu sehen, wann die Klausuren stattfinden.

**Projektumfeld:** (2)

Das Projekt werden die vier angegebenen Schüler gemeinsam umsetzen. Die Umsetzung des Projekts erfolgt größtenteils in der Schule vor Ort. Teilaufgaben werden aber auch nach der Schule zu Hause erledigt. Als Schnittstelle zur Datenbank wird Node.js verwendet und mit “MySQL” werden Daten aus der Datenbank abgerufen, geändert, gelöscht und erstellt.

**Ziel:** (3)

Das Ziel des Projekts ist es, für die Brühlwiesenschule eine Web-Applikation zu entwickeln, die die Verwaltung und Bearbeitung von Klausuren so unkompliziert wie möglich macht. Wichtig dabei ist, dass Lehrer und Schüler von jedem Ort, wo man Internet hat, darauf zugreifen kann.

**Beschreibung:** (1)

Der Klausurenplaner hat folgende Anforderungen. Es soll möglich sein, Klausurtermine *erstellen*, *ändern* und *löschen* zu können. Folgende Attribute sollten enthalten sein:   
***- Datum & Uhrzeit bzw. Schulstunde(n)   
- Klasse   
- Fach***

***- Raum  
- Klausurthemen (Kann-Feld)***

Klassendaten können eingepflegt werden mit einem „Admin“-Account. Dieser Account soll folgende Attribute besitzen:

***- Klasse***

***- Passwort***

***- Import***

***- Funktion für Klassensätze (Klasse mit Passwort)***

***- Reset-Funktion für alle Termine eines wählbaren Zeitraumes***

Die User-Verwaltung sollte folgende Dinge enthalten:

- ***Lehrer und/oder Admin mit Namen, Mail, Passwort (Lehrer können auch Admin sein)***

***- Import-Funktion für Lehrerdaten mit Passwort***

Für das Front-End wird HTML, CSS und JavaScript genutzt. Für das Back-End wird MySQL und Node.js genutzt. Für die Entwicklung des Projekts wird eine agile Methodik gewählt, es wird Scrum angewendet. Diese Methodik bietet demnach eine dynamische Sicht auf den Entwicklungsprozess.

**Zeitplanung:** (2)

1. Dokumentation (Gesamt: 20 Stunden)

1.1 Projektbeschreibung (Verantwortlichkeiten in der Projektbeschreibung markiert) - 5 Stunden

1.2 Lastenheft (Jason) - 4 Stunden

1.3 Installationsanleitung für Admins (Kadri) - 3 Stunden

1.4 Benutzerhandbuch für Benutzer (Joshua) - 4 Stunden

1.5 Test-Protokolle (Jaime) - 2 Stunden

1.6 Team-Protokolle (Jason) - 2 Stunden

2. Front-End (Gesamt: 20 Stunden)

2.1 Grafische Oberfläche aller Unterseiten (Joshua) - 10 Stunden

2.2 JavaScript Funktionalitäten (Jason) - 10 Stunden

3. Backend (Gesamt: 20 Stunden)

3.1 Implementierung der API (Jaime) - 10 Stunden

3.2 Implementierung der Datenbank und Tabellen (Kadri) - 5 Stunden

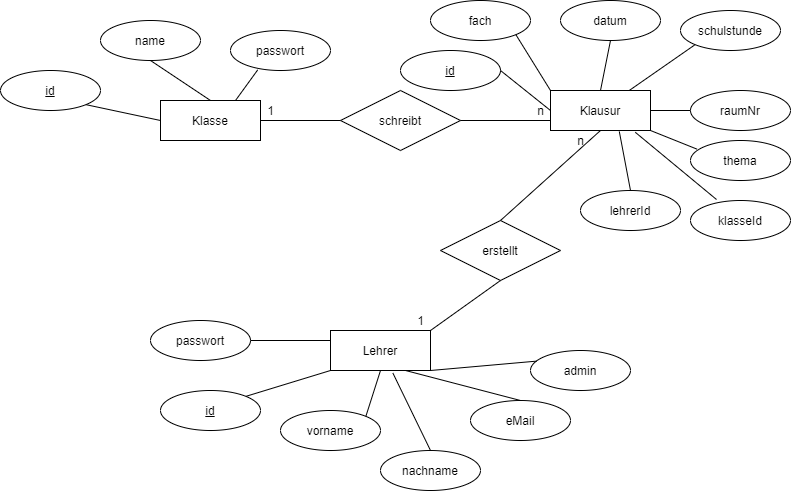
3.3 MySQL-Statements schreiben (Kadri) - 5 Stunden

4. Testen (Gesamt: 12 Stunden)

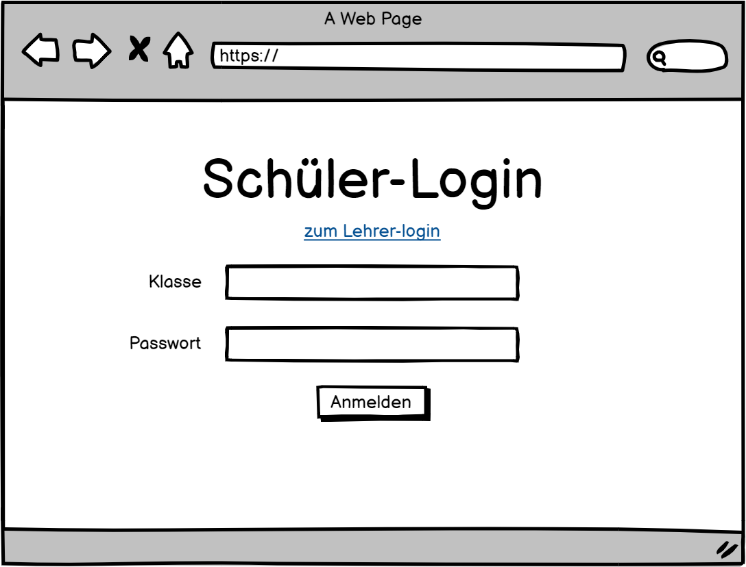
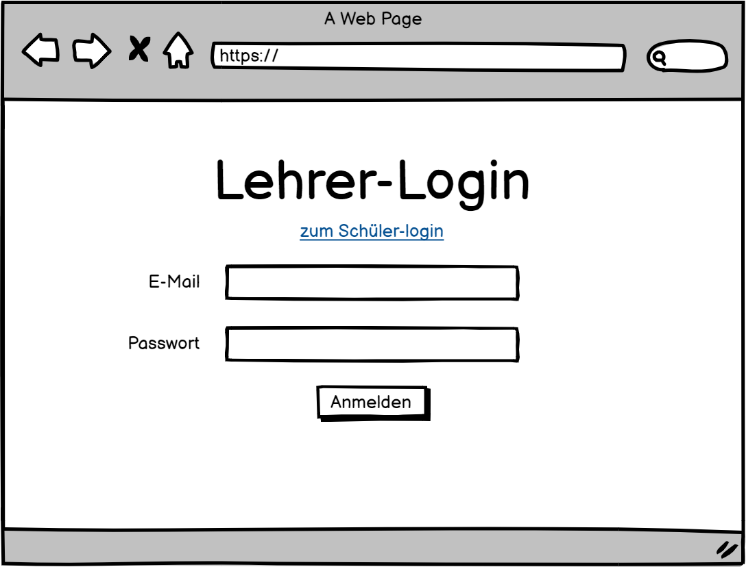
4.1 Test und Qualitätssicherung (Jason, Joshua, Jaime, Kadri) - je 3 Stunden

**Budgetplanung:** (4)

Kosten die eventuell anfallen würden, wären Kosten für Balsamiq Mockups 3. Ein Programm, womit man sich Mockups erstellen kann. Die Web-App würde 9 $ im Monat oder 90 $ im Jahr kosten. **Kosten entfallen**, da *Kadri Kalesi* eine Lizenz dafür besitzt. Weitere Kosten, die entstehen, sind die Serverkosten, welche aber die Brühlwiesenschule übernimmt, da die Server bereits in der Schule vorhanden sind.



*ERM-Datenmodell (2) A*



Lehrer-Login (1), (2) C

Schüler-Login / Startseite (1), (2) B

