HTBLuVA St. Pölten
Höhere Lehranstalt für Elektronik und Technische Informatik
Ausbildungsschwerpunkte Embedded- & Wireless Systems

DIPLOMARBEIT

Workshop Scheduling System

Ausgeführt im Schuljahr 2018/19 von:

Bernhard Würfel
Kiril Vereshchagin

Betreuer/Betreuerin:

AV Dipl.-Ing. Wolfgang Kuran

St. Pölten, am 6. Februar 2021

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Bernhard Würfel

Kiril Vereshchagin

HÖHERE TECHNISCHE BUNDESLEHRANSTALT ST. PÖLTEN

Abteilung: **Elektronik und Technische Informatik**
Ausbildungsschwerpunkt: Embedded Systems

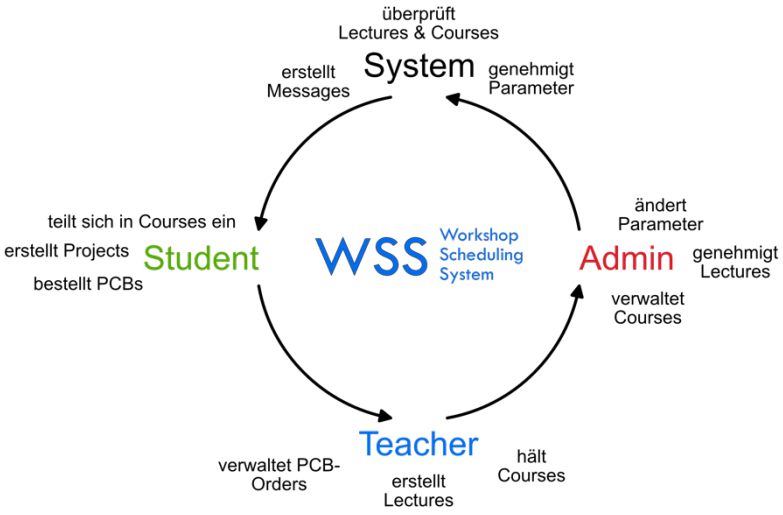
DIPLOMARBEIT
DOKUMENTATION

Namen der Verfasser/innen	Bernhard Würfel/Kiril Vereshchagin
Jahrgang / Klasse Schuljahr	5BHELS/2018-2019
Thema der Diplomarbeit	Workshop Scheduling System
Kooperationspartner	HTBLuVA St. Pölten

Aufgabenstellung	In Werkstätten sollen Schüler an individuellen Projekten arbeiten. Die Benützung der Geräte setzt die Betreuung durch Lehrer voraus. Damit die Schüler Zugriff auf Geräte in verschiedenen Räumen erhalten, soll mit einem Buchungssystem Zeit reserviert werden. Dafür soll die DA eine Weboberfläche für Anfragen, sowie ein System zur Einteilung von Terminen, Räumen und Lehrern bereitstellen. • Bereitstellung einer klar strukturierten Weboberfläche mit 4-facher Usergruppierung (Administrator, Werkstättenleiter, Lehrer, Schüler), übersichtlichem Design und verständlicher Navigation • das Erstellen der Datenbank für das Speichern von Räumen, Zeitplänen und Usern, mit der im weiteren Verlauf des Projektes interagiert werden soll • Entwicklung einer serverseitigen Anwendung, die mit Hilfe eines Algorithmus eines Drittanbieters die Daten von einer Datenbank entnimmt, verarbeitet und wieder in die Datenbank speichert • Implementierung der Möglichkeit zur Interaktion eines Administrators mit der Anwendung; der Administrator soll den Grad der Automatisierung bestimmen und weitere Parameter und ganze Zeitpläne ändern können

Realisierung	Mit einer Weboberfläche (Sprachen: HTML, CSS, JS, JQuery, PHP), die für das Einreichen der Anfragen und Anzeigen der Termine zuständig ist, einer Datenbank (MySQL), die Anfragen durch die Website und vorhandenen Eigenschaften der Räume, Zeitpläne und Lehrer beinhaltet und einer serverseitigen Anwendung zur Änderung der Automatisierung und manuellen Änderung von Zeitplänen wird das Ziel erreicht.

Ergebnisse	Die Website und alle nötigen Backend-Funktionen wurden fertiggestellt und deren Funktion durch eigene Tests und eine Beta-Testphase der Abteilung Elektronik und Technische Informatik sichergestellt. Des Weiteren wurde ein Feedback der Schüler und Lehrer eingeholt, das ebenfalls in dieser Arbeit behandelt wird.

Typische Grafik	 Konzept der Interaktion der Nutzer mit der Website

Teilnahme an Wettbewerben, Auszeichnungen	keine

Möglichkeiten der Einsichtnahme in die Arbeit	Einsichtnahme ab sofort in der HTBLuVA St. Pölten möglich

Approbation (Datum / Unterschrift)	Prüfer/in	Dipl.-Ing. W. U. KURAN Abteilungsvorstand

htl bildung mit zukunft	**COLLEGE of ENGINEERING ST. PÖLTEN**
	Department: **Electronics and computer engineering** Educational focus: Embedded systems

DIPLOMA THESIS
DOCUMENTATION

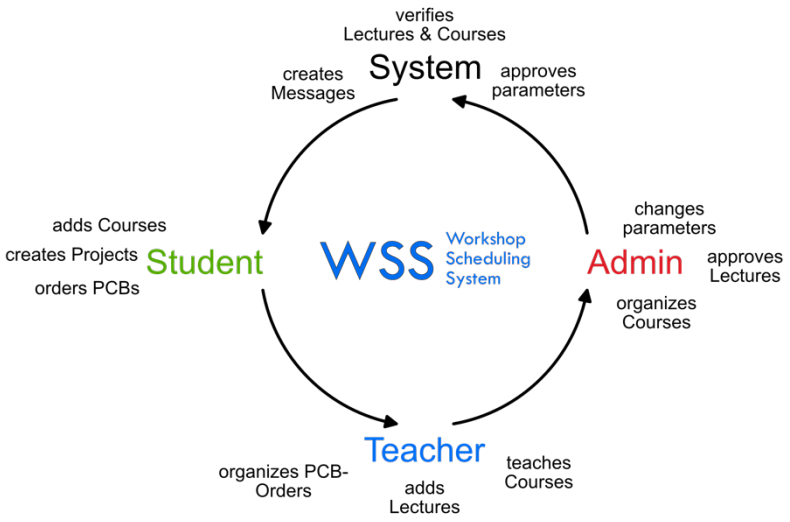
Author(s)	Bernhard Würfel/Kiril Vereshchagin
Form Academic year	5BHELS/2018-2019
Topic	Workshop Scheduling System
Co-operation partners	COLLEGE of ENGINEERING ST. PÖLTEN

Assignment of tasks (conceptual formulation/job definition)	Within the workshop-classes, students are supposed to work at their own projects. Therefore, they have to use multiple units of equipment, which have to be supervised by a qualified teacher. As every student needs different devices, there shall be a possibility to book a timeframe requiring a single room. For that matter, this thesis should provide an online-platform for requests, as well as a system for arranging appointments, rooms and teachers. • Deployment of a clear-structured online-platform by using a 4-stage user-assembly (admin, head of workshop, teacher, student), neat design and intelligible navigation • Construction of the database for storing rooms, schedules and users • Development of a server-sided application, which processes the data stored by the database by using a third-party utility • Implementation of parameter-changing and modifying schedules by the admin

Realization	The tasks are achieved by an online-platform (coding-languages: HTML, CSS, JS, JQuery, PHP), which is responsible for the receipt of requests and displaying schedules, a database (MySQL), which stores requests of the website, attributes of rooms, schedules and teachers, and a server-sided application for the alteration of the automation and manual changing of schedules.

Results	The website and all necessary server-sided functions were finalised and their function were ensured by tests within the team of developers and also by conducting a beta-test of the department of electronics and computer engineering. Furthermore, all participants did give feedback, which is discussed within this thesis.

htl bildung mit zukunft	**COLLEGE of ENGINEERING ST. PÖLTEN**
	Department: **Electronics and computer engineering** Educational focus: Embedded systems

Illustrative graph	 Concept of interaction by users with the website

Participation in competitions Awards	None

Accessibility of diploma thesis	Insight can be gained at the COLLEGE of ENGINEERING ST. PÖLTEN

Approval (Date / Sign)	Examiner	Dipl.-Ing. W. U. KURAN Head of Department

Inhaltsverzeichnis

1. Einleitung

1.1. Kurzfassung der Zielsetzung

Das Ziel dieser Diplomarbeit ist die Durchführung einer Studie über den Einsatz eines Buchungssystems im Rahmen des Werkstättenunterrichts. Dieses System soll Schülern die Möglichkeit bieten, ihre Projekte zu organisieren. Die Buchungen sollen über eine Web-Oberfläche erfolgen.
Durch eine serverseitige Anwendung sollen die Anfragen möglichst zeit- und platzeffizient verarbeitet werden, wobei vorhandene Räume - denen jeweils ein Lehrer zugeteilt ist - und deren Kapazitätsgrenzen berücksichtigt werden müssen.

1.2. Aufgabenteilung

1.2.1. Verantwortliche Person für das Core-System

Die verantwortliche Person für das Core-System (dt. Kern-System) ist Bernhard Würfel. Das Core-System verwaltet die Anfragen von Schülern, die sich in Räume einteilen wollen, sowie die Anzahl der Personen pro Raum und mögliche Problemfälle bei der Aufteilung der Schüler und Termineinhaltung.

Bernhard Würfel ist Schüler der 5BHELS und hat die Vertiefung *Embedded Systems* gewählt.

1.2.2. Verantwortliche Person für das Communication-System

Verantwortliche Person für das Communication-System (dt. Kommunikationssystem) ist Kiril Vereshchagin. Das Communication-System stellt die Verbindung zwischen Benutzer und der Datenbank dar und ist daher u. a. für die Profil-Verwaltung, die *Zugriffskontrolle*[1] und das Anfordern von Raumbelegungen notwendig.

Kiril Vereshchagin ist Schüler der 5BHELS und hat die Vertiefung *Embedded Systems* gewählt.

1.2.3. Betreuer

Der Betreuer dieser Diplomarbeit ist Dipl.-Ing. Wolfgang Kuran, Abteilungsvorstand der Abteilung Elektronik und Technische Informatik (EL) und Professor für Hardwareentwicklung (HWE), Laboratorium (LA1), Softwaretechnik (SOTE) und Fachspezifische Softwaretechnik (FSST).
Die Idee für diese Diplomarbeit wurde von AV Kuran persönlich eingebracht und daher war es naheliegend - zusätzlich zu seinen Erfahrungen und seinem Fachwissen - ihn als Betreuer zu wählen.

[1] Darunter versteht man die Tatsache, dass nicht jeder Benutzer der Web-Oberfläche dazu berechtigt ist, jeden Inhalt aufzurufen. Außerdem beschreibt sie die dynamische Separation des Inhaltes entsprechend der Benutzerebene. Mehr dazu im Unterabschnitt 3.2.3

1.3. Meilensteine

Aufgabe	Person
bis 03.09.2018	
Informationen über Algorithmen zur Sortierung wurden eingeholt	Bernhard Würfel
Informationen über Bootstrap bzw. Alternativen wurden eingeholt	Kiril Vereshchagin
bis 03.11.2018	
Datenbank und grundsätzlicher Aufbau der Website wurden fertiggestellt	Kiril Vereshchagin
bis 03.02.2019	
Algorithmus für individuelle Anwendung ist funktionstüchtig	Bernhard Würfel
PHP-Teil der Website ist funktionstüchtig	Kiril Vereshchagin
bis 03.03.2019	
Oberfläche der Desktop-Anwendung[a] wurde perfektioniert	Bernhard Würfel
grafische Gestaltung der Website wurde perfektioniert	Kiril Vereshchagin
bis 03.04.2019	
Dokumentation der Diplomarbeit wurde fertiggestellt	beide

Tabelle 1.1.: Meilensteine

[a] wurde im Verlauf der Entwicklung durch eine Weboberfläche ersetzt

2. Grundlagen und Methoden

2.1. Ausgangslage

Die Schüler der vierten Jahrgänge der Höheren Technischen Bundes Lehr- und Versuchs-Anstalt St. Pölten (HTBLuVA St. Pölten), Abteilung für Elektronik und Technische Informatik, sind derzeit zur Durchführung eines fachgerechten Projektes im Rahmen des Werkstättenunterrichts in Gruppen oder einzeln verpflichtet. Das Ziel dieses Verfahrens ist die Vorbereitung auf die Diplomarbeit und Arbeitswelt. Dabei werden die Bereiche Projektplanung, Projektmanagement, Teamarbeit und *übliche*[1] technische Fähigkeiten gefördert. Die entstehende Problematik ist die Umsetzung.
Der Werkstättenunterricht sieht vor, dass sich die Schüler in n Gruppen aufteilen, damit diese dann unter x verschiedenen Lehrern zur Betreuung in x verschiedene Räume (jedem Lehrer ist ein Raum zugeteilt) gegliedert werden können. Die Gruppen rotieren dann x mal im Schuljahr zwischen den Lehrern und ihren Räumen. Dieses Prinzip ist gut, jedoch nicht ideal. Es stellt sicher, dass jeder Schüler bspw. die Leiterplattenfertigung oder das Netzwerktechniklabor besucht hat. Dabei ist der Zeitpunkt fest vorgegeben und wird nicht an die Bedürfnisse des Schülers angepasst, wodurch die Methodik sich in diesem Fall als unflexibel erweist. Will ein Schüler aus der Leiterplattenfertigung wieder in das Netzwerktechniklabor, so muss er ein halbes Jahr warten, sofern dieses im nächsten Semester im Lehrplan vorgeschrieben ist. Diese Problematik ist dem Personal bekannt, weswegen oft für verschiedene Räume gleiche Geräte angeschafft werden. Damit versucht man vorzubeugen, dass einem Schüler die notwendige Apparatur zur Durchführung seines Projektes fehlt. Räume, die in ihrer Ausstattung zu teuer oder zu speziell sind, werden nicht reproduziert. Außerdem bleibt das Personal (die Lehrer) an die jeweiligen Räume gebunden. Findet eine Rotation statt, so ist der vorherige Lehrer im Unterricht für den Schüler nicht mehr zugänglich. Allgemein wurde festgestellt, dass innerhalb eines Zyklus der persönliche Kontakt nur mit einem Lehrer für jeden Schüler der Gruppe N möglich ist.

[1] Das sind Fähigkeiten, welche von den Schülern während ihrer schulischen Laufbahn erlernt wurden.

Schülerprojekte setzen oft nicht nur das Fachwissen oder die Konsultation der jeweiligen Experten voraus, sondern auch die Benutzung mancher Räume. Festgestellt wurde, dass nicht jeder Raum und jeder Lehrer für jedes Projekt benötigt wird. So erfordert bspw. ein Projekt über "Netzwerke und Server" (Name frei erfunden) die Zugänglichkeit zum Netzwerktechniklabor, benötigt aber keinen Besuch in der Leiterplattenherstellung. Hier tritt also ein Fall des "Erschöpfenden Ressourcenverbrauchs" auf, bei dem ein Schüler keinen bestimmten Nutzen aus dem erteilten Lehrer für sein Projekt ziehen kann, jedoch Platz beansprucht.
Außerdem unterbindet diese Rotation den Ansatz der selbstständigen Projektplanung des Schülers. Im Optimalfall würde sich der Schüler stets am Standort des maximalen Gewinns für sein Projekt befinden. Dadurch könnten auch redundante Geräte aus manchen Räumen entfernt und in jenen, in denen sie in größerer Anzahl benötigt werden, hinzugefügt werden.

2.1.1. Ziel

Es ist essenziell, die Effektivität von Schülerprojekten zu erhöhen, aber auch die Souveränität der Lehrer über jeden Schüler zu bewahren. Schließlich sollen Lehrer hierbei nicht nur als "Personal", sondern als "Mentoren" agieren. Diese Mentoren sollen über gewisse Hoheiten oder Vorteile den Schülern gegenüber verfügen, um notfalls ihren Status durchsetzen zu können.
Die Erhöhung der Effizienz des Projekts eines Schülers erfordert allerdings mehr Subsistenz und Autonomie des Schülers. Wird jedoch einem Schüler zu viel Autonomie gewährt, so besteht die Gefahr des Missbrauchs oder des *paradoxen Optimalfalls*[2].

2.1.2. Probleme und Hindernisse

Problem des paradoxen Optimalfalls

Dieser Begriff deutet an, dass der Optimalfall erreicht wurde. Der Schüler befindet sich also in der Position des maximalen Gewinns. Diese Position kann stets der selbe Ort sein, was im schlimmsten Fall dazu führt, dass keine Rotation zwischen Räumen stattfindet. Im "besten" Fall für den Schüler ist dieser für manche Lehrer unzugänglich. Somit kann ein Lehrer diesem Schüler den vorgesehenen Lehrinhalt nicht vermitteln, der, trotz möglicher fehlender Relevanz für das Projekt, notwendig ist. Dieses Recht können sie nicht vollziehen, wenn manche Schüler für sie nicht zugänglich sind. Ein Optimalfall für Schüler

[2]Dieser Begriff mag fachlich nicht ganz korrekt sein, jedoch beschreibt er die Situation am besten. Man kann die hier geschilderte Lage auch mit dem mathematischen Problem der lokalen und globalen Maxima in Verbindung bringen.

ist ein Hindernis für Lehrer - deshalb spricht man hier vom *paradoxen Optimalfall*. Zusammenfassend soll also ein Mittelweg zwischen Schüler-Autonomie und Lehrer-Souveränität gefunden werden.

Problem des Missbrauchs

Der Fall des Missbrauchs ist ganz ähnlich. Dabei befindet sich der Schüler nicht am Ort des maximalen Gewinns, sondern hängt sich ganz bewusst an einen Raum oder einen Lehrer fest. Dies hat ganz wie in der Situation des paradoxen Optimalfalls die Folge, dass manche Lehrer dem jeweiligen Schüler ihren Lehrinhalt nicht vermitteln können. Dies zieht auch die Tatsache nach sich, dass der Platz, den der Schüler beansprucht, einem anderen mehr Nutzen erbringen würde.

Hindernis des Platzlimits

Man geht davon aus, dass der Platz in Räumen beschränkt ist. Im besten Fall entspricht die Gesamtkapazität der Räume der Gesamtzahl der Schüler. Weiterhin besteht ein Problem, wenn ein Raum n Plätze hat und mehr als n Schüler diesen - dank gewährter Autonomie - beanspruchen wollen. Dies kann sowohl einen parasitären, als auch einen legitimen Hintergrund haben.

2.2. Lösungsansatz

Um das Ziel (im Unterabschnitt 2.1.1 beschrieben) zu erreichen, wurde die Erstellung einer Software beschlossen. Es soll ein Computerprogramm[3] entwickelt werden, welches sowohl den Lehrern, als auch den Schülern verhilft, ihre Tätigkeiten miteinander zu koordinieren.
Mit Hilfe dieses Programms sollen sich die Schüler selbst aussuchen können, welche Lehrperson sie zu welchen Zeiten besuchen. Im Nachfolgenden werden solche Einheiten als *Kurse* bezeichnet. Natürlich bleibt dabei die Unterrichtszeit vorgegeben; lediglich der Standort und das dazugehörige Lehrpersonal sollen variabel sein. Die Unterrichtszeiten können dabei entweder durch einen Verantwortlichen (z.B.: einen Administrator) festgelegt, oder direkt aus WebUntis ausgelesen werden.
Da Projekte oft in Gruppen von bis zu drei Personen vollzogen werden, soll es möglich sein, über dieselbe Software erkenntlich zu machen, wer die Projektteilnehmer eines Projektes sind. Im Optimalfall ist dies die Aufgabe des Schülers. Dementsprechend muss es eine Möglichkeit geben, eine kurze Projektbeschreibung und einen Projektnamen anzugeben. In Voraussicht auf die Diplomarbeit

[3] Die Rede ist zwar von einem "Computerprogramm", damit soll aber die Gesamtheit aller Softwaremöglichkeiten gemeint sein.

soll es zu jedem Projekt einen Projektbetreuer geben. Da man davon ausgehen kann, dass es zu jedem Zeitpunkt mehr Projekte als Betreuer gibt, müssen manche Betreuer mehr Projekte annehmen als andere. Diese haben dann auch die Möglichkeit, alle ihre betreuten Projekt und deren Teilnehmer zu sehen. Der Projektbetreuer soll einer der verfügbaren Werkstättenlehrer sein.
Gleichzeitig tragen die Lehrer Verantwortung über ihre Räume und Unterrichtseinheiten. Daher sollen sie die Möglichkeit haben, Kapazitätsgrenzen für ihre Räume festzulegen, Pflichtkurse für bestimmte Schüler zu erstellen und Schüler aus ihren Kursen zu entfernen.
Die Einteilung in Kurse soll durch jeden Schüler individuell stattfinden. Teilt sich ein Schüler bis zum Termin nicht ein, so muss er durch das System automatisch eingeteilt werden.

2.2.1. Lösung mittels eigenständiger Software

In Abschnitt 2.2 wurde bereits evaluiert, dass zur Erreichung der erwünschten Ergebnisse eine Software entwickelt werden muss. Jedoch wurde die Art jener noch nicht festgelegt.

Es besteht die Möglichkeit, ein *eigenständiges Computerprogramm* zu entwickeln, welches auf Benutzer-Endgeräten installiert werden muss. Dieses Programm kann sowohl in C, C++, Java oder in einer anderen Programmiersprache geschrieben werden. Die Wahl einer *objektorientierten*[4] oder *prozeduralen*[5] Programmiersprache obliegt den Kenntnissen und Anforderungen der Entwickler. Das Endziel kann theoretisch mittels beider Techniken erreicht werden, im nachfolgenden Abschnitt "Vorteile" wird dennoch nur auf die Vorteile objektorientierter Programmiersprachen eingegangen, da diese zahlreicher ausfallen.

Vorteile

- Dank zahlreicher Möglichkeiten der objektorientierten Programmiersprachen gibt es viele vordefinierte Zugriffsmethoden, Klassen und Objekte, die verwendet werden können, wodurch der Programmieraufwand minimiert wird.
- Dank diverser graphischer Entwicklungsverfahren, wie Scene Builder für JavaFX fällt das Entwickeln einer graphischen Benutzeroberfläche *recht einfach* aus.
- Komplexe Aufgaben, wie Optimierungsaufgaben, können besser/effizienter erledigt werden, als bei der Entwicklung eines Websystems.

[4]Objektorientierte Programmierung, Definition siehe Glossar
[5]Prozedurale Programmierung, Definition siehe Glossar

- In Integrated Development Environment (dt. integrierte Entwicklungsumgebung) (IDE) integrierte Debugger gestalten das Auffinden von Fehlerquellen einfach und unproblematisch.
- Da der Code lokal[6] ausgeführt wird, können Parameter wie Skalierbarkeit oder Leistungseffizienz, im Gegensatz zu serverseitigen Verfahren, eher vernachlässigt werden, bzw. reichen die internen Optimierungsalgorithmen völlig aus.

Nachteile

- Eine dedizierte Software birgt Schwierigkeiten bei der plattformübergreifenden Nutzung. Oft müssen Dinge wie Prozessor-Architekturen, Betriebssysteme oder Endgerätetypen[7] beachtet werden.
  Dies ist von Sprache zu Sprache unterschiedlich. Bspw. funktioniert ein Java Programm unter Windows und Linux; für eine ARM Architektur (wie bei mobilen Endgeräten) muss jedoch eine zusätzliche App entwickelt werden.
- Es gibt nie eine Garantie dafür, dass eine Software, die unter Intel Core I7[8] der aktuellen Generation entwickelt wurde, auch auf den Schulrechnern läuft.
- Bezogen auf den vorherigen Punkt, ist das Testen der Software viel aufwändiger. Es kann niemals jede Kombination von Plattformen, Prozessoren und Betriebssystemen getestet werden.
- Sicherheitslücken können viel schwerer behoben werden. Es müssen dedizierte *Patches* geschrieben werden, dessen Installation man nicht überprüfen kann. Dies gilt ebenso für reguläre *Software Updates*.
- Bezogen auf den vorherigen Punkt, sind dementsprechend viele verschiedene Softwareversionen im Umlauf. Enthält die Software auch Datenbankoperationen, kann das zu sicherheitstechnischen Problemen führen. Zusätzlich können ältere Versionen nicht mehr wirksame Befehle enthalten.
- Die Installation ist ebenso problematisch. Zuerst muss sichergestellt werden, dass das Endgerät die notwendige Programmiersprache installiert hat. Dann muss entschieden werden, ob man dem Benutzer das Kompilieren von Quell-Dateien (Source Compiling) zutraut, oder fertige Binärdateien mit einer Installtionssoftware ausgeliefert werden müssen.

[6] Das bedeutet, dass das Ausführen des Codes/Programms auf dem jeweiligen Endgerät und nicht auf einem Server stattfindet.

[7] z.B. PC, Laptop, Tablet, Mobiltelefon etc.

[8] Ein CISC (x64) Prozessortyp der Firma Intel; überwiegend in PCs oder Laptops verbaut.

2.2.2. Lösung mittels Web-Oberfläche

Eine Internet-Schnittstelle, also eine Web-Oberfläche, ist in der Entwicklung (subjektiv) deutlich aufwändiger. Zudem wird die Anzahl der verwendbaren Sprachen auf sehr wenige eingeschränkt. Zugleich müssen mehrere Sprachen verwendet werden. Unter anderem kann eine Webseite ohne *Auszeichnungssprachen* (korrekterweise spricht man hier von *Markup Languages*), wie Hypertext Markup Language (dt. Hypertext Auszeichnungssprache) (HTML) oder Cascading Style Sheets (dt. gestufte Gestaltungsbögen) (CSS), wobei CSS eigentlich eine "Gestaltungssprache" ist, nicht funktionieren. Darüber hinaus müssen eine oder mehrere Sprachen für Server-seitige Operationen und Datenbankinteraktionen gewählt werden. Somit wird ein sehr großer Teil der rechenintensiven Aufgaben auf die Serverseite übertragen. Dementsprechend muss in die Serverinfrastruktur investiert werden (sowohl an Arbeitsleistung, als auch an wirtschaftlicher Leistung). Im Folgenden werden die Vor- und Nachteile detailliert geschildert.

Vorteile

- Das Aktualisieren einer Webseite ist wesentlich einfacher, als das eines *Computerprogramms*. Es müssen lediglich die Dateien auf dem Server aktualisiert werden. Dem Benutzer wird somit keine zusätzliche Verantwortung über das Aktuellhalten der Software auferlegt.

- Es müssen keine speziellen Patch- oder Update-Skripte geschrieben werden.

- Bezogen auf den ersten Punkt, können die Entwickler sich stets sicher sein, dass jeder Benutzer die aktuellste Version der *Software* verwendet; somit muss die Abwärtskompatibilität nicht beachtet werden.

- Da Teile des Codes, die für die Sicherheitsfunktionen verantwortlich sind, auf dem Server und nicht beim Benutzer liegen, sind *reverse engineering-Attacken*[9] praktisch auszuschließen.

- Wesentlich benutzerfreundlicher ist das Wegfallen der Installation auf einem Endgerät. Der Benutzer benötigt lediglich einen Webbrowser. Man kann davon ausgehen, dass jeder, der (2019) einen Computer, ein Smartphone etc. besitzt, auch über einen Webbrowser verfügt.

[9] Jemand versucht dem Hersteller der Software Schaden zuzufügen, indem er den Code mit Hilfe diverser Werkzeuge rekonstruiert und manipuliert. Meistens ist das Ziel die Beschädigung der Datenbank oder im Falle einer lizenzpflichtigen Software die illegale Verbreitung (Software-Piraterie).

- Gerätekompatibilität muss nur kosmetisch realisiert werden. Da die technischen Operation auf dem Server ausgeführt werden, muss die CPU-Architektur oder Grafikkarte des Benutzers nicht beachtet werden.

Nachteile

- Man kann sich, im Gegensatz zur gewöhnlicher Computer Software, die benutzten Sprachen kaum aussuchen, da die Entwicklung für das World Wide Web (dt. weltweites Netz) (WWW) auf einige wenige beschränkt ist.

- Für das Entwickeln einer solchen Webseite (wie in Abschnitt 2.2 beschrieben) wird viel mehr technisches *Know-How*, und dementsprechend viele Technologien, benötigt.

- Für die Sprachen der Web-Entwicklung gibt es derzeit keine IDEs per se. Zur Verfügung stehen lediglich Texteditoren, die in ihrer Funktionalität variieren. Bestenfalls wird ein eingebauter Hypertext Preprocessor (PHP)-Debugger bereitgestellt.

- Klassischerweise werden in der Web-Entwicklung Datenbank-Interaktionen mit PHP vollzogen. Die Entwicklung wird zusätzlich durch die dauerhafte Notwendigkeit eines Webservers erschwert, da PHP nur mit diesem ausgeführt werden kann.

- Generell fällt die Gestaltung der Benutzeroberfläche viel schwieriger aus, als bspw. mit Java. Es muss ein HTML-Gerüst, sowie eine CSS-kaskadierte Gestaltung der Elemente erstellt werden. Bei einem *Computerprogramm* wird diese Aufgabe von diversen graphischen Werkzeugen erheblich erleichtert.

2.2.3. Datenbank als Zusatz

Da eine anderwertige Speicherung von Stundenplan-, Projekt- und Benutzerinformationen und eine Synchronisation mehrerer Benutzer mit sehr viel Aufwand verbunden ist, muss eine Datenbank als zusätzlicher Teil des Projekts entwickelt werden. Dies ist völlig unabhängig davon, welcher konzeptuelle Lösungsansatz gewählt wird.

Alternativ können solche Informationen auch in Comma-Separated Values (CSV) Tabellen abgelegt werden. Betrachtet man aber die Vielfalt der Einträge, sowie Faktoren wie Skalierbarkeit, so ergibt sich keine andere Möglichkeit, als die einer Datenbank.

2.3. Web-Oberfläche als Endziel

Betrachtet man alle Vor- und Nachteile, die in Unterabschnitt 2.2.1, sowie Unterabschnitt 2.2.2 erhoben wurden, so ergibt sich Folgendes: Die Entwicklung einer Web-Oberfläche erscheint auf den ersten Blick viel aufwändiger, verglichen mit der eines Computerprogramms. Doch betrachtet man den Aufwand über die gesamte Zeitspanne, so verringert sich dieser. Am umständlichsten ist der Entwicklungsbeginn: Die Serverinfrastruktur, sowie der einheitliche Auftritt aller Webseiten-Elemente und die Modellierung der Datenkbank müssen erstellt werden. Hat man allerdings diese anfänglichen Hürden überwunden und eine Projektstruktur festgelegt, so ist die eigentliche Entwicklung recht unproblematisch. Erste Testversionen können dem Kunden schnell vorgeführt oder rasch aus der Umlaufbahn genommen werden. Ist eine Version der Webseite erstmal im Einsatz, kann diese nahezu aufwandlos, durch das schlichte Erneuern oder Hinzufügen von Dateien, aktualisierst werden. Einer bestimmten Gruppe von Benutzern kann ebenso ein Testkonto gewährt werden, um Informationen über Fehler oder erwünschte Funktionen zu sammeln.

Im Ganzen erscheint eine Web-Oberfläche in diesem Fall der bessere und modernere Ansatz zu sein, zumal eine Webseite, wenn richtig gestaltet, immer und auf nahezu allen Endgeräten funktionieren wird.

2.4. Projekteinsatz

Da das *Workshop Scheduling System (dt. Werkstättenplannungs-System) (WSS)* nur eine Machbarkeitsstudie darstellt, wird es zwar an der HTBLuVA St. Pölten getestet, jedoch aufgrund der Schulbestimmungen nicht in naher Zukunft an dieser eingesetzt. Dafür müsste eine Änderung des derzeitigen Ablaufs des Werkstättenunterrichts erfolgen und sichergestellt werden, dass der vorgesehene Lehrplan trotz der Projekte vermittelt wird.

2.4.1. Projektfunktion

Wie in den vorangegangenen Abschnitten bereits detailreich geschildert wurde, bietet das *WSS* eine Plattform zur Verwaltung von Schülereinteilungen, Projekten und der Bearbeitung dieser. Zusätzlich wird durch die Implementierung eines Nachrichten-Systems ebenfalls eine bessere Kommunikation zwischen Schülern und Lehrern außerhalb des Unterrichts gefördert.

Durch den flexiblen Aufbau soll das System jederzeit auf eine gesamte Schule,

dessen Stundenplan und die damit verbundene Ressourcenverwaltung, skaliert werden können.

2.4.2. Gewinn für die Forschung

Im Gesamten stellt das *Workshop Scheduling System* einen weiteren Schritt für die moderne Pädagogik dar; mit dem *WSS* wird das Konzept der Waldorfpädagogik, erstmals in Österreich, auf eine höhere technische Lehranstalt erweitert.

Durch Gewährung von Autonomie und Entscheidungsfreiheit soll der Unterricht weniger aufgezwungen, sondern viel mehr als Teil der Freizeit maskiert werden. Die Schüler dürfen größtenteils unabhängig ihre eigenen technischen Projekte in Projektgruppen von bis zu drei Teilnehmern, wobei jedem Teilnehmer sein eigener Verantwortungsbereich obliegt, planen, durchführen, dokumentieren und präsentieren. Das Resultat: die Schüler, wenn auch in einem eingeschränkten Rahmen, agieren als Wegweiser für den Unterricht und nicht umgekehrt. Daraus entspringt eine aktive Förderung für die wichtigsten aller Eigenschaften: kreative, soziale und kognitive Intelligenz. Im Sinne der "kreativen Intelligenz" ist es die Findung von Ideen, Umsetzungsmöglichkeiten und Lösungswegen, sowohl für das ganze Projekt, als auch für die eigene Branche. Die "soziale Intelligenz" wird gefördert, indem die Schüler unter sich die Hierarchie und die Arbeitsteilung ausmachen. Dabei wird ein Projektleiter bestimmt, der dementsprechend viel Verantwortung über die zeitgerechte Fertigstellung des Projektes trägt. Die Förderung kognitiver Fähigkeiten findet in Form der Durchführung per se statt. Jeder Schüler muss sein technisches *Know-How* mit logischen, künstlerischen und anderen Fähigkeiten kombinieren, um das Ziel zu erreichen.

Dadurch, dass die Lehrkräfte keine autoritäre sondern vielmehr eine unterstützende Rolle annehmen, werden Vertrauen und Empathie beidseitig gefördert.

Das Konzept ist für eine HTL - soweit bekannt - einzigartig und die Ergebnisse des Versuchs wegweisend. Auf jeden Fall besteht die Möglichkeit, die Resultate dieser Fallstudie für weitere Forschung zu nutzen, oder diese zur Verbesserung der Unterrichtsqualität an HTLs und anderen Schulen in ganz Österreich anzuwenden.

3. Individuelle Zielsetzung

3.1. Core-System (Bernhard Würfel)

3.1.1. Verarbeitung der Einteilungen

Wie bereits in 2.1.2 beschrieben, treten ab einer zu hohen Anzahl an Schülern, die zur gleichen Zeit im gleichen Raum arbeiten möchten, Platzprobleme auf. Zusätzlich sollen diese nicht im gesamten Semester bei einer Lehrperson bleiben, sondern möglichst viele zu ihrem Projekt passende Räumlichkeiten - und damit Lehrer - aufsuchen können.

Schüler können jedoch nicht gezwungen werden, einen bestimmten Raum aufzusuchen, da dies die Freiheit und damit die Grundidee dieser Diplomarbeit einschränken würde. Deshalb muss eine Methode gefunden werden, die diese dazu bringt, freiwillig verschiedene Lehrer innerhalb eines Semesters zu frequentieren.

Zusammenfassend soll also die Freiheit der Schüler, als auch die faire Nutzung gegenüber anderen Mitmenschen, sichergestellt werden.

3.1.2. Administrative Ebene

Über eine eigene Oberfläche, die in die Web-Oberfläche (3.2.3) implementiert ist, sollen berechtigte Personen (die als *Admins* eingestuft werden) wichtige Parameter ändern und alle Ressourcen verwalten können. Darunter fallen das Bearbeiten und Hinzufügen von Räumen, Zeiteinteilungen, PCB-Bestellungen, sowie das Erstellen von Lehrer- und Schülerkonten. Der *Admin* ist ebenso dazu berechtigt, alle Einteilungen anzusehen, diese zu verschieben und zu löschen.

3.1.3. Nachrichten-System

Einen weiteren Punkt, der in die Web-Oberfläche (3.2.3) integriert ist, stellt das Nachrichten-System da. Dabei lehnt sich dieses an E-Mailing an, da keine Chats[1] aufgebaut werden, sondern Nachrichten mit Betreffen erstellt werden, auf die referenziert oder geantwortet werden kann.

Dieses stellt vor allem eine Verbesserung für die Kommunikation zwischen Lehrern und Schülern außerhalb des Unterrichts dar. Nachrichten müssen dadurch nicht über das herkömmliche E-Mail-System oder per SMS/Messenger getätigt werden, bei denen sie mit anderen Themen vermischt werden.

Die Gesamtzahl der Nachrichten, die der Nutzer gleichzeitig sieht, beschränkt sich auf projektbezogene Benachrichtigungen. So können die Übersicht und der Fokus leicht bewahrt werden, denn die Mitteilungen können nicht zwischen Spam-Emails und anderen Meldungen untergehen.

[1] Anglizismus für "sich unterhalten", meistens über das Internet, in kurzen, oft belanglosen, Diskussionen.

3.2. Communication-System (Kiril Vereshchagin)

3.2.1. Server-Infrastruktur

Die Server-Infrastruktur bezieht sich sowohl auf die Wahl des Betriebssystems, des Webservers und der Datenbank, als auch auf die Bereitstellung der Server-Hardware, sowie der Sicherstellung, dass alle Komponenten zusammenspielen und stets die aktuellste Version der Webseite der Öffentlichkeit zur Verfügung steht.

3.2.2. Datenbank

Im Rahmen der Diplomarbeit - wie bereits in Unterabschnitt 2.2.3 evaluiert - soll auch eine Datenbank modelliert und erstellt werden. Mit dieser soll anschließend die Web-Oberfläche kommunizieren, um von ihr Informationen abzurufen oder Benutzeranfragen zu speichern. Natürlich spielt auch Hr. Würfel beim Erstellen der Datenbank eine Rolle; so bringt er Informationen darüber ein, welche Tabelleneinträge oder Tabellen er für seinen Teil der Diplomarbeit benötigt.

3.2.3. Web-Oberfläche

a) **Geschlossenes System**
Sehr wichtig für das Projekt ist die Tatsache, dass nur bestimmte Angehörige der HTBLuVA St. Pölten darauf zugreifen können. Dies ist wichtig, um sicherzustellen, dass das System nicht missbraucht wird; schließlich sollen nur berechtigte Schüler, Projekte, Stundenpläne etc. buchen dürfen.

b) **Benutzerebenen**
Das System soll auf die Verwendung durch Benutzer mit verschiedenen Berechtigungsstufen ausgelegt werden. Grundsätzlich soll zwischen folgenden Stufen unterschieden werden:

- Super-Admin
- Administrator
- Lehrer (u. a. Professor)
- Schüler

Die Reihenfolge der Auflistung entspricht gleichzeitig der korrekten Hierachie. Der Benutzer soll seine Rechte vom jeweils höhergestellten Benut-

zer empfangen oder durch diesen selbst angelegt werden. Dieses Kriterium setzt die Tatsache voraus, dass es mindestens einen Benutzer der höchsten Hierarchie-Stufe per Voreinstellung (korrekterweise sagt man hier auch "per Default") gibt. In diesem Fall *muss* ein Super-Admin zu jedem Zeitpunkt existieren.

c) **Zugriffskontrolle**
Hr. Vereshchagin ist auch dafür zuständig, dass bestimmte Benutzer auf bestimmte Seiten nicht zugreifen dürfen, oder stattdessen auf ihre eigenen *semiäquivalenten*[2] Seiten umgeleitet werden.

d) **Gestaltung**
Das Aussehen, sowie die logische Struktur, wie die Navigation, die Zugänglichkeit oder das Offenlegen von Elementen der Seite, sind ebenso im Verantwortungsbereich von Herrn Vereshchagin. Es soll eine möglichst übersichtliche, *einfache und verständliche*[3] Benutzerschnittstelle entwickelt werden. Dazu gehört aber auch das Erscheinungsbild des *WSS* (korrekterweise spricht man hier von: "Corporate Design"), was z.B. Logos beinhaltet, und das Auftreten von WSS als Ganzes (korrekterweise, "Corporate Identity"), was sich auf die "Gesamtheit aller Merkmale die ein Unternehmen kennzeichnen"[1](Wikipedia) bezieht.

3.2.4. Sicherheit

Eine sekundäre, jedoch nicht zu vernachlässigende Aufgabe, ist die Sicherstellung diverser Sicherheitsfunktionen. Diese reichen von der Serversicherheit, wie z.B. der Sicherheit des Zugriffsprotokolls, über die Sicherheit der Webseitenfunktionen, wie dem Schutz vor *SQL-Injections*[4], -Angriffen oder *Cross-Site-Scripting (dt. webseitenübergreifendes Scripting) (XSS)*[5]-Attacken.

[2]Hierunter versteht man in diesem Kontext eine zum Gesamtsystem untergeordnete Seite, die jenen Funktionen und Berechtigungen entspricht; wie diejenige, die der Benutzer versucht hat aufzurufen - mit dem Unterschied, dass diese neue Seite für die Zugriffsberechtigung des Benutzers vorgesehen ist.

[3]Diese Begriffe sind subjektiv, da jeder Mensch darunter etwas anderes versteht. Jedoch lässt sich von jedem Individuum ein bestimmtes Niveau an “Simplizität“ definieren. Jenes Niveau muss im Rahmen der Diplomarbeit möglichst einheitlich dezidiert werden.

[4]Eine ”Injektion” von Datenbank-Anfragen über bspw. Eingabefelder einer Webseite durch einen Angreifer, um z.B. Informationen einer Tabelle zu extrahieren, oder gar die ganze Datenbank unbrauchbar zu machen.

[5]Darunter versteht man bösartigen Script-Code, meistens realisiert mit JavaScript, welcher auf eine Ziel-Seite - oft über Eingabefelder - durch einen Angreifer eingeschleust wurde, um einem anderen Nutzer Schaden zuzufügen

3.3. Umsetzung durch beide Diplomanden

3.3.1. Durchführung von Umfragen und Interpretation

Ein wesentlicher Teil der Diplomarbeit ist die Erhebung von Benutzererfahrungen über die Nutzung des Systems unter Alltagsbedingungen. Konkret ist erwünscht, das Verhalten der Benutzer, mit dem Ziel der Feststellung der Benutzerfreundlichkeit, sowie der Handlichkeit des Systems, zu revidieren. Die beiden Gruppen, die in etwa 99% aller Benutzer ausmachen sind:

I. Lehrer

II. Schüler

Für die Administratoren reicht eine Testung unter klinischen Bedingungen.
Die Resultate der Revision können verwendet werden, um Fehler zu beheben oder die allgemeine Benutzererfahrung zu verbessern; so soll im Allgemeinen festgestellt werden, wie intuitiv oder verwirrend das System tatsächlich ist. Außerdem soll Feedback darüber eingeholt werden, welche Funktionen noch eingebaut werden müssen, welche überflüssig sind, und welche Bereiche evtl. anders gelöst werden können.
Dabei ist es von besonderer Wichtigkeit, auf die Rückmeldung des Zielpublikums einzugehen. Denn möchte man einen Erfolg für das gesamte Projekt, sowie eine mögliche Weiterentwicklung, sicherstellen, so darf es auf keinen Fall daran scheitern, dass das System vom "Kunden" abgelehnt wird.

Aufgrund der Fallstudie ist es ebenso kritisch zu evaluieren, wie das neue Verfahren zur Projekteinteilung, -Durchführung und Stundeneinteilung in der Praxis, unter keinen Idealbedingungen, funktioniert. Außerdem ist festzustellen, ob die Problematik, wie in Abschnitt 2.1 geschildert, gelöst wurde.

3.3.2. Testung und Fehlersuche

Praktischerweise muss bereits während der Entwicklung gezielt nach Fehlern gesucht werden, um diese zu beheben. Dabei sollte nicht nur der eigene, sondern auch der Code des jeweils anderen getestet und revidiert werden. Dies hat den Zweck, dass vor der Auslieferung an den Endbenutzer bereits die meisten Fehler behoben sind und das System funktionsfähig ist. Solche Art von Testung ist zwar eher klinisch, jedoch ebenso wichtig. Sie stellt sicher, dass die Software, zumindest nach Intentionen der Entwickler, funktioniert.

4. Realisierungswünsche der Anwender

4.1. Allgemein

Geleitet durch die Idee von AV Dipl.-Ing. Kuran, wurden mehrere Fachlehrer besucht und mögliche Vorschläge besprochen. Ob diese innerhalb der vorhandenen Zeit bzw. durch ein Team von zwei Personen realisierbar ist, wurde hierbei, ähnlich einem Brainstorming, nicht berücksichtigt. Aus diesem Grund wird im Folgenden bei jedem Punkt gesondert erwähnt, ob dieser im fertigen Produkt enthalten ist oder die Implementierung aus rechtlichen/zeitlichen Hintergründen nicht in Betracht gezogen wurde.

4.2. Einteilung pro Halbtag

- Wunsch von Fachlehrer Halmetschlager, Fachlehrer Prantl und Fachlehrer Rzepa

- in das System integriert

Anfänglich war eine stundenweise Einteilung geplant, da so die höchstmögliche Flexibilität gewährt wird. Aufgrund des hohen Aufwandes für den Lehrer, der beim Kontrollieren der Anwesenheit anfallen würde, wurde die Periode auf einen halben Schultag erweitert. Somit kann auch leichter überprüft werden, welcher Schüler zuletzt mit welchem Gerät und auf welchem Arbeitsplatz gearbeitet hat, wenn dieses/dieser nicht ordnungsgemäß zurückgelassen wurde. Durch eine zu hohe Frequenz könnte der Urheber durch die große Anzahl an Personen nicht eindeutig herausgefunden werden. Dies wird noch deutlich erschwert, wenn dieser Missstand erst nach einigen Stunden auffällt.

In der Software wurde das Regeln der Dauer einer Einteilung flexibel gestaltet, sodass diese bis auf eine Minute reduziert werden kann. Diese Änderung unterliegt jedoch nur den Administratoren.

4.3. Projektmanagement

- Wunsch von Fachlehrer Halmetschlager und Fachlehrer Rzepa
- in das System integriert

Ähnlich wie bei einer Diplomarbeit oder vorwissenschaftlichen Arbeit gibt es einen Bereich auf der Website, der den Projekten gewidmet ist. Hier können bis zu drei Schüler mit einem Fachlehrer/Professor ein Projekt erstellen, das gewisse Themenbereiche abdeckt. Ebenso müssen Meilensteine festgelegt, und bei der Fertigstellung alle notwendigen Dateien für die Benotung hochgeladen werden.

4.4. Gruppeneinteilung per WebUntis

- Wunsch von Fachlehrer Halmetschlager, Fachlehrer Prantl und Fachlehrer Rzepa
- aus rechtlichen Gründen nicht im System integriert

Wie im bisherigen Schulalltag üblich, wird weiterhin WebUntis zum Eintragen der Anwesenheit verwendet. Alle Befragten haben den Wunsch nach einer möglichen Übertragung der Schüler pro Lehrer nach WebUntis/Untis geäußert. Dies lässt sich jedoch aufgrund der Konfiguration von Untis seitens der HTBLuVA St. Pölten, bei dem Schülergruppen nur jahresweise geändert werden können, nicht umsetzen. Außerdem protokolliert Untis als einziges System die Anwesenheit der Schüler und kann deshalb im Zweifelsfall, bspw. bei Gerichtsfällen, herangezogen werden.

Entscheidet nun eine externe Software (wie das WSS) darüber, welcher Schüler bei welchem Lehrer sein soll, kann dadurch eine Inkonsistenz entstehen.

4.5. Grundkompetenzen als Theoriestunden

- Wunsch von Fachlehrer Halmetschlager, Fachlehrer Prantl und Fachlehrer Rzepa
- in das System integriert

Damit während den frei wählbaren Stunden auch die laut Lehrplan notwendigen Grundkompetenzen vermittelt werden können, steht es Lehrern frei, sogenannte *Lectures* (dt. Theoriestunden), zu erstellen. In diesen können sie ausgewählten Schülern den nötigen Lehrstoff weitergeben. Diese Funktion wird durch den Admin streng kontrolliert, wodurch jede einzelne Theoriestunde von diesem zugelassen werden muss.

4.6. Wochenbericht

- Wunsch von Fachlehrer Rzepa
- aus zeitlichen Gründen in der aktuellen Version nicht enthalten

Wochenberichte stellen ein Protokoll für die Arbeiten jedes Schülers pro Arbeitstag bis hin zu einzelnen Stunden dar und werden bisher von diesen selbst geführt. Damit die Berichte dauerhaft vom Fachlehrer/Professor einsehbar sind und auch ein Verlorengehen unterbunden werden kann, ist es der Wunsch von Fachlehrer Rzepa, diese in einer digitalen Form in das *Workshop Scheduling System* einzubinden.

5. Technologien der Webentwicklung

5.1. Entwicklungsumgebungen

Da sowohl die Präferenzen jedes Einzelnen, als auch die Eigenschaften und die damit verbundenen Vor- und Nachteile jeder IDE unterschiedlich sind, werden in diesem Projekt mehrere verwendet. Es ist jedoch von äußerster Relevanz, dass jede IDE eine integrierte Unterstützung für *GitHub* und damit *Git* (in den Unterabschnitten 5.1.3 und 5.1.4 weiter ausgeführt) anbietet, damit jeder Projektteilnehmer jederzeit ohne den Verlust von Änderungen eines anderen an seinem Teil weiterarbeiten kann. Sollte eine IDE diese Funktion nicht enthalten, kann durch die Verwendung von *GitHub* eine Aktualisierung der Daten erfolgen. Möglichen Interferenzen von Code-Abschnitten können jedoch nicht gelöst werden.

5.1.1. Atom

Atom ist ein für mehrere Betriebssysteme entwickelter Texteditor, der von *GitHub* entwickelt und als Open-Source-Software angeboten wird. Er basiert auf *Electron*[1] und zeichnet sich durch standardmäßiges Syntax-Highlighting, automatische Vervollständigung von Befehlen und einer, von der Community hochgradig unterstützten, Plug-in-Integrierung aus.

[1] "Electron besteht aus dem Webbrowser Chromium und dem JavaScript-Framework Node.js und erlaubt es, beliebige Anwendungen mit JavaScript, HTML und CSS zu erstellen " Wikipedia [2]

5.1.2. Visual Studio Code

Ähnlich zu Atom wird *Visual Studio Code* als Open-Source-Software für mehrere Betriebssysteme bereitgestellt. Es wird von *Microsoft* entwickelt und basiert auf *Electron*, wodurch viele Features denen von Atom gleichen. Zusätzlich ist ein Debugger für JavaScript und PHP integriert.
Der Name mag an das von Microsoft entwickelte *Visual Studio* erinnern; *Visual Studio Code* hat jedoch bis auf einige Funktionen wie *IntelliSense*[2] keine Gemeinsamkeiten mit diesem.

5.1.3. Git

Git spielt eine zentrale Rolle bei der Verwaltung von Dateien und Dateiversionen. Es protokolliert Veränderungen in Dokumenten, gliedert diese möglichst effizient in die alte Version ein und weist böswillige Änderungen, die zur Zerstörung führen würden, ab. Der letzte Punkt kann zusätzlich duch andere am Projekt beteiligte Personen verhindert werden, indem sie Neuerungen abblocken.

Im Gegensatz zu anderen Versionskontrollsystemen speichert *Git* nicht die Änderungen ab, sondern den Inhalt zu jenem Zustand. Um dabei trotzdem effizient zu bleiben, werden nur veränderte Dateien neu abgespeichert, während alle anderen als Verknüpfung angelegt werden.

Vorteile von Git:

- nicht-lineare Entwicklung und Erstellung von mehreren aktuellen Versionen
- kein zentraler Server, der eine dauerhafte Internetverbindung veraussetzen würde
- Sicherheit gegen Manipulation der Versionsgeschichte
- zurückgenommene Aktionen werden zwischengespeichert, bis sie explizit gelöscht werden

[2] "IntelliSense ist ein von Microsoft angebotenes Hilfsmittel zur automatischen Vervollständigung bei der Bearbeitung von Quellcode durch einen Programmierer." Wikipedia [3]

Befehle und Bezeichnungen

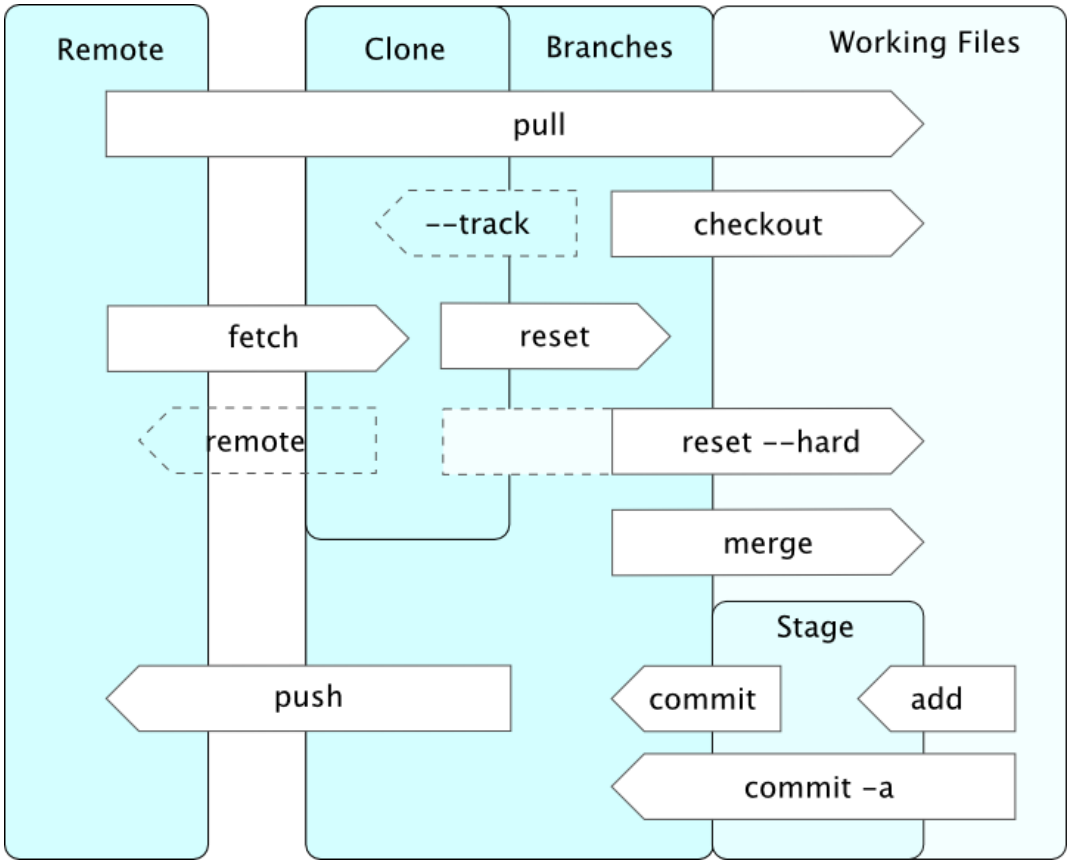


Abbildung 5.1.: Operationen von Git und ihr Wirkungsbereich
(Quelle: https://de.wikipedia.org/wiki/Datei:Git_operations.svg)

Repository
Ein *Repository* ist eine Summe von Dateien und Ordnern, die lokal gespeichert werden. Wird an einem Dokument etwas geändert, so findet diese Änderung zuerst nur im Arbeitsverzeichnis statt, wodurch alle Objekte im *Repository* unverändert bleiben.

Stage & Commit
Möchte man Änderungen auch in den Dateien im *Repository* übernehmen, so prüft *Git* zuerst den aktuellen Stand. Danach werden die Dateien verglichen und Geänderte durch den Befehl *add* in den *Staging*-Bereich überführt. Schließlich werden durch einen *Commit* die betroffenen Objekte im *Repository* geändert.

Push
Mit *Push* die Änderungen im *Repository* in den *Remote*-Bereich geladen, der alle aktuellen Dateien enthält und jederzeit von jedem Nutzer durch den Befehl *fetch* angesteuert werden kann, um etwaige Neuerungen festzustellen.

Fetch
Mit dem Befehl *Fetch* werden alle *Commits* und Dateien in das eigene *Repository* geladen. Sollten keine Änderungen vorliegen, werden die bereits im *Repository* vorhandenen Dokumente weiter benutzt.

Pull
Sollten durch den Befehl *fetch* Änderungen festgestellt worden sein, kann das eigene *Repository* damit abgeglichen werden. Treten beim *Mergen* (also dem Zusammenfügen von Dateien) Konflikte auf, so müssen diese vom Benutzer behandelt werden, indem er entweder die neuen Teile akzeptiert, oder die Bisherige bestehen lässt.

Branching
Ein Feature, das im Laufe dieser Diplomarbeit nicht verwendet wurde, jedoch in manchen Anwendungsfällen sehr hilfreich sein kann, ist das *Branching*. Hierbei wird eine Funktion extrahiert, sodass sie alleingestellt weiterentwickelt werden kann. Sollte die Entwicklung abgeschlossen sein, kann die *Branch* durch das *Mergen* wieder zum eigentlichen Projekt (*Master*) hinzugefügt werden.

5.1.4. GitHub

GitHub ist die Online-Version von *Git* und ermöglicht die Veröffentlichung von Dateien. Außerdem macht es gemeinsames Arbeiten möglich, indem es eine leicht zugängliche Plattform für den *Remote*-Teil von *Git* darstellt. Zusätzlich gibt es eine plattformübergreifende Anwendung von *GitHub*, die den Ablauf aller Befehle in einem Graphical User Interface (dt. grafische Benutzeroberfläche) (GUI) ermöglicht und eine Übersicht aller Versionen bereitstellt.
Grundsätzlich bietet die kostenlose Version kein gemeinsames Arbeiten an Projekten ohne deren Veröffentlichung an, jedoch wurde dieses Problem gelöst, indem alle Projektteilnehmer dasselbe Konto nutzen, wodurch der Projektordner nicht öffentlich auf der Seite von *GitHub* erscheint. Dadurch werden beim *Pushen* von Änderungen statt Nutzernamen statische, gerätegebundene IDs verwendet, wodurch der Urheber leicht identifiziert werden kann.

5.2. Script-Sprachen und Languages

Im Gegensatz zu konventionellen Programmiersprachen wie C++ oder C, werden Script-Sprachen und Languages nicht kompiliert sondern interpretiert. Sie sind somit ein sehr hohes Konstrukt und werden oft, wenn Zeit kein Faktor ist und aus leistungs-steigernden Gründen in maschinennahe Programmiersprachen übersetzt (z.B.: Python zu C).

Die wesentlichen Unterschiede zwischen Compiler und Interpreter bestehen darin, dass Compiler den ganzen Code einlesen und dann in Maschinencode umwandeln, während Interpreter diesen Block für Block in Maschinencode übersetzen. Generell ist Kompilieren langsamer als Interpretieren, dafür läuft das Programm aber wesentlich performanter. Da Interpreter um einiges schneller den Quellcode übersetzen, eignen sich diese deutlich besser für Web-Anwendungen. Deswegen werden die meisten „Web-Sprachen" wie JavaScript, Ruby On Rails, JQuery, PHP und selbstverständlich HTML, interpretiert.

Der Unterschied zwischen (Programmier/Script)-Sprache und Language ist eher verschwommen, zudem ist es nicht ganz klar ob der Begriff "Language" überhaupt noch gebraucht wird. Generell definieren wir eine Language so: *Die Language ist ein imperatives Konstrukt - es folgen also Befehl auf Befehl - ohne jegliche „Extravaganzen", wie Schleifen, Rekursionen, Subroutinen, etc., welche von herkömmlichen Programmiersprachen angeboten werden. Zudem wird eine Language interpretiert.* Klassische Languages sind HTML oder LateX.

5.2.1. HTML

HTML steht für Hypertext Markup Language und ist eine Auszeichnungssprache, die rein auf Text basiert. Sie ermöglicht das Struktieren von Dokumenten, die bspw. Text, Aufzählungen, Links und Bilder enthalten.
HTML arbeitet mit sogenannten "Tags". Das sind Code-Blöcke, welche bestimmte *Funktionen* erfüllen. Z.B.: <**p**> für Text, <**img**> für ein Bild oder <**h1**> für die erste Überschrift. Tags werden mit einem Backslash geschlossen: </**p**>. Alle Tags werden vom <**html**>-Tag eingeschlossen; das einzige Element darüber ist die Definition <**!doctype html**>. Anhand dieser erkennt der Browser, dass es sich um HTML-Code handelt, wobei moderne Browser sehr großzügig bei der Interpretation des HTML-Codes vorgehen, sodass viele Programmierfehler einfach *überlesen* werden.

Grundsätzlich muss man davon ausgehen, dass alle Inhalte innerhalb des *<html>-Tags* beim Empfänger ankommen und von dessen Browser verarbeitet werden. Zwar werden nicht alle direkt angezeigt, der Nutzer kann sie jedoch durch Anzeigen des Quellcodes sichtbar machen. Daher sollte darauf geachtet werden, wie gewisse Bereiche gestaltet werden und dass Daten, die der Nutzer nicht erblicken darf, auch nicht in Kommentaren versteckt werden.

Durch bestimmte HTML-Funktionen lassen sich Benutzer-Eingaben in Feldern einschränken: z.B.: type=”email”, maxlength=”12” oder type=”date”. Da der HTML-Code allerdings **immer** auf der Benutzer-Seite ausgeführt wird, und die Benutzer diesen **jederzeit** direkt im Browser bearbeiten können, müssen die Ergebnisse von Formularfeldern, die an den Server geschickt werden, stets mit großer Vorsicht behandelt werden.
Hat man z.B.: im HTML ’type=”date”’ gesetzt, darf man nicht davon ausgehen, dass der Client[3] tatsächlich ein gültiges Datum abschickt; sollten die empfangenen Informationen für die ordnungsgemäße Funktionalität der Seite wesentlich sein, so müssen diese entweder mittels eines PHP-Scripts, oder einer anderen serverseitigen Anwendung geprüft werden. Mehr dazu in: Abschnitt 8.1.

5.2.2. CSS

Die Abkürzung *CSS* steht für „Cascading Style Sheets” und ist eine *Language*, die für die visuelle Gestaltung von Inhalten verwendet wird. Dabei schränkt sich *CSS* nicht nur auf Webseiten ein, sondern findet häufig Verwendung in vielen graphischen Protokollen.
In der Webentwicklung kann durch das Adressieren von HTML Tags wie *div*, *section*, *p*, etc. das Erscheinungsbild von einzelnen oder mehreren Bereichen verändert werden. Es können aber auch sog. *ids* oder *classes* verwendet werden, um Elemente anzusprechen.

[3]Ein Programm auf dem Benutzer-Endgerät, welches auf Informationen-Austausch mit einem *Server* oder *Host* angewiesen ist.

Der Code ist sehr einfach zu produzieren, und schränkt sich lediglich auf das Kombinieren von einzelnen Befehlen ein:

```
1  body {
2    background-color: lightblue;
3  }
4  h1 {
5    color: white;
6    text-align: center;
7  }
8  p {
9    font-family: verdana;
10   font-size: 20px;
11 }
```

Listing 5.1: Beispiel zur Adressierung und Gestaltung von Elementen mittels CSS

Hier wird zunächst der gesamte *HTML-Body* adressiert, bei dem die Hintergrundfarbe auf Hellblau gesetzt wird. Die Erste Überschrift „**h1**" wird in die Mitte gerückt und dessen Farbe auf weiß geändert. Für alle Paragraphen „**p**" wird die Schriftart auf *Verdana* gesetzt und die Textgröße auf 20 quadratische Pixel beschränkt. Der CSS-Code wird ebenfalls auf der *Client*-Seite ausgeführt, und kann durch den Endbenutzer leicht manipuliert werden.

Abbildung 5.2.: Das Ergebnis des in 5.1 aufgeführten Codes

Der HTML-Code wurde in dem Exempel 5.1 bewusst weggelassen, da dieser sich zum einem nur auf wenige Zeilen beschränkt und zum anderem nicht wesentlich für die Demonstration von *CSS* ist. Natürlich ist das oben bearbeitete Beispiel nur eine sehr vereinfachte Darstellung von dem, was mittels *CSS* möglich ist.

Auch bei *CSS*-Dateien muss darauf geachtet werden, dass der Inhalt vollständig vom Browser des Benutzers interpretiert wird. Ist dem jedoch nicht der Fall, so ist dies meist nicht besonders problematisch, da die Fehler meist nur kosmetischer Natur sind. Natürlich gibt es auch sehr komplexe Projekte, die

sich auf *CSS* für eine einwandfreie Funktion der Webseite verlassen. Einfachere Seiten, wie Blogs, Zeitungen u. Ä. sollten jedoch, auch ohne das graphische „Markup" verwendbar sein.

Grundsätzlich arbeiten die *Rendering Engines* der verschiedenen Browser (jene Unterprogramme, die für die graphische Darstellung verantwortlich sind) bei der Interpretation des CSS nicht so einheitlich, wie bei der HTML-Interpretation. Obwohl dies in den letzten Jahren immer besser wurde, müssen immer noch, des öfteren Engine-spezifische Befehle wie „webkit-" verwendet werden.

Bootstrap

Damit das Rad nicht immer wieder neu erfunden werden muss, gibt es mittlerweile unzählige *CSS-Frameworks*[4], die das Gestalten von Webseiten, durch bspw. vordefinierte *Classes* und Definitionen für einen Standard-Auftritt von Elementen, deutlich einfacher gestalten. Einer der größten und bekanntesten Anbieter ist **Bootstrap**.

Bootstrap ist ein Open Source-Baukasten für die Entwicklung mit den *üblichen* Web-Sprachen: HTML, CSS und JavaScript. Es bietet eine bequeme und schnelle Möglichkeit, Ideen zu prototypisieren oder gar ganze Web-Anwendungen zu erstellen. Zu den Stärken von *Bootstrap* gehören vordefinierte Variablen, Klassen, ein responsives Gitter-Raster-System, sowie viele andere vorgefertigte Komponenten und auf jQuery basierende Plugins[5].
Natürlich gibt es viele Alternativen, wie z.B.: „Skeleton", „Foundation" oder „UIKit", dennoch ist die Entscheidung auf *Bootstrap*, wegen der umfangreichen Dokumentation, großen *Community* (Internet Gemeinschaft), stetiger Weiterentwicklung, sowie der Vertretung durch globale Dienste wie **Twitter**, gefallen.

[4]Ein Framework ist kein fertiges Programm, sondern viel eher ein Programmiergerüst, mit dessen Hilfe ein Programm erstellt werden soll. Frameworks implementieren oft viele fertige „Baublöcke", die baukastenartig zusammengesetzt, und für den eigenen Zweck angepasst werden können.

[5]Ein Plugin, häufig auch als Add-on bezeichnet, erweitert ein bestehendes Programm (möglicherweise auch eine Entwicklungsumgebung oder ein Framework) um zusätzliche Funktionalitäten.

Templates

Dank der fähigen und vielfältigen *Community* von *Bootstrap*, gibt es sehr viele sog. *Templates*, also fertige Vorlagen, die im Gegensatz zu einem Framework bereits **ganz** fertige Webseiten oder Webseitenteile wie Login-Formulare, Navigationsleisten, Dashboards etc. zur freien Verwendung zur Verfügung stellen. Solche Templates sind u. a. auf *GitHub* (`https://github.com/`) oder auf der *Bootstrap Webseite* (`https://getbootstrap.com/`) meist zum kostenlosen Download erhältlich.

Das gewählte Template "Kiaalap Admin Template" ist eine sehr umfangreiche "Administrative-Vorlage" mit vielen Unterseiten. Sie wurde von dem *GitHub*-Benutzer Aigars Silkalns alias "Puikinsh", unter Verwendung der Bootstrap Version 4, zu Bedingungen der **MIT Lizenz** (siehe nachfolgende Seite), entwickelt; sie darf also ohne Einschränkungen weiterverwendet und -verbreitet werden.

Das **Kiialap Admin Template** bietet bereits eine sehr gute Vorlage für das Werkstätten-Management System WSS und ist generell optimal für jegliche Verwaltungs-Webseiten, die sich auf Universitäten oder Schulen richten, geeignet. Es können viele Teile (Formulare, Navigationsleisten, Nachrichten-System, Benutzerprofil-Vorlagen etc.) übernommen werden. Durch eine großflächige *JavaScript*-Implementierung sind alle Seiten *responsiv*[6] und interaktiv.

Das Template bietet allerdings nur das Design und den HTML-Code. Dies ist zwar ein guter Start und eine große Zeitersparnis, dennoch steckt der größte Aufwand in den PHP- und Datenbank-Algorithmen. Dazu kommt noch, dass bei der Erstellung des Templates keine *Templating-Systeme* (mehr dazu in den Folgeabschnitten) verwendet wurden, viele Code-Teile also statisch oder völlig redundant und für einen dynamischen Einsatz mit vielen Benutzern dementsprechend gänzlich ungeeignet sind. Infolgedessen müssen große Teile des Codes entfernt oder umgeschrieben werden.

[6] Bezeichnet die Fähigkeit einer Website, sich verschiedensten Endgeräten und den damit verbundenen Bildschirmgrößen anzupassen.

MIT Lizenzbestimmungen

MIT Lizenz oder *X11-Lizenz* ist eine freizügige Open-Source-Lizenz, ähnlich zur BSD-Lizenz[7]. Sie wurde vom Massachusetts Institute of Technology entworfen.

Originaltext: Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the „Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. [...] [5]

Übersetzung (sinngemäß): Jeder Person, die eine Kopie dieser Software und der dazugehörigen Dokumentation (die „Software") erhält, wird hiermit die Erlaubnis zur uneingeschränkten Behandlung der Software erteilt; einschließlich der Rechte zur Verwendung, zum Kopieren, Ändern, Zusammenführen, veröffentlichen, verteilen, unterlizenzieren und/oder verkaufen und Personen, denen die Software zur Verfügung gestellt wird, dies zu tun, unter den folgenden Bedingungen:

Der obige Urheberrechtsvermerk und dieser Erlaubnisschein sind in allen Kopien oder wesentlichen Teilen der Software enthalten. [...]

5.2.3. PHP

PHP ist eine der wenigen Skriptsprachen, die nicht direkt am *Client* ausgeführt werden, sondern vorher am *Server* durch einen *PHP-Interpreter* verarbeitet werden. Damit bietet es die Möglichkeit, logische Operationen durchzuführen und einen dynamischen Inhalt zu erzeugen. In diesem Fall bedeutet dynamisch jedoch nicht, dass dieser während dem Anzeigen auf dem Endgerät verändert werden kann - dafür müsste dieser neu angefordert werden. Durch diesen Umstand werden PHP-Dateien auch nicht mit dem geschriebenen Code an den Nutzer übertragen, sondern in der kompilierten Form. Während JavaScript und HTML sich vom Benutzer verändern und manipulieren lassen, kann dies bei PHP-Scripts nicht geschehen. PHP wird also zur Überprüfung von Werten, Einfügen von Datensätzen in die Datenbank (unter Einbindung von SQL) und

[7] Berkeley Software Distribution, Software unterliegend der BSD-Lizenz ist frei verwendbar.

Durchführung vieler anderer Programmier-üblicher (mathematischer) Aufgaben, eingesetzt.

5.2.4. Smarty

Wie im vorigen Unterabschnitt beschrieben wurde, kann durch *PHP*-Code ein dynamischer Inhalt erzeugt werden. Um diesen anzuzeigen, kann man ihn direkt im *HTML*-Teil einbinden oder ihn über eine *Template-Engine* einfügen. Ersteres hat den Nachteil, dass die Operationslogik und Webseitenstruktur vermischt werden; dadurch erscheint der Code sehr unstrukturiert und undurchschaubar. Möchte man nun Logik und Strukturierung bzw. Gestaltung trennen, sodass sie von zwei Personen respektive verändert werden können, benötigt man eine sogenannte Template-Engine. Diese erzeugt eigene Variablen, welche mit den Werten des *PHP*-Scripts befüllt und dann mit einer zusätzlichen Datei in das *HTML*-Gerüst eingebunden werden.
Dafür verwendet *Smarty* (`https://www.smarty.net/`) das Datei-Format *TPL*. Diese Dateien werden beim Anfordern der Website auf der Server-Seite in *PHP*-Dateien übersetzt und an den Client gesendet, wo sie im Browser dargestellt werden können. Smarty wurde gewählt, weil es zu den bekanntesten Vertretern der Szene gehört, für sicher gilt und stets aktuell ist. Außerdem verfügt Smarty über eine sehr umfangreiche und gut verständliche Dokumentation, sowie eine aktive Community-Unterstützung.

Laut Prof. Diernegger ergeben sich dadurch folgende Vorteile: („Smarty“ [6])

- flexible Trennung zwischen HTML und PHP
- verbesserte Code-Übersicht
- reibungslose Wartbarkeit des Codes
- leichte Wiederverwendbarkeit von Code-Teilen

Anwendungsbeispiel

Zur Verwendung von Smarty gehören grundsätzlich drei Dinge:

1. Die Seite wird als PHP-Datei an den Client gesendet (.php).
2. Die PHP-Datei beinhaltet u. a. den funktionalen Code und ruft eine Smarty-Template-Datei auf (.tpl).
3. Die Smarty-Template-Datei beinhaltet den Darstellungscode (HTML), kann aber auch andere TPL-Dateien aufrufen oder eigene, Smarty-spezifische Befehle zur Darstellung von dynamischen Inhalten verwenden.

```php
<?php
    /* Smarty Bibliothek einbinden */
    include('smarty-master/libs/Smarty.class.php');

    /* PHP Variable anlegen */
    $username = $_SESSION['username'];

    /* Smarty Objekt erstellen */
    $smarty = new Smarty;

    /* PHP Variable einer Smarty Variable zuweisen */
    $smarty->assign('username_smarty', $username);

    /* Template Datei aufrufen */
    $smarty->display('templates/index.tpl');
?>
```

Listing 5.2: index.php: Aufruf eines Smarty Templates

```
<head>
    <!-- Metadaten -->
    <title>Seitenname</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
    <link rel="stylesheet" href="css/index.css">
</head>
<body>
    <!-- Smarty ruft ein weiteres TPL File auf -->
    {include file='templates/top-nav.tpl'}

    <!-- Smarty Variable wird ausgegeben -->
    <p>Hello {$username_smarty}!</p>
    <br>

    <!-- Smarty Variable wird abgefragt -->
    {if $username_smarty === "counter"}

        <!-- Eine Schleife wird mit Smarty initiiert -->
        {for $foo=1 to 5}
            <li>{$foo}</li>
        {/for}

    {/if}

    <footer>
        <!-- Jahr wird mittels Smarty-Funktion ausgegeben -->
        <p>&copy; Copyright 2018-{'Y'|date}</p>
    </footer>
</body>
```

Listing 5.3: index.tpl Verwendung einer Template-Datei

Bei dem Beispiel fällt auf, dass zu einem der PHP-Code sehr sauber und ohne jegliches HTML geschrieben wird; und zu anderem, dass sich dynamische Smarty-Befehle mühelos in den HTML-Code integrieren. (Bei *index.tpl* wurde die HTML Syntax zwecks Lesbarkeit etwas vereinfacht/verkürzt.) Der Benutzername:

```
$username = $_SESSION['username'];
```

wird in dem Beispiel beim Login aus der Datenbank gelesen und in die *superglobale* Variable $_SESSION gespeichert; dieser Wert ist für jeden Benutzer dementsprechend unterschiedlich. Für einen bestimmten Benutzer („counter") wird eine Aufzählung ausgeführt. Dies soll nur die Möglichkeit veranschaulichen, verschiedene Benutzergruppen (z.B.: Schüler, Administratoren, Lehrer; der Status würde in jenem Fall abgefragt werden), unterschiedlich behandeln zu können.

Smarty bringt viele Vorteile, jedoch auch einen wesentlichen Nachteil: Jegliche PHP Dateien, die mit Smarty Templates ausgestattet wurden und an den Benutzer gesendet werden, müssen bevor sie durch den PHP-Interpreter verarbeitet werden können, durch die Smarty Bibliothek mit den TPL (Template) Files zusammengeführt und in reinen PHP/HTML Code übersetzt werden. Dies erfordert mit steigender Benutzerzahl (Informations-Varietät) exponentiell mehr Rechenleistung, aufgrund von schwacher Hardware sollte diese unbedingt geschont werden, deswegen wurde u.A. *Smarty Caching* implementiert; dieses soll das Abrufen redundanter Informationen verhindern.

Mehr zu Smarty-Caching im Unterabschnitt 7.2.1.

5.2.5. JavaScript

JavaScript ist eine Skriptsprache, die *typischerweise* Client-seitig ausgeführt wird, wodurch eine Veränderung des geladenen Inhalts ohne weiterer *Anfragen* erfolgen kann. Dadurch können bspw. Schaltflächen mit Funktionen hinterlegt werden, die die Seite verändern. Den grundlegenden Vorteil stellt hierbei die Entlastung des Servers dar, da so die Website ohne Nachladen abgewandelt werden kann. Somit können einige Funktionen, die sonst über *PHP* gelöst werden, direkt beim Client verarbeitet werden. Ein gravierender Nachteil ist jedoch, dass der Benutzer den Code manipulieren und *JavaScript* generell deaktivieren kann. Deshalb sollten Sicherheitsfunktionen, wie das Überprüfen von Eingaben, immer zweifach (also mit *JavaScript* Client-seitig und mit *PHP* Server-seitig) durchgeführt werden.

JQuery

JQuery ist eine *JavaScript*-Bibliothek, die unter der MIT-Lizenz steht und damit frei zur Bearbeitung und Wiederverwendung bereitgestellt wird. Mit dieser werden viele *JS*-Funktionen, die selbst zusammengestellt werden müssten, unter einem Befehl vereint. Vorrangig wird es für die Navigation und Manipulation des Document Object Model (dt. Dokumenten-Objekt-Model) (DOM) genutzt.

AJAX

AJAX ist ein Akronym für *Asynchronous JavaScript and XML* und wird für die asynchrone Datenübertragung zwischen dem Server und Webbrowser genutzt. Hierbei werden spezifische Teile in einem *HTML*-Dokument durch *JavaScript* und dem Aufrufen eines *PHP*-Files geändert. Dabei bleibt das gesamte Neuladen einer Website aus und Veränderungen wirken deutlich nahtloser.
Die wichtigsten Informationen einer Internetseite sollten im Optimalfall trotzdem direkt im *HTML*-File integriert sein, da das Nachladen von Teilen durch *AJAX* nur mit aktivem *JavaScript* funktioniert und ein geringer Teil der Nutzer dieses dauerhaft deaktiviert haben könnte.

5.2.6. SQL

SQL (Structured Query Language, dt. strukturierte Abfrage-Sprache) ist eine Datenbanksprache zum Festlegen von Strukturen in **relationalen** Datenbanken. Zugleich dient sie zum Bearbeiten und Abfragen der bestehenden Datensätze. SQL ist der Nachfolger von IBM Sequel und wird häufig als „Sequel" [´si:kwel] ausgesprochen.

In relationalen Datenbanken gibt es derzeit (Q1 2019) keine andere Alternative als SQL. Sobald also eine relationale Datenbank verwendet wird, muss auch in SQL *programmiert* werden. Allerdings kommt SQL in vielen unterschiedlichen Abwandlungen (mit verschiedenen Datenbanken), z.B. PostgreSQL, MariaDB, MySQL, Oracle Database uvm. vor. Die Befehle unterscheiden sich aber von Datenbank zu Datenbank nur marginal. Manche Datenbanken bieten größere oder komplexere Befehlssätze, die grundlegenden Befehle sind allerdings in ihrer Funktionalität und Syntax meistens gleich. Man kann bspw. davon ausgehen, dass Befehle wie UPDATE, TRUNCATE, INSERT INTO etc. überall gleich funktionieren; dennoch sollte man sich stets mit dem Benutzerhandbuch der jeweiligen Datenbank vertraut machen.
SQL kann entweder direkt in der Konsole des jeweiligen Datenbankmanagementsystems, oder mit Hilfe einer anderen Skript- oder Programmiersprache (wie PHP oder Java) ausgeführt werden.

MySQL

MySQL ist ein Datenbankverwaltungssystem (DBMS), das als Open-Source-Software zu einem der weltweit am häufigsten eingesetzten Systeme zählt. Es zeichnet sich besonders durch umfangreiche Dokumentation, hinreichende Community-Unterstützung, sowie eine Vielzahl von Beispielen aus diversen Quellen aus. Zudem wird MySQL von den größten (Internet-) Unternehmen der Welt verwendet, zu diesen zählen: NASA, YouTube, Bank of America, Verizon, CERN, Bayer, uvm. Vor allem sind beide Entwickler bereits mit MySQL vertraut, weswegen es als die beste Wahl zur Modellierung der Datenbank für das WSS erscheint.

Seit 2010 wird MySQL von der Oracle Corporation entwickelt und bietet folgende **Kenngrößen**:

- Die maximale Datenbankgröße ist nur durch den verfügbaren Speicher begrenzt.
- Die maximale Spaltenzahl beträgt ca. 4096 Spalten pro Tabelle (Datentypenabhängig), typisch jedoch zw. 250 und 1600 Spalten.
- Die maximale Anzahl von Zeilen ist pro Tabelle uneingeschränkt.
- Die interne Darstellung einer Tabelle hat eine maximale Zeilengrößenbegrenzung von 65.535 Byte.
- Die maximale Größe einer Tabelle beträgt 32 TB.
- Die maximale Größe eines Datensatzes beträgt 1,6 TB.
- Die maximale Größe einer Zelle ist auf etwa 1 GB beschränkt.

Obige Informationen wurden von (`https://dev.mysql.com/doc/`)[7] entnommen.

Wesentliche Eigenschaften

- verschachtelte Abfragen mit Unterabfragen (Subselects)
- referentielle Integrität (u.a. Constraints und Fremdschlüssel)
- Mengenoperationen
- Vererbung von Tabellen
- Schnittstellen zu vielen Programmiersprachen, u.a. C++, Object Pascal, Java, PHP, Perl, Python, Ruby, .NET

- Erweiterbarkeit durch Funktionen, selbstdefinierbare Datentypen und Operatoren

Die aufgeführten Kenngrößen und Eigenschaften machen deutlich, dass MySQL bereits in dessen einfachster Ausführung (non Enterprise) eine sehr starke und uneingeschränkte Grundlage für den Aufbau einer Datenbank darstellt. Zudem sind die meisten unter „Eigenschaften" aufgeführten *Limitationen* so absurd hoch gesetzt, dass es für das Projektteam überhaupt nicht in Frage kommt, sich um diese zu sorgen. Bspw. ist die maximale Tabellengröße auf 32 TB beschränkt, das Projektteam rechnet aber mit einer Größe von < 1 GB, im gesamten Lebenszyklus der Software für die **gesamte Datenbank**.

Ebenso wenig sind die Spezialfunktionen wie Objektorientierung von Relevanz, da sich die Entwicklung im Wesentlichen nur auf die simpelsten und grundlegendsten Funktionalitäten einer Datenbank beschränkt.

MySQLi

MySQLi (wobei das *i* für „improved" steht) ist eine Erweiterung von PHP, wobei der Vorgänger seit Version 7.0 nicht mehr verfügbar ist. Ein Vorteil gegenüber der ursprünglichen Version ist die Objektorientierung. Außerdem können durch sog. *Prepared Statements SQL-Injections* verhindert werden.

Beispiel

```php
<?php
    /* Mit der Datenbank verbinden */
    $conn = new mysqli($db_host, $db_user, $db_pass, $db_name);
    /* Bei Fehlschlag Fehlermeldung ausgeben */
    if ($conn->connect_error) {
        die("Verbindung fehlgeschlagen: " . $conn->connect_error);
    }
    /* Anweisung vorbereiten */
    $stmt = $conn->prepare("SELECT * FROM Table1 WHERE ID = ?");
    /* Variable einbinden */
    $stmt->bind_param("s", $id);
    /* Anweisung vollziehen */
    if(false === $stmt->execute()) {
        die("Fehler beim Auslesen der Daten.");
    }
    /* Daten in ein Array speichern */
    $res = $stmt->get_result();
    /* Anweisung schliessen */
    $stmt->close();
?>
```

Listing 5.4: *MySQLi*, Verwendung von Prepared Statements

Insbesondere fällt auf, dass bei einem *Prepared Statement* die Variablen nicht als *Strings* als Teil der Abfrage (Query) eingefügt, sondern mit der Methode:

```
$stmt->bind_param("s", $id);
```

übergeben werden; dabei gibt die Folge der „?" (Fragezeichen) im Query auch die chronologische Abfolge und Anzahl der von mit „bind_param(...);" eingebundenen Variablen an. Im ersten Parameter der Methode wird für jede Variable chronologisch der Datentyp etabliert. Man unterscheidet:

- *s* - String
- *i* - Integer
- *b* - Blob (Binary Large Objects)

Dabei können Integers auch als Strings übergeben werden, denn PHP führt automatisch eine Typumwandlung durch; dennoch, sollte der Rechenleistung wegen, stets der korrekte Datentyp verwendet werden.

Genau diese Methodik ist der springende Punkt, der SQL-Injection-Attacken verhindern soll. Denn bei einer SQL-Injection werden die Variablen angegriffen, die in Queries enthalten sind; Prepared Statements schließen jene Sicherheitslücke und machen zudem den Code viel lesbarer.

6. Datenbank-Aufbau

Dieses kurze Kapitel behandelt überwiegend den internen Aufbau und die Funktionalität der Datenbank. In Abschnitt 5.2.6 wurde bereits etabliert, dass als DBMS, das von der Oracle Corporation entwickelte **MySQL**, verwendet wird. Dort wurden auch dessen Vor- und Nachteile geschildert.

6.1. Normalformen und Regeln

Eine Datenbank soll stets mit den beiden wichtigsten Eigenschaften im Hinterkopf modelliert werden:

I Redundanzfreiheit

II Datenintegrität

Redundanzfreiheit liegt dann vor, wenn jede Information genau einmal in der Datenbank vorkommt. Dies ist eigentlich leicht erreichbar; der Entwickler darf nur die Übersicht über seine Tabellen und Datensätze nicht verlieren. Es soll also stets geprüft werden, ob die zu speichernde Information sicher einzigartig ist.

Datenintegrität oder auch *Datenkonsistenz* bezieht sich auf die Korrektheit gespeicherter Daten (Attributwerte). Inhalte einer Datenbank sind dann konsistent, wenn sie eindeutig und einheitlich sind, d.h. u.a., dass Daten wie bspw. Adressdaten korrekt aufgenommen, formatiert und gespeichert wurden. Solche Fehler können einem leicht unterlaufen; bspw. stellt die Speicherung von Umlauten bei falsch formatierten Tabellen (Relationen) ein Problem dar. Ebenso werden Namen oft nicht einheitlich gespeichert. Wenn z.B. Vor- und Nachnamen verwechselt werden, oder ein Kunde mehrere Vornamen führt, sind jene Datensätze schwer verwertbar.

Um die beiden Eigenschaften *Redundanzfreiheit* und *Datenintegrität* leichter verfolgen zu können, wurden Normalformen eingeführt, nach denen entwickelt werden soll. Es gibt genau *sechs* Normalformen, die letzten beiden beziehen sich aber auf hochwertige **m:n** Beziehungen, weswegen diese bei kleinen Datenbanken vernachlässigt werden können.

Die Normalformen wie sie nach Prof. DI Martin Walter im Rahmen des FSST Unterrichts erläutert wurden:

1. Normalform

Eine Relation ist in erster Normalform, wenn alle Attributwerte elementar sind. In einfachen Worten: Jeder Attributwert ist eindeutig und somit ein eigenes Element. [8]

2. Normalform

Eine Relation ist in zweiter Normalform, wenn sie in erster Normalform ist, und jedes Nicht-Schlüssel-Attribut von jedem Schlüssel *voll funktional abhängig* ist. [8]

Funktional abhängig: Es gibt für jeden Wert A genau einen Wert B.

Voll funktional abhängig: ein Nicht-Schlüssel-Attribut ist funktional abhängig vom zusammengesetzten Schlüssel.

3. Normalform

Eine Relation ist dann in dritter Normalform, wenn sie in zweiter Normalform ist und alle Nicht-Schlüssel-Attribute frei von *transitiven Abhängigkeiten* sind. Eine *transitive Abhängigkeit* besteht dann, wenn die Attributmenge B einer Relation von der Attributmenge A abhängt, und eine Menge C funktional abhängig von B ist. C ist hier transitiv abhängig von A. [8]

Boyce-Codd-Normalform (BCNF)

Eine Relation ist dann in BCNF, wenn sie in dritter Normalform ist und jede Determinante einen Schlüssel bilden kann (ein Superschlüssel ist).

Determinate sind Attribute, die entweder durch sich selbst, oder durch Kombinationen mit Attributen andere Attribute bestimmen können. BCNF besagt, dass Determinante so eindeutig sein müssen, dass sie als sog. „Kandidatenschlüssel" dienen könnten, also den Primärschlüssel ersetzen könnten.
[9] E. F. Codd

6.2. Relationales Datenbankmodell

Datenbanken werden entweder in Form von Entity-Relationship-Modellen (ER-Modellen) oder technischen, relationalen Konstrukten dargestellt. Relationale Datenbankmodelle repräsentieren die Datenbank am besten und zeichnen sich durch Schlüsselverknüpfungen von Relationen, sowie durch die Darstellung von Attributen aus. Solche Visualisierungen sind leicht nachvollziehbar, und enthalten bereits alle Schlüssel und Attribute.
ER-Modelle sind weniger technisch; sie enthalten oft nur die jeweiligen Tabellen und kennzeichnen, in welchem Zusammenhang diese mit anderen Tabellen stehen. Es wurde ein **relationales Datenbankmodell** von der fertigen WSS-Datenbank angefertigt:

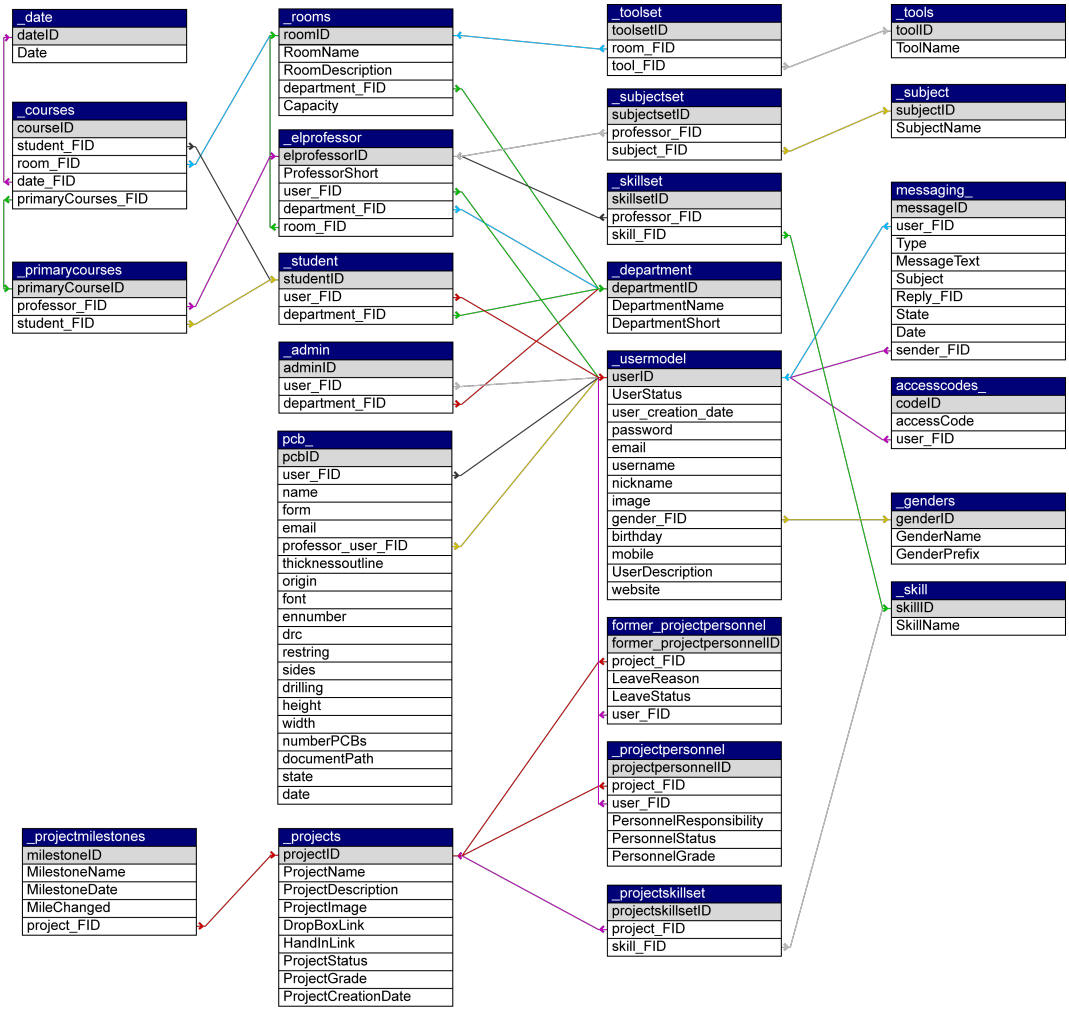


Abbildung 6.1.: WSS-Datenbankstruktur, illustriert im relationalen Zusammenhang

6.3. Beschreibung der Relationen

Insgesamt besteht die Datenbank aus 24 Relationen (Tabellen). Dabei wurden direkt interaktive Tabellen durch einen *Unterstrich* vor dem Namen, und funktionale Tabellen durch einen *Unterstrich* hinter dem Namen gekennzeichnet. In der Abbildung 6.1 wurden die Tabellennamen *blau* und die Primärschlüssel *grau* gefärbt. Schlüsselbeziehungen wurden durch farbige Linien dargestellt. Primärschlüssel (Primary Key) führen immer den Zusatz **ID** und stehen ganz oben in der Tabelle, während Fremdschlüssel (Foreign Key) irgendwo in der Tabelle stehen können, jedoch immer den Zusatz **FID** führen. Da nicht alle Tabellen von besonderer Wichtigkeit sind, werden hier nur die wichtigsten geschildert.

6.3.1. _usermodel

Dies ist eine der wichtigsten Tabellen der gesamten Datenbank. Sie stellt die Grundlage für das gesamte Multischicht-Benutzersystem des WSS. Jeder registrierte Benutzer ist in jener Tabelle gespeichert. Das Feld **UserStatus** klassifiziert die Berechtigungsstufe des Anwenders (siehe Abschnitt 7.3). Im Feld **user_creation_date** wird bei Erstellung des Nutzerkontos ein Zeitstempel gespeichert. Felder **email**, **username** und **password** sind Pflichtinhalte und werden bei der Erstellung des Kontos befüllt. Andere Felder speichern persönliche Daten, die meistens nach Wunsch angegeben werden können und auf dem Profil angezeigt werden.

_usermodel ist mit der Tabelle **_genders** verknüpft, diese beinhaltet verschiedene Geschlechter und deren Präfixe, z.B. Male - Mr. Das Geschlecht kann auf dem Benutzerprofil *freiwillig* angegeben werden, und wird u. a. auf der öffentlichen Ansicht des Profils angezeigt.

6.3.2. _admin

Alle Benutzer, die *Administratoren* oder *Superadmins* sind, werden zusätzlich in dieser, von *_usermodel* abgeleiteten Tabelle, durch den Fremdschlüssel jener, referenziert. Die Tabelle beinhaltet lediglich noch eine zusätzliche Verknüpfung mit **_department**, in der die Abteilungen und deren Kürzel gespeichert werden. Die Angabe des *Departments* ist freiwillig.

6.3.3. _elprofessor

Jeder Benutzer, der einen Raum betreuen und somit Stunden halten kann, also ein Lehrer ist, wird zusätzlich in dieser Tabelle referenziert. Sie beinhaltet

dementsprechend eine Schlüsselverknüpfung mit der Tabelle **_rooms**, außerdem einen einzigartigen Lehrerkürzel, der als *Kandidatenschlüssel* gewertet werden kann und ist ebenso wie **_admin**, mit der Tabelle **_department** verknüpft.

6.3.4. _student

Jeder Anwender, der ein Schüler ist, somit Stunden buchen und Projekte erstellen darf, wird in dieser, von **_usermodel** abgeleiteten Tabelle, durch ihren Fremdschlüssel referenziert. Sie ist mit der Tabelle **_admin** völlig ident und beinhaltet keine Verknüpfungen zu schülerspezifischen Tabellen, sondern wird durch jene referenziert.

6.3.5. _rooms

Diese Tabelle beinhaltet Informationen über verfügbare Räume, solche sind Kapazität, Beschreibung, Name und die Abteilung. Klarerweiße wird hier also ebenfalls die Tabelle **_department** referenziert.

6.3.6. _courses

Courses, also sinngemäß Stunden oder Einteilungen, werden durch Schüler erstellt, besucht und durch Lehrer gehalten. Die Tabelle ist mit der Relation **_date** verknüpft. *Dates* sind Zeitpunkte, zu welchen *Courses* stattfinden, die durch Administratoren vorgegeben (erstellt) werden. Die Tabelle referenziert außerdem die vorher beschriebene Tabelle **_rooms**, sowie **_primarycourses**. *Primary Courses* oder Pflichtkurse sind eine Ableitung von *Courses*, die zwar dieselben sind, für bestimmte Schüler aber durch Lehrer angelegt werden können und somit verpflichtend sind. **_courses** beinhaltet nur eine Referenz zum Schüler, aber keine zum Lehrer, weil der Lehrer nicht an Kurse, sondern an Räume (einen Raum) gebunden ist.

6.3.7. _projects

_projects stellt die Grundlage einer weiteren, wesentlichen Funktion von WSS - den Projekten - dar. Von dieser Tabelle sind dementsprechend *vier* weitere, auf jenes Feature bezogene Relationen, abhängig. Wenn ein Projekt durch einen Schüler angelegt wird, werden die Informationen dessen in dieser Tabelle gespeichert; z.B. Name, Beschreibung, Pfad zum Projektbild, Status etc. Ebenso wird bei der Projekterstellung ein Zeitstempel gespeichert, sowie die Note, wenn das Projekt abgegeben und durch den Betreuungslehrer benotet wurde.

_projectmilestones

Beinhaltet Informationen über sogenannte *Projekt-Meilensteine*, und ob diese verändert wurden.

6.3.8. _projectpersonnel

Es ist auf der Projektebene die äquivalente Tabelle zu **_usermodel**, ist aber gleichzeitig von dieser abgeleitet. Hier werden alle Teilnehmer eines Projektes, ihre Berechtigungsstufen am Projekt, sowie persönliche Merkmale wie Verantwortungsbereiche oder individuelle Noten gespeichert. Die Tabelle enthält dementsprechend eine Schlüsselverknüpfung zu **_projects**.

former_projectpersonnel

Diese Tabelle ist für jegliche Funktionalität unwesentlich; sie beinhaltet lediglich alle ausgetretenen Projektteilnehmer aus allen Projekten, sowie deren Austrittsgründe und Berechtigungsstufen. Sie dient zur Statistikerstellung, sowie Aufzeichnung vom Userverhalten.

6.3.9. accesscodes_

Die Registrierung der Schüler erfolgt über sogenannte *Access-Codes* (Zugangsschlüssel), die nach ihrer Generierung in dieser Tabelle gespeichert werden. Dementsprechend enthält die Tabelle die Zugangsschlüssel, sowie einen Fremdschlüssel von der Tabelle **_usermodel**, um feststellen zu können, welcher Benutzer welchen Schlüssel verwendet hat; oder viel eher welche Schlüssel bereits vergeben wurden, damit ein Schlüssel nicht mehrmals verwendbar ist.

6.3.10. messaging_

Jene Relation ist das Rückgrat des internen Kurznachrichtensystems. Wenn ein Benutzer oder das System eine Nachricht versendet, wird deren Inhalt, die Versender-, sowie Empfänger-IDs, Typ, Betreff, Datum und Zustand - also ob sie gelesen wurde - in der Tabelle *messaging_* gespeichert.

6.3.11. pcb_

Das WSS bietet ihren Benutzern die Möglichkeit Leiterplattenbestellungen aufzugeben, welche dann durch berechtigte Personen eingesehen und verarbeitet werden können. Die Ergebnisse jener Bestellungen werden in der Tabelle *pcb_* gespeichert. Sie beinhaltet eine *doppelte* Verknüpfung zu **_usermodel**: Dabei referenziert ein Fremdschlüsselattribut den Auftraggeber (meistens Schüler),

während sich das andere auf den *Betreuer* (Lehrer/Admin) jener Bestellung bezieht. Die Tabelle steht großteils für technische Daten der Bestellung zur Verfügung, beinhaltet aber auch ein zusätzliches Email-Attribut, welches die Möglichkeit bietet, zusätzliche zur bereits bekannten, diese zu archivieren.

6.3.12. m:n Tabellen

Manche Tabellen sind über Zwischentabellen miteinander verknüpft. Diese **Zwischentabellen** sind:

- _toolset
- _subjectset
- _skillset
- _projektskillset

Deren **korrespondierende Tabellen** sind:

- _tools - *Geräte für Räume*
- _subject - *Fächer für Lehrer*
- _skill - *Fähigkeiten für Lehrer*

Eine Zwischentabelle dient nur dazu, zwei Tabellen über ihre Schlüssel-Attribute zu verknüpfen, beinhaltet somit nur Fremdschlüssel. Dies ist notwendig, wenn die Anzahl der Verknüpfungen zweier Tabellen > 1 ist. Z.B.: Verknüpft die Tabelle *_toolset*, **_rooms** und **_tools** miteinander, wobei in *_tools* die Geräte eines Raumes gespeichert werden. Durch eine solche Verknüpfung ist es möglich, dass ein Raum mehrere Geräte besitzt.

7. Funktionen der Website

Das nachfolgende Kapitel erläutert sowohl die technischen Aspekte und Algorithmen, nach denen das System operiert; als auch behandelt es die Möglichkeiten und Operationen, welche den Benutzern der Software zur Verfügung stehen.

7.1. Erreichbarkeit

Die Webseite ist entweder lokal unter `10.53.57.201/wss` oder global über die URL `https://prada.privatevoid.net/wss` erreichbar. Lokal bedeutet konkret, dass der Zugriff innerhalb des Netzwerks der *HTBLuVA St. Pölten*, über die dort vergebene (lokale) *IP-Adresse*, erfolgen kann. Dementsprechend ist die *HTL* der Ort, an dem sich die WSS-Rechenwerke befinden. Der lokale Zugriff ist meistens marginal schneller, da die globale Alternative (über das *WWW*) sich auf einen *Virtual Private Network (VPN)*-Server verlässt, der den gesamten Datenverkehr lediglich auf den *WSS-Webserver* in der *HTL* umleitet. Allerdings hat der Zugang über das *WWW* einige Vorteile.
Zum einen ist dies die Möglichkeit einer sicheren Verbindung über das *Hypertext Transfer Protocol Secure (HTTPS)*. Als Folge dessen wurde auch die Sicherheitsfunktion *HTTP Strict Transport Security (HSTS)* durch die Kollegen von ***Private Void*** implementiert. Hierbei erzwingt der Server durch Informationen im *„HTTP Response Header*[1]" eine verschlüsselte Verbindung über *HTTPS* mit dem Browser des Benutzers. *HSTS* soll auch vor *Downgrade Attacken*[2] und *Session Hijacking*[3] schützen.
Nicht zuletzt gibt es auch einen Domain-Namen. Der Benutzer muss sich somit nicht die IP-Adresse (Abfolge von Ziffern), sondern nur eine kurze Zeichenkette merken.

---

[1] *„Hypertext Transfer Protocol (HTTP)-Antwort"* beinhaltet Header-Informationen des Servers (u. a. Metadaten) und den eigentlichen anzuzeigenden Inhalt.

[2] Form eines kryptografischen Angriffs auf ein Computersystem oder ein Kommunikationsprotokoll, das dazu führt, dass ein qualitativ hochwertiger Betriebsmodus (z. B. eine verschlüsselte Verbindung) zugunsten eines älteren Betriebsmodus niedrigerer Qualität aufgegeben wird.

[3] Auch als Cookie-Hijacking bezeichnet; ist die Nutzung einer gültigen Sitzung (*Session*), um einen unberechtigten Zugriff auf Informationen oder Dienste in einem Computersystem zu erhalten.

7.2. Caching

Caching ist ein Verfahren, bei dem Ressourcen anhand bestimmter Kriterien in einem temporären Speicher (Cache) abgelegt werden, mit dem Ziel redundante Datenübertragungen und Serveranfragen zu verhindern, sowie Latenz- und Wartezeiten zu verringern. Dies führt Client-seitig zu einer Entlastung des Internet-Traffics und viel wichtiger, zu einer Server-seitigen Leistungseinsparung. *Caching* wird häufig auf der Anwender-Seite als eine Funktion des Browsers implementiert. Dabei werden häufig Bilder, Bibliotheken (z.B.: Bootstrap) und CSS-Dateien lokal abgelegt. Dies verbessert Seiten-Ladezeiten oft erheblich und sorgt für eine optimierte Benutzererfahrung.

Caching kann aber auch auf dem Server implementiert werden, um bspw. Abfragen aus der Datenbank zu reduzieren, indem jene Daten für eine kurze Zeit entweder in den Arbeitsspeicher geladen, oder in separaten Dateien abgelegt werden. Dabei haben Datenbank-Managementsysteme (DBMS) in den meisten Fällen bereits ihre eigenen, internen Optimierungsalgorithmen sowie Caching-Verfahren. Jegliche Art von effizienzsteigerndem Code, wirkt sich aber erfahrungsgemäß positiv auf das Gesamtsystem aus. Es darf jedoch nicht außer Acht gelassen werden, dass *Caching* **immer** nur ein temporäres Verfahren ist, und keine Datenbank ersetzen soll. *Caches* sollen eine möglichst kurze *Lebenszeit*[4] haben, um Dateninkonsistenz zu vermeiden.

7.2.1. Caching in Smarty

Smarty bringt viele Vorteile, jedoch auch einen wesentlichen Nachteil: Jegliche PHP-Dateien, die an den Benutzer gesendet werden, müssen, bevor sie durch den PHP-Interpreter verarbeitet werden können, durch die Smarty Bibliothek mit den TPL (Template) Files zusammengeführt und in reinen PHP/HTML Code übersetzt werden. Zwar werden die Resultate jenes Zusammenführens in sogenannten „.tpl.cache.php" Dateien abgelegt, diese sind jedoch nur verwendbar, wenn sehr wenige dynamische Inhalte, dazu sehr selten, abgefragt werden. Denn je öfter variable Inhalte angefordert werden, desto öfter muss neu-*kompiliert* werden; dies ist zwar bei geringen Benutzerzahlen nicht problematisch, jedoch steigt mit zunehmender Userzahl die Auslastung in etwa exponentiell. Wenn also sehr viel Benutzer jene Seite simultan verwenden und jeder andere Inhalte anfordert, belastet es nicht nur die Datenbank, sondern auch zusätzlich den Webserver durch das permanente neu-kompilieren.
Dieses Problem ist den Entwicklern von Smarty durchaus bekannt, weswegen Möglichkeiten für *dynamisches* und *statisches* Caching integriert sind.

[4] Zeit, nach der bestimmte Dateien erneuert oder gelöscht werden.

Smarty bietet also nicht nur die Möglichkeit *statische* Caches zu erzeugen, also Caches, die einfach nur die gesamte Seite speichern, sondern auch *dynamische* Caches anzulegen.
Caching wird als Teil des eigentlichen PHP-Codes implementiert; Cacheanweisungen können aber auch durch bestimmte Smarty-Befehle in den Template-Dateien vorkommen (z.B.: *{nocache}*, wenn ein Inhalt unter keinen Umständen temporär gespeichert werden darf). Dynamisches Caching hat den Vorteil, dass bestimmte Seitenteile unterschiedlich behandelt werden können. Beispielsweise kann der Teil der Seite mit Datenbank-Abfragen nur dann zum Kompilieren abgerufen werden, wenn kein bestehender Cache zur Verfügung steht. Während andere Teile der Seite, so wie jene, die Benutzereingaben empfangen (z.B.: auf abgesendete HTML-Formulare prüfen), also **immer** andere Daten bekommen und somit nicht gecached werden **können**; vom Zwischenspeichern ausgeschlossen werden müssen. Zudem kann auch eine Cache-Lebenszeit (Cache Lifetime) festgelegt werden, nach der der Cache spätestens geleert werden muss. Der Cache kann aber auch manuell geleert werden.

Das WSS implementiert Caching so, dass für jeden Benutzer, auf bestimmten Seiten wie Benutzerprofilen, eine eigene Cache Datei angelegt wird, die also nur für diesen einen Benutzer gültig ist. Beim Aktualisieren von Informationen (z.B.: durch den Benutzer) auf jener Seite wird der Seitencache im Prozess geleert und das Template bei der nächsten Anforderung neu-kompiliert; der Cache dabei neu erstellt.
Es lässt sich allerdings nicht vermeiden, dass manche Informations-Änderungen das Leeren des Caches nicht verursachen (absichtlich oder aus Fahrlässigkeit) und es somit zu einer *visuellen* Informationen-Inkonsistenz kommen kann. Die Einträge sind zwar richtig in der Datenbank gespeichert, es werden allerdings durch alten Cache obsolete Informationen angezeigt.
Deswegen wurde eine Cache-Lebenszeit von **maximal** 12 Stunden implementiert. Der Cache wird also nach 12 Stunden automatisch gelöscht und beim nächsten Aufruf neu generiert. Meistens beträgt die Lifetime allerdings 30 Minuten bis zwei Stunden.

Beispiel: Implementierung

```
1  // Smarty Objekt erstellen
2  $smarty = new Smarty();
3  // Freischaltung von Caching
4  $smarty->caching = 2;
5  // Betroffene Dateien sollen nach Aenderungen untersucht werden
6  $smarty->compile_check = true;
7  // Lebenszeit von 12 Stunden
8  $smarty->cache_lifetime = 43200;
9
10 if(!$smarty->isCached('templates/Profile.tpl', $_SESSION['user']))
11 {
12     // Daten aus der Datenbank auslesen
13     ...
14 } else {
15     // WSS-VDA
16     $valid_genID =
17     json_decode(file_get_contents('vars/valid_genID'), true);
18 }
```

Listing 7.1: Implementierung der Smarty-Cachingfunktion in profile.php

Das Beispiel entspricht 1:1 dem Code auf *profile.php*. Dabei fällt auf, wie einfach und unproblematisch das Smarty Caching aktiviert und in den Code implementiert werden kann. Die meiste Arbeit fällt auf die Abfrage in Zeile **10**: Hier wird geprüft, ob ein gültiger Cache vorhanden ist. Wenn ja, wird die *else* Anweisung ausgeführt, wo das hauseigene Archivierungsprotokoll **WSS-VDA** zum Einsatz kommt. Sollte allerdings kein gültiger Cache vorhanden sein, werden diverse Datenbank-Abfragen ausgeführt, um die angeforderten Daten herzustellen und einen neuen Cache zu kompilieren.

7.2.2. WSS-VDA

Workshop Scheduling System - Variable Data Archiving ist das hauseigene Protokoll zum temporären Archivieren von Daten in Form von *JSON*[5] *kodierten Arrays*, die durch das Caching-Protokoll von Smarty unzugänglich gemacht oder nicht gespeichert wurden.
WSS-VDA geht also Hand in Hand mit Smarty und stellt dessen inoffizielle Erweiterung dar, die für einen *sehr* spezifischen Einsatz auf der WSS-Seite von Kiril Vereshchagin entwickelt wurde.

Das WSS bietet häufig die Möglichkeit, aus einer Reihe von Optionsschaltflächen einen Wert auszuwählen, bspw. „Skills" bei *project-profile.php*. Wenn

[5] JavaScript Object Notation ist ein, für den Menschen lesbares, Textformat, das häufig für Datenaustausch zwischen Anwendungen verwendet wird.

ein Wert ausgewählt wurde, wird die jeweilige Feld-ID (Zahl) an das PHP weitergegeben, wo sie in die Datenbank gespeichert wird.
Da jegliche HTML-Werte allerdings durch den Benutzer verändert werden können, also auch IDs, werden Benutzereingaben Server-seitig auf Gültigkeit geprüft (siehe Unterabschnitt 8.1.4). Dabei muss das Skript wissen, welche IDs gültig sind, um diese mit der gesendeten ID vergleichen zu können; konkret ist die Funktion *select_validator* gemeint. Die gültigen IDs werden zusammen mit anderen Werten aus der Datenbank abgerufen und in einer Variable gespeichert. Da jene Variable aber aus Sicherheitsgründen nirgends im TPL-File vorkommt, wird sie durch Smarty nicht gecached. Das WSS-VDA bietet eine Möglichkeit, die Werte jener Variablen temporär in Dateien auf dem Server abzulegen. Diese wären zwar theoretisch immer noch öffentlich einsehbar, jedoch nicht veränderbar, und bieten nicht über mehr Informationen als das HTML-Formular selbst.

Das Protokoll ist gegeben simpel implementiert und stellt nur einen Versuch dar, möglichst effizienten Code zu entwickeln, indem redundante Aufrufe und Abfragen durch *fragliche* Methoden eingespart werden. Weiß ein Entwickler also nicht, dass ein solcher Caching-Zusatz implementiert wurde, könnten ihm potentielle Fehler über Daten-Inkonsistenz unterlaufen.

Technische Implementierung

Ähnlich wie im Beispiel 7.1 dargestellt, muss sichergestellt werden, dass Daten überhaupt aus der Datenbank gelesen wurden. Dies passiert immer, wenn der Smarty Cache abgelaufen, oder nicht vorhanden ist (siehe Bsp. 7.1, Zeile 10). Wenn also Werte aus der Datenbank gelesen werden, um später für den Benutzer dargestellt werden zu können, werden auch gültige IDs in einer Variable gespeichert. Diese IDs werden durch das WSS-VDA im JSON-Arrayformat in einer Datei auf dem Server abgelegt. Dabei bietet die PHP-Standardfunktion *json_encode*, sowie dessen Kontra *json_decode*, eine hervorragende Möglichkeit zum Kodieren/Dekodieren solcher Arrays. Der Name der Datei kann durch den Programmierer frei gewählt werden, denn diese wird später zum Auslesen jener gültigen *IDs* verwendet, siehe dazu Bsp. 7.1, Zeile 17. Die IDs werden dann erneut in eine Variable gespeichert und können der jeweiligen Validierungsfunktion (*select_validator*) übergeben werden.

So könnten die Inhalte einer JSON-kodierten Datei aussehen:

```
["14","6","11","10","9","12","16","4","3","1","2","5"]
```

Dabei fällt auf, dass jede *ID* durch doppelte Anführungszeichen eingeschlossen, sowie durch Kommas getrennt ist, während sich die gesamte Zeichenkette in

eckigen Klammern befindet. Dies erlaubt es theoretisch mehrere solcher *Strings* in einem File abzulegen, wobei es eher empfohlen wird, für jede ID-Kette eine neue Datei anzulegen. Ebenso muss darauf geachtet werden, dass die Dateien stets aktualisiert werden, bspw. sollen die Dateien bei jedem neu-Kompilieren des Smarty Caches neu angelegt werden.

7.3. Benutzerebenen

Das WSS ist eine administrative Web-Oberfläche zum Organisieren von Stundenplänen, Projekten, Pflichtveranstaltungen etc. Dementsprechend ist es keine statische Webseite, sondern eine dynamische Software, die auf Benutzer mit unterschiedlichen Berechtigungsstufen ausgelegt ist. Das WSS ist ein geschlossenes System, wodurch es nur durch ausgewählte Teilnehmer verwendbar ist.

Es gibt 4 Benutzerarten:

- *Superadmin*
- *Admin*
- *Professor*
  - *PCB-Professor*
- *Student*

Die Hierarchie ist hier eindeutig: Je weiter oben man ist, desto mehr Möglichkeiten hat man. Dabei ist der *PCB-Professor* eine Ableitung von *Professor*, dessen Berechtigungen auf die Verwaltung von Leiterplattenbestellungen **erweitert** wurden. Es gibt allerdings für jede Berechtigungsstufe auch Ebenenspezifische Fähigkeiten; dies wird in den nachfolgenden Unterabschnitten noch genauer erläutert.

Grundsätzlich gilt Folgendes: Ein *Superadmin* kann Konten für Administratoren und Lehrer anlegen. Ein Administrator kann nur Lehrerkonten anlegen. Schüler müssen sich immer **selbst**, mit Hilfe der sogenannten **Access-Codes** (Zugangscodes), registrieren. Diese können von jeder der anderen drei Benutzergruppen angefordert werden. Die Registrierung mittels *Access-Code* wurde überwiegend aus Datenschutzgründen, aber auch zur Entlastung von Lehrern und Administratoren, implementiert.
Der einzige Unterschied zwischen dem *Superadmin* und den *Admins* besteht darin, dass der Superadmin Administrator-Konten anlegen und löschen kann. Es gibt dementsprechend auch mehr Admins als Superadmins, außerdem besteht keine handliche Möglichkeit, ein Superadmin-Konto zu erzeugen. Dazu müssen die Berechtigungen manuell in der Datenbank geändert werden.

7.4. Registrierung

7.4.1. Access-Codes

Access-Codes werden für die selbstständige Registrierung von Schülern benötigt. Schüler sollen sich selbst registrieren können, damit weniger Aufwand für Administratoren oder Lehrer entsteht. Dabei ist jedoch nicht ausschließbar, dass diese die Funktion missbrauchen und mehrere Accounts pro Schüler erstellen. Damit kann einerseits nicht mehr herausgefunden werden, welcher Account zu welchem Schüler gehört und andererseits wird die Datenbank mit unnötig vielen Eintragen befüllt.

Deshalb muss bei jeder Registrierung eine valide Kombination von Ziffern und Buchstaben eingegeben werden (der sogenannte *Access-Code*), damit diese erfolgreich durchgeführt wird. Schüler können diese bei einem beliebigen Lehrer beantragen, der standardmäßig auf eine Liste mit 2000 Einträgen im WSS zugriff hat. Dabei liegt die Verantwortung bei ihm, einem Schüler nicht mehrere Codes zu geben. Sobald dieser Code verwendet wurde, wird dies in der Datenbank vermerkt und es ist für jeden Lehrer/Administrator einsehbar, wer welchen *Access-Code* benutzt hat.

Mit einem Lehrer- oder Administratorkonto (darunter auch *Superadmins*) kann man auf die Übersicht aller Codes, sowohl benutzte als auch noch offene, zugreifen. Diese kann in beliebiger Anzahl in eine *Excel*-, *JSON*-, *XML*-, *CSV*-, *TXT*- oder *SQL*-Datei exportiert und anschließend ausgedruckt werden.

Um die *Access-Codes* zu generieren, wurde die Online-Anwendung *RANDOM.ORG* verwendet. Betrieben von Randomness und Integrity Services Ltd. verspricht das Unternehmen die Generierung *echter* Zufallsergebnisse, welche nicht vorhersehbar sind. **Auf ihrer Webseite (`https://www.random.org/`) wird geschildert:**

„RANDOM.ORG bietet jedem im Internet echte Zufallszahlen. Die Zufälligkeit kommt von atmosphärischem Rauschen, das für viele Zwecke besser ist als die Pseudozufallszahlenalgorithmen, die typischerweise in Computerprogrammen verwendet werden." [11] (sinngemäß übersetzt)

Dabei wurde der „Random String Generator" mit den folgenden Einstellungen verwendet:

- Generate: *2000* random Strings
- Each string should be: *8* characters long
- Allow:
  - *Numeric digits (0-9)*
  - *Uppercase letters (A-Z)*
  - *Lowercase letters (a-z)*
- Each string should be *unique*

Dies lieferte eine Liste mit 2000 Einträgen im *text/html*-Format, welche problemlos in eine .CSV Datei kopiert und in die **accesscodes_** Tabelle der Datenbank importiert werden konnten.

Die Access-Codes sehen in etwa wie gefolgt aus:

```
tCzqLE2m
C4x4tmUc
YExWmlND
izbPYwp1
HRbBeoOQ
HvTQkA5X
Rp9vdDv2
BNMLTSia
```

7.4.2. Technische Aspekte der Registrierung

Das Registrierformular beinhaltet **fünf Pflichtfelder** (dabei keine Optionalfelder):

- *Full Name* - vollständiger Name
- *Password* - Passwort
- *Repeat Password* - Passwort wiederholen
- *Email Address* - Emailadresse
- *Access Code* - Zugangscode

Außerdem beinhaltet es **zwei Kästchen**, die abgehakt werden müssen. Diese sind *Terms of Service* oder die Allgemeinen Geschäftsbedingungen, und *Data Protection Regulations* oder die Datenschutzerklärung (siehe Anhang). Vom Schüler wird erwartet, dass er diese durchliest und akzeptiert. Beide stehen im handlichen PDF-Format zum Download zur Verfügung.

Der Benutzer wird durch eine, mittels JavaScript implementierten Anzeige, bei der Wahl des Passworts unterstützt. Diese impliziert mit den Werten: *VERY WEAK*, *WEAK*, *GOOD* und *VERY GOOD* die Stärke des Passworts. Die Kriterien, nach denen das Skript prüft, sind:

- mindestens **8 Zeichen**
- Anwesenheit von **Kleinbuchstaben**
- Anwesenheit von **Großbuchstaben**
- Vorhandensein von **Zahlen**
- Vorhandensein von **Sonderzeichen**

Dabei sind *Sonderzeichen* nicht verpflichtend. Die Stärke des Passworts wird nochmals Server-seitig überprüft (siehe Abschnitt 8.1 *Hilfsfunktionen*), sollte das Passwort unzureichend sein, oder mit dem *Repeat Password* Feld nicht übereinstimmen, werden alle Anforderungen nochmals durch eine Meldung angezeigt und der Benutzer aufgefordert, die Eingabe nochmals zu tätigen.

Das PHP

Sobald das Formular abgeschickt wird, wird zunächst geprüft, ob alle Felder befüllt wurden. Dabei werden die Eingaben auch auf Authentizität gewertet (z.B., ob die angegebene Emailadresse einer tatsächlichen Email entspricht etc). Dazu werden diverse Validierungsfunktionen verwendet (siehe Abschnitt 8.1). Sollte eine Eingabe nicht in Ordnung sein, wird dem Benutzer eine Fehlermeldung angezeigt.

Des Weiteren wird geprüft, ob der angegebene *Access Code* sich tatsächlich in der Datenbank befindet und *frei* ist. Sollte dem der Fall sein, wird zunächst das Passwort mittels *BCRYPT*, einer kryptologischen Hashfunktion, die speziell für das Speichern von Passwörtern entwickelt wurde, gehashed[6]. So werden die Passwörter **nie** in Klartext in der Datenbank gespeichert und bleiben geheim. Anschließend wird noch identifiziert, ob die Email schon einmal benutzt

[6]Hashing ist ein Verfahren, bei dem eine große Eingabemenge auf eine fixe Ausgabemenge reduziert (komprimiert) wird.

wurde. Wenn nicht, werden alle Datensätze in die Datenbank (in die Tabelle *_usermodel*) gespeichert und eine *Session* wird durch die Setzung der *Superglobalen* **$_SESSION** initiiert. Dabei werden diesem Array der Status („student"), die Benutzer-ID, die Email, sowie der Benutzername übergeben. Diese werden auf WSS-Subseiten ausgelesen und teilweise bei SQL-Abfragen verwendet oder angezeigt. Bei erfolgreichem Abschluss wird der nun registrierte Benutzer auf die Indexseite (*index.php*) weitergeleitet.

Eine *Session* läuft für gewöhnlich nach 30 Minuten ab (Webservereinstellungsabhängig), nach Ablauf muss sich der Benutzer erneut einloggen.

7.5. Login

Versucht ein User auf eine der untergeordneten WSS-Seiten, die nicht *login.php* oder *register.php* ist, zuzugreifen, während er nicht angemeldet ist, so wird er automatisch auf die Login-Seite (*login.php*) umgeleitet. Dort kann er sich entweder anmelden oder auf die Registrierseite (*register.php*) wechseln, um sich mittels **Access-Code** zu registrieren.

Um festzustellen ob ein Besucher angemeldet ist, wird die *Superglobale* **$_SESSION** an der Stelle ['user'] überprüft (die Benutzer-ID). Ist das Array nicht gesetzt, so ist der Anwender nicht angemeldet.

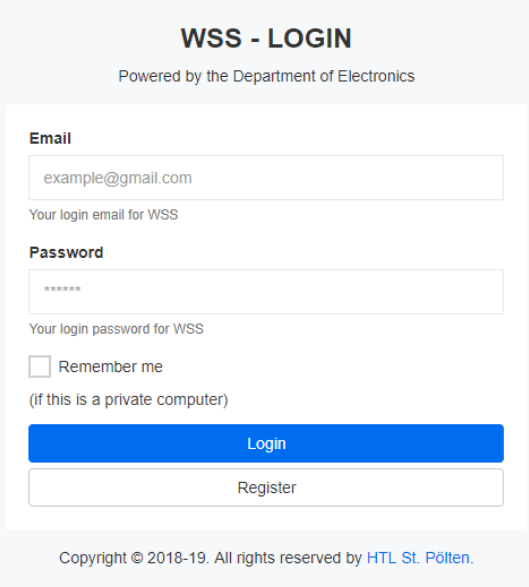


Abbildung 7.1.: Das Anmeldeformular (*login.php*)

Durch Anhaken des Kästchens *„Remember me”*, kann das Setzen von Cookies erlaubt werden - somit würde die Sitzung nicht nach 30 Minuten ablaufen, sondern würde erst beim Löschen der Cookies beendet werden. Mit der Verwendung von Cookies müsste das WSS die Benutzerinformationen nicht aus der *Superglobalen* **$_SESSION**, sondern aus lokal gespeicherten Daten lesen - diese genießen für gewöhnlich eine viel längere *Lebenszeit* (Lifetime).

Das setzen der Cookies wurde noch nicht implementiert. (Stand: 01.04.2019)

7.5.1. Technische Aspekte

Sobald der Benutzer alle Felder ausgefüllt, und auf die Schaltfläche „Login” gedrückt hat, werden die Daten *Email* und *Passwort* an das PHP weitergeleitet. Zuerst wird identifiziert, ob die Email in der Tabelle **_usermodel** vorhanden ist, ist dem der Fall, so existiert der Benutzer und das Passwort kann überprüft werden. Gibt es die Email allerdings nicht, so hat sich der Benutzer womöglich vertippt oder verwendet eine falsche Email; darüber wird er mittels einer Fehlermeldung vom Server informiert.

Ist die Email also in der Datenbank vorhanden, kann der dazugehörige Passwort-Hash zur Verifizierung des angegebenen Passworts verwendet werden. Dazu wird die PHP-Standardfunktion *password_verify* wie gefolgt eingesetzt:

```
1 if (password_verify($password, $usermodel['password'])) {
2     ...
3 }
4 else {
5     // falsches Passwort
6 }
```

Listing 7.2: Verifizierung des Passworts mittels *password_verify*.

$password ist das durch den Benutzer angegebene Passwort in Klartext.
$usermodel['password'] ist der Hash-Wert aus der Datenbank.

Dank der Tatsache, dass jede Email in der Datenbank nur einmal vorkommt, kann diese effektiv als Ersatztschlüssel für den Primärschlüssel verwendet werden. Mit Hilfe der Email werden also der Passwort-Hash, der Benutzername, der eigentliche Primärschlüssel, sowie die Zugangsberechtigung („status”) aus der Datenbank gelesen. Diese Daten werden dann in der **$_SESSION**-Variable gespeichert, um später von den WSS-Subseiten ausgelesen zu werden.

Die assoziativen Arraypositionen *'user'* (Primärschlüssel) und *'status'* sind dabei besonders wichtig. Anhand derer wird identifiziert, ob der Anwender eingeloggt ist und ob er auf die jeweilige Seite zugreifen darf.

7.6. Nutzerspezifische Funktionen

Aufgrund des mehrschichtigen Berechtigungssystems des WSS muss der Webauftritt für *drei* bis *fünf* Benutzergruppen angepasst werden. Dabei sind die Differenziertesten **Schüler**, **Lehrer** und **Administratoren**. Funktionsunterschiede bei Superadmins und PCB-Lehrern weichen so wenig von ihren Vorstufen ab, dass diese zunächst vernachlässigt werden können.

Dank der Verwendung von *Smarty* können dynamische Elemente sehr aufwandlos in die Seite integriert werden. Es soll immer der *Status* - also die Berechtigungsstufe des jeweiligen Benutzers - an das TPL File übergeben werden. Dort kann mittels *Smarty* überprüft werden, um welchen Benutzer es sich handelt und je nach *Status* werden dann die entsprechenden Inhalte angezeigt.

Bsp. all-professors.tpl

Auf dieser Subseite werden alle Professorprofile in Form von Kärtchen mit kurzen Zusammenfassungen angezeigt. Für Admins/Superadmins soll eine Schaltfläche zum Löschen dieser erscheinen.

```
1 {foreach from=$profRows item=profRow}
2     ...
3     {if $session_status == "admin" OR $session_status ==
4         "superadmin"}
5         <hr/>
6         <a id="_stylelink" onclick="deleteProof({$...}, {$...},
              '...')" style="color: #ff3841;">Delete User</a>
7     {/if}
8     ...
9 {/foreach}
```

Listing 7.3: Einsatz von Smarty zur Erstellung Benutzergruppen-spezifischer Elemente (Übergabeparameter zur besseren Lesbarkeit durch '...' ersetzt)

Zuvor muss in der dazugehörigen PHP Datei (*all-professors.php*) die Variable *$session_status* deklariert werden:

```
$smarty->assign('session_status', $_SESSION['status']);
```

Im Codebeispiel sieht man gleichfalls, wie *Smarty* verwendet wurde, um mittels einer *foreach-Schleife* alle 'Kärtchen' auszugeben. In Zeile **3** wird der beim *Login* determinierte *Benutzerstatus* überprüft. Entspricht dieser dem „Admin" oder dem „Superadmin ", so wird ein zusätzlicher HTML-Code zur Darstellung einer Schaltfläche zum Löschen des dazugehörigen Profils angezeigt.

Ergebnisse im Vergleich:

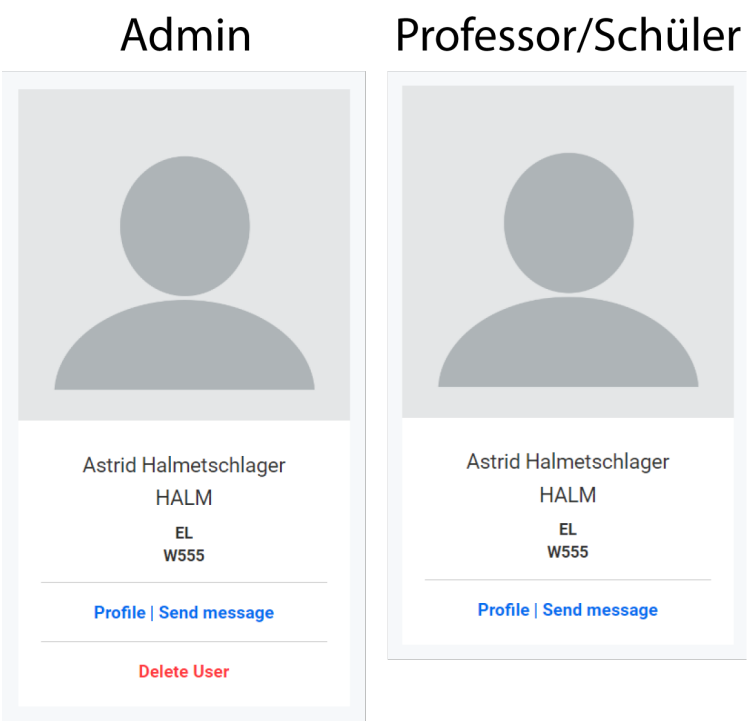


Abbildung 7.2.: *all-professors.php*: Vergleich der Ansicht zwischen Schüler/Professor und Admin

Es fällt auf, dass für den Administrator nun eine rote Schaltfläche verfügbar ist, die es dem jeweiligen Nutzer erlaubt, das dazugehörige Lehrerprofil zu löschen. Ist man allerdings als Lehrer oder Schüler eingeloggt, so wird die Schaltfläche nicht angezeigt.

7.6.1. Semiäquivalente Seiten

Es ist oft allerdings einfacher, für jede Berechtigungsstufe eine eigene Seite zu entwickeln - eine *semiäquivalente Seite*. Hierunter versteht man also eine zum Gesamtsystem untergeordnete Seite, die jenen Funktionen und Berechtigungen entspricht; wie diejenige, die der Benutzer versucht hat aufzurufen - mit dem Unterschied, dass diese neue Seite für die Zugriffsberechtigung des Anwenders vorgesehen ist.

Ebenso kann mit Seitenteilen, wie Navigationsleisten, Seitenleisten, Bannern etc. vorgegangen werden. So wurde bspw. für jede Nutzergruppe eine eigene Seitenleiste (TPL Datei) geschrieben. In der Praxis muss nur geprüft werden, um welche Art des Benutzers es sich handelt, und die dementsprechende Datei

geladen werden.

Bsp. Seitenleisten

```
{if $session_status == "student"}
    {include file='LeftMenu.tpl'}
{elseif $session_status == "professor" }
    {include file='LeftMenuProfessor.tpl'}
{elseif $session_status == "admin"}
    {include file='LeftMenuAdmin.tpl'}
{/if}
```

Listing 7.4: Implementierung dynamischer Elemente mittels Smarty

Hier wird ganz ähnlich, wie in Listing 7.3, an der *richtigen* Stelle im TPL File geprüft, um welche Berechtigungsstufe es sich handelt, je nachdem wird das korrekte Seitenmenü geladen.

Deviieren die Seiten allerdings zu viel von einander, wie im Falle der **Indexseiten**, so muss bereits in der PHP Datei geprüft werden, um welche Art des Users es sich handelt, dieser kann dann auf seine *semiäquivalente Seite* umgeleitet werden. Existiert eine solche Seite nicht (z.B. *add-professor.php* ist nur für Admins verfügbar), so kann der Zugriff entweder durch die HTML Fehlermeldung '403' verboten, oder der Nutzer auf eine andere beliebige Seite umgeleitet werden.

Im Falle von *index.php* sieht die Überprüfung wie gefolgt aus:

```
if ($_SESSION['status'] == "professor") {
    header("Location: professor.php");
}
elseif ($_SESSION['status'] == "admin") {
    header("Location: admin.php");
}
elseif ($_SESSION['status'] != "student") {
    header("Location: 403.html");
}
```

Listing 7.5: Implementierung semiäquivalenter Seiten mittels PHP

Dabei ist das Verfahren zu dem im Vorbeispiel ident. Die Überprüfung passiert jetzt allerdings nicht im TPL File, sondern bereits *ganz oben* in der PHP Datei.

7.7. Funktionen für Schüler

Ein Schüler hat alle Funktionen, die in den folgenden Unterabschnitten behandelt werden und jene, die für alle Benutzerebenen zur Verfügung stehen (siehe 7.10).

7.7.1. Einteilen in Räume

Schüler können sich halbtagesweise in einen Raum mit zugehörigem Lehrer einteilen. Diese werden im WSS als *Courses* bezeichnet und können sinngemäß als *Einteilungen*, jedoch nicht als *Kurse* übersetzt werden.

Auswählen der Räume/Lehrer/Fähigkeiten

Beim Erstellen bzw. Hinzufügen eines *Courses* kann der Schüler zuerst bestimmen, welche Auswahloptionen er angezeigt bekommen möchte - also entweder *Teacher* (Lehrer), *Room* (Räume) oder *Skill* (Fähigkeiten eines Lehrers). Sollte eines davon ausgewählt werden, wird eine *JavaScript*-Funktion mittels *onchange* ausgeführt. Diese ruft über *AJAX* ein PHP-File auf, das die weiteren Auswahlmöglichkeiten (abhängig von der vorherigen Auswahl) lädt und über *Smarty* in einem TPL-File darstellt.

```
1 <select ... onchange="showSelection(this.value)" ...>
2     ...
3 </select>
4 <div id="chosenValue" class="form-group"></div>
5 <div id="furtherInfos" class="form-group"></div>
```

Listing 7.6: templates/index.tpl: Aufuf einer *JS*-Funktion durch *onchange*

```
1  function showSelection(str) {
2
3      document.getElementById("furtherInfos").innerHTML = null;
4
5      selectionTR = str;
6
7      if(!str == "") {
8
9          if (window.XMLHttpRequest) {
10             // code for IE7+, Firefox, Chrome, Opera, Safari
11             xmlhttp = new XMLHttpRequest();
12         }
13         xmlhttp.onreadystatechange = function() {
14             if (this.readyState == 4 && this.status == 200) {
15                 document.getElementById("chosenValue").innerHTML =
                       this.responseText;
16             }
17         };
18
19         console.log(str);
20
21         xmlhttp.open("GET","php/getSelection.php?selection="+str+"
               &onChange=true",true);
22         xmlhttp.send();
23
24     }
25
26 }
```

Listing 7.7: js/draganddropcourse.js: Aufgerufene Funktion in 7.6

In 7.6 wird durch *onchange* die Funktion *showSelection* in 7.7 aufgerufen. Dabei wird der *Value* der derzeit ausgewählten *Option* durch *this.value* an die Funktion übergeben.

getSelection.php
Zu Beginn wird sichergestellt, dass alle alten *HTML*-Einträge im *DIV furtherInfos* entfernt werden. Dieses wird benötigt, wenn *Skill* ausgewählt wurde, da daraufhin ein Lehrer gewählt werden muss. Ist die Auswahl *Room* (Raum) oder *Teacher* (Lehrer) aktiv, fällt diese Notwendigkeit weg, da hier lediglich eine Möglichkeit (für jeden Lehrer gibt es nur einen Raum und vice versa) möglich ist.
Alle folgenden Befehle ab inkl. Zeile 7 dienen dem Aufruf und Anzeigen der Ausgabe von *getSelection.php*, dessen Inhalt - ein *Select* - mit dem Befehl in Zeile 18 in ein *DIV* geschrieben wird.

furtherInfos.php
In die Variable *selectionTR* wird der ausgewählte *Value* für den späteren Aufruf

der *JavaScript*-Funktion *furtherInfos* gespeichert. Diese ruft, wie *showSelection*, mit *AJAX* die Datei *furtherInfos.php* auf. Weiters werden je nach Auswahl entweder die Informationen über den zugehörigen Lehrer (wenn der Raum ausgewählt wurde) oder den zugehörigen Raum (wenn der Lehrer gewählt wurde) geladen. Sollte der einzutragende Kurs ein *Fixed Course* (Erklärung siehe weiter unten) sein, wird dies vermerkt.

Auswählen des Datums/der Zeit

Grundsätzlich wird der Zeitraum der Eintragung ausgewählt, indem man das *DIV* mit allen ausgewählten *Options* durch Drag-And-Drop[7] in ein durch das Datum/die Uhrzeit gekennzeichnetes *DIV* zieht. Sollte in diesen Feldern bereits ein Eintrag sein, kann dieser ersetzt werden. Da es sich um einen eigens Erstellten handelt, kann dieser jederzeit unabhängig von *Fixed Courses* verändert und gelöscht werden.
Sollte die Website auf einem Gerät aufgerufen worden sein, dass *HTML5*-natives *Drag-And-Drop* nicht unterstützt - bspw. ein Tablet oder Smartphone, wofür zusätzliche Funktionen notwendig wären - kann der Zeitraum durch ein *Select* gewählt werden. Diese Möglichkeit wird jedoch ausgeblendet, sobald das Datum durch Drag-And-Drop ausgewählt wurde.

Fixed Courses

Ein *Fixed Course* ist ein *Course*, der nicht durch Einteilungen von anderen Schülern ersetzt werden kann. Dadurch wird sichergestellt, dass der Schüler diesen Lehrer zu dieser Zeit besuchen und dass ein Schüler innerhalb eines Halbjahres bei jedem Lehrer x-mal sein kann. Ohne dieses Attribut könnten andere *Courses* verhindern, dass ein Schüler einen Raum buchen kann, da dieser zu jedem Zeitpunkt schon vollständig besetzt ist.
Deshalb kann ein *Fixed Course* bereits eingetragene Kurse ersetzen, die dieses Attribut nicht besitzen.
Primär sind *Fixed Courses* implementiert, da sonst Schüler, die ein rechtzeitiges Anmelden verpasst haben, womöglich diesen Raum in einem Halbjahr nicht erneut besuchen und dadurch ihr Projekt nicht fertigstellen können. Sollte ein Schüler keine *Fixed Courses* mehr offen haben (die Anzahl wird durch die Admins bestimmt) und trotzdem unbedingt ein weiteres Mal diesen Raum oder diesen Lehrer aufsuchen müssen, kann dies durch das Eintragen/Verschieben eines Eintrages durch einen Admin geschehen.

[7] Elemente können durch Drücken und Halten der linken Maustaste gezogen werden und durch Loslassen dieser einem anderen Bereich zugeordnet werden - Grundfunktion von *HTML5* in Kombination mit *JavaScript*.

7.7.2. Erstellen von Projekten

Die *Projekte* sind ein zentrales Feature des WSS. Sie bieten Schülern die Möglichkeit, an einer gemeinsamen Sache zu arbeiten. Dabei können Teams von bis zu *drei* Schülern gebildet werden, wobei immer ein Betreuungslehrer (der sog. Supervisor) aus einer Liste verfügbarer Lehrer gewählt werden muss. Das Erstellen des Projekts ist durch das Anklicken der Schaltfläche „My Project", in der linken Seitenleiste möglich. Sollte **kein aktives** Projekt vorhanden sein, wird einem die Möglichkeit geboten, eines zu erstellen.
Bei der Projekterstellung müssen die Pflichtfelder **Supervisor**, **Name**, **Beschreibung**, sowie **3 Meilensteine** - welche je Datum und Kurzbeschreibung beinhalten - befüllt werden.
Zusätzlich können optional zwei zusätzliche Schüler, aus einer Liste verfügbarer Kandidaten, gewählt werden. Die Liste stellt sich dynamisch auf jene Schüler ein, die ebenfalls kein aktives Projekt haben. Ebenso können *Project Skills*, also „Projekt Fähigkeiten" aus einer Liste von Fähigkeiten, sowie ein charakteristisches Projektbild gewählt werden.

Dank einer JavaScript Anwendung im Hintergrund ist das Auswählen zweier (oder mehr) gleicher *Project Skills* oder *Projektteilnehmer* nicht möglich. Selbst wenn JavaScript deaktiviert ist, werden jene Kriterien ebenfalls Serverseitig geprüft.

Nach der Erstellung des Projektes bekommt jeder ausgewählte Benutzer eine System-Nachricht, mit dem Link zum Projekt, sowie der Information, dass jene Person nun ein Teilnehmer des Projektes in der Position X ist. Auf den Teilnehmerprofilen wird automatisch eine Schaltfläche zum aktuellen Projekt eingeblendet und die Menüoption *My Projects* führt nun zum aktuellen Projekt.

7.7.3. Verwalten von Projekten

Ähnlich wie das gesamte WSS, ist das Projektsystem hierarchisch aufgebaut. Dabei führt der Projektersteller die Position **Team Leader**, also „Teamleiter". Die anderen beiden, *optional* gewählten Schüler, halten die Positionen **Second in Charge**, also „Zweiter in der Verantwortung" und **First Officer**, also „Erster Offizier". Dabei hat der letztere die geringsten Möglichkeiten zur Projektverwaltung, der Name dessen Position wurde absichtlich so gewählt, dass sich der Schüler nicht „weniger wichtig" vorkommt.

Der Betreuungslehrer heißt in der Projekthierarchie **Supervisor** und hat wenig Einfluss auf die Bearbeitung des Projektprofils, er dient mehr zur Wegweisung,

Beratung, Kontrolle und anschließend zur Bewertung des Projektes. Das Projektmanagement soll dabei möglichst zur Gänze von den Schülern übernommen werden. **Das Projektprofil** beinhaltet außerdem Links zu den persönlichen Profilen aller Teilnehmer, Informationen über ihre *Duties* - Verantwortungsbereiche, die *Projektskills*, Meilensteine und Status (*Work in Progress, Submitted* oder *Graded*)

Fähigkeiten des Supervisors

Der Supervisor ist der Projektbetreuer, also ein Lehrer und kann theoretisch unbegrenzt viele Projekte unterstützen. Möchte er allerdings ein Projekt nicht betreuen, so hat er die Möglichkeit, unter Angabe eines Austrittsgrundes, das „Projektteam" zu verlassen. Sonst beschränken sich seine Bearbeitungsmöglichkeiten lediglich auf das Verändern von Meilensteinen. Diese kann er unbegrenzt oft editieren.
Betreut ein Lehrer Projekte, so wird durch das Drücken auf die Schaltfläche *My Projects*, in der linken Seitenleiste, eine Liste laufender Arbeiten angezeigt. Wurde ein Projekt durch dessen *Team Leader* abgegeben, so erscheint in der Liste, in der Zeile des abgegebenen Projektes, ein **Download Button**. Durch das Betätigen dessen, werden die durch den Projektleiter hochgeladenen Dateien, heruntergeladen. Anschließend erscheinen auch neben den Namen der Teammitglieder Auswahlfelder, durch welche jedem eine persönliche Arbeitsbewertung (*Sehr Gut, Gut, Befriedigend, Genügend, Nicht Genügend*) vergeben werden kann. Es erscheint auch ein großes Auswahlfeld, durch welches eine Note für das gesamte Projekt vergeben werden kann. Noten können beliebig oft geändert werden.

Fähigkeiten des Team Leaders

Der Teamleiter, also auch der Projektersteller, führt die größte Projektmacht. Er kann beliebig oft den Namen, die Skills und die Projektbeschreibung ändern. Ebenso kann er ein Projektbild hochladen und den Teilnehmern ihre *Duties*, also Pflichten, einteilen. Außerdem kann er als einziger das Projekt, nach Hochladen eines zip-Archivs mit der Dokumentation, abgegeben. Nach der Abgabe wechselt der Status des Projekts von *Work in Progress* zu *Submitted*, und steht nun dem *Supervisor*, zum Benoten zur Verfügung. Einst abgegeben, gibt es keine Möglichkeit mehr, die Abgabe zurückzuziehen.
Befindet sich das Projekt im Status *Submitted*, so kann ein neues erstellt werden. Jeder Schüler kann also simultan nur **ein einziges** *aktives* Projekt haben. Dabei kann das Projekt erst **sieben** Tage nach dessen Erstellung abgegeben

werden. Dies wurde implementiert, um Spam[8] zu verhindern.

Der Teamleiter ist auch der Einzige von den Schülern, der die Meilensteine ändern kann, allerdings jedes nur einmal. Danach muss die Änderung der Meilensteine beim zuständigen Projektbetreuer angefragt werden. Dies kann entweder mittels einer WSS-internen Nachricht oder mündlich bewerkstelligt werden.

Es gibt derzeit (01.04.2019) noch keine Möglichkeit, die Position des Teamleiters abzugeben, oder Teammitglieder zu entfernen. Wobei das Entfernen der Teammitglieder dem Teamleiter nicht gerne zugetraut wird (Spam), weswegen diese Option auch *noch* nicht implementiert wurde. Projektteilnehmer können das Projekt allerdings *freiwillig* verlassen. Sollten die beiden Positionen *Second in Charge* und *First Officer* bei der Projekterstellung nicht befüllt werden, oder durch das Austreten von Mitgliedern leer sein, so können diese durch den Teamleiter freien Schülern zugeteilt werden.
Sollte der *Supervisor* das Projekt verlassen haben, so kann auch dieser durch den Teamleiter neu gewählt werden.

Fähigkeiten des Second in Charge

Der „Zweite in der Verantwortung" ist hierarchisch gesehen eine Stufe unter dem *Team Leader*. Seine Möglichkeiten über Projektmanagement sind jedoch recht beschränkt. Er kann bspw. keine Mitglieder hinzufügen, die Meilensteine auch kein einziges mal verändern, das Projekt abgeben oder dessen Namen ändern. Er kann allerdings die Projektbeschreibung, die *Skills* und das charakteristische Projektbild setzen. Ebenso hat er die Möglichkeit, seine eigenen *Duties*, sowie die des *First Officers* zu bearbeiten.
Wenn er möchte, kann er auch das Projekt verlassen und ein eigenes als *Team Leader* erstellen.

Fähigkeiten des First Officers

Der „Erste Offizier" ist das am geringsten gewertete Projektmitglied, er sollte mehr als *Mitwirkender* betrachtet werden. Seine Möglichkeiten bezüglich des Projektmanagements beschränken sich auf das Bearbeiten der eigenen *Duties*. Selbstverständlich steht es ihm frei, das Projekt zu verlassen und ein Eigenes als Team Leader zu gründen. Der Name „First Officer" wurde gewählt, damit sich das Mitglied nicht „weniger wichtig" fühlt.

[8] Unerwünschte Nachrichten, oder andere digitale Aktivitäten, die dem Empfänger gegen seinen Willen auferlegt werden.

7.8. Funktionen für Lehrer

Ein Lehrer hat alle Funktionen, die in den folgenden Unterabschnitten behandelt werden und jene, die für alle Benutzerebnen zur Verfügung stehen (siehe 7.10).

7.8.1. Ansehen und Verwalten von betreffenden Eintragungen

Ein Lehrer kann alle Eintragungen pro Zeitspanne sehen, die ihn und seinen Raum betreffen. Sollte er einen Schüler nicht an diesem Tag in seinem Raum haben wollen, kann er diesen durch das Klicken auf das Müllcontainer-Symbol aus dem *Course* entfernen. Dadurch bekommt der Schüler eine Benachrichtigung, dass sein Eintrag durch den Lehrer gelöscht wurde. Diese Funktion stellt jedoch nur einen Kick[9] dar, wodurch der Schüler sich erneut für diese Zeitspanne in diesem Raum eintragen kann. Dadurch soll verhindert werden, dass ein Lehrer, der einen Schüler nicht in seinem Raum haben will, diesen dauerhaft ausschließen kann. Vielmehr soll diese Funktion dazu dienen, Platz für Schüler zu schaffen, die den Lehrer oder die Geräte in diesem Raum dringender benötigen.

7.8.2. Meldung des Fernbleibens vom Unterricht

Wie auch im normalen Unterricht kann es passieren, dass Lehrer nicht rechtzeitig oder gar nicht erscheinen können, da sie bspw. krank geworden sind. Dadurch wären alle Schüler, die sich bei dieser Lehrperson eingetragen haben, ohne Betreuung und könnten nicht weiterarbeiten. Deshalb wurde eine Funktion eingebaut, mit der der Lehrer kurzfristig alle Schüler für eine Zeitspanne entfernen kann. Jedoch wird diesen nicht nur die Einteilung entfernt, wie beim Entfernen von einzelnen Schülern (vgl. 7.8.1), sondern bei Möglichkeit ein neuer *Course* bei einem Lehrer, der möglichst wenige Eintragungen hat und am Besten zum Schüler passt, erstellt. Diese Änderung wird dem Schüler per Nachricht mitgeteilt.

[9] Einen User von einem Server *kicken* bedeutet im Allgemeinen, dass die derzeitige Verbindung beendet wird, der Nutzer diese jedoch erneut aufbauen kann.

7.8.3. Verwalten von Leiterplatten-Bestellungen

Nur für Lehrer zugänglich, die die Berechtigung zur Verwaltung der Leiterplattenbestellungen besitzen.

Leiterplatten (im WSS nach der englischen Übersetzung *PCB* benannt) können durch jeden Benutzer für sich selbst oder in Stellvertretung bestellt werden (siehe 7.10.1).
Lehrer, die die Berechtigung zur Verwaltung der Leiterplattenbestellung haben (wird vom *Admin* vergeben), können gemeinsam mit allen *Admins* auf diese zugreifen.

Auf der Website werden alle Bestellungen nach Status sortiert (*New*, *In Progress*, *Finished*) gelistet. Durch einen Klick auf den *Info*-Button in der jeweiligen Zeile können alle notwendigen Eigenschaften (Auflistung dieser siehe 7.10.1) angezeigt werden, darunter der Besteller und ein Vermerk, ob die Bestellung in Vertretung erstellt wurde.

Mit einem Klick auf *Download Files* werden alle Dateien, die bei der Bestellung hochgeladen wurde, als *ZIP*- oder *RAR*-Datei heruntergeladen und so begutachtet werden. Sollte ein Detail nicht passen oder noch Fragen bestehen, kann mit einem Klick auf den *Message*-Button eine Nachricht an den Auftraggeber gesendet werden. Ebenso kann der Status der Bestellung jederzeit durch das Drücken des zugehörigen Buttons geändert werden.

7.8.4. Verwalten des eigenen Raumes

Jedem Lehrer wird beim Erstellen eines Lehrer-Accounts durch den Admin ein Raum zugewiesen. Dabei kann ein Lehrer nur einen Raum haben und vice versa. Diese Limitierung erlaubt die Bearbeitung eines Raumes durch den Lehrer, der die Raumnummer, die Anzahl an Plätzen und die Beschreibung für diesen ändern kann. Meist hat eine Lehrperson ihre Räumlichkeit dauerhaft, wodurch alle Eigenschaften dem Raum zugeteilt und nicht pro Lehrer gespeichert werden.

7.8.5. Hinzufügen und Verwalten von eigenen Theoriestunden

Grundsätzlich bieten die flexiblen *Courses* für Schüler beinahe nur Vorteile. Bei Lehrern kann jedoch das Problem auftreten, dass diese den vorgesehenen Lehrstoff nicht oder geringfügig vermitteln können. Deshalb besteht die Möglichkeit *Lectures* (dt. Vorlesungen) zu halten, in denen das nötige Wissen ausgewählten Schülern vorgetragen wird.

Hinzufügen einer Lecture

Um eine neue *Lecture* zu erstellen, kann ein Lehrer die Seite *add-lecture.php* aufrufen und dort einen Antrag mit dem Namen, der Beschreibung, dem Datum - hierbei kann eine Person nur eine Theoriestunde pro Zeitbereich anlegen -, den Schülern und optional den vermittelten *Skills* einreichen. Dieser Vorschlag muss vom Admin, der nach dem Einreichen eine Benachrichtigung erhält, überprüft werden und kann von diesem entweder bewilligt oder gelöscht werden. Andernfalls kann er auch dem Ersteller die zu bearbeitenden Punkte mitteilen.

Sollte der Admin die *Lecture* bestätigen, wird sie in den Stundenplan des Lehrers und der betroffenen Schüler eingetragen. Sollte ein Schüler zu diesem Zeitpunkt schon in einem anderen Theorieunterricht eingeteilt sein, so ist dieser von der neuen *Lecture* nicht betroffen. Alle verhinderten Schüler werden dem Lehrer in einer Nachricht bekannt gegeben.

Bearbeiten einer eingetragenen Lecture

Sollte der Lehrer mit einer bereits bewilligten Theoriestunde nicht zufrieden sein, kann er diese jederzeit löschen, wobei alle Schüler über diese Änderung benachrichtigt werden, da sie sich nun einen neuen *Course* erstellen müssen. Sollen gewisse Schüler nicht mehr an dieser *Lecture* teilnehmen, können diese vom Lehrer entfernt werden. Ein Hinzufügen von Schülern ist jedoch nicht möglich, da dies der Bestätigung des Admins unterliegt. Um eine Theoriestunde zu bearbeiten, kann er einen Admin bitten, die Bewilligung zurückzuziehen. In diesem Modus kann sie vom Lehrer wieder verändert werden.

7.9. Funktionen für Administratoren

Ein Admin hat alle Funktionen, die in den folgenden Unterabschnitten behandelt werden und kann zusätzlich *Leiterplatten-Bestellungen* und *Access-Codes* verwalten (siehe 7.10.1 und 7.4.1). Außerdem stehen ihm alle Funktionen, die jeder Benutzerebene bereitgestellt werden, bereit (siehe 7.10).

7.9.1. Anzeigen und Bearbeiten von allen Einteilungen

Als Admin können alle Einteilungen pro Raum pro Zeitspanne auf der Seite *admin.php* angezeigt werden. Sollten Änderungen vorgenommen werden müssen, so kann er einzelne Schüler per *Drag-And-Drop* in andere Räume verschieben und Eintragungen löschen. Auch in diesem Fall wird der Schüler über diese Aktion informiert.

7.9.2. Anzahl der Fixed Courses

Die Bedeutung von *Fixed Courses* wurde bereits in 7.7.1 besprochen. Damit ein Schüler pro Lehrer auch mehrere garantierte Eintragungen haben kann, ist die Anzahl dieser auf der Seite *admin.php* veränderbar. Dabei bezieht sich die angegebene Anzahl immer auf einen Lehrer.

Die Veränderung nach oben ist in jedem Fall möglich, jedoch kann die Anzahl nicht so weit verringert werden, dass bereits verwendete *Fixed Courses* dadurch gelöscht werden. Eine Funktion, die dies umgehen würde, ist zum Schutz der Schüler nicht implementiert.

7.9.3. Hinzufügen, Bearbeiten und Löschen von Räumen

Auf der Seite *add-lecture-hall.php* werden alle angelegten Räume angezeigt. Durch Inline-Editing können einzelne Werte, wie der Name, der zugeteilte Lehrer, die Anzahl an Plätzen und die Beschreibung geändert werden. Sollte ein erstellter Raum nicht benötigt werden bzw. physisch oder rechtlich nicht mehr vorhanden sein, kann dieser durch einen Klick auf den *Müllcontainer*-Button gelöscht werden.

Soll ein Raum gelöscht werden, so muss dieser Vorgang durch die Eingabe von *Yes* in einem *JavaScript-Alert* bestätigt werden. Damit werden alle *Courses* in diesem Zeitraum entfernt, und alle betroffenen Schüler darüber per Nachricht informiert.

Im unteren Abschnitt der Seite kann ein neuer Raum hinzugefügt werden. Die einstellbaren Werte gleichen denen beim Bearbeiten, wobei zusätzlich die Abteilung definiert werden kann.

7.9.4. Hinzufügen, Bearbeiten und Löschen von Fähigkeiten und Unterrichtsfächern

Fähigkeiten

Fähigkeiten werden im WSS as *Skills* bezeichnet (direkte Übersetzung ins Englische) und können auf der Seite *add-skill.php* angezeigt werden. Voreingestellt gibt es 21, die nach Belieben umbenannt werden können, sollte die Formulierung nicht passen bzw. ein Punkt davon redundant geworden sein. Weitere Skills können im unteren Bereich der Website hinzugefügt werden.

Skills können einerseits von Lehrern verwendet werden, um sich zu beschreiben - damit helfen sie Schülern, die spezielle Fähigkeiten einer Lehrperson benötigen, jedoch nicht genau wissen, welche diese besitzt -, andererseits können Schüler bei ihren Projekten angeben, welche *Skills* jenes beinhaltet.

Unterrichtsfächer

Diese werden im WSS as *Subjects* bezeichnet (direkte Übersetzung ins Englische) und können von Lehrern verwendet werden, wenn diese mehrere Fächer zusätzlich zum Werkstättenunterricht unterrichten. Damit kann ein Schüler annehmen, dass die Lehrperson den Lehrstoff und das damit verbundene Wissen kennt, und kann diesem auch Fragen bezüglich des aktuell vorgetragenen Inhalts eines anderen Faches stellen. Schüler können beim Erstellen von *Courses* nicht nach *Subjects* suchen, da diese nicht zur Werkstätte gehört.

Wie bei *Skills*, können alle Unterrichtsfächer vom Admin unter dem Link *add-subject.php* aufgerufen werden und *inline* bearbeitet werden. Ebenso können Weitere im unteren Teil der Website erstellt werden.

7.9.5. Hinzufügen, Bearbeiten und Löschen von Zeiträumen

Zeiträume spielen beim Einteilen von *Courses* eine zentrale Rolle und werden im WSS als *Dates* bezeichnet. Ähnlich wie Räume können sie durch Inline-Editing bearbeitet und ggf. gelöscht werden. Damit ein besserer Überblick entsteht, wie viele Schüler sich bereits bei einem *Date* eingetragen haben, wird die Anzahl der Eintragungen pro Zeitraum in einer Spalte angezeigt. Alle Einstellungen können unter *add-date.php* getätigt werden.

Das Löschen von *Dates* ist durch einen *JavaScript-Alert* zweifach gesichert. Wird der Vorgang bestätigt, so werden damit alle Eintragungen in diesem Raum gelöscht und betroffene Schüler darüber per Nachricht informiert.

Im unteren Teil der Website können neue *Dates* hinzugefügt werden. Diese müssen in der Zukunft liegen, dürfen jedoch nicht weiter als ein Jahr voraus sein.

7.9.6. Hinzufügen und Löschen von Lehrern

Hinzufügen von Lehrern

Über die Seite *add-professor.php* kann ein neues Konto für einen Lehrer erstellt werden, der noch nicht im WSS eingetragen ist. Dabei sind nur die Attribute *Full Name* (Vor- und Familienname), *Abbreviation* (Lehrer-Kürzel), *Email* und *Password* auszufüllen. Alle weiteren Felder, wie *Department* (Abteilung) und *Technical Information* können auf freiwilliger Basis eingegeben werden.

Im Reiter *Personal Informationen* (zu Beginn ausgewählt) kann eingestellt werden, ob dieser Lehrer Zugriff auf die Verwaltung der Leiterplattenbestellungen hat.

Löschen von Lehrern

Das Löschen von Lehrern geschieht auf der Seite *all-professors.php*, die von jedem Benutzer aufgerufen werden kann. Als Admin hat man deshalb zusätzlich die Option zum Löschen von Lehrern freigeschalten. Hierbei wird mit einem *JavaScript-Alert*-Fenster gefragt, ob man das Lehrerkonto tatsächlich entfernen möchte, da damit alle seine Daten verloren gehen. Beim Löschen eines Lehrers werden alle in Verbindung stehenden Kurse gelöscht und eingeteilte Schüler per Nachricht informiert.

7.9.7. Löschen von Schülern

Analog zum Entfernen von Lehrerkonten wird die Funktion zum Löschen auf der von allen Nutzern zugänglichen Seite *all-students.php* freigeschalten. Auch hier erfolgt eine doppelte Bestätigung durch einen *JavaScript-Alert*.

7.9.8. Bestätigen und Löschen von Theoriestunden

Lehrer haben die Möglichkeit zur Erstellung von Theoriestunden. Da diese unter Umständen alle Schüler betreffen können und die Anzahl von *Lectures* nicht begrenzt ist, muss ein Admin diese unter *all-lectures.php* bestätigen. Sollte er mit dem Entwurf einer Theoriestunde nicht zufrieden sein, kann er den Urheber per Nachricht dazu auffordern, diesen zu ändern. Andernfalls kann er ihn auch vollständig löschen.

Sobald ein Admin eine *Lecture* bestätigt hat, wird diese in den Zeitplan der betroffenen Schüler und den des Lehrers eingetragen. Sollten sich im weiteren Verlauf die eingetragenen Theoriestunden als ungünstig oder unmöglich erweisen, kann der Admin diese jederzeit wieder austragen, wobei er diese entweder löschen oder nur den Status *Not Granted* (dt. *nicht bewilligt*) zuweisen kann, wodurch die Option besteht, diese nochmals in den Stundenplan einzugliedern.

7.10. Funktionen für alle Benutzerebenen

Diese Funktionen können von jedem Benutzer ausgeführt werden und sind vor allem für die Bereitstellung von Informationen zuständig. Zusätzlich stehen für manche Benutzerebenen innerhalb dieser weitere Aktionen zur Verfügung, die jeweils einzeln angeführt werden.

7.10.1. Leiterplatten-Bestellung

Bisher werden Leiterplatten in der Werkstatt folgendermaßen bestellt:

1. Komprimierung aller notwendigen Dateien in ein *ZIP*-Archiv
2. Ausfüllen des Bestellformulars mit Angabe des Namens des *ZIP*-Verzeichnisses
3. Hochladen des *ZIP*-Verzeichnisses auf den File-Server der Schule
4. Abgeben des Formulars bei Fachlehrer Bauer

Nun muss der Besteller regelmäßig den Raum von Fachlehrer Bauer aufsuchen und nachfragen, ob der Print bereits gefertigt wurde. Bei Fehlern des Layouts wird der Auftraggeber jedoch schon bisher per E-Mail kontaktiert. Wird die Bestellung durch einen Lehrer abgegeben, können die Prints i. d. R. bei diesem wieder abgeholt werden.

Grundsätzlich liegt jedoch eine Ungewissheit vor, bis wann der Print fertiggestellt ist. Deshalb wurde im WSS eine Funktion zur Bestellung von Leiterplatten eingebaut.

Online-Bestellung mit Rückmeldung

Auf der Seite *pcborder.php* ist das bisherige Formular nachgebildet worden. Dadurch kann die Verarbeitung der Bestellungen mit der derzeitigen Methode weitergeführt werden. Der Besteller gibt den Namen (dieser ist bei Bestellung in Vertretung der Name der Person, für die der Print gefertigt werden soll), die E-Mail für Benachrichtigungen, die Klasse und den für das Projekt verantwortlichen Lehrer an. Weiters werden Informationen zum Print angegeben und das *ZIP*-Archiv mit den notwendigen Dateien hochgeladen. Sollte man eine Filmvorlage ausgedruckt haben, kann diese Option ausgewählt werden und der Upload des Ordners fällt weg. Daraufhin muss die Filmvorlage bei Fachlehrer Bauer zur Fertigung abgegeben werden.

Bei jeder Änderung des Status der Bestellung (*New* (dt. *noch nicht verarbeitet*), *In Progress* (dt. *wird verarbeitet*), *Finished* (dt. *fertig gestellt*)) wird der Besteller benachrichtigt. Bei Fehlern im Layout und anderen Komplikationen kann der Kunde per Nachricht verständigt werden.

7.10.2. Nachrichten

Nachrichten bilden einen der wichtigsten Bestandteile des WSS und werden *Messages* genannt. Über diese Funktion können alle Benutzer miteinander kommunizieren und werden über ihre *Courses*, *Lectures*, Leiterplattenbestellungen und generelle Änderungen im System informiert.

Im Widerspruch zur Integration dieser Funktion steht die generelle Einstellung bezüglich der Nutzung von neuen *Messengern*. Die meisten Personen nutzen lieber wenige Wege zur Kommunikation, die bereits von den meisten Bekannten verwendet werden, anstatt neue Dienste zu probieren, die mehr Vorteile bieten. "»Niemand würde einfach so zu uns wechseln«, wusste [Eric Ries]. »Die gefühlten Wechselkosten [zu seinem neuen Messenger] waren viel zu hoch. Seine Freunde herüber auf einen Messenger zu locken, macht einfach keiner.«" *Silicon Valley* [12].

Die Benachrichtigungen über Änderungen im System erfordern jedoch die Implementierung und so kam auch die Kommunikation zwischen Nutzern in Frage. Der größte Vorteil der systeminternen Mitteilungen ist die Spezialisierung auf das Thema Werkstatt und Projekte. Während an eine E-Mail-Adresse eines Lehrers neben werkstättenspezifischen Nachrichten auch Schulspezifische, gesammelt mit Privaten und Werbung geschickt werden, bleibt der Fokus im Nachrichtensystem des WSS auf das Wesentliche erhalten. Außerdem ersparen sich Nutzer das Öffnen eines Messengers oder des Mail-Programms, sollten sie bereits auf der Seite aktiv sein.

Schnellzugriff auf die Nachrichtenfunktion

Die neuesten Nachrichten oder eine Übersicht aller Benachrichtigungen können über das *Brief*-Symbol in der Menüleiste der Website aufgerufen werden. An diesem Punkt kann eine Mitteilung auch als *gelesen* markiert oder geöffnet werden.

Seiten, die Informationen über Lehrer, Räume, Schüler, *Lectures* und Leiterplattenbestellungen enthalten, haben in der jeweiligen Zeile des Eintrags ein *Chat*-Symbol, mit dem der Nutzer direkt kontaktiert werden kann.

Übersicht der Nachrichten

Auf der Seite *message.php* können alle Nachrichten, die empfangen, gesendet oder gelöscht wurden, eingesehen werden. Hierbei werden jeweils zehn Mitteilungen der ausgewählten Art (*Inbox* für Empfangene, *Sent* für Gesendete und *Trash* für Gelöschte) tabellarisch angezeigt. Durch das Drücken der Button für neuere Nachrichten (*Pfeil-nach-links*-Symbol) und ältere Nachrichten (*Pfeil-nach-rechts*-Symbol) werden weitere Mitteilungen geladen.

Ungelesene Benachrichtigungen werden durch einen hervorgehobenen Absender und Betreff gekennzeichnet. Dies gilt auch für den *Sent*-Bereich, wodurch überprüft werden kann, ob der Empfänger die Nachricht bereits gelesen hat.
Eine Nachricht kann als gelesen/ungelesen markiert oder gelöscht werden (wenn die Übersicht der gelöschten Nachrichten ausgewählt ist, wird sie wieder in den Ordner *Inbox* verschoben), indem die jeweilige Nachricht ausgewählt (die ID der Nachricht wird über *JavaScript* gespeichert) und dann der zutreffende Button (ein *Haken*-Symbol für *gelesen/ungelesen*, ein *Papierkorb*-Symbol für *gelöscht*) gedrückt wird.

Schreiben von Nachrichten

Über die Seite *newMessage.php* kann jeder Nutzer Mitteilungen an Admins, Lehrer und Schüler schicken. Dabei wählt er zuerst die Kategorie aus (*Single admin*, *Single teacher* oder *Single student*), durch die die jeweiligen Personen in einem weiteren *Select* angezeigt werden. Die Funktion ist gleich jener auf der Seite *index.php* aufgebaut (siehe 7.7.1). Nachdem ein *Subject* (dt. Betreff) eingetragen wurde, kann der Inhalt der Nachricht verfasst werden. Dieser muss mindestens aus 6 Zeichen bestehen und darf 512 Zeichen nicht überschreiten. Zeilenumbrüche werden unterstützt und in die Datenbank übertragen.

Gruppennachrichten

Damit mehrere Nutzer die gleiche Nachricht erhalten, gibt es die Möglichkeit bei der Kategorie Gruppen auszuwählen (*All students*, *All teachers*, *All admins*, *All teachers and students*). Die Gruppierung der eigenen Berechtigungsstufe und der Darüberliegenden kann nicht gewählt werden, da die Funktion primär auf die Benachrichtigung wichtiger Änderungen limitiert ist. Schüler können folgendermaßen keine Gruppennachrichten verfassen. Die einzige Ausnahme liegt bei Lehrern vor, da diese eine Nachricht an alle Admins erstellen können. Dies ist beispielsweise für die Änderung von *Courses* wichtig, da die Möglichkeit besteht, dass ein einzelner Admin längere Zeit nicht im WSS aktiv ist.

Antworten
Wie aus dem E-Mail-System oder der Referenz-Funktion in *Whatsapp* bekannt, bietet das WSS eine Funktion zum Antworten auf Nachrichten. Hierbei wird eine neue Benachrichtigung erstellt und in der Datenbank die referenzierte Mitteilung vermerkt. Zusätzlich wird standardmäßig der Inhalt des Betreffs auf *Re: [Betreff der referenzierten Nachricht]* gesetzt. Dieser kann wie bei E-Mails geändert werden.

Beim Anzeigen der Nachricht wird nach dem eigentlichen Inhalt jener der referenzierte Nachricht angezeigt - ähnlich wie bei den meisten E-Mail-Programmen. Durch einen Klick auf diesen wird der Nutzer automatisch auf die Nachricht weitergeleitet.

Limitierung geschriebener Nachrichten
Damit Spam unterbunden wird, können Schüler nur eine Nachricht pro Admin und fünf Nachrichten pro Lehrer in einer Stunde verfassen. Dies gilt auch für Antworten.

7.10.3. Anzeigen von Räumen

Eine Auflistung aller Räume kann durch das Betätigen der Schaltfläche *Lectures* und dann dessen Unterordnung *All Rooms* in der linken Seitenleiste, eingesehen werden. Die Auflistung beinhaltet den Namen des Raumes, das Kürzel der Abteilung, das dem Raum zugewiesen wurde, sowie dessen Kapazität, Kurzbeschreibung und verantwortlichen Lehrer. Wobei der Name des Lehrers gleichzeitig ein Link zu seinem persönlichen WSS-Nutzerprofil ist.

Ist man als Administrator eingeloggt, so erscheint neben jedem Eintrag ein Mülleimer-Symbol. Mit dessen Betätigung wird der Raum gelöscht.

7.10.4. Anzeigen aller Lehrer

Durch das Betätigen der Schaltfläche *Professors*, dann *All Professors* kommt der Benutzer zu einer Auflistung der Lehrerprofile in alphabetischer Folge. Die Profile werden in Form von „Kärtchen" dargestellt. Diese Kärtchen charakterisieren sich durch eine großzügige Anzeige des Profilbildes, unter dem sich der vollständige Name mit dem Geschlechtspräfix (z.B. Ms. oder Mr.) - nur wenn das Geschlecht durch den Nutzer angegeben wurde - befindet. Des Weiteren zeigen die Kärtchen den einzigartigen Lehrerkürzel (z.B.: JANE), die Abteilung, der der Lehrer angehört, dessen Raum (wenn vorhanden), sowie zwei Schaltflächen. Eine ist ein Link zum Profil, die andere führt zu einer Möglichkeit, eine sofortige Nachricht an den ausgewählten Lehrer zu senden.

Ist man als Administrator eingeloggt, so gibt es noch die zusätzliche Schaltfläche *Delete User*. Mit dessen Betätigung, kann das Zielprofil aus dem System entfernt werden. Siehe Abbildung 7.2.

Die Auflistung der Lehrer ist viel großzügiger als jede andere, da man davon ausgehen kann, dass die Anzahl der Professoren verhältnismäßig gering ist.

7.10.5. Anzeigen aller Schüler

Betätigt man zuerst die Schaltfläche *Students*, dann dessen Unterordnung *All Students*, so kommt man zu einer tabellarischen Auflistung aller Schüler. Die Liste ist standardmäßig alphabetisch sortiert, es lassen sich aber auch die Schüler eines bestimmten Kurses (*Course*) anzeigen.

Die Tabelle beinhaltet die Profilbilder aller Schüler, wobei diese viel kleiner angezeigt werden, als die der Lehrer. Vielmehr enthält sie die Namen der Schüler,

welche gleichzeitig als Link zum persönlichen Profil fungieren. Ebenso werden ihre Abteilungen und die Emails angezeigt. Durch den Klick auf die Email eines Schülers öffnet sich der Standard-Maildienst. Ebenso wird neben jedem Eintrag ein Sprechblasensymbol angezeigt, drückt man auf dieses, so kommt man zum WSS-internen Nachrichtendienst, wo eine schnelle Nachricht an den jeweiligen Schüler verfasst werden kann.

Ist man als Administrator eingeloggt, so erscheint neben dem Sprechblasensymbol ein zusätzliches Mülleimersymbol. Durch dessen Betätigung kann das jeweilige Profil *liquidiert* werden.

7.10.6. Profile von Nutzern

Nach erfolgreicher Registrierung hat jeder WSS-Teilnehmer die Möglichkeit, seinen Auftritt im System zu personalisieren. Dies ermöglichen vor allem die sogenannten Benutzerprofile oder schlicht *Profile*. Hierbei verfolgt das WSS den typischen *Social Media* Ansatz. Benutzer können nach Belieben Informationen hinzufügen, wie Geburtsdatum, Geschlecht, Kurzbiografie etc. Außerdem können Profilbilder (z.B. eigenes Portrait) gesetzt werden, diese helfen den Nutzern sich gegenseitig zu identifizieren und zugleich sich von der Masse abzuheben. Ebenso sorgen diese für einen persönlicheren Bezug zum System.

Lehrkräfte haben zusätzlich noch die Möglichkeit, aus vordefinierten Listen von *Skills* (Fähigkeiten), und *Subjects* (Unterrichtsfächern) bis zu jeweils *vier* auszuwählen, damit diese ebenfalls auf ihrem Benutzerprofil angezeigt werden können. Hat ein Lehrer *Skills* oder *Subjects* gesetzt, so verschafft dessen Profil den Schülern eine bessere Übersicht über seine technischen Fähigkeiten. Der Name sowie die Beschreibung des betreuten Raumes werden ebenfalls auf den Lehrerprofilen angezeigt.

Lehrer und Administratoren (inklusive Superadmins) besitzen als einzige die Fähigkeit, eigene Login-Daten, also Email und Passwort zu verändern. Diese Option wurde für Schüler *noch* nicht implementiert, da zusätzlich ein System integriert werden müsste, welches den Missbrauch jenes Features verhindert. Den Lehrern sowie Admins kann allerdings zugetraut werden, dass diese ausreichend vernünftig sind.

Führt der Schüler ein **Projekt**, so wird eine Schaltfläche angezeigt, die zum jeweiligen *Projektprofil* führt, wo alle Informationen zum Projekt eingeholt werden können. Hat ein Schüler sein Profil nicht aktualisiert, so wird eine entsprechende Meldung angezeigt. Diese soll ihn ermutigen, Informationen hinzuzufügen.

Auf Profilen von Administratoren wird zusätzlich verwiesen, dass jene Person der jeweiligen Usergruppe angehört. Dies soll dem jeweiligen Betrachter klarmachen, dass jener Nutzer besser nicht mit unseriösen Nachrichten adressiert werden soll. Dementsprechend befindet sich auf jedem Nutzerprofil eine Schaltfläche *„Send message"*, Durch Betätigen derer wechselt der jeweilige Betrachter in die Nachrichten-Ansicht des WSS mit bereits ausgefülltem Adressfeld.

8. Hilfsfunktionen

8.1. functions.php

Diese Datei beinhaltet die wichtigsten PHP-bezogenen *Hilfsfunktionen*, die zur Verbesserung der Erweiterbarkeit, Lesbarkeit und Konsistenz des Codes erstellt wurden. Viele der .php Dateien bauen auf diese auf.

8.1.1. Funktion sanitize

Die sicherlich am häufigsten verwendete Funktion ist *"sanitize"*. Sie wird überall dort verwendet, wo mit Benutzereingaben gearbeitet wird. Sie dient dazu, die vom Benutzer gesendete Information (z.B. Text), wie der Name vermuten lässt, aufzuräumen; u. a. zur Sicherstellung, dass keine böswilligen Absichten hinter der gesendeten Information stecken.

```
1 function sanitize($data) {
2     $data = trim($data);
3     $data = stripslashes($data);
4     $data = htmlspecialchars($data);
5     return $data;
6 }
```

Listing 8.1: php/functions.php: Funktionsdefinition für *sanitize*

Als einzigen Parameter erwartet die Funktion eine Zeichenkette (vorzugsweise eine, die sich durch Benutzerinteraktion ergeben hat). Dieser wird dann durch die Funktionen *trim*, *stripslashes* und *htmlspecialchars* gereicht. Ein sich durch diese Funktionen ergebender, "gesäuberter" Wert wird anschließend zurückgeliefert; mit diesem kann dann in weiterer Folge **sicher** gearbeitet werden.

trim - entfernt zunächst parasitäre Zeichen von beiden Enden einer Zeichenkette. Diese sind z.B. Leerzeichen, Tabulatoren, "Carriage Return" etc.

stripslashes - entfernt alle *Backslashes* von einer Zeichenkette. Dies ist vor allem nützlich, wenn mit HTML-Formularen gearbeitet wird.

htmlspecialchars - die Funktion konvertiert einige vordefinierte Zeichen in HTML-gerechte Zeichen. Z.B.: < wird zu < **&** wird zu & usw. Dies entschärft vor allem die Möglichkeit einer XSS-Atacke.

8.1.2. Funktion SafeDisplay

Eine unscheinbare Funktion, die rein zur Verbesserung der Benutzererfahrung dient. Technisch ist sie eher irrelevant. Sollte eine Benutzereingabe durch die Ausgabe einer Fehlermeldung gescheitert sein (z.B. auf add-professor.php), dann wird *SafeDisplay* verwendet um die *textalen* Eingaben wiederherzustellen. Natürlich setzt dies auch den Einsatz von *HTML* und *Smarty* voraus. Als einzigen Übergabeparameter benötigt die Funktion lediglich die per **POST** oder **GET** übergebenen Daten (z.B. Benutzername).

Die Eingaben-Wiederherstellungsfunktion mittels *SafeDisplay* wurde nicht auf allen Seiten implementiert, da diese rein kosmetische Funktionalität recht aufwendig ist und nur wertvolle Arbeitszeit beansprucht.

8.1.3. Funktion isNULL

Manchmal reicht es nicht aus, zu prüfen, ob ein Wert leer ist. Ebenso ist es häufig unzureichend, zu prüfen, ob ein Wert **NULL**[1] ist. Es kommt vor, dass ein Wert zwar eigentlich *leer* ist, die PHP-interne Funktion *empty* diesen jedoch nicht als leer sieht, weil er mit **NULL** befüllt ist und v. v.
Aus diesem simplen Grund wurde die häufig verwendete Funktion *isNULL* entwickelt. Sollte der Wert *NULL* oder *leer* sein wird *true* retourniert, im umgekehrten Fall *false*.

Es ist oft wichtig festzustellen, ob der Wert *leer*, also *NULL*, oder wirklich *leer* ist, denn die MySQL-Datenbank akzeptiert nur befüllte Werte. Selbst wenn der Wert *leer sein* ***muss***, soll er mit *NULL* gefüllt werden, bevor er in die Datenbank gespeichert wird. Wird dies nicht vollzogen, kommt es zu einem Einfüge-Fehler („Insertion Errror at Querry Execution“). *isNULL* kommt vor allem bei optionalen Werten, wie den Eingabefeldern auf *add-professor.php* zum Einsatz.

[1] **NULL** ist in der Informatik ein Wert, der die Abwesenheit eines Wertes darstellt; es ist paradox. Bereits 1979 hat E. F. Codd in seinem Buch über rationale Datenbankmodelle *Extending the Database Relational Model to Capture More Meaning* zwischen zwei Arten von **NULL** unterschieden: die Abwesenheit eines Wertes, weil keiner existiert „property inapplicable“, oder die Abwesenheit, da man den Wert (noch) nicht kennt „value at present unknown“[9].

8.1.4. Funktionen zur Daten-Validierung

Da das WSS-System sehr viel mit Benutzereingaben arbeitet, wurden spezielle Funktionen zur Überprüfung dieser erstellt. Zwar sind diese nicht unbedingt notwendig, jedoch ist es aus administrativer Sicht sinnvoll, die Eingaben nach bestimmten Kriterien zu prüfen, bevor diese in die Datenbank gespeichert werden. Solche Kriterien sind zum Beispiel: Eingabelänge, Zeichensatz, Passwortauthentizität etc.

In Fällen wo Dateien hochgeladen werden können, und dann wieder zum Download angeboten werden (z.B. Projektabgabe), müssen diese aus Sicherheitsgründen überprüft werden. Ein Angreifer könnte versuchen, Scripte hochzuladen, die beim Download automatisch auf dem Server ausgeführt werden. Solche Scripte könnten eine *Reverse Shell* beinhalten, mit der der Angreifer auf den Server zugreifen kann.

Funktion password_ok

Die Funktion ist zur Überprüfung von Passwörtern gedacht, sie wird u. a. auf *register.php* und *add-professor.php* eingesetzt. Wenn jemand versucht sich zu registrieren, oder das bestehende Passwort zu ändern, muss festgestellt werden, ob das neue Passwort den folgenden Kriterien entspricht:

i. Passwort beinhaltet mind. einen Großbuchstaben

ii. Passwort beinhaltet mind. einen Kleinbuchstaben

iii. Passwort beinhaltet mind. eine Zahl

iv. Passwort ist mindestens *acht* Zeichen lang

Diese Kriterien wurden zur persönlichen Sicherheit von Benutzer-Konten festgelegt, schließen aber die Verwendung von unsicheren Passwörtern wie "1234Andi" nicht aus.

Intern verwendet die Funktion eine *Regular Expression*[2], konkret:

```
1 '/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[a-zA-Z\d]{8,}$/'
```

Listing 8.2: php/functions.php: Regular Expression zur Überprüfung von Passwörtern

[2]Ein **Regulärer Ausdruck** (engl. Regular Expression), oder *RegEx* genannt, ist eine Folge von Zeichen, die ein Suchmuster mittels spezieller Syntax definieren. Normalerweise werden *Regex*(es) verwendet, um in Zeichenketten (Strings) Übereinstimmungen zu finden, oder Eingaben zu überprüfen.

Funktion email_validator

Da Email-Adressen sehr wichtig sind, um die Benutzer des Systems verlässlich kontaktieren zu können, muss sichergestellt werden, dass die angegebenen Adressen stets gültig sind. Die sicherste Methode, dies zu gewährleisten, ist das Versenden einer Mail mit einem Verifizierungslink an die angegebene Email. Aus Gründen der Benutzerfreundlichkeit wurde auf diese zusätzliche Verifizierungsfunktion jedoch verzichtet, da bereits sog. *Accesscodes* (Zugangscodes) verwendet werden, um sich auf der Plattform überhaupt registrieren zu können. Deshalb ist eine *genaue* Überprüfung der Email-Adressen aus Entwicklersicht redundant; stattdessen wurde eine rudimentäre Validierung mittels *email_validator* implementiert.

Intern verwendet die Funktion sog. Filter, die von PHP, bereits ab Version *5.2*, zur Verfügung stehen. Es ist nicht bekannt, ob die Filter mit Einsatz von *Regular Expressions* umgesetzt oder ob sie anders implementiert wurden. Jedenfalls wurde absichtlich darauf verzichtet, eine *RegEx* zu verwenden, da diese korrekterweise nach dem RFC 5322-Standard[3] zu implementieren ist. Ein solcher *Regulärer Ausdruck* ist aber sehr lang und aus Sicht der Entwickler völlig ineffizient.

Beim Funktionsaufruf kann zusätzlich angegeben werden, welche maximale Länge die Email haben darf, und ob das Eingabefeld leer sein darf. Beim Verifizierungsfehlschlag wird ein Feld von Elementen (Array) mit der Fehlermeldung und zusätzlichen Parametern zurückgeliefert, sonst *true*.

Funktion uepp_validator

Der Name bezieht sich auf die Übergabeparameter der Funktion. Diese sind der Benutzername (Username), die Email, das Passwort und ein Wert, welcher *dem Passwort entspricht*, anzugeben. **uepp_validator** baut auf den Funktionen **password_ok** und **email_validator** auf. Mit ihrer Hilfe wird festgestellt, ob die Felder leer sind (unzulässig), sowie ob die Email gültig ist und ob das Passwort den Kriterien, die in 8.1.4 *Funktion password_ok* definiert wurden, entspricht. Zusätzlich wird festgestellt, ob die beiden Passwörter übereinstimmen; dies kommt vor allem bei dem Registrierungsformular (register.php) zum Einsatz; u. a. wird die Funktion auch bei "profile-professor.php" und "add-professor.php" benötigt.

[3] **RFC** (*Request for Comments*) ist eine Reihe technischer Dokumente, die diverse Funktionalitäten betreffend des Internets deklarieren. **RFC 5322** ist ein Standard, der die Implementierung von Email-Adressen definiert. RFC 5322 *Internet Message Format* wurde von Paul Resnick entworfen.

Die Funktion retourniert bei Fehler ein Feld mit Elementen, welches u. a. die Fehlerinformationen beinhaltet, welche Art von Fehler aufgetreten ist usw. Die Fehlermeldung soll dann dem Benutzer angezeigt werden.

Funktion select_validator

Durch das Ankreuzen von Kontrollkästchen (Checkboxes) oder Optionsschaltflächen (Radio Buttons) in diversen HTML-Formularen werden Zahlenwerte (IDs) gesendet, welche dann in die Datenbank gespeichert werden sollen. Da allerdings HTML-Formulare sehr leicht zu manipulieren sind, kann es passieren, dass ein Benutzer, entweder mit Absicht oder aus Versehen, falsche Werte abschickt. Die Funktion *select_ validator* soll dies verhindern.
Die Funktion bekommt *vier* Werte übergeben:

- der zu überprüfende Wert
- ein Feld (Array) mit gültigen Werten
- den "echten" Namen der Optionsschaltfläche (für eine bessere Fehlerausgabe)
- eine Variable, die angibt, ob der zu überprüfende Wert leer sein darf, oder nicht (*true* oder *false*)

Der empfangene Wert wird mit der Liste der gültigen Werte verglichen, bei erfolgreichem Treffer wird *true* retourniert. Bei Fehlschlag wird ein Feld zurückgegeben, welches die Fehlermeldung und weitere Parametern beinhaltet. Diese soll dann dem Benutzer angezeigt werden.

Funktion stdin_validator

Der Name "stdin_validator" bezieht sich auf **Standard Input**, was im Programmierumfeld auf eine standardmäßige Eingabemethode deutet. Die Funktion ist eine sehr häufig verwendete, da sie überall dort eingesetzt wird, wo mit textalen Eingabefeldern in HTML-Formularen gearbeitet wird.

Die Funktion ist sehr vielseitig und benötigt daher *sechs* Parameter bei ihrem Aufruf:

- der zu überprüfende Wert
- der zulässige Typ des zu überprüfenden Wertes
  - DIGIT (nur Zahlen)
  - WORDS (nur Wörter)
  - CHAR (nur Zeichen des deutschen Alphabets)
  - TEXT (jeglicher Text mit Zeilenumbruch)
  - NONE (keine Überprüfung)
- minimale Zeichenzahl
- maximale Zeichenzahl
- den "echten" Namen des Feldes (für eine detaillierte Fehlerausgabe)
- eine Variable, die angibt, ob der zu überprüfende Wert leer sein darf (*true* oder *false*)

Die Funktionalität ist ähnlich wie bei der Funktion *select_ validator* (Abschnitt 8.1.4). Der zu überprüfende Wert wird nach den Kriterien des erlaubten Typs und der Zeichenzahl bewertet. Bei Fehlschlag wird ein Feld mit der Fehlermeldung und diversen weiteren Parametern zurückgeliefert. Bei Erfolg wird *true* zurückgegeben.

Funktion date_validator

Da Benutzerprofile die Möglichkeit bieten, ein Geburtsdatum anzugeben, und weil ein Datum unter vielen andern Umständen angegeben werden muss, soll geprüft werden, ob dieses ein wirkliches Datum ist und *authentisch wirkt*. Authentisch ist im Falle des Geburtsdatums ein Mindestalter des Benutzers; es kann bspw. angenommen werden, dass jeder Schüler, der den Werkstättenunterricht besucht, mindestens 14 Jahre alt ist. Im Falle des Hinzufügens von Unterrichtseinheiten, sollen diese nur in der Zukunft liegen dürfen.
Die Funktion benötigt zwei Übergabeparameter: das zu überprüfende Datum **$date** und einen *zulässigen* Abstand zum aktuellen Datum **$age** (positiv, wenn es nur in der Vergangenheit liegen darf und negativ, wenn das Datum nur in der Zukunft sein darf). Der **Funktions-Algorithmus** sieht wie gefolgt aus:

```
1  function date_validator($date, $age) {
2      ...
3      if (empty($date)) {
4          ...
5      }
6      elseif (false === strtotime($date)) {
7          ...
8      }
9      elseif (time() < strtotime($age, strtotime($date))) {
10         ...
11     }
12     else {
13         list($year, $month, $day) = explode('-', $date);
14         if(!checkdate($month, $day, $year)) {
15             ...
16         }
17     }
18     return $errRep;
19 }
```

Listing 8.3: php/functions.php: Algorithmus der Funktion *date_ validator*

Zuerst wird festgestellt, ob der zu prüfende Wert **$date** leer ist, wenn dem nicht der Fall ist, wird mit Hilfe der Funktion *strtotime* (Bestandteil von PHP ab der Version 4.0) versucht, den gegebenen Wert in einen UNIX-Zeitstempel umzuwandeln. Sollte die Umwandlung fehlschlagen, wird eine Fehlermeldung retourniert - kein Datum, oder Datum im falschen Format.

Ist die Umwandlung jedoch erfolgreich, wird geprüft, ob **$date** innerhalb des mit **$age** spezifizierten Abstandes liegt. Ist dem der Fall, wird nochmal mittels der ab PHP Version 4.0 bereitgestellten Funktion *checkdate* geprüft, ob die einzelnen Bestandteile des Wertes **$date** tatsächlich dem Format des Gregorianischen Kalenders entsprechen. Bei Fehlschlag wird wie üblich ein *Array* mit der Fehlermeldung, sowie konkreten Details über den Fehler retourniert. Bei Erfolg liefert die Funktion *true* zurück.

Funktion date_validator_onempty

Der **date_validator_onempty** wurde als eine verbesserte Version der Funktion **date_validator** entwickelt. Im Gegensatz zum Vorgänger bietet sie die Möglichkeit, einen gültigen Zeitbereich zu spezifizieren; der Vorläufer konnte lediglich prüfen, ob das Datum in der Zukunft oder der Vergangenheit liegt. Außerdem kann nun angegeben werden, ob der zu überprüfende Wert leer sein darf. Dies ist vor allem für optionale Felder notwendig, die aber dennoch zusammen mit anderen abgeschickt werden; z.B.: *profile.php, profile-professor.php oder profile-admin.php.*

Deklaration

```
1 function date_validator_onempty($date, $margin_top, $margin_bottom
      , $fieldEmptyYes) {
2     ...
3 }
```

Listing 8.4: php/functions.php: Deklaration von *date_ validator_ onempty*

Verwendung

```
1 // Zeitraum von zwei Jahren (Vergangenheit und Zukunft)
2 date_validator_onempty($date, '-1 years', '+1 years', false);
3 // Nur ein Jahr in die Zukunft
4 date_validator_onempty($date, '-1 years', '-0 years', false);
5 // Nur ein Jahr in die Vergangenheit
6 date_validator_onempty($date, '-0 years', '+1 years', false);
```

Listing 8.5: php/functions.php: Funktionsaufruf von *date_ validator_ onempty*

Intern gibt es keinen Unterschied beim Prüfungsverfahren des Datums, es wurden lediglich die beiden neuen *Eigenschaften* Zeitrahmen und Möglichkeit der Akzeptanz eines leeren Wertes hinzugefügt. Die ursprüngliche Funktion **date_validator** wurde aus Kompatibilitätsgründen nicht entfernt. Es wird jedoch empfohlen den Nachfolger zu verwenden.

8.1.5. Funktionen zur Daten-Validierung mit Aktion

Der Unterschied zu Unterabschnitt 8.1.4 (Funktionen zur Daten-Validierung) ist, dass die Daten nicht nur auf Authentizität und Gültigkeit geprüft, sondern gleich auf den Server hochgeladen werden. Zwar werden die Daten vorerst nicht in die Datenbank gespeichert, dennoch ist es im Falle von bspw. Bild-Uploads sinnvoll diese nach erfolgreicher Validierung gleich und ohne Zwischenschritte, hochzuladen. Dies spart marginal Rechenleistung und ist aus der Sicht der Entwickler am *effektivsten*. Die Informationen über den Pfad der hochgeladenen Datei werden erst im nächsten Schritt in die Datenbank eingetragen.

Funktion image_validator

Da das WSS ihren Benutzern die Möglichkeit bietet, eigenen Profile zu verwalten und u. a. aus Gründen der Personalisierung ein Profilbild zu setzen, muss eine Möglichkeit über Bild-Uploads geboten werden.
Da man allerdings nicht immer davon ausgehen kann, dass der Anwender ein gültiges Dateiformat verwendet, oder die Datei überhaupt ein Bild ist, müssen diese Kriterien geprüft werden. Dazu wurde „image_validator" entwickelt: Die Funktion prüft diverse Parameter der Datei, stellt fest, ob es sich tatsächlich um ein Bild handelt und speichert das Material anschließend auf dem Server. Sie wird überall dort eingesetzt, wo Bilder hochgeladen werden können z.B.:

profile.php, profile-professor.php, project-profile.php, add-project.php etc.

Um Dateiuploads überhaupt vornehmen zu können, muss dem HTML-Formular das Attribut **'enctype="multipart/form-data"'** beigefügt werden. Der Parameter **„enctype"** gibt an, wie die abgesendeten Daten codiert werden sollen, dabei ist
„application/x-www-form-urlencoded" der Standard-Wert und muss nicht explizit spezifiziert werden.

Deklaration

```
1 function image_validator($post, $dir, $fileEmptyYes){
2     ...
3 }
```

Listing 8.6: php/functions.php: Deklaration von *image_ validator*

Dabei ist im Parameter **$post** der abgesendete Formularwert, welcher die Bilddatei enthält, anzugeben. **$dir** (String) steht für *Directory* und spezifiziert dementsprechend das Verzeichnis auf dem Server, in welchem die Datei abgelegt werden soll.
$fileEmptyYes erwartet *true* oder *false*, je nachdem, ob der Wert der übergebenen Variable **$post** leer sein darf oder nicht (bspw. für optionale Felder).

image_validator verifiziert das hochzuladende Bild in drei Schritten. Zuerst ist es wichtig zu wissen, dass jede Datei, die hochgeladen werden soll, in der *superglobalen* Variable **$_FILES** gespeichert wird. Dabei ist $_FILES ein multidimensionales Array und wirft in simpelsten Fehlerfällen an der Stelle $_FILES[$post]['error'] folgende Fehlerkonstanten:

- UPLOAD_ERR_OK - kein Fehler erkannt (alles o.k.)
- UPLOAD_ERR_NO_FILE - keine Datei ausgewählt
- UPLOAD_ERR_INI_SIZE - maximale Dateigröße überschritten (festgelegt in der entsprechenden Direktive in *php.ini* auf dem Webserver)
- UPLOAD_ERR_FORM_SIZE - maximale im HTML Formular angegebene Dateigröße überschritten
- UPLOAD_ERR_PARTIAL - Datei wurde nur teilweise hochgeladen

Im ersten Schritt wird also geprüft, ob eine der Konstanten gesetzt wurde, im Falle von UPLOAD_ERR_OK wird die Überprüfung übersprungen. Im Fehlerfall wird eine **Exception** geworfen, die dann von der Funktion als Rückgabewert retourniert wird.

Dabei kann den Angaben von $_FILES nur bedingt vertraut werden, weil File-Uploads häufig als Angriffsflächen benutzt werden und Inhalte von $_FILES potentiell manipuliert werden können. Somit sollte die Überprüfung der Konstanten eine weitere Validierung nach sich ziehen.

Im nächsten Schritt folgt nochmal die Überprüfung der Dateigröße (diese sollte unter 2MB liegen); dazu wird wieder $_FILES an der Stelle [$post]['size'] exploriert. Allerdings kann dieser Angabe, wie bereits erwähnt, nicht wirklich vertraut werden. Dann wird noch mittels der Funktion **is_array** geprüft, ob sich mehrere Dateien in $_FILES befinden. Sollte dies der Fall sein, wird eine Fehlermeldung retourniert.

Die bisher wichtigste Überprüfung wird durch die Erstellung des Objekts

```
$finfo = new finfo(FILEINFO_MIME_TYPE);
```

ermöglicht. Dieses wird verwendet, um den tatsächlichen Inhalt der hochzuladenden Datei über den MIME-Typ[4] festzustellen. Dabei wird geprüft, ob dieses einem der drei zugelassenen Bildformate entspricht:

```
'image/jpeg'
'image/gif'
'image/png'
```

Dabei kann der MIME-Typ nicht so leicht wie die Dateiendung manipuliert werden; wird also bspw. eine .php Datei in .jpg umbenannt und hochgeladen, so wird jene Datei immer noch als .php identifiziert.
Solche Überprüfungen sind besonders wichtig, da ein Angreifer bspw. versuchen könnte, ein PHP Skript hochzuladen, mit dem Ziel Dateien auf dem Server zu manipulieren oder eine *Reverse-Shell*[5] auszuführen.

Ist die Prüfung des MIME-Typs erfolgreich, so wird nun eine **zuverlässige** Prüfung der Dateigröße, sowie der Bildweite und -Höhe (in quadratischen Pixel), mittels der Funktion **getimagesize** vollzogen. Dabei darf das hochzuladende Bild nicht kleiner als 120px x 120px und nicht größer als 1080px x 1080px sein. Dies wurde durch die Entwickler festgelegt, damit keine unnötig großen Bilder hochgeladen werden können.

[4] Internet Media Type (MIME) - wird in der Kommunikation zw. einem Client und einem Server gesendet, um Auskunft über die Art der übertragenen Datei anzugeben. Dieser wird bestimmt, indem die Datei-Header und teilweise Inhalte analysiert werden.

[5] Eine (unsichere) Remote-Shell (telnet), die vom Ziel eingeleitet wird. Das ist das Gegenteil einer „normalen" Remote-Shell, die von der Quelle initiiert wird.

Im letzten Schritt gilt die Validierung der Bilddatei als abgeschlossen. Mittels der MIT-lizenzierten Bibliothek *„ImageResize"*, dementsprechend den Methoden des Objektes *ImageResize*, wird das Bild nun auf eine definierte Größe von 600px x 600px zugeschnitten. Dies ist für einen einheitlichen Auftritt der Profile wichtig. Dabei verwendet die Methode die Mitte des Bildes als Ausgangsvektor.

Um auszuschließen, dass Bilder mit gleichen Dateinamen kollidieren, wird an den Namen der Datei eine zufällige Zeichenkette angehängt. Erst danach wird die Datei im angegebenen Pfad **$dir** auf dem Server, mittels der PHP Standard-Funktion **move_uploaded_file**, gespeichert.
Der image_validator retourniert ein Array, welches u. a. den Dateinamen (Pfad als Teil des Dateinamen), die Fehlermeldung, den Fehlertyp sowie den Parameter $formvalid (*true* oder *false*) beinhaltet. Anhand des Wertes von $formvalid soll erkannt werden, ob die Subroutine erfolgreich (fehlerfrei) ausgeführt wurde. Der Dateiname kann des Weiteren für den jeweiligen Benutzer in die Datenbank gespeichert werden; so ist das Darstellen der Datei zu einem späteren Zeitpunkt möglich.

Funktion zip_validator

In manchen Fällen müssen keine Bilder, sondern andere Dateien hochgeladen werden, u. a. zip-Archive (bspw. bei der Abgabe von Projekten oder Dateien für die Leiterplattenfertigung). Dazu wurde eine separate Funktion **zip_validator** entwickelt.

Die Funktionalität unterscheidet sich vom *image_validator* nur marginal. Es wird ebenso die superglobale Variable **$_FILES** nach Fehler-Konstanten untersucht, die Dateigröße wird ermittelt (max. 6MB), und eine MIME-Typ Überprüfung wird vollzogen. Am Ende wird der Datei eine zufällige Zeichenkette angehängt, und das Archiv anschließend auf dem Server gespeichert. Dabei sind die Übergabeparameter, sowie die Rückgabewerte der Funktion dieselben wie bei *image_validator*. Der einzige Unterschied besteht darin, dass auf eine Untersuchung der Bild-Auflösung verzichtet wurde und bei der MIME-Typ Überprüfung andere MIMEs verwendet wurden. **Diese sind:**

```
'application/zip'
'application/octet-stream'
'application/x-zip-compressed'
'multipart/x-zip'
'application/x-rar-compressed'
```

Aus den Informationen wird erkenntlich, dass die beiden Archivtypen *.zip* und *.rar* zugelassen sind. Dabei kommen zip-Archive in scheinbar mehreren Varianten, weswegen auch mehrere MIMEs zugelassen werden.

Generell könnten die beiden Funktionen *image_validator* und *zip_validator* unter einer Funktion zusammengefasst werden, bei der die Art der hochzuladenden Datei als Parameter spezifiziert werden kann. Dies wurde allerdings unterlassen, weil Uploads anderer Art als Bilder oder Archive, nicht stattfinden werden, und jene Aufgabe nur zusätzlichen Programmieraufwand bereitet.

8.1.6. Funktionen zur Daten-Manipulation

„Daten-Manipulation" bezieht sich in dem Fall auf das Verändern von Datenbankeinträgen. Manchmal wird der Benutzer aufgefordert, Daten auf seinem Profil oder Projekt zu aktualisieren. Damit der Nutzer allerdings nicht alle Eingabefelder neu ausfüllen muss, werden sie mit den *alten* Informationen befüllt und **alle** abgeschickt, dabei muss überprüft werden, welche dieser Felder *alte* Informationen beinhalten (vor dem Funktionsaufruf), welche davon leer sind (evtl. in der Funktion zu prüfen), und welche davon tatsächlich mit *neuen* Informationen befüllt wurden.
Die Ergebnisse jener Felder können dementsprechend nach der Validierung (siehe Unterabschnitt 8.1.4 *Funktionen zur Daten-Validierung*) in der Datenbank aktualisiert werden. Dabei ist es geschickt, eine Subroutine zu verwenden, da sich jene Daten nur durch Variation von wenigen Parametern (Tabellenname, -Spalte, etc) unterscheiden.

Was wie ein Umweg erscheint, wurde zum persönlichen Komfort des Benutzers implementiert.

Funktion db_adventure

Der spielerisch gewählte Name bezieht sich auf die Ineffizienz eines solchen Verfahrens, was jedoch marginal leistungmindernd ist. Es erlaubt eine bessere Codelesbarkeit und minimiert den Aufwand sowie die Zeilenzahl.

Deklaration

```
1  function DB_adventure($tablename, $content, $contentNew, $IDname,
       $ID) {
2      ...
3      switch (is_NULL($contentNew)) {
4          case false:
5          $stmt = $conn->prepare("UPDATE " . $tablename . " SET " .
               $content . "=? WHERE " . $IDname . "=?");
6          $stmt->bind_param("ss", $contentNew, $ID);
7          if(false === $stmt->execute()) {
8              header("Location: ../501.html");
9              exit;
10         }
11         $stmt->close();
12         break;
13     ...
14 }
```

Listing 8.7: php/functions.php: Deklaration und Operationsweise der Funktion *db_adventure*

Im Wesentlichen wird zuerst geprüft, ob der Parameter **$contentNew** leer ist; sollte dem der Fall sein, wird die Ausführung der Funktion terminiert, da es keine Daten zum aktualisieren gibt. Ist der Parameter jedoch nicht leer, wird ein einfaches *Prepared Statement* ausgeführt, welches den entsprechenden Eintrag in die Datenbank einfügen soll. Die Lokation jenes Eintrages wird durch weitere Übergabeparameter festgestellt. **Die Übergabeparameter sind:**

- $tablename - Tabellenname
- $IDname - Name des eindeutigen Identifikationsparameters in der Datenbank (ID)
- $ID - Inhalt des Identifikationsparameters in der Datenbank
- $content - Name der Spalte in der Datenbank
- $contentNew - Inhalt der eingefügt werden soll

db_adventure verfügt über keine Rückgabewerte.

Funktion db_trip

Diese Funktion ist in ihrer Operation mit **db_adventure** absolut ident, der einzige Unterschied besteht darin, dass auf eine Überprüfung von der Anwesenheit des Wertes in $contentNew verzichtet wurde. Dies ist nützlich, wenn bereits vor dem Funktionsaufruf festgestellt wurde, ob der einzufügende Wert leer ist. Dadurch lässt sich also redundanter Code vermeiden. *db_trip* verfügt ebenfalls über keine Rückgabewerte.

8.1.7. Funktion message

Diese Funktion wird von jedem *PHP*-Skript aufgerufen, wenn *Messages* benötigt werden. Hierzu zählen:

- die letzten ungelesenen Nachrichten
- ungelesene und gelesene Nachrichten
- gesendete Nachrichten
- gelöschte Nachrichten

Zusätzlich können ältere Nachrichten geladen werden, wofür die *messageID* der letzten Mitteilung in einer *SESSION*-Variable abgespeichert wird:

```
1  function message($userID, $actualise, $next) {
2  (...)
3  if($next == "nextinbox") {
4      $stmt = $pdo->prepare("SELECT * FROM messaging_
5                             (...)
6                             WHERE (...)
7                             AND messageID < :messageID
8                             (...)
9                             LIMIT 10");
10     $stmt->bindValue(":messageID", intval($_SESSION['lastMessage'
           ]), PDO::PARAM_INT);
11 }
12 (...)
13 $_SESSION['lastMessage'] = $_SESSION['messages'][count($_SESSION['
       messages'])-1]['messageID'];
14 (...)
15 }
```

Listing 8.8: php/functions.php: Auslesen weiterer Nachrichten für *Inbox*

In der Funktion *message* (vgl. 8.8) wird durch den Übergabeparameter *$next* bestimmt, welcher Teil der Nachrichten ausgelesen werden soll. In diesem Code-Ausschnitt wird gezeigt, wie beim Nachladen von *Messages* in der *Inbox*-Ansicht alle Einträge aus der Datenbank gelesen werden, deren *messageID* kleiner als jener Wert von *$_SESSION['lastMessage']* ist. Dabei werden immer nur zehn Datensätze durch *LIMIT 10* pro durchgeführter Abfrage ausgelesen.

Sollte die Abfrage erfolgreich gewesen und tatsächlich zehn Eintrage ausgelesen worden sein, wird die *messageID* der letzten Nachricht in die *SESSION*-Variable *messageID* gespeichert.

Auf diese Weise werden Nachrichten für die Ansichten *Sent* und *Deleted* geladen bzw. ältere Nachrichten nachgeladen.

8.1.8. Funktion auto_courses

Sollte ein Schüler sich nicht bis zu einem Tag vor einem Zeitbereich zu einem Lehrer zugeteilt haben, wird er vom System automatisch eingeteilt. Dieser wird über die Aktion per *Message* informiert.

Für diesen Prozess werden zuerst alle Kurse des Schülers und danach die Raumauslastung für denjenigen Zeitbereich analysiert. Dadurch kann er einem Lehrer zugeteilt werden, der die wenigsten Teilnehmer in Relation zur Raumgröße hat und gleichzeitig von diesem Schüler bisher am wenigsten besucht wurde. Die Funktion *auto_courses* verwendet denselben Algorithmus, der für *Recommended Courses* zum Einsatz kommt.

8.2. function.js

Bis auf *deleteProof* dienen alle Funktionen der *JavaScript*-Datei *function.js* zum Manipulieren von *Select*-Feldern. Wenn auf einer Seite mehrere *Select*-Felder die gleichen Einträge zur Auswahl anbieten, werden die bereits ausgewählten Eintrage von den anderen Feldern entfernt, sodass der Benutzer keine Möglichkeit zum Auswählen dieser hat. Sollte dieser trotzdem clientseitig einen Weg finden, wird dies durch ein *PHP*-Skript serverseitig verhindert.

8.2.1. Klasse sVariable

Diese Klasse stellt die Funktion *createArray* bereit, mit der ein mehrdimensionales *Array* mit der benötigten Länge instanziiert wird. Da dies in *JavaScript* nicht primär vorgesehen ist, erfordert es mehrere Befehle:

```
1  class sVariable{}
2
3    function createArray(length) {
4      var arr = new Array(length || 0),
5          i = length;
6
7      if (arguments.length > 1) {
8          var args = Array.prototype.slice.call(arguments, 1);
9          while(i--) arr[length-1 - i] = createArray.apply(this,
               args);
10     }
11
12     return arr;
13 }
```

Listing 8.9: js/function.js: Erstellen eines Arrays der Klasse *sVariable*

Da der Übergabeparameter *length* in 8.9 aus mehreren Argumenten aufgebaut sein kann, wird mit *arguments.length* (siehe Zeile 7) überprüft, ob ein mehrdimensionales *Array* erstellt werden soll. Ein vier-dimensionales *Array* mit jeweils drei Feldern, die aus je zwei Feldern bestehen wird bspw. mit *createArray(4,3,2)* erstellt. Die Funktion ruft sich selbst auf und knüpft die erzeugten *Arrays* an das eigentliche Array an, sodass ein n-dimensionales Array instanziiert werden kann. Dies wird nur durch die Beschränkung des für den Prozess zugelassenen Speichers limitiert.

8.2.2. Funktion unsign_value

Diese Funktion wird aufgerufen, sobald bei einem *Select* eine *Option* ausgewählt wurde. Voraussetzung dafür ist, dass diese mit dem Befehl *onchange* belegt ist und die richtigen Übergabeparameter besitzt. Hierbei werden der *Value* und die *ID* des *Selects*, sowie das *Keyword* für die Suche innerhalb der Funktion und die Stelle des Arrays übergeben.

Zuerst wird die Anzahl der *Selects* mit denselben *Options* gesucht, weshalb das *Keyword* einen Übergabeparameter darstellt:

```
1 var keyword = "outer-" + key;
2
3 var outerSkill = document.getElementById(keyword).innerHTML;
4 var count = outerSkill.split("selectlimitation").length - 1;
```

Listing 8.10: js/function.js: Suchen aller *Selects*

Daraufhin wird überprüft, ob die Größe des vorhandenen *Arrays* zur Speicherung der entfernten *Options* genügt, andernfalls wird dieses erweitert:

```
1 while (sVariable.staticVariable.length <= nr) {
2       sVariable.staticVariable.push(createArray(1));
3 }
```

Listing 8.11: js/function.js: Erweiterung eines Arrays

Sofern von dem *Select* bereits eine *Option* ausgewählt und deshalb von den restlichen Betroffenen entfernt wurde, wird sie mit *add_value* hinzugefügt (siehe 8.2.3). Danach wird die derzeit ausgewählte *Option* von allen *Selects* ausgenommen dem Initiierenden gelöscht (siehe 8.2.4).
Nachdem sowohl *add_value* als auch *remove_value* erfolgreich ausgeführt wurden, wird die hinzufügte *Option* aus dem jeweiligen *Array* gelöscht und die Entfernte diesem hinzugefügt.

8.2.3. add_value

Diese Funktion fügt jedem übergebenen *Select* die zuletzt von dem initiierenden *Select* entfernte *Option* aus dem *Array* an letzter Stelle hinzu.

8.2.4. remove_value

Anfangs wird das ausgewählte *Select* mit der übergebenen *ID* durch *document.getElementById* ausgewählt. Danach wird über alle *Options* iteriert und bei Übereinstimmen des Wertes mit der übergegebenen *Option* diese entfernt.

8.2.5. deleteProof

Das Löschen von Einträgen, die im System keine signifikante Änderung hervorrufen und mühelos wieder erstellt werden können, bspw. das Entfernen von einzelnen Schülern aus *Lectures*, kann ohne doppelte Bestätigung ausgeführt werden. Sollte es sich jedoch um das Löschen von Nutzerkonten, Räumen oder Zeiteinheiten handeln, muss der Nutzer dieses durch eine zusätzliche Zustimmung innerhalb eines *JavaScript-Prompts* veranlassen. Da *JavaScript* clientseitig deaktiviert werden kann, ist ein Löschvorgang dieser Art nur mit aktivem *JavaScript* durchführbar.

Klickt der User auf einen *Löschen*-Button, der diese Funktion aufruft, öffnet sich bei ihm ein *JavaScript-Prompt*. Dieses fragt, ob man tatsächlich den folgenden Eintrag löschen möchte, und fordert den Nutzer auf, *Yes* (exakte Schreibweise mit Beachtung der Groß- und Kleinschreibung) einzutippen. Bei Abbrechen des *Prompts* oder falscher Eingabe wird der Vorgang nicht weiter ausgeführt.

Bei richtiger Eingabe wird das *PHP*-Skript *delete.php* aufgerufen (vgl. 8.12) und der Eintrag wird vollständig aus dem System und der Datenbank ohne Möglichkeit zur Wiederherstellung entfernt.

```
1 var methodSelection = prompt("Do you really want to delete " +
    name + "? (Type in 'Yes' to confirm)");
2 if (methodSelection != null) {
3     if(methodSelection == 'Yes') {
4     (...)
5     xmlhttp.open("GET","php/delete.php?(...)",true);
6     xmlhttp.send();
7     }
8 }
```

Listing 8.12: js/function.js: Doppelte Überprüfung eines Löschvorgangs

9. Öffentlicher Test

9.1. Fragebögen

Da diese Diplomarbeit eine Machbarkeitsstudie für den Einsatz eines Werkstättenplaners/-Managementtools ist, wurden im Prozess auch Tests mit mehreren Klassen und in der Werkstätte unterrichtenden Lehrpersonen durchgeführt. In den folgenden Abschnitten wird die Durchführung dieser und die damit verbundene Aufnahme von Meinungen und Verbesserungsvorschlägen beschrieben und die Auswertung dieser dargelegt.

9.1.1. Fragebögen für Schüler

Der Fragebogen für Schüler beinhaltet Fragen über allgemeine Informationen zur Person für die Einteilung in Alters- und Schulklassen, Fragen über das bisherige Projekt, die Methode zum Verwalten von Lehrstoff und über das getestete System (WSS).

Strukturierung

- Persönliche Informationen
  - Klasse
  - Alter
  - Kategorie des bisherigen Projekts (Hardware, Software, beides)
  - Thema des Projektes
  - Methode zum Notieren von Informationen (analog, digital, *im Kopf*)
- Informationen über das WSS
  - Benutzerfreundlichkeit (schlecht - perfekt)
  - Design (unansehnlich - schön)
  - Hinzufügen von Kursen (kompliziert - sehr einfach)
  - Empfehlungen von Räumen/Lehrern (unnötig - sinnvoll)
  - fehlende Funktionen

  - freiwillige Nutzung des Systems (in keinem Fall - ja)
  - WSS gegenüber derzeitiger Verwaltung von Projekten (besser/nicht besser)

9.1.2. Fragebögen für Lehrpersonen

Der Fragebögen für Lehrpersonen ist grundsätzlich analog zu jenem für Schüler aufgebaut: Es gibt Fragen zur Person betreffend dem unterrichtenden Fach und der Methode zum Verwalten von Informationen. Außerdem wird das Empfinden des WSS behandelt.

Strukturierung

- Persönliche Informationen
  - Kategorie des unterrichtenden Fachs (Hardware, Software, beides)
  - Methode zum Notieren von Informationen (analog, digital, *im Kopf*)
- Informationen über das WSS
  - Benutzerfreundlichkeit (schlecht - perfekt)
  - Design (unansehnlich - schön)
  - Ansicht der Schüler pro Woche (kompliziert - sehr einfach)
  - Kommunikation mit anderen Lehrern/Schülern (schlecht umgesetzt - sehr gut umgesetzt)
  - fehlende Funktionen
  - freiwillige Nutzung des Systems (in keinem Fall - ja)
  - WSS gegenüber derzeitiger Verwaltung von Projekten (besser/nicht besser)

9.1.3. Durchführung der Tests

Das Testteam besteht aus einem unterrichtenden Werkstättenlehrer und der zu diesem Zeitpunkt unterrichteten Klasse. Dabei hat sich der Lehrer in den meisten Fällen schon geringfügig in das System eingearbeitet. Da sich Schüler nur mit einem *Access Code* registrieren können, den sie vom Lehrer bekommen, sehen diese das WSS erstmals.

Ein Test beginnt grundsätzlich damit, dass den Probanden das Projekt und dessen Zweck kurz erklärt wird. Daraufhin registrieren sich alle Schüler mit den zuvor ausgeteilten *Access Codes* und verwenden das System für mindestens eine dreiviertel Stunde. Die maximale Testdauer beträgt zwei Stunden, abhängig davon, wie aktiv oder interessiert das Team sich verhält. Während der Tests stehen die beiden Entwickler für Fragen bereit, beeinflussen jedoch den Ablauf so gering wie möglich. Alle aufgetretenen Meinungen und Bugs, die innerhalb dieser Zeit mündlich geäußert werden, werden aufgeschrieben und ebenfalls verwertet.

Eine halbe Stunde vor vorgesehenem Testende werden die weiter oben genannten Feedbackbögen ausgeteilt, sodass die Probanden genug Zeit haben, die Fragen gewissenhaft auszufüllen. Anfangs sollen sich die Tester auf das eigentliche und unabhängige Testen konzentrieren, weshalb die Fragebögen zu Beginn noch nicht ausgehändigt sind.

9.2. Feedback

Die Auswertung der Feedbackbögen wird auf die Gruppe der Lehrer und jene der Schüler aufgeteilt, der Durchschnitt der Antworten pro Fragestellung ermittelt und in Diagrammen dargestellt und interpretiert. Zusätzlich werden einzelne Aussagen und Meinungen erwähnt, die wegen der Individualität nicht zusammengefasst und durch Zahlen bewertet werden können.

9.2.1. Feedback der Schüler

Es wurden sowohl Klassen befragt, die derzeit an einem eigenen Projekt arbeiten, als auch jene, die damit noch keine Erfahrung haben. Letztere sind deshalb von Relevanz, da das System bereits zum Zeitpunkt des Erstellens eines Projekts benützt werden soll.

Position	Durchschnittswert
Alter	16,35
Geschlecht	männlich
Bereiche des Projekts	ausgeglichen zwischen Hardware und Software
Zeitinvestition in das Projekt	so viel schulische Zeit, wie möglich
Relevanz des Projekts	ziemlich wichtig
Medium zum Notieren von Informationen	digital

Tabelle 9.1.: Allgemeine Informationen zu den Probanden (Schüler)

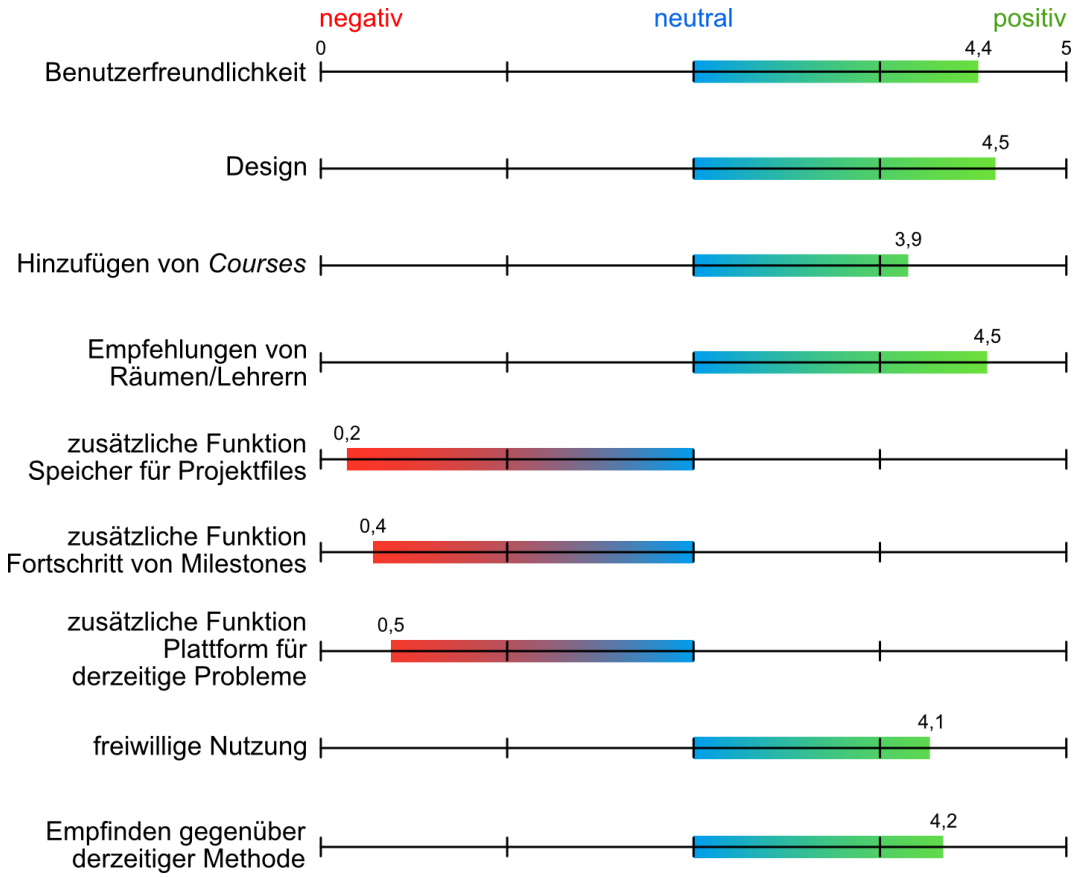


Abbildung 9.1.: Durchschnittliche Antworten von Schülern

Wie in 9.1 ersichtlich ist, wird die Software durchschnittlich als *sehr gut* bewertet und die im Fragebogen vorgeschlagenen zusätzlichen Funktionen werden als eher nicht nötig empfunden. Selbst wenn das System nicht vorgeschrieben ist, würden es viele freiwillig nutzen und finden es besser als die derzeitige Verwaltung von Projekten. Das Hinzufügen von *Courses* (3. Zeile in 9.1) wird im Vergleich zu der gesamten Benutzerfreundlichkeit als etwas schwieriger gewertet. Dessen ungeachtet liegt der Mittelwert der Antworten deutlich im positiven Bereich, d. h. es wird als *unkompliziert* bis *leicht zu nutzen* gewertet.

Empfehlungen von Räumen (4. Zeile) werden allgemein als sehr sinnvoll erachtet, und im gesamten Test wertete nur ein Schüler diese Funktion als *unnötig*. Selbst Schüler, die die Projekte an sich als unwichtig empfinden, würden das WSS gerne nutzen. Anzunehmen ist, dass sich dies durch Funktionen ergibt, die nicht unbedingt an das Projekt gebunden sind, wie das Bestellen von Leiterplatten oder die Kommunikation zwischen Lehrern und Schülern.

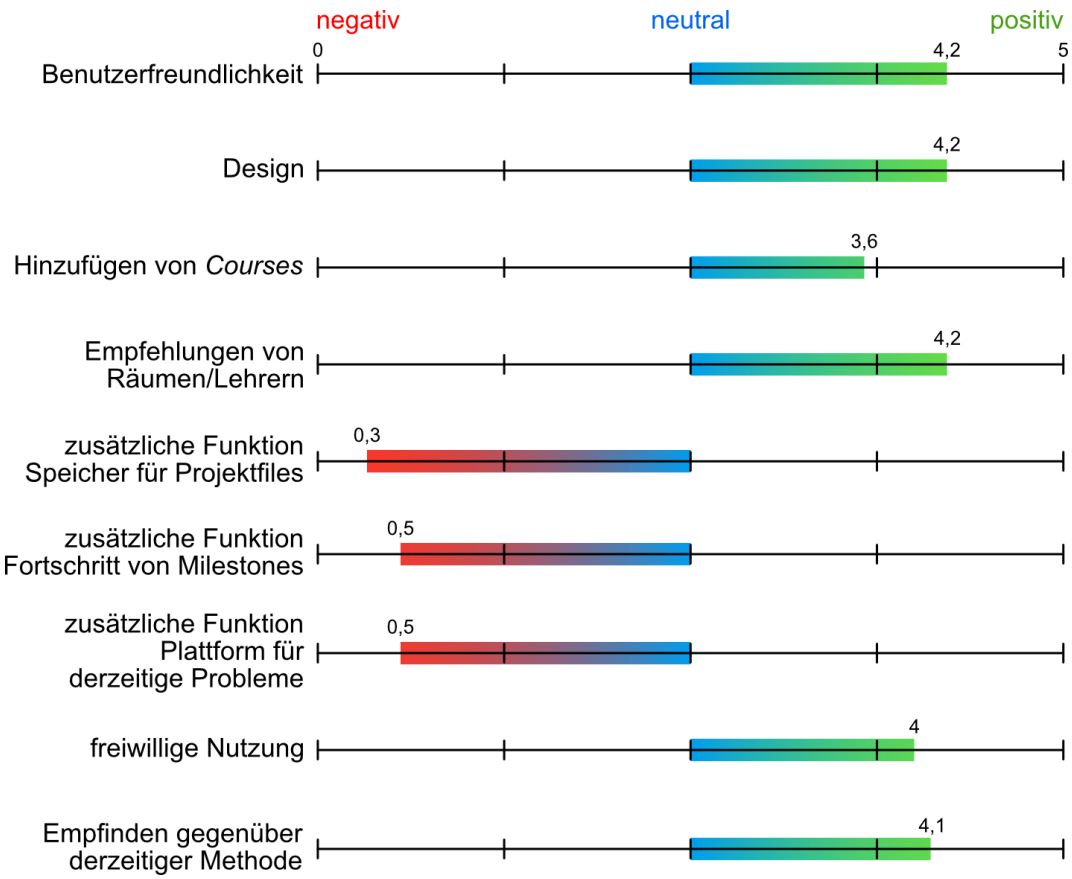


Abbildung 9.2.: Durchschnittliche Antworten der 4. Klasse

Betrachtet man die Antworten der 4. Klasse (ohne Fachschule) (vgl. 9.2), die derzeit an Projekten arbeiten, abgeschottet, so fallen die Bewertungen in allen Bereichen etwas negativer im Vergleich zum Durchschnitt aus (um ca. 0,2 von 5 Punkten). Bei diesem Test haben außerdem mehrere Schüler angegeben, dass sie gerne zusätzliche (eigens definierte, in 9.2 nicht ersichtliche) Funktionen in das System implementiert wissen möchten.

Fragt man nach der *freiwilligen Nutzung* und dem *Empfinden gegenüber* der *derzeitige(n) Methode*, so fallen die Antworten nur um 0,1 Punkte geringer aus. Eine Interpretationsmöglichkeit kann lauten, dass diese mehr Erfahrung im Testen und Nutzen von derartigen Systemen besitzen und deshalb kritischer sind. Im Endeffekt würden sie das Produkt jedoch genauso gerne, wie alle Schüler-Probanden, nutzen.

9.2.2. Feedback der Lehrer

Wie bei den Tests mit Schülern wurden sowohl projektbetreuende Lehrer als auch jene, die mehr auf Frontalunterricht setzen, als Probanden eingesetzt. Somit erhält man einen Überblick, wie das System im Hinblick auf die spezielle Projektverwaltung und die Nutzung von *Lectures*, Nachrichten und weiteren Funktionen bewertet wird.

Position	**Durchschnittswert**
Geschlecht	männlich
Bereiche des Unterrichts	ausgeglichen zwischen Hardware und Software
Medium zum Notieren von Informationen	abhängig von der Situation

Tabelle 9.2.: Allgemeine Informationen zu den Probanden (Lehrern)

Anmerkung: Die Anzahl der befragten Lehrer ist aufgrund dem in jeder Schule ähnlich vorliegendem Lehrer-Schüler-Verhältnis deutlich geringer, wodurch die Zahl der Rückmeldungen relativ niedrig ausfällt.

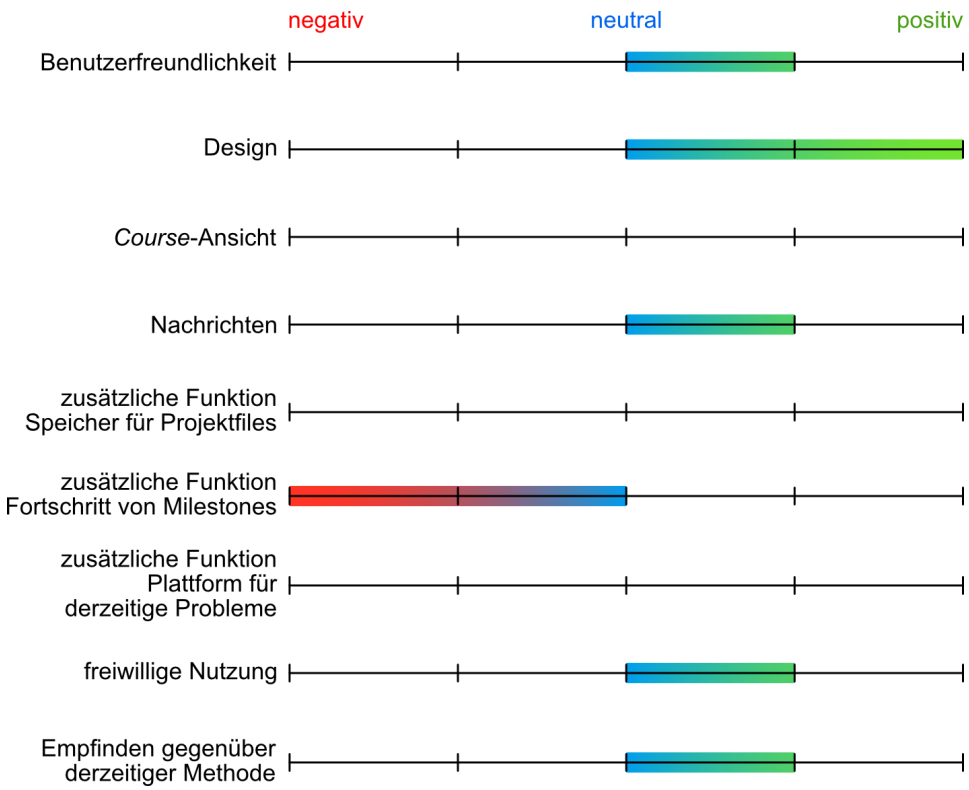


Abbildung 9.3.: Durchschnittliche Antworten der Probanden (Lehrer)

Insgesamt wird das Design als *schön* angesehen (vgl. 9.3). Ebenso werden die Benutzerfreundlichkeit und das Nachrichten-System als *gut* erachtet. Einzig die *Course-Ansicht* wird im Durchschnitt als *mittelmäßig* (d. h. weder gut noch schlecht) gewertet - sie könnte also noch vereinfacht bzw. übersichtlicher gestaltet werden.

Als nützliche zusätzliche Funktion wird der *Speicher für Projektfiles* und eine *Plattform für derzeitige Probleme* angegeben, der *Fortschritt von Milestones* wäre jedoch ein wenig genutztes Element.

Analog zur Befragung der Schüler können sich im Durchschnitt alle Lehrer vorstellen, das System auch auf freiwilliger Basis zu benutzen, und bewerten es *besser* als die derzeit benutzte Methode.

Gewünschte Funktionen
Einige Lehrer wünschen sich zum zusätzlichen Stand des WSS noch das Benutzen von mehreren Räumen für einen Tutor (besonders für *Lectures*), das Kopieren von Theoriestunden und die Zuordnung von Schülern zu *Projects* durch die betreuende Lehrperson.

9.3. Einzelfeedback von Fachlehrer Rzepa

Dieser Paragraf wird Fachlehrer Ing. Reinhard Rzepa und seinem Feedback gewidmet, das er freundlicherweise in einem frühen Stadion der Software innerhalb kürzester Zeit und mit genauen, sehr hilfreichen Angaben zur Verbesserung und Erweiterung bereitgestellt hat. Dafür möchten sich die beiden Diplomanden bereits an dieser Stelle herzlichst bedanken.

9.3.1. Inhalt des Feedbacks

Antworten auf Nachrichten

Das Nachrichten-System bestand zu diesem Zeitpunkt noch aus einem sehr einfachen Aufbau mit einem undurchsichtigen Design. Durch den Vorschlag, eine Antwortmöglichkeit einzubauen, wurde das Nachrichtensystem neu überdacht und mit einem Design-Template aufgebaut, weshalb es nun für deutlich mehr Nachrichten geeignet ist.

Durch das Antworten auf Nachrichten und einen Betreff können diese deutlich einfacher wiedergefunden und eine Konversation komfortabel geführt werden.

Anzeige der gesendeten Nachrichten

Durch das neue Nachrichtensystem ist es möglich, die gesendeten Nachrichten erneut zu öffnen und anzusehen. Durch diese Funktion kann nachverfolgt werden, ob eine Mitteilung bereits vom Empfänger gelesen wurde.

Erhöhung der maximale Länge der Nachrichtenlänge

Wie im Unterabschnitt oberhalb erwähnt, war die frühere Version des Nachrichtensystems nur für sehr kurze Nachrichten und System-Benachrichtigungen geplant. Da mit dem Betreff und Beantworten von *Messages* umfangreichere Konversationen entstehen können, wurde das Limit der Nachrichtenlänge von 255 Zeichen auf 3072 erhöht.

Bearbeiten des eigenen Raumes

Im Grundkonzept wurde definiert, dass ein Raum durch den Admin erstellt und bearbeitet wird. In den meisten Fällen weiß jedoch der unterrichtende Lehrer über diesen besser Bescheid, da er täglich darin arbeitet. Deshalb wurde das Bearbeiten vom eigenen Raum durch den Lehrer eingebaut. Damit jedoch die Kontrolle durch den Admin bestehen bleibt, wird die Räumlichkeit durch diesen zugeteilt.

Expansion des aktuellen Menüpunktes in der Seitenleiste

Anfangs war in der Seitenleiste dauerhaft das Menü *Education* geöffnet. Durch die Anregung von Fachlehrer Rzepa ist nun das jeweilige Verzeichnis erweitert, von dem ein Punkt gewählt wurde.

Erweiterung durch Lectures

Da die Grundvoraussetzung das Hinzufügen von *Courses* durch Schüler war, wurde anfänglich die Funktion für Theoriestunden nicht implementiert. Durch das erneute Fordern dieser Funktion von Ing. Rzepa (das Feature wurde bereits bei den Expertisen von einigen Lehrern gefordert, vgl. 4.5) wurde diese in das System eingebaut.

9.3.2. Effekt des Feedbacks

Wie im vorherigen Unterabschnitt ersichtlich, wurden einige vom derzeitigen Stand aus betrachtete essenzielle Funktionen nachgereicht und das System dadurch deutlich verbessert. Außerdem war es so möglich, das WSS rechtzeitig an die Bedürfnisse der Anwender anzupassen und diese weitgehend zu erfüllen.

10. Betriebswirtschaftliche Kalkulation

10.1. Einzelkosten

Die Einzelkosten werden zum einen für das derzeitige Projekt als Neuentwicklung und im zweiten Schritt als angepasste Version für weitere Unternehmen berechnet. Da das WSS sehr flexibel aufgebaut ist, kann es mit wenigen Änderungen für andere Kunden und deren Wünsche aufbereitet und so von mehreren Firmen verwendet werden.

In beiden Fällen werden ausschließlich die Arbeitskosten für die Entwicklung herangezogen, wodurch angenommen wird, dass die gesamte Hardware vom Konsumenten bereitgestellt wird.

10.1.1. Neuentwicklung

Variable Kosten

Die variablen Kosten bestehen aus den notwendigen Arbeitsstunden:

Kostenart	Person	€ pro h	Stunden	Gesamt [€]
Entwicklungskosten	Kiril Vereshchagin	30	204	6.120
	Bernhard Würfel	30	195	5.850
Gesamt			399	11.970

Tabelle 10.1.: Variable Kosten für die Neuentwicklung

Gewinnzuschlag

Um das Produkt gewinntreibend zu verkaufen, wird ein Gewinnzuschlag in der Höhe von 50% hinzugerechnet.

Position	Prozentuelle Menge [%]	Absolute Menge [€]
Variable Kosten	-	11.970
Gewinnzuschlag	50	5.985
Gesamt	-	17.955

Tabelle 10.2.: Variable Kosten mit Gewinnzuschlag für die Neuentwicklung

10.1.2. Anpassung für den Wiederverkauf

Die variablen Kosten bestehen wie bei der Neuentwicklung aus den notwendigen Arbeitsstunden, deren Anzahl in diesem Fall deutlich geringer ist. Da jeder Kunde andere Wünsche hat, ist die notwendige Zeit zur Adaption sehr schwer vorhersagbar - in folgender Berechnung wird ein Durchschnitt von 30 Stunden pro Person angenommen:

Kostenart	Person	€ pro h	Stunden	Gesamt [€]
Entwicklungskosten	Kiril Vereshchagin	30	30	900
	Bernhard Würfel	30	30	900
Gesamt			60	1.800

Tabelle 10.3.: Variable Kosten für eine Adaption

Gewinnzuschlag

Um das Produkt gewinntreibend zu verkaufen, wird ein Gewinnzuschlag auf den fertigen Artikel in der Höhe von 50% aufgeschlagen.

Position	Prozentuelle Menge [%]	Absolute Menge [€]
Variable Kosten	-	1.800
Gewinnzuschlag	50	900
Gesamt	-	2.700

Tabelle 10.4.: Variable Kosten mit Gewinnzuschlag für eine Adaption

10.1.3. Fixkosten

Für diese Produktion ist die Summe der Fixkosten und jene der Gemeinkosten dieselbe, da sowohl für eine Neuentwicklung, als auch für eine Überarbeitung die gleichen Ressourcen benötigt werden. Dieser Punkt wird daher in 10.2 abgehandelt.

10.2. Gemeinkosten

Zu den *Overheads* (dt. Gemeinkosten) zählen alle Aufwände, die keinem einzelnen Kostenträger direkt zugeteilt werden können. In diesem Fall sind das Ausgaben für die Bereitstellung und Instandhaltung des Büroraums für die Entwicklung und Lagerung aller betriebsrelevanten Unterlagen.

In folgender Tabelle ist eine grobe Abschätzung der *Overheads* ersichtlich:

Position	**Kosten/Monat [€]**
Miete Büroraum	660
Allgemeine Instandhaltungskosten	200
Internet + Telefonie	30
Gesamt	890

Tabelle 10.5.: Gemeinkosten

10.3. Break-Even-Point

Der *Break-Even-Point* beschreibt den Zeitpunkt, an dem das Produkt die Gewinnzone erreicht hat und alle Kosten gedeckt sind. Im Fall dieser Diplomarbeit wird mit einem Mehrfachverkauf der Software gerechnet, um eine dynamischere Gewinnfunktion zu erzielen. Daher wird nach der Basisversion eine abgewandelte Version an weitere Kunden verkauft, wodurch laufend Kosten entstehen, jedoch auch Erträge erwirtschaftet werden.

Der folgende *BEP* wird in der Annahme berechnet, dass nach der fertigen Entwicklung einmal die Basisversion und danach fortlaufend die für die weiteren Kunden angepasste Version verkauft wird. Der Verkaufspreis bleibt jedoch gleich und wird als 5.000 € angenommen.

Es werden 7 Monate Entwicklungszeit angenommen, in denen jeweils die Ausgaben als gesamter Verlust anfallen. Erst nach diesen Monaten wird mit einem monatlichen Ertrag von 5.000 € gerechnet.

$$\text{Entwicklungskosten für die Basisversion } EEK \text{ (einmalig)} = 11.970 \text{ €} \tag{10.1}$$

$$\text{Entwicklungskosten für die Adaption } AEK \text{ (monatlich)} = 1.800 \text{ €} \tag{10.2}$$

$$\text{Gemeinkosten } GK \text{ (monatlich)} = 890 \text{ €} \tag{10.3}$$

$$\text{Ertrag } E \text{ (monatlich)} = 5000 \text{ €} \tag{10.4}$$

$$E * t - [GK * (7 + t) + EEK + AEK * t] > 0 \tag{10.5}$$

$$t > 7.88 \text{ Monate} \tag{10.6}$$

Das Produkt und deren Adaptionen müssen länger als 8 Monate verkauft werden, um als gewinnbringend zu gelten.

11. Ergebnisse

Alle angegebenen Ziele und teilweise zusätzlich geforderten Wünsche seitens der Lehrer wurden erfüllt und das System für den Einsatz in der Werkstätte einer technischen Schule bereitgestelltt. Wie in Kapitel 7 erwähnt wurde, ist das WSS lokal und global erreichbar.

Wie aus den Tests und Befragungen hervorgegangen ist (siehe Kapitel 9), ist der derzeitige Stand für die Großzahl der Personen geeignet und beweist, dass eine derartige Methode zum Gestalten und Verwalten des Unterrichts möglich und erwünscht ist. Besonders aus Schülersicht erweist sich diese Art als lukrativ und ermöglicht einfacheres, effizienteres, und vor allem eigenständiges Arbeiten. Aus der Sicht einer Lehrperson bietet die Übersicht von Projekten und Schülern Vorteile. Durch das Einführen von *Lectures*/Theoriestunden wurde auch die größte Sorge dieser - der zu geringen Zeit zum Vortragen des Lehrstoffes - weitgehend beseitigt.

Für den tatsächlichen Einsatz in einer spezifischen Schule sind geringe Änderungen und Erweiterungen, die von den dort anwesenden Lehrern und Schülern erwünscht wären, notwendig. Da das System in der HTBLuVA St. Pölten getestet und für diese weiterentwickelt wurde, ist ein sofortiges Arbeiten mit dem WSS möglich.

A. Aufwand

In den folgenden Tabellen wird der Aufwand der beteiligten Personen aufgeschlüsselt. Die Arbeitsbeschreibung ist nach Kalenderwochen und die aufgewendete Zeit nach Monaten gegliedert.
Die Gesamtarbeitszeit beträgt:

Bernhard Würfel	224h
Kiril Vereshchagin	246h

KW	Arbeitsinhalt
	September (insg. 10h)
36	Projektmanagement
37-39	Besprechungen bezgl. Anforderungen und derzeitiger Plattform
	Oktober (insg. 25h)
40	Konzepterstellung
41	Besprechung des Konzepts
42	Besprechung mit Werkstättenlehrern und Printerei bzgl. Vorstellungen und Verantwortung
43	Expertisen mit einzelnen Lehrern
	November (insg. 55h)
44	Erarbeitung und Besprechung der angeforderten Funktionen
45	Erstellung eines Zeitplans für Durchführung
46	Planung der *Course*-Einteilungen und deren Optimierung
47	*Courses*-Implementierung für alle Benutzerebenen
	Dezember (insg. 70h)
48-49	Planung und Implementierung von *Fixed Courses*
50	Implementierung von *Messages*
51	Implementierung der Leiterplattenbestellung
52	interne Testphase

KW	Arbeitsinhalt
	Jänner (insg. 25h)
01	Bug-Fixing
02	Besprechung des aktuellen Stands mit Werkstättenlehrern
03	Beginn der Dokumentation
04	Präsentation des Systems vor dem Testteam
	Februar (insg. 20h)
05	Einholung der Testergebnisse
06	Besprechungen mit Teilnehmern des Test
07-08	Schreiben an Dokumentation
	März (insg. 19h)
09-11	Schreiben an Dokumentation
12-13	Fertigstellung der Dokumentation, Fehlerausbesserung

Tabelle A.1.: Aufwandaufschlüsselung Bernhard Würfel

KW	Arbeitsinhalt
	September (insg. 10h)
36	Projektmanagement
38	Erstellung von Designkonzepten
	Oktober (insg. 24h)
40	Erforschung technischer Möglichkeiten
41	Erstellung des Grundgerüsts
43	Grundlegende Modellierung der Datenbank
	November (insg. 48h)
44	Serverinfrastruktur bereitgestellt
45	Angeforderte Funktionen aus technischer Sicht
46	Registrierungs- und Loginsystem implementiert
47	Erweiterung der Datenbank
	Dezember (insg. 40h)
48	Sicherheitsfunktionen am Server
49	Entwicklung diverser Sicherheitsfeatures der Webseite
50	Entwicklung der Validierungsfunktionen

KW	Arbeitsinhalt
	Jänner (insg. 50h)
01	Berechtigungsebenen implementiert
02	Benutzerprofile, sowie deren öffentliche Ansicht implementiert
03	Hinzufügen von Lehrern implementiert, sowie interne Testung
04	Präsentation des Systems
	Februar (insg. 30h)
05	Sekundäre Features implementiert
06	Management sekundärer Features integriert
07-08	Beginn der Dokumentation
	März (insg. 44h)
09	Raummanagement für Lehrer
10	Behebung primärer Fehler
11-13	Fertigstellung der Dokumentation

Tabelle A.2.: Aufwandschlüssel Kiril Vereshchagin

B. Abkürzungsverzeichnis

EL	Elektronik und Technische Informatik
HTBLuVA St. Pölten	Höhere Technische Bundes Lehr- und Versuchs-Anstalt St. Pölten
HWE	Hardwareentwicklung
LA1	Laboratorium
SOTE	Softwaretechnik
FSST	Fachspezifische Softwaretechnik
AV	Abteilungsvorstand
WSS	Workshop Scheduling System (dt. Werkstättenplannungs-System)
XSS	Cross-Site-Scripting (dt. webseitenübergreifendes Scripting)
SQL	Structured Query Language (dt. strukturierte Abfrage-Sprache)
HTML	Hypertext Markup Language (dt. Hypertext Auszeichnungssprache)
CSS	Cascading Style Sheets (dt. gestufte Gestaltungsbögen)
PHP	Hypertext Preprocessor
CSV	Comma-Separated Values
GUI	Graphical User Interface (dt. grafische Benutzeroberfläche)
DOM	Document Object Model (dt. Dokumenten-Objekt-Model)
PC	Personal Computer
WWW	World Wide Web (dt. weltweites Netz)
IP	Internet Protocol

VPN	Virtual Private Network
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
HSTS	HTTP Strict Transport Security
IDE	Integrated Development Environment (dt. integrierte Entwicklungsumgebung)
ARM	Advanced RISC Machine
RISC	Reduced Instruction Set Computer (dt. Rechner mit reduziertem Befehlssatz)
CISC	Complex Instruction Set Computer (dt. Rechner mit komplexem Befehlssatz)

Glossar

Caching Ist ein Verfahren aus der Softwaretechnik, um Ressourcen anhand bestimmter Kriterien in einem temporären Speicher (Cache) abzulegen. Caching wird hauptsächlich mit dem Ziel eingesetzt, redundante Datenübertragungen und Serveranfragen zu vermeiden, sowie Latenz- und Wartezeiten zu verringern. Dies führt in den meisten Fällen zu einer erheblichen Entlastung des Servers. 37, 52, 53

Client Client oder zu dt. „Klient" (kaum gebräuchlich), ist ein *Computerprogramm*, das auf der Seite des Benutzer-Endgeräts installiert ist und auf die Kommunikation mit einem *Server* oder *Host* angewiesen ist; dies ist meistens zum Austausch von Informationen notwendig. 30, 31, 35, 37, 52, 94

Community Im Deutschen üblicher Anglizismus oder technischer Jargon für „Internet-Gemeinschaft". Solche Gemeinschaften bilden sich hauptsächlich zwecks Informationsaustausch, Kommunikation oder Entwicklung einer *Open Source*, also lizenzfreier, Software. 32, 33, 35, 39

Debugger Ist eine Software, meistens Teil einer IDE, die zur Diagnose von Computer-Programmen oder (Code-)Fehlern in diesen verwendet wird. 9, 11

Drag-And-Drop -Elemente können durch Drücken und Halten der linken Maustaste gezogen und durch Loslassen dieser einem anderen Bereich zugeordnet werden - Grundfunktion von *HTML5* in Kombination mit *JavaScript*. 67

Framework (Softwaretechnik) Ist ein Programmiergerüst, mit dessen Hilfe ein Programm erstellt werden kann. Frameworks implementieren oft viele fertige „Baublöcke" (Programmteile/-Vorlagen), die baukastenartig zusammengesetzt und für den eigenen Zweck angepasst werden können. 32, 33

Inline-Editing Inline-Bearbeitung ist die Bezeichnung der Eigenschaft einer Webseite, über die gegebene Möglichkeit Inhalte direkt, so wie sie dargestellt sind, zu bearbeiten. 74, 76

Kick Einen Benutzer von einem Server *kicken* bedeutet im Allgemeinen, dass die derzeitige Sitzung beendet wird, der Nutzer diese jedoch erneut aufbauen kann. 71

Objektorientierte Programmierung "Unter Objektorientierung versteht man in der Entwicklung von Software eine Sichtweise auf komplexe Systeme, bei der ein System durch das Zusammenspiel kooperierender Objekte beschrieben wird. Der Begriff Objekt ist dabei unscharf gefasst; wichtig an einem Objekt ist nur, dass ihm bestimmte Attribute (Eigenschaften) und Methoden zugeordnet sind und dass es in der Lage ist, von anderen Objekten Nachrichten zu empfangen beziehungsweise an diese zu senden." (Wikipedia)[14]. Dadurch ergibt sich ein umso besser wartbarer und erweiterbarer Code. 8

Patch Ist eine Art von *Software-Update*, bei dem speziell auf das Beheben von Fehlern geachtet wird. 9, 10

Plugin Häufig als „Plug-in" geschrieben, ist eine Komponente der Softwaretechnik und wird abhängig vom Branchen-Zweig (bspw. Videospiele) auch als „Add-on" bezeichnet. Ein Plugin erweitert ein bestehendes Programm, eine Entwicklungsumgebung oder ein Framework, um zusätzliche Funktionalitäten. Die adressierte Software muss jedoch fähig sein, Plugins zu integrieren; in manchen Fällen müssen sie speziell für eine bestimmte Software entwickelt werden. 32

Prozedurale Programmierung "Die prozedurale Programmierung ergänzt das imperative Konzept aufeinander folgender Befehle um den Ansatz, einen Algorithmus in überschaubare Teile zu zerlegen. Je nach Programmiersprache werden diese Teile Unterprogramm, Routine, Prozedur oder Funktion genannt." (Wikipedia)[15]. Dies erspart dem Programmierer die Notwendigkeit, wiederkehrende Code-Teile neu zu schreiben und verbessert somit die Code-Lesbarkeit. 8

Regular Expression Ein *Regulärer Ausdruck* (engl. regular expression) oder *RegEx* genannt, ist eine Folge von Zeichen, die ein Suchmuster mittels spezieller Syntax definiert. Normalerweise wird dieses Muster verwendet, um in Zeichenketten (Strings) Übereinstimmungen zu finden. Unter anderem werden *Reguläre Ausdrücke* auch häufig zur Eingabeüberprüfung verwendet. *Regex* ist ein Instrument der theoretischen Informatik. 87, 88

Skalierbarkeit Beschreibt die Möglichkeit, ein System (Server, Netzwerk) ohne eine Neuplanung oder eine komplette Überholung zwecks Leistungsgewinn oder Leistungsverlust zu erweitern. 9, 11

Template Im allgemeinen Sinn ist ein Template eine Vorlage für digitale Medien. Solche Medien sind typischerweise Text-Dokumente, Spread-Sheets und illustrative Materialien. Sie können aber auch für komplexe Analyse-Instrumente oder Webseiten Vorlagen sein. 33

Update (Software) Ist eine Aktualisierung - entweder auf eine Nachfolgeversion, eine neuere Version oder eine andere Version der Software. 9, 10

WebUntis Ist ein Teil von Untis - einem Tool zur Erstellung und Wartung von Stundenplänen, Personal, Räumen und Schülern. 22

C. Abbildungsverzeichnis

D. Listings

E. Tabellenverzeichnis

F. Danksagung

Wir bedanken uns bei

AV Dipl.-Ing. Wolfgang Kuran für die Idee dieser Diplomarbeit,
seine aktive Unterstützung und die hilfreichen Tipps,

den beiden Kollegen **M. Suchy und A. Lohr** aus der Abteilung Informatik,
und ihrem gemeinsamen Unternehmen
Private Void, für die Bereitstellung von Hosting- und VPN-Services,

den Herren Fachlehrern **Bauer** und **Rzepa** für die Testung, Ideeneinbringung
und Verbesserungsvorschläge vor und während der Entwicklung,

Frau Fachlehrerin **Halmetschlager**, sowie den Herren Fachlehrern
Prantl und **Rzepa** für die Expertisen und Ideeneinbringungen,

und besonders den Herren Fachlehrern **Janeczek** und **Rzepa** für ihre ausschlaggebende Direktive, sowie bei
allen teilnehmenden Schülern für die Tests und Rückmeldungen.

G. Literaturverzeichnis

[1] Wikipedia. *Corporate Identity*. (2018). URL: https://de.wikipedia.org/wiki/Corporate_Identity (besucht am 16. 02. 2019).

[2] Wikipedia. *Atom (Texteditor)*. (2019). URL: https://de.wikipedia.org/wiki/Atom_(Texteditor) (besucht am 16. 02. 2019).

[3] Wikipedia. *IntelliSense*. (2019). URL: https://de.wikipedia.org/wiki/IntelliSense (besucht am 17. 02. 2019).

[4] Aigars Silkalns alias "Puikinsh". *Kiaalap Admin Template*. (2018). URL: https://github.com/puikinsh/kiaalap (besucht am 19. 03. 2019).

[5] Massachusetts Institute of Technology. *MIT Lizenzbedingungen*. (1988). URL: https://opensource.org/licenses/MIT (besucht am 24. 03. 2019).

[6] Herwig Diernegger. „Smarty". In: *Smarty* 1.0 (2017), S. 4.

[7] Oracle Corporation. *MySQL*. (2010). URL: https://www.mysql.com/de/ (besucht am 27. 03. 2019).

[8] Martin Walter. „Datenbanken". In: (2017), S. 2–5.

[9] E. F. Codd. *Extending the Database Relational Model to Capture More Meaning*. Bd. 4. 1979, S. 397–434.

[10] M. Suchy und A. Lohr. *Private Void*. (2018). URL: https://www.privatevoid.net/ (besucht am 01. 04. 2019).

[11] Randomness und Integrity Services Ltd. *RANDOM.ORG*. (2018). URL: https://www.random.org/ (besucht am 11. 10. 2018).

[12] Christoph Keese. *Silicon Valley*. München: Penguin Verlag, 2014.

[13] Paul Resnick. *Internet Message Format*. (2008). URL: https://tools.ietf.org/html/rfc5322 (besucht am 07. 03. 2019).

[14] Wikipedia. *Objektorientierte Programmierung*. (2018). URL: https://de.wikipedia.org/wiki/Objektorientierte_Programmierung (besucht am 20. 02. 2019).

[15] Wikipedia. *Prozedurale Programmierung*. (2018). URL: https://de.wikipedia.org/wiki/Prozedurale_Programmierung (besucht am 20. 02. 2019).

htl bildung mit zukunft	**HTBLVA St. Pölten** **Elektronik und Technische Informatik** **Ausbildungsschwerpunkt Embedded Systems**	**Reife- und Diplomprüfung**

Betreuungsprotokoll zur Diplomarbeit lfd. Nr.: 1

Themenstellung: Workshop Scheduling System
Kandidaten/Kandidatinnen: Kiril Vereshchagin (5BHELS), Bernhard Würfel (5BHELS)

Jahrgang: 5BHELS 2018/19
Betreuer/in: W. Kuran
Ort: HTBLuVA St. Pölten / Raum: W111
Datum: 01.10.2018
Zeit: 13:10 – 14:50

Besprechungsinhalt:

Name	Notiz
Vereshchagin	• Webspace Zugangsfreigabe • Präsentation des Konzepts für das Web Interface • Erstellung des finalen Entwurfs für das Web Interface • Erläuterung der Serverkonfiguration • Einigung auf die Rechte der einzelnen Userebenen • Ideenvorstellung zur Vereinigung mit dem Lehrplan
Würfel	• Besprechung des Punktesystems für überlappende Termine • Erläuterung der Anforderungen seitens der Lehrer/Verantwortlichen • Struktur der Eingabe/Ausgabe • Kontakt mit Werkstättenlehrer/-leiter

Aufgaben:

Name	Notiz	zu erledigen bis
Vereshchagin	Grundgerüst für die Datenbank	03.11.2018
	Grundlegende Funktionen des Webinterfaces	
Würfel	Finalisierung der Eingabestruktur	03.11.2018
	Anforderungen der Lehrer wurden eingeholt	

htl bildung mit zukunft	**HTBLVA St. Pölten** **Elektronik und Technische Informatik** **Ausbildungsschwerpunkt Embedded Systems**	**Reife- und Diplomprüfung**

Betreuungsprotokoll zur Diplomarbeit lfd. Nr.: 2

Themenstellung: Workshop Scheduling System
Kandidaten/Kandidatinnen: Kiril Vereshchagin (5BHELS), Bernhard Würfel (5BHELS)

Jahrgang: 5BHELS 2018/19
Betreuer/in: W. Kuran
Ort: HTBLuVA St. Pölten / Raum: W111
Datum: 12.11.2018
Zeit: 11:30 – 12:30

Besprechungsinhalt:

Name	Notiz

Vereshchagin	• Erläuterung der Serverkonfiguration
	• Stress-/Penetrations-Test des Servers
	• öffentlicher Domain-Zugriff
	• spezielle Behandlung der Printerei
	  ○ Aufträge von Schülern sollen durch Skript gelöst werden
	  ○ Rückmeldung der Printerei soll möglich sein (Rückgabetermin etc.)
	• Möglichkeit zur Eingabe von „Seminaren“ durch Lehrer

Würfel	• Ablehnung der Einbindung von WebUntis • Keine automatischen Vorschläge durch den Algorithmus aufgrund Projektnamen • Rubriken für Räume/Lehrer zur Vereinfachung der Zeit-/Raum-Eingabe des Schülers • Berücksichtigung der Printerei • Berücksichtigung von Seminaren

Aufgaben:

Name	Notiz	zu erledigen bis
Vereshchagin	Fertigstellung aller PHP-Skripte der Website	03.02.2019
Würfel	Fertigstellung des Algorithmus für Überprüfung	03.02.2019

	HTBLVA St. Pölten Elektronik und Technische Informatik Ausbildungsschwerpunkt Embedded Systems	Reife- und Diplomprüfung

Betreuungsprotokoll zur Diplomarbeit lfd. Nr.: 3

Themenstellung: Workshop Scheduling System
Kandidaten/Kandidatinnen: Kiril Vereshchagin (5BHELS), Bernhard Würfel (5BHELS)

Jahrgang: 5BHELS 2018/19
Betreuer/in: W. Kuran
Ort: HTBLuVA St. Pölten / Raum: W111
Datum: 10.12.2018
Zeit: 13:20 – 14:20

Besprechungsinhalt:

Name	Notiz
Vereshchagin	• Erklärung über die einzelnen PHP Skripte, Implementierung von JavaScript Elementen • Sicherheitsfeatures der Input-Felder, Möglichkeit zur Eingabe von „Seminaren“ durch Lehrer • Erläuterung der aktuellen Lage bezgl. Webinterface • detaillierte Besprechung über die Rechte einzelner User-Gruppen (Admin, Lehrer, Schüler)
Würfel	• garantiert fixe Kurse für Schüler • Kurse, die durch garantiert fixe Kurse ersetzt werden können
Allgemein	• Start einer Testphase mit einer geringen Zielgruppe von Anwendern. • Erfahrungen sollen anhand der Ergebnisse der Testphase zur Verbesserungszwecken gesammelt werden • Programmfehler und Benutzerverhalten soll evaluiert werden um das Tool anschließend anzupassen • Verbesserungsvorschläge sollen seitens der Zielgruppen „Schüler“ und „Lehrer“ entgegengenommen werden und evtl. implementiert werden • eine finale Anwendungsphase soll Ende Jänner mit verbesserter Software stattfinden

Aufgaben:

Name	Notiz	zu erledigen bis
Vereshchagin	Fertigstellung aller PHP-Skripte der Website. Fertigstellung des HTML Grundgerüsts und CSS.	03.02.2019
Würfel	Fertigstellung des Algorithmus zur Überprüfung.	03.02.2019

	HTBLVA St. Pölten Elektronik und Technische Informatik Ausbildungsschwerpunkt Embedded Systems	Reife- und Diplomprüfung

Betreuungsprotokoll zur Diplomarbeit lfd. Nr.: 4

Themenstellung: Workshop Scheduling System
Kandidaten/Kandidatinnen: Kiril Vereshchagin (5BHELS), Bernhard Würfel (5BHELS)

Jahrgang: 5BHELS 2018/19
Betreuer/in: W. Kuran
Ort: HTBLuVA St. Pölten / Raum: V121
Datum: 09.01.2019
Zeit: 7:50 – 9:00

Besprechungsinhalt:

Name	Notiz
Vereshchagin	• Stand der aktuellen Software und neuester Erweiterungen
Würfel	• Möglichkeit der Einbindung (auslesen) von WebUntis Einträgen durch JSON Schnittstelle
Allgemein	• Verhalten der möglichen Zielgruppe, welche derzeit dieselbe Arbeit ohne Software erledigt ist möglicherweise „problematisch“ • Vorbereitung einer Präsentation

Aufgaben:

Name	Notiz	zu erledigen bis
Allgemein	• aktuellen Projekte der Anwender-Zielgruppe „Schüler“ sollen begutachtet werden, um Nützlichkeit der Software für jene Anwender zu beurteilen • Bereitstellung des Prototyps einer gewählten Testgruppe	16.01.2019
	• „finale“ Software soll der Anwender-Zielgruppe „Lehrer“ präsentiert werden • „finale“ Software soll öffentlich zugänglich gemacht werden und anwendungsbereit sein	25.01.2019

	HTBLVA St. Pölten Elektronik und Technische Informatik Ausbildungsschwerpunkt Embedded Systems	Reife- und Diplomprüfung

Betreuungsprotokoll zur Diplomarbeit lfd. Nr.: 5

Themenstellung: Workshop Scheduling System
Kandidaten/Kandidatinnen: Kiril Vereshchagin (5BHELS), Bernhard Würfel (5BHELS)

Jahrgang: 5BHELS 2018/19
Betreuer/in: W. Kuran
Ort: HTBLuVA St. Pölten / Raum: V121
Datum: 11.02.2019
Zeit: 13:10 – 14:10

Besprechungsinhalt:

Name	Notiz
Vereshchagin	• Anzahl der Skills/Fächer pro Lehrer
Würfel	• Einbindung von Kursen von Lehrern
Allgemein	• Präsentation des aktuellen Systems • Möglichkeiten der Erweiterungen • Feedback der Tests • Fixpunkte für die Diplomarbeit und Handbuch

Aufgaben:

Name	Notiz	zu erledigen bis
Allgemein	• Perfektion des Systems • letzte notwendige Funktionalitäten, wie das Benoten von Projekten wurden eingebaut • erster Entwurf der Dokumentation wurde zur Verfügung gestellt	04.03.2019

H. Anhang

H.1. Benutzerhandbuch

H.1.1. Allgemeines und Funktionsübersicht

Im Folgenden werden alle Funktionen des WSS nach Nutzergruppen durch Text und Screenshots des Systems beschrieben. Zur besseren Übersicht werden zuerst die Hauptfunktionen und danach alle Nebenfunktionen jeder Seite beschrieben.

Damit einzelne Funktionen schneller gefunden werden, sind alle in folgender Tabelle alphabetisch mit Kapitel- und Seitenzahl angeführt:

H.1.2. Information

Damit das System erwartungsgemäß funktioniert, empfehlen wir eine Nutzung nach diesem Benutzerhandbuch und ohne Verwendung von browserspezifischen Funktionen zur Manipulation der Webseite.
Allgemein ist das WSS gegen jede Form des Missbrauchs geschützt.

H.1.3. Allgemeine Begriffe

In diesem Unterabschnitt werden jene Begriffe beschrieben, die systemspezifisch sind und durch Außenstehende falsch interpretiert werden können. Damit Sie von Anfang an die im Handbuch verwendeten Bezeichnungen richtig auslegen, empfehlen wir, die Erklärung sorgfältig durchzulesen.

Courses

Courses stellen den Hauptbestandteil des WSS dar. Damit sich ein Schüler für einen Zeitbereich (bspw. einen Halbtag) in einem Raum mit einer Lehrperson eintragen kann, muss dieser einen *Course* - im Folgenden auch als *Kurs* bezeichnet - erstellen. Somit werden dem Lehrer alle Schüler angezeigt, die in einem Zeitbereich in seinem Raum anwesend sein sollen. Dabei stellt dieser vor allem eine Ansprechperson für Probleme, Fragen bezüglich Technologien und Umsetzungsmöglichkeiten und eine Aufsichtsperson für die anwesenden Schüler dar, die evtl. in diesem Raum Maschinen bedienen, die eine Aufsicht oder Einweisung benötigen.

Da sich Schüler aus eigenem Verlangen diesem Lehrer bzw. Raum zuteilen, kann ein Lehrer selbst keine *Courses* erstellen, um einzelne Schüler zu sich einzuteilen. Die einzige Möglichkeit zur Regulierung der eingeteilten Schüler ist das Löschen einzelner *Courses*, womit ein Schüler zum ausgewählten Zeitraum aus dem Raum entfernt wird.

Courses stellen also keine Theoriestunden dar, in denen eine Lehrperson seinen vorgesehenen Lehrstoff vermittelt. Dafür sind im WSS *Lectures* vorgesehen, die im übernächsten Punkt behandelt werden.

Fixed Courses

Ein *Fixed Course* ist ein Attribut für einen *Course*. Dieses verhindert das Ersetzen eines Kurses durch einen anderen Schüler, der ebenso zu diesem Zeitbereich in diesen Raum möchte. Dieser Fall tritt ein, wenn alle Plätze in einer Räumlichkeit besetzt sind.

Grundsätzlich können einzelne Kurse keine anderen ersetzen. Damit jedoch ein Schüler garantiert in einem Halbjahr (oder jeder beliebigen größeren Zeitspanne) jeden Raum zumindest einmal besuchen kann - die Anzahl wird von den Admins bestimmt -, gibt es im WSS *Fixed Courses*. Auf diese Weise kann ein Schüler in einem ausgelasteten Raum einen Kurs - vorausgesetzt dieser besitzt nicht ebenfalls das Attribut *Fixed Course* - mit seinem eigenen ersetzen. Dieser Kurs kann nicht von anderen *Fixed Courses* ersetzt und somit nur von einem Lehrer oder Admin entfernt werden.

Lectures

Lectures bezeichnen im WSS Theoriestunden. Diese können von Lehrern erstellt werden, wobei er alle Schüler auswählt, die daran teilnehmen müssen. Einem Schüler steht es daher nicht frei, einer *Lecture* fernzubleiben. Ebenso können sich einzelne Schüler nicht selbst in einen vorhandenen Theoriekurs einteilen.

Diese strikten Vorgaben sollen ermöglichen, dass ein Lehrer den vorgesehenen Lehrstoff an genau jene Schüler vermitteln kann, von denen er vermutet, dass diese ihn benötigen bzw. interessiert aufnehmen. Damit *Lectures* gewissenhaft geplant werden, muss der Lehrer bei der Erstellung jener eine Beschreibung, das Datum des Stattfindens und die betroffenen Schüler festlegen. Optional kann er die involvierten Themengebiete angeben.

Da Schüler sofort eingeteilt werden und sich nicht aus einer *Lecture* austragen können, muss ein Admin diese nach dem Einreichen eines Lehrers akzeptieren, bevor diese in den Plan übernommen wird. Bis zum Akzeptieren steht es dem Ersteller frei, diese weiterhin zu bearbeiten. Nachdem eine *Lecture* in den Zeitplan übernommen wurde, kann sie jedoch nicht mehr verändert werden.

Projects

Das WSS soll vor allem in einer Unterrichtsumgebung verwendet werden, in der Schüler primär an eigenen Projekten arbeiten, deren Ideen von diesen stammen und deren Umsetzung auch von diesen verantwortet wird. Daher steht unabhängig von *Courses* und *Lectures* die Erstellung und Verwaltung von *Projects* - im Folgenden auch Projekte genannt - im Mittelpunkt des Systems.

Projekte werden von einem Schüler initiiert, der sich vorab mit anderen an dieser Idee interessierten Schülern abgesprochen hat. Zusätzlich benötigt er für die Umsetzung des Projekts einen betreuenden Lehrer, der bei der Umsetzung zur Seite steht und nach Abgabe des Projekts auch für die Benotung dieser

verantwortlich ist. Sollten alle grundlegenden Punkte geklärt sein, legt der verantwortliche Schüler (als *Team Leader* bezeichnet) ein *Project* an. Bei dieser gibt er die involvierten Schüler und den betreuenden Lehrer, gemeinsam mit Milestones und einer Beschreibung des Projekts, ein.

Die Laufzeit eines Projekts beträgt typischerweise einige Wochen bis wenige Monate. Dieses hat systemintern keine Auswirkung auf die Erstellung von *Courses* oder *Lectures*; es gibt einem Schülerteam jedoch einen Leitfaden für Tätigkeiten in *Courses*.

Ein Schüler kann parallel nur ein Projekt haben, damit er sich vollständig auf dieses konzentriert. Währenddessen kann ein Lehrer mehrere *Projects* betreuen und damit benoten. Dieser Aufbau wurde an das bestehende Projektverfahren der HTBLuVA St. Pölten angelehnt.

Dates

Dates bezeichnen im WSS einzelne Zeitbereiche, die mit dem Beginnzeitpunkt gekennzeichnet werden. Zu diesen, die von einem Admin festgelegt werden, können *Courses*, *Lectures* etc. angelegt werden.

Access Codes und Registrierung

Grundsätzlich können sich nur Schüler registrieren - alle anderen Konten werden von der jeweils höheren Instanz oder darüber erstellt.

Damit Schüler ihre Registrierung durchführen können, benötigen sie einen achtstelligen Code, den *Access-Code*. Dieser wird von einem Lehrer oder Admin vergeben, die diese in einer Übersicht auslesen bzw. von dieser exportieren können.

H.1.4. Allgemeine Funktionen

Menüleiste

Über die Menüleiste kann die seitliche Leiste minimiert, das Nachichten-Menü und das Profil-Menü geöffnet werden.

Abbildung H.1.: Menüleiste

Nachrichten-Menü

Mit einem Klick auf das Nachrichten-Symbol (orange Umrandung in H.1) wird das Nachrichten-Menü aufgerufen. In diesem werden alle ungelesenen Nachrichten angezeigt. Möchte man den vollen Inhalt anzeigen, so klickt man auf diese (schwarzer Rahmen in H.2). Um sie ohne Öffnen als gelesen zu markieren, klicken Sie auf den blauen Button *Mark as read* (orange Umrandung in H.2). Die Übersicht aller Nachrichten wird aufgerufen, wenn man auf *View/Send Messages* klickt (grüner Rahmen in H.2).

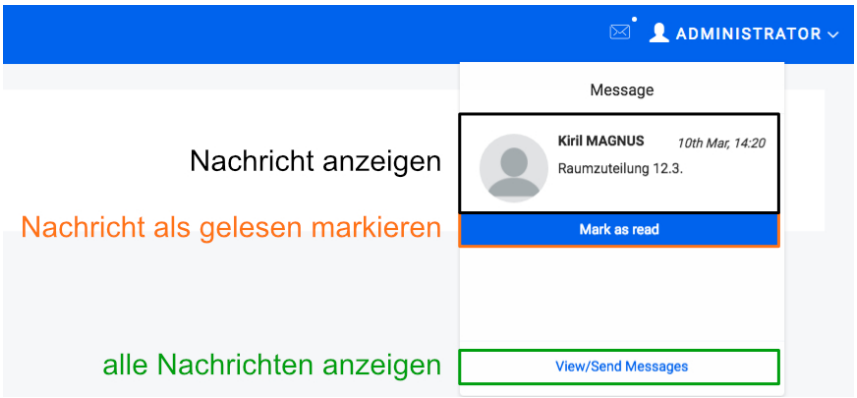


Abbildung H.2.: Nachrichten-Menü

Profil-Menü

Das Profil-Menü wird mit einem Klick auf den Benutzernamen aufgerufen (grüne Umrandung in H.1). Über *My Account* wird das Dashboard angezeigt

(schwarze Umrandung in H.3), mit einem Klick auf *My Profile* wird man auf das eigene Profil weitergeleitet (orange Umrandung) und das Abmelden aus dem WSS erfolgt durch einen Klick auf *Log Out* (durch grünen Rahmen gekennzeichnet).

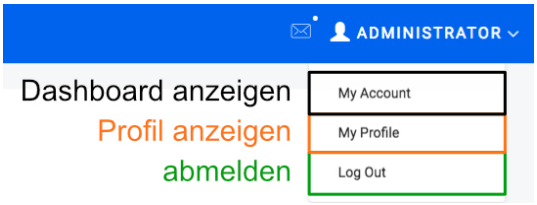


Abbildung H.3.: Profil-Menü

Seitenleiste

Über die Seitenleiste kann zu jeder Seite des WSS navigiert werden. Dabei ist die Anzahl an Optionen an die Rechte des Nutzers angepasst. Wird die Seitenleiste minimiert (orange Umrandung in H.4), so sind die Punkte des bisher geöffneten Menüs auf der rechten Seite weiterhin angeführt.

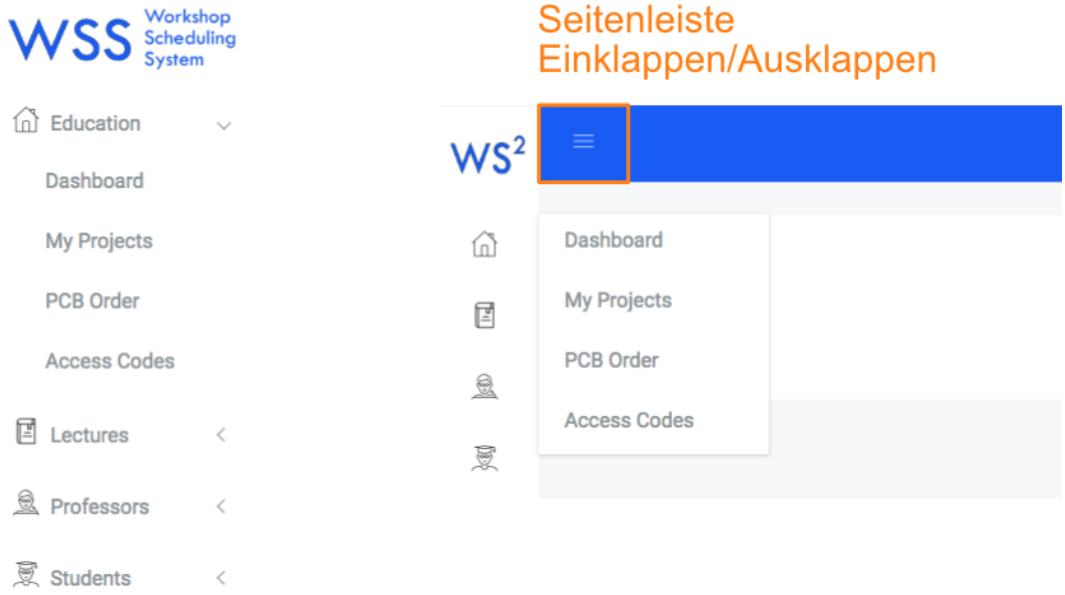


Abbildung H.4.: Seitenleiste maximiert (links) und minimiert (rechts)

Dashboard

Das Dashboard ist der zentrale Ausgangspunkt des WSS und beinhaltet die wichtigsten Funktionen für jeden Nutzer. Wie die Seitenleiste ist auch das Dashboard an jede Benutzerebene angepasst - in den folgenden Punkten werden die Funktionen und der Aufbau für jede erklärt.

Student (Schüler)

Beim Schüler teilt sich das Dashboard in zwei Teile: dem Bereich zum Erstellen eines *Courses* (in H.5 orange markiert) und der Übersicht aller Zeitbereiche (grün markiert).

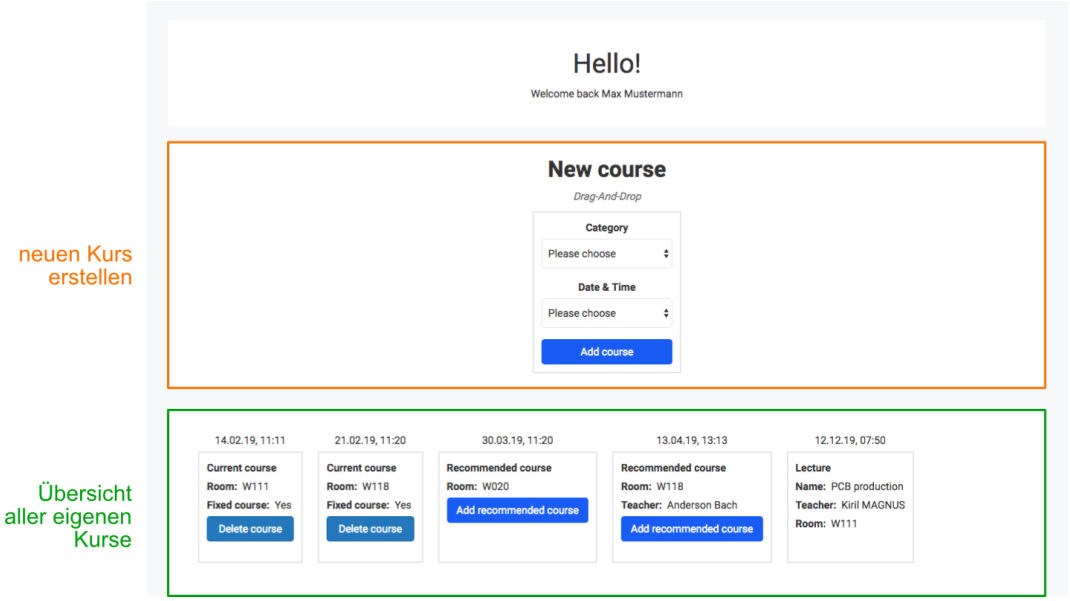


Abbildung H.5.: Dashboard eines Students (Schülers)

Beim Erstellen eines *Courses* gibt es drei Filtermöglichkeiten zur Suche von Räumen oder Lehrern. Wie in H.6 ersichtlich, kann nach Räumen (*Room*), Lehrern (*Teacher*) oder Fähigkeiten (*Skill*) gesucht werden.
Wird der Raum oder Lehrer ausgewählt, so wird der korrespondierende Teil zur Information angezeigt (welcher Lehrer in diesem Raum unterrichtet und vice versa). Wählt man eine Fähigkeit aus, so werden alle Lehrer mit dieser Fähigkeit und deren Räume in einem weiteren Auswahlfeld angezeigt und eine dieser Optionen steht zur Auswahl.

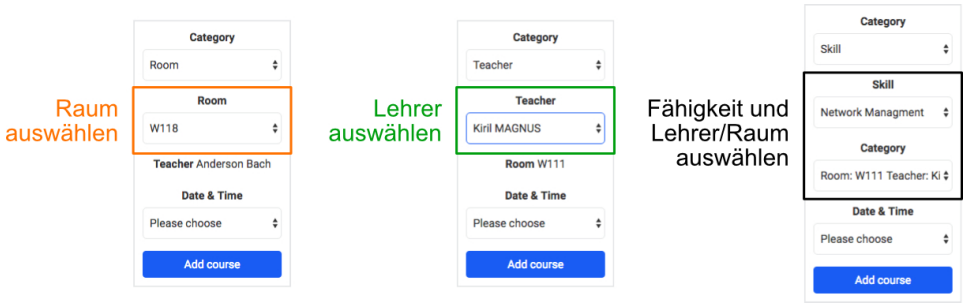


Abbildung H.6.: Filtermöglichkeiten zur Einteilung eines Kurses

Um den Zeitbereich für den *Course* auszuwählen, kann man das gesamte Kästchen per Drag&Drop in einen der in der Übersicht angezeigten Zeitbereiche ziehen und dort fallen lassen (vgl. grüne Umrandung in H.7). Dabei werden alle auswählbaren Zeitbereiche durch eine blau-leuchtende Umrandung gekennzeichnet.

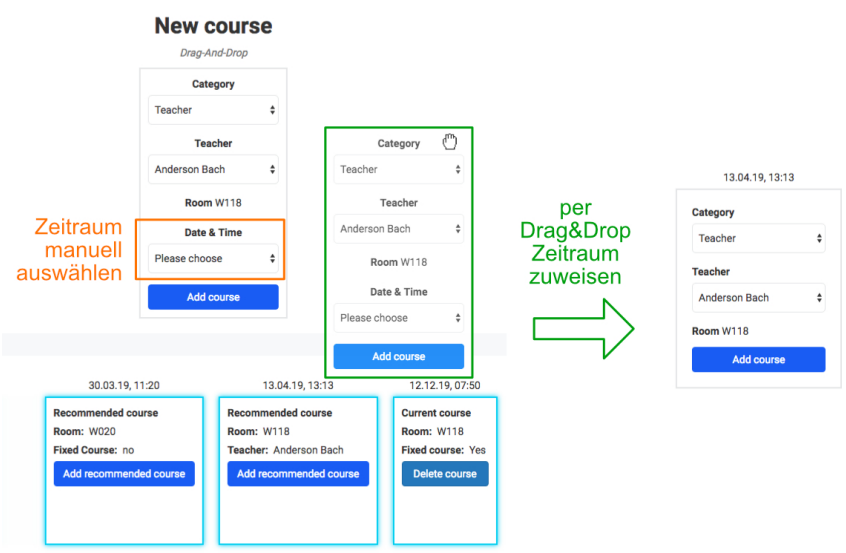


Abbildung H.7.: Course einen Zeitbereich zuweisen

Sollte man Drag&Drop nicht verwenden wollen, kann der Zeitraum auch manuell ausgewählt werden (in orange gekennzeichnet). In beiden Fällen muss zum vollständigen Eintragen des *Courses* der blaue Button mit der Beschriftung *Add course* gedrückt werden.
Wurde zu diesem Zeitpunkt schon ein Kurs eingetragen, so wird dieser ersetzt. Bei Drag&Drop wird dies zusätzlich explizit durch den Button *Replace course* angegeben (orange Umrandung in H.8).

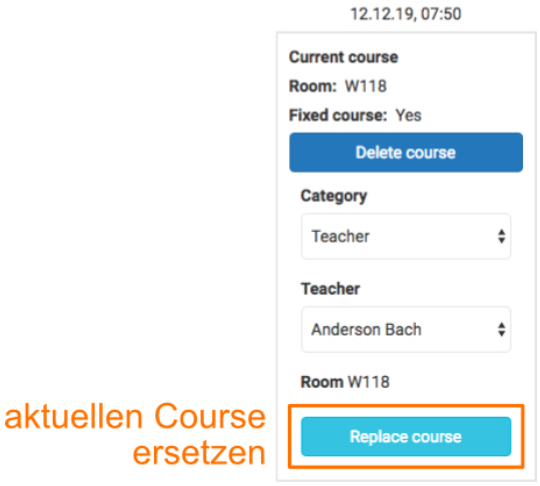


Abbildung H.8.: Course bei Drag&Drop ersetzen

In der Übersicht aller eigenen Kurse werden für jeden Zeitbereich die bereits eingetragenen *Courses*, Vorschläge für diese und *Lectures* angezeigt. Bereits eingetragene Kurse können mit einem Klick auf den Button *Delete course* wieder entfernt werden, sofern diese noch nicht stattgefunden haben (orange Umrandung in H.9).
Recommended Courses werden durch einen Algorithmus des WSS berechnet und bieten eine komfortable Möglichkeit, den besten Kurs - dieser ist von der Kapazität der Räume, deren Belegung und den bisherigen besuchten Lehrern abhängig - sofort festzulegen (grüne Umrandung).

Theoriestunden, die von einem Lehrer festgelegt wurden, werden *Lectures* genannt und können nicht selbstständig eingetragen oder entfernt werden. In diesen Theoriestunden, die nur für einzelne Schüler festgelegt werden, trägt eine Lehrperson lehrplanrelevante Themen vor. Sollte man selbst eine bestimmte *Lecture* besuchen wollen, so kann man vom Vortragenden zu dieser hinzugefügt werden.
Da eine eingetragene Theoriestunde nicht gelöscht werden kann, können für diesen Zeitpunkt auch keine Kurse erstellt werden. Sobald eine *Lecture* eingetragen wird, werden für diesen Zeitraum bereits eingetragene *Courses* gelöscht.

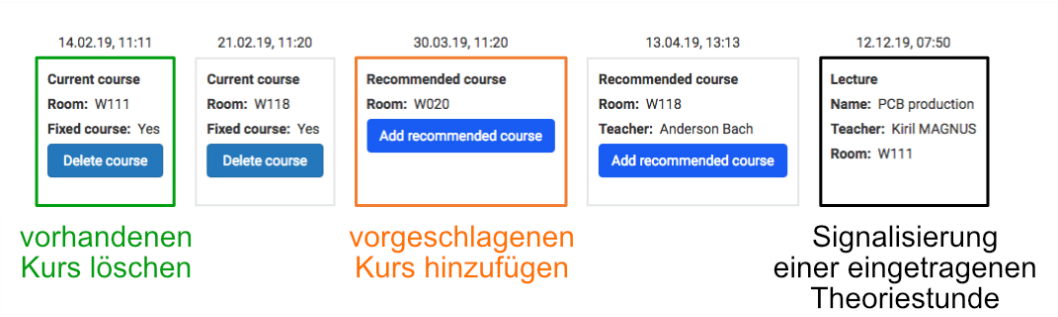


Abbildung H.9.: Übersicht aller eigenen Kurse

Teacher (Lehrer)

Das Dashboard eines Lehrers teilt sich in zwei Bereiche - den Aktionsbereich zum Eintragen des eigenen Fernbleibens und der Übersicht aller eigenen Kurse pro Zeitbereich.

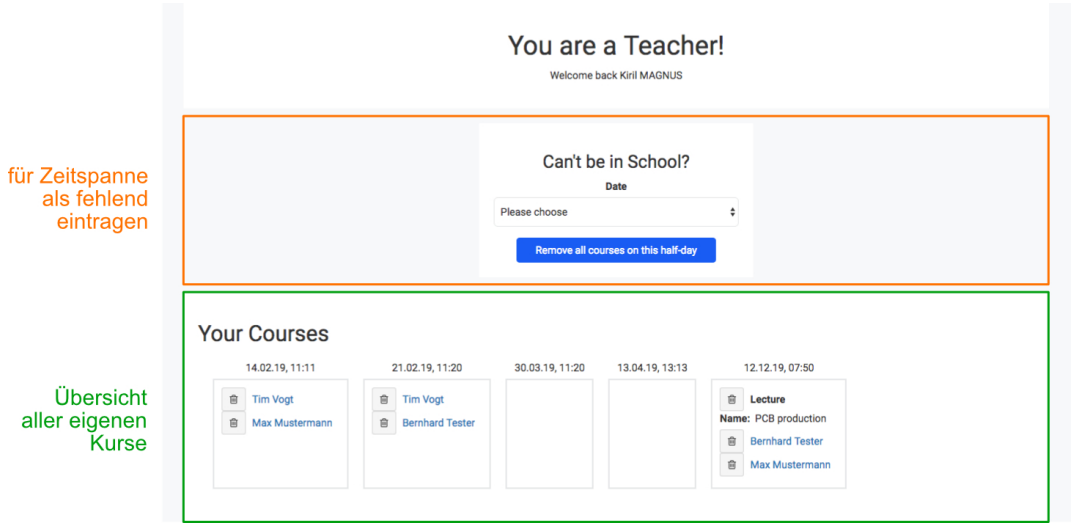


Abbildung H.10.: Dashboard eines Teachers (Lehrers)

Sollte man als Lehrer an einem Tag kurzfristig nicht in die Schule kommen - bspw. weil man erkrankt oder durch zu hohes Verkehrsaufkommen nicht rechtzeitig den Unterrichtsraum erreicht - so kann man alle eingetragenen Kurse für diesen Zeitbereich entfernen lassen.
Dafür wählt man im Drop-Down-Menü innerhalb des *Can't be in School?*-Fensters (in H.10 orange markiert) den Zeitbereich aus und bestätigt diesen mit einem Klick auf den blauen Button mit der Beschriftung *Remove all courses on this half-day.*

Der zweite Bereich des Lehrer-Dashboards zeigt alle eingetragenen Schüler pro Zeitbereich an. Diese *Courses* werden von den Schülern selbst erstellt und werden nicht vom Lehrer bestimmt (genauere Erklärung siehe H.1.3). Sollten Sie als Lehrer einzelne Schüler zu einem Zeitbereich aus dem Raum entfernen wollen, können Sie das Löschen mit einem Klick auf das Müllcontainer-Icon durchführen (orange Umrandung in H.11). Durch einen Klick auf den Namen werden Sie direkt auf das Profil des jeweiligen Schülers geleitet (grün markiert).

Abbildung H.11.: Löschen von Courses und Lectures

Da *Lectures* (genaue Erklärung siehe H.1.3) von einem Lehrer erstellt werden, können diese jederzeit durch einen Klick auf das Müllcontainer-Icon (schwarze Umrandung in H.11) gelöscht werden. Danach können diese jedoch nicht wiederhergestellt werden. Um eine Lecture vorübergehend zu deaktivieren, bspw. um Schüler hinzuzufügen oder andere Details zu bearbeiten, bitten Sie einen Admin um die Rückziehung der Bewilligung. Dadurch wird sie ohne dem Entfernen aus dem System dem Plan entzogen und steht dem Ersteller wieder frei zur Gestaltung.

Um einen einzelnen Schüler aus einem Theoriekurs zu entfernen, drücken Sie das Müllcontainer-Icon neben dem Namen des jeweiligen Schülers (rote Umrandung).

Admin (Administrator)

Im Gegensatz zum Daåshboard eines Schülers oder Lehrers teilt sich jenes eines Admins in drei Bereiche auf: dem Auswählen des angezeigten Zeitbereichs der Übersicht, der Übersicht selbst und einem Aktionsbereich zum Ändern der Anzahl der *Fixed Courses*.

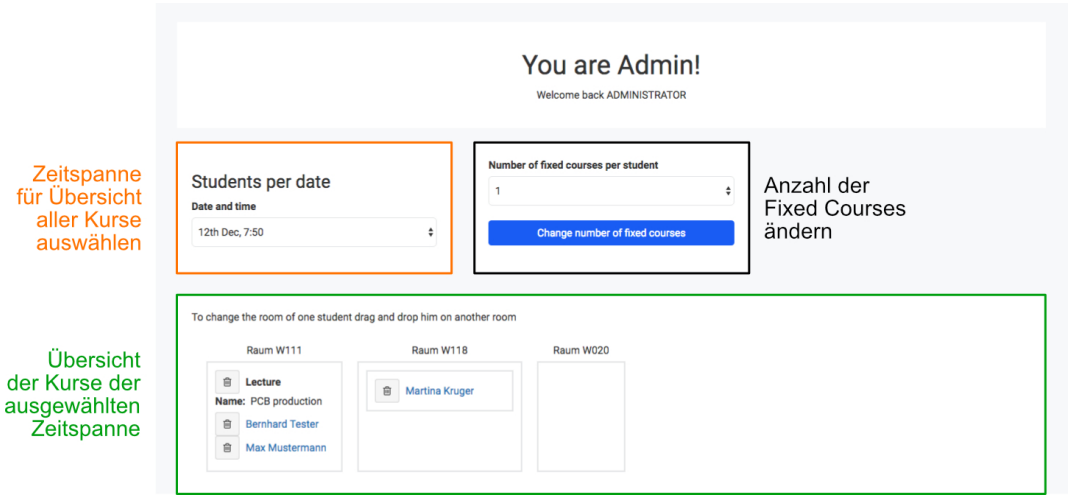


Abbildung H.12.: Dashboard eines Admins (Administrators)

Die Bedeutung von *Fixed Courses* wurde bereits in H.1.3 erläutert. Die Anzahl dieser kann von Ihnen als Admin verändert werden. Dabei bezieht sich diese auf die Anzahl der *Fixed Courses*, die ein Schüler pro Lehrer/Raum hat. Im Dropdown-Menü unter *Number of fixed courses per student* (schwarze Umrandung in H.12) kann diese zwischen 0 und 5 ausgewählt werden und durch einen Klick auf den blauen Button mit der Beschriftung *Change number of fixed courses* bestätigt werden.

Im unteren Bereich das Dashboards werden alle *Courses* und *Lectures* in den jeweiligen Räumen zum ausgewählten Zeitbereich angezeigt. Dieser wird durch den ausgewählten Wert des Dropdown-Menüs unterhalb *Students per Date* ausgewählt (orange Umrandung).
Wie bei Lehrern können einzelne Schüler und Lectures durch einen Klick auf das Müllcontainer-Icon neben dem jeweiligen Eintrag entfernt werden. Ebenso kann das Profil eines einzelnen Schülers durch einen Klick auf dessen Namen aufgerufen werden.

Eine Admin-spezifische Funktion ist das Verschieben von Schülern in andere Räume innerhalb des ausgewählten Zeitbereichs. Durch Drag&Drop kann ein Eintrag durch Klicken und Halten der linken Maustaste ausgewählt und in den gewünschten Raum verschoben werden. Dazu bewegen Sie den ausgewählten Schüler mit der Maus über einen Raum und lassen die linke Maustaste los. Nun wird die Seite neu geladen und jener *Course* wurde erfolgreich verschoben (vgl. H.13). Der jeweilige Schüler wird über die Änderung benachrichtigt.

Abbildung H.13.: Schüler in andere Räume verschieben

PCB-Order

Eine PCB-/Leiterplattenbestellung kann von jedem Benutzer abgegeben werden. Diese werden von einem berechtigten Lehrer, der für die Leiterplattenherstellung verantwortlich ist, oder einem Admin verwaltet.

Um zur Bestellseite zu gelangen, öffnet man in der Seitenleiste das *Education*-Menü und klickt darin auf den Punkt *PCB Order*.
Im Folgenden sind jene auszufüllenden Felder beschrieben, die nicht vollständig selbsterklärend sind.

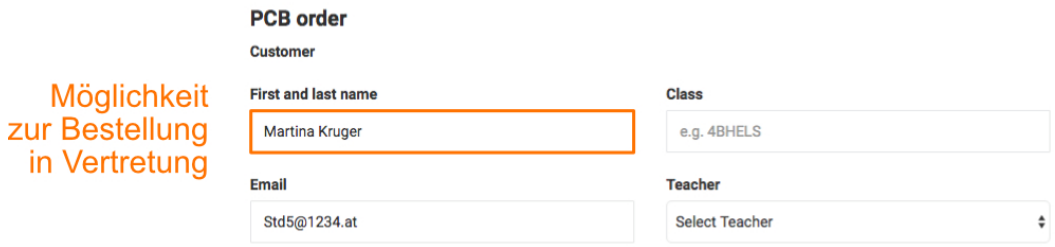


Abbildung H.14.: Bestellungen in Vertretung abgeben

In H.14 ist ersichtlich, dass Bestellungen auch für andere Personen abgegeben werden können. Hierfür wird der Name der Person, für die die Bestellung abgegeben wird, in das Feld *First and last name* (orange umrandet) eingetragen. Ebenso wird die Klasse dieser Person angegeben.

Die E-Mail-Adresse sollte nur geändert werden, wenn man nach der Bestellung nicht mehr selbst über etwaige Fehler informiert werden möchte, sondern die Person, für die der Print gefertigt wird. Der verwaltende Lehrer kann sich jedoch trotzdem dafür entscheiden, den Besteller über die systeminterne Nachrichtenfunktion zu informieren, der in diesem Fall Sie selbst sind. Ebenso werden die Nachrichten über eine Statusänderung (*in Progress*, *Finished*) an Sie geschickt.

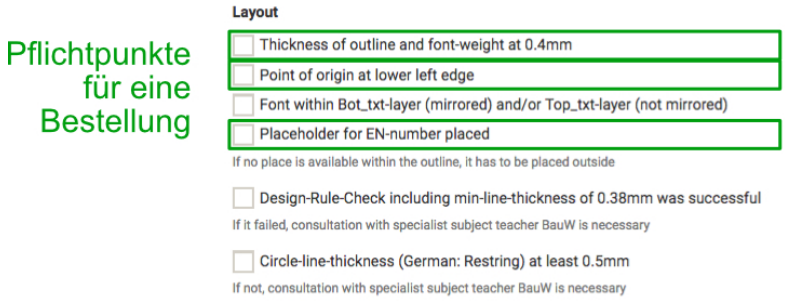


Abbildung H.15.: Pflichtfelder für die Bestellung bezgl. des Layouts

Unter dem Punkt *Layout* müssen drei Punkte verpflichtend aktiviert werden. Diese sind in H.15 grün umrandet. Diese Information wird auf der Bestellseite nicht angegeben, da sie so evtl. ohne besondere Beachtung vom Nutzer aktiviert werden, obwohl dieser sie beim Layout-Design nicht beachtet hat. Selbstverständlich wird der Nutzer bei der Abgabe darauf hingewiesen, dass diese Pflichtfelder aktiviert werden müssen.

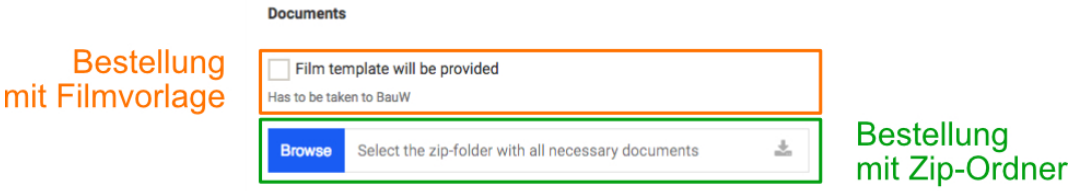


Abbildung H.16.: Möglichkeiten der Bereitstellung der PCB-Dateien

Sie haben zwei Möglichkeiten zur Abgabe der notwendigen Dateien zur Leiterplattenherstellung. Einerseits können Sie selbst dem verantwortlichen Lehrer der PCB-Produktion eine Filmvorlage bringen - dafür aktivieren Sie die Option *Film template will be provided* (in H.16 orange umrandet) - oder Sie lassen diese Option deaktiviert und laden einen ZIP-Ordner mit den notwendigen Dateien hoch. Diese wählen Sie mit einem Klick auf den Button *Browse* (grün umrandet) aus.

Die notwendigen Dateien, die in der ZIP-Datei enthalten sein müssen, werden unter dem Link *lp-fertigung-htlstp.jimdo.com* aufgeführt, der auch im unteren Bereiche der Bestellseite angeführt ist.

Benachrichtigungen über Statusänderungen

Bei einer Statusänderung wird eine Benachrichtigung an den Besteller gesendet, in der die Nummer der Bestellung und der neue Status angegeben werden.

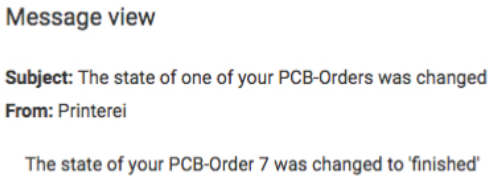


Abbildung H.17.: Benachrichtigung bei Statusänderung einer PCB-Bestellung

Übersicht aller Räume

Eine Übersicht aller Räume steht jedem Benutzer offen, woraus dieser ablesen kann, wie viele Plätze ein Raum bietet und welcher Lehrer in diesem Raum unterrichtet. Durch die Kapazität kann ein *Student* abschätzen, wie früh er sich für diesen Raum einen *Course* erstellen muss, damit dieser nicht ausgebucht ist.

Um zur Raumübersicht zu gelangen, öffnet man in der Seitenleiste das *Lectures*-Menü und klickt darin auf den Punkt *All Rooms*.

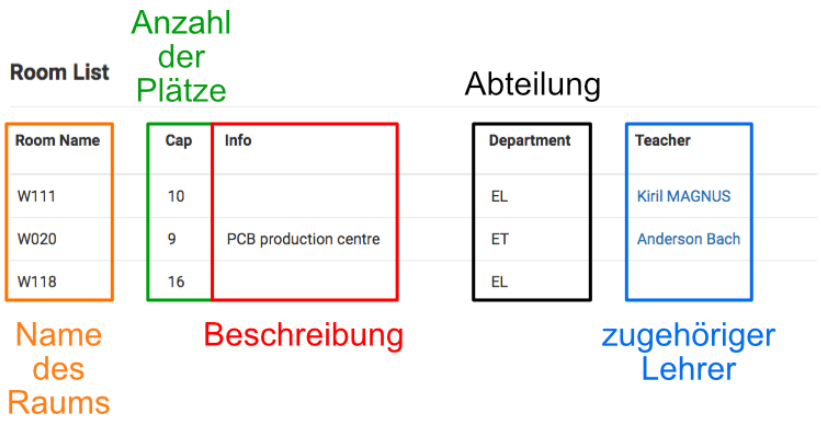


Abbildung H.18.: Übersicht aller Räume

Übersicht aller Professoren

Wie die Übersicht aller Räume kann auch die Übersicht aller Professoren von jedem Benutzer aufgerufen werden. Diese Ansicht führt für jeden Lehrer die Abkürzung, die Abteilung, den zugeteilten Raum, sowie die Möglichkeit, das Profil des Lehrers aufzurufen und diesem eine Nachricht zu senden, auf.

Um zur Lehrerübersicht zu gelangen, öffnet man in der Seitenleiste das *Professors*-Menü und klickt darin auf den Punkt *All Professors*.

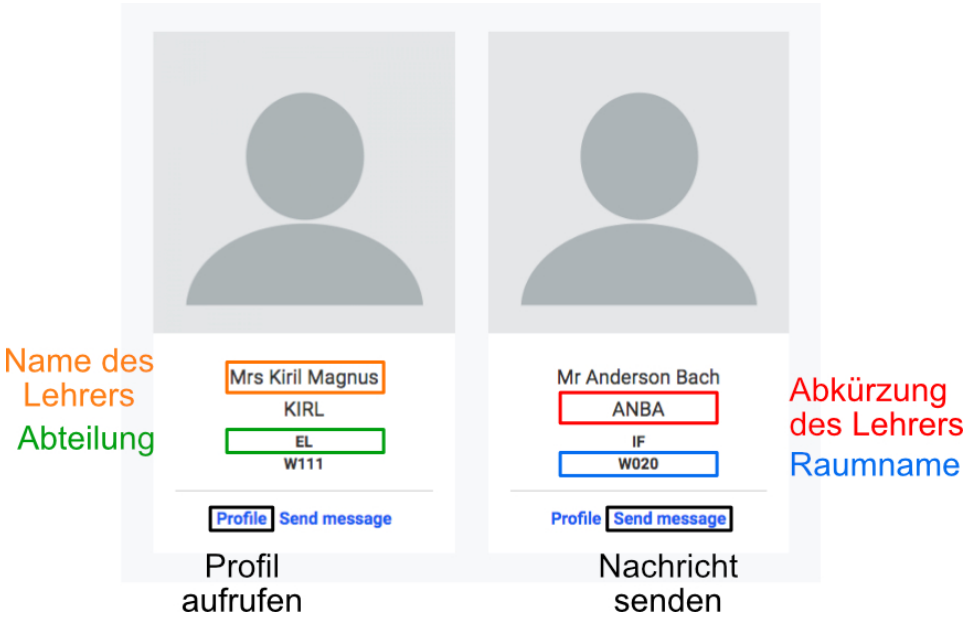


Abbildung H.19.: Übersicht aller Professoren

Übersicht aller Schüler

Um eine Übersicht aller registrierten Schüler zu bekommen, öffnet man in der Seitenleiste das *Students*-Menü und klickt darin auf den Punkt *All Students*.

Im Auswahlmenü unter *Selected course* (orange Umrandung in H.20) kann ein Raum ausgewählt werden, und es werden alle Schüler aufgelistet, die in diesem Raum einen *Course* erstellt haben. Um alle Schüler anzuzeigen, wählt man *All students* aus, das standardmäßig aktiv ist.

Unterhalb werden nun alle Schüler mit ihrem Profilbild, Namen, der zugehörigen Abteilung, ihrer E-Mail und einem Button zum Senden einer Nachricht an diesen tabellarisch aufgelistet.

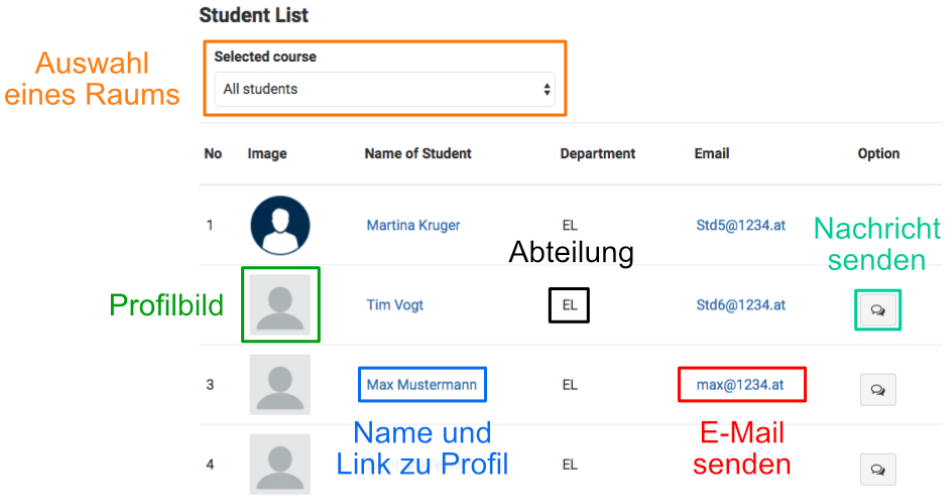


Abbildung H.20.: Übersicht aller Schüler

Profile

In den vorherigen Unterabschnitten *Übersicht aller Professoren* und *Übersicht aller Schüler* wurde erklärt, wie Sie auf das Profil eines Lehrers oder Schülers navigieren. Im gesamten WSS besteht die Möglichkeit, das Profil einer jeweiligen Person aufzurufen, wenn deren Name blau markiert ist.

Grundsätzlich gibt es zwei verschiedene Arten, wie das Profil eines anderen Benutzers aufgebaut sein kann - diese werden im Folgenden besprochen. Zusätzlich wird danach auf das Anzeigen und Bearbeiten des eigenen Profils eingegangen, das entweder das eines *Admins*, *Teachers* oder *Students* ist.

Profil eines Lehrers

Das Profil eines Lehrers besteht aus den persönlichen Details und der Möglichkeit, eine Nachricht an diesen zu senden auf der linken und genaueren Beschreibungen auf der rechten Seite. Standardmäßig sind bei jedem Lehrer der Name, die Abteilung und die E-Mail angegeben - alle anderen Informationen sind vom Kontoinhaber optional auszufüllen.

Zu den lehrerspezifischen Angaben zählen die *Skills* (dt. Fähigkeiten), *Subjects* (dt. unterrichtende Schulfächer) und der *Room* (dt. zugeteilter Raum) mit dessen Beschreibung.

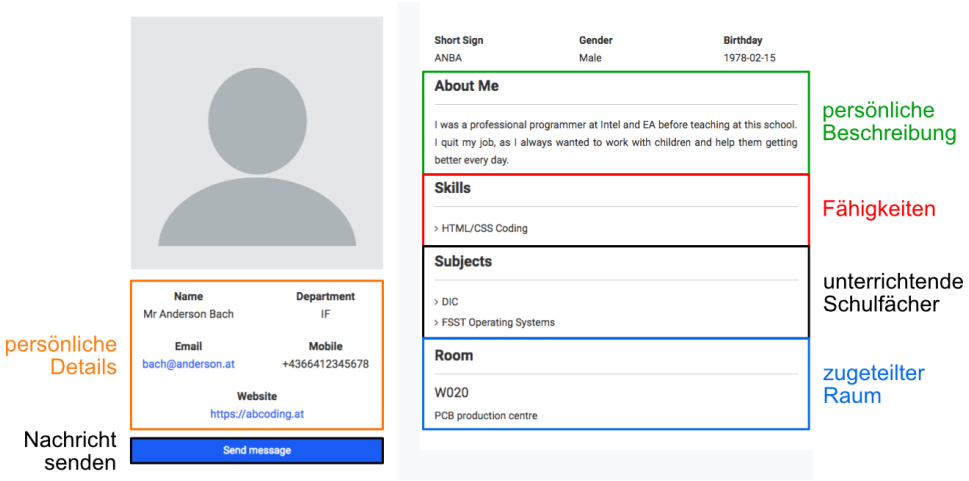


Abbildung H.21.: Profil eines Lehrers

Profil eines Schülers

Das Profil eines Schülers ist grundsätzlich wie ein Lehrer-Profil aufgebaut. Ein Schüler kann jedoch keine *Skills*, *Subjects* und keinen *Room* haben, weshalb diese nicht auf dem Profil angezeigt werden.

Zu den schülerspezifischen Angaben gehört das *Project* (dt. eigenes Projekt), dessen Name mit einer Verlinkung zu diesem auf der Profilseite aufgeführt wird.

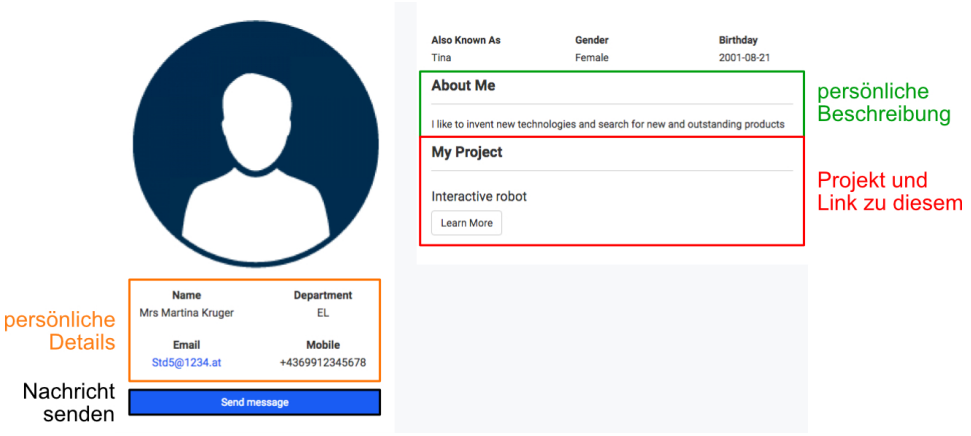


Abbildung H.22.: Profil eines Schülers

Profil bearbeiten

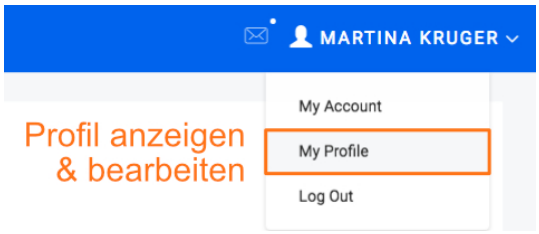


Abbildung H.23.: Profil bearbeiten

Um das eigene Profil aufzurufen, klicken Sie auf den eigenen Namen in der Menüleiste und dann auf den Punkt *My Profile*. Nun können Sie abhängig von der Benutzerebene Details verschiedener Kategorien ändern:

- Student
  - Update Details
- Teacher
  - Update Details
  - Update Technical Info
  - Update Login
- Admin
  - Update Details
  - Update Login

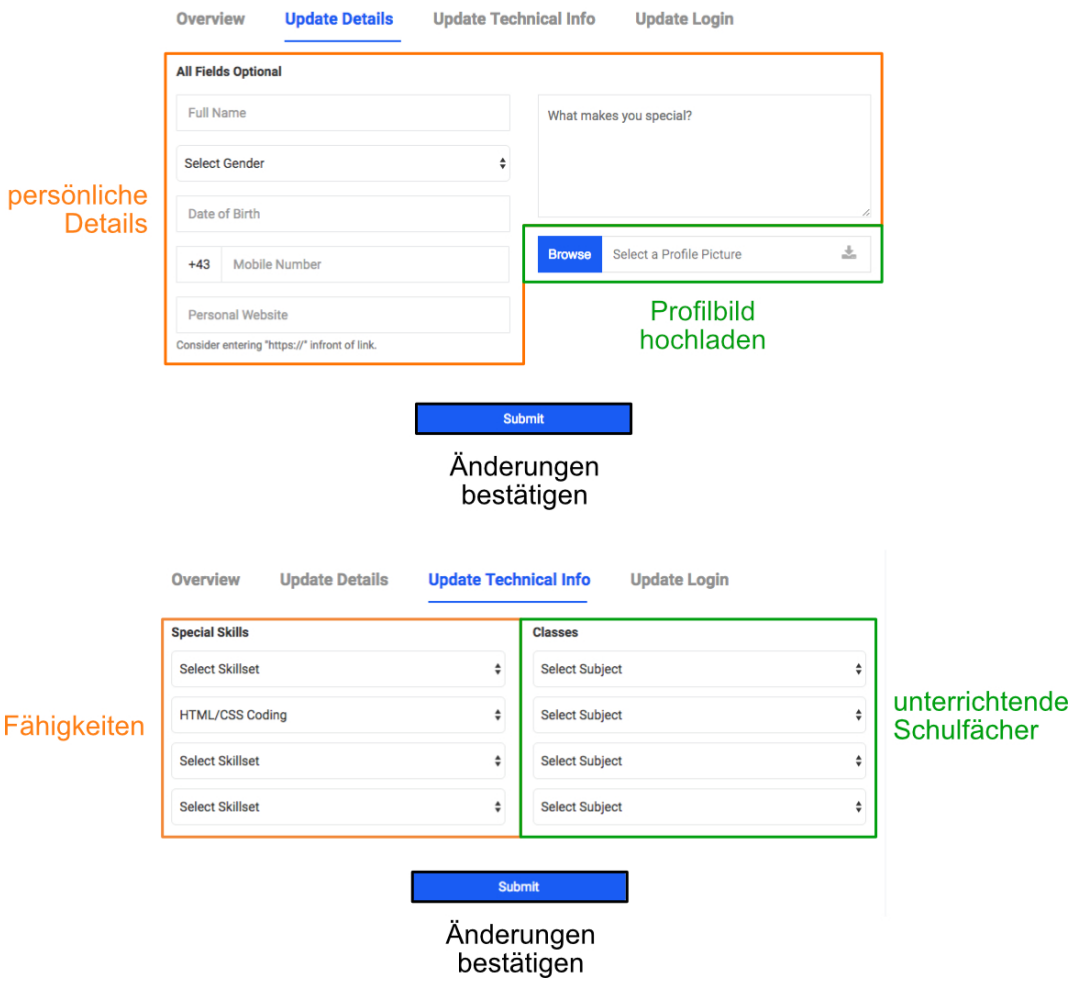


Abbildung H.24.: Update Details (persönliche Details ändern) & Update Technical Info (technische Details ändern)

Das Ändern von persönlichen Details (*Update Details*, obere Grafik in H.24) ist jedem Benutzer vorbehalten. Beim Ersetzen von Informationen werden nur jene verändert, die neu ausgefüllt wurden. Wenn Sie eine persönliche Website angeben, müssen Sie den Link *https://* voranstellen, da dieser sonst von anderen Nutzern nicht direkt aufgerufen werden kann.
Beim Hochladen eines Profilbildes muss beachtet werden, dass dessen Größe zwischen 120px x 120px und 1080px x 1080px liegt und das Dateiformat *JPEG* oder *PNG* ist.

Technische Details können nur von Lehrern geändert werden. Diese können jeweils vier *Special Skills* (dt. Fähigkeiten) und vier *Subects* (unterrichtende Fächer) wählen.

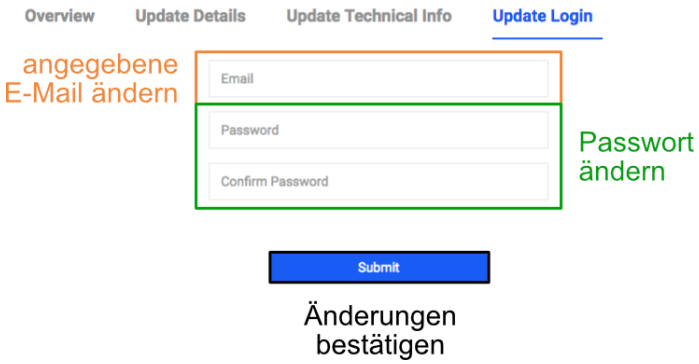


Abbildung H.25.: Update Login (Login-Daten ändern)

Als Lehrer oder Admin können Sie Ihre Login-Daten, also Ihre E-Mail und Ihr Passwort, ändern. Dabei müssen Sie nicht beide gleichzeitig ändern, sondern können dies getrennt machen. Das System ändert bei einem Klick auf den *Submit*-Button automatisch den Wert, den Sie eingegeben haben.

Nachrichten

Um zur Auflistung aller Nachrichten zu gelangen, klicken Sie in der Menüleiste auf das Nachrichten-Symbol und dann auf *View/Send Messages* (in H.26 orange markiert).

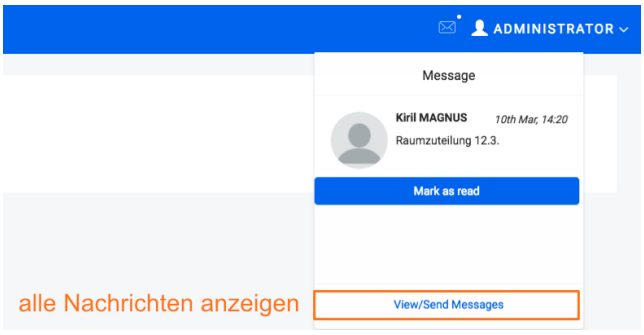


Abbildung H.26.: Aufrufen der Nachrichten

Nun befinden Sie sich auf der Hauptseite der Nachrichtenfunktion. Im linken Teil finden Sie die Möglichkeit der Erstellung einer neuen Nachricht und das Auswählen der Kategorie der angezeigten *Messages*. Im rechten Teil werden Ihnen, gemeinsam mit den Buttons zur Veränderung des Status (in H.27 blau markiert) und dem Laden weiterer Nachrichten (grüne Markierung), die Nachrichten der ausgewählten Kategorie angezeigt.

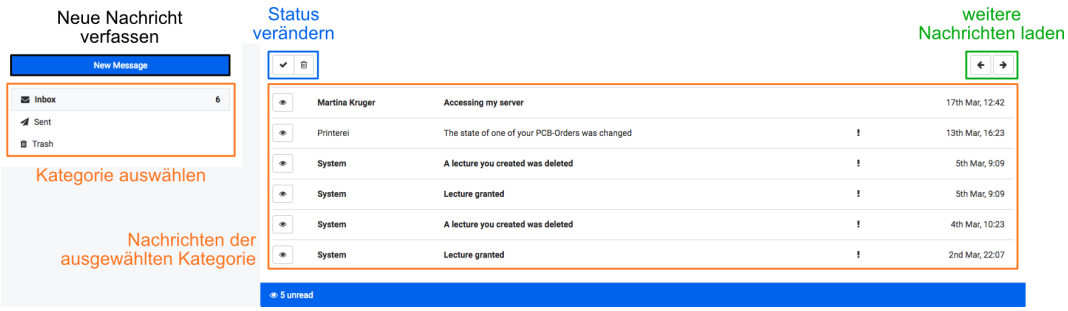


Abbildung H.27.: Hauptseite der Nachrichten-Funktion

Um eine Nachricht zu verfassen, klicken Sie auf den blauen Button mit der Aufschrift *New Message* (in H.27 schwarz umrandet). Die weitere Vorgehensweise finden Sie weiter unten.
Darunter befinden sich die drei Kategorien *Inbox* (dt. Posteingang), *Sent* (dt. gesendete Nachrichten) und *Trash* (dt. gelöschte Nachrichten). Sobald Sie auf eine dieser klicken, werden rechts alle Nachrichten dieser Kategorie angezeigt, wobei jeweils zehn Nachrichten gleichzeitig angezeigt werden, die nach dem Sendezeitpunkt sortiert sind. Um die nächsten zehn Nachrichten anzuzeigen, klicken Sie auf den Pfeil nach rechts (grün markiert). Um wieder die neueren Benachrichtigungen anzuzeigen, klicken Sie auf den Pfeil nach links.

Um eine Nachricht zu öffnen, betätigen Sie den *Augen*-Button neben der jeweiligen Nachricht. Ob eine *Message* noch ungelesen ist, wird durch den fett gedruckten Absender und Betreff signalisiert. Das Rufzeichen links neben dem Sendedatum lässt eine hohe Relevanz der Nachricht erkennen.
Durch Markieren der Nachricht (Linksklick auf diese) und einen Klick auf das *Haken*-Symbol wird die Nachricht als gelesen bzw. ungelesen markiert. Durch das Klicken auf das *Müllcontainer*-Symbol wird eine ausgewählte Nachricht als gelöscht markiert. Ist die Kategorie *Trash* ausgewählt, so wird die ausgewählte Nachricht in *Inbox* verschoben.

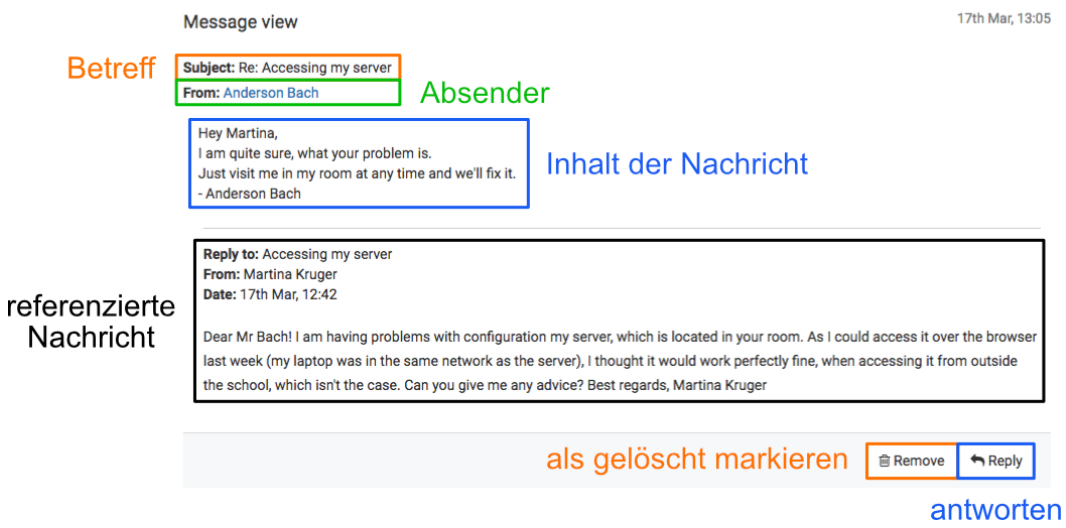


Abbildung H.28.: Geöffnete Nachricht

In der geöffneten Nachricht befinden sich oben der Betreff und Sender der Nachricht, dessen Profil mit einem Klick auf den Namen aufgerufen wird. Darunter wird der eigentliche Inhalt der Nachricht angezeigt.
Sollte die *Message* eine Antwort auf eine andere Nachricht sein, so wird diese im unteren Teil angeführt. Mit einem Klick auf diese öffnet man sie.

Mit den Buttons *Remove* und *Reply* kann eine Nachricht als gelöscht markiert bzw. eine Antwort auf diese verfasst werden.

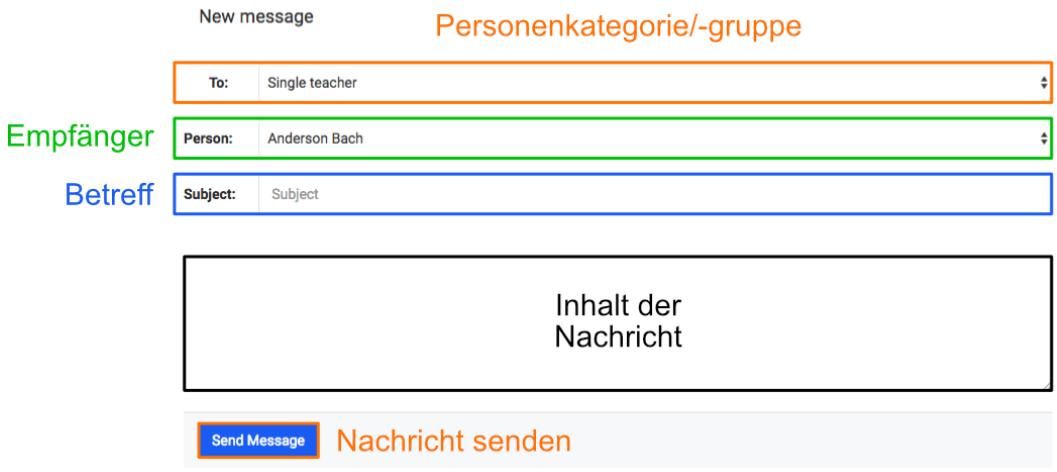


Abbildung H.29.: Nachricht verfassen

Möchte man eine neue Nachricht verfassen, sie klickt man auf der Hauptseite der Nachrichten auf den blauen Button mit der Aufschrift *New Message* bzw. auf ein *Brief*-Symbol oder den Button *Send message* bei einem Benutzer. Im Endeffekt wird dieselbe Seite aufgerufen, wobei bei einer neuen Nachricht alle Felder leer sind, während die Personenkategorie und der Empfänger über den Nachrichten-Button bereits ausgefüllt sind.

Bei *To:* (orange Umrandung in H.29) können je nach Benutzer folgende Personenkategorien/-gruppen ausgewählt werden:

- für Schüler
  - Single admin (einzelner Admin)
  - Single teacher (einzelner Lehrer)
  - Single student (einzelner Schüler)
- für Lehrer
  - Single admin (einzelner Admin)
  - Single teacher (einzelner Lehrer)
  - Single student (einzelner Schüler)
  - All admins (alle Admins)
  - All students (alle Schüler)
- für Admins
  - Single admin (einzelner Admin)
  - Single teacher (einzelner Lehrer)
  - Single student (einzelner Schüler)
  - All teachers and students (alle Lehrer und Schüler)
  - All teachers (alle Lehrer)
  - All students (alle Schüler)

Wird die Kategorie *Single admin*, *Single teacher* oder *Single student* ausgewählt, so muss ein Empfänger (grün markiert) ausgewählt werden. Dieses Dropdown-Menü wird bei Gruppennachrichten nicht angezeigt, da hier alle Empfänger schon durch die Auswahl der Kategorie definiert wurden. In jedem Fall muss ein Betreff (blau markiert) eingegeben werden.

Nun wird der Inhalt der Nachricht eingegeben (schwarze Markierung), der maximal 3072 Zeichen umfassen darf. Die Mitteilung wird abgesendet, wenn Sie auf den blauen Button *Send Message* klicken. Sollten Sie die Seite ohne Senden verlassen, geht die Nachricht verloren.

Sollten Sie bei einer bestehenden Nachricht auf *Reply* geklickt haben, werden Sie auf die gleiche Seite weitergeleitet. Hierbei sind die Felder *To:*, *Person:* und *Subject:* (Re: [Betreff der referenzierten Nachricht]) bereits ausgefüllt. Außerdem wird systemintern die Nachricht, auf die geantwortet wird, vermerkt. Sollten Sie während des Verfassens der Antwort dieses Vorhaben abbrechen, müssen Sie zum Erstellen einer Nachricht an einen anderen Benutzer zurück auf *Inbox* gehen und dort erneut auf *New Message* klicken, da sonst die referenzierte Nachricht auch bei einer Änderung der Personengruppe, des Empfängers oder des Betreffs weiterhin verlinkt wird.

H.1.5. Schülerspezifische Funktionen

Projekte

Projekte können von einem Schüler erstellt werden, der diesem bis zu zwei weitere Schüler hinzufügen kann, sofern diese kein bestehendes Projekt haben. Außerdem muss jedes Projekt einen betreuenden Lehrer haben, der dieses nach der Projektfertigstellung und -abgabe benotet. Für die genaue Beschreibung und Begriffserklärung siehe H.1.3.

Um zur Hauptseite der Projekte zu kommen, öffnen Sie als Schüler in der Seitenleiste das Untermenü *Education* und klicken in diesem auf *My Project* (in H.30 orange markiert).

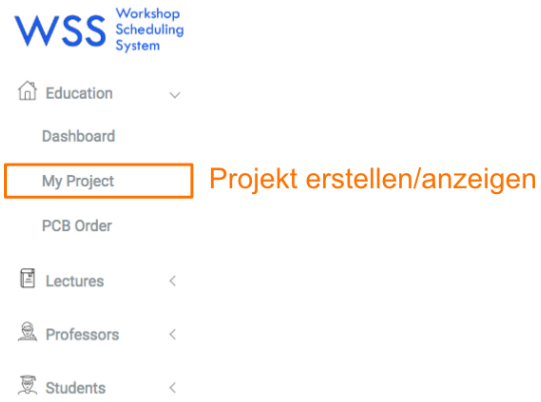


Abbildung H.30.: Aufrufen der Projekt-Hauptseite

Der dargestellte Inhalt auf dieser Seite ist davon abhängig, ob Sie bereits ein Projekt erstellt haben bzw. in ein Projektteam hinzugefügt wurden. Ist dies nicht der Fall, so können Sie über einen Klick auf den blauen Button mit der Aufschrift *Create Project* (in H.31 grün markiert) ein neues Projekt erstellen. Besteht bereits ein Projekt, in dem der Schüler aktiv ist, so wird dieses angezeigt.

Abbildung H.31.: Hauptseite der Projekte - kein Projekt vorhanden (links) & Projekt vorhanden (rechts)

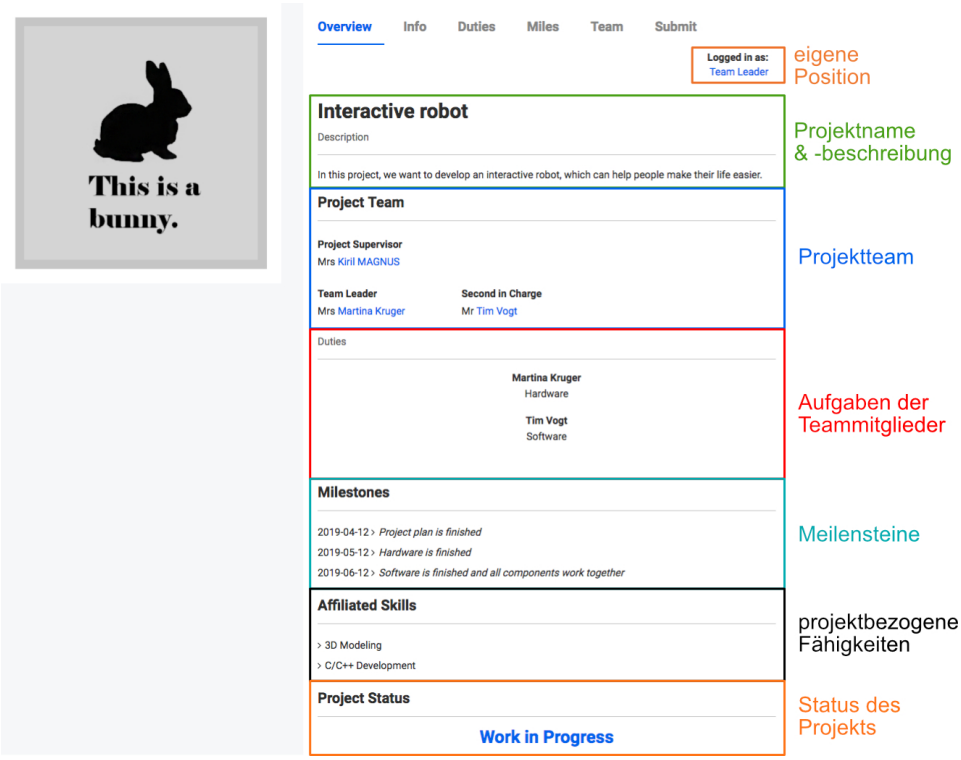


Abbildung H.32.: Hauptseite der Projekte bei vorhandenem Projekt

Bei einem vorhandenen Projekt (vgl. H.32) sind auf dem *Overview*-Tab für jedes Mitglied des Teams alle Informationen ersichtlich. Dazu zählen Ihre eigene Position (rechts oben, orange markiert), der Name und die Beschreibung des Projekts, die Zusammensetzung des Projektteams, die jeweiligen Aufgaben der Teammitglieder, die festgelegten Meilensteine, die projektbezogenen Fähigkeiten und der Status des Projekts.

Sollten Sie noch keinem Projekt zugeordnet sein, können Sie als Schüler durch einen Klick auf *Create Project* (in H.31 grün markiert) ein neues anlegen.
Beim Anlegen eines neuen Projekts müssen Sie zuerst einen betreuenden Lehrer aus dem Dropdown-Menü auswählen (in H.33 blau markiert). Im Gegensatz zu den anderen Teammitgliedern kann dieser mehrere Projekte parallel betreuen, es steht im jedoch frei, das Projekt jederzeit mit Angabe eines Grundes zu verlassen. Weiters sind der Projektname, dessen Ziel und Beschreibung, und die Meilensteine mit Datum auszuwählen. Projektname, -Ziel und -Beschreibung sind dauerhaft veränderbar, während die Meilensteine vom Teamleiter/Ersteller nur ein Mal verändert werden können.

Optional können bei der Erstellung bis zu zwei Teammitglieder, die projektbezogenen Fähigkeiten und eine Projektgrafik ausgewählt werden. Als Teammitglieder können nur andere *Students* ausgewählt werden, die derzeit kein aktives Projekt haben. Außerdem können diese, wie der betreuende Lehrer, jederzeit mit Angabe eines Grundes das Team verlassen und können nicht vom Teamleiter entfernt werden. Die Möglichkeit, weitere Teammitglieder hinzuzufügen und die Fähigkeiten und Projektgrafik zu verändern ist jederzeit vorhanden. Wie bei den Profilbildern dürfen die Abmessungen der Grafik nur zwischen 120px x 120px und 1080px x 1080px liegen und als Dateiformat wird *JPEG* und *PNG* akzeptiert.

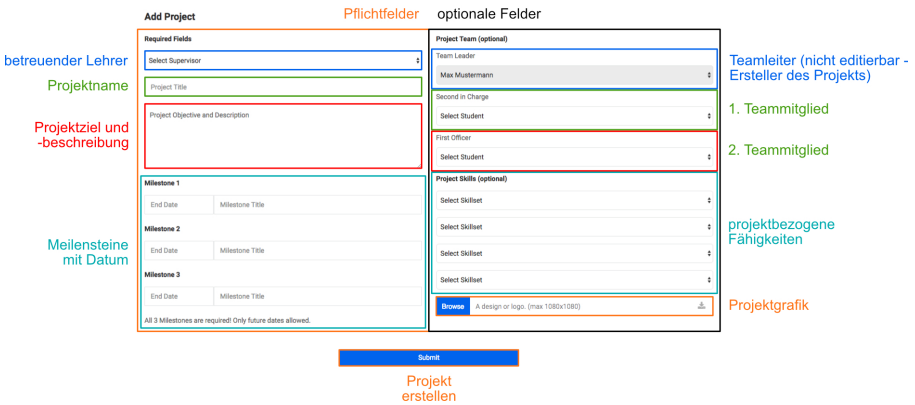


Abbildung H.33.: Neues Projekt erstellen

Grundsätzlich kann jeder Teilnehmer des Projekts bestimmte Teile bearbeiten. Dabei bestimmt die Position im Projekt, welche Informationen geändert werden dürfen. Auf die Änderungsmöglichkeiten des betreuenden Lehrers wird im Unterabschnitt H.1.6 (Lehrerspezifische Funktionen) eingegangen.
Änderungsmöglichkeiten (in Tabs zusammengefasst) nach Teammitglied:

- Teamleiter (*Team Leader*)
  - Info
  - Duties (von allen Teammitgliedern)
  - Milestones (einmalig)
  - Team (neue Teammitglieder hinzufügen)
  - Submit
- 1. Teammitglied (*Second in Charge*)
  - Info (keine Änderung des Projekt-Titels)
  - Duties (der eigenen Person)
  - Leave Project-Team
- 2. Teammitglied (*First Officer*)
  - Duties (der eigenen Person)
  - Leave Project-Team

Info-Tab
Im Info-Tab können der Projekttitel, die projektbezogenen Fähigkeiten, die Projektbeschreibung und die -Grafik geändert werden.

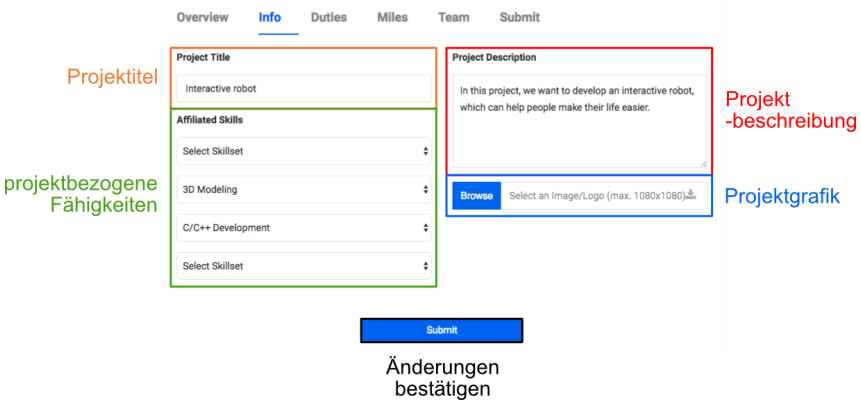


Abbildung H.34.: Änderungen im Info-Tab

Duties-Tab
Die Aufgaben der einzelnen Teammitglieder können im Tab *Duties* geändert werden.

Abbildung H.35.: Änderungen im Duties-Tab

Miles-Tab
Um die einzelnen Meilensteine zu verändern, kann der Projektleiter dies im Tab *Miles* durchführen. Dabei muss beachtet werden, dass diese nur einmal geändert werden können. Sollten sie trotzdem weitere Male eine Änderung benötigen, kann dies vom betreuenden Lehrer gemacht werden.

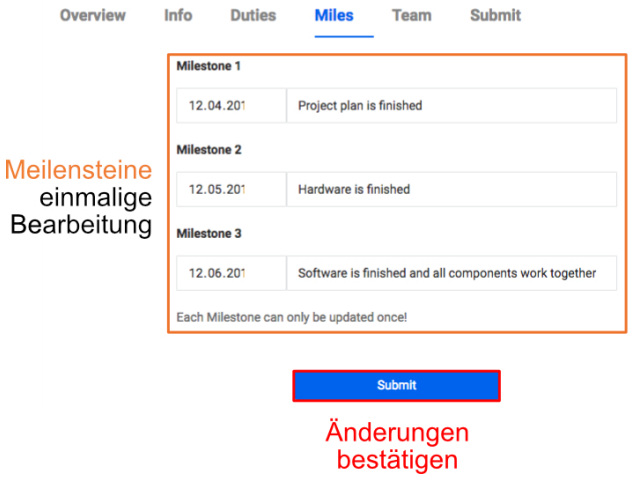


Abbildung H.36.: Änderungen im Miles-Tab

Team-Tab
Sollte ein Mitglied das Projekt-Team verlassen und Sie möchten deshalb ein neues hinzufügen oder das Team nach dem Erstellen des Projekts erweitern, so können Sie das im Tab *Team* machen. Dazu klicken Sie auf die Position, die noch nicht besetzt wurde und wählen einen Schüler des Dropdown-Menüs aus.

Abbildung H.37.: Änderungen im Team-Tab

Submit-Tab
Sobald das Projekt fertiggestellt ist und der Zeitpunkt des Erstellens eine Woche zurückliegt, kann das Projekt über den Tab *Submit* abgegeben und dem Lehrer zur Bewertung freigegeben werden. Dazu fassen Sie alle projektrelevanten Dateien in einem ZIP-Archiv zusammen und laden dieses über den blauen Button mit der Aufschrift *Browse* (in H.38 orange markiert) hoch.

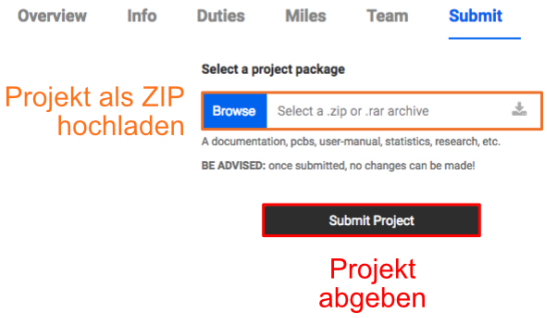


Abbildung H.38.: Projektabgabe

Leave Project-Team-Tab
Sollte man als Teammitglied das Projekt verlassen wollen, so kann man dies unter Angabe eines Grundes tun. Selbstverständlich können Sie als Projektleiter das Team nicht verlassen, da ansonsten kein Schüler alle Berechtigungen und die gesamte Verantwortung über das Projekt innehat. Dies soll auch ein Anreiz sein, ein Projekt nur anzulegen, wenn dies auch tatsächlich durchgeführt werden möchte.

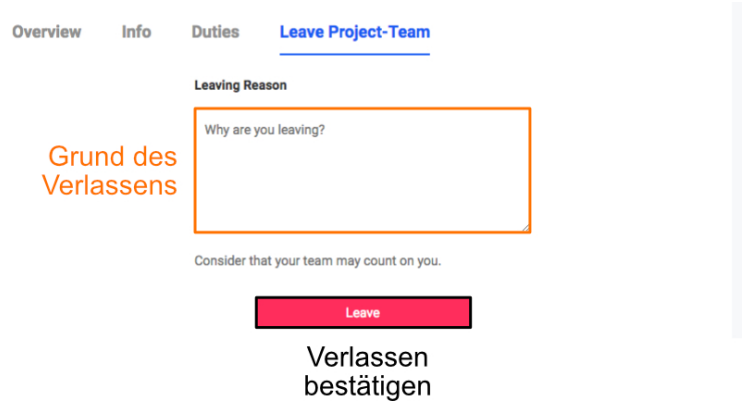


Abbildung H.39.: Projekt verlassen

H.1.6. Lehrerspezifische Funktionen

Projekte

Als *Teacher* haben Sie die Möglichkeit, zugeteilte Projekte anzusehen, deren Details zu bearbeiten und Sie nach der Abgabe zu bewerten. Dabei können Sie bei mehreren Projekten gleichzeitig als betreuender Lehrer wirken. Sollten Sie einem Projekt zugeteilt werden, das Sie nicht betreuen möchten, so können Sie jederzeit mit Angabe eines Grundes das Team verlassen - ein selbstständiges Zuteilen zu einem Projekt ist jedoch nicht möglich.

Um zur Übersichtsseite der Projekte zu kommen, öffnen Sie als Lehrer in der Seitenleiste das Untermenü *Education* und klicken in diesem auf *My Projects* (in H.40 orange markiert).

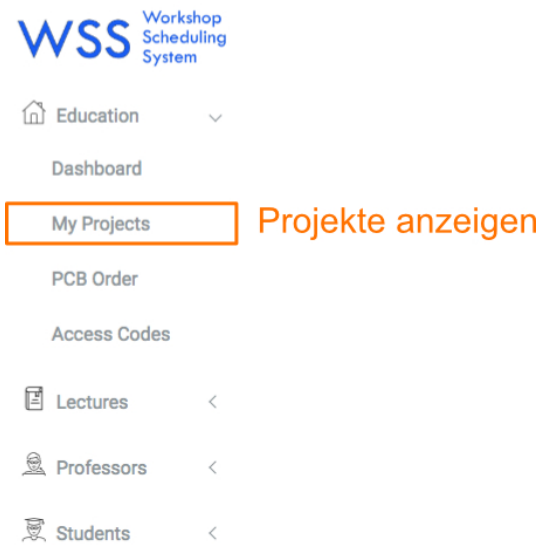


Abbildung H.40.: Aufrufen der Projektübersicht

In der Projektübersicht sehen Sie alle Projekte, bei denen Sie als betreuender Lehrer eingetragen sind. Durch einen Klick auf den Projektnamen (in H.41 orange markiert) gelangen Sie auf die Projektseite mit allen Informationen und Bearbeitungsmöglichkeiten. Wenn ein Projekt abgegeben wurde - sichtbar durch den Wert *Assigned* in der Spalte *Status* (rot markiert) -, können Sie dieses gesamt (schwarz umrandet) und jeden einzelnen Projektteilnehmer (grüne Markierung) benoten. Um die ZIP-Datei mit allen relevanten Dateien herunterzuladen, klicken Sie auf den Schriftzug *Download* (blau markiert).

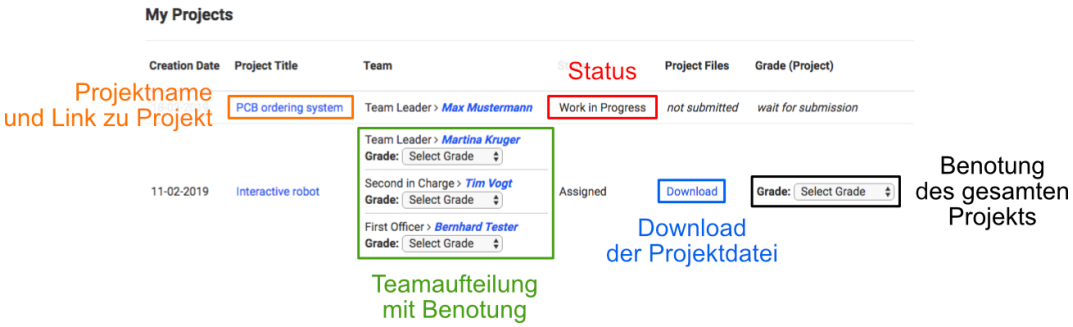


Abbildung H.41.: Projektübersicht

Nachdem Sie auf den Namen des Projekts geklickt haben, werden Sie auf die Übersichtsseite weitergeleitet, auf der Sie alle Informationen im Detail sehen (vgl. H.32 in *Schülerspezifische Funktionen*). Im Reiter *Miles* können Sie die Meilensteine des Projekts anpassen. Sollten Sie das Projekt verlassen wollen, so können Sie dies im Tab *Leave Project-Team* machen (genaue Erklärung siehe Unterabschnitt H.1.5).

Access Codes

Die Relevanz und Anwendung von *Access Codes* wurde bereits in H.1.3 beschrieben. Um zur Auflistung dieser zu gelangen, öffnen Sie als Lehrer in der Seitenleiste das Untermenü *Education* und klicken in diesem auf *Access Codes* (in H.42 orange markiert).

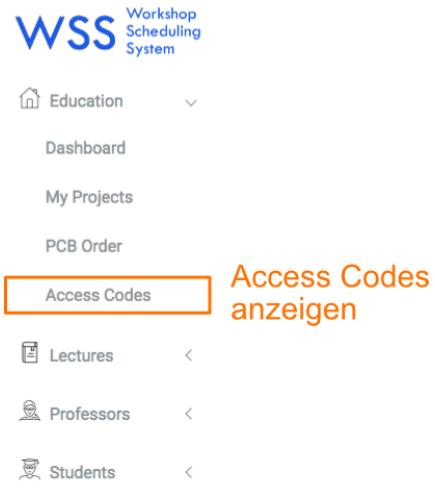


Abbildung H.42.: Aufrufen der Access Codes

In der Übersicht der *Access Codes* werden diese (in H.43 rot markiert) mit dem verbundenen Benutzer (schwarz markiert) angezeigt. Ist die Benutzerzelle leer, so kann dieser *Access Code* zum Registrieren eines neuen Schülers verwendet werden. Um nur unbenutzte *Access Codes* anzuzeigen, wählen Sie im Dropdown-Menü unter dem fett gedruckten Schriftzug *Access Codes* die Kategorie *Unused accesscodes* aus (orange umrandet). Abhängig von der ausgewählten Option im grün markierten Dropdown-Menü werden alle *Access Codes* der ausgewählten Kategorie (bei *Export All*) oder alle Angezeigten (bei *Export Shown*) exportiert. Zum Herunterladen der exportierten Datei klicken Sie auf den in H.43 blau markierten Button, in dessen Dropdown-Menü Sie den Dateityp auswählen können. Durch Klicken auf den gewünschten Typen wird die Datei heruntergeladen.

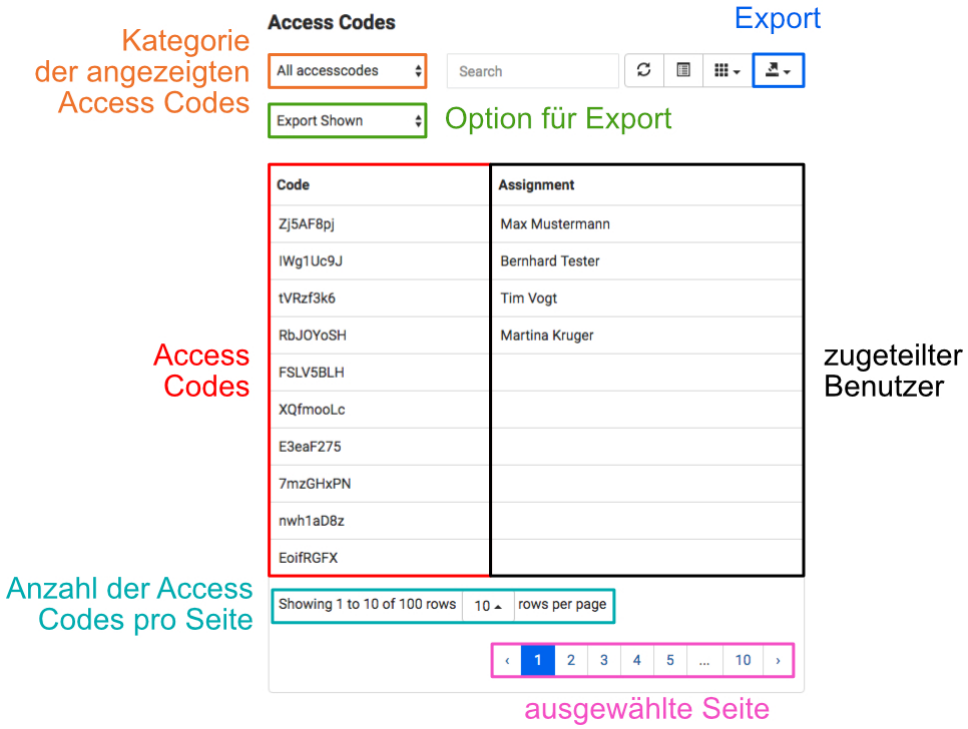


Abbildung H.43.: Übersicht der Access Codes

Lectures

Als Lehrer können Sie eigene Theoriestunden, sogenannte *Lectures*, erstellen und bis zur Bewilligung eines Admins bearbeiten. Sobald eine *Lecture* bewilligt ist, wird diese in den Zeitplan eingetragen, und Sie können nur noch einzelne Schüler bzw. die gesamte Theoriestunde entfernen. Die nähere Erklärung von *Lectures* finden Sie in H.1.3.

Um die Übersicht Ihrer *Lectures* aufzurufen, öffnen Sie als Lehrer in der Seitenleiste das Untermenü *Lectures* und wählen in diesem *My Lectures* (in H.44 orange markiert). Sollten Sie eine Theoriestunde erstellen wollen, so klicken Sie im selben Untermenü auf *Add Lecture* (schwarz umrandet).

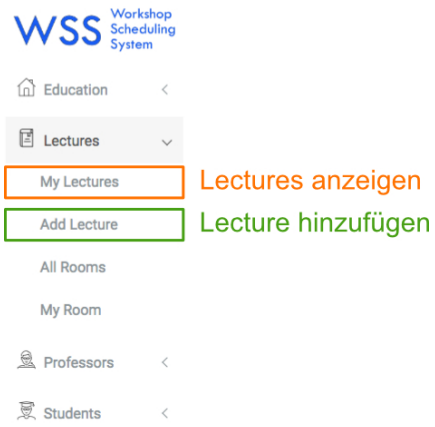


Abbildung H.44.: Aufrufen der Lectures

Auf der Seite zum Hinzufügen einer *Lecture* befinden sich auf der linken Seite die Pflichtfelder (in H.45 orange markiert) und auf der rechten die optionalen *Lecture*-spezifischen Fähigkeiten (schwarz umrandet), die Sie in vierfacher Ausführung angeben können. Damit helfen Sie dem Admin beim Bewilligen von Theoriestunden, schneller die Relevantesten herauszufiltern und einzuteilen.
Zu den Pflichtfeldern zählen der Titel, die Beschreibung, das geplante Datum des Stattfindens und die zugeteilten Schüler. Da Schüler sich selbst nicht in *Lectures* einteilen können, müssen Sie hier all jene angeben, die diese Theoriestunde besuchen sollen. Beim Bewilligen der *Lecture* wird diese, wenn der Schüler nicht bereits in einer anderen Theoriestunde eingeteilt ist, bindend in seinen Zeitplan gebucht.

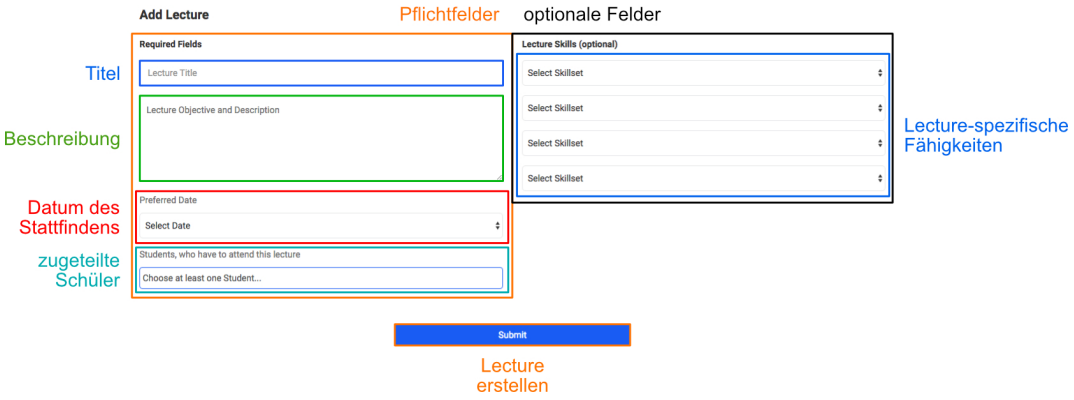


Abbildung H.45.: Erstellen einer Lecture

Haben Sie im Untermenü *Lectures* auf *My Lectures* geklickt, so werden Sie auf die Übersicht Ihrer Theoriestunden weitergeleitet. In dieser sehen Sie diese mit dem Titel (in H.46 orange markiert), dem Datum des Stattfindens (grün markiert), den Status der Bewilligung (rot umrandet), den Aktion-Buttons zum Anzeigen weiterer Details (blau markiert), Löschen einer *Lecture* (schwarz umrandet) und Bearbeiten dieser (türkis gekennzeichnet), tabellarisch aufgelistet. Letzteres ist nur in der Zeit möglich, in der die Theoriestunde nicht bewilligt ist.
Auf der rechten Seite (innerhalb des orangen Kastens) werden alle Details der ausgewählten *Lecture* angezeigt, darunter die spezifischen Fähigkeiten und die zugeteilten Schüler.

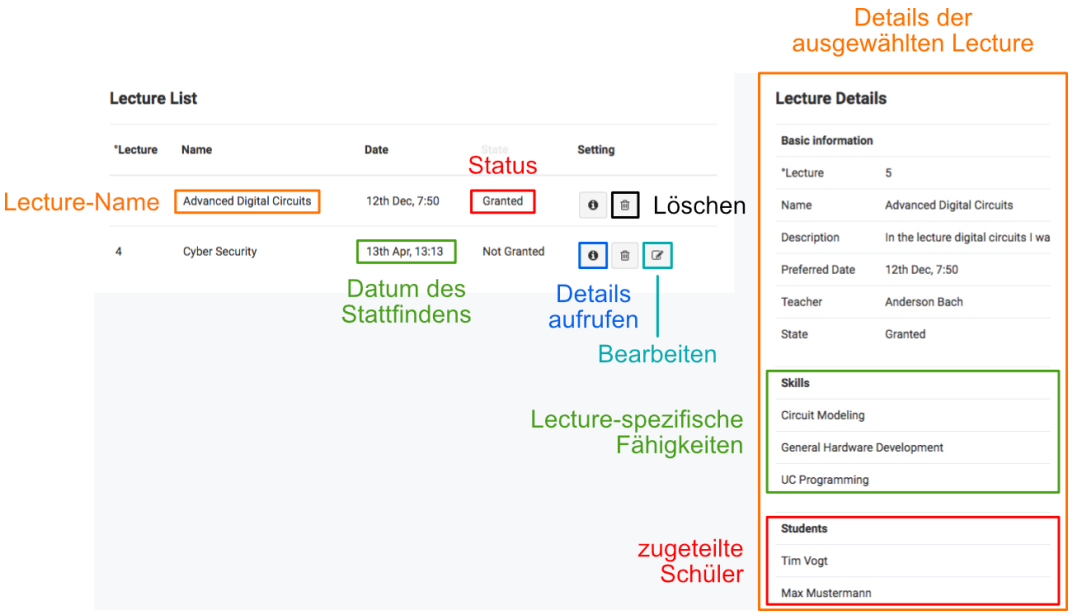


Abbildung H.46.: Übersicht der eigenen Lectures

Wird eine *Lecture* von einem Admin als *Granted* (dt. bewilligt) markiert, so wird sie in den Zeitplan der ausgewählten Schüler und in Ihren eigenen Zeitplan eingetragen. Nun können Sie noch einzelne Schüler durch einen Klick auf das Müllcontainer-Icon neben ihrem Namen entfernen (in H.47 rot gekennzeichnet) bzw. die gesamte Theoriestunde durch einen Klick auf das Müllcontainer-Icon neben dem Titel dieser löschen (schwarz umrandet).

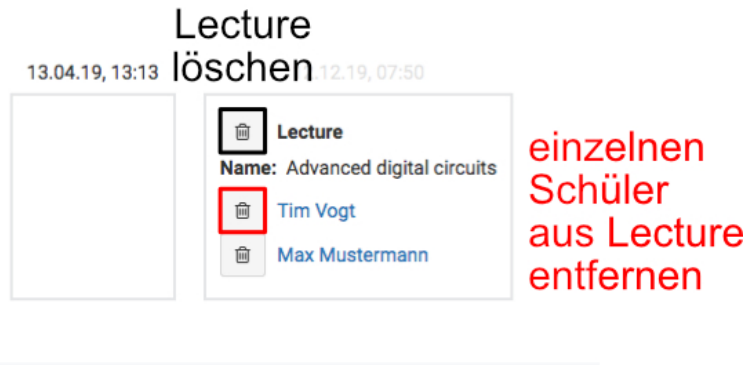


Abbildung H.47.: Lectures im eigenen Zeitplan

Bearbeiten des eigenen Raumes

Jedem Lehrer wird von einem Admin ein Raum zugeteilt. Dieser ist i. d. R. der bereits im Regelunterricht benutzte Raum. Um den eigenen Raum zu bearbeiten, öffnen Sie als Lehrer im Seitenmenü das Untermenü *Lectures* und wählen in diesem *My Room* aus.

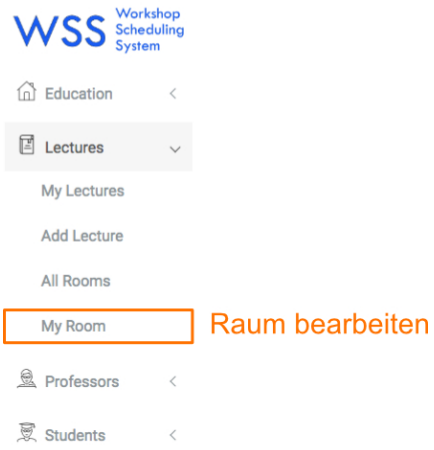


Abbildung H.48.: Aufrufen der Raumbearbeitung

Auf der aufgerufenen Seite können Sie die Raumbezeichnung (in H.49 orange markiert), die Anzahl der Sitz-/Arbeitsplätze verändern (grün umrandet) und eine Beschreibung hinzufügen (rot gekennzeichnet).

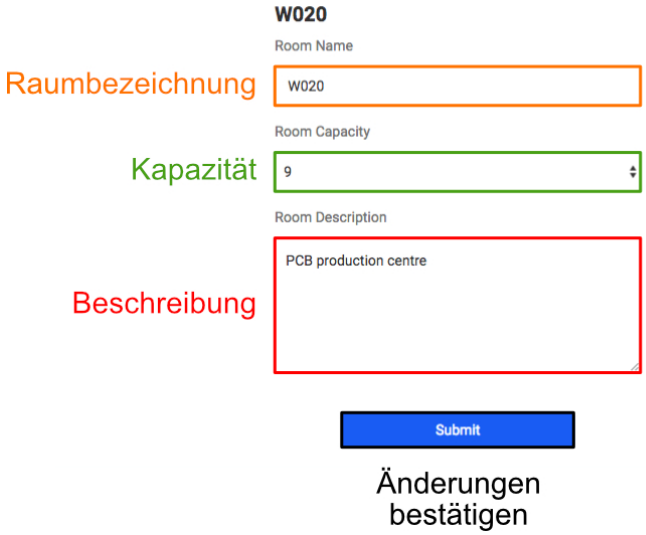


Abbildung H.49.: Ansicht der Raumbearbeitung

Verwaltung der Leiterplatten-Bestellungen

Sollten Sie als Lehrer von einem Admin die Berechtigung zur Leiterplatten-Verwaltung genehmigt bekommen haben, so sehen Sie im Seitenmenü unter dem Untermenü *Education* den Punkt *All PCB Orders*. Über diesen gelangen Sie zur Übersicht aller Leiterplattenbestellungen, die im WSS abgegeben wurden.

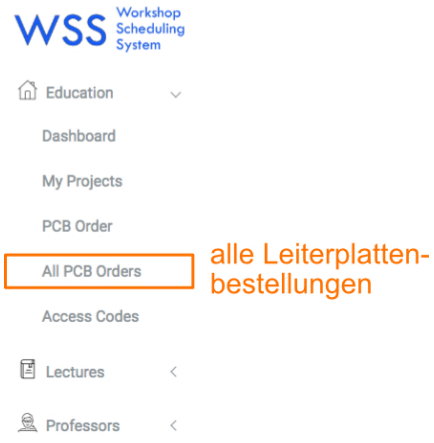


Abbildung H.50.: Aufruf der Übersicht der PCB-Bestellungen

Auf der Übersichtsseite befinden finden Sie auf der linken Seite alle Bestellungen tabellarisch aufgelistet. Bei diesen werden der Name des Bestellers (in H.51 orange markiert), dessen Klasse (grün umrandet), der Link zum Download (rot gekennzeichnet), sofern keine Filmvorlage abgegeben wird, den Zeitpunkt der Abgabe (schwarz), den derzeitigen Status (pink markiert) und ein Aktionsmenü zum Aufrufen der Details (blau markiert), dem Ändern des Status (schwarz umrandet) und dem Senden einer Nachricht angezeigt (türkis). Auf der rechten Seite finden Sie weitere Details der ausgewählten Bestellung. In diesem werden u. a. die Eigenschaften und die Größe & Anzahl des bestellten Prints erläutert.

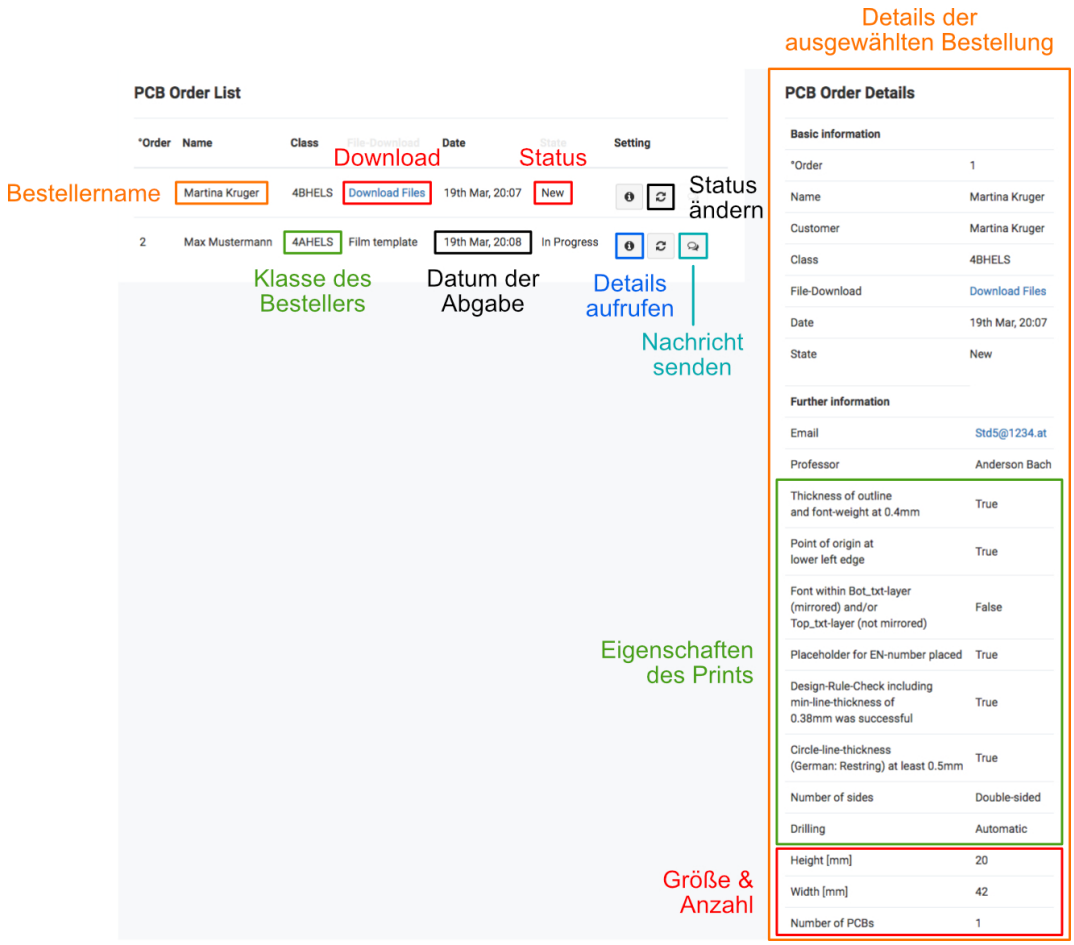


Abbildung H.51.: Übersicht der PCB-Bestellungen

Wenn Sie auf den Button zum Ändern des Status klicken, öffnet sich ein *Alert*, in dem Sie den neuen Status in einem Dropdown-Menü (in H.52 orange markiert) auswählen und diesen mit einen Klick auf *OK* bestätigen können.

Abbildung H.52.: Ändern des Status einer Leiterplattenbestellung

H.1.7. Adminspezifische Funktionen

Lectures

Das Anzeigen und Löschen von *Lectures* wurde bereits in H.1.6 geklärt. Im Folgenden werden nur noch das Bewilligen und Benachrichtigen des Erstellers besprochen.

Um eine Theoriestunde zu bewilligen, bzw. deren Bewilligung wieder zu entziehen, drücken Sie in der Übersicht der *Lectures* auf den *Änderungs*-Button (in H.53 orange markiert). Im folgenden *Alert* klicken Sie nun auf *OK*. Um den Ersteller der Theoriestunde zu benachrichtigen, klicken Sie auf das *Chat*-Symbol (grün umrandet).

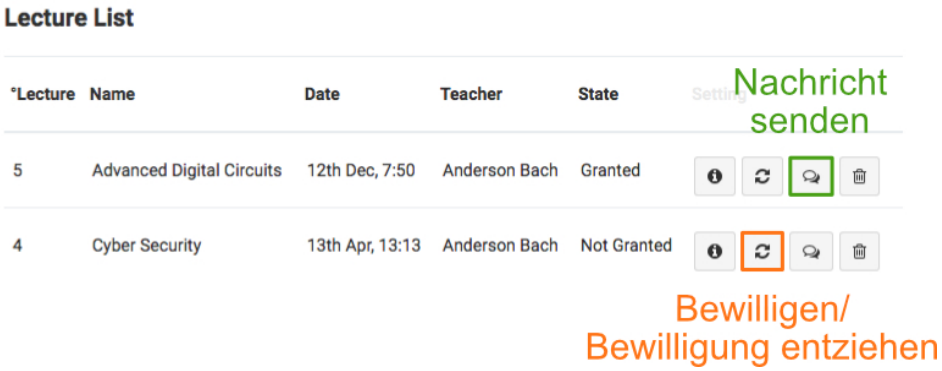


Abbildung H.53.: Adminspezifische Aktionen in der Lecture-Übersicht

Verwaltung von Räumen

Als Admin sind Sie dafür verantwortlich, dass alle benötigen Räume im WSS vorhanden sind. Zu diesem Zweck können Sie diese eintragen, bearbeiten und einem bestehenden Lehrer zuteilen.

Um die Seite der Raumverwaltung aufzurufen, öffnen Sie zuerst in der Seitenleiste das Untermenü *Lectures* und wählen in diesem den Punkt *Add/Edit Rooms*.

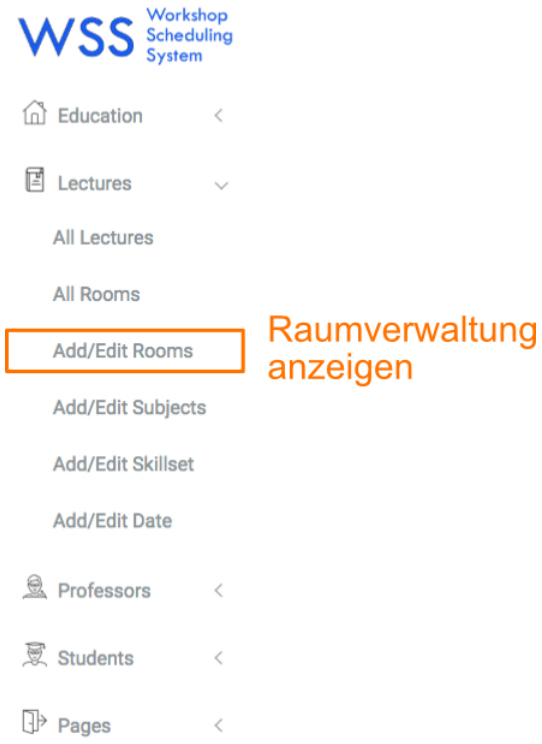


Abbildung H.54.: Aufruf der Raumverwaltung

In dieser sehen Sie im oberen Teil die im WSS bereits vorhandenen Räume, bei denen Sie durch *Inline-Editing* (ändern des Wertes, indem Sie auf diesen Klicken und dann den Neuen eingeben) oder Auswählen eines Wertes aus dem entsprechenden Dropdown-Menü diese bearbeiten können und im unteren Teil den Aktionsteil zum Hinzufügen eines Raumes.

Durch *Inline-Editing* können Sie den Namen des Raumes (in H.55 scharz umrandet) und die Beschreibung dieses (pink markiert) ändern. Wollen Sie den zugeteilten Lehrer (blau umrandet) oder die Anzahl der verfügbaren Plätze (rot gekennzeichnet) ändern, so können Sie dies durch das Auswählen eines neuen Wertes aus dem jeweiligen Dropdown-Menü tun. Durch Drücken des *Müllcontainer*-Logos in der Zeile des jeweiligen Raums, können Sie diesen aus dem System entfernen.

Um einen neuen Raum hinzuzufügen, tragen Sie im unteren Teil der Website unterhalb der Überschrift *Add Room* den Namen, die Anzahl der Arbeitsplätze und die Abteilung ein und bestätigen die Erstellung durch einen Klick auf den blauen Button mit der Aufschrift *Submit*. Optional können Sie vor der Erstellung eine Beschreibung ergänzen.

Abbildung H.55.: Ansicht der Raumverwaltung

Verwaltung von *Subjects*

Um die Verwaltung von *Subjects* (dt. unterrichtete Fächer) aufzurufen, klicken Sie im Seitemenü auf *Lectures* und wählen den Punkt *Add/Edit Subjects* aus.

Abbildung H.56.: Aufruf der *Subject*-Verwaltung

Nun können Sie im oberen Teil der Seite durch *Inline-Editing* den Namen des Theoriefaches ändern (innerhalb der orangenen Umrandung in H.57 schwarz markiert) und im unteren Teil unterhalb der Überschrift *Add Subject* durch Eingabe des Namens des *Subjects* einen neuen Eintrag erstellen.

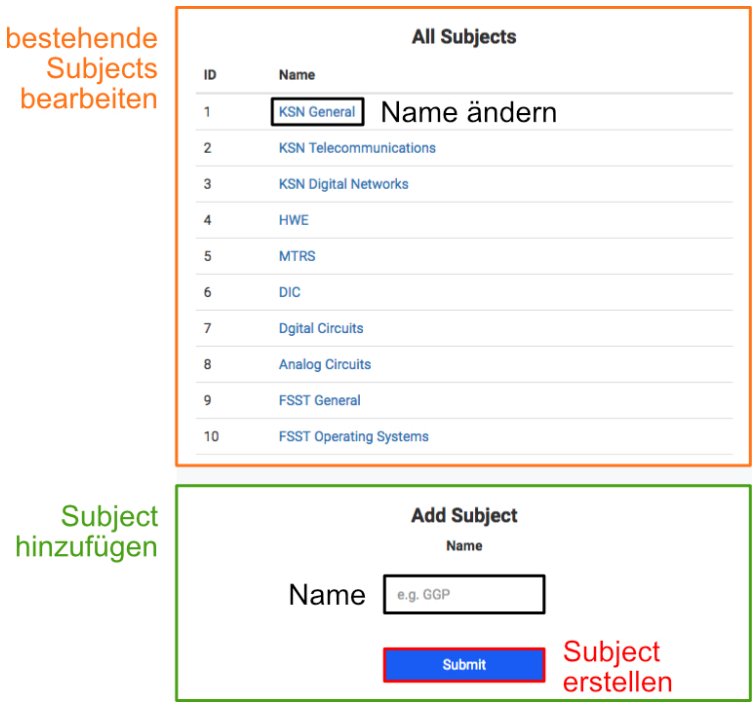


Abbildung H.57.: Ansicht der *Subject*-Verwaltung

Verwaltung von Skills

Die Verwaltung und Editierung von *Skills* ist ähnlich jener der *Subjects*. Zum Aufrufen klicken Sie im Seitenmenü auf *Lectures* und wählen den Punkt *Add/Edit Skillset* aus.

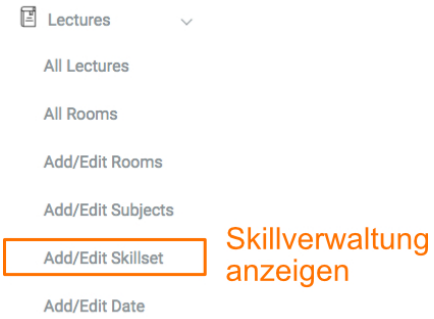


Abbildung H.58.: Aufruf der *Skillset*-Verwaltung

Dementsprechend ist auch die Ansicht der *Skillset*-Verwaltung analog der *Subject*-Verwaltung. Im oberen Bereich können Sie die bestehenden Fähigkeiten bearbeiten und im unteren Teil Neue hinzufügen.

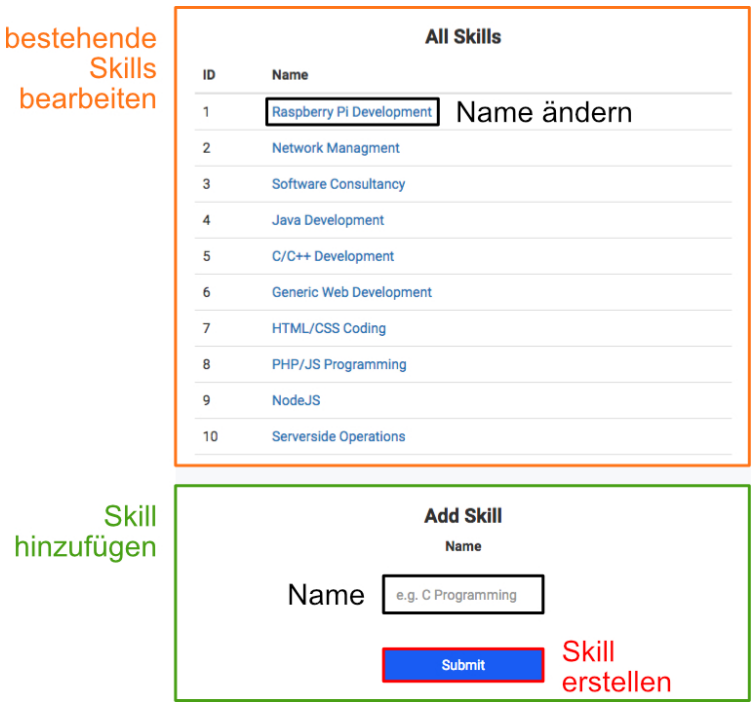


Abbildung H.59.: Ansicht der *Skillset*-Verwaltung

Verwaltung von *Dates*

Die Verwaltung von Dates ist ebenfalls der von *Subjects* nachempfunden. Zum Aufrufen klicken Sie im Seitenmenü auf *Lectures* und wählen den Punkt *Add/Edit Date* aus.

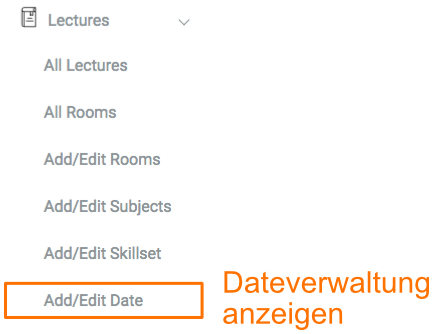


Abbildung H.60.: Aufruf der *Date*-Verwaltung

Im oberen Bereich der Website können die bisherigen Zeitpunkte durch *Inline-Editing* verändert werden bzw. durch einen Klick auf das *Müllcontainer*-Symbol in der jeweiligen Zeile gelöscht werden. Im unteren Bereich kann durch angeben eines Datums mit der Startzeit ein neues *Date* erstellt werden.

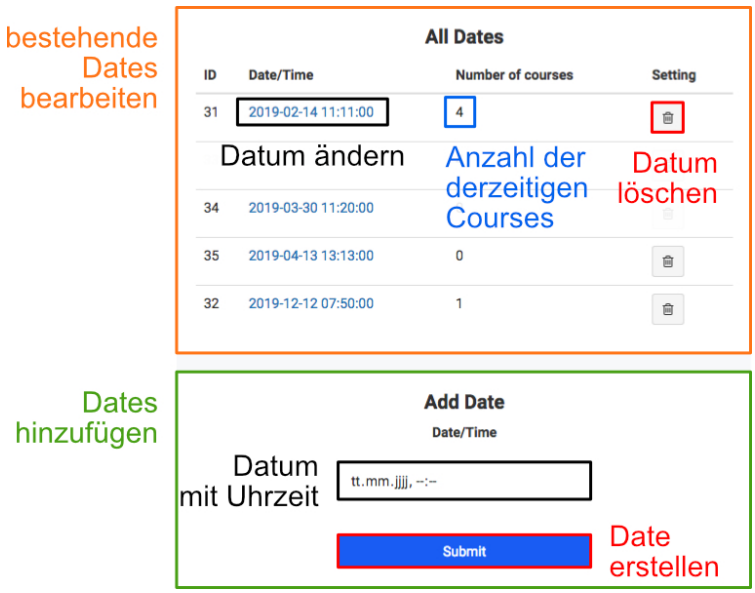


Abbildung H.61.: Ansicht der *Date*-Verwaltung

Verwaltung von Professoren

Als Admin sind Sie u. a. für das Hinzufügen von Konten für Professoren verantwortlich. Außerdem können Sie ein Lehrer-Konto löschen, wenn dieser nicht mehr im WSS tätig ist.

Auf das Hinzufügen bzw. Entfernen von Lehrern greifen Sie zu, indem Sie in der Seitenleiste das Menü *Professors* öffnen und in diesem den Punkt *Add Professor* zum Hinzufügen eines Lehrerkontos auswählen bzw. *All Professors* auswählen, um eines zu löschen.

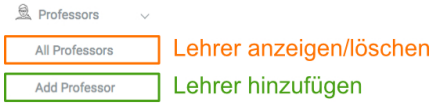


Abbildung H.62.: Aufruf der Professoren-Verwaltung

Wenn Sie einen Lehrer erstellen, befinden Sie sich zuerst auf dem Tab *Personal Information*, wo Sie die persönlichen Details eines Lehrers eingeben können. Um einen Lehrer anzulegen, müssen Sie dessen Namen (in H.63 innerhalb des orangen Kastens blau markiert), Kürzel (grün umrandet), E-Mail (rot gekennzeichnet) und ein Passwort (türkis) angeben, das Sie diesem mitteilen.
Optional können Sie auf der rechten Seite (schwarzer Kasten) die Abteilung (blau umrandet), das Geschlecht (grün markiert), Geburtsdatum (rot), die Telefonnummer (türkis gekennzeichnet), eine Beschreibung zur Person (orange markiert) und die URL zur persönlichen Website (blau umrandet) angeben.
Soll der Lehrer die Leiterplattenbestellungen verwalten können, so aktivieren Sie das Kästchen neben *Access to PCB production* (rot gekennzeichnet).

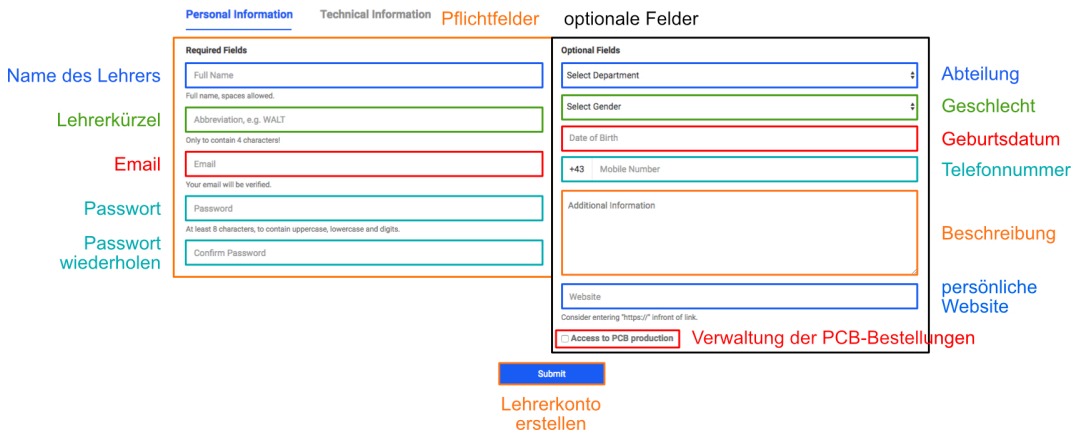


Abbildung H.63.: Erstellen eines Lehrerkontos - persönliche Details

Nun können Sie entweder das Profil durch einen Klick auf den blauen Button mit der Aufschrift *Submit* erstellen, oder im zweiten Tab *Technial Information* weitere fachbezogene Details zum Lehrer eingeben.

Dazu zählen auf der linken Seite der Raum des Lehrers (in H.64 orange markiert), seine Fähigkeiten (grün gekennzeichnet) und auf der rechten Seite die unterrichtenden Fächer (rot umrandet).
Nun erstellen Sie durch einen Klick auf den *Submit*-Button das Lehrerkonto.

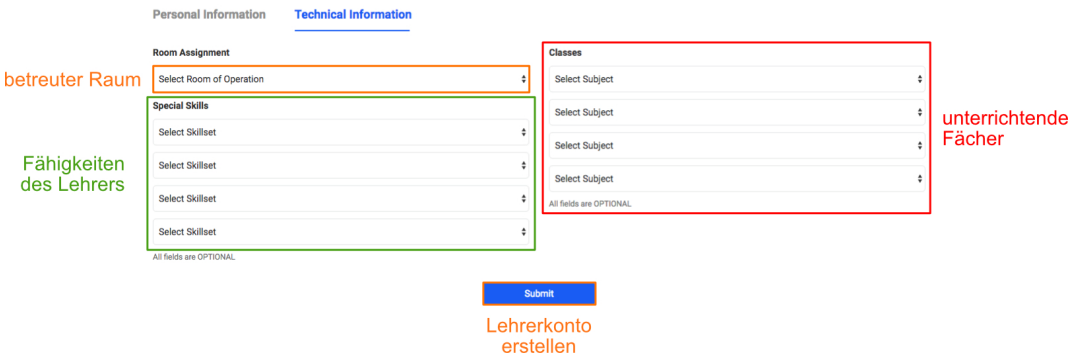


Abbildung H.64.: Erstellen eines Lehrerkontos - fachbezogene Details

Um einen Lehrer zu löschen, klicken Sie im Seitenmenü unter *Professors* auf *All Professors*. Nun befinden Sie sich in der Auflistung aller Lehrer, in der Sie durch einen Klick auf *Delete User* das Profil des jeweiligen Lehrers löschen. Um den Löschvorgang zu vervollständigen, geben Sie im folgenden *Alert Yes* ein und Klicken auf *OK*.

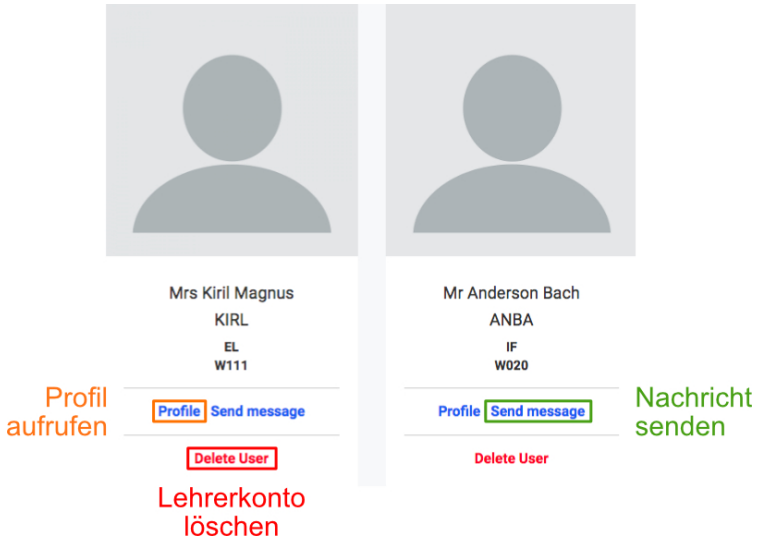


Abbildung H.65.: Löschen eines Lehrerkontos

Verwaltung von Schülern

Als Admin können Sie die Konten von Schülern auf der Übersicht aller Schüler löschen. Dazu öffnen Sie im Seitenmenü das Untermenü *Students* und klicken darin auf den Punkt *All Students*.

Abbildung H.66.: Aufruf der Schülerverwaltung

In dieser Übersicht sehen Sie die Schüler der links oben ausgewählten Kategorie (*Selected course*, in H.67 orange markiert). Die ausgewählte Kategorie zeigt alle Schüler an, die sich für diesen Raum eingetragen haben bzw. alle registrierten Schüler bei der Auswahl *All students*.
Um das Konto eines Schülers zu löschen, klicken Sie auf das *Müllcontainer*-Symbol in der jeweiligen Zeile (rot gekennzeichnet) und geben im erscheinenden *Alert Yes* ein. Sie bestätigen den Vorgang mit einem Klick auf *OK*.

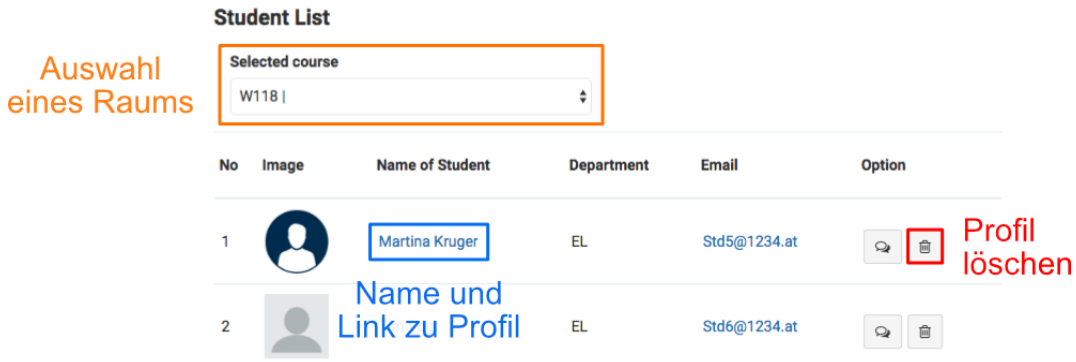


Abbildung H.67.: Löschen eines Schülers

WSS - WORKSHOP SCHEDULING SYSTEM

ALLGEMEINE GESCHÄFTSBEDINGUNGEN

……

(Workshop Scheduling System GesbR, Waldstraße 3, 3101 St. Pölten, Telefon: +43 2742 750 51-0, Email: office@htlstp.at)

I. Allgemeines

Das Unternehmen *Workshop Scheduling System GesbR* (im Folgenden mit WSS abgekürzt) bietet die im Anhang angeführten Dienstleistungen nach dem Stand der Technik und mit der Sorgfalt eines ordentlichen Unternehmers an. Dies unter bestmöglicher Wahrung der Interessen des Kunden.

1. Allgemeine Bestimmungen

1.1. Allgemeine Geschäftsbedingungen

Die Allgemeinen Geschäftsbedingungen, welche ausgehängt sind, gelten für sämtliche Dienstleistungen des WSS.
Sofern nichts anderes vereinbart, gelten diese Geschäftsbedingungen als maßgeblicher Vertragsbestandteil des zwischen WSS und dem Kunden geschlossenen Auftrags.

2. Auftragserteilung

2.1. Grundsätzlich wird der erteilte Auftrag in einem Auftragsschein festgehalten. Dort werden die zu erbringenden Leistungen genau bezeichnet. Der Kunde erhält eine Abschrift.

3. Preise

3.1. Das Service des WSS wird in vollem Umfang kostenlos angeboten. Sollten in Zukunft kostenpflichtige Optionen hinzugefügt werden, wird der User darüber informiert. Alle bisherigen Optionen bleiben dabei kostenfrei.

II. **Service**

4. Sicherheit vom Stattfinden von *Courses*

4.1. Ist der User als Typ *Schüler* registriert, kann dieser sich in *Courses* eintragen. Sofern nicht angegeben ist, dass dieser als *Fixed course* gewertet ist, kann der *Course* jederzeit gelöscht werden. Bei einer Löschung wird der User informiert.

4.2. Die Anzahl von *Fixed courses* wird vom Admin vorgegeben und ist für jeden Schüler pro Lehrer gleich.

WSS Workshop Scheduling System

Datenschutzerklärung

Wir verarbeiten Ihre personenbezogenen Daten, die unter folgende **Datenkategorien** fallen:

- Name,
- Geschlecht,
- Email,
- Geburtsdatum,
- Telefonnummer,
- persönliche Website

Sie haben uns Daten über sich freiwillig zur Verfügung gestellt und wir verarbeiten diese Daten auf Grundlage Ihrer **Einwilligung** zu folgenden Zwecken:

- Betreuung des Kunden sowie
- Vereinfachung der Kontaktherstellung innerhalb der registrierten User

Sie können diese Einwilligung jederzeit widerrufen. Ein **Widerruf** hat zur Folge, dass wir Ihre Daten ab diesem Zeitpunkt zu oben genannten Zwecken nicht mehr verarbeiten. **Für einen Widerruf wenden Sie sich bitte an:** office@htlstp.at

Die von Ihnen bereit gestellten Daten sind weiters zur **Vertragserfüllung** bzw. zur Durchführung vorvertraglicher Maßnahmen erforderlich. Ohne diese Daten können wir den Vertrag mit Ihnen nicht abschließen.

Wir **speichern** Ihre Daten für die gesamte Dauer Ihres Schulbesuchs.
Außerdem verwendet unser System **Cookies**, um ein automatisches Log-In zu ermöglichen.

Die Datenverarbeitung erfolgt automatisch durch das System und ggf. durch den Admin.

Ihre Daten bleiben ausschließlich innerhalb der HTBLuVA St. Pölten und werden unter keinen Umständen an dritte weitergegeben.

Rechtsbehelfsbelehrung
Ihnen stehen grundsätzlich die Rechte auf Auskunft, Berichtigung, Löschung, Einschränkung, Datenübertragbarkeit und Widerspruch zu. Dafür wenden Sie sich an uns. Wenn Sie glauben, dass die Verarbeitung Ihrer Daten gegen das Datenschutzrecht verstößt oder Ihre datenschutzrechtlichen Ansprüche sonst in einer Weise verletzt worden sind, können Sie sich bei der Aufsichtsbehörde beschweren. In Österreich ist die Datenschutzbehörde zuständig.