

**Project Simple Studies**

**in Bezug auf Moodle**

**PROJEKTARBEIT**

im Studiengang Angewandte Informatik

an der Dualen Hochschule Baden-Württemberg Mosbach

von

**Louis Mathuni, Nicolas Messerschmidt, Matthias Lay und Marco Erne**

Abgabedatum 12.06.2022

Matrikelnummern 3226047, 8509440, 2759750, 6772873

Betreuer der DHBW Martin Lüttecke

Eigenständigkeitserklärung

Hiermit erklären wir, dass wir diese Arbeit selbständig verfasst und keine anderweitigen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Alle wörtlich übernommenen Textstellen aus fremden Quellen wurden kenntlich gemacht.

Bad Mergentheim, den 12.06.2022

Marco Erne, Matthias Lay, Nicolas Messerschmidt und Louis Mathuni

Inhaltsverzeichnis

[Vorwort 3](#_Toc105959344)

[1. Sprint 4](#_Toc105959345)

[2. Sprint 5](#_Toc105959346)

[3. Sprint 6](#_Toc105959347)

[4. Sprint 8](#_Toc105959348)

[Produktvorstellung 10](#_Toc105959349)

[Lessons learned 11](#_Toc105959350)

[Fazit 12](#_Toc105959351)

# Vorwort

Aus Gründen der Lesbarkeit wird in dieser Arbeit nur die männliche Personenform verwendet.

Die ihnen vorliegende Ausarbeitung fertigen wir im Rahmen unseres Angewandten Informatik Studiums an der Dualen Hochschule Baden Württemberg, am Campus Bad Mergentheim, im Fach Anwendungsprojekt Informatik, an.

Während unserer bisherigen Studienzeit an der Dualen Hochschule Baden Württemberg, entwickelte sich bei uns die Idee zu diesem Thema. Da wir selbst Studenten sind, sind wir selbst auch Teil unserer Zielgruppe und können somit beurteilen, welche Funktionen hilfreich sind.

Gemeinsam als vierer Team, konnten wir diese Ausarbeitung verfassen. Dabei nahm Louis Mathuni die Rolle des Scrum Master, Nicolas Messerschmidt die des Product Owners, Matthias Lay die des Senior Developer und Marco Erne die des Junior Developers, ein.

# Sprint

Sprint Planning:

Wir haben uns anfangs auf die fundamentalen Dinge unseres Projektes konzentriert. Dazu war unser Plan uns ein konsistentes GitHub Projekt zu erstellen, auf welchem die einzelnen Entwickler, also wir, gleichzeitig arbeiten können. Zu Beginn der Arbeit lädt man sich den aktuellen Stand herunter und sobald ein Arbeitsschritt erledigt ist, lädt man den Status wieder hoch. So gewährleistet man, dass jeder immer vom aktuellen Stand aus arbeiten kann und es keine Überschneidungen gibt. Unsere Entwicklungsumgebung soll dabei Visual Studio 2022, mit Windows Forms sein und unser Projektmanagement Tool ein Trello Board.

Um keine größeren Kollisionen bei der Erstellung des Datenbankcodes und der Projektstruktur zu verursachen, wurde das Ziel gesetzt, das Layout des Programmes zu entwickeln, sodass jeder Entwickler die gleiche Basis hatte. Zudem wurde das generelle Design-Theme geplant. Dazu zählen wir die Auswahl einer stimmigen Farbpalette, sowie die Größe der Knöpfe etc. Zudem ist die Einbindung der benötigten Datenbanken geplant, diese zu erstellen und einzubinden.

Die erste Kernfunktion unseres Programmes, die Stundenplanansicht, zu implementieren wurde ebenfalls auf die Liste der im ersten Sprint zu erledigenden Punkte gesetzt.

Sprint Verlauf:

Von vornherein war uns bewusst, dass die Verwendung der Tools wie GitHub und Visual Studio mit Windows Forms einige Probleme verursachen würde. Die meisten von uns hatten teilweise noch gar nicht mit Github / Visual Studio gearbeitet oder nur mit anderen Programmen, welche sich zwar oft sehr ähneln, dann im Detail jedoch noch stark unterschiedlich sein können.

Zunächst gab es bei der Installation der zu verwendenden IDE Visual Studio einige Probleme, die dann aber relativ schnell durch gemeinsames Begutachten behoben wurden.

Außerdem gab es Schwierigkeiten im Umgang mit der Versionsverwaltungssoftware GitHub, da es nicht selten vorkam, dass durch falsches Hochladen einer neuen Projektversion das gesamte Projekt überschrieben wurde. Eine weitere Gefahr bestand darin, dass man manchmal seinen neuen Stand nicht hochgeladen hat dann aber herunterlädt, überschreibt es die Arbeiten, die man selbst vorgenommen hatte. Das Projektmanagement Tool Trello Board wurde von allen gut verstanden und auch gerne genutzt.

Bei den ganzen Problemen war es wichtig, dass wir uns untereinander mit den Problemen halfen, um einen weitestgehend flüssigen Arbeitsablauf zu garantieren.

Der restliche Sprint lief weitestgehend reibungslos und am Ende des ersten Sprints hatten wir einiges erreicht.

Sprint Review:

Für unseren ersten Sprint waren wir mehr als zufrieden. Die anfänglichen Schwierigkeiten wurden durch Teamarbeit gut bewältigt und auch die Weiterbildung jedes einzelnen in den benutzten Tools hat maßgeblich zum Erfolg des Sprints beigetragen.

Die Erstellung unseres Layouts sowie das Design-Theme des Programms wurde entwickelt und bietet uns eine wunderbare Basis für unsere weiteren Sprints. Die Datenbanken wurden ebenfalls passend erstellt und auch in unser Programm eingebunden.

Die erste Kernfunktion unseres Programmes, die Stundenplanansicht, haben wir zeitlich nicht mehr umgesetzt bekommen. Das ist hauptsächlich auf die anfänglichen Probleme mit vor allem dem Tool GitHub zurückzuführen.

Im ersten Sprint wurde also das Fundament gebaut, welches wir für die weitergehende Entwicklung benötigen.

# Sprint

Sprint Planning:

Nachdem nun eine grundlegende Struktur des Programms vorhanden war, haben wir uns zum Ziel gesetzt, einige der geplanten Funktionen zu

implementieren. So war der Plan, den Lernplan sowie die Hausaufgabenverwaltung und die E-Mail Adressen Ansicht zu entwickeln.

Im Lernplan sollten die Studierenden sich nach ihren Bedürfnissen einplanen können, an welchen Tagen sie welches Thema für die Klausurenphase lernen.

Für den chaotischen Studentenalltag soll so eine Abhilfe geschaffen werden, damit die Klausuren, das wichtigste, nicht verhauen werden.

Außerdem ist klar geworden, dass eine weitere Funktion zum Eintragen der Kurse benötigt wird, um essentielle Funktionen des Programms umzusetzen.

Es müssen nämlich die Kurse im Programm vorhanden sein, damit in den jeweiligen Funktionen dediziert zu den einzelnen Kursen Einträge

vorgenommen werden können. Des Weiteren sollte der nicht fertiggestellte Stundenplan aus dem ersten Sprint nun vollständig implementiert werden.

Bei diesem gab es im ersten Sprint noch Probleme beim Einbinden.

Sprint Verlauf:

Auch im zweiten Sprint lief nicht alles nach Plan. Vor allem mit GitHub gab es immer wieder Probleme, da Konflikte in den Versionen nicht

immer aufgelöst werden konnten. Allerdings wurden auch diese Probleme im Team gelöst und überwunden, sodass keine größere Hindernisse auftraten.

Der Stundenplan konnte dann relativ schnell umgesetzt werden und war einsatzbereit. Im Verlauf des Sprints wurden dann auch der Lernplan und die Einträge der Kurse implementiert. Zum Ende des zweiten Sprints fiel die Funktion des Sidepanels wieder für eine kurze Zeit aus.

Die Ursache für diesen Ausfall konnte dann nach einigem Suchen gefunden und zurückgesetzt werden. Schließlich war auch der zweite Sprint schon zu Ende.

Sprint Review:

Im zweiten Sprint konnten wir unsere Pläne weitestgehend umsetzen. Der Product Owner war mit den Ergebnissen der Developer in hohem Maße zufrieden und die Stakeholder bewunderten den Fortschritt.

Nun war die Hälfte der vorgesehenen Entwicklungszeit schon vorbei und es war Zeit, in die zweite Halbzeit überzugehen. Mit dem allgemeinen Fortschritt waren wir im Großen und Ganzen zufrieden, obwohl sich manche Arbeiten

durch die genannten Probleme etwas verzögert hatten. Deshalb gab es leichte Bedenken, ob im Verlaufe der zweiten Hälfte auch größere Probleme auftreten könnten, die dann die Entwicklung erheblich verzögern können.

# Sprint

Sprint Planning:

Im dritten Sprint sollten insbesondere die Epics Lernplan, Notenübersicht, E-Mail und NewsFeed zum Leben erweckt werden. Zudem sollten im Rahmen des Epics Projektpräsentation ein erster grober Fahrplan inklusive Aufgabenunterteilung angefertigt werden. Hierbei wurde zunächst diskutiert, in welcher Form der NewsFeed erscheinen und welche Informationen er enthalten soll. Dabei stand auf der einen Seite der Vorschlag eines nutzerabhängigen, individuellen NewsFeeds, der im Bezug auf das Product Goal selbstverständlich wünschenswert gewesen wäre, allerdings nach Abschätzung des notwendigen Aufwands der Implementierungsarbeiten einstimmig verworfen wurde. Stattdessen setzte sich der Gegenvorschlag durch, der einen einheitlichen NewsFeed für die Benutzer des Endprodukts vorsieht. Auch beim Epic E-Mail mussten der Product Owner und der zuständige Entwickler zunächst festlegen, welche Mailadressen aufgeführt werden sollten und woher diese zu beziehen seien. Vor allem die Diskussion über die zentrale Speicherung privater Mailadressen erhitzte die Gemüter, da der Grundsatz der bestmöglichen Usability gegen den Datenschutz aufgewogen werden müssen und nicht auf einem zentralen Server von uns bereitgestellt werden. Letztendlich wurde sich darauf geeinigt, dass E-Mail-Adressen vom Benutzer selbst eingetragen werden. Beim Epic Aufschriebeverwaltung wurde schnell klar, dass es schwer werden würde, dies im Zeitrahmen des Projektes umzusetzen und dass man sich eventuell beim Auflisten der anfangs geplanten Features etwas übernommen hatte. Daraus resultierte die Meinungsverschiedenheit darüber, ob die Aufschriebeverwaltung in Gänze verworfen werden sollte oder ob eine Reduktion des Features auf das Hinzufügen von eigenen Aufschrieben durch den Benutzer im Rahmen des Product Goals zufriedenstellend wäre. Da letztere Option aber nicht mit dem Product Goal, das eine sorgfältige Implementierung und die Überzeugung des Teams von der Nützlichkeit des hinzuzufügenden Features verlangt, nicht zu vereinbaren war, wurde dieser Vorschlag und somit auch das Feature Aufschriebeverwaltung einstimmig verworfen. Im Epic Notenübersicht konnte sich schnell auf die anfangs vorgesehenen Tasks und die entsprechenden Funktionen geeinigt werden.

Die entsprechenden Änderungen im Product Backlog sowie Sprint Backlog wurden dann vom Scrum-Master in Trello übernommen.

Sprintverlauf:

Im ersten Daily wurden die Tasks der einzelnen Entwickler noch einmal mit dem Product Owner abgesprochen und im Rahmen dessen die Entscheidungen aus den Diskussionen im Sprint Planning nochmals wiederholt, bevor sich jeder seinen Aufgaben widmete.

Zu Beginn führte Senior Entwickler Matthias Lay das Team in die Grundlagen und Verwendung des Entity Frameworks ein, um ein breiteres Verständnis und Know-How im Team und damit ein flexiblere Aufgabenunterteilung zu ermöglichen. Im Laufe des Sprints machten sich die mangelnde Erfahrung mit Entity Framework dennoch bemerkbar, womit weitere Beratung durch den Senior Entwickler nicht zu vermeiden war. Ein weiteres Problem im Sprint entstand durch die Verwendung von GitHub, da bei einigen Teammitgliedern einige Softwarekomponenten dieses sonst zuverlässigen und hilfreichen Tools fehlten. Dies hatte zur Folge, dass beim Hochladen des eigenen Entwicklungsstatus in das Gesamtprojekt die vorangegangenen Änderungen anderer Teammitglieder vollständig rückgängig gemacht oder verändert wurden, was dazu führte, dass einige Implementierungen zunächst unbemerkt überschrieben wurden und verloren gingen. Nach Recherche der Fehlerquelle und Ergänzung dieser fehlenden Softwarekomponenten bei den betroffenen Entwicklern konnte diese Hürde allerdings überwunden und für den weiteren Verlauf beseitigt werden.

In einem Teammeeting wurde dann auch die Projektpräsentation und insbesondere Vorschläge zur interaktiven Gestaltung dieser diskutiert. Zeitlich soll die Organisation dieser vor allem vom Product Owner und Scrum Master im 4. Sprint stattfinden. Auch die Übungstermine sollen teils in Präsenz, teils online aufgrund weiter Anfahrten realisiert werden.

Sprint Review

Trotz der oben erwähnten Schwierigkeiten konnten im dritten Sprint alle beim Sprint Planning gesetzten Ziele zufriedenstellend erreicht werden. Bei diesem dritten Sprint handelte es sich um einen umfangreichen und maßgeblichen Sprint, da auf der einen Seite zwar viel erreicht wurde, auf der anderen Seite aber der hohe Arbeitsumfang für viel Anspannung sorgte und infolge dessen auch den Wunsch nach einer besseren Aufteilung des Workloads im Team nach sich zog. Die mangelhafte Unterteilung resultierte dabei auch nicht zuletzt im Verwerfen eines Epics. Weiterhin wäre eine frühzeitigere Ansetzung der Schulungstermine dieses Sprints sicherlich sinnvoll gewesen, da eine Einarbeitungszeit und somit die Chance, sich selbst Erfahrung im neuen Thema zu erarbeiten, gegeben wäre. Damit hätten Zeitverluste , die durch diese Wissens-/Erfahrungslücken auftauchten (z.B. ständiges Nachfragen), vermieden werden können.

Da schnell offensichtlich wurde, dass der zu erwartende Arbeitsumfang den geringen Vorlesungsstunden im Kurs Projektmanangement überwiegt, wurde beschlossen, im Team zusätzlich flexibel zu arbeiten und bei Bedarf spontan zusätzliche Termine zwischen den Teammitgliedern zu organisieren.

# 4. Sprint

Sprint Planning:

Die wesentlichen Punkte des Sprints beziehen sich auf die schöne Gestaltung des Programmes, die Verfassung der Ausarbeitung, die Erstellung der Powerpoint Präsentation und eines Kahoots - ein Know-How Check am Ende der Präsentation - durch welches wir uns mehr Aufmerksamkeit während der Präsentation erhoffen, da die Mitschüler dieses Kahoot als interne Challenge sehen.

Um möglichst effizient arbeiten zu können werden die anstehenden Aufgaben auf die Teammitglieder aufgeteilt. Eine Person kümmert sich um die schöne Gestaltung des Programmes, zwei Personen werden die Powerpoint erstellen, die Ausarbeitung wird auf alle vier Personen aufgeteilt und das Kahoot wird noch von einer Person erstellt.

Sprint Verlauf:

Die Gestaltung des Programmes wurde in einem gemeinsamen Discord Anruf überarbeitet, was die Anpassung auf unsere gewählte Farbpalette und das Design-Theme beinhaltete. Auch die Anordnung der Knöpfe und die Fenstergröße waren Thema davon. Die Verfassung der Ausarbeitung wurde erst in folgende Punkte unterteilt: 1.Sprint, 2.Sprint, 3.Sprint, 4.Sprint, Lessons learned, Vorwort, Fazit und den Rest und anschließend auf uns vier Teammitglieder fair aufgeteilt.

Dabei fiel uns auf, dass die Zeit nicht mehr reicht, unser Programm benutzerfreundlich und ästhetisch zu gestalten.

Unsere Powerpoint Präsentation wurde in enger Zusammenarbeit mit dem Inhalt der Ausarbeitung erstellt. Die Folien wurden übersichtlich, mit Bildern gespickt, jedoch informativ erstellt. Das Kahoot wurde mit gemeinsamen, aus den Powerpoint Folien, erstellten Fragen entworfen.

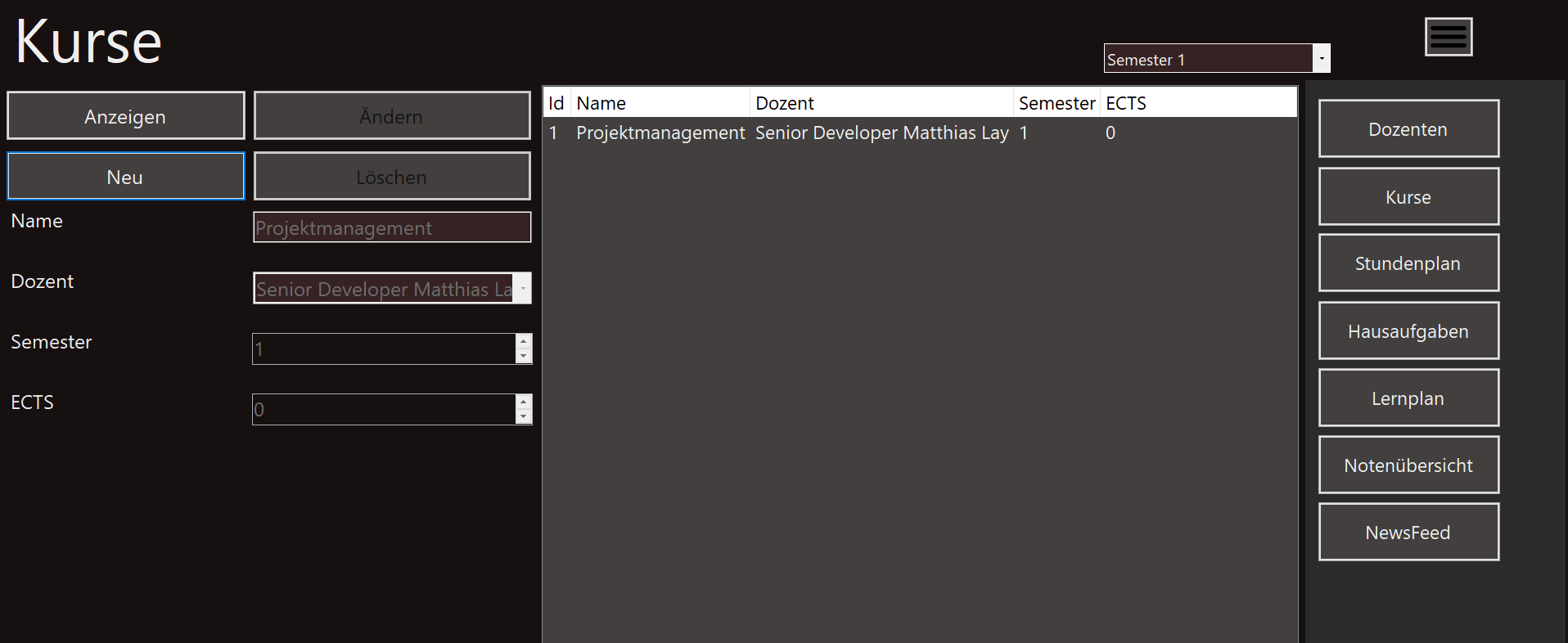
Sprint Review:

Der Sprint lief reibungslos und ohne große Komplikationen. Durch die im Voraus erstellte Aufteilung bezüglich der ToDos, war die Effizienz im Team sehr hoch und jeder konnte ohne auf einen anderen warten zu müssen arbeiten, was nicht in jedem Sprint gewährleistet war.

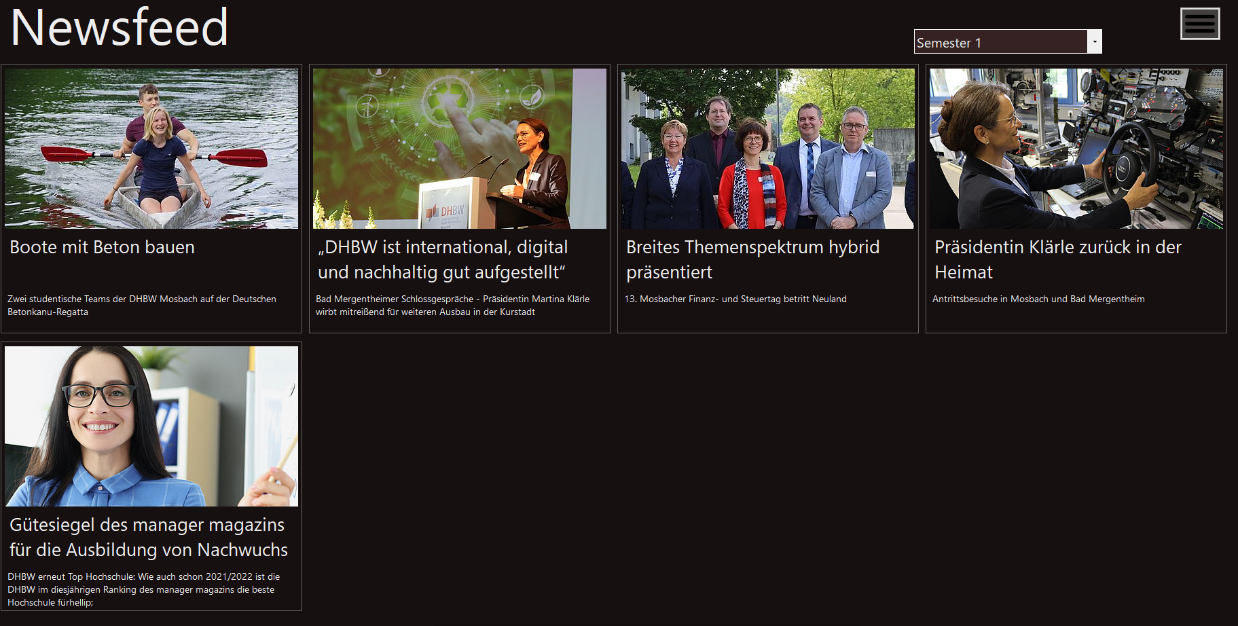
Im vierten Sprint wurde also der Feinschliff an unserem Programm vorgenommen, was die Benutzung nun sehr angenehm macht. Zudem wurde die Ausarbeitung und die Powerpoint, sowie das Kahoot erstellt.

# Produktvorstellung

Unsere Software wurde mit .Net6 in Windows Forms zusammen mit Entity Framework erstellt.

Die einzelnen Ansichten sind über die Seitenleiste (siehe Screenshot) erreichbar. Die Daten werden in einer lokalen Datenbank gespeichert, dadurch sind sie auch noch nach einem Neustart des Programmes erreichbar.

Der Newsfeed greift seine Informationen direkt von der DHBW – Webseite ab. Der Stundenplan wird direkt vom Intranet geladen.



Auf den anderen Ansichten kann der Benutzer individuelle Einträge einfügen, ändern und löschen.

Die folgenden Ansichten haben wir fertiggestellt: Dozenten, Kurse, Stundenplan, Hausaufgaben, Lernplan, Notenübersicht und NewsFeed.

# Lessons learned

Unsere Lessons learned während des Projektes bestehen vor allem aus drei Hauptthemen. Das erste Thema ist der Umgang mit Problemen und Verzögerungen bei der Entwicklung. Es gab wie zu erwarten einige Probleme, da nicht jeder Entwickler schon Erfahrungen mit den benutzten Tools hatte und sich zunächst teilweise einarbeiten musste.

Zu Beginn haben wir die Probleme zunächst versucht, selbst zu lösen, um die anderen nicht bei der Entwicklung abzuhalten mit der Hoffnung, zeitnah alleine eine Lösung zu finden. Allerdings haben wir nach einiger Zeit herausgefunden, dass es mehr Sinn macht, sich bei größeren Problemen mit mindestens einer anderen Person zusammenzusitzen und gemeinsam eine Lösung zu finden.

So kommt man meistens schneller zu einer Lösung und selbst wenn dann keine Lösung gefunden wird, kann man sich bezüglich des weiteren Vorgehens absprechen. Nachdem wir diese Vorgehensweise angewandt haben, konnten Probleme deutlich schneller behoben werden.

Das zweite Thema ist der Umgang mit agilem Projektmanagement. Zu Beginn haben wir uns noch relativ gut an die Vorgaben gehalten und die Rahmenbedingungen des Scrums befolgt.

Allerdings konnte diese klare Struktur nicht immer eingehalten werden, die Einhaltung der Rahmenbedingungen war von niedrigerer Priorität als das Entwickeln des Programms selbst. Dadurch dass dann einige Tools nicht auf dem neusten Stand gehalten wurden, gab es manchmal Schwierigkeiten, den Überblick zu behalten und einem roten Faden für die Entwicklung zu folgen.

Vor allem die Definition of Done wurde nicht immer eingehalten. Oftmals wurden zu implementierende Funktionen als "eigentlich fertig" bezeichnet, ohne dass diese tatsächlich ausgiebig getestet wurden. Hierdurch kam es dann bei späteren Sprints zu Problemen, bei bereits vermeintlich fertiggestellten Features festgestellt.

Das dritte Thema war der fehlende Fokus auf das Endprodukt. Hier wurde sich zu sehr auf die einzelnen Funktion fokussiert und das Design nicht mit angemessener Priorität behandelt. Zum Zeitpunkt dieser Erkenntnis war es jedoch schon zu spät, das UI ästhetisch zu gestalten. In Zukunft sollten wir uns nicht nur auf die Funktionalität konzentrieren, sondern immer zuerst grundlegende Dinge, wie das Design, festlegen.

Des Weiteren haben wir auch kleinere Lehren aus diesem Projekt gezogen. Es wurde uns allen mit der Zeit klar, dass es unvermeidbar ist, auf Probleme und Verzögerungen zu stoßen. Wir mussten lernen, dass manche Probleme gelöst werden können, aber manche auch zunächst ignoriert werden müssen, damit wir im Zeitfenster bleiben. Falls Probleme fürs erste umgangen werden können, ist dies manchmal nötig, damit diese nicht die Vollendung des Projektes in Gefahr bringen.

# Fazit

Alles in allem kann das Projekt „Simple Studies“ als ein Erfolg verbucht werden. Wir haben es innerhalb der gegebenen 8 Wochen geschafft, den größten Teil des geplanten Programms fertig zu bekommen. Funktional haben wir alle anfänglichen definierten Anforderungen erfüllt. Leider haben wir durch unseren Fokus auf die Funktionen den fast genauso wichtigen Teil der Präsentation des Programms vernachlässigt. So ist die Software zwar stabil und kann benutzt werden, jedoch ist das Design und die Usability noch sehr ausbaufähig.

Allgemein verliefen alle Sprints holperig. Zwar gab es wie zuvor beschrieben immer kleinere Schwierigkeiten, aber am Ende wurden die Features rechtzeitig fertig gestellt, die definierten Termine und Meetings wurden immer wie geplant ausgeführt und das Agile Mindset wurde nach und nach mehr gelebt. Die einzelnen Positionen haben ihre Aufgaben sorgfältig erledigt, auch wenn es hin und wieder etwas Hilfe von den anderen Mitgliedern gebraucht hat.

Die wichtigste Lektion, die wir hierbei gelernt haben, ist, das große Ganze immer im Auge zu behalten. Es bringt dem Programm nichts zwei bis drei Funktionen mehr zu haben, wenn dann keine Zeit mehr dafür bleibt die Software aufzuhübschen, sodass das Benutzen auch Spaß macht. Hier muss bei der Planung immer darauf geachtet werden, die Prioritäten im Sprint mit Blick auf die Deadline zu setzen.

Zusammenfassend wurde das Agile Projektmanagement mehr oder weniger befolgt. Probleme hat man während Sprints erkannt und diese in den Reviews angesprochen und gelöst. Die einzelnen Aufgaben hat jedes Teammitglied ernst genommen und erledigt. Am Ende haben wir es geschafft, in der Zeit eine lauffähige Software zu erstellen.