Inhalt

[1. Unsere Desktopanwendung 1](#_Toc40268323)

[2. Scrum 2](#_Toc40268324)

[2.1 Teambuilding und Organisation 2](#_Toc40268325)

[2.2 Scrum Vorbereitungen 4](#_Toc40268326)

[2.2.1 Definition of Ready 4](#_Toc40268327)

[2.2.2 Definition of Done 5](#_Toc40268328)

[2.3 Schätzgrößen 5](#_Toc40268329)

[2.4 Priorisieren 6](#_Toc40268330)

[2.5 Sprint Inkremente 7](#_Toc40268331)

[3. Die Desktopanwendung 7](#_Toc40268332)

[3.1 Produktvision und Anforderungen an das Produkt 7](#_Toc40268333)

[3.2 Auswahl der Programmiersprache 7](#_Toc40268334)

[3.3 Gestaltung der Anwendung 7](#_Toc40268335)

[3.4 Umsetzung 8](#_Toc40268336)

[3.5 Ergebnis 8](#_Toc40268337)

[3.6 Problemdokumentation 8](#_Toc40268338)

[3.7 Anleitung 8](#_Toc40268339)

[4. Retrospektiven 8](#_Toc40268340)

[5. Reflexionen der einzelnen Mitglieder 10](#_Toc40268341)

[6. Trelloboards 10](#_Toc40268342)

[7. Lop Liste 10](#_Toc40268343)

[8. Quellen 11](#_Toc40268344)

# Unsere Desktopanwendung

Die Aufgabe in unserem Projekt ist es eine Desktop-App für Sekretariate zu entwickeln und zwar ein Excel-Moodle-“Gateway“ für Noten und Feedback. Die Sekretäre an Hochschulen haben gegen Ende eines Semesters oft viel Arbeit mit dem Eintragen der Noten für die Studierenden. Im Moment läuft dies alles händisch ab. Das heißt die Dozenten geben im Sekretariat meist eine Excel-Tabelle ab, in der die Noten für alle Studierenden eines Kurses eingetragen sind. Teilweise sind dort auch nicht die endgültigen Noten angegeben, sondern Teilnoten. Diese Teilnoten müssen dann unter Berücksichtigung der Gewichtung noch verrechnet werden müssen. In Studiengängen mit viel Praxisbezug kann die Zahl der Einzelnoten auch schnell sehr groß werden. Dagegen wollen wir Abhilfe schaffen, indem wir eine Desktopanwendung entwickeln, die es einem Sekretär oder einer Sekretärin mit wenigen Schritten ermöglicht, Noten schnell und unkompliziert hochzuladen. Dazu haben wir uns einen genauen Plan gemacht, wie dies funktionieren soll. Wir erstellen Excelvorlagen, die die Dozenten nutzen können, um darin die Noten für ihre Studenten einzutragen. Dazu haben wir verschiedene Vorlagen angelegt, die wir in unserer App zum Download zur Verfügung stellen. Eine basiert auf Matrikelnummern, die andere auf Namen. Diese können in der Anwendung dann ausgewählt und ganz einfach per Klick in Moodle importiert werden. So ist es nicht notwendig die Noten für jeden Studenten einzeln einzutragen. Aus der Anwendung sollen diese dann einfach über eine eingegeben URL in Moodle zu dem richtigen Studierenden eingetragen werden. Das war unsere Grundidee, die wir durch unsere User Stories und innerhalb der einzelnen Sprint To-Do´s genauer definiert haben. Dadurch soll vor allem Sekretariaten das Leben erleichtert werden. Die Arbeit, die vorher viel Zeit gekostet hat, wird verkürzt und mit wenigen Klicks erledigt. So ist es auch möglich, dass Sekretäre ihren Fokus besser auf andere Dinge legen können. Wichtig für die Anwendung ist aber auch, dass Dozenten die vorgegeben Excel-Tabellen verwenden. Das erleichtert ihnen ebenfalls diesen die Arbeit und die Originaldateien sind in Zukunft auch einheitlich. Das Entwicklungsprojekt wird hier mit der agilen Projektmanagement-Methode Scrum umgesetzt. Da es aber eine vorgegebene Anzahl an Sprints gibt, ist unser Projekt in dem momentanen Stand noch nicht vollendet. Die Dokumentation dient deshalb auch dazu, dass daran weitergearbeitet werden kann und verstanden wird, was bereits gemacht wurde und wie wir dabei vorgegangen sind. Zunächst soll deshalb erläutert werden, wie wir uns innerhalb Scrum organisiert haben und im zweiten Teil dann näher auf die Umsetzung eingegangen werden.

# Scrum

## 2.1 Teambuilding und Organisation

Noch während der Vorlesung zu Scrum haben wir bereits Maßnahmen zum Teambuilding getroffen. Folgende Regeln haben wir für die Zusammenarbeit definiert:

* Respektvoller Umgang miteinander
* Ausreden lassen und zuhören
* Pünktlich kommen und Aufgaben erledigen (unmittelbar, rechtzeitig absagen)
* Offene Kommunikation
* Aufgaben sorgfältig erledigen
* Gleichberechtigung
* Kompromissbereitschaft
* 10 Minuten vor einem Meeting Internetverbindung checken und einloggen

Diese sind sehr wichtig, damit wir untereinander ein gutes Arbeitsklima schaffen. Außerdem helfen diese Regeln die Zusammenarbeit zu fördern und Probleme leicht zu beheben, beziehungsweise sie gar nicht erst entstehen zu lassen.

Zur Organisation der Dokumente und Kommunikationsmittel haben wir uns auch Gedanken gemacht. Als Kommunikationskanäle haben wir uns für WhatsApp, Discord, Whereby und Trello entschieden. Die Übersicht zeigt genauer, für was wir welchen Kanal benutzen möchten.

|  |  |  |  |
| --- | --- | --- | --- |
| WhatsApp | Discord | Whereby | Trello |
| Für spontane Treffen und kurze Fragen,  🡪 Erleichterung dieser Kommunikation | Gruppentreffen (Video, Screenshare), gespeicherter Chat | Gruppentreffen während Vorlesung und Treffen mit Stakeholdern | Organisation von Aufgaben, Übersicht |

Außerdem haben wir folgende Kanäle für unsere Dokumentenablage ausgewählt.

|  |  |  |
| --- | --- | --- |
| GitHub | E-Mail | Google Drive |
| Dokumentenablage, versionskontrolliertes Speichern, gleichzeitiges Arbeiten | Daten teilen, Dokumente nur für einzelne Mitglieder | Dokumentenablage/ - austausch |

Die Kommunikation über diese Kanäle hat sehr gut funktioniert. Jedes Mitglied war dadurch immer auf dem neuesten Stand und hatte immer die aktuelle Version des Projekts.

Als nächstes hat sich jedes Teammitglied seine Stärken vor allem in Hinsicht auf dieses Projekt überlegt. Daraus haben wir auch eine Übersicht erstellt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pascal Feinauer | Christian Dänzer | Jonas Althoff | Marco Scotellaro | Katharina Schmitt |
| Menschen überzeugen, Planung | Für Gruppe sprechen, Programmieren | Technik, Programmieren | Design, Präsentation | Organisation, Beratung |

Darauf aufbauend haben wir zum Schluss die Rollen des Scrum Teams auf unsere Mitglieder verteilt. Die Verteilung sieht folgendermaßen aus.

Product Owner: Pascal Feinauer

Scrum Master: Katharina Schmitt

Entwicklungsteam: Christian Dänzer, Jonas Althoff, Marco Scotellaro

## 2.2 Scrum Vorbereitungen

Zur Scrum Vorbereitung gehört auch das definieren der „Definition of Ready“ und der „Definition of Done“. Diese haben wir ebenfalls gemeinsam festgelegt, damit jedes Mitglied des Scrum-Teams das gleiche Verständnis davon hat.

## 2.2.1 Definition of Ready

Unsere User Stories sind unabhängig voneinander. Zusätzlich sind sie klein genug, damit sie innerhalb eines Sprints umgesetzt werden können. Dadurch haben wir zu jedem Release eine funktionsfähige Software und können in jedem Sprint ein Inkrement fertigstellen. (Sie können, wenn das Entwicklungsteam den Anspruch hat, vom Product Owner zerlegt werden.)

Sie sind ausreichend weit gefasst. Dadurch haben wir genug Spielraum, um zu besprechen, was genau entwickelt werden soll und wir können uns Änderungen vorbehalten. (Gespräch zwischen Entwicklungsteam und Product Owner)

Außerdem liefern alle unserer User Stories einen Wert. Daraus ergibt sich, dass sobald eine User Story überholt wird und diese somit keinen Wert mehr hat, sie gelöscht wird.

Unser Aufwand der Umsetzung ist gut zu schätzen, da wir genügend Informationen zu den Anforderungen unseres Stakeholders haben. (Herr Mester)

Wir haben zu jeder User Story mindestens ein Akzeptanzkriterium. Dadurch können wir überprüfen, ob die Umsetzung planmäßig erfolgt ist. So kann sichergestellt werden, ob es einen Wert für den Stakeholder hat.

## 2.2.2 Definition of Done

* User Stories wurden in Tasks geteilt
* Alle Akzeptanzkriterien wurden erfüllt
* Der Code ist vollständig implementiert und kommentiert
* Der Code ist immer aktuell auf Git
* Ein Code Review wurde durchgeführt
* Der Code wurde getestet und funktioniert
* Es gibt keine kritischen Fehler

Für die Definition of Done haben wir einmal einen Text und einmal Stichpunkte geschrieben. Dadurch konnten wir beides ausprobieren und sehen, womit wir besser zurechtkommen. Wir konnten mit beidem gut arbeiten. Jedoch waren die Stichpunkte etwas leichter. Hier konnten die einzelnen Punkte direkt erkannt werden, die erfüllt sein müssen.

## 2.3 Schätzgrößen

Außerdem gehörte zur Vorbereitung das Festlegen unserer Schätzgrößen. Dabei haben wir uns auf Gebäck geeinigt. Unsere Skala sieht folgendermaßen aus.

Cakepops

Sehr einfach

Sehr schwer

Macarons

Muffins

Cupcakes

Kuchen

Torten

Hochzeitstorte

Nach dieser Skala haben wir dann auch alle unsere User Stories geschätzt. Im Verlauf unseres Projekts haben wir außerdem noch Magic Estimation ausprobiert und auch hiermit nochmal neu geschätzt. Dabei sind wir aber auf gleiche, beziehungsweise sehr ähnliche, Ergebnisse gekommen. Ab dem zweiten Sprint haben wir dann beide Methoden verwendet und unsere „Gebäckskala“ noch um Zahlen der Fibonacci-Reihe erweitert:

1 - Cakepops

Sehr einfach

Sehr schwer

2 - Macarons

3 - Muffins

5 - Cupcakes

8 - Kuchen

13 - Torten

20 - Hochzeitstorte

## 2.4 Priorisieren

Für das Priorisieren der einzelnen User-Stories haben wir die Methode „Buy a Feature“ kennengelernt. Damit haben wir dann unsere User-Stories neu priorisiert. Jeder hatte hier zu Beginn 556 Geldeinheiten zur Verfügung. Dabei sind wir dann auf Folgendes Ergebnis gekommen, bei dem eine verkaufte User-Story durch den Namen des Mitglieds unseres Scrum-Teams markiert ist.

Als User möchte ich eine Vorlage haben, in die ich meine Noten eintragen kann.

10 Jonas

Als User möchte ich eine Anwendung mit einer ansprechenden Benutzeroberfläche haben, damit ich mich gut zurecht finden.

250

Als User möchte eine plattformübergreifende Desktopanwendung, damit ich das Programm benutzen kann.

800 Pascal und Marco

Als User möchte ich in der Anwendung problemlos auf den Darkmode wechseln können.

10 Jonas

Als User möchte ich eine Excel-Datei, die Noten enthält, in Moodle importieren, damit ich die in Excel stehenden Noten dem in Moodle hinterlegten Studenten für eine bestimmte Leistungserhebung nicht alle einzeln eintragen muss.

1000 Christian und Katharina

Als User möchte ich mich über die Anwendung in Moodle einloggen können, damit ich dann von dort aus den Import starten kann.

900 alle

Als User möchte ich Dateien in einem Auswahlfeld auswählen können und per Mausklick z.B. in Button-Form in Moodle exportieren können, damit ich nicht extra die Anwendungen verlassen muss.

100

Als User möchte ich Noten für nur einen Studenten eintragen können damit noch nicht finale Noten ausgespart werden.

700

Als User möchte ich Excel-Tabellen kombinieren können und die Prozentanteile der Noten zuteilen können, damit ich das nicht händisch machen muss.

400

## Sprint Inkremente

In unserem ersten Spint hatten wir mehrere Inkremente zum Vorstellen. Das erste war eine GUI in XD. Das zweite war die Umsetzung des MockUps mit HTML. Das dritte Inkrement war die Funktion in der Anwendung Excel-Tabellen auszuwählen.

Der zweite Sprint hatte als Inkrement das Hochladen der Noten in Moodle zum richtigen Studenten mit Anmeldevorgang aus der Desktopanwendung heraus. Zusätzlich haben wir einen dynamischen Darkmode vorgestellt. Dieses Inkrement hatte nicht die höchste Priorität. Da aufgrund von einer unerwarteten Verzögerung Kapazitäten an anderer Stelle zur Verfügung standen und der Darkmode schnell umgesetzt war, haben wir uns gemeinsam dafür entschieden, dieses Inkrement noch in den Sprint mit aufzunehmen.

Da wir in unserem zweiten Sprint auf massive Probleme mit der Schnittstelle von unserer Anwendung zu Moodle gestoßen sind, haben wir in Sprint 2 nicht alle User Stories zu Ende bringen können. Wir haben die angefangenen User Stories dann in den Sprint 3 mitaufgenommen. Aufgrund fehlender Unterstützung konnten wir diese User Stories allerdings nicht weiter bearbeiten. Deshalb hatten wir in Sprint 3 als Inkremente eine Fehlerdokumentation, die Ausarbeitung der Projektpräsentation und das Gestalten des Logos, da wir dies trotzdem bearbeiten konnten.

## Retrospektiven

Während unseren Sprints sind wir natürlich auch auf verschiedene Probleme gestoßen, die uns bei der Arbeit behindert haben. Diese haben wir immer in der Sprint Retrospektive analysiert und besprochen. Dabei sind wir in der ersten Retrospektive zu folgenden Ergebnissen gekommen.

Was ist gut gelaufen?

* Teamkommunikation
* Zeitliche Einhaltung
* Aufgabenverteilung
* Bestätigung der richtigen Rollenverteilung
* Jeder hatte gleiches Verständnis der User Stories

Was ist schlecht gelaufen?

* Verständnisproblem mit Herr Mester 🡪 mehrmals Sprint Planning neu machen
* Sprint Backlog nicht immer direkt nach Erledigung der Aufgabe verändert
* Product Owner keine Idee was als nächstes kommen soll
* Termine zu spät vorbereitet

Was wollen wir ändern?

* Entwicklungsteam soll Sprint Backlog immer aktuell halten und direkt anpassen
* Product Owner vor Sprint Review schon Gedanken über nächsten Schritt machen

Wie setzen wir Änderung um?

* Jeder Entwickler seine Aufgaben direkt anpasst und weiterverschiebt, wenn es erledigt ist 🡪 dazu in WhatsApp Erinnerungen an Entwicklungsteam geben
* Product Owner soll sich vor Sprint Review nochmal das Product Backlog anschauen

Was wollen wir beibehalten?

* Kommunikation beibehalten
* Schauen, dass wir weiterhin zeitlich im Rahmen bleiben

Die Veränderungen haben wir gut umgesetzt. Es waren keine Erinnerungen mehr an das Entwicklungsteam nötig, da die fertiggestellten To-Do’s nun selbstständig weitergeschoben wurden. Der Product Owner hatte sich vor dem nächsten Review nochmal mit dem Product Backlog vertraut gemacht.

In der Retrospektive 2 hatten wir dann dieses Ergebnis:

Was ist gut gelaufen?

* Teamarbeit
* Kommunikation
* Aufteilung Aufgaben und Hilfe
* Abnehmen von Aufgaben, wenn einer zu viel hat
* Fehlersuche und Alternativen finden
* Hilfe angefragt

Was ist schlecht gelaufen?

* Externer Support
* Überforderung mit Umsetzung
* Ziele nicht erreicht
* Wissen noch nicht genau, wie wir weitermachen sollen

Was wollen wir ändern?

* Zu Alternativen direkt Beispiele zeigen
* Nachfragen wie es weitergehen soll

Wie setzen wir Änderung um?

* Direkt in Kontakt mit Herr Mester treten

Was wollen wir beibehalten?

* Kommunikation im Team
* Gute Zusammenarbeit

Auch diese Änderung haben wir umgesetzt. Wir haben noch am gleichen Tag den Kontakt mit dem Stakeholder gesucht. Leider hatten wir aufgrund verspäteter Rückmeldung keine Einflussmöglichkeiten mehr auf ein erfolgreiches Projektergebnis.

In unserer letzten Retrospektive sind wir dann zu Folgendem Ergebnis gekommen:

Wir haben es hierbei immer geschafft unsere Änderungen in unserem Handlungsrahmen umzusetzen.

# Die Desktopanwendung

## Produktvision und Anforderungen an das Produkt

Generell hatten wir als Vision den Umgang mit Scrum, das Vertraut machen mit dem JavaScript Framework ElectronJS, das Erstellen einer Präsentation und eines Projektberichts sowie das eigentliche Ergebnis: das Erstellen einer Software, um Excel-Dateien smarter in Moodle zu importieren.

## 3.2 Auswahl der Programmiersprache

Die Anforderungen an das Projekt sind kurz gesagt:

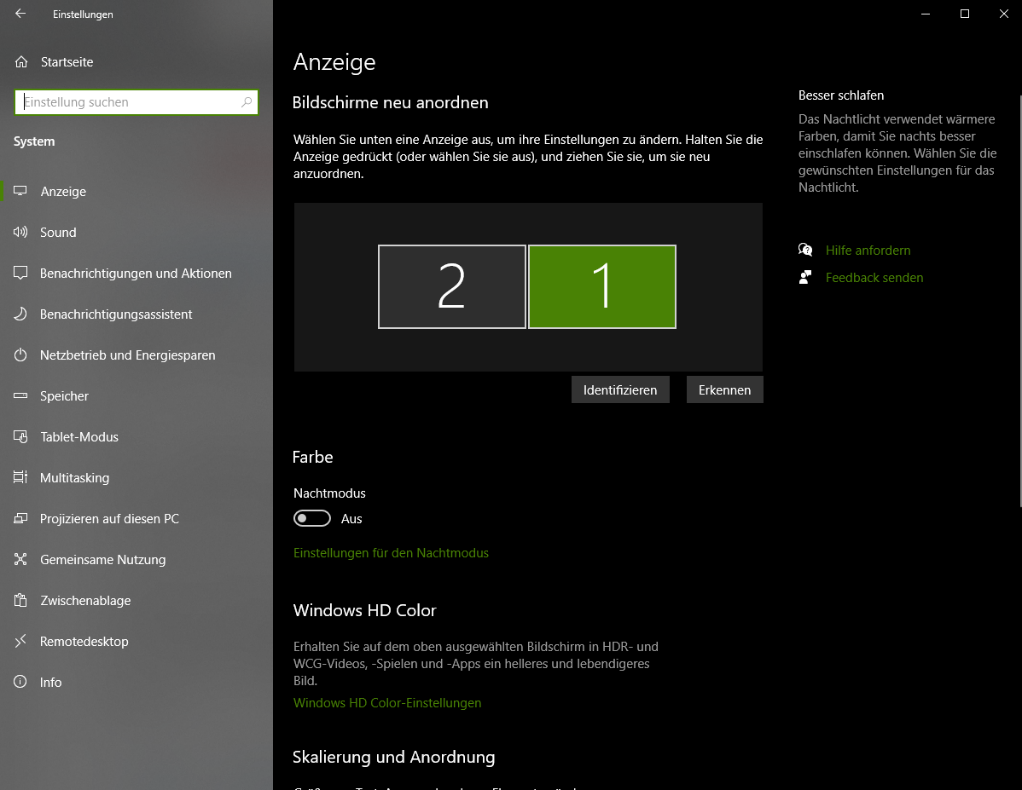
* Desktop-App erstellen
* plattformübergreifende Anwendung
* mit JavaScript, TypeScript, StencilJS oder Java umsetzen
* Zugriff auf Dateisystem muss ermöglicht werden.

Nach ein wenig Recherche hat sich ElectronJS als das passende Framework herausgestellt.[[1]](#footnote-1) Die Apps können mit HTML, CSS und JavaScript entwickelt werden. Das fällt genau in den Bereich rein, mit dem wir bereits gelernt haben umzugehen. Die Anwendungen können für Windows, Linux und MacOS entwickelt und sie können auf allen Plattformen ausgeführt werden.[[2]](#footnote-2) Außerdem kann man mit ElectronJS Daten im Dateisystem lesen und schreiben – einem ausschlaggebenden Punkt, da wir mit den Excel-Tabellen arbeiten müssen. Außerdem gibt es eine wachsende Community und eine gute Dokumentation. Wir hatten vor dem Projekt keine Erfahrung mit dem Framework. Daher war ein guter Support im Internet ein wichtiger Aspekt.

## 3.3 Gestaltung der Anwendung

Bevor die Gestaltung des Entwicklerteams begann, überlegte sich das Team, welchem Designstil sie folgen möchten um die Anwendungsoberfläche (GUI) optisch ansprechend und neumodisch zu gestalten. Zur Auswahl standen zahlreiche Design Stile aber entschieden hat sich das Team für das Flat-Design. Wir als Team, haben uns für diesen Stil bewusst entschieden, da unserer subjektiven Meinung nach ein minimalistischer Gestaltungsstil optisch äußerst ansprechend für den Nutzer ist. Dieser Gestaltungsstil steht im Gegensatz zum Skeuomorphismus, welche nicht das flache und anschaulich schlichte Design unserer Desktopapplikation fördern würde.

Das Flat Design ist äußerst erfolgreich heutzutage und findet seine Wiederverwendung z.B. bei Windows (in der nachfolgenden Abbildung wird ein Screenshot aus den Einstellungen zur Verdeutlichung veranschaulicht).

Weitere sehr bekannte Software Firmen wie Apple sind ebenfalls wegweisende Repräsentanten des Designstils.

Mit der Auswahl des Designstils begann für das Entwicklungsteam die Ausarbeitung der einzelnen Elemente in einem sogenannten Style-Tile. Wir haben uns als Entwicklungsteam für die Software Adobe Illustrator entschieden aufgrund der zahlreichen grafischen Features des Programms und aufgrund dessen, da alle Beteiligten der Gruppe T6 das Programm installiert auf dem Rechner besitzen.

Zuerst befassten wir uns mit der Farbauswahl für die Desktop Applikation. Wir haben uns hier für eine einen schwarz-weißen Primärstil entschieden. In ihm sind sowohl sehr dunkle Grautöne als auch helle Weißtöne enthalten. Sie sollen insgesamt einen stimmigen Kontrast bilden, welcher allerdings kein Absolut Kontrast sein sollte, da wir eine harmonischere Farbempfindung herbeirufen möchten und dem Auge möglichst den absoluten Kontrast ersparen wollen.

Da allerdings der reine schwarz-weiß Kontrast ein sehr eintöniges Anschauungsbild ergibt und man mit diesem Kontrast wichtige Dinge nicht ausreichend hervorheben kann, entschieden wir uns Akzentfarben einzubauen mit denen wichtige Dinge in Form von Buttons, Boxen oder Icons hervorgehoben werden können. Die Akzentfarben bei uns sind zum einen die Farbe Rot und die Farbe Grün. Die Farbe Rot haben wir ausgewählt, da sie als Signalfarbe angesehen wird und sehr oft auf wichtiges hindeutet. Beispielweise wird die Farbe oftmals mit Gefahrenzeichen verbunden. Dies zieht die Aufmerksamkeit des Nutzers auf sich. Die Farbe Grün findet seine Anwendung nur bei einem Haken, welcher erscheint, wenn der Upload auf Moodle erfolgreich abgelaufen ist. Grün also als Semantisches Zeichen für den Erfolg, da unsere Gesellschaft schon seit der Schulzeit beigebracht wird, dass Grün für richtig und Rot für falsch steht.

Als nächstes befassten wir uns mit der Schriftauswahl. Schnell wurde ich deutlich bewusst, dass die Verwendung einer seifenlosen Schrift für die Applikation von essenzieller Bedeutung ist. Zum einen handelt es sich um das Flat Design, welches ein neumodisches und aktuelles Design ist. Zum anderen ist die Applikation am Computer verfügbar. Auf einem Computer wird vorzugsweise eine serifenlose Schrift gewählt, da diese besser zu lesen ist. Mit der Entscheidung der der Klassifikation haben wir uns dann für die Schrift „Roboto“ entschieden. Diesen Schriftstil haben wir vor allem aufgrund seiner Schlichtheit gewählt. Wir haben uns hier bewusst für die Quelle Google Fonts entschieden, da diese Schriften einfach per link in das HTML eingebaut werden kann und daher von jedem Rechner aus betrachtet werden kann. Wir haben uns nun für die Verwendung von insgesamt zwei Schriftschnitten entschieden. Regular für die Hauptüberschriften und Light für Unterüberschriften und die Mengentexte. Hier wird ebenfalls durch die Abstufung von Schriftgrößen immer in 10pt Schritten dafür gesorgt, dass der Nutzer sofort den Unterschied zwischen den Abstufungen erkennt und sich so auf der Anwendungsseite zurechtfinden kann oder sich ggf. an Haltepunkten orientieren kann.

Des Weiteren haben wir im Style-Tile wichtige Elemente festgehalten. Wir haben die verwendeten Tabellen konstruiert um zu visualisieren wie das Ergebnis sein soll. Hier haben wir uns ebenfalls für ein sehr schlichtes Design entschieden allerdings mit Trennlinien. Diese sollen dem Nutzer die Möglichkeit geben sich in den Zeilen und Spalten zu orientieren um so seine Einträge sicher und korrekt zu bewerkstelligen.

Für das Design des / der Buttons haben wir uns passend zum Flat Design für eine platte Oberfläche im nicht aktiven Modus geeinigt. Wir haben uns hier für ein soliden Rahmen entschieden, welche keine Füllfarbe besitzt. Der textuelle Inhalt des Buttons ist im Style-Tile nur anhand eines Beispiels vorhanden, da sich dieser eben von Situation zu Situation verändern kann. Ebenso gibt es mehrere Farben für den Button zur Auswahl. Zum einen die Auswahl für einen roten Button, einen hellgrauen Button oder einen dunkelgrauen Button.

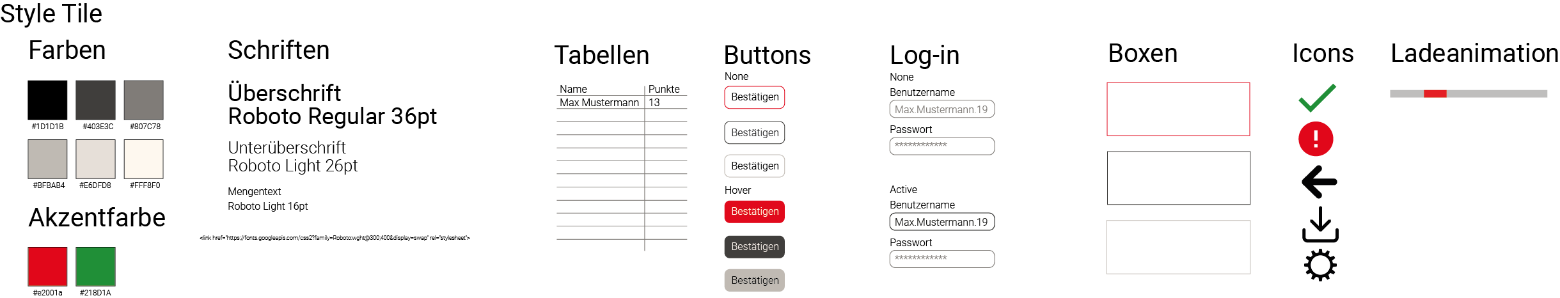
Der Button iss jeweils ein Button mit runden Ecken, da diese unserer subjektiven Meinung nach einer größeren Ästhetik aufweisen als eckige Button.

Diese Buttons besitzen sogenannte Hover-Effekte. Diese werden aktiv, wenn der Nutzer seinen Mauszeiger über den Button bewegt und verändert damit das Styling des Buttons. Hier bekommt der jeweilige Button eine hintergrundfüllende Farbe, welche dem Rand entspricht. Ebenso verändert sich hier die Schriftfarbe nach weiß, bzw. schwarz, je nachdem welche füllende Farbe der Button besitzt sodass die Schrift einen leserlichen Farbkontrast besitzt und so optisch besser entziffert werden kann.

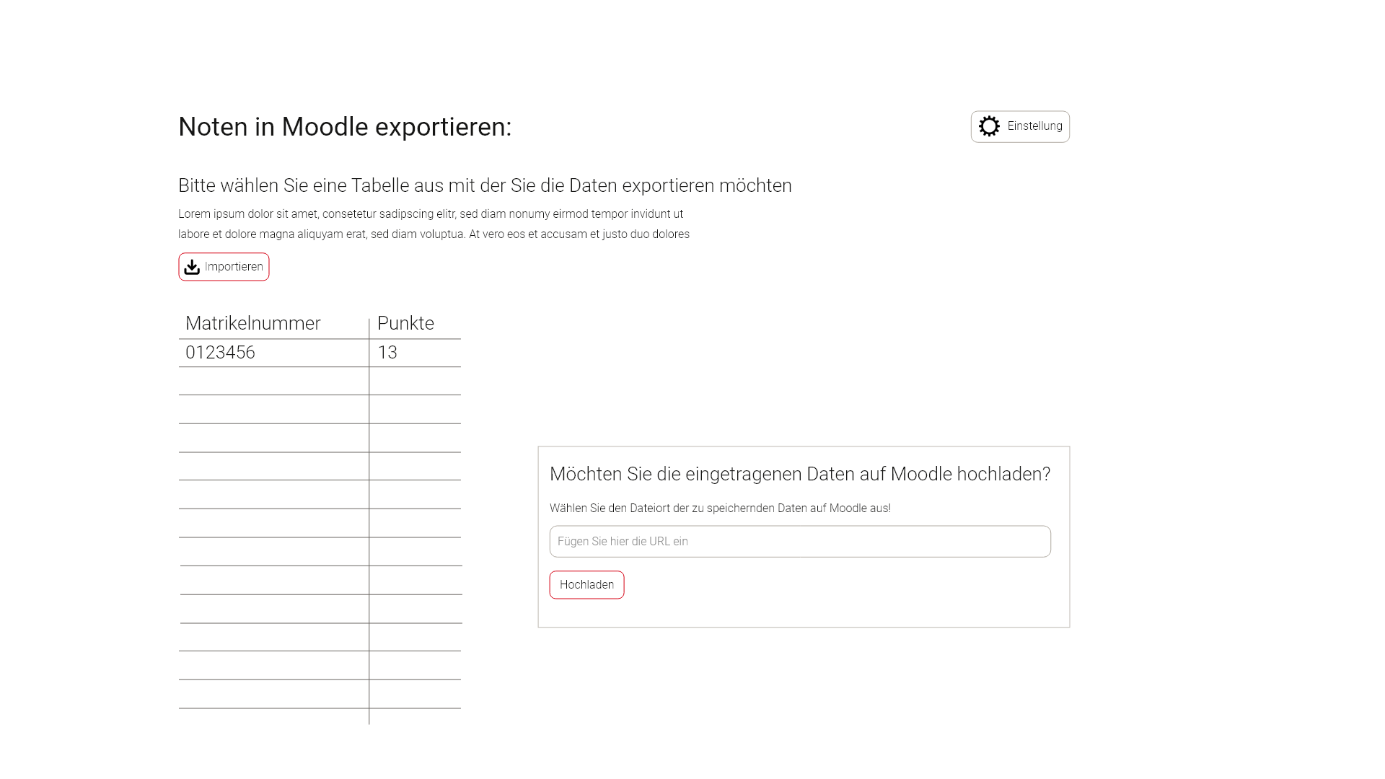
Als nächstes haben wir uns um den Log-In gekümmert. Wir wollten auch hier ein schlichtes Design im Stil des Flat Design. Hierbei haben wir und für einen hellgrauen Rand entschieden. Allerdings haben wir auch hier eine Animation eingebaut, sodass sich eine optische Hervorhebung ergibt aufgrund einer Farbänderung des Inhalts des Formulars sowie eine Farbänderung des Formularfeldrahmens.

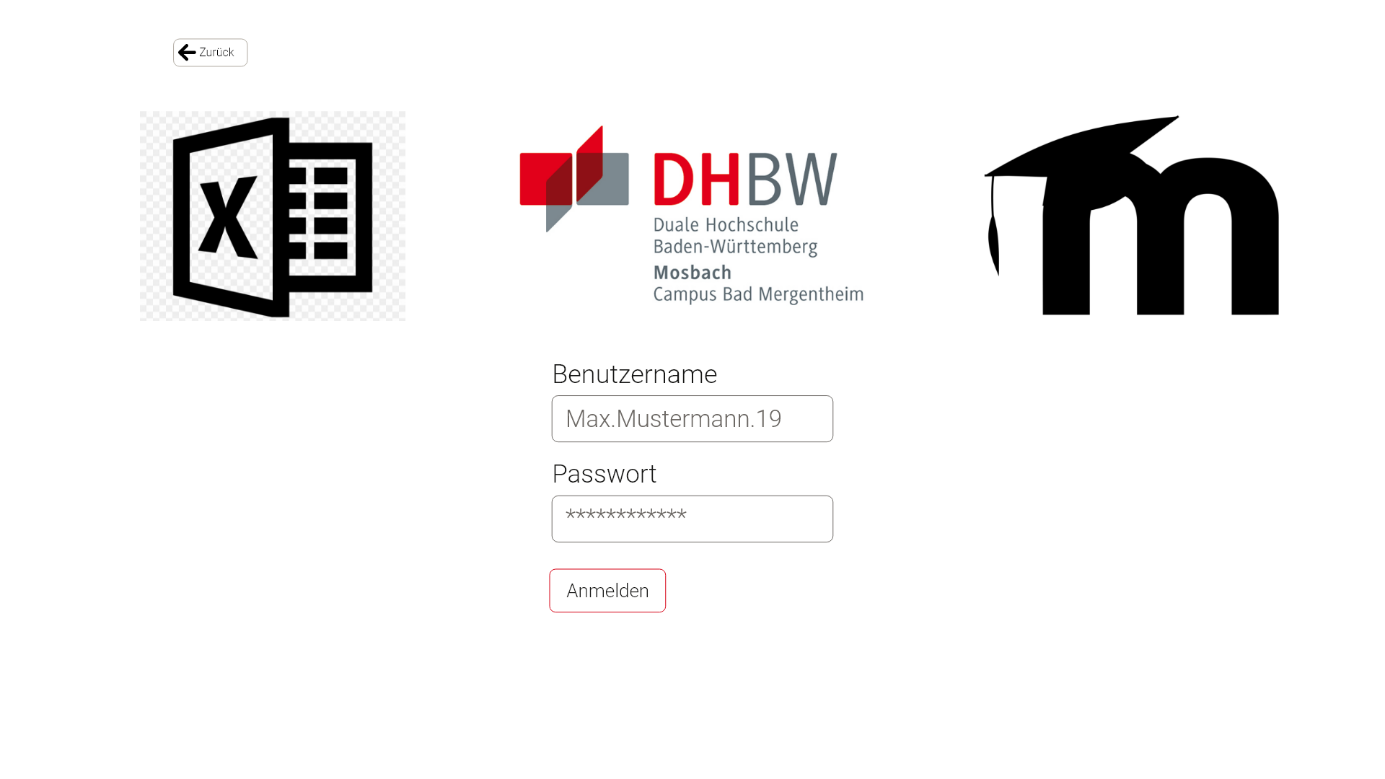
Weiter haben wir uns um Boxen gekümmert. Aber warum haben wir die Boxen mit spitzen Ecken ausgestattet, wenn wir die Buttons bewusst mit runden Ecken versehen haben? Der Grund für diese Entscheidung ist, dass die Boxen ein Kontrast zu den Buttons bilden sollen um für Abwechslung zu sorgen. Die Farben sind auch hier ein Hellgrau, ein Dunkelgrau und ein Rot. In diesen Boxen sollen wichtige Inhalte platziert werden, welche die Aufmerksamkeit des Nutzers bündeln sollen.

Als nächstes haben wir uns dazu entschieden eine Auswahl an Icons zu erstellen, welche verschiedene repräsentative Ziele haben sollen. Der Hauptgrund für die Verwendung von Icons ist die visuelle Unterstützung und die visuelle Sprache der Applikation. Wir haben diese in Buttons angebracht um die Wahrnehmung von Schrift durch ein passendes Bild zu ergänzen und dem Nutzer durch die direkte Zuordnung mithilfe des Gesetzes der Erfahrung auch die Bedienung zu erleichtern. Die verwendeten Farben sind Schwarz, Grün und Rot.

Die letzte optische Definition haben wir in Form von einer Ladeanimation definiert. Während des Uploads der Daten auf Moodle sollte dem Nutzer eine Animation zur Verfügung stehen, sodass der Nutzer bei der Wartezeit eine kleine Unterhaltung erhält. Hier wurde ein Balken mit einem inneren kleinen Balken verwendet, welcher sich immer abwechselnd von links nach rechts und umgekehrt bewegt. Zusätzlich dazu haben wir noch Texte definiert, welche in einem bestimmten Zeittakt auf dem Bildschirm zu sehen sind, sodass der Nutzer unterhalten wird (Untenstehend ist das komplette Style-Tile angefügt).

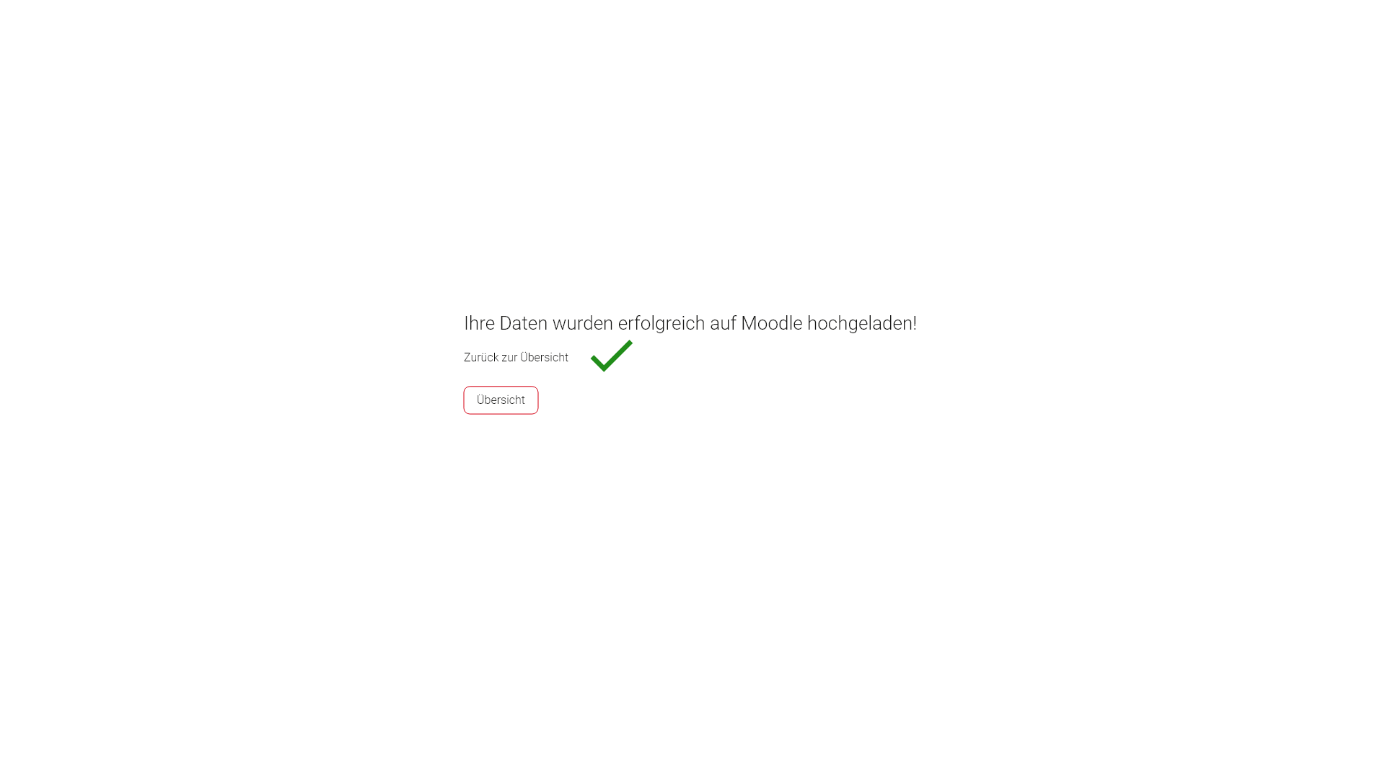
Die rein lose Umsetzung in Illustrator war für uns als Team jedoch nicht aussagekräftig genug. Daher entschieden wir uns dazu ein Mock-Up mit der Software Adobe XD zu erstellen. Wir haben uns auch hier für ein Adobe Produkt für die Umsetzung entschieden, da die Software sehr gut mit der Software Illustrator zusammenarbeitet und so einzelne Elemente übernommen werden können. Wir werden hier im Bericht allerdings nur die Funktionen der einzelnen Seiten kurz ansprechen, da die Optik schon durch das Style-Tile durch die Erklärung der einzelnen Elemente erfolgte. Hier wird lediglich ggf. ein noch nicht erwähnter Teil hinzufügend ergänzt. Ebenfalls zu sagen ist, dass die jetzige Erklärung das Mock-Up beschreibt und in der Anwendung einige Parts verändert wurden.

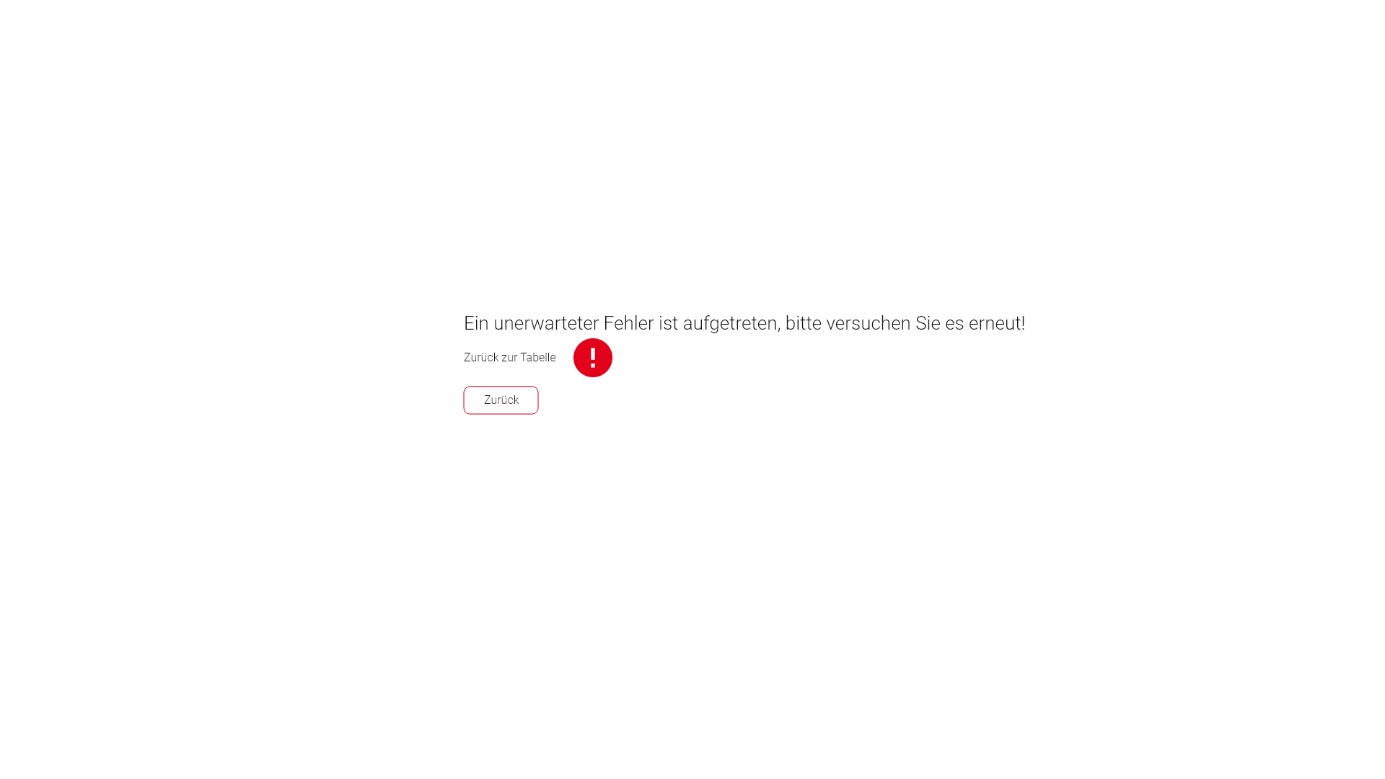
Die Idee der Startseite sollte wieder sehr spartanisch schlicht sein. Zu sehen ist die Funktion eine Exceltabelle vom Programm „einlesen“ zu lassen. Dann kann man die Ziel URL angeben und das Ganze hochladen. Ein Hauptfokus liegt auch hier auf dem Weißraum. Dieses stilistische Mittel soll ebenfalls für eine Seriosität im Hinblick auf die Optik ausstrahlen. Der Spruch „Weniger ist mehr!“ begleitet uns durch die Designumsetzung. 

Als nächstes wurde die Log-In Seite entwickelt. Hier wurden Platzhalter Icon für das Logo eingesetzt, da das Logo erst in einem späteren Teil entwickelt wurde. Hier haben wir darauf geachtet, dass sich der Fokus auf die Mitte des Bildschirms begrenzt und die Anmeldung groß und deutlich erfolgen kann. 

Darauf hin haben wir uns um den Ladescreen gekümmert. Ebenfalls ist hier der Inhalt auf der optischen Mitte platziert um eine angenehme visuelle Gestaltung zu ermöglichen. 

Die letzte Seite soll mit folgender Logik ausgestrahlt werden:

1. Wenn der Upload der Daten erfolgreich verläuft, soll dem Nutzer eine Seite dargestellt werden, welche den Erfolg wiederspiegelt. 

2. Wenn der Upload fehlschlägt, soll dem Nutzer eine Seite dargestellt werden, welche einen Fehler beim Hochladen ausgibt. 

Die Entwicklung des Logos beruht auf einfachen Gegebenheiten. Wir als Gruppe T6 haben uns auf den Namen „The Gate“ für die Anwendung geeinigt. Damit einhergehend stand die Logogestaltung auf dem Zeitplan. Das Logo besitzt Pfeile, welche für einen Up- und Download stehen. Darüber hinaus eine Wolke, so sollte die Datenübertragung als Medium visualisiert werden und einen Schlüssel. Dieser Schlüssel soll der Schlüssel zur Anwendung des „Tors“ sein. Somit ist das Icon z.B. in der Taskleiste der Schlüssel zu unserem Tor. Darunter steht nochmals der Name der Anwendung. Dieser in Verbindung mit dem Bild soll also zur Wort-Bild-Marke beitragen, um so die Erinnerung der Nutzer und den Wiedererkennungswert zu stärken und zu festigen.

## 3.4 Umsetzung

## Bei der Umsetzung galt es zunächst einmal ein Projekt aufzusetzen. Für das Öffnen eines neuen Electron-Fensters sowie das Schließen der Anwendung gibt es bereits Code-Snippets, die verwendet werden können. Diese mussten wir nur minimal auf unsere Bedürfnisse anpassen. Im Anschluss wurden die ersten Seiten mittels HTML und CSS aufgebaut. Diese haben sich dann an der in XD entworfenen GUI orientiert. Für den JavaScript-Code haben wir uns einzelne Bausteile überlegt und diese dann jeweils mit einzelnen Dateien voneinander abgegrenzt. Durch passende Dateinamen war es uns somit besser möglich, den Überblick zu bewahren. Bausteine waren das Hochladen der zu importierenden Excel-Datei, das Bearbeiten der Tabelle, die die Namen und Matrikelnummern enthält, der Log-In in Moodle, das Suchen des Namens in der Tabelle, das Eintragen der Note beim jeweiligen Studenten, das Klicken der Button um zur nächsten Seite zu kommen und das Generieren eines Ladebalkens mit Texten zur Unterhaltung. An den Stellen, die einen Zugriff auf die Moodle-Website benötigt haben, sind wir leider nicht weitergekommen. Das wird allerdings später bei der Problemschilderung genauer ausgeführt. Um trotzdem kleinere Funktionen testen zu können, und weil wir noch auf Unterstützung gehofft haben, haben wir die benötigten Moodle-Seiten nachgebaut. Dabei haben wir auf die gleiche Benamung der Ids und die gleiche Verschachtelung der Tags geachtet.

## 3.5 Ergebnis

Mit screenshots

## Problemdokumentation

Dies ist auch das Inkrement unseres dritten Sprints.

## Anleitung

Da unsere Anwendung noch nicht fertig ist, lässt sich diese auch noch nicht leicht installieren. Deshalb sieht dieser Weg im Moment folgendermaßen aus:

1. Herunterladen des git-Repositories
2. Dieses mit zum Beispiel Visual Studio Code öffnen
3. Danach über eine Eingabe in einem Terminal Electron mit npm install –g electron installieren
4. In den Ordner 01\_App navigieren und Befehl npm install eingeben
5. Dann npm start eingeben und die Anwendung öffnet sich

In unserer Anwendung sind dann folgende Schritte zu machen, die aber auch in der Anwendung selbst zu lesen sind:

1. Zuerst muss eine Tabelle ausgewählt werden, die hochgeladen werden soll und die Noten der Studierenden enthält. Der Button dafür ist direkt auf der ersten Seite zu finden
2. Um diese dann auch Moodle hochzuladen, muss in dem dafür vorgesehenen Feld, auf derselben Seite, die Ziel-URL eingegeben werden und anschließend der Button „Importieren“ gedrückt werden
3. Danach kommt man auf die Seite, die die Log-In-Daten für Moodle abfragt. Diese müssen hier eingegeben werden und danach der Button „Anmelden“ geklickt werden.
4. Dann sollte ein Ladebildschirm erscheinen der erkenntlich macht, wann die Daten hochgeladen sind.

Im Moment funktioniert nur Schritt 1 wie angegeben. Die restlichen Punkte lassen sich auch ausprobieren und bedienen, aber im Moment wird man weder in Moodle angemeldet, noch werden Daten dort hochgeladen. Damit diese Schritte funktionieren, hätten wir mehr Sprints benötigt. Nach unseren Rahmenbedingungen sind wir hier aber nicht bis dorthin gekommen.

# Reflexionen der einzelnen Mitglieder

## 4.1 Einzelreflexionen

## 4.1.1 Reflexion Katharina Schmitt

Ich war in unserem Scrum-Projekt der Scrum Master. Dabei habe ich versucht, dem Team so gut wie möglich Scrum verstehen zu geben. Dazu gehört insbesondere das Moderieren unserer Scrum Artefakte und die Einhaltung der zeitlichen Vorgaben. Das war nicht immer so einfach, da wir gerade am Anfang des Projekts im Daily Scrum etwas länger als die 15 Minuten gebraucht haben. Ich habe außerdem versucht Probleme anzusprechen und diese bis zu einer Lösung voranzubringen. Dabei habe ich viel gelernt, das mich auch persönlich weitergebracht habe. Gerade die Aufgabe, die einzelnen Meetings zu moderieren, fiel mir am Anfang etwas schwer. Jedoch konnte ich in jedem Treffen etwas dazulernen. Dadurch wurde unsere Zusammenarbeit auch immer besser und wir hatten nach kurzer Zeit schon einen geregelten Ablauf. Generell ist mir aufgefallen, dass unsere Gruppenarbeit sehr gut funktioniert hat. Wir haben sehr viel kommuniziert und dadurch Verständnisfragen schnell klären können. In meiner Rolle als Scrum Master konnte ich außerdem auch immer wieder Tipps geben, wie Scrum genau umzusetzen ist. Während das Entwicklungsteam sich um den Code gekümmert hat, habe ich versucht diesen die Arbeit um das Projekt herum so gut es geht abzunehmen. Dadurch konnten sich meine Kommilitonen besser auf ihre Aufgaben fokussieren und wir haben immer die Strukturen und Rollen des Scrum Teams beibehalten. Damit unser Team immer gut auf die einzelnen Meetings vorbereitet ist, habe ich ihnen auch eine Übersicht erstellt, wer für was genau innerhalb des Scrum-Teams zuständig ist und in welchem Meeting welche Unterlagen gebraucht werden. Hier gab es am Anfang nämlich Probleme, da nicht jeder genau wusste für was welches Meeting nötig ist. Diese konnte ich so aber schnell beseitigen. Außerdem habe ich dem Team nochmal eine Zusammenfassung generell über Scrum erstellt, damit jeder auf einen Blick schnell das finden kann, bei dem er sich noch nicht so gut auskennt. Doch auch neben Scrum habe ich technisch etwas dazugelernt. Unser Entwicklungsteam hat uns immer auf dem neuesten Stand gehalten. Dadurch konnten alle den Code verstehen. Für mich war dies sehr hilfreich, da ich mich in diesem Teil noch nicht so gut auskenne. Ich habe aber dadurch auch ganz neue Möglichkeiten kennengelernt.

## 4.1.2 Reflexion Pascal Feinauer

## 4.1.3 Reflexion Marco Scotellaro

Ich war ein Teil des Entwicklungsteams und habe dort fungiert als Chief of Design. Meine Hauptaufgabe bestand darin, alles rum das Thema Design zu entwickeln. Daher konnte ich meine Stärke gut in das Projekt einbringen und meine Gruppenmitglieder von meiner Stärke profitieren lassen. Die ständigen Rücksprachen und Kontrollen von Gruppenmitgliedern haben dafür gesorgt, dass das ganze Team auf einem Stand ist und auch bespricht, ob alle dem Designvorschlag zustimmen. Ich habe meiner Kreativität versucht freien Lauf zu lassen bei der Gestaltung, da für mich dadurch dieses Projekt ein Ausgleich zum Technik-lastigen Semester ist. Für mich persönlich hat das Arbeiten in der Gruppe super geklappt, was ich hier nochmal positiv erwähnen möchte. Ohne unsere stetige Kommunikation und Absprachen hätten wir vermutlich noch weniger die Chance erhalten das Projekt mit einem erfolgreichen Ergebnis abzuschließen. Wir hatten bei der Entwicklung der Anwendung jedoch enorme Probleme, jedoch zeigt die Offline-Tauglichkeit der Applikation, dass der Code funktioniert und das Problem bei Moodle liegt. Ich hatte zuvor in diesem Stil mit ElectronJS noch keinen Berührpunkt. Durch das Erklären des Codes von meinen Teamkammeraden Chris und Jonas habe ich aber viel mitnehmen können, da mein Wissenstand davor bei null war. Das Thema agiles Projektmanagement mit Scrum durfte ich in den vergangenen Wochen lernen, was mir persönlich enorm viel weiterhilft, da meine Ausbildungsfirma momentan ein Projekt bearbeitet, welches auch in Sprints abläuft. Dadurch verstehe ich nun Abläufe besser und kenne die Hintergründe hinter bestimmten Meetings und Zeitintervallen der Sprints. Nicht so gut gelaufen ist für mich relativ, allerdings gab ein HTML- und CSS-Problem, welches mich Stunden gekostet hat. Jedoch kann ich nun mit der Lösung und der einhergehenden Erkenntnis das Projekt beenden.

## 4.1.4 Reflexion Christian Dänzer

Als Mitglied des Entwicklungsteams habe ich hauptsächlich am Code der Anwendung gearbeitet. Zu Beginn war ich noch sehr zuversichtlich, da es zu ElectronJS und JavaScript viele Quellen im Internet gibt und ich auch schon einiges mit JavaScript umgesetzt habe. Nachdem ich dann nach einiger Zeit aber an mehreren Stellen in einer Sackgasse gelandet bin, deren Ende jedes Mal Moodle hieß, wurde meine Hoffnung auf einen erfolgreichen Abschluss des Projekts immer geringer. Da wir jedoch ohne programmiertechnische Unterstützung dastanden – obwohl wir diese mehrfach angefordert haben – wurde mir während des dritten Sprints klar, dass wir kein funktionierendes Projekt vorstellen können. Das hat mich sehr stark unter Druck gesetzt und verunsichert. Da wir aber eine super Projektgruppe mit einer gut funktionierenden Kommunikation waren, haben wir das beste daraus gemacht: Wir haben für eine ausführliche Problemdokumentation gesorgt, damit es nachfolgenden Entwicklern leichter fällt. Trotz der Schwierigkeiten konnte ich sehr vieles Lernen. Ich konnte meine Programmierkenntnisse weiterverbessern. Das geschah zum einen durch das Programmieren selbst, zum anderen aber auch durch das Erklären des Codes, damit jeder im Team diesen versteht. Außerdem konnte ich vieles über Scrum lernen. An vielen Stellen fand ich die Arbeit mit dieser Projektmanagement-Methode sinnvoll, an manchen Punkten habe ich es jedoch nur als zusätzlichen Arbeitsaufwand empfunden. Auch in den Präsentationen konnte ich weitere Dinge dazulernen. Durch die Sprint Reviews kamen mehrere Präsentationen vor Kunden dazu, in denen ich gelernt habe, mit Kritik am Objekt umzugehen.

## 4.1.5 Reflexion Jonas Althoff

## 4.2 Unsere Erfahrungen und Tipps für die Zukunft

Wir als Gruppe hatten eine sehr gute Zusammenarbeit. Dies liegt vor allem daran, dass wir sehr viel und ausführlich miteinander geredet haben. Dazu zählt natürlich auch das Einhalten der Daily Scrums, da wir dabei sehr schnell Probleme erkennen konnten. Deshalb ist es immer wichtig eine offene und stetige Kommunikation im Team zu haben. Außerdem haben wir den Tipp sich schon vor dem Projekt mit dem Stakeholder zu treffen. Fragen können direkt geklärt werden und ein eventuelles Neuplanen eines Sprints wird vermieden. Diesen Fehler haben wir am Anfang gemacht und auch in der Retrospektive festgestellt. Dadurch haben wir das auch nur einmal nicht beachtet und konnten in den weiteren Sprints problemlos anfangen zu arbeiten. Außerdem sollte man sich bei Problemen immer direkt um Hilfe kümmern. Dabei haben wir leider nicht immer die erwartete Hilfe bekommen und mussten uns selbst versuchen zu helfen. Auch wenn wir zu diesem Zeitpunkt schon an dem Punkt waren, an dem wir selbst nicht mehr weitergekommen sind und schon mehrere Lösungswege ausprobiert haben. Deshalb haben wir den Tipp sich direkt bei Fehlern externe Hilfe zu suchen.

## 4.3 Erfahrungen mit virtueller Gruppenarbeit

Wir haben remote sehr gute Erfahrungen machen können. Unsere Kommunikationstools Discord und Whereby haben immer funtkioniert und wir hatten damit keine technischen Probleme. Wichtig ist auch hier die gute und stetige Kommunikation, die bei uns sehr gut funktioniert hat…Dabei waren wir vor allem auf die Screenshare-Funktion der Anwendungen angewiesen. Denn ohne diese lässt es sich sehr schwer zusammen am Code arbeiten, über Designs diskutieren oder Logos besprechen. Auch das pünktliche Erscheinen zu verabredeten Terminen ist wichtig. Hierbei muss im Vergleich zur realen Welt eben nicht Staus, sondern mögliche technische Probleme, als Verzögerung eingeplant werden. Mit unseren Gruppenregeln haben wir aber dafür vorgesorgt.

# Trelloboards

Scrum Backlog:

<https://trello.com/b/pH1fFIXw/scrum-backlog>

Backlog Sprint 1:

<https://trello.com/b/68dD38lM/backlog-sprint-1>

Backblog Sprint 2:

<https://trello.com/b/gV3w8Bs3/backlog-sprint-2>

Backlog Sprint 3:

<https://trello.com/b/AvTnkseR/backlog-sprint-3>

# Lop Liste



# Quellen

Die für Entwicklung verwendet wurden

https://docs.sheetjs.com/

https://www.electronjs.org/docs

https://stackoverflow.com/

1. 5 Best JavaScript Frameworks For Desktop Apps, 2018. . Blog Brainhub.eu. URL https://brainhub.eu/blog/javascript-frameworks-for-desktop-apps/ (zuletzt aufgerufen am 05.13.20). [↑](#footnote-ref-1)
2. Electron | Plattformübergreifende Desktop-Anwendungen mit JavaScript, HTML und CSS entwickeln, n.d. URL https://www.electronjs.org/ (zuletzt aufgerufen am 05.13.20). [↑](#footnote-ref-2)