

DIPLOMARBEIT

Gesamtprojekt
DigitalSchoolNotes

User- und Rollenmanagement

Philipp Adler 5BHIT

Betreuer: Michael Borko, Bakk. techn.
Mag. DI.(FH) Christoph Brein

Datenmanagement

Selina Brinnich 5BHIT

Betreuer: Michael Borko, Bakk. techn.
Mag. DI.(FH) Christoph Brein

Infrastruktur und Testing

Niklas Hohenwarter 5BHIT

Betreuer: Michael Borko, Bakk. techn.
Mag. DI.(FH) Christoph Brein

Optical Character Recognition

Adin Karic 5BHIT

Betreuer: Michael Borko, Bakk. techn.
Mag. DI.(FH) Christoph Brein

Parallel Working System

Thomas Stedronsky 5BHIT

Betreuer: Michael Borko, Bakk. techn.
Mag. DI.(FH) Christoph Brein

ausgeführt im Schuljahr 2015/16

Abgabevermerk:

Datum: 1. April 2016

übernommen von:

- Ⓐ Ausdruck
- Ⓔ Formulierungen

Ziffern ausschreiben

! Das Schriftbild innerhalb eines Dokumentes ändert sich immer wieder - ist das der Drucker od. ein Fehler - bitte prüfen!

Generell:

Ich find das vorliegende Dokument ist eigentlich ganz gut formuliert, ich habe mir aber Mühe gegeben auch überall entsprechendes hinzuschreiben.


Ein bisschen weniger ITP - Screen am Beginn und dafür mehr technisches im Hauptteil hätte mir besser gefallen, aber wie gesagt nicht schlecht.


Lg Brein

Eidesstattliche Erklärung

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

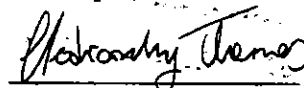
Wien, am 31.03.2016


Philipp Adler


Selina Brinnich


Niklas Hohenwarter


Adin Karic


Thomas Stedronsky

Danksagung

Wir wollen uns zunächst bei jenen Menschen bedanken, die auf ihre herzliche und unterstützende Art an der erfolgreichen Umsetzung dieses Diplomprojekts beigetragen haben. Ohne sie hätten wir bei manchen Problemen vielleicht doppelt so lange gebraucht und hätten wahrscheinlich nur halb so viel Spaß an der Umsetzung gehabt.

Wir wollen vor allem auch unseren Familien und Freunden danken. Im Zuge dieses Diplomprojektes sind einige Herausforderungen auf uns zugekommen, die wir ohne die Motivation und Unterstützung aus dem Familien- und Freundeskreis nicht so gut gemeistert hätten. Wir möchten besonders unseren Eltern danken, die uns auch nach langen Arbeitssitzungen stets wieder aufgerappelt haben.

Ein besonderer Dank geht an unsere beiden Betreuer, Prof. Borko und Prof. Brein. Ohne die zielgerichteten Verbesserungsvorschläge und die technische Beratung von Prof. Borko wäre unser Weg zur Fertigstellung des Produkts ein deutlich längerer gewesen. Prof. Brein stand uns mit seiner langjährigen Erfahrung und benutzerorientierten Sicht zur Seite.

Abschließend wollen wir uns auch noch bei unserem äußerst motivierten Abteilungsvorstand Prof. Koppensteiner und der Abteilung für Informationstechnologie bedanken. Der Abteilungsvorstand hatte immer ein offenes Ohr für uns, egal mit welchen Anliegen wir zu ihm kamen. Wir haben uns, im Zuge unserer fünfjährigen Ausbildung am TGM, immer als ein Teil dieser Familie gesehen und werden auch in Zukunft mit Stolz auf unsere Zeit hier zurückblicken.

„Wer aufhört, besser zu werden, hat aufgehört, gut zu sein.“

-Philip Rosenthal

Gender Erklärung

Aus Gründen der besseren Lesbarkeit wird in dieser Diplomarbeit die Sprachform des generischen Maskulinums angewendet. Es wird an dieser Stelle darauf hingewiesen, dass die ausschließliche Verwendung der männlichen Form geschlechtsunabhängig verstanden werden soll.

Abstract

Nowadays most people can't think of a world without internet anymore. It is used almost always and everywhere you are - also in schools. Students are getting more and more into digital notes, especially in technical oriented schools, where they have their laptop always with them. There is only one problem: These digital notes are often written in different programs. They are unorganized and mostly left in some directory where they aren't opened once afterwards.

The project DigitalSchoolNotes was formed exactly because of this problem. Our webapplication can organize digital notes and make it easier to write them. It is made from students themselves to fit the needs of students optimal. The application can be accessed from almost every device - personal computers, laptops and even tablets. Furthermore it is possible to use a mobile phone to take a photo and put it into the digital notes easy and conveniently. Particularly fitted for technical schools there is also the possibility to insert some code-snippets into the digital notes, which are correctly formatted and highlighted.

All of which is to allow students to write tidy, organized, digital notes and to make learning from these easier and more efficient.

Kurzfassung

Die heutige digitale Welt ist aus den Köpfen der meisten Menschen gar nicht mehr wegzudenken. Das Internet wird beinahe immer und überall verwendet - so auch immer mehr in Schulen. Vor allem in technischen Schulen werden Mitschriften aus dem Unterricht immer öfters digitalisiert. Das Problem dabei: Diese Mitschriften werden meist in unterschiedlichen Programmen verfasst, sie sind unorganisiert und werden oft in irgendeinem Ordner ohne jemals wieder angesehen zu werden.

Das Projekt DigitalSchoolNotes setzt genau bei diesem Problem an. Unsere Web-Applikation soll das Führen einer digitalen Mitschrift einfacher und organisierter machen. Dabei wollen wir speziell auf die Bedürfnisse der Schüler eingehen. Der Zugriff auf die Applikation ist sowohl von Desktop Systemen und Laptops, als auch von Tablets über eine Webseite möglich. Ein eingeschränkter Zugriff ist zudem für Handys möglich, um beispielsweise Tafelbilder schnell und bequem in die Mitschrift einfügen zu können. Speziell für technisch orientierte Schulen gibt es auch die Möglichkeit, Teile von Programmcode richtig formatiert erstellen zu können.

Das alles dient dazu, Schülern eine ordentliche, organisierte, digitale Mitschrift zu erleichtern und das Lernen aus diesen effizienter zu gestalten.

~„and even“ wäre sogar

Inhalt

1	Einleitung	1
2	Problembeschreibung	2
2.1	Umfeldanalyse	2
2.2	Projektidee	3
2.3	Projektkoordination	4
2.3.1	Kurzeinführung in Scrum	6
2.3.2	Scrum im Team	11
3	Stand der Technik	15
3.1	Frameworks	15
3.1.1	Web Frameworks	15
3.1.2	JS-Frameworks	18
3.1.3	CSS-Frameworks	21
3.1.4	Element-Frameworks	24
3.1.5	GUI-Testing	26
3.2	Technologien	28
3.2.1	MySQL	28
3.2.2	PostgreSQL	29
3.2.3	MongoDB	30
3.2.4	Couchbase	31
3.2.5	Vergleich	32
4	Design	33
4.1	Software-Architektur	33
4.1.1	Ablauf	33
4.1.2	DSN Architektur	36
4.2	Graphische Oberfläche	38
4.2.1	Layout	38
4.2.2	Einfache Menüstruktur	39
4.2.3	Unterstützende Bildzeichen	40
4.2.4	Heftansicht	41
4.3	Javascript Optimierung	43
4.3.1	Code Style	43
4.3.2	Debugging	44
4.3.3	Scripteinbindung	45
4.3.4	Optimierung	45
5	Implementierung	46
5.1	Infrastruktur und Testing	46

5.1.1	Infrastruktur	46
5.1.2	Testing	53
5.2	User und Rollenmanagement	56
5.2.1	Authentisierung	56
5.2.2	Userbereich	60
5.2.3	Berechtigungen	61
5.2.4	Adminbereich	61
5.3	Datenmanagement	63
5.3.1	Persistierung von Daten	63
5.3.2	Datenmodell	63
5.3.3	Datenzugriff	66
5.4	Heftelemente	71
5.4.1	Textelement	71
5.4.2	Codeelement	74
5.4.3	Bildelement	77
5.5	Parallel Working System	78
5.5.1	Bestehende Systeme	78
5.5.2	Umsetzung	80
5.5.3	Probleme	85
5.5.4	Ausblick	85
5.6	Optical Character Recognition	87
5.6.1	OCR-Verfahren	88
5.6.2	ICR und IWR	92
5.6.3	Implementierung	92
5.6.4	Ausblick	97
6	Auswertung und Benchmarks	98
6.1	Benutzerakzeptanz	98
6.2	Änderungsvorschläge	99
7	Ausblick	100
7.1	Bezahlsystem	100
7.2	Einblendung von Werbung	101
7.3	Zeichenelement	101
7.4	Video einbindung	101
7.5	Drag and Drop von Bildern	101
7.6	PDF Download	102
7.7	Shortcuts	102
7.8	Filehoster-Anbindung	102
7.9	Web Untis Anbindung	103
7.10	LDAP-Anbindung	103
8	Zusammenfassung	104

1 Einleitung

Am Anfang des Diplomprojektes Digital School Notes (DSN) stand der Ideenfindungsprozess. Das Team wollte an einem sinnvollen Diplomprojekt arbeiten, also ein Projekt, welches nachher weitergeführt werden und verkauft werden könnte. Durch diese Art von Projekt versprach sich das Team eine höhere Arbeitsmoral.

Die Idee kam während einer normalen Unterrichtsstunde. Mal wieder schrieb niemand mit und ~~auch~~ ~~kaum~~ ~~passte~~ ~~einer~~ auf. Man hatte schon länger versucht eine Idee für ein Diplomprojekt zu finden und auf einmal war es doch so offensichtlich. Das Diplomprojekt musste den Alltag in der Schule verbessern. Da kam die Idee für Digital School Notes.

Was, wenn mitschreiben auf einmal wieder interessanter ist? Wenn die Mitschrift am Laptop geführt wird und nicht mehr am Papier? Das würde so vieles einfacher machen. Ein digitales Schulheft. Sofort begannen wir die technische Machbarkeit zu prüfen. Die Software müsste eine Web Applikation sein um alle Betriebssysteme zu unterstützen. Sie müsste leicht zu bedienen und komfortabel sein. Es müsste möglich sein mehrere Schulen zu unterstützen, um eine große Community aufzubauen. Und natürlich müsste die Software attraktiv genug sein, um zahlende Kunden anzulocken und somit die Kosten der Infrastruktur decken zu können.

Nachdem die Idee fest stand wurde ein Lastenheft verfasst. Mit diesem Lastenheft hat das Team dann mehrere Lehrer kontaktiert, um weitere Ideen und Verbesserungsvorschläge einzuholen. Die Resonanz war gut und das ~~motiviert~~ ~~das~~ Team umso mehr. Das Projekt wurde genehmigt und die Implementierung begann.

Diese Diplomarbeit fasst die Arbeit am Projekt zusammen. Sie gibt Aufschluss über die Probleme und Lösungen, an welchen das Team gearbeitet hat.

) Formulierung
(F)

motiviert

2 Problembeschreibung

Im Schulunterricht mitzuschreiben ist wichtig, jedoch immer weniger Schüler machen dies auch. Falls sie doch mitschreiben, dann verschwindet der Zettel oder Collegeblock meistens, oder wird nie wieder angesehen. Es ist kompliziert den in immer größeren Teilen digital ablaufenden Unterricht auf Papier festzuhalten und seine Mitschriften zu organisieren. Als Resultat dieser Umstände stirbt die händisch geführte Mitschrift langsam aber sicher aus.

Statt der händischen Mitschrift wird die digitale Mitschrift am im Unterricht verwendeten Laptop immer beliebter. Diese hat allerdings bis heute ähnliche Probleme. Es ist kompliziert die Struktur des Tafelbildes in z.B. ein Word-File zu übernehmen. Außerdem passiert es häufig, dass das File nicht mehr gefunden wird, verloren geht oder einfach einen unpassenden Dateinamen hat, welcher nicht die entsprechende Mitschrift innerhalb der Datei vermuten lässt. Mit aktuellen Mitteln ist es ebenfalls mühsam digital mitzuschreiben.

?

2.1 Umfeldanalyse

Es existiert bereits einiges an Software, welche sich mehr oder weniger dafür eignet digital mitzuschreiben. Die Features der einzelnen Produkte sind großteils bekannt, deshalb werden diese nur oberflächlich beschrieben ~~werden~~:

- **Microsoft Word:** Am weitesten verbreitete Textverarbeitungssoftware; kostenpflichtig; auf Windows, Mac, Android & iOS verfügbar
- **Libre Office Writer:** Textverarbeitung; OpenSource; kostenlos; auf Windows, Mac & Linux verfügbar
- **OpenOffice Writer:** Textverarbeitung; OpenSource; kostenlos; auf Windows, Mac & Linux verfügbar
- **Kingsoft WPS Writer:** Textverarbeitung; kostenlos oder kostenpflichtig; auf Windows, Mac, Android & iOS verfügbar; sehr stark an Microsoft Word angelehnt
- **Microsoft OneNote:** Notizprogramm; Notizen können beliebig platziert und angeordnet werden; Organisation in Notizbüchern; gleichzeitiges Bearbeiten möglich; auf Windows, Mac, Android & iOS verfügbar; kostenlos
- **Google Docs:** Textverarbeitung; kostenlos; Betriebssystemunabhängig (Browser); gleichzeitiges Arbeiten möglich

1) Wenn etwas nicht mehr gefunden wird ist es verloren!

- 4) Dienen behält hatten wir schon
2) Konformen ausschreiben usw. → und so weiter
z.B. ...

2.2 Projektidee

Die Idee hinter dem Diplomprojekt „Digital School Notes“ war es eine Web-Applikation zur Führung einer digitalen Mitschrift zu entwickeln. Der Zugriff und das Bearbeiten soll von Desktop Systemen, Laptops und Tablets über eine Website möglich sein. Des Weiteren soll die Applikation mit Ausnahme der Heftbearbeitung auch auf Handys verfügbar sein. Das System sollte eine hohe Verfügbarkeit und gleichzeitig, für Schüler optimiert, eine leichte Bedienung bieten.

Es sollte ein System entwickelt werden, welches extra auf die Bedürfnisse von Schülern angepasst ist. Das Produkt soll es ermöglichen, ganz einfach und schnell verschiedene externe Medien, wie zum Beispiel Bilder, Links auf andere Webseiten oder Code-Snippets in ein „Heft“ einzufügen und zu verwalten. Durch eine sehr simpel gestaltete Benutzeroberfläche sollte die Bedienung zum Einfügen solcher Medien, im Vergleich zu anderen Produkten auf dem Markt, deutlich vereinfacht werden. Das Anbieten von verschiedenen Dateiformaten, sowie der sichere Umgang mit den Daten der Nutzer war ebenso ein Anliegen.

Durch einen eigenen Stundenplan sollte das Rundumpaket, welches wir den Schülern bieten wollen, zusätzlich bereichert werden. Der Nutzer kann im Stundenplan die Dauer, sowie die Anfangs- und Endzeit der jeweiligen Unterrichtseinheiten festlegen. Des Weiteren kann er dann die entsprechenden Einheiten im Stundenplan eintragen. Diese Angaben sind beliebig oft änderbar. Beim Klick auf eine der eingetragenen Unterrichtseinheiten soll sich dann automatisch das dazugehörige Heft des Users öffnen (wenn zuvor ein Heft dieser Unterrichtseinheit bzw. diesem Fach zugeteilt wurde).

Schüler sollten einfach Bilder hochladen und den Text aus den Bildern automatisch in ihr Heft einfügen können. Dies sollte mit Optical Character Recognition (optische Zeichenerkennung) ermöglicht werden. Das OCR-System sollte ein Bild des Users entgegennehmen, dieses mit OCR-Algorithmen analysieren und schließlich den aus dem Bild erkannten Text in das Heft einfügen. Somit soll den Schülern ein vereinfachtes Einfügen von gedruckten Angaben oder Hausübungen, welche dann direkt bearbeitbar sind, ermöglicht werden.

Da Gruppenarbeiten und Referate im schulischen Bereich keine Seltenheit sind, wurde auch an eine Optimierung in dieser Hinsicht gedacht. Durch ein eigenes „Parallel Working System“ sollte das gemeinsame Arbeiten für Schüler deutlich vereinfacht werden. Jeder Schüler soll die Möglichkeit haben, einige seiner Hefte mit anderen Schülern zu teilen. Diese geteilten Hefte können dann von allen dazu berechtigten Nutzern bearbeitet werden. Ein besonderes Augenmerk ist dabei auf die Konsistenz der bearbeiteten Hefte zu legen. Durch verschiedene Sperrmechanismen muss die Integrität der Daten in den geteilten Heften gesichert werden.

Um mit dem Produkt Geld zu verdienen, sind nach einer kostenlosen Probezeit von 90 Tagen kostenpflichtige Benutzerkonten vorgesehen. Die Nutzer sollen also nach der Probezeit einen geringen monatlichen Betrag entrichten, um das Produkt nutzen zu können. Zusätzlich ist der Vertrieb des Produkts an Schulen angedacht. Diese würden dann einen gewissen Betrag pro erworbener Lizenz zahlen, wobei diese Lizenz dann nicht zeitlich begrenzt ist und das Hosting auf den Servern der jeweiligen Schule stattfindet. Als zusätzliche Einnahmequelle könnte man noch, in Kooperation mit verschiedenen Werbenetzwerken, Werbung auf der Webseite schalten. Dabei wäre, aufgrund der vergleichsweise langen Verweildauer, die Seite der Heftansicht ein interessanter Platz für eine klassische Bannerwerbung oder Ähnliches.

2.3 Projektkoordination

Eine gut durchdachte Projektkoordination ist die Voraussetzung für die erfolgreiche Umsetzung eines Projektes. Dabei kann man generell zwischen zwei Arten des Projektmanagements unterscheiden: dem klassischen Projektmanagement und dem agilen Projektmanagement.

Das klassische Projektmanagement zeichnet sich durch am Projektbeginn klar formulierte Ziele und die Unterteilung des Projektes in sogenannte Meilensteine und weitgehend voneinander abgegrenzte Phasen aus.

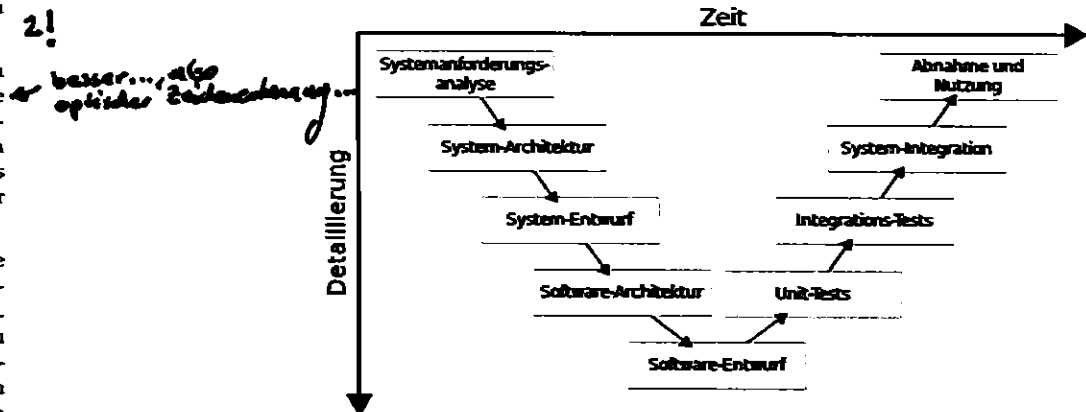


Abbildung 1: Phasen des klassischen V-Modells

Das agile Projektmanagement geht davon aus, dass die meisten Softwareentwicklungsprozesse zu komplex sind, um diese auf klassische Art und Weise zu planen. Agile Ansätze zählen also auf die Dynamik der verschiedenen Projektparameter, wie zum Beispiel das Umfeld oder die Projektziele. Einige bekannte Vorgehensweisen des agilen Projektmanagements sind SCRUM, TDD (test-driven development) oder XP (Extreme Programming).

Alle agilen Entwicklungsmethoden haben aber eine gemeinsame Basis: das agile Manifest. Das agile Manifest ist eine Ansammlung von agilen Prinzipien, welches erstmals 2001 einer breiten Öffentlichkeit zugänglich wurde. In dem Manifest legen die damaligen Erstunterzeichner die agilen Handlungsgrundsätze für eine erfolgreiche agile Softwareentwicklung fest.

„Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

- Individuen und Interaktionen ^{Sind} mehr als Prozesse und Werkzeuge
- Funktionierende Software ^{ist} mehr als umfassende Dokumentation
- Zusammenarbeit mit dem Kunden ^{ist} mehr als Vertragsverhandlung
- Reagieren auf Veränderung ^{ist} mehr als das Befolgen eines Plans

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.“ [uA01]

Mit der Annahme, dass Softwareentwicklungsprozesse empirische Prozesse sind, wurde für die Durchführung des Projektes „Digital School Notes“ das agile Projektmanagement gewählt. Da die Softwareumgebung selten vollständig definiert ist, die Anforderungen veränderlich sind und das Wissen und Können für den besten Lösungsansatz oft unvollständig ist, wird ein agiles Vorgehen einem klassischen vorgezogen. Konkret wurde für die Projektkoordination die agile Projektmanagementmethode SCRUM angewendet.

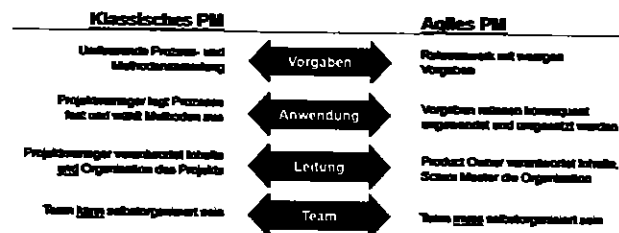


Abbildung 2: Klassisches PM vs. Agiles PM

2.3.1 Kurzeinführung in Scrum

Scrum bedeutet übersetzt „Gedränge“ und stellt ursprünglich einen Spielzug im Rugsbysport dar. Mit der Analogie zum „Gedränge“ wollten die beiden Ideengeber Ikujiro Nonaka und H. Takeuchi darauf anspielen, dass kleine, vernetzte Teams die besten Resultate liefern. Scrum ist eine Umsetzung des „Lean Developments“ (schlanke Entwicklung) für das Projektmanagement.

2.3.1.1 Rollen in Scrum

** Fremdworte erklären! ev. in einer kurzen Tabelle*

Scrum definiert als agiler Ansatz drei unterschiedliche Rollen: den Product Owner, den Scrum Master und das Entwicklungsteam.

Product Owner

Der Product Owner ist hauptsächlich für die zu implementierenden Funktionen, sowie den wirtschaftlichen Erfolg des Produkts verantwortlich. Er entscheidet maßgeblich, welche Eigenschaften das Produkt aufweist und welche nicht. Dabei stehen die Befriedigung der Bedürfnisse der Stakeholder, sowie der maximale wirtschaftliche Nutzen des Produktes im Mittelpunkt. Der Product Owner erstellt, priorisiert und entscheidet über die Produkteigenschaften und schließlich wann diese umgesetzt werden. Zur Festlegung dieser Eigenschaften erstellt der Product Owner ein Product Backlog. Eine weitere wichtige Aufgabe des Product Owners ist der ständige Kontakt mit den Stakeholdern des Produkts, sowie die Vertretung der Interessen selbiger.

Scrum Master

Der Scrum Master ist für die erfolgreiche Umsetzung von Scrum als Projektmanagementmethode verantwortlich. Er definiert die Regeln des Scrum und achtet auf deren Einhaltung. In enger Zusammenarbeit mit dem Entwicklungsteam ist er dafür verantwortlich, dass Hindernisse, die eine erfolgreiche Umsetzung gefährden könnten, aus dem Weg geräumt werden. Der Scrum Master stellt gegenüber dem Entwicklungsteam eine „dienende“, situative Führungskraft dar und erteilt selbst keine Arbeitsanweisungen.

Entwicklungsteam

Das Entwicklungsteam ist in erster Linie für die Implementierung der vom Product Owner festgelegten Produktfunktionalitäten zuständig. Ein Entwicklungsteam besteht meistens aus drei bis neun Mitgliedern. Es organisiert sich weitgehend selbst und hält die Qualitätsstandards aufrecht. Eine interdisziplinäre Besetzung des Entwicklungsteams ist essentiell, um die in den Sprints anstehenden Aufgaben möglichst unabhängig voneinander lösen zu können. Die Teammitglieder selbst sind neben speziellen Fähigkeiten aber auch Generalisten, da sie bei Ausfällen eines Teammitglieds dessen Arbeit fortführen können müssen. Eine entsprechende Dokumentation sollte hier hilfreich sein. Eine wichtige Aufgabe des Entwicklungsteams ist zudem die Schätzung des Aufwands der verschiedenen Einträge im Product Backlog. Dies wird üblicherweise mit Story Points realisiert, welche für eine bestimmte Zeitdauer stehen.

Wie zuvor angeführt wurde ich zunächst eine Mini Tabelle mit allen Scrum Begriffen, deren Übersetzung und Miniertklärung bringen

2.3.1.2 Artefakte in Scrum

Product Backlog (Übersetzung) * ich würde die Kapitel umdrehen u. das vor dem ersten Kapitel bringen

Das Product Backlog stellt eine Menge an Anforderungen an das zu entwickelnde Produkt dar. Es ist keineswegs statisch und kann ständig verändert werden. Für die Gestaltung des Product Backlogs ist der Product Owner zuständig. Es stellt den Ausgangspunkt für die umzusetzenden Funktionalitäten des Produkts dar. Diese Anforderungen sind jedoch nicht technisch, sondern eher anwenderfreundlich als sogenannte User Stories definiert. In so einer User Story wird in meist einem knappen Satz der Nutzen der Funktionalität für den Endnutzer dargestellt.

Release Backlog

Das Release Backlog fasst alle User Stories zusammen, welche in einem bestimmten Release (Version) des Produkts implementiert werden sollen. Die User Stories stammen aus dem Product Backlog und folglich kann das Product Backlog in mehrere Release Backlogs unterteilt werden.

Sprint Backlog

Das Sprint Backlog definiert die für einen bestimmten Sprint umzusetzenden Produktfunktionalitäten (diese stammen aus einem bestimmten Release Backlog). Das Sprint Backlog wird laufend von den Teammitgliedern aktualisiert, also zum Beispiel wenn eine User Story umgesetzt wurde. Aus den verschiedenen Parametern eines Sprints lässt sich schließlich ein Burndown chart für jeden Sprint generieren.

Defect Backlog

Im Defect Backlog werden mögliche Fehler oder Hindernisse bei der Implementierung der User Stories festgehalten. Es dient dazu sich einen Überblick über die verschiedenen Arten und das Aufkommen von Bugs zu machen. Ein Sprint wird erst dann abgenommen, wenn alle User Stories des betreffenden Sprints weitgehend fehlerfrei sind.

Impediment Backlog

Ähnlich wie beim Defect Backlog besteht das Impediment Backlog aus Hindernissen. Diese Hindernisse sind aber keine technischen Fehler, sondern meist organisatorische Hindernisse oder jene, welche das Projektmanagement betreffen. Für die Erstellung des Impediment Backlogs, sowie für die Beseitigung der Hindernisse, ist der Scrum Master verantwortlich.

Sprint-Burndown-Chart

Das Sprint-Burndown-Chart dient zur Visualisierung des Arbeitsfortschritts im aktuellen Sprint. Es wird nach Erledigen der User Stories im Sprint laufend aktualisiert und kann dem Team und dem Product Owner gefährliche Verzögerungen zeigen. Dabei ist auf der Ordinate der (Rest-)Aufwand in Stunden (oder eben Story Points) und auf der Abszisse die Zeit aufgetragen. Mit der Erledigung von User Stories sinkt der Restaufwand und es vergeht natürlich Zeit. Dadurch lässt sich der tatsächliche Fortschritt sowie der prognostizierte Termin für das Fertigstellen aller User Stories (Restaufwand = 0) darstellen.

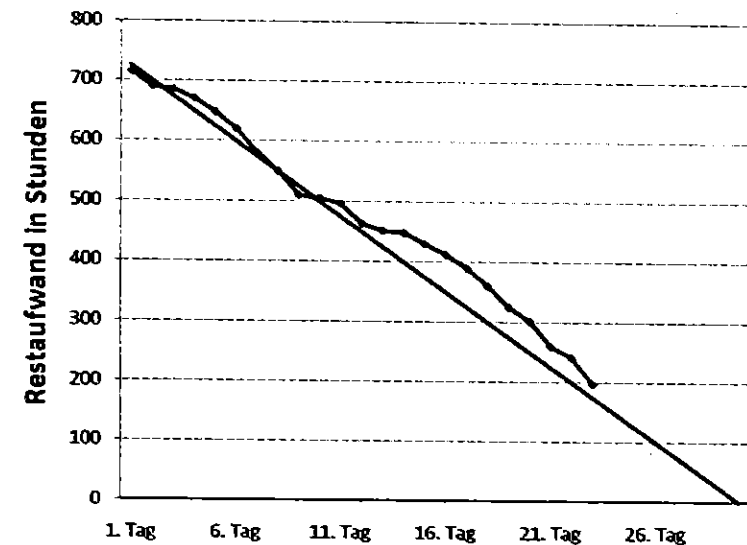


Abbildung 3: Beispiel eines Sprint-Burndown-Charts

Sprint-Endbericht

Immer wenn ein Sprint beendet wird, wird ein Sprint-Endbericht vom Product Owner erstellt. Dieser umfasst den Sprint-Verlauf, also die umgesetzten Funktionalitäten, die Hindernisse, sowie das Sprint-Burndown-Chart.

2.3.1.3 Scrum-Ablauf

Scrum beginnt mit der Festlegung der verschiedenen Rollen. Es werden also Product Owner, Scrum Master, sowie das Entwicklungsteam bestimmt. Der nächste Schritt ist das Erstellen und Sammeln der sogenannten User Stories. Diese User Stories stellen im Grunde die Funktionen des zukünftigen Produktes dar. Aus diesen User Stories wird nun ein Product Backlog definiert. Dabei entscheidet der Product Owner maßgeblich, welche User Stories hineinkommen und welche gestrichen werden. Der Product Owner und das Entwicklungsteam schätzen dann den Aufwand der verschiedenen User Stories in Story Points oder Zeiteinheiten. Als Nächstes priorisiert der Product Owner, welche Funktionalitäten wann (in welchem Sprint) umgesetzt werden.

Aus dieser Priorisierung ergibt sich nun das Sprint Planning (ein Release Planning davor ist optional). Das Ergebnis des Sprint Planning ist das Sprint Backlog. Es enthält alle funktionalen Anforderungen, die in diesem bestimmten Sprint umgesetzt werden sollen.

Ein Sprint dauert normalerweise zwei bis vier Wochen und stellt den Zeitraum dar, in dem die Anforderungen aus dem jeweiligen Sprint Backlog umgesetzt werden sollen. Innerhalb des Sprints findet täglich ein kurzes Daily Scrum Meeting statt. Das Daily Scrum Meeting ist ein Stand-up-Meeting und dient zum Informationsaustausch zwischen Entwicklungsteam, Product Owner und Scrum Master. Bei jedem Daily Scrum Meeting werden prinzipiell diese drei Fragen gestellt:

- Was haben Sie seit dem letzten Meeting getan?
- Was werden Sie heute machen?
- Hatten Sie etwaige Probleme?

Durch diese Fragen möchte der Scrum Master ein Maximum an Information gewinnen. Durch das kurze Meeting halten sich alle Beteiligten über den Projektfortschritt am Laufenden und der Scrum Master hält mögliche Probleme fest und versucht diese zu lösen.

Am Ende eines Sprints steht die Sprint Review. Sie stellt die Abnahme der in diesem Sprint implementierten Funktionalitäten dar. Der Product Owner nimmt, meist gemeinsam mit dem Auftraggeber, diese Funktionen ab und erstellt daraus den Sprint-Endbericht.

Nach dem Sprint Review erfolgt die Sprint Retrospektive. Hierbei geht es nicht um die umgesetzten Anforderungen sondern allein um das Feedback des Teams hinsichtlich der Zusammenarbeit und Organisation. Dabei nennen die Mitglieder positive und negative Aspekte des Sprints. Abschließend wird zu jedem negativen Punkt eine Verbesserungsmaßnahme definiert.

Sollte es der Fall sein, dass alle User Stories aus dem Product Backlog abgearbeitet sind, findet nach diesem Sprint eine Endabnahme statt und das fertige Produkt liegt vor. Ansonsten wird das Ganze wiederholt und ein neuer Sprint mit neuem Sprint Backlog wird gestartet. Das Ergebnis eines jeden Sprints ist ein Produktinkrement.

↑
wir verstehen das, aber
sonst keine :)

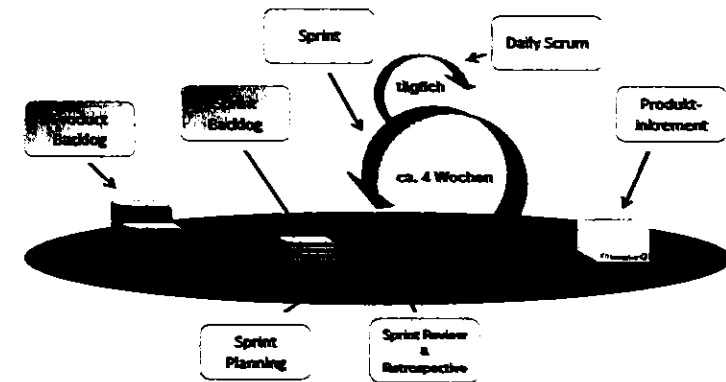


Abbildung 4: Scrum-Ablauf

2.3.1.4 Vorteile von Scrum

Scrum bringt, als agiler und schlanker Ansatz des Projektmanagements, einige Vorteile mit sich, weswegen sich das Projektteam für Scrum entschieden hat. Durch Scrum erhält der Auftraggeber schon sehr früh einen Einblick in den Entwicklungsstand und kann daher mit einer Änderung von Anforderungen oder Ähnlichem auf die Produktentwicklung Einfluss nehmen. Das ist einer der großen Vorteile von Scrum. Der Auftraggeber weiß zu jeder Zeit, wie es mit dem Projektfortschritt steht und kann gegebenenfalls eingreifen, falls etwas schief läuft.

Ein weiterer Vorteil ist die aktive Miteinbeziehung des Auftraggebers, was in meist zufriedenstellender Software mündet. Durch Scrum ist eine Änderung der Anforderungen jederzeit möglich. Im Gegensatz zu der statischen, langen Planung einer klassischen Vorgehensweise, ist eine Änderung der Anforderungen bei Scrum jederzeit möglich. Zudem steigt die Produktivität, weil nur die wirklich effektiv benötigten Funktionen umgesetzt werden. Außerdem erhöht Scrum, durch die selbstorganisierten Teams, maßgeblich die Motivation der Teammitglieder.

↳ Das haben wir eigentlich schon

2.3.2 Scrum im Team

Das Diplomprojektsteam hat sich für die Durchführung des Projektes für den agilen Ansatz Scrum entschieden. Die Gruppe zählte fünf Personen. Niklas Hobenwarter wurde auf der Projektwoche in Kärnten vom Team zum Product Owner gewählt. Selina Brinnich fungierte als technische Architektin. Das Entwicklungsteam bestand aus Philipp Adler, Adin Karic und Thomas Stedronsky, wobei hier der Begriff Entwicklungsteam nicht zu wörtlich zu nehmen ist, denn es waren alle fünf Teilnehmer aktiv an der Entwicklung des Produktes beteiligt. Als Scrum Master stellte sich Prof. Borko zur Verfügung.

Die Projektwoche in Kärnten im Oktober stellte den Kick Off für das Projekt dar. Dort wurden wichtige Dokumente geschrieben und Rollen vergeben. Auch die Erstellung der User Stories im Team wurde dort erledigt. Dabei wurde auch über mögliche Akzeptanzkriterien festgelegt. Die User Stories wurden schließlich in dem Online-Scrum-Tool OnTime von Axosoft eingefügt. OnTime wurde auch für den weiteren Verlauf des Projekts als Unterstützung für den Scrum-Prozess verwendet. Als Sprintlänge wurde eine Dauer von drei Wochen bestimmt.

443	Als Besucher möchte ich mich über ein Formular registrieren können
649	Ein Formular inklusive Captcha wird angezeigt
650	Neuer User wird erstellt
651	Formulardaten werden sinnvoll validiert

Abbildung 5: Beispiel für eine User Story in OnTime

Das Team hat sich feste Arbeitszeiten ausgemacht und kam so auf eine durchschnittliche Arbeitszeit von 13 Stunden pro Woche. Zur Steigerung der Produktivität wurde die sogenannte Pomodoro-Technik angewandt. Diese Technik besagt, dass man durch häufige, kleine Pausen viel effizienter arbeiten kann. Die Arbeitseinheiten werden als Pomodori (Einzahl Pomodoro) bezeichnet. In unserer Anwendung der Technik dauerte ein Pomodoro 25 Minuten. Nach jedem Pomodoro wurde eine fünfminütige Pause eingelegt. Nach jedem vierten Pomodoro gab es eine lange Pause von 15 Minuten.



Abbildung 6: Ansicht eines Pomodoro-Timers

Die User Stories wurden vom Product Owner priorisiert und gemeinsam mit dem Team wurde der Aufwand in Story Points geschätzt. In unserer Kalkulation entsprach ein Story Point einer Arbeitszeit von vier Stunden.

Durch die Priorisierung wurden die ersten User Stories in ein Sprint Backlog gepackt. Der „0. Sprint“ diente hauptsächlich der Evaluierung von Tools und Technologien, speziell jener, die in den jeweiligen Spezialgebieten der Mitglieder von Relevanz sein könnten. Das Ergebnis dieses Sprints war die Implementierung von funktionstüchtigen Prototypen, die zu einem späteren Zeitpunkt des Projektes eine wichtige Rolle einnehmen würden.

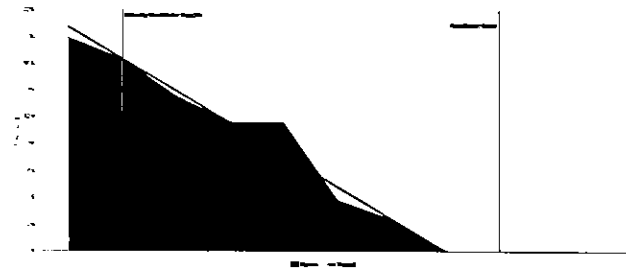


Abbildung 7: Burndown-Chart des Evaluierungssprints

Im ersten richtigen Sprint ging es vor allem um die Implementierung der System-Basis für unser Projekt. Dabei wurden Funktionen wie das Registrieren und Anmelden der Nutzer, die Profile der Nutzer, einfache Hefterstellung, allgemeine Benutzerverwaltung, sowie ein einfacher Stundenplan umgesetzt.



Abbildung 8: Burndown-Chart des ersten Sprints

Der zweite Sprint befasste sich hauptsächlich mit der Umsetzung des Stundenplans, sowie den allgemeinen Kontoeinstellungen der Nutzer. Zusätzlich wurde in diesem Sprint an trivialen Heftoperationen wie zum Beispiel dem Löschen von Elementen gearbeitet.

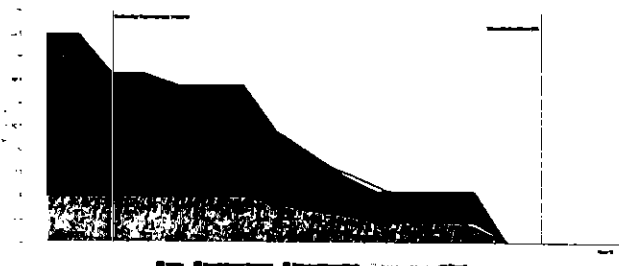


Abbildung 9: Burndown-Chart des zweiten Sprints

Im dritten Sprint ging es meist um die Heftfunktionen und die Heftansicht. In diesem Sprint wurde die Ansicht des Heftes neu gestaltet, das Blättern in Heften wurde ermöglicht und das Importieren von Heftseiten aus anderen Heften wurde implementiert.

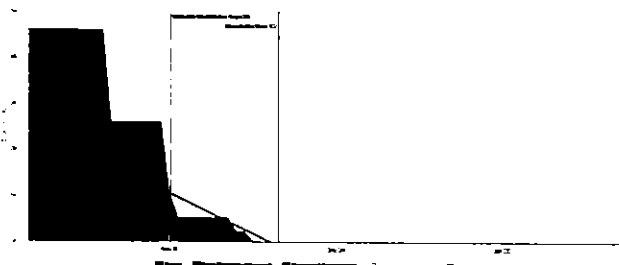


Abbildung 10: Burndown-Chart des dritten Sprints

Die Planung des vierten Sprints war die Folge einer Restrukturierung und der Neupriorisierung des Product Owners. Oberste Maxime war die Fertigstellung der drei Heftelemente: das Textelement, das Programmelement und das Bildelement.

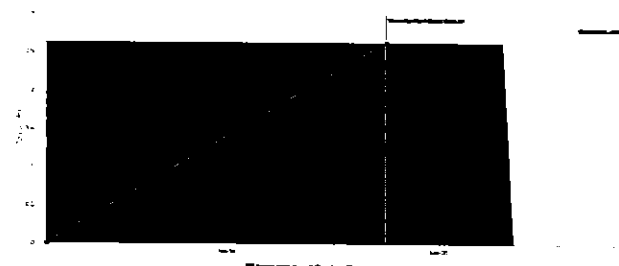


Abbildung 11: Burndown-Chart des vierten Sprints

Der letzte Sprint befasste sich vor allem mit zwei Funktionen des Produktes. Das gemeinsame Arbeiten (Parallel Working System) und das OCR-Modul (optische Zeichenerkennung) wurden umgesetzt. Zudem wurden allfällige Mängel und Fehler behoben.

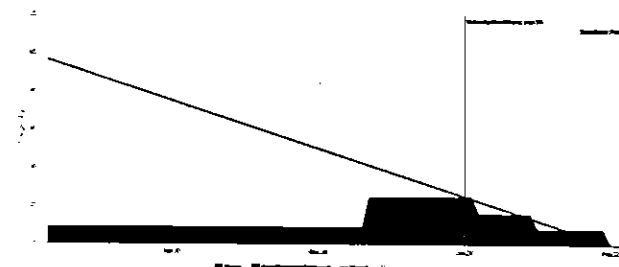


Abbildung 12: Burndown-Chart des letzten Sprints

Nach jedem Sprint fand eine Sprint-Abnahme durch den Product Owner und den Auftraggeber statt. Dabei wurden die funktionalen Anforderungen, die in dem jeweiligen Sprint umzusetzen waren, überprüft und abgenommen. Nach der Sprint-Abnahme erfolgte das Sprint-Retrospektive-Meeting in dem positive und negative Aspekte der Zusammenarbeit und Organisation in diesem Sprint diskutiert wurden. Der letzte Schritt stellte die Formulierung von Verbesserungsvorschlägen und deren Umsetzung für die folgenden Sprints dar.

3 Stand der Technik

Die globale, schnelle IT Welt bietet verschiedenste Technologien, die Entwickler in ihrer Aufgabenstellung unterstützen. Bei diesem Überangebot ist es schwierig, das Passende für seine Anforderung zu finden.

Dieses Kapitel befasst sich mit dem Vergleich nützlicher Systeme, um sich für die Geeigneten zu entscheiden. *Auswahl A*

3.1 Frameworks

Frameworks sind eine Sammlung von Softwareprodukten, die Entwicklungsansätze zur Verfügung stellen. Sozusagen ermöglichen sie durch vordefinierte Grundbausteine eine schnelle, reibungslose Entwicklung. Mithilfe von Schnittstellen und Bibliotheken soll dem Entwickler Arbeit abgenommen werden.

Für unser System benötigen wir:

- *ein* Web Framework als Grundgerüst
- *JS* JS-Framework um auf Useraktionen zu reagieren
- CSS-Framework für das grafische Design
- Element-Frameworks als Grundlage für die Funktion in den Schulheften
- GUI-Testing um zu garantieren, dass alle Funktionen reibungslos laufen

3.1.1 Web Frameworks

Web Frameworks haben das Ziel, dem Entwickler einer Webanwendung mit Hilfe von vordefinierten Klassen zu unterstützen. Sie sorgen dafür, dass eine Verbindung mit der Datenbank aufgebaut wird, Inhalte dynamisch angezeigt werden und ein Anmeldesystem zur Verfügung steht.

Web Frameworks teilen das Projekt in zwei Teile, dem Backend und Frontend.

Für die Auswahl unserer Web Frameworks wurden Flask, Django und Play verglichen. Dabei wurde darauf geachtet, *2* wie groß der Aufwand der Installation und Konfiguration ist, wie weit es ausbaufähig ist und welche Funktionalität bereits zur Verfügung steht.

3.1.1.1 Flask

„Flask ist ein microframework für Python, basierend auf Werkzeug, Jinja 2.“ [Ron16]
Es ist frei verfügbar und aufgrund seiner geringen Größe schnell und einfach zu installieren und konfigurieren. Da es zu den kleineren Web Frameworks zählt, wird wenig Funktionalität geboten, welche aber durch einfache Erweiterungen behoben werden kann. [Ron16]

Die Installation von Flask ist ziemlich simpel. Es kann mit folgendem Befehl installiert werden: *in welchem Zielsystem?*

```
pip3 install Flask
```

Listing 1: Installation von Flask [Ron16]

Für ein einfaches „Hello World“-Programm, geschrieben in Flask, braucht es eine Funktion, die die Message ausgibt und eine Main-Methode, welche die Applikation startet.

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello Flask World!"

if __name__ == "__main__":
    app.run()
```

Listing 2: Flask Hello-World [Ron16]

Mit dem Befehl `python3 hello.py` lässt sich das Programm ausführen.

3.1.1.2 Django

schnell, od. schneller?
Django ist ein Open-Source Python Web Framework für die schnelle Entwicklung einer Webanwendung. Es bietet eine detaillierte und umfangreiche Dokumentation über alle Funktionalitäten und wird außerdem durch eine große Community unterstützt. Obwohl es mit sehr vielen Features ausgestattet ist, ist es einfach erweiterbar und kann mit wenig Aufwand installiert und konfiguriert werden. [Fou16a]

Installiert wurde das Web Framework folgenderweise: *in Zielsystem*

```
apt-get install python3-django
```

Listing 3: Installation von Django [Fou16b]

Zur Erstellung eines „Hello-World“-Programmes muss zu Beginn ein Projekt erzeugt werden. Im nächsten Schritt ist die Datenbank entsprechend der Project-Settings zu konfigurieren. Anschließend wird im Unterordner *app* ein neues File „hello.py“ mit folgenden Code erstellt:

```
from django.http import HttpResponse

def hello_world(request):
    return HttpResponse("Hello Django World!")
```

Listing 4: Django Hello-World [sun15]

Wird der Server mit dem Befehl *python manage.py runserver* gestartet, erscheint die Message im Browserfenster.

3.1.1.3 Play

Play, ein leichtgewichtiges Web Framework basierend auf Java und Scala, steht für den kommerziellen Gebrauch kostenlos zur Verfügung. Allerdings ist die offizielle Dokumentation nicht besonders ausführlich. ^(A)

Die Installation gestaltete sich zeitaufwendiger und dauerte im Vergleich zu anderen Web Frameworks länger. [Pla15b]

Um Play zu installieren muss das Framework zunächst von <https://www.playframework.com/download> heruntergeladen werden. Das downgeladene zip-File muss entpackt und der darin enthaltene Activator zum PATH hinzugefügt werden. Danach wird der Activator mit folgendem Befehl gestartet [Pla15a]:

```
activator ui
```

Listing 5: Konfiguration von Play [Pla15a]

Für das „Hello-World“-Programm braucht es ein neues Projekt. Im Unterordner */controllers* muss folgende Methode hinzugefügt werden:

```
public static Result hello() {
    return ok(main.render("Hello World", new
        play.twirl.api.Html("Hello Play World!")));
}
```

Listing 6: Play Hello-World [Pla15a]

Zu guter Letzt wird der Prototyp mit dem Befehl *activator run* im Projekt-Verzeichnis ausgeführt.

3.1.1.4 Vergleich

Das DSN-Team hatte im Vorfeld mit der Programmierung von Web Frameworks noch keine Erfahrungswerte.

Aufgrund der leicht verständlichen Dokumentation und der gut vernetzten Community, hat sich Django klar herauskristallisiert. Als wichtige Entscheidungshilfe galten die erstellten Prototypen. Die schnelle, einfache Entwicklung, sowie die Installation und Konfiguration des Frameworks, hinterließ beim Team einen guten Eindruck. ✓

3.1.2 JS-Frameworks

JavaScript-Frameworks sind dazu da, um Userinteraktionen entgegenzunehmen, diese Daten zu validieren und zu verarbeiten und am Ende das Ergebnis zu retournieren. Im Falle von DSN dienen sie als Schnittstelle zum Webserver und außerdem dazu, den Content dynamisch zu ändern. Verglichen wurden die bekanntesten, populärsten JavaScript-Frameworks in den Bereichen:

- Browserunterstützung
- Dokumentation & Community
- Lizenz und Kosten
- Schwierigkeit bei der Erstellung eines Prototypen

3.1.2.1 Dojo

Dojo ist eine JS-Bibliothek von Dojo Foundation, die Entwicklern JS- und Ajax-basierende Module anbietet. Das Framework ist modular aufgebaut, ^(B) das für eine gute Übersicht im Code sorgt. Dojo ist frei anwendbar und plattformunabhängig. Die Dokumentation besteht aus einer übersichtlichen Auflistung von step-by-step Tutorials. [Bol16]

Das Installieren bzw. Anwenden des Frameworks ist sehr einfach. Eine Möglichkeit besteht darin, den gesamten Source-Ordner herunterzuladen, in dem sich der Core mit den Kernfunktionalitäten befindet, oder ganz einfach den Link der *dojo.js* Datei im Header anzugeben.

```
<script src="//ajax.googleapis.com/ajax/libs/dojo/1.10.4/dojo/dojo.js">
</script>
```

Listing 7: Dojo einbinden [Fou16c]

Der Prototyp, ein Drag & Drop Beispiel, ließ sich dank verständlicher Erklärung ohne Probleme umsetzen. Allerdings muss sich der Entwickler im Klaren sein, welche Module für eine lauffähige Applikation einzubinden sind. [Ste15]

3.1.2.2 Sencha ExtJS

Das ExtJS JavaScript Framework von Sencha dient der Realisierung von komplexen Anwendungen. Sencha bietet eine gut ausgebaute, strukturierte API. Neben einer großen Community von über 500.000 Mitgliedern unterstützt ExtJS alle marktführenden Webbrowser, sowie Smartphones.

Sencha ist unter der Commercial License oder GNU General Public License verfügbar. Für den kommerziellen Nutzen benötigt es eine Lizenz im Wert von \$895.00.[Mü15]

Sencha ExtJS hat den Vorteil, dass es unabhängig ist, ohne Backend ausgeliefert wird und viele fertige Komponenten (Charts, Grids, Forms) schon vorhanden sind. Basierend auf einem MVC-Modell ist das Klassensystem objektorientiert aufgebaut. Für das Testing bietet die Sencha Suite keine eigenen Frameworks, aber es besteht die Möglichkeit, mit verschiedenen Drittanbieter-Produkten zu testen.[Tec15][Sen16]

Für die Verwendung von Sencha müssen die benötigten Module im Header des HTML-Files eingebunden werden. Die Umsetzung des Prototyps gestaltete sich einfach und konnte den Bedürfnissen entsprechend angepasst werden. Da Sencha HTML und JavaScript strikt trennt, ist nur die Einbindung von fertigen JS-Files notwendig.

3.1.2.3 jQuery

jQuery ist ein plattformunabhängiges JavaScript Framework von der jQuery Foundation. Die Open-Source-Software steht frei für jegliche Verwendung zur Verfügung, solange im Projekt überall die Copyrights stehen. Mit einer umfangreichen Klassenbibliothek unterstützt jQuery den Umgang mit DOM (Document Object Model). DOM wandelt alle HTML-Elemente in Objekte um, die während der Laufzeit dynamisch angepasst werden.[Nie12]

Für die Nutzung ist die jQuery.js Datei in den Header einzubinden.

```
<script src="//code.jquery.com/jquery-1.12.0.min.js"></script>
```

Listing 8: jQuery einbinden[jF16]

Für den Entwickler ist jQuery ein leicht verständliches Framework mit einer Vielzahl von Funktionen. In kurzer Zeit ist es möglich große Fortschritte zu erzielen. So war auch die Erfahrung bei der Umsetzung des Prototyps. Hauptsächlich musste nur eine Funktion geschrieben werden, welche auf eine id zeigt. [Pat07]

3.1.2.4 AngularJS

AngularJS ist ein JavaScript Framework von Google, welches einen MVC-Ansatz verfolgt. Die 4 wichtigsten Browser Chrome, Firefox, Safari und Edge werden unterstützt. *Lauschen*

(F)

Obwohl die Community relativ wenig Mitglieder beherbergt, beinhaltet die Dokumentation ausführliche Informationen über Funktionen, die zusätzlich mit Beispielen untermauert sind. Als OpenSource Framework, unter der MIT Lizenz veröffentlicht, erlaubt es die freie Nutzung und Veränderung des Codes.

Für die Verwendung des Frameworks wird der HTML-Code einfach um AngularJS-Attribute erweitert. Dadurch besteht eine strikte HTML und JS Trennung, welches die Lesbarkeit, Übersichtlichkeit und Testbarkeit des Codes verbessert. [Pla12]

Um AngularJS in der Praxis einsetzen zu können, muss die JS-Bibliothek im Header inkludiert werden:

```
<script  
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.2/angular.min.js">  
</script>
```

Listing 9: AngularJS einbinden[Goo16a]

Verpflichtend für die Anwendung von AngularJS ist das Setzen des ng-app Attributes im HTML-Root Tag. Es soll dem Framework mitteilen, wo sich das Root-Element befindet. Anfangs gestaltete sich die Implementation der Drag & Drop Applikation als kompliziert, doch nach zunehmender Erfahrung mit dem Framework konnte in sehr kurzer Zeit viel erreicht werden.

3.1.2.5 Meteor

Meteor ist ein full-stack JavaScript Framework, welches alle Plattformen unterstützt. Erwähnenswert ist, dass Meteor mit NoSQL Datenbanken, insbesondere MongoDB, kooperiert. Ähnlich wie Java besitzt es eine große, strukturierte API, dessen einzelne Methoden ausführlich beschrieben sind. Die Nutzung von Meteor ist frei. Dank einer eigenen Community, namens Meteorpedia, steht einer Problemlösung nichts im Weg. Ein weiterer Vorteil ist das live deploying. Dadurch wird es dem Entwickler ermöglicht, Änderungen automatisch in realtime hochzuladen und zu deployen. [GM15]

Die Installation dauerte, im Gegensatz zu anderen JavaScript Frameworks, länger. Es gab nicht die Möglichkeit einer Header-Einbindung stattdessen musste es mit folgendem Befehl installiert und getestet werden:

```
curl https://install.meteor.com/ | sh
meteor create ~/my_cool_app
cd ~/my_cool_app
meteor
```

System?
eh klar Linux - aber bitte aufpassen

Listing 10: Installation von Meteor [met16]

`meteor create` erstellt ein neues Meteor Projekt. Durch die Angabe von `~/my_cool_app` wird ein default Projekt erstellt, das gleichzeitig als Prototyp genutzt werden kann. Bei diesem Prototypen handelt es sich um einen Klickzähler, der bei jedem Klick die Variable um 1 erhöht und anschließend ausgibt.

3.1.2.6 Vergleich

Wichtig für das DSN Team ist es, die Kosten so gering wie möglich zu halten, sowie das Vorhandensein einer ausführlichen Dokumentation, zum besseren Verständnis mit Beispielen untermauert. Die Entscheidung fiel zugunsten AngularJS und jQuery aus, weil es neben den oben genannten Forderungen viele Funktionalitäten anbietet und mit dem Web Framework Django harmoniert.

3.1.3 CSS-Frameworks

Nicht nur das Backend spielt im System eine wichtige Rolle, sondern auch Design und Gestaltung einer Website. CSS-Frameworks sprechen User an, denen Klarheit und Übersichtlichkeit wichtig ist. Sozusagen interagiert es mit dem Anwender und animiert ihn, das System zu verwenden.

Wichtig erscheint uns, dass ein CSS-Framework:

- einfach zu installieren und konfigurieren ist
- browserunabhängig ist und sich dynamisch an Devices anpasst
- zahlreiche, vordefinierte Funktionen bietet, die ausführlich dokumentiert sind

Ein Framework MACHT
kein Design! Es bietet nur
bessere Umsetzungsstrategien!
Bitte umformulieren

3.1.3.1 Yaml

Yaml ist ein CSS-Framework von Dirk Jesse, welches von allen modernen Browsern, wie Chrome, Firefox, Opera, Safari und Internet Explorer, unterstützt wird. Es passt sich an jeden Screen dynamisch an, sei es bei diversen Browsern oder auch auf mobilen Endgeräten wie iPhone und iPad. Hinzuzufügen ist, dass das Framework modular aufgebaut ist. Neben dem Kernmodul, welches flexible Layouts, variable Spaltenbreiten, sowie Grid-Layouts mit fester Breite beinhaltet, können nach Belieben weitere Module eingebunden werden. Yaml bietet eine umfangreiche deutschsprachige Dokumentation, die alle notwendigen Informationen enthält. Leider ist die Suche nach bestimmten Elementen nicht möglich. [Jes13b]

Das Framework wird unter der Creative Commons Attribution 2.0 Lizenz (CC-BY 2.0) veröffentlicht, die privaten als auch kommerziellen Gebrauch erlaubt.

Die aktuellste Version kann von der Hauptseite <http://www.yaml.de> heruntergeladen und entpackt werden. Im entpackten Ordner befinden sich alle notwendigen .css Dateien und Demos, die für den Prototyp essentiell sind. Bei der Verwendung muss in den HTML Files der Pfad zum CSS File und bei HTML Tags die gewünschte class angegeben werden.

```
<link rel="stylesheet" href="yaml/core/base.css" type="text/css"/>
<link rel="stylesheet" href="css/styles.css" type="text/css"/>
```

Listing 11: YAML einbinden [Jes13a]

Dadurch wird automatisch nach dem Style des CSS-Frameworks formatiert. 7. Sinn?
So ließ sich in sehr kurzer Zeit ein userfreundliches Anmeldeformular erstellen.

3.1.3.2 Pure

Pure ist eine nur 4kb große, komprimierte CSS-Datei, dass heißt im Vergleich zu Anderen ist es sehr klein. Hauptsächlich besteht es aus einem Set von CSS Modulen. Die Dokumentation ist übersichtlich gegliedert, sodass sich jeder schnell zurechtfindet. Mithilfe des Navigationsbalkens können bestimmte Elemente schnell gefunden werden. Bei Auftreten von Problemen mit Pure stehen zahlreiche Foren zur Verfügung. Das Framework wird unter der BSD-Lizenz veröffentlicht, und kann kostenlos genutzt werden.

Der Unterschied zu anderen CSS-Frameworks, wie Bootstrap oder Yaml, ist, dass es ohne JavaScript Plugins ausgeliefert wird. JavaScript wird nur für Dropdown-Menüs und fixed top- bzw. bottom-Navigation eingesetzt. Pure bietet sechs Bausteine, welche die Hauptanforderungen eines jeden Entwicklers abdecken. Da Pure wegen der kleinen Größe nur die notwendige Funktionalität anbietet, ist es bei Bedarf erweiterbar.

Für die Verwendung von Pure muss hauptsächlich der Link zu dem CSS File im Header des HTML-Files eingefügt werden. [Yah14]

```
<link rel="stylesheet"
href="http://yui.yahooapis.com/pure/0.6.0/pure-min.css">
```

Listing 12: Pure einbinden [Yah14]

Der Prototyp, ein Anmeldeformular, konnte durch das gut beschriebene Tutorial, untermauert mit Beispielen, sehr schnell und leicht nachgebildet werden.

3.1.3.3 Bootstrap

Das frei verfügbare CSS-Framework Bootstrap von Twitter unterstützt alle Elemente, die essentiell für eine Webseite sind. Dem Entwickler steht es frei, welche Komponenten er verwenden möchte. Bootstrap passt sich dynamisch an jedes Device an. Es ist plattformunabhängig und bietet eine sehr genaue Dokumentation mit Navigationsbalken.

Für den Einsatz muss Bootstrap von der offiziellen Seite <http://getbootstrap.com> heruntergeladen werden. Die Installation bezieht sich lediglich auf das Einbinden der bereitgestellten Dateien in das eigene Projekt. Bootstrap wird als ein ZIP-Archiv bereitgestellt. In diesem befindet sich eine CSS-Datei und eine Javascript Datei. Die beiden Dateien müssen anschließend in den <head> der Webseite eingebunden werden. [Lei13]

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js">
</script>
```

Listing 13: Bootstrap einbinden [Lei13]

3.1.3.4 Vergleich

Aufgrund der erstellten Prototypen und der ausführlichen Dokumentation, fiel die Entscheidung auf Bootstrap. Bootstrap unterstützt alle Browser passt sich dynamisch dem Bildschirm an und ist mit wenig Aufwand einsetzbar. Die Syntax des CSS-Frameworks ist klar verständlich und hat außerdem den Vorteil, dass es viele Möglichkeiten der Formatierung und Validierung gibt.

das ist falsch!

3.1.4 Element-Frameworks

In diesem Kapitel werden alle notwendigen Elemente dargestellt, die in den Schulheften der DSN-Usern zur Verfügung stehen. Gemeint sind das Codeelement, Textelement, Zeichentool und die Bildergalerie.

Dem DSN-Team ist es wichtig, dass sich die Elemente den Anforderungen entsprechend anpassen können.

3.1.4.1 CodeMirror

CodeMirror ist ein Texteditor basierend auf JavaScript, welcher dank MIT Lizenz kommerziell genutzt werden darf. CodeMirror wird von Firefox, Chrome, Safari, IE und Opera unterstützt. Durch die Einbindung von Frameworks ist es möglich, ein Textfeld darzustellen. Dank unterschiedlicher Sprachunterstützungen erkennt das System automatisch Syntax und hebt diese farbig hervor. Zur besseren Übersicht werden am linken Rand Zeilennummern eingeblendet. [Cod16]

```
1 <!-- Create a simple CodeMirror instance -->
2 <link rel="stylesheet" href="lib/codemirror.css">
3 <script src="lib/codemirror.js"></script>
4 <script>
5   var editor = CodeMirror.fromTextArea(myTextarea, {
6     lineNumbers: true
7   });
8 </script>
```

Abbildung 13: CodeMirror

3.1.4.2 Sketch.js

Sketch.js ist ein freiverfügbares Zeichentool. Diverse Parameter erlauben dem User Farbe und Pinselgröße auszuwählen. Außerdem ermöglicht das plattformunabhängige Framework einen reibungslosen Download der angefertigten Zeichnung im PNG- oder JPEG-Format. [BI11]



Abbildung 14: Sketch.js

3.1.4.3 CKEditor

Der CKEditor von CKSource ist ein Texteditor, der sich ohne Mühe in Websites einbinden und verwenden lässt. Als vorteilhaft erweist sich die Kompatibilität mit den meistverwendeten Webbrowsern. Durch dieses Framework ist es möglich, einen Textblock zu erstellen und diesen individuell zu bearbeiten.

Außerdem ist es möglich, seinen CKEditor an die eigenen Bedürfnisse anzupassen. [CKS16]

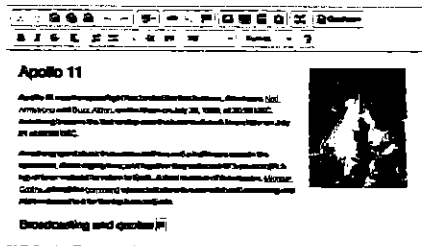


Abbildung 15: CKEditor

3.1.4.4 Photoswipe

Das Framework Photoswipe kann Bilder in einer Slideshow anzeigen und durch diese navigieren. Photoswipe ist auf allen Browsern vertreten, hat gute Grundstrukturen, braucht aber für unsere Zwecke ein paar kleine Anpassungen. [Pho14]



Abbildung 16: Slideshow

Was sind gute Grundstrukturen

3.1.5 GUI-Testing

Um eine Software auf seine einwandfreie Funktion zu testen, wird ein geeignetes GUI-Testing Framework benötigt. Die Aufgabe besteht darin, automatisiert Fehler ausfindig zu machen, mögliche Benutzerinteraktionen zu testen und auch grafische Elemente zu überprüfen.

3.1.5.1 Sahi OS

Sahi OS ist ein automatisiertes Open Source Testing Framework. Neben der kostenlosen Version mit einer eingeschränkten Anzahl an Features gibt es eine kostenpflichtige, mit umfangreichen Features. Die Testfälle lassen sich nur auf Firefox, Chrome und IE ausführen. Bei Auftreten von Fehlern, hilft die gut beschriebene, aber etwas unübersichtliche Dokumentation.

Die Installation erwies sich als sehr einfach. Auf <http://sahi.sourceforge.net/install.html#install> gibt es einen Installer zum Download, welcher das Tool auf Port 9999 startet. Obwohl kaum Probleme beim Erstellen der Testfälle mittels GUI Tool und per Script aufgetreten sind, ließ sich der Prototyp nicht ausführen. Die sehr kleine Community konnte uns auch nicht weiterhelfen.

3.1.5.2 Watir

Watir ist ein Testing-Tool basierend auf Ruby. Egal, in welcher Sprache das Projekt geschrieben oder welcher Browser verwendet wird, Watir unterstützt es. Die Dokumentation war schwierig zu finden und ist außerdem sehr unübersichtlich. [Wat14a]

Um Watir zu installieren müssen folgende Befehle in der Konsole ausgeführt werden:

```
sudo apt-get install ruby ruby-dev
sudo apt-get install rubygems
gem update --system --no-rdoc --no-ri
gem install watir --no-rdoc --no-ri
```

Listing 14: Installation von Watir [Wat14b]

Die Ausführung der Tests war aus zwei Gründen nicht umsetzbar. Zum Einen ließ sich Rubygem Watir-Web-Framework nicht installieren. Zum Anderen konnten die Testfälle aus unerklärlichen Gründen nicht ausgeführt werden.

(das ist keine Mystery Serie :))

3.1.5.3 Robot Framework

Das Testing Framework Robot Framework ist auf Akzeptanztests ausgelegt. Es ist frei unter Apache 2.0 Lizenz verfügbar. Die Dokumentation ist *viel zu unübersichtlich* und die Beispiele werden nicht *step-by-step* erklärt, was die Umsetzung des Prototypen unmöglich machte. [Rob16a] *Schritt für Schritt*

Die Installation verlief unter Python mit den Befehlen:

```
sudo apt-get install python-pip
sudo pip install robotframework
sudo pip install docutils
```

Listing 15: Installation von Robot Framework [Rob16b]

Obwohl die Installationsanleitung korrekt befolgt wurde, konnten bestimmte Libraries nicht gefunden werden. Fazit: die Tests konnten nicht ausgeführt werden.

3.1.5.4 Selenium

Eines der bekanntesten freien Testing Frameworks ist Selenium. Selenium unterstützt die meisten Browser und *besteht aus einer* großen Community. Die Testfälle können auf allen gängigen Programmiersprachen erzeugt werden.

Die Bibliothek lässt sich entweder als *stand-alone* Software oder als Plugin in einer IDE z.B. Eclipse installieren. Für die Ausführung müssen lediglich die Bibliotheken in das Projekt eingebettet werden. Es besteht die Möglichkeit, die Tests mittels GUI Tool oder per Script zu erstellen.

Innerhalb von 5 Minuten funktionierte alles reibungslos.

3.1.5.5 Vergleich

Da Selenium als einziges Produkt innerhalb kürzester Zeit den gewünschten Erfolg brachte, entschied sich das Team für dieses. Desweiteren bietet es eine sehr große Community und lässt sich komplett automatisieren und in CI Tools integrieren.

da die Formulierungen sind teilweise etwas untechnisch

3.2 Technologien

Abgesehen von einigen Frameworks, wird bei der Umsetzung der Applikation zudem eine wichtige Technologie benötigt, nämlich eine Datenbank. Es existieren eine Vielzahl von unterschiedlichen Datenbanken. Eine richtige Auswahl zu treffen ist besonders wichtig, da die Applikation die benötigten Daten möglichst schnell zurückliefern soll und auf keinen Fall Daten verloren gehen sollen.

Bei der Persistierung von Daten wird dabei zwischen zwei Konzepten von Datenbanken unterschieden: relationale Datenbanken und NoSQL Datenbanken. Um einen Vergleich aufstellen zu können, wurden jeweils zwei Datenbanken für jedes der beiden Konzepte ausgewählt und genauer betrachtet. Die Auswahl betrifft dabei folgende vier Datenbanken:

- Relationale Datenbanken
 - MySQL
 - PostgreSQL
- NoSQL Datenbanken
 - MongoDB
 - Couchbase

Im Folgenden wird genauer auf jede der aufgezählten Datenbanken eingegangen, um einen Vergleich aufstellen zu können. Dabei wird der Aufwand der Installation und Konfiguration, sowie die Einfachheit der grundlegenden Operationen Erstellen, Lesen, Bearbeiten und Löschen genauer betrachtet.

3.2.1 MySQL

MySQL ist eine Open-Source Datenbank von Oracle. Sie verspricht vor allem Einfachheit in der Anwendung. [Cor16]

Die Installation von MySQL ist sehr einfach und schnell erledigt. Die Datenbank kann mithilfe folgender Befehle installiert werden:

```
sudo apt-get install mysql-server mysql-client
sudo mysqladmin -u root -h localhost password 'mypassword'
sudo apt-get install php5-mysql
```

Listing 16: Installation von MySQL [dV07]

Bei der Konfiguration von MySQL muss zunächst ein neuer Benutzer erstellt und anschließend die entsprechenden Rechte gesetzt werden:

```
mysql -u root -p
CREATE USER 'eval'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON eval.* TO 'eval'@'localhost';
CREATE DATABASE eval;
FLUSH PRIVILEGES;
mysql -u eval -p
```

Listing 17: Konfiguration von MySQL [Sve12]

Nach der Installation und Konfiguration können weitere Operationen ausgeführt werden.

Ein neuer Datensatz kann folgendermaßen in die Datenbank eingefügt werden:

```
INSERT INTO test VALUES ("Xandra Mcknight");
```

Um die vorhandenen Datensätze wieder auszulesen, wird folgende Zeile benötigt:

```
SELECT * FROM test;
```

Ein bestehender Datensatz kann mit folgender Zeile verändert werden:

```
UPDATE test SET name='Peter Test' WHERE name='Xandra Mcknight';
```

Um einen vorhandenen Datensatz zu löschen kann folgender Befehl ausgeführt werden:

```
DELETE FROM test WHERE name='Xandra Mcknight';
```

3.2.2 PostgreSQL

PostgreSQL ist eine Open-Source Datenbank. Sie verspricht vor allem eine hohe Zuverlässigkeit und Datenintegrität. [Gro16]

Die Installation von PostgreSQL ist sehr einfach. Sie verläuft sehr schnell und es muss nur ein einziger Befehl ausgeführt werden. Die Datenbank kann mithilfe folgendem Befehl installiert werden:

```
apt-get install postgresql postgresql-client
```

Listing 18: Installation von PostgreSQL [Wik15]

Bei der Konfiguration von PostgreSQL muss ein neuer Benutzer erstellt und anschließend die entsprechenden Rechte gesetzt werden:

```
su postgres
psql
CREATE USER mypguser WITH PASSWORD 'mypguserpass';
CREATE DATABASE mypgdatabase OWNER mypguser;
\q
vim /etc/postgresql/X.Y/main/pg_hba.conf
Change: local all all peer
To: local all all md5
/etc/init.d/postgresql reload
psql -d mypgdatabase -U mypguser
```

Listing 19: Konfiguration von PostgreSQL [Wik15]

Nach der Installation und Konfiguration können weitere Operationen ausgeführt werden.

Ein neuer Datensatz kann folgendermaßen in die Datenbank eingefügt werden:

```
INSERT INTO test VALUES ("Xandra Mcknight");
```

Um die vorhandenen Datensätze wieder auszulesen, wird folgende Zeile benötigt:

```
SELECT * FROM test;
```

Ein bestehender Datensatz kann mit folgender Zeile verändert werden:

```
UPDATE test SET name='Peter Test' WHERE name='Xandra Mcknight';
```

Um einen vorhandenen Datensatz zu löschen kann folgender Befehl ausgeführt werden:

```
DELETE FROM test WHERE name='Xandra Mcknight';
```

3.2.3 MongoDB

MongoDB ist eine Open-Source NoSQL Datenbank. Sie verspricht vor allem eine hohe Flexibilität, Skalierbarkeit und Performance. [Mon16]

Die Installation von MongoDB verläuft folgendermaßen:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv 7F0CEB10
echo "deb http://repo.mongodb.org/apt/debian wheezy/mongodb-org/3.0 main" |
sudo tee /etc/apt/sources.list.d/mongodb-org-3.0.list
sudo apt-get update
sudo apt-get install -y mongodb-org
sudo service mongod start
```

Listing 20: Installation von MongoDB [Doc16]

Eine weitere Konfiguration ist nicht notwendig.

Nach der Installation können weitere Operationen ausgeführt werden.
 Ein neuer Datensatz kann folgendermaßen in die Datenbank eingefügt werden:
`db.test.insert({"name": "Test", "email": "test@email.com"})`
 Um die vorhandenen Datensätze wieder auszulesen, wird folgende Zeile benötigt:
`db.test.find({"name": "Test"})`
 Ein bestehender Datensatz kann mit folgender Zeile verändert werden:
`db.test.update({"name": "Test"},{$set:{"name": "Test2"}})`
 Um einen vorhandenen Datensatz zu löschen kann folgender Befehl ausgeführt werden:
`db.test.remove({"name": "Test"})`

3.2.4 Couchbase

Couchbase ist eine Open-Source NoSQL Datenbank. Sie verspricht vor allem eine hohe Skalierbarkeit.[Cou16]

Die Installation von Couchbase verläuft über ein externes .deb Packet. Zur Installation werden folgende Befehle benötigt:

```
wget http://packages.couchbase.com/releases/4.0.0-rc0/couchbase-
server-community_4.0.0-rc0-ubuntu14.04_amd64.deb
dpkg -i couchbase-server-community_4.0.0-rc0-ubuntu14.04_amd64.deb
```

Listing 21: Installation von Couchbase [Doc15]

Bei der Konfiguration von Couchbase muss das CLI Programm in die Bashrc hinzugefügt werden:

```
if [ -d "/opt/couchbase/bin" ] ; then
    export PATH="/opt/couchbase/bin:$PATH"
fi
```

Listing 22: Konfiguration von Couchbase

Nach der Installation und Konfiguration können weitere Operationen ausgeführt werden.
 Ein neuer Datensatz kann folgendermaßen in die Datenbank eingefügt werden:
`cbtransfer CSV_FILE http://IP:PORT -B TARGET_BUCKET -u USERNAME -p PASSWORD`
 Um die vorhandenen Datensätze wieder auszulesen, wird folgende Zeile benötigt:
`cbtransfer http://IP:PORT csv:FILE -b SOURCE_BUCKET -u USERNAME -p PASSWORD`
 Ein bestehender Datensatz kann mit folgender Zeile verändert werden:
`cbtransfer FILE http://IP:PORT -B TARGET_BUCKET -destination-operation=add/set -u USERNAME -p PASSWORD`

Um einen vorhandenen Datensatz zu löschen kann folgender Befehl ausgeführt werden:
`couchbase-cli bucket-delete -c IP:PORT -bucket=NAME -u USERNAME -p PASSWORD`

3.2.5 Vergleich

Anhand der zuvor genauer betrachteten Datenbanken und deren erstellter Prototypen kann nun eine Entscheidung darüber getroffen werden, welche der Datenbanken bei der Applikation zur Verwendung kommen soll.

Die beiden relationalen Datenbanken haben den großen Vorteil, dass sie für die grundlegenden Operationen eine standardisierte Sprache verwenden: SQL. Dadurch wird die Verwendung deutlich erleichtert. Allerdings ist bei der Applikation die Verwendung einer relationalen Datenbank nicht sehr sinnvoll, da sich die Daten vor allem innerhalb der Hefte heft unterscheiden können und keine eindeutige Struktur dafür festgelegt werden kann. Bei relationalen Datenbanken muss so eine fixe Struktur allerdings existieren. NoSQL Datenbanken hingegen haben den großen Vorteil, dass sie unterschiedliche Strukturen unterstützen und nicht von Anfang an festgelegt werden muss, wie die Daten genau aufgebaut sind. Daher ist die Verwendung einer NoSQL Datenbank bei der Applikation sinnvoller.

Nun bleibt noch die Entscheidung, welche der beiden NoSQL Datenbanken verwendet werden soll. Der große Vorteil von Couchbase ist die hohe Skalierbarkeit und gute Performance. Allerdings ist die Verwendung von MongoDB deutlich einfacher. Zudem existiert für MongoDB eine wesentlich größere Community und bessere Tutorials, die während der Entwicklung sehr hilfreich sein und viel Arbeit und vor allem Zeit ersparen können. Aus diesem Grund wurde MongoDB als besser geeignet für das Projekt angesehen und wird demnach für die Persistierung der Daten für die Applikation eingesetzt.

4 Design

Für die Umsetzung einer guten Idee braucht es ein strukturiertes und gut geplantes Design. Dadurch ist bereits zu Projektbeginn das Ausmaß des Gesamtsystems erkennbar. Das Team kann sich dadurch einen Überblick bezüglich Herausforderungen und vorraussichtlichen Zeitaufwand verschaffen.

4.1 Software-Architektur

Die Software-Architektur soll die Bauweise eines Systems abbilden und stellt den Ausgangspunkt eines erfolgreichen Systems dar. Es beschreibt die Vernetzung der Software- und Hardwaresegmente. Des Weiteren spielt die Platzierung, sowie die Zusammenarbeit und Anordnung der Softwarekomponenten eine wichtige Rolle. Welche Schnittstellen und Beziehungen stehen zwischen den Elementen, wie findet die Interaktionen zwischen Client und Server statt? All das sind Fragen, die bei der Entwicklung eines Systems bedacht werden müssen. [AST08]

4.1.1 Ablauf

DSN ist eine Web-Applikation, welche den Benutzern, im Speziellen Schülern, helfen soll, seine Mitschriften organisierter und einfacher zu verwalten. Doch was versteckt hinter einer so großen und komplexen Anwendung? Das folgende Diagramm soll den Ablauf zwischen dem Client und Server verdeutlichen. Daraus wird klar ersichtlich, welche Komponenten beim Anmelden eines Users zum Einsatz kommen. Das System wird folgendermaßen aufgebaut:

- **Ebene 1: Client**
Unser Anwender, der mittels Browser auf unserer Website surft und seine Mitschriften verwaltet.
- **Ebene 2: Web-Server**
Nginx nimmt HTTP-Anfragen (GET & POST) entgegen und überprüft die vom User übermittelten Parameter.
- **Ebene 3: Applikations-Server**
Hinter Ebene 3 verbirgt sich die eigentliche Geschäftslogik, die maßgebend für unser System ist. Bei DSN werden alle HTTP-Anfragen, die an /api/ gehen, an das Web-Framework Django weitergeleitet. Das ist sozusagen die Schnittstelle zwischen Datenbank und Client.

- **Ebene 4: Datenbank-Server**

Die Datenbank hat die Aufgabe, wichtige, geheimzuhaltende Daten zu persistieren und bei Anfragen schnell Antworten zu liefern.

Es muss sich nicht unbedingt um einen Datenbank-Server handeln, sondern es gibt die Möglichkeit, seine Daten auf mehreren Stationen aufzuteilen. Weiters können wichtige von unwichtigen Daten getrennt werden. Das steigert die Performance, sowie die Ausfallsicherheit.

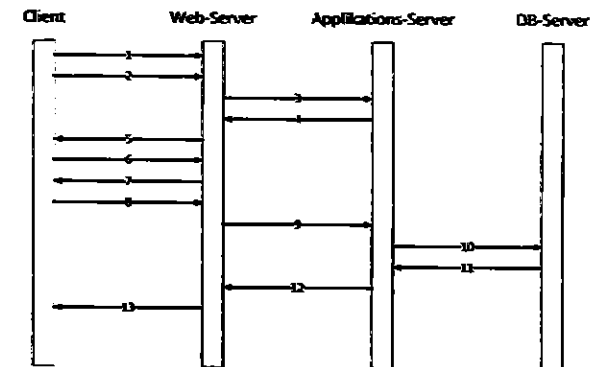


Abbildung 17: Ablaufdiagramm

1. Im ersten Schritt ruft der Anwender mittels einer GET-Anfrage die DSN-Webseite zum ersten Mal auf. Im Hintergrund werden alle notwendigen CSS- und JavaScript-Files geladen. *als Vollgröße Deutsch?*
2. Bevor die Hauptseite erscheint, überprüft der Server die Sprachauswahl. Es besteht die Wahl zwischen zwei Weltsprachen, nämlich Englisch oder Deutsch. Standardmäßig, wird ein POST an `api/change_lang` mit dem JSON Parameter `{language: "de"}` gesendet.
3. Da es sich um eine /api/ Funktion handelt, leitet der Nginx-Server, der nur für den statischen Teil zuständig ist, die Anfrage weiter zum Django-Server.
4. Hinter der gesendeten Adresse befindet sich eine Funktion, welche den Inhalt der DSN-Seite anhand der übergebenen Parameter auf die gewählte Sprache ändert.
5. Das Ergebnis, die Homepage in der gewählten Sprache, wird an den Client zurückgeliefert.
6. Im nächsten Schritt möchte sich der bestehende DSN-User beim System anmelden. Dafür klickt er auf den Login-Button, wodurch eine GET(/login) Anfrage an dem Server geschickt wird.

Wir machen keine Homepage Website oder Webseite

Bleibe so komplexe Prozesse auch für Nichterfahrene verständlich formulieren!

7. Mittels REST, Representational State Transfer, wird automatisch zum Login weitergeleitet. Eine erfolgreiche Anmeldung benötigt eine bereits registrierte Email-Adresse, sowie das mindestens 8 stellige Passwort.
8. Durch die Eingabe der Benutzerdaten schickt der Anwender seine Email-Adresse und das verschlüsselte Passwort als JSON-Objekt an den Web-Server. Das POST wird an api/login gesendet.
9. Dynamische Inhalte werden an Django weitergeleitet.
10. Hinter api/login verbirgt sich eine Funktion, die vom JSON-Objekt die Email-Adresse holt. Mit dieser Info stellt der Server bei der Datenbank die Anfrage, ob der User registriert ist und sich rechtmäßig anmelden darf.
11. Die MongoDB-Datenbank, bestehend aus mehreren Schemas, sucht in der Collection *user*. Im Falle eines positiven Ergebnisses, wird der gefundene User an den Web-Server zurückgegeben.
12. Aufgrund des komplizierten Umgangs mit JSON-Objekten wird das Datenbankergebnis mit einem User-Objekt gemappt. Durch den objektorientierten Ansatz kann ohne viel Aufwand das Passwort kontrolliert werden. Sind alle Überprüfungen fehlerlos, wird der User angemeldet.
13. Je nach Response erfolgt die Weiterleitung auf die Management-Page oder es erscheint eine Fehlermeldung.

4.1.2 DSN Architektur

Unser System setzt sich aus einem Frontend-Server, einem Backend-Server und einer dokumentenbasierten Datenbank zusammen. Das Frontend ist für die Darstellung statischer Daten verantwortlich. Im Gegensatz dazu kümmert sich das Backend mit MongoDB um den dynamischen Teil.

4.1.2.1 Interaktionen zwischen Frameworks

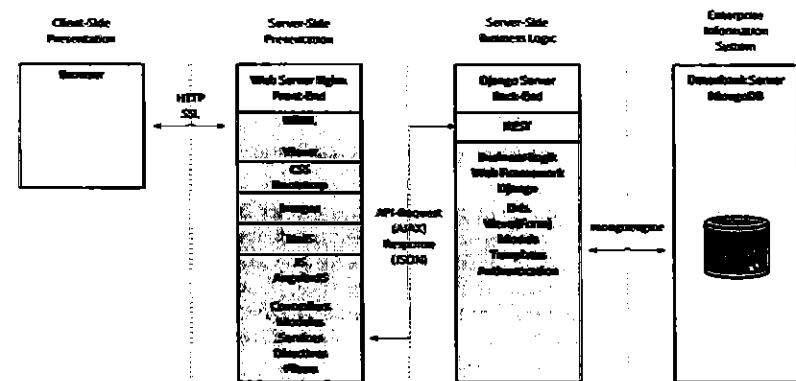


Abbildung 18: Technologische Übersicht

Der Client kann über HTTP oder SSL die DSN-Webseite aufrufen. Je nach Anfrage wird eine Verbindung mit dem Frontend- oder Backend-Server aufgebaut.

Der Nginx-Server welcher für das Frontend zuständig ist, beinhaltet statische Daten, wie HTML-Seiten, CSS- & JS-Dateien und Bilder. Diese werden durch ein JavaScript File namens *routes.js* verwaltet. Eingehende HTTP-Requests werden in diesem File gemappt. Fordert ein User z.B. mit einem GET die Loginpage, wird auf diese weitergeleitet. Jede HTML-Seite hat ihren eigenen Controller, basierend auf JavaScript. Er kümmert sich um die Useraktionen und ändert je nach Anforderung den Inhalt der Website.

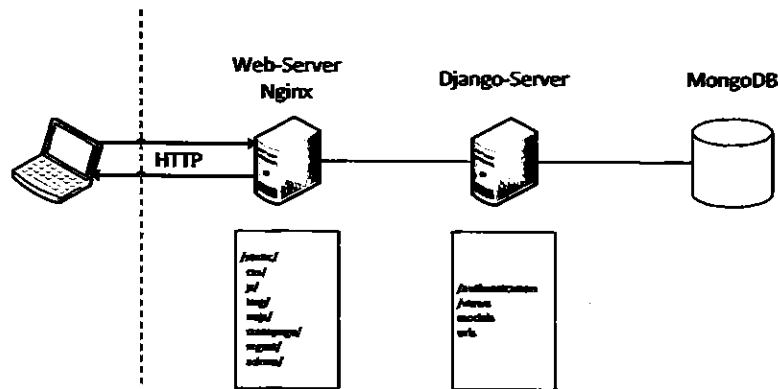


Abbildung 19: Aufbau der Server

Sollte es zu komplexeren Aufgaben kommen, wo statischer Inhalt keine Hilfe ist, wird die Kommunikation mit dem Django-Backend-Server erforderlich. Django ist ein Web Framework basierend auf Python. Wie beim Frontend, existiert ein File, das alles managert. Die `urls.py` Datei nimmt HTTP-Anfragen entgegen und delegiert diese auf die jeweiligen Funktionen im `views` Ordner. Diese hantieren mit den empfangenen JSON-Objekten. Handelt es sich bei der Anfrage um die Auflistung von Schulheften oder Heftinhalten, ist eine DB-Abfrage erforderlich. Dank dem `models.py` File liefert MongoDB kein JSON, sondern spezifizierte Objekte, die die Arbeit erleichtern.

4.2 Graphische Oberfläche

Eine intuitive Oberfläche ist besonders wichtig, um Benutzern eine einfache und schnelle Verwendung der Applikation zu ermöglichen. Daher wurde beim Erstellen eines Designs für die grafische Oberfläche darauf besonders viel Wert gelegt.

Um eine ansprechende grafische Oberfläche erstellen zu können, wurde das CSS-Framework *Bootstrap* eingebunden und verwendet.

4.2.1 Layout

Für die Applikation sollte ein möglichst einfaches Layout erstellt werden, sodass nicht zu viel Inhalt auf nur einer Seite zu sehen ist, ~~damas könnte die Benutzer überfordern.~~ *könnale* Daher folgt das Layout folgendem Entwurf: *was die*

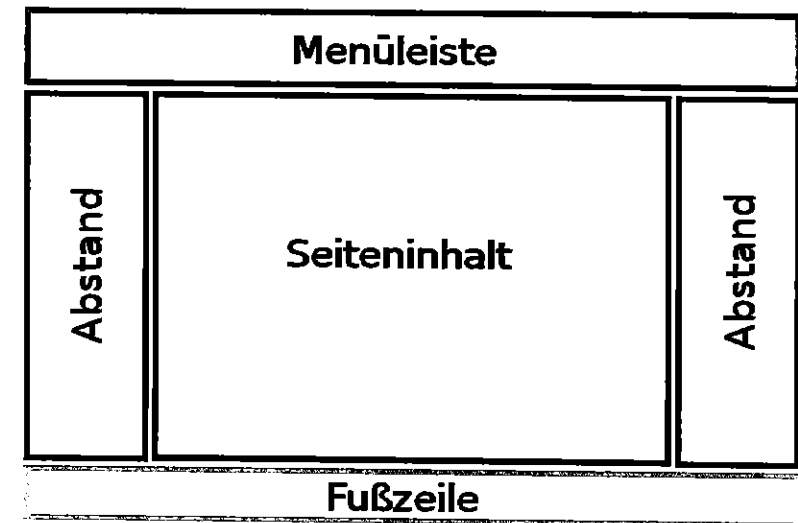


Abbildung 20: Layout der grafischen Oberfläche

Die Menüleiste ist, wie zu sehen, am oberen Rand zu finden, die Fußzeile am unteren Rand. Außerdem ist auf der Webseite nur noch der eigentliche Seiteninhalt zu sehen, wobei auch dieser möglichst einfach gehalten wird. Dadurch, dass der Seiteninhalt die Mitte der Webseite beansprucht, wird der Fokus der Benutzer vor allem darauf gelenkt.

Umgesetzt wurde dieses Layout mithilfe von Bootstrap. Bootstrap bietet eine Möglichkeit, das Layout responsive zu gestalten. Dies geschieht mit Reihen und Spalten, in die das Layout aufgeteilt wird. Dabei beinhaltet jede Reihe zwölf Spalten. Diese Spalten passen sich je nach verfügbaren Pixeln automatisch an den jeweiligen verfügbaren Platz an. Das obige Layout würde dann aus drei Reihen bestehen: Menüleiste, Abstände und Seiteninhalt, sowie Fußzeile. Menüleiste und Fußzeile würden jeweils volle zwölf Spalten einnehmen. Die mittlere Reihe würde aufgeteilt werden. Beispielsweise könnten zwei Spalten für den linken Abstand, acht Spalten für den Seiteninhalt und wieder zwei Spalten für den rechten Abstand vergeben werden.

Durch den Aufbau mittels Reihen und Spalten wird das Layout auf allen Computern sehr ähnlich dargestellt. Das Design bleibt überall erhalten. Zudem ist die Unterstützung für Handys und Tablets integriert. Das System der Reihen und Spalten funktioniert auf allen Geräten, auch auf Handys und Tablets wird die Breite der Spalten so angepasst, dass das Layout so gut wie möglich erhalten bleibt.

4.2.2 Einfache Menüstruktur

Mithilfe einer einfachen Menüstruktur finden sich Benutzer auf der Webseite sofort zu recht und müssen nicht lange nach der gewünschten Funktion suchen. Eine möglichst einfache Menüstruktur wurde umgesetzt, indem es keinerlei Untermenüs gibt. Die Menüleiste besteht aus einzelnen Einträgen, bei denen sofort erkenntlich ist, welche Funktion sie haben.

besser ..
als flache
Hierarchie
implementiert ..



Abbildung 21: Menüleiste der Hauptseite

Die Menüleiste der Hauptseite ist auf drei Menüpunkte beschränkt. Klickt der Benutzer auf *Funktionen* oder *Preise*, wird auf der Seite automatisch gescrollt, bis der jeweilige Inhalt zu sehen ist. Diese beiden Menüpunkte dienen dazu, den Benutzer über die Applikation zu informieren. Der Menüpunkt *Anmelden* dient Benutzern, die sich bereits an der Applikation registriert haben, dazu, sich an der Applikation anzumelden, um sie verwenden zu können. Dieser Menüpunkt ist mit dunklerer Farbe hervorgehoben, um Benutzer sofort darauf aufmerksam zu machen.



Abbildung 22: Menüleiste nach dem Anmelden

Sobald sich ein Benutzer an der Applikation angemeldet hat, wird obige Menüleiste angezeigt. Nach dem Anmelden wird der Benutzer auf den Menüpunkt *Stundenplan* geleitet. Über die Menüpunkte *Hefte* und *Kontoeinstellungen* kann der Benutzer in der Applikation navigieren. Zudem ist in dieser Menüleiste ein Suchfeld, um nach anderen Benutzern der Applikation zu suchen, sowie ein Menüpunkt *Abmelden*, um sich von der Applikation abmelden zu können.



Abbildung 23: Menüleiste der Admin-Seite

Ein Administrator hat die Möglichkeit, die oben abgebildete Menüleiste auf der Admin-Seite zu sehen. Hier gibt es nur zwei Menüpunkte: *User Management*, um Benutzer der Applikation verwalten zu können und *Logout*, um sich von der Applikation abzumelden.

Jedes der oben beschriebenen Menüs hat, wie zu sehen, keine Unterpunkte und nur sehr wenige Menüpunkte. Dadurch wird ein schnelleres Navigieren innerhalb der Webseite und eine einfachere Verwendung der Applikation ermöglicht.

4.2.3 Unterstützende Bildzeichen

Unterstützende Bildzeichen, in Bootstrap *Glyphicons* genannt, werden überall in der Applikation eingesetzt. Mithilfe dieser Symbole können Benutzer bestimmte Funktionen schneller identifizieren und damit verwenden.



Abbildung 24: Glyphicons innerhalb der Applikation

Abbildung 24 zeigt ein Beispiel, wie Glyphicons innerhalb der Applikation verwendet werden. Hier werden zusätzlich zu den Menüpunkten einer Menüleiste unterstützende Bildzeichen eingesetzt. Das Symbol eines Kalenders soll den Stundenplan darstellen, ein Heft soll auf die Ansicht aller Hefte hindeuten und das Zahnrad-Symbol unterstützt die Funktion der Kontoeinstellungen.

Mithilfe dieser Symbole wird den Benutzern eine schnellere Navigation innerhalb der Webseite ermöglicht.

4.2.4 Heftansicht

Die Heftansicht dient der Verwendung der Kernfunktion der Applikation, daher ist ein passendes Design für diese Ansicht besonders wichtig.

Das Design für die Heftansicht wurde zu Beginn des Projektes wie in Abbildung 25 gestaltet.

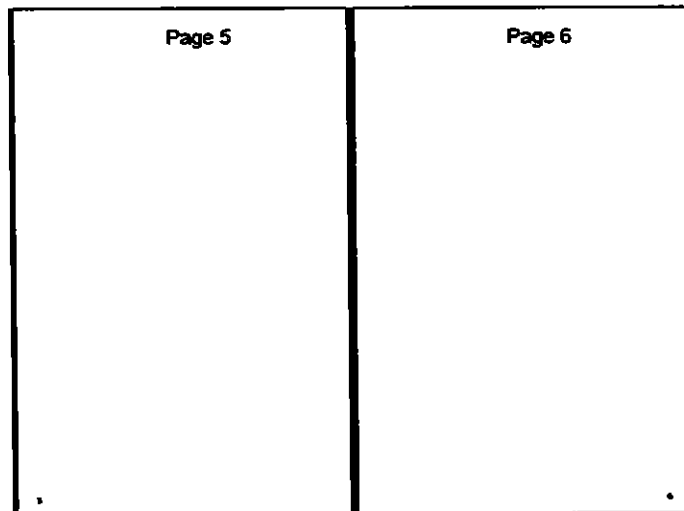


Abbildung 25: Frühes Design der Heftansicht

Zu späterem Zeitpunkt hat sich allerdings herausgestellt, dass dieses Design zu einigen Problemen führt. Das größte Problem dabei war, dass die Ansicht des Heftes dieses Designs zu klein war. Ein effektives Arbeiten innerhalb des Heftes wäre für Benutzer kaum möglich. Daher musste ein neues, verbessertes Design erstellt werden. Nach einem Redesign kam eine Ansicht wie in Abbildung 26 zu sehen zustande.

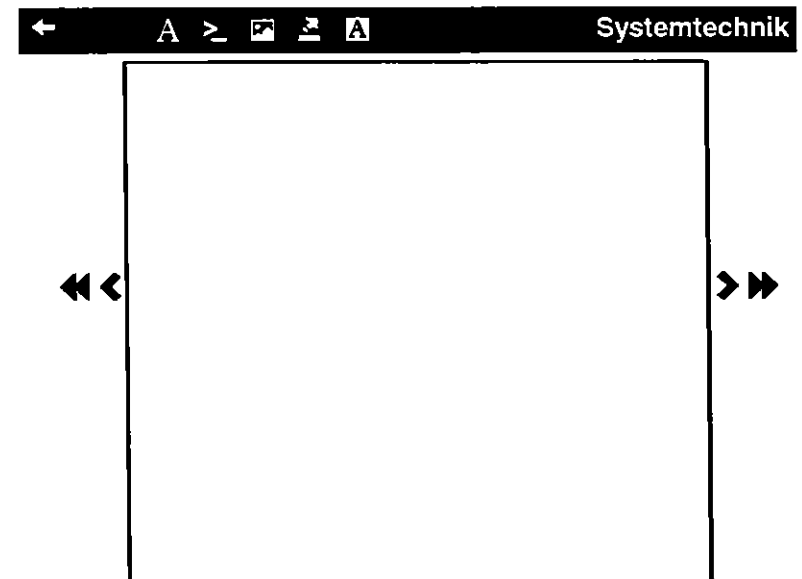


Abbildung 26: Verbessertes Design der Heftansicht

Das neue Design hat eine verbesserte Menüleiste, um Heftelemente innerhalb des Heftes einfügen zu können. Außerdem wurde die Ansicht des Heftes deutlich vergrößert. Dadurch kann den Benutzern ein effektiveres Arbeiten und eine bessere und einfachere Verwendung der Heftfunktionen ermöglicht werden.

✓

4.3 Javascript Optimierung

Im Laufe des Projektes gab es immer mehr Probleme im Zusammenhang mit Javascript. Einfache Befehle wie `console.log()` funktionierten nicht mehr ordnungsgemäß. Das Problem konnte bis zum Ende des Projektes nicht behoben werden, jedoch soll dieses Kapitel für zukünftige Projekte hilfreich sein.

4.3.1 Code Style

In Javascript gibt es meist viele Möglichkeiten den selben Code zu schreiben. Dadurch ist es schwierig, sich in neue Frameworks oder anderes einzuarbeiten. Wenn ein bestimmtes Problem auftritt und eine Lösung im Internet gesucht wird, ~~dann~~ kann ein Anfänger nicht unterscheiden, ob diese Lösung nun nur eine andere Schreibweise einer anderen Lösung ist oder ein komplett anderer Ansatz.

Es hat natürlich auch Vorteile eine Sprache so schwach reguliert zu gestalten. Jeder Programmierer kann sich einen Stil aussuchen, der ihm gefällt und mit welchem er gut arbeiten kann. Des Weiteren sind unterschiedliche Schreibweisen des Codes auch unterschiedlich effektiv.

Doch ist etwas wirklich nötig? Es gibt genug Sprachen in welchen ganz genau festgelegt ist, wie etwas durchzuführen ist. Javascript gehört hier leider nicht dazu. Es würde auf jeden Fall das Verständnis der Sprache vereinfachen.

```
function A(){};           // function declaration
var B = function(){};     // function expression
var C = (function(){});   // function expression with grouping operators
var D = function foo(){}; // named function expression
// immediately-invoked function expression (IIFE) that returns a function
var E = (function(){
    return function(){}
})();
var F = new Function();   // Function constructor
var G = new function(){}; // special case: object constructor
```

Listing 23: Unterschiedliche Möglichkeiten eine Funktion zu deklarieren[Cal11]

Doch wie kann dieses Problem nun umgangen werden? Am besten ist es, sich am Anfang eines Projektes auf eine Schreibweise für z.B. Funktionsdeklarationen zu einigen. Dies macht zumindest den Code im eigenen Projekt um einiges vergleichbarer. Wenn sich das Team einen gemeinsamen Code Style aneignet, ~~dann~~ kann der bereits verfasste Code besser wiederverwendet und verstanden werden.

4.3.2 Debugging

Der Javascript Code im Projekt wurde größtenteils mit `console.log()` und `alert()` debuggt. Da diese beiden Befehle nach einer gewissen Zeit nicht mehr zuverlässig funktionierten, wurde es um einiges schwieriger Fehler aufzuspüren. Die Befehle wurden manchmal richtig ausgeführt, manchmal nicht an der Stelle an welcher sie geschrieben wurden und manchmal wurden sie gar nicht beachtet.

Hier sollten die Debugger in den Browsern weiterhelfen. Jeder moderne Browser, wie z.B. Firefox oder Chrome, hat einen solchen eingebaut. Mit diesem lassen sich wie in jedem anderen Debugger Breakpoints setzen. In Folge dessen kann das Programm dann Schritt für Schritt durchgestept werden. Des Weiteren kann auch der Inhalt der Variablen eingesehen werden.

Ein Firefox Plugin namens Firebug unterstützt diese Variablenüberprüfung und einige andere Features, welche es in normalen Browser Debuggern nicht gibt und sollte daher unbedingt in zukünftigen Projekten verwendet werden. Hiermit hätte während des Projektes nach einer gewissen Einarbeitungsphase einiges an Zeit gespart werden können. *Beim nächsten Projekt Bar. fragen! :)*

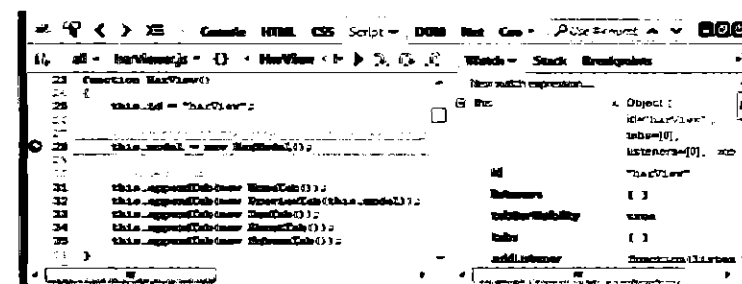


Abbildung 27: Firebug Javascript Debugger

Die Fehlfunktion von `console.log()` und `alert()` erklärte sich das Team durch die vielen eingebundenen Libraries. Eine dieser Libraries könnte diese Funktion vielleicht überschrieben haben.

4.3.3 Scripteinbindung

Je weniger Scripts eingebunden werden müssen desto besser. Müssen jedoch, wie in diesem Projekt relativ, viele Javascript Frameworks und Scripts eingebunden werden, dann sollten die Imports organisiert werden und ein paar Regeln folgen.

Falls möglich sollten die Scripts über ein CDN eingebunden werden, um die Performance zu erhöhen. Des Weiteren wird damit auch Bandbreite und Traffic gespart.

Alle Scripts sollten mit der gleichen Schreibweise des Script Tags eingebunden werden, um die Imports übersichtlich zu halten.

Falls ein Framework aus vielen Javascript Files besteht, dann sollten diese, falls möglich, in weniger Files zusammengefasst werden. Dadurch werden die Imports übersichtlicher und die Scripts können schneller geladen werden.

4.3.4 Optimierung

Um die Ladezeit der Scripts zu optimieren, sollten diese komprimiert werden. Dies geschieht, indem alle Leerzeichen, Absätze und Kommentare entfernt werden. Dadurch kann erheblich die Dateigröße reduziert und somit die Übertragungszeit minimiert werden.

Scripts, welche nicht sofort benötigt werden, sollten nachgeladen werden. Dies kann mit den *async* und *defer* Statements bei der Scripteinbindung erreicht werden. Doch Achtung: nicht bei allen Scripts funktioniert das. Wenn die falschen Scripts nachgeladen werden, ~~dann~~ funktioniert die ~~ganze~~ Seite nicht mehr.

gesamte

5 Implementierung

Dieses Kapitel befasst sich mit der eigentlichen Implementierung des Projektes. Hier beschreibt jedes Teammitglied sein Spezialgebiet, welches auf Wunsch des Ministeriums festgelegt wurde.

5.1 Infrastruktur und Testing

Infrastruktur und Testing sind für jedes Projekt essentiell. Die Infrastruktur ist notwendig, damit das gesamte Team mit den gleichen Versionen der Frameworks arbeiten kann und um eine stabile Version der Software hosten zu können. Tests werden benötigt, um die Funktionalität der Software zu überprüfen.

5.1.1 Infrastruktur

Eine stabile und sichere Infrastruktur ist heutzutage ein Muss für jedes IT Projekt. Die Infrastruktur ist wichtig, da in der Vergangenheit oft kleine Projekte bereits wenige Tage nach Veröffentlichung von sehr hohen Userzahlen berichten konnten. Wenn hier zuvor die Infrastruktur gut geplant und implementiert wurde, ist es kein Problem viele User zu bewältigen.

5.1.1.1 Serverhosting

Die wichtigste technische Grundlage für das Projekt DigitalSchoolNotes ist der Projektserver. Auf diesem Server, wird das Projekt entwickelt und getestet. Hier ist es besonders wichtig, dass das gesamte Team mit der gleichen Umgebung arbeitet, da sonst die einzelnen Codeteile des Teams nicht zusammen funktionieren. Des Weiteren wird der Server dazu verwendet, die Zwischenversionen des Projektes öffentlich verfügbar zu machen. Dies ist für das Team essentiell, da dadurch der Stakeholder jederzeit Zugriff auf eine aktuelle und stabile Version des Projektes hat. Dadurch kann das Team Änderungswünsche des Stakeholders leichter erfassen und realisieren.



Für die Auswahl des Serverhosters wurden einige Kriterien festgelegt. Diese lauten wie folgt:

- **Serverstandort:** Der Standort des Projektserver sollte möglichst nahe beim Endbenutzer sein, um die Latenz gering zu halten.
- **Verfügbarkeit:** Der Server sollte eine hohe Mindestverfügbarkeit haben. Dadurch kann sich der Endbenutzer darauf verlassen, dass das Service erreichbar ist. Der Minimalwert für die Verfügbarkeit wurde auf 99,6% festgelegt. Das bedeutet, dass der Server für maximal 35h im Jahr nicht verfügbar ist.
- **Support:** Der Hoster sollte Support unter der Woche und in Notfällen rund um die Uhr bieten.
- **Preis:** Um die Entwicklungskosten möglichst gering zu halten, wurde der maximale Monatspreis auf 10 Euro festgelegt.
- **Wartung:** Der Server sollte sich über ein Webinterface warten lassen.

Die oben genannten Kriterien reduzierten die Anzahl der möglichen Hoster stark. Das Team entschied sich für den Hoster netcup GmbH mit Sitz in Deutschland. Dieser erfüllte alle Anforderungen und Teile des Teams hatten bereits gute Erfahrungen mit dieser Firma gemacht.

Das ausgewählte Produkt der netcup GmbH heißt "Root-Server M v6". Dieser bietet folgende Features:

- **Virtualisierungstechnik:**KVM
- **CPU:**Intel®Xeon® E5-26xxV3 2,3GHz 2Cores
- **RAM:**6GB DDR4
- **Speicher:**120GB SSD

5.1.1.2 Erreichbarkeit

Der Server ist unter der IP-Adresse 37.120.161.195 erreichbar. Da IP-Adressen schwer zu merken sind, wurde ebenfalls eine Domain für das Projekt gekauft. Diese lautet "digitalschoolnotes.com" und löst auf die oben genannte IP-Adresse auf.

5.1.1.3 Benutzerverwaltung am Projektserver

Jedes Projektteam Mitglied hat einen eigenen Unix Account auf dem Projektserver. Der Vorname der Person ist der Benutzername. Das Benutzerpasswort ist von jedem Teammitglied selbst gewählt. Alle Teammitglieder haben sudo Rechte.

5.1.1.4 Mailsystem

Das Projektteam hat einen Email-Verteiler mit der Adresse info@digitalschoolnotes.com. Jedes Teammitglied hat eine E-Mail Adresse nach dem Schema des TGMS(z.B. nho-henwarter@digitalschoolnotes.com). Der Scrummaster ist unter scrummaster@digitalschoolnotes.com erreichbar.

5.1.1.5 Serverzugriff

Um den Server zu konfigurieren und zu verwalten, wird mit dem Protokoll SSH darauf zugegriffen. Aus Sicherheitsgründen wurde die Anmeldung mit Passwort verboten und es können hierfür nur noch SSH Keys verwendet werden. Diese sind um einiges sicherer.

5.1.1.6 Firewall

Um den Server vor Angriffen und unerwünschten Zugriffen zu schützen, wurde eine Firewall installiert. Diese blockiert alle unerwünschten Anfragen. Prinzipiell sind alle Ports geschlossen. Es werden nur Ports geöffnet, welche für das Betreiben des Projektes notwendig sind.

Es folgt eine Liste der freigegebenen Ports:

- 22 SSH
- 53 DNS
- 80 HTTP
- 443 HTTPS
- 5001-5005 Django Development

Die Konfiguration der Firewall des Projektserver sieht wie folgt aus:

```
# Flush the tables to apply changes
iptables -F

# Default policy to drop 'everything' but our output to internet
iptables -P FORWARD DROP
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT

# Allow established connections (the responses to our outgoing traffic)
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Allow local programs that use loopback (Unix sockets)
iptables -A INPUT -s 127.0.0.0/8 -d 127.0.0.0/8 -i lo -j ACCEPT
iptables -A FORWARD -s 127.0.0.0/8 -d 127.0.0.0/8 -i lo -j ACCEPT

#Allowed Ports
iptables -A INPUT -p tcp --dport 22 -m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp --dport 53 -m state --state NEW -j ACCEPT
iptables -A INPUT -p udp --dport 53 -m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp --dport 5001 -m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp --dport 5002 -m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp --dport 5003 -m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp --dport 5004 -m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp --dport 5005 -m state --state NEW -j ACCEPT
```

Listing 24: Firewall Rules des Projektserver

Da normalerweise nach einem Reboot des Servers die Firewallkonfiguration verloren geht, musste diese persistiert werden. Das wird durch das Paket *iptables-persistent* erledigt. Die Konfiguration dieses Paketes geschieht wie folgt[Kre15b]:

```
# Install
sudo apt-get install iptables-persistent

# Save Rules
iptables-save > /etc/iptables/rules.v4
```

Listing 25: Installation iptables-persistent

5.1.1.7 Bruteforce Prevention

Um Bruteforce Angriffe auf den SSH Dienst zu erschweren, wurde am Server fail2ban eingerichtet. Dieses Tool zählt fehlgeschlagene Anmeldeversuche mit und sperrt die IP Adresse des Angreifers nach einer festgelegten Anzahl an Versuchen. Dieses Verfahren ist äußerst effektiv, da der Angreifer dadurch keine Chance hat, eine große Anzahl an Passwörtern auszuprobieren(z.B. Wörterbuchangriff). Da auf dem Projektserver die Anmeldung nur mit SSH Key möglich ist, hat der Client, welcher sich zum Server verbinden will, sechs Versuche einen korrekten SSH Key zu übermitteln.

5.1.1.8 Webserver

Als Webserver für unsere Applikation wurde Nginx gewählt. Dieser wurde vor allem gewählt, da das Team bereits in der Vergangenheit mit dieser Software gearbeitet hat. Mithilfe von Nginx kann ebenfalls ein Loadbalancer realisiert werden. Dies ist ein wichtiger Punkt, um die Software skalierbar zu halten.

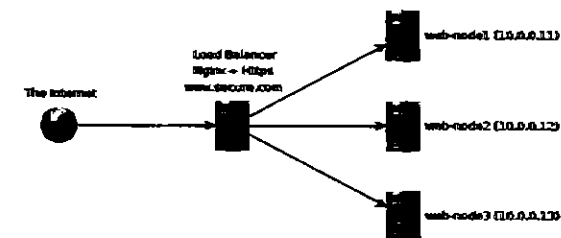


Abbildung 28: Loadbalancing

Der Webserver ist hauptsächlich für den statischen Content(HTML, Javascript, CSS, Bilder...) zuständig. Dies funktioniert, indem alle statischen Inhalte in einem Ordner abgelegt werden. Damit weiß Nginx, dass er für diese Inhalte zuständig ist.

5.1.1.9 SSL

Um die Daten und Privatsphäre unserer Kunden zu schützen, wird bei allen Aufrufen der Website mit SSL verschlüsselt. Um eine legitime SSL Verschlüsselung zu gewährleisten, ist ein valides Zertifikat notwendig. Dieses muss von einer Zertifizierungsstelle erworben werden. Das verwendete Zertifikat für das Projekt wurde von der Zertifizierungsstelle Namens thawte Inc. erworben.

Das Zertifikat validiert die Domain (Domain Validated). Das bedeutet, dass zur Ausstellung des Zertifikates eine Email an den Besitzer der Domain geschickt wird. Wenn der Besitzer der Domain der Zertifizierung zustimmt, wird diese durchgeführt. ✓

Um das Zertifikat nun verwenden zu können, muss es mit dem Intermediate Zertifikat der Zertifizierungsstelle verbunden werden. Dadurch ist ein eindeutiger Zertifizierungsfluss hergestellt. Danach kann es auf den Webserver deployed werden. ✓

5.1.1.10 Produktivbetrieb

Im Produktivbetrieb ist der Betrieb der Software auf zwei Dienste aufgeteilt. Der statische Teil der Applikation wird, wie bereits vorhin beschrieben, von Nginx dem User zur Verfügung gestellt.

Der dynamische Teil - das Backend - stellt unser Django Server dar. Hier werden die kritischen Operationen, wie Datenbankzugriffe oder die Authentifizierung, durchgeführt. Wird nun unsere API (der dynamische Teil) aufgerufen, leitet Nginx die Anfrage an den Django Server weiter. ✓

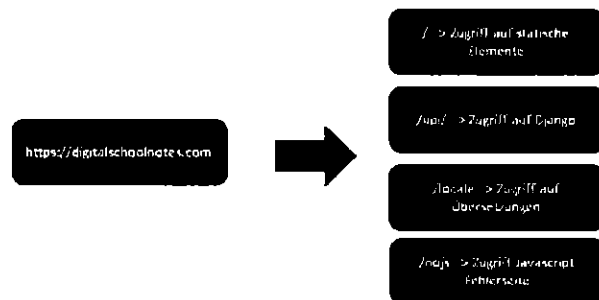


Abbildung 29: Deployment Struktur

Der Django Server läuft mittels gunicorn. Gunicorn ist ein WSGI HTTP Server und somit die Schnittstelle zwischen dem Webserver und Django. Gunicorn startet für Django mehrere Worker Prozesse, wodurch die Anfragen theoretisch auf mehrere CPU Kerne aufgeteilt werden können. Des Weiteren startet es die einzelnen Prozesse automatisch neu, falls diese abstürzen. ✓

5.1.1.11 Testbetrieb

Der Testbetrieb läuft relativ ähnlich wie der Produktivbetrieb ab. Jedes Teammitglied arbeitet an einer eigenen Instanz des Codes. Dadurch behindert sich das Team nicht gegenseitig, falls Fehler auftreten. Um dies zu ermöglichen, hat jede Person einen eigenen Port zugewiesen bekommen, auf der seine Version der Applikation zu erreichen ist. ✓

Hat diese Person nun eine Änderung am Code vorgenommen, muss dieser auf den Server hochgeladen werden. Nun kann innerhalb der IDE der Django Server gestartet und beendet werden. Dies ist wichtig, da dadurch auch die Fehlermeldungen von Django innerhalb der IDE sichtbar sind. Für den statischen Teil ist auch hier Nginx zuständig.

5.1.1.12 Verfügbarkeit

Um in Zukunft die Verfügbarkeit zu verbessern, gibt es viele Möglichkeiten. Um die Wahrscheinlichkeit eines kritischen Serverausfalles zu reduzieren könnten, mehrere Server angemietet werden. Eine andere Möglichkeit wäre es, das Hosting in eine Cloud auszulagern (Amazon Web Services, Google, Azure...). ✓

Um die Performance und Verfügbarkeit zu erhöhen, sollte ein Loadbalancer verwendet werden. Dieser teilt die eingehenden Anfragen auf mehrere andere Server auf und reduziert damit die Last der einzelnen Server und erhöht die Performance. Hierbei ist darauf zu achten, dass der Loadbalancer Healthchecks durchführen muss. Mithilfe dieser kann der Loadbalancer überprüfen, ob die Server noch aktiv sind und somit Anfragen entgegennehmen und bearbeiten können. Sollte ein Server ausfallen, leitet der Loadbalancer keine Anfragen mehr an diesen weiter. Der Loadbalancer sollte redundant ausgeführt sein, da ohne ihn die Seite nicht mehr erreichbar ist. *oder setzt Wartungsdauern*

Um die Performance der Datenbank zu erhöhen, sollte diese auf einem eigenen Server betrieben werden. Des Weiteren sollte hier Loadbalancing oder Clustering betrieben werden, um die Ausfallsicherheit zu erhöhen.

Die Applikation sollte in Containern (z.B. Docker) mit Containermanagement deployed werden, um die Verfügbarkeit zu erhöhen. Somit könnte bei einem Fehler der Container einfach entfernt und ein Neuer gestartet werden.

Um die Erreichbarkeit der Domain zu gewährleisten, sollten redundante DNS Server verwendet werden. Hier gibt es auch die Möglichkeit, ein CDN Network (z.B. Cloudflare, Cloudfront...) zu verwenden, um einen Performancevorteil zu erlangen. Durch das CDN wird auch die Serverlast reduziert.

5.1.2 Testing

Um eine Einwandfreie Benutzung einer Software zu ermöglichen, muss diese getestet werden. Mittels Tests können Fehler gefunden und repariert werden. Um Zeit zu sparen, werden diese Tests mit Testing-Frameworks automatisiert.

5.1.2.1 Testing Level

Beim Softwaretesting wird in zwei Kategorien eingeteilt: Functional Testing und Non-functional Testing. Functional Testing testet definierte Spezifikationen. Damit soll sichergestellt werden, dass die Applikation sich so verhält, wie erwartet. Non-functional Testing testet Dinge wie Performance und Erreichbarkeit. [tut15]

Das Functional Testing lässt sich in vier Level aufteilen, welche auch in folgender Grafik dargestellt werden.

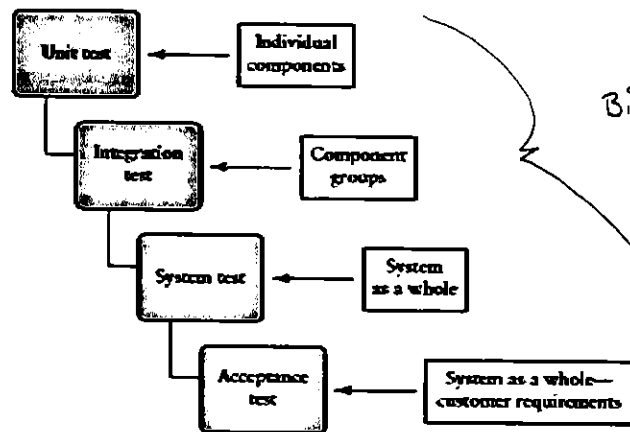


Abbildung 30: Functional Testing Level

Mithilfe von Unit Tests wird die Funktionalität des Programmes überprüft. Hier werden Methoden oder Programmteilen - sogenannten Units - Daten übergeben. Diese werden dann verarbeitet und das Ergebnis wird überprüft. Diese Tests werden bereits während der Entwicklung erstellt und stellen sicher, dass neue Änderungen am Code den bereits vorhandenen Code nicht kompromittieren. [Pea15]

Integration Tests prüfen die Kompatibilität zwischen den einzelnen Units. Es kann sein, dass Units alleine funktionieren, jedoch zusammen mit anderen Units nicht. Des Weiteren wird hier auch die Performance überprüft. Jede einzelne Unit kann effizient sein, sind die Units jedoch schlecht integriert, entsteht trotzdem ein ineffizientes Programm. [Pea15]

Bei System Tests wird die ganze Applikation getestet. Hier ist es das Ziel, die Software auf Erfüllung der Qualitätsstandards und Anforderungen zu testen. Diese Tests werden von eigenständigen Testern durchgeführt, welche nicht an der Entwicklung des Programmes beteiligt waren. [Pea15]

Acceptance Tests sind das letzte Test Level. Hier wird geprüft, ob die Softwareversion für eine Veröffentlichung bereit ist. Dazu müssen die Endanwender ausprobieren, ob die Software wie gewünscht funktioniert. [Pea15]

5.1.2.2 Testerstellung

Die einzelnen Tests werden in Java programmiert. Beim Ausführen der Tests wird ein Browser gestartet, welcher dann die gewünschten Websiteaufrufe tätigt. Tests können zu Test Suites zusammengefasst werden. Dadurch werden alle Tests in der Suite nacheinander ausgeführt.

Um im Browser auf ein Element mit Selenium klicken zu können, muss dieses eindeutig identifizierbar sein. Hierbei gibt es mehrere Möglichkeiten. Eine eindeutige Identifizierung ist über ID, Name, CSS Klasse, Link Text oder XPATH möglich.

```
driver.findElement(By.id("email")).click();
driver.findElement(By.name("email")).clear();
driver.findElement(By.className("cssclass")).click();
driver.findElement(By.partialLinkText("Klick mich")).click();
driver.findElement(By.xpath("//*[contains(text(), 'Mail senden')]")).click();
```

Listing 26: Selenium Element Selektoren

Selenium kann auch Formulare ausfüllen. Dazu muss das Element eindeutig angesprochen werden und dann kann wie oben im Beispiel statt dem `.click()` ein `.sendKeys()` verwendet werden.

Ein kompletter Test sieht dann wie folgt aus:

```
public void testInvalidUser() throws Exception {
    driver.get(baseUrl + "/login");
    driver.findElement(By.name("email")).clear();
    driver.findElement(By.name("email")).sendKeys("testinactive@test.test");
    driver.findElement(By.name("pwd")).sendKeys("12341234");
    driver.findElement(By.id("submit")).click();
    Thread.sleep(Parameters.SLEEP_PAGELOAD);
    String page = driver.getPageSource();
    if (!page.contains("Bitte bestaetige zuerst deine E-Mail Adresse!")) throw
        new NotFoundException();
}
```

Listing 27: Selenium Test

5.1.2.3 Probleme

Die Verwendung des Selenium Frameworks bereitet einige Probleme. Selenium kann nur mit Elementen interagieren, welche auch sichtbar sind. Dadurch können z.B. Buttons, welche nur bei Hover angezeigt werden, schwer getestet werden. Hierfür muss Selenium zuerst die Maus in die Hover Region bewegen und kann erst dann den Button drücken

Um ein Element testen zu können, muss es eindeutig identifizierbar sein. Dies ist allerdings bei vielen Elementen in unserer Software nicht möglich. Zum Beispiel sind die einzelnen Buttons im Texteditor im Heft nicht eindeutig identifizierbar. Diese können also nicht automatisiert getestet werden. Stattdessen wurden hier geführte Acceptance Tests durchgeführt. Das bedeutet, dass Selenium die Seite aufruft, eine Anweisung an den User gibt und danach fragt, ob das gewünschte Ergebnis eingetroffen ist.

Die auf DSN verwendeten Captchas bei der Registrierung und dem Passwort Reset können ebenfalls nicht automatisiert getestet werden. Captchas sollen im Endeffekt die automatische Ausfüllung eines Formulars ja auch verhindern. Bei einem Test füllt Selenium das Formular bis auf das Captcha aus. Das Captcha muss dann händisch gelöst werden.

Die Tests werden relativ langsam ausgeführt. So dauert es ca. 30 Minuten alle DSN Tests auszuführen. Dies liegt vor allem an der Netzwerklatenz der einzelnen Websiteaufrufe. Die Tests könnten parallelisiert auf mehreren Servern ausgeführt werden, jedoch ist die Einrichtung eines solchen Testing Clusters kompliziert.

5.2 User und Rollenmanagement

Usermanagement steht für das Erzeugen, Verwalten und Löschen von Benutzerkonten. Es soll dazu dienen, jeden registrierten User eindeutig zu identifizieren und zu kontrollieren, z.B. ob die monatliche Rate für den Pro-Account überwiesen wurde. Außerdem soll die bereits in Anspruch genommene Speicherkapazität überwacht werden. Dazu ist es notwendig, dass jeder DSN-User, durch eine Kombination von Daten, differenzierbar ist.

5.2.1 Authentisierung

Authentisierung bedeutet Nachweis der behaupteten Identität. Im Falle von DSN handelt es sich hierbei um die eindeutige Email-Adresse, welche einmalig im System verwendet wird. Es soll Gewissheit darüber geben, von wem die Information stammt. Dadurch kann jedes Handeln jemandem zugewiesen werden. Falls die vorgegebenen Richtlinien nicht eingehalten werden, kann DSN sofort reagieren.

5.2.1.1 Registrierung

Neben der Email-Adresse ist ein weiterer Identitätspunkt das geheime Passwort. Aus Sicherheitsgründen muss es mindestens 8 Zeichen beinhalten, wodurch Cyberkriminellen das Knacken von Passwörtern erschwert wird. Zum Abschluss der Registrierung müssen noch die Nutzungsbedingungen akzeptiert werden.

„Allgemeine Geschäftsbedingungen (AGB) sind vertragliche Klauseln, die zur Standardisierung und Konkretisierung von Massenverträgen dienen. Sie werden von einer Vertragspartei einseitig gestellt und bedürfen daher einer bes. Kontrolle, um ihren Missbrauch zu verhindern.“[DCH15]
[AST08][Kre15a]

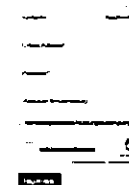


Abbildung 31: Registrierung bei DSN

Ein CAPTCHA soll bei der Registrierung verhindern, dass eine Software oder ein Bot einen Account erzeugt.

Generell dient es der Sicherheit und hat die Aufgabe, jede Eingabe auf ihre Herkunft zu prüfen. Um ein Captcha anzuzeigen, ist die Einbindung einer JavaScript Library notwendig. [Shal5].

```
<script src="https://www.google.com/recaptcha/api.js?
onload=vcRecaptchaApiLoaded&render=explicit" async defer></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/angular-recaptcha/2.2.5/
angular-recaptcha.min.js"></script>
```

Listing 28: Einbindung der JS-Library Recaptcha

Bei jeder Registrierung generiert sich das Captcha neu, dadurch kann es nur einmal eingelöst werden.

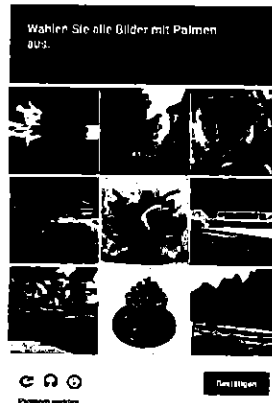


Abbildung 32: Lösen des Captchas

5.2.1.2 Datenbank

Nach der Authentisierung wird der zukünftige Benutzer, nach dem im Kapitel 5.3.2.1 beschriebenen Datenmodell, in die Datenbank persistiert. Der Registrierungsprozess ist erst nach Einlösen des Validierungstokens abgeschlossen. Bis dahin ist es dem Benutzer nicht möglich, sich auf DSN anzumelden.

5.2.1.3 Email

Um den Registrierungsprozess zu beenden, wird dem User ein Token per Mail von unserem Mailserver zugesandt. Dieser Token dient der Identifizierung und Authentifizierung und könnte folgendermaßen aussehen:

```
http://digitalschoolnotes.com/validate/
dad9574635aad7d6549536db3817839c042f7704b3bd74acc4271075d0601470
```

Listing 29: Validierungstoken für die Aktivierung des DSN-Accounts

Bei der Erstellung eines solchen Tokens wird die Email-Adresse des Benutzers und das aktuelle Datum kombiniert. Die beiden Werte werden miteinander verknüpft und in einen Hash umgewandelt. Dieser Hash dient als Aktivierungstoken.

Dem Token wird mittels Datum eine Lebensdauer zugeteilt. Falls die Einlösung des Hash nicht innerhalb der nächsten 24 Stunden erfolgt, verfällt dieser und muss vom User neu angefordert werden.

5.2.1.4 Anmeldung

Hat der Benutzer den Registrierungsprozess erfolgreich abgeschlossen, steht es ihm jetzt frei, sich anzumelden. Der Login erfolgt anhand der Angabe der Email-Adresse plus Passwort oder er nutzt die Möglichkeit einer OAuth-Anmeldung.

OAuth bietet dem Nutzer die Möglichkeit, Daten über einen Webservice auszutauschen. „OAuth sichert die Programmschnittstelle von Web-Anwendungen und verwendet für die Übertragung der Nutzeridentifikation dessen Passwort und einen Token“ [DI15]. Beim Zugriff auf sensible Daten muss der Benutzer keine zusätzlichen Information und auch keine Identität preisgeben. Der Dienstanbieter holt sich die Benutzerdaten von Facebook oder Google+ und erstellt für den User einen Account. Sozusagen besteht für unsere Endanwender die Möglichkeit, den vorher beschriebenen Registrierungsprozess auszulassen und sich direkt über OAuth anzumelden.

Abbildung 33: Klassische Anmeldung oder mittels OAuth

Beim Anmelden wird kontrolliert, ob die angegebene Email-Adresse in der Datenbank existiert. Ist sie keinem Benutzer zugeordnet, existiert dieser nicht und die Authentifikation schlägt fehl. Andernfalls wird das eingegebene Passwort überprüft. Ist dieses korrekt, wird der Anwender in den Userbereich weitergeleitet. *wirklich?!*

Surft ein Benutzer auf unserer Seite, arbeitet im Hintergrund ein Session Timeout, welches die Aktivität des eingeloggten Benutzers überprüft.

Im Falle einer Inaktivität von einer Stunde, wird der Anwender automatisch abgemeldet und zur Anmeldeseite zurückgeleitet. Es dient als Vorbeugemaßnahme und kontrolliert unauthorisierte Aktivitäten.

```
try:
    user = User.objects.get(email='example@example.com')
except:
    user = None
if user is not None and user.check_password('myPassword'):
    user.backend = 'mongoengine.django.auth.MongoEngineBackend'
    login(request, user)
    request.session.set_expiry(60 * 60 * 1) # 1 hour timeout
```

Listing 30: Session Timeout

5.2.2 Userbereich

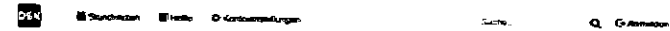


Abbildung 34: Navigationsleiste als User

In der Navigationsleiste befinden sich:

- **Stundenplan**
Im Stundenplan kann jeder seine Schulstunden und Schulfächer manuell eintragen.
- **Hefte**
Hier werden die Schulhefte aufgelistet, welche während der Schulstunden zum Einsatz kommen.
- **Kontoeinstellungen**
Die Kontoeinstellungen sind in drei Bereiche gegliedert.
 1. **User-Daten bearbeiten**
Ändern der Benutzerinformationen
 2. **Passwort ändern**
Überschreiben des alten Passworts
 3. **Account löschen**
Alle angelegten Hefte, sowie der DSN-Account werden gelöscht
- **Suche**
Jeder Benutzer hat die Möglichkeit, nach Freunden oder anderen registrierten Anwendern, mittels Vorname, Nachname oder Email-Adresse, zu suchen. Auf dessen Profil ist der vollständige Name, die Email-Adresse und die Berechtigungsstufe, ob Standardbenutzer, Pro-Benutzer oder Administrator, zu sehen. Ein weiterer wichtiger Punkt sind die öffentlichen Hefte. Jedem User steht es frei, von diesen, einzelne Heftseiten in seine eigenen Hefte zu importieren.
- **Abmelden**
Abmelden des Accounts

5.2.3 Berechtigungen

Im System befinden sich drei verschiedene Berechtigungsstufen, nämlich: Standard-Benutzer, Pro-Benutzer und Administrator. Jeder registrierte Anwender ist zu Beginn ein Standard-Benutzer.

- **Standard-Benutzer**

Der Standard-Benutzer hat Gelegenheit, als Erstanwender die vielen Vorteile, wie OCR oder PWS, zu nutzen. Unser System bietet dafür eine Testdauer von 90 Tagen an und steht in Form von digitalen Heften in begrenzter Stückzahl zur Verfügung.

- **Pro-Benutzer**

In Zukunft wollen wir Standard-Benutzern die Möglichkeit geben, durch eine geringen monatlichen Beitrag, seinen Account auf einen Pro-Account upzugraden. Dadurch werden dem Schüler erweiterte Funktion angeboten, wie z.B. eine unbegrenzte Heftanzahl, keine Speicherbeschränkung, sowie keine Werbung. Das derzeitige Limit liegt bei 1GB.

Das Konzept gibt es bereits, aber die Umsetzung ist noch in der Entwicklungsphase.

- **Administrator**

Die letzte Berechtigungsstufe sind Administratoren. Sie sind ebenfalls Pro-User, haben aber im Gegensatz zu den Pro-Usern einen eigenen Admin-Bereich.

5.2.4 Adminbereich

Sämtliche Daten über Benutzer werden im `/admin` Bereich aufgelistet.



Abbildung 35: Navigationsleiste als Admin

5.2.4.1 User Management

Die User Management Page listet alle Benutzer von DSN tabellarisch auf. Der Admin hat Einsicht auf die Email-Adresse, Vorname, Nachname und auf die Berechtigungsstufe.

Dem Administrator steht es frei, Benutzer zu löschen, deren Berechtigungsstufe zu ändern oder mittels Mail auf Probleme etc. hinzuweisen. Neben der Auflistung aller User kann im Bedarfsfall nach Einzelnen gesucht werden. Die Sucheingabe wird mit vorhandenen Daten verglichen. Zusätzlich besteht die Möglichkeit, die Tabelle in alphabetischer Reihenfolge zu sortieren.

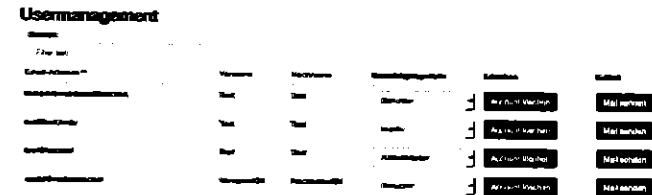


Abbildung 36: Usermanagement-Page

Unsere Usermanagement-Page wurde so designed, dass durch kurze Ladezeit dem Administrator diese Seite rasch zur Verfügung steht. Die Performancesteigerung ergibt sich daraus, dass nur eine bestimmte Anzahl an DSN-Usern in Form einer Tabelle vom Server geladen wird. Ist das Ende einer Tabellenseite erreicht, kann auf die nächste Seite umgeblättert werden. Alle User finden sich so in der Liste wieder.

5.2.4.2 Benachrichtigung

Pro-User, die den Zahlungen nicht nachkommen, geben dem Administrator das Recht, diesen zu löschen. DSN gibt dem User die Möglichkeit, seine Daten bzw. Hefte zu sichern, bevor er gelöscht wird. Der Benutzer empfängt rechtzeitig eine Aufforderungs-Mail, in welcher darauf hingewiesen wird, dass sein Account und alle dazugehörigen Daten nach 7 Tagen gelöscht werden. Am Server von DSN läuft ein Cron-Daemon, welcher täglich prüft, wann der zu löschende User entfernt werden soll. „Der Cron-Daemon ist ein Dienst, der automatisch Skripte und Programme zu vorgegebenen Zeiten starten kann.“[ubu15]

Im Falle, dass jemand länger als 3 Monate inaktiv ist, wird ihm ebenfalls eine Informationsmail geschickt. Wenn sich der User binnen 7 Tagen nicht einloggt, wird der Account mit allen Daten gelöscht. [Fou15a][ubu15]

```
# m h dom mon dow command
# * * */1 * * python3 /home/stable/dsn/manage.py inform
# * * */1 * * python3 /home/stable/dsn/manage.py delete
```

Listing 31: Cronjob für die Überprüfung der Inaktivität und Löschung

5.3 Datenmanagement

Ein gutes Datenmanagement ist eine Grundvoraussetzung für eine gut funktionierende Applikation. Alle Daten müssen persistiert werden, um nicht verloren zu gehen. Zudem soll durch ein gut organisiertes Datenmanagement eine einfache und effiziente Verwaltung der Daten gesichert werden.

5.3.1 Persistierung von Daten

Die Persistierung von Daten ist besonders wichtig. Das Abspeichern aller Daten dient dem Zweck, die Daten auch zu einem späteren Zeitpunkt oder beispielsweise nach einem Reboot des Servers noch abrufen zu können. Um das zu erreichen, wird eine Datenbank benötigt, die sich um die Persistierung kümmert. Dabei wird zwischen zwei Arten von Datenbankkonzepten unterschieden: Relationale Datenbanken und NoSQL Datenbanken. Im folgenden werden beide Konzepte näher erläutert und ein Vergleich zwischen den beiden Arten erstellt.

das kam schon vor!

5.3.2 Datenmodell

Um einen einheitlichen Zugriff auf Daten zu ermöglichen, muss ein Datenmodell entwickelt werden. Ein Datenmodell bezeichnet eine Struktur beziehungsweise den Aufbau der einzelnen Datensätze. Dabei ist zu beachten, dass in der Applikation Daten unterschiedlichster Art gespeichert werden. Das bedeutet, dass diese unterschiedlichen Typen logischerweise auch unterschiedliche Datenmodelle zugrunde haben müssen. Zur Organisation dieser unterschiedlichen Arten von Daten gibt es in MongoDB sogenannte Collections. Diese Collections beinhalten alle Datensätze eines Typs, beispielsweise alle Einträge von Benutzern. In unserer Applikation gibt es folgende Collections:

- **user**
Zur Abspeicherung von Benutzeraccounts
- **notebook**
Zur Abspeicherung von Schulheften; beinhaltet auch alle Daten innerhalb des Heftes (Texte, Bilder,...)
- **time_table**
Zur Abspeicherung eines Stundenplans; beinhaltet die einzelnen Stundenzeiten sowie eine Fachbezeichnung, einen Lehrer und einen Raum pro Tag und Stunde



5.3.2.1 User

Die Collection *user* beinhaltet alle relevanten Daten zur Abspeicherung von Benutzeraccounts. Folgendes Datenmodell liegt dem zugrunde:

- **_id**
Eine Object-ID zur eindeutigen Identifizierung eines Eintrags
- **email**
Die E-Mail Adresse des jeweiligen Benutzers
- **first_name**
Der Vorname des jeweiligen Benutzers
- **last_name**
Der Nachname des jeweiligen Benutzers
- **password**
Ein Hash eines selbst definierten Passwortes des jeweiligen Benutzers
- **is_prouser**
Beschreibt, ob der jeweilige Benutzer einen Pro-Account besitzt
- **is_active**
Beschreibt, ob der jeweilige Benutzer bereits seine E-Mail Adresse bestätigt hat
- **is_superuser**
Beschreibt, ob der jeweilige Benutzer ein Administrator der Applikation ist
- **last_login**
Das Datum, an dem der jeweilige Benutzer sich das letzte Mal erfolgreich an der Applikation angemeldet hat
- **date_joined**
Das Datum, an dem der jeweilige Benutzer sich an der Applikation registriert hat
- **validatetoken**
Ein Hash, der als Token zur Bestätigung der E-Mail Adresse des jeweiligen Benutzers dient, sofern dieser die Bestätigung noch nicht durchgeführt hat
- **passwordreset**
Ein Hash sowie ein Datum, welche zur Zurücksetzung des Passwortes des jeweiligen Benutzers dienen, sofern dieser angegeben hat, sein Passwort vergessen zu haben

Benutzer werden innerhalb der Applikation mithilfe ihrer E-Mail Adresse identifiziert. Das bedeutet, dass die Eigenschaft *email* bei jedem Eintrag in der Datenbank einzigartig sein muss, ebenso wie die ID des Eintrags.



5.3.2.2 Notebook

Die Collection *notebook* beinhaltet die Daten, die ein einzelnes Heft betreffen. Dazu wird folgende Struktur definiert:

- **id**
Eine Object-ID zur eindeutigen Identifizierung eines Eintrags
- **name**
Der Anzeigename des Heftes, der vom Benutzer festgelegt wurde
- **is_public**
Beschreibt, ob das Heft öffentlich (von allen Benutzern der Applikation) einsehbar ist
- **create_date**
Das Erstellungsdatum des Heftes
- **last_change**
Das Datum, an dem das Heft das letzte Mal bearbeitet wurde
- **email**
Die E-Mail Adresse des Besitzers des Heftes
- **numpages**
Die Anzahl an Seiten, die das Heft besitzt
- **current_page**
Die Seite, die aufgeschlagen wird, sobald der Benutzer das Heft das nächste Mal öffnet
- **content**
Eine Liste, die alle Inhalte des Heftes (Texte, Bilder,...) beinhaltet
- **collaborator** Eine Liste an E-Mail Adressen von Benutzern der Applikation, die neben dem Besitzer des Heftes ebenfalls die Inhalte des Heftes bearbeiten dürfen

Der Name eines Heftes ist pro Benutzer einzigartig zu vergeben, um das Heft identifizieren zu können. Abgesehen von der Eigenschaft *id* kann ein Heft also auch mithilfe der beiden Eigenschaften *name* und *email* eindeutig identifiziert werden. Die Eigenschaft *content* besteht aus einer Liste. Diese Liste beinhaltet individuell viele JSON-Objekte, die jeweils ein Objekt innerhalb des Heftes darstellen (Text, Bild,...). Diese JSON-Objekte bestehen wiederum aus einer ID zur Identifizierung, der Art des Elementes (Textelement, Bildelement,...), der genauen Position innerhalb des Heftes (Seitenzahl und x-Koordinate, sowie y-Koordinate auf dieser Seite) und dem eigentlichen Inhalt, beispielsweise dem Text den der Benutzer eingegeben hat, sollte es sich um ein Textelement handeln.

5.3.2.3 Timetable

In der Collection *time_table* werden alle Daten der Stundenpläne von Benutzern gespeichert. Das beinhaltet sowohl die einzelnen Fächer, Lehrer und Räume pro Stunde, als auch die per Benutzer definierten Zeiten für jede Stunde. Zur Persistierung wird folgendes Datenmodell verwendet:

- **id**
Eine Object-ID zur eindeutigen Identifizierung eines Eintrags
- **email**
Die E-Mail Adresse des Benutzers, dem der Stundenplan zugeordnet ist
- **times**
Eine Liste, die die Anfangs- und Endzeiten jeder Stunde im Stundenplan enthält
- **fields**
Eine Liste, die alle einzelnen Stunden im Stundenplan mit Fach, Lehrer und Raum enthält

Der Stundenplan wird einem Benutzer mithilfe der Eigenschaft *email* zugeordnet und damit auch eindeutig identifiziert, da jeder Benutzer nur einen Stundenplan besitzen kann.

Die Eigenschaften *times* und *fields* sind jeweils Listen, die mehrere JSON-Objekte enthalten. Ein JSON-Objekt in der Liste *times* enthält die jeweilige Stunde (1-xx) und die Anfangs- und Endzeit für diese Stunde. Ein JSON-Objekt in der Liste *fields* enthält eine ID zum Identifizieren der jeweiligen Stunde (Zusammengesetzt aus Reihenzahl und Spaltenzahl im Stundenplan), eine Bezeichnung des Faches, das in dieser Stunde unterrichtet wird, dem Namen des Lehrers, der das jeweilige Fach unterrichtet und dem Raum, in dem der Unterricht stattfindet, sowie ein Heftname eines Heftes, das dem jeweiligen Benutzer gehört. Mithilfe des Heftnamens kann ein Heft mit einer Stunde im Stundenplan verknüpft und zugeordnet werden.

5.3.3 Datenzugriff

Der Zugriff auf Daten in der Datenbank kann über zwei Arten erfolgen. Entweder es wird direkt über die Konsole von MongoDB zugegriffen, oder der Datenzugriff erfolgt über die Applikation.

Der Zugriff auf Daten bezeichnet dabei unterschiedliche Operationen. Es können neue Daten erstellt, vorhandene Daten ausgelesen und bestehende Daten bearbeitet oder gelöscht werden. Wie all diese Operationen jeweils auf beide der genannten Arten des Zugriffs durchgeführt werden können, wird im folgenden näher erläutert.

5.3.3.1 Direkter Zugriff auf die Datenbank

Ein direkter Zugriff auf Daten in der Datenbank kann über die Konsole von MongoDB ausgeführt werden. Die Konsole kann nach Installieren von MongoDB mithilfe dem Befehl `mongo` geöffnet werden. Anschließend können weitere Operationen durchgeführt werden.

Um eine Datenbank für den Datenzugriff auszuwählen, wird der Befehl `use dbname` benötigt. Dieser erstellt auch gleichzeitig eine neue Datenbank, sollte noch keine mit dem angegebenen Namen existieren. Nach Auswählen der gewünschten Datenbank können nun Collections angelegt werden, um die Daten später besser zu organisieren. Dies geschieht mittels dem Befehl `db.createCollection(collname)`. MongoDB kann allerdings auch automatisch eine neue Collection erstellen, sobald versucht wird einen neuen Datensatz in eine bisher nicht existente Collection einzufügen.

Nachdem eine Datenbank und Collections angelegt wurden, können weitere Operationen durchgeführt werden.

Mithilfe des Befehls `db.collname.insert(document)` kann ein neuer Datensatz in eine gewünschte Collection eingefügt werden. `document` bezeichnet dabei einen JSON-String. Dieser kann beispielsweise folgendermaßen aussehen:

```
{ "name": "Test", "email": "test@email.com" }
```

Der String kann dabei aus beliebig vielen Key-Value Paaren bestehen.

Der Befehl `db.collname.find()` kann dazu verwendet werden, bestehende Datensätze innerhalb einer Collection zu suchen und anzuzeigen. Dabei kann optional ein Parameter angegeben werden, um Suchkriterien festzulegen. Um beispielsweise Benutzer mit einem bestimmten Namen zu suchen, kann folgendermaßen vorgegangen werden:

```
db.user.find({ "name": "Test" })
```

Hierbei würden nun alle Benutzer mit dem Namen "Test" angezeigt werden. Sollte nur ein Suchergebnis erwartet werden, kann zudem auch der Befehl `.findOne()` verwendet werden.

Mithilfe von `db.collname.update(criteria, updated)` kann ein bestehender Datensatz verändert werden. `criteria` entspricht dabei einem JSON-String, der als Suchkriterium dient. Nur Datensätze, die diesem Kriterium entsprechen, werden verändert. `updated` bezeichnet den neuen JSON-String, durch den der alte Datensatz ersetzt werden soll. Soll nicht der ganze Datensatz ersetzt werden, sondern lediglich ein oder mehrere Eigenschaften des bestehenden Datensatzes, so kann `$set` folgendermaßen verwendet werden:

```
db.user.update({ "name": "Test" }, { $set: { "name": "Test2" }, { multi: true } })
```

Hier werden alle Datensätze in der Collection `user`, die den Namen "Test" haben so bearbeitet, dass nun "Test2" als Name festgelegt wird. Der zusätzliche Parameter `{multi:true}` besagt dabei, dass nicht nur der erste Datensatz, der auf das Suchkriterium zutrifft, bearbeitet werden soll, sondern alle, die dem Muster entsprechen.

Um einen bereits bestehenden Datensatz wieder aus der Collection zu löschen, kann der Befehl `db.collname.remove()` verwendet werden. Sollte hierbei kein Parameter angegeben werden, werden alle Datensätze aus der Collection gelöscht. Mithilfe eines JSON-Strings als Parameter kann das Löschen auf Datensätze beschränkt werden, die einem bestimmten Kriterium entsprechen.

5.3.3.2 Zugriff aus der Applikation

Um innerhalb der Applikation auf Daten aus der Datenbank zugreifen zu können, muss Django entsprechend konfiguriert werden. Django unterstützt standardmäßig lediglich relationale DBMS. Da im Projekt jedoch als Datenbank die NoSQL-Datenbank "MongoDB" verwendet wird, muss dies extra konfiguriert werden.

Dazu muss `MongoEngine`, ein Verbindungsstück zwischen Django und MongoDB, installiert werden. Dabei sollte die Version 0.9.0 verwendet werden, da die neueste Version 0.10.0 aktuell einen Bug hat, durch welchen man sich nicht mehr mit einer Datenbank verbinden kann.

```
pip3 install mongoengine==0.9.0
```

Um Django mitzuteilen, dass MongoDB im Hintergrund verwendet werden soll, müssen im File `prototype/settings.py` folgende Zeilen auskommentiert beziehungsweise hinzugefügt werden:

```
#DATABASES = {
#    'default': {
#        'ENGINE': 'django.db.backends.sqlite3',
#        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
#    }
#}
import mongoengine

DATABASES = {
    'default': {
        'ENGINE': '',
    },
}
SESSION_ENGINE = 'mongoengine.django.sessions'

_MONGODB_HOST = 'localhost' #Hostname
_MONGODB_NAME = 'dsn' #DB-Name
_MONGODB_DATABASE_HOST = 'mongodb://ls' % (_MONGODB_HOST)
mongoengine.connect(_MONGODB_NAME, host=_MONGODB_DATABASE_HOST)
```

Listing 32: MongoDB Konfiguration in Django

Um von Django aus auf Daten in MongoDB zugreifen beziehungsweise neue Daten hineinspeichern zu können, müssen vorher sogenannte Models erstellt werden. Diese sind im File *models.py* definiert und bilden die Struktur der Daten in der Datenbank ab. Dies kann beispielsweise folgendermaßen aussehen:

```
from mongoengine import *

class Notebook(Document):
    name = StringField(max_length=30)
    is_public = BooleanField()
    create_date = DateTimeField()
    last_change = DateTimeField()
    email = EmailField()
```

Listing 33: Beispiel für ein Datenbankmodell in Django

Der Klassenname (hier: Notebook) entspricht in der Datenbank später dem Collection-Namen. Wichtig bei der Klassendefinition ist, dass von *Document* geerbt wird. *Document* definiert Funktionen zum Auslesen der Daten der entsprechenden Collection aus der Datenbank, zum Erstellen eines neuen Objektes und zum Abspeichern des Objektes in der Datenbank. Die einzelnen Attribute der Klasse werden in der Datenbank als JSON abgebildet, wobei der Key der Name des Attributes ist.

Nachdem MongoEngine innerhalb von Django konfiguriert wurde und alle benötigten Datenbankmodelle aller verwendeten Collections erstellt wurden, können Operationen zum Hinzufügen, Lesen, Bearbeiten und Löschen von Datensätzen in der Datenbank in den entsprechenden Collections ausgeführt werden.

Mittels *collectionName.objects()* können bestehende Datensätze der entsprechenden Collection ausgelesen werden. Dabei liefert MongoEngine alle Objekte aus der angegebenen Collection. Wenn nicht alle Objekte einer Collection gefragt sind, sondern nur eine gewisse Anzahl, können diese mit folgendem Befehl abgefragt werden:
Tabellenname.objects[x:y]

Um bestimmte Objekte innerhalb einer Collection nach selbst definierten Suchkriterien zu filtern kann folgende Syntax verwendet werden:

```
user = User.objects()
user(
    Q(email__icontains=suchtext) |
    Q(first_name__icontains=suchtext) |
    Q(last_name__icontains=suchtext)
)
```

Listing 34: Syntax für eine Suchanfrage an die Datenbank in Django

Hierbei wird nach allen Benutzern gesucht, die *suchtext* entweder in der E-Mail Adresse oder im Vor- beziehungsweise Nachnamen angegeben haben. *icontains* gibt dabei an, dass die Groß- und Kleinschreibung bei der Suche nicht relevant ist. Sollte jedoch gewünscht sein, dass auch die Groß- und Kleinschreibung bei der Suche beachtet wird, muss statt *icontains* das Stichwort *contains* verwendet werden.

Objekte können auch nach einer bestimmten Spalte sortiert werden. Um die Objekte aufsteigend zu sortieren, kann folgender Befehl verwendet werden:

users.order_by(spaltenname)

Zum absteigenden Sortieren wird folgende Syntax benötigt:

users.order_by('-'+spaltenname)

Falls ein vorhandenes Objekt aus der Collection gelöscht werden soll, muss dieses zuvor mithilfe einer Suchanfrage abgefragt werden und dann mit der Funktion *.delete()* entfernt werden.

Nachdem ein Ergebnis auf eine Suchanfrage zurückgeliefert wurde, können einzelne Attribute des erhaltenen Objektes verändert werden. Um diese Änderungen anschließend zu persistieren, muss die Funktion *.save()* auf das entsprechende Objekt ausgeführt werden.

5.4 Heftelemente

Um Inhalt in ein Heft einzufügen gibt, es die Ebene der Elemente. Diese verschiedenen Elemente bauen alle auf der selben Grundstruktur auf und sind nach Art des Elements weiter spezifiziert.



Abbildung 37: Heftelemente

Mittels Mausklick auf ein bestimmtes Icon ⁱⁿ der ^{Werkzeugleiste} wird das jeweilige Element in das Heft eingefügt.

Um das Element bearbeiten zu können, wird beim Überfahren des Elements ein Bleistift eingeblendet, welcher dem Benutzer erlaubt, den bestehenden Inhalt des Elements zu editieren. Neben dem Bleistift erscheint ebenfalls ein Miskübel, mit dem das Element entfernt werden kann. Außerdem ist es möglich, ein Element beliebig im Heft zu verschieben und dadurch beliebig im Heft zu platzieren.

Diese Grundfunktionen sind in jedem Element enthalten. Jedes Element ist für den jeweiligen Anwendungsbereich genauer spezifiziert.

5.4.1 Textelement

Um Informationen im Sinne von Text in ein Heft einzufügen gibt es ein Textelement. Mit diesem Textelement lässt sich Text eingeben und bearbeiten. Es können Funktionen wie Fett, Kursiv, Durchgestrichen, Liste, Aufzählung, etc. auf den Text angewendet werden. Außerdem können Tabellen in dem Textelement erzeugt werden, welche beliebig groß definiert werden können. Sollte der Benutzer Sonderzeichen einbinden wollen, geht dies mit dem Sonderzeichen Tool im Textelement. Außerdem kann ein Hyperlink eingebunden werden, der den Link automatisch in einem neuen Tab öffnet.

Dieses Element wurde mittels des Frameworks CKEditor implementiert. Mit diesem Framework war es möglich, einen Texteditor in das bereits bestehende Heft einzufügen. Der CKEditor konnte frei konfiguriert werden und somit optimal auf die Anforderungen von DigitalSchoolNotes angepasst werden.



Abbildung 38: Textelement

Mittels Klick auf das A Icon ist es möglich, ein Textelement im Heft zu platzieren.

Im Bearbeitungsmodus ist es möglich, den Text mit bestimmten Effekten hervorzuheben oder Hyperlinks einzufügen.



Der Mount Everest ist mit 8848 m der höchste Berg im Himalaya und der höchste Berg der Erde. Er gehört zu den 14 Achttausendern und zu den Seven Summits. Der Mount Everest ist seit 1856 nach dem britischen Landvermesser George Everest benannt. Auf Nepali heißt der Berg Sagarmatha, auf Tibetisch Qomolangma (deutsche Aussprache „Tschomolangma“, englische Umschrift Chomolungma).

body

Abbildung 39: Textelement im Bearbeitungsmodus

Nach Verlassen des Bearbeitungsmodus wird die angewendeten Effekte übernommen und entsprechend angezeigt.

Der Mount Everest ist mit 8848 m der höchste Berg im Himalaya und der höchste Berg der Erde. Er gehört zu den 14 Achttausendern und zu den Seven Summits. Der Mount Everest ist seit 1856 nach dem britischen Landvermesser George Everest benannt. Auf Nepali heißt der Berg Sagarmatha, auf Tibetisch Qomolangma (deutsche Aussprache „Tschomolangma“, englische Umschrift Chomolungma).

Abbildung 40: Textelement im Ansichtsmodus

Ein besonders kritischer Punkt beim Textelement war die Speicherung beziehungsweise die Übernahme des geschriebenen Inhaltes in den Ansichtsmodus. CKEditor verwendet hierbei eine eigene Speicherstruktur, welche vor der Verwendung auf die neuen Anforderungen angepasst werden musste. Nach erfolgreicher Anpassung ist es nun möglich den geschriebenen Inhalt inklusive hervorgehobenem Text in den Ansichtsmodus zu übernehmen.

Die Speicherung wird von on-Events übernommen, die nach einer bestimmten Aktion den Updatevorgang auslösen.

```
function updateModel() {
    scope.$apply(function () {
        ngModel.$setViewValue(ck.getData());
    });
}

ck.on('dataReady', updateModel);
ck.on('key', updateModel);
ck.on('paste', updateModel);
ck.on('selectionChange', updateModel);
```

Listing 35: Speicherung des Textelements

Das Textelement wurde wie folgt in das Heft integriert. Dieser Abschnitt wird nach dem Klick auf das Icon ausgeführt.

```
<span ng-if="element.art == 'textarea'" style="word-wrap: break-word;">
  <!-- Edit -->
  <textarea ng-if="editMode == element.id"
    ng-model="models[element.art][element.id][0]" ckeditor>

  </textarea>
  <span ng-show="models[element.art][element.id][2] == 'red'"
    class="glyphicon glyphicon-exclamation-sign"
    ng-style="{ 'color': models[element.art][element.id][2],
      'cursor': 'pointer' }">
  </span>
  <!-- Read-Only -->
  <div ng-if="editMode != element.id"
    style="border: 1px solid black; min-height: 20px;
      min-width: 20px; text-align: left; padding: 5px;">
    <div ng-bind-html="models[element.art][element.id][0]"
      style="word-wrap: break-word; ">
    </div>
  </div>
</span>
```

Listing 36: Einbindung des Textelements

5.4.2 Codeelement

Zur Realisierung des Codeelements wurde das Framework codemirror [Hav15] benutzt. Das Codeelement soll es dem Benutzer ermöglichen, auf einfache und intuitive Weise Codesnippets in seiner Mitschrift zu erstellen und zu bearbeiten. Besondere Features sind hier das Anzeigen der Zeilennummern, sowie das Syntax-Highlighting (abhängig von der jeweiligen Programmiersprache). In der vorliegenden Implementation werden folgende Sprachen unterstützt:

- XML, HTML
- JavaScript
- C-ähnliche Sprachen (C, C++, C#, Java, etc.)
- Python
- SQL
- Shell
- Ruby
- PHP

Wenn der Benutzer nun konkret ein Codeelement hinzufügen will, betätigt er zunächst diesen Code-Button.



Abbildung 41: Code-Button

Nach Betätigen des Code-Buttons erscheint eine Auswahl von (Programmier-)Sprachen. Hier legt der Benutzer fest, wie der Inhalt im Codeelement hervorgehoben wird. (Dies ist von Sprache zu Sprache verschieden und kann auch im Nachhinein geändert werden)

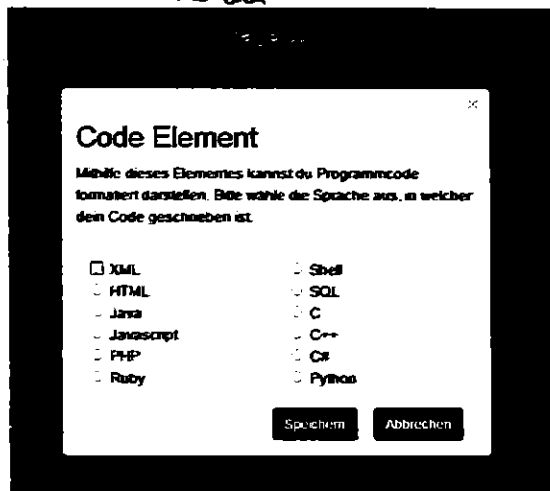


Abbildung 42: Programmiersprache auswählen

Anschließend erscheint das leere Codeelement im Heft.



Abbildung 43: Leeres Code-Element

Unter dem Codeelement gibt es noch drei Buttons, die nur sichtbar sind, wenn man mit der Maus darüber fährt. Mit Druck auf den ersten Button (also den Mülleimer) kann das Element (nachdem man nochmals bestätigt hat) gelöscht werden. Um in den Bearbeitungsmodus des Codeelements zu wechseln betätigt der User den zweiten Button (also den Stift). Um im Nachhinein die Auswahl der Programmiersprache zu ändern (siehe Dialog oben) muss auf den dritten Button, also das Zahnrad, geklickt werden.

also entweder
Mülleimer od. Mülleimer
nicht beides

```
1 public static void bearbeitungsmodus(String[] args){
2     int x;
3     return x;
4 }
```

Bearbeitungsmodus aktiviert! Beenden

Abbildung 44: Code-Bearbeitung

Bei Klick auf den Code-Button wird die Methode `codeElementCreate()` ausgeführt. Dort wird ein `ngDialog` mit dem `codeElementSettings`-template geöffnet, in welchem der User zum ersten Mal die Programmiersprache auswählen kann. Nachdem dies ausgewählt wurde, wird die Methode `addCodeElement()` ausgeführt.

```
$scope.addCodeElement = function() {
    if(!$scope.wf) {
        data = "{ \"data\": \"\", \"language\": \"\" + $scope.codeLanguage + \"\" }";
        console.log(data);
        $scope.addcodeelement('code', data);
    } else {
        $scope.wf = false;
        single_object = $filter('filter')($scope.content, function (d) {
            return d.id === $scope.cid;
        })[0];
        data = "{ \"data\": \"\" + single_object['data'] + \"\", \"language\": \"\" + $scope.codeLanguage + \"\" }";
        $scope.editcodeelement($scope.cid, 'code', data);
    }
}
```

Listing 37: Code-Element Persistierung

Bei Klick auf das Zahnrad kann dann noch die eingestellte Programmiersprache geändert werden. Dabei wird die Methode `setCodeElementLanguage()` ausgeführt.

5.4.3 Bildelement

Durch das Bildelement können Grafiken aus dem Unterricht, oder Internet in ein Heft integriert werden. Durch die Einbindung von Grafiken kann einfacher gelernt werden und der Schüler hat mehr Möglichkeiten zur Gestaltung seines Schulheftes.

Um ein Bild einfügen zu können, muss der Bilder Button in der Werkzeugleiste im Heft geklickt werden (siehe Abbildung 37). Danach öffnet sich folgende Ansicht im Heft:

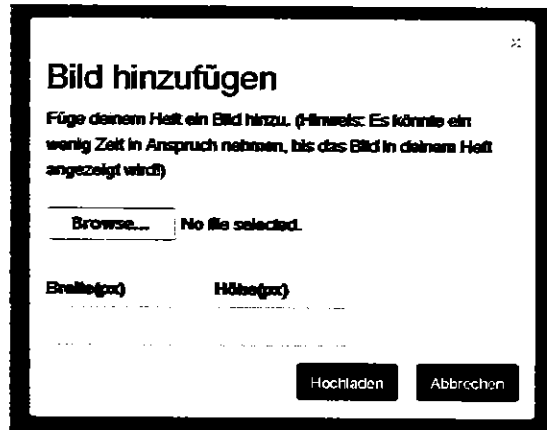


Abbildung 45: Bild zum Heft hinzufügen

Durch einen Klick auf Browse öffnet sich ein Dateieexplorer, über welchen ein Bild, welches hochgeladen werden soll, ausgewählt werden kann. Hierbei wurden nur Bilddateien erlaubt, andere Dateiformate können nicht ausgewählt werden. Des Weiteren kam über zwei Eingabefelder die Breite und Höhe des Bildes angegeben werden. Das Bild wird hierbei jedoch nicht zugeschnitten, sondern nur verkleinert dargestellt. Dadurch haben die hochgeladenen Dateien höhere Dateigrößen, jedoch kann im Nachhinein das Bild durch eine Änderung der Größe größer dargestellt werden.

Nachdem alle Formularfelder korrekt ausgefüllt wurden, kann der Button "Hochladen" gedrückt werden. Nach einem Klick auf den Button wird das Bild auf den Projektserver hochgeladen. Dort wird dann überprüft, ob es sich um ein Bild handelt. Falls die Datei ein Bild ist, wird sie mittels Bildverarbeitung optimiert, um Speicherplatz zu sparen. Nach diesem Schritt wird das Bild in die Amazon S3 Cloud hochgeladen. Das Bild wird nun von Amazon redundant gespeichert. Um es später ins Heft einbinden zu können, gibt Amazon einen Link zum Bild zurück. Danach kann das Bild vom Projektserver gelöscht werden und der Link zum Bild wird in der Datenbank abgespeichert. Nun kann das Bild mittels HTML in das Heft eingebunden werden.

5.5 Parallel Working System

Mit dem Parallel Working System ist es dem Benutzer möglich, Hefte mit anderen Benutzern zu teilen und diese anschließend gleichzeitig zu bearbeiten. Es kann immer ein Element pro User bearbeitet werden. Während dieses Element bearbeitet wird, können andere Benutzer dieses Element weder löschen, bearbeiten, noch verschieben.

5.5.1 Bestehende Systeme

Um einen besseren Überblick über bereits bestehende Collaboration Systeme zu bekommen, wurden anfangs solche analysiert. Im Zuge der Evaluierung wurden deswegen die Systeme von Google Docs und Microsoft Portal unter die Lupe genommen.

5.5.1.1 Google Docs

Mit Google Docs wurde einer der bekanntesten Vertreter der Realtime Web-Kommunikation analysiert. Die Grundidee von Google Docs ist allerdings eine andere, hierbei kann man in Realtime im gesamten Textdokument arbeiten. Bei DSN ist das gleichzeitige Arbeiten auf Elemente beschränkt.

Google Docs verwendet eine eigene Google Realtime API, welche die Collaboration Vorgänge abwickelt. Diese JavaScript Library enthält Events und Methoden, die für die Erstellung einer kollaborativen Software verwendet werden können.

Die Google Realtime API arbeitet mit einem Aktualisierungsprinzip, das nur dann aktualisiert, wenn eine Änderung am Shared Medium vorgenommen wurde. Im Hintergrund wird dann eine Benachrichtigung an die entsprechenden User gesendet, um das Dokument aktuell zu halten. Diese Realtime API arbeitet mit einem Daten Modell, das sich „eventually consistent“ nennt. Dadurch ist es möglich, dass jeder dieselben Daten sieht. [Goo16b]

Die Google Realtime API konnte allerdings nicht beim Parallel Working System verwendet werden, weil die Google API ein eigenes Datenmodell benötigt und nur schwer auf einem bereits bestehendes Datenmodell aufbauen kann. Außerdem kann diese API nicht mit Javascript-Frameworks zusammengefügt werden. Da bei DigitalSchoolNotes eine Reihe von Frameworks verwendet wurden, hätte dies einen großen Aufwand erfordert. Darum wurde diese Google API nicht für die Implementierung des Parallel Working Systems verwendet.

5.5.1.2 Microsoft Portal

Mit Microsoft Portal ist es möglich, im Web-Browser Office-Apps zu verwenden, unter anderen Word Online. Diese Online Textverarbeitung ist eine abgeschwächte Version des klassischen Desktop Words.

Allerdings geht durch die eingeschränkte Funktionalität einiges an Nutzerkomfort verloren. Dadurch ist es lediglich möglich, Standardfunktionen zu verwenden die sich im Raster Desktop Word in den Reitern Start, Einfügen, Seitenlayout, Überprüfen und Ansicht befinden. Es sind jedoch nicht alle Funktionen dieser Reiter implementiert.

Bis zu Office 2013 war der Microsoft Portal Dienst nur im Internet Explorer verfügbar. Dies hat zur Folge, dass es in anderen Browsern öfter zu Fehlern kommen kann. In diesem Punkt hat das System noch Aufholbedarf.

Microsoft Portal verwendet, wie Google Docs, ein Pull und Push Prinzip zur Aktualisierung der Dokumente. Allerdings gibt es keine Möglichkeit, die verwendeten APIs zu verwenden, da die verwendeten Bibliotheken nicht öffentlich zugänglich sind. Dadurch könnten lediglich Ideen aus dieser bestehenden Software mitgenommen werden und keine verwendbaren APIs und dergleichen.

5.5.1.3 Fazit

Google Docs und Microsoft Portal verfolgen dasselbe Ziel, beide wollen eine konsistente Echtzeit Kommunikation in Dokumenten schaffen. Wobei man sagen muss, dass dies bei Google Docs mit weniger Fehlern funktioniert.

Bei Microsoft Portal ist es schade, dass es lediglich eine Lite-Version von Office in der Online Variante gibt, dadurch verliert das Produkt an Wert. valide.

Ein großes Augenmerk liegt auf der Vorgehensweise der einzelnen Systeme. Beide arbeiten mit einem Pull und Push Verfahren. Da Microsoft die verwendeten APIs nicht öffentlich zur Verfügung stellt, war dies keine Hilfe. (A)

Bei Google gibt es mehr Einblick, durch die öffentliche Google Realtime API konnten erste Erkenntnisse daraus gewonnen werden. Dadurch wurde, wie bereits erwähnt, eine eigene Lösung angestrebt.

Im Vergleich schneidet Google Docs besser ab als Microsoft Portal ~~das~~ liegt daran, dass die Technologie bei Google Docs ausgereifter ist und dadurch ohne große Fehler einzusetzen ist.

5.5.2 Umsetzung

Das Parallel Working System besteht aus zwei Modulen. Mit dem ersten Modul ist es möglich, Hefte mit anderen Nutzern zu teilen. Das andere Modul ist für die ständige Aktualisierung und Synchronisation des Heftinhaltes verantwortlich.

5.5.2.1 Hefte teilen

Jedes Heft hat ein Attribut *collaborator* (siehe Kapitel 5.3.2.2). Dieses Attribut ist die Voraussetzung, um das Heft mit anderen Benutzern zu teilen. Um anderen Nutzern ein bestimmtes Heft freizugeben, gibt es folgende Oberfläche:

ft freizugeben, gibt es folgende Oberfläche:

Collaborators

strmitch@gmx.at

philippadler13@gmail.com

Abbildung 46: Hinzufügen eines weiteren Nutzers

Wie man in Abbildung 46 sehen kann schlägt das System nach Eingabe User vor, ~~das~~ diese hinzuzufügen. Durch diese Funktion soll der Komfort beim Hinzufügen eines Collaborators gefördert werden. Bei diesen Suchvorschlägen handelt es sich um die E-Mail Adresse, die gleichzeitig die eindeutige Identifizierung des Users darstellt. Ist ein Heft nun für einen User freigegeben, so wird das Heft in dessen Heftkollektion unter 'Für mich freigegebene Hefte' angezeigt.

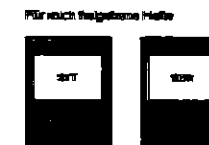


Abbildung 47: Für mich freigegebene Hefte

Im Prinzip kann jedes Heft mit beliebig vielen Nutzern geteilt werden. Dabei spielt es keine Rolle, ob es ein privates oder öffentliches Heft ist. Der Benutzer dem ein Heft freigegeben wurde, hat nur die Bearbeitungsrechte, er kann weder den Namen ändern, noch das Heft löschen. Wird das Notebook vom Besitzer gelöscht, dann wird das Heft auch bei allen Mitbesitzern gelöscht. Hefte können nicht übergeben werden.

Um die Teilung eines Heftes aufzuheben, gibt es zwei Möglichkeiten. Entweder der Besitzer des Heftes löscht den User als collaborator mittels des roten Minus neben dem Namen (siehe Abbildung 46), oder der User entfernt ein geteiltes Heft selbst.

5.5.2.2 Aktualisierung und Synchronisation

Jedes NotebookContent-Element besitzt folgende Attribute:

```
is_active = BooleanField(default=False)
is_active_by = EmailField()
```

Listing 38: Parallel Working System Attribute

✓

Mit dieser Struktur kann darauf geschlossen werden, ob ein Element gerade aktiv ist und wer der aktive Nutzer ist. Diese Grundstruktur ist der Baustein, worauf das Gesamte Parallel Working System aufbaut.

Mit diesen Attributen werden alle Anfragen, welche vom Nutzer auftreten, abgefragt und dementsprechend abgehandelt. Es ist wichtig zu wissen, wer dieses Element gerade bearbeitet, um aktive Nutzer nicht durch Aktualisierungen zu behindern. Die Attribute *is active* und *is active by* ändern sich, wenn der Bearbeitungsmodus eines Elements aktiviert oder deaktiviert wird.

Bei einer Aktivierung und folgenden Bearbeitung des Elements, wird das Attribut *is active* auf *true* gesetzt.

```
$scope.editelement(id, art, {"data": $scope.models[art][id][0],true});
```

Listing 39: Bearbeitungsmodus true - PWS

Sollte der Bearbeitungsmodus vom Benutzer wieder verlassen werden, so wird das Attribut wieder auf den Default-Wert *false* zurückgesetzt.

```
$scope.editelement(id, art, {"data": $scope.models[art][id][0],false});
```

Listing 40: Bearbeitungsmodus false - PWS

Um die derzeitigen User in einem bestimmten Heft auszumachen, gibt es folgende Funktion:

```
def view_get_is_active(request):
    if not request.user.is_authenticated():
        return JsonResponse({})
    if request.method == "POST":
        notebook = Notebook.objects.get(id=request.POST.get('notebook'))
        findnotebook = None
        content = notebook.content
        for item in content:
            if str(item["id"]) == str(request.POST.get('content_id')) and
               item["art"] == request.POST.get('content_art'):
                findnotebook = item
                break
        return JsonResponse({"active": findnotebook.is_active, "active_by":
                             findnotebook.is_active_by})
```

Listing 41: Abfrage der aktiven Nutzer - PWS

Mit dieser Funktion kann das System die aktiven Nutzer herausfiltern und anschließend die Daten weiter verwenden, um die Aktualisierung und Synchronisation zu steuern. Diese Daten befinden sich in einem Log in der Datenbank. Dieses Log speichert nur die aktiven Nutzer im Heft, um nachvollziehen zu können wer sich gerade in welchem Heft aufhält.

Das Prinzip ist, dass immer nur ein Benutzer des Heftes an einem Element arbeiten kann und in dieser Phase von keinem anderen User gestört werden kann. Dadurch soll verhindert werden, dass zwei oder mehr Nutzer an einem Element Änderungen vornehmen und dadurch ein Lost-Update verursachen. Durch die Maßnahme, dass nur eine Person ein Element editieren kann sind die Elemente unmittelbar nach der Aktualisierung für alle User aktuell. Dadurch wird dem Lost-Update Problem entgegen gewirkt. Um zu vermeiden, dass ein User ein Element unbegrenzt blockiert, gibt es ein Timeout, welches den Benutzer nach einer bestimmten Zeit vom Element abmeldet, wenn dieser im Element keinen Tastendruck oder Mausklick ausführt. Wenn dieser User vom System abgemeldet wird, ist das Element anschließend wieder für alle User verfügbar.

Um den anderen Benutzern zu signalisieren, dass ein Element gerade aktiv ist, wird allen anderen Usern ein rotes Rufzeichen über dem Element angezeigt. Ist dies der Fall, ist das gesamte Element nicht editierbar.

Der Begriff deutschsprachige beziehungsweise deutsche Literatur bezeichnet die literarischen Werke in deutscher Sprache aus dem deutschen Sprachraum der Vergangenheit und Gegenwart.

Abbildung 48: Element gesperrt - PWS

Dieses Zeichen verschwindet automatisch, sobald das Element für alle User wieder zugänglich ist. Durch diese Methodik sollen Missverständnisse ausgeschlossen werden. Versucht der Benutzer trotzdem das Element zu verschieben, zu löschen oder in den Bearbeitungsmodus zu gelangen, wird eine Meldung angezeigt, dass dies erst geht sobald der aktive Benutzer den Bearbeitungsmodus verlassen hat.

Die Aktualisierung und Synchronisation funktioniert in zwei Schritten. Zuerst gibt es ein definiertes Aktualisierungsintervall, mit diesem Intervall wird gesteuert, in welchen Abständen das System nach neuen Heftinhalten sucht. Die Funktion, die dieses Intervall steuert, sieht wie folgt aus:

```
$scope.poll = function(){
  $timeout(function() {
    var content = $scope.notebook['content'];
    $http({
      method: 'POST',
      url: '/api/get_notebook',
      data: {id: $stateParams.id}
    }).success(function (data) {
      $scope.notebook = JSON.parse(data['notebook']);
      $scope.content = $scope.notebook['content'];
      if (JSON.stringify($scope.content) != JSON.stringify(content)) {
        $scope.update();
      }
      $scope.poll();
    });
  }, 10000);
};

$scope.poll();
```

Listing 42: Aktualisierung - PWS

Die Funktion ist eine rekursive Funktion. Sie ruft sich selbst in einem vordefinierten Intervall auf. Durch den selbstständigen Funktionsaufruf ist es dem System möglich, die Funktion im Hintergrund laufen zu lassen. Sobald ein Heft aufgerufen wird, beginnt diese Funktion und hört erst dann wieder auf, wenn sich der Benutzer aus dem Heft abmeldet. Somit soll gewährleistet werden, dass der Benutzer gleich nach Eintritt in das Heft auf den aktuellen Stand zurückgreifen kann. Es wird dem User die manuelle Aktualisierung abgenommen. Das System kümmert sich komplett eigenständig darum, dass der User immer den aktuellen Stand der einzelnen Elemente angezeigt bekommt. Bevor allerdings eine Aktualisierung beim User im Heft ausgeführt wird, muss überprüft werden, ob sich der entsprechende Benutzer nicht gerade im Bearbeitungsmodus befindet. Dies muss gemacht werden, um einen eventuellen Fortschritt des User nicht durch eine Aktualisierung zu löschen. Mit dieser Sicherheitsvorkehrung wird dieser erst nach Verlassen des Bearbeitungsmodus aktualisiert. Sobald der Bearbeitungsmodus verlassen wird, gibt es immer eine Aktualisierung des Heftinhaltes auf der jeweiligen Seite. Das System erkennt, auf welcher Seite sich der User gerade befindet und aktualisiert genau diese Seite. Sobald der Benutzer die Seite ändert, wird diese automatisch neu geladen. Somit können Ressourcen eingespart werden und das System gewinnt an Performance.

Um wirklich nur dann zu aktualisieren, wenn tatsächlich eine Veränderung des Heftinhaltes vorliegt, gibt es in der Aktualisierungsfunktion eine zusätzliche Überprüfung.

```
if (JSON.stringify($scope.content) != JSON.stringify(content)) {
  $scope.update();
}
```

Listing 43: Synchronisation - PWS

Diese Anweisung gleicht ab, ob sich der Heftinhalt in der Datenbank vom Angezeigten unterscheidet. Dies wird gemacht, um nicht ständig zu aktualisieren, obwohl keine Änderung vorliegt. Dies ist beispielsweise der Fall, wenn ich mich gerade allein im Heft befinde dann können keine Änderungen außer die selbst vorgenommenen vorliegen. Ist diese if-Anweisung allerdings erfüllt, dann wird die Oberfläche mit der Funktion `scope.update()` aktualisiert.

Durch diese beiden Schritte ist das System immer auf aktuellem Stand und ermöglicht somit eine Umgebung, in der mehrere User gleichzeitig an einem Heft arbeiten können.

Das Parallel Working System wurde speziell für Gruppenarbeiten entwickelt. Dadurch soll der Komfort für ein gemeinsam geführtes Dokument gesteigert werden. Da das Heft immer auf dem aktuellen Stand ist, können gemeinsame Projekte optimal mit DigitalSchoolNotes und dem Parallel Working System durchgeführt werden.

5.5.3 Probleme

Es gibt zahlreiche Collaboration-Frameworks, die eine Echtzeitkommunikation ermöglichen. Die erste Überlegung war, so ein Framework zu verwenden, wie beispielsweise APE Project [APE16]. Mit diesem Framework ist es möglich, Web Kommunikation in Echtzeit abzuhandeln. Allerdings gab es einige Probleme, da dieses Framework nur sehr schwer in das bereits bestehende Projekt integriert werden konnte. Daraufhin wurde diese Idee verworfen und es wurde eine eigenständige Echtzeit Lösung angestrebt. Dadurch konnte das Parallel Working System besser auf dessen Bedürfnisse angepasst werden dies hat den Implementierungsprozess deutlich vereinfacht. (F) (A)

Ein deutlich größeres Problem trat bei der Implementierung der Sperrung der Elemente auf. Die Problematik war, dass trotz aktiver Bearbeitung eines Elements mehrere Nutzer dieses Element bearbeiten konnten. Dadurch kam es zu Synchronisationsfehlern. Durch eine zusätzliche Abfrage vor Eintritt in den Bearbeitungsmodus, wurde diesem Problem entgegengewirkt. (A) *

Die stetige Aktualisierung der Heftseite zeigte einige Schwierigkeiten auf. Durch das ständige Neu laden der Seite wurde sehr viel an Traffic verschwendet. Dadurch gab es Performance Verluste. Um dieses Problem zu lösen, wurde die bereits erwähnte Zusatzfunktion, mit der nur bei einer Änderung aktualisiert wird, eingebaut. Durch diese Maßnahme konnten Ressourcen eingespart werden. konnten. wodurch

Da die zahlreichen Heftelemente mit verschiedenen JavaScript-Frameworks implementiert wurden, gab es Probleme diese einheitlich in das Parallel Working System zu integrieren. Dadurch, dass ein einheitliches Datenmodell verwendet wurde, konnte das PWS an die Bedürfnisse der einzelnen Elemente angepasst werden.

5.5.4 Ausblick

Um das Parallel Working System weiter zu verbessern, gibt es Ideen, um dieses System schneller und performanter zu machen.

Um eine höhere Performance zu erlangen, gibt es die Möglichkeit, das Aktualisierungsverfahren zu verändern. Die Aktualisierung in gleichbleibenden Abständen ist nicht die performanteste Lösung. Es gibt ein Pull und Push Prinzip, wie es beispielsweise die Google Realtime API verwendet. Mit diesem Verfahren wird nach jeder Änderung am System eine Push-Benachrichtigung an den Server geschickt. Anschließend werden die aktiven Nutzer benachrichtigt und der geänderte Inhalt kann aktualisiert werden. ✓

* Durch die wiederholten Aktualisierungen kam es aber auch zu leistungsmäßigen Engpässen, da das so genannte Polling eine entsprechende Systemlast hervorrief

^{müsste}
Um dieses Verfahren in das Parallel Working System zu integrieren, ~~musste~~ das System an mehreren Stellen angepasst werden, ~~durch diese Anpassungen würde das System aber effizienter arbeiten und würde noch näher an eine Echtzeit-Kommunikation herankommen.~~ *

Diese Änderung hätte zur Folge das keine Differenz zwischen Änderung und Aktualisierung mehr ~~herrscht~~ ^{sich} dadurch könnte besser gemeinsam an einer Heftseite gearbeitet werden. ~~Konflikte, wodurch~~

Um weiteren Nutzerkomfort zu schaffen, müsste das Prinzip, dass nur ein Nutzer pro Element arbeiten darf, optimiert werden. Es gäbe die Möglichkeit, diese Regel aufzuheben und ein gleichzeitiges Arbeiten an einem Element zu erlauben. Dies wäre der letzte Schritt zur endgültigen Echtzeit-Kommunikation. Allerdings müsste mit dieser Änderung das ganze Konzept von DigitalSchoolNotes überarbeitet werden, um dies zu ermöglichen. Denn mit der derzeitigen Version wird erst ab Verlassen des Bearbeitungsmodus der aktuelle Stand gespeichert und für alle Nutzer im Heft zur Verfügung gestellt. Mit diesem neuen Prinzip müsste jeder Tastendruck in Echtzeit an die aktiven Nutzer übermittelt werden, um keine Konflikte aufkommen zu lassen. Diese Funktion könnte nur mit einem dementsprechenden Framework durchgeführt werden. Es müsste eine Machbarkeitsstudie durchgeführt werden, um abzuklären, ob eine solch aufwendige Änderung mit dem aktuellen System umsetzbar wäre. (A)

* Die Integration solcher Verfahren in das PWS, würde natürlich Anpassungen an mehreren Stellen erfordern. Der Mehrwert bestünde aber unbestritten in einem performanteren System, das in Richtung einer Echtzeit-Kommunikation ginge.

5.6 Optical Character Recognition

Optical Character Recognition oder auch Optische Zeichenerkennung bezeichnet die automatisierte Texterkennung innerhalb von Bildern. Mit diesem Verfahren können also Bilder eingelesen werden und die darin enthaltenen Zeichen als diese erkannt und ausgegeben werden. OCR ermöglicht es den Nutzern von Digital School Notes Bilder auf den Server zu laden, den textuellen Inhalt aus diesen Bildern zu filtern und als bearbeitbares Textelement in ein Heft einzufügen.

Wir gehen schnell, um die Küh
wohl, daß wir an der hellen Sc
hellen Sonne ...

Wir gehen schnell, um die Küh
wohl, daß wir an der hellen Sc
hellen Sonne ...

Wir gehen schrigJL um die Küh
wohl, daß wir an der hellen Son
hellen Sonne ...

Abbildung 49: Beispiel einer Texterkennung

Die ersten Versuche einer automatischen Zeichenerkennung lassen sich auf die 30er Jahre datieren. Die damals entworfene Maschine für das künstliche Lesen konnte mit einem Mustervergleich die zehn arabischen Ziffern automatisch erkennen. Solche Texterkennungssysteme konnten sich vor allem sehr früh bei Banken und Versicherungen etablieren. Bereits zu Beginn der 60er Jahre war es möglich, mit entsprechenden Rechnern, geschriebene oder gedruckte Zeichen auf Belegen zu identifizieren.

In den folgenden Jahren gelang dann aber auch der Durchbruch beim Erkennen von Buchstaben. Dabei war die Anzahl der benötigten Schriftarten jedoch sehr begrenzt. Die erste Schriftfamilie, OCR-A, beinhaltete neben Ziffern und Großbuchstaben auch 30 Sonderzeichen. Die Weiterentwicklung davon, OCR-B, ermöglichte auch die Erkennung von Kleinbuchstaben.

In den 70er Jahren haben sich bei Banken und Versicherungen schließlich intelligentere Systeme etabliert, die neben ungenormten Schriftarten auch Handschriften erkennen konnten. Da diese Lesegeräte Komplettsysteme waren, also aus Hardware und Software bestanden, waren die Preise dementsprechend hoch.

Erst seit Mitte der 80er Jahre etablierten sich, aufgrund der Entwicklung von Flachbett- und anderen Scannern, sowie leistungsfähigeren Computern, billigere Systeme, die gute Lösungen anboten. Diese OCR-Lösungen haben sich im Laufe der Jahrzehnte stetig weiterentwickelt, bis zu der professionellen Software, die man heute im Handel oder frei verfügbar im Internet findet.

5.6.1 OCR-Verfahren

Das Verfahren zur optischen Zeichenerkennung teilt sich grob gesagt in drei Teile:

- Seiten- und Gliederungserkennung
- Mustererkennung
- Umwandlung in das Ausgabeformat

Zudem gibt es verschiedene Arten der konkreten Zeichenerkennung.

5.6.1.1 Bitmustervergleich

Ein besonders einfaches und wahrscheinlich das älteste Verfahren zur Zeichenerkennung ist der Bitmustervergleich (Matrizenvergleich). Das in den 30er Jahren entwickelte Verfahren vergleicht Zeichen miteinander, indem es die Differenz des digitalisierten Zeichens zu den gespeicherten Zeichennustern betrachtet. Wenn ein betrachtetes Zeichen eine geringe Differenz zum gespeicherten Muster zeigt, wird es als dieses Zeichen identifiziert. Für jedes Zeichen existiert also eine Matrix, in der dieses Zeichen in ein Pixelmuster zerlegt wird.

Quelle deiner
Informationen!

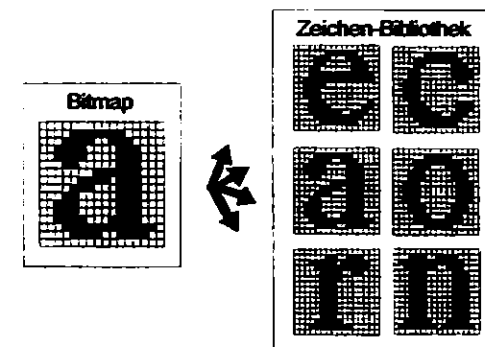


Abbildung 50: Bitmustervergleich

✓

✓

✓

✓



5.6.1.3 Topologische Analyse

Diese Methode wird auch strukturelle Formenanalyse genannt und orientiert sich, ebenso wie die Klassifikation, an spezifischen Eigenschaften der Zeichen und nicht an umfangreichen Mustervergleichen.

Der erste Schritt bei der topologischen Analyse ist die Reduzierung des Pixelbilds auf dessen Grundelemente. Dieser Vorgang wird Vektorisierung genannt. Die überflüssigen Pixel werden also alle entfernt und es entsteht ein „Zeichenskelett“. Danach folgt die eigentliche Zeichenanalyse. Das Verfahren sucht nun nach Beschreibungen, die mit den Einzelelementen des reduzierten Zeichens übereinstimmen.



Abbildung 52: Vektorisierung eines Zeichens

Diese Methode ermöglicht eine sehr fehlertolerante Zeichenerkennung. Sie benutzt komplexe Algorithmen, konzentriert sich aber nur auf wenige Elemente und erreicht dadurch eine angemessene Geschwindigkeit.

5.6.1.4 Neuronale Netze

Neuronale Netze sollen die Schwächen der bereits genannten Verfahren beheben. Mit solchen neuronalen Netzen wird versucht, intelligentes Verhalten auf Computern nachzuvollziehen. Dabei stellt die Neuroinformatik eine Alternative zur klassischen algorithmischen Informatik dar. Die parallele und verteilte Informationsverarbeitung selbst geschieht mit Hilfe einer Vielzahl von Elementen, den sogenannten Neuronen.

Eine wesentliche Eigenschaft bei diesem Verfahren ist die Selbstprogrammierung. Das System kann sein Wissen durch selbstständiges Lernen erweitern. Das neuronale Netz muss aber zunächst kalibriert werden, dies geschieht beispielsweise mit einem Training aus Identifikationsmustern. Sobald das System ein vorliegendes Problem dann mit einem vernachlässigbaren Fehler lösen kann, kann die Kalibrierung abgeschlossen werden.

Die Synapsen eines neuronalen Netzes, also die Verbindungen zwischen den Neuronen, sind nicht statisch und können sich während des Lernprozesses verändern. Es können neue Synapsen entstehen, bestehende Kontakte können ihren Übertragungswiderstand ändern oder ganz aufgelöst werden. Besonders bei der handschriftlichen Erkennung haben neuronale Netze in den letzten Jahren bessere Ergebnisse geliefert als die konkurrierenden Verfahren. Durch eine zusätzliche Erhöhung der Rechenleistung mittels GPU-Implementierungen, konnten Wissenschaftler mit dieser Methode viele Wettbewerbe der Mustererkennung gewinnen. [MP93]

5.6.2 ICR und IWR

Bei der Intelligent Character Recognition werden die digitalen Abbilder der Zeichen mit Wörterbüchern verglichen und anschließend wird die wahrscheinliche Fehlerfreiheit der Zeichen analysiert. Abhängig von dieser wahrscheinlichen Fehlerfreiheit wird das Zeichen identifiziert oder einer erneuten Bewertung unterzogen.

Die Intelligent Word Recognition vergleicht Einzelzeichen, die nicht separat erkannt werden können, mit Wörterbüchern. Die Erkennungsgenauigkeit ist von der Größe des eingebundenen Wörterbuchs abhängig. IWR liefert besonders bei der Handschrifterkennung gute Erfolge.

5.6.3 Implementierung

In den folgenden Kapiteln ist die Implementierung des OCR-Moduls beschrieben.

5.6.3.1 Tesseract OCR und pytesseract-Modul

Der erste Schritt zur Implementierung war die ^{an}Evaluierung der geeigneten OCR-Software. Diese wurde im Evaluierungssprint des Projektes evaluiert. Es wurden drei Softwarepakete evaluiert: Tesseract OCR (pytesseract), OCRopus und OmniPage Ultimate. Nach dem Vergleich dieser Lösungen hinsichtlich mehrerer Kriterien fiel die Entscheidung auf die tesseract-Engine und das dazugehörige Python-Modul pytesseract.

Um Tesseract-OCR (engine) zu installieren, installiert man die folgenden Pakete auf dem Server:

- tesseract-ocr
- tesseract-ocr-deu

kein Tutorialdeutsch

Zum Testen der Analyse führt man tesseract mit folgendem Befehl in der Commandline aus:

```
tesseract test3.jpg out3
```

Listing 44: tesseract-Ausführung

[Sch15] [fos13]

Hier erkennt man das analysierte Bild, sowie den Output der tesseract-Engine.



Abbildung 53: Beispielbild für OCR-Analyse

Output:

THE.FRESH.STEP

Adamski, Coric, Schwertberger

Entwicklung einer neuen Webseite.

die alle Abteilungen, Werkstätten und A / m

Für die Installation der Python-Anbindung pytesseract sind folgende Befehle notwendig.

```
apt-get install python3  
apt-get install python3-pip  
pip3 install pytesseract  
apt-get install python3-pil
```

Listing 45: pytesseract-Installation

[Fou15b] [Del14] [Gri15]

Bitte kein Tutorial schreiben! -> Besser ... es wird...

Mit einem Testscript in Python kann die optische Zeichenanalyse ausgeführt werden.

```
import pytesseract  
from PIL import Image  
print (pytesseract.image_to_string(Image.open('test3.jpg')))
```

Listing 46: pytesseract-Code

5.6.3.2 Implementierung des OCR-Moduls

Um Bilder hochladen zu können und eine Bild-zu-Text-Analyse zu ermöglichen, wurde ein OCR-Modul implementiert. Die Idee dahinter ist, dass die Benutzer unserer Hefte möglichst einfach Bilder auswählen und einer Zeichenanalyse (OCR) unterziehen können. Dabei werden prinzipiell die Dateiformate .jpeg .png und .gif akzeptiert.

Bei der optischen Zeichenanalyse (OCR) wird der am Bild vorhandene Text durch eine OCR-Engine analysiert und in ein Textformat umgewandelt. Für unsere Lösung wurde das python-Framework pytesseract mit der OCR-Engine tesseract benutzt.

Um ein Bild zu analysieren, klickt der User zunächst auf folgenden Button im Heft:



Abbildung 54: OCR-Button

Darauf öffnet sich ein Dialog mit einem Dateieingabefeld:

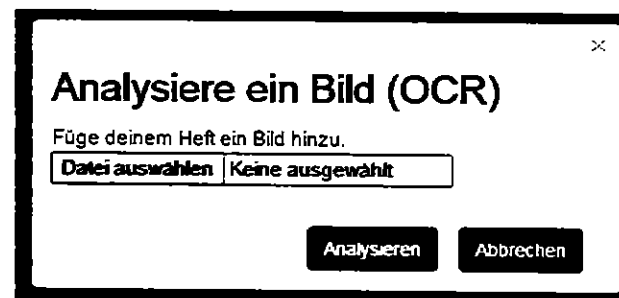


Abbildung 55: OCR-Dialog

Nachdem auf den Button „Analysieren“ geklickt wurde, wird das Bild auf den Server geladen und mittels OCR analysiert. Das Ergebnis ~~das~~ ist ein neu erstelltes Textelement im Heft mit dem textuellen Inhalt des angegebenen Bildes.

Wie schon erwähnt, erscheint bei Klick auf den OCR-Button ein Dialog, in welchem die zu analysierende Bilddatei angegeben werden kann. Bei Klick auf den Button „Analysieren“ wird folgende Subroutine aufgerufen:

```
$scope.uploadOCRFile = function(){
    var file = $scope.ocrFile;
    if((file.type == "image/jpeg" || file.type == "image/png" ||
        file.type == "image/gif") && file.size < 5242880) {
        $scope.errorMessage = "";
        var uploadUrl = "/api/analyseOCR";
        var message = fileUpload.uploadFileToUrl(file, uploadUrl);
        message.then(function(data) {
            data_data = "{\"data\":\""+data['ocrt']+"\"}";
            $scope.addelement('textarea', data_data);
            $window.location.reload();
            $window.location.reload();
        });
        ngDialog.close({
            template: 'ocrFileDialog',
            controller: 'notebookEditCtrl',
            className: 'ngdialog-theme-default',
            scope: $scope
        });
    } else {
        if(file.size > 5242880) {
            $scope.errorMessage = "file size is more than 5MB";
        } else {
            $scope.errorMessage = "filetyp is not supported";
        }
    }
};
```

Listing 47: Upload OCR-File

Hier ist zu sehen, dass zunächst der Dateityp und die Dateigröße überprüft wird. Wenn alles den Kriterien entspricht wird die Methode `uploadFileToUrl()` aufgerufen. Unter der URL `url(r'api/analyseOCR', 'dsn.views.notebook.views.view_analyseOCR', name=„analyseOCR“)` findet sich die Python-Methode `view_analyseOCR()`.

In dieser Methode erhält das File einen neuen, zufällig generierten Namen und wird auf den Server lokal hochgeladen. Dann wird schließlich die Methode, die für das eigentliche Analysieren zuständig ist, aufgerufen:

```
ocrtext=analyseOCR(os.getcwd()+"/dsm/static/upload/"+filename+"."+typ)
```

Listing 48: OCR-Analyse

Schließlich kommt in der Methode `analyseOCR` das `pytesseract`-Framework zum Einsatz:

```
def analyseOCR(file):
    s = str(pytesseract.image_to_string(
        Image.open(file)).encode(sys.stdout.encoding,
        errors='replace'))
    s = s[2:-1]
    s = s.replace("\n", "<br />").replace("\\x", "")
    return s
```

Listing 49: analyseOCR mittels pytesseract

Nach der Analyse wird der erkannte Text als `JSONResponse` zurückgegeben und das lokal hochgeladene Bild wird wieder vom Server gelöscht.

```
os.remove(os.getcwd()+"/dsm/static/upload/"+filename+"."+typ)
return JsonResponse({'ocrt': ocrtext})
```

Listing 50: Fileentfernung und Rückgabe des OCR-Textes

Abschließend wird mit dem daraus analysierten Text ein neues Textelement im Heft erzeugt. Dazu wird die `addelement()` Methode aufgerufen. Aus folgendem Bild ergibt sich dann das folgende Textelement.



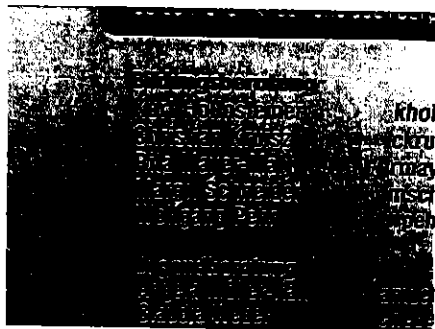


Abbildung 56: Beispielbild für OCR-Analyse

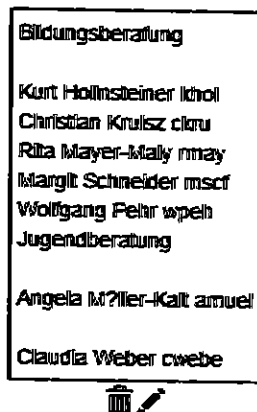


Abbildung 57: Erzeugtes Textelement

5.6.4 Ausblick

Eine Optimierung des OCR-Moduls hinsichtlich Erkennungsgenauigkeit und Handschrifterkennung wurde angedacht. In Zukunft könnte man das OCR-Modul durch entsprechendes (automatisiertes) Training in dieser Hinsicht weiterentwickeln. Eine höhere Fehlertoleranz bei Bildern mit schlechter Qualität ist ebenso wünschenswert.

6 Auswertung und Benchmarks

Dieses Kapitel befasst sich mit den messbaren Ergebnissen des Projektes und der technischen Auswertung. Es wird überprüft, ob die richtigen Frameworks zur Realisierung verwendet wurden und ob die Software ausreichend performant ist und vom Benutzer akzeptiert wird.

6.1 Benutzerakzeptanz

Um die Benutzerakzeptanz bzw. das Interface Design zu überprüfen, gibt es eine Vielzahl von Richtlinien. Des Weiteren gibt es auch ganze Bücher und Papers, welche sich mit diesen Themen befassen. In unserem Projekt haben wir uns nur auf zehn Richtlinien[And11] beschränkt, an welche wir uns halten wollten.

Es ist wichtig, dem User mittels einem eindeutigen Design die Website zu vermitteln. Dieses Design soll vom Benutzer wiedererkannt werden, damit er immer weiß, ob er sich gerade auf der DSN Website befindet. Dazu wurde innerhalb der ganzen Applikation grün in großen Mengen als Signalfarbe verwendet. Des Weiteren ist in jeder Ansicht außer der Heftansicht immer das DSN Logo sichtbar. *Wie es eigentlich?*

Eine eindeutige Benutzerführung ist ebenfalls essentiell. Aus diesem Grund ist das Registrierungsformular direkt auf der Startseite positioniert. Nach der Registrierung erhält der Benutzer eine Email, welche ihn zum Login leitet. Von dort aus wird der Benutzer zum Stundenplan geleitet, welchen er dann eintragen kann. Über ein großes Hauptmenü am oberen Bildschirmrand können alle anderen gewünschten Funktionen erreicht werden.

Eine klare Call-to-Action Führung findet sich ebenfalls auf der Website. Ein in Zukunft erstelltes Werbevideo klärt dem Benutzer beim ersten Besuch der Website über die Funktionen von DSN auf. Im Video wird der User zu einer Registrierung animiert.

Das Feedback an den User fehlt leider ein bisschen auf der DSN Website. Es ist nicht immer klar, ob die Website gerade lädt oder ob einfach nichts passiert. Hier werden in Zukunft noch Ladeanimationen eingefügt, um den Benutzer über einen Ladevorgang zu informieren.

Es ist wichtig, bei der Registrierung so wenig Informationen wie möglich abzufragen, um die Hemmschwelle der Registrierung möglichst gering zu halten. Dies ist vor allem durch die OAuth Registrierung gut realisiert, da hier nur ein Button gedrückt werden muss.

Die Applikation hat in allen Eingabefeldern Default-Werte, um den User besser durch das Formular zu führen. Dies hilft dem User beim Ausfüllen und erklärt ihm, welche Daten eingegeben werden müssen.

Der User sollte immer das Gefühl haben zu wissen, was mit seinen eingegebenen Daten passiert. Dies ist notwendig, um die Eingabe von richtigen Daten zu fördern und um dem User zu erklären, wozu die Daten benötigt werden. (A)

Um den Ärger über die Applikation gering zu halten gibt es keinen Button, welcher das gesamte Formular zurücksetzt. Somit kann der User nicht aus Versehen alle mühsam eingegebenen Formularfelder unabsichtlich löschen. (F)

In der DSN Applikation gibt es viele eindeutige Fehlermeldungen, um den Benutzer über seine Fehleingabe aufzuklären und um eine richtige Eingabe zu fördern.

In DSN gibt es beabsichtigt keine Breadcrumbs, da diese den Benutzer oft verwirren. Die Führung der Menüs ist sehr einfach, somit werden diese auch nicht benötigt.

6.2 Änderungsvorschläge

Im Nachhinein betrachtet wäre es eine gute Idee gewesen, das Full Stack Framework MeteorJS zu verwenden. Durch die Verwendung von Meteor hätten sich im Laufe des Projektes einige Vorteile ergeben.

Zum Einen war das Team bezüglich der vielschichtigen Softwarearchitektur verwirrt. Es war für viele nicht klar, welche Änderung sich auf welchen Teil der Software auswirkt und wie diese Änderung aktiv geschaltet wird. Dadurch wurde viel Zeit verloren. Hier hätte ein Full Stack Framework den Vorteil gehabt, dass die gesamte Software mit einem Befehl zentral deploybar gewesen wäre.

Des Weiteren bietet Meteor viele Features, die händisch implementiert wurden. So ist die Benutzerverwaltung oder die Anmeldung über OAuth bereits ein fertiges Modul in Meteor. Außerdem werden die Daten automatisch zwischen Server und Client synchron gehalten, wodurch es um einiges einfacher gewesen wäre, das PWS zu integrieren. ✓

7 Ausblick

Um den stetigen Verbesserungsprozess von DNS zu gewährleisten, gibt es weitere Ideen, um die Software mit mehr Funktionalität auszustatten und diese somit noch attraktiver für die Nutzer zu machen. ✓

7.1 Bezahlungssystem

Um das System in die Wirtschaft zu bringen, müsste ein Bezahlungssystem eingebaut werden, das die Kostenabwicklung übernimmt. Dieses System wäre die Voraussetzung, um zahlende Kunden für das Produkt zu gewinnen.

Es gibt bereits ein Geschäftsmodell und anhand dessen ~~müsste~~ dieses System aufgebaut sein. müsste

TEST	PRO ACCOUNT	EDUCATIONAL
90 Tage gratis	€4.99 / Monat	10€ / Lizenz
1GB Speicher für Bilder	1GB Speicher für Bilder	Unbegrenzt ab 500 Accounts
Alle Features	Alle Features	Lizenz unbegrenzt gültig
Unbegrenzte Bilder	Unbegrenzte Bilder	Schulbildung Hosting
Begrenzter Anzahl an Medien	JETZT KAUFEN!	KONTAKTIERE UNS!
JETZT REGISTRIEREN!		

Abbildung 58: Geschäftsmodell

Die Bezahlung sollte über bekannte Online-Bezahlösungen wie PayPal, Amazon Payments, etc. abgewickelt werden, um sich rechtlich abzusichern. Durch dieses Bezahlungssystem wäre es dann möglich, die 90 Tage Testaccounts auf Pro Accounts upzugraden. Außerdem gibt es extra für Schulen zugeschnittene Lizenzen, mit denen Schulen eine zeitlich unbegrenzte Lizenz erwerben können, wobei die Schule dann das Hosting des Servers selber übernehmen muss.

7.2 Einblendung von Werbung

Durch die Einblendung von Werbung in den 90 Tage Testversionen sollen neue Einnahmequellen geschaffen werden. Dadurch sollten diese Probeaccounts finanziert werden. Durch diese Verbesserung wäre es vorstellbar, dass User, die störungsfrei arbeiten wollen, ihren 90 Tage Probeaccount schneller verlängern und somit dauerhafter Nutzer von DigitalSchoolNotes werden. Diese Werbungen sollen über das Google AdSense Werbenetzwerk bezogen werden. Diese Art von Werbung ist in Webapplikationen eine gängige Methode, um Werbung einzublenden. ✓

7.3 Zeichenelement

Um die Palette an Elementen weiter aufzustocken, gibt es die Idee eines Zeichenelements. Mit diesem Element soll es dem User möglich sein, direkt im Heft eine Skizze anzulegen und diese dann nach Belieben in späterer Folge weiter zu editieren. Dadurch würde einiges an Aufwand wegfallen, um eine Skizze, beziehungsweise eine Zeichnung, in einem Heft zu platzieren.

Hierbei soll gefördert werden, dass die Nutzer Gedanken im Sinne von Skizzen, Mindmaps, etc. schnell in einem Heft anlegen können, wie sie es auch in einem Handgeschriebenen Heft machen würden. Mit diesem Tool würde man näher an ein komplettes Notizheft heranrücken.

*die Idee eines handschriftlichen
Mikroheft*

7.4 Video einbindung

Um das Heft noch interaktiver zu gestalten könnte man die Elemente um ein Video-Element erweitern. Mit diesem Element wäre es dem User möglich, zum Beispiel Videos, die den Unterricht betreffen, in das interaktive Schulheft einzubinden. Somit könnten Video Tutorials direkt im Schulheft angesehen werden. Um Ressourcen zu sparen, wäre die Idee, diese Videos nicht auf dem DigitalSchoolNotes Server abzuspeichern, sondern lediglich Links in das Heft einzubinden. Somit müssten nur Links aus Videoportalen eingebunden werden und diese könnten anschließend direkt im Notebook abgespielt werden. Durch diese Erweiterung der Elemente könnten klare Abgrenzungen zu normalen Textdokumenten geschaffen werden. Es würde ein interaktives Notizheft mit vielen verschiedenen Inhalten entstehen. *würde*

7.5 Drag and Drop von Bildern

Durch eine zusätzliche Drag and Drop Funktion um Bilder einzufügen, könnten dem User viel an Aufwand abgenommen werden. Somit müsste ein Bild nur mehr in das Heft hineingezogen werden und das Bild wäre im Heft platziert. Mit dieser Methode könnten Bilder schneller in ein Heft hinzugefügt werden. *um das zu*

7.6 PDF Download

Durch die Möglichkeit eines PDF-Downloads des Heftes wäre es dem User möglich geschriebenen Inhalt auch offline verfügbar zu haben. Durch diese Methode könnte man das jeweilige Heft zum Beispiel ausdrucken und archivieren. Außerdem würde mit dieser Funktion das Problem der maximalen Heftanzahl aufgelockert werden. Es wäre anschließend möglich, nicht mehr gebrauchte Hefte als PDF herunterzuladen und danach zu löschen, *wodurch* somit könnte Platz für aktuelle Fächer beziehungsweise Hefte geschaffen werden. *könnte*

7.7 Shortcuts

Für eine bessere Bedienbarkeit von DigitalSchoolNotes könnten Shortcuts in das System integriert werden, um so schneller gewünschte Funktionen auszuführen. Somit könnten beispielsweise einfache Elemente mittels einer simplen Tastenkombination in das Heft eingebunden werden. Durch diese geringfügige Verbesserung würde der Userkomfort gesteigert werden. Durch dieses Feature könnte effizienter mit der Web-Applikation DigitalSchoolNotes gearbeitet werden.

Es könnte auch unter anderem eine Suchfunktion innerhalb eines Heftes geben, wie es in den meisten Systemen mit Str + F üblich ist. Dadurch könnten einzelne Elemente untersucht werden und dann automatisch zu diesen verwiesen werden. Damit wäre es möglich, besser und schneller in zum Beispiel geteilten Heften nach dem passenden Inhalt zu suchen.

! Seite verknüpfen !

7.8 Filehoster-Anbindung

Derzeit ist es möglich *beispielsweise* zum Beispiel Bilder in ein Heft hochzuladen. Diese Bilder werden dann auf dem DigitalSchoolNotes Server abgespeichert. Allerdings benötigen diese Bilder trotz Komprimierung eine Menge an Ressourcen. Um dem User zu ermöglichen, beliebig viele Bilder in sein Heft zu importieren, gibt es eine Möglichkeit. Durch die Anbindung an ein Filehosting-System, wie beispielsweise Dropbox, könnten Bilder direkt aus dem Shared Folder in das Heft eingebunden werden. Dadurch könnte jeder User seinen Speicher von außerhalb verwenden und hätte somit alle Dateien auf dem selben Ort abgelegt, *was auch hilft* *an*

Dies hätte den großen Vorteil, dass schnell und effizient mehrere Systeme gleichzeitig auf diese Medien zugreifen können. Durch die Verbindung mit bereits verwendeter Software könnte der Benutzer einiges an Administrationsaufwand abgenommen werden. Dadurch könnte man DigitalSchoolNotes überall verwenden und hätte alle Dateien, die sich in solchen Shared Foldern befinden, jederzeit zur Verfügung.

7.9 Web Untis Anbindung

Da DigitalSchoolNotes als Zielgruppe Schulen hat, sollte darüber nachgedacht werden, eventuell Stundenplansysteme wie WebUntis in die Web-Applikation einzubinden. Somit müsste der Stundenplan auf der DSN Website nicht mehr manuell eingegeben werden, sondern würde direkt nach Verbindung mit dem ~~Web Untis System~~ der Schule synchronisiert werden.

Durch diese Neuverbindung würde dem User einiges an Administrationsaufwand abgenommen werden. Es müssten maximal kleine Anpassungen vorgenommen werden, um den Stundenplan im selben Rahmen zu nutzen, wie ihn Web Untis zur Verfügung stellt. Dadurch müssten lediglich die Hefte mit dem neuen Stundenplan verbunden werden und danach könnte man diesen genau wie einen manuell eingegebenen Stundenplan nutzen.

Stundenplansystem

• beispielweise

Ⓐ

7.10 LDAP-Anbindung

Um bereits bestehende Lösungen, die in einer Schule eingesetzt werden, zu nutzen, gibt es die Idee, bestehende ~~LDAP~~ Systeme einer Schule mit DigitalSchoolNotes zu verbinden. Somit müssten sich die Nutzer nicht mehr extra für DSN anmelden, sondern könnten bereits verwendete Accounts verwenden. Dadurch verringert sich der Administrationsaufwand drastisch.

verringern würde, wodurch

4 Verzeichnisdienste

8 Zusammenfassung

DigitalSchoolNotes ist eine Web-Applikation zur einfachen Führung einer digitalen Mitschrift.

Für die Registrierung bei DSN gibt es zwei Möglichkeiten. Einerseits gibt es die Möglichkeit, sich mittels Eingabe der E-Mail Adresse, sowie Name und Passwort zu registrieren. Die weitaus einfachere Variante ist hierbei allerdings die Registrierung mit OAuth. Dabei verwendet man ein bestehendes Konto eines sozialen Netzwerkes wie Facebook oder Google+, um sich bei DSN zu registrieren. Anschließend kann man sich erfolgreich in die Applikation einloggen.

Ist man zusätzlich zu einem normalen User noch ein Administrator, gibt es die Möglichkeit der Nutzung einer eigenen Plattform, um die Nutzer zu administrieren. Mit diesem Interface ist es dem Administrator möglich, User zu benachrichtigen oder diese bei Bedarf zu löschen. Außerdem können die Rechte der einzelnen Nutzer manuell administriert werden.

Für eine optimale schulische Organisation ist ein Stundenplan implementiert. In diesen können manuell Unterrichtszeiten, sowie Fächer, Räume und Lehrer eingetragen werden. ~~Man kann~~ *geben wird* dem Nutzer mehr Freiraum in der Gestaltung seines eigenen Stundenplans.

- 1 Die weitaus wichtigste Funktion ist das ~~Hefte~~ *der* Anlegen und die Bearbeitung dieser. Es können Hefte angelegt werden, die einen bestimmten Namen erhalten und es wird entschieden, welche Sichtbarkeit das Heft hat entweder öffentlich oder privat. Diese Angabe ist wichtig, denn Öffentliche werden auf dem DSN-Profil angezeigt. *Ⓐ*

In dem erstellten Heft können dann diverse Elemente eingebunden werden, um den Inhalt in das Heft zu bringen. Diese verschiedenen Elemente können beliebig im Heft platziert werden. Durch ~~Klicken~~ *ziehen* und Ziehen des Elements kann das jeweilige Element verschoben werden. Mittels Hover können Elemente schnell und leicht wieder aus dem Heft entfernt werden.

Um einfachen Text in das Heft einzufügen gibt es das Textelement. Mit diesem ist es dem Benutzer möglich, einfachen Text einzutragen und diesen dann entsprechend zu editieren. Um den Text außerdem hervorzuheben, können Effekte, wie Fett, Kursiv, Unterstrichen, etc. auf den Text angewendet werden. Des Weiteren können Tabellen in dieses Element eingefügt werden. Durch Hyperlinks kann zusätzlicher Inhalt außerhalb von DigitalSchoolNotes angemerkt werden.

1 ist das Anlegen und die Bearbeitung d. Hefte.

Da in einem Heft Erklärungsgrafiken oder zusätzliche Medien nicht fehlen dürfen, gibt es ein eigenes Bild-Element. Bilder können auf den DigitalSchoolNotes Server hochgeladen werden und anschließend in das Heft eingebunden werden. Die Bilder können eine beliebige Größe annehmen und beliebig im Heft platziert werden. Die Größe kann laufend geändert werden.

Dadurch, dass eine digitale Mitschrift besonders in technischen Schulen vorkommt, gibt es ein eigenes Codeelement, mit dem Code Snippets eingebunden werden können. Diese Codezeilen können je nach Programmiersprache oder Scriptsprache verschieden hervorgehoben werden. Das Codeelement unterstützt unter anderem XML, HTML, JAVA, JavaScript, PHP, Python, C, C++, C# und SQL. Um noch eine bessere Lesefreundlichkeit zu schaffen, sind auf der linken Seite Zeilennummern automatisch integriert, um den Code besser im Überblick zu behalten und eventuelle Referenzierungen anlegen zu können. Dadurch können einzelne Codeabschnitte optimal in die digitale Mitschrift integriert werden

Es kann vorkommen, dass man einen gewissen Teil aus öffentlichen Heften interessant findet und diesen gerne im eigenen Heft hätte. Kein Problem. Mithilfe eines Features ist es möglich, Seiten aus öffentlichen Heften in mein eigenes Heft zu importieren. Dadurch können Inhalte geteilt und anschließend erweitert werden.

Um bereits niedergeschriebenen Text weiter zu editieren, gibt es eine Funktion, welche die optische Zeichenerkennung (Optical Character Recognitions) unterstützt. Mithilfe dieses Features kann der User gedruckten Text in bearbeitbaren Text umwandeln. Durch eine OCR-Engine kann ein Bild auf den Server hochgeladen werden, das nach der Analyse wieder gelöscht wird. Das hochgeladene Bild wird von der integrierten OCR-Engine analysiert und anschließend in ein Textelement ausgegeben. Dieses Element kann nach der Analyse wie ein normales Textelement verwendet werden. Somit kann an bereits bestehendem Text mühelos weitergearbeitet werden *hier* *undurch*

Gruppenarbeiten oder Referate mit Mehrpersonenteams sind im Schulalltag kaum mehr wegzudenken. Darum gibt es mit DigitalSchoolNotes das Parallel Working System. Mit diesem System ist es möglich, Hefte mit anderen User zu teilen und diese dann gleichzeitig zu bearbeiten. Mit diesem System kann man beliebig viele Nutzer an der Bearbeitung eines Hefes teilhaben lassen. Diese User haben dann die Möglichkeit, geschriebenen Heftinhalt anderer User in Echtzeit zu verfolgen. Durch dieses integrierte System können Gruppenarbeiten optimal durchgeführt werden.

Anhang

Glossar

API Application Programming Interface.....	51
CAPTCHA Completely Automated Public Turing test to tell Computers and Humans Apart	57
CSS Cascading Style Sheets.....	15
DNS Domain Name Service	48
DSN Digital School Notes	1
HTTP Hypertext Transfer Protocol.....	48
HTTPS Hypertext Transfer Protocol Secure	48
IDE Integrated Development Environment	52
KVM Kernel-based Virtual Machine	47
Latenz Verzögerung zwischen Anfrage und Antwort (Ping)	47
OAuth Open Authentication	58
REST Representational State Transfer	35
SSH Secure Shell	48
SSL Secure Sockets Layer	50
Stakeholder Eine Person welche Interesse am Ergebnis des Projektes hat, jedoch nicht an der Entwicklung des Projektes an sich beteiligt ist.	46
TGM Technologisches Gewerbe Museum	48
WSGI Web Server Gateway Interface	51

Abbildungen

1	Phasen des klassischen V-Modells [MP10]	4
2	Klassisches PM vs. Agiles PM [Mö15]	5
3	Beispiel eines Sprint-Burndown-Charts [Dre12a]	8
4	Scrum-Ablauf [Dre12b]	10
5	Beispiel für eine User Story in OnTime (selfmade)	11
6	Ansicht eines Pomodoro-Timers (selfmade)	11
7	Burndown-Chart des Evaluierungssprints (selfmade)	12
8	Burndown-Chart des ersten Sprints (selfmade)	12
9	Burndown-Chart des zweiten Sprints (selfmade)	13
10	Burndown-Chart des dritten Sprints (selfmade)	13
11	Burndown-Chart des vierten Sprints (selfmade)	14
12	Burndown-Chart des letzten Sprints (selfmade)	14
13	CodeMirror [Cod16]	24
14	Sketch.js [BI11]	24
15	CKEditor [CKS16]	25
16	Slideshow [Pho14]	25
17	Ablaufdiagramm (selfmade)	34
18	Technologische Übersicht (selfmade)	36
19	Aufbau der Server (selfmade)	37
20	Layout der grafischen Oberfläche (selfmade)	38
21	Menüleiste der Hauptseite (selfmade)	39
22	Menüleiste nach dem Anmelden (selfmade)	40
23	Menüleiste der Admin-Seite (selfmade)	40
24	Glyphicons innerhalb der Applikation (selfmade)	40
25	Frühes Design der Heftansicht (selfmade)	41
26	Verbessertes Design der Heftansicht (selfmade)	42
27	Firebug Javascript Debugger [Fir14]	44
28	Loadbalancing [Cep14]	50
29	Deployment Struktur (selfmade)	51
30	Functional Testing Level [Bur03]	53
31	Registrierung bei DSN (selfmade)	56
32	Lösen des Captchas (selfmade)	57
33	Klassische Anmeldung oder mittels OAuth (selfmade)	59
34	Navigationsleiste als User (selfmade)	60
35	Navigationsleiste als Admin (selfmade)	61
36	Usermanagement-Page (selfmade)	62
37	Heftelemente (selfmade)	71
38	Textelement (selfmade)	72
39	Textelement im Bearbeitungsmodus (selfmade)	72
40	Textelement im Ansichtsmodus (selfmade)	72
41	Code-Button (selfmade)	74
42	Programmiersprache auswählen (selfmade)	75
43	Leeres Code-Element (selfmade)	75
44	Code-Bearbeitung (selfmade)	76
45	Bild zum Heft hinzufügen (selfmade)	77
46	Hinzufügen eines weiteren Nutzers (selfmade)	80
47	Für mich freigegebene Hefte (selfmade)	80
48	Element gesperrt - PWS (selfmade)	83
49	Beispiel einer Texterkennung [mic11]	87
50	Bitmustervergleich [MP93]	88
51	Klassenbaum für Merkmalsanalyse [MP93]	90
52	Vektorisierung eines Zeichens [MP93]	91
53	Beispielbild für OCR-Analyse (selfmade)	93
54	OCR-Button (selfmade)	94
55	OCR-Dialog (selfmade)	94
56	Beispielbild für OCR-Analyse (selfmade)	97
57	Erzeugtes Textelement (selfmade)	97
58	Geschäftsmodell (selfmade)	100

Listings

1	Installation von Flask [Ron16]	16
2	Flask Hello-World [Ron16]	16
3	Installation von Django[Fou16b]	16
4	Django Hello-World [sun15]	17
5	Konfiguration von Play [Pla15a]	17
6	Play Hello-World [Pla15a]	17
7	Dojo einbinden[Fou16c]	18
8	jQuery einbinden[jF16]	19
9	AngularJS einbinden[Goo16a]	20
10	Installation von Meteor [met16]	21
11	YAML einbinden [Jes13a]	22
12	Pure einbinden [Yah14]	23
13	Bootstrap einbinden [Lei13]	23
14	Installation von Watir [Wat14b]	26
15	Installation von Robot Framework [Rob16b]	27
16	Installation von MySQL [dV07]	28
17	Konfiguration von MySQL [Sve12]	29
18	Installation von PostgreSQL [Wik15]	29
19	Konfiguration von PostgreSQL [Wik15]	30
20	Installation von MongoDB [Doc16]	30
21	Installation von Couchbase [Doc15]	31
22	Konfiguration von Couchbase	31
23	Unterschiedliche Möglichkeiten eine Funktion zu deklarieren[Cal11]	43
24	Firewall Rules des Projektsservers	49
25	Installation iptables-persistent	49
26	Selenium Element Selektoren	54
27	Selenium Test	55
28	Einbindung der JS-Library Recaptcha	57
29	Validierungstoken für die Aktivierung des DSN-Accounts	58
30	Session Timeout	59
31	Cronjob für die Überprüfung der Inaktivität und Löschung	62
32	MongoDB Konfiguration in Django	68
33	Beispiel für ein Datenbankmodell in Django	69
34	Syntax für eine Suchanfrage an die Datenbank in Django	69
35	Speicherung des Textelements	73
36	Einbindung des Textelements	73
37	Code-Element Persistierung	76
38	Parallel Working System Attribute	81
39	Bearbeitungsmodus true - PWS	81
40	Bearbeitungsmodus false - PWS	81
41	Abfrage der aktiven Nutzer - PWS	82
42	Aktualisierung - PWS	83

43	Synchronisation - PWS	84
44	tesseract-Ausführung	93
45	pytesseract-Installation	93
46	pytesseract-Code	94
47	Upload OCR-File	95
48	OCR-Analyse	96
49	analyseOCR mittels pytesseract	96
50	Fileentfernung und Rückgabe des OCR-Textes	96

Quellen

- [And11] Paul Andrew. 10 essential web application usability guidelines. <https://speckyboy.com/2011/03/31/10-essential-web-application-usability-guidelines/>, 2011.
- [APE16] APE. Ape (ajax push engine). <http://ape-project.org/>, 2016.
- [AST08] Maarten van Steen Andrew S. Tanenbaum. *Verteilte Systeme*. PEARSON, 2 edition, 2008.
- [BI11] Michael Bleigh and Intridea. Sketch.js. <http://intridea.github.io/sketch.js/>, 2011.
- [Bol16] Max Bold. Javascript-frameworks. <http://www.maxbold.com/js-frameworks.php>, 2016.
- [Bur03] Ilene Burnstein. Software testing level chart. http://images.books24x7.com/bookimages/id_16280/fig6-1.jpg, 2003.
- [Cal11] David Calhoun. Different ways of defining functions in javascript (this is madness!). <http://davidcalhoun.com/2011/different-ways-of-defining-functions-in-javascript-this-is-madness/>, 2011.
- [Cep14] Lukasz Cepowski. Loadbalancing with nginx. <http://cepa.io/image/6>, 2014.
- [CKS16] CKSource. Ckeditor. <http://ckeditor.com/>, 2016.
- [Cod16] CodeMirror. Codemirror. <https://codemirror.net>, 2016.
- [Cor16] Oracle Corporation. About mysql. <http://www.mysql.com/about>, 2016.
- [Cou16] Couchbase. About couchbase. <http://www.couchbase.com/about>, 2016.
- [DCH15] Dr. Dr. Jörg Berwanger Dr. Cordula Heldt. Allgemeine geschäftsbedingungen (agb). <http://wirtschaftslexikon.gabler.de/Archiv/5433/allgemeine-geschaeftsbedingungen-agb-v9.html>, 2015.
- [Del14] Delimity. Installing tesseract for python on ubuntu 14.04. <http://delimity.blogspot.co.at/2014/10/installing-tesseract-for-python-on.html>, 2014.
- [DI15] Klaus Lipinski Dipl.-Ing. OAuth(open authentication). <http://www.itwissen.info/definition/lexikon/OAuth.html>, 2015.
- [Doc15] Couchbase Docs. Ubuntu/debian installation. <http://docs.couchbase.com/admin/admin/Install/Ubuntu-install.html>, 2015.
- [Doc16] MongoDB Docs. Install mongodb on debian. <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-debian>, 2016.
- [Dre12a] Eric Dreyer. Beispiel eines sprint-burndown-charts. WWU Münster, 2012.
- [Dre12b] Eric Dreyer. Scrum-ablauf. WWU Münster, 2012.
- [dV07] Ariejan de Vroom. How to install mysql on ubuntu/debian. <https://ariejan.net/2007/12/12/how-to-install-mysql-on-ubuntu-debian>, 2007.
- [Fir14] Firebug. Firebug. <http://getfirebug.com/perch/resources/scriptpanel.png>, 2014.
- [fos13] fosshelp. How to convert jpg to tiff for ocr with tesseract. <http://fosshelp.blogspot.co.at/2013/04/how-to-convert-jpg-to-tiff-for-ocr-with.html>, 2013.
- [Fou15a] Django Software Foundation. Writing custom django-admin commands. <https://docs.djangoproject.com/en/1.9/howto/custom-management-commands/>, 2015.
- [Fou15b] Python Software Foundation. pytesseract 0.1.6. <https://pypi.python.org/pypi/pytesseract>, 2015.
- [Fou16a] Django Software Foundation. Django. <https://www.djangoproject.com>, 2016.
- [Fou16b] Django Software Foundation. Third-party distributions of django. <https://code.djangoproject.com/wiki/Distributions>, 2016.
- [Fou16c] The Dojo Foundation. Get the dojo toolkit. <https://dojotoolkit.org/download/>, 2016.
- [GM15] Paul GM. K7 striking features of meteorjs. <http://ezinearticles.com/??-Striking-Features-of-Meteorjs&id=9212264>, 2015.
- [Goo16a] Google. Angularjs. <https://angularjs.org>, 2016.
- [Goo16b] Google. Google realtime api. <https://developers.google.com/google-apps/realtime>, 2016.
- [Gri15] Grimhacker. Installing pytesseract - practically painless. <http://grimhacker.com/wordpress/2014/11/23/installing-pytesseract-practically-painless/>, 2015.
- [Gro16] The PostgreSQL Global Development Group. About. <http://www.postgresql.org/about>, 2016.
- [Hav15] Marijn Haverbeke. Codemirror. <http://codemirror.net/>, 2015.
- [Jes13a] Dirk Jesse. Quick start demos. <http://www.yaml.de/docs/index.html>, 2013.
- [Jes13b] Dirk Jesse. Yaml 4 css framework. <http://www.yaml.de>, 2013.
- [jF16] The jQuery Foundation. Downloading jquery. <http://jquery.com/download/>, 2016.

- [Kre15a] Thomas Krenn?? Brute-force-angriffe?? http://www.password-depot.de/know-how/brute_force_angriffe.htm, 2015.
- [Kre15b] Thomas Krenn. Iptables firewall regeln dauerhaft speichern. https://www.thomas-krenn.com/de/wiki/Iptables_Firewall_Regeln_dauerhaft_speichern, 2015.
- [Lei13] Wlad Leirich. Twitter bootstrap: Einführung in das css-framework [teil 1]. <http://www.work-paper.de/twitter-bootstrap-css-framework/>, 2013.
- [met16] meteoris. Installing meteor on os x, linux and windows. <https://www.meteor.com/install>, 2016.
- [mic11] michael.s. Beispiel einer texterkennung. <https://de.wikipedia.org/w/index.php?curid=6197497>, 2011.
- [Mon16] Inc. MongoDB. About us. <https://www.mongodb.com/company>, 2016.
- [MP93] Christian Klitscher Mathias Petri. *Scannen und optische Zeichenerkennung*. ADDISON-WESLEY, 1 edition, 1993.
- [MP10] S. Seyfert Michael Pätzold. Phasen des v-modells. <https://commons.wikimedia.org/w/index.php?curid=31080279>, 2010.
- [Mö15] Thor Möller. Klassisches pm vs. agiles pm. <https://www.youtube.com/watch?v=LjUeJUP4YU>, 2015.
- [Mü15] Georg Müller. Be powerful sencha extjs. <http://www.treefish.de/loesungen/extjs/>, 2015.
- [Nie12] Alexander Niemann. jquery. <http://www.itwissen.info/definition/lexikon/jQuery.html>, 2012.
- [Pat07] Patrick. jquery - ein mächtiges und effizientes werkzeug. <http://www.ajaxschmiede.de/jquery/jquery-ein-maechtiges-und-effizientes-werkzeug/>, 2007.
- [Pea15] LaTonya Pearson. The four levels of software testing. <http://www.seguetech.com/blog/2013/07/31/four-levels-software-testing>, 2015.
- [Pho14] PhotoSwipe. Photoswipe. <http://photoswipe.com>, 2014.
- [Pla12] Florian Plag. Javascript-frameworks: Einführung in angularjs. <http://html5-mobile.de/blog/angularjs-javascript-mvc-framework-tutorial>, 2012.
- [Pla15a] Play. Installing play. <https://www.playframework.com/documentation/2.4.x/Installing#Quick-Start>, 2015.
- [Pla15b] Play. Play framework. <https://www.playframework.com>, 2015.
- [Rob16a] Robotframework. Robot framework. <http://robotframework.org>, 2016.
- [Rob16b] Robotframework. User guide. <http://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html>, 2016.
- [Ron16] Armin Ronacher. Flask. <http://flask.pocoo.org>, 2016.
- [Sch15] Heinrich Schwietering. Tesseract-ocr. <http://wiki.ubuntuusers.de/tesseract-OCR>, 2015.
- [Sen16] Sencha. Sencha licensing. <https://www.sencha.com/legal/>, 2016.
- [Sha15] Rahil Shaiksh. Google recaptcha with angularjs. <http://code.ciphertrick.com/2015/05/19/google-recaptcha-with-angularjs/>, 2015.
- [Ste15] Ralph Steyer. 25.2.1 download des dojo toolkits oder direkte einbindung aus dem internet. <https://goo.gl/iGPYzD>, 2015.
- [sun15] sundip. Hello world in django [python]. <http://code.runnable.com/UWRVp6lLuONCAABD/hello-world-in-django-for-python>, 2015.
- [Sve12] Etel Sverdllov. How to create a new user and grant permissions in mysql. <https://www.digitalocean.com/community/tutorials/how-to-create-a-new-user-and-grant-permissions-in-mysql>, 2012.
- [Tec15] TechFerry. Extjs vs angularjs. <http://www.techferry.com/articles/ExtJS-vs-AngularJS.html>, 2015.
- [tut15] tutorialspoint. Software testing - levels. http://www.tutorialspoint.com/software_testing/software_testing_levels.htm, 2015.
- [ua01] Kent Beck und Andere. Manifesto for agile software development. <http://agilemanifesto.org/>, 2001.
- [ubu15] ubuntuusers. Cron. <https://wiki.ubuntuusers.de/Cron/>, 2015.
- [Wat14a] Watir. Automated testing that doesn't hurt. <http://watir.com>, 2014.
- [Wat14b] Watir. Installation. <http://watir.com/installation/>, 2014.
- [Wik15] Debian Wiki. Postgresql. <https://wiki.debian.org/PostgreSQL>, 2015.
- [Yah14] Yahoo. Pure.css. <http://purecss.io>, 2014.

Zeitaufzeichnung

Die Aufzeichnung der Arbeitszeit wurde online in einem Google-Spreadsheet durchgeführt. Jedes Teammitglied war selbst für das Eintragen seiner Arbeitszeit verantwortlich. Die folgende Tabelle zeigt die Summe der Arbeitsstunden pro Teammitglied. Das komplette Spreadsheet kann unter <https://goo.gl/SuY1RF> gefunden werden.

	Anzahl der Arbeitsstunden
Philipp Adler	229
Selina Brinnich	278
Niklas Hohenwarter	275
Adin Karic	224
Thomas Stedronsky	229
Summe	1235

