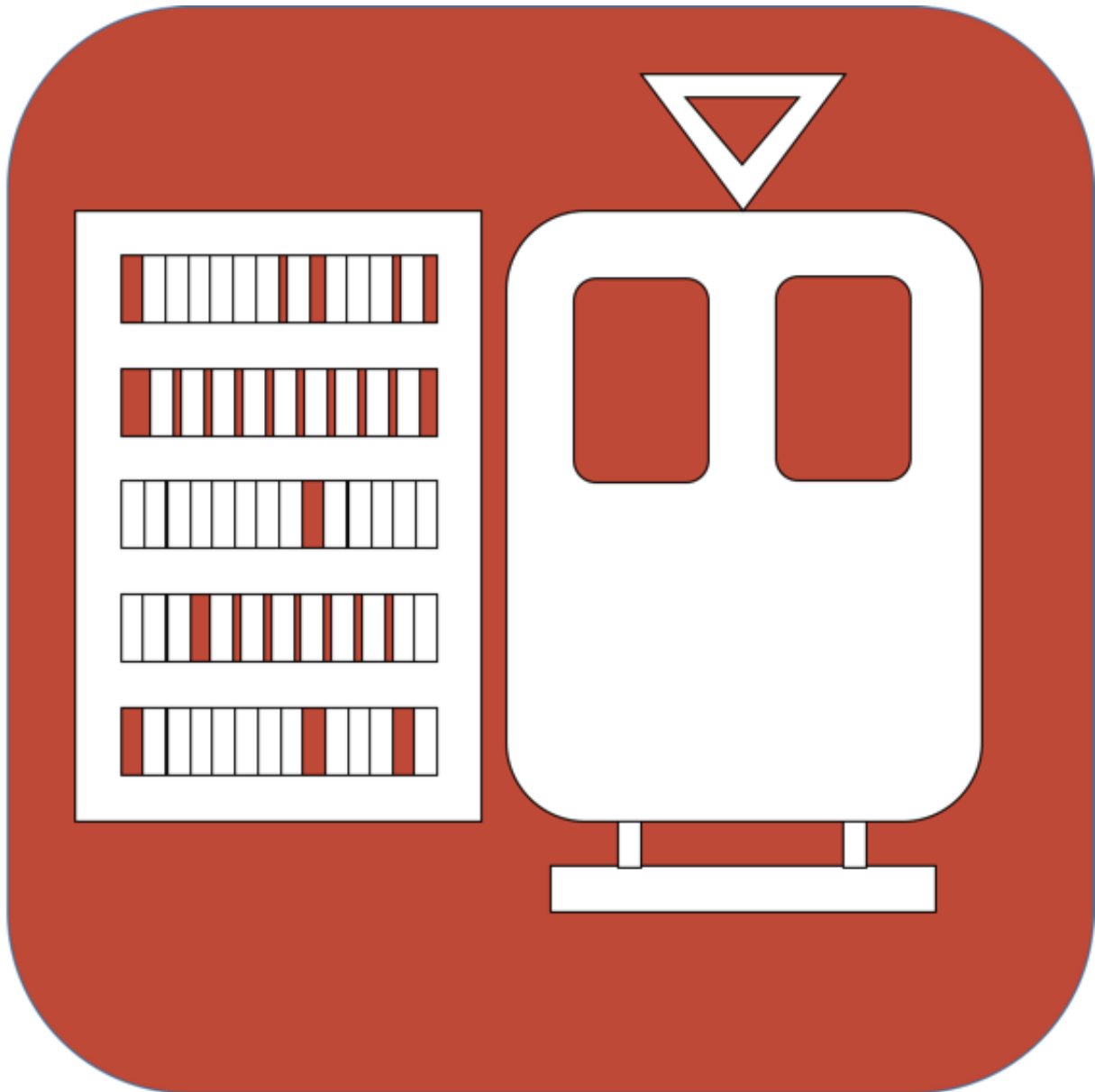


Webbasierte Verwaltung für Reservematerial



Christoph Grossmann / Thomas Baechler
L-TIN-16-Fr-a
Bruno Hammer
Schweizerische Fachschule TEKÖ Luzern
20. Mai 2018

Inhaltsverzeichnis

INHALTSVERZEICHNIS	2
<u>1 MANAGEMENT SUMMARY.....</u>	<u>5</u>
<u>2 EINLEITUNG</u>	<u>6</u>
2.1 VORWORT.....	6
2.2 PROJEKTTEAM	6
2.3 BEGRIFFLICHKEITEN	7
2.4 VORGEHEN	7
2.5 PROJEKTSTRUKTURPLAN	8
2.6 RESULTATE.....	8
2.7 AUSSICHT	8
<u>3 PROJEKTAUFTRAG</u>	<u>9</u>
3.1 RAHMENBEDINGUNGEN.....	9
3.2 UMFELD.....	9
<u>4 PFLICHTENHEFT.....</u>	<u>10</u>
4.1 ZIELBESTIMMUNGEN	10
4.1.1 MUSS-KRITERIEN	10
4.1.2 WUNSCH-KRITERIEN	10
4.1.3 ABGRENZUNG.....	10
4.2 EINSATZ.....	10
4.3 FUNKTIONEN	10
4.3.1 SORTIEREN	10
4.3.2 SUCHEN	10
4.3.3 BEARBEITEN	10
4.3.4 HINZUFÜGEN	10
4.3.5 LÖSCHEN.....	10
4.3.6 EIN-/AUSBUCHEN (WUNSCH)	11
4.4 BENUTZEROBERFLÄCHE (LAYOUT, ZUGRIFFSRECHTE)	11
4.4.1 ADMIN:	11
4.4.2 BENUTZER:.....	11
4.5 TECHNISCHES UMFELD.....	11
<u>5 ANALYSE</u>	<u>12</u>
5.1 LÖSUNGSVARIANTEN	12
5.1.1 HARDWARE	12
5.1.1.1 Variante Hosting bei einem Hosting Anbieter.....	12
5.1.1.2 Variante System auf eigener Umgebung aufbauen	12

5.1.2	SOFTWARE	13
5.1.2.1	Variante Microsoft: Windows Maschine mit IIS und SQL Datenbank	13
5.1.2.2	Variante: Apache und MySQL	13
5.1.2.3	Variante «TEKO» OpenSource: Raspbian, Apache Tomcat und Mongo DB	14
5.2	VARIANTENENTSCHEID	14
5.2.1	HARDWARE	15
5.2.2	SOFTWARE	17
5.2.2.1	Webserver	17
5.2.2.2	Datenbank	18
5.2.3	ENTSCHEID	19
5.3	WIRTSCHAFTLICHKEIT	19
5.4	RISIKOANALYSE	20
5.5	ZEITABLAUF	21
5.6	ARBEITSJOURNAL	22

6 DESIGN23

6.1	DER ERSTE ENTWURF VOM WEBDESIGN	23
6.2	DIE EINZELNEN FUNKTIONEN	23
6.2.1	WEBDESIGN: DIE STARTSEITE	24
6.2.2	WEBDESIGN: DIE SUCHE	25
6.2.3	WEBDESIGN: ERFASSEN	25
6.2.4	WEBDESIGN: LÖSCHEN	26
6.2.5	WEBDESIGN: DRUCKEN	27
6.2.6	WEBDESIGN: LOGIN	28
6.2.7	WEBDESIGN: REGISTRIERUNG	28
6.3	KLASSENDIAGRAMM	29
6.4	AKTIVITÄTSDIAGRAMM	30
6.5	ÜBERSICHT RASPBERRY PI	34
6.6	MONGODB	34
6.7	APACHE TOMCAT	35
6.8	NO-IP	35
6.9	NETBEANS IDE	35

7 PROGRAMMIERUNG36

7.1	SYSTEMANFORDERUNGEN	36
7.2	KLASSEN UND METHODEN	36
7.2.1	MYUI	36
7.2.2	LOGINPAGE	36
7.2.3	AUTHENTICATION	36
7.2.4	DBCONNECTION	36
7.2.5	SEARCHTAB	36
7.2.6	ADDTAB	36
7.2.7	EDITTAB	37
7.2.8	PRINTTAB	37
7.2.9	MATERIALVERWALTUNDAO	37

7.2.10	ARTICLETODOCUMENT	37
7.2.11	DOCUMENTTOARTICLE	37
7.2.12	ARTICLE	37
8	<u>TESTING</u>	<u>38</u>
8.1	TESTFÄLLE	38
8.2	TESTPROTOKOLL	39
9	<u>PROJEKTABSCHLUSS</u>	<u>40</u>
9.1	SCHLUSSPRODUKT	40
9.2	ERFAHRUNGEN	40
9.2.1	UMGANG MIT DATEN DER MONGODB UND JAVA	40
9.2.2	UMGANG MIT GIT	40
9.2.3	UMGANG MIT DEM VAADIN FRAMEWORK	40
9.2.4	UMGANG MIT CSS	40
9.2.5	UMGANG MIT DYNDNS	40
9.3	REFLEXION	41
9.4	LESSONS LEARNED	42
9.4.1	ANALYSE UND DESIGN	42
9.4.2	RECHERCHEN EINPLANEN	42
9.4.3	VARIANTENENTSCHEID DURCHDENKEN	42
9.4.4	ZUSÄTZLICHE FUNKTIONEN	42
9.5	FAZIT	42
10	<u>ANHANG</u>	<u>43</u>
10.1	QUELLENVERZEICHNIS	43
10.2	ABBILDUNGSVERZEICHNIS	44
10.3	ARBEITSJOURNAL	45
10.4	PROJEKTANTRAG	46

1 Management summary

Die Webbasierte Materialverwaltung hilft im Tages Geschäft eines Sicherungsanlagen Techniker der SBB jederzeit abrufen zu können wo er benötigte Ersatzteile findet.

Auf den über 20 Standorten der Region Mitte der SBB befinden sich weit über 4000 Gegenstände an Lager für den täglichen Gebrauch im Störfall. Es ist beinahe unmöglich den Überblick zu behalten wo was ist. Jeder Techniker ist mit einem Smartphone, Tablet und Laptop ausgerüstet, eines der drei Geräte ist ihm immer zur Hand und einsatzbereit. Momentan existiert eine Excel-Liste die circa 30 Sekunden braucht um zu öffnen, somit unbrauchbar. Hier kommt unser Programm ins Spiel. Es soll eine Möglichkeit bieten benötigtes Material in möglichst kurzer Zeit auffindig zu machen, plattformunabhängig und rund um die Uhr.

Unsere Anwendung ist mobil einsetzbar und läuft kostengünstig auf einem Raspberry Pi mit installierter Datenbank und Webserver. Sie ist nicht für den kommerziellen Gebrauch gedacht, somit sollten keine Performanceprobleme auftreten mit der gewählten Hardware. In einem Team sind maximal 10 Techniker die vereinzelt darauf zugreifen müssen. Es konnten alle Anforderungen für den Betrieb umgesetzt werden. Für zusätzliche Funktionen muss ein weiteres Projekt in Auftrag gegeben werden. Das Projekt war somit erfolgreich im Rahmen der Aufgabenstellung.

2 Einleitung

Wir erhalten jedes Semester von der schweizerischen Fachschule TEKO einen Projektauftrag, damit wir uns mit den Themen Projektmanagement und objektorientierte Programmierung auseinandersetzen. Es geht darum das Wissen in diesen Bereichen zu steigern und uns stetig weiterzuentwickeln. Die Projektarbeit wird in zweier Gruppen durchgeführt. Als Team fördern wir das Zusammenarbeiten und geben unser Wissen bestmöglich weiter.

2.1 Vorwort

Die ersten Grundlagen der Programmierung wie auch die Grundlagen mit Datenbanken durften wir bereits an der Schule erlernen. Es geht nun darum mit den Grundlagen ein erstes selbständiges Projekt zu realisieren und das erlernte umzusetzen.

Wir Programmieren als Projektarbeit für das vierte Semester eine Webapplikation mit Datenbankanbindung, um die bestehende Excel Materialliste der SBB abzulösen.

2.2 Projektteam



2: Thomas Baechler



3: Christoph Grossmann

Name	Baechler	Grossmann
Vorname	Thomas	Christoph
Geburtsdatum	29.12.1980	19.07.1989
Arbeitgeber	InfoSoft Systems GmbH	SBB
Beruf	System- und Netzwerktechniker	Techniker für Sicherungsanlagen Innen
Funktion	Entwicklung, Dokumentation	Projektleitung, Entwicklung, Dokumentation
Kommunikation	Telefon, WhatsApp, E-Mail und Face-to-Face	
Entwicklungstools	Netbeans via GIT	

2.3 Begrifflichkeiten

Begriffe	Bedeutung
TEKO	Schweizerische Fachschule
OOA	Objektorientierte Analyse
OOD	Objektorientiertes Design
OOP	Objektorientiertes Programmieren
SBB	Schweizerische Bundesbahn
Open Source	Öffentlicher Quelltext der Anwendung, kostenlos nutzbar
VCS	Version Control System
GIT	Git Open Source Software für das VCS
SQL	Datenbanksprache für eine relationale Datenbank
MySQL	SQL Datenbanksystem von Oracle, Open Source / Kommerziell
MongoDB	Domukentorientierte NoSQL Datenbank, Open Source
NoSQL	Nicht relationale Datenbank
NetBeans	Programmieroberfläche, stellt Frameworks und Libraries zur Verfügung
Apache Tomcat	Open Source Webserver
IIS	Microsoft Webserver
Raspberry Pi	Einplatinencomputer
DynDNS	Dynamische DNS, synchronisieren einer öffentlichen IP Adresse
NoIP	Kostenloser Anbieter von DynDNS
DB	Datenbank
UI	Benutzeroberfläche
CSS	Cascading Style Sheets, Gestaltungsanweisungen für Webinhalte
Java	Objektorientierte Programmiersprache von Sun (Oracle)

2.4 Vorgehen

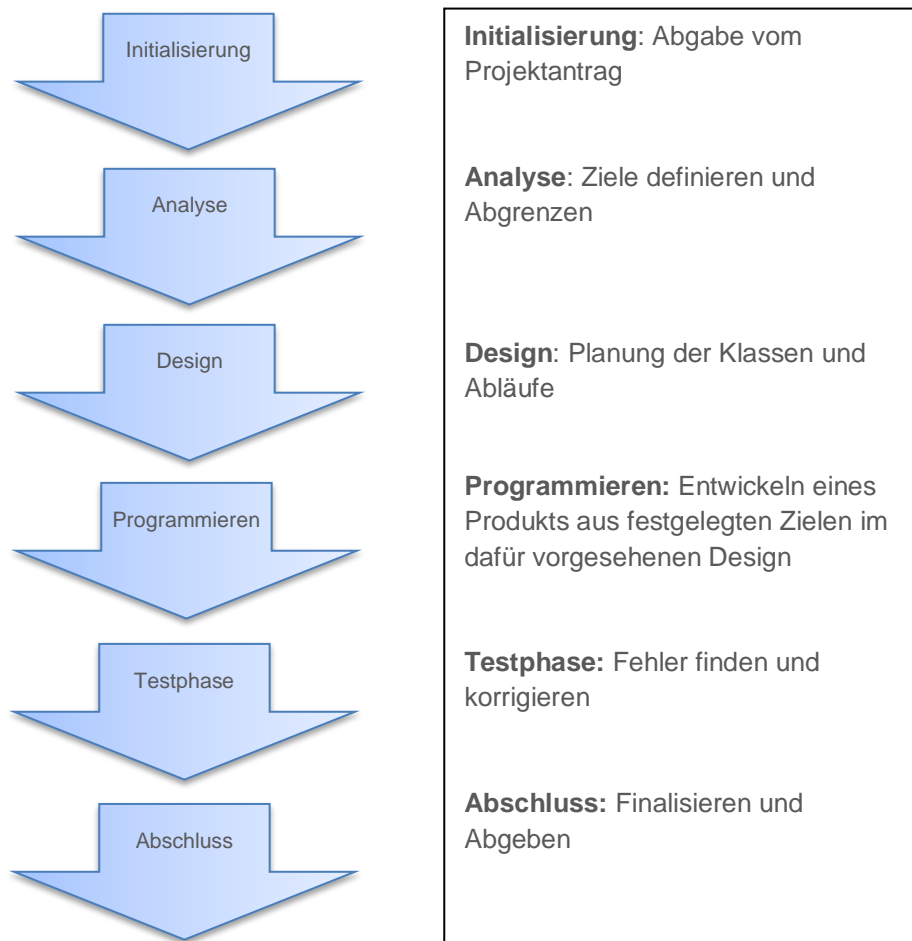
Wir haben uns dazu entschieden, dass wir uns regelmässig nach der Schule und nach der Arbeit treffen und gemeinsam am Projekt arbeiten. Dadurch können wir jeden weiteren Schritt besprechen, Probleme effizient beseitigen und unser Wissen jederzeit weitergeben.

Nach jedem treffen legen wir fest, wann wir uns das nächste Mal treffen und an welcher Stelle wir weiterarbeiten. Wodurch wir uns optimal vorbereiten können.

Als Treffpunkt haben wir uns für den Arbeitsort von Thomas in Emmen festgelegt. Der Arbeitsort hat eine gute Autobahnanbindung und wir können uns jeweils nach der Arbeit in einem klimatisierten Sitzungszimmer an die Arbeit machen. Im Sitzungszimmer haben wir WLAN und auch einen Flip Chart um unsere Ideen aufzuzeichnen.

2.5 Projektstrukturplan

Der Projektstrukturplan zeigt das grobe Vorgehen unserer Semesterarbeit auf. Er unterstützt uns beim Umsetzen der einzelnen Phasen mit dem Zeitplan.



2.6 Resultate

Die von uns programmierte Webbasierte Applikation für Reservematerial dient dazu, Material in einer Datenbank abzulegen und mittels Webinterface anzuzeigen.

2.7 Aussicht

Die Webapplikation ist nicht für den Verkauf gedacht. Ob die Software wirklich produktiv geht ist noch nicht entschieden. In erster Linie dient die Applikation das in der Schule erlernte Wissen weiterentwickeln.

3 Projektauftrag

Die aufgeführten Richtlinien wurden von der TEKO unter Anweisung von Bruno Hammer gesetzt. Die Arbeit geht über die Dokumentation, das Software Engineering, das Engineering von Datenbanken und die Kenntnisse mit Raspberry Pi.

3.1 Rahmenbedingungen

Umfang

- Das Projekt wird als zweier Gruppe realisiert
- Die Richtzeit beträgt 60-100 Stunden pro Person, also insgesamt maximal 200 Stunden

Umfeld

- Es werden nur Freeware Programme und öffentliche API's verwendet
- Für das Projekt beschaffen sich die Teilnehmer ein Raspberry Pi 3 Modell B+

Plattform / IDE

- GIT (<https://elad.ch/gitblit/>)
- Netbeans IDE 8.2
- MongoDB 3.6
- Apache Tomcat 8
- Raspberry Pi 3 Model B+

Programmiersprache

- Java mit Vaadin Framework

Präsentation

- Maximal 13 Minuten am Freitag, 28. September 2017, 18:30-20:00 Uhr

Projektsitzungen

- Teamsitzung 2-3-mal pro Monat
- Meilensteinsitzungen insgesamt 2

Termine

- Projektantrag: Mittwoch, 2. Mai 2018
- Entscheid Projektantrag: Freitag, 4. Mai 2018
- Projektstart: Freitag, 4. Mai 2018
- Abgabe Dokumentation: Samstag, 22. September 2018, 12.00 Uhr
- Präsentation: Freitag, 28. September 2017, 18:30-20:00 Uhr

Betreuung

- Fachlehrer Bruno Hammer (hammer@ext.teko.ch), Schweizerische Fachschule TEKO, Luzern

3.2 Umfeld

Das Projekt wird in einem kleinen autonomen Netzwerk umgesetzt, ist mobil und somit überall einsetzbar.

4 Pflichtenheft

4.1 Zielbestimmungen

4.1.1 MUSS-Kriterien

- Die Zugänglichkeit über Mobilien Geräten wie: Smartphone und Tablet muss gewährleistet sein.
- Zugriff auf die Webapplikation innert 15 Sekunden über das Mobile Netzwerk.
- Unterschiedliche Zugriffsrechte: Anzeigen der Liste oder die Möglichkeit die Einträge zu Bearbeiten
- 24h-Verfügbarkeit
- Artikel Suchen nach: Bezeichnung, Station, Element, ...

4.1.2 WUNSCH-Kriterien

- Benutzer können einen Artikel ein-/ausbuchen.
- Eine Erinnerung kann erstellt werden, wenn ein Artikel zur Reparatur eingesendet werden kann.

4.1.3 Abgrenzung

- Keine aufwendigen Animationen die die Performance beeinträchtigen.
- Nicht mehr als 10 Benutzer sollen gleichzeitig, ohne Performance-Einbussen, die Anwendung benutzen können.
- Die Anwendung ist nicht für den Kommerziellen gebraucht gedacht.

4.2 Einsatz

Die Anwendung ist für den internen Einsatz des Sicherungsanlagen Team der SBB gedacht. Die Anwender kennen das Umfeld und die Artikel die sie suchen. Oft kommt es vor das in einem Pikettfall mitten in der Nacht ein Ersatzteil benötigt wird. Mit Hilfe der Anwendung soll es ermöglicht werden, schnell und von überall her Einsicht in die verschiedenen Reservestandorte zu erlangen. So können Anfahrts- und Störungszeiten enorm verkürzt werden. Zugriffen wird via Smartphone, Tablet und Laptop über SIM Karte.

4.3 Funktionen

4.3.1 Sortieren

Die Artikelliste kann in verschiedenen Spalten alphabetisch geordnet werden. Durch Druck auf die Titelspalte.

4.3.2 Suchen

Beim Zugriff kann jeder ohne sich anzumelden die Artikeldatenbank durchsuchen. Gesucht werden kann nach Bezeichnung, Standort, Typ und Art

4.3.3 Bearbeiten

Der Administrator kann Artikel mit Klick auf den "Bearbeiten" Button bearbeitet werden.

4.3.4 Hinzufügen

Der Administrator kann Artikel im "Hinzufügen" Tab neu erfassen.

4.3.5 Löschen

Der Administrator kann Artikel mit Klick auf den "Löschen" Button aus der Datenbank entfernen.

4.3.6 Ein-/Ausbuchen (Wunsch)

Angemeldete Benutzer können Artikel aus- oder einbuchen. Um die Aktuelle Menge des an Lager befindenden Artikels aktuell zu halten.

4.4 Benutzeroberfläche (Layout, Zugriffsrechte)

4.4.1 Admin:

- Hinzufügen
- Löschen
- Bearbeiten
- Drucken

4.4.2 Benutzer:

- Suchen
- Drucken
- (Ein-/Ausbuchen)

4.5 Technisches Umfeld

Die Website läuft auf einem Webserver auf den immer zugegriffen werden kann. Mit Hilfe der von der SBB zur Verfügung gestellten Arbeitsmittel muss die Anwendung zu bedienen sein.

Laptop: Lenovo 13"

Tablet: iPad Air2

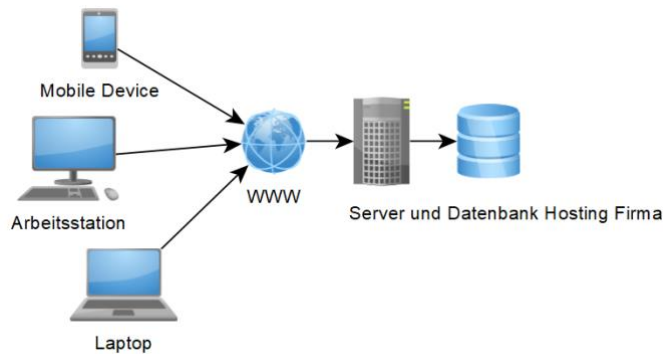
Smartphone: Samsung Galaxy S5, iPhone

5 Analyse

5.1 Lösungsvarianten

5.1.1 Hardware

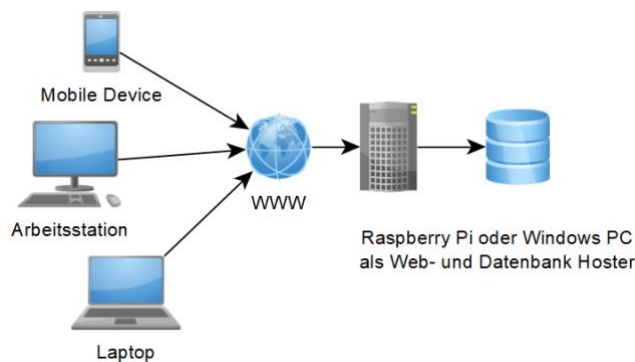
5.1.1.1 Variante Hosting bei einem Hosting Anbieter



4: Lösungsvariante Server wird Fremdgehostet

Die Server werden bei einem Webhoster fremdgehostet. Dadurch ist der Server überall erreichbar und die Datensicherung wird vom Hoster sichergestellt. Ebenfalls sind die Systeme je nach Anbieter redundant abgesichert. Die Systeme haben eine Garantierte Verfügbarkeit. Bei Problemen muss jedoch der Hoster kontaktiert werden und je nach Problem muss gewartet werden, bis der Hoster das Problem behoben hat. Der persönliche Lerneffekt ist kleiner, da die Software meistens als Paket zur Verfügung steht und nur die Konfiguration gemacht werden muss. Je nach Anbieter gibt es nur eine beschränkte Auswahl der Datenbank- und Webserversoftware.

5.1.1.2 Variante System auf eigener Umgebung aufbauen



5: Lösungsvariante Eigenes Hosting mit Raspberry Pi

Die Webseite und die Datenbank werden auf einer eigenen Umgebung wie zum Beispiel Raspberry Pi oder Windows Client installiert. Wir hosten die Umgebung selber an einem eigenen Internetanschluss oder beim Arbeitgeber. Für den Ausfall und die Sicherheit sind wir selber verantwortlich. Bei Problemen muss das Problem selber analysiert und behoben werden. Wir erarbeiten uns das benötigte Wissen wie ein Webserver und ein Datenbankserver aufgesetzt und betrieben wird. Die Softwareauswahl ist unbeschränkt. Der Lerneffekt ist grösser, da wir uns mit der Hardware wie auch Software intensiver auseinandersetzen.

5.1.2 Software

5.1.2.1 Variante Microsoft: Windows Maschine mit IIS und SQL Datenbank

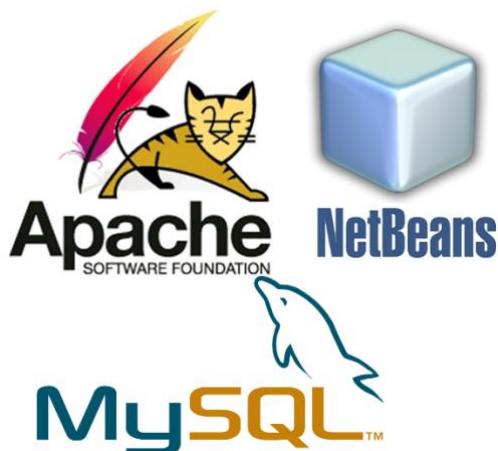


6: Lösungsvariante Software von Microsoft

Wir installieren oder lassen uns ein Microsoft Windows Betriebssystem hosten. Auf dem System wird der Webserver IIS (Internet Information Services) eingerichtet und installieren die kostenlose Datenbank Software Microsoft SQL Express. Das Betriebssystem ist weltweit sehr verbreitet. Der Webserver ist im Betriebssystem integriert, was für eine hohe Stabilität spricht. Der Microsoft SQL Server ist ein relationales Datenbankmanagementsystem, welcher für Windows Betriebssysteme konzipiert wurde. Der MS SQL Server kann nicht auf dem Raspberry betrieben werden.

Für das Betriebssystem benötigen wir einen Computer. Natürlich kann auch auf den Raspberry Pi ein Windows 10 installiert werden, dabei handelt es sich aber nur um eine «Core» (Windows-10-Kern) Variante, also ohne grafische Oberfläche. Das Operating System ist vor allem für die Konfiguration und Steuerung aus der Ferne gedacht. Aus diesem Grund würden wir uns bei dieser Variante für einen PC entscheiden. Microsoft IIS und Microsoft SQL haben wir bisher im Unterricht nicht eingesetzt, somit müssen wir uns das Wissen zuerst beschaffen.

5.1.2.2 Variante: Apache und MySQL



7: Lösungsvariante Apache und MySQL

Wir installieren oder lassen uns die weltweit verbreitetsten Open-Source Webapplikationen Apache und MySQL hosten. MySQL ist ein relationales Datenbankverwaltungssystem, welches für verschiedenste Betriebssysteme verfügbar ist und für die Datenspeicherung von Webservices dient. Apache ist ein frei verfügbarer Webserver, welcher auf so ziemlich jedem Betriebssystem installiert werden kann (Linux, Solaris, Digital UNIX, MacOS und Windows).

Diese Variante wird bei so ziemlich allen Hoster angeboten. Es ist wie bereits geschrieben eines der weltweit verbreitetsten Varianten. Im Unterricht hatten wir bisher den Apache Tomcat, jedoch kein MySQL. Das Wissen für MySQL müssten wir uns komplett selber erarbeiten.

5.1.2.3 Variante «TEKO» OpenSource: Raspbian, Apache Tomcat und Mongo DB



8: Lösungsvariante Apache mit mongoDB

Wir installieren oder lassen uns die im Fach Datenbankdesign, SQL Teil II kennengelernte Variante Apache Tomcat mit MongoDB hosten. Im Gegensatz zu den anderen beiden Softwarevarianten, handelt es sich bei MongoDB um eine dokumentenorientierte NoSQL-Datenbank. Da die Datenbank dokumentenorientiert ist, kann sie Sammlungen von Dokumenten verwalten. Die Daten werden zwar in komplexen Hierarchien verschachtelt, bleiben aber immer abfragbar und indizierbar. Als Webserver kommt Apache zusammen mit Tomcat als Java-Umgebung zum Einsatz. Der Tomcat wird zum Erstellen von Java serverseitig Inhalten verwendet.

Wie bereits geschrieben, konnten wir im Unterricht schon erste Erfahrungen mit dieser Kombination sammeln. Das Wissen kann mit dieser Variante weiter vertieft werden.

5.2 Variantenentscheid

Wir haben sehr lange Diskutiert und auch andere Personen gefragt, welchen Weg wir einschlagen sollen. Alle Vor- und Nachteile mehrfach durchgegangen und uns als erstes für die Plattform entschieden. Weder der Software noch der Hardwareentscheid konnte am Anfang gefällt werden. Wir wussten das es ein Kombinationsentscheid werden muss.

5.2.1 Hardware



9: Gewichtung der Variante Hardware

Die Frage bei der Hardware stellte sich vor allem, wollen wir die Webapplikation Fremd hosten oder ein System (Raspberry Pi oder Windows System) selber aufsetzen?

Fremd Hosting:

Vorteile

- Garantierte Verfügbarkeit
- Ausfallsicherheit
- Erreichbarkeit
- Zeitaufwand

Nachteile

- Abhängig vom Hoster
- Initialkosten und laufende Kosten
- Persönlicher Lerneffekt
- Suche nach einem Hoster der die gewünschte Software Variante anbietet
- Nach dem Entscheid ist ein Wechsel Aufwändig (Kosten und Zeit)

Eigenes Hosting:

Vorteile

- Persönlicher Lerneffekt
- Flexibel: Jede Variante der Software realisierbar
- Die Software kann auch jederzeit gewechselt werden
- Keine Abhängigkeit
- Geringere Kosten: Eigener Internetanschluss bereits vorhanden, Hardwarebeschaffung gering

Nachteile

- Zeitaufwand: Installation, Problembehebung und einrichten
- Verfügbarkeit schwer Abschätzbar

Eigenes Hosting mit Raspberry Pi:

Wir setzen auf eine bereits bekannte sehr flexible Hardware und nutzen dabei das bereits erlernte, erweitern dieses mit der Installation von einer Datenbank und Webservice. Geringe Kosten und Stromverbrauch. Variante Windows fällt jedoch weg.

Eigenes Hosting mit Windows PC:

Wir beschaffen uns einen Windows PC (neu oder Occasion), installieren darauf einen Webserver und einen Datenbankserver. Wir legen uns als Betriebssystem auf Microsoft Windows fest und müssen mit allen Vor- und Nachteilen leben. Alle Softwarevarianten sind möglich. Stromverbrauch höher als beim Raspberry und falls wir kein Occasionsgerät finden, sind die Anschaffungskosten höher als beim Raspberry.

Der Entscheid fiel am Ende knapp auf ein eigenes Hosting mit Raspberry Pi. Der Lerneffekt ist am grössten und das neu erlernte werden wir höchstwahrscheinlich im Verlaufe der Ausbildung weiterverwenden können. Bis auf die Software Variante Windows kann mit dem Raspberry alles umgesetzt werden. Die TEKÖ setzt ebenfalls auf Open Source, somit gehen wir denselben Weg und können für Probleme oder Rückfragen auch direkt mit den Lehrern kommunizieren.

5.2.2 Software

Beim Softwareentscheid war die Hauptschwierigkeit der Datenbankserver. Wir haben uns zum Zeitpunkt des Variantenentscheids im Unterricht in erster Linie mit relationalen Datenbanken auseinandergesetzt. Der Begriff MongoDB wurde erläutert, jedoch konnten wir uns noch nicht vorstellen wie damit gearbeitet wird.

5.2.2.1 Webserver



10: Gewichtung der Variante Webserver

IIS:

Vorteile

- Integriert in Windows
- Einfach Installation
- Kostenlos

Nachteile

- Keine persönlichen Erfahrungen
- Muss auf einem Windows System installiert werden
- Persönlicher Lerneffekt in Bezug auf die Schule fraglich

Apache Tomcat:

Vorteile

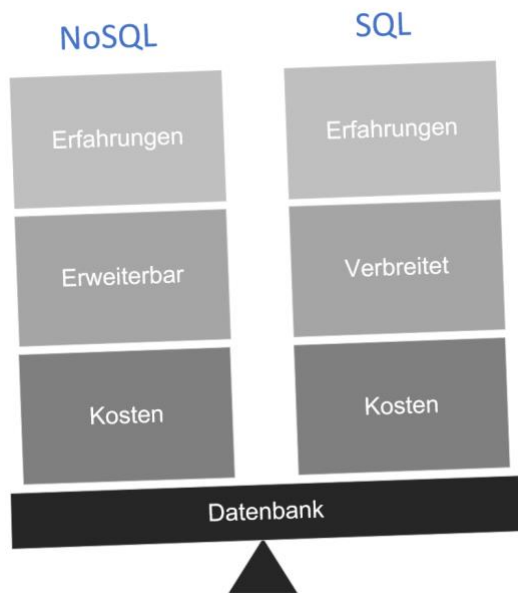
- Flexibel: Kann auf ziemlich jeder Plattform installiert werden
- Bereits erste Projekte und Erfahrungen an der Schule gesammelt
- Weit verbreitet: Diverse Anleitungen auch mit Raspberry Pi vorhanden
- Kostenlos

Nachteile

- Keine

Der Entscheid beim Webserver fiel aufgrund von keinen nennenswerten Nachteilen auf den Apache Tomcat. Ein weiteres wichtiges Kriterium waren die ersten Erfahrungen während dem Unterricht. Wir wussten das die Semesterarbeit sicher mit Apache Tomcat umgesetzt werden kann.

5.2.2.2 Datenbank



11: Gewichtung der Variante Datenbank

Relationale Datenbank:

Vorteile

- Bereits erste Projekte und Erfahrungen an der Schule gesammelt
- Viele Programme laufen auf einer relationalen Datenbank
- Kostenlos

Nachteile

- Erweitern ist nicht ganz einfach

NoSQL Datenbank:

Vorteile

- Erfahrungen mit Projekten in der Schule gesammelt (Juli 2018)
- Erweitern ist nicht ganz einfach
- Kostenlos

Nachteile

- Noch keine Erfahrungen gesammelt

Die erste Entscheidung fiel auf die relationale Datenbank. Wir hatten bereits erste Projekte erfolgreich umgesetzt und dachten am Anfang auch, dies ist die richtige Variante. Wir erstellten das Datenbankmodell und machten uns intensiv Gedanken wie wir die vorhandene Excel Tabelle sinnvoll abbilden können. Leider machte das Ganze aber keinen wirklichen Sinn. Wir sprachen uns nochmals mit Rainer Strauss unserem Lehrer im Fach Datenbank und mussten unseren ersten Entscheid verwerfen. Der neue Entscheid war somit eine NoSQL Datenbank mit MongoDB.

5.2.3 Entscheid

Schlussendlich haben wir uns für eine eigene Umgebung mit Raspberry Pi entschieden. Wir beschaffen uns je ein Raspberry und installieren darauf die von uns festgelegte Software: Apache Tomcat und Mongo DB. Der Lerneffekt ist somit viel grösser und wir können selber entscheiden, ob die Software auch wirklich dafür geeignet ist oder bei Bedarf gewechselt wird. Die Aufwendungen bestehen einfach aus den Anschaffungskosten vom Raspberry Pi 3. Die Software ist Open Source, also kostenlos. Laufende Kosten entstehen vorerst keine, da wir für den externen Zugriff einen DynDNS von <https://www.noip.com/> einsetzen. Die kostenlose Version muss einfach einmal monatlich bestätigt werden. Der Raspberry Pi wird am Privaten Internetanschluss betrieben und ist über DynDNS von überall erreichbar.

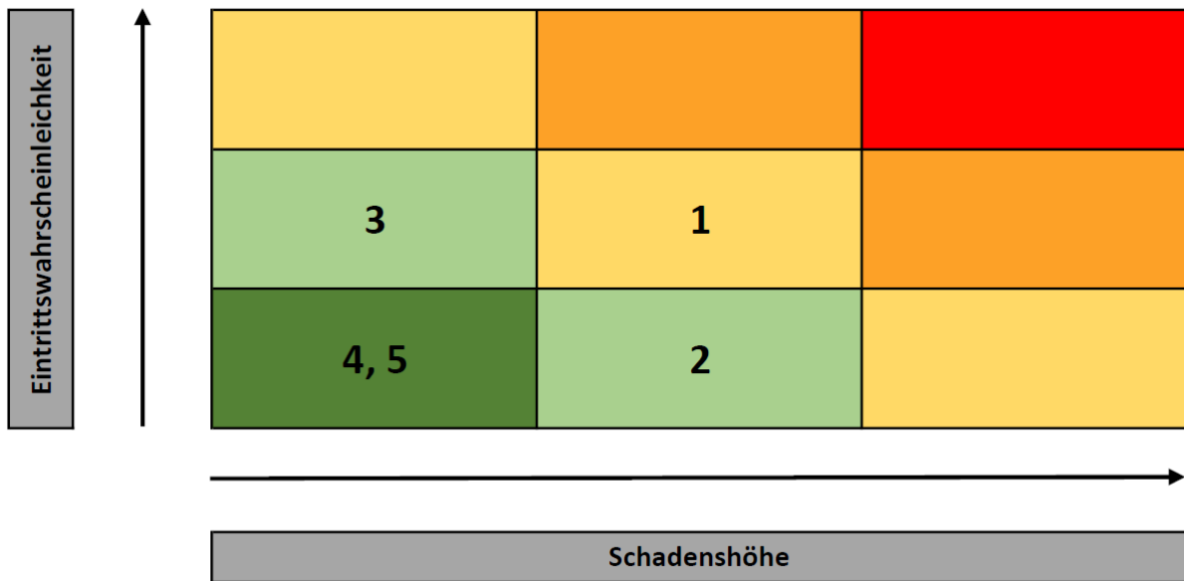
5.3 Wirtschaftlichkeit

Wir beschaffen uns jeweils einen neuen Raspberry Pi 3 für 70 CHF. Diese Kosten tragen wir selber. Damit die Webseite auch erreichbar ist, nutzen wir den kostenlosen DynDNS Dienst von NoIP. Der Raspberry wird im Privaten Netzwerk am eigenen Privaten Internetanschluss eingebunden. Somit entstehen auch keine zusätzlichen Kosten für den Betrieb der Webseite. Der Raspberry kann somit an jedem x-beliebigen Internetanschluss betrieben werden.

Unsere Semesterarbeit ist zum aktuellen Zeitpunkt weder zum Verkauf, noch für den produktiven Einsatz bei uns Zuhause vorgesehen. Aus diesem Grund entstehen bis auf die Anschaffungskosten keine weiteren Unterhaltskosten.

5.4 Risikoanalyse

Mit der Risikoanalyse wird gezeigt welche Risiken bestehen und welche Gewichtung sie haben. Zugleich wird zu jedem Task (Risiko) eine Massnahme erstellt.

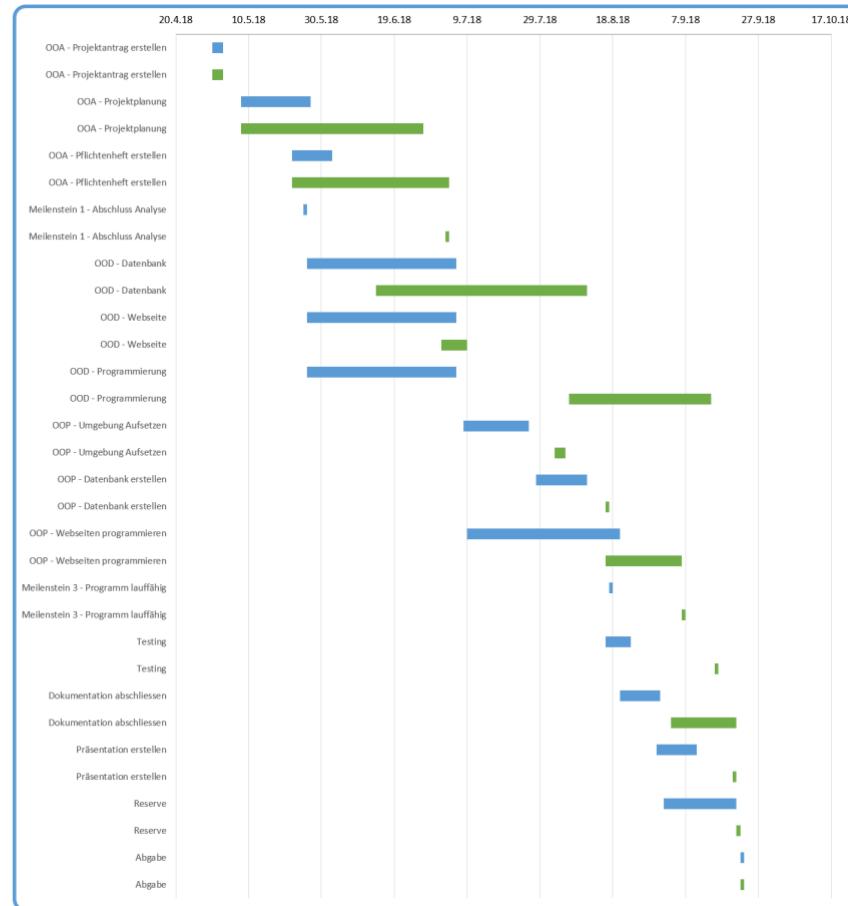


12: Risikoanalyse

Nr.	Risiko	Lösung	Einstufung (Eintritt – Schaden)
1	Zeitmanagement	Planung überdenken evtl. anpassen	mittel - mittel
2	Person fällt aus	Die Projektarbeit kann trotzdem beendet werden, da unser Team aus 2 Personen besteht	niedrig - mittel
3	Hardware defekt (Notebook, Raspberry)	Backup erstellen (GIT)	mittel - niedrig
4	Betreuer fällt aus	Stellvertreter übernimmt	niedrig - niedrig
5	Diverses (Weltuntergang)	Diesem Risiko wird keine Beachtung geschenkt	niedrig - niedrig

5.5 Zeitablauf

Aufgabe	Start Datum	End Datum	Dauer (in Tagen)
OOA - Projektantrag erstellen	30.04.2018	03.05.2018	3
OOA - Projektantrag erstellen	30.04.2018	03.05.2018	3
OOA - Projektplanung	08.05.2018	27.05.2018	19
OOA - Projektplanung	08.05.2018	27.06.2018	50
OOA - Pflichtenheft erstellen	22.05.2018	02.06.2018	11
OOA - Pflichtenheft erstellen	22.05.2018	04.07.2018	43
Meilenstein 1 - Abschluss Analyse	25.05.2018	26.05.2018	1
Meilenstein 1 - Abschluss Analyse	03.07.2018	04.07.2018	1
OOD - Datenbank	26.05.2018	06.07.2018	41
OOD - Datenbank	14.06.2018	11.08.2018	58
OOD - Webseite	26.05.2018	06.07.2018	41
OOD - Webseite	02.07.2018	09.07.2018	7
OOD - Programmierung	26.05.2018	06.07.2018	41
OOD - Programmierung	06.08.2018	14.09.2018	39
OOP - Umgebung Aufsetzen	08.07.2018	26.07.2018	18
OOP - Umgebung Aufsetzen	02.08.2018	05.08.2018	3
OOP - Datenbank erstellen	28.07.2018	11.08.2018	14
OOP - Datenbank erstellen	16.08.2018	17.08.2018	1
OOP - Webseiten programmieren	09.07.2018	20.08.2018	42
OOP - Webseiten programmieren	16.08.2018	06.09.2018	21
Meilenstein 3 - Programm lauffähig	17.08.2018	18.08.2018	1
Meilenstein 3 - Programm lauffähig	06.09.2018	07.09.2018	1
Testing	16.08.2018	23.08.2018	7
Testing	15.09.2018	16.09.2018	1
Dokumentation abschliessen	20.08.2018	31.08.2018	11
Dokumentation abschliessen	03.09.2018	21.09.2018	18
Präsentation erstellen	30.08.2018	10.09.2018	11
Präsentation erstellen	20.09.2018	21.09.2018	1
Reserve	01.09.2018	21.09.2018	20
Reserve	21.09.2018	22.09.2018	1
Abgabe	22.09.2018	23.09.2018	1
Abgabe	22.09.2018	23.09.2018	1



13: Zeitablauf

Zusatzinformation:
 Blauer Balken = Soll
 Grüner Balken = IST

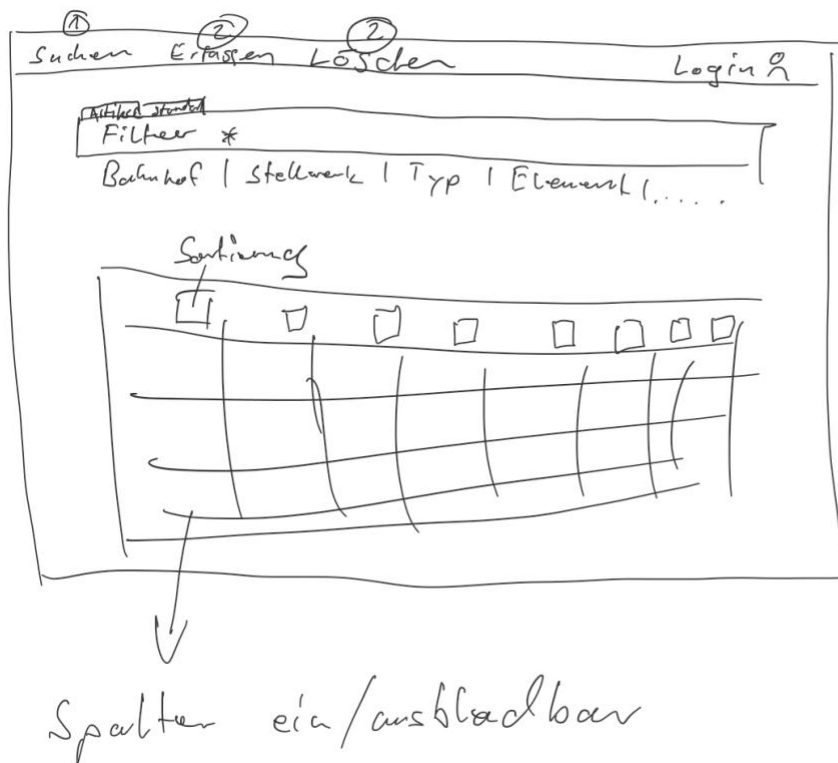
5.6 Arbeitsjournal

Wir schreiben parallel zur Umsetzung der Semesterarbeit ein Arbeitsjournal. Im Journal ist ersichtlich welche Person, zu welcher Zeit, was umgesetzt hat. Das Arbeitsjournal kann im Anhang gefunden werden.

6 Design

6.1 Der erste Entwurf vom Webdesign

Für unser erstes Webdesign haben wir verschiedenste Webseiten im Internet und Intranet angeschaut und uns die in Frage kommenden Webseiten rausgesucht. Anhand der Auswahl, haben wir uns dann eine Skizze gemacht, welche Steuerelemente wo platziert werden und wo der Inhalt dargestellt wird. Wichtig war uns, die Webapplikation muss einfach zu bedienen und auch die Anordnung einem gewissen Standard entsprechen. Wir haben uns darauf konzentriert, nur so wenig wie möglich und so viel wie nötig beim Design einzubeziehen.



* - Over design
- Test

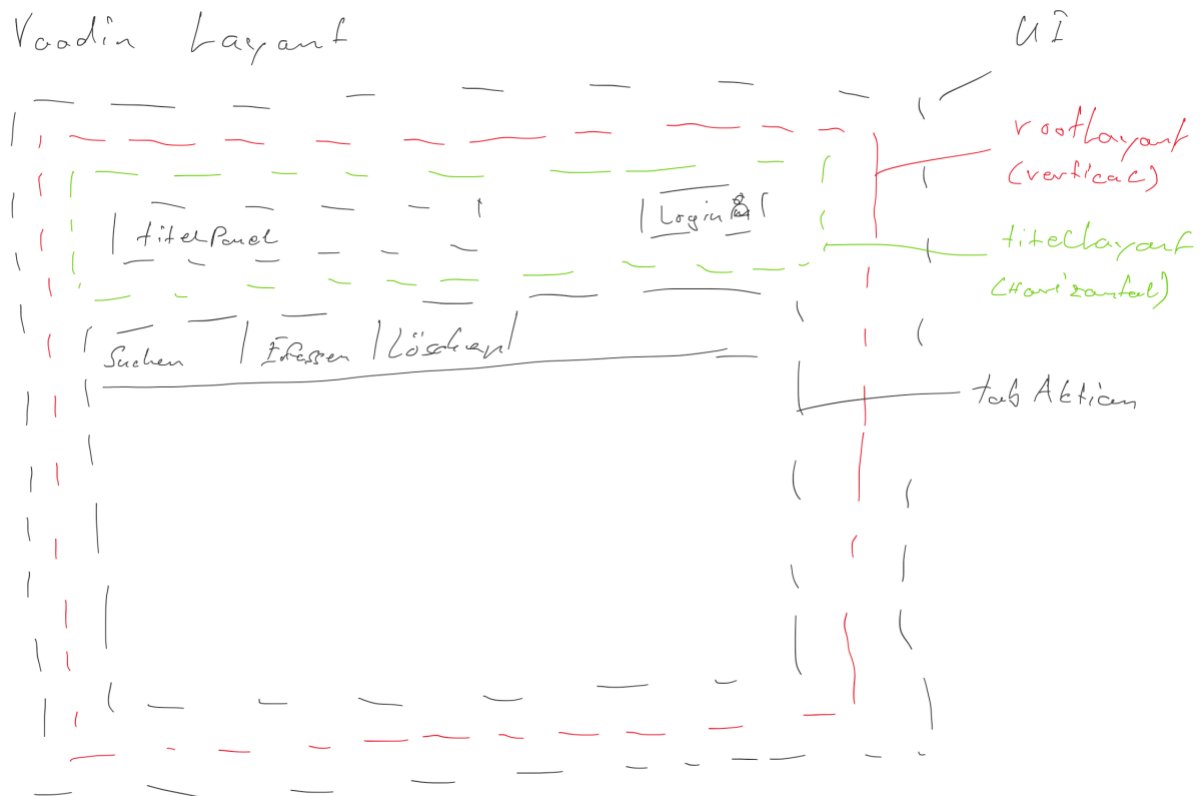
14: erster Entwurf Weblayout

6.2 Die einzelnen Funktionen

Nachdem wir den ersten Entwurf skizziert haben, haben wir uns mit den einzelnen Seiten und deren Funktionen befasst. Wie muss die Webseite aufgebaut werden und wie werden die einzelnen Frames angeordnet. Anhand der weiteren Entwürfe für die einzelnen Funktionen (Suchen, Löschen, etc.) kann unser Gedankenfluss nachvollzogen werden.

6.2.1 Webdesign: Die Startseite

Beim Öffnen der Webapplikation gelangt man erstes auf die Startseite unseres User Interface kurz UI. Die verschiedenen Layout Ebenen werden anhand der unten aufgeführten Grafik erstellt. Die Erstellung erfolgt von aussen nach innen. Zuerst das Allgemein UI, dann das root Layout und darin dann das titel Layout wie auch das tab Aktion. Je nach Berechtigung sind die einzelnen Steuerelemente im tabAktion vorhanden oder nicht.

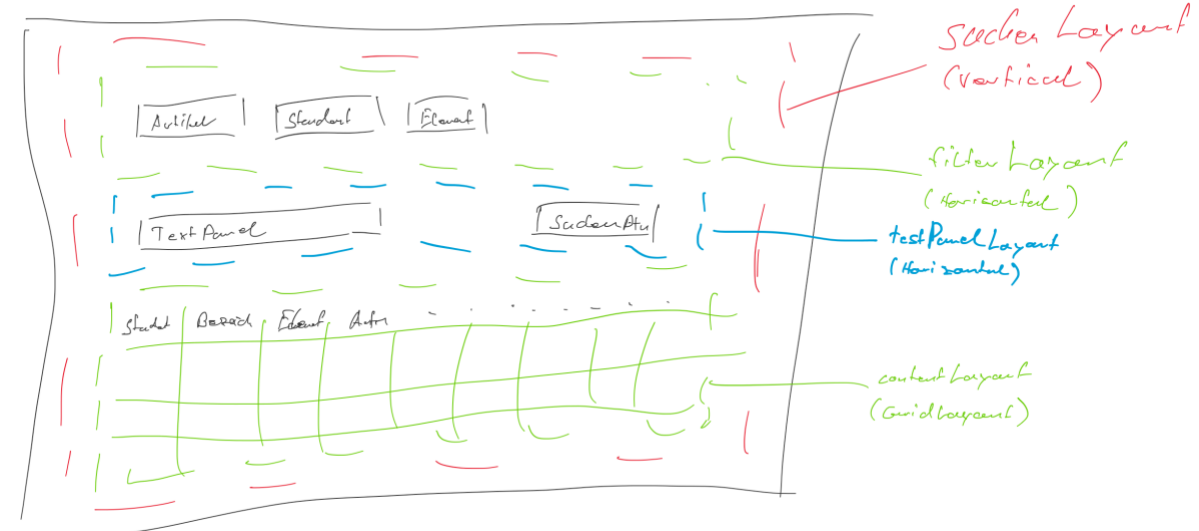


15: Design Layout Skizze: Startseite

6.2.2 Webdesign: Die Suche

Wurde auf der Startseite das Steuerelement Suchen ausgewählt, erscheint nun im tab Aktion eine Suchmaske. Die verschiedenen Layer beinhalten weitere Steuerelemente wie auch Darstellungselemente wie z.B. das contentLayout also den Suchinhalt. Ziel der Suche ist es, den gewünschten Artikel schnell und einfach zu finden.

Suchen Tab

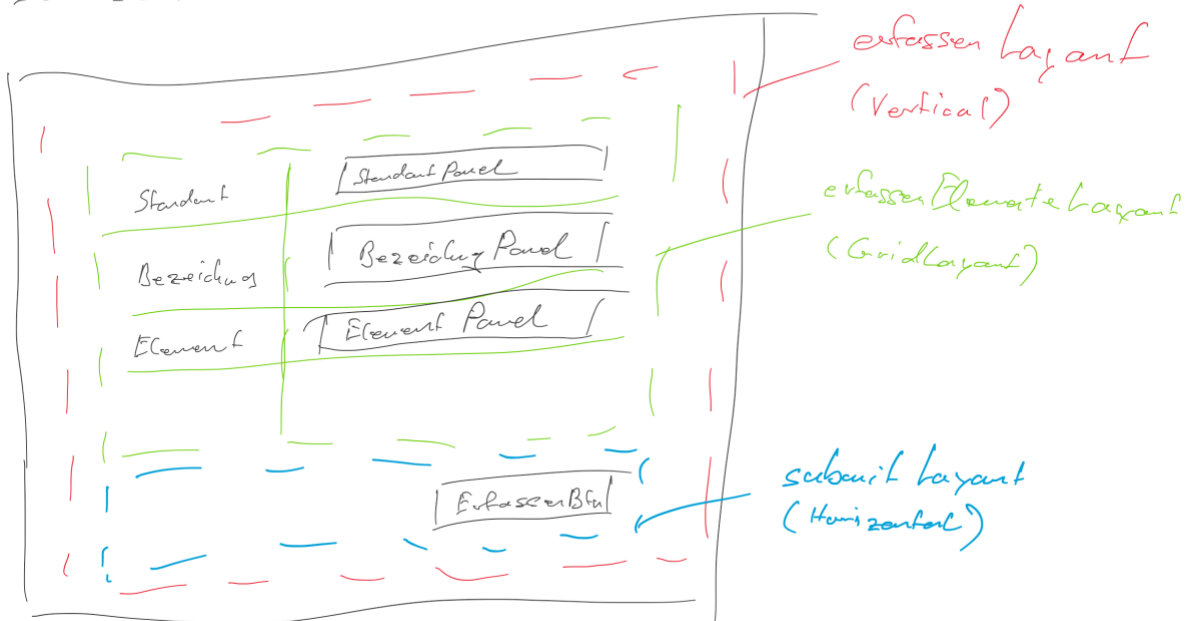


16: Design Layout Skizze: Suchen

6.2.3 Webdesign: Erfassen

Wurde auf der Startseite das Steuerelement Erfassen ausgewählt, erscheint nun im tab Aktion eine Maske um Artikel zu erfassen. Beim Tab Erfassen, können neue Artikel schnell und einfach erfasst werden. Auf der unten aufgeführten Skizze, können die einzelnen Layer und die Anordnung der einzelnen Elemente für den Aufbau der Seite nachvollzogen werden.

Erfassen Tab

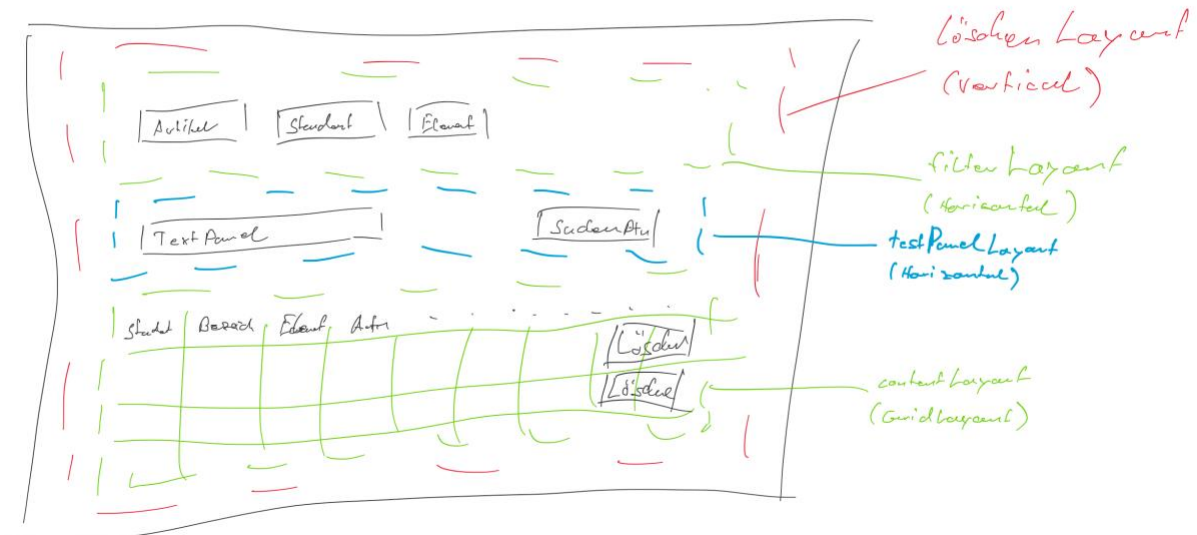


17: Design Layout Skizze: Erfassen

6.2.4 Webdesign: Löschen

Wurde auf der Startseite das Steuerelement Löschen ausgewählt, erscheint nun im tab Aktion eine Maske um Artikel zu entfernen. Beim Tab Löschen, können nicht mehr benötigte oder nicht mehr vorhandene Artikel aus der Datenbank entfernt werden. Wie auch bei den anderen Funktionen haben wir eine Skizze erstellt, damit unsere Gedankengänge zum Layout nachvollzogen werden können. Ziel ist es, dass der gewünschte Artikel schnell und zuverlässig aus der Datenbank entfernt wird.

Löschen Tab

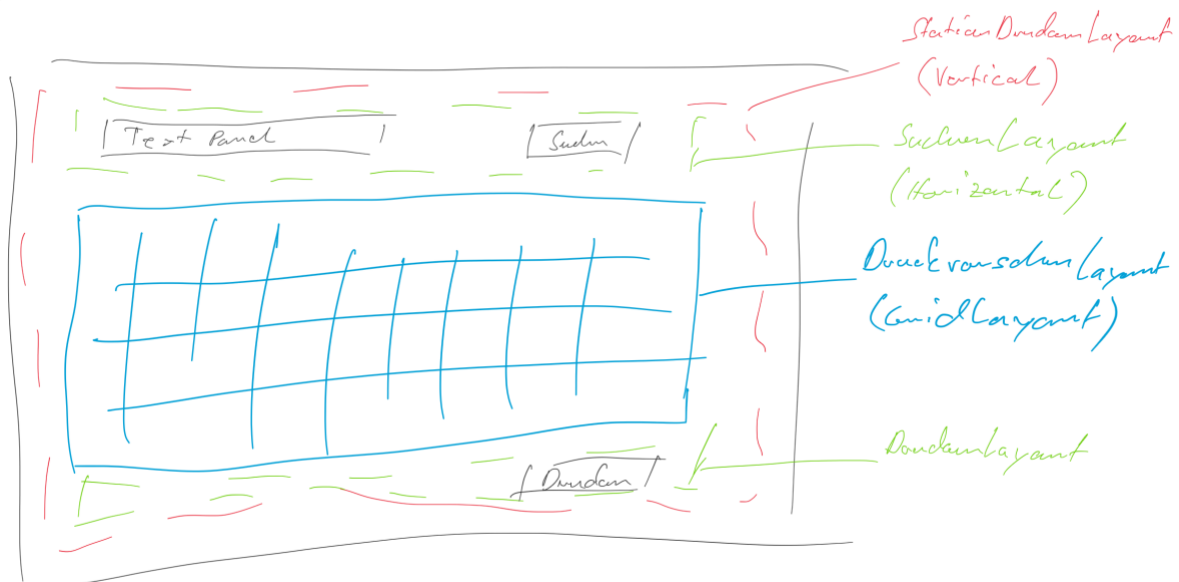


18: Design Layout Skizze: Löschen

6.2.5 Webdesign: Drucken

Was natürlich nicht fehlen darf, ist die Möglichkeit Artikel von einem Standort auszudrucken. An jedem Standort hat es eine Liste in Papierform, welche den Mitarbeitenden den aktuellen Materialbestand vom entsprechenden Standort anzeigt. Dies ist notwendig, da nicht an jedem Standort der Handyempfang gewährleistet ist. Die verschiedenen Layout Ebenen werden anhand der unten aufgeführten Grafik erstellt. Ziel ist es, die Liste schnell und übersichtlich auf ein Stück Papier zu bringen.

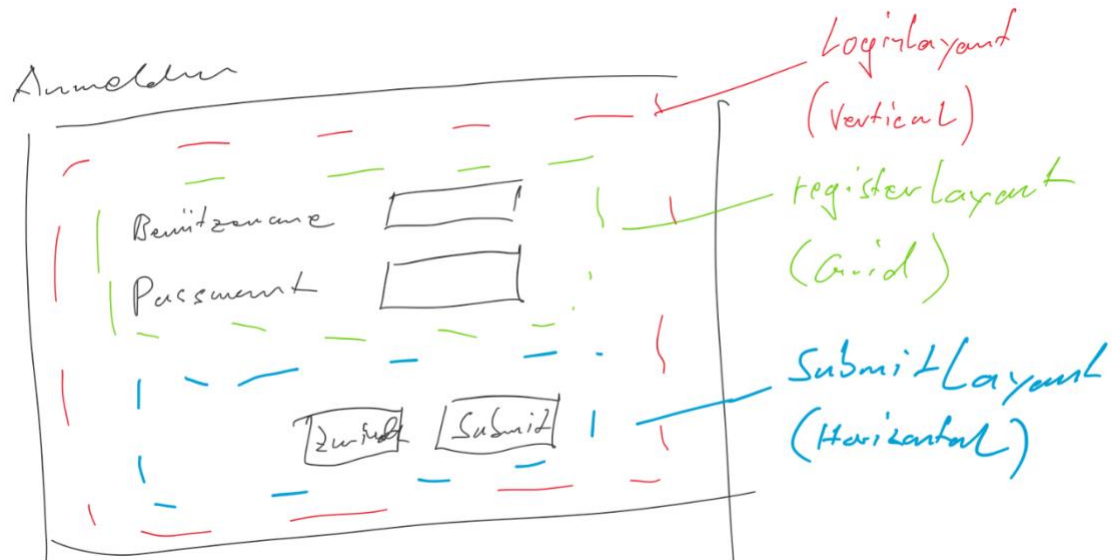
Station Drucken Tab



19: Design Layout Skizze: Drucken

6.2.6 Webdesign: Login

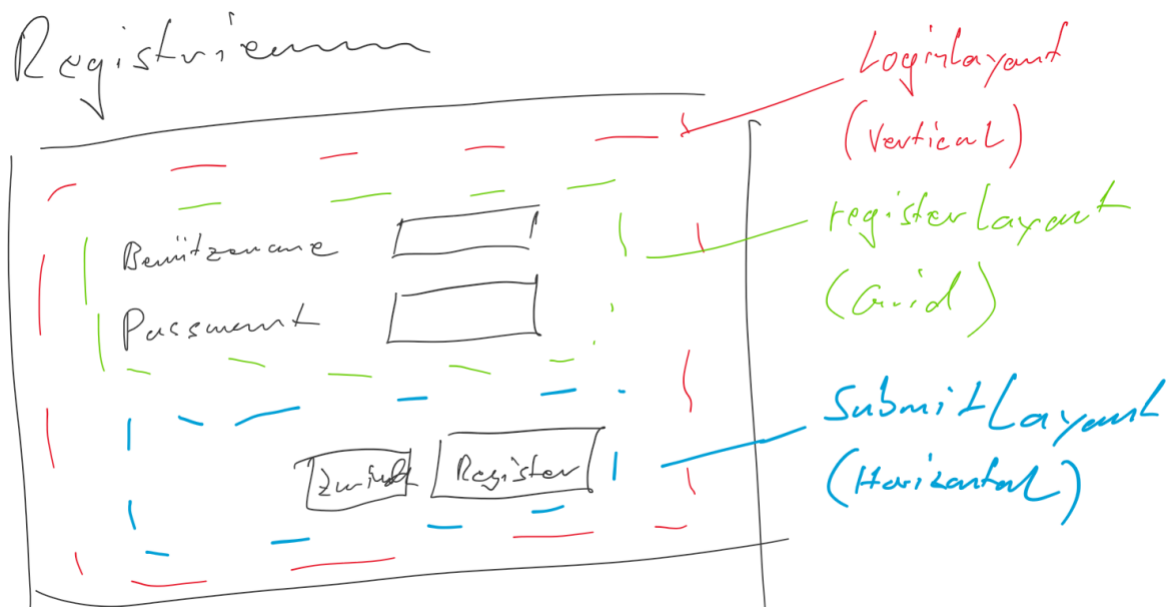
Der Anmeldebildschirm dient dazu, damit der Anwender sich Authentifizieren kann und ihm die korrekte Berechtigung zugewiesen wird. Je nach Berechtigungsstufe kann dieser Artikel Suchen, Artikel erfassen oder auch Artikel löschen. Der Aufbau der verschiedenen Layer ist auf der unten erstellen Skizze ersichtlich



20: Design Layout Skizze: Login

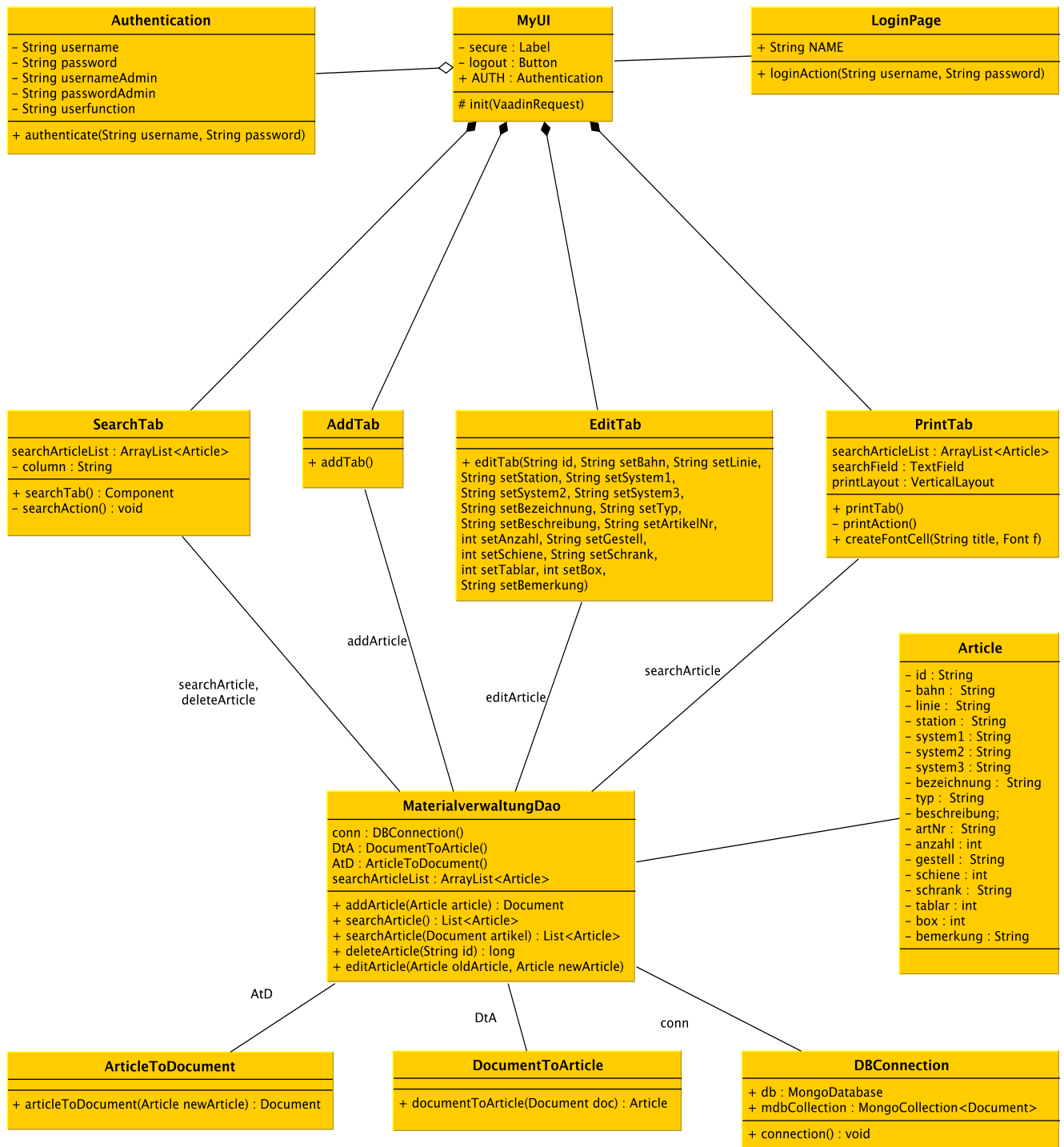
6.2.7 Webdesign: Registrierung

Bei der Registrierung, können sich neue Anwender Registrieren. Das Layout entspricht fast jenem vom Login, der Unterschied liegt wie auf der Skizze ersichtlich bei der Schaltfläche Register.



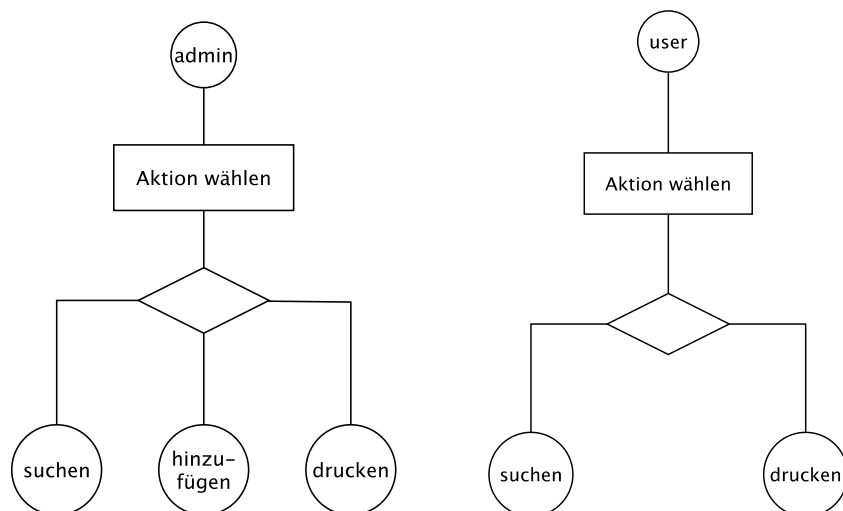
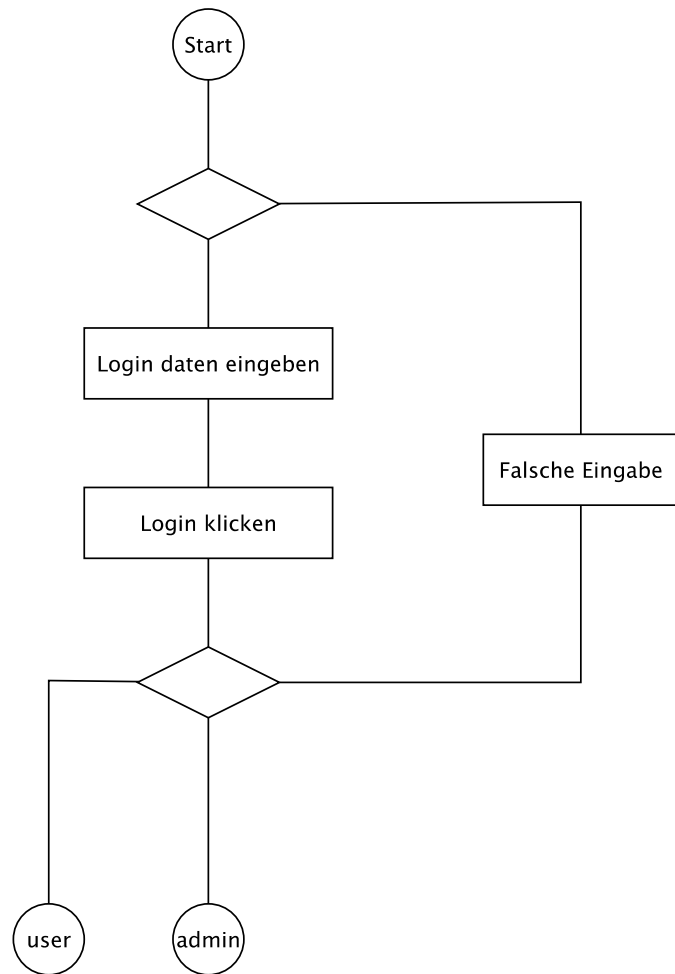
21: Design Layout Skizze: Registrierung

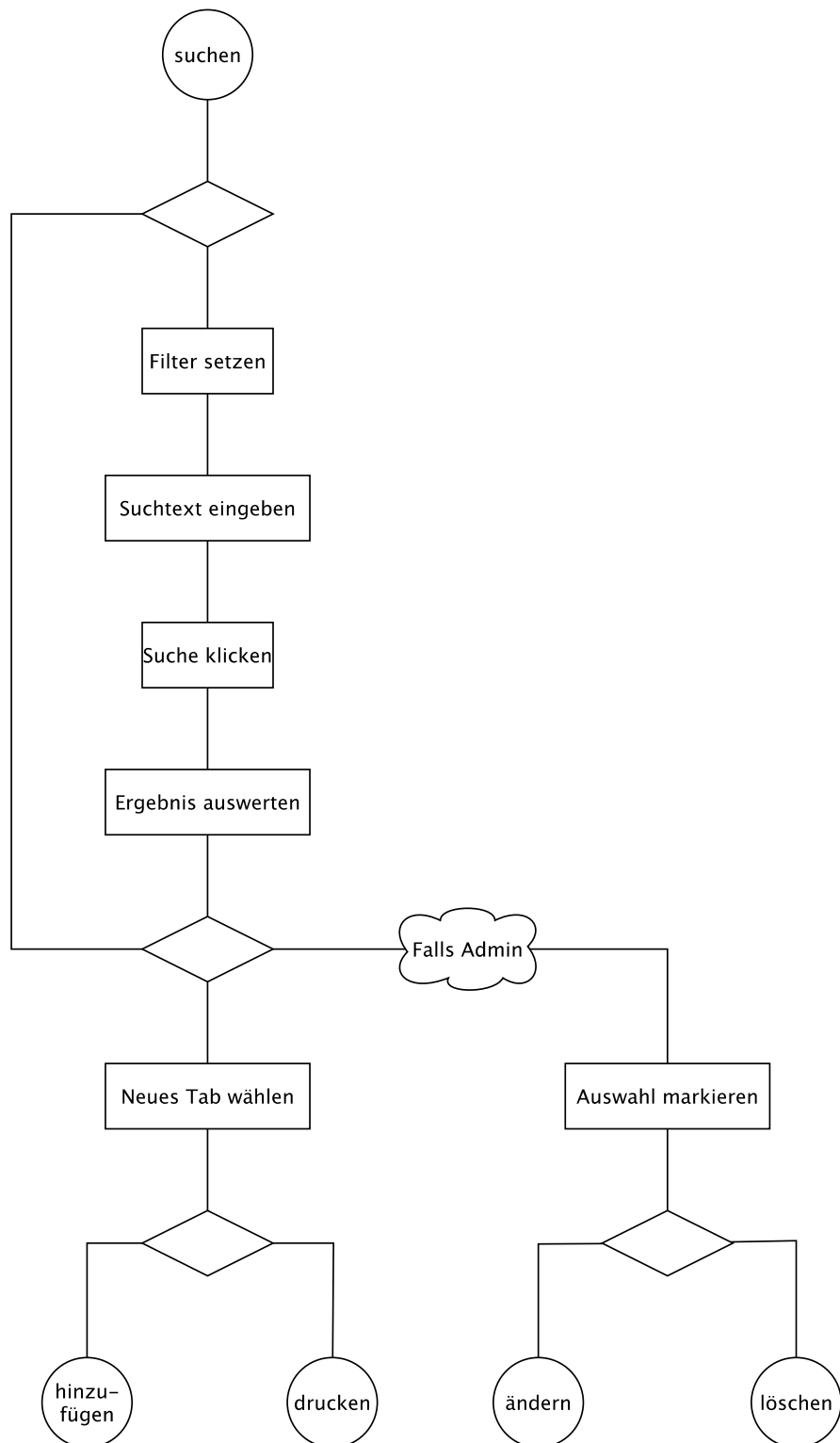
6.3 Klassendiagramm

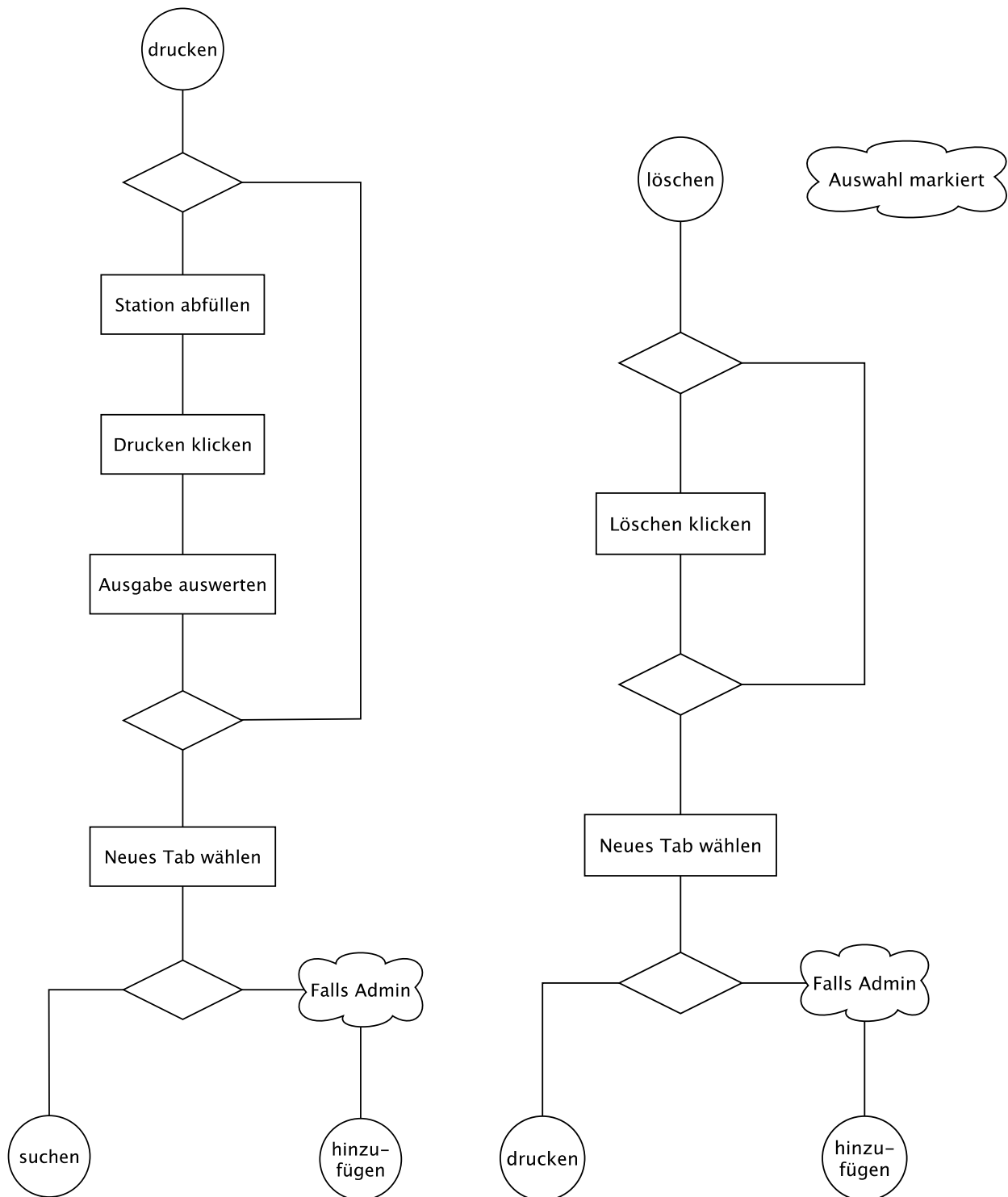


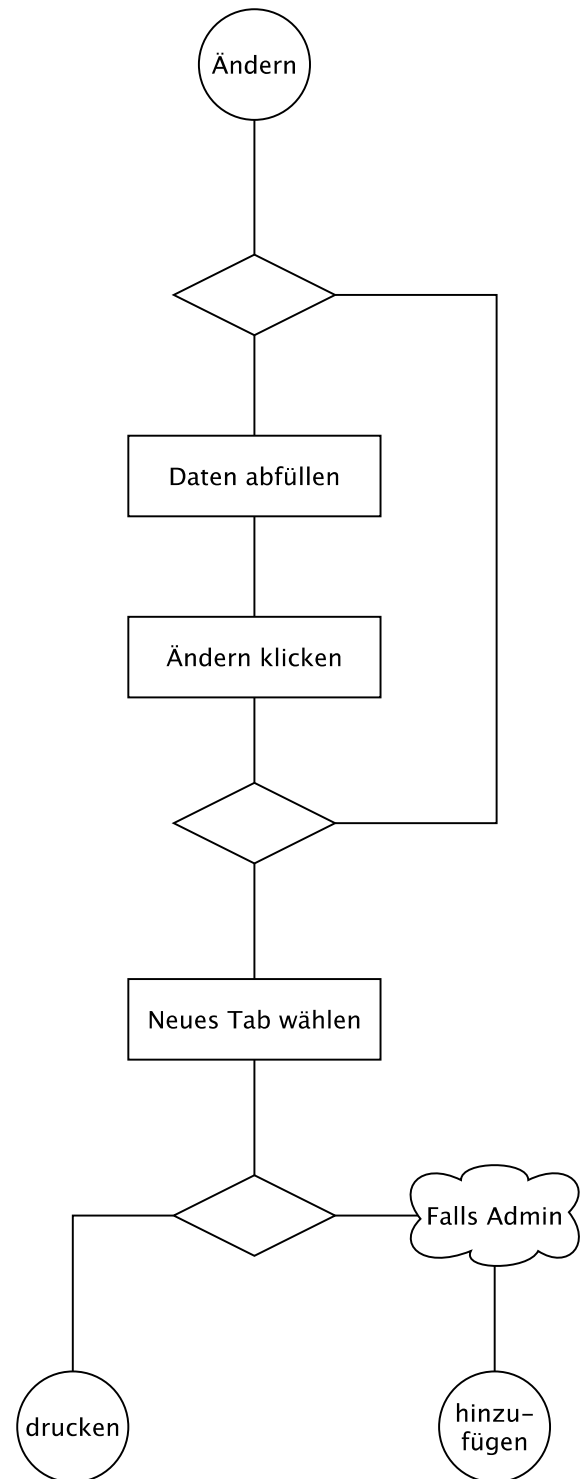
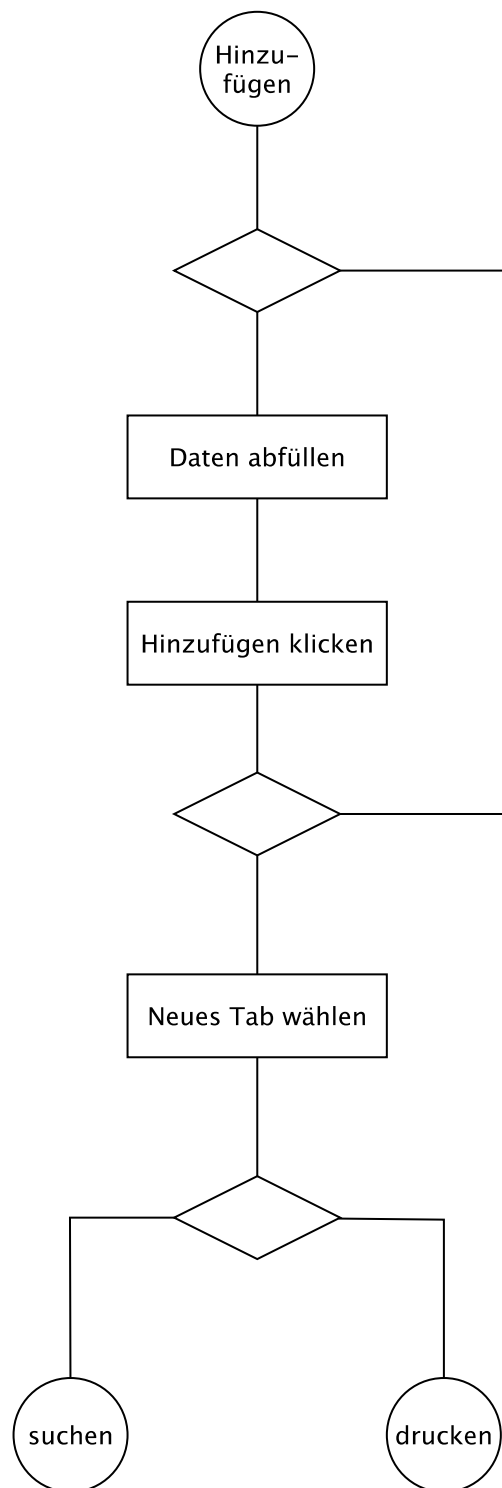
22: UML Klassendiagramm

6.4 Aktivitätsdiagramm









6.5 Übersicht Raspberry Pi

Erst mal ein paar grundlegende Informationen zum Raspberry Pi. Ursprünglich wurde der Einplatinencomputer zu Schulungszwecken entwickelt. Aufgrund seines günstigen Preises wurde er unter Bastlern sehr schnell beliebt. Die Größe einer Kreditkarte, aber dennoch ein vollwertiger Computer. Er bietet Schnittstellen für HDMI, Ethernet, USB, Audio, frei programmierbare GPIO-Ports (werden noch erläutert) und einen SD-Karten Einschub, worauf auch das Betriebssystem seinen Platz findet. Die Stromversorgung erfolgt über ein Micro-USB Anschluss.

Als Betriebssystem kommt ein Linux zum Einsatz. Die bekannteste Distribution auf dem Raspberry ist dabei Raspbian, eine angepasste Variante von Debian. Auf dem Debian installieren wir den Datenbankserver MongoDB wie auch einen Webserver von Apache Tomcat. Diese Software wird benötigt, damit unsere Webapplikation auf dem Raspberry betrieben werden kann.



24: Raspberry Pi 3

6.6 MongoDB

MongoDB, der ist Name abgeleitet aus dem englischen humongous also gigantisch, ist eine dokumentenorientierte NoSQL-Datenbank. Da die Datenbank im Gegensatz zu SQL Datenbanken dokumentenorientiert ist, werden in der Datenbank Sammlungen von JSON-ähnlichen Dokumenten verwaltet. So können viele Anwendungen Daten auf natürlichere Weise modellieren, da die Daten zwar in komplexen Hierarchien verschachtelt werden können, dabei aber immer abfragbar bleiben und indiziert werden kann.

Die ersten Erfahrungen mit MongoDB haben wir im Fach Datenbankdesign, SQL Teil II bei Rainer Strauss gesammelt. Aufgrund der Erfahrungen und einem ersten Projekt während dem Unterricht, haben wir uns dazu entschieden (siehe auch Entscheid der Analyse) für unsere Semesterarbeit eine MongoDB einzusetzen. Die Ursprungsdaten der Materialverwaltung befindet sich einer Excelliste also auch eine Dokumentenbasierten Datenbank. Was eignet sich also Besser als eine Mongo Datenbank.

MongoDB auf dem Raspberry Pi installieren

```
-----
nano /etc/apt/sources.list
** Einfügen **
deb http://repo.mongodb.org/apt/debian stretch main
** Speichern **
sudo apt-get update
apt-get install mongodb
nano /etc/mongodb.conf
** Ende von Datei Zeile hinzufügen **
smallfiles=true
** Speichern **
/etc/init.d/mongodb start
** Konsole starten
mongo
```

25: Installation MongoDB auf dem Raspberry Pi

6.7 Apache Tomcat

Der Apache Tomcat ist ein kostenloser Open Source Webserver und Webcontainer, der die Spezifikation für Java Servlets und Java Server Pages (JSP) implementiert und es damit erlaubt, in Java geschriebene Webapplikationen auf Servlet- respektive JSP-Basis auszuführen.

Auch hier konnten wir die ersten Erfahrungen im Fach Datenbankdesign, SQL Teil II bei Rainer Strauss sammeln. Wir mussten für die letzte Prüfung eine Webseite programmieren, welche auf dem Apache Tomcat lief. Da die Handhabung und Integration mit Netbeans sehr einfach ist, werden wir bei der Semesterarbeit auch wieder mit diesem Webserver arbeiten.

Apache Tomcat installieren

```
-----  
apt-get update  
apt-get install tomcat8 tomcat8-admin libtcnative-1  
usermod -a -G gpio tomcat8
```

26: Installation Apache Tomcat auf dem Raspberry Pi

6.8 No-IP

No-IP ist ein Anbieter von dynamischen DNS Einträgen, mit welchem sich Heimnetzwerke mit einem leicht zu merkenden Domänen Namen verknüpfen lassen. Wer von unterwegs auf sein Heimnetzwerk zugreifen möchte, der muss wissen wie die Öffentliche IP-Adresse von seinem Internetanschluss lautet. Da dieser jedoch meistens dynamisch ist, sich also alle ca. 24 Stunden wechselt, ist dies nicht ganz so einfach. Eine stets gleichbleibende Domain ist somit eine grosse Erleichterung.

DNS ist die Kurzform von Domain-Name-System. Damit wird der Dienst im Internet bezeichnet, der Adressen wie www.google.ch in die entsprechenden IP-Adressen übersetzt. Die Aufgabe vom DNS gleicht einem Telefonbuch. Die Domäne entspricht dem Namen und die IP-Adresse der Telefonnummer des Gesprächspartners. Nötig ist diese Übersetzung, da wir uns besser Namen als Zahlen merken können. Hinzu kommt noch das die Zahlen in vier dreier Blöcken dargestellt werden (z.B. 151.242.353.64).

Damit also unsere Webapplikation von überall und einfach erreicht werden kann. Haben wir uns für den Einsatz vom kostenlosen Dienst von No-IP entschieden. Einziger Nachteil ist, das der Domänenname jeden Monat neu bestätigt werden muss.

No-IP installieren

```
-----  
mkdir /home/pi/noip  
cd /home/pi/noip  
wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz  
tar vzxvf noip-duc-linux.tar.gz  
cd noip-2.1.9-1  
sudo make  
sudo make install  
sudo /usr/local/bin/noip2  
sudo noip2 --S (Capital "S")
```

27: Installation No-IP auf dem Raspberry Pi

6.9 NetBeans IDE

NetBeans ist eine kostenlose Entwicklungsumgebung, welche in Java geschrieben wurde und auf der NetBeans Plattform läuft. In Netbeans kann mit verschiedensten Programmiersprachen entwickelt werden, hauptsächlich wird aber mit Java entwickelt. Im Unterricht an der TEKO entwickeln wir ebenfalls Java Programme mit NetBeans. Aus diesem Grund werden wir unser Projekt ebenfalls mit Netbeans entwickeln.

7 Programmierung

7.1 Systemanforderungen

Beim Raspberry Pi müssen folgende Funktionen aktiviert/installiert sein:

- Netzwerkschnittstelle (WLAN oder LAN)
- SSH aktivieren für Zugriff von aussen
- Apache Tomcat 8
- MongoDB
- Java 8 mit Maven

7.2 Klassen und Methoden

7.2.1 MyUI

- `init`: Diese Methode ist der Einstiegspunkt des Programms und beinhaltet die Erstellung des Vaadin typischen Layouts für Komponenten des UI's. Als Erstes wird man auf einer Login Seite gebeten sich anzumelden. Dies geschieht mit Hilfe der Klasse `Authentication` und `LoginPage`. Als zentrales Element wurde ein `TabSheet` gewählt, welches es uns erlaubt einfach zwischen den gewünschten Aktionen zu wechseln.

7.2.2 LoginPage

- Die Komponenten für das Login wird hier erstellt und über die `loginAction` zum Funktionieren gebracht.
- `loginAction`: Es wird validiert ob die eingegebenen Daten korrekt sind oder nicht. Falls `true` zurückgeliefert wird die Session erstellt mit der jeweiligen Berechtigungsstufe. Dies unterscheidet dann welche Optionen einem zur Verfügung stehen. In einer Admin Session wird zusätzlich die Option Ändern oder Löschen angeboten.

7.2.3 Authentication

- `authenticate`: Hier werden die Eingaben aus der vorgegangenen Login Maske als Parameter übergeben und validiert ob die Eingaben zu einem der Profile passt.

7.2.4 DBConnection

- `connection`: Mit dem Parameter der Collection wird diese erstellt oder darauf zugegriffen in der DB "Materialverwaltung".

7.2.5 SearchTab

- In dieser Klasse werden die UI Komponente für das Suchen-Tab erstellt. Je nach aktueller Berechtigung werden nach der Suche unterschiedliche Aktionen angeboten. Als User wird ausschliesslich die Suchen angeboten. Als Admin wird nachträglich das Ändern oder Löschen angeboten.
- `searchTab`: Hier werden die Komponenten erstellt und in die vorgesehenen Layouts abgefüllt. Sobald der Suche Button betätigt wird, entweder per Mausklick oder Enter, wird die `searchAction` ausgelöst.
- `searchAction`: Falls das Textfeld der Suche leer ist wird die ganze Datenbank durchsucht und in eine Tabelle abgefüllt. Sollte ein Filter und ein Text eingegeben sein wird nur der gewünschte Eintrag gesucht. Nach dem Erstellen der Tabelle wird geprüft welche Berechtigung aktiv ist und erstellt allenfalls noch die Buttons Ändern und Löschen.

7.2.6 AddTab

- Hier werden die Komponenten für das Hinzufügen von neuen Artikeln erstellt und dem Layout übergeben. Im Tab Hinzufügen wird das Ganze dann visualisiert, falls man als Admin angemeldet ist.
- `addTab`: Erstellen der Komponente zur Eingabe der Daten für den neuen Artikel. Durch den Hinzufügen Button werden alle Daten abgefangen die in den Textfeldern stehen und danach wieder zurückgesetzt. Durch die Methoden `addArticle` in der `MaterialverwaltungDao` Klasse wird diese in die Datenbank geschrieben.

7.2.7 EditTab

- Im SuchenTab wird, falls man als Admin angemeldet ist, nach einer Suche der Button Ändern angeboten. Durch das Markieren eines Eintrages in der Tabelle und Klicken des Buttons Ändern, wird dieser ausgelesen und in die Eingabemaske eingefüllt um ihn dann anzupassen.
- editTab: Hier werden alle Werte des markierten Eintrages in die Textfelder eingetragen. Durch das Ändern wird der alte Eintrag gelöscht und ein neuer gespeichert.

7.2.8 PrintTab

- Um die Listen auf den jeweiligen Standorten abzulegen kann hier eine Station eingegeben werden und als PDF Datei gespeichert oder gedruckt werden.
- printAction: Durch die Eingabe einer Station wird ein Filter erstellt nachdem in der Datenbank alle Einträge gesucht werden. Diese werden in einer Tabelle im PDF dargestellt.

7.2.9 MaterialverwaltungDao

- Die zentrale Klasse für die Kommunikation mit der Datenbank. Alle Operationen mit der Datenbank laufen über die MaterialverwaltungDao Klasse. Sie beinhaltet keinerlei grafische Komponente. In den Einzelnen Methoden wird jeweils eine Verbindung zur Datenbank erstellt.
- addArticle: Der übergebene Artikel wird in ein Document abgefüllt um ihn in die Datenbank einzufügen. Dies geschieht mit der ArticleToDocument Klasse.
- searchArticle: Diese Methode wird überlagert einmal ohne Parameter und einmal mit Parameter für eine spezifische Suche. In der Variante ohne Filter wird die Komplette Datenbank ausgelesen und in einer Liste zurückgegeben. Andernfalls kann ein Filter mitgegeben werden nachdem in der Datenbank gesucht wird und nur diese Einträge zurückgibt. Die ausgelesenen Einträge werden in ein Artikel umgewandelt mit Hilfe der DocumentToArticle Klasse.
- deleteArticle: Mit Hilfe der eindeutigen ObjektId der Datenbank Einträge wird der gewünschte Eintrag gelöscht. Es wird zur Kontrolle ein long Wert zurückgeliefert wieviel Einträge gelöscht wurden.
- editArticle: Hier werden zwei Artikel übergeben, alt und neu. Im Alten Objekt ist nur die ObjektId vorhanden und wird zuerst gelöscht. Falls der Eintrag erfolgreich gelöscht werden konnte wird der neue Artikel in die Datenbank hinzugefügt.

7.2.10 ArticleToDocument

- Wandelt ein übergebenes Article Bean in ein Document um. Dies wird benötigt um Einträge für die Datenbank bereitzustellen.
- articleToDocument: Erstellt ein Document und füllt diese ab.

7.2.11 DocumentToArticle

- Wandelt ein übergebenes Document in ein Article Bean um. Dies wird benötigt für das darstellen in einer Tabelle im UI.
- documentToArticle: Erstellt ein Article Bean und füllt es ab.

7.2.12 Article

- Stellt alle benötigten Attribute und Konstruktoren zur Verfügung um die Datenkonsistenz der Datenbank zu gewährleisten.
- Bietet getter und setter für alle Attribute.

8 Testing

Testverfahren

Zum Testen der Materialverwaltung haben wir verschiedene Testfälle erstellt, um die Funktionalitäten des Programms auf ihre Funktionsweise zu testen. Diese sind unter Testfälle aufgelistet. Wir haben die Tests bewusst auf unterschiedlichen Endgeräten ausgeführt. Christoph mit einem MacBook und Thomas mit einem Windows Notebook.

Testspezifikation

In den nachfolgenden Testfällen wird die Funktionalität des Programms überprüft.

8.1 Testfälle

Nr.	Testfall	Beschreibung	Erwartetes Ergebnis
01	Login	Login mit korrekten wie auch falschen Informationen testen. Versuchen ob die Seite ohne korrektes Login zugreifbar ist.	Programm nur mit korrekten Benutzer ersichtlich.
02	Daten suchen	Die Suche mit verschiedenen Filter durchführen. Ergebnisse überprüfen in der MongoDB kontrollieren.	Korrekte Datensätze werden angezeigt.
03	Daten erfassen	Datensätze hinzufügen und in MongoDB kontrollieren.	Neue korrekte Datensätze in der Datenbank vorhanden.
04	Daten bearbeiten	Bestehende Daten ändern und in MongoDB kontrollieren.	Die Datensätze können bearbeitet werden.
05	Daten löschen	Bestehende Daten aus der Datenbank löschen.	Die Datensätze werden gelöscht.
06	Daten Drucken	PDF mittels Suche generieren. Verschiedene Suchen als PDF ausgeben.	Die Daten im PDF entsprechen der Suche.
07	Kontrolle Berechtigungen	Admin Benutzer sieht alle Menüpunkte. Benutzer User kann Suchen und Drucken.	Die Berechtigungen Funktionieren wie gewünscht.
08	Logout	Nach dem Ausloggen kann nicht mehr auf den Webseiteninhalt zugegriffen werden.	Nach einem Logout kein Zugriff mehr.

8.2 Testprotokoll

Testperson: Christoph Grossmann

Nr.	Datum	Testergebnis	Bemerkungen
01	15.09.2018	Bestanden	
02	15.09.2018	Bestanden	
03	15.09.2018	Bestanden	
04	15.09.2018	Bestanden	
05	15.09.2018	Bestanden	
06	15.09.2018	Bestanden	
07	15.09.2018	Bestanden	
08	15.09.2018	Bestanden	

Testperson: Thomas Baechler

Nr.	Datum	Testergebnis	Bemerkungen
01	16.09.2018	Bestanden	
02	16.09.2018	Bestanden	
03	16.09.2018	Bestanden	
04	16.09.2018	Bestanden	
05	16.09.2018	Bestanden	
06	16.09.2018	Bestanden	
07	16.09.2018	Bestanden	
08	16.09.2018	Bestanden	

9 Projektabschluss

9.1 Schlussprodukt

Das von uns vorgenommene Projekt konnten wir erfolgreich umsetzen. Wir haben eine Webapplikation entwickelt, auf welcher die Materialverwaltung der SBB vorgenommen werden kann. Alle Tests waren erfolgreich und das Raspberry kann an einem Internetanschluss in Betrieb genommen werden. In unserem Projekt sind aus Datenschutzgründen nur Testdaten vorhanden. Die Produktiven Daten können in die MongoDB mit .JSON importiert werden.

9.2 Erfahrungen

9.2.1 Umgang mit Daten der MongoDB und Java

Die verschiedenen Aktionen wie einfügen, ändern, löschen, suchen, gefilterte suche stellten sich als nicht ganz einfach anzuwenden heraus. Es gibt eine Vielzahl an Variationen um mit den Daten in der MongoDB zu arbeiten. Wir haben einiges ausprobiert und haben schlussendlich die für uns passendsten verwendet. Die Architektur einer MongoDB war uns bis dato nur wenig vertraut durch den Unterricht im Datenbankdesign. Trotz der kleinen Vorahnung sind wir schnell damit zurechtgekommen. Die Möglichkeit des erstellen mehrerer JSON Dateien in einer verschachtelten Datei machte das Design der Datenbank aufwendiger als gedacht. Es wurden viele Möglichkeiten durchgespielt bis wir uns schliesslich für eine der Simpelsten entschieden. Der Umgang mit dem MongoDB Treiber und dem Erstellen einer Verbindung zur DB machte uns nicht grosse Probleme.

9.2.2 Umgang mit GIT

Es war für uns beide das erste Mal GIT in einem Team zu verwenden indem gleichzeitig an verschiedenen Funktionen gearbeitet wird. Wir haben es so gehandhabt das jeder sich eine Funktion vornimmt die er implementieren möchte und dafür einen neuen Branch macht. Sobald die Funktion getestet ist und funktionierte wurde sie in den Master gemerged um dem anderen die neue Funktion ebenfalls bereit zu stellen. Dies geschah reibungslos und wir sind grösstenteils ohne Konflikte davongekommen. Es wurde sehr darauf geachtet das wir uns mit dem neuen Programmcode nicht in die Quere kommen. Jedoch konnte es nicht immer vermieden werden. Die Übung merge Konflikte zu lösen kam uns aber ebenfalls zugute und stellt nun keine Probleme mehr dar.

9.2.3 Umgang mit dem Vaadin Framework

Vaadin war uns beiden völlig unbekannt. Durch genaueres informieren haben wir etwas tiefer in das Framework hineingesehen und waren begeistert das es uns für Möglichkeiten bietet um eine Webapplikation zu schreiben. Die Tatsache das ein UI nur mit der Programmiersprache Java geschrieben wird und mit CSS praktisch grenzenlos angepasst werden kann hat uns eine neue Welt eröffnet. Im Vaadin Framework sind bereits die wichtigsten Elemente zur Anordnung in Layouts wie Horizontal, Vertikal oder in einer Tabellenform vorhanden. Es hat uns Spass bereitet mit den Möglichkeiten zu spielen und für uns die richtige Form zu finden. Natürlich hat nicht alles auf Anhieb gepasst und musste in mühsamer Kleinstarbeit an den rechten Fleck gebracht werden.

9.2.4 Umgang mit CSS

Fluch und Segen kann über einen kommen während man mit CSS arbeitet. Es ist sehr schwierig bei sehr begrenzter Erfahrung mit CSS die richtige Feineinstellung zu finden. Wenn man seinen Style gefunden hat sind die Möglichkeiten praktisch endlos.

9.2.5 Umgang mit DynDNS

Das Erstellen einer kostenlosen DDNS Domain machte uns keine Probleme. Wir konnten uns schnell für einen Anbieter entscheiden und waren zufrieden. Das grössere Problem ergab sich aus den Einstellungen um die Website von aussen zugänglich zu machen. Einstellungen wie Portweiterleitung und das Einstellen des Webserver. Thomas ging das leichter von der Hand und hat schlussendliche alles richtig Konfiguriert um unsere Webapplikation der Welt zugänglich zu machen.

9.3 Reflexion

Eine eigene Webapplikation zu realisieren war nicht ganz einfach. Wir einigten uns, dass Projekt zuerst gut zu planen und erst nach einer erfolgreichen Planung alles Umzusetzen. Bei der Planung stellten wir schnell fest, dass es gar nicht so einfach ist, eine Webapplikation für eine Excelliste umzusetzen. Die Programmiersprache Java war gegeben, jedoch auf welchem Webserver und mit welcher Datenbank werden wir die Semesterarbeit realisieren? Mit dieser Frage startete das grosse Recherchieren und brachte unseren Zeitplan das erste Mal durcheinander. Die Auswahl an Produkten ist beinahe endlos, jedoch ist dieser Entscheid auch davon Abhängig wo die Webapplikation am Ende laufen wird. Wollen wir diese Fremd Hosten oder doch bei uns betreiben. Auch diese Analyse kostete uns sehr viel Zeit. Es war uns bewusst, dass wir für die Analysephase viel Zeit investieren müssen, doch mit so viel Zeit hatten wir nicht gerechnet.

Wir entschieden uns dafür, dass es am Besten ist, wenn wir die Applikation auf einem Raspberry Pi selber betreiben und somit auch selber entscheiden können, mit welcher Software das Projekt umgesetzt wird. Die Vielfältigkeit wie auch die Flexibilität ist einfach am grössten.

Beim Webserver einigten wir uns schnell auf den Apache Tomcat, doch in welche Datenbank werden wir unsere Daten abspeichern? SQL oder MongoDB? Der Entscheid fiel uns nicht leicht, da unsere Kenntnisse mit Webapplikation ziemlich bescheiden sind. Wir starteten zuerst mit PostgreSQL und erstellen ein Datenbankmodell. Beim Erstellen kamen die ersten Zweifel und als wir am Ende die Anzahl Tabellen für die Anzahl Daten sahen, wussten wir, dass SQL vielleicht doch nicht die Optimale Lösung sein wird. Der Entscheid war ein wichtiger Wegweiser, welchen wir aber aufgrund von unserem Wissen nicht alleine tragen konnten. Wir einigten uns darauf das Thema mit Rainer Strauss, Lehrer vom Fach Datenbankdesign zu besprechen und auf sein Wissen zurückzugreifen. Er riet uns dazu, dass Projekt mit MongoDB umzusetzen, was wir dann auch so machten. Die Analyse war abgeschlossen.

Weiter ging es mit dem Design. Wir suchten im Internet nach Ideen und erstellten Skizzen vom Design der Webseite. Wichtig war uns, dass die Webseite nur soviel wie nötig, doch so wenig wie möglich enthält. Das Design der Datenbank war nicht so Aufwändig. Wir achteten einfach darauf, dass alle Daten in einem logischen Datentyp abgespeichert werden. Das Klassendiagramm und das Aktivitätsdiagramm erstellten wir Anhand vom Webdesign. Alle Funktionalitäten wurden mit beim Erstellen vom Pflichtenheft festgelegt. Auch zu diesem Zeitpunkt waren wir weiter hinter unserem Zeitplan. Endlich konnten wir an das Programmieren.

Beim Programmieren und Dokumentieren teilten wir die Arbeiten jeweils auf, die erledigten Arbeiten kontrollierten wir gegenseitig und besprachen wöchentlich das weitere Vorgehen. Somit konnten wir selbständig und unabhängig voneinander Entwickeln. Sehr viel Zeit verloren wir beim platzieren der Steuerelemente. Oft wollten sich diese nicht wie gewünscht platzieren lassen. Auch das Erstellen vom gewünschten PDF erwies sich nicht als so einfach. Das PDF selber zu erstellen war nicht das Hauptproblem, sondern alle Daten auf einer A4 Seite darzustellen. Beim Programmieren mussten wir sehr viel Recherchieren und probieren, da unsere Kenntnisse und Erfahrungen mit Webapplikationen sehr bescheiden sind. Aufgrund vom Zeitmangel begnügten wir uns gegen Schluss auch damit, dass gewisse Dinge Funktionieren und nicht perfekt Programmiert sind. Dazu fehlte einfach die Zeit.

Beim Testverfahren haben wir uns für das manuelle Testen entschieden. Der Code wurde immer vorneweg getestet. Dazu haben wir noch verschiedene Tests für einen Abschlusstest festgelegt. Wobei wir unabhängig voneinander das Programm nochmals getestet haben. Zum Glück waren alle Tests erfolgreich, denn wir hätten am Ende keine Zeit mehr gehabt, grössere Probleme zu beheben.

9.4 Lessons Learned

9.4.1 Analyse und Design

Für ein erfolgreiches Projekt das allen Anforderungen genügt braucht es ein angemessenes Mass an Planung. Wir haben es etwas unterschätzt wieviel Zeit es in Anspruch nimmt eine Webapplikation von Grund auf selbst zu entwerfen und umzusetzen. Eine vollumfängliche Analyse erspart spätere Stolpersteine oder Leerläufe. Wir nehmen diese Erkenntnis für das nächste Projekt mit und werden dies in unserem nächsten Zeitplan berücksichtigen.

9.4.2 Recherchen einplanen

Die meisten Funktionen die wir in unserer Webapplikation verwenden mussten waren komplettes Neuland für uns. Es kostete viel Zeit und Kraft laufend, jeden weiteren Schritt im Projekt neu zu recherchieren und sich das benötigte Wissen anzueignen. Ein grosser Teil davon muss bereits in der Analyse und dem Design abgeklärt werden. Wir nehmen aus diesem Projekt die Erkenntnis mit, dass bereits vor dem eigentlichen Implementieren des Programmcodes die ungefähre Richtung bestimmt sein muss. Unbekannte Klassen oder Methoden müssen ausfindig gemacht werden damit man darauf zu arbeiten kann diese zu verwenden und nicht komplett im Dunkeln steht, wenn es dazu kommt.

9.4.3 Variantenentscheid durchdenken

Als wir uns für das Eingenhosting entschieden und unser Raspberry Pi aufsetzten, ahnten wir noch nicht was für ein Problem auf uns zukommen würde. Kurz vor Fertigstellung unseres Produktes führten wir das Deployment auf dem Webserver des Raspberry Pi durch. Alle Tests waren bereits Lokal abgeschlossen und die Webapplikation voll funktionsfähig. Wir mussten feststellen das die Verfügbare Version der MongoDB auf einem 32 Bit System höchstens 2.6 beträgt. Unsere verwendete Version für die Webapplikation ist aber 3.8. Diese waren nicht miteinander kompatibel und somit nicht lauffähig. Im Eiltempo suchten wir nach einer Lösung des Problems und fanden eine. Ein experimentelles 64 Bit OS das auf dem Raspberry läuft und somit die höheren Versionen der MongoDB unterstützt. Abermals wurden wir enttäuscht. Diese OS läuft nur bis zum Modell Raspberry Pi 3B und wir hatten für unser Projekt das Modell 3B+ besorgt. Da wir jedoch durch die TEKO im Besitz eines Modell 3B waren gelang es uns dieses aufzusetzen und unsere Webapplikation zu deployen. Leider mussten wir ebenfalls feststellen das die Funktion Drucken nicht mehr funktioniert. Wir haben aus dieser Misere gelernt eine tiefreichende vorab Recherche kann essenzielle Probleme in der Endphase beseitigen.

9.4.4 Zusätzliche Funktionen

Bereits bei der Ausarbeitung des Projektantrages muss man sich versuchen auf das Wesentliche zu konzentrieren und nicht bereits zusätzliche Funktionen versuchen einzubauen. Man unterliegt sehr schnell der Versuchung das bestmögliche und allumfassende Programm zu kreieren. Von dieser Vorstellung muss man sich direkt am Anfang trennen und sich für die gegebene Zeit bzw. Budget die wichtigsten Grundfunktionen herausnehmen. Ein stabiles Grundprogramm kann einfach erweitert und verschiedensten Wünschen angepasst werden. Wir lernen daraus sich auf die Essenzielle Aufgabe des Programms zu fokussieren und keine unnötigen Extras einzubauen.

9.5 Fazit

Das Projekt war durchgehend interessant. Vor allem das Sammeln der ersten Erfahrungen mit Webapplikationen und dem Vaadin Framework.

10 Anhang

10.1 Quellenverzeichnis

== und equals

<https://www.gutefrage.net/frage/java-unterschied-zwischen--und-equals>

No-IP

<https://www.noip.com/>

MongoDB

<https://www.mongodb.com/>

Raspberry Pi

<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

Apache Tomcat

<http://tomcat.apache.org/>

Java Overview

<https://docs.oracle.com/javase/8/docs/api/index.html?overview-summary.html>

Vaadin Framework Sampler

<https://demo.vaadin.com/sampler/>

Vaadin Documentation

<https://vaadin.com/api/platform/11.0.0/overview-summary.html>

CSS Tutorial

<https://www.w3schools.com/css/default.asp>

Diverses von OLAT Bruno Hammer

<https://elad.ch/olat/dmz/>

10.2 Abbildungsverzeichnis

Abbildungsnummer	Beschreibung
1	Titelbild / Logo
2	Thomas Baechler
3	Christoph Grossmann
4	Lösungsvariante Server wird Fremdgehostet
5	Lösungsvariante Accesspoint
6	Lösungsvariante Software von Microsoft
7	Lösungsvariante Apache und MySQL
8	Lösungsvariante Apache mit MongoDB
9	Gewichtung der Variante Hardware
10	Gewichtung der Variante Webserver
11	Gewichtung der Variante Datenbank
12	Risikoanalyse
13	Zeitablauf
14	Erster Entwurf Weblayout
15	Design Layout Skizze: Startseite
16	Design Layout Skizze: Suchen
17	Design Layout Skizze: Erfassen
18	Design Layout Skizze: Löschen
19	Design Layout Skizze: Drucken
20	Design Layout Skizze: Login
21	Design Layout Skizze: Registrierung
22	UML Klassendiagramm
23	UML Aktivitätsdiagramm
24	Raspberry Pi 3
25	Installation MongoDB auf dem Raspberry Pi
26	Installation Apache Tomcat auf dem Raspberry Pi
27	Installation No-IP auf dem Raspberry Pi

10.3 Arbeitsjournal

Datum	Arbeit	Christoph	Thomas
28.04.2018	Projektantrag anhand von Ideen erstellen	2	2
02.05.2018	Arbeitsjournal erstellen und erste Einträge erfassen		1
08.05.2018	Sitzung in Emmen: Projektablauf definieren, Zeiten festlegen, Analyse	3	3
12.05.2018	Vorlage Excel Liste anpassen, Icon erstellen, Ideensammlung Pflichtenheft	2	
18.05.2018	Installationanleitung für Raspberry Pi suchen, DynDNS erstellen		2
22.05.2018	Sitzung in Emmen: Umsetzungsvarianten Besprechen und festhalten. Plattform, Datenbank, Webserver und Programmiersprache festlegen	3	3
27.05.2018	Erste Dokumentation Variantenentscheid		2
14.06.2018	Sitzung in Emmen: Weiteres Vorgehen besprechen, Datenbank mit SQL aufzeichnen, DB Variante noch offen	3	3
23.06.2018	Projekt mit Rainer besprechen, Variantenvorenscheid DB - MongoDB. Flexibler Plattformunabhängig	1	1
26.06.2018	Projektablauf überarbeiten, Arbeitsjournal ergänzen, Besprechung Variantenentscheid Datenbank		
04.07.2018	Pflichtenheft erstellen und in die Dokumentation einfügen	2	
10.07.2018	Sitzung in Emmen: Besprechung weiteres Vorgehen. Wie DDNS funktioniert. Netbeans Merge, Branch anschauen. Erstes grobes Klassendiagramm. MongoDB keep	3	3
11.07.2018	Raspberry Pi 64 bit aufsetzen und MongoDB installieren		8
14.07.2018	Testwebseite erstellen	4	
31.07.2018	Sitzung in Emmen: Besprechung weiteres Vorgehen und Aktualisieren Arbeitsjournal	3	3
11.08.2018	Telefonische Besprechung Stand und weiteres Vorgehen	1	1
11.08.2018	Dokumentation Variantenentscheid		2
12.08.2018	Dokumentation Variantenentscheid		8
15.08.2018	Dokumentation Design und Umgebung		10
16.08.2018	Sitzung in Emmen: Dokumentation durchgehen, DBConnection und Start Programmierung Insert, MongoDB Document erstellen	3	3
16.08.2018	Erstellen des Programmcodes für den insert von Daten in die MongoDB	5	
19.08.2018	Erstellen des Programmcodes für das suchen von Artikeln	5	
21.08.2018	Weiterführen der suchen Funktion und kleiner Änderungen wie z.B. deutsche Texte	4	
27.08.2018	Erstellen des Programmcodes für das Ändern von Einträgen.	5	
28.08.2018	Erstellen des Programmcodes für das Ändern von Einträgen.	4	
29.08.2018	Raspberry im Web verfügbar machen		3
31.08.2018	Login Programmieren		4
01.09.2018	Weiterführung des Programmcodes für das Ändern, Drucken.	4	4
03.09.2018	Erstellen des Programmcodes für das Löschen von Einträgen	5	
04.09.2018	Erstellen des Programmcodes für das Löschen von Einträgen	5	
06.09.2018	Drucken Programmieren		4
08.09.2018	Weiterführende Programmierung für das Drucken und Erstellen von Javadoc Kommentaren	6	4
10.09.2018	Drucken mit DB Abfrage verknüpfen		4
14.09.2018	Dokumentation nachführen		2
15.09.2018	Dokumentation nachführen und Deployment der Materialverwaltung auf dem RaspberryPi	5	10
17.09.2018	Dokumentation nachführen Beschreibung Klassen und Methoden	3	
		81	90
			171

10.4 Projektantrag

Projekt: 004-Webbasierte Verwaltung
für Reservematerial

TEKO Schweizerische Fachschule | L-TIN-16-T-a

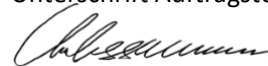
Projektantrag

Empfänger	Absender	
Bruno Hammer < teko@dasycon.ch >	Christoph Grossmann Thomas Baechler	< ch.grossmann@gmx.ch > < t.baechler@bluewin.ch >

Projektname	Webbasierte Verwaltung für Reservematerial
Beweggründe	Die Excel basierte Lagerliste benötigt bereits zum Öffnen mehrere Minuten. Das Aufbereiten von Drucklisten dauert sehr lange und blockiert das System. Es ist keine Kompatibilität für Mobilgeräte vorhanden. Kein gemeinsames Schreibrecht zur laufenden Bearbeitung. Keine Kontrolle über die Lagerbestände an über 20 Standorten.
Beschreibung	Reservematerial an über 20 verschiedenen Standorten unter Kontrolle zu haben ist kein Ding der Leichtigkeit. Um Waren Aus- und Eingänge sowie Reparaturen im Griff zu halten, wird ein Programm benötigt. Dieses soll einfach von Unterwegs aktualisiert werden können, so das Verzögerungen im Bestellprozess vermindert werden können.
Ziel	Die aktuellen Bestände des Reservematerials sollen von überall mit mobilen Geräten eingesehen und gemeinsam bearbeitet werden können.
Aufwand	Der Aufwand wird auf ca.100 Stunden pro Student festgelegt.
Termine	Start: 4.Mai.2018 1. Zwischengespräch: vor Sommerferien 2. Zwischengespräch: nach Sommerferien Abgabe: 22.September.2018

Unterschrift Auftragsempfänger:

Unterschrift Auftragsteller:



Christoph Grossmann



Thomas Baechler