Teil 1: IPA Dokumentation

|  |  |  |  |
| --- | --- | --- | --- |
| **IPA-Daten** | | | |
| **Projektname** | Inventar App | | |
| **Firmenname** | Technische Fachschule Bern, Abt. Informatik | | |
| **Berufsschule** | Technische Fachschule Bern | | |
| **Autor** | Josiah Schiess | | |
| **Experten** | VEX:  HEX: Iseli Erich Reto  NEX: | | |
| **Verantwortliche Fachkraft** | Amati Stefano | | |
| **Berufsbildner** | Giulio Iannattone | | |
| **Fachrichtung** | BET | | |
| **Projektvorgehensmodell** | HERMES 5 | | |
| **Jahrgang und Kanton** | IPA 2020, Kanton Bern | | |
| **Ausgabedatum** | 28.02.2020 | | |
| **Status** | In Arbeit | In Prüfung | Zur Nutzung genehmigt |

Tabelle 0-1: Teil 1: IPA Dokumentation

# Kurzfassung des IPA Berichtes

## Informationen

Innerhalb der 10 Arbeitstage, wurde einen Full-Stack JavaScript Web-Applikation, die Lokal auf einem Rechner läuft erstellt und dokumentiert. Dabei wurden diese Zentrale Frameworks gebraucht:

* Express (REST API)
* Vue (Frontend/Darstellung)
* Knex (Datenbankzugriff)

Es wurde eine Datenbank mit Knex, ein Querry-Building Sprache erstellt. Dazu wurde eine REST API mit Express erstellt. Die Benutzeroberfläche wurde mit dem Visuelle Framework Vuetify, ein Plugin von Vue erstellt.

## Grobe Ausgangssituation

Es wird eine Applikation für die Inventarisierung der vorhandenen Lehr- und Lernmaterialien für die Lehrkräfte und die Lernenden aufgebaut. Dabei sollen die Materialien (Einzelteile) verwaltet werden können und entsprechende Listen ausgegeben werden können. Es wird eine Datenbank mit WEB Interface aufgebaut.

## Umsetzung

Das Projekt wurde vom 12.02.2020 bis 28.02.2020 durchgeführt. Sie wurde als IPA durchgeführt und nach vorgaben von PkOrg.ch gemacht. Die gewählte Projektmethode war Hermes 5.

Insgesamte Arbeitszeit betrug: 81 Stunden.

Das Projekt wurde ausschliesslich in JavaScript geschrieben.

## Ergebnisse

Die Ergebnisse wurden innerhalb des Zeitrahmens der IPA erzielt.

Es wurden alle Anforderungen, der Aufgabenstellung erfüllt, und ist für den Einsatz im Betrieb geeinigt. Es besteht eine Benutzeranleitung, welches im Anhang auffindbar ist. Die Test wurden ohne grosse Mangeln erfolgreich durchgeführt.

Inhaltsverzeichnis

[Teil 1: IPA Dokumentation 1](#_Toc33750770)

[Kurzfassung des IPA Berichtes 2](#_Toc33750771)

[Informationen 2](#_Toc33750772)

[Grobe Ausgangssituation 2](#_Toc33750773)

[Umsetzung 2](#_Toc33750774)

[Ergebnis 2](#_Toc33750775)

[2 Aufgabenstellung 10](#_Toc33750776)

[2.1 Titel der Arbeit 10](#_Toc33750777)

[2.2 Ausgangslage 10](#_Toc33750778)

[2.3 Detaillierte Aufgabenstellung 10](#_Toc33750779)

[2.3.1 Anwendungsfälle 10](#_Toc33750780)

[2.3.2 Ziele: 11](#_Toc33750781)

[2.3.3 Funktionale Anforderungen: 11](#_Toc33750782)

[2.3.4 Nichtfunktionale Anforderungen: 12](#_Toc33750783)

[2.4 Mittel und Methoden 12](#_Toc33750784)

[2.5 Vorkenntnisse 12](#_Toc33750785)

[2.6 Vorarbeiten 12](#_Toc33750786)

[2.7 Neue Lerninhalte 12](#_Toc33750787)

[2.8 Arbeiten in den letzten 6 Monaten 13](#_Toc33750788)

[3 Standards 14](#_Toc33750789)

[4 IPA-Schutzbedarfsanalyse 15](#_Toc33750790)

[4.1 Zugriff auf lokale Computer 15](#_Toc33750791)

[4.2 Zugriff auf GitHub 15](#_Toc33750792)

[4.3 IPA Daten 15](#_Toc33750793)

[5 Organisation der IPA Ergebnisse 16](#_Toc33750794)

[5.1 Arbeitsumgebung 16](#_Toc33750795)

[5.1.1 Arbeitsplatz 16](#_Toc33750796)

[5.1.2 Laptop 16](#_Toc33750797)

[5.2 Datensicherung Der IPA 17](#_Toc33750798)

[5.2.1 Filestruktur der Gespeicherten Daten 17](#_Toc33750799)

[5.2.2 Datensicherung der IPA 17](#_Toc33750800)

[5.2.3 Namenskonzept der Gespeicherten Daten 17](#_Toc33750801)

[5.2.4 Wiederherstellung 17](#_Toc33750802)

[5.2.5 Test der Wiederherstellung von Dokumenten 18](#_Toc33750803)

[6 Projektvorgehen 19](#_Toc33750804)

[6.1 Projektvorgehensmodell 19](#_Toc33750805)

[6.2 Phasen 20](#_Toc33750806)

[6.3 Abweichungen 20](#_Toc33750807)

[6.4 Meilensteine 21](#_Toc33750808)

[7 IPA Projektorganisation 21](#_Toc33750809)

[7.1 Projekt Organigramm 21](#_Toc33750810)

[7.2 Projektrollen 22](#_Toc33750811)

[8 Risikoanalyse 23](#_Toc33750812)

[8.1 Legende 24](#_Toc33750813)

[8.1.1 Schadensausmass 24](#_Toc33750814)

[8.1.2 Eintrittswahrscheinlichkeit 24](#_Toc33750815)

[8.2 Risikograph 25](#_Toc33750816)

[8.2.1 vor Massnahmen 25](#_Toc33750817)

[8.2.2 Nach Massnahmen 26](#_Toc33750818)

[9 Zeitplan 27](#_Toc33750819)

[9.1 Legende 28](#_Toc33750820)

[10 Arbeitsjournale 29](#_Toc33750821)

[10.1 Erster Tag: Mittwoch 12.02.2020 29](#_Toc33750822)

[10.2 Zweiter Tag: Donnerstag 13.02.2020 31](#_Toc33750823)

[10.3 Dritter Tag: Freitag 14.02.2020 33](#_Toc33750824)

[10.4 Vierter Tag: Montag 17.02.2020 (Halbtag) 35](#_Toc33750825)

[10.5 Fünfter Tag: Mittwoch 19.02.2020 36](#_Toc33750826)

[10.6 Sechster Tag: Donnerstag 20.02.2020 37](#_Toc33750827)

[10.7 Siebter Tag: Freitag 21.02.2020 38](#_Toc33750828)

[10.8 Achter Tag: Montag 24.02.2020 (Halbtag) 40](#_Toc33750829)

[10.9 Neunter Tag: Mittwoch 26.02.2020 41](#_Toc33750830)

[10.10 Zehnter Tag: Donnerstag 27.02.2020 42](#_Toc33750831)

[10.11 Elfter Tag: Freitag 28.02.2020 44](#_Toc33750832)

[11 Abschlussbericht 45](#_Toc33750833)

[11.1 Vergleich Ist/Soll (Anforderungen, Zeit, Einsatzmittel) 45](#_Toc33750834)

[11.2 Fazit zur IPA 45](#_Toc33750835)

[11.3 Persönliches Fazit 45](#_Toc33750836)

[11.4 Schlussreflexion 45](#_Toc33750837)

[Teil 2: Projekt-Dokumentation 46](#_Toc33750838)

[11.5 Einführung 46](#_Toc33750839)

[12 Initialisierung 47](#_Toc33750840)

[12.1 IST-Situation 47](#_Toc33750841)

[12.1.1 Abgrenzungen 47](#_Toc33750842)

[12.2 SOLL-Situation 47](#_Toc33750843)

[12.2.1 Systemziele 48](#_Toc33750844)

[12.2.2 Systemanforderungen 49](#_Toc33750845)

[12.3 Vorgehensziele 51](#_Toc33750846)

[13 Variantenvergleich 51](#_Toc33750847)

[14 Konzept 52](#_Toc33750848)

[14.1 Namenskonzept 52](#_Toc33750849)

[14.1.1 Benutzer 52](#_Toc33750850)

[14.1.2 Datenbank Relationen 52](#_Toc33750851)

[14.2 Mockups 53](#_Toc33750852)

[14.2.1 Login Seite 53](#_Toc33750853)

[14.2.2 Start Seite Lehrpersonen 54](#_Toc33750854)

[14.2.3 Lehrer Startseite Menu 54](#_Toc33750855)

[14.2.4 Schüler Startseite 55](#_Toc33750856)

[14.2.5 Schüler Startseite Menu 55](#_Toc33750857)

[14.2.6 Lehrer Neues Material erfassen 56](#_Toc33750858)

[14.2.7 Inventarliste Lehrer 56](#_Toc33750859)

[14.2.8 Ausleihungen 57](#_Toc33750860)

[14.2.9 Inventarliste Schüler 57](#_Toc33750861)

[14.3 Datenbank Konzept 58](#_Toc33750862)

[14.3.1 Konzept Datenbank in der ersten Normalform 59](#_Toc33750863)

[14.3.2 Konzept Datenbank in der zweiten Normalform 59](#_Toc33750864)

[14.3.3 Konzept Datenbank in der dritten Normalform 60](#_Toc33750865)

[14.3.4 Verifizierung 60](#_Toc33750866)

[14.4 Programmstruktur 61](#_Toc33750867)

[14.4.1 Entwicklungsumgebung Architektur 61](#_Toc33750868)

[14.4.2 Frontend Package Diagramm 61](#_Toc33750869)

[14.4.3 Back-End Package Diagramm 62](#_Toc33750870)

[14.4.4 konzeptionelles Login verfahren 62](#_Toc33750871)

[14.5 Vue 63](#_Toc33750872)

[14.5.1 Vuex 63](#_Toc33750873)

[14.5.2 Vue-router 63](#_Toc33750874)

[14.5.3 Vue Dateien 63](#_Toc33750875)

[14.6 Konzeptionelle Schnittstellen 64](#_Toc33750876)

[14.6.1 Diagramm 64](#_Toc33750877)

[14.6.2 öffentliche Routen 65](#_Toc33750878)

[14.6.3 Schüler Routen 65](#_Toc33750879)

[14.6.4 Lehrer Routen 65](#_Toc33750880)

[14.7 Testkonzept 66](#_Toc33750881)

[14.7.1 Testdaten 66](#_Toc33750882)

[14.7.2 Fehlerklassen 67](#_Toc33750883)

[14.7.3 Testarten 67](#_Toc33750884)

[14.7.4 Testvorgehen 67](#_Toc33750885)

[14.7.5 Testziele 67](#_Toc33750886)

[14.7.6 Testkategorien 68](#_Toc33750887)

[14.7.7 Testvorlage 68](#_Toc33750888)

[14.7.8 Testfälle 69](#_Toc33750889)

[15 Realisierung 76](#_Toc33750890)

[15.1 Projektumgebung 76](#_Toc33750891)

[15.1.1 API (Backend) 76](#_Toc33750892)

[15.1.2 Frontend 77](#_Toc33750893)

[15.1.3 Datenbank 78](#_Toc33750894)

[15.2 Datenbank 79](#_Toc33750895)

[15.2.1 Aufbau 79](#_Toc33750896)

[15.2.2 Abweichungen vom Konzept 80](#_Toc33750897)

[15.2.3 Testdaten 81](#_Toc33750898)

[15.2.4 Datenbankschema 82](#_Toc33750899)

[15.2.5 Beziehungen 83](#_Toc33750900)

[15.3 design 83](#_Toc33750901)

[15.3.1 pug.js 83](#_Toc33750902)

[15.3.2 Material erfassen 84](#_Toc33750903)

[15.3.3 Material ausleihen 88](#_Toc33750904)

[15.3.4 Nachrichten 90](#_Toc33750905)

[15.3.5 Material anzeigen 92](#_Toc33750906)

[15.4 Programmstruktur 93](#_Toc33750907)

[15.4.1 Frontend Package Diagramm 93](#_Toc33750908)

[15.4.2 Backend Package Diagramm 94](#_Toc33750909)

[15.5 API 95](#_Toc33750910)

[15.5.1 API Flowchart 95](#_Toc33750911)

[15.5.2 Öffentliche Schnittstellen 95](#_Toc33750912)

[15.5.3 Schüler Schnittstellen 96](#_Toc33750913)

[15.5.4 Lehrkraft Schnittstellen 96](#_Toc33750914)

[15.6 Sicherheit 97](#_Toc33750915)

[15.6.1 Passwörter 97](#_Toc33750916)

[15.6.2 Login 97](#_Toc33750917)

[15.6.3 Benutzer Autorisierung 98](#_Toc33750918)

[15.6.4 Frontend Routeguarding 100](#_Toc33750919)

[15.6.5 Backend Routeguarding 102](#_Toc33750920)

[15.7 Testfälle 103](#_Toc33750921)

[15.7.1 Testfall 1 103](#_Toc33750922)

[15.7.2 Testfall 2 104](#_Toc33750923)

[15.7.3 Testfall 3 105](#_Toc33750924)

[15.7.4 Testfall 4 106](#_Toc33750925)

[15.7.5 Testfall 5 107](#_Toc33750926)

[15.7.6 Testfall 6 108](#_Toc33750927)

[15.7.7 Testfall 7 109](#_Toc33750928)

[15.8 Nachfolgearbeiten 110](#_Toc33750929)

[15.8.1 Implementierung 110](#_Toc33750930)

[15.8.2 Ausbau der Applikation 110](#_Toc33750931)

[Teil 3: (Formaler Teil 2) 111](#_Toc33750932)

[16 Versioniserungs Nachweis 111](#_Toc33750933)

[17 Abbildungsverzeichnis 112](#_Toc33750934)

[18 Tabellenverzeichnis 114](#_Toc33750935)

[19 Informationsquellen 116](#_Toc33750936)

[20 Abkürzungsverzeichnis 117](#_Toc33750937)

[21 Glossar 118](#_Toc33750938)

[22 Phasen Freigabe 119](#_Toc33750939)

# Aufgabenstellung

## Titel der Arbeit

Inventar-Applikation

## Ausgangslage

In der Abteilung Informatik sind verschiedenste Materialien für den Unterricht und die Werkstatt vorhanden. Diese Materialien sind über mehrere Standorte und Schränke verteilt. Die Materialien sind jedoch nicht inventarisiert und es gibt auch keine "Ausleih-Verwaltung". Mit dieser neuen Webapplikation soll den Nutzern der Materialien eine elektronische Verwaltung dieser Materialien ermöglicht werden.

## Detaillierte Aufgabenstellung

Es wird eine Applikation für die Inventarisierung der vorhandenen Lehr- und Lernmaterialien für die Lehrkräfte und die Lernenden aufgebaut. Dabei sollen die Materialien (Einzelteile) von den Lehrkräften verwaltet werden können (Neu – Mutation – Löschen). Die Standorte der Materialien werden angegeben. Das Material kann sowohl von den Lehrkräften wie auch den Lernenden ausgeliehen werden. Wird es nicht mehr benötigt kann es wieder freigegeben werden. Es wird eine Datenbank mit WEB Interface aufgebaut, auf den man mit einem Login zugreifen kann.

Die Datensicherung, die Benutzerverwaltung und die Standortverwaltung sind nicht Bestandteil dieser IPA! Es werden lediglich Testnutzer erstellt.

### Anwendungsfälle

1. Eine Lehrkraft kann sich mit Benutzername und Passwort ins Programm einloggen.
2. Eine Lehrkraft kann ein vorhandenes Lehr- bzw. Lernmaterial erfassen.
3. Die Daten eines bereits erfassten Lehr- bzw. Lernmaterials können von einer Lehrkraft verändert werden.
4. Ein Lehr- bzw. Lernmaterial kann von einer Lehrkraft gelöscht werden.
5. Eine Lehrkraft kann ein Lehr- bzw. Lernmaterial ausleihen und als ausgeliehen markieren und wenn er es wieder zurückbringt als nicht ausgeliehen markieren.
6. Eine Lehrkraft kann eine Lagerliste für die verfügbare Menge eines Materials abfragen.
7. Eine Lehrkraft kann eine Ausleihliste, welche für einen bestimmten Benutzer angibt, welche Materialien er/sie ausgeliehen hat, abfragen.
8. Eine Lernende / ein Lernender kann sich mit Benutzername und Passwort ins Programm einloggen.
9. Eine Lernende / ein Lernender kann ein Lehr- bzw. Lernmaterial ausleihen und als ausgeliehen markieren und wenn sie/er es wieder zurückbringt als nicht ausgeliehen markieren.
10. Eine Lernende / ein Lernender kann eine Ausleihliste abfragen, welche ihre/seine ausgeliehenen Materialien anzeigt.

### Ziele:

1. Es ist ein Login implementiert, welches Lernende und Lehrkräfte unterscheidet.
2. Es besteht ein Webinterface, welches den Lehrkräften ermöglicht, Materialien neu zu erfassen, zu mutiert und zu löschen, sowie Abfragen zu den Materialien und Ausleihe von Materialien vorzunehmen. Das Interface ermöglicht den Lernenden Abfragen zu den Materialien und Ausleihe von Materialien vorzunehmen.
3. Zu den Materialien können Nummer, Bezeichnung, Hersteller, Typ, Beschreibung/Eigenschaften, Seriennummer, Menge und Standort erfasst werden.
4. Es kann eine Lagerliste, welche die verfügbare Menge eines Materials angibt und eine Ausleihliste, welche für einen bestimmten Benutzer angibt, welche Materialien er/sie ausgeliehen hat, auf dem Bildschirm ausgegeben werden.
5. Die Tabellen der DB werden mittels SQL-Script erstellt; die Datenbank ist in der 3NF.
6. Webserver und MySql werden für die IPA lokal installiert.
7. Es besteht eine Benutzeranleitung zur Verwendung der Inventar-Applikation.

### Funktionale Anforderungen:

1. Je drei Testnutzer aus den beiden Benutzergruppen für die Lehrkräfte und Lernenden sollen mit Namen und Passwort per Script in der Datenbank erstellt werden (Z1).
2. Die Lehrkraft kann sich mit Benutzername und Passwort anmelden und erhält die entsprechenden Rechte (Z1)
3. Die Lernende / der Lernende kann sich mit Benutzername und Passwort anmelden und erhält die entsprechenden Rechte (Z1)
4. Die Passwörter werden als gehashte Strings in der Datenbank abgelegt (Z1)
5. Eine Lehrkraft kann Materialien über das Web-Interface erfassen (Z2)
6. Eine Lehrkraft kann Materialien über das Web-Interface löschen (Z2)
7. Eine Lehrkraft kann Materialien über das Web-Interface mutieren (Z2)
8. Eine Lehrkraft kann Materialien über das Web-Interface ausleihen. Dabei wird das Material als ausgeliehen markiert und mit seinem Namen verknüpft(Z2)
9. Eine Lernende/ein Lernender Materialien über das Web-Interface ausleihen. Dabei wird das Material als ausgeliehen markiert und mit seinem Namen verknüpft(Z2)
10. Zu den Materialien sollen die folgenden Attribute erfasst werden können: Nummer, Bezeichnung, Hersteller, Typ, Beschreibung/Eigenschaften, Seriennummer und Standort. (Z3)
11. Es kann, sowohl von Lehrkräften wie auch Lernenden, eine Lagerliste in der Applikation bzw. auf dem Bildschirm ausgegeben werden, welche die verfügbare (nicht ausgeliehene) Menge eines Materials angibt. (Z4)
12. Die Lehrkraft kann eine Ausleihliste in der Applikation bzw. auf dem Bildschirm ausgegeben lassen, welche für einen Benutzer angibt, welche Materialien er/sie ausgeliehen hat. Dabei kann die Lehrkraft für allen Benutzer die Auswahlliste ausgeben lassen. (Z4)
13. Die Lernende/der Lernender kann für sich selbst eine Ausleihliste in der Applikation bzw. auf dem Bildschirm ausgegeben lassen, welche anzeigt, welche Materialien sie/er ausgeliehen hat. (Z4)
14. Die Tabellen der DB werden mittels SQL-Script erstellt (Z5)
15. Der Webserver ist lokal installiert (Z6)
16. Die Datenbank ist lokal installiert (Z6)
17. Es besteht eine Benutzeranleitung mit den Themen Login, Erfassen von Materialien, Mutieren von Materialien, Löschen von Materialien, Ausleihen von Materialien, Ausgabe der Lagerliste und Ausleihliste (Z7)

### Nichtfunktionale Anforderungen:

1. Die Lagerliste und die Ausleihliste werden in der Applikation bzw. auf dem Bildschirm ausgegeben. (Z4)
2. Die Datenbank ist in der 3. NF (Z5)
3. Die Applikation wird mit JavaScript erstellt (Z6)
4. Als Datenbank wird MySql verwendet (Z6)
5. Die Benutzeranleitung ist als separate Datei realisiert und nicht in die Applikation integriert. (Z7)

## Mittel und Methoden

Projektmethode:

* Als Projektmethode wird HERMES 5 angewandt.

Technologie-Stack:

* Frontend : HTML, CSS, JS (Framework: Vue.js)
* Back-End : Node.js ,erweitert durch express.js & knex.js, Bcrypt, JWT (JsonWebToken)

DB: MySQL

Es gelten die Coding Conventions der ICT Berufsbildung.

Zur Entwicklung steht ein Laptop (Windows 10) mit installierter Entwicklungsumgebung (Visual Studio Code) zur Verfügung.

* Node.js ist lokal installiert.
* MySQL und MySql Workbench 8.0 CE sind lokal installiert.

Dies wird vorgängig zur IPA von dem Lernenden installiert bzw. bereitgestellt.

## Vorkenntnisse

HTML, CSS, JavaScript, Vue.js, SQL

## Vorarbeiten

Einarbeitung in Bcrypt und JWT

## Neue Lerninhalte

* Vertiefung Vue.js/Vuetify
* gehashte Passwörter in DB Speichen

## Arbeiten in den letzten 6 Monaten

Quiz-App erstellen mit SQLite und knex/vue/express Webauftritt erstellen DB in Web-App einbinden

# Standards

Die Technische Fachschule Bern besitzt keine Firmenstandards. Für die IPA werden deshalb die Coding-Conventions, die auf PkOrg auffindbar sind beachtet.

|  |  |
| --- | --- |
| Standard | Beschreibung |
| Dokumentvorlage | Dieses Dokument und ihre Beilagen wurden nach den Vorgaben von Pkorg.ch erstellt |
| Code/ Skripte / Kommentare | Aller selbsterstellte Code ausschnitte und Skripte werden nach den Code-Conventions, die auf Pkorg.ch auffindbar sind erstellt. |
| Sicherheitskonzept | Die Informatik Abteilung an der Technischen Fachschule Bern verfügen über keine Sicherheitskonzept. Ein eigenes Sicherheitskonzept wird deshalb für das Projekt verwendet |
| Projektmethode | Die Technische Fachschule Bern verfügt über keine eigene Projektmethode. Die Entscheidung liegt bei dem Projektleiter |

Tabelle 3-1: Standards

# IPA-Schutzbedarfsanalyse

## Zugriff auf lokale Computer

An meinen Laptop, der kann nur mit einem Passwort eingeloggt werden.

## Zugriff auf GitHub

Alle IPA Daten werden mit GitHub versioniert und gesichert Weitere Informationen sind Im Kapitel Organisation der IPA Ergebnisse.

## IPA Daten

Sämtliche Daten der IPA werden nur autorisierten Personen durch GitHub zur Verfügung gestellt. Die Daten werden täglich gesichert. Weitere Informationen Dazu sind im Kapitel Organisation der IPA Ergebnisse zu finden.

# Organisation der IPA Ergebnisse

## Arbeitsumgebung

### Arbeitsplatz

Während der ganze IPA wird an den folgenden Arbeitsplatz gearbeitet.



### Laptop

|  |  |  |
| --- | --- | --- |
| Bild | Eigenschaften | |
|  | Name | Lenovo ThinkPad E470 |
| Betriebssystem Architektur | 64 Bit |
| Betriebssystem Version | Windows 10 Pro |
| Speicherkapazität SSD/Flash | 256 GB |
| Daten Anschlüsse | 2 x USB 3.0, USB 2.0 und RJ-45 (LAN) |

Tabelle 5-1: Laptop

## Datensicherung Der IPA

### Filestruktur der Gespeicherten Daten



Abbildung 1: Filestruktur der Gespeicherten Daten

### Datensicherung der IPA

Alle IPA Dokumente und Projekt Dateien werden mittels Git versioniert und auf GitHub, als Back-up hochgeladen. Das Back-up wird manuell zwei Mal Täglich erstellet. Am Ende von jedem Tag wird ein Branch erstellt und auf GitHub hochgeladen.

Der Zugriff auf die Daten ist geschützt durch ein Passwort, das nur den Projektleiter bekannt ist.

### Namenskonzept der Gespeicherten Daten

Alle Dateien werden gekennzeichnet durch den Namen der Datei, den Letzt bekannten Änderungsdatum, und die Aktuelle Version der Datei. Dabei ist jeder Abschnitt mit einem Unterstrich getrennt.

Das Datum muss folgendes Format **DD.MM.YYYY** entsprechen.

**Beispiel**:

**IPA-Dokumentation-beispiel\_12.02.2020\_v01.xyz**

### Wiederherstellung

Die Wiederherstellung von Projekt daten erfolgt durch GitHub oder per Konsole. Dazu ist ein Passwort benötigt.

### Test der Wiederherstellung von Dokumenten

#### Wiederherstellung durch Konsole



Abbildung 2: Test der Wiederherstellung von Dokumenten

C:\Users>git Clone git@github.com:HoloArcher/IPA2020\_Dokumentation.git

Mit Git Clone kann das Verzeichnis wiederhergestellt werden

#### Wiederherstellung durch Github.com



Abbildung 3: Wiederherstellung durch Github.com

# Projektvorgehen

## Projektvorgehensmodell

Dieses Projekt richtet sich nach der Projektmethode HERMES 5.

Abbildung 4: Projektmethode

## Phasen

|  |  |
| --- | --- |
| Phase | Beschreibung |
| Initialisierung | Die Initialisierung schafft eine definierte Ausgangslage für das Projekt und stellt sicher, dass die Projektziele mit den Zielen und Strategien der Organisation abgestimmt sind. Die Projektgrundlagen und der Projektauftrag werden erarbeitet und der Entscheid zur Projektfreigabe wird getroffen |
| Konzept | Die in der Phase Initialisierung gewählte Variante wird konkretisiert. Die Ergebnisse werden so detailliert erarbeitet, dass die Projektbeteiligten das Produkt bzw. das IT-System auf einer verlässlichen Grundlage planen, offerieren und realisieren können. |
| Realisierung | Das Produkt bzw. das IT-System werden realisiert und getestet. Die nötigen Vorarbeiten werden geleistet, um die Einführungsrisiken zu minimieren. |
| Einführung | Der sichere Übergang vom alten zum neuen Zustand wird gewährleistet. Der Betrieb wird aufgenommen und so lange durch das Projekt unterstützt, bis er stabil ist. |

Tabelle 6-1: Hermes Phasen

## Abweichungen

Die Phase Einführung ist nicht Bestandteil dieser IPA.

## Meilensteine

|  |  |
| --- | --- |
| Nr. | Meilensteine |
| 0 | Start der IPA |
| 1 | Freigabe Initialisierungsphase |
| 2 | Teil 1 - Administratives abgeschlossen |
| 3 | Freigabe Konzeptphase |
| 4 | Freigabe Realisierungsphase |
| 5 | Projektabschluss |

Tabelle 6-2: Meilensteine

# IPA Projektorganisation

## Projekt Organigramm

Abbildung 5: Organigramm

## Projektrollen

|  |  |
| --- | --- |
| Person / Rolle | Kontaktdaten |
| Auftraggeber | Technische Fachschule Bern  Lorrainestrasse 3  3013 Bern  Telefon: 031 337 37 37  E-Mail: info@tfbern.ch |
| VEX | Hurst Michael |
| HEX | Erich Reto Iseli |
| NEX | Enrico Buchs |
| Berufsbildner | Giulio Iannattone  Telefon: 031 337 38 28  E-Mail: giulio.iannattone@tfbern.ch |
| Verantwortliche Fachkraft | Stefano Amati |
| Tester | Jonas Gerber  Hezekiah Bakare |
| Projektleiter | Josiah Schiess |
| Fachspezialist |

Tabelle 7-1: Projektrollen

# Risikoanalyse

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nr. | Risikobeschreibung | Auswirkung | Vor Massnahme | | Massnahmen | Nach Massnahme | |
| Schadensausmass | Eintrittswahrscheinlichkeit | Schadensausmass | Eintrittswahrscheinlichkeit |
| R1 | Zeitmangel | Das Projekt kann nicht rechtzeitig fertiggestellt werden, oder ist mangelhaft | **S4** | **W3** | Einen Realistischen und konkrete Zeitplan erstellen.  Mit Überstunden kompensieren.  Im Zeitplan ungeplante stunden miteinrechnen | **S2** | **W2** |
| R2 | Krankheit / Unfall | Wegen unvorhersehbarer Krankheit kann die nötige Arbeit nicht geleistet werden | **S4** | **W2** | Sofort den Hauptexperten melden und sich ein Zeugnis vom Arzt besorgen. Folgendes Vorgehen mit dem Hauptexperten besprechen | **S1** | **W2** |
| R3 | Datenverlust | Die Aktuelle Version der Dokumente oder der Projektdaten kann nicht aufgefunden werden | **S4** | **W2** | Durch eine Konkrete und Einheitliche Back-up Konzept werden die Backups Halbtag erstellt. | **S2** | **W1** |
| R4 | Abbruch der Internetverbindung | Ressourcen aus den Internet Fehlen | **S3** | **W2** | Eine Verbindung mit dem Mobiltelefon kann hergestellt bis das Problem erhoben ist. | **S1** | **W2** |
| R5 | Systemausfall | Aufgrund eines Systemausfalls kann die IPA nicht fortgeführt bzw. nicht pünktlich abgeschlossen werden. | **S3** | **W2** | Bei einem auftretenden Systemausfall wird unverzüglich der IPA HEX informiert.  Es wird sofort den Chefexperte gemeldet, um weiteres Vorgehen zu besprechen | **S1** | **W2** |

Tabelle 8-1: Risikoanalyse

## Legende

### Schadensausmass

|  |  |
| --- | --- |
| Abkürzung | **Beschreibung** |
| **S1** | führt zu keiner Abwertung |
| **S2** | geringe Abwertung |
| **S3** | hohe Abwertung |
| **S4** | führt zu nicht bestehen |

Tabelle 8-2: Schadensausmass

### Eintrittswahrscheinlichkeit

|  |  |
| --- | --- |
| Abkürzung | **Beschreibung** |
| **W1** | unvorstellbar |
| **W2** | unwahrscheinlich |
| **W3** | eher vorstellbar |
| **W4** | wahrscheinlich |
| **W5** | sehr wahrscheinlich |

Tabelle 8-3: Eintrittswahrscheinlichkeit

## Risikograph

### vor Massnahmen

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Eintrittswahrscheinlichkeit | Sehr wahrscheinlich | |  |  |  |  |
| **Wahrscheinlich** | |  |  |  |  |
| **vorstellbar** | |  |  |  | R1 |
| **Unwahrscheinlich** | |  |  | R4, R5 | R2, R3 |
| **Unvorstellbar** | |  |  |  |  |
|  | |  | **Keine Abwertung** | **geringe Abwertung** | **hohe Abwertung** | **führt zu Nichtbestehen** |
| Schadensausmass | | | | | | | |

Tabelle 8-4: Risikograph vor Massnahmen

### Nach Massnahmen

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Eintrittswahrscheinlichkeit | Sehr wahrscheinlich | |  |  |  |  |
| **Wahrscheinlich** | |  |  |  |  |
| **vorstellbar** | |  |  |  |  |
| **Unwahrscheinlich** | | R2, R4, R5 | R1 |  |  |
| **Unvorstellbar** | |  | R3 |  |  |
|  | |  | **Keine Abwertung** | **geringe Abwertung** | **hohe Abwertung** | **führt zu Nichtbestehen** |
| Schadensausmass | | | | | | | |

Tabelle 8-5: Risikograph nach Massnahmen

# Zeitplan

## Legende

|  |  |
| --- | --- |
| Legende | |
|  | Soll Zeit |
|  | Ist zeit |
|  | Weniger als soll |
|  | Überstunden |
| VM | Vormittag |
| NM | Nachmittag |
|  | Meilenstein |

Tabelle 9-0-1: Legende

# Arbeitsjournale

## Erster Tag: Mittwoch 12.02.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Beteiligte Personen** | **Aufwand geplant (Std)** | **Aufwand effektiv (Std)** |
| **Sitzung mit Verantwortliche Fachkraft** | Josiah Schiess  Stefano Amati | 1 | 1 |
| **Zeitplan** | Josiah Schiess | 2 | 2 |
| **Erste Version des Dokuments** | Josiah Schiess | 1 | 1 |
| **Standards, IPA-Schutzbedarfanalyse** | Josiah Schiess | 1 | 1 |
| **Risikoanalyse, Aufgabenstellung** | Josiah Schiess | 1 | 1 |
| **Projektvorgehen, Projektorganisation** | Josiah Schiess | 1 | 1 |
| **Organisation Der IPA Ergebnisse** | Josiah Schiess | 1 | 1 |
| **Arbeitsjournal, Backup erstellen** | Josiah Schiess | 1 | 1 |
| ***Aufwand Total*** |  | **9** | **9** |
| **Tages Ablauf** | | | |
| In der Startsitzung mit meine Verantwortliche Fachperson haben wir den Ablauf der IPA besprochen. Dazu auch die Meilensteine angesprochen und wir haben abgemacht, dass ein Phasenfreigabe Blatt erstellt werden soll. Die Sitzungstermine wurden auch angesprochen.  Nach der Sitzung fing ich mit dem Zeitplan an. Ich habe versucht die Aufgabenpakete nach den Vorgaben, die auf PkOrg auffindbar sind zu erstellen. Dann habe ich den formellen teil ausgefüllt. | | | |
| **Erfolge und Misserfolge** | | | |
| Ein Problem ist bei der Word Formatierung aufgetaucht. Wenn eine Tabelle, über eine schon vorhandene Tabelle mit einer Beschriftung, in dem Dokument hinzugefügt wird, wird die Beschriftung nicht automatisch erneuert. | | | |
| **Erfolge und Misserfolge** | | | |
| **Lösung:**  Nach einige Minuten Recherche habe ich die folgende Seite gefunden die, mir die Lösung zu mein Problem gegeben hat.  <https://www.officetooltips.com/word_2016/tips/adding_captions_with_autocaption.html>  Die Lösung, die ich online gefunden hab ist in Word eingebaut als Auto Beschriftung. So kann ich nachträglich über schon bestehende Tabellen hinzufügen ohne alle Beschriftungen umbeschreiben zu müssen. | | | |
| **Reflexion** | | | |
| Meine Arbeit ist gut vorangekommen. Ich habe dank einer guten Vorlage nicht sehr viel Zeit an den Zeitplan verbracht und konnte schnell anfangen. Auch bei der Ersten Version von der Dokumentation hatte ich eine MS Word Vorlage mit Standardisierte Header und Table design. Dank der Gesparten Zeit konnte ich Teil 1, mit Ausnahme vom Abschlussbericht am ersten Tag abschliessen.  Die Aufgabenstellung von PkOrg übertragen hat länger gebraucht als anfangs gedacht habe. Wegen den Format unterschied von HTML und MS Word. Ich habe dann später gemerkt das, man mit Regular-Expression die Zeilenseparatoren ersetzen könnte, um die Tabellen zu generieren. | | | |
| **Weiteres Vorgehen** | | | |
| Fachgespräch mit experten, Anfangen Teil 2: Projektdokumentation | | | |

Tabelle 10-1: Erster Tag: Mittwoch 12.02.2020

## Zweiter Tag: Donnerstag 13.02.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Beteiligte Personen** | **Aufwand geplant (Std)** | **Aufwand effektiv (Std)** |
| **Expertenbesuch** | Iseli Erich Reto  Stefano Amati  Josiah Schiess | 1 | 2 |
| **Zeitplan** | Josiah Schiess | 0 | 1 |
| **Einführung** | Josiah Schiess | 1 | 1 |
| **IST / SOLL Situation** | Josiah Schiess | 2 | 1 |
| **Vorgehensziele, Variantenvergleich** | Josiah Schiess | 1 | 1 |
| **Arbeitsjournal, Backup erstellen** | Josiah Schiess | 1 | 1 |
| **Sitzung mit Verantwortliche Fachkraft** | Josiah Schiess | 1 | 1 |
| **Namenskonzept** | Josiah Schiess | 1 | 0 |
| ***Aufwand Total*** |  | **8** | **8** |
| **Tages Ablauf** | | | |
| Der Tag fing mit dem ersten Expertenbesuch an. Es ging länger als erwartet. Was genau besprochen wurde, ist im Sitzungsprotokoll im Anhang aufgeführt. Anschliessend machte ich mich direkt an die Arbeit, meinen Zeitplan zu überarbeiten. | | | |
| **Erfolge und Misserfolge** | | | |
| Nach den Expertenbesuch habe ich meinen Zeitplan angepasst und das Sitzungsprotokoll nachgetragen. Dies hat dazu geführt, dass ich ca. zwei Stunden in Verzug kam.  Zurzeit sind keine Varianten ersichtlich. Gemäss Experten wäre es möglich ohne Variantenvergleich fortzufahren. Falls sich allerdings in der weitern Arbeit allfälligen Varianten ergeben würden kann dies noch ergänzt werden. Dadurch das der Variantenvergleich aktuell wegfiel, konnte ich ca. eine Stunde wiedergutmachen.  Trotz der Zeitverzögerung konnte ich die Initialisierungsphase Abschliessen und mit meiner Verantwortliche Fachkraft die Freigabe machen.  Ich hatte Schwierigkeiten die Einführung zu Schreiben da, mir nicht im Klaren war was genau vorkommen muss. Ich habe dann kurz einen Arbeitskollegen gefragt, ob es wüsste was gemeint ist. | | | |
| **Reflexion** | | | |
| Der Expertenbesuch hat mir gross geholfen. Das Feedback zu meinem Zeitplan war sehr hilfreich. Ich habe gewisse Arbeitsschritte aufgeteilt zu kleineren Einheiten und andere zusammengefügt, um die Übersicht des Projekts zu verbessern. | | | |
| **Weiteres Vorgehen** | | | |
| Start der Phase: Konzept. | | | |

Tabelle 10-2: Zweiter Tag: Donnerstag 13.02.2020

## Dritter Tag: Freitag 14.02.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Beteiligte Personen** | **Aufwand geplant (Std)** | **Aufwand effektiv (Std)** |
| **Namenskonzept** | Josiah Schiess | 0 | 1 |
| **Mockups** | Josiah Schiess | 3 | 3 |
| **Datenbank Konzept / Diagramm** | Josiah Schiess | 3 | 3 |
| **Programm-Struktur** | Josiah Schiess | 1 | 0 |
| **Arbeitsjournal, Backup erstellen** | Josiah Schiess | 1 | 1 |
| ***Aufwand Total*** |  | **8** | **8** |
| **Tages Ablauf** | | | |
| Ich habe mit der Erstellung von Mockup angefangen. Diese erfolgten mit Adobe XD. Danach habe ich das Namenskonzept erstellt. Mit dem Namenskonzept erledigt, konnte ich mich an das Datenbankschema machen. Ich habe Diagramm in Lucidchart erstellt. Da ich immer noch im Verzug bin konnte ich noch nicht mit der Programm-Struktur anfangen. | | | |
| **Erfolge und Misserfolge** | | | |
| Als ich den bei der Datenbankschema arbeitete, trat ich auf ein Problem bei der Normalisierungsprozesse. Es War mir nicht mehr klar im Sinne was der Unterschied zwischen der 2. Und 1. Normalform.  Ich habe online keine zufriedenstellende Erklärung gefunden. Deshalb habe ich mir Hilfe von Giulio Iannatone gesucht, der es dann auch verifiziert hat.  Ich habe gemerkt, das für mich Deutsche Fachtexte zu Schreiben sehr schwierig ist, und ich mich viel Mühe geben muss, dies zu korrigieren und in der besser zu machen. Ich werde mit der eingebauten Grammatik-Checker von Word und https://languagetool.org/ arbeiten, um hoffentlich das Kriterium abzudecken. Idealerweise würde ich jemanden die Arbeit zum Überprüfung geben. Dies wird sich wahrscheinlich schwer erweisen, da sie täglich bearbeitet wird. | | | |
| **Reflexion** | | | |
| Der Expertenbesuch hat mir viel geholfen. Das Feedback zu meinem Zeitplan war sehr hilfreich. Ich habe gewisse Arbeitsschritte aufgeteilt zu kleineren Einheiten und andere zusammengefügt, um die Übersicht des Projekts zu verbessern. | | | |
| **Weiteres Vorgehen** | | | |
| Programm-Struktur, Testkonzept | | | |

Tabelle 10-3: Dritter Tag: Freitag 14.02.2020

## Vierter Tag: Montag 17.02.2020 (Halbtag)

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Beteiligte Personen** | **Aufwand geplant (Std)** | **Aufwand effektiv (Std)** |
| **Programm-Struktur** | Josiah Schiess | 3 | 3 |
| **Testkonzept** | Josiah Schiess | 1 | 0 |
| **Arbeitsjournal, Backup erstellen** | Josiah Schiess | 1 | 1 |
| ***Aufwand Total*** |  | **4** | **4** |
| **Tages Ablauf** | | | |
| Ich startete die neue Woche mit der Programm-Struktur. Ich entschied mich für 2 Package Diagramme, welche einen Einblick in den Programmaufbau geben sollten. Diese wurden mit Lucidchart erstellt, ein gratis online Tool, welches ich zu dem Erstellen von Grafiken und Diagramme nutze. Da ich etwas früher fertig war, als ich ursprünglich gedacht hatte, habe ich mit dem Testkonzept begonnen. Da ich nur ein paar Minuten daran gearbeitet habe, beschloss ich, es nicht in den Zeitplan zu schreiben, sondern es einfach wegzulassen. | | | |
| **Erfolge und Misserfolge** | | | |
| Ich hatte zuerst geplant das Online-Tool draw.io brauchen. Doch ich merkte schnell, dass das Tool nicht für meine Zwecke sich eigenen würde. Deshalb habe ich online nach einem anderen Tool gesucht und fand so Lucidchart.  Lucidchart hat jedoch das Problem das, wenn man nicht mehr als 3 einzelne Diagramme online speichern kann.  Die Lösung, die ich fand, war alle Diagramme bei einer Workspace zu erstellen und dann nach dem Exportieren zurechtzuschneiden | | | |
| **Reflexion** | | | |
| Ich bin mit der Arbeit sehr gut vorangekommen. Für meinen Geschmack war es etwas zu laut, deshalb habe ich den größten Teil des Vormittags Kopfhörer getragen. | | | |
| **Weiteres Vorgehen** | | | |
| Testkonzept, Phasenfreigabe: Konzept | | | |

Tabelle 10-4: Vierter Tag: Montag 17.02.2020 (Halbtag)

## Fünfter Tag: Mittwoch 19.02.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Beteiligte Personen** | **Aufwand geplant (Std)** | **Aufwand effektiv (Std)** |
| **Testkonzept** | Josiah Schiess | 3 | 5 |
| **Projektumgebung** | Josiah Schiess | 1 | 1 |
| **Datenbank / Testdaten erstellen und Einfügen** | Josiah Schiess | 1 | 0 |
| **Reservezeit** | Josiah Schiess | 1 | 0 |
| **Sitzung mit verantwortlicher Fachkraft** | Josiah Schiess  Stefano Amati | 1 | 1 |
| **Arbeitsjournal, Backup erstellen** | Josiah Schiess | 1 | 1 |
| ***Aufwand Total*** |  | **8** | **8** |
| **Tages Ablauf** | | | |
| Ich verbrachte den grössten Teil des Vormittags mit der Arbeit am Testkonzept. Danach hatte ich die Phasenfreigrabe mit Stefano. Schliesslich richtete ich meine Projektumgebung ein. Technisches gab es bis jetzt noch nicht darum kann ich im Tagesablauf auch nichts nennen, das technisch bezogen ist. | | | |
| **Erfolge und Misserfolge** | | | |
| heute hatte ich Konzentrationsprobleme. ich war gestresst, weil ich mit der Konzeptphase fast fertig bin und ich das Gefühl habe, dass ich die Arbeitsmenge, die ich vor mir habe, unterschätzt habe. doch ich glaube, es sind nur meine Nerven. Ich muss mich nur auf den nächsten Schritt konzentrieren, da ich immer noch gut in der Zeit bin. | | | |
| **Reflexion** | | | |
| Während der Arbeit am Testkonzept wurde mir klar, dass ich möglicherweise unterschätzt hatte, wie viel Zeit es dauern würde, dieses Projekt abzuschließen. | | | |
| **Weiteres Vorgehen** | | | |
| Datenbankkonzept, API erstellen | | | |

Tabelle 10-5: Fünfter Tag: Mittwoch 19.02.2020

## Sechster Tag: Donnerstag 20.02.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Beteiligte Personen** | **Aufwand geplant (Std)** | **Aufwand effektiv (Std)** |
| **Datenbank / Testdaten erstellen und Einfügen** | Josiah Schiess | 3 | 4 |
| **API-Routing** | Josiah Schiess | 3 | 0 |
| **API-Logik** | Josiah Schiess | 1 | 0 |
| **Frontend UI Komponente** | Josiah Schiess | 0 | 3 |
| **Arbeitsjournal, Backup erstellen** | Josiah Schiess | 1 | 1 |
| ***Aufwand Total*** |  | **8** | **8** |
| **Tages Ablauf** | | | |
| Heute habe ich an der Realisierung meines Datenbankkonzepts gearbeitet. Nachher machte ich mich an das Frontend UI design und die Umsetzung des Konzepts. | | | |
| **Erfolge und Misserfolge** | | | |
| Der nächste Schritt wäre, mit der Arbeit an der API zu beginnen. Ich beschloss jedoch, dass ich zuerst mit dem Frontend-UI-Design beginnen würde, da ich es schwierig fand, zu entscheiden, wie die Ein- und Ausgaben der API aussehen sollten. Ich dachte, es wäre einfacher, mit dem Frontend zu beginnen, da ich mir eine Vorstellung von den Objekten machen könnte, die die UI benötigen würde. Da ich die Mockups mit Materialdesign im Hinterkopf erstellt habe, war es einfach, ein Design zu erstellen, das dem Original sehr nahekommt.  Ich habe bei Erstellen von Diagrammen gemerkt, dass Lucidchart auch eine begrenzte Anzahl kostenlose Objekt hat. Diese laufen schnell aus. Da zurzeit ich kein Mobiltelefon besitze, konnte ich nicht auf die Smartphone App von meiner Online-Bank zugreifen, um meine Kreditkarte zu brauchen. Lösung war einfach mit einem anderen Mailadresse anzumelden, um weiterhin die kostenfreie brauchen zu können. | | | |
| **Reflexion** | | | |
| Ich versuchte, die Vue-Komponenten über ein Adobe XD-Plugin zu exportieren, dies funktionierte jedoch letztendlich nicht. Trotzdem fand ich es leicht das Design zu erstellen, dank der Mockups. | | | |
| **Weiteres Vorgehen** | | | |
| UI Logik / Routing | | | |

Tabelle 10-6: Sechster Tag: Donnerstag 20.02.2020

## Siebter Tag: Freitag 21.02.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Beteiligte Personen** | **Aufwand geplant (Std)** | **Aufwand effektiv (Std)** |
| **API-Autorisierung (Login verfahren)** | Josiah Schiess | 1 | 1 |
| **API-Logik** | Josiah Schiess | 1 | 0 |
| **Frontend UI Komponente** | Josiah Schiess | 3 | 1 |
| **Frontend Routing / Logik** | Josiah schiess | 2 | 4 |
| **API-Routing** | Josiah Schiess | 0 | 1 |
| **Arbeitsjournal, Backup erstellen** | Josiah Schiess | 1 | 1 |
| ***Aufwand Total*** |  | **8** | **8** |
| **Tages Ablauf** | | | |
| heute habe ich dort weitergemacht, wo ich gestern aufgehört habe, bei der Frontend-UI-Komponenten. danach habe ich an der Logik und dem Routing der Anwendung gearbeitet. Am frühen Nachmittag habe ich das Login verfahren bei in der API erstellt.  ich bemerkte, dass es ein Problem mit einer der Ziele in der Aufgabenstellung geben könnte, und besprach mit Stefano darüber. er empfahl mir, dass bei einer Abweichung von einem gegebenen ziel es klar dokumentiert und begründend sein müsste. denn in A7 heißtes "Eine Lehrkraft kann Materialien über das Web-Interface mutieren". Aber in meiner Datenbank haben die Reale Materialien nur die Seriennummer als Attribut, und ich fragte mich, ob es einen Sinn hat, die Funktion zu dem Bearbeiten der Attribute über die Benutzeroberfläche hinzuzufügen sollte, auch wenn es tatsächlich nur einen Attribut ist, das geändert werden kann.  Ich habe mich entschieden, es beizubehalten, weil es für zukünftige Implementierungen praktisch wäre, diesen Endpunkt und diese UI-Komponente bereits zu haben, auch wenn sie in dem Zustand, in dem sie sich jetzt befindet, nicht sehr viel bewirkt.  Zum Schluss als ich das mit dem Frontend fertig war, habe ich mich an das Routing der API gemacht. | | | |
| **Erfolge und Misserfolge** | | | |
| Bei der Konzeption der API hätte ich mir mehr zeit nehmen sollen alle möglichen daten zu überlegen, die das Frontend brauchen würde. | | | |
| **Reflexion** | | | |
| Besonders bei der Konzeption der Schnittstellen hätte ich mir mehr zu jeder Schnittstelle aufschreiben sollen. Es wäre hilfreich gewesen, wenn ich die ein- und ausgaben der verschiedene Endpunkte definiert hätte. Eine ungeplante Arbeit was, dass das Objekt die manche Vuetify Komponente benötigen, um zu funktionieren, nicht so von Knex ausgegeben werden. Ich musste an einer stelle eine Funktion in dem Backend schreiben, welch die Resultate von Knex umformatiert und dann ausgibt. | | | |
| **Hilfestellung** | | | |
| Für die UI design hatte ich immer die Dokumentation von Vuetify auf einen sekundären Bildschirm offen. | | | |
| **Weiteres Vorgehen** | | | |
| API-Routing / API Logik  Evtl. Frontend anpassen  Idealerweise mit Testing anfangen. | | | |

Tabelle 10-7: Siebter Tag: Freitag 21.02.2020

## Achter Tag: Montag 24.02.2020 (Halbtag)

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Beteiligte Personen** | **Aufwand geplant (Std)** | **Aufwand effektiv (Std)** |
| **API-Routing** | Josiah Schiess | 0 | 1 |
| **API-Logik** | Josiah Schiess | 0 | 2 |
| **Test durchführen** | Josiah Schiess | 1 | 0 |
| **Frontend Routing / Logik** | Josiah Schiess | 2 | 0 |
| **Arbeitsjournal, Backup erstellen** | Josiah Schiess | 1 | 1 |
| ***Aufwand Total*** |  | **4** | **4** |
| **Tages Ablauf** | | | |
| An diesen Tag habe ich am API weitergearbeitet. Ich habe die meiste Zeit mit Bug-Fixing verbracht.  Ich habe einen Fehler bei meiner Datenbank Konzept entdecket. Ich habe die Standort bei der ItemsClass definiert, dies macht aber bei keinen Sinn, da nicht alle Items in der Klasse am gleichen Standort sein werden. | | | |
| **Erfolge und Misserfolge** | | | |
| Ich habe den Fehler der Datenbank beheben können.  Ich musste wieder einen neuen Benutzer für Lucidchart erstellen da ich das Objekt-Limit erreicht habe. | | | |
| **Reflexion** | | | |
| Wie erhofft, hatte ich keine grossen Probleme beim Erstellen des API. Schwieriger war es die API mit dem Frontend zu verbinden. Dies hätte ich besser bei dem Zeitplan einrechnen können. | | | |
| **Hilfestellung** | | | |
| keine | | | |
| **Weiteres Vorgehen** | | | |
| Testphase anfangen. | | | |

Tabelle 10-8: Achter Tag: Montag 24.02.2020 (Halbtag)

## Neunter Tag: Mittwoch 26.02.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Beteiligte Personen** | **Aufwand geplant (Std)** | **Aufwand effektiv (Std)** |
| **Expertenbesuch** | Josiah Schiess  Stefano Amati  Iseli Erich Reto | 1 | 1 |
| **Test durchführen** | Josiah Schiess  Hezekiah Bakare | 3 | 3 |
| **Reservezeit** | Josiah Schiess | 2 | 0 |
| **Benutzeranleitung** | Josiah Schiess | 1 | 3 |
| **Arbeitsjournal, Backup erstellen** | Josiah Schiess | 1 | 1 |
| ***Aufwand Total*** |  | **8** | **8** |
| **Tages Ablauf** | | | |
| Der Tag fing mit dem Experten Besuch an. Dann machte ich mich an die Testingphase dran. Leider war meine Vorhergesehener Tester wegen Krankheit abwesend. Ich habe die Hilfe von, Hezekiah Bakare, einen Schüler im 2. Lehrjahr stattdessen beantragt. Nach dem Testing machte ich mich dran, die Benutzeranleitung zu schreiben. | | | |
| **Erfolge und Misserfolge** | | | |
| Mir geling die Benutzeranleitung fast Komplet abzuschliessen.  Ich habe es nicht hinbekommen das Sitzungsprotokoll vom Expertenbesuch bis am Abend in PkOrg.ch hochzuladen, da ich nicht rechtzeitig dran gedacht habe. | | | |
| **Reflexion** | | | |
| Ich war am morgen noch ein wenig ablenkbar, trotz dessen habe ich es die Testingphase abschliessen können. Am Nachmittag habe ich besser arbeiten können. | | | |
| **Hilfestellung** | | | |
| Ich habe Stefano kurz gebeten ein Teil meines Projekt anzuschauen, um eine andere perspektive zu gewinnen. Dies hat mir geholfen kleine Fehler vor der Testingphase zu korrigieren. | | | |
| **Weiteres Vorgehen** | | | |
| Sitzungsprotokoll hochladen, Benutzeranleitung fertigstellen. | | | |

Tabelle 10-9: Neunter Tag: Mittwoch 26.02.2020

## Zehnter Tag: Donnerstag 27.02.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Beteiligte Personen** | **Aufwand geplant (Std)** | **Aufwand effektiv (Std)** |
| **Sitzung mit verantwortliche Fachkraft** | Josiah Schiess  Stefano Amati | 0 | 1 |
| **Benutzeranleitung: Erfassen, Mutieren, Löschen von Materialien** | Josiah Schiess | 3 | 1 |
| **Benutzeranleitung: Ausleihen von Materialien** | Josiah Schiess | 3 | 0 |
| **Dokumentation überarbeiten** | Josiah Schiess | 1 | 5 |
| **Arbeitsjournal, Backup erstellen** | Josiah Schiess | 1 | 1 |
| ***Aufwand Total*** |  | **8** | **8** |
| **Tages Ablauf** | | | |
| ich begann den Tag mit der Arbeit an der Benutzeranleitung und habe diese fertiggestellt. Dann arbeitete ich an die Dokumentation. Ich hatte die Phasenfreigabe von der Realisierung und besprach mein Projekt mit meiner verantwortlichen Fachperson. | | | |
| **Erfolge und Misserfolge** | | | |
| Ich habe heute die Benutzeranleitung schnell abschliessen können. Dies hat mir sehr geholfen, da meine Dokumentation viel Überarbeitung braucht.  Bei der Dokumentation habe ich sehr viele Rechtschreib- und Grammatikfehler entdeckt. | | | |
| **Reflexion** | | | |
| Keine | | | |
| **Hilfestellung** | | | |
| Keine | | | |
| **Weiteres Vorgehen** | | | |
| Dokumentation überarbeiten, Rechtschreibung korrigieren, Abgabe | | | |

Tabelle 10-10: Zehnter Tag: Donnerstag 28.02.2020

## Elfter Tag: Freitag 28.02.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Beteiligte Personen** | **Aufwand geplant (Std)** | **Aufwand effektiv (Std)** |
| **Dokumentation überarbeiten** | Josiah Schiess | 4 |  |
| **Frontend Komponente** | Josiah Schiess | 0 | 1 |
| **Reservezeit** | Josiah Schiess | 2 | 0 |
| **Arbeitsjournal, Backup erstellen** | Josiah Schiess | 1 | 1 |
| ***Aufwand Total*** |  | **8** | **8** |
| **Tages Ablauf** | | | |
| Ich Habe den Ganzen Tag an der Dokumentation verbracht. | | | |
| **Erfolge und Misserfolge** | | | |
| Als durch alle Anforderungen der Applikation durchgeschaut habe, ist mir aufgefallen, dass ich die 11. Anforderung: «Es kann, sowohl von Lehrkräften wie auch Lernenden, eine Lagerliste in der Applikation bzw. auf dem Bildschirm ausgegeben werden, welche die verfügbare (nicht ausgeliehene) Menge eines Materials angibt. (Z4).». überschaut habe. Ich habe dies dann schnell korrigiert | | | |
| **Reflexion** | | | |
| Ich bin erleichtert das die IPA vorbei ist, und werde noch mehr froh sein, wenn die Präsentation auch vorbei ist. Ich Glaube/Hoffe, dass die Arbeit, die ich geleistet habe, ausreichend sein wird. Da habe ich aber ich habe meine Bedenken. | | | |
| **Hilfestellung** | | | |
| Keine | | | |
| **Weiteres Vorgehen** | | | |
| Dokumentation überarbeiten, Rechtschreibung korrigieren, Abgabe | | | |

Tabelle 10-11: Elfter Tag: Freitag 28.02.2020

# Abschlussbericht

## Vergleich Ist/Soll (Anforderungen, Zeit, Einsatzmittel)

## Fazit zur IPA

## Persönliches Fazit

## Schlussreflexion

Teil 2: Projekt-Dokumentation

## Einführung

Der Beruf Informatikpraktiker/in EBA hat die TFBern seit ca. 10 Jahren angeboten. Im Jahr 2016 wurde der Lehrgang Informatiker/in EFZ Betriebsinformatik eingeführt. Der neue Lehrgang wurde in den Annex eingerichtet. Wegen der wachsende Anzahl Lernende, und die Aufteilung von Standorten gab es immer mehr den Bedarf nach einem Inventarisierung Tool. Dazu gibt es auch das Projekt von einem eigenen Labor Netzwerk für Lehrzwecke aufzubauen. Dies führt auch zu mehr Materialien, die angeschafft werden müssen. Somit wird es immer schwerer die vorhandenen Lehrmaterialien in Blick zu behalten.

Es besteht eine rudimentäre Lösung mit einer lokalen Access Datenbank. Dies deckt aber lange nicht den Bedarf ab.

Aus diesen Gründen entstand die Idee einer Webapplikation als Auftrag für die Lernenden von den ersten und zweiten Lehrjahren zu gestalten. Als das Projekt vorgeschlagen wurde habe ich mich als Freiwilliger gemeldet, bevor das Projekt klar definiert war oder anfing. Ich habe mich für diesen Auftrag, im Rahmen einer IPA gemeldet, da dieser Themenbereich meinen Stärken entspricht. Das Projekt wurde schliesslich für die IPA eingeschränkt, damit es in dem Zeitrahmen passt.

Abbildung 6: LOA903

# Initialisierung

## IST-Situation

|  |  |
| --- | --- |
| Nr. | Situation |
| 1 | Es besteht eine lokale MS Access Datenbank. |
| 2 | Die aktuelle Lösung verfügt über Keine Ausleifunktion. |
| 3 | Die Inventargegenstände können nicht verfolgt werden. |
| 4 | Zugriff auf der Datenbank sehr beschränkt. |

Tabelle 12-1: IST-situation

### Abgrenzungen

Es Besteht eine Rudimentäre MS Access DB. Diese wird nicht für das Projekt beachtet oder gebraucht. Als Folgeprojekt können die bestehenden Einträge der MS Access DB in die neu erstellte MySql DB übertragen werden.

## SOLL-Situation

Wegen der Zeiteinschränkung der IPA, wird das Projekt auf die Entwicklung der Applikation begrenzt. Nicht beachtet werden:

* Serveraufsetzung
* Netzwerkeinrichtungen
* Netzwerkkonzept
* IP-konzept
* ADDS Einbindung
* Datenerfassung
* Benutzererfassung

Diese können nach der IPA als separate Projekte implementiert werden.

### Systemziele

|  |  |  |
| --- | --- | --- |
| Nr. | Ziel | Beschreibung |
| 1 | Login | Es ist ein Login implementiert, welches Lernende und Lehrkräfte unterscheidet. |
| 2 | Web interface | Es besteht ein Webinterface, welches den Lehrkräften ermöglicht, Materialien neu zu erfassen, zu mutiert und zu löschen, sowie Abfragen zu den Materialien und Ausleihe von Materialien vorzunehmen. Das Interface ermöglicht den Lernenden Abfragen zu den Materialien und Ausleihe von Materialien vorzunehmen. |
| 3 | Attribute der Materialien sind konkret gesetzt | Zu den Materialien können Nummer, Bezeichnung, Hersteller, Typ, Beschreibung/Eigenschaften, Seriennummer, Menge und Standort erfasst werden. |
| 4 | Lagerlisten Ansicht | Es kann eine Lagerliste, welche die verfügbare Menge eines Materials angibt und eine Ausleihliste, welche für einen bestimmten Benutzer angibt, welche Materialien er/sie ausgeliehen hat, auf dem Bildschirm ausgegeben werden. |
| 5 | SQL Skripte | Die Tabellen der DB werden mittels SQL-Script erstellt; die Datenbank ist in der 3NF. |
| 6 | Webserver und MySql sind installiert | Webserver und MySql werden für die IPA lokal installiert. |
| 7 | Benutzeranleitung zur Applikation | Es besteht eine Benutzeranleitung zur Verwendung der Inventar-Applikation. |

Tabelle 12-2: Systemziele

### Systemanforderungen

#### Funktionale Anforderungen:

|  |  |  |
| --- | --- | --- |
| Nr. | Anforderung | Ziel |
| A1 | Je drei Testnutzer aus den beiden Benutzergruppen für die Lehrkräfte und Lernenden sollen mit Namen und Passwort per Script in der Datenbank erstellt werden | (Z1) |
| A2 | Die Lehrkraft kann sich mit Benutzername und Passwort anmelden und erhält die entsprechenden Rechte | (Z1) |
| A3 | Die Lernende / der Lernende kann sich mit Benutzername und Passwort anmelden und erhält die entsprechenden Rechte | (Z1) |
| A4 | Die Passwörter werden als gehashte Strings in der Datenbank abgelegt | (Z1) |
| A5 | Eine Lehrkraft kann Materialien über das Web-Interface erfassen | (Z2) |
| A6 | Eine Lehrkraft kann Materialien über das Web-Interface löschen | (Z2) |
| A7 | Eine Lehrkraft kann Materialien über das Web-Interface mutieren | (Z2) |
| A8 | Eine Lehrkraft kann Materialien über das Web-Interface ausleihen. Dabei wird das Material als ausgeliehen markiert und mit seinem Namen verknüpft | (Z2) |
| A9 | Eine Lernende/ein Lernender Materialien über das Web-Interface ausleihen. Dabei wird das Material als ausgeliehen markiert und mit seinem Namen verknüpft | (Z2) |
| A10 | Zu den Materialien sollen die folgenden Attribute erfasst werden können: Nummer, Bezeichnung, Hersteller, Typ, Beschreibung/Eigenschaften, Seriennummer und Standort. | (Z3) |
| A11 | Es kann, sowohl von Lehrkräften wie auch Lernenden, eine Lagerliste in der Applikation bzw. auf dem Bildschirm ausgegeben werden, welche die verfügbare (nicht ausgeliehene) Menge eines Materials angibt. | (Z4) |
| A12 | Die Lehrkraft kann eine Ausleihliste in der Applikation bzw. auf dem Bildschirm ausgegeben lassen, welche für einen Benutzer angibt, welche Materialien er/sie ausgeliehen hat. Dabei kann die Lehrkraft für allen Benutzer die Auswahlliste ausgeben lassen. | (Z4) |
| A13 | Die Lernende/der Lernender kann für sich selbst eine Ausleihliste in der Applikation bzw. auf dem Bildschirm ausgegeben lassen, welche anzeigt, welche Materialien sie/er ausgeliehen hat. | (Z4) |
| A14 | Die Tabellen der DB werden mittels SQL-Script erstellt | (Z5) |
| A15 | Der Webserver ist lokal installiert | (Z6) |
| A16 | Die Datenbank ist lokal installiert | (Z6) |
| Nr. | Anforderung | Ziel |
| A17 | Es besteht eine Benutzeranleitung mit den Themen Login, Erfassen von Materialien, Mutieren von Materialien, Löschen von Materialien, Ausleihen von Materialien, Ausgabe der Lagerliste und Ausleihliste | (Z7) |

Tabelle 12-3: funktionale Anforderungen

#### Nicht Funktionale Anforderungen:

|  |  |  |
| --- | --- | --- |
| Nr. | Anforderung | Ziel |
| NA1. | Die Lagerliste und die Ausleihliste werden in der Applikation bzw. auf dem Bildschirm ausgegeben. | (Z4) |
| NA2. | Die Datenbank ist in der 3. NF | (Z5) |
| NA3. | Die Applikation wird mit JavaScript erstellt. | (Z6) |
| NA4. | Als Datenbank wird MySql verwendet. | (Z6) |
| NA5. | Die Benutzeranleitung ist als separate Datei realisiert und nicht in die Applikation integriert. | (Z7) |

Tabelle 12-4: nicht Funktionale Anforderungen

## Vorgehensziele

|  |  |  |
| --- | --- | --- |
| Ziel ID | Ziel | Beschreibung |
| V1 | HERMES 5 verwenden | Die Projektmethode Hermes 5 wird angewendet. |
| V2 | Zeitplan einhalten | Der Zeitplan wird eingehalten |
| V3 | Arbeitsjournal | Es wird jeden Tag ein Arbeitsjournal geführt. |
| V4 | Pünktliche Abgabe | Die Abgabe Zeit wird eingehalten. Am Vormittag des Abgabetermins wird eine Version hochgeladen zum Testen der Upload Funktion |
| V5 | Arbeitsort | Die Arbeit wird an der TFBern durchgeführt |

Tabelle 12-5: Vorgehensziele

# Variantenvergleich

Wegen der Ausführlichen Aufgabenstellung, konnte keine Varianten gefunden werden.

# Konzept

## Namenskonzept

Da die IPA in einer eigenen Projektumgebung stattfindet, welche nicht mit der Domäne der TF Bern verknüpft ist, und somit nicht diese Benutzerdaten verwendet, wird ein neues Namenskonzept erstellt.

### Benutzer

Für Lehrer wird der Benutzername aus dem ersten und zweiten Buchstaben des Nachnamens und dem ersten Buchstaben des Vornamens kombiniert.

**Beispiel**:

**Name**: Marco Schneider

**Username**: SCM

Für Lernende wird er Benutzername folgendermassen zusammengesetzt: die ersten 2 Buchstaben der Abteilung, in welcher er/sie befindet. Annschliessend folgt das Jahr, in welchem er/sie die Lehre angefangen hat. Schliesslich endet der Benutzername mit den ersten zwei Buchstaben des Nachnamens und den ersten zwei des Vornamens.

**Beispiel**:

**Name**: Marco Schneider

**Abteilung**: Informatik

**Jahr**: 2020

**Username**: in20scma

### Datenbank Relationen

Die Technische Fachschule Bern verfügt über keine Namenskonvention. Deshalb habe ich mir eine für diese Projekt erstellt.

|  |  |
| --- | --- |
| **Attribut** | **Beschreibung** |
| Primärschlüssel | Primärschlüssel werden immer mit ‘PK’ gekennzeichnet. Anschliessend folgt nach einem Unterstrich der Name der Tabelle. Danach folgt nach einem Unterstrich ‘ID’.  **Beispiel: PK\_TabellenName\_ID** |
| Fremdschlüssel | Primärschlüssel werden immer mit ‘FK’ gekennzeichnet. Anschliessend folgt nach einem Unterstrich der Name der Tabelle. Danach folgt nach einem Unterstrich ‘ID’.  **Beispiel: FK\_TabellenName\_ID** |
| Tabellenname | Der Name der Tabelle muss auf English sein und in Camel Case geschrieben sein. |
| Zwischentabellen | Der Name einer Zwischentabellen wird durch die Namen der beiden Tabellen, separiert durch einen Unterstrich, gebildet.  **Beispiel: TabellenName\_Tabellenname** |

Tabelle 14-1: Datenbank Namenskonzept

## Mockups

Die Mockups wurden mit einem Material design Vorlage von Adobe, welches den Stil von Vuetify sehr nah liegt. Dazu habe ich auch von der Vuetify Dokumentation Seite, Komponente verglichen und wie geschildert mit Adobe XP dargestellt.

Die Mockups wurden erstellet, um einen Leitfaden zu haben bei der UI Entwicklung. Die effektiven Webseiten sollen die Mockups ähneln.

### Login Seite

Öffentliche Seite die jeder sehen kann.



### Start Seite Lehrpersonen

Die Startseite der Lehrpersonen



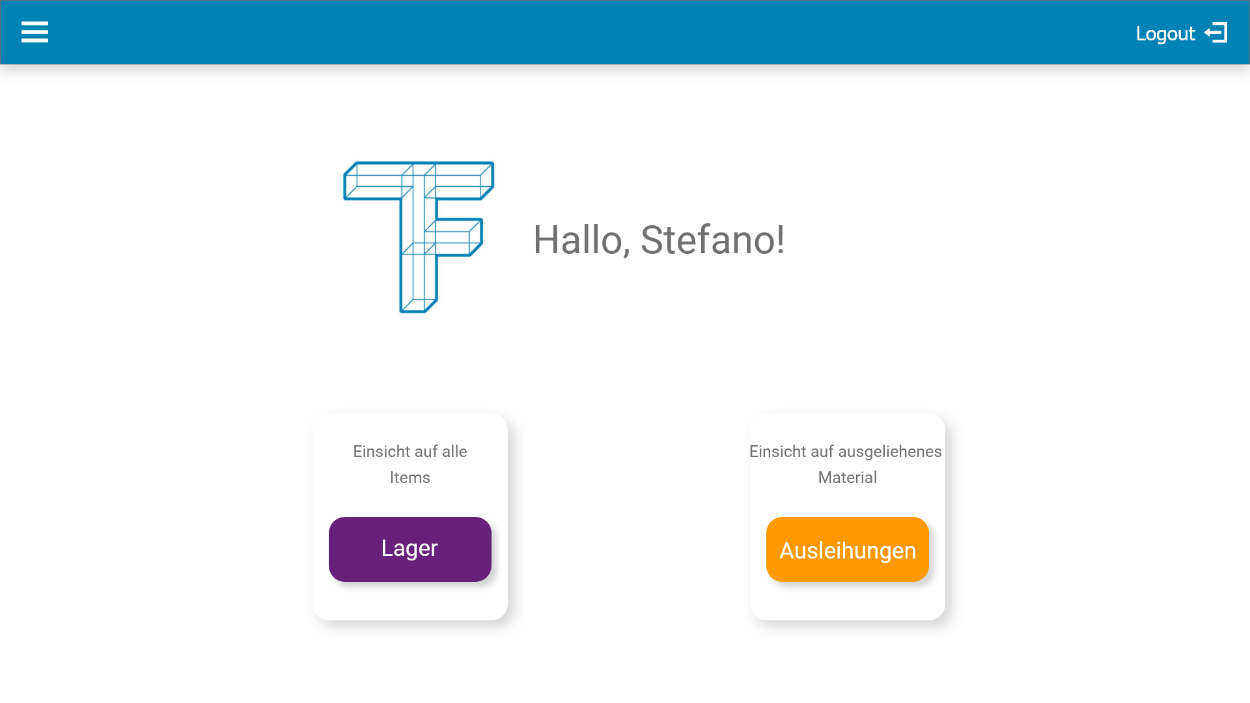
### Lehrer Startseite Menu

Das Menu für die Lehrpersonen. Das Menu sollte von jeder Seite verfügbar sein.



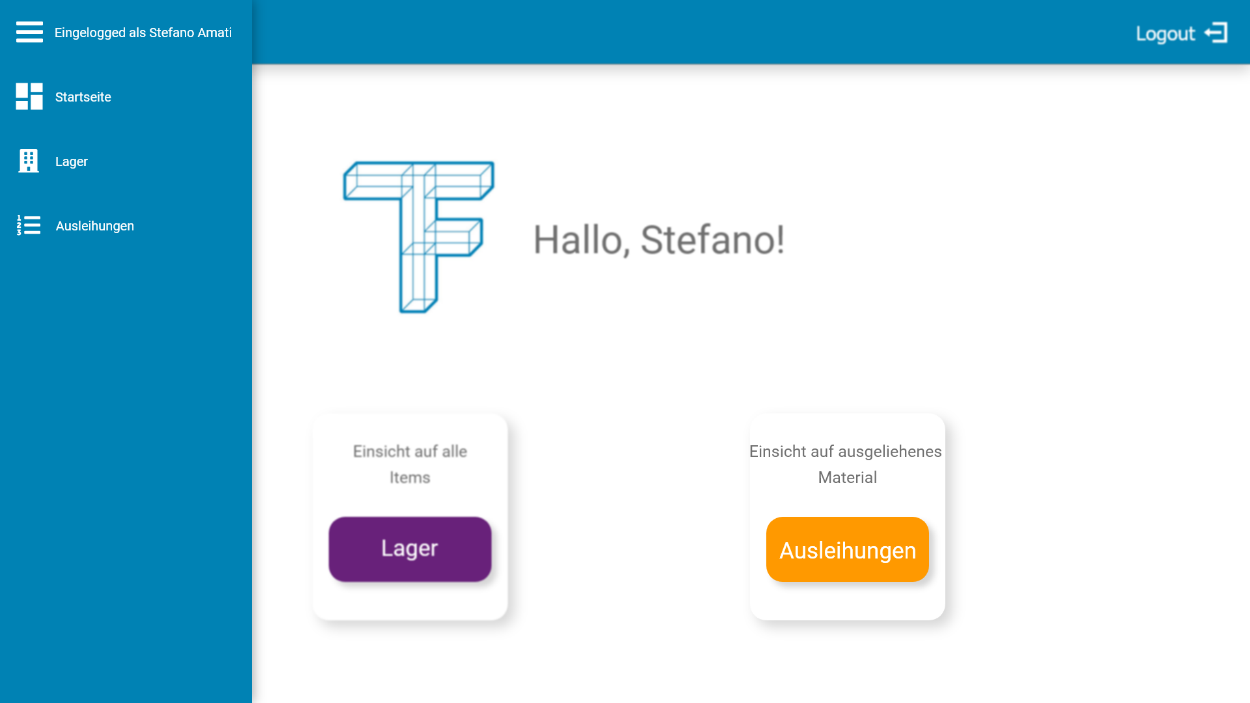
### Schüler Startseite

Die Startseite der Schüler. Nach dem Anmeldeverfahren wird man auf dieser Seite weitergeleitet



### Schüler Startseite Menu

Das Menu sollte von jeder Seite verfügbar sein.



### Lehrer Neues Material erfassen

Hier ist das Formular, um neues Material zu erfassen.



### Inventarliste Lehrer

Auflistung von allen Materialien. Von dieser Seite soll die Lehrkraft Materialien bearbeiten und löschen können



### Ausleihungen

Hier werden die eigenen Ausleihungen angezeigt. Wenn eine Lehrkraft angemeldet ist kann hier alle Ausleihungen angezeigt werden



### Inventarliste Schüler

Schüler können von dieser Seite aus, alle Materialien anschauen und ausleihen



## Datenbank Konzept

Die einzelnen Artikel werden als individuelle Entitäten in der Tabelle 'Items' gespeichert. Jeder Artikel in der Tabelle ‘Items’ gehört zu einer Klasse. Die Klasse enthält Informationen über die Produktklasse.

|  |  |
| --- | --- |
| Attribut | Beschreibung |
| **ID (Nummer)** | Inkrementell erhöhte Nummer |
| **Name (Bezeichnung)** | Name des Produkts  **Beispiel:**   * Raspberry PI Model 4 * Lenovo E420s |
| **Manufacturer (Hersteller)** | Firma, die das Produkt herstellt  **Beispiel:**   * Samsung * Intel * Raspberry PI |
| **Type (Typ)** | Überbegriff zum Klassifizieren  **Beispiel:**   * Bildschirm * Tastatur * Raspberry PI |
| **Description (Beschreibung/Eigenschaften)** | Eigenschaften des Produkts als String  **Beispiel:**  ‘a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote teaching of basic computer science in schools and in developing countries.’ |
| **Serialnumber (Seriennummer)** | Eindeutige Nummer, welches den Artikel identifiziert |
| **Username** | Name des Benutzers |
| **Password** | Passwort des Benutzers |
| **Role (Rolle)** | Rolle des Benutzers |

Tabelle 14-2: Attribute

### Konzept Datenbank in der ersten Normalform

Eine Relation ist in der 1. Normalform, wenn alle zugrundeliegenden Attribute atomare Werte enthalten und ein eindeutiger Primärschlüssel gegeben ist. Dieses Schema erfüllt das Kriterium. Die erste Normalform könnte aber auch in einer Tabelle.

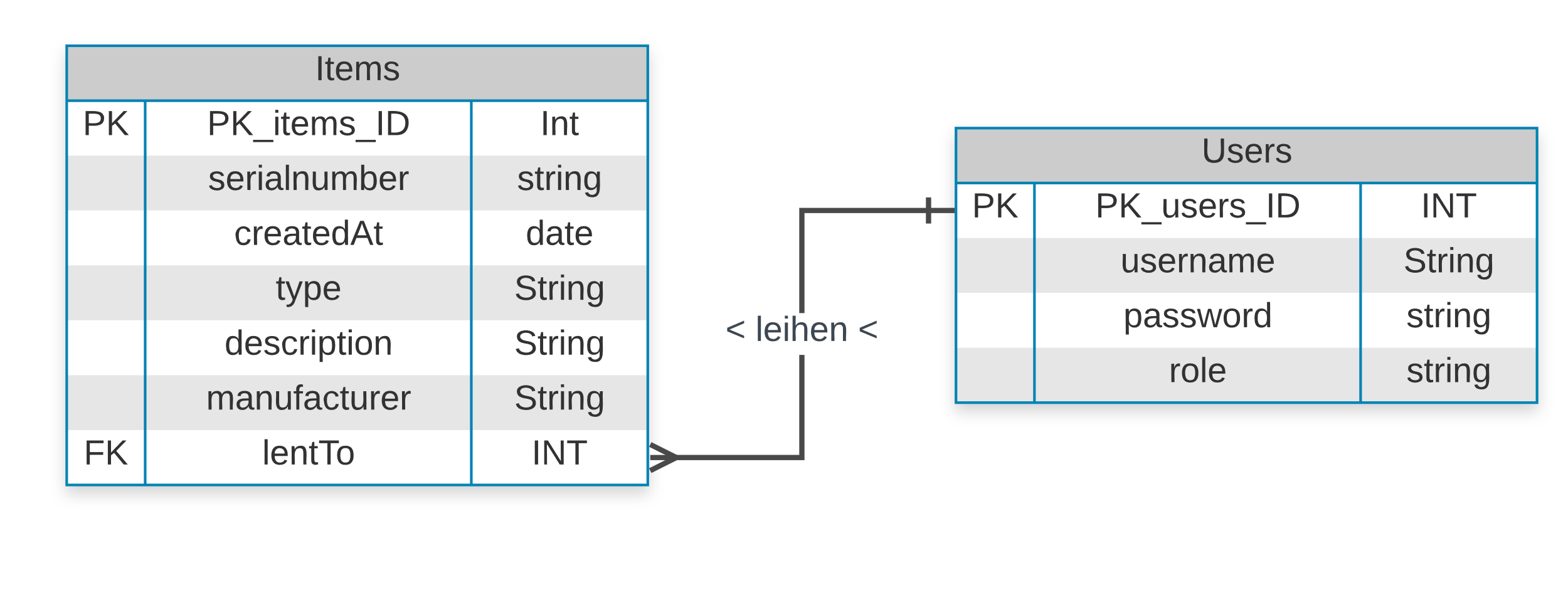


Abbildung 7: Konzept: erste Normalform

### Konzept Datenbank in der zweiten Normalform

Um die zweite Normalform zu erfüllen, müssen alle nicht-Schlüssel Attribute von einem einzigen Schlüssel abhängen. Dies wird erreicht, indem man die Items Tabelle aufteilt.

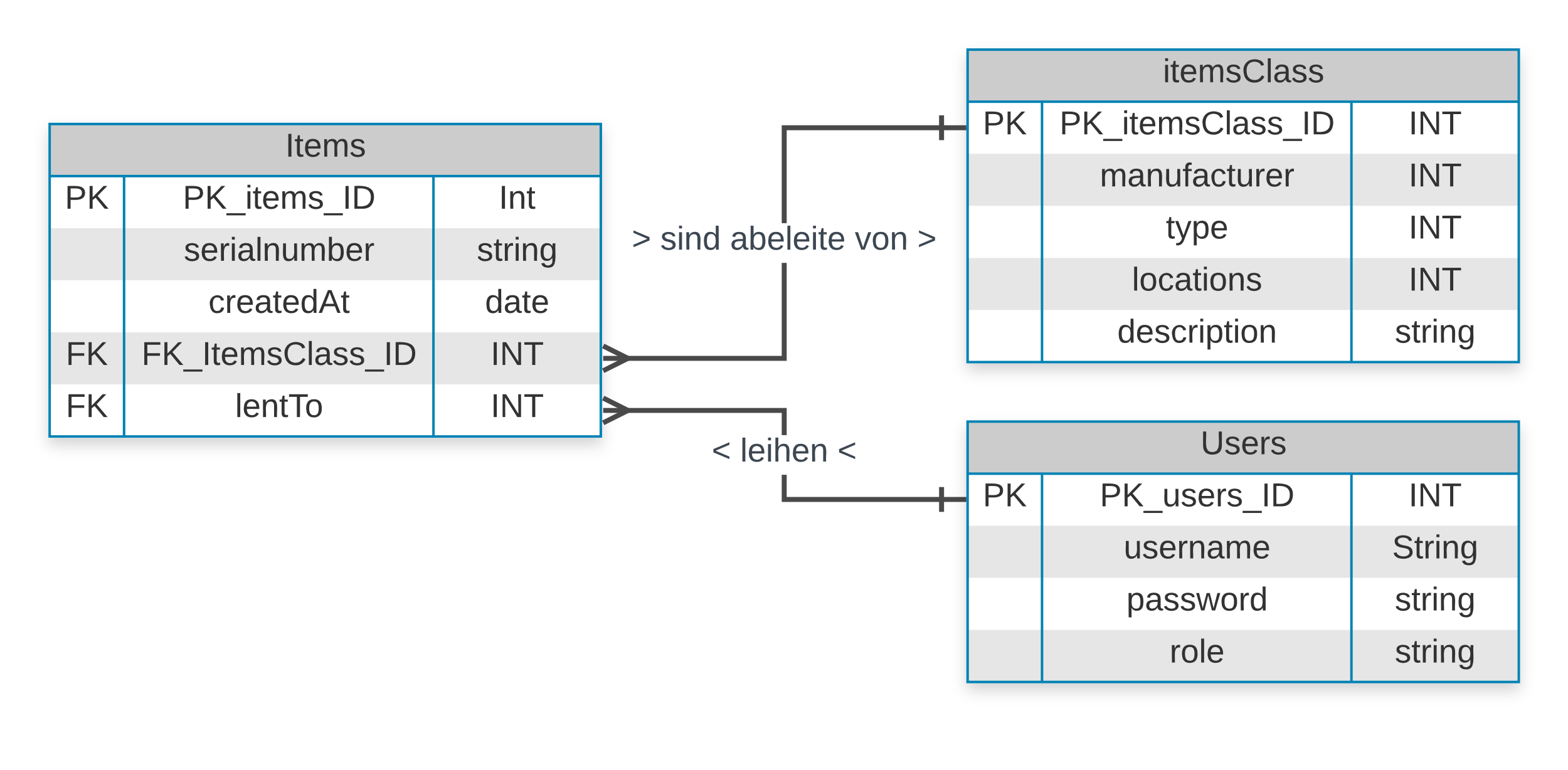


Abbildung 8: Konzept: zweite Normalform

### Konzept Datenbank in der dritten Normalform

Die dritte Normalform ist erreicht, wenn die zweite Normalform erreicht ist und wenn keine transitive Abhängigkeit in einer Tabelle vorhanden ist.

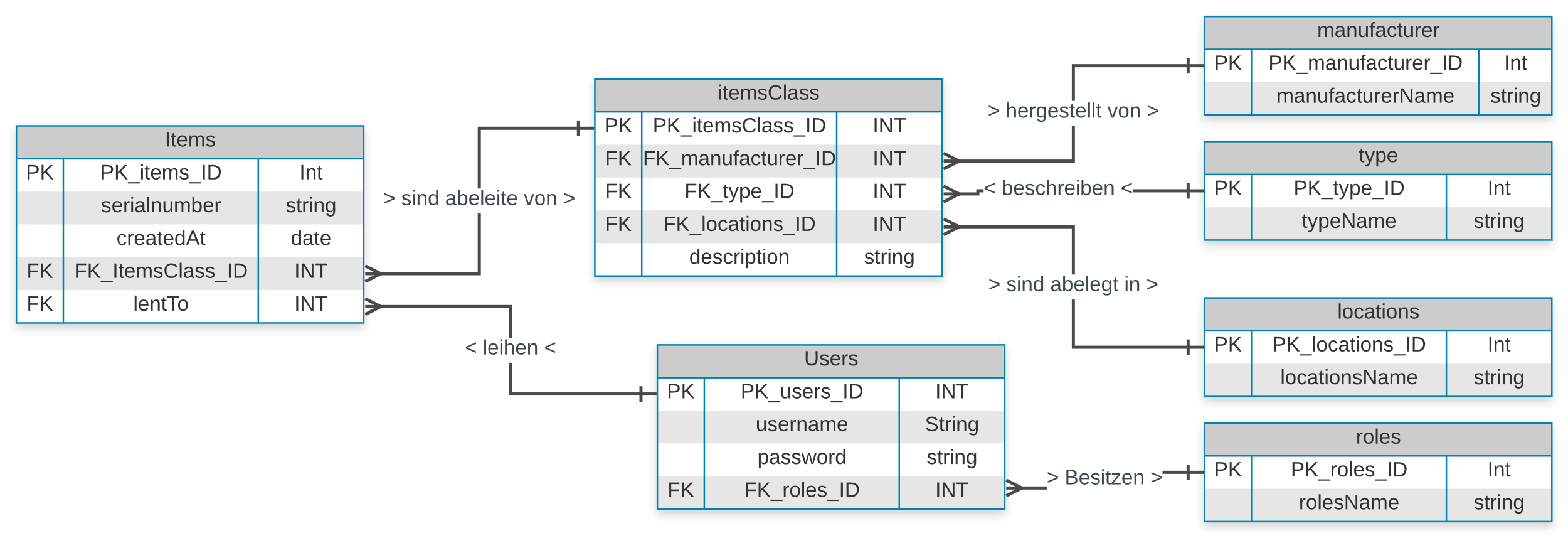
****

Abbildung 9: Konzept: dritte Normalform

### Verifizierung

Die Normalisierungsprozess wurde von Giulio Iannattone verifiziert.

## Programmstruktur

### Entwicklungsumgebung Architektur

Dieses Diagramm zeigt die Architektur der Entwicklungsumgebung.

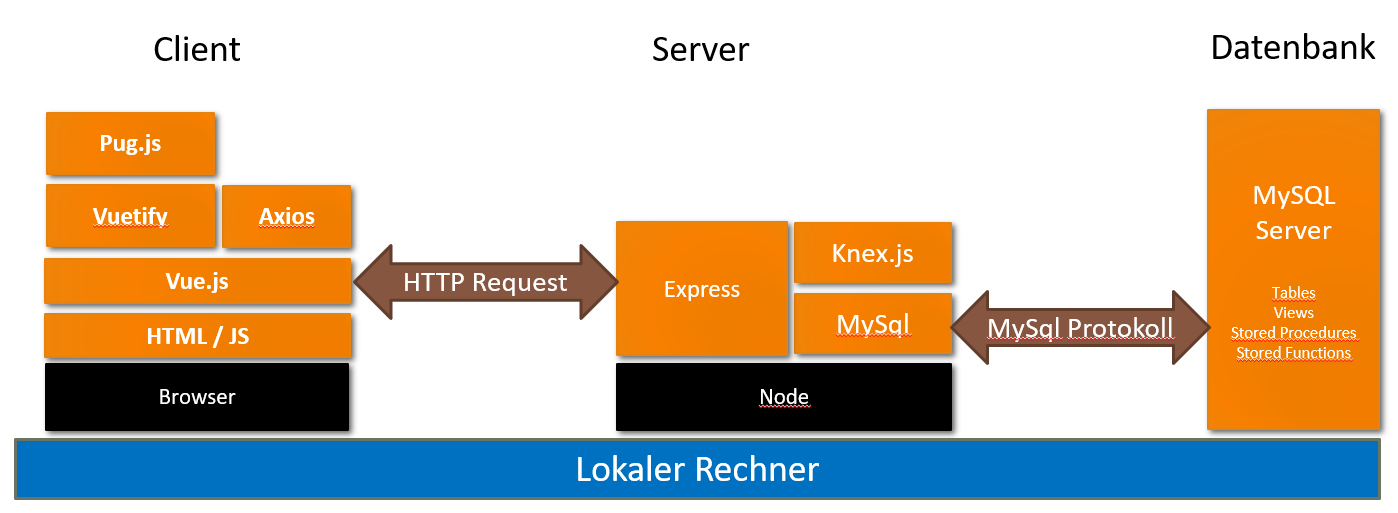


Abbildung 10: Konzept: Entwicklungsumgebung Architektur

### Frontend Package Diagramm

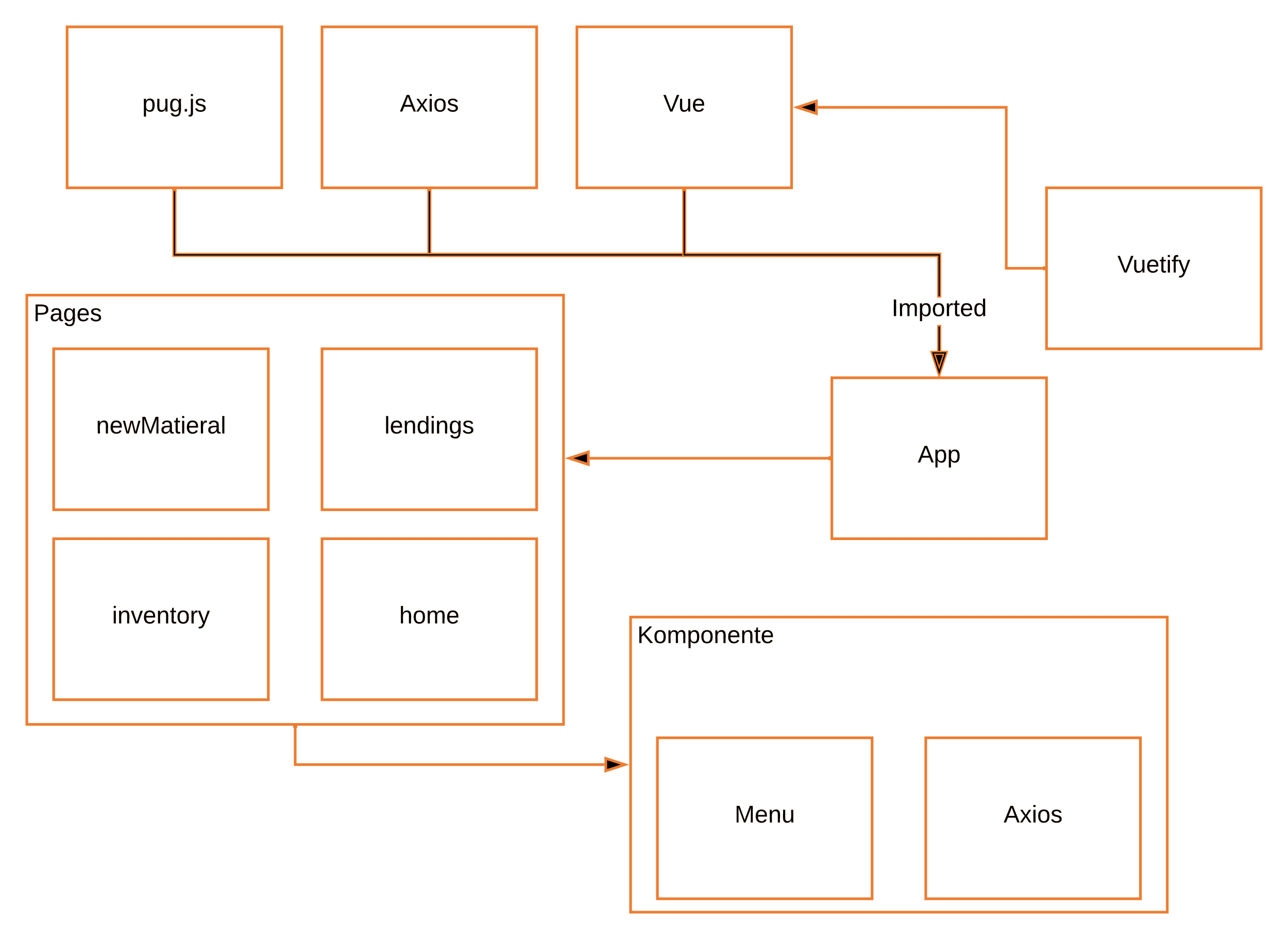
Dieses Diagramm zeigt das Package-Schema für den Front-End Teil des Programms.

Abbildung 11: Frontend Package Diagramm

### Back-End Package Diagramm

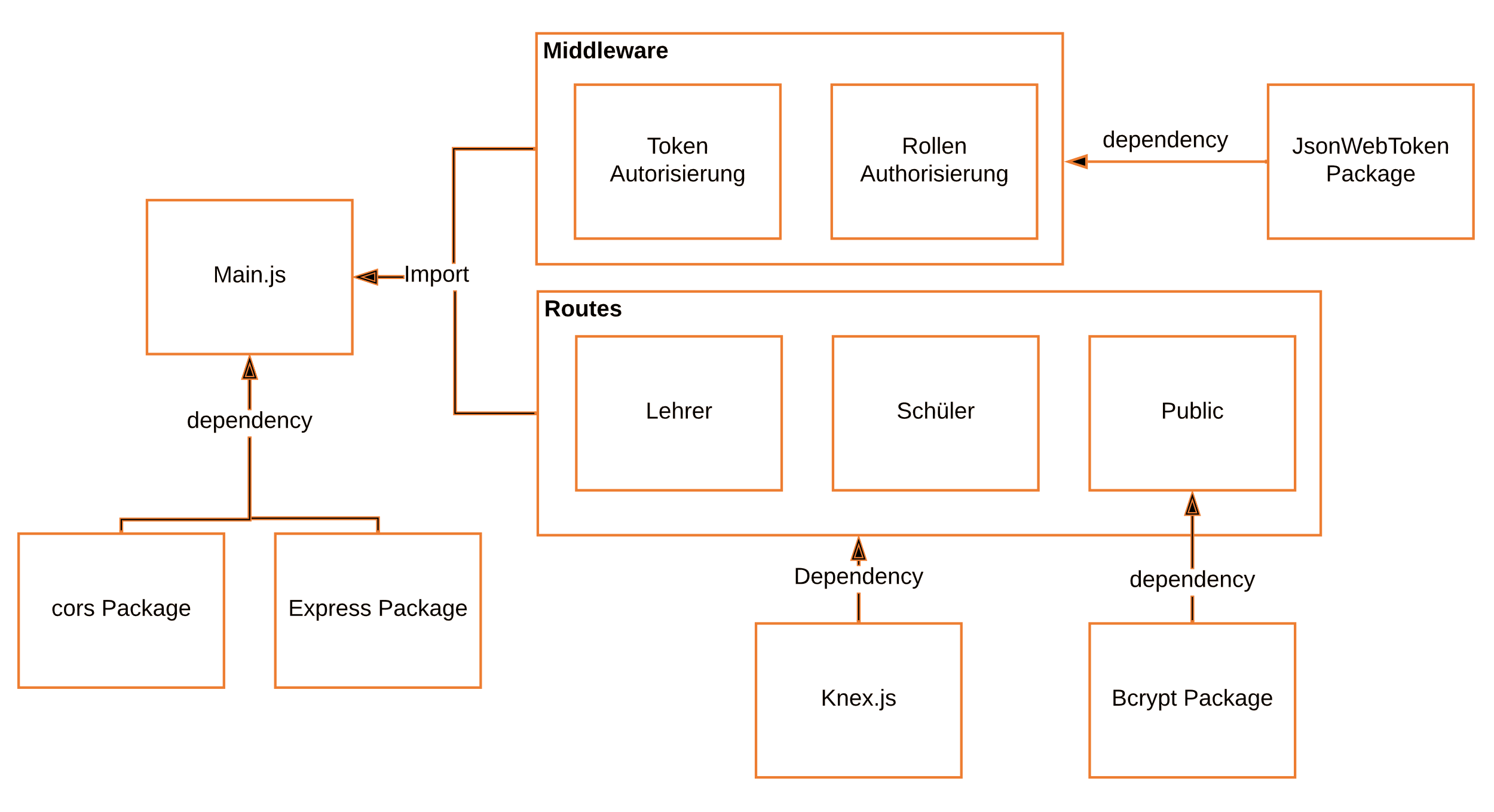
Dieses Diagramm zeigt das Package-Schema für den Back-End Teil des Programms

Abbildung 12: Back-End Package Diagramm

### konzeptionelles Login verfahren

Dieses Sequenzdiagramm beschreibt das konzeptionelle Anmeldeverfahren.

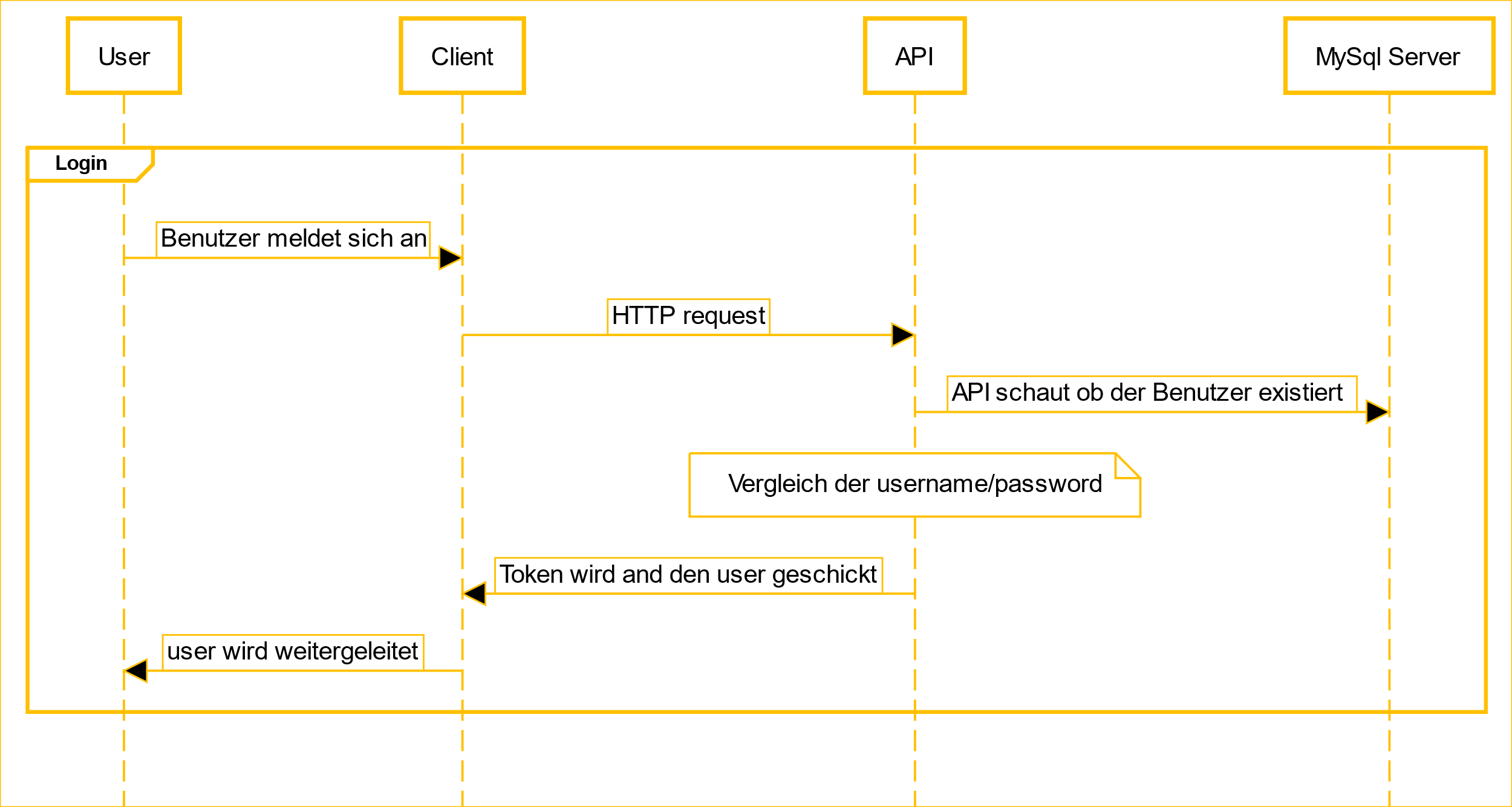


Abbildung 13: konzeptionelles Login verfahren

## Vue

Vue ist ein JavaScript Framework, welches gebraucht wird, um Progressive Webapplikationen zu erstellen. Vue ist relativ klein. Um komplexere Funktionen zu verwenden, wie Routing, State-Management, etc. werden weitere Packages und Plugins verwendet.

### Vuex

Vuex wird verwendet, um das Local State Management Schema umzusetzen. Zu Vuex wird das vuex-persistedstate Package installiert. Dieses wird gebraucht, um direkt von Vuex in den Lokalen Speicher des Browsers zu schreiben.

* Der State, die Quelle der Wahrheit, die unsere Anwendung antreibt;
* Die View (Vue components), eine visuelle Abbildung des States;
* Die Actions lösen Mutations aus.
* Die Mutations werden nur verwendet, um das State zu ändern.

vuex-persistedstate verknüpft den State mit dem lokalen Speicher des Browsers. So kann die Seite neugeladen werden, ohne nochmal anzumelden.



Abbildung 14: Vuex

### Vue-router

Der Router ist zuständig, für das Page-Routing. Heisst er steuert die Navigation zwischen den Webkomponenten.

### Vue Dateien

Die einzigen Vue Komponente sind in Dateien aufgeteilt. Vue Dateien habe die .vue Datentyp.

## Konzeptionelle Schnittstellen

Die API sollte zwischen Lehrer und Schüler unterscheiden. Deshalb benötigt die API mindestens 2 Routen. Die Schüler sollten nur auf ihre Route kommen. Die Lehrer sollte jedoch auch die Lehrer-Route kommen wie auch auf die Schüler-Route. Dazu braucht es auch eine öffentliche Route für das Anmeldeverfahren.

### Diagramm

Abbildung 15: Konzeptionelles Schnittstellendiagramm

### öffentliche Routen

Diese Endpunkte sind für aller erreichbar.

|  |  |  |  |
| --- | --- | --- | --- |
| Methode | Endpunkt | Beschreibung | Anforderung |
| POST | /login | Anmeldung | A3 |

Tabelle 14-3: Login

### Schüler Routen

Die Schüler Route sollte über den Endpunkt: /student erfolgen.

|  |  |  |  |
| --- | --- | --- | --- |
| Methode | Endpunkt | Beschreibung | Anforderung |
| GET | /Inventory | Holt sich die verfügbaren Daten. | A11 |
| GET | /lendings | Eigene Ausleihungen werden angezeigt | A13 |
| POST | /lendings | Material wird ausgeliehen. | A9 / A8 |
| DELETE | /lendings | Material wird zurück gegeben | - |

Tabelle 14-4: Schüler Routen

### Lehrer Routen

Die Schüler Route sollte über den Endpunkt: /admin erfolgen.

|  |  |  |  |
| --- | --- | --- | --- |
| Methods | Endpunkt | Beschreibung | Anforderung |
| GET | /lendings/:id | Für ein bestimmter Benutzer werden alle ausgeliehenen Materialien angezeigt | A12 |
| POST | /inventory | Neues Material erfassen | A5 |
| PUT | /inventory/:id | Ein Bestimmtes Material bearbeiten | A7 |
| DELETE | /inventory/:id | Ein bestimmtes Material löschen | A6 |

Tabelle 14-5: Lehrer Routen

## Testkonzept

### Testdaten

Daten, die mit Hilfe der Anwendung nicht eingefügt werden können, werden mit Skripten eingefügt.

#### Benutzerdaten

|  |  |  |
| --- | --- | --- |
| username | password (vor verschlüsselung) | Rolle |
| scjo | kek | Lehrer |
| peju | iluvtfBern | Lehrer |
| sika | okBoomer | Lehrer |
| in17scma | gibbix12345 | Schüler |
| in19sika | passwort1 | Schüler |
| In18frch | Csgopro1337 | Schüler |

Tabelle 14-6: Benutzertestdaten

#### types

|  |  |
| --- | --- |
| PK\_types\_ID | typesName |
| 1 | Tastatur |
| 2 | Bildschirm |
| 3 | Raspberry PI |

Tabelle 14-7: Types

#### Locations

|  |  |
| --- | --- |
| PK\_locations\_ID | locationsName |
| 1 | LOA03 |
| 2 | LOA04 |
| 3 | LOA10 |

Tabelle 14-8: Locations

#### manufacturers

|  |  |
| --- | --- |
| PK\_manufacturers\_ID | manufacturersName |
| 1 | Samsung |
| 2 | Logitech |
| 3 | Raspberry PI |

Tabelle 14-9: Manufacturers

### Fehlerklassen

|  |  |  |
| --- | --- | --- |
| Klassen ID | Fehlerklassen | Beschreibung |
| M0 | Kein Mangel | Das erwartete sowie das Tatsächliche Resultat stimmen überein. |
| M1 | Leichter Mangel | Grundlegend funktioniert alles, kleinere Mängel ersichtlich |
| M2 | Schwerer Mangel | Die Anforderung wird nur teilweise erfüllt. |
| M3 | Kritischer Mangel | Die Anforderungen werden sehr schlecht bis gar nicht erfüllt. |

Tabelle 14-10: Fehlerklassen

### Testarten

|  |  |
| --- | --- |
| Testart | Beschreibung |
| Blackbox-Testing | Funktion orientierte Tests. Test werden ohne Kenntnisse die Systeme oder innere Funktionen durchgeführt. |
| Whitebox-Testing | Struktur orientierte Tests. Innere Funktionen einzelne Komponenten werden getestet |

Tabelle 14-11: Testarten

### Testvorgehen

Die Tests erfolgen nach der Realisierungsphase des Projekts. Alle Tests werden in Folge von einander gemacht. Bei Auftreten von Fehler, werden diese beschrieben und wenn möglich gelöst.

### Testziele

Fehlerfreie Durchführung von den Tests.

### Testkategorien

|  |  |  |
| --- | --- | --- |
| ID | Kategorie | Beschreibung |
| K1 | Login | Alle Testfälle, die mit dem Login zu tun haben. |
| K2 | Inventar | Alle Testfälle, die mit dem Inventar zu tun hat. |
| K3 | Ausleihungen | Alle Testfälle, die mit dem Ausleihen, oder zurückbringen von Material zu tun hat. |

Tabelle 14-12: Testkategorien

### Testvorlage

|  |  |
| --- | --- |
| **Testfall ID** | Testfall ID (T1) |
| **Testmethode** | Welche Test Methode gebraucht wird |
| **Testkategorie** | Testkategorie ID |
| **Testfall** | Beschreibung des Testfalls |
| **Voraussetzungen** | Voraussetzungen die getroffen werden müssen |
| **Beschreibung** | Beschreibung des Vorgangs |
| **Test Schritte** | Die Tests werden in Schritte aufgeteilt |
| **Test Informationen** | Nötige Informationen zum Test |
| **Erwartete Resultate** | Geplante Resultate |
| **Erreichte Resultate** | Effektive Resultate |
| **Kommentar** | Kommentar zum Fehlschlag |
| **Fehlerklasse** | Fehlerklasse (M0-M3) |

Tabelle 14-13: Testvorlage

### Testfälle

#### Testfall 1: Login

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Testfall ID** | T1 | | | | |
| **Testmethode** | Blackbox | | | | |
| **Testkategorie** | K1 (Login) | | | | |
| **Testfall** | Login | | | | |
| **Voraussetzungen** | Kein Benutzer ist eingeloggt. Browsercache ist lehr | | | | |
| **Beschreibung** | Keine eingaben | Kein Benutzer | Kein Passwort | Richtiger Benutzer  Falsches Passwort | Richtiger Benutzer & Passwort |
| **Test Schritte** | 1. Username eingeben 2. Passwort eingeben 3. Auf Login klicken | | | | |
| **Test Informationen** | Benutzer: -  Passwort: - | Benutzer: -  Passwort: kek | Benutzer: scjo  Passwort: - | Benutzer: scjo  Passwort: 1234 | Benutzer: scjo  Passwort: kek |
| **Erwartete Resultate** | Fehlermeldung 403: Benutzer oder Passwort falsch | | | | Weiterleitung auf das Dashboard |
| **Erreichte Resultate** |  |  |  |  |  |
| **Kommentar** |  |  |  |  |  |
| **Fehlerklasse** |  |  |  |  |  |

Tabelle 14-14: Testfall 1: Login

#### Testfall 2: Material Klassen erstellen

|  |  |  |  |
| --- | --- | --- | --- |
| **Testfall ID** | T2 | | |
| **Testmethode** | Whitebox | | |
| **Testkategorie** | K2 (Inventar) | | |
| **Testfall** | Material Klassen erfassen | | |
| **Voraussetzungen** | * Einen Benutzer mit der rolle Lehrer ist eingeloggt * Man ist auf der ‘Material erfassen’ Seite * Der ‘neue Klasse’ Schalter ist eingeschaltet | | |
| **Beschreibung** | Keine eingaben | Unvollständigen Eingaben | Korrekte Eingaben |
| **Test Schritte** | 1. Das Formula ausfüllen 2. Den «Erfassen» Knopf klicken | | |
| **Test Informationen** | Name:  Hersteller:  Typ:  Beschreibung: | Name: Raspberry Pi 4 Model B  Hersteller: Raspberry pi  Typ:  Beschreibung: | Name: Raspberry Pi 4 Model B  Hersteller: Raspberry pi  Typ: Raspberry Pi  Beschreibung: Micro Rechner |
| **Erwartete Resultate** | Der Knopf ist nicht anklickbar | Der Knopf ist nicht anklickbar | Status 200: material erfolgreich erfasst. |
| **Erreichte Resultate** |  |  |  |
| **Kommentar** |  |  |  |
| **Fehlerklasse** |  |  |  |

Tabelle 14-15: Testfall 2: Material erfassen

#### Testfall 3: Material Erfassen

|  |  |  |  |
| --- | --- | --- | --- |
| **Testfall ID** | T3 | | |
| **Testmethode** | Whitebox | | |
| **Testkategorie** | K2 (Inventar) | | |
| **Testfall** | Material Erfassen | | |
| **Voraussetzungen** | * Einen Benutzer mit der rolle Lehrer ist eingeloggt * Man ist auf der ‘Material erfassen’ Seite * Der ‘neue Klasse’ Schalter ist nicht eingeschaltet | | |
| **Beschreibung** | Keine eingaben | Unvollständigen Eingaben | Korrekte Eingaben |
| **Test Schritte** | 1. Einen Benutzer mit der rolle Lehrer ist eingeloggt 2. Man ist auf der ‘Material erfassen’ Seite 3. Der ‘neue Klasse’ Schalter ist eingeschaltet | | |
| **Test Informationen** |  | Seriennummer: 1133224  Standort: | Seriennummer: 1133224  Standort: LOA03 |
| **Erwartete Resultate** | Der Knopf ist nicht anklickbar | Der Knopf ist nicht anklickbar | Status 200: material erfolgreich erfasst. |
| **Erreichte Resultate** |  |  |  |
| **Kommentar** |  |  |  |
| **Fehlerklasse** |  |  |  |

Tabelle 14-16: Testfall 3: Material Erfassen

#### Testfall 4: Material löschen

|  |  |  |
| --- | --- | --- |
| **Testfall ID** | T4 | |
| **Testmethode** | Whitebox | |
| **Testkategorie** | K2 (Inventar) | |
| **Testfall** | Material löschen | |
| **Voraussetzungen** | * Einen Benutzer mit der rolle Lehrer ist eingeloggt * Man ist auf der ‘Inventar’ Seite | |
| **Beschreibung** | Kein Produkt auswählen | Ein Produkt ausgewählt |
| **Test Schritte** | 1. Der ‘löschen’ Symbol klicken. 2. Seite neu laden | |
| **Test Informationen** |  | |
| **Erwartete Resultate** | Eintrag wird aus der Tabelle verschwinden | |
| **Erreichte Resultate** |  |  |
| **Kommentar** |  |  |
| **Fehlerklasse** |  |  |

Tabelle 14-17: Testfall 4: Material löschen

#### Testfall 5: Material Bearbeiten

|  |  |  |
| --- | --- | --- |
| **Testfall ID** | T5 | |
| **Testmethode** | Whitebox | |
| **Testkategorie** | K2 (Inventar) | |
| **Testfall** | Material Bearbeiten | |
| **Voraussetzungen** | * Einen Benutzer mit der rolle Lehrer ist eingeloggt * Man ist auf der ‘Inventar’ Seite * Enter, oder vom Textfeld klicken * Bestätigen | |
| **Beschreibung** | Fehlende Eingaben | Korrekte angaben |
| **Test Schritte** | 1. Unter eine beliebige Klasse auf ihre Serien Nummer klicken 2. Feld bearbeiten 3. Enter, oder vom Textfeld klicken 4. Bestätigen | |
| **Test Informationen** |  | |
| **Erwartete Resultate** | Eingabe kann nicht bestätigt werden | Auch nach neu laden der Seite Serien Nummer geändert |
| **Erreichte Resultate** |  |  |
| **Kommentar** |  |  |
| **Fehlerklasse** |  |  |

Tabelle 14-18: Testfall 5: Material Bearbeiten

#### Testfall 6: Ausleihungen machen

|  |  |  |
| --- | --- | --- |
| **Testfall ID** | T6 | |
| **Testmethode** | Whitebox | |
| **Testkategorie** | K3 (Ausleihungen) | |
| **Testfall** | Ausleihung machen | |
| **Voraussetzungen** | * Einen Benutzer ist eingeloggt * Man ist auf der ‘Inventar’ Seite Bestätigen | |
| **Beschreibung** | Kein Produkt auswählen | Ein Artikel unter der klasse ‘Raspberry Pi 4 Model B’ auswählen |
| **Test Schritte** | 1. Vom Menu auf die Inventarliste navigieren 2. wählen Sie ein oder mehrere Artikel aus der Liste aus 3. auf den ‘Ausleihen’ Knopf drücken | |
| **Test Informationen** |  | |
| **Erwartete Resultate** | ‘Ausleihen’ Knopf ist nicht anklickbar | Das ausgewählte Artikel erscheint in der Ausleihungen Seite |
| **Erreichte Resultate** |  |  |
| **Kommentar** |  |  |
| **Fehlerklasse** |  |  |

Tabelle 14-19: Testfall: 6 Ausleihungen machen

#### Testfall 7: Ausleihungen löschen

|  |  |  |
| --- | --- | --- |
| **Testfall ID** | T7 | |
| **Testmethode** | Whitebox | |
| **Testkategorie** | K3 (Ausleihungen) | |
| **Testfall** | Ausleihung machen | |
| **Voraussetzungen** | * Einen Benutzer ist eingeloggt * Man ist auf der ‘Inventar’ Seite Bestätigen | |
| **Beschreibung** | Kein Artikel | Einen Artikel |
| **Test Schritte** | 1. wählen keine oder mehrere Artikel aus der Liste aus 2. auf den ‘Ausleihen’ Knopf drücken | |
| **Test Informationen** | Kein Artikel ausgewählen | Ein Artikel unter der klasse ‘Raspberry Pi 4 Model B’ auswählen |
| **Erwartete Resultate** | Knopf nicht anklickbar | Das ausgewählte Artikel erscheint in der Ausleihungen Seite |
| **Erreichte Resultate** |  |  |
| **Kommentar** |  |  |
| **Fehlerklasse** |  |  |

Tabelle 14-20: Testfall 7: Ausleihungen löschen

# Realisierung

## Projektumgebung

In das 04\_projekt Verzeichnis werden 3 Unterverzeichnisse kreiert. API, database und frontend.

### API (Backend)

Als Vorarbeit wurde Node auf der lokalen Maschine installiert. Ebenso wurde die Vue-CLI global installiert.

Es wird ein package.json Datei angelegt.

>npm init

In das API-Verzeichnis werden die nötigen Abhängigkeiten installiert:

>npm i --save express Bcrypt jsonwebtoken knex mysql

Weitere Abhängigkeiten werden für entwicklungszwecke installiert.

>npm i -D nodemon eslint babel-eslint

* JsonWebToken

Dient zur Authentifizierung des Users.

* Knex

Dient zur Datenbankabfragen

* Eslint / babel-eslint

verwende ich, um mein Coding-Style konsistent zu behalten und Coding-Conventions einzuhalten

* Nodemon

verwende ich, um den code automatisch neuzuladen, wenn Änderungen gemacht werden.

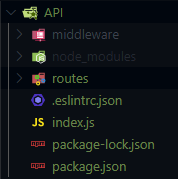
* Bcrypt

Um Bcrypt korrekt zu verwenden muss eine bestimmte Version installiert werden. Diese häng von der Node Version ab. Mit dem folgenden Kommando kann man seine aktuelle Node Version ermitteln

>node -v

Die entsprechenden Versionen von Bcrypt finden Sie hier: <https://www.npmjs.com/package/bcrypt>. Diese muss man manuell in die package.json nachtragen.

#### Grundgerüst

* **/middleware**

Hier werden Middleware Funktionen hinterlegt. Diese werden von dem index.js importiert.

* **/routes**

Hier werden die verschiedenen routen hinterlegt. Diese werden von dem index.js importiert.

* **/index.js**

Entry point

### Frontend

Mit der Vue CLI kann man durch eine UI sein Projekt Initialisieren. Es wird automatisch die richtige Ordner Struktur kreiert. In der UI kann man die packages und Plugins aussuchen, die man für das Projekt brachen will, aussuchen.

>vue UI

Für dieses Projekt werdende ich folgende Vue-plugins

* **Vuex**

Global Store

* **Vuex-persistedstate**

Erleichtert den Speicher in der Lokalen Browser Storage

* **Pug.js**

HTML Templater. Erleichtert das Schreiben von HTML.

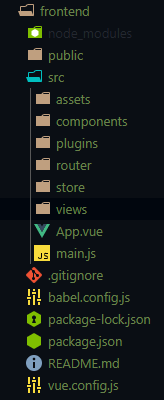
* **Vue-router**

Offizieller Vue Router Plugin

* **Vuetify**

Vuetify ist ein Materialdesign-Komponenten-Framework für Vue.js.

#### Grundgerüst

* **Public**

Globale Ressourcen die von alle Seiten zugreifbar sind

* **Assets**

Bilder und logs werden hier hinterlegt

* **Components**

Hier werden eigene Komponente hinterlegt

* **Plugins**

Hier werden die Plugins hinterlegt

* **Router**

Hier werden die verschieden routen hinterlegt.

* **Store**

Globaler Speicherort von Vuex

* **Views**

Hier werden die effektiven Seiten hiterlegt.

* **App.vue**

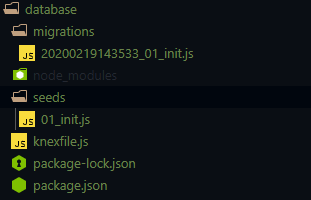
Die Startseite. Vergleichbar mit index.html bei traditioneller HTML code

* **Main.js**

Entry Point des Programms

### Datenbank

In der /database Verzeichnis werden die Knex Skripts abgelegt.

In den Unterverzeichnis /Migration sind die Skripts abgelegt, die zur erstellen der Datenbank zuständig sind abgelegt. Um die Reihenfolge zu behalten tut Knex automatisch ein Zeitstempel vor den Namen setzen des Files setzen

* **/01\_init.js**

Alles Tabellen werden erstellen.

* **/02\_relations.js**

Beziehungen werden aufgestellt.

Testdaten werden in der /seeds Verzeichnis abgelegt

* **/01\_init.js**

Alles Testaten werden eingefügt

Abbildung 16: Datenbank Ordnerstruktur

## Datenbank

Die SQL-Skripts werden mit Knex erstellt. Für sämtliche Tabellen liegen vollständige und korrekte Knex Skripts im Anhang vor. Siehe folgende Code-Dateien:

|  |  |
| --- | --- |
| Dateiname | Beschreibung |
| 20200219143533\_01\_init.js | Erstellt alle Tabellen. |
| 20200220082855\_02\_relations.js | Erstellt die Beziehungen zwischen die Tabellen |

Tabelle 15-1: SQL-Skripts

### Aufbau

Die Umsetzung des Konzepts erfolgte mit Knex, einem Query-Building-Tool für JavaScript.

In dem Verzeichnis /database wurde Knex als Abhängigkeit installiert.

Beispiel von ein Knex Skript:

**await** knex**.***schema***.**createTable('roles', **table** **=>** {

      table**.**integer('PK\_roles\_ID')**.**primary()

      table**.**string('rolesName')

});

In diesem Beispiel erstellen wir eine Tabelle mit dem Namen Rollen.

Diese Tabelle hat zwei Attribute, PK\_roles\_ID und rolesName. Durch das Setzen von '.primary()' hinter das Attribut wird es eindeutig und primär gesetzt. Knex tut aus den oberen Skript ein SQL-Query generieren.

create table `roles` (`PK\_roles\_ID` int, `rolesName` varchar(255));

alter table `roles` add prmary key `roles\_pkey`(`PK\_roles\_ID`);

Ein Problem mit Knex, das ich beheben musste, war, dass die Methode .increments() automatisch das Attribut auf unsigned gesetzt hat. In MySQL ist es nicht möglich Abhängigkeiten mit unsigned Werte zu machen. Ich habe online nach Leuten gesucht, die das gleiche Problem hatten, aber niemanden gefunden. Ich wusste von einem anderen Pxrojekt, dass ich mit Knex durchgeführt hatte, dass es möglich war, ganze SQL-Abfragen zur Ausführung einzufügen. Die einzige Lösung war also die Tabelle mit der Methode .integer() zu erstellen und dann die Tabelle mit SQL in AUTO\_INCREMENT zu ändern.

**await** knex**.**raw("ALTER TABLE `inventory`.`users` CHANGE COLUMN `PK\_users\_ID`

`PK\_users\_ID` INT (11) NOT NULL AUTO\_INCREMENT")

### Abweichungen vom Konzept

Es gab doch ein paar Änderungen. Ich gemerkt, dass ich die itemClass Tabelle keinen Namen gegeben habe. Der Name wird gebraucht um, bei material erfassen die itemClass leichter zu unterscheiden. Ausserdem waren die Tabellen: manufacturers und types Singular geschrieben.

Die Grösste Abweichung war, den Standort zu in die Items Tabelle verschieben. Das ist weil, nach einer Besprechung mit Stefano, sind wir zu dem Entschluss gekommen, dass es wenig Sin macht den Standort bei der Klasse abzulegen, da nicht alle Artikel der Klasse an dem gleichen Ort abgelegt werden.

### Testdaten

In diesen Code-Ausschnitt werden die Testbenutzer definiert. Dabei wird über ein Array von Benutzer iteriert und für jedes Objekt in das Array ein Eintrag erstellt.

var users **=** [

      {

         username: 'scjo',

         password: 'kek',

         role: 1,

      },

      {

         username: 'peju',

         password: 'iluvtfBern',

         role: 1,

      },

      {

         username: 'sika',

         password: 'okBoomer',

         role: 1,

      },

      {

         username: 'in17scma',

         password: 'gibbix12345',

         role: 2,

      },

      {

         username: 'in19sika',

         password: 'passwort1',

         role: 2,

      },

      {

         username: 'In18frch',

         password: 'Csgopro1337',

         role: 2,

      },

   ]

**await** knex('users')**.**del()

**.**then(async function () {

**for** (var user **of** users)

**await** knex('users')**.**insert([

               {

                  username: user**.***username*,

                  password: **await** bcrypt**.**hash(user**.***password*, 10),

                  FK\_roles\_ID: user**.***role*,

               },

            ]);

      });

};

### Datenbankschema

In der Unteren Abbildung ist die Umgesetzte Datenbank sichtbar. Diese wurde mit von der Laufenden Datenbank Reverse-Engineered und dann in Lucidchart abgebildet

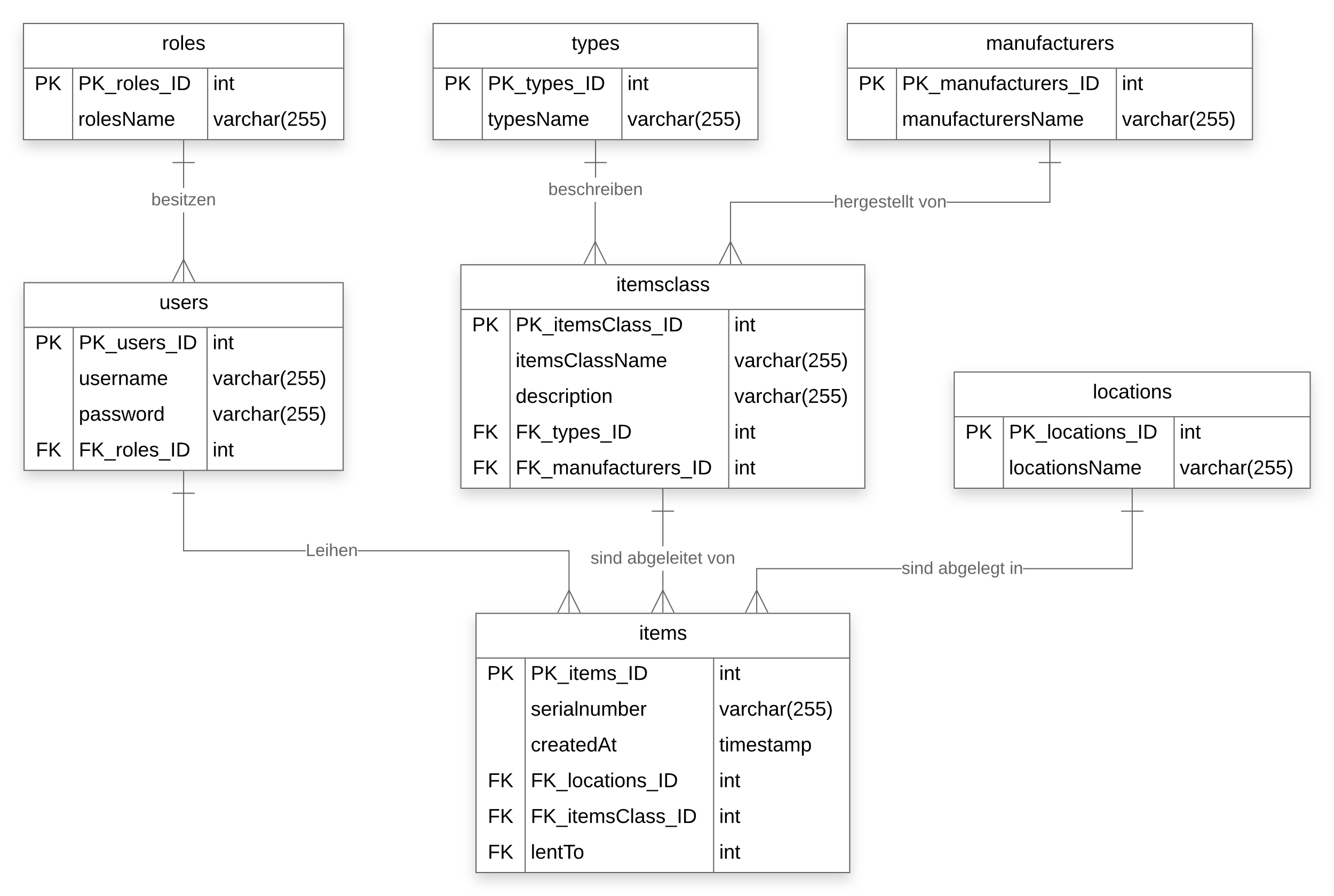


Abbildung 17: Datenbankschema

### Beziehungen

|  |  |  |  |
| --- | --- | --- | --- |
| Tabelle | Kardinale Beziehung | Tabelle | Beziehung |
| roles | 1 - m | users | besitzen |
| users | c - mc | Items | leihen |
| Items | 1 - m | itemsClass | sind abgeleitet von |
| ItemsClass | mc - 1 | Manufcturers | hergestellt von |
| Types | 1 - mc | itemsClass | beschreiben |
| ItemsClass | mc - 1 | Manufcturers | sind abgelegt in |

Tabelle 15-2: Datenbank Beziehungen

## design

### pug.js

Pug.js ist ein Sprache, die zu HTML kompiliert. Ich habe diese Sprache gebraucht, weil ich in der Vergangenheit diese zum Schreiben von HTML gebraucht habe und mir sehr gefiel. Pug.js ist eine ‘Whitespace sensitive’ Sprache. Hier ist ein Beispiel von der GitHub-Seite von Pug.js:

   doctype html

   html(*lang*="en")

   head

      title= pageTitle

      script(*type*='text/javascript').

**if** (foo) bar(1 **+** 5)

   body

      h1 Pug - node template engine

*#container.col*

**if** youAreUsingPug

         p You are amazing

**else**

         p Get on it!

         p.

         Pug is a terse and simple templating language with a

         strong focus on performance and powerful features.

Pug.js verwandelt den Obern Code Stück zu:

<!DOCTYPE *html*>

<html *lang*="en">

<head>

   <title>Pug</title>

   <script *type*="text/javascript">

**if** (foo) bar(1 **+** 5)

   </script>

</head>

<*body*>

   <*h1*>Pug **-** node template engine**</**h1**>**

   <div *id***=**"container" *class***=**"col">

      <p>You are amazing</p>

      <p>Pug is a terse and simple templating language with a strong focus on performance and powerful features.</p>

   </div>

**</**body**>**

**</**html**>**

### Material erfassen

Um Material Klassen und Materialien zu unterscheiden gibt es sein switch, welches ein anderes Formular aufruft. Bei Beiden Formulare kann man die abfrage nur bestätigen, wenn alle Felder ausgefüllt sind. Da beider Funktionen sehr ähnlich sind, wird nur die für Material erfassen abgebildet.

Unten ist das Formular abgebildet, welches gebraucht wird, um Material Klassen zu erfassen.

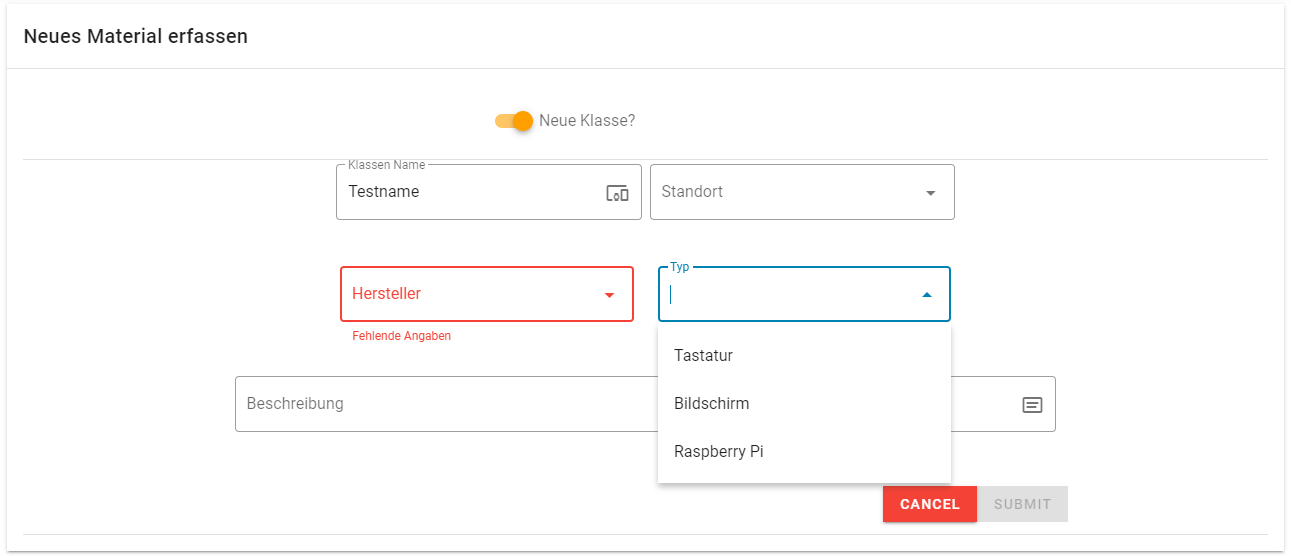


Abbildung 18: Material Klassen erfassen



Hier abgebildet ist das Formular, um neues Material zu erfassen. Das ‘Material Klasse’ Feld, wenn angeklickt, verwandelt sich zu einem Dropdown-Menu, bei dem man die Klasse auswählen kann

Um Vorschläge im Drop-Down-Menu zu machen, müssen die Daten vom Backend geholt werden.

Abbildung 19: Material erfassen

Diese Line code **if**(*this***.**$refs**.***form***.**validate() ruft eine Funktion vom v-form Komponent auf, welches die eingaben validiert und ein Boolean zurückgibt.

Die Daten werden aus dem ‘form’ Objekt geholt. Dann wird, mit axios die Anfrage gestellt, wobei, die Daten als Parameter mitgegeben werden.

async submitNewItem() {

**if**(*this***.**$refs**.***form***.**validate()) {

**try** {

               var data **=** {

                  serialNumber: *this***.***form***.***serialNumber*,

                  FK\_locations\_ID: *this***.***form***.***locationsId*,

                  FK\_itemsClass\_ID: *this***.***form***.***itemsClassId*

               }

**await** axios()**.**post('/teacher/items', data)

*this***.**$emit("message", {

                  type: "success",

                  text: 'Eintrag erstellt',

                  timeout: 1000

               });

*this***.**$refs**.***form***.**reset()

            } **catch** (error) {

               console**.**error(error);

*this***.**$emit("message", {

                  type: "error",

                  text: error**.***message*,

                  timeout: 0 });

            }

         }

      }

#### Diagramm

Das Diagramm beschreibt den Ablauf, um Daten zu erfassen.

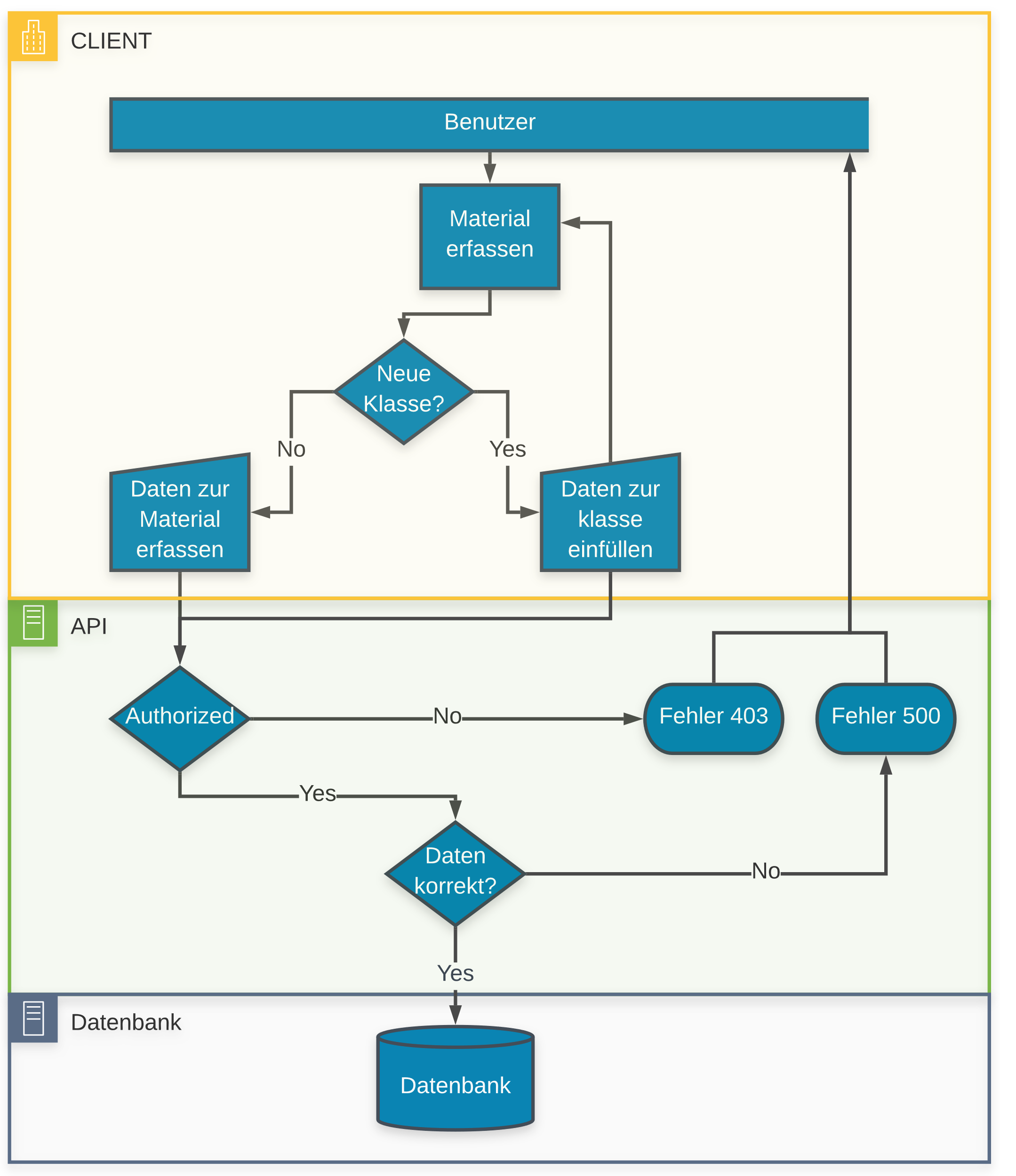


Abbildung 20: Material erfassen

### Material ausleihen

Um material auszuleihen wählt man das Item aus, welches man ausleihen will, dann klickt auf das Ausleihen Knopf, oben rechts in der Ecke.

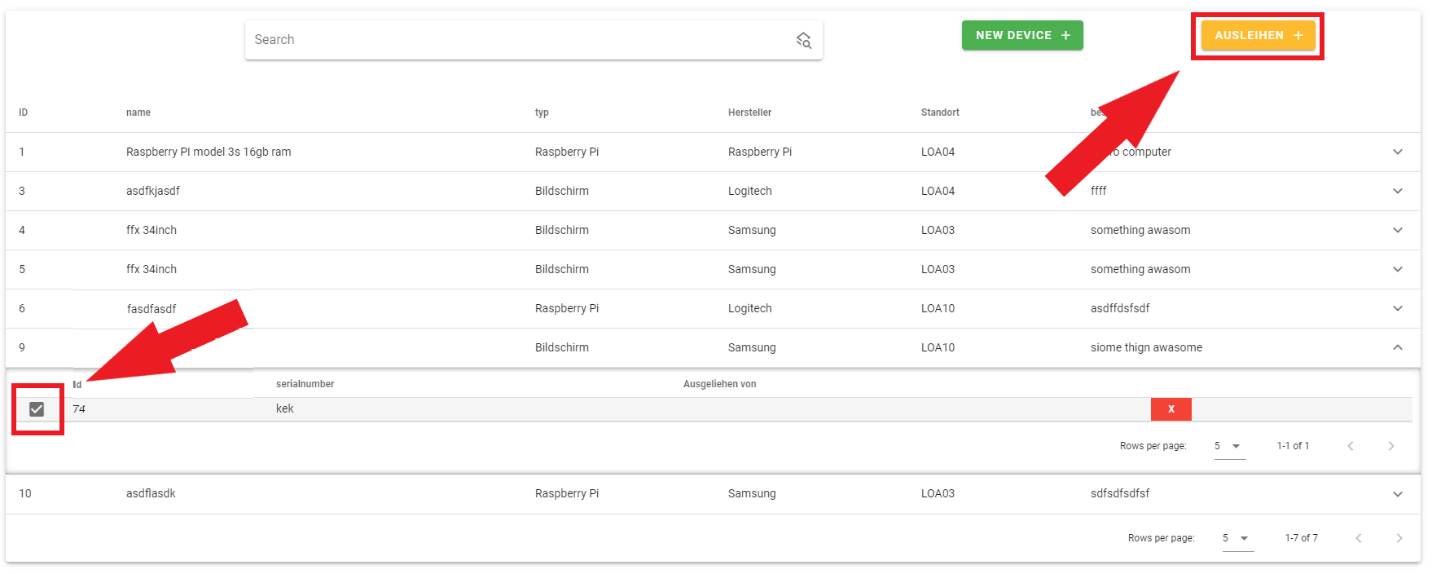


Abbildung 21: Material ausleihen

In diesen Code-Ausschnitt wird der Ausleihprozess im Frontend gemacht. Zuerst formt es die Daten um, so das es das Backend richtig verarbeiten kann. Der HTML-Request erfolgt mit Axios und sendet die ID-liste mit den ausgewählten Items mit. Danach erfolgt die Fehlerbehandlung. Zuletzt werden die auserwählten Items abgewählt, und die Liste neu geladen.

async lendItems() {

         var idList **=** *this***.***selectedItems***.**map(**item** **=>** item['PK\_items\_ID'])

**try** {

**await** axios()**.**post('/student/lendings/', idList )

*this***.**$emit("message", {

               type: "success",

               text: 'Material ausgeliehen',

               timeout: 1000

            });

         } **catch** (error) {

            console**.**error(error);

*this***.**$emit("message", {

               type: "error",

               text: error**.***message*,

               timeout: 0

            });

         }

*this***.***selectedItems* **=** [];

*this***.**loadItems()

      }

Hier ist der Code vom Backend ersichtlich welches für die Ausleihungen zuständig ist.

Es wird die Benutzer-ID und Array mit IDs entgegengenommen., dann wird für jede ID in das Array eine Datenbankanfrage an die Datenbank gemacht.

In Knex update Methoden wird immer 0 oder 1 als Ausgabe gegeben. Deshalb werden die Errors abgefangen, da es von sich keine Errors ausgibt.

Wenn alles wie es sollte funktioniert, wird der Status 200 gesendet

router**.**post('/lendings', async (**req**, **res**) **=>** {

*// id of user from taken from token*

   var userId **=** req**.***decodedToken***.***userId*;

*// list of ids to lend*

   var idList **=** req**.***body*;

**try** {

**for** (var id **of** idList) {

*// database request*

         var result **=** **await** knex('items')

**.**where('lentTo', null)

**.**where('PK\_items\_ID', id)

**.**update({ lentTo: userId });

**if** (result **===** 0) {

*// log error*

            console**.**error('no entry found ');

*// send Status 500*

            res**.**sendStatus(500);

         } **else** {

*// send Status 200*

            res**.**sendStatus(200);

         }

      }

*// error handling*

   } **catch** (error) {

*// log error*

      console**.**error(error);

*// send Status 500*

      res**.**sendStatus(500);

   }

});

### Nachrichten

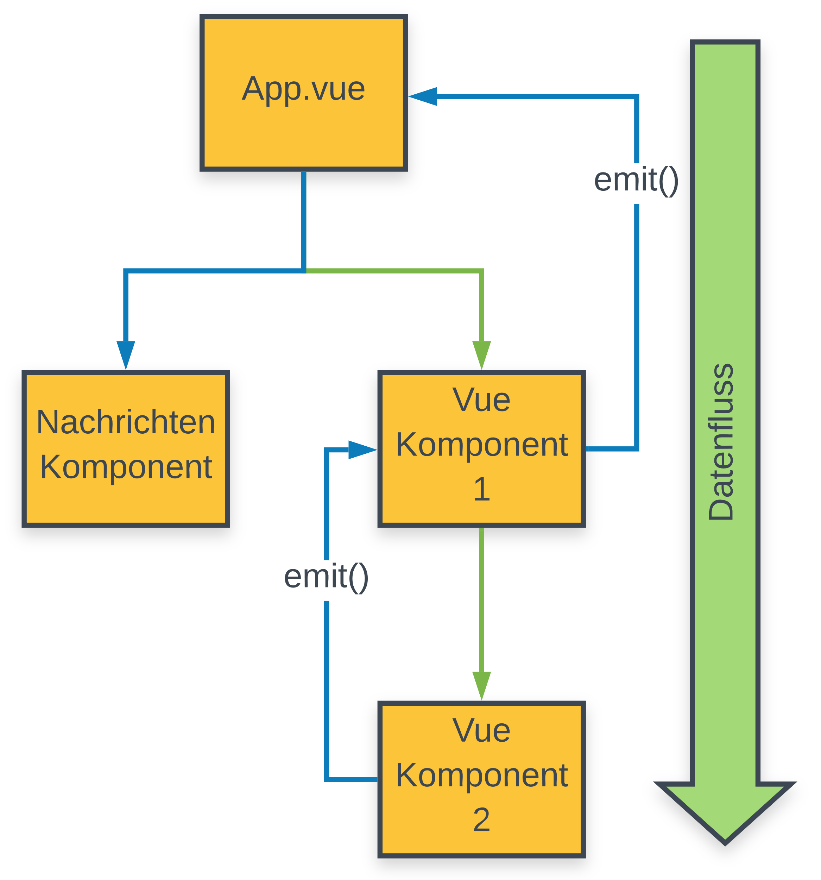
Nachrichten werden im obersten Vue Komponent dargestellt und verwaltet. Aber wegen der Hierarchischen Aufbau von Vue ist es schwer daten von unter-komponente anzuzeigen. Ein Variante wäre mit emit(). Im Grunde ist emit() ein Eventlistener der in der obersten Komponente ist, welches mit einem Funktionsaufruf verbunden ist, das Daten als Argument aufnimmt. Die Unterkomponente löst dann den Listener aus und übergibt die Daten.

Abbildung 22: Nachrichten Flussdiagramm

#### Aufbau

Diese Methode tut die Nachricht es bekommt zu ein Array pushen, welches dann mit der UI angezeigt wird.

message(**message**) {

*this***.***snacks* **=** (**!***this***.***snacks*) **?** [] **:** *this***.***snacks*

*this***.***snacks***.**push(message);

      },

Das Vuetify-Komponent sieht so aus.

v-snackbar(

*top*

*v-for*="snack **in** snacks"

         :*key*='snacks**.**indexOf(snack)'

*v-model*="snacks"

         :*timeout*='snack**.***timeout*'

         :*color*='snack**.***type*'

      ) {{ snack**.***text* }}

         v-btn( @*click*="snacks**.**splice(snacks**.**indexOf(n)), 1" *dark* *text*) close

In diesen Kodeausschnitt wird eine Nachricht von child-component in die parent-component geschickt.

*this***.**$emit("message", { type: "error", text: error**.***message*, timeout: 0 });

#### Fehlermeldung:

Fehlermeldungen werden so ausgegeben

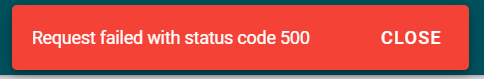


Abbildung 23: Fehlermeldung

#### Erfolgsmeldung

Erfolgsmeldungen werden so ausgegeben



Abbildung 24: Erfolgsmeldung

### Material anzeigen

Um die untergeordnete Table zu erstellen habe ich zweimal das gleiche Komponent gebraucht.

Das ‘template’ Element kann gebraucht werden um, die Standard Komponente, die von Vuetify Gesetz sind, mit selbstdefinierte Komponente zu ersetzen. In diesen Fall habe ich eine weitere Tabelle hinzugefügt.

v-data-table(

*:search*="search"

*:items-per-page*='999'

*item-key*='PK\_itemsClass\_ID'

*:expanded.sync*="expanded"

*show-expand*

*:loading*='loading'

*:items*='items'

*:headers*='headers'

)

   template( *v-slot:expanded-item*="{ headers, item }" )

      td( *style*="padding:5px 10px;" :*colspan***=**'headers.length')

         v-data-table(

*dense*

*:search*="search"

*show-select*

*item-key*='PK\_items\_ID'

*:items-per-page*=1000

*class*="elevation-0"

*:items*='item.items'

*:headers*='subheaders'

*hide-default-footers*

*v-model*="selectedItems"

         )

Hier sieht man das Ergebnis:

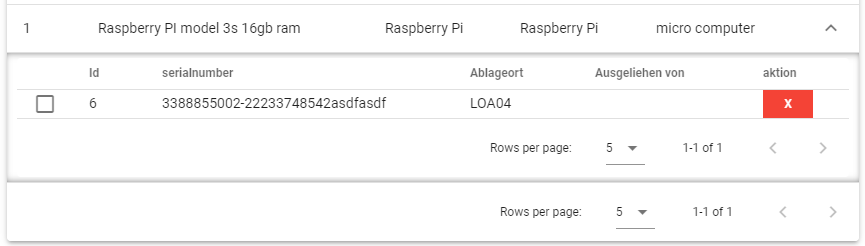


Abbildung 25: Material Anzeigen Komponent

## Programmstruktur

Abbildung 26: Frontend Package Diagramm

### Frontend Package Diagramm

Rechts ist das Frontend Diagramm dargestellt.

#### Plugins

Plugins werden Vue gebraucht, um Tätlichkeiten zu erledigen, die nicht in Vue standardmässig abgedeckt sind.

Ich habe. Das Visual Framework Vuetify gebraucht, um vorgemachte Komponente zu brauchen, welches die Arbeit erheblich erleichtert.

Pug.js wird in einen Abschnitt weiter unten erklärt.

#### Components und Views

Components sind wiederverwendbare Vue-Instanzen mit einem Namen. Wir können diese Komponente als ein benutzerdefiniertes Element innerhalb einer Root-Vue-Instanz verwenden

Views sind die Webseiten, die im Browser angezeigt werden. Diese, werden aber durch den Router Verwalten, welches wiederum in Main.js verwaltet wird. Deshalb kann man auch von Components reden, da sie auch wiederverwendbare Vue-Instanzen mit einem Namen sind.

#### Packages

Packages sind von npm ausgeben Code-Ausschnitte.

### Backend Package Diagramm

Abbildung 27: Backend Package Diagramm

Hier abgebildet ist das Backend Package Diagramm. Die Pfeile repräsentieren die Abhängigkeiten

Daer Index ist das Root-Programm. Von hier werden die Aufgaben auf die verschiedenen Module delegiert. Die effektive Route-Handling und die Konsolen Ausgaben werden hier gemacht.

#### Routes

Hier sind die Endpunkte definiert.

#### Middleware

Das Middleware, ist in meinem Fall zuständig für die Autorisierung der Benutzer und Verifizierung des Tokens.

#### Knex.js

Das Knex.js sorgt dafür das die SQL abfragen mit der Richtigen Konfiguration gemacht werden.

var config **=** {

   client: 'mysql',

   version: '8.0',

   connection: {

      host: 'localhost',

      database: 'database',

      user: 'root',

      password: 'password',

      insecureAuth: true,

      port: 3306

   }

};

## API

### API Flowchart

Unten wird der Lebenslauf ein Anfragen an der API, durch ein Flowchart dargestellt.

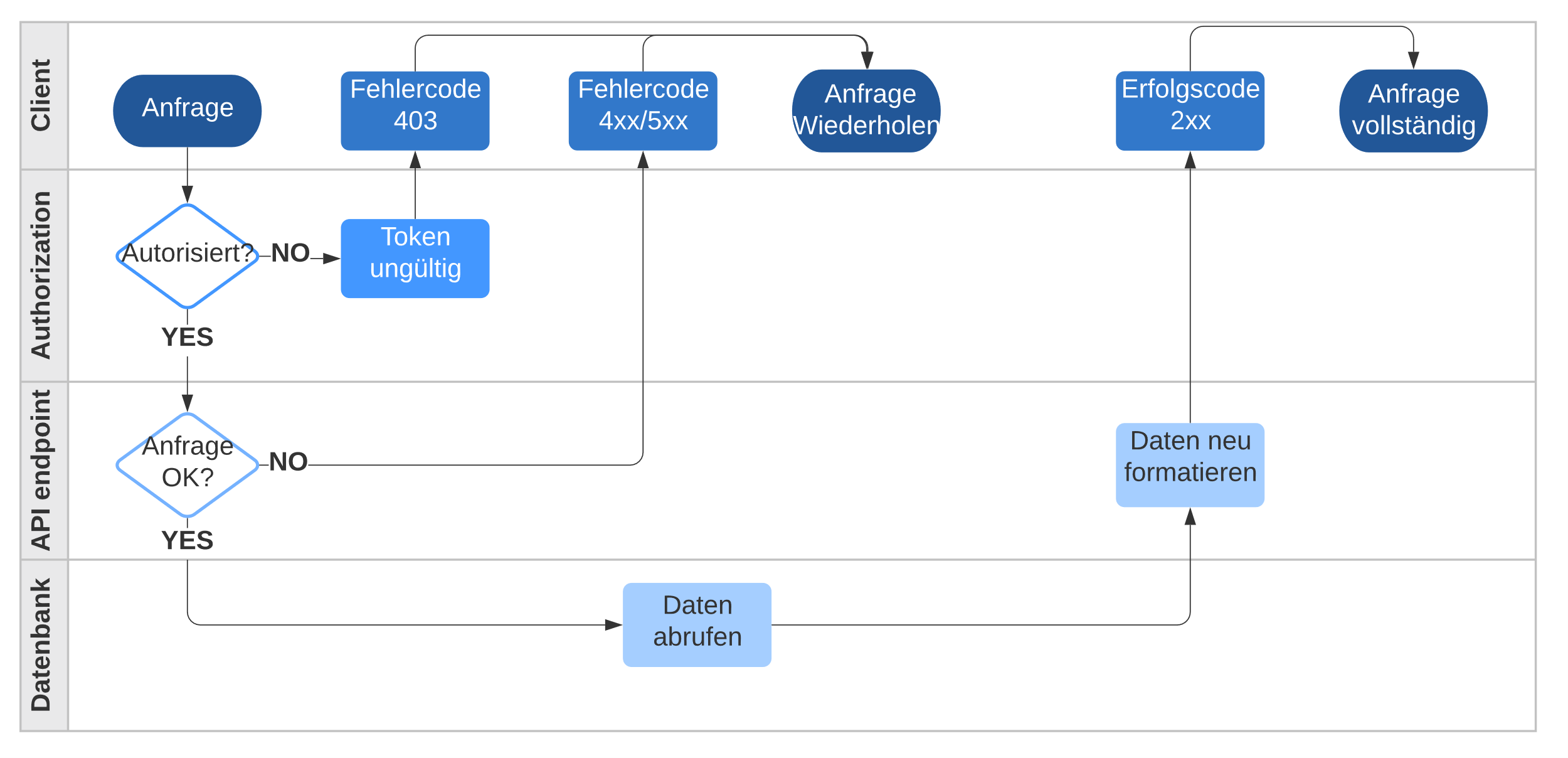


Abbildung 28: API Flowchart

### Öffentliche Schnittstellen

Schnittstellen die öffentlich sind.

|  |  |  |
| --- | --- | --- |
| Methode | Endpunkt | Beschreibung |
| POST | /login | Benutzerdaten werden an die API geschickt und ein Token wird zurückgeschickt. |

Tabelle 15-3: Öffentliche Schnittstellen

### Schüler Schnittstellen

Nur ein eingeloggter Benutzer kann auf diese Endpunkte gelangen.

|  |  |  |
| --- | --- | --- |
| Methode | Endpunkt | Beschreibung |
| GET | /inventory | Er kommen daten zur Inventar zurück. |
| GET | /lendings | Ausleihungen vom Benutzer werden ausgegeben. |
| POST | /lendings | Eine neue Ausleihung wird gemacht erstellt. |
| DELETE | /lendings/:id | ID der Ausleihung wird der URL mittgeschickt. Zurück kommt lediglich ein Status-Code. Nur eigene Ausleihungen können gelöscht werden. |

Tabelle 15-4: Schüler Schnittstellen

### Lehrkraft Schnittstellen

Nur ein eingeloggter Benutzer mit der rolle Lehrer, kann auf diese Endpunkte zugreifen.

Die Endpunkte in den dunkelten Felder werden zurzeit nicht von der Applikation verwendet.

|  |  |  |
| --- | --- | --- |
| Methode | Endpunkt | Beschreibung |
| GET | /locations | Gibt alle Standorte aus mit entsprechende ID. |
| GET | /types | Gibt alle Typen aus mit entsprechende ID. |
| GET | /manufacturers | Gibt alle Hersteller aus mit entsprechende ID. |
| GET | /itemsClass | Gibt alle Materialklassen aus mit entsprechende ID. |
| GET | /lendings | Ausleihungen vom allen Benutzer werden ausgegeben. |
| PUT | /inventory/:id | Ändert das Material. |
| POST | /items | Neues Material wird erfasset. |
| POST | /manufacturers | Erfasst einen neuen Hersteller. |
| POST | /types | Erfasst einen neuen Typ. |
| POST | /ItemsClass | Erfasst eine neue Klasse. |
| DELETE | /lendings/:id | Tut die Ausleihung auflösen. Kann für jede Ausleihung gemacht werden. |
| DELETE | /inventory/:id | Löscht ein Material. |

Tabelle 15-5: Lehrkraft Schnittstellen

## Sicherheit

### Passwörter

Für das Testen der Applikation wurden Testbenutzer erstellt. Für jeden Benutzer wird das Passwort mit Bcrypt gehascht.

### Login

Im Login-Prozess sendet der Client seine Logindaten ans Backend. Das erhaltene Passwort wird mit einer Funktion von Bcrypt mit dem gehaschten Passwort in der Datenbank verglichen. Falls die Daten übereinstimmen, bekommt der Client einen Token zurück.

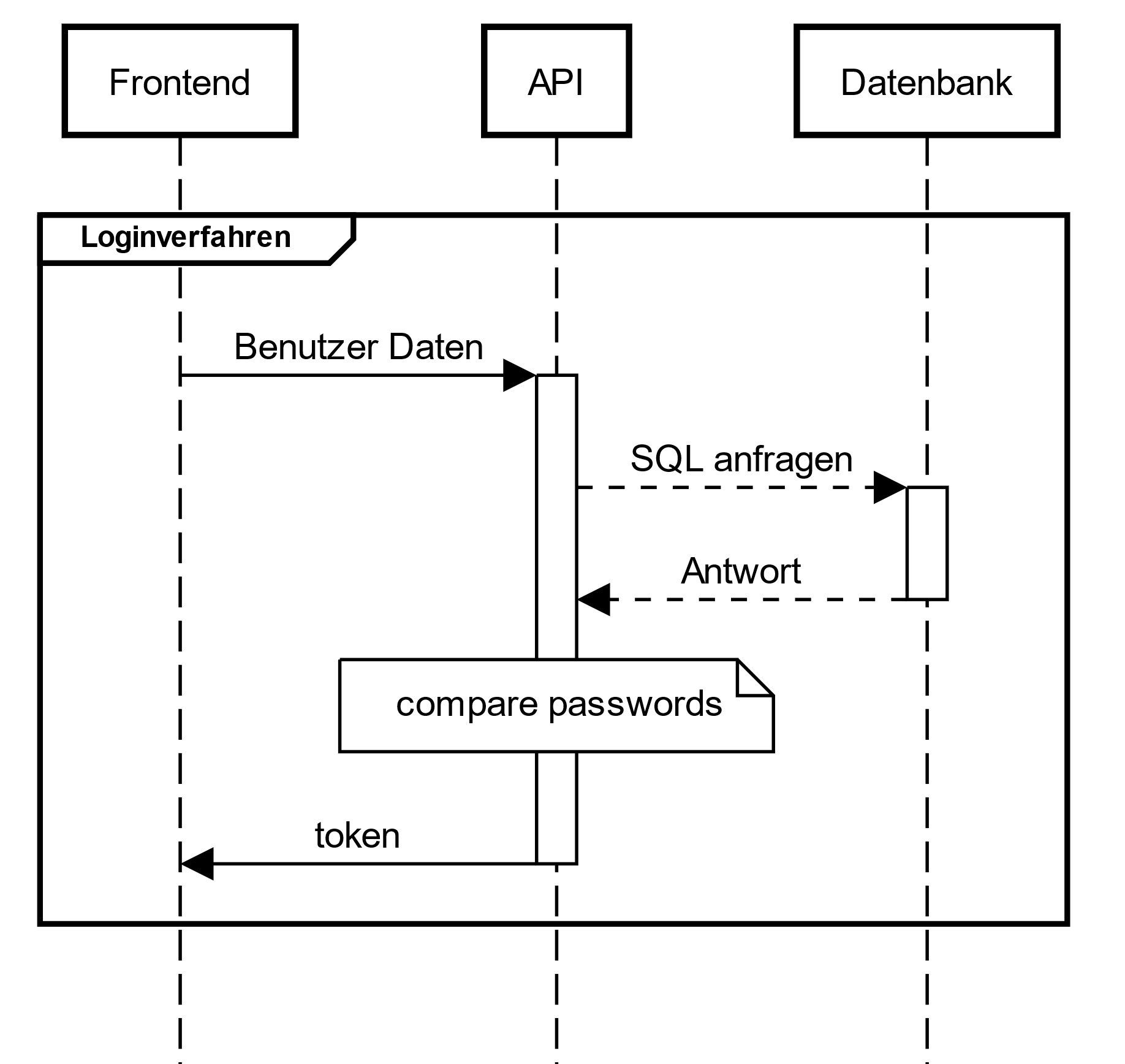


Abbildung 29: Login verfahren

### Benutzer Autorisierung

Das Token, welches von der API zurückgeschickt wird, wird von dem Frontend mit Vuex gespeichert.

Im code ist es so umgesetzt:

login: async function() {

*// if the form has valid data*

**if** (*this***.**$refs**.***form***.**validate()) {

*// send request to /login*

            var response **=** **await** axios()**.**post("/login", *this***.***data*);

*// get token from response.data*

            var token **=** response**.***data***.***auth***.**split(" ")[1];

            var userData **=** response**.***data***.***tokenData*

*// var user = response.data.user;*

*// save token in vuex store*

*this***.**$store**.**dispatch('login', {token, userData})

*// emit to parent element to display message*

*this***.**$emit("message", { type: "success", text: 'erfolgreich eingeloggt', timeout: 1000 });

*// send user to path home*

*this***.**$router**.**push("/");

         }

      }

Auf dieser Linie wird eine Funktionsaufruf in Vuex gemacht, welches die Daten in dem Store schreibt.

*this***.**$store**.**dispatch('login', {token, userData})

Vuex Store:

 login: function (**context**, **data**) {

context**.**commit('setToken', data**.***token*)

      context**.**commit('setUserData', data**.***userData*)

      context**.**commit('setLoggedIn', true)

}

Im backend wird als Middleware die checkAuthorization Funktion gebraucht, um den Benutzer zu autorisieren.

Diese Funktion wird für jeden Endpunkt, ausser den öffentlichen Schnittstellen aufgerufen.

Wenn die Validierung des Tokens erfolgreich ist, gelangt man weitergeleitet zu dem Endpunkt. Die Funktion gibt das decodierte Token im req (request) Parameter mit. Dieser kann von allen, im Stack folgende Funktionen benutzt werden. Beispielsweise um die Rolle eines Benutzers zu überprüfen, ohne das Token jede mal wieder decodieren zu müssen.

var jwt **=** require('jsonwebtoken');

async function checkAuthorization (**req**, **res**, **next**) {

   var head **=** req**.***headers***.***authorization*;

**if** (**typeof** head **!==** 'undefined') {

      var token **=** req**.***headers***.***authorization***.**split(' ')[1];

      req**.***token* **=** token;

**try** {

         req**.***decodedToken* **=** jwt**.**verify(token, process**.**env**.**JWT\_TOKEN);

      }

**catch** (error) {

         console**.**error(error);

         res**.**send('invalid token')**.**status(403);

      }

      next();

   }

**else** {

      res**.**sendStatus(403);

   }

}

### Frontend Routeguarding

Diese Funktion wird vor jeder Route Änderung aufgerufen. Zuerst werden alle Seiten definiert, die für jeden Benutzer verfügbar sind. Danach die Seiten, auf die nur die Lehrkraft zugriff, haben soll. Dann wird geschaut, ob man eingeloggt sein muss, um auf die Seite zu gelangen. Dann wird geschaut, ob man eingeloggt ist. Und schließlich, schautes, ob man ein Lehrer ist oder nicht.

Wenn man nicht eingeloggt ist und eingeloggt sein muss, um auf die Seite zu gelangen, wird man auf die Login-Seite zurückgewiesen. Wenn man eingeloggt ist und auf die Login-Seite gehen will, wird man auf das Dashboard gelangen. Und schliesslich, wenn man nicht eine Lehrkraft ist und auf eine Lehrkraft Seite gelangen will wird auch auf das Dashboard gelangen.

router**.**beforeEach((**to**, **from**, **next**) **=>** {

*// pages accessable to everyone*

   const publicPages **=** ['/login']

*// teacher routes*

   const teacherRoutes **=** ['/newMaterial']

*// if a page is not in the public pages array, it requires login*

   const authRequired **=** **!**publicPages**.**includes(to**.***path*);

*// checks if the user is loggedIn*

   const loggedIn **=** store**.***state***.***token*

*// checks if the user is a teacher*

   const isTeacher **=** store**.***state***.***role* **===** 'teacher'

*// if the user is not logged in and if authorization is required*

**if** (authRequired **&&** **!**loggedIn) {

*// send user to login page*

**return** next('/login')

*// or if the user is logged in and is going to the login page*

   } **else** **if** (loggedIn **&&** to**.***path* **===** '/login') {

*// it redirects him to the dashboard*

      next('/')

*// if not a teacher and trying to access teacher route*

   } **else** **if** (**!**isTeacher **&&** teacherRoutes**.**includes(to**.***path*)) {

      next(from**.***path*)

   }

*// if none apply it continues*

   next();

})

#### User Interface

Viele Benutzeransichten mussten für die verschiedenen Rollen angepasst werden. Weil Benutzer mit der Rolle ‘Lehrer’ Komponente brauchen können, die für Benutzer mit der Rolle ‘Schüler’ unsichtbar sein sollten. Dieses Problem kann auf verschiedene Arten gelöst werden. Ich habe es so gelöst, das einzelne Komponente bedingt sichtbar sind.

Hier ist ein Beispiel, wobei es sich um ein Knopf handelt, welches für Benutzer der Rolle ‘Schüler’ nicht sichtbar sein sollte:

template(

*v-slot:item.action*='{ item }'

*v-if*='user.role === "teacher"'

   )

   v-btn*.elevation-0*(

*@click*="deleteItem( item )"

*small*

*tile*

*color*="red"

*dark*

   ) x

Die Variable ‘user’ ist eine Funktion, die den ‘user’ von der Vuex Store abruft:

user: function() {**return** *this***.**$store**.***state***.***user*},

### Backend Routeguarding

In diesen Code-Ausschnitt werden die verschiedenen Routen definiert.

var isTeacher **=** require('./middleware/isTeacher');

var authorization **=** require('./middleware/authorization');

var studentRoutes **=** require('./routes/student');

var teacherRoutes **=** require('./routes/teacher');

var publicRoutes **=** require('./routes/public');

*// public routes*

app**.**use('/', publicRoutes);

*// student routes*

app**.**use('/student', authorization, studentRoutes);

*// teacher routes*

app**.**use('/teacher', authorization, isTeacher, teacherRoutes);

Die ‘/public’ Route hat keine Middleware Funktionen, da diese von jeder Person zugreifbar sein sollte und deshalb keine Autorisierungsfunktionen benötigt. Die ‘/student’ Route, hat ‘authorization’ als Middleware, diese wird, bevor es zu der Route gelangt aufgerufen und ist im Kapitel: 15.6.2 beschrieben. Die ‘/teacher’ Route hat zwei Middleware Funktionen, da es überprüfen muss, ob man eingeloggt ist und ob man die ‘teacher’ Rolle besitzt. Das wird mit der isTeacher Funktion gemacht.

Es benutzt das decodierte Token, um die Rolle des Benutzers abzufragen. Mit der ‘next()’ Aufruf wird der Request weitergeleitet.

function isTeacher(**req**, **res**, **next**) {

   var role **=** req**.***decodedToken***.***role*;

**if** (role **===** 'teacher') {

      next();

   } **else** {

      res**.**sendStatus(403);

   }

}

**module.exports** **=** isTeacher;

## Testfälle

### Testfall 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Testfall ID** | T1 | | | | |
| **Testmethode** | Blackbox | | | | |
| **Testkategorie** | K1 (Login) | | | | |
| **Testfall** | Login | | | | |
| **Voraussetzungen** | Kein Benutzer ist eingeloggt. Browsercache ist lehr | | | | |
| **Beschreibung** | Keine eingaben | Kein Benutzer | Kein Passwort | Richtiger Benutzer  Falsches Passwort | Richtiger Benutzer & Passwort |
| **Test Schritte** | 1. Username eingeben 2. Passwort eingeben 3. Auf Login klicken | | | | |
| **Test Informationen** | Benutzer: -  Passwort: - | Benutzer: -  Passwort: kek | Benutzer: scjo  Passwort: - | Benutzer: scjo  Passwort: 1234 | Benutzer: scjo  Passwort: kek |
| **Erwartete Resultate** | Fehlermeldung 403: Benutzer oder Passwort falsch | | | | Weiterleitung auf das Dashboard |
| **Erreichte Resultate** | Fehlermeldung Request failed with status code 403 | | | | Erwartetes Resultat erzielt |
| **Kommentar** | Nicht die genau erwartete Resultate | | | |  |
| **Fehlerklasse** | M1 | | | | M0 |

Tabelle 15-6: Testfall 1

### Testfall 2

|  |  |  |  |
| --- | --- | --- | --- |
| **Testfall ID** | T2 | | |
| **Testmethode** | Whitebox | | |
| **Testkategorie** | K2 (Inventar) | | |
| **Testfall** | Material erfassen | | |
| **Voraussetzungen** | * Einen Benutzer mit der rolle Lehrer ist eingeloggt * Man ist auf den Dashboard | | |
| **Beschreibung** | Keine eingaben | Unvollständigen Eingaben | Korrekte Eingaben |
| **Test Schritte** | 1. Auf die ‘Material erfassen’ Knopf klicken 2. den Material Klasse Textfeld ausfüllen 3. Den «Erfassen» Knopf klicken | | |
| **Erwartete Resultate** | Das Formular ist nicht validiert. | | Status 200: material erfolgreich erfasst. |
| **Erreichte Resultate** | Das Formular ist nicht validiert. | | Status 200: material erfolgreich erfasst. |
| **Kommentar** |  | |  |
| **Fehlerklasse** | M0 | | M0 |

Tabelle 15-7: Testfall 2

### Testfall 3

|  |  |  |  |
| --- | --- | --- | --- |
| **Testfall ID** | T3 | | |
| **Testmethode** | Whitebox | | |
| **Testkategorie** | K2 (Inventar) | | |
| **Testfall** | Material Erfassen | | |
| **Voraussetzungen** | Ein Benutzer mit der Rolle ‘Lehrer’ ist eingeloggt | | |
| **Beschreibung** | Keine eingaben | Unvollständigen Eingaben | Korrekte Eingaben |
| **Test Schritte** | 1. Einen Benutzer mit der rolle Lehrer ist eingeloggt 2. Man ist auf der ‘Material erfassen’ Seite 3. Der ‘neue Klasse’ Schalter ist eingeschaltet | | |
| **Test Informationen** | Keine | Seriennummer: 1133224  Standort: | Seriennummer: 1133224  Standort: LOA03 |
| **Erwartete Resultate** | Der Knopf ist nicht anklickbar | Der Knopf ist nicht anklickbar | Status 200: material erfolgreich erfasst. |
| **Erreichte Resultate** | Das Formular ist nicht validiert. | Das Formular ist nicht validiert. | Status 200: material erfolgreich erfasst. |
| **Kommentar** | Nicht die genau erwarteten Resultate. | | |
| **Fehlerklasse** | M1 | M1 | M0 |

Tabelle 15-8: Testfall 3

### Testfall 4

|  |  |  |
| --- | --- | --- |
| **Testfall ID** | T4 | |
| **Testmethode** | Whitebox | |
| **Testkategorie** | K2 (Inventar) | |
| **Testfall** | Material löschen | |
| **Voraussetzungen** | * Einen Benutzer mit der rolle Lehrer ist eingeloggt * Man ist auf der ‘Inventar’ Seite | |
| **Beschreibung** | Kein Produkt auswählen | Ein Produkt ausgewählt |
| **Test Schritte** | 1. Das ‘löschen’ Symbol klicken. 2. Seite neu laden | |
| **Test Informationen** |  | |
| **Erwartete Resultate** | Eintrag wird aus der Tabelle verschwinden | |
| **Erreichte Resultate** | Eintrag verschwinden aus der Tabelle | |
| **Kommentar** |  | |
| **Fehlerklasse** | M0 | |

Tabelle 15-9: Testfall 4

### Testfall 5

|  |  |  |
| --- | --- | --- |
| **Testfall ID** | T5 | |
| **Testmethode** | Whitebox | |
| **Testkategorie** | K2 (Inventar) | |
| **Testfall** | Material Bearbeiten | |
| **Voraussetzungen** | * Einen Benutzer mit der rolle Lehrer ist eingeloggt * Man ist auf der ‘Inventar’ Seite * Enter, oder vom Textfeld klicken * Bestätigen | |
| **Beschreibung** | Fehlende Eingaben | Korrekte angaben |
| **Test Schritte** | 1. Unter eine beliebige Klasse auf ihre Serien Nummer klicken 2. Feld bearbeiten 3. Enter, oder vom Textfeld klicken 4. Bestätigen | |
| **Test Informationen** |  | |
| **Erwartete Resultate** | Eingabe kann nicht bestätigt werden | Auch nach neu laden der Seite Serien Nummer geändert |
| **Erreichte Resultate** | Eingabe wird angenommen | Eingabe wird angenommen |
| **Kommentar** | Muss noch angepasst werden |  |
| **Fehlerklasse** | M1 | M0 |

Tabelle 15-10: Testfall 5

### Testfall 6

|  |  |  |
| --- | --- | --- |
| **Testfall ID** | T6 | |
| **Testmethode** | Whitebox | |
| **Testkategorie** | K3 (Ausleihungen) | |
| **Testfall** | Ausleihung machen | |
| **Voraussetzungen** | * Einen Benutzer ist eingeloggt * Man ist auf der ‘Inventar’ Seite Bestätigen | |
| **Beschreibung** | Kein Produkt auswählen | Ein Artikel unter der klasse ‘Raspberry Pi 4 Model B’ auswählen |
| **Test Schritte** | 1. Vom Menu auf die Inventarliste navigieren 2. wählen Sie ein oder mehrere Artikel aus der Liste aus 3. auf den ‘Ausleihen’ Knopf drücken | |
| **Test Informationen** |  | |
| **Erwartete Resultate** | ‘Ausleihen’ Knopf ist nicht anklickbar | Das ausgewählte Artikel erscheint in der Ausleihungen Seite |
| **Erreichte Resultate** | ‘Ausleihen’ Knopf ist nicht anklickbar | Der ausgewählte Artikel erscheint in der Ausleihungen Seite. Das Kästchen ist nicht auswählbar |
| **Kommentar** |  | |
| **Fehlerklasse** | M0 | M0 |

Tabelle 15-11: Testfall 6

### Testfall 7

|  |  |  |
| --- | --- | --- |
| **Testfall ID** | T7 | |
| **Testmethode** | Whitebox | |
| **Testkategorie** | K3 (Ausleihungen) | |
| **Testfall** | Ausleihung machen | |
| **Voraussetzungen** | * Einen Benutzer ist eingeloggt * Man ist auf der ‘Inventar’ Seite Bestätigen | |
| **Beschreibung** | Kein Artikel | Einen Artikel |
| **Test Schritte** | 1. wählen keine oder mehrere Artikel aus der Liste aus 2. auf den ‘Ausleihen’ Knopf drücken | |
| **Test Informationen** | Kein Artikel ausgewählen | Ein Artikel unter der klasse ‘Raspberry Pi 4 Model B’ auswählen |
| **Erwartete Resultate** | Knopf nicht anklickbar | Das ausgewählte Artikel erscheint in der Ausleihungen Seite |
| **Erreichte Resultate** | Knopf nicht anklickbar | Der ausgewählte Artikel erscheint in der Ausleihungen Seite. Benutzername erscheint bei der Spalte: ‘Ausgeliehen von’. |
| **Kommentar** |  | |
| **Fehlerklasse** | M0 | M0 |

Tabelle 15-12: Testfall 7

## Nachfolgearbeiten

### Implementierung

Der nächsten Schritte wären, die Website auf einem Server zu hosten, einen MySQL-Server einrichten und eine Verbindung zu dem bestehenden ADDS-Dienst herstellen. Es ist noch unklar, welche Netzwerkgeräte zur Verfügung stehen werden, da einige Schüler ihre IPA zum bestehenden Netzwerk schreiben und Priorität haben, wenn es darum geht Änderungen zu machen. Deshalb werde ich die Implementation erst nach Abschluss aller IPAs durchführen. Bis dann können aber Ausbauarbeiten an der Applikation gemacht werden.

### Ausbau der Applikation

In der ersten Auflage der Aufgabenstellung (vor Abschluss der Validierung) war geplant, dass die einzelnen Elemente zu einer größeren Einheit zusammengefügt werden konnten. Dies wurde jedoch aus Zeitgründen aus der Aufgabenstellung herausgenommen. Es war am Anfang auch die Rede von Reservierungen statt Ausleihungen. Dies musste auch angepasst werden.

Dazu müssen auch Datenfassungen gemacht werden.

Teil 3: (Formaler Teil 2)

# Versioniserungs Nachweis

Hier werden die Versionen von GitHub angezeigt.

# Abbildungsverzeichnis

[Abbildung 1: Filestruktur der Gespeicherten Daten 17](#_Toc33766772)

[Abbildung 2: Test der Wiederherstellung von Dokumenten 18](#_Toc33766773)

[Abbildung 3: Wiederherstellung durch Github.com 19](#_Toc33766774)

[Abbildung 4: Projektmethode 19](#_Toc33766775)

[Abbildung 5: Organigramm 21](#_Toc33766776)

[Abbildung 6: LOA903 46](file:///C:\Users\Josiah\Documents\IPA2020_JOSIAH_SCHIESS\01_DOKUMENTATION\IPA-Dokumentation_28.02.2020_v11.docx#_Toc33766777)

[Abbildung 7: Konzept: erste Normalform 59](#_Toc33766778)

[Abbildung 8: Konzept: zweite Normalform 59](#_Toc33766779)

[Abbildung 9: Konzept: dritte Normalform 60](#_Toc33766780)

[Abbildung 10: Konzept: Entwicklungsumgebung Architektur 61](#_Toc33766781)

[Abbildung 11: Frontend Package Diagramm 61](#_Toc33766782)

[Abbildung 12: Back-End Package Diagramm 62](#_Toc33766783)

[Abbildung 13: konzeptionelles Login verfahren 62](#_Toc33766784)

[Abbildung 14: Vuex 63](#_Toc33766785)

[Abbildung 15: Konzeptionelles Schnittstellendiagramm 64](#_Toc33766786)

[Abbildung 16: Datenbank Ordnerstruktur 78](file:///C:\Users\Josiah\Documents\IPA2020_JOSIAH_SCHIESS\01_DOKUMENTATION\IPA-Dokumentation_28.02.2020_v11.docx#_Toc33766787)

[Abbildung 17: Datenbankschema 82](#_Toc33766788)

[Abbildung 18: Material Klassen erfassen 84](#_Toc33766789)

[Abbildung 19: Material erfassen 85](#_Toc33766790)

[Abbildung 20: Material erfassen 87](#_Toc33766791)

[Abbildung 21: Material ausleihen 88](#_Toc33766792)

[Abbildung 22: Nachrichten Flussdiagramm 90](file:///C:\Users\Josiah\Documents\IPA2020_JOSIAH_SCHIESS\01_DOKUMENTATION\IPA-Dokumentation_28.02.2020_v11.docx#_Toc33766793)

[Abbildung 23: Fehlermeldung 91](#_Toc33766794)

[Abbildung 24: Erfolgsmeldung 91](#_Toc33766795)

[Abbildung 25: Material Anzeigen Komponent 92](#_Toc33766796)

[Abbildung 26: Frontend Package Diagramm 93](file:///C:\Users\Josiah\Documents\IPA2020_JOSIAH_SCHIESS\01_DOKUMENTATION\IPA-Dokumentation_28.02.2020_v11.docx#_Toc33766797)

[Abbildung 27: Backend Package Diagramm 94](file:///C:\Users\Josiah\Documents\IPA2020_JOSIAH_SCHIESS\01_DOKUMENTATION\IPA-Dokumentation_28.02.2020_v11.docx#_Toc33766798)

[Abbildung 28: API Flowchart 95](#_Toc33766799)

[Abbildung 29: Login verfahren 97](#_Toc33766800)

# Tabellenverzeichnis

[Tabelle 0-1: Teil 1: IPA Dokumentation 1](#_Toc33766707)

[Tabelle 3-1: Standards 14](#_Toc33766708)

[Tabelle 5-1: Laptop 16](#_Toc33766709)

[Tabelle 6-1: Hermes Phasen 20](#_Toc33766710)

[Tabelle 6-2: Meilensteine 21](#_Toc33766711)

[Tabelle 7-1: Projektrollen 22](#_Toc33766712)

[Tabelle 8-1: Risikoanalyse 23](#_Toc33766713)

[Tabelle 8-2: Schadensausmass 24](#_Toc33766714)

[Tabelle 8-3: Eintrittswahrscheinlichkeit 24](#_Toc33766715)

[Tabelle 8-4: Risikograph vor Massnahmen 25](#_Toc33766716)

[Tabelle 8-5: Risikograph nach Massnahmen 26](#_Toc33766717)

[Tabelle 9-0-1: Legende 28](#_Toc33766718)

[Tabelle 10-1: Erster Tag: Mittwoch 12.02.2020 30](#_Toc33766719)

[Tabelle 10-2: Zweiter Tag: Donnerstag 13.02.2020 32](#_Toc33766720)

[Tabelle 10-3: Dritter Tag: Freitag 14.02.2020 34](#_Toc33766721)

[Tabelle 10-4: Vierter Tag: Montag 17.02.2020 (Halbtag) 35](#_Toc33766722)

[Tabelle 10-5: Fünfter Tag: Mittwoch 19.02.2020 36](#_Toc33766723)

[Tabelle 10-6: Sechster Tag: Donnerstag 20.02.2020 38](#_Toc33766724)

[Tabelle 10-7: Siebter Tag: Freitag 21.02.2020 39](#_Toc33766725)

[Tabelle 10-8: Achter Tag: Montag 24.02.2020 (Halbtag) 40](#_Toc33766726)

[Tabelle 10-9: Neunter Tag: Mittwoch 26.02.2020 42](#_Toc33766727)

[Tabelle 10-10: Zehnter Tag: Donnerstag 28.02.2020 43](#_Toc33766728)

[Tabelle 10-11: Elfter Tag: Freitag 28.02.2020 44](#_Toc33766729)

[Tabelle 12-1: IST-situation 47](#_Toc33766730)

[Tabelle 12-2: Systemziele 48](#_Toc33766731)

[Tabelle 12-3: funktionale Anforderungen 50](#_Toc33766732)

[Tabelle 12-4: nicht Funktionale Anforderungen 50](#_Toc33766733)

[Tabelle 12-5: Vorgehensziele 51](#_Toc33766734)

[Tabelle 14-1: Datenbank Namenskonzept 53](#_Toc33766735)

[Tabelle 14-2: Attribute 58](#_Toc33766736)

[Tabelle 14-3: Login 65](#_Toc33766737)

[Tabelle 14-4: Schüler Routen 65](#_Toc33766738)

[Tabelle 14-5: Lehrer Routen 65](#_Toc33766739)

[Tabelle 14-6: Benutzertestdaten 66](#_Toc33766740)

[Tabelle 14-7: Types 66](#_Toc33766741)

[Tabelle 14-8: Locations 66](#_Toc33766742)

[Tabelle 14-9: Manufacturers 67](#_Toc33766743)

[Tabelle 14-10: Fehlerklassen 67](#_Toc33766744)

[Tabelle 14-11: Testarten 67](#_Toc33766745)

[Tabelle 14-12: Testkategorien 68](#_Toc33766746)

[Tabelle 14-13: Testvorlage 68](#_Toc33766747)

[Tabelle 14-14: Testfall 1: Login 69](#_Toc33766748)

[Tabelle 14-15: Testfall 2: Material erfassen 70](#_Toc33766749)

[Tabelle 14-16: Testfall 3: Material Erfassen 71](#_Toc33766750)

[Tabelle 14-17: Testfall 4: Material löschen 72](#_Toc33766751)

[Tabelle 14-18: Testfall 5: Material Bearbeiten 73](#_Toc33766752)

[Tabelle 14-19: Testfall: 6 Ausleihungen machen 74](#_Toc33766753)

[Tabelle 14-20: Testfall 7: Ausleihungen löschen 75](#_Toc33766754)

[Tabelle 15-1: SQL-Skripts 79](#_Toc33766755)

[Tabelle 15-2: Datenbank Beziehungen 83](#_Toc33766756)

[Tabelle 15-3: Öffentliche Schnittstellen 95](#_Toc33766757)

[Tabelle 15-4: Schüler Schnittstellen 96](#_Toc33766758)

[Tabelle 15-5: Lehrkraft Schnittstellen 96](#_Toc33766759)

[Tabelle 15-6: Testfall 1 103](#_Toc33766760)

[Tabelle 15-7: Testfall 2 104](#_Toc33766761)

[Tabelle 15-8: Testfall 3 105](#_Toc33766762)

[Tabelle 15-9: Testfall 4 106](#_Toc33766763)

[Tabelle 15-10: Testfall 5 107](#_Toc33766764)

[Tabelle 15-11: Testfall 6 108](#_Toc33766765)

[Tabelle 15-12: Testfall 7 109](#_Toc33766766)

[Tabelle 19-1: Informationsquellen 116](#_Toc33766767)

[Tabelle 20-1 :Abkürzungen 117](#_Toc33766768)

[Tabelle 21-1: Glossar 118](#_Toc33766769)

[Tabelle 22-1: Phasenfreigabe 119](#_Toc33766770)

[Tabelle 23-1: Anhang 120](#_Toc33766771)

# Informationsquellen

|  |  |
| --- | --- |
| Quellen | Beschreibung |
| HERMES 5 Projektmanagementmethode für alle Projekte Referenzhandbuch - Release 5.1 | Hermes 5 Beschreibung |
| <https://knexjs.org/> | Framework Dokumentation |
| <https://cdn.materialdesignicons.com/4.8.95/> | Material design Icon liste |
| <https://devhints.io/> | Weitere Dokumentationen / cheatsheets |
| <https://vuetifyjs.com/en> | Vuetify library |
| <https://www.lucidchart.com/> | Graphiken, Diagramme |
| <https://www.deepl.com/translator> | Übersetzer |
| <https://languagetool.org/> | Übersetzer |
|  |  |

Tabelle 19-1: Informationsquellen

# Abkürzungsverzeichnis

|  |  |
| --- | --- |
| Abkürzung | Bedeutung |
|  |  |
| IPA | Individuelle Praktische Arbeit |
| JSON | JavaScript Object Notation |
| JWT | JSON Web Token |

Tabelle 20-1 :Abkürzungen

# Glossar

|  |  |
| --- | --- |
| Begriff | Erklärung |
| API | Application programming interface |
| Bcrypt | Framework für Hashes generieren. |
| JS | JavaScript |
| JWT | JSON Web Token |
| Node.js | Node.js ist eine serverseitige Plattform in der Softwareentwicklung zum Betrieb von Netzwerkanwendungen. |
| NPM | Node Package Manager |
| Pug.js | Pug.js ist eine HTML-Templating-Engine, was bedeutet, dass Sie viel einfacheren Pug-Code schreiben können, den Pug in HTML-Code kompiliert, den der Browser verstehen kann. |
| Vuex | Vuetify |

Tabelle 21-1: Glossar

# Phasen Freigabe

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Datum | Unterschrift Lernende | Unterschrift Verantwortliche Fachkraft |
| Initialisierung Freigabe |  |  |  |
| Konzept Freigabe |  |  |  |
| Realisierung Freigabe |  |  |  |

Tabelle 22-1: Phasenfreigabe

# Anhang

|  |  |
| --- | --- |
| Datei | Beschreibung |
| Sitzungsprotokolle\_26.02.2020\_v01.pdf | Beide Sitzungsprotokolle des Expertenbesuches. |
| Benutzeranleitung\_26 |  |
| Datenbank ( /database ) | |
| /seeds/01\_init.js | Hier werden die Testdaten in die Datenbank geschrieben. |
| /migrations/20200219143533\_01\_init.js | Alle Tabellen werden erstellt. |
| /migrations/20200220082855\_02\_relations.js | Alle Beziehungen werden zwischen den Tabellen gemacht. |
| /knexfile.js | Hier wird die Verbindung zu der Datenbank definiert. |
| Backend ( /API ) | |
| /middleware/authorization.js | Token wird verifiziert und in den Request-Header weitergeschickt. |
| /middleware/isTeacher.js | Schaut ob die Rolle ‘teacher’ entspricht. |
| /routes/public.js | Öffentliche Routen. |
| /routes/student.js | Schüler Routen. |
| /routes/teacher.js | Lehrer Routen. |
| /index.js | Haupt-Programm |
| /knex.js | Hier wird die Verbindung zu der Datenbank definiert. |
| /package.json | Hauptfunktion ist es die Abhängigkeiten zu definieren. |
| Frontend ( /frontend ) | |
| /components/Menu.vue | Menu Komponent. |
| /router/index.js | Hier wird der Router definiert. |
| /store/index.js | Global Vuex Store. |
| /views/Dashboard.vue | Die Startseite. |
| /views/Inventory.vue | Die Inventar Seite. |
| /views/Lendings.vue | Die Ausleihliste Seite. |
| /views/Login.vue | Die Login Seite. |
| /views/NewMaterial.vue | Material erfassen Seite. |
| /api.js | Verbindung mit der API wird hier definiert. |
| /App.vue | Das Haupt-Vue Komponent. |
| /main.js | Die Hauptinstanz von Vue. |

Tabelle 23-1: Anhang