Oregon

Eine Team-basierte Web-Applikation zur Stoff-Vertiefung

Bachelor-Thesis

zur Erlangung des akademischen Grades B.Sc.

Patrick Hilgenstock

2203656

Hochschule für Angewandte Wissenschaften Hamburg
Fakultät Design, Medien und Information
Department Medientechnik

Erstprüfer: Prof. Dr. Edmund Weitz

Zweitprüfer: Prof. Dr Torsten Edeler

Hamburg, 20.02.2017

Inhaltsverzeichnis

1 Einleitung

1.1 Zielsetzung

Das Ziel dieser Arbeit ist es eine Webapplikation zu erstellen, welche bei der Gestaltung einer Mathematik-Vorlesung helfen soll. Dabei werden von der Anwendung Aufgaben generiert und an den Nutzer gesendet. Dieser kann diese nun lösen und seine Antwort abschicken. Nach der Validierung bekommt er ein Feedback und bei dem korrekten Lösen eine neue Aufgabe, sowie einige Punkte für sein Team. Der Administrator kann dabei den ganzen Vorgang überwachen und den aktuellen Punktestand einsehen.
Einer der Kernpunkte der Anwendung wird es sein, dass dynamisch Aufgaben generiert werden. Das bedeutet, dass wenn der Nutzer mehrere Aufgaben anfordert er jedes mal eine Aufgabe bekommt welche zwar aus dem selben Themenbereich kommen aber doch jedes mal unterschiedliche Variablen beinhalten und so ein erneutes Berechnen erfordern.

1.2 Bereits existierende Software

Wenn man sich die bereits existierenden Lösungen für dieses Problem anschaut stößt man auf sehr viele Programme die sich mit dem Vertiefen von Stoff befassen. Allerdings setzen diese lediglich darauf den Nutzer Aufgaben zu stellen und die Aufgaben zu validieren. Es gibt keinerlei Möglichkeit die vorhandenen Aufgaben zu erweitern oder die Ergebnisse mit anderen Leuten zu teilen, geschweige denn automatisch eine Übersicht über die Ergebnisse eines ganzen Kurses zu erschaffen. Ebenfalls als Nachteil aufzufassen ist hier die Tatsache, dass es sich bei fast allen Lösungen um Programme handelt welche käuflich zu erwerben sind

Als Variante zur Validierung der Ergebnisse eines ganzen Kurses besteht die Webanwendung “MARS” (Minimal Audience Response System). Dieses ist sehr gut dafür geeignet statische Fragen zu stellen und zu validieren. Allerdings ist es hier nicht möglich dynamisch Aufgaben zu erstellen. Zusätzlich kann nur ein einziges Mal eine Antwort abgegeben werden, es ist also nicht möglich mehrere Fragen des selben Bereiches zu beantworten und so sein Wissen intensiver zu testen.

Ansonsten gibt es bereits viele Webseiten die sich mit dem Stellen von Aufgaben aus verschiedenen Themenbereichen kümmern, zum Beispiel RegexGolf (https://

alf.nu/RegexGolf). Hierbei handelt es sich um eine Anwendung bei der dem Nutzer eine Liste von Wörtern gegeben wird. Die Aufgabe besteht nun darin einen regulären Ausdruck zu finden welcher für einen Teil der Liste matcht und bei der anderen nicht.

1.3 Vorteile der neuen Lösung

Bei der neuen Anwendung soll es sich um ein Projekt handelt welches alle Vorteile existierender Programme vereint, während es die Nachteile auslässt. Das bedeutet, dass die Anwendung viel Wert darauf legt erweiterbar zu sein, ein möglichst großes Spektrum an Aufgaben abdecken zu können und dynamisch Aufgaben zu generieren können.
Gleichzeitig wird es sich bei der resultierenden Webapplikation um Open Source handeln, das heißt der Quellcode ist frei verfügbar und kostenlos, ebenso wie die Webapplikation selbst.

2 Konzeption der Anwendung

2.1 Analyse des Anwendungsfeldes

Bevor man sich nun an die Entwicklung der Anwendung macht bleibt eine Frage offen. Für wen ist sie eigentlich gemacht und wie soll sie eingesetzt werden?
Als grobe Eingrenzung lässt sich sagen, dass diese Anwendung für die Lehre geschaffen wird, um genau zu sein für den Teilbereich der Mathematik. Wichtig ist hierbei zu beachten, dass sich die Applikation nach dem momentanen Konzept nur während der Vorlesung / des Unterrichts benutzen lässt. Dies liegt daran, dass die Idee Teams vorsieht, welche zusammenarbeiten und so Punkte sammeln. Die Arbeit als einzelne Person ist nicht vorgesehen.
Da die Anwendung vorsieht, dass jeder Nutzer einen eigenen Zugang zum Internet hat verlagert sich das Hauptnutzungsfeld in höhere Bildungsstufen, zum Beispiel Universität und Fachhochschule, da dort jeder ein eigenes Smartphone oder sogar einen Laptop benutzt.
Zusammenfassend lässt sich also sagen, dass die Anwendung hauptsächlich im Bereich der Lehre der Mathematik in Universitäten und Fachhochschulen zum Einsatz kommen wird.

2.2 Was ist eine Aufgabe

Als letzten Schritt ist nun noch zu definieren was genau eine Aufgabe eigentlich ist, beziehungsweise wie sie im Sinne dieser Anwendung gesehen wird.
Für diese Arbeit werden wir Aufgaben immer in drei Teilen betrachten:

1. Die Variablen
   Als erstes betrachten wir die Variablen. Sie werden für jede Aufgabe neu generiert. Dadurch, dass diese dynamisch generiert wird jedes mal wenn eine Aufgabe angefragt wird.

2. Die Darstellung
   An zweiter Stelle befassen wir uns mit der Darstellung der Aufgabe. Um die vorher generierten Variablen für den menschlichen Nutzer angemessen darzustellen wird hier eine Zeichenkette generiert, welche die Aufgabe repräsentiert, in die dann die Variablen eingesetzt werden.

3. Die Lösung
   Nachdem die Aufgaben generiert und dargestellt wurden bleibt noch eine Sache übrig. Die Validierung der Aufgabe. Es wird die Lösung des Nutzers übergeben und festgestellt ob diese korrekt gelöst wurde oder nicht.

2.3 Analyse der Anforderungen

2.4 Verwendete Technologien

Für die Anwendung ist nun bereits klar, dass es sich um eine Webanwendung handeln wird. Doch stehen noch einige Fragen zu dieser offen. Wie sieht das Frontend aus? Welcher Technologie wird zum Ausliefern der Backend Daten genutzt? Diese Fragen werden nachfolgend geklärt.

2.4.1 MariaDB

Bereits am Anfang der Entwicklung stand fest, dass viele Daten sowie die Accounts von Administratoren über mehrere Sitzungen hinweg persistiert werden müssen. Ebenfalls stand die Idee im Raum die durchschnittlichen Ergebnisse von Aufgaben zu speichern um so langfristige Beobachtungen durchführen zu können. Daher war recht früh klar, dass auf eine Datenbank kein Verzicht sein kann.

Wenn man sich nun aber überlegt was für eine Datenbank in einem Projekt eingesetzt werden sollte landet man vor einer großen Frage: Dokument-basiert oder doch lieber relational?
Bei relationalen Datenbanken besteht einer der Hauptvorteile liegt in ihrer Struktur. Ihr Aufbau in mehrere Tabellen erlaubt es,dass es sehr leicht ist Duplikate von Daten zu vermeiden (auch bekannt unter dem Begriff “Normalisierung”). Dadurch bleibt die Datenbank kleiner und leichter zu durchsuchen, was gut für die Performanz ist.

Sollte man allerdings Flexibilität benötigen so sind Dokument-basierte Datenbanken die richtige Wahl. Dadurch, dass die Daten einfach in ein Dokument abgelegt werden und nicht an ein Schema gebunden sind an das sich gehalten werden muss können die einzelnen Dokumente ohne Probleme sehr voneinander abweichen. Am ehesten vergleichbar ist dies mit einem normalen JSON-Objekt. Da die Daten hier nicht über mehrere Tabellen verteilt sind lässt sich ein Dokument jederzeit mit einem einzelnen Query aus der Datenbank extrahieren. Daher sind Dokument-basierte Datenbanken immer dann die richtige Wahl wenn viele Datensätze existieren, welche allerdings keine Beziehung zu einander haben (nicht-relationale Datensätze)

Da in dem Projekt die Notwendigkeit war eine Relation zwischen Nutzern und ihren Rolen (Administrator / Nutzer) herzustellen ist die Wahl auf eine relationale

Datenbank gefallen.

Bei der Suche nach einer passenden Datenbank sind zwei in Frage gekommen : MariaDB oder MySQL. Hier die Wahl zu treffen ging recht schnell, da es sich bei MariaDB um nicht viel mehr als einen Fork von MySQL handelt, welcher auf einer performanteren Engine und einer intensiveren Betreeung durch das Entwickler-Team basiert.

2.4.2 Dependency-Injection

„Als Dependency Injection (englisch dependency ‚Abhängigkeit' und injection ‚Injektion'; Abkürzung DI) wird in der objektorientierten Programmierung ein Entwurfsmuster bezeichnet, welches die Abhängigkeiten eines Objekts zur Laufzeit reglementiert: Benötigt ein Objekt beispielsweise bei seiner Initialisierung ein anderes Objekt, ist diese Abhängigkeit an einem zentralen Ort hinterlegt – es wird also nicht vom initialisierten Objekt selbst erzeugt. " [1]

Bei der Dependency-Injection handelt es sich um ein Pattern welches aus der heutigen Welt der Programmierung kaum wegzudenken ist. Anstatt das einzelne Komponenten ihre Abhängigkeiten selber erzeugen geben sie lediglich an, dass sie eine Abhängigkeit benötigen welche einen bestimmten Satz an Aufgaben erfüllen kann (oft durch ein Interface definiert). Daraufhin sucht der Dependency-Injector nach einer Komponente (in den meisten Frameworks als "Service" bezeichnet) die diese Anforderungen erfüllt und fügt sie in den Anfragenden ein (bei diesem Prozess handelt es sich dann um die "Injection")

Beispiel für eine Komponente zur Abfragung von Daten (Datenbankzugriff)

```
public interface ScriptRepository extends CrudRepository<ScriptEntity,
    Integer> {
   public ScriptEntity findByName(String name);
   public List<ScriptEntity> findAll();
}
```

Service

Der Begriff Dienst (auch Service oder Daemon) beschreibt in der Informatik allgemein eine technische, autarke Einheit, die zusammenhängende Funktionalitäten zu einem Themenkomplex bündelt und über eine klar definierte Schnittstelle zur Verfügung stellt.

[1] https://de.wikipedia.org/wiki/Dependency_Injection

Beispiel für einen Service zum Sammeln von Fehlern

```
public interface GlobalErrorService {
   public void appendError(String message);
   public List<String> getErrors();
}
```

Durch das Pattern der Dependency-Injection wird die Trennung zwischen der "Business-Logik" und der technischen Implementation stark getrennt. Dies fördert die Nachvollziehbarkeit von Programmabschnitten. Ebenfalls wird dank der loseren Kopplung die Wiederverwertbarkeit von Code verbessert. Daher wird Dependency-Injection immer dann verwendet wenn auf Modularität Wert gelegt wird

Geschäftslogik
„Geschäftslogik (englisch business logic, auch Anwendungslogik) ist ein abstrakter Begriff in der Softwaretechnik, der eine Abgrenzung der durch die Aufgabenstellung selbst motivierten Logik eines Softwaresystems von der technischen Implementierung zum Ziel hat. " [2]

Ebenfalls zu erwähnen ist die Verbesserung des Verständnisses von Programmcode, insbesondere wenn man nicht den gesamten Kontext der Anwendung kennt. Dies liegt daran, dass jeder Service immer für einen Satz an Aufgaben steht welche logisch zusammenhängen (zum Beispiel das Verwalten der Punktzahl von Nutzern). Sollte ein Service ebenfalls Abhängigkeiten haben werden diese ebenfalls über den Dependency-Injectior eingefügt.

Als letzter Punkt steht die Verbesserung der Testbarkeit. Dadurch, dass die einzelnen Komponenten ihre Abhängigkeiten an einer zentralen Stellle anfragen anstatt sie selber zu erzeugen können an dieser simplere Versionen der Abhängigkeiten hinterlegt werden (sogennante Mock's), wodurch die Funktionen der zu testenden Komponente steigt da Fehler durch die Abhängigkeiten in dieser Umgebung nicht auftreten können.

Mock
„Mock-Objekte (auch Attrappe, von englisch to mock ‚etwas vortäuschen') sind in der Softwareentwicklung Objekte, die als Platzhalter für echte Objekte innerhalb von Modultests verwendet werden. Diese werden umgangssprachlich auch Mocks genannt. " [3]

Das Pattern der Dependency-Injection wird von denn im Nachhinein folgenden Frameworks Spring(-Boot) sowie Angular 2 implementiert und intensiv genutzt.

[2] https://de.wikipedia.org/wiki/Geschäftslogik
[3] https://de.wikipedia.org/wiki/Mock-Objekt

2.4.3 Spring(-Boot)

Auf der Suche nach einem Framework für die Programmierung des REST-Backends stolpert man oft über drei die sich wiederholen. Express-JS, Sinatra und Spring-Boot.

„Express ist ein einfaches und flexibles Node.js-Framework von Webanwendungen, das zahlreiche leistungsfähige Features und Funktionen für Webanwendungen und mobile Anwendungen bereitstellt. “ [4]

Schaut man sich als erstes Express an stößt man auf leichtes Package-Managment (dank NPM), eine Sprache die exzellent für das Bearbeiten und Manipulieren von JSON gemacht ist. Gleichzeitig ist das Problem, dass viele Bibliotheken unterschiedliche Standards benutzen was der Übersicht in größeren Projekten schadet. Zusätzlich kommt das Problem, dass Javascript durch fehlende starke Typisierung eine Sprache ist die bei wachsenden Code-Mengen immer unübersichtlicher wird, während nicht für alle Bibliotheken eine Typescript Implementation vorhanden ist die die Arbeit erleichtern würde. Dazu kommt, dass die Nutzung mehrerer Kerne (und so die Handhabung vieler Anfragen an den Server) nur mittels Prozess-Managern möglich ist, welche ebenfalls wieder Konfiguration und Testaufwand bedeuten. Als letztes ist zu sagen, dass Express-JS eine relativ kleine Entwickler-Basis besitzt, was zu weniger Möglichkeiten für Hilfestellungen bei Problemen führt

„Das Spring Framework (kurz Spring) ist ein quelloffenes Framework für die Java-Plattform. Ziel des Spring Frameworks ist es, die Entwicklung mit Java/Java EE zu vereinfachen und gute Programmierpraktiken zu fördern. Spring bietet mit einem breiten Spektrum an Funktionalität eine ganzheitliche Lösung zur Entwicklung von Anwendungen und deren Geschäftslogiken; dabei steht die Entkopplung der Applikationskomponenten im Vordergrund. “ [5]

Schaut man sich als letztes das Spring-Framework an stößt man auf ein System welches über ein inzwischen gigantisches Ökosystem an Erweiterungen verfügt. Seit März 2004 hat es immer mehr an Popularität und an Features dazu gewonnen. Doch ein wirklich großer Sprung war der Release von Spring-Boot. Dieses übernahm ein Großteil der Konfiguration einer Anwendung und kam bereits mit integriertem Webcontainer mit. Einer der vielen Vorteile die Spring mitbrachte war die Dependency-Injection, welche von Haus aus implementiert war und direkt verwendet werden konnte. Die Absicherung einer Anwendung wird durch Spring-Security ebenfalls erheblich vereinfacht, welches direkt Schutz gegen einige Angriffe bietet, sowie einige Authentifizierungsverfahren erheblich vereinfacht. Eines der Probleme ist allerdings, dass es sich bei Spring (und all seinen Erweiterungen) inzwischen um ein immens großes Projekt handelt, was es sehr schwer machen kann mit Spring anzufangen und die erste

[4] http://expressjs.com/de/)
[5] https://de.wikipedia.org/wiki/Spring_(Framework)

Webanwendung zu schreiben.

Alternative : Express-JS, Sinatra

Für das Backend ist die Wahl auf Spring-Boot gefallen. Bei Spring-Boot handelt es sich um eine Erweiterung des bekannten Java-Frameworks Spring, welches den Großteil der Konfiguration automatisiert, beziehungsweise vereinfacht.

Kurz gesagt hilft das Framework dabei eine Anwendung modular aufzubauen, was insbesondere bei großen Anwendungen sehr hilfreich ist, da diese übersichtlich bleiben. Drei Kernkonzepte die dabei besonders wichtig sind sind die Controller, sowie die bereits erklärten Services und die Dependency-Injection

Controller

Bei einem Controller handelt es sich um eine Klasse welche für die Verwaltung von Anfragen zuständig ist. Jede Anfrage kommt immer bei einem Controller an, von wo die Anfrage an den zuständigen Service weitergeleitet wird. Das Ergebnis von diesem wird dann über den Controller zurück an den Sender der Anfrage gesendet

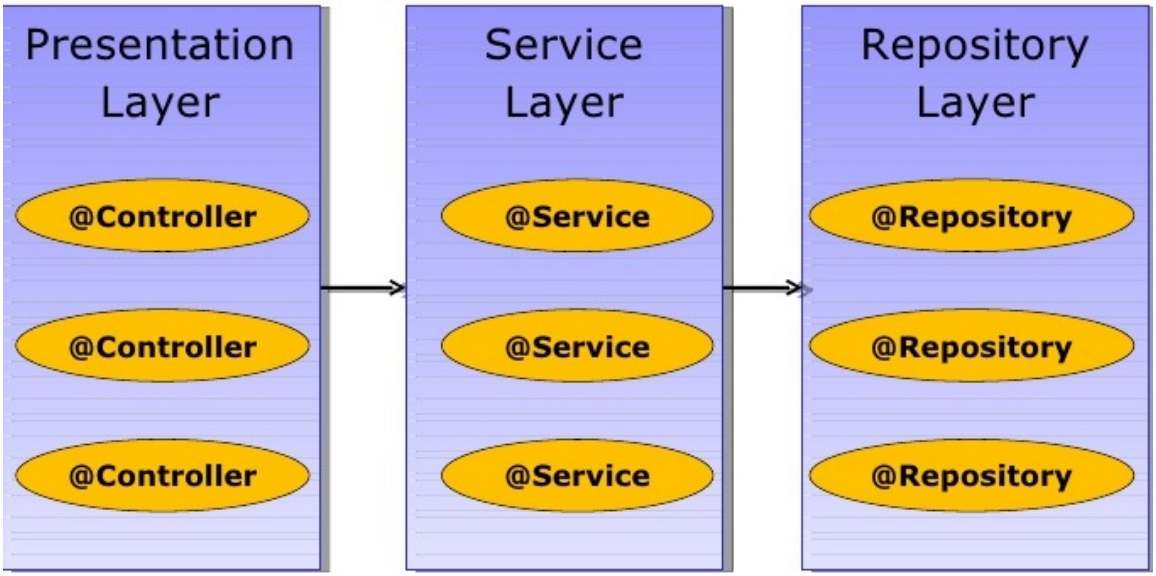

Abbildung 2.1: Konzept einer Spring-Anwendung

Gleichzeit stellt Spring-Boot auch direkt einen Container für die Webapplikation zur Verfügung. Das bedeutet, dass die Programmierung sich komplett mit der REST-API und ihren Funktionalitäten beschäftigen kann und nur sehr wenig Zeit für die Konfiguration des Web-Servers verbraucht wird.

Ein großer Teil der Konfiguration findet außerhalb des Codes statt. In dem Projekt kann eine Datei mit dem Namen “application.properties” hinterlegt werden. Hier können dann mehrere Sachen eingestellt werden, wie zum Beispiel der Port unter

dem der Server laufen soll.
Konfiguration des Server-Ports

```
spring.server.port=8080
```

2.4.4 Spring Data JPA

„Implementing a data access layer of an application has been cumbersome for quite a while. Too much boilerplate code has to be written to execute simple queries as well as perform pagination, and auditing. Spring Data JPA aims to significantly improve the implementation of data access layers by reducing the effort to the amount that's actually needed. As a developer you write your repository interfaces, including custom finder methods, and Spring will provide the implementation automatically. " [6]

Bei Spring Data JPA (im nachfolgenden nur noch als JPA bezeichnet) handelt es sich um eines der Projekte aus dem Spring-Ökosystem welches in wahrscheinlich jedem Spring Projekt heutzutage genutzt wird. Das Hauptziel des Projektes ist es, dass überflüssiger Code der jedes mal geschrieben wird automatisch generiert werden kann und so dem Entwickler viel Arbeit abnimmt. Der Zugriff auf eine Datenbank wird enorm erleichtert, indem die Verbindung automatisch aufgebaut und die Querys automatisch aus Interfaces generiert werden.
Als erstes muss JPA also die Information erlangen wie die Datenbank erreichbar ist und wie die Zugangsdaten lauten. Dies wird in der application.properties angegeben

Konfiguration der Datenbankverbindung

```
spring.datasource.url=jdbc:mysql://localhost:3306/bachelor
spring.datasource.username=root
spring.datasource.password=root
```

Nach dem JPA nun die Verbindung mit der Datenbank aufbauen kann muss definiert werden wie die Datenbank-Entitäten in Java-Objekte umgewandelt werden und in welchen Tabellen sie sich befinden. In JPA wird dies dadurch umgesetzt, dass für jede Entität eine Klasse angelegt wird in der durch Annotationen festgelegt wird wie die Daten in das Objekt gelegt werden.

[6] http://projects.spring.io/spring-data-jpa/

Beispiel für eine Datenbank-Entität

```
@Entity
@Table(name="scripts")
public class ScriptEntity implements Serializable{

   @Id
   @GeneratedValue(strategy= GenerationType.IDENTITY)
   private Integer id;

   @Column(nullable=false, name="name")
   private String name;

   @Column(nullable=false, name="variablescript")
   private String variableScript;

   @Column(nullable=false, name="solutionscript")
   private String solutionScript;

   @Column(nullable=false, name="mathjaxscript")
   private String mathjaxScript;

   /*Getter/Setter/Konstruktor*/
}
```

Zu guter Letzt muss ein Interface erzeugt werden, welches definiert was für Querys zur Verfügung stehen. Um das zu realisieren wird ein Interface erzeugt, welches die Querys erzeugt indem es bei der Benennung der Methoden bestimmten Konventionen folgt. Wichtig ist hierbei, dass es das bereits gegebene Interface CrudRepository erweitern muss. Dadurch wird es von JPA erkannt und einige Funktionalität direkt gegeben (wie zum Beispiel findAll, save und delete). Um nun eigene Querys hinzufügen gibt es zwei Möglichkeiten. Entweder über eine Annotation oder über die Benennung der Funktion

Beispiel für eine eigene Querys

```
public interface ScriptRepository extends CrudRepository<ScriptEntity,
    Integer> {

   public ScriptEntity findByName(String name);

   @Query("SELECT p FROM ScriptEntity s WHERE s.name = :name")
   public ScriptEntity customQuery(@Param("name")String name);
}
```

2.4.5 Bootstrap

Gegenstück : Material

„Bootstrap ist ein anpassungsfähiges und zuerst für mobile Geräte entwickeltes Framework. Egal, wie gut du dich auskennst, kannst du damit einfacher und schneller kleine wie große Projekte entwickeln, die auf Geräten in allen erdenklichen Formen funktionieren. “ [7]

Aus der heutigen Welt der Webentwicklung ist Bootstrap kaum wegzudenken. Durch die leichte Entwicklung für alle vorhandenen Formate und Endgeräte hat es sich als äußerst robust herausgestellt wenn es um die schnelle Entwicklung von Webseiten geht.
Gleichzeitig bietet es ein großes Repertoire an CSS-Klassen für alle Elemente der HTML-Entwicklung was es sogar Personen mit wenig Erfahrung im Design-Bereich ermöglicht spektakuläre Ergebnisse mit wenig Arbeit zu erzielen.

2.4.6 Typescript / Angular 2

Bevor man sich mit der Technologie beschäftigt die benutzt wurde um das Frontend zu bauen muss man sich vorher mit einer anderen Errungenschaft der Webentwicklung auseinander setzen. Typescript.

„TypeScript is a free and open-source programming language developed and maintained by Microsoft. It is a strict superset of JavaScript, and adds optional static typing and class-based object-oriented programming to the language.
TypeScript is designed for development of large applications and transcompiles to JavaScript. As TypeScript is a superset of JavaScript, any existing JavaScript programs are also valid TypeScript programs. “ [8]

Besonders für Entwickler welche aus der Welt der stark-typisierten Programmierwelt kamen war TypeScript ein großer Fortschritt. Durch die Möglichkeit Variablen stark zu typisieren entstand die Möglichkeit Schnittstellen klarer zu definieren. Gleichzeitig wurden die bestehenden Möglichkeiten Entwicklungsumgebungen mit Auto-Vervollständigung auszustatten verbessert.
Die Vorteile die sich daraus für die Entwicklung sind recht eindeutig: Eine Beschleunigung der Entwicklung, während die Testbarkeit vereinfacht wird und Fehler besser vermieden werden können. Kurz gesagt: Bessere Software in weniger Zeit.

Was für die Anwendung relativ früh klar war war, dass es sich um eine Single Page Application (kurz SPA) handeln sollte. Doch was genau ist das und wieso sollte es verwendet werden?

[7] http://holdirbootstrap.de//
[8] https://en.wikipedia.org/wiki/TypeScript

„Als Single-Page-Webanwendung (englisch single-page application, kurz SPA) oder Einzelseiten-Webanwendung wird eine Webanwendung bezeichnet, die aus einem einzigen HTML-Dokument besteht und deren Inhalte dynamisch nachgeladen werden. Diese Art von Web-Architektur steht im Gegensatz zu klassischen Webanwendungen, die aus mehreren, untereinander verlinkten HTML-Dokumenten bestehen. Hierdurch wird allerdings die Grundlage geschaffen, eine Webanwendung in Form einer Rich-Client- bzw. Fat-Client-Verteilung zu entwickeln. Eine verstärkte clientseitige Ausführung der Webanwendung ermöglicht eine Reduzierung der Serverlast sowie die Umsetzung von selbstständigen Webclients, die beispielsweise eine Offline-Unterstützung anbieten. “ [9]

Warum also eine Single-Page-Application? Der Hauptvorteil liegt darin, dass bei dem Initialisieren der Seite fast alle benötigten Informationen geladen werden, somit fallen die Ladezeiten nach der ersten Initialisierung sehr gering aus. Das sorgt insbesondere bei Nutzern von mobilen Endgeräten für eine bessere User-Experience, da bei diesen (bedingt durch schwächere Prozessoren) die Ladezeiten besonders ins Gewicht fallen. Gleichzeitig reduzieren sich die Anforderungen an den Webserver, da der Content nur ein einziges mal ausgeliefert werden muss. Lediglich durch die Abfragung einiger Daten zur Laufzeit (wie zum Beispiel das Erhalten einer neuen Aufgabe) wird der Server nach dem Ausliefern der Webseite kontaktiert.

Nachdem nun evaluiert wurde, dass eine SPA ein Gewinn ist also die Frage nach dem Framework / der Library um das umzusetzen. Im Raum stehen zwei Varianten. Zum einen ReactJS (welches seit längerem existiert und weit bekannt ist) oder das gerade in der Beta angekommene Angular 2.
Direkt zu sagen ist, dass Angular 2 von Haus aus mit Typescript gebaut ist. Das bedeutet bester Typescript bei der Entwicklung von Angular 2, was bei ReactJS nur teilweise ist und viele Varianten wie man am besten mit ReactJS und Typescript arbeitet sich widersprechen da keine klare Konvention herrscht.
Zusätzlich implementiert Angular 2 eine bessere Trennung der verschiedenen Komponenten. Während in ReactJS HTML und Javascript Code in der selben Klasse stehen werden in Angular 2 klare Mechanismen aufgezeigt wie diese sauber zu trennen und in verschiedenen Dateien abzulegen sind. Was gleichzeitig auch einer der Nachteile von Angular 2 ist. Dadurch, dass HTML und Script sauber getrennt werden ist es schwerer die Script-Variablen im HTML zu referenzieren und alle Namen im Kopf zu behalten ohne die zweite Datei irgendwo geöffnet zu haben.
Bei Angular 2, sowie ReactJS handelt es sich standardmäßig um Technologien bei denen “Client-Side Rendering” betrieben wird. Das beudetet, dass vom Server alle Dateien ausgeliefert werden die benötigt werden. Der Client nimmt diese entgegen und rendert die einzelnen HTML-Komponenten und fügt diese korrekt zusammen. Das bedeutet, dass die Last des Rendering auf den Client übertragen wird. Das be-

[9] https://de.wikipedia.org/wiki/Single-Page-Webanwendung

deutet weniger Aufwand für den Server, gleichzeitig aber eine bessere Performanz für den Server (da dieser nicht selber rendern muss). Was bei Angular2 aber auch unterstützt wird ist die Möglichkeit auf der Seite des Server zu rendern. Das bedeutet wenn man die Ladezeiten auf dem Client reduzieren will und einen performanten Server zur Hand hat ist dies die korrekte Lösung. Möglich ist dies durch eine kürzlich herausgekommene Erweiterung für Angular 2 https://universal.angular.io/.

Am Ende fiel die Wahl auf Angular 2, auf Grund der oben genannten Vorteile zusätzlich dazu, dass hinter Angular 2 ein Team von Entwicklern steht welche Erfahrungen aus einem vorherigen Projekt (AngularJS) gezogen haben um daraus von neu an ein komplett neues Framework zu erstellen welches aus den Problemen der alten Version gelernt hat und diese ausgemerzt hat.

2.4.7 SocketIO

3 Ein Rundgang durch das Frontend

3.1 Die Ansicht des Nutzers

3.1.1 Die Auswahl des Teams

3.1.2 Erhalten und bearbeiten der Aufgaben

3.2 Die Sicht des Administrators

3.2.1 Bearbeitung der Aufgaben-Generatoren und ihren Hilfsmitteln

3.2.2 Starten eines neuen Aufgaben-Generators

3.2.3 Übersicht über die laufende und letzte Aufgabe

3.2.4 Der Fehler-Log

4 Ein Blick unter die Haube - das Backend

4.1 Die Sicherung der REST-API

4.2 Dokumentation der REST-Schnittstellen

4.3 Die Generierung und Validierung der Aufgaben

4.4 Die Sicherung der Aufgaben-Generierung

5 Fazit

5.1 Das Ergebnis

5.2 Wie kann die Anwendung verbessert / erweitert werden

Abbildungsverzeichnis

Literaturverzeichnis

Spring-Dokumentation https://spring.io/docs, letzter Zugriff: 17.12.2016

Spring-Boot-Dokumentation http://docs.spring.io/spring-boot/docs/current-SNAPSHOT/reference/htmlsingle/, letzter Zugriff: 17.12.2016
https://blog.php-dev.info/2014/04/mariadb-vs-mysql/
https://www.quora.com/What-are-the-advantages-and-disadvantages-of-using-Sinatra-vs-Express-for-a-web-service-API
https://medium.freecodecamp.com/angular-2-versus-react-there-will-be-blood-66595faafd51.ufvfcrtqo

Blu-ray Disc Association: *White paper Blu-ray Disc Format 2.B Audio Visual Application, Format Specifications for BD-ROM*, http://www.blu-raydisc.com/Assets/downloadablefile/2b_bdrom_audiovisualapplication_0305-12955-15269.pdf, 2005, letzter Zugriff: 1. 10. 2012

Dooley, Wesley L. & Streicher, Ronald D.: „M–S Stereo: A Powerful Technique for Working in Stereo“, *Journ. Audio Engineering Society* vol. 30 (10), 1982

Kuttruff, Heinrich: *Room Acoustics*, 3. Aufl., Elsevier 1991

Spehr, Georg (Hrsg.): *Funktionale Klänge*, transcript 2009

Sowodniok, Ulrike: „Funktionaler Stimmklang – Ein Prozess mit Nachhalligkeit“, in: Spehr, Georg (Hrsg.): *Funktionale Klänge*, transcript 2009

Stephenson, Uwe: „Comparison of the Mirror Image Source Method and the Sound Particle Simulation Method“, *Applied Acoustics* vol. 29, 1990

Ich versichere, die vorliegende Arbeit selbstständig ohne fremde Hilfe verfasst und keine anderen Quellen und Hilfsmittel als die angegebenen benutzt zu haben. Die aus anderen Werken wörtlich entnommenen Stellen oder dem Sinn nach entlehnten Passagen sind durch Quellenangaben eindeutig kenntlich gemacht.

Ort, Datum Patrick Hilgenstock