

## Diplomarbeit

Höhere Technische Bundeslehranstalt Leonding  
Informationstechnologie mit Schwerpunkt Medientechnik

## Tauschbörse

Eingereicht von: **Antonia Kriegner, 5AHITM**  
**Melanie Pittersberger, 5AHITM**  
Datum: **April 5, 2019**  
Betreuer: **Prof. Mag. Robert Reder**  
Projektpartner: **Gemeinde Natternbach**

## **Declaration of Academic Honesty**

Hereby, I declare that I have composed the presented paper independently on my own and without any other resources than the ones indicated. All thoughts taken directly or indirectly from external sources are properly denoted as such.

This paper has neither been previously submitted to another authority nor has it been published yet.

Leonding, April 5, 2019

Antonia Kriegner, Melanie Pittersberger

## **Eidesstattliche Erklärung**

Hiermit erkläre ich an Eides statt, dass ich die vorgelegte Diplomarbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Gedanken, die aus fremden Quellen direkt oder indirekt übernommen wurden, sind als solche gekennzeichnet.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Leonding, am 5. April 2019

Antonia Kriegner, Melanie Pittersberger

## **Gender-Klausel**

Für die bessere Lesbarkeit wird in unserer Arbeit auf geschlechtsspezifische Formulierungen verzichtet. Es wird ausdrücklich darauf hingewiesen, dass im Text personenbezogene Formulierungen immer gleichermaßen auf alle Geschlechter bezogen sind.

## **Zusammenfassung**

Die Pfarrgemeinde Natterbach führt eine Tauschbörse unter dem Namen "Talente-tauschkreis Hausruck Nord". Die Tauschbörse hat den Zweck, dass Produkte und Dienstleistungen getauscht werden können. Bis zum jetzigen Zeitpunkt wurden die ganzen Tauschprozesse handschriftlich mitgeschrieben. Nun haben sich die zuständigen Personen dazu entschieden, die Tauschbörse zu digitalisieren, um so die Durchführung der Tauschprozesse zu vereinfachen und das Programm für noch mehr Leute zugänglich zu machen. Außerdem kann die Tauschbörse so ihre Benutzer laufend über bevorstehende Termine informieren. Der Benutzer muss sich zuerst registrieren, um an der Tauschbörse teilnehmen zu können. Wenn der Benutzer Inserate sehen möchte, beziehungsweise auf sein Konto zugreifen möchte, muss er sich einloggen. Der Benutzer kann ein Inserat für sein Produkt oder für seine Dienstleistung erstellen und so dafür werben. Andere Benutzer können auf das Inserat eingehen und das Produkt kaufen oder die Dienstleistung in Anspruch nehmen. Das System basiert auf der virtuellen Währung "Talente". Talente sind sozusagen Punkte, die der Benutzer für sein jeweils zu erstellendes Inserat individuell vergeben kann. Ein Benutzer hat ein Konto, auf dem zu Beginn ein Startguthaben vorhanden ist. Wenn er etwas verkauft, kommen diese Talente auf sein Konto. Wenn er selbst etwas kauft, werden diese Talente von seinem Konto abgezogen.

Für den Server wurde Java als Programmiersprache verwendet und für den Client das Framework Angular. Als OR-Mapper wurde JPA verwendet und als Datenbank wurde MariaDB gewählt. Außerdem wurde die Scrum Methode des Projektmanagements angewendet.

Das fertige Produkt soll dem Benutzer erlauben Inserate erstellen zu können, Produkte oder Dienstleistungen von anderen Benutzern in Anspruch zu nehmen und den Benutzern, für deren Inserate sich die Person interessiert, zu kontaktieren. Außerdem werden die Talente automatisch berechnet und es kann eine Rechnung ausgestellt werden, die der Benutzer erhält.

## **Abstract**

The parish of Natternbach manages a barter under the name "Talentetauschkreis Hausruck Nord". The purpose of the barter is to exchange products and services. Up to the present time the whole exchange processes were handwritten. Now the responsible persons have decided to digitize the exchanges in order to simplify the execution of the exchange processes and to make the program accessible to even more people. In addition, the "Talentetauschkreis Hausruck Nord" can continuously inform its users about upcoming appointments. The user must first register in order to participate in the program. If the user wants to see the things, other users have to offer, or access his account, he has to log in. The user can create an ad for his or her product or service and advertise for it. Other users may respond to the listing and purchase the product or use the service. The system is based on the virtual currency "Talente". "Talente" are points that the user can individually award for each advertisement to be created. A user has an account with a starting credit. If he sells something, these "Talente" come into his own account. If he buys something himself, those points will be deducted from his account.

Java was used as the programming language for the server and the Angular framework for the client. As an OR mapper JPA was used and as a database MariaDB was chosen. In addition, the Scrum method of project management was applied.

The finished product should allow the user to create advertisements for their products or services, to use products or services of other users and to contact the users whose ads the person is interested in. In addition, the talents are calculated automatically and an invoice can be issued, which the user receives.

## **Danksagung**

Wir bedanken uns bei unserem Diplomarbeitsbetreuer Herrn Prof. Mag. Robert Reder, der uns immer mit Rat und Tat zur Seite stand und stets bemüht war, uns bestmöglichst zu unterstützen.

Außerdem bedanken wir uns bei der Pfarrgemeinde Natternbach für die tolle Zusammenarbeit und für die ständige Unterstützung.

Zum Schluss möchten wir uns noch bei unseren Familien und Freunden bedanken, die uns während der Diplomarbeit stets zur Seite standen.

# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>5</b>
1.1 Abkürzungen . . . . .	5
1.2 Ausgangssituation (MP) . . . . .	5
1.3 Zieldefinition (AK) . . . . .	5
1.4 Auftraggeber (AK) . . . . .	6
1.5 Autorinnen der Diplomarbeit (MP) . . . . .	7
1.6 Aufgabenverteilung (AK) . . . . .	9
1.6.1 Aufgaben von Pittersberger Melanie . . . . .	9
1.6.2 Aufgaben von Kriegner Antonia . . . . .	9
1.6.3 Gemeinsame Aufgaben (MP AK) . . . . .	10
<b>2 Technologien</b>	<b>12</b>
2.1 Sprachen (MP) . . . . .	12
2.1.1 Hypertext Markup Language (HTML) . . . . .	12
2.1.2 Cascading Style Sheets (CSS) . . . . .	13
2.1.3 TypeScript (TS) . . . . .	13
2.1.4 Wie unterscheidet sich TypeScript von JavaScript? . . . . .	14
2.1.5 Angular . . . . .	14
2.1.6 Java (AK) . . . . .	18
2.2 Werkzeuge . . . . .	21
2.2.1 Visual Studio Code (MP) . . . . .	21
2.2.2 NetBeans (AK) . . . . .	22
2.2.3 Maven Project (AK) . . . . .	25
2.2.4 Adobe Illustrator (AK) . . . . .	27
2.3 Projektmanagement . . . . .	28
2.3.1 Lastenheft (MP) . . . . .	28
2.3.2 Pflichtenheft (MP) . . . . .	28
2.3.3 Meilensteinliste (MP) . . . . .	30
2.3.4 Gitlab (AK) . . . . .	33
2.3.5 Scrum und Tool ScrumDo (AK) . . . . .	35

<b>3 Realisierung</b>	<b>37</b>
3.1 Überblick (MP) . . . . .	37
3.2 Grund der Aufteilung (MP) . . . . .	39
3.3 Informationsseite - Wordpress . . . . .	39
3.3.1 Elementor Plugin (AK) . . . . .	39
3.4 Applikation . . . . .	40
3.4.1 Architektur (AK) . . . . .	40
<b>4 Server</b>	<b>43</b>
4.1 Datenbank (AK) . . . . .	43
4.1.1 MariaDB . . . . .	43
4.1.2 Vorteile . . . . .	43
4.1.3 Nachteile . . . . .	44
4.1.4 Installation . . . . .	45
4.1.5 Alternativen . . . . .	47
4.2 Datenbankstruktur . . . . .	48
4.2.1 Entitäten (MP) . . . . .	48
4.2.2 Klassendiagramm (CLD) - (AK) . . . . .	50
4.2.3 Beziehungen (AK) . . . . .	50
4.3 Aufbau der Projektstruktur (AK) . . . . .	51
4.4 Lazy Loading (AK) . . . . .	53
4.5 Security (MP) . . . . .	53
4.5.1 Worauf muss geachtet werden? . . . . .	53
4.5.2 Message-Digest Algorithm (MD5)- Allgemein . . . . .	54
4.5.3 Vorteile . . . . .	54
4.5.4 Nachteile . . . . .	55
4.5.5 Wie haben wir MD5 in unserem Projekt eingesetzt? . . . . .	55
4.5.6 MessageDigest . . . . .	56
4.5.7 Andere Hash-Funktionen . . . . .	56
4.5.8 E-Mail senden . . . . .	58
<b>5 Client</b>	<b>61</b>
5.1 Navigationsstruktur (MP) . . . . .	61
5.2 Cards/Panel (AK) . . . . .	63
5.3 Slideshow - Plugin (AK) . . . . .	64
5.3.1 Ngx-gallery . . . . .	64
5.3.2 Alternativen zur Ngx – gallery . . . . .	65
5.4 Mittel für die responsive Darstellung (AK) . . . . .	67
5.4.1 Bootstrap . . . . .	67
5.4.2 Media queries . . . . .	67
5.5 Filter (AK) . . . . .	69
5.6 Nachrichten – Nachrichtenformular (AK) . . . . .	70
5.7 Abrechnung (AK) . . . . .	72
5.8 Drucken (AK) . . . . .	74

<b>6 Wissenswertes</b>	<b>76</b>
6.1 Registrierung (MP) . . . . .	76
6.1.1 Registrierungs-Formular . . . . .	76
6.1.2 JSON Objekt eines Benutzers . . . . .	78
6.1.3 Benutzer löschen . . . . .	78
6.1.4 Registrieren Methode . . . . .	79
6.2 Login (MP) . . . . .	80
6.2.1 Login Methode am Client . . . . .	80
6.2.2 Login Formular . . . . .	81
6.3 Inserate erstellen (MP) . . . . .	82
6.3.1 Inserate erstellen Formular . . . . .	82
6.3.2 Image Upload . . . . .	85
6.3.3 Inserat erstellen Methode . . . . .	85
<b>7 Schlussfolgerung</b>	<b>86</b>
<b>A Anhang</b>	<b>98</b>
A.0.1 Plichtenheft . . . . .	98

# **Kapitel 1**

## **Einleitung**

### **1.1 Abkürzungen**

Die Abkürzungen AK und MP stehen für Antonia Kriegner und Melanie Pittersberger. Diese Kürzel werden immer bei einer Überschrift angegeben und markieren somit, welche Abschnitte von wem geschrieben wurden.

### **1.2 Ausgangssituation (MP)**

Die Gemeinde Natternbach führt schon seit längerem eine Tauschbörse unter dem Namen „Talentetauschkreis Hausruck Nord“. Die Mitglieder dieser Tauschbörse treffen sich in regelmäßigen Abständen zu sogenannten Tauschtreffen, bei welchen die momentanen Angebote angepriesen werden und wo auch Gegenstände und Dienstleistungen getauscht werden können. Allerdings wurden die Tauschprozesse bis jetzt handschriftlich abgewickelt. Das heißt, wenn etwas getauscht wurde, wurde dies auf einem Blatt Papier festgehalten. Unsere Auftraggeber haben sich nun dazu entschieden, dass sie das Ganze auf eine Webseite auslagern möchten, sodass die Tauschbörse digital funktioniert und für jedermann zugänglich ist. Die digitale Währung der Tauschbörse sind "Talente". Die schon bestehenden Kunden werden natürlich weitergeführt, sie verlieren dadurch nicht ihre bereits vorhandenen Talente. Es gibt ein Startkapital von 10 Talenten. Wenn jemand eine Dienstleistung in Anspruch nimmt oder ein Produkt kauft, werden ihm diese Talente von seinem Konto abgezogen. Im Umkehrschluss werden der Person, die diese Dienstleistung beziehungsweise das Produkt anbietet, die Talente auf das Konto gebucht.

### **1.3 Zieldefinition (AK)**

Mit dieser Anwendung soll erreicht werden, dass die Tauschbörse mehr Mitglieder bekommt und es mehr Tauschprozesse gibt. Es soll eine Erleichterung für die Personen sein, die zurzeit die Buchhaltung führen, da die Talente nun online verbucht werden können. Ein weiterer Vorteil ist, dass neue Angebote schneller verbreitet werden und die Angebo-

te dadurch stets aktuell sind. Zudem wird die Kontaktaufnahme mit jemandem, der an einem Tausch interessiert ist, einfacher. Es ist nicht mehr notwendig, die Telefonnummer zu kennen, um mit der jeweiligen Person kommunizieren zu können. Zusätzlich soll es eine Anregung für die jüngere Generation sein, auch am Talentetausch teilzunehmen.

## 1.4 Auftraggeber (AK)

Die Diplomarbeit ist in Kooperation mit der Pfarrgemeinde Natternbach entstanden, dabei hat uns ein Team aus sechs Leuten tatkräftig unterstützt. Unsere Ansprechperson bei allen Fragen waren Hildegard Wiesinger, Rita Haim-Harasek, Maria Scheuringer, Monika Buchinger, Christine Schöfberger und Ernestine Dornetshuber. Zur Kommunikation mit den Auftraggebern bildeten wir eine WhatsApp Gruppe und über E-Mails tauschten wir Dateien aus. Zusätzlich telefonierten wir manchmal mit einer Person, welche anschließend die Ergebnisse des Gesprächs an die anderen Personen weiterleitete.

Zu Beginn der Arbeit besprachen wir viel mit Hildegard Wiesinger. Mit ihr vereinbarten wir, dass wir das Projekt als Diplomarbeit starten möchten. Zudem hatten wir auch mit ihr das erste Treffen, bei dem uns der ganze Funktionsaufwand der Diplomarbeit erklärt wurde.

Rita Haim-Harasek war während der Projektentwicklung die Hauptansprechperson, da Hildegard Wiesinger aus Zeitgründen nicht immer leicht zu erreichen war. Also übernahm Rita Haim-Harasek das Ruder und unterstützte uns beim Organisieren von neuen Treffen. Weiters konnte sie uns hilfreiche Verbesserungsvorschläge für die Anwendung geben.

Maria Scheuringer war gemeinsam mit Rita Haim-Harasek eine der wichtigsten Ansprechpersonen bezüglich der Texte auf der Startseite. Die beiden überarbeiteten die Texte oft mehrmals und ließen uns so korrekte und inhaltlich perfekt ausgearbeitete Ergebnisse zukommen.

Monika Buchinger unterstützte uns mit zahlreichen von ihr per Hand gezeichneten Grafiken für die Informationswebseite. Sie brachte mit ihren Bildern eine neue und vor allem bunte Seite in die Webseite.

Christine Schöfberger und Ernestine Dornetshuber waren bei den jeweiligen Treffen nicht anwesend, aber dafür bei Gemeinschaftsfragen immer involviert. Wenn eine Frage auftauchte und diese in der WhatsApp-Gruppe behandelt wurde, engagierten sie sich. Außerdem wirkten sie beim Erstellen des neuen Logos mit guten Ideen mit.

Für die nächsten Schritte außerhalb der Diplomarbeit wird uns Michael Barth zur Verfügung stehen, der uns einen Speicherplatz am Server organisiert, diesen für uns aufsetzt und die Anwendung darauf platzieren wird.

## 1.5 Autorinnen der Diplomarbeit (MP)

**Antonia Kriegner:**

**Persönliche Daten:**

**Name:** Antonia Kriegner

**Geburtsdatum:** 30.05.2000

**Adresse:** Polstergrub 3, 4724 Neukirchen am Walde

**E-Mail:** antonia.kriegner@gmail.com

**Schulausbildung:**

**seit 09/2014** HTBLA Leonding, Medientechnik

**09/2010 – 07/2014** Hauptschule Neukirchen am Walde

**09/2006 – 07/2010** Volksschule Neukirchen am Walde

**Berufserfahrung:**

- Ferialpraktikum bei Maschinenring
- Ferialpraktikum bei Times Up



Abbildung 1.1: Schulfotograf - Antonia Kriegner

**Melanie Pittersberger:**

**Persönliche Daten:**

**Name:** Melanie Pittersberger

**Geburtsdatum:** 17.07.2000

**Adresse:** Voralpenstraße 27, 3351 Weistrach

**E-Mail:** m.pittersberger@gmail.com

**Schulausbildung:**

**seit 09/2014** HTBLA Leonding, Medientechnik

**09/2010 – 07/2014** Hauptschule St. Peter/Au

**09/2006 – 07/2010** Volksschule Weistrach

**Berufserfahrung:**

- Ferialpraktikum bei EDBS GmbH
- Ferialpraktikum im Landesklinikum Mauer



Abbildung 1.2: Schulfotograf - Melanie Pittersberger

## **1.6 Aufgabenverteilung (AK)**

### **1.6.1 Aufgaben von Pittersberger Melanie**

#### **Benutzerverwaltung**

Die Benutzerverwaltung gliedert sich in die Registrierung und die Anmeldung. Bei der Registrierung müssen alle Daten des neuen Benutzers angegeben werden und anschließend wird eine E-Mail geschickt, die einen Bestätigungslink enthält. Durch Drücken auf den Link wird die Registrierung vervollständigt und es wird auf die Anmeldungsseite weitergeleitet. Ein wichtiges Thema bei der Registrierung ist, dass das Passwort des Benutzers verschlüsselt wird. Wenn man eingeloggt ist, gibt es einen Reiter, wo die Daten des Nutzers nochmals geändert werden können.

#### **Verkaufsangebotmanagement**

Für das Erstellen von Einträgen gibt es ein Eingabeformular, bei dem der Benutzer die Daten einer Dienstleistung oder eines Produktes eingeben kann. Wichtig ist es, dass auch Bilder hochgeladen werden können, diese dürfen aber nur eine bestimmte Größe haben. Die Eingabe soll durch viele Überprüfungen möglichst benutzerfreundlich gestaltet werden und Eingabefehler sollen ausgeschlossen sein.

### **1.6.2 Aufgaben von Kriegner Antonia**

#### **Angebotsansicht für den Verkaufsprozess**

Es werden alle vorhandenen und sichtbar gesetzten Einträge angezeigt. Die Inserate sind durch eine Rasteransicht von Boxen dargestellt, wozu es einige Suchoptionen gibt. Zuerst soll gefiltert werden, ob es sich um ein Produkt oder eine Dienstleistung handelt. Zudem sollte die Suche auf bestimmte Kategorien eingegrenzt werden können. Eine weitere Funktion, die wichtig ist, ist die Suchleiste, wo nach bestimmten Begriffen gesucht werden kann. Beim Klick auf einen Eintrag wird auf die Detailansicht des Eintrages weitergeleitet, wo eine Anfrage an den Ersteller des Inserates gesendet werden kann. Zudem ist eine Slideshow ersichtlich und die genauen Daten des Inserates. Als Benutzer können die eigenen Einträge gesichtet werden und durch einen Klick anschließend auch wieder bearbeitet werden.

#### **Nachrichtenportal des Tauschabwicklungsprozesses**

Dies inkludiert das Schreiben der Nachrichten sowie das Anzeigen der Nachrichten. Es gibt einen Reiter, wo die Anfragen und Antworten eines Benutzers zu sehen sind. Beim Schreiben einer Nachricht werden die Daten, bis auf den Text, immer automatisch ausgefüllt. Bei der Nachrichtenanzeige ist zu sehen, von wem die Nachricht ist und welches Inserat es betrifft, zudem ist ersichtlich ob es eine Anfrage oder eine Antwort ist. Ähnlich wie bei bekannten Messengern kann durch Klicken auf einen Button gleich eine Antwort verfasst werden, so wird das Kommunizieren stark vereinfacht.

## Talentebuchungsprozess

Das Abbuchen der Talente gliedert sich in die Talenteverwaltung und die Belegsansicht. Wenn ein Tausch erfolgt ist, kann anschließend ein Beleg erstellt werden. Durch Bestätigen des Beleges vom Gegenüber werden die Talente verrechnet, das heißt, es werden die Talente des Anfragenden abgebucht und auf das Konto des Anbietenden aufgebucht. Zum Schluss können beide Benutzer den Tausch in einer Tabelle ansehen und, falls gewünscht, einen Beleg ausdrucken.

### 1.6.3 Gemeinsame Aufgaben (MP AK)

#### Grunddesign der Datenbankstruktur

Zu Beginn der Diplomarbeit überlegten wir uns, wie wir das Datenbankdesign aufbauen wollen. Zu diesem Zweck arbeiteten wir handschriftliche Skizzen aus. Nachdem wir die Grundstruktur erarbeitet hatten, stellten wir das vorläufige Design mithilfe von dem Programm "Visio" dar.

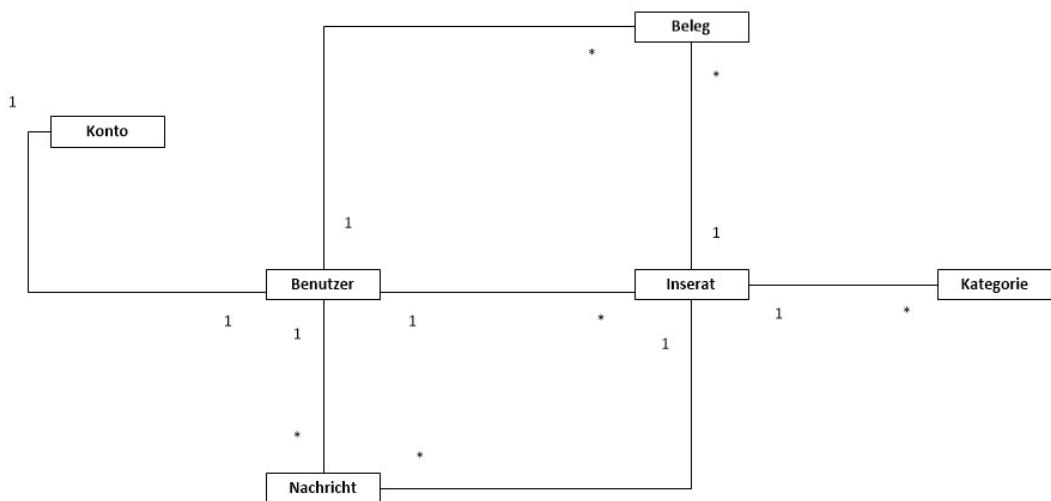


Abbildung 1.3: Grundstruktur der Datenbank - Screenshot Antonia Kriegner

Am Anfang waren die Entitäten Konto, Benutzer, Nachricht, Inserat, Beleg und Kategorie geplant.

Im Laufe der Diplomarbeit kam noch die Entität Registrierung hinzu, um das Registrieren von Benutzern zu vereinfachen.

## **Wahl der Technologien**

Ein weiterer wesentlicher Punkt, welchen wir uns zu Beginn unserer Diplomarbeit überlegen mussten, war die Wahl der Technologien.

Für uns war von Anfang an klar, dass wir die Technologien, welche wir bereits im Unterricht kennengelernt hatten, verwenden möchten. Wir sahen darin den Vorteil, dass wir uns schon gut mit den jeweiligen Technologien auskannten und außerdem eine zusätzliche Übung für die Matura hatten.

Für den Server verwendeten wir die Programmiersprache Java, für den Client das Framework Angular und als OR-Mapper benutzten wir JPA.

Für das Projektmanagement verwendeten wir Scrum, erstellten ein Pflichtenheft und orientierten uns an Meilensteinen.

Zum Schluss ging es noch darum, welche Datenbank wir verwenden möchten. Wir recherchierten, welche Datenbanken uns kostenlos zur Verfügung stehen. Schlussendlich fiel die Entscheidung auf MariaDB. Auf die Gründe für unsere Entscheidung sowie die Vor- und Nachteile dieser Datenbank wird im Kapitel 4.1 eingegangen.

# Kapitel 2

# Technologien

In diesem Kapitel beschreiben wir die verwendeten Technologien und Bibliotheken.

Der Server wurde in der Programmiersprache "Java" geschrieben und der Client wurde mit dem auf Typescript basierenden Framework "Angular" gecodet.

## 2.1 Sprachen (MP)

### 2.1.1 Hypertext Markup Language (HTML)

HTML steht für Hypertext Markup Language und ist eine Grundlage des Internets. Damit werden zum Beispiel Bilder und Texte auf einer Seite strukturiert dargestellt, so auch bei der Tauschbörse. Es werden die grundlegenden Elemente einer Seite, wie zum Beispiel das Navigationsmenü mit Verlinkungen, dargestellt. Außerdem wurde das Formular für die Registrierung ebenso mit HTML dargestellt, es beinhaltet verschiedene div-Container sowie Labels und input-Felder. Die visuelle Darstellung übernimmt allerdings die Stylesheet-Sprache CSS. [38] [17]



Abbildung 2.1: Logo HTML

### 2.1.2 Cascading Style Sheets (CSS)

CSS ist eine Stylesheet-Sprache und ermöglicht es in Kombination mit HTML, Webseiten zu gestalten. Mit CSS wurden nicht nur die Formulare und die Menüs gestaltet, sondern es wurde auch der ganzen Tauschbörse ein eigenes Design gegeben. Außerdem erleichtert es die Anpassung der Formulare, somit sieht das Formular von der Registrierung ähnlich wie das Formular für das Erstellen der Inserate aus. Ein weiterer wesentlicher Aspekt von Cascading Style Sheets ist, dass die Position eines Objektes, beziehungsweise eines div-Containers (eines Bereichs-Containers), bestimmt werden kann. Dies kann in Pixel oder auch Prozent angeben werden. Der CSS Teil wird in einem eigenen File ausgelagert. [17]



Abbildung 2.2: Logo CSS

### 2.1.3 TypeScript (TS)

TypeScript ist eine objektorientierte Programmiersprache, wesentliche Punkte davon sind Variablen, Klassen, Module, Vererbungen und Funktionen. Mit TypeScript wurden die Funktionen auf der Client-Seite geschrieben. Die Daten für den Client werden über die REST Service Klasse aus der Datenbank geholt. Bei Benutzereingaben werden die Daten bevor sie an den Server geschickt werden, auf Fehler überprüft. [24] [107] [80]



Abbildung 2.3: Logo TypeScript

## 2.1.4 Wie unterscheidet sich TypeScript von JavaScript?

Für dieses Unterkapitel wurden folgende Referenzen verwendet: [24] [107] [80] [45]  
Der erste wesentliche Unterschied zwischen TypeScript und JavaScript ist, dass bei TypeScript der Typ einer Variable von Bedeutung ist, anders als bei JavaScript, wo die Typisierung relativ egal ist und sich während des Ablaufes vom Programm verändern kann. Bei TypeScript muss, ähnlich wie bei Java, beim Deklarieren einer Variable gleich ein Typ gesetzt werden.

Der TypeScript Code wird, wenn er kompiliert wird, zu JavaScript Code umgewandelt. Für TypeScript gibt es einige Autocomplete Programme, welche beim Programmieren helfen können.

Der Vorteil von TypeScript besteht, wie oben schon erwähnt, in den Deklarationen der Typen, dadurch werden die Programme um einiges leichter lesbar und ein Softwarefremder kann sich schneller in einen Code einlesen. Außerdem werden durch die Typisierung Fehler vermieden.

## 2.1.5 Angular

Für dieses Kapitel wurden folgende Referenzen verwendet: [6] [4] [8] [7] [9] [5][10] [3] [21] [11] [13]

Angular wird zur Front-End Webentwicklung verwendet und ist ein auf TypeScript basierendes Framework. Der Vorgänger davon ist AngularJS.

### Wie ist ein Angular Projekt aufgebaut?

Ein Angular Projekt besteht grundsätzlich aus Komponenten, Modulen und Klassen.

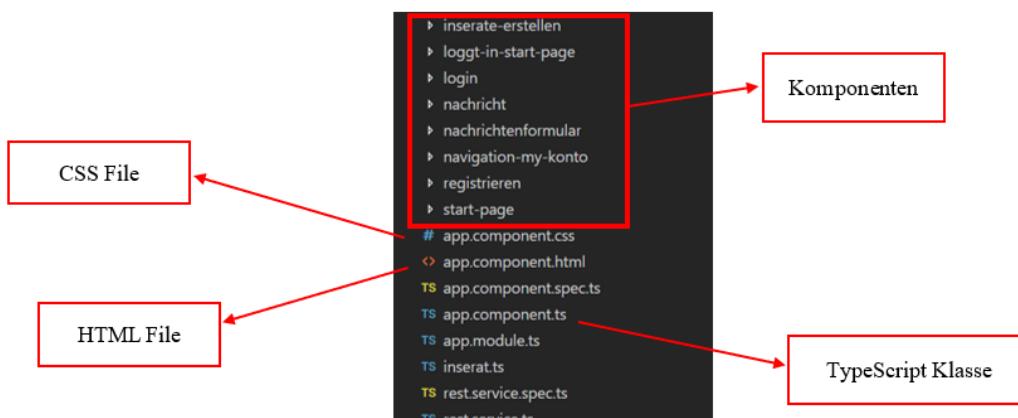


Abbildung 2.4: Projektstruktur - Screenshot Melanie Pittersberger

## Komponenten

Command um eine neue Komponente zu erzeugen:

---

```
ng generate component [name]
```

---

Komponenten grenzen einzelne Bereiche von einander ab. In unserem Projekt gibt es zum Beispiel die "registrieren" Komponente, welche nur für das Registrieren von neuen Benutzer zuständig ist, sie beinhaltet die Daten, Funktionen, sowie auch die Darstellung für diesen Bereich.

In einem Angular Projekt können beliebig viele Komponenten generiert werden. Wenn ein Projekt neu erstellt wird, wird automatisch eine sogenannte Root Komponente (Wurzelkomponente) erstellt. Oft wird die App Struktur mit einer Baumstruktur verglichen, da die verschiedenen Komponenten ineinander verzweigt sind. Eine Komponente ist folgendermaßen aufgebaut:

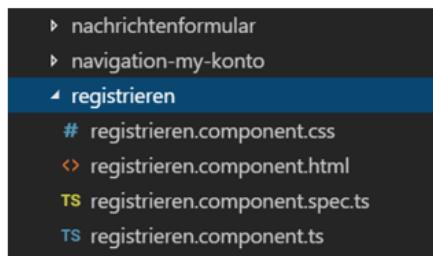


Abbildung 2.5: Komponentenstruktur - Screenshot Melanie Pittersberger

Sie besteht aus einem CSS File, einer HTML File, einer vorgefertigten TypeScript Klasse und einer TypeScript Klasse, welche später für die Methoden zuständig ist. Natürlich ist es möglich, auch noch weitere Klassen in einer Komponente zu erstellen.

## HttpClient

Command um ein neues Service zu erzeugen:

---

```
ng generate service [name]
```

---

Ein Service ist dazu da, Daten vom Server zu holen. Jede Komponente kann auf das Service zugreifen.

Server und Client kommunizieren über das HTTP Protokoll (Hyper Text Transfer Protocol). Der HttpClient beinhaltet verschiedene Funktionen, er behandelt hauptsächlich Requests und Responses der beiden. Normalerweise werden Services geschrieben, um die Darstellung der Daten von dem eigentlichen Datenzugriff zu trennen, es wird dann in den einzelnen Komponenten auf das Service verwiesen.

```

    /**
     * Holt alle Inserate, die in der Datenbank gespeichert sind.
     */
    getInserate(){
        return this.http.get('http://localhost:8080/api/service/inserate');
    }

    /**
     * Holt die Inserate aus der Datenbank, die dem mitübergebenen Filter entsprechen.
     */
    getInserate2(filter){
        return this.http.post('http://localhost:8080/api/service/match', filter);
    }

```

Abbildung 2.6: Auszug aus rest.service.ts - Screenshot Melanie Pittersberger

In der Abbildung 2.6 sieht man zwei REST Aufrufe. Bei dem ersten Aufruf getInserate() werden alle Inserate, welche in der Datenbank gespeichert sind, geholt. Im zweiten Aufruf getInserate2(filter) werden alle Inserate mit dem mitübergebenen Filter aus der Datenbank geholt.

```

setInserate(){
    if(this.matchKatFilter == ''){
        this.rest.getInserate().subscribe(data => {
            this.inserate=data;
            //console.log(this.inserate);
        });
    }
    else{
        this.rest.getInserate2(this.matchKatFilter).subscribe(data => {
            this.inserate=data;
            //console.log(this.inserate);
        });
    }
}

```

Abbildung 2.7: loggt-in-start-page.component.ts - Screenshot Melanie Pittersberger

Wenn sich der Benutzer in der Tauschbörse eingeloggt hat, kommt er zu einer Übersicht aller Inserate. In der Abbildung 2.7 sieht man, wie diese Daten mit Hilfe des Get-Requests verarbeitet werden. Wenn kein Filter ausgewählt wird, wird die Methode getInserate() aus der vorigen Abbildung aufgerufen. Wenn ein Filter angewendet wurde, werden nur die Inserate mit dem jeweiligen Filter vom Server geholt.

Nun hat der Client die Daten vom Server bekommen und kann sie darstellen.

```

<!-- Darstellungsbox -->


<div class="ordnen" *ngFor="let i of inserate, let zahl=index" (click)="act(i);">
    <div *ngIf="checkIfExists(i.art)">
      <div class="container" *ngIf="matchesFilter(i)">
        <div class="box" [style.border] = "i.type == 'suche' ? '2px solid #222222':"
          <div class="imageOption">
            <div *ngIf="i.mediapfad.length>0">
              
              
            </div>
          </div>
          <div class="titel">
            <h3>{{i.titel}}</h3>
          </div>
          <hr>
          <div class="beschreibung">{{i.beschreibung}}</div>
          <div class="footer">
            <h4><i>{{i.art}}</i></h4>
          </div>
        </div>
      </div>
    </div>
  </div>


```

Abbildung 2.8: Darstellung der Daten - Screenshot Melanie Pittersberger

## Wie installiert man Angular?

Um Angular verwenden zu können, muss man ”Node.js” und den ”npm package manager” installiert haben. Um das Angular CLI, das Command Line Interface, zu installieren, wird folgender Command verwendet:

---

```
npm install -g @angular/cli
```

---

Ein Angular Projekt befindet sich immer in einem sogenannten Workspace. Dieser Workspace kann ein oder mehrere Projekte enthalten. Um einen neuen Workspace und ein neues leeres Projekt zu erstellen, muss folgender Command in die Command Line eingegeben werden:

---

```
ng new tauschboerse
```

---

”tauschboerse” steht hier für den Namen des Projektes und des Workspaces.

## Wie wird ein Angular Projekt gestartet?

Das Projekt wird lokal gebaut und ausgeführt. Um das Angular Projekt auszuführen, muss zuerst im Terminal zum Workspace navigiert werden. Danach wird mit folgendem Command das Projekt ausgeführt:

---

```
ng serve -open
```

---

Diese Zeile startet den Angular Server und baute das Projekt immer wieder neu, wenn Änderungen vorgenommen werden. Außerdem wird das Projekt automatisch im Browser geöffnet.

### 2.1.6 Java (AK)

Java ist eine objektorientierte Programmiersprache und eher einfach gehalten, da auf komplexe Sprachkonstrukte verzichtet wird. Zudem ist Java sehr sicher, da sie einen Interpreter, die Java Virtual Maschine, verwendet. [25][42]



Abbildung 2.9: Logo Java

### Wie funktioniert JAVA?

Zuerst wird der Code des Programmes geschrieben und diese Anwendung wird dann mit der Dateiendung .java gespeichert:

---

```
public class TheMain {  
  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
        System.out.println("-----");  
    }  
}
```

---

Der Quellcode wird dann vom Javacompiler umgewandelt in eine .class – Datei, die anschließend den Bytecode enthält. Dieser Bytecode wird dann von der Java Virtual Machine interpretiert und daraufhin in Maschinencode umgewandelt. Der große Vorteil daran ist, dass der Code somit Plattform unabhängig ist.[93]

### Geschichte von Java

Java wurde von Sun Microsystems entwickelt und erschien 1995 am Markt. Der Hauptentwickler war James Gosling und die Sprache wurde innerhalb eines Projektes namens "The Green Project" entwickelt. Sie hätte ursprünglich für die Steuerung von Geräten eingesetzt werden sollen, da mit den vorhandenen Programmiersprachen dies nicht möglich war. Anfangs hieß die Sprache Oak, die kurze Zeit darauf in Java umbenannt wurde, da es rechtliche Probleme mit dem Namen Oak gab. [44]

### 2.1.6.1 Java Persistence API (JPA)

JPA ist eine Sammlung von Klassen und Methoden, um das Speichern von Daten in die Datenbanken zu unterstützen. Es hilft dabei, normale JAVA-Objekte als relationale Datenobjekte abzubilden. [104]

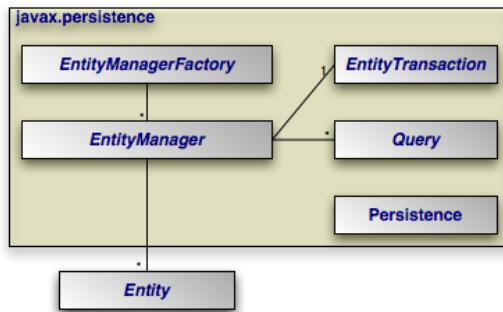


Abbildung 2.10: Darstellung von JPA

#### Entity ManagerFactory und Entity Manager

Die EntityManagerFactory stellt eine Instanz vom EntityManager für die Verbindung zu derselben Datenbank bereit. Alle Instanzen weisen dieselben Einstellungen auf. Der EntityManager wird verwendet, um die Datenbankinteraktion, welche in der Tabelle 2.1 erklärt werden, durchzuführen. [47]

---

```
static EntityManager em =
    Persistence.createEntityManagerFactory("PersistenceUnit").createEntityManager();
```

---

Wichtige Entity Manager Methoden:

<code>persist</code>	Neue Objekte werden eingefügt.
<code>merge</code>	Neue Objekte werden eingefügt und vorhandene Objekte werden aktualisiert.
<code>find</code>	Es wird der Eintrag gesucht, indem man den Primary Key angibt und wenn er existiert, wird das gefundene Objekt zurückgeliefert.
<code>remove</code>	Löscht die Entity Instanz.
<code>createQuery</code>	Dient zum Erstellen einer JPQL - Abfrage.
<code>createNativeQuery</code>	Dient zum Erstellen einer SQL - Abfrage.

Tabelle 2.1: Wichtige EntityManger Methoden [43] [46] [47]

## Entity

Eine Entity ist ein Objekt, das in der Datenbank gespeichert oder bearbeitet wird. In NetBeans können die Entity Klassen automatisch generiert werden, dies hat den wesentlichen Vorteil, dass sie dann auch gleich in der persistence.xml eingetragen werden. Es kann aber auch von einer normalen Java Klasse ausgegangen werden, dann werden aber folgende Schritte benötigt.

Zuerst muss die Annotation @Entity hinzugefügt werden, damit die Klasse als Entity erkannt wird. Als nächsten Schritt braucht die Klasse einen Primary Key, der mit @Id gekennzeichnet wird. In unserem Beispiel wollen wir den Primary Key automatisch generieren, dazu benötigen wir die @GeneratedValue(strategy = GenerationType.AUTO) Annotation.

Als letzten Schritt muss die Entity noch in die persistence.xml eingetragen werden. Die persistence.xml ist eine Datei, die für das Konfigurieren der Datenbank und Entity-Klassen zuständig ist. [108]

```
@Entity
public class Kategorie implements Serializable {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int id;
    private String kategorienname;

    public Kategorie() {
    }

    public Kategorie(String kategorienname) {
        this.kategorienname = kategorienname;
    }

    public String getKategorienname() {
        return kategorienname;
    }

    public void setKategorienname(String kategorienname) {
        this.kategorienname = kategorienname;
    }
}
```

Abbildung 2.11: Entity Klasse - Screenshot Antonia Kriegner

Wichtige Annotationen für das Erstellen von Entities:

@Id	Markiert den Primary Key.
@NotNull	Die Zeile darf nicht null sein.
@Column (unique=true)	Die Einträge in dieser Spalte müssen einzigartig sein.
@JoinColumn	Dient zum Überschreiben des Default-Namens der Join-Spalte. [105]
@GeneratedValue (strategy = GenerationType.AUTO)	Wird benutzt, um den Primary Key automatisch erstellen zu lassen.

Tabelle 2.2: Wichtige Annotationen

## Beziehungen zwischen Entities

Zwischen Entitäten herrschen Beziehungen. Dabei sind folgende Beziehungen möglich:

- @ManyToOne: Ist eine \* zu 1 Beziehung. Diese kann angewendet werden, wenn zum Beispiel zu mehreren Terminen ein Ersteller gespeichert wird.
- @OneToMany: Ist eine 1 zu \* Beziehung. Diese Beziehung haben wir in unserer Anwendung mehrmals benutzt. Zum Beispiel bei der Verbindung von Benutzer und Inserat bzw. Benutzer und Nachricht.
- @OneToOne: Ist eine 1 zu 1 Verbindung. Diese Beziehung haben wir benötigt, um zu einem Benutzer genau ein Konto zuzuordnen.
- @ManyToMany: Ist eine \* zu \* Beziehung. Ein Beispiel dafür ist: Ein Schüler arbeitet an mehreren Projekten und ein Projekt wird von mehreren Schülern bearbeitet.

## 2.2 Werkzeuge

### 2.2.1 Visual Studio Code (MP)

Visual Studio Code ist ein kostenloser Editor von Microsoft. Der Editor wird von den Betriebssystemen Windows, macOS und Linux unterstützt. Es werden viele Sprachen unterstützt und die Syntax ist durch unterschiedliche Schriftdarstellungen sehr leicht lesbar. Außerdem gibt es Auto vervollständigungstools und ein integriertes Terminal. [83]

Um ein Angular Projekt zu öffnen, klickt man auf die Schaltfläche "File" und dann auf die Schaltfläche "Open Folder" und wählt das gewünschte Projekt aus. Danach erscheint die Projektstruktur auf der linken Seite des Editors und der Code des jeweils ausgewählten Files auf der rechten Seite.

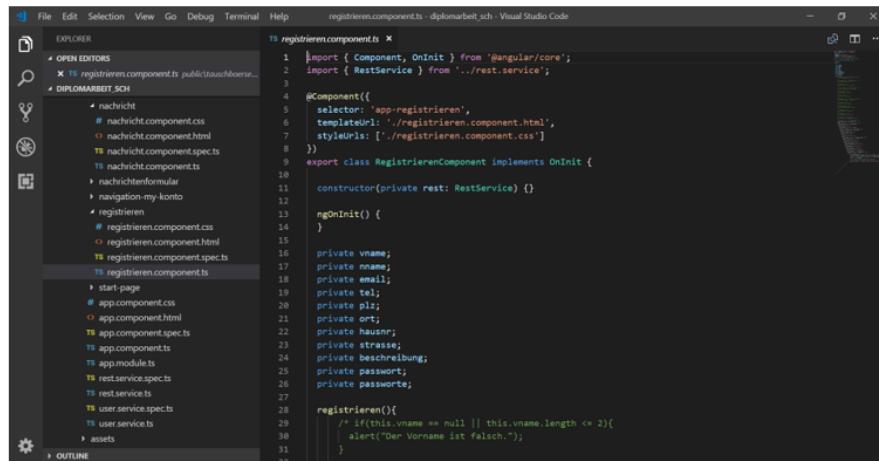


Abbildung 2.12: Oberfläche von Visual Studio Code - Screenshot Melanie Pittersberger

## 2.2.2 NetBeans (AK)

NetBeans ist eine Entwicklungsumgebung, die hauptsächlich für Java entwickelt wurde, aber auch für das Programmieren von C und C++ verwendet werden kann. Durch Installieren eines Plugins kann NetBeans auch für Sprachen wie Python, LaTeX, Ruby und noch viele mehr erweitert werden. [61]

### Oberfläche

NetBeans ist standardmäßig so aufgebaut, dass rechts zwischen den Registerkarten „Services“, „Favorites“ und „Projects“ gewechselt werden kann. Unter „Services“ befinden sich die Einstellungen für die Datenbank. Hier können neue Datenbankconnections erstellt werden. Unter „Favorites“ können für das Arbeiten notwendige Ordner geöffnet werden. Bei der Registerkarte „Projects“ wird eine Liste von den aktuell geöffneten Projekten angezeigt. Ein weiteres wichtiges Fenster für das Entwickeln in NetBeans ist das Fenster für die Ausgaben, dieses „Output-Fenster“ wird meistens in der unteren Ecke angezeigt.

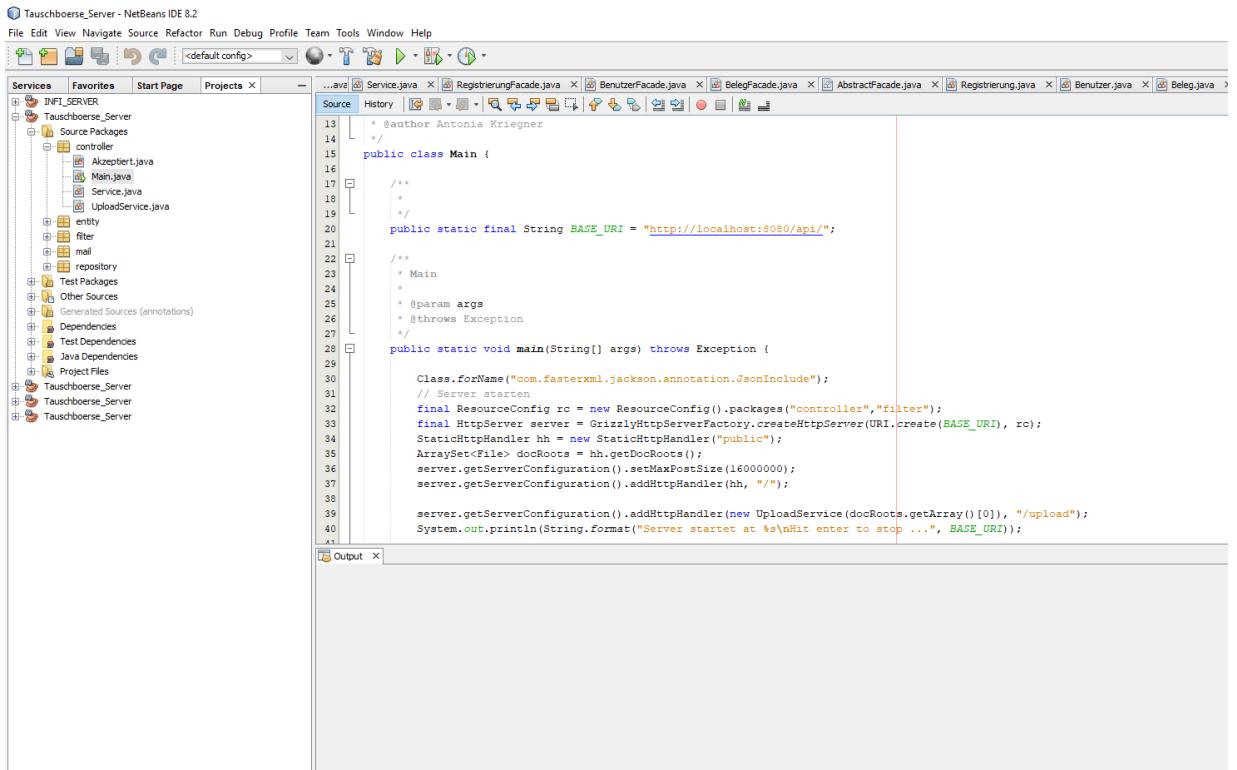


Abbildung 2.13: Oberfläche von NetBeans - Screenshot Antonia Kriegner

## Debuggen in NetBeans

Das Debuggen hilft dabei, in einem Programm Fehler zu finden, da die Werte der Variablen zum Zeitpunkt des Durchlaufens der Methode angesehen werden können. Referenz für nachfolgenden Inhalt: [34]

Um den Debug Modus nützen zu können, muss vor dem Starten ein Breakpoint gesetzt werden. Dieser unterbricht die Programmausführung an der ausgewählten Stelle und ermöglicht das Inspizieren des Status. Ein Haltepunkt kann gesetzt werden, indem man neben dem Code, bei der Zeilennummerierung, auf die linke Maustaste klickt. Dieser Punkt wird anschließend als rotes Rechteck neben dem Sourcecode dargestellt und die Zeile wird rot hinterlegt.

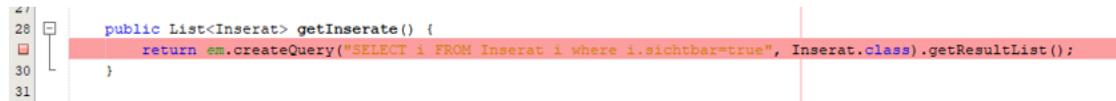


Abbildung 2.14: Breakpoint in NetBeans - Screenshot Antonia Kriegner

Durch Klicken auf den Button wird das Debuggen gestartet. Das Programm wird dort abgefangen, wo der Breakpoint gesetzt wurde. Dabei wird die Zeile grün hinterlegt und mit einem Pfeil gekennzeichnet. Diese Stelle markiert, was im nächsten Schritt ausgeführt werden soll.

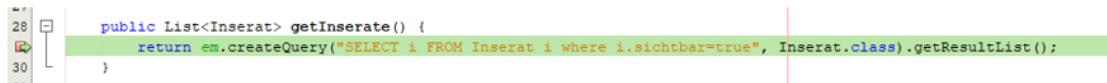


Abbildung 2.15: aktivierter Breakpoint in NetBeans - Screenshot Antonia Kriegner

Zum Navigieren wird eine Palette an Werkzeugen bereitgestellt. Der erste rote Button beendet das Debuggen und der grüne Button ist für das weitere Ausführen zuständig. Der erste blaue Button überspringt das Ganze, der zweite überspringt den Ausdruck, der dritte geht in eine Funktion hinein und der letzte geht aus einer Funktion heraus.



Abbildung 2.16: Buttons zum Debuggen - Screenshot Antonia Kriegner

Die Werte der Variablen können durch „hover“ auf die jeweilige Variable ersichtlich werden oder in einem eigenen Fenster eingesehen werden. Dabei kann durch das Klicken auf das „+“ jedes einzelne Attribut eines Objekts betrachtet werden.

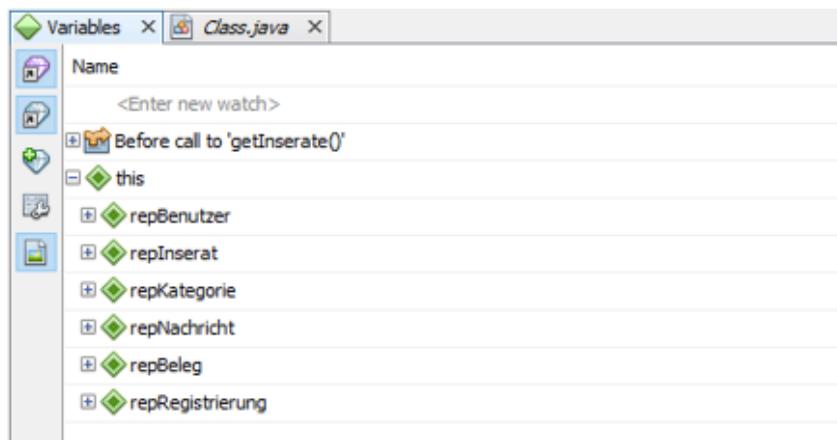


Abbildung 2.17: Variablenfenster beim Debuggen - Screenshot Antonia Kriegner

### 2.2.3 Maven Project (AK)

#### Allgemein

Maven ist ein kostenloses Build-Tool. Es wird verwendet, um Programme beim Kompilieren, Packen und Testen zu unterstützen. Maven ist standardmäßig für Java-Programme ausgelegt. Das Design von Maven basiert auf einer Plugin-Architektur. Dabei können die Plugins verwendet werden, ohne diese direkt installieren zu müssen. [14]

#### Installation

Zuerst muss eine aktuelle Version von Maven heruntergeladen werden. Diese zip-Datei muss anschließend in einem beliebigen Ordner entpackt werden. Als letzten Schritt muss der Pfad des bin-Verzeichnisses in NetBeans richtig gesetzt werden.

#### Erstellen eines Maven Projektes in NetBeans

Um ein Maven Projekt zu erstellen, geht man auf „neues Projekt erstellen“, unter dem Ordner Maven wählt man „Java Application“ aus und klickt auf „weiter“. Anschließend wird ein Name ausgewählt und auf „Finish“ geklickt. Die Felder GroupId, ArtifactId und Version werden automatisch befüllt und können in den meisten Fällen unverändert bleiben.

Dabei gibt die GroupId die eindeutige Identifikation des Herstellers an. Die ArtifactId ist der Name für das Artefakt, dabei gibt die ID den Auslieferungsnamen des Projekts an und die Version gibt den aktuellen Versionsstand des Projekts an. [27]

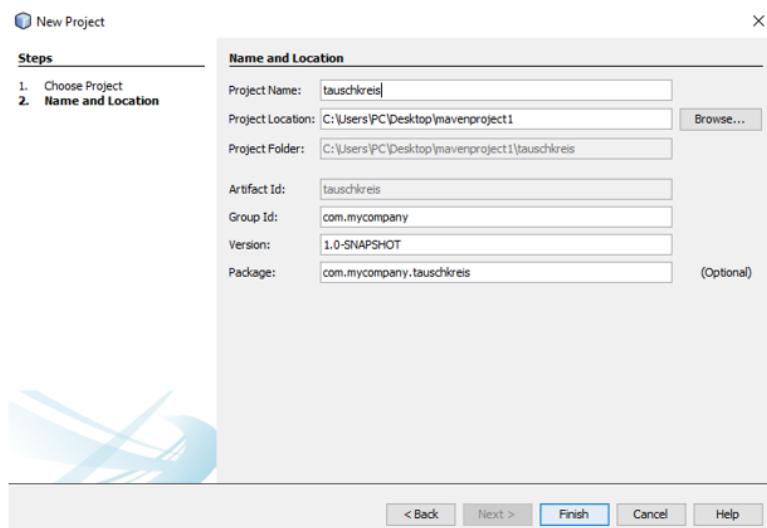


Abbildung 2.18: Erstellen eines Maven Projects - Screenshot Antonia Kriegner

## Was ist pom?

Pom ist eine Arbeitseinheit in Maven und steht für Project Object Model. Dieses pom-File ist vom Dateityp XML. Dabei enthält sie Daten des Projektes und Konfigurationsdetails, die zum Erstellen des Projektes beitragen. Im POM-File können Angaben wie project dependencies, plugins, usw. stehen. Alle POM-Files enthalten ein Project Element und die oben angeführten Felder (groupId, artifactId, version). [109] [54]

Aufbau eines POM-Files:

---

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" m
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>
  <groupId>htl</groupId>
  <artifactId>tauschboerse</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <dependencies>
    <dependency>
      <groupId>javax.mail</groupId>
      <artifactId>mail</artifactId>
      <version>1.4.7</version>
    </dependency>
  </dependencies>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
  </properties>
  <name>Tauschboerse_Server</name>
</project>
```

---

## Super POM

Das Super-POM File ist der Standard von Maven. Alle anderen POM Files erben von dieser Datei. Die Default-Einstellungen können mit dem Befehl ausgelesen werden:

---

```
mvn help:effective-pom
```

---

[109]

## 2.2.4 Adobe Illustrator (AK)

Illustrator ist ein Programm, mit dem vektorbasierte Grafiken erstellt werden können. Es wird vorzugsweise im professionellen Bereich eingesetzt. Der große Vorteil gegenüber anderen Programmen, die pixelorientiert sind, ist, dass die Grafiken unendlich hochskaliert werden können, ohne einen Qualitätsverlust feststellen zu können. Adobe Illustrator ist kostenpflichtig.

[2]

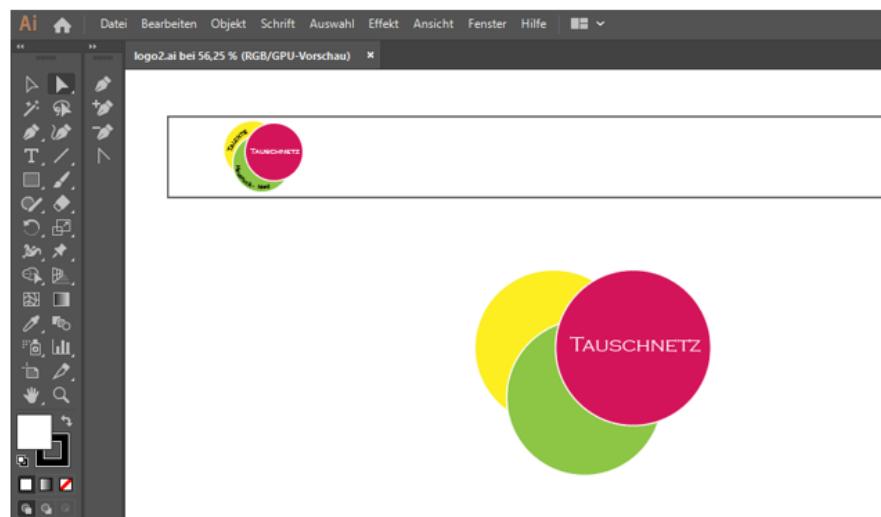


Abbildung 2.19: Oberfläche von Illustrator - Screenshot Antonia Kriegner

Unten rechts befindet sich das Fenster für die Ebenen. Es ist wichtig, mit mehreren Ebenen zu arbeiten, um Objekte überlagern zu können bzw. mehrere Arbeitsschritte am selben Objekt ausführen zu können.

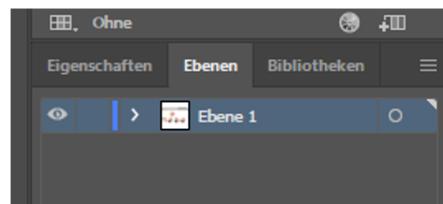


Abbildung 2.20: Ebenen Fenster - Screenshot Antonia Kriegner

## 2.3 Projektmanagement

In diesem Kapitel geht es um die Organisationselemente unserer Diplomarbeit, daher werden hier Werkzeuge wie das Lastenheft, das Pflichtenheft, die Meilensteinliste, Gitlab und Scrum erklärt.

### 2.3.1 Lastenheft (MP)

Das Lastenheft wird vom Auftraggeber vorgegeben. Es beinhaltet die Anforderung des Auftraggebers an ein Programm. Es darf allerdings noch keine Lösung für die Realisierung beinhalten.

Die wesentlichen Punkte eines Lastenheftes:

- Die Ausgangslage beschreiben
- Was soll das Projekt am Ende können?
- Die notwendigen Funktionen
- Meilensteine [84]

### 2.3.2 Pflichtenheft (MP)

#### Allgemein

Ein Pflichtenheft wird vom Auftragnehmer erstellt und beinhaltet die technische Umsetzung des Projektes. Die Grundlage für das Pflichtenheft ist das Lastenheft. Der Grundgedanke bei einem Pflichtenheft ist es, abgrenzen zu können was das Endprogramm können muss, welche Technologien für die Realisierung verwendet werden und was die Rahmenbedingungen sind. Das heißt, gibt es ein Budget, wenn ja, wie groß ist es und gibt es einen begrenzten Zeitraum für die Entwicklung des Projektes.

Ein weiterer wichtiger Punkt, welcher vor der Erstellung eines Pflichtenheftes abzuklären ist, ist, ob schon ein Teil des Projektes vorhanden ist, das heißt, der Ist-Zustand muss abgeklärt werden. Falls dies der Fall ist, muss im Pflichtenheft der Soll-Zustand, das heißt, was in Zukunft gemacht werden muss, gut erkennbar vom Ist-Zustand getrennt sein.

Außerdem wird das Layout im Pflichtenheft schon grob skizziert. [85]

Im Anhang befindet sich unser vollständiges Pflichtenheft.

#### Gliederung des Pflichtenheftes:

- Zielbestimmungen
  - Musskriterien
  - Wunschkriterien
  - Abgrenzungskriterien

- Produkteinsatz
  - Anwendungsbereiche
  - Zielgruppen
  - Betriebsbedingungen
- Produktumgebung
  - Software
  - Hardware
  - Orgware
- Produktfunktionen
  - Benutzerfunktionen
    - \* Benutzer-Kennung
    - \* Persönliche Daten
    - \* Persönliche Konfiguration
    - \* Persönliches Profil
  - Talente
  - Inserate
- Produktdaten
- Produktleistungen
- Benutzungsoberfläche
  - Dialogstruktur
    - \* Startseite
    - \* Hauptseite
    - \* Benutzermenü
  - Bildschirmlayout
- Qualitätszielbestimmungen
- Globale Testszenarien und Testfälle
- Entwicklungsumgebung
  - Software
  - Hardware
  - Orgware
- Ergänzungen
- Glossar

Die Struktur des Pflichtenheftes ist angelehnt an die folgenden Referenzen: [67][1][68]

### 2.3.3 Meilensteinliste (MP)

Meilensteinliste:

Nr	Meilenstein	Termin	Tatsächliches Enddatum
1	Pflichtenheft fertiggestellt	22.07.2018	01.08.2018
2	Einträge können erstellt werden	25.08.2018	25.11.2018
3	Einträge können angezeigt werden	03.09.2018	29.10.2018
4	Nachrichtenportal fertig implementiert	20.09.2018	02.11.2018
5	Abbuchung der Talente implementiert	15.10.2018	08.03.2019
6	Login implementiert	10.11.2018	17.11.2018
7	Prototyp/Funktionsumfang fertig implementiert	03.12.2018	08.03.2019
8	Entwurf für die Probelesung der Abschrift bzw. Diplomarbeit fertig gestellt	04.03.2019	04.03.2019
9	Endgültige Abgabe der Diplomarbeit	04.04.2019	04.04.2019

Tabelle 2.3: Meilensteinliste

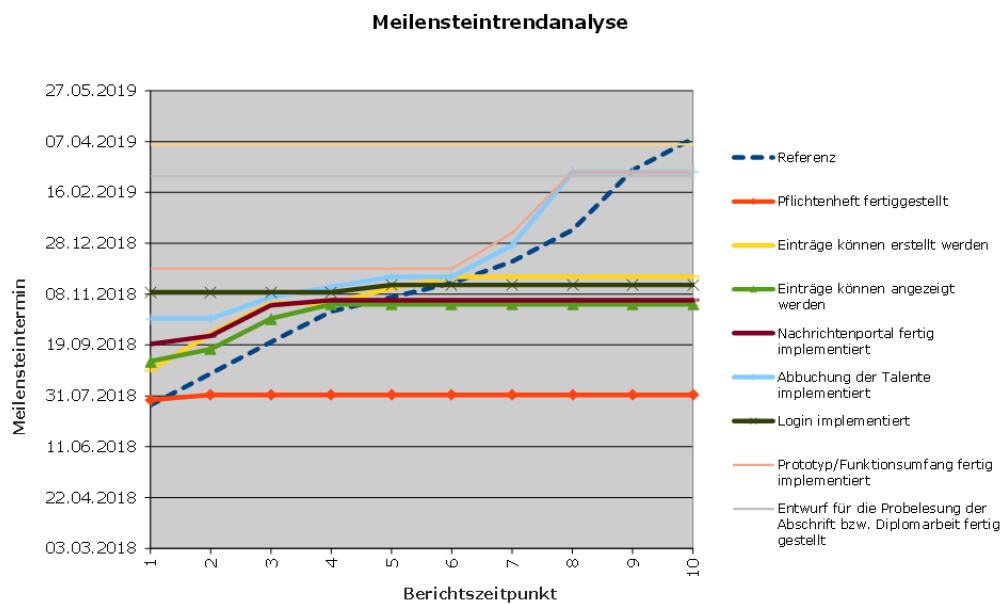


Abbildung 2.21: Meilensteintrendanalyse - Screenshot Antonia Kriegner

Wir haben insgesamt acht Meilensteine definiert. Die obenstehenden Meilensteine definieren den Zeitpunkt, an dem die jeweilige Aktivität abgeschlossen sein soll. Grundsätzlich verwendet man Meilensteine in einem Projekt, um dem Ganzen Struktur zu geben. Für einen Meilenstein gibt es immer ein Datum, wann die Aktivität fertig sein soll, dieses Datum kann sich jedoch auch nach hinten verschieben, wenn es zu Komplikationen kommt, beziehungsweise wenn Änderungen im Ablauf beschlossen werden. [87]

Die Fertigstellung der Meilensteine verzögerte sich grundsätzlich etwas, da wir die Zeit für die einzelnen Punkte in den Ferien sehr knapp bemessen hatten. Der Grund dafür war, dass Antonia Kriegner das Fach Naturwissenschaften vorgezogen maturierte und Melanie Pittersberger in den Sommerferien arbeitete. Deshalb gab es zu Beginn des Schuljahres eine Verzögerung von drei bis vier Wochen, welche sich durch das Projekt durchzog. Über diese Verzögerung wurden die Auftraggeber informiert, dies stellte kein Problem dar.

#### **1.) Pflichtenheft fertiggestellt:**

Der erste Meilenstein war „Pflichtenheft fertiggestellt“. Das Pflichtenheft befindet sich im Anhang. Dieser Meilenstein wurde zehn Tage nach dem geplanten Termin fertig gestellt, da wir das Pflichtenheft noch von Prof. Mag. Reder absegnen lassen mussten.

#### **2.) Einträge können erstellt werden:**

„Einträge können erstellt werden“ wurde drei Monate nach dem angesetzten Termin fertig gestellt. Es kam zu einer Verzögerung, da wir den Zeitaufwand unterschätzten.

#### **3.) Einträge können angezeigt werden:**

Der dritte Meilenstein wurde fast zwei Monate später als angesetzt, fertig implementiert, da Antonia Kriegner vorgezogen das Fach Naturwissenschaften maturierte.

#### **4.) Nachrichtenportal fertig implementiert:**

„Nachrichtenportal fertig implementiert“ verzögerte sich aus demselben Grund wie der vorherige Meilenstein.

#### **5.) Abbuchung der Talente implementiert:**

Wir beschlossen, die schriftliche Arbeit vorzuziehen, weshalb sich einige Änderung im Ablauf ergaben. Dadurch verzögerte sich die Implementierung der Talenteabbuchung

#### **6.) Login implementiert:**

Der Meilenstein „Login implementieren“ verschob sich nur um sieben Tage. Er verzögerte sich aufgrund der oben schon genannten Gründe.

**7.) Prototyp/Funktionsumfang fertig implementiert:**

Dieser Meilenstein wurde später fertig, da wir noch ein paar kleine Änderungen zu machen hatten und uns zuerst auf den schriftlichen Teil konzentrieren wollten.

**8.) Entwurf für die Probelesung der Abschrift bzw. Diplomarbeit fertig gestellt:**

Bei diesem Meilenstein wird die fertige schriftliche Arbeit zum Korrekturlesen an Herrn Prof. Mag. Reder geschickt. In diesem Meilenstein ist noch nicht mitbegripen, dass die Diplomarbeit von anderen Leuten auch Korrektur gelesen wurde.

**9.) Endgültige Abgabe der Diplomarbeit**

Bei diesem Meilenstein wird die Diplomarbeit fertig abgegeben.

### 2.3.4 Gitlab (AK)

Gitlab ist für das gemeinsame Entwickeln eines Projektes sehr gut geeignet. Es bietet die Möglichkeit, das Projekt laufend zu aktualisieren, dabei gehen die alten Versionen nicht verloren.

#### Aufbau von Git

Git basiert auf folgenden Ebenen: Arbeitsverzeichnis, Staging Area und dem Git Verzeichnis. Das Arbeitsverzeichnis ist der Bereich, in dem Bearbeitungen erfolgen können. Wenn Änderungen gemacht wurden, dann befindet man sich im Zustand „modified“. Nach dem Bearbeiten der Dateien wird in den Zustand „staged“ gewechselt, den man mit dem Befehl „add“ erreicht. In der Staging Area befindet sich eine Datei, in der die für den nächsten Commit relevanten Daten gespeichert werden. Nach diesem Schritt werden die Dateien, die in der Staging Area vorgemerkt sind, mit dem Befehl „commit“ in das Git Verzeichnis geladen. Der Zustand des Projektes ist nun „committed“. [30] [90]

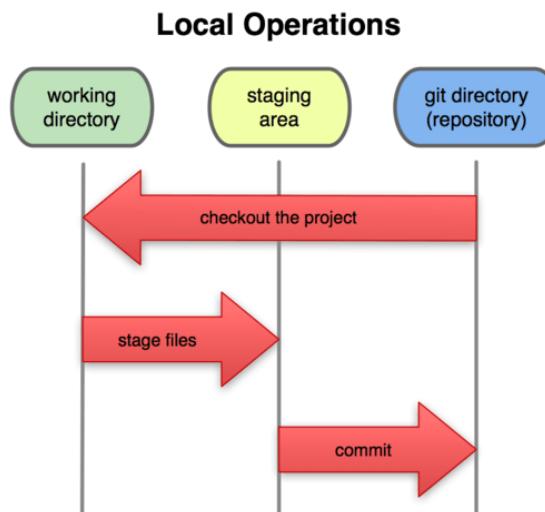


Abbildung 2.22: Funktionsweise von Git

#### Projekt auf GitLab stellen

Zuerst muss ein Projekt erstellt werden, in welchem später der Code hochgeladen werden kann. Nach Anlegen eines Projektes muss ein Repository erzeugt werden, in dem ein bestehendes Projekt eingefügt wird. Dazu öffnen wir die Kommandozeile und wechseln mit „cd“ auf den richtigen Ordner, der hochgeladen werden soll. Um das Projekt nun auf Git geben zu können, wird im ersten Schritt ein Git Repository erzeugt, mit dem Befehl:

---

```
git init
```

---

Im nächsten Schritt wird ein Remote Repository erstellt:

---

```
git remote add origin https://gitlab.com/m.pittersberger/tauschboerseDA.git
```

---

Anschließend wird der Befehl

---

```
git add .
```

---

ausgeführt, der alle Dateien in die Staging Area ladet und diese Dateien werden beim nächsten Commit erfasst. Nach dem Hinzufügen wird "git commit" ausgeführt und alle Änderungen werden gespeichert.

[57] [31]

## Branches

Ein Branch besteht aus dem aktuellen Zustand des Projektes. Das Projekt besteht aus einem „master“ Branch, der den Zustand „protected“ hat. Dieser kann nicht gelöscht werden. Zudem können nun Branches vom Masterbranch erstellt werden, in denen man entwickeln kann, diese werden anschließend wieder in den Master gemerged, außer ein Stand soll nicht in den bestehenden Master gegeben werden, da er zum Beispiel noch nicht einwandfrei funktioniert. Der Master sollte immer eine lauffähige Version haben.



```
Y master default protected
-> 9b658479 · Merge branch 'Verbesserungen_Kriegner' into 'master' · 1 month ago
```

Abbildung 2.23: Branches in Git - Screenshot Antonia Kriegner

## Graph

Durch den Graph werden die Branches grafisch dargestellt, dadurch ist ersichtlich, welcher Branch woher abgeleitet wurde. In diesem Beispiel wurde vom ersten Commit der Branch „Einträge anzeigen“ erstellt und dieser wurde anschließend wieder in den „master“ gemerged:

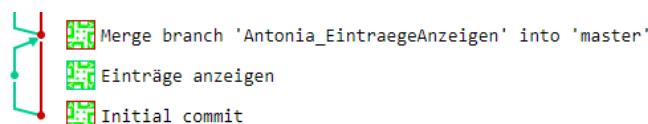


Abbildung 2.24: Graph in Git - Screenshot Antonia Kriegnerr

### 2.3.5 Scrum und Tool ScrumDo (AK)

In diesem Kapitel wird unsere Projektmanagement Methode Scrum erläutert. Neben Scrum gibt es auch zum Beispiel die Methode Kanban. Für die Ausarbeitung haben wir folgende Quellen verwendet: [71] [110] [98] [77] [78] [91]

Scrum ist eine agile Vorgehensweise, die beim Entwickeln von Applikationen eingesetzt werden kann. Dabei wird in sogenannten „Sprints“ das Projekt entwickelt und am Ende jedes Sprints gibt es ein Treffen mit dem Auftraggeber, dem der Fortschritt präsentiert wird. Wichtig ist hierbei, dass nur die Teile dem Kunden präsentiert werden, die auch zu hundert Prozent funktionieren. Halbfertige Versionen dürfen hier nicht gezeigt werden. Um die Meinung des Auftraggebers miteinzubeziehen, werden die Anmerkungen des Gesprächs schriftlich festgehalten und später in der Planungsphase berücksichtigt.

In der Planungsphase wird das sogenannte „Sprint Planning“ angefertigt. Dieses enthält die jeweiligen User Storys, welche Teilbereiche des Projektes sind, die bis zum nächsten Treffen realisiert werden sollen. Bei der User Story werden folgende Attribute angegeben:

- Titel, der immer aus einem Nomen und einem Prädikat besteht
- Beschreibung, die angibt, was die jeweilige User Story können soll
- Spezifikation, hier wird alles genau definiert – was alles umgesetzt werden soll, was alles beachtet werden soll
- Layout, Skizze der späteren Darstellung
- Aufwand, wie lange dafür benötigt wird
- Priorität, wie wichtig diese User Story ist
- Nicht funktionale Anforderungen
- Definition of Done, hier wird definiert, was funktionieren muss, um die User Story als fertig zu kennzeichnen

Zu Beginn des Projektes wird ein ProductBacklog angelegt, der alle User Storys beinhaltet, die im Laufe der Zeit realisiert werden. Wenn größere Anmerkungen seitens Auftraggeber bei der Besprechung gemacht werden, dann werden diese im ProductBacklog in Form einer neuen User Story festgehalten. Der Backlog besteht aus einem Titel und einer Beschreibung der jeweiligen User Story.

## Sprint-Retrospektive

Bei der Sprint Retrospektive wird diskutiert, was beim letzten Sprint gut gelaufen ist, welche Tools funktioniert haben, aber auch die Probleme werden erläutert. Was im nächsten Sprint besser gemacht werden soll und wie das Projektteam das umsetzen möchte. Die Sprint Retrospektive dient dazu, einen schnellen Projekterfolg zu erreichen und weniger effektive Strategien zu ändern.

## Vorteil von Scrum

Es wird laufend mit dem Auftraggeber entwickelt, das heißt, die Chance, dass das Projekt nicht den Anforderungen und Wünschen des Auftraggebers entspricht, ist geringer, da er immer am laufenden Stand ist und Kritikpunkte sofort anmerken kann. Dadurch erspart man sich auch Geld, da meist das Endprodukt auch abgenommen wird, da die Anwendung so wird, wie der Kunde es sich auch vorstellt.

## Nachteil von Scrum

Meetings sind sehr oft anzusetzen und das ist sehr zeitintensiv. Oft gibt es auch das Problem, dass sich die Termine nicht richtig vereinbaren lassen, da sehr viele Personen an einem Gespräch teilnehmen sollen und dies zu koordinieren nicht immer einfach ist.

## ScrumDo

ScrumDo ist ein Tool, in welchem man pro Sprint die Tasks aufteilen kann und dann sieht, wo sich der Task gerade befindet, zum Beispiel ob er noch in der Warteschlange ist, gerade bearbeitet wird oder schon fertiggestellt wurde. Zusätzlich kann einem Task eine Person zugewiesen werden, die diesen Task bearbeiten muss. Dem Task kann ebenfalls ein Datum hinzugefügt werden, bis wann er erledigt werden muss.

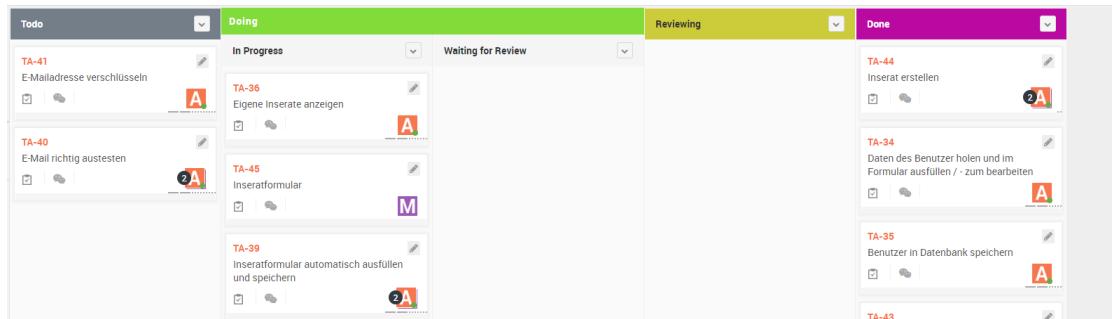


Abbildung 2.25: Oberfläche von ScrumDo - Screenshot Antonia Kriegner

Das Tool inkludiert auch einen Notizblock, wo jemand eine Notiz erzeugen kann und auch die anderen Mitglieder des Teams einen Kommentar dazu abgeben können.

# Kapitel 3

## Realisierung

### 3.1 Überblick (MP)

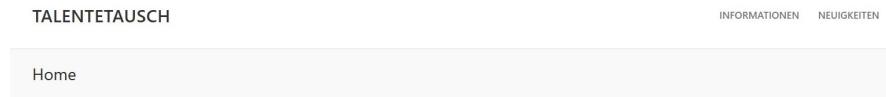


Abbildung 3.1: Wordpress Menüleiste - Screenshot Melanie Pittersberger

In der Abbildung 3.1 sieht man die Menüleiste von unserem Wordpress Anteil.

Der Wordpress Teil besteht aus der Startseite, Informationen und Neuigkeiten. Die Startseite ist die erste Seite, welche man sieht, wenn man die Webseite aufruft. Oben rechts gibt es die Menübuttons "Informationen" und "Neuigkeiten".

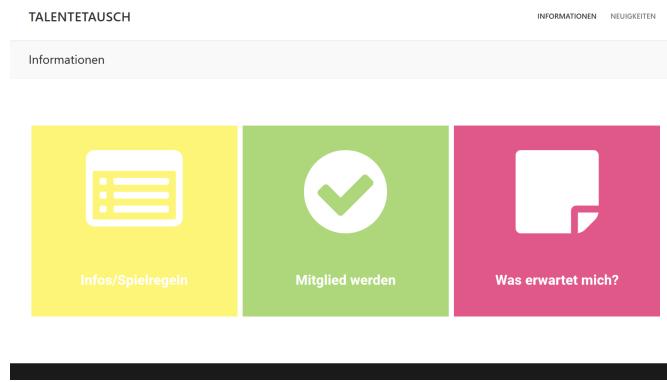


Abbildung 3.2: Wordpress Informationen Seite - Screenshot Melanie Pittersberger

Wenn man auf den „Informationen“ Button klickt, kommt man auf eine Unterseite, wie in Abbildung 3.2 dargestellt. Diese Seite enthält wieder die Menüleiste und die drei Unterseiten „Infos/Spielregeln“, „Mitglied werden“ und „Was erwartet mich?“.

TALENTETAUSCH

INFORMATIONEN NEUIGKEITEN

Mitglied werden

TeilnehmerIn wird man nach Registrierung, Unterzeichnung der Teilnahmeerklärung, sowie nach Bezahlung des ersten Mitgliedsbeitrages.

**Mitmachen können**  
Frauen und Männer sowie Jugendliche ab 16 Jahren.  
Alle, die einen neuen Weg der Nachbarschaftshilfe gehen wollen, sind herzlich eingeladen.  
Je mehr mitmachen, desto bunter und reichhaltiger wird das Angebot.

---

**Registrierung**  
Als erster Schritt hat eine Anmeldung sowie eine gleichzeitige Registrierung zu erfolgen. Es erfolgt eine automatische Bestätigungsmaile.

---

**Einzahlung und Freigabe**  
Anschließend ist die Teilnahmegebühr bei Ersteintritt zu entrichten und auf das Konto des Tausch-und Talentenetzes Hausruck Nord zu überweisen.

Abbildung 3.3: Wordpress Unterseite Mitglied werden - Screenshot Melanie Pittersberger

In der Abbildung 3.3 ist die Unterseite „Mitglied werden“ dargestellt. Diese Seite beinhaltet die Anleitung zur Anmeldung.

TALENTETAUSCH

INFORMATIONEN NEUIGKEITEN

Neuigkeiten

Willkommen bei den Neuigkeiten



UNCATEGORIZED

**Jahreshauptversammlung des Talentetauschkreises**

TALENTEADMIN / JANUARY 16, 2019 / 1 REPLY

Wir laden euch sehr herzlich zu der Jahreshauptversammlung des Talentetauschkreises Hausruck Nord ein!

[Read More](#)

---



UNCATEGORIZED

...

Abbildung 3.4: Wordpress Neuigkeiten - Screenshot Melanie Pittersberger

In der Abbildung 3.4 ist die Neuigkeiten-Seite abgebildet. Diese soll einen Überblick über die kommenden Veranstaltungen, wie zum Beispiel Tauschtreffen geben. Die Grundinhalte werden mithilfe von Image Boxen dargestellt.

## 3.2 Grund der Aufteilung (MP)

Wir haben uns dazu entschieden, den Informationen und Neuigkeiten Teil unseres Projektes auf Wordpress auszulagern, da unsere Auftraggeber diese Seiten ständig updaten können möchten. Am Anfang waren diese zwei Seiten zwar als fixe Bestandteile des Projektes, welche mithilfe von Angular dargestellt werden, geplant. Im Laufe der Diplomarbeit entstand allerdings der Wunsch seitens Auftraggeber, diese Seiten ständig ändern zu können. Mithilfe von Wordpress ist dies möglich, da es den Administrator Bereich gibt und die zuständigen Personen schnell und einfach die Texte wechseln können.

## 3.3 Informationsseite - Wordpress

### 3.3.1 Elementor Plugin (AK)

Elementor ist ein Plugin, das hilft, Front-End-Designs schneller und einfacher zu gestalten. Zudem ist es ein Open Source Produkt, es bietet die Möglichkeit, laufend weiterentwickelt zu werden. Wenn das Plugin installiert und aktiviert wurde, dann wird der Editor in der Bearbeitungsansicht angezeigt. [28]

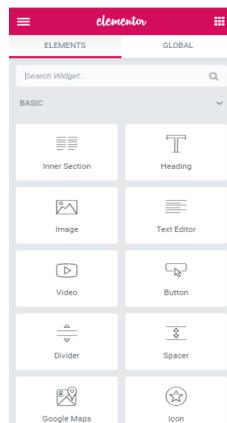


Abbildung 3.5: Elementoroberfläche - Screenshot Antonia Kriegner

Ein wesentlicher Vorteil von Elementor liegt darin, dass es ideal für die Anpassung des Layouts ist. Es können Abstände geregelt werden, Höhen und Breiten von Spalten angepasst werden und die Position des Inhaltes verändert werden. Die Gratisversion, die wir bei unserer WordPress Seite verwendet haben, inkludiert 30 Basis Widgets und den Drag and Drop Editor. Zu den bekanntesten kostenlosen Features gehören das Einfügen von Bildern, Texten, Videos, Buttons, Google Maps, Icons und Tabellen. [28]

## 3.4 Applikation

### 3.4.1 Architektur (AK)

#### REST

Die Ausarbeitung zum Thema REST wurde mit Hilfe der folgenden Quellen gemacht:  
[85][92][22]

REST steht für Representational State Transfer und ist eine Programmierschnittstelle zwischen Client und Server. Die Anfragen vom Client werden über HTTP-Methoden an den Server geschickt und dieser liefert als Antwort die Daten, meist als JSON Format. Wenn der Server einen Response schickt, dann muss im Header angegeben werden, welchen Content-Type er hat.

HTTP – Methoden:

- GET – holt Daten vom Server
- POST – sendet Daten an den Server
- DELETE – löscht Daten vom Server
- PUT – ändert Daten am Server

Response - Status:

200 (OK)	Erfolgreicher http - Request
400 (BAD REQUEST)	Der Request kann nicht ausgeführt werden aufgrund einer fehlerhaften Syntax.
404 (NOT FOUND)	Die Ressource wurde nicht gefunden. Es kann sein, dass sie gelöscht wurde oder dass sie noch nicht existiert.

Tabelle 3.1: Response – Status

Der wesentliche Vorteil von REST ist, dass Client und Server unabhängig voneinander sind. Somit kann am Client etwas geändert werden, ohne den Server zu beeinflussen und umgekehrt. Es müssen nur die gesendeten Nachrichten übereinstimmen. Das World Wide Web verfügt über die Infrastruktur des REST, somit können kleine Aufrufe auch direkt im Browser gemacht werden.

#### JSON

JSON steht für JavaScript Object Notation. Es ist ideal geeignet für den Datenaustausch und aufgrund der einfachen Syntax ist es gut lesbar.

Referenzen für den Inhalt des Kapitels: [49] [23] [45] [89]

JSON basiert auf zwei Strukturen:

Die erste Struktur ist eine Menge von untergeordneten Schlüssel-Wert-Paaren.

---

```
{  
    "idAuto" : 3,  
    "titel" : "Rasenmähen",  
    "preis" : 50,  
    "art" : "Dienstleistung",  
    "type" : "biete",  
    "beschreibung" : "Ich kann für sie den Rasen mähen."}
```

---

Die zweite Struktur ist eine Liste an Werten.

---

```
[{  
    "idAuto" : 3,  
    "titel" : "Rasenmähen",  
    "preis" : 50,  
},  
{  
    "idAuto" : 3,  
    "titel" : "Rasenmähen",  
    "preis" : 50,  
}]
```

---

## Unterschied XML - JSON

XML steht für Extensible Mark-up Language. Der wesentliche Vorteil von XML ist, dass es leichter auszulesen ist als JSON, wenn die Daten tief verschachtelt sind.

Im Gegensatz ist JSON schneller zu lesen und zu schreiben. Zudem kann man mit JSON auch Arrays schicken, was mit XML nicht möglich ist. Ein weiterer Vorteil von JSON ist, dass es kürzer ist, kompakter ist und somit nicht so viel Speicher benötigt. Zusätzlich kann man einzelne Knoten in einer XML-Datei nicht leicht ansprechen. [50] [95] [48]

JSON:[50]

---

```
{  
    "benutzer": [  
        { "vname": "Antonia", "nname": "Kriegner" },  
        { "vname": "Melanie", "nname": "Pittersberger" }]  
}
```

---

XML: [50]

---

```
< benutzern>
  < benutzer >
    < vname > Antonia </ vname >
    < nname > Kriegner </ nname >
  </ benutzer >
  < benutzer >
    < vname > Melanie </ vname >
    < nname > Pittersberger </ nname >
  </ benutzer >
</ benutzern >
```

---

### Alternative zu JSON - YAML

YAML steht für "YAML Ain't Markup Language" und ist eine Auszeichnungssprache, die von der Form her an XML angelehnt ist. Das Prinzip von YAML ist, dass sich jede Datenstruktur mit assoziativen Listen, Arrays oder Werten abbilden lässt. Der wesentliche Vorteil von YAML ist, dass die Daten einfach weiterverarbeitet werden können. [96] [97]

Beispielcode: [96]

---

```
---
- Beispiel1
- Beispiel2
---
[List1, List2, List3]
```

---

# Kapitel 4

## Server

### 4.1 Datenbank (AK)

#### 4.1.1 MariaDB

*”MariaDB ist ein freies, relationales Open-Source-Datenbankmanagementsystem, das durch eine Abspaltung aus MySQL entstanden ist.”* [52] MariaDB ist bekannt als ein schnelles und sehr stabiles System. Zudem bietet MariaDB ein hohes Maß an Kompatibilität mit MySQL und hat eine exakte Übereinstimmung mit den Befehlen und APIs. [69]

Für die Vorteile und Nachteile ist folgende Quelle als Referenz anzusehen: [53]

#### 4.1.2 Vorteile

##### 1. Engines:

Es werden viele Engines unterstützt, wie zum Beispiel Aria, TokuDB, Spider oder ScaleDB.

##### 2. Weiterentwicklung:

Es wird ständig an neuen Funktionen gearbeitet, um MariaDB an aktuelle Standards anzupassen.

##### 3. Performance:

Die MariaDB ist für die Performance optimiert, das bedeutet, dass auch eine große Ansammlung von Daten gespeichert werden kann.

#### **4.1.3 Nachteile**

##### **1. Updates:**

Bei Umstellungen werden oft die notwendigen Libraries nicht bereitgestellt, was bei einem Update zu Fehlern führen kann.

##### **2. Unterstützung:**

Es kann leicht vorkommen, dass MariaDB nicht vom Betriebssystem oder von der Hosting Umgebung unterstützt wird. Man kann dann aber wieder auf MySQL umsteigen, was ein wesentlicher Vorteil ist.

#### 4.1.4 Installation

Die MariaDB haben wir lokal installiert, da wir von den Auftraggebern noch keinen Serverplatz bekommen haben, wo wir die Datenbank aufsetzen können.

Wenn das Betriebssystem Windows verwendet wird, muss zuerst die .msi Datei heruntergeladen werden und anschließend ausgeführt werden. Diese Datei kann unter folgenden Link gedownloadet werden: <https://downloads.mariadb.org/mariadb/10.3.13/> Dabei muss ein Passwort eingegeben werden. Im nächsten Schritt wird der Servicename, ein Port und die Größe eingegeben. Dann klicken wir auf „Install“ und die Aktion wird abgeschlossen.

Zu Beginn wird der installierte Client geöffnet und das bei der Installation gesetzte Passwort abgefragt. Nach der Eingabe befindet man sich nun auf der eingeloggten Seite. Im nächsten Schritt muss eine Datenbank erstellt werden. Dabei wird folgender Befehl ausgeführt:

---

```
CREATE DATABASE 'databasemariadb';
```

---

Die erstellten Datenbanken kann man mit dem Befehl „show databases“ einsehen. Wenn man auf eine Datenbank wechseln will, nutzt man den Befehl „use databaseother“. Dann befindet man sich auf einer anderen Datenbank. Unsere Tabellen werden mit Hilfe von JPA erzeugt und können nach dem Starten des Programmes und dem Einfügen von Daten abgerufen werden. Dies kann mit dem Befehl „show tables“ gemacht werden:

Tables_in_databasemariadb
beleg
benutzer
benutzer_beleg
benutzer_benutzer
benutzer_inserat
inserat
inserat_beleg
inserat_kategorie
inserat_nachricht
kategorie
konto
nachricht
registrierung
sequence

Abbildung 4.1: Tabellen auf der Datenbank - Screenshot Antonia Kriegner

Gesamte Inhalte einer Tabellle können mit einem select \* from table abgerufen werden, einzelne Zeilen einer Tabelle mit zum Beispiel select \* from table where id=1;

Der gesamte Inhalt wurde durch nachschlagen auf folgenden Seiten erarbeitet: [39] [35] [19]

Zum Verbinden der lokal installierten Datenbank und der Anwendung muss in NetBeans eine neue Datenbank angelegt werden. Da in NetBeans eine Connection zur MariaDB nicht automatisch möglich ist, muss ein zusätzliches „jar“ File heruntergeladen werden und anschließend muss ein neuer Driver hinzugefügt werden. Beim Erstellen des Drivers muss die heruntergeladene Datei ausgewählt werden und beim Punkt Driver Class auf Find geklickt werden. Anschließend wird dieses Feld ausgefüllt und es muss nur noch ein Name angegeben werden und das Anlegen des Drivers ist fertig.

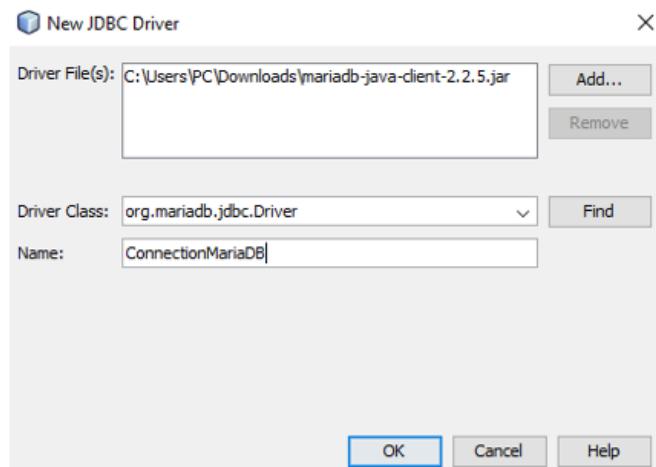


Abbildung 4.2: Driver anlegen - Screenshot Antonia Kriegner

Im nächsten Schritt wird ein Rechtsklick auf den gerade erstellten Driver gemacht und auf „Connect Using“ geklickt. In diesem Fenster müssen anschließend nur noch der Username und das Passwort für die Datenbank und die URL eingegeben werden, somit ist die Verbindung zur Datenbank fertig. Die URL besteht im ersten Schnitt aus den Schlüsselwörtern jdbc:mariadb. Der nächste Abschnitt gibt die IP-Adresse an, auf der die Datenbank läuft, in unserem Fall ist das der localhost, der durch die IP-Adresse 127.0.0.1 abgebildet werden kann. Anschließend muss der Port angegeben werden und im letzten Schritt die Datenbank, die wir zu Beginn angelegt haben (databasemariadb).

Der vollständige Pfad ist in diesem Fenster ersichtlich:

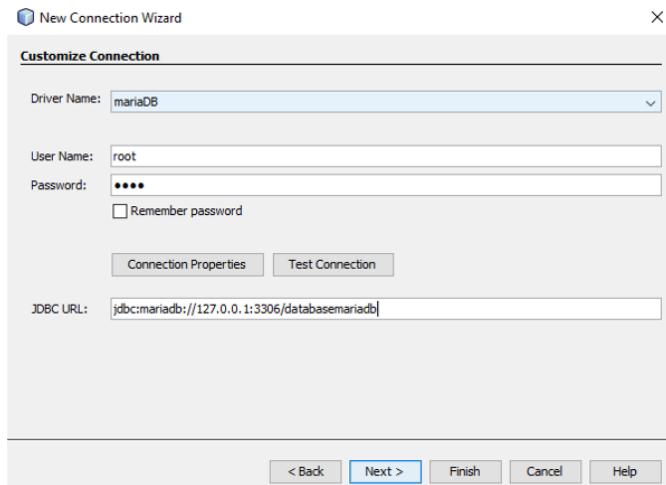


Abbildung 4.3: Connection erstellen - Screenshot Antonia Kriegner

#### 4.1.5 Alternativen

##### MySQL

MySQL ist ein relationales Datenbankverwaltungssystem. Dabei besteht dieses System aus Tabellen. Die MySQL Datenbank wird meistens bei Webservices und dort oft in Verbindung mit dem Apache Server eingesetzt. Ein Vorteil an dieser Datenbank ist, dass es ein Open-Source-Produkt ist. [69] [60]



Abbildung 4.4: MySQL Logo

##### Oracle Database

Die Database von Oracle ist eine Datenbankmanagementsystem-Software. Dabei werden relationale Daten und objektrelationale Daten gespeichert. Die Oracle Database ist weit verbreitet und beim Thema Sicherheit sehr fortgeschritten. Ein Nachteil an der Oracle Database ist, dass sie kostenpflichtig ist. [69] [66]



Abbildung 4.5: Oracle Logo

### MongoDB

Die MongoDB ist in der Sprache C++ geschrieben und basiert auf einer dokumentenorientierten Struktur. Die Daten werden in einem JSON-Format gespeichert, dies bedeutet, dass bei diesem Prinzip keine Tabellen verwendet werden. Zudem ist die MongoDB eine Open-Source Datenbank, womit eine gewisse Qualität gesichert ist.[69] [59]



Abbildung 4.6: MongoDB Logo

## 4.2 Datenbankstruktur

### 4.2.1 Entitäten (MP)

#### 4.2.1.1 Allgemein

Den Typ des Datums haben wir immer als „String“ gewählt, da es für die weitere Verarbeitung um einiges leichter war und wir den Typ „date“ hier nicht benötigen. Außerdem ist ein „String“ leichter zum Darstellen.

#### 4.2.1.2 Beleg:

Die Entität „Beleg“ beinhaltet alle Attribute, die ein Beleg benötigt. Sie besteht aus einer automatisch generierten id (int), aus einem Zweck (String), welcher den Titel des Inserates angibt, aus einem Betrag (int), das sind die Talente, aus einem Datum (String), welches den Tag des Tausches angibt, aus der Art (String) des Beleges und ob der Beleg schon akzeptiert worden ist oder nicht.

#### **4.2.1.3 Benutzer:**

Die Entität „Benutzer“ beinhaltet alle Attribute, welche für einen Benutzer nötig sind. „Benutzer“ besteht aus einer generierten id (int) und den jeweiligen „String“ wie dem Vornamen und Nachnamen, der E-Mail, der Telefonnummer, der Adresse und der Beschreibung, die ein Benutzer individuell über sich verfassen kann, und einem Passwort. Außer der id sind alle Attribute vom Typ „String“. Weiters gibt es in der Entität Referenzen auf Inserat, Nachricht, Beleg, Benutzer und Konto.

#### **4.2.1.4 Inserat:**

„Inserat“ besteht aus folgenden Attributen: einer generierten id, einem Titel, einer Art, einem Typ, einer Beschreibung, einem Start- und Enddatum, welche alle vom Typ String sind. Außerdem gibt es ein String Array Mediapfad, welches die Bildpfade beinhaltet. Weiteres gibt es noch einen Boolean, ob das Inserat auf sichtbar gestellt wurde oder nicht. „Inserat“ hat Referenzen zu Kategorie, Nachricht und Beleg.

#### **4.2.1.5 Kategorie:**

„Kategorie“ besteht aus einer generierten id (int) und einem Kategorienamen (String).

#### **4.2.1.6 Konto:**

„Konto“ besteht aus einer generierten id (int) und einem Kontostand (int). Dieser Kontostand sind Talente (die virtuelle Währung).

#### **4.2.1.7 Nachricht:**

„Nachricht“ besteht aus einer generierten id (int), den Strings Type, Titel, Text und Datum. Außerdem gibt es noch den Boolean checkMessage, welcher angibt, ob die Nachricht schon gelesen wurde oder nicht.

#### **4.2.1.8 Registrierung:**

Die Entität „Registrierung“ hat wieder eine automatisch generierte id und folgende Strings: Nachname, Vorname, E-Mail, Telefonnummer, Hausnummer, PLZ, Straße, Ort, Beschreibung und Passwort.

Die Entitäten beinhalteten jeweils noch die benötigten Setter, Getter und Konstruktoren.

#### 4.2.2 Klassendiagramm (CLD) - (AK)

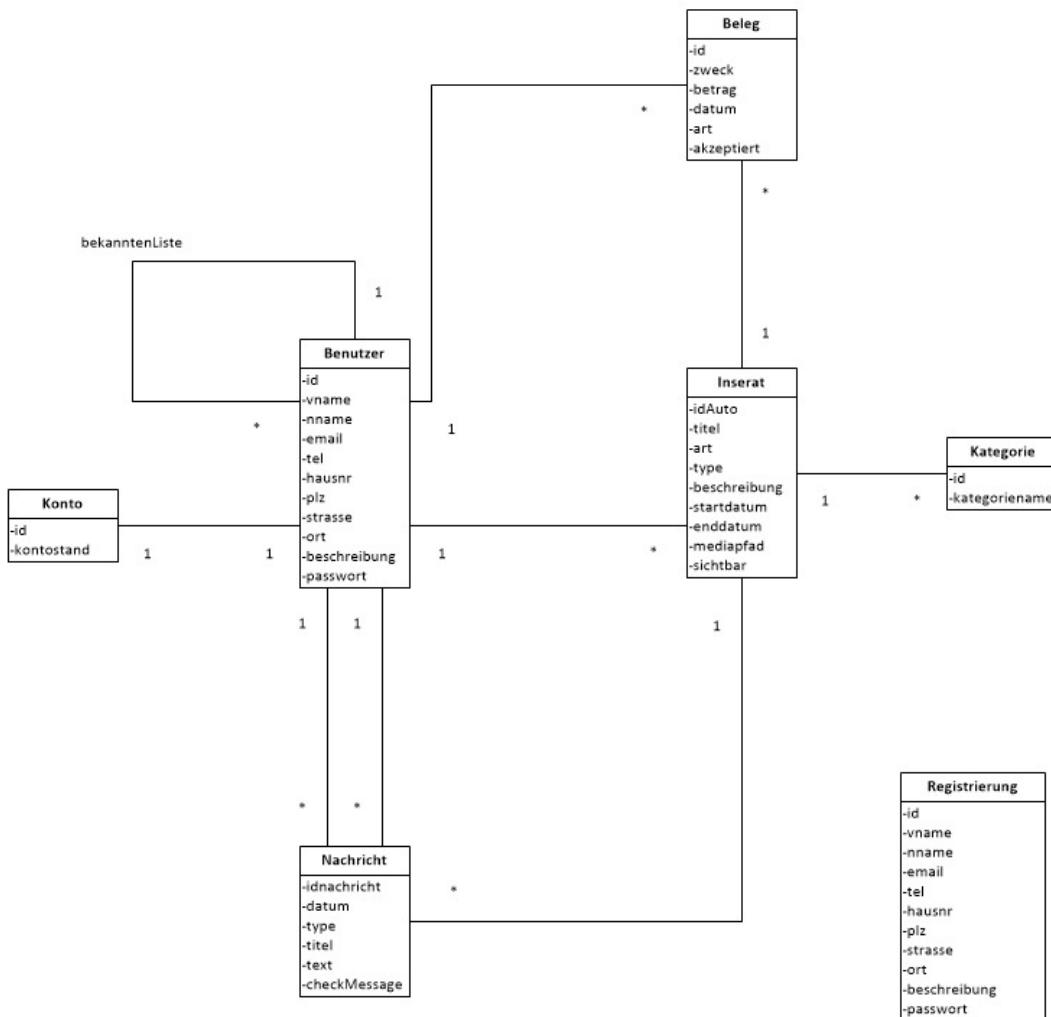


Abbildung 4.7: Klassendiagramm - Screenshot Antonia Kriegner

#### 4.2.3 Beziehungen (AK)

Ein Benutzer hat eine Beziehung zum Inserat, das bedeutet, dass ein Benutzer mehrere Inserate erstellen kann. Ein Inserat hat eine Menge Daten und wiederrum eine Beziehung zur Kategorie. Am Anfang war die Kategorie eine Spalte im Inserat, aber da diese Version nicht so gut wartbar war und mit dieser Lösung die Kategorien nicht erweiterbar waren, wurde die Spalte auf eine eigene Tabelle ausgelagert, worauf das Inserat eine Referenz hat.

Zu einem Inserat gehört ein Beleg, der vom Benutzer, der das Inserat erstellt hat, gemacht wird. Es wurde eine eins zu Stern – Beziehung gewählt, da die Inserate in Produkte und Dienstleistungen unterteilt sind und eine Dienstleistung auch mehrmals ausgeführt werden kann. Zugleich braucht der Beleg noch eine zusätzliche Verbindung zu einem Benutzer, welcher der Partner der Aktion ist. Dieser Benutzer kann auch mehrere Belege haben.

Um Eingabefehler beim Erstellen eines Beleges zu vermeiden, hat jeder Benutzer eine Bekanntenliste, in die jemand beim ersten Kontakt hinzugefügt wird. Der Nutzer kann aus dieser Liste auswählen. Ein Benutzer hat zusätzlich noch ein Konto, wo nach dem Bestätigen des Beleges die Talente hin und her gebucht werden. Hier ist eine 1 zu 1 Beziehung vorhanden, da ein Benutzer nur ein Konto haben kann und umgekehrt ein Konto nur zu einem Benutzer gehören kann.

Zur Verständigung gibt es in der Applikation ein Nachrichtenformular. Auf den ersten Blick sieht die zweifache Verbindung zwischen Benutzer und Nachricht komisch aus, doch dies war unbedingt notwendig. Für den ersten Schritt im Kommunikationsverkehr würde eine einfache Verbindung reichen, da der Benutzer, der die Nachricht verschickt, der Sender ist und der Ersteller des Inserates der Empfänger ist. In unserer Anwendung ist es aber kein monotoner Nachrichtenverkehr, das heißt, man kann auf die Nachricht direkt antworten. Bei einer einfachen Verbindung tritt das Problem auf, dass nicht mehr unterschieden werden kann, ob die Nachricht jetzt empfangen oder verschickt wurde und darum braucht man vom Benutzer zwei Beziehungen zur Nachricht.

### 4.3 Aufbau der Projektstruktur (AK)

Unser Projekt ist so aufgebaut, dass wir unsere Klassen in fünf Packages aufteilen. Das erste Package ist der Controller, in dem sich zwei Services befinden, einmal das für die normalen Abfragen und dann noch das für den Bilderupload. Weiteres befindet sich darin dann auch noch die Main und eine Enum-Klasse. Das zweite Package ist für die Entities. Hier sind die Klassen Beleg, Benutzer, Inserat, Nachricht, Kategorie, Konto und Registrierung enthalten. Also all jene, die anschließend eine Tabelle in der Datenbank darstellen. Das dritte beinhaltet den Filter, der dafür zuständig ist, dass die Kommunikation zwischen Server und Client funktioniert und keinen Http-Response Fehler wirft. Diese Klasse kann man auch umgehen, indem man im Browser das Corse Plugin installiert.

Das vierte Package beinhaltet die Klasse "EMailVersendenUtil.java". Diese Klasse wird bei der Registrierung aufgerufen und dann wird eine E-Mail an den Nutzer geschickt, der diese bestätigen muss.

Das fünfte Package ist das Repository. Dieses beinhaltet Facade Klassen und zusätzlich noch eine Abstract Facade. Die Facade Klassen sind von der Abstract Facade abgeleitet und können dann alle Standard Methoden der Abstract Facade nutzen. Dazu gehören die Methoden:

- `find()`, die ein bestimmtes Objekt mit dem Primary Key sucht
- `findAll()`, die den Inhalt einer ganzen Tabelle ausgibt
- `save()`, die einen neuen Eintrag speichert
- `remove()`, die einen Eintrag entfernt

Um all diese für jede Entity nutzen zu können, müssen diese Methoden in der Abstract Facade typisiert sein. Für die Typisierung gibt man beim Namen der Klasse in spitzen Klammern ein T an und bei den Methoden verwendet man ebenfalls ein T.

---

```
public abstract class AbstractFacade<T> {

    public void save(T entity){
        em.getTransaction().begin();
        em.persist(entity);
        em.getTransaction().commit();
    }
}
```

---

Zudem wird in der AbstractFacade ein Konstruktor erstellt und darin dann die entity-Class gesetzt:

---

```
public AbstractFacade(Class<T> entityClass) {
    this.entityClass = entityClass;
}
```

---

Die abgeleiteten Klassen werden mit `extends AbstractFacade < Entity >` gekennzeichnet und anschließend wird im Konstruktor der Parentkonstruktor aufgerufen und die Entity.class mit übergeben, die dann in der AbstractFacade gesetzt wird.

---

```
public class KategorieFacade extends AbstractFacade<Kategorie>{

    public KategorieFacade() {
        super(Kategorie.class);
    }
}
```

---

## 4.4 Lazy Loading (AK)

Lazy Loading bedeutet übersetzt „faules Laden“ und hilft dabei, nicht alles aus der Datenbank zu laden. In unserer Anwendung gibt es einen Benutzer und dieser referenziert auf eine Liste von Inseraten, eine Liste von Nachrichten und eine Liste von Belegen. Das Problem dabei ist, dass wenn wir den Benutzer aus der Datenbank laden, werden alle Listen mitgeladen und das JSON-Array wird enorm groß. Um dies zu vermeiden, kann bei der Beziehung der FetchType.LAZY angegeben werden:

---

```
@OneToMany(fetch=FetchType.LAZY)
```

---

Der FetchType.Lazy lädt die Listen nur, wenn zum Beispiel mit einem Getter darauf zugegriffen wird. Durch diesen Zugriff wird ein Aufruf auf die Datenbank ausgelöst, um die Elemente abzurufen. Diese Methode mit dem Getter ist aber nur bei kleinen Listen von Objekten sinnvoll, da es sonst zu Performance Problemen kommen kann, da durch aufrufen des Getters jeweils eine eigene SQL-Anweisung ausgeführt wird.

In unserem Beispiel wollten wir nur den Benutzer laden und haben deshalb den FetchType.LAZY verwendet, um das JSON-Array so klein wie möglich zu gestalten und nicht gebrauchte Entities zu „entfernen“. Dadurch wird die Anwendung zeit- und ressourcen-sparender und das Laden des Requests, bei dem ein Benutzer geladen wird, schneller.

Neben FetchType.Lazy gibt es auch noch FetchType.Eager. Bei diesem Typ werden alle Listen sofort geladen, wenn der „Parent“ aufgerufen wird. In unserem Beispiel würde es bedeuten, dass, wenn der Benutzer abgerufen wird, alle Listen sofort geladen werden.

[33][29][51]

## 4.5 Security (MP)

In diesem Kapitel werden die Sicherheitsmaßnamen, welche wir getroffen haben, um die Sicherheit der Daten der Benutzer zu wahren, erklärt und beschrieben.

### 4.5.1 Worauf muss geachtet werden?

Generell muss darauf geachtet werden, dass das Passwort, welches der Benutzer bei der Registrierung angibt, nicht im Klartext in die Datenbank gespeichert wird, sondern, dass es verschlüsselt gespeichert wird. Das haben wir mit der Hash Funktion MD5 implementiert. Ein weiterer wichtiger Punkt war, dass sich eine Person nicht mit einer falschen E-Mail Adresse, bzw. mit einer E-Mail Adresse von einer anderen Person registrieren kann. Das haben wir so gelöst, indem nach jeder Registrierung der Benutzer eine Bestätigungs-E-Mail gesendet bekommt.

In dieser befindet sich ein Bestätigungslink, welcher angeklickt werden muss. Wenn der Link nicht angeklickt wird, wird der Benutzer noch nicht als Benutzer abgespeichert, sondern wird in der Tabelle Registrierung zwischen gespeichert.

#### 4.5.2 Message-Digest Algorithm (MD5)- Allgemein

MD5 ist, wie schon erwähnt, eine Hashfunktion, welche Nachrichten, Passwörter und generell Texte verschlüsseln kann. Zuerst wird ein Hashwert ermittelt, dieser ist 128 Bit lang. Dieser Hashwert ist für jeden Text, der verschlüsselt wird, anders. [86] [55]

Wenn zum Beispiel das Wort „Diplomarbeit“ mit einem großen D beginnt, sieht die Zeichenkette, welche wir durch MD5 heraus bekommen, folgendermaßen aus: „1c31a944c8449e5bbdd890f863d8dbff“. Wenn das Wort mit einem kleinen d beginnt, also „diplomarbeit“, dann sieht die Zeichenkette so aus: „a4aac7f8e2c0d16766cffac18b7f62a3“. Wie man sieht, handelt es sich um fast das selbe Wort, nur mit einem anderen Anfangsbuchstaben und es kommt trotzdem eine ganz andere Zeichenkette heraus. Ein wesentlicher Vorteil von MD5 ist, dass man diese Zeichenkette nicht wieder in einen Text entschlüsseln kann. Darauf werden wir im nächsten Kapitel genauer eingehen. [86] [55]



Abbildung 4.8: MD5 Beispiel

#### 4.5.3 Vorteile

Wie schon erwähnt ist ein wesentlicher Vorteil von MD5, dass man die Zeichenkette nicht wieder zu einem Text zurück verschlüsseln kann. Der Grund dafür ist, dass ein Hashwert nicht eindeutig ist. Demnach können zwei verschiedene Nachrichten die selbe Zeichenkette haben. Das ist allerdings sehr unwahrscheinlich. Gerade bei Passwörtern kommt das so gut wie nie vor. [86]

#### 4.5.4 Nachteile

Der Nachteil an MD5 ist, dass man theoretisch ausprobieren kann, viele Passwörter in einen Hash Wert zu verschlüsseln und diese dann mit dem eingegebenen Passwort zu vergleichen. Dadurch könnte man eventuell heraus finden, welches Passwort nun verwendet wurde. Allerdings gibt es für einen Hash Wert, wie schon erwähnt, mehrere Möglichkeiten. Um das zu verhindern, muss in unserem Projekt das Passwort, welches bei der Registrierung angegeben wird, aus mindestens zehn Zeichen, Groß- und Kleinbuchstaben und zwei Zahlen bestehen. Denn es ist viel wahrscheinlicher, dass das Passwort durch eine solche Überprüfung gefunden wird, wenn es ein normales Wort wie zum Beispiel Haus oder Hund ist. So verringern wir die Wahrscheinlichkeit, dass das Passwort geknackt werden könnte. [86]

#### 4.5.5 Wie haben wir MD5 in unserem Projekt eingesetzt?

Für dieses Unterkapitel haben wir folgende Referenzen verwendet: [58] [40] [75] [79]

Um MD5 in unserem Projekt zu verwenden, haben wir in der BenutzerFacade die Methode "verschlüsseln()" erstellt:

---

```
public String verschluesseln(String str) {
    String passwordToHash = str + "platzhalter";
    String generatedPassword = null;

    try {
        MessageDigest md = MessageDigest.getInstance("MD5");

        md.update(passwordToHash.getBytes());

        byte[] bytes = md.digest();

        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < bytes.length; i++) {
            sb.append(Integer.toString((bytes[i] & 0xff) + 0x100,
                16).substring(1));
        }

        generatedPassword = sb.toString();
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }

    return generatedPassword;
}
```

---

In dieser Methode wird das Passwort im String str mitübergeben. In passwordToHash wird das Passwort mit einem weiteren String zusammen gehängt, um das Risiko, das Passwort heraus zu finden, zu verringern. Der String generatedPassword wird mit null initialisiert, dieser wird später die generierte Zeichenkette beinhalten. Danach wird in einem try - catch - Block eine Instanz der MessageDigest Klasse für MD5 erstellt. Dann werden die Bytes des Passwortes auf diese Instanz gespeichert. Danach werden die Bytes der Zeichenkette ermittelt. Da die Bytes der Zeichenkette in einem Dezimalformat abgespeichert wurden, müssen sie jetzt in ein Hexadezimalformat konvertiert werden. Nun wird die gesamte Zeichenkette auf generatedPassword gespeichert. Der return Wert ist das generatedPassword.

#### 4.5.6 MessageDigest

Die MessageDigest Klasse ist dafür da, um Texte mithilfe von MD5, SHA-1 und SHA-256 in Zeichenketten umzuwandeln. Sie stellt verschiedene Methoden zur Verfügung. Zu Beginn der Verwendung dieser Klasse wird eine Instanz davon generiert. Danach wird die update() Methode angewendet. Bei der update() Methode werden die Bytes für die jeweilige Hash Funktion benötigt. Es gibt außerdem die Methode reset(), welche zu jedem beliebigen Zeitpunkt die Instanz wieder zurücksetzen kann. Eine weitere Methode, welche die Klasse bereitstellt, ist die digest() Methode. Diese wird zum Schluss aufgerufen und schließt die Berechnung der Zeichenkette ab. [58]

#### 4.5.7 Andere Hash-Funktionen

Neben MD5 gibt es auch noch andere Hash Funktionen, wie zum Beispiel SHA-1 und SHA-256. Diese werden wir uns nun etwas genauer anschauen und außerdem erörtern, wieso wir diese nicht verwendet haben. Generell ist zu empfehlen immer auf die Hashfunktionen umzusteigen, welche aktuell dem Stand der Sicherheitstechnik entsprechen. Daher wird für den Livebetrieb diese Thematik noch weiter überprüft.

##### 4.5.7.1 Secure Hash Algorithm 1 (SHA-1)

Für dieses Unterkapitel wurden folgende Referenzen verwendet: [72] [103] [18] [106]

SHA-1 wird zum Beispiel bei digitalen Zertifikaten, bei digitalen Signaturen, als auch bei der Verschlüsselung von Passwörtern angewendet.

Der erste wesentliche Unterschied von SHA-1 gegenüber MD5 ist, dass der Hash Wert länger ist. Der Hash Wert von SHA-1 hat eine Länge von 160 Bits und der von MD5 hat eine Länge von 128 Bits.

Ein paar Jahre nach der Veröffentlichung des SHA-1 wurde bekannt, dass dieser einigermaßen leicht zu berechnen ist, vor allem mit Hochleistungsrechnern. Da die Geschwindigkeit von Computern immer schneller wird, wird es immer einfacher, die Zeichenketten zu berechnen.

Name	Hashgrösse	Jahr
MD2	128	1989
MD4	128	1990
MD5	128	1992
MD6 <sup>5</sup>	variabel < 512	2008
SHA(0)	160	1992 - durch SHA1 ersetzt
SHA1	160	1995
SHA224 (SHA2)	224	2004
SHA256 (SHA2)	256	2001
SHA384 (SHA2)	384	2001
SHA512 (SHA2)	512	2001
SHA3	?	Wettbewerb läuft

Abbildung 4.9: Übersicht gängige Hash Funktionen

Eine weitere Schwäche dieser Hash Funktion ist, dass zwei PDF's den selben Hash Wert haben können. Dies haben Forscher von der CWI Amsterdam (das nationale niederländische Forschungsinstitut für Mathematik und Informatik) und Google belegt.

Da SHA-1 zu große Schwächen aufweist, wurden weitere SHA Algorithmen entwickelt. Einer davon ist SHA-256, welchen wir uns jetzt auch noch etwas genauer anschauen werden.

#### 4.5.7.2 Secure Hash Algorithm 256 (SHA-256)

Die Hashfunktion SHA-256 fällt unter den Begriff SHA-2. Es ist der Nachfolger von SHA-1 und auch sicherer. Der Hash-Wert der SHA-256 Funktion hat eine Länge von 256 Bits. SHA-2 beinhaltet neben SHA-256 noch folgende Funktionen: SHA-224, SHA-384, SHA-512, SHA-512/224 und SHA-512/256. Der Unterschied zwischen den einzelnen Funktionen ist die Ausgabenlänge. Da SHA-2 einer der neuesten Algorithmen ist, wurden noch keine erfolgreichen Angriffe getätigt. [73] [82]

Der gleiche Text wird mit unterschiedlichen Hash Funktionen verschlüsselt:

MD5	Heute regnet es.	099e2770a846915ac3ee5fdec4124816
SHA-1	Heute regnet es.	735fb5e64a6d7a718bd4e244637d2e044ab8ce93
SHA-256	Heute regnet es.	A40F5C8A39EDBF368E5AB7E21C2C36BC7F5346AD0 BBC16CF24ED34AB3D38ADDD

Tabelle 4.1: Gleicher Text mit unterschiedlichen Hash Funktionen verschlüsselt

[55] [74] [76]

#### 4.5.8 E-Mail senden

Für das folgende Unterkapitel wurden folgende Referenzen verwendet: [32] [26] [41]

Ein weiterer Weg, die Sicherheit der Tauschbörse zu gewährleisten, ist das Senden einer E-Mail mit einem Bestätigungslink bei der Registrierung. Das heißt, jeder Benutzer, der sich neu im System registrieren möchte, muss im Registrierungsformular seine E-Mail Adresse angeben. An diese Adresse wird dann automatisch eine Mail geschickt. Diese Mail beinhaltet einen Bestätigungslink, auf welchen der Benutzer klicken muss, um im System als Benutzer gespeichert zu werden.

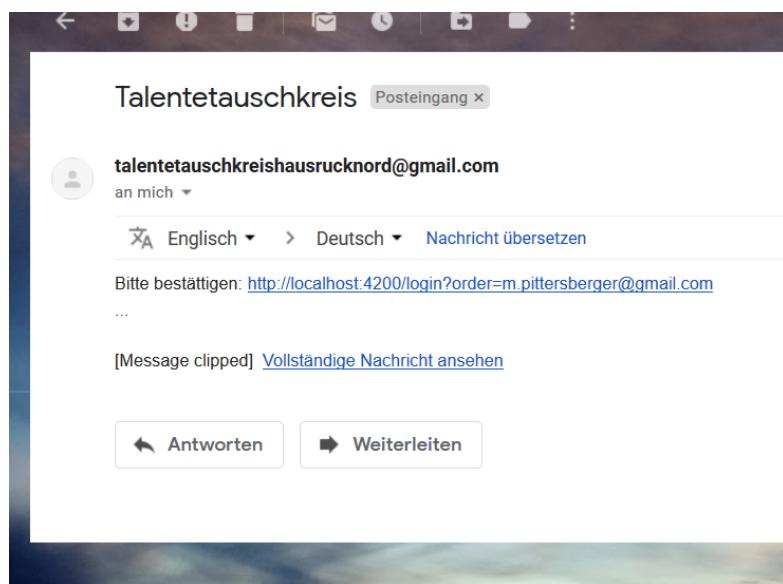


Abbildung 4.10: Bestätigungsmaill - Screenshot Melanie Pittersberger

Der Vorteil davon ist, dass sich somit niemand mit einer E-Mail Adresse von einer anderen Person registrieren kann, beziehungsweise kann man sich so auch nicht mit einer Fake Adresse anmelden.

Solange der Benutzer noch nicht auf den Link geklickt hat, ist er in der Datenbank noch in der Tabelle Registrierung zwischengespeichert. Sobald er auf den Link klickt, wird er aus der Registrierungs-Tabelle gelöscht und in die Benutzer-Tabelle gespeichert.

Für das Versenden der E-Mail haben wir ein neues Package mit dem Namen "mail" angelegt. Darin befindet sich die Klasse "EMailVersendenUtil". Diese beinhaltet die Methode "sendEmail", die für das Versenden zuständig ist.

---

```
public static void sendEmail(String emailMessage, String email) {

    try {

        final String fromEmail = "talentetauschkreishausrucknord@gmail.com";
        final String password = "passwort";
        final String toEmail = "email";

        System.out.println("TLSEmail Start");
        Properties props = new Properties();
        props.put("mail.smtp.host", "smtp.gmail.com");
        props.put("mail.smtp.port", "587");
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.starttls.enable", "true");

        Authenticator auth = new Authenticator() {

            @Override
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication(fromEmail, password);
            }
        };
        Session session = Session.getInstance(props, auth);

        MimeMessage message = new MimeMessage(session);
        message.setFrom(new InternetAddress(fromEmail));
        message.addRecipient(Message.RecipientType.TO, new
            InternetAddress(toEmail));

        message.setSubject("Talentetauschkreis");
        message.setText(emailMessage);

        Transport.send(message);

    } catch (Exception ex) {
        System.out.println("Mail fail");
        System.out.println(ex);
    }
}
```

---

Im Methodenkopf der sendEmail - Methode wird die Nachricht, welche die Mail beinhaltten soll, und die E-Mail Adresse, an die die Nachricht geschickt werden soll, mit übergeben. Der String fromEmail wird mit der E-Mail Adresse des Talentetauschkreises Hausruck Nord initialisiert. Diese Adresse haben wir im vorhinein erstellt, von dort aus werden dann die automatischen Mails versendet. Damit das funktioniert, muss es eine Gmail Adresse sein.

Beim Erstellen des E-Mail Kontos haben wir außerdem ein Passwort vergeben müssen, welches im String password gespeichert werden muss. Im String toEmail wird die Adresse gespeichert, mit der sich der Benutzer registriert. Das kann jede beliebige Adresse sein, das heißt, es muss kein Gmail Konto sein.

Danach werden essentielle Daten wie der SMTP Host, der TLS Port und ob die Authentication richtig oder falsch ist, gespeichert. Nun müssen wir ein Authenticator Objekt erstellen, sodass wir es der Session mitgeben können. Anschließend wird ein Objekt message von MimeMessage erstellt, welchem die Session mitübergeben wird. Auf message wird die fromEmail und die toEmail gespeichert. Außerdem wird der Betreff der E-Mail und der Text gesetzt. Zum Schluss wird die message abgesendet.

Falls das Ganze nicht funktioniert hat, wird auf der Konsole „Mail fail“ ausgegeben.

Um die E-Mail versenden zu können, mussten wir bei dem Google Konto, also talenteauschkreishausrucknord@gmail.com, folgende Einstellung treffen:



Abbildung 4.11: E-Mail zulassen - Screenshot Melanie Pittersberger

Man muss unter diesem Link

---

<https://myaccount.google.com/lesssecureapps>

---

die weniger sicheren Apps zulassen. [88]

# Kapitel 5

## Client

### 5.1 Navigationsstruktur (MP)

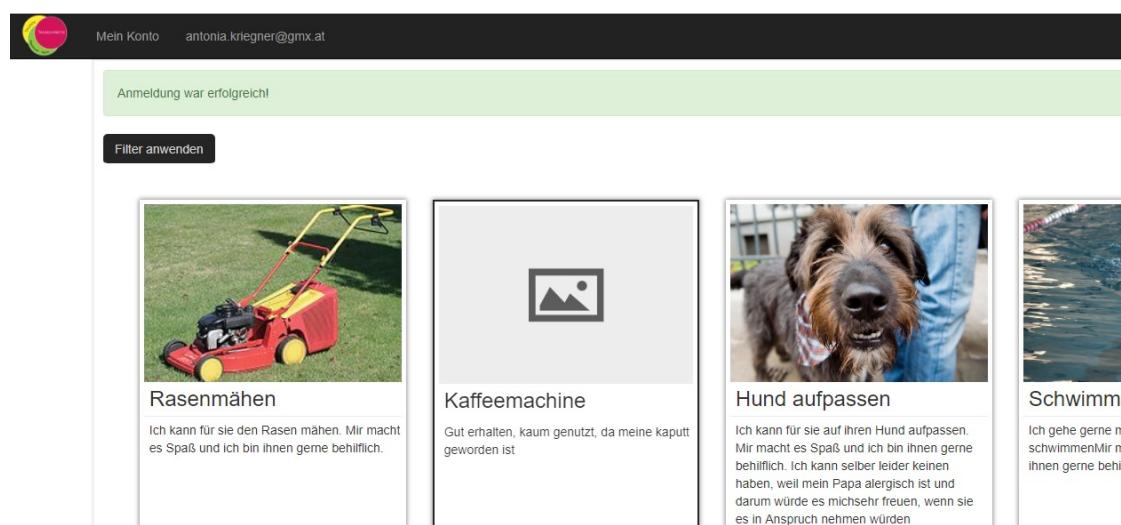


Abbildung 5.1: Navigationsstruktur (MP) - Screenshot Melanie Pittersberger

In der Abbildung 5.1 ist die Navigationsstruktur zu sehen. Dieses Menü besteht aus einem Logo und dem Button "Mein Konto". Wenn man mit der Maus über die einzelnen Menüpunkte fährt, dann wird die Schrift weiß, das ist ein sogenannter "Hover Effekt".

```
[ngClass]="{ 'in': isIn }"
<ul class="nav navbar-nav">
  <li><a [routerLink]="'[naviagtion']" (click)="toggleState()">Mein Konto</a></li>
  <li><a (click)="logout()" data-toggle="tooltip" title="{{act.email}}>{{act.email}}</a></li>
</ul>
```

Abbildung 5.2: Navigationsstruktur Code (MP) - Screenshot Melanie Pittersberger

---

```
<ul>
```

---

Der ul-Tag ist ein HTML Tag, welcher eine ungeordnete Liste definiert. Im Gegensatz dazu gibt es den ol-Tag, welcher geordnete Listen definiert. Wie jedem anderen HTML-Tag kann man ul natürlich auch ein Styling mithilfe von Cascade Style Sheets (CSS) geben.

Wie man in Abbildung 5.2 erkennen kann, haben wir die Navigationsstruktur mit dem ul-Tag begonnen. Wir haben diesen Tag gewählt, da eine Menüleiste keine geordnete Liste ist und wir daher den ol-Tag nicht brauchen. Dieser Tag muss, nachdem die ganzen Elemente der Liste aufgelistet werden, geschlossen werden.

---

```
<li>
```

---

Der li-Tag definiert ein neues Listenelement. Diesen haben wir jeweils für einen neuen Menübutton benutzt. Das heißt, er wurde beim Button "Mein Konto" eingesetzt. [37]

---

```
<li><a [routerLink]=["'home'"] (click)="toggleState()">Infos</a></li>
```

---

Der li-Tag wird, nachdem die Seiten verlinkt werden und der Name des Menüpunktes geschrieben wurde, wieder geschlossen.

---

```
<a>
```

---

Der a-Tag ist für Verlinkungen zuständig. Durch diesen Tag kann man auf andere Seiten verlinken. [36]

---

```
<a [routerLink]=["'login'"] (click)="toggleState()">
```

---

Zuerst wird auf die jeweilige Seite verlinkt, wie hier auf die Login-Seite, auf welcher sich der Benutzer einloggen kann. Die Methode toggleState() ist für die Ansicht am Handy zuständig.

## 5.2 Cards/Panel (AK)

Cards sind Container, die sich in Header, Body und Footer unterteilen lassen. Es gibt verschiedene Anbieter, zum Beispiel Bootstrap oder Foundation. Die Bootstrapcards sind eine Ablöse der vorherigen Panels. Die Breite der Cards kann mithilfe von CSS über bestimmte Einstellungen geändert werden. Eine wichtige Funktion, die Bootstrapcards ebenfalls bietet, ist die Einbindung von Bildern. Ein Bild kann oben und unten platziert werden und ebenfalls in den Hintergrund gesetzt werden. Dieser Effekt nennt sich Overlay. Eine weitere Option, welche Bootstrapcards zur Verfügung stellt, ist das Zusammenfassen von Cards zu einer Gruppe beziehungsweise zu Spalten.

Die typische Darstellung einer Card sieht wie folgt aus: [16]

---

```
<div class="card">
  <div class="card-header">
    Header
  </div>
  <div class="card-body">
    Body
  </div>
  <div class="card-footer">
    Footer
  </div>
</div>
```

---

Die Darstellung von Panels: [15]

---

```
<div class="panel panel-default">
  <div class="panel-heading">Panel Heading</div>
  <div class="panel-body">Panel Content</div>
  <div class="panel-footer">Panel Footer </div>
</div>
```

---

Bei der Darstellung der eigenen Inserate haben wir noch das veraltete Format der Panels verwendet, da es hier eine horizontale Version gibt. Dabei haben wir im Header das Bild, in der Mitte den Text und im Footer zwei Buttons eingefügt. Der eine Button verlinkt zum „Beleg ausstellen“ und der zweite setzt die Sichtbarkeit des Inserates. Beim Text gibt es zusätzlich noch eine Verlinkung, damit der Eintrag bearbeitet werden kann.

Referenzen für Cards und Panels: [100] [16] [15]

Abbildung 5.3: Ansicht der eigenen Inserate - Screenshot Antonia Kriegner

### 5.3 Slideshow - Plugin (AK)

Wir haben das Ngx-gallery Plugin für unsere Slideshow gewählt. Dabei gilt der angegebene Link als Referenz für den folgenden Inhalt: [65]

#### 5.3.1 Ngx-gallery

Um die Ngx-gallery zu verwenden, muss im ersten Schritt der Befehl ausgeführt werden:

---

```
npm i ngx-gallery
```

---

Anschließend muss in der HTML das Element <ngx-gallery></ngx-gallery> eingebunden werden. Diesem Element kann ein Array mit Bildern zugewiesen und die Optionen gesetzt werden.

---

```
<ngx-gallery [options]="galleryOptions" [images]="galleryImages"></ngx-gallery>
```

---

Durch einen Breakpoint wird gesetzt, ab welcher Fensterbreite sich die Größe der Slideshow ändern soll. In dem angeführten Beispiel beträgt die maximale Fensterbreite 800px. Wenn das Fenster breiter als diese Pixelanzahl wird, dann wird auf die Einstellungen der ersten geschwungenen Klammern gewechselt. Der Breakpoint gibt sozusagen die max-width an. Zusätzlich ist in den ersten beiden Klammern die Art der thumbnailColumns angegeben und die Art der Animation. Die Anzahl der Spalten gibt an, wie viele Bilder in der Vorschau sichtbar sein sollten.

---

```
this.galleryOptions = [
  {
    width: '100%',
    height: '500px',
    thumbnailsColumns: 4,
    imageAnimation: NgxGalleryAnimation.Slide
  },
  {
    breakpoint: 800,
    width: '500px',
    height: '200px',
    imagePercent: 80,
    thumbnailsPercent: 20,
    thumbnailsMargin: 20,
    thumbnailMargin: 20
  }
]
```

---

Im nächsten Schritt wird ein Array mit Bildpfaden dynamisch gefüllt. Dazu gehen wir das Array an Bildpfaden durch und erstellen pro Bild ein JSONObject, welches drei Key-Value Paare hat, und zwar für die Ansicht in klein, mittel und groß. Diese Objekte fügen wir anschließend in das Array ein, das später dem Bildarray für die Slideshow entspricht.

---

```
for (let i in this.dataFromParent.mediapfad) {
  this.jsonObjArray.push({
    small: this.dataFromParent.mediapfad[i],
    medium: this.dataFromParent.mediapfad[i],
    big: this.dataFromParent.mediapfad[i]
  });
}
```

---

### 5.3.2 Alternativen zur Ngx – gallery

Eine weitere Option für eine andere Slideshow wäre das Bootstrap Carousel gewesen. Bei dieser Slideshow wäre allerdings das dynamische Befüllen sehr umständlich gewesen, da hier die Slideshow im HTML Code eingebunden wird und die erste Slide einen anderen class - Typ hat, als der Rest. Die Slideshow hätte man nur durch das Einsetzen von class Properties verwenden können. Dabei hätte aber die Performance gelitten, da jeweils vor der Darstellung überprüft werden müsste, ob es sich um die erste Slide handelt oder nicht, und dann erst der Typ gesetzt wird. [101]

Bootstrap Carousel - Code: [101]

---

```
<div id="carouselExampleSlidesOnly" class="carousel slide"
    data-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
</div>
```

---

Weiters hätte es auch die Möglichkeit gegeben, den ng2-Image-Viewer zu verwenden. Dieser ist aber leider in seiner Darstellung sehr dominant und kaum durch CSS-Formatierungen zu beeinflussen. Leider war auch die Abbildung von Bildern nicht sehr ansprechend. Dies führte dazu, dass wir uns gegen diesen Slider entschieden haben. [64]

Eine weitere optisch sehr ansprechende Image View ist die Angular 2 Image Gallery. Das Problem an dieser View war die Installation. Das Herunterladen der Gallery hat einwandfrei funktioniert und es sind keine Fehler aufgetaucht. Auch das Einbinden der Slide in die app-modules.ts war erfolgreich. Aber der Import von „hammerjs“ wurde in unserem Programm nicht erkannt. Auch Deinstallieren und erneutes Installieren haben keine Wirkung gezeigt. [99]

## 5.4 Mittel für die responsive Darstellung (AK)

### 5.4.1 Bootstrap

Indem man mit der Bootstrap Tabelle arbeitet, kann die responsive Darstellung erreicht werden. Bootstrap verwendet Container, Reihen und Spalten, um das Layout darzustellen. Eine Reihe einer Tabelle besteht aus 12 Spalten, diese Anzahl darf nicht überschritten werden, sonst verschiebt sich die Spalte in die nächste Reihe.

In diesem Beispiel werden drei gleich breite Spalten generiert:

---

```
<div class="container">
  <div class="row">
    <div class="col">
      1. Spalte
    </div>
    <div class="col">
      2. Spalte
    </div>
    <div class="col">
      3. Spalte
    </div>
  </div>
</div>
```

---

Tabellen Optionen:

Extra small	<576px	.col-
Small	$\geq 576\text{px}$	.col-sm-
Medium	$\geq 768\text{px}$	.col-md-
Large	$\geq 992\text{px}$	.col-lg-
Extra large	$\geq 1200\text{px}$	.col-xl-

Tabelle 5.1: Bootstrap - Tabellen Optionen

Alle Inhalte wurden dieser Quelle entnommen: [102]

### 5.4.2 Media queries

Media queries bestehen aus einem Medientyp, welcher angibt, welches Format diese Regel befolgen muss, und einem Ausdruck, der sich auf Größen wie zum Beispiel Höhe und Breite bezieht. Wenn der Ausdruck erfüllt ist (true), dann wird die CSS ausgeführt. Media queries besitzen Operatoren, diese sind not, and und only. Mit „and“ können mehrere Ausdrücke miteinander verbunden werden.

Der „not“ Operator dreht eine Aussage um. Zum Beispiel, wenn bei max-width: 700px ein not eingefügt wird, dann wird die Aussage true, wenn es größer als 700px ist. Der only Operator wird dazu verwendet, dass die CSS nur angewendet wird, wenn der ganze Ausdruck true ist.

Mediatyp:

- all
- print
- screen
- speech
- braile/projection/tv

Wenn der Browser 600px hat oder kleiner ist, dann gilt die CSS innerhalb der geschwungenen Klammern:

---

```
@media only screen and (max-width: 600px){}
```

---

Wenn der Browser 600px hat oder größer ist, dann gilt die CSS innerhalb der geschwungenen Klammern:

---

```
@media only screen and (min-width: 600px){}
```

---

Neben den normalen Ausdrücken wie Höhe und Breite gibt es noch den „orientation“ Operator. Dieser kann zwei Werte haben, entweder „landscape“ oder „portrait“. Der landscape – Operator wird verwendet, wenn die Displaygröße breiter ist als höher und der portrait-Operator, wenn die Displaygröße höher ist als breiter. Diese Funktion findet Anwendung beim Drehen eines Smartphones.

---

```
@media all and (orientation: portrait) { ... }  
 @media all and (orientation: landscape) { ... }
```

---

Die Inhalte des Kapitels Media queries wurden folgenden Seiten entnommen: [70] [81][56]

## 5.5 Filter (AK)

Auf der Startseite gibt es drei Filter, die auf die Inserate angewendet werden können. Der erste Filter ist eine Suchleiste, die nur den Titel überprüft und jeweilige Übereinstimmungen liefert. Der zweite bezieht sich auf die Suche in bestimmten Kategorien. Diese Suche wurde am Server abgewickelt, dazu wird durch das Setzen einer Kategorie ein Aufruf gemacht und ein Kategorienstring mitgeschickt. Dann werden alle Inserate abgeglichen und die, die übereinstimmen, zurückgeschickt. Die Funktion am Server sieht wie folgt aus:

---

```
(getInserate()).forEach((i) -> {
    for (Kategorie k : i.getKategorieList()) {
        for (String d : matches) {
            if (k.getKategoriename().equals(d)) {
                if (!list.contains(i)) {
                    list.add(i);
                }
            }
        }
    }
});
```

---

Zuerst werden mit der Methode getInserate() alle Inserate geholt, diese werden dann nacheinander durchlaufen. Pro Inserat gibt es eine Liste von Kategorien. Darum erzeugt man eine for-Schleife, die alle Kategorien durch geht und anschließend nochmals eine for-Schleife, die alle Kategorien des Suchstrings durchwandert. Anschließend erfolgt eine Überprüfung, ob eine Übereinstimmung herrscht, und dann wird das Inserat, insofern es nicht schon enthalten ist, in die Liste eingefügt, die später an den Client geschickt wird.

Der dritte Filter inkludiert 2 Regler, die ein- und ausgeschaltet werden können. Dadurch wird die Ansicht der Inserate zwischen Dienstleistung und Produkt geregelt.

## 5.6 Nachrichten – Nachrichtenformular (AK)

Es gibt einen Menüpunkt, in welchem alle Nachrichten angezeigt werden. Durch Klicken auf den Pfeil wird auf ein Formular umgeleitet. Die Nachrichten werden nach dem Datum sortiert, das heißt, die neuesten Nachrichten werden ganz oben dargestellt. Diejenigen, die noch nicht gesichtet wurden, werden mit grünem Rand dargestellt. Dies wird mit der folgenden Zeile verwirklicht:

---

```
[style.border] = "messages[in].checkMessage ? '1px solid lightgrey':'2px solid #8CC644'
```

---

Dieser Code überprüft, ob die Nachricht schon angesehen wurde. Wenn ja, dann wird ein grauer Rand mit 1px gezeichnet, ansonsten ein grüner Rand.

Das Formular besteht aus drei Zeilen, die automatisch gefüllt werden und nicht änderbar sind. So werden Eingabefehler bezüglich Sender und Empfänger vermieden. Zusätzlich wird für die Identifizierung der Nachricht ein Titel mitabgespeichert, der durch Klicken auf ein bestimmtes Inserat gesetzt wird. Das einzige bearbeitbare Feld ist jenes, in dem der Text geschrieben werden kann. Das Nachrichtenformular ist so aufgebaut, dass es nur ein Formular gibt und die nötigen Details werden in der jeweiligen Komponente mitübergeben. Es gibt somit eine Komponente „nachrichtenformular“, die wie folgt eingebunden wird:

---

```
<app-nachrichtenformular [empf]="empf" [type]="type" [inseratId]="inseratId" [titel]="titel" [help]=false></app-nachrichtenformular>
```

---

So wird zum Beispiel in der ersten Komponente der Typ „Anfrage“ übergeben und in der anderen der Typ „Antwort“. Weiters wird noch die InseratId für die eindeutige Identifikation der Nachricht mitübergeben und weitere relevante Attribute für die Darstellung. In „nachrichtenformular.ts“ werden die mitübergebenen Werte wie folgt angesprochen:

---

```
@Input()  
private inseratId;
```

---

In der Nachrichten Komponente werden anschließend die noch fehlenden Daten für die Nachricht ermittelt und es wird ein JSON-String an den Server geschickt. Zum Setzen des Senders haben wir den eingeloggten Benutzer ermittelt. Der Empfänger wurde bei einer „Anfrage“ durch den Ersteller des Inserates ermittelt und beim „Antworten“ erhalten wir den Empfänger, indem wir den Sender der vorherigen Nachricht ermittelten. Anschließend wird die Nachricht an den Server geschickt.

Am Server wird mit den Daten eine neue Nachricht erzeugt und die Beziehungen zwischen den Benutzern gesetzt. Dabei werden die Benutzer aus der Datenbank gelesen und einer wird als Sender und der andere als Empfänger markiert. Damit dies umgesetzt werden

kann, müssen vom Benutzer ausgehend zwei Referenzen zur Nachricht bestehen. Mit JPA gab es dabei das Problem, dass zwei Listen vom gleichen Typ nicht auf den gleichen Primary Key referenzieren konnten. Die Fehlermeldung sah wie folgt aus:

---

Field 'senderofNachricht\_IDNACHRICHT' doesn't have a default value

---

Dieser Fehler konnte mit Hilfe der Verwendung von @JoinColumn gelöst werden. Zusätzlich musste das updateable auf true gesetzt werden, sonst gab es den Fehler, dass der Primary Key doppelt ist.

---

```
@JoinColumn(name="sendidnachricht", updatable = true)
@OneToMany(cascade = CascadeType.ALL, fetch=FetchType.LAZY)
private List<Nachricht> senderofNachricht= new LinkedList<>();

@JoinColumn(name="empfidnachricht", updatable = true)
@OneToMany(cascade = CascadeType.ALL, fetch=FetchType.LAZY)
private List<Nachricht> empfaengerofNachricht= new LinkedList<>();
```

---

Eine weitere Funktion, die eingebaut wurde, ist das Setzen der Bekanntenliste. Wenn ein Benutzer zum ersten Mal in Kontakt mit jemanden tritt, also eine „Anfrage schickt“, wird er in die Bekanntenliste des Empfängers eingefügt. Die Funktion ist in der Klasse Benutzer durch eine auf den Benutzer referenzierende Referenz realisiert worden. Dies dient dazu, dass später beim Ausstellen von Belegen keine Fehler passieren können.

## 5.7 Abrechnung (AK)

Die Abrechnung wird im System abgewickelt. Dabei muss der Ersteller des Inserates auf das betroffene Inserat gehen und anschließend auf „Beleg ausstellen“ klicken, dann wird er auf die Datenerfassungsseite weitergeleitet. Dabei werden die Daten über den eingeloggten User ausgefüllt. In unserer Anwendung werden „Suchen“ und „Angebote“ dargestellt. Wenn jemand etwas gesucht hat und es erhalten hat, dann ist er im Beleg als Käufer und der andere als Verkäufer abgespeichert. Wenn er jedoch etwas anbietet ist er der Verkäufer. Die Unterscheidung erhalten wir durch das Inserat, in welchem abgespeichert wurde, um welchen Typ es sich handelt. Dadurch können wir die Daten des eingeloggten Nutzers automatisch setzen und somit können keine Fehler in der Abbuchung von Talenten passieren. Der Partner ist mithilfe eines Dropdowns auszuwählen. Im Dropdown erscheint er, sobald er in Kontakt mit dem Ersteller des Inserates tritt.

Buchungsbeleg

Angebot

Verkäufer:	Kriegner, Antonia
Käufer:	
Zweck:	Rasenmähen
Betrag:	50 Talente
Datum:	Wähle ein Datum aus

Schicken

Februar 2019						
MO	DI	MI	DO	FR	SA	SO
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28			

Abbildung 5.4: Formular für den Buchungsbeleg - Screenshot Antonia Kriegner

Weiters ist das Datum mit Hilfe eines Kalenders auszuwählen, somit können auch hier keine Fehler in der Angabe des Formates passieren. Anschließend wird ein Preis angegeben und der Beleg abgeschickt.

Der Partner kann diesen Beleg kontrollieren und wenn etwas nicht stimmt, kann er den Beleg ablehnen. Wenn er mit dem Preis einverstanden ist, klickt er auf "Bestätigen". Dies hat zur Folge, dass der Beleg von beiden Seiten akzeptiert wurde und die Abbuchung der Talente erfolgen kann. Sobald diese Aktion vervollständigt wurde, kann der Beleg unter „Abgeschlossene Tauschgeschäfte“ eingesehen werden. Wichtig hierbei ist, dass auch das Gegenüber den Beleg sehen kann. Das heißt, sobald eine Transaktion erfolgt ist, wird der Beleg sowohl für den Verkäufer als auch für den Käufer sichtbar. Der grobe Überblick wird in einer Tabelle dargestellt. Durch Klicken auf einen Button öffnet sich ein PopUp und es werden zusätzliche Daten angezeigt. Dabei gibt es auch die Funktion, dass ein Beleg gedruckt werden kann. Die zusätzlichen Daten werden am Server mit einer Funktion ermittelt und dabei wird nur der Vorname und der Nachname der Beteiligten geschickt. Um dies umsetzen zu können, haben wir eine Format Methode geschrieben:

---

```
public JSONObject format(List<Benutzer> list) {
    JSONObject obj = new JSONObject();

    obj.append("kauf", ((Benutzer) list.get(0)).getNname() + " " +
               ((Benutzer) list.get(0)).getVname());
    obj.append("verkauf", ((Benutzer) list.get(1)).getNname() + " " +
               ((Benutzer) list.get(1)).getVname());

    return obj;
}
```

---

Diese Methode wenden wir dann wie folgt an:

---

```
repBenutzer.format(data);
```

---

Die Abbuchung am Server funktioniert so, dass zuerst ermittelt wird, um welchen Typ es sich handelt und somit definiert wird, wem der Betrag aufgebucht und wem abgebucht wird. Dann werden die aktuellen Kontostände aus der Datenbank gelesen und der Betrag addiert beziehungsweise subtrahiert. Die neuen Beträge werden anschließend wieder gesetzt und in die Datenbank geschrieben.

---

```
Konto kaufKonto = (Konto) query.getSingleResult();
kaufKonto.setKontostand(kaufKonto.getKontostand() - b.getBetrag());

Konto verkaufKonto = (Konto) query2.getSingleResult();
verkaufKonto.setKontostand(verkaufKonto.getKontostand() + b.getBetrag());
```

---

## 5.8 Drucken (AK)

Die Funktion „Drucken“ ist bei den Belegen eingebunden und sowohl für den Verkäufer als auch für den Käufer sichtbar. Dies ist wichtig, damit jeder eine Bestätigung der Abwicklung des Tausches erhält. Es gibt viele Möglichkeiten, das Drucken in die Anwendung einzubinden. Es kann zum Beispiel der Code: [94]

---

```
window.print();
```

---

aufgerufen werden und dann wird die ganze Seite, wie sie abgebildet ist, gedruckt. Wenn hierbei unerwünschte Elemente auftauchen, können diese auf unsichtbar gesetzt werden. Dies kann mit dem Media query vom Typ print umgesetzt werden: [20]

---

```
@media print
{
  .print {display:none;}
}
```

---

Eine Alternative wäre die Verwendung von angular-print gewesen. Als Referenz für den folgenden Inhalt gilt diese Seite: [12]

Diese Druckfunktion wird mit folgendem Befehl installiert:

---

```
npm i angular-print
```

---

Der wesentliche Vorteil ist, dass hier bei den einzelnen Boxen definiert werden kann, ob sie gedruckt werden sollen oder nicht. Die zu druckende Box wird mit „print-section“ gekennzeichnet. In dieser Sektion können Bereiche auch als unsichtbar markiert werden. Dabei gibt es „printHide“, wo die Elemente zwar versteckt werden, aber durch einen leeren Platz abgebildet werden. „printRemove“ entfernt das ganze Element vollständig.

---

```
<div print-section>
  <div print-remove>
    Entfernen
  </div>

  <div print-hide>
    Verstecken
  </div>

  <div> Sichtbar </div>
</div>
```

---

Ein Nachteil an angular-print ist, dass eine Installation notwendig ist, um diese Funktion verwenden zu können. Wenn die Bereiche markiert wurden, dann kann mit Hilfe eines Elements, das mit „print-btn“ versehen ist, die Druckfunktion aufgerufen werden. Zum Beispiel mit Hilfe eines Buttons:

---

```
<button print-btn> Print </button>
```

---

# Kapitel 6

## Wissenswertes

### 6.1 Registrierung (MP)

In diesem Kapitel wird der Vorgang der Registrierung erläutert.

#### 6.1.1 Registrierungs-Formular

Das Registrierungs-Formular besteht aus elf Eingabefeldern und drei Buttons.

Die Eingabefelder für Vorname, Nachname, E-Mail, Telefonnummer, PLZ, Ort, Straße und Hausnummer sind ganz normale Text input Felder. Vor dem input-Feld befindet sich immer ein Label mit dem jeweiligen Attribut. Über das ngModel wird der eingegebene Wert an die ts-Klasse, welche die Daten weiterverarbeitet, weitergegeben.

---

```
<label>Vorname</label>
<input type="text" name="vname" [(ngModel)]="vname">

<label for="nname">Nachname</label>
<input type="text" name="nname" [(ngModel)]="nname">

<label for="fname">Email</label>
<input type="text" name="email" [(ngModel)]="email">

<label for="fname">Telefonnummer</label>
<input type="text" name="tel" [(ngModel)]="tel">

<label for="fname">PLZ</label>
<input type="text" name="plz" [(ngModel)]="plz">

<label for="fname">Ort</label>
<input type="text" name="ort" [(ngModel)]="ort">

<label for="fname">Strasse</label>
```

```
<input type="text" name="strasse" [(ngModel)]="strasse">  
  
<label for="fname">Hausnummer</label>  
<input type="text" name="hausnr" [(ngModel)]="hausnr">
```

---

Weiteres gibt es auch noch zwei Eingabefelder für das Passwort. Das Passwort muss ein zweites Mal eingegeben werden, denn falls sich der Benutzer beim ersten Mal vertippt, kann es zu Fehlern kommen und so wird dies vermieden.

Die Passwörter werden auch wieder über ngModel an die ts-Klasse übergeben. Dort wird dann überprüft, ob die Passwörter übereinstimmen.

```
<label for="fname">Passwort</label>  
<input type="password" name="passwort" [(ngModel)]="passwort">  
  
<label for="fname">Passwort erneut eingeben</label>  
<input type="password" name="passworte" [(ngModel)]="passworte">
```

---

Dann gibt es noch eine sogenannte TextArea. Das ist ein größeres Feld zur Eingabe von Nachrichten. In unserem Projekt verwenden wir es dafür, dass der Benutzer eine kurze Beschreibung über sich selbst eingeben kann. Der Placeholder hat den Wert "Beschreibung..", welcher im dargestellten Formular steht. Außerdem kann man mit height noch die Größe des Feldes bestimmen.

```
<label for="subject">Beschreibung</label>  
<textarea id="subject" name="subject" placeholder="Beschreibung.." style="height:200px" [(ngModel)]="beschreibung">  
</textarea>
```

---

Zum Schluss gibt es noch einen Button.

Der Button ist zum Registrieren. Er ruft die registrieren() Methode in der ts-Klasse auf.

```
<input type="submit" value="Registrieren" (click)="registrieren()">
```

---

### 6.1.2 JSON Objekt eines Benutzers

```
getConst(){
  const reg = {"id": this.id, "vname":this.vname, "nname":""+this.nname+"",
    "email":""+ this.email"",
    "tel":this.tel,"plz": this.plz,"ort":""+this.ort+"","hausnr": this.hausnr,
    "strasse":""+this.strasse+"","beschreibung":""+this.beschreibung+"",
    "passwort": this.passwort};
  return reg;
}
```

Bei dieser Methode wird ein JSON Objekt mit den Daten des Benutzers erstellt. Ein JSON Objekt besteht immer aus einem Key Value Pair. Der Value wird immer auf den Key gespeichert. In unserem Fall werden die eingegebenen Daten des Benutzers auf den jeweiligen Key gespeichert und als fertiges JSON Objekt zurück gegeben. Dieses Objekt wird benötigt, um den Benutzer am Server in die Datenbank speichern zu können.

### 6.1.3 Benutzer löschen

```
removeKonto(){
  this.rest.removeBenutzer(JSON.stringify(this.getConst())).subscribe();
  alert("Ihr Konto wurde gelöscht. Sie sind jetzt kein Mitglied mehr.")
}
```

Falls ein Benutzer nicht mehr Teil der Tauschbörse sein möchte, muss er die Möglichkeit haben, sein Konto löschen zu können. Dies wird mit der Methode removeKonto() möglich. Die Methode beinhaltet den rest Aufruf removeBenutzer, welchen man in der untenstehenden Abbildung 6.1 sehen kann. Durch diesen rest Aufruf wird der jeweilige Benutzer aus der Datenbank gelöscht. Danach wird noch ein alert Fenster mit der Meldung "Ihr Konto wurde gelöscht. Sie sind jetzt kein Mitglied mehr." dargestellt.

```
/**
 * Löschen des Benutzers
 * @param body
 */
removeBenutzer(body){
  return this.http.post('http://localhost:8080/api/service/removeBenutzer', body+"");
}
```

Abbildung 6.1: Benutzer löschen (Rest Aufruf) - Screenshot Melanie Pittersberger

#### 6.1.4 Registrieren Methode

In der registrieren() Methode werden die Eingaben auf Fehler überprüft. Wenn die Daten fehlerlos eingegeben wurden, wird der Benutzer registriert.

Hier ist ein Auszug aus den Fehlerüberprüfungen:

---

```
if(this.vname == null || this.vname.length <= 2){  
    alert("Der Vorname ist entweder zu kurz oder wurde nicht eingegeben.");  
}  
else{  
    if(this.nname == null || this.nname.length <= 2){  
        alert("Der Nachname ist entweder zu kurz oder wurde nicht eingegeben.");  
    }  
    else{  
        if(this.email == null){  
            alert("Die E-Mail ist nicht vorhanden.");  
        }  
        else ...  
    }  
}
```

---

Es gibt für jedes Feld Überprüfungen, vor allem wird bei jedem Feld überprüft, ob etwas eingegeben wurde. Außerdem wird auf Validität geprüft.

Außerdem wird das Passwort auf Richtigkeit überprüft. Der Benutzer muss es zweimal eingeben und diese beiden Eingaben werden dann miteinander verglichen, ob sie übereinstimmen. Falls dies nicht der Fall ist, wird eine Fehlermeldung ausgegeben.

Wenn alle Überprüfungen bestanden wurden, wird eine Meldung ausgegeben, dass die Registrierung erfolgreich war und dass der Benutzer nun eine Bestätigungsemail erhalten hat.

---

```
... else{  
    if(this.passwort != this.passworte){  
        alert("Die Passwörter stimmen nicht überein.");  
    }  
    else{  
        this.rest.registrieren(JSON.stringify(this.getConst())).subscribe();  
        alert("Vielen Danke für die Registrierung. Es wurde eine Bestätigungsemail an  
              Sie gesendet. Bitte öffnen Sie diese Mail und klicken Sie zur Bestätigung  
              Ihrer EMail-Adresse auf den Link in der Mail.");  
    }  
}
```

---

Außerdem wird der Benutzer mithilfe des rest Aufrufes "registrieren", welchen man in der Abbildung 6.2 sieht, an den Server geschickt und in die Datenbank gespeichert. In der Datenbank wird er in die Tabelle Registrierung gespeichert.

```

    /**
     * Registrierung eines Benutzers
     * @param body
     */
    registrieren(body){
        console.log(body);
        return this.http.post('http://localhost:8080/api/service/registrieren', body+"");
    }

```

Abbildung 6.2: Registrieren (Rest Aufruf) - Screenshot Melanie Pittersberger

## 6.2 Login (MP)

Sobald sich der Benutzer registriert hat und den Bestätigungslink angeklickt hat, wird er in der Datenbank aus der Registrierung Tabelle gelöscht und in die Benutzer Tabelle gespeichert. Er kann sich von nun an einloggen.

Der Benutzer braucht für den Login seine E-Mail Adresse und sein Passwort.

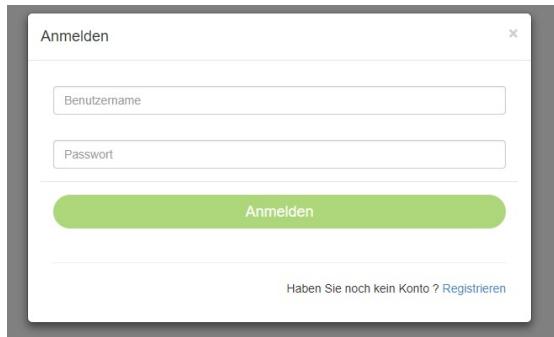


Abbildung 6.3: Login - Screenshot Melanie Pittersberger

### 6.2.1 Login Methode am Client

---

```

loginUser(e) {
e.preventDefault();

const email = e.target.elements[0].value;
const password = e.target.elements[1].value;

const body = {"email":" "+email+"","password":" "+password+""};

this.rest.login(body)
.subscribe(data => {
console.log(data);
}

```

```

if(data!=null){
    this.user.setUserLoggedIn();
    this.user.setUser(data.benutzer);

    localStorage.setItem('benutzer', JSON.stringify(data.benutzer));

    document.getElementById("buttonOpen").click();

    this.router.navigate(["loggtIn"]);
}

else{
    alert("Falsche Eingabe!");
}
}

```

---

Die Login Methode hat einen Parameter e, der mitübergeben wird. Aus diesem Parameter werden zuerst die Daten herausgelesen. Dann werden diese in email und password gespeichert. Als nächstes wird ein JSON Objekt gebaut mit zwei Key-Value Pairs. Der erste Key ist "email", auf den wird die E-Mail Adresse gespeichert. Der Zweite ist "passwort", auf diesen wird das Passwort gespeichert. Dann wird die eigentliche Loginfunktion mithilfe des rest Services aufgerufen. Falls ein Fehler auftritt, wird die Nachricht "Falsche Email oder falsches Passwort eingegeben!" ausgegeben und ein Error geworfen. Wenn die eingebenen Daten stimmen, dann wird der Benutzer auf eingeloggt gesetzt und es wird zur loggtIn Seite weiter geleitet.

### 6.2.2 Login Formular

Das Login Formular besteht aus zwei Input Feldern für die E-Mail Adresse und für das Passwort. Weiteres gibt es noch einen Anmelden Button, auf welchen der Benutzer klicken muss, um sich einzuloggen.

---

```

<input id="username" class="form-control" type="text"
placeholder="Benutzername" >

<input id="password" type="text" class="form-control" type="text"
placeholder="Passwort" >

<input type="submit" name="submit" value="Anmelden">

```

---

Falls der Benutzer noch kein Konto hat, kann er auf die Verlinkung klicken und kommt somit zum Registrieren Formular.

---

```

Haben Sie noch kein Konto ?
<a href="/regist">Registrieren</a>

```

---

### 6.3 Inserate erstellen (MP)

Um ein neues Inserat erstellen zu können, muss der Benutzer auf "Mein Konto" gehen und auf die Schaltfläche "Inserate erstellen" klicken.

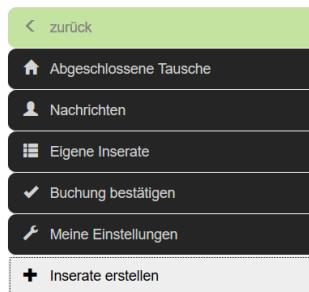


Abbildung 6.4: Inserate erstellen Schaltfläche - Screenshot Melanie Pittersberger

Dort kommt er zu einem Formular, in welchem er die Daten seines zu erstellenden Inserates eingeben kann.

#### 6.3.1 Inserate erstellen Formular

Das Formular erscheint rechts neben der Schaltfläche. Das Formular besteht aus Text-Eingabefeldern für den Titel, die Beschreibung und den Preis in Talenten. Außerdem kann ein Start- und Enddatum, an welchem das jeweilige Inserat gilt, festgelegt werden. Dazu kann man das Datum mithilfe eines DatePickers auswählen. Für Art, Typ und Kategorien gibt es CheckBoxen mit den jeweiligen Attributen.

Zum Schluss des Formulares hat der Benutzer noch die Möglichkeit, Bilder hochzuladen, um sein Produkt beziehungsweise seine Dienstleistung noch besser zu präsentieren.



Abbildung 6.5: Inserate erstellen Kalender - Screenshot Melanie Pittersberger

Abbildung 6.6: Inserate erstellen Formular - Screenshot Melanie Pittersberger

### 6.3.1.1 Text Input Felder

Dies ist ein Auszug aus dem html-File, in welchem das Formular für das Erstellen der Inserate geschrieben wurde:

---

```
<div class="row">
<div class="col-25">
<label>Titel</label>
</div>
<div class="col-75">
<input type="text" name="titel" [(ngModel)]="titel">
</div>
</div>
```

---

Die Texteingabefelder wurden alle mit einem input-Feld erstellt. Davor gibt es jeweils ein label, welches den Namen des input-Feldes darstellt. Durch das ngModel werden die eingegebenen Daten an die ts-Klasse übergeben, welche sie dann weiter verarbeiten kann.

### 6.3.1.2 DatePicker

Das Start- und Enddatum kann man mithilfe eines DatePickers auswählen. Dieser wird mit dem untenstehenden Tag dargestellt.

---

```
<div class="row">
<div class="col-25">
<label for="fname">Von</label>
</div>
<div class="col-75">
<ng-datepicker [options]=option name="von" [(ngModel)]="von"></ng-datepicker>
</div>
</div>
```

---

[63] [62]

### 6.3.1.3 CheckBoxen

Die Art, der Typ und die Kategorien werden mit CheckBoxen angegeben. Zuerst wird wieder das label angegeben, danach werden die weiteren label für die CheckBoxen an sich angegeben.

Um CheckBoxen zu bekommen, muss man beim input-Feld den type auf radio setzen. CheckBoxen werden auch RadioBoxen genannt.

---

```
<div class="col-25">
<label for="fname">Art</label>
</div>
<div class="col-75">
<form>
<label for="Produkt">Produkt</label>
<input type="radio" [(ngModel)]="art" name="same" value="Produkt">
<br>
<label for="Dienstleistung">Dienstleistung</label>
<input type="radio" [(ngModel)]="art" name="same" value="Dienstleistung">
</form>
<br/>
</div>
```

---

### 6.3.2 Image Upload

Ein weiteres Feature des Formulars ist, dass der Benutzer ein oder mehrere Bilder zu seinem Inserat hochladen kann.

Dies wurde mit einem File Upload gelöst:

Zuerst werden die Daten aus dem input-Feld geholt. Dann wird mit einer for-Schleife durchgegangen. Der Pfad wird ermittelt und in eine Base64 String konvertiert. Die Anzahl der Bytes werden in ein ArrayBuffer gespeichert und in diesem Buffer wird eine View zum Darstellen der Bilder erstellt. Falls das Hochladen des Bildes nicht funktioniert, wird eine Fehlermeldung ausgegeben.

### 6.3.3 Inserat erstellen Methode

Wenn das Formular fertig ausgefüllt wurde, muss das Inserat noch erstellt werden. Ein fertiges Inserat wird in die Inserat Tabelle in der Datenbank gespeichert. Durch die untenstehende for-Schleife wird zum Hochladen der richtige Bildpfad ermittelt. Die beiden darunterstehenden Zeilen sind für das Setzen des Datums zuständig. Sie formatieren das Datum richtig und es wird auf den Variablen datum1 und datum2, welche für Start- und Enddatum stehen, gespeichert.

Danach wird ein neues Inserat Objekt erstellt, auf diesen werden dann die einzelnen Attribute, welche im Formular eingegeben wurden, gespeichert. Mit dem sendInserat() Aufruf wird das Inserat an den Server geschickt, welcher dieses dann in die Datenbank speichert, sodass es dargestellt werden kann.

---

```
var imageType = /image.*/;

for (var i=0; i<this.file.length;i++) {
var file=this.fileInput.files[i]
if(file.type.match(imageType)&&this.file[i].size<1097152){
this.path[i] = ("http://localhost:8080/upload_image/"+this.file[i].name)
}
}

this.datum1 = this.von.getDate()+"."+this.von.getMonth()+"+"
+(this.von.getYear()+ 1900);
this.datum2 = this.bisd.getDate()+"."++(this.bisd.getMonth()+1)+"."
+(this.bisd.getYear()+ 1900);

this.ins = new Inserat(this.titel, this.preis, this.art, this.typ,
this.beschreibung, this.datum1, this.datum2, this.path, true);

this.rest.sendInserat(this.ins, this.user.getUser(),
this.matchKatFilter).subscribe();
```

---

## Kapitel 7

# Schlussfolgerung

Das Projekt wird nach Abschluss der Diplomarbeit nochmals vor den Auftraggebern präsentiert. Falls noch kleine optische Verbesserungswünsche auftreten, werden diese noch geändert. Anschließend wird von einem der Auftraggeber ein Cloud-Server organisiert und auf diesem wird unser Projekt laufen. Nach dem Aufsetzen werden die bestehenden Mitglieder in die Datenbank eingefügt und der generierte Zugangscode wird ihnen mitgeteilt.

Rückblickend würden wir sagen, dass sich die arbeitsreichen Monate gelohnt haben, da sich das Endprodukt der Diplomarbeit sehen lassen kann. Wir haben viel Erfahrung gesammelt und im Bereich Projektentwicklung enorm viel Wissen dazugewonnen.

# Literaturverzeichnis

- [1] Eingesehen am: 2019-03-31. URL: <https://pi4.informatik.uni-mannheim.de/pi4.data/content/courses/2002-ss/pm/dokumente/vorlagen/pflichtenheft.html>.
- [2] Adobe illustrator. Eingesehen am: 2019-02-20. URL: [https://de.wikipedia.org/w/index.php?title=Adobe\\_Illustrator&oldid=181102989](https://de.wikipedia.org/w/index.php?title=Adobe_Illustrator&oldid=181102989).
- [3] Angular. Eingesehen am: 2019-03-31. URL: <https://de.wikipedia.org/w/index.php?title=Angular&oldid=186961949>.
- [4] Angular - architecture overview. Eingesehen am: 2019-04-01. URL: <https://angular.io/guide/architecture>.
- [5] Angular - getting started. Eingesehen am: 2019-04-01. URL: <https://angular.io/guide/quickstart>.
- [6] Angular - HTTP. Eingesehen am: 2019-04-01. URL: <https://angular.io/tutorial/toh-pt6>.
- [7] Angular - introduction to components. Eingesehen am: 2019-04-01. URL: <https://angular.io/guide/architecture-components>.
- [8] Angular - introduction to modules. Eingesehen am: 2019-04-01. URL: <https://angular.io/guide/architecture-modules>.
- [9] Angular - introduction to services and dependency injection. Eingesehen am: 2019-04-01. URL: <https://angular.io/guide/architecture-services>.
- [10] Angular CLI. Eingesehen am: 2019-04-01. URL: <https://cli.angular.io/>.
- [11] Angular gallery. Eingesehen am: 2018-07-10. URL: <https://lukasz-galka.github.io/ngx-gallery-demo/#async-images>.
- [12] angular-print. Eingesehen am: 2019-03-04. URL: <https://www.npmjs.com/package/angular-print>.
- [13] @angular/cli. Eingesehen am: 2019-04-01. URL: <https://www.npmjs.com/package/@angular/cli>.

- [14] Apache maven. Eingesehen am: 2019-03-02. URL: [https://de.wikipedia.org/w/index.php?title=Apache\\_Maven&oldid=181149550](https://de.wikipedia.org/w/index.php?title=Apache_Maven&oldid=181149550).
- [15] Bootstrap panels. Eingesehen am: 2019-03-04. URL: [https://www.w3schools.com/bootstrap/bootstrap\\_panels.asp](https://www.w3schools.com/bootstrap/bootstrap_panels.asp).
- [16] Bootstrap panels - examples & tutorial. basic & advanced usage. Eingesehen am: 2019-03-04. URL: <https://mdbootstrap.com/docs/jquery/components/panels/>.
- [17] Cascading style sheets. Eingesehen am: 2019-03-31. URL: [https://de.wikipedia.org/w/index.php?title=Cascading\\_Style\\_Sheets&oldid=184417116](https://de.wikipedia.org/w/index.php?title=Cascading_Style_Sheets&oldid=184417116).
- [18] Centrum wiskunde & informatica. Eingesehen am: 2019-03-04. URL: [https://de.wikipedia.org/w/index.php?title=Centrum\\_Wiskunde\\_%26\\_Informatica&oldid=182717063](https://de.wikipedia.org/w/index.php?title=Centrum_Wiskunde_%26_Informatica&oldid=182717063).
- [19] CREATE DATABASE. Eingesehen am: 2019-03-24. URL: <https://mariadb.com/kb/en/library/create-database/>.
- [20] css - how do i hide an element when printing a web page? Eingesehen am: 2019-03-26. URL: <https://stackoverflow.com/questions/355313/how-do-i-hide-an-element-when-printing-a-web-page>.
- [21] Datenmodellierung in angular: Die architektur einer angular-applikation richtig gestalten | t3n – digital pioneers. Eingesehen am: 2019-03-31. URL: <https://t3n.de/magazin/datenmodellierung-angular-architektur-angluar-applikation-242317/>.
- [22] Design of a reflective REST-based query API - PDF. Eingesehen am: 2019-03-04. URL: <https://docplayer.org/8958351-Design-of-a-reflective-rest-based-query-api.html>.
- [23] Design of a reflective REST-based query API - PDF. Eingesehen am: 2019-03-04. URL: <https://docplayer.org/8958351-Design-of-a-reflective-rest-based-query-api.html>.
- [24] Drei gründe warum TypeScript das bessere JavaScript ist. Eingesehen am: 2019-03-31. URL: <https://www.communardo.de/techblog/drei-gruende-warum-typescript-das-bessere-javascript-ist/>.
- [25] Eigenschaften von java. Eingesehen am: 2019-03-31. URL: <https://www.java-tutorial.org/java-eigenschaften.html>.
- [26] Einfacher e-mail-versand mit SMTP in java. Eingesehen am: 2019-03-31. URL: <https://www.torsten-horn.de/techdocs/java-smtp.htm>.

- [27] Einführung in maven. Eingesehen am: 2019-03-02. URL: <http://home.edvsz.fh-osnabrueck.de/skleuker/CSI/Werkzeuge/Maven/index.html>.
- [28] Elementor for developers - build your own widgets & integrations. Eingesehen am: 2019-03-04. URL: <https://elementor.com/development/>.
- [29] Entity mappings: Introduction to JPA FetchTypes. Eingesehen am: 2019-03-04. URL: <https://thoughts-on-java.org/entity-mappings-introduction-jpa-fetchtypes/>.
- [30] Git - git grundlagen. Eingesehen am: 2019-02-20. URL: <https://git-scm.com/book/de/v1/Los-geht%80%99s-Git-Grundlagen>.
- [31] git\_gitflow.pdf. Eingesehen am: 2019-02-20. URL: [https://pgi-jcns.fz-juelich.de/pub/doc/git\\_gitflow.pdf](https://pgi-jcns.fz-juelich.de/pub/doc/git_gitflow.pdf).
- [32] gmail - com.sun.mail.smtp.SMTPSendFailedException: 530-5.5.1 authentication required (java mail). Eingesehen am: 2019-03-31. URL: <https://stackoverflow.com/questions/26728132/com-sun-mail-smtp-smtpsendfailedexception-530-5-5-1-authentication-required-ja/26902310>.
- [33] hibernate - difference between FetchType LAZY and EA-GER in java persistence API? Eingesehen am: 2019-03-04. URL: <https://stackoverflow.com/questions/2990799/difference-between-fetchtype-lazy-and-eager-in-java-persistence-api>.
- [34] How to install NetBeans for java programming on windows, mac OS and ubuntu. Eingesehen am: 2019-03-04. URL: [http://www.ntu.edu.sg/home/ehchua/programming/howto/netbeans\\_howto.html](http://www.ntu.edu.sg/home/ehchua/programming/howto/netbeans_howto.html).
- [35] How to view MariaDB databases in SSH | InMotion hosting. Eingesehen am: 2019-03-04. URL: <https://www.inmotionhosting.com/support/website/databases/mariadb-view-databases>.
- [36] HTML links. Eingesehen am: 2019-03-31. URL: [https://www.w3schools.com/html/html\\_links.asp](https://www.w3schools.com/html/html_links.asp).
- [37] HTML lists. Eingesehen am: 2019-03-31. URL: [https://www.w3schools.com/html/html\\_lists.asp](https://www.w3schools.com/html/html_lists.asp).
- [38] Hypertext markup language. Eingesehen am: 2019-03-31. URL: [https://de.wikipedia.org/w/index.php?title=Hypertext\\_Markup\\_Language&oldid=185825621](https://de.wikipedia.org/w/index.php?title=Hypertext_Markup_Language&oldid=185825621).
- [39] Installing MariaDB MSI packages on windows. Eingesehen am: 2019-03-04. URL: <https://mariadb.com/kb/en/library/installing-mariadb-msi-packages-on-windows/>.

- [40] java - how can i generate an MD5 hash? Eingesehen am: 2019-03-04. URL: <https://stackoverflow.com/questions/415953/how-can-i-generate-an-md5-hash>.
- [41] java - what is wrong here when i sending mail? Eingesehen am: 2019-03-31. URL: <https://stackoverflow.com/questions/49897310/what-is-wrong-here-when-i-sending-mail>.
- [42] Java definition. Eingesehen am: 2019-02-19. URL: <https://www.gruenderszene.de/lexikon/begriffe/java>.
- [43] java-persistence-api JPA EntityManager: Warum persist () über merge () verwenden? - CODE q&a gelöst. Eingesehen am: 2019-02-19. URL: <https://code.i-harness.com/de/q/1053a8>.
- [44] Java-technologie. Eingesehen am: 2019-03-31. URL: <https://de.wikipedia.org/w/index.php?title=Java-Technologie&oldid=186818848>.
- [45] JavaScript object notation. Eingesehen am: 2019-03-04. URL: [https://de.wikipedia.org/w/index.php?title=JavaScript\\_Object\\_Notation&oldid=186234730](https://de.wikipedia.org/w/index.php?title=JavaScript_Object_Notation&oldid=186234730).
- [46] jpa - JPA ist der EntityManager createQuery() vs createNamedQuery() vs createNativeQuery(). Eingesehen am: 2019-02-19. URL: <https://im-coder.com/jpa-ist-der-entitymanager-createquery-vs-createnamedquery-vs-createnativequery.html>.
- [47] JPA EntityManager - hibernate EntityManager. Eingesehen am: 2019-02-19. URL: <https://www.journaldev.com/17379/jpa-entitymanager-hibernate>.
- [48] JSON als XML-alternative. Eingesehen am: 2019-03-04. URL: <https://www.webmasterpro.de/coding/article/json-als-xml-alternative.html>.
- [49] JSON introduction. Eingesehen am: 2019-03-04. URL: [https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp).
- [50] JSON vs XML. Eingesehen am: 2019-03-04. URL: [https://www.w3schools.com/js/js\\_json\\_xml.asp](https://www.w3schools.com/js/js_json_xml.asp).
- [51] Lazy loading. Eingesehen am: 2019-03-04. URL: <https://de.wikipedia.org/w/index.php?title=Lazy>Loading&oldid=145284115>.
- [52] MariaDB. Eingesehen am: 2019-03-24. URL: <https://de.wikipedia.org/w/index.php?title=MariaDB&oldid=186840883>.
- [53] MariaDB vs MySQL - vorteile und nachteile im Überblick. Eingesehen am: 2019-03-24. URL: <https://blog.php-dev.info/2014/04/mariadb-vs-mysql/>.

- [54] Maven – introduction to the POM. Eingesehen am: 2019-03-02. URL: <https://maven.apache.org/guides/introduction/introduction-to-the-pom.html>.
- [55] MD5 generator und passwortsicherheit. Eingesehen am: 2019-03-04. URL: <https://www.md5-generator.de/>.
- [56] Media queries in CSS. how do they work? • silo creativo. Eingesehen am: 2019-03-04. URL: <https://www.silocreativo.com/en/media-queries-css-work/>.
- [57] Melanie pittersberger / tauschboerseDA. Eingesehen am: 2019-02-20. URL: <https://gitlab.com/m.pittersberger/tauschboerseDA>.
- [58] MessageDigest (java platform SE 7 ). Eingesehen am: 2019-03-04. URL: <https://docs.oracle.com/javase/7/docs/api/java/security/MessageDigest.html>.
- [59] MongoDB. Eingesehen am: 2019-03-31. URL: <https://de.wikipedia.org/w/index.php?title=MongoDB&oldid=186818117>.
- [60] MySQL. Eingesehen am: 2019-03-31. URL: <https://de.wikipedia.org/w/index.php?title=MySQL&oldid=187035826>.
- [61] NetBeans IDE. Eingesehen am: 2019-03-04. URL: [https://de.wikipedia.org/w/index.php?title=NetBeans\\_IDE&oldid=184965630](https://de.wikipedia.org/w/index.php?title=NetBeans_IDE&oldid=184965630).
- [62] ng-datepicker. Eingesehen am: 2019-03-31. URL: <https://www.npmjs.com/package/ng-datepicker>.
- [63] ng2-datepicker. Eingesehen am: 2019-03-31. URL: <https://www.npmjs.com/package/ng2-datepicker>.
- [64] Ng2-image-viewer demo. Eingesehen am: 2019-03-04. URL: <https://brenoprata10.github.io/ng2-image-viewer/>.
- [65] ngx-gallery. Eingesehen am: 2019-03-04. URL: <https://www.npmjs.com/package/ngx-gallery>.
- [66] Oracle (datenbanksystem). Eingesehen am: 2019-03-31. URL: [https://de.wikipedia.org/w/index.php?title=Oracle\\_\(Datenbanksystem\)&oldid=186716093](https://de.wikipedia.org/w/index.php?title=Oracle_(Datenbanksystem)&oldid=186716093).
- [67] Pflichtenheft - software engineering i WS 2012/2013. URL: <https://www.isf.cs.tu-bs.de/cms/teaching/2012w/se1/pflichtenheft.pdf>.
- [68] Pflichtenheft.pdf. Eingesehen am: 2019-03-31. URL: [https://studium.jumland.de/Software\\_Entwic/XML\\_Import/Pflichtenheft.pdf](https://studium.jumland.de/Software_Entwic/XML_Import/Pflichtenheft.pdf).
- [69] The pros and cons of 8 popular databases. Eingesehen am: 2019-03-04. URL: <https://www.keycdn.com/blog/popular-databases>.

- [70] Responsive web design media queries. Eingesehen am: 2019-03-04. URL: [https://www.w3schools.com/css/css\\_rwd\\_mediaqueries.asp](https://www.w3schools.com/css/css_rwd_mediaqueries.asp).
- [71] Scrum kurz und bündig erklärt - wibas. Eingesehen am: 2019-03-04. URL: <https://www.wibas.com/de/scrum/>.
- [72] Secure hash algorithm. Eingesehen am: 2019-03-04. URL: [https://de.wikipedia.org/w/index.php?title=Secure\\_Hash\\_Algorithm&oldid=184302459](https://de.wikipedia.org/w/index.php?title=Secure_Hash_Algorithm&oldid=184302459).
- [73] SHA-2. Eingesehen am: 2019-03-31. URL: <https://de.wikipedia.org/w/index.php?title=SHA-2&oldid=183548521>.
- [74] SHA1 generator - online checksum / hash generieren. Eingesehen am: 2019-03-31. URL: <http://www.sha1generator.de/>.
- [75] SHA1 in java. Eingesehen am: 2019-03-04. URL: <http://www.sha1-online.com/sha1-java/>.
- [76] SHA256 hash generator online. Eingesehen am: 2019-03-31. URL: <https://passwordsgenerator.net/sha256-hash-generator/>.
- [77] Sprint retrospective: Ziel einer sprint retrospektive. Eingesehen am: 2019-03-24. URL: <https://www.mvdn.de/sprint-retrospective>.
- [78] Sprint retrospektive. Eingesehen am: 2019-03-24. URL: <https://digitalisierungscoach.com/digitalisierung/scrum/scrum-meetings/sprint-retrospektive/>.
- [79] Stackoverflow verschluesselung. Eingesehen am: 2019-04-01. URL: <https://stackoverflow.com/questions/29782631/simple-password-encryption-how-do-i-do>.
- [80] TypeScript. Eingesehen am: 2019-03-31. URL: <https://de.wikipedia.org/w/index.php?title=TypeScript&oldid=185852506>.
- [81] Using media queries. Eingesehen am: 2019-03-04. URL: [https://developer.mozilla.org/de/docs/Web/CSS/Media\\_Queries/Using\\_media\\_queries](https://developer.mozilla.org/de/docs/Web/CSS/Media_Queries/Using_media_queries).
- [82] Verfahren der kryptographie, teil 16: SHA-2 ist noch für einige jahre sicher | dipl.-inform. carsten eilers. Eingesehen am: 2019-03-31. URL: <https://www.ceilers-news.de/serendipity/807-Verfahren-der-Kryptographie,-Teil-16-SHA-2-ist-noch-fuer-einige-Jahre-sicher.html>.
- [83] Visual studio code. Eingesehen am: 2019-03-31. URL: [https://de.wikipedia.org/w/index.php?title=Visual\\_Studio\\_Code&oldid=186694407](https://de.wikipedia.org/w/index.php?title=Visual_Studio_Code&oldid=186694407).
- [84] Was ist ein lastenheft und was gehört rein? Eingesehen am: 2019-03-31. URL: <https://wiki.induux.de/Lastenheft>.

- [85] Was ist eine REST API? Eingesehen am: 2019-03-04. URL: <https://www.cloudcomputing-insider.de/was-ist-eine-rest-api-a-611116/>.
- [86] Was ist md5? – PHP lernen. Eingesehen am: 2019-03-04. URL: <https://www.php-einfach.de/diverse/was-ist-md5/>.
- [87] Was sind eigentlich meilensteine? Eingesehen am: 2019-03-31. URL: <https://projekte-leicht-gemacht.de/blog/definitionen/definition-meilensteine/>.
- [88] Weniger sicheren apps den zugriff auf ihr konto gestatten - google-konto-hilfe. Eingesehen am: 2019-04-01. URL: <https://support.google.com/accounts/answer/6010255?hl=de>.
- [89] Wer ist dieser JSON? Eingesehen am: 2019-03-24. URL: <https://www.it-talents.de/blog/it-talents/wer-ist-dieser-json>.
- [90] What does 'stage' mean in git? Eingesehen am: 2019-03-23. URL: <https://softwareengineering.stackexchange.com/questions/119782/what-does-stage-mean-in-git>.
- [91] What is a sprint retrospective? Eingesehen am: 2019-03-24. URL: <https://www.scrum.org/resources/what-is-a-sprint-retrospective>.
- [92] What is REST? Eingesehen am: 2019-03-04. URL: <https://www.codecademy.com/articles/what-is-rest>.
- [93] Wie java funktioniert. Eingesehen am: 2019-03-31. URL: <http://www.straub.as/java/what-you-need/how-it-works.html>.
- [94] Window print() method. Eingesehen am: 2019-03-26. URL: [https://www.w3schools.com/jsref/met\\_win\\_print.asp](https://www.w3schools.com/jsref/met_win_print.asp).
- [95] XML vs. JSON | MobDok - mobile dokumentation. Eingesehen am: 2019-03-04. URL: <http://mobdok.de/2015/05/xml-vs-json/>.
- [96] YAML. Eingesehen am: 2019-03-04. URL: <https://de.wikipedia.org/w/index.php?title=YAML&oldid=177841983>.
- [97] YAML ain't markup language (YAML<sup>TM</sup>) version 1.2. Eingesehen am: 2019-03-04. URL: <https://yaml.org/spec/1.2/spec.html#id2759572>.
- [98] Atlassian. Scrum – was es ist, was nicht und warum es so großartig ist. Eingesehen am: 2019-03-04. URL: <https://de.atlassian.com/agile/scrum>.
- [99] Benjamin Brandmeier. Image gallery built with angular 2+, node.js and GraphicsMagick: BenjaminBrandmeier/angular2-image-gallery. Eingesehen am: 2019-03-04. URL: <https://github.com/BenjaminBrandmeier/angular2-image-gallery>.

- [100] {and} Bootstrap Mark Otto contributors, Jacob Thornton. Cards. Eingesehen am: 2019-03-04. URL: <https://getbootstrap.com/docs/4.0/components/card/>.
- [101] {and} Bootstrap Mark Otto contributors, Jacob Thornton. Carousel. Eingesehen am: 2019-03-04. URL: <https://getbootstrap.com/docs/4.0/components/carousel/>.
- [102] {and} Bootstrap Mark Otto contributors, Jacob Thornton. Grid system. Eingesehen am: 2019-03-04. URL: <https://getbootstrap.com/docs/4.0/layout/grid/>.
- [103] Wolfgang Dautermann. Soviel mathematik wie noetig, sogenig wie moeglich. URL: <https://wolfgang.dautermann.at/vortraege/CLT2010-Hashfunktionen.pdf>.
- [104] Invest. Was ist JPA? Eingesehen am: 2019-02-19. URL: <https://zkmablog.com/2016/11/30/was-ist-jpa/>.
- [105] Bernd Müller and Harald Wehr. *Java Persistence API 2: Hibernate, EclipseLink, OpenJPA und Erweiterungen*. Carl Hanser Verlag GmbH Co KG. Google-Books-ID: 86RPAgAAQBAJ.
- [106] heise online. Todesstoß: Forscher zerschmettern SHA-1. Eingesehen am: 2019-03-04. URL: <https://www.heise.de/security/meldung/Todesstoss-Forscher-zerschmettern-SHA-1-3633589.html>.
- [107] Janna Tillmann. Was ist eigentlich typescript? Eingesehen am: 2019-03-31. URL: <https://t3n.de/news/eigentlich-typescript-859869/>.
- [108] tutorialspoint.com. JPA - entity managers. Eingesehen am: 2019-02-19. URL: [https://www.tutorialspoint.com/de/jpa/jpa\\_entity\\_managers.htm](https://www.tutorialspoint.com/de/jpa/jpa_entity_managers.htm).
- [109] tutorialspoint.com. Maven POM. Eingesehen am: 2019-03-02. URL: [https://www.tutorialspoint.com/maven/maven\\_pom.htm](https://www.tutorialspoint.com/maven/maven_pom.htm).
- [110] www.tectrain.at. Kurs: Agile SCRUM foundation training und zertifizierung in wien und graz | tecTrain GmbH. Eingesehen am: 2019-03-04. URL: <https://www.tectrain.at/seminare/scrum/agile-scrum-foundation/agile-scrum-foundation>.

# Abbildungsverzeichnis

1.1	Schulfotograf - Antonia Kriegner . . . . .	7
1.2	Schulfotograf - Melanie Pittersberger . . . . .	8
1.3	Grundstruktur der Datenbank - Screenshot Antonia Kriegner . . . . .	10
2.1	<a href="https://en.wikipedia.org/wiki/HTML">https://en.wikipedia.org/wiki/HTML</a> . . . . .	12
2.2	<a href="https://www.leankoala.com/de/akademie/checks/css-selector-checker">https://www.leankoala.com/de/akademie/checks/css-selector-checker</a> . . .	13
2.3	<a href="https://raw.githubusercontent.com/remojansen/logo.ts/master/ts.png">https://raw.githubusercontent.com/remojansen/logo.ts/master/ts.png</a> . .	13
2.4	Projektstruktur - Screenshot Melanie Pittersberger . . . . .	14
2.5	Komponentenstruktur - Screenshot Melanie Pittersberger . . . . .	15
2.6	Auszug aus rest.service.ts - Screenshot Melanie Pittersberger . . . . .	16
2.7	loggt-in-start-page.component.ts - Screenshot Melanie Pittersberger . . .	16
2.8	Darstellung der Daten - Screenshot Melanie Pittersberger . . . . .	17
2.9	<a href="https://de.wikipedia.org/wiki/Java-Technologie/media/File:Java-Logo.svg">https://de.wikipedia.org/wiki/Java-Technologie/media/File:Java-Logo.svg</a>	18
2.10	<a href="https://openjpa.apache.org/builds/1.2.3/apache-openjpa/docs/img/jpa-arch.png">https://openjpa.apache.org/builds/1.2.3/apache-openjpa/docs/img/jpa-arch.png</a>	19
2.11	Entity Klasse - Screenshot Antonia Kriegner . . . . .	20
2.12	Oberfläche von Visual Studio Code - Screenshot Melanie Pittersberger . .	22
2.13	Oberfläche von NetBeans - Screenshot Antonia Kriegner . . . . .	23
2.14	Breakpoint in NetBeans - Screenshot Antonia Kriegner . . . . .	23
2.15	aktivierter Breakpoint in NetBeans - Screenshot Antonia Kriegner . . . .	24
2.16	Buttons zum Debuggen - Screenshot Antonia Kriegner . . . . .	24
2.17	Variablenfenster beim Debuggen - Screenshot Antonia Kriegner . . . . .	24
2.18	Erstellen eines Maven Projects - Screenshot Antonia Kriegner . . . . .	25
2.19	Oberfläche von Illustrator - Screenshot Antonia Kriegner . . . . .	27
2.20	Ebenen Fenster - Screenshot Antonia Kriegner . . . . .	27
2.21	Meilensteintrendanalyse - Screenshot Antonia Kriegner . . . . .	30
2.22	<a href="https://git-scm.com/figures/18333fig0106-tn.png">https://git-scm.com/figures/18333fig0106-tn.png</a> . . . . .	33
2.23	Branches in Git - Screenshot Antonia Kriegner . . . . .	34
2.24	Graph in Git - Screenshot Antonia Kriegnerr . . . . .	34
2.25	Oberfläche von ScrumDo - Screenshot Antonia Kriegner . . . . .	36
3.1	Wordpress Menüleiste - Screenshot Melanie Pittersberger . . . . .	37
3.2	Wordpress Informationen Seite - Screenshot Melanie Pittersberger . . . .	37
3.3	Wordpress Unterseite Mitglied werden - Screenshot Melanie Pittersberger	38
3.4	Wordpress Neuigkeiten - Screenshot Melanie Pittersberger . . . . .	38

3.5	Elementoberfläche - Screenshot Antonia Kriegner . . . . .	39
4.1	Tabellen auf der Datenbank - Screenshot Antonia Kriegner . . . . .	45
4.2	Driver anlegen - Screenshot Antonia Kriegner . . . . .	46
4.3	Connection erstellen - Screenshot Antonia Kriegner . . . . .	47
4.4	<a href="https://de.wikipedia.org/wiki/MySQL">https://de.wikipedia.org/wiki/MySQL</a> . . . . .	47
4.5	<a href="https://juanjoselo.files.wordpress.com/2017/10/screen-shot-2017-08-15-at-1-35-54-pm.png?w=300">https://juanjoselo.files.wordpress.com/2017/10/screen-shot-2017-08-15-at-1-35-54-pm.png?w=300</a> . . . . .	48
4.6	<a href="https://www.cloudsigma.com/de/einrichten-und-optimieren-einer-mongodb-auf-oeffentlichen-cloud-servern/">https://www.cloudsigma.com/de/einrichten-und-optimieren-einer-mongodb-auf-oeffentlichen-cloud-servern/</a> . . . . .	48
4.7	Klassendiagramm - Screenshot Antonia Kriegner . . . . .	50
4.8	<a href="https://zdnet4.cbsistatic.com/hub/i/r/2014/10/04/f680563a-4b5d-11e4-b6a0-d4ae52e95e57/resize/370xauto/a1b02cbb47b7dec7939c2576574576bc/screen-shot-2012-06-07-at-140213.png">https://zdnet4.cbsistatic.com/hub/i/r/2014/10/04/f680563a-4b5d-11e4-b6a0-d4ae52e95e57/resize/370xauto/a1b02cbb47b7dec7939c2576574576bc/screen-shot-2012-06-07-at-140213.png</a> . . . . .	54
4.9	<a href="https://wolfgang.dautermann.at/vortraege/CLT2010-Hashfunktionen.pdf">https://wolfgang.dautermann.at/vortraege/CLT2010-Hashfunktionen.pdf</a>	57
4.10	Bestätigungsmaill - Screenshot Melanie Pittersberger . . . . .	58
4.11	E-Mail zulassen - Screenshot Melanie Pittersberger . . . . .	60
5.1	Navigationsstruktur (MP) - Screenshot Melanie Pittersberger . . . . .	61
5.2	Navigationsstruktur Code (MP) - Screenshot Melanie Pittersberger . . . . .	61
5.3	Ansicht der eigenen Inserate - Screenshot Antonia Kriegner . . . . .	64
5.4	Formular für den Buchungsbeleg - Screenshot Antonia Kriegner . . . . .	72
6.1	Benutzer löschen (Rest Aufruf) - Screenshot Melanie Pittersberger . . . . .	78
6.2	Registrieren (Rest Aufruf) - Screenshot Melanie Pittersberger . . . . .	80
6.3	Login - Screenshot Melanie Pittersberger . . . . .	80
6.4	Inserate erstellen Schaltfläche - Screenshot Melanie Pittersberger . . . . .	82
6.5	Inserate erstellen Kalender - Screenshot Melanie Pittersberger . . . . .	83
6.6	Inserate erstellen Formular - Screenshot Melanie Pittersberger . . . . .	83

# Tabellenverzeichnis

2.1	Wichtige EntityManager Methoden [43] [46] [47] . . . . .	19
2.2	Wichtige Annotationen . . . . .	21
2.3	Meilensteinliste . . . . .	30
3.1	Response – Status . . . . .	40
4.1	Gleicher Text mit unterschiedlichen Hash Funktionen verschluesselt . . . . .	57
5.1	Bootstrap - Tabellen Optionen . . . . .	67

# **Anhang A**

## **Anhang**

### **A.0.1 Plichtenheft**



# Tauschbörse

Version 1: 18.07.2018

Version 1.1 Überarbeitet am 24.07.2018

Version 1.2 Überarbeitet am 30.07.2018

—

## Pflichtenheft

Antonia Kriegner

Melanie Pittersberger



## Inhaltsverzeichnis

1. Zielbestimmungen
  - 1.1. Musskriterien
  - 1.2. Wunschkriterien
  - 1.3. Abgrenzungskriterien
2. Produkteinsatz
  - 2.1. Anwendungsbereiche
  - 2.2. Zielgruppen
  - 2.3. Betriebsbedingungen
3. Produktumgebung
  - 3.1. Software
  - 3.2. Hardware
  - 3.3. Orgware
4. Produktfunktionen
  - 4.1. Benutzerfunktionen
    - 4.1.1. Benutzer-Kennung
    - 4.1.2. Persönliche Daten
    - 4.1.3. Persönliche Konfiguration
    - 4.1.4. Persönliches Profil
  - 4.2. Talente
  - 4.3. Inserate
5. Produktdaten
6. Produktleistungen
7. Benutzungsoberfläche
  - 7.1. Dialogstruktur
    - 7.1.1. Startseite
    - 7.1.2. Hauptseite
    - 7.1.3. Benutzermenü
  - 7.2. Bildschirmlayout

- 
8. Qualitätszielbestimmungen
  9. Globale Testszenarien und Testfälle
  10. Entwicklungsumgebung
    - 10.1. Software
    - 10.2. Hardware
    - 10.3. Orgware
  11. Ergänzungen
  12. Glossar

## 1 Zielbestimmungen

Die Tauschbörse stellt einen Internetdienst dar, welcher ähnlich zu willhaben, einen Marktplatz hat. Dort werden von verschiedenen Benutzern Produkte oder Dienstleistungen angeboten, welche gegen andere Produkte oder Dienstleistungen getauscht werden können.

### 1.1 Musskriterien

- Der Benutzer-Account
  - Der Benutzer kann sich selbst auf der Seite registrieren
  - Der Benutzer kann sich selbst auf der Seite an- und abmelden
  - Der Benutzer kann seine Anmelddaten anfordern, falls er diese Daten vergessen hat
  - Der Benutzer kann seine eigenen Daten einsehen
  - Der Benutzer kann die Inserate der anderen Benutzer einsehen
  - Der Benutzer kann Anfragen an andere Benutzer schicken
  - Der Benutzer kann seine eigenen Inserate ansehen
  - Der Benutzer kann Anfragen von anderen Benutzern auf seine eigenen Inserate sehen
  - Der Benutzer kann sich abgeschlossene Tauschgeschäfte ansehen
  - Der Benutzer kann nach gewissen Kriterien filtern
  - Der Benutzer kann nach bestimmten Produkten suchen
  - Der Benutzer kann sich eine Seite ansehen, auf welcher erklärt wird, wie das Prinzip der Tauschbörse funktioniert
  - Der Benutzer hat ein Startkapital von 10 Talenten, dieser Wert kann bis zu 300 Talente ins negativen gehen



- Das Inserat
  - Das Inserat muss eine Beschreibung haben
  - Das Inserat muss die Kontaktdaten des Benutzers beinhalten, um eine Anfrage senden zu können
  - Das Inserat muss einen Zeitraum beinhalten
  - Das Inserat muss Kategorien haben
  - Das Inserat muss einen Preis haben in Form von Talenten
  - Das Inserat muss filterbar sein
- Ein Talent
  - Der Preis in Form von Talenten wird an den Benutzer gegeben, der etwas verkauft
  - Der Preis in Form von Talenten wird von dem Benutzer abgezogen, der etwas kauft
  - Der Preis ist immer eine ganzzahlige Einheit

## 1.2 Wunschkriterien

- Der Benutzer kann sich eine Seite mit der Idee hinter der Tauschbörse ansehen
- Der Benutzer kann sich eine Seite mit Neuigkeiten ansehen

## 1.3 Abgrenzungskriterien

- nur Tauschgeschäfte möglich, d.h. keine Barabköse, es wird mit Talenten gerechnet
- gesetzliche Abklärung, inwieweit etwas als Tausch gilt, bzw. steuerliche Aspekte sind nicht Gegenstand dieser Arbeit
- AGBs werden vom Verein definiert
- keine Mehrsprachigkeit des Systems



## 2 Produkteinsatz

### 2.1 Anwendungsbereiche

Personen, welche im Rahmen der noch zu definierenden AGBs als berechtigte Nutzer definiert sind, können die Tauschbörse verwenden, um Produkte oder Dienstleistungen mit anderen Personen zu tauschen. Diese Plattform soll dazu dienen, Produkte oder Dienstleistungen nicht mit Geld, sondern mit Talenten, eine virtuelle Währung, zu bezahlen.

### 2.2 Zielgruppen

Personen, die eine gewisse Dienstleistung anbieten, wie z.B. Rasen mähen, Babysitten, Nachhilfe

Personen, die gewisse Produkte anbieten, wie z.B. Brot, Marmelade, Milch

Privatpersonen, die Dinge tauschen möchten, wie z.B. Haushaltsgeräte tauschen, wobei es egal ist ob das Produkt neu oder gebraucht ist

Personen, die die oben genannten Produkte und Dienstleistungen in Anspruch nehmen möchten

### 2.2 Betriebsbedingungen

Dieses System soll sich bezüglich der Betriebsbedingungen nicht wesentlich von anderen Internetdiensten bzw. -anwendungen unterscheiden.

- Betriebsdauer: täglich, 24 Stunden
- Die Sicherung der Datenbank muss manuell vom Administrator durchgeführt werden.
- Falls nötig, ist der Administrator für die Behebung eines Problems zwischen Benutzern verantwortlich. (jemand aus der Gemeinde Natternbach)



## 3 Produktumgebung

Das Produkt ist weitgehend unabhängig vom Betriebssystem, sofern folgende Produktumgebung vorhanden ist.

### 3.1 Software

- Client
  - HTML
  - CSS
  - Angular2
- Server
  - Java
  - MariaDB

### 3.2 Hardware

- Client
  - Internetfähiger Rechner
- Server
  - Internetfähiger Server
  - Rechner, der die Ansprüche der o.g. Server-Software erfüllt
  - Ausreichend Rechen- und Festplattenkapazität (Annahme: 40 Anfragen/h)

### 3.3 Orgware

- Gewährleistung, dass der Softwareanteil langfristig nicht durch Programmierfehler zum Programm- oder Systemabsturz führt
- Administrator muss den Internetdienst starten und die Betriebsparameter (Datenbank installieren, Programm hochladen) konfigurieren

## 4 Produktfunktionen

### 4.1 Benutzerfunktionen

#### 4.1.1 Benutzer-Kennung

Ein im System registrierter Benutzer kann das System erst nutzen, wenn er angemeldet ist.

##### **Registrieren:**

Ein beliebiger Benutzer, welcher in den AGBs definiert ist, kann sich über die Login-Seite des Systems schnell und bequem registrieren lassen.

Zum Registrieren sind mindestens folgende Angaben erforderlich:

- Vorname
- Nachname
- E-Mail
- Passwort (Hashcode)

Weitere Daten die aber nicht erforderlich sind:

- Straße
- Hausnummer
- Postleitzahl
- Ort
- Telefon

Die Registrierung ist erfolgreich, wenn die E-Mail-Adresse innerhalb des Systems jeweils eindeutig sind. Die E-Mail-Adresse wird auf ihre Gültigkeit geprüft.

Mit dem erfolgreichen Abschließen des Registrierungsvorgangs ist der neue Benutzer am System angemeldet.

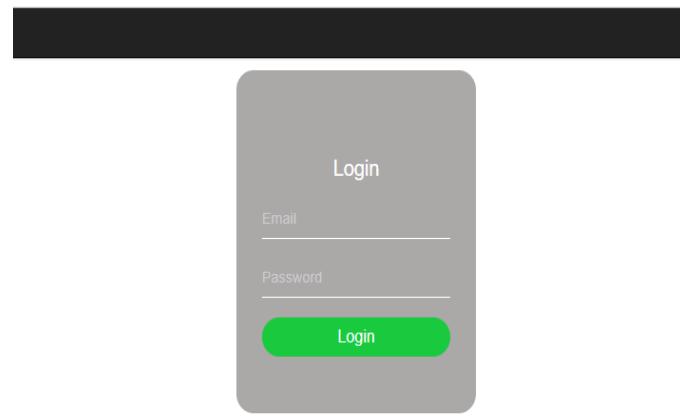


### Anmelden:

Ein bereits registrierter Benutzer kann sich über die Login-Seite des Systems schnell und bequem anmelden.

Dazu ist seine Kennung erforderlich:

- seine E-Mail-Adresse
- sein Passwort



(Design noch nicht abgestimmt)

### Abmelden:

Der angemeldete Benutzer kann sich jeder Zeit wieder vom System abmelden.

### Kennung anfordern:

Falls ein bereits registrierter Benutzer seine Anmelddaten vergessen haben sollte, so kann er diese über den Administrator anfordern. Dem Benutzer wird unter Angabe

- seiner eMail-Adresse

ein neues generiertes Passwort via E-Mail zu gesendet.

## 4.1.2 Persönliche Daten

Der Benutzer verfügt über seine persönlichen Daten.

### Anzeige der eigenen, persönlichen Daten:

Der Benutzer kann sich seine persönlichen Daten, außer sein Passwort vom System anzeigen lassen.

### Ändern der eigenen, persönlichen Daten:

Der Benutzer kann seine persönlichen Daten aktualisieren bzw. ändern.

### Anzeige der persönlichen Daten anderer Benutzer:

Der Benutzer kann sich von anderen Benutzern die Basisdaten anzeigen lassen.

## 4.1.3 Persönliche Konfiguration

Die Nutzungsumgebung eines Benutzers ist das Layout, das Design, aber auch diverse logische Einstellungen, die die individuelle Handhabung des Systems vereinfachen können.

Individuell einstellbar für den Benutzer sind:

- die Filterung der Inserate

Persönliche Konfigurationen können verwaltet werden.

### Anzeige der persönlichen Konfiguration:

Der Benutzer kann sich alle einstellbaren Werte seiner persönlichen Konfiguration seiner Nutzungsumgebung vom System anzeigen lassen.

### Ändern der persönlichen Konfiguration:

Der Benutzer kann alle einstellbaren Werte seiner persönlichen Konfiguration ändern oder die voreingestellte Konfiguration wiederherstellen.

### Speichern der persönlichen Konfiguration:

Der Benutzer kann seine persönliche Konfiguration speichern.



### **Löschen der persönlichen Konfiguration:**

Der Benutzer kann bereits gesicherte Konfigurationen entfernen.

## **4.1.4 Persönliches Profil**

Der Benutzer verfügt über ein persönliches Profil. Dieses kann man in zwei Teile gliedern:

- allgemeines Profil
  - die Daten des Benutzers, die er später auch noch ändern kann. (siehe 5.Produktdaten -> Benutzerdaten)
- seine Inserate

Das persönliche Profil wird je nach Bedarf vom System automatisch aktualisiert.

### **Anzeige des eigenen, persönlichen Profils:**

Der Benutzer kann sich sein persönliches Profil anzeigen lassen.

### **Anzeige der persönlichen Profile anderer Benutzer:**

Der Benutzer kann sich von anderen Benutzern die persönlichen Profile anzeigen lassen.

## **4.2 Talente**

Der Benutzer verfügt über ein Konto mit Talenten.

### **Talente werden abgebucht:**

Eine bestimmte Anzahl von Talenten werden abgebucht, wenn der Benutzer etwas kauft.

### **Talente werden aufgebucht:**

Eine bestimmte Anzahl von Talenten werden aufgebucht, wenn der Benutzer etwas verkauft.

## 4.3 Inserate

### Inserat wird freigeschaltet:

Wenn das Inserat fertig ausgefüllt ist, wird es freigeschaltet.

### Inserat wird gelöscht:

Wenn ein Produkt verkauft worden ist, kann das Inserat auf „unsichtbar“ gesetzt werden.

### Inserat ist sichtbar:

Ein Inserat muss auf sichtbar gestellt werden, wenn der Benutzer, welcher es verfasst hat, für andere Benutzer anzeigen möchte.

### Inserat ist unsichtbar

Ein Inserat kann auf unsichtbar gestellt werden, wenn der Benutzer, welcher es verfasst hat, es für andere Benutzer nicht mehr anzeigen lassen möchte.

The screenshot shows a web application for creating an advertisement. At the top, there's a navigation bar with links for 'Project Manager', 'Infos', 'Mitglied werden', and 'Anmelden'. Below the navigation, a title 'Hund aufpassen' is displayed above a large image of a brown, shaggy dog. To the right of the image, there's a brief description: 'in Aufpassen, Urlaub, Tiere, Haushalt' and a note: 'Ich kann für sie auf ihren Hund aufpassen. Mir macht es Spaß und ich bin Ihnen gerne behilflich. Ich kann selber leider keinen haben, weil mein Papa allergisch ist und darum wurde es mich sehr freuen, wenn sie es in Anspruch nehmen würden'. Below this, a date range 'von 01.06.2019 bis 11.10.2019' is shown. Underneath the main image, there are three smaller thumbnail images of dogs. The bottom section contains a form for the user to fill in their details: 'Pittersberger Melanie', 'Vorname' (Your name...), 'Nachname' (Your last name...), and 'Nachricht' (Write something...). A green 'Submit' button is located at the bottom right of the form area.

The screenshot shows a user interface with a navigation bar at the top containing links for 'Project Manager', 'Infos', 'Mitglied werden', and 'Anmelden'. A green banner below the navigation bar displays the message 'Anmeldung war erfolgreich!'. Below this, there is a 'Filter' section with checkboxes for categories like 'Haushalt', 'Kinder', 'Tiere', 'Schule', 'Garten', and 'Urlaub'. A search bar and a 'Clear Filter' button are also present. Two service offers are listed in boxes: 'Rasenmähen' (with an image of a red lawnmower) and 'Babysitten'.

## 5 Produktdaten

Benutzerdaten:

- BenutzerID (eindeutig), ist die "Kundennummer" eines Mitgliedes
- Kennung
  - E-Mail (eindeutig)
  - Passwort (verschlüsselt)
- Persönliche Daten
  - Informationen zur eigenen Person
    - Vorname
    - Nachname
  - Kontaktinformationen
    - Straße
    - Hausnummer
    - Postleitzahl
    - Ort



- Telefon
- E-Mail
- Passwort (Hashcode)
- Inserate
  - Sichtbarkeit (check)
  - Titel
  - Kategorie
  - Preis in Talenten
  - Beschreibung
  - Startdatum
  - Enddatum

## 6 Produktleistungen

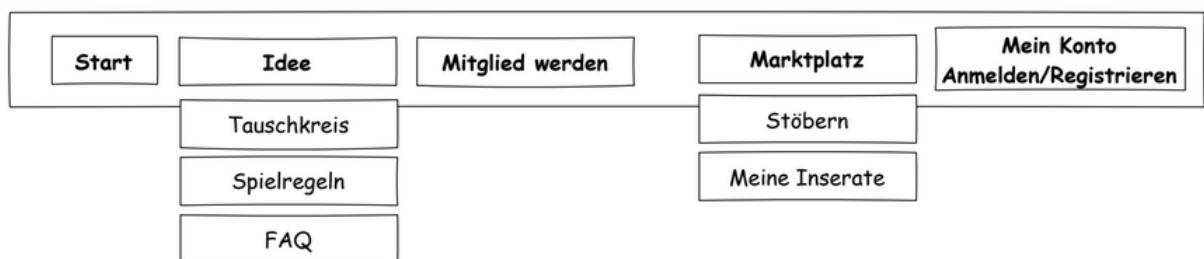
**Talente:** Die Talente müssen berechnet werden

Bereitstellung von Angeboten, um daraus potenziell entstehende Tauschgeschäfte bzw. Verkäufe (Bezahlung mit virtueller Währung) abwickeln zu können.

Abwicklung von Verkauf und Kauf bzw. Tauschgeschäften.

## 7 Benutzungsoberfläche

### 7.1 Menüstruktur





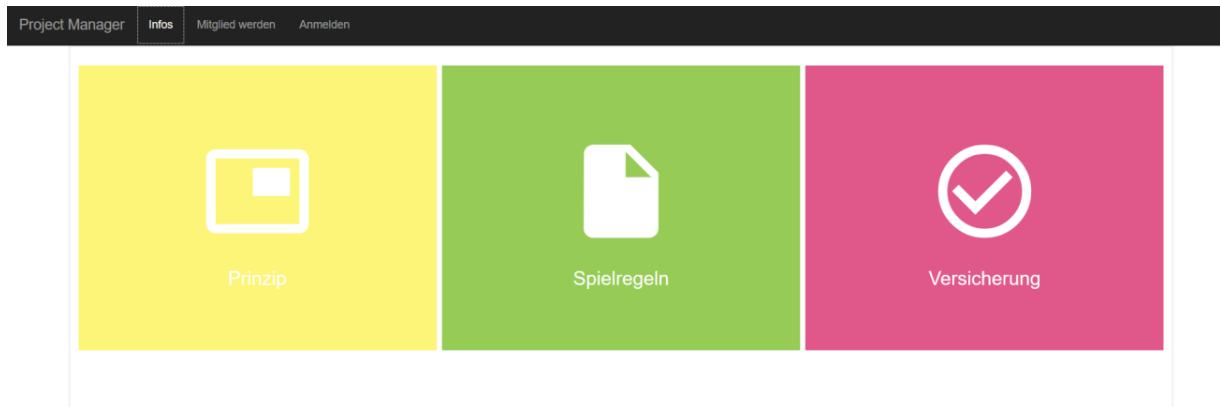
## 7.2 Bildschirmlayout

Das Layout sowie das Design wird mit Hilfe der Softwareframeworks Angular (<https://angular.io/>) gemacht und ist über das gesamte System konsistent bzw. einheitlich.

Die Farben sind gelb, grün, pink.

- `rgb : (252, 238, 33);`
- `rgb : (140, 198, 68);`
- `rgb : (212, 20, 90);`

Außerdem wird ein bestehendes Logo etwas abgeändert und verwendet.



**Funktionsprinzip**

Wie funktioniert der Tauschkreis?

Getauscht wird **ohne** Geld, es werden keine Euros benötigt um trotzdem etwas zu bekommen. Die Währung heißt **TALENTE** und wird in Euros gemessen. Diese Talente werden jenem auf ein Konto gutgeschrieben der etwas gegeben hat und dem der etwas bekommen hat, abgezogen. Jeder der bei diesem Tauschkreis dabei ist, kann mit jedem (kreuz & quer) tauschen oder Dienstleistungen in Anspruch nehmen (Z.B. Gräber gießen, Babysitten; Marmelade, Pflanzenstecklinge, Haushaltsgeräte tauschen; Nachhilfe, Besuche machen; usw.)

Beispiel:

*Frau Ypsilon geht gerne Gräber gießen und gibt dies als Inserat bekannt. Herr Lix kann auf Grund seiner Operation das Familiengrab zurzeit nicht betreuen, deshalb ruft er Frau Ypsilon an und die beiden machen sich aus, wie und wann das Grab zu gießen ist. Daraufhin werden drei Buchungsbelege ausgefüllt. Einen Buchungsbeleg bekommt Frau Ypsilon, einen Herr Lix und einen Frau Maria Scheuringer. Frau Maria Scheuringer bucht nun den ausgemachten Betrag auf das Konto von Fr. Ypsilon und von Hr. Lix wird der Betrag abgebucht. Herr Lix wiederum kann sehr gut zuhören und hat das als Inserat angegeben. Herr Zett ist bettlägerig und möchte gerne manchmal Besuch bekommen. Er kontaktiert Herrn Lix und beide machen eine Zeit aus, wann Herr Lix auf Besuch kommen kann. Nun wieder dasselbe Verfahren. Es werden wieder drei Belege ausgefüllt. Herrn Lix werden die Talente nun auf sein Konto gutgeschrieben und von Herrn Zett abgezogen. Herr Zett hat reiche Obstternte, kann dieses weder ernnten noch verbrauchen, Frau Alpha hat kein Obst und nimmt Herrn Zett das überschüssige Obst ab.*

Erneut derselbe Vorgang wie oben. Es wird kreuz & quer getauscht, je nach Belieben. Gerade „ehrenamtliche“ unbezahlte Tätigkeiten werden durch das TALENTE - Kontoguthaben enorm aufgewertet.

## 8 Qualitätsbestimmung

	sehr wichtig	wichtig	weniger wichtig	unwichtig
Robustheit		x		
Zuverlässigkeit	x			
Korrektheit	x			
Benutzerfreundlichkeit	x			
Effizienz	x			
Portierbarkeit			x	
Kompatibilität			x	

## 9 Globale Testszenarien und Testfälle

Jede Produktfunktion wird anhand von konkreten Testfällen getestet.

### **Registrieren:**

Herr Max Mustermann registriert sich mit der gewünschten E-Mail-Adresse [max.mustermann@gmail.com](mailto:max.mustermann@gmail.com) und dem Passwort root am System. Außerdem füllt er die Pflichtfelder Vorname und Nachname aus.

Frau Maria Musterfrau registriert sich ebenfalls mit der gewünschten E-Mail-Adresse [maria.musterfrau@gmail.com](mailto:maria.musterfrau@gmail.com) und dem Passwort maria123 am System. Außerdem füllt sie die Pflichtfelder Vorname und Nachname aus.

### **Anmelden:**

Max Mustermann meldet sich am System unter Benutzung seiner E-Mail-Adresse und seines Passwortes am System an.

### **Abmelden:**

Max Mustermann meldet sich vom System wieder ab.

### **Kennung anfordern:**

Maria Musterfrau hat ihre Kennung vergessen und fordert unter Angabe ihrer E-Mail-Adresse ihre Kennung an.

### **Inserat erstellen:**

Maria Musterfrau möchte ein Inserat erstellen. Sie muss den Titel, die Beschreibung, Start- und Enddatum und ein Bild angeben.

**Titel:** Nachhilfe

**Beschreibung:** Ich suche jemanden, der meinem Kind Nachhilfe in dem Fach Mathematik gibt. Mein Kind besucht im Moment die 5. Klasse eines Gymnasiums

**Startdatum:** 01.05.2018

**Enddatum:** 31.10.2018

**Bild:****Anfrage schicken:**

Antonia Kriegner möchte eine Anfrage für das Inserat von Maria Musterfrau schicken. Dazu klickt sie auf das Inserat und verfasst eine kurze Nachricht.

Kriegner Antonia

<b>Titel</b>	Rasenmähen
<b>Name</b>	Kriegner, Antonia
<b>Nachricht</b>	Write something..

**Submit**

**Titel:** Rasenmähen (automatisch gesetzt)

**Name:** Kriegner, Antonia (automatisch gesetzt)

**Nachricht:** Hallo Maria!

Ich hätte in dem von dir genannten Zeitraum, Zeit deinen Rasen zu mähen. Ich liebe es draußen im Garten zu sein und es würde mich freuen, wenn ich dir helfen kann.

Mit freundlichen Grüßen

Antonia Kriegner



## 10 Entwicklungsumgebung

Es wird darauf geachtet, dass alle Entwicklungstools kostenlos sind.

Plattform:

- Server: Java
- Datenbank: MariaDB
- OS: Windows

Tools:

- Netbeans
- Visual Studio Code

Browser:

- Client: Angular
- Mozilla Firefox 61.0.1
- Google Chrome 67.0

## 11 Ergänzungen

### 12 Glossar

**Talente** ist eine virtuelle Währung, welche zum Kaufen von Produkten bzw. Dienstleistungen verwendet wird.

**Kennung** ist das Passwort und die E-Mail-Adresse des Benutzers, er braucht diese, um sich anmelden zu können.

**Konfiguration** Jeder einzelne Benutzer kann seine Nutzungsoberfläche individuell gestalten. Diese Einstellung wird als persönliche Konfiguration bezeichnet.



**öffentlich** Unter öffentlich versteht man die Lesbarkeit nur innerhalb der Spielgemeinschaft des Systems, soweit nicht näher beschrieben.

**registrieren** Erstmaliges Anmelden eines beliebigen Internet-Benutzers, für den noch keine Kennung für das System vorliegt.

**Tauschgemeinschaft** Die Menge aller Benutzer des Systems.

**Loginseite** ist die Seite, die angezeigt wird, wenn ein beliebiger Internet-Benutzer auf den Internetdienst gelangt.

**System** ist ein Synonym für die Tauschbörse