

Leo-Survey

DIPLOMARBEIT

verfasst im Rahmen der

Reife- und Diplomprüfung

an der

Höheren Abteilung für Medientechnik

Eingereicht von:

Christine Raffeiner
Nina Weissengruber

Betreuer:

Thomas Stütz

Leonding, April 2022

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

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

Die vorliegende Diplomarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Leonding, April 2022

C. Raffeiner, N. Weissengruber

Zur Verbesserung der Lesbarkeit wurde in diesem Dokument auf eine geschlechtsneutrale Ausdrucksweise verzichtet. Alle verwendeten Formulierungen richten sich jedoch an beide Geschlechter.

Danksagung

Unser Dank gilt vor allem unserem Betreuer Herr Prof. Stütz ohne dessen mentale und technische Unterstützung unsere Diplomarbeit sicher nicht fertig geworden wäre.

Ebenfalls möchten wir unseren Dank an Dorfinger Jonas, Scholl Sebastian, Primetzhofer Vanessa, Wallinger Marc und Sljivic Emina aussprechen.

Vielen Dank auch an all unsere Professoren und Professorinnen die uns erfolgreich durch die HTL begleitet haben.

Zum Schluss möchten wir uns bei all jenen bedanken, die sich für das Korrekturlesen Zeit genommen haben.

Zusammenfassung

Im Rahmen der vorliegenden Arbeit wurde ein Online-Fragebogen implementiert, da die Befragungen von Personengruppen im Rahmen von Home-Office und Distance-Learning zugenommen haben und auch zusätzlich, um die Umwelt zu schonen.



Die vorliegende Arbeit ist speziell für die Schüler und Schülerinnen und Lehrer und Lehrerinnen der HTL Leonding ausgerichtet und beinhaltet das Erstellen, Beantworten und Auswerten von Fragebögen. Mitunter war es ein Anliegen, dass oft verwendete Fragebögen schon vorgefertigt sind und die Benutzer und Benutzerinnen frei über ihre Daten verfügen können.

Im Verlauf dieser Arbeit werden aufgetretene Probleme erklärt und deren Lösungsansätze näher beschrieben. Weiters werden die verwendeten Technologien und der Ablauf der Arbeit näher erläutert und die Erfahrungen, die im Laufe der Arbeit gesammelt wurden, beschrieben.

Summary

In the context of the present work, an online questionnaire was implemented, since the surveys of groups of people, in the context of Home-Office and Distance Learning has increased and also additionally to reduce the environmental pollution.

The present work is especially designed for the students and teachers of the HTL Leonding and includes the creation, answering and evaluation of questionnaires. It was a concern that often used questionnaires are already prepared and that the users can freely access their data.

In the course of this work, encountered problems are explained and their solutions are described in more detail. Furthermore, the technologies used and the process of the work are explained in more detail and the experience gained in the course of the work is described.



Inhaltsverzeichnis

1. Einleitung	1
2. Ausgangssituation	2
3. Umfeldanalyse	3
3.1. Vorgänger Projekte	3
3.2. Online Fragebogenwerkzeuge	3
4. Problembeschreibung	10
5. Aufgabenstellung	11
6. Ziele	13
7. Systemarchitektur	14
7.1. Aktivitätsdiagramm	14
7.2. Klassendiagramm und Entity-Relationship-Modell	15
8. Technologien	21
8.1. Angular	21
8.2. Docker	29
8.3. Keycloak	33
8.4. Traefik	37
8.5. PlantUML	38
8.6. Markdown	38
8.7. Java	39
8.8. Quarkus	40
8.9. PostgreSQL	42
8.10. Rest-API	42
8.11. Software-Testing	47

9. Tools	51
9.1. Adobe XD	51
9.2. Oracle SQL-Developer	51
10. Ausgewählte Aspekte	53
10.1. Besonders gut gelöste Programmteile	53
10.2. Besondere Probleme, die gelöst wurden	67
10.3. Entwurfsentscheidungen	71
11. Implementierung	73
11.1. Endpoints	73
12. Projektorganisation	88
12.1. Anforderungen	88
12.2. Mögliche Hilfsmittel	88
12.3. Problemlösung	91
13. ScreenDesign	96
13.1. Handskizzen	96
13.2. Screendesign Prototyp	98
14. Ergebnis	103
14.1. Zusammenfassung	103
14.2. Frontend	103
15. Resümee	107
Anhang	108
A. Teamvorstellung	109
B. Wichtige Lektionen	111
C. Protokolle	112
D. Meilensteinliste und -analyse	127
E. Arbeitsverteilung	129
Literaturverzeichnis	131

Abbildungsverzeichnis	138
Tabellenverzeichnis	140
Quellcodeverzeichnis	141

1. Einleitung

Befragungen von Personengruppen werden oft benötigt, besonders Onlinebefragungen werden im Rahmen von Home-Office und Distancelearning verwendet. Wie zum Beispiel an unserer Schule, der HTL-Leonding, finden Feedbackbögen Verwendung. Bestimmte Lehrer verwenden immer noch Fragebögen aus Papier. Dies ist nicht nur schlecht für die Umwelt sondern auch sehr umständlich und zeitaufwändig in der anschließenden Auswertung.

Deshalb war das Ziel der Diplomarbeit unter anderem das einfache Gestalten von Fragebögen mit verschiedenen Fragetypen, sowie das Durchführen und Auswerten von bereits erstellten Befragungen.

2. Ausgangssituation

Feedback ist in allen Bereichen ein wichtiger Teil. Vor allem bei Dienstleistungen möchte man Feedback über die erbrachten Leistungen bekommen, um zu wissen, in welchen Bereichen man Gutes geleistet hat und in welchen man noch Verbesserungsbedarf hat. Auch Lehrer sind Dienstleister und ihr Unterricht ist ihre Dienstleistung. Diese Lehrer bekommen auch gerne eine Rückmeldung von den Schülern, welche Aspekte sie in ihrem Unterricht gut umgesetzt haben und welche Verbesserungsvorschläge es für die Zukunft gibt.

Zurzeit besteht für Lehrer die Möglichkeit, ihr Feedback über ein kommerzielle Angebote des Ministeriums einzuholen. Eine weitere Möglichkeit besteht darin, das Feedback mit Fragebögen auf Papier einzuholen.

3. Umfeldanalyse

3.1. Vorgänger Projekte

Die zwei folgenden Vorgängerprojekte wurden für die vorliegende Arbeit als Inspirationsquelle verwendet.

1. QuestionZ [1]
2. Leo-survey [2]

QuestionZ ist ein Projekt bestehend aus Frontend, Backend und Datenbank und somit der Systemarchitektur der vorliegenden Arbeit sehr ähnlich. Das Frontend von QuestionZ wurde jedoch mit JavaServerFaces (JSF) implementiert und war im Betrieb nicht stabil. Obwohl JSF immer noch weiterentwickelt wird, ist es für Neuprojekte oft nicht die erste Wahl, insbesondere wenn keine JavaEE bzw. JakartaEE-Backend verwendet wird. Das Backend von QuestionZ wurde in JavaEE mit einem Application Server umgesetzt. JavaEE existiert mittlerweile nicht mehr und wurde von JakartaEE abgelöst. [1]

Leo-survey, der Namensvetter der Diplomarbeit, besteht im Gegensatz dazu aus einem Angular Frontend und einem Java Backend. Für die Datenbank wurde Derby gewählt. [2]

Leo-survey wurde mit Programmiersprachen implementiert, mit denen das Team vertraut war. Grundlegende Ideen zur Herangehensweise der Implementierung einiger Funktionen im Backend und Ideen für Layout und Design für die Entwicklung des Frontends wurden aus den Vorgängerprojekten geschöpft. Aus dem Projekt QuestionZ entstand die erste stabile Version des Klassendiagramms.

3.2. Online Fragebogenwerkzeuge

Für eine Konkurrenzanalyse und Überlegungen, wie die vorliegende Arbeit von anderen Produkten abzugrenzen ist, wurden eine Reihe von Online-Fragebogen-Webseiten getestet und analysiert. Weiters wurden interessante Funktionen und ansprechende

Designideen gesammelt. Dabei wurde beachtet, dass die Arbeit hauptsächlich für den schulinternen Gebrauch gedacht ist.

Dies ist eine Auflistung der Webseiten, die analysiert wurden:

- 2ask.com [3]
- appinio.com [4]
- ingress.com [5]
- questionpro.com [6]
- survio.com [7]
- umfrageonline.com [8]

3.2.1. 2ask.com

Dieses Tool bietet ähnliche Funktionen zu den Anforderungen der Arbeit, weshalb sie als Referenzobjekt sehr geeignet ist. Besonderes Augenmerk wurde auf die Funktionen *Fragebogen erstellen*, *Umfragen durchführen* und *Ergebnisse auswerten* gelegt. Aufgefallen sind dabei die Erstellung eines QR-Codes, um Umfragen auszufüllen, der Datenexport der Auswertung in ein CSV-File und die Visualisierung der Auswertung mit verschiedenen Diagrammtypen. Negativ anzumerken ist, dass dieses Tool sehr viele Fragetypen besitzt, die die Erstellung eines Fragebogens verkomplizieren. [3]

Vorhandene Fragetypen sind:

- Beschreibung
- Textfeld
- Formular
- Multiple-Choice
- Matrix
- Gruppen-Matrix
- Schieberegler (Sterne)
- Rangliste (Hierarchie)
- Konstantsummen

Was muss an Ihrem Arbeitsplatz verbessert werden?

Mehrfachantworten sind möglich.

Belüftung
 Mehr Tageslicht
 Bessere Beleuchtung
 Kantine / Pausenzimmer
 Ausstattung Arbeitsplatz
 Sonstiges

0 von 1070

Abbildung 1.: Multiple-Choice-Frage von [3]

Von Interesse für designtechnische Aspekte ist die Angabe von sonstigen Antwortoptionen (siehe Abb. 1) in Multiple-Choice-Fragen. Dieses Funktion deckt die Möglichkeit ab, dass keine der angegebenen Optionen für den Benutzer oder die Benutzerin in Frage kommt und optional eigene Antwortmöglichkeiten verwendet werden können.

Ein tolles Feature für die Implementierung in die vorliegende Arbeit wäre die automatische Konvertierung sonstiger Optionen in vordefinierte Antwortmöglichkeiten. Umsetzen könnte man diese Konvertierung mittels algorithmischer Berechnungen.

3.2.2. appinio.com

Bei der Analyse dieses Tools stellte sich heraus, dass es sehr viele Funktionen beinhaltet. Die Auswertung liefert sehr ausführliche und mit vielen Details gespickte Informationen. Weiters ergab sich, dass das Tool für den Rahmen der Arbeit zu aufwendig gestaltet ist.

< Zurück zur Übersicht

Neue Umfrage erstellen

1. Zielgruppe definieren 2. Fragen erstellen 3. Zusammenfassung

1. Einzel-/Mehrfachauswahl

Frage
Wie häufig trinkst Du Kaffee?

+ Anweisung für Teilnehmer hinzufügen + Bild hinzufügen

Täglich
Mehrmals wöchentlich

Antwort C (max. 50 Zeichen)
Antwort D (max. 50 Zeichen)

+ Antwort hinzufügen + Freitext-Antwort hinzufügen

Mehrere Antworten möglich Randomisieren

Kostenübersicht

Preis pro Teilnehmer
€ 1.00

Anzahl der Teilnehmer
600

Basispreis
€ 1.00

Gesamtpreis zzgl. Mwst.
€ 600

Unsere Zielgruppe

Geschlecht
Männer, Frauen

Alter
14 bis 55+ Jahre

Abbildung 2.: Beispiel für die Erstellung eines Fragebogen von [4]

Das Design (siehe Abb. 2) des Prozesses der Erstellung von Fragebögen ist sehr übersichtlich und optisch ansprechend umgesetzt. [4]

3.2.3. ingress.com

Das Tool Ingress deckt den Umfang der Arbeit mit seinen Funktionen ab. Mit ihm lassen sich Fragebögen erstellen, beantworten und auswerten. Während des Beantwortungsprozesses stellt Ingress ein Dashboard zur Verfügung, um den derzeitigen Stand der Umfrage zu überwachen.

PROGRESS 42%

How long are you going to stay at the Research Club Networking party?

less than 30 min

30 minutes - 1 hour

1 - 2 hours

2 - 3 hours

3 hour or more

not sure

BACK **CONTINUE ...**

Abbildung 3.: Fortschrittsanzeige bei der Beachtung der Fragebögen von [5]

Hervorzuheben bei diesem Tool ist die Fortschrittsanzeige, die während der Beantwortung von Umfragen (siehe Abb. 3) angezeigt wird. Sie gibt den Benutzern eine direkte Rückmeldung über den Fortschritt und den Restaufwand, der betrieben werden muss, um das Ausfüllen des Fragebogens zu beenden. [5]

3.2.4. questionpro.com

Dieses Tool zeichnet sich durch das Analysieren der Daten in Echtzeit, die große Auswahl an Auswertungsdiagrammen und die resultierende genaue Auswertung aus.



Abbildung 4.: Fortschrittsanzeige bei der Beachtung der Fragebögen und Antwortoptio-nendesign von [6]

Dieses Tool bietet eine Fortschrittsanzeige, um Benutzerinnen und Benutzern eine direkte Rückmeldung über ihren derzeitigen Fortschritt zu geben. (siehe Abb. 4). Designtechnisch ist das Layout der Antwortoptionen von Interesse, da Optionen anders als Vergleichsobjekte nebeneinander statt untereinander angezeigt werden. Die Möglichkeit, diese Optionen nebeneinander anzuzeigen, ist nur dann sinnvoll, wenn die Optionen für jede Frage unverändert bleiben. Sollte das nicht der Fall sein, kann dadurch die Übersicht verloren gehen, da über jeder Antwortmöglichkeit ein anderer Text steht.

[6]

3.2.5. survio.com

Bei diesem Tool war besonders das Herunterladen der Ergebnisse der Umfragen in vielen Formaten, beispielweise: PDF, DOCX, XLSX, CSV einprägsam.

Designtechnisch wurde auf die Verwendung von Radiobuttons verzichtet und stattdessen eine Auswahloption verwendet. Diese Option bietet dieselbe Funktionalität wie Radiobuttons, der / die Benutzer/in erkennt jedoch nicht auf den ersten Blick, welche Antwortmöglichkeiten die Frage hat. [7]

3.2.6. umfrageonline.com

Dieses Tool wurde genauer unter die Lupe genommen und es folgen die Erkenntnisse, die über das Design und die Funktionalitäten geschlossen wurden.

The screenshot shows a survey titled "Customer Satisfaction Survey". The first question is "How satisfied were you with the following:". Below it are two sections: "The way your call was answered" and "The speed your call was answered". Each section contains a list of response options with radio buttons. In the "The way your call was answered" section, the "Very Satisfied" option is selected (indicated by a checked checkbox). In the "The speed your call was answered" section, the "Very Unsatisfied" option is selected.

Section	Options
The way your call was answered	Very Unsatisfied, Unsatisfied, Satisfied, Very Satisfied, Can't say
The speed your call was answered	Very Unsatisfied

Abbildung 5.: Visuelle Trennung der Fragen von den Antwortoptionen von [8]

Das Tool zeigt (siehe Abb. 5) eine visuelle Abtrennung von Fragen und Antwortmöglichkeiten, die hilft, nicht den Überblick bei einer großen Anzahl an Fragen und Antwortmöglichkeiten zu verlieren. Zudem dient es dazu, den Blick des Benutzers auf die Fragen zu fokussieren.

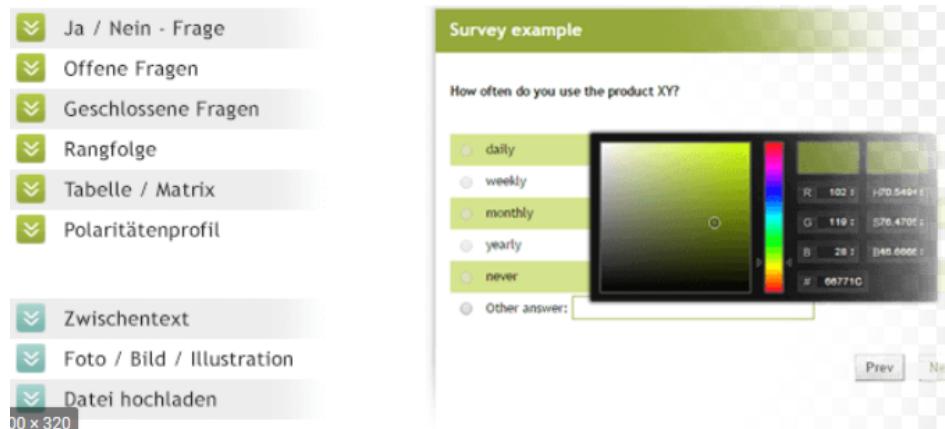


Abbildung 6.: Anpassung und Dateien zu einer Frage von [8]

Bilder oder Dateien, die zum Beantworten einer Frage nötig sind, zu einer Frage abzuspeichern, ist eine Funktionalität, die ein Fragebogentool von einem Anderen unterscheiden und abheben kann. Zudem ist die Funktion praktisch, um wichtige Informationen zur Frage bereitzustellen. (siehe Abb. 6 als Beispiel). Den Umfang der Arbeit betrachtend, ist die in der Abbildung gezeigte Anpassung von Farben zu jedem Fragebogen zu aufwendig. Weiters wurden die Fragetypen analysiert. Die Implementierung aller Typen würde den Umfang der Arbeit überschreiten.

Customer Satisfaction Survey

0 %

How satisfied were you with the following:

Very Unsatisfied Unsatisfied Satisfied Very Satisfied Can't say

The way your call was answered	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
The speed your call was answered	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
The answer to your query	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Accuracy of the given information	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

Abbildung 7.: Antwortoptionendesign von [8]

Dieses Tool bietet die Möglichkeit Antwortoptionen nebeneinander darzustellen. (siehe Abb. 7). [8], [9]

4. Problembeschreibung

Bei bereits existierenden Plattformen ist nicht sichergestellt, was mit den jeweiligen Daten passiert. Die Frage besteht darin, was mit den Daten geschieht, wie sie verwaltet werden oder ob sie an Dritte weiter gegeben werden. Außerdem ist die Anonymität nicht sicher gestellt, da unsicher ist, ob irgendwelche persönlichen Daten zu den Personen, die die Umfrage durchführen, gesammelt oder an andere weitergeleitet werden.

5. Aufgabenstellung

Die Aufgabenstellung der vorliegenden Arbeit, ist das Erstellen einer Fragebogen Plattform, bei der sichergestellt wird, dass die erhobenen Daten unter der Kontrolle des Fragenstellers bleiben, um so sämtliche datenschutzrechtlichen Aspekte zu erfüllen

In folgender Abbildung sieht man das Use-Case Diagramm für unsere Arbeit.

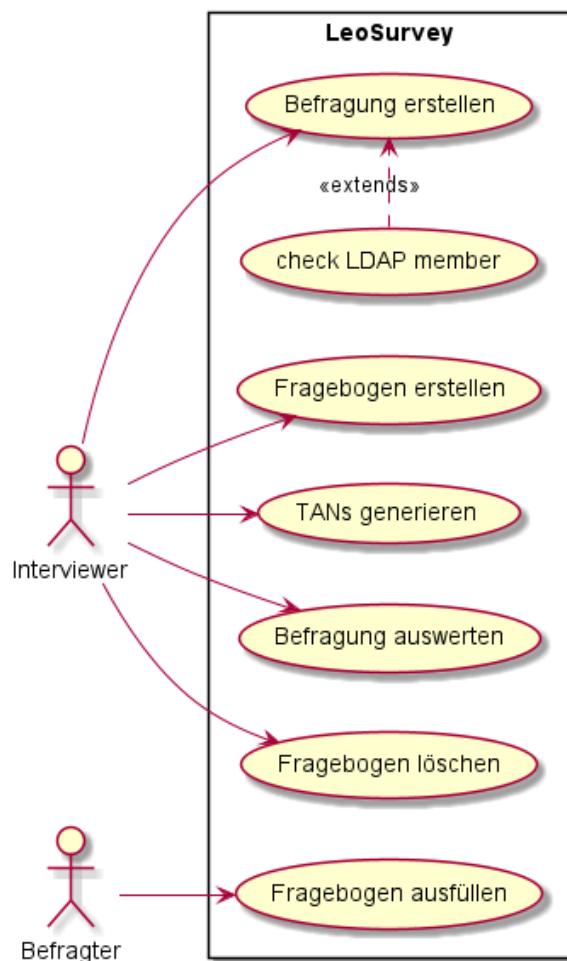


Abbildung 8.: Use Case Diagramm

Zunächst ist ein Fragebogen mit seinen Fragen zu erstellen. Anschließend wird eine Umfrage geplant (Bezeichnung der Umfrage, ev. Zeitraum der Umfrage, Teilnehmer der Umfrage). Danach werden Transaktionscodes (TAN) erstellt, um den Befragten eine anonyme Teilnahme an der Umfrage zu ermöglichen. Schließlich werden diese

TANS an die Befragten übermittelt, woraufhin diese die Fragen des Fragebogens beantworten.

6. Ziele

Das Ziel dieser Diplomarbeit ist es, Personen eine simple und schnelle Möglichkeit zu bieten, um Umfragen zu erstellen. Dabei kann die Beantwortung der Fragen anonym ablaufen. Im Nachhinein sind die Daten leicht und schnell auswertbar.

7. Systemarchitektur

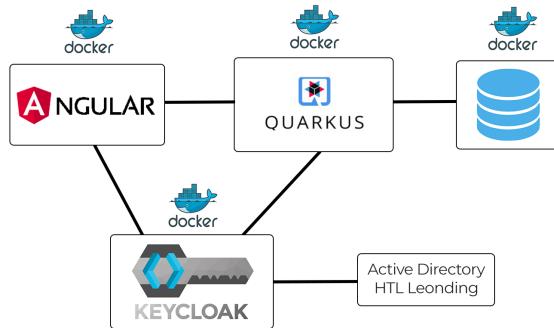


Abbildung 9.: Systemarchitektur

In Abbildung 9 sieht man die Systemarchitektur. Für diese Anwendung wurde für das Backend Quarkus verwendet, das auf eine PostgreSQL Datenbank zugreift. Für das Frontend wurde Angular verwendet. Außerdem wurde ein Keycloak, der Zugriff auf das Active Directory der HTL Leonding hat und mit Frontend und Backend kommuniziert, eingebaut.

7.1. Aktivitätsdiagramm

Ein Aktivitätsdiagramm stellt eine Reihe von dynamischen Beziehungen in einem System dar.

Abbildung 10 ist das Aktivitätsdiagramm für diese Arbeit. Beim Start des Programms gibt es als erstes die Entscheidung, ob man sich einloggen will oder nicht. Möchte man sich nicht einloggen, kann man gleich die Befragung ausfüllen. Entscheidet man sich zum einloggen, steht man vor der Entscheidung, ob man einen bereits erstellten Fragebogen verwenden möchte. Wenn ja, kann man eine Befragung erstellen. Möchte man keinen vorgefertigten Fragebogen verwenden, kann man selbst einen Fragebogen erstellen. Danach kann eine Frage erstellt und im Zuge dessen kann eine Antwortmöglichkeit/können Antwortmöglichkeiten erstellt werden. Danach ist es möglich, eine Befragung zu erstellen, um anschließend Transactions Codes zu erstellen. Danach kann man die Befragung ausfüllen und daraus die Befragung auswerten.

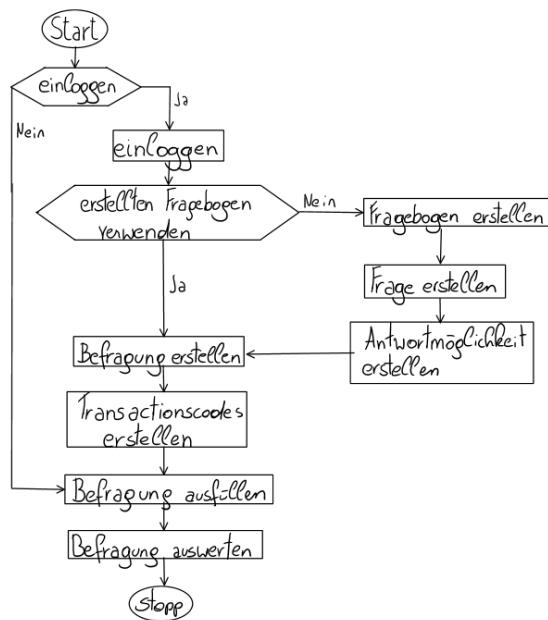


Abbildung 10.: Aktivitätsdiagramm

7.2. Klassendiagramm und Entity-Relationship-Modell

Beim Start der Arbeit wurden im Zuge der Implementierung ein Klassendiagramm und ein Entity-Relationship-Modell erstellt.

7.2.1. Unterschied

Bei beiden Diagrammtypen werden die Datenstrukturen und ihre Beziehungen zueinander abgebildet. Während jedoch Klassendiagramme dem objektorientierten Paradigma entsprechen (mit Collections, Objektidentität, usw.), bilden ERDs die in den Datenbanktabellen gespeicherten Datenstrukturen und deren Beziehungen ab. ERDs entsprechen dem relationalem Paradigma. (inkl. Erfüllung der Normalformen, des Schlüsselprinzips, usw.)

Die erste Version der Diagramme wurden mithilfe des SQL-Developers erstellt. Mehr Informationen zum SQL-Developer befinden sich im Kapitel 9.2. [10], [11], [12]

7.2.2. Erste Version

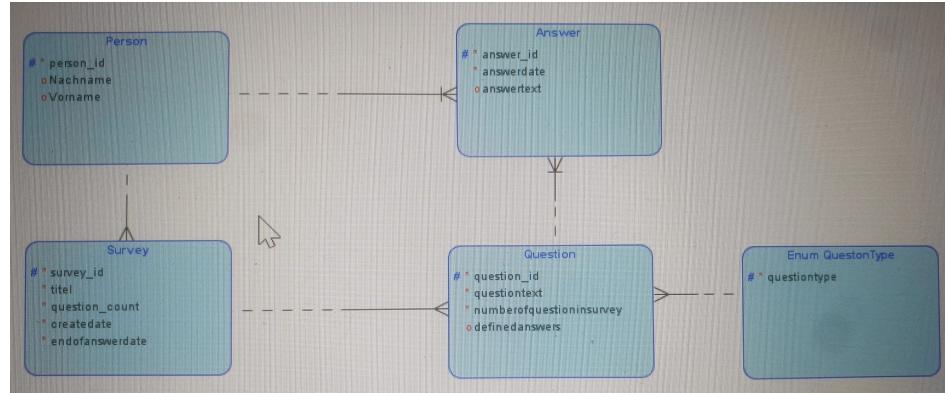


Abbildung 11.: Erste Version des Datenmodелles (ERD)

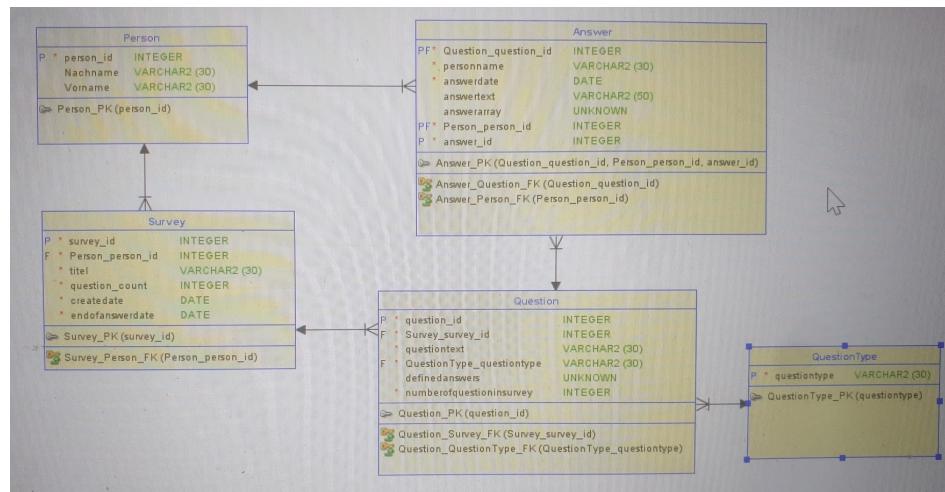


Abbildung 12.: Erste Version des Datenmodell (CLD)

Das Datenmodell (siehe Abb. 11 und 12) bildet eine einfache Art Fragebögen, zugehörige Befragungen und Antworten zu speichern, ab. Für die leichtere Implementierung der grundlegenden Funktionen wurde in der ersten Version die Klasse Person noch zur Implementierung vorgesehen. Diese sollte in einer Endversion jedoch entfernt und durch die Anbindung an den Keycloak ersetzt werden.

Das Problem dieses Datenmodells bestand darin, dass nicht alle Anforderungen der Arbeit durch das Datenmodell abgebildet werden konnten und es auch nicht normalisiert wurde.

Die erste Normalform wurde in dem Modell (siehe Abb. 11 und 12) durch das Attribut definedanswers missachtet. Es war vorgesehen, alle möglichen Antworten in einem Attribut durch Trennzeichen abgetrennt zu speichern.

Es wurde dagegen entschieden, das Datenmodell zu überarbeiten und stattdessen ein

neues Datenmodell von einem anderen Projekt als Vorlage zu nehmen. (siehe Kapitel 3).

7.2.3. Normalisierung

Unter Datenbanknormalisierung versteht man den Vorgang, Daten innerhalb einer Datenbank so effizient wie möglich zu organisieren. Bei der Verwendung einer relationalen Datenbank kann die Normalisierung dazu beitragen, die Daten fehlerfrei zu halten und sicherzustellen, dass die Größe der Datenbank nicht durch doppelte Daten vergrößert wird. [13]

1. Normalform

Die Erste Normalform (1NF) ist dann gegeben, wenn alle Informationen in einer Tabelle atomar vorliegen. [14]

7.2.4. Umstieg zu Markdown und PlantUML

Alle weiteren Versionen des Datenmodells wurden mit PlantUML erstellt. Der Hauptgrund für den Wechsel bestand darin, dass die Software IntelliJ IDEA, die zum Programmieren des Backends verwendet wurde, diese Diagramme erstellen und anzeigen kann. Diese können in Markdown eingebettet oder in eigenständigen Dateien mit einem Plugin gebildet werden. Die Teammitglieder waren mit PlantUML vertraut, was den Wechsel zusätzlich bestärkte.

Dateien mit Markdown-formatiertem Text können mit vielen Anwendungen geöffnet werden. Markdown-Dateien können einfach in andere Markdown-Anwendungen importiert werden. Mehr Informationen über Markdown und PlantUML befinden sich im Kapitel 8.5 und 8.6.

7.2.5. Zweite Version

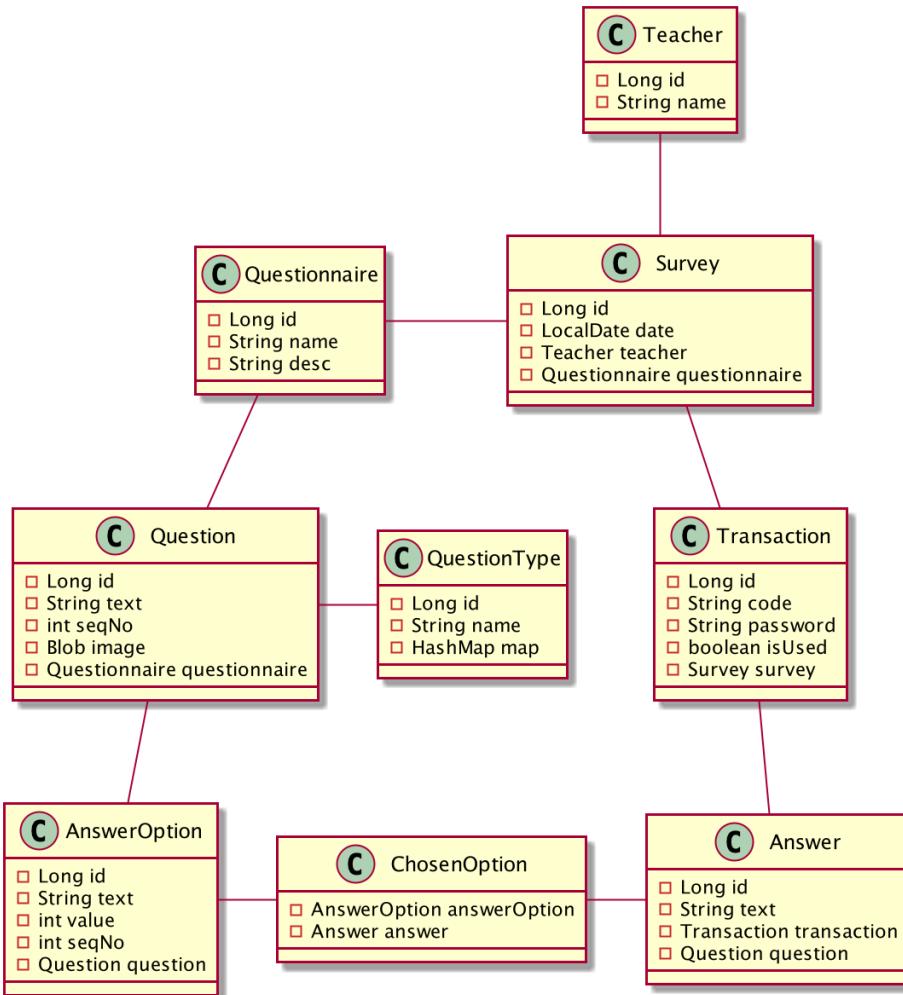


Abbildung 13.: Zweite Version des Datenmodells

Die zweite Version des Datenmodells (siehe Abb. 13) ist auf dem Modell eines Vorgängerprojektes aufgebaut (siehe Kapitel 3). Erweiterungen wurden im Bereich Beantwortung der Befragung und Auswertung getätigt. Mit den Erweiterungen konnten nun Single-Choice, Multiple-Choice und Text-Antworten besser abgebildet werden. Zudem wurde das Modell um die Funktion der optionalen Speicherung eines Bildes zu Fragen ergänzt.

Questionnaire stellt den Fragebogen und Survey die Befragungen dar. Es ergibt sich, dass für einen Fragebogen mehrere Befragungen gleichzeitig erstellt werden können. Die Transaktionscodes stehen stellvertretend für den/die Benutzer/in und garantieren die Anonymität der Person, die den Fragebogen ausfüllt.

7.2.6. Stabile Version

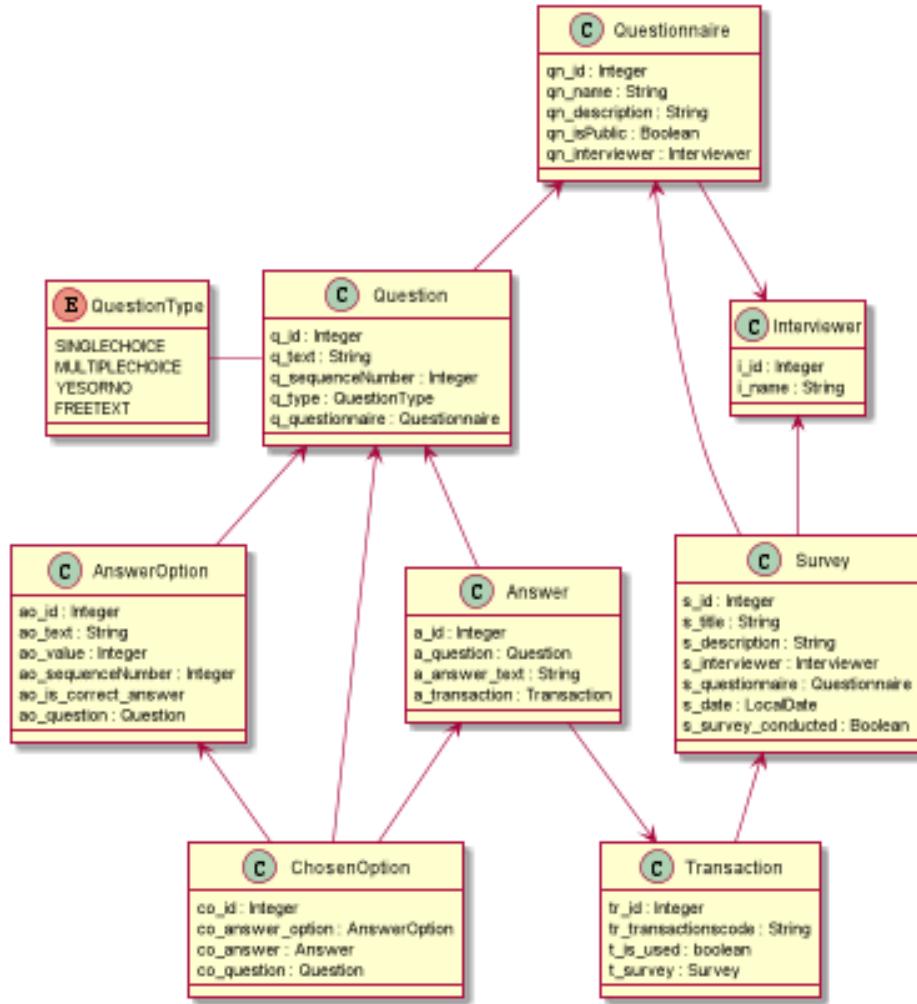


Abbildung 14.: Zwischenversion des Datenmodells

Im Laufe der Implementierung sind weitere Änderungen am Klassendiagramm vorgenommen worden (siehe Abb. 14), um den wachsenden Anforderungen gerecht zu werden. Dadurch ist beispielsweise ein zusätzliches Attribut im Fragebogen hinzugefügt worden (`is_public`). Dieses Attribut stellt sicher, dass jede/r Benutzer/in der Applikation Fragebögen öffentlich zur Verfügung stellen kann. Sichtbar sind alle öffentlichen und privaten Fragebögen des Benutzers.

Der Befragung wurde um das Attribut `interviewer` erweitert, um anderen Benutzern die Möglichkeit zu geben, öffentliche Fragebögen auszufüllen. In der letzten signifikanten Änderung wurde das Attribut `survey_conducted` hinzugefügt, dass besagt, ob und wie lange Personen den Fragebogen ausfüllen können. Dies dient dazu, den Fragebogen als Leistungsfeststellung zu verwenden.

7.2.7. Endversion

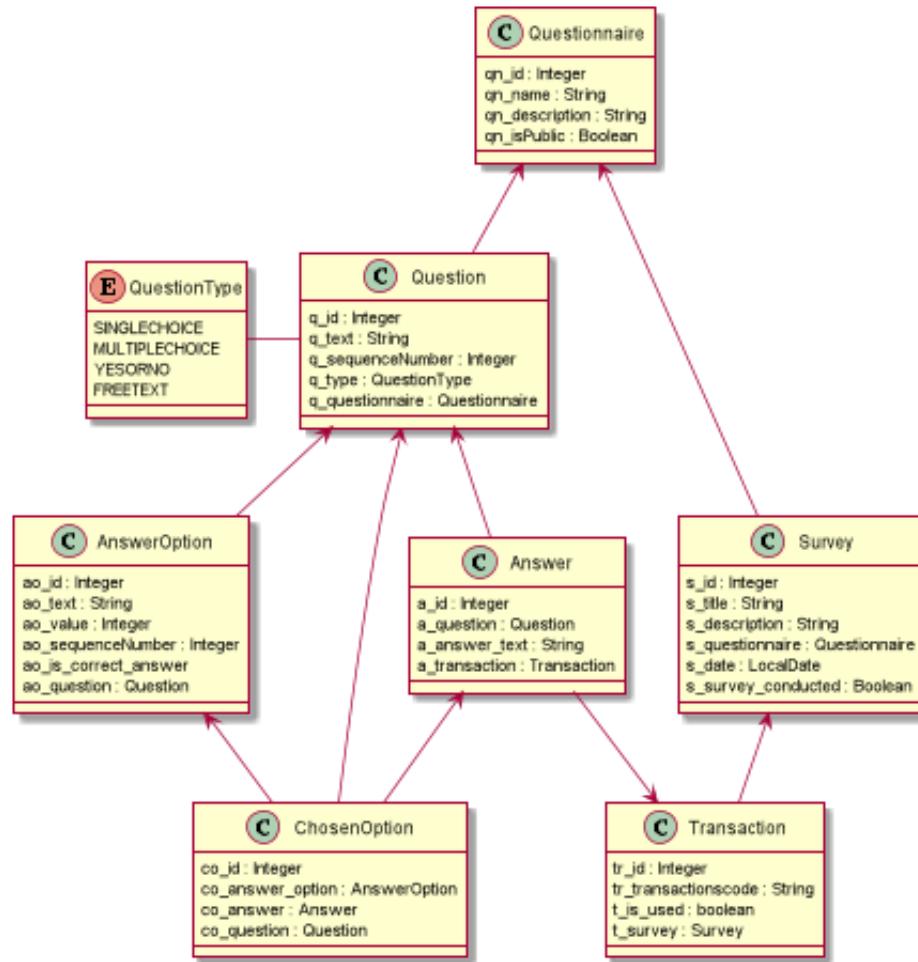


Abbildung 15.: Letzte Version des Datenmodells

Mit der Implementierung des Keycloak entfiel die Klasse Person. Die Speicherung der Nutzer/innen wurde durch den Keycloak übernommen. (siehe Abb. 15)

8. Technologien

8.1. Angular

Angular ist ein TypeScript-basiertes, clientseitiges Webapplikations- und JavaScript-Framework was bedeutet, dass es für die Entwicklung von Webseiten und Webapplikationen ausgetragen ist. Es unterstützt die Verwendung des MVC(Model-View-Controller)-Pattern und vereinfacht die Erstellung und das Layoutieren von Single-Page-Applications (SPA) mit TypeScript, HTML und einer Formatierungssprache. Mögliche Formatierungssprache sind CSS, SCSS SASS und Less.



Abbildung 16.: Angular Logo von <https://angular.io>

Angular unterscheidet im Prinzip zwischen zwei Arten von Modulen. Zum Einen, das App-Modul, welches das Root-Module beinhaltet und als erstes Modul der Anwendung geladen und initialisiert wird und zum Anderen weitere Feature-Module, die sogenannten Components, Services und andere notwendige Dateien, die für ein bestimmtes Feature relevant sind. Durch den Gebrauch des Model-View-Controller-Pattern von Angular wird die Wiederholung von Code mithilfe der Erstellung und Wiederverwendbarkeit von Komponenten, die statische HTML-Tags mit dynamischen Inhalten verbinden, verhindert.

Ein großer Vorteil von Angular ist es, dass es im Gegensatz zu anderen Frameworks über die Funktion einer bidirektionalen Verbindung verfügt. Das heißt, dass die Änderung eines Wertes einer Variable in einem Textfeld im HTML ebenfalls Auswirkungen auf die Variable im TypeScript hat und vice versa. Weitere Vorteile von Angular sind unter anderem, dass Angular ein Open Source Framework mit einer MIT Lizenz ist. Weiteres stellt Angular umfangreiche Funktionen zur Verfügung, die direkt bereitgestellt sind. Wie diese Funktionen zu implementieren sind, wird von Angular bereits vorgegeben und erspart somit die Überlegung der Implementierung. Zudem werden Zeit und fallweise Kosten gespart, da diese nicht erst selbst programmiert werden müssen.
[15], [16]

8.1.1. TypeScript

TypeScript ist eine Programmiersprache, die von Microsoft entwickelt und gewartet wird. Sie ist eine streng syntaktische Variante von JavaScript und fügt der Sprache eine optionale statische Typisierung hinzu. Es kann zur Entwicklung von JavaScript-Anwendungen sowohl für die clientseitige als auch für die serverseitige Ausführung verwendet werden.

TypeScript ist für die Entwicklung großer Anwendungen konzipiert und lässt sich in JavaScript umwandeln.

TypeScript ist keine Programmiersprache für sich allein. Es ist vielmehr eine Kombination aus Werkzeugen und optionalen, entfernbaren Typen. TypeScript ermöglicht es einem beliebigen JavaScript-Tools, -Bibliotheken und -Frameworks zu verwenden.

TypeScript bezieht sich auf mehrere verschiedene Dinge:

- einen Type-Checker für die beiden Sprachen TypeScript und JavaScript
- eine übergeordnete Sprache, die die Typensyntax zu JavaScript hinzufügt
- den offizielle Compiler, der Code typisiert und umwandelt

All diese Komponenten werden benötigt, um JavaScript eine statische Typisierung hinzuzufügen und Werkzeuge bereitzustellen, die die Verwendung einfach und komfortabel machen.

Der TypeScript-Compiler ist selbst in TypeScript geschrieben und zu JavaScript kompiliert. Er ist unter der Apache License 2.0 lizenziert. [17], [18]

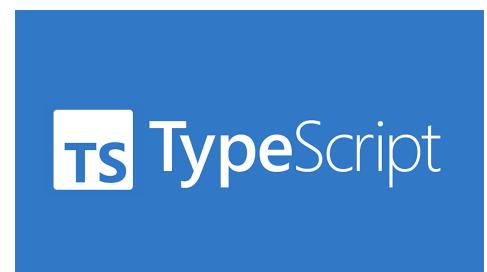


Abbildung 17.: TypeScript Logo von <https://thenewstack.io/typescript-helps-enterprise-developers/>

JavaScript

JavaScript (kurz JS) ist leichtgewichtige, interpretierte Skriptsprache. Ihre Bekanntheit hat sie hauptsächlich als Skriptsprache für Webseiten erhalten, um Benutzerinteraktionen auszuwerten, Inhalte zu verändern, nachzuladen oder zu generieren. JavaScript wird auch außerhalb von Browsern angewandt und dient ebenfalls für die Programmierung von Prototypen. Die Sprache bietet sowohl objektorientierte, aber klassenlose, imperative als auch deklarative Programmierung an. Der Standard für JavaScript ist ECMAScript. [19], [20], [21]

8.1.2. CSS

CSS auch Cascading Style Sheets ist eine Formatierungssprache für HTML-, SVG- und XML-Dokumente. Somit behandelt CSS das Design oder den Stil und nicht den Inhalt von Webseiten. Beispielsweise können damit Schriftarten, Farben, Linien, Höhen, Breiten und Positionierung auf einer Webseite definieren werden. CSS ermöglicht es ein einmal erstelltes Design schnell und einfach in ein anderes Projekt zu übertragen. CSS stellt einen Standard dar und wird vom World Wide Web Consortium (W3C) gemanagt und weiterentwickelt. Darüber hinaus wird eine responsive Darstellung von CSS unterstützt. Anders gesagt heißt dass, das mittels CSS passende Darstellungsformen für unterschiedliche Geräte angefangen von Monitoren zu Druckern bis hin zu Smartphones definiert werden können. Mittels Media Queries können in CSS auch Geräteeigenschaften Breite und Höhe des Browserfensters ausgelesen werden. Eigenschaften, die bestimmt werden können sind:

- Breite und Höhe des Gerätes
- Orientierung (Quer- oder Hochformat)
- Bildschirmauflösung

CSS wird von allen gängigen Browsern unterstützt, allerdings kann es zu Einschränkungen bzw. Fehlern bei der Darstellung des Layouts in Browsern kommen - übermäßig davon betroffen sind veraltete Browser. [22], [23], [24], [25]



Abbildung 18.: CSS Logo

World Wide Web Consortium

Beim World Wide Web Consortium (W3C) handelt es sich um ein Gremium, dass technische Spezifikationen und Richtlinien zur Standardisierung der Techniken im World Wide Web entwickelt. Andere bekannte Beispiele für Technologien, die durch das W3C standardisierte werden sind, HTML SVG und XML. [26], [27]

8.1.3. SCSS

SCSS ist die erweiterte Version von CSS. Es können Variablen definiert werden, die zur Folge haben, dass der Code verkürzt wird. [28]

8.1.4. SASS

SASS basiert auf Ruby und heißt ausgeschrieben SassScript. Es steht unter der MIT-Lizenz und gilt daher als Open Source. SASS bietet im Gegensatz zu CSS zusätzlich die Funktion von:

- Variablen (wie SCSS)
- Mathematische Funktionen wie Multiplizieren, Dividieren, Addieren und Subtrahieren (+, -, *, /)
- Funktionen
- Schleifen
- Fallunterscheidungen (wie if und else)
- Mixins (Vorlagen, die selbst erstellt oder bei der Verwendung eines Frameworks einfach in den eigenen Code eingebettet werden können)
- Vererbung (mittels Selektor)

Ein großer Nachteil von SASS ist, dass es zuerst kompiliert werden muss. Das heißt anders als in CSS, wo Änderungen in der CSS-Datei sofort Auswirkungen auf der Webseite haben, müssen in SASS die Änderungen erst in CSS übersetzt werden. Im Gegensatz dazu spricht für SASS die freie Auswahl der Syntax. Ob man nun die vorgegebene Syntax benutzt oder lieber die gewohnte SCSS- / CSS-Syntax verwendet. [29], [30], [31]

8.1.5. Less

Genau wie SCSS oder SASS ist Less eine Obermenge von CSS- und somit ebenfalls eine Stylesheet-Sprache. Vergleichsweise zu SASS ist die Verwendung von Mixins (Vorlagen), Variablen, Berechnungen und Verzweigungen. Less ist in JavaScript geschrieben und muss ebenfalls kompiliert werden. Allerdings können Mithilfe eines Watch-Mode Änderungen automatisch im Webbrowser angezeigt werden. [32], [33]

8.1.6. HTML

HTML - ausgeschrieben Hypertext Markup Language - ist eine bekannte textbasierte Auszeichnungssprache für die Anzeige und Strukturierung von Webseiten. HTML macht sich



sogenannte Tags zu Nutze, um die verschiedensten Inhalte wie Text, Bilder oder Videos anzuzeigen. Der derzeitige Standard von HTML ist XHTML (Extensible Hypertext Markup Language) und HTML5. XHTML gehört zu XML, einer Auszeichnungssprache und Dateiformat und erweitert die Auszeichnungssprache HTML.

HTML5 enthält detaillierte Verarbeitungsmodelle, um mehr kompatible Implementierungen zu fördern: Es erweitert, verbessert und vereinfacht das für Dokumente verfügbare Format und führt Formatierungs- und Anwendungsprogrammierschnittstellen (APIs) für komplexe Webanwendungen ein. HTML5 ist vermehrt für Suchmaschinen optimiert und fördert im Gegensatz zu alten Versionen userorientierte Webseiten. Zusätzlich zu den Anzeigeelementen verfügt HTML über nicht sichtbare Inhalte namens Metadaten. Diese Daten sind im normalen Gebrauch und für User nicht sichtbar. Suchmaschinen allerdings können sie lesen und nicht nur das, sondern auch ihre Suchergebnisse dahingehend anpassen. Befinden sich keine Metadaten im HTML-Code, versuchen Suchmaschinen sich etwas von der Webseite zusammenzulesen. HTML-Quellcode kann in jedem Editor geschrieben und von jedem Browser interpretiert werden. [34], [35], [36]

8.1.7. NodeJS

Node.js ist ein plattformübergreifendes Open-Source Framework, die der MIT-Lizenz unterliegt, und für die Entwicklung eigenständiger JavaScript-Programme, Netzwerktools und Webapplikationen verwendet wird. Der JavaScript-Code kann außerhalb von Webbrowsern laufen und bietet neben Netzwerk-orientierten Kommandozeilen-Tools auch Werkzeuge für die Systemadministration. Node.js wird in Googles JavaScript-Laufzeitumgebung V8, einer prozess-basierten virtuellen Maschine ausgeführt, die den Code mithilfe eines Compilers in Maschinensprache übersetzt.



Abbildung 20.: NodeJS Logo von <https://de.wikipedia.org>

Besonders für die Verwendung von Angular ist NodeJS von Bedeutung, da die geschriebene Applikation mit dem Node Package Manager, kurz npm mit beliebigen Modulen erweitert werden kann. Der Node Package Manager kann dabei allerdings nicht nur Module samt ihrer Abhängigkeiten installieren, sondern diese ebenfalls entfer-

nen, kompilieren und auf neuere oder bestimmte Version aktualisieren. Diese Prozesse werden mithilfe des Node.js-Repository zur Verfügung gestellt. [37], [38], [39]

8.1.8. Angular Materials

Die Angular Material UI ist eine Bibliothek, die es ermöglicht, verschiedene Komponenten zu importieren und zu verwenden, um Benutzeroberflächen in Angular zu erstellen. Diese importierten Komponenten können gegebenenfalls in einem vordefinierten Rahmen an die jeweiligen Anforderungen angepasst werden. Das heißt, dass in einer ausgewählten Komponente einzelne Teilbereiche zum Beispiel ausgeblendet (nicht verwendet) oder eingeblendet (verwendet und definiert) werden können. Die Verwendung von Materials spart einiges an Zeit, da man nicht alle Funktionen von Grund auf selbst programmieren muss. Zusätzlich zu den bereits implementierten Funktionen liefert Angular Materials auch Layoutierungsregeln, Darstellungsregeln und Animationen (vordefiniertes CSS beziehungsweise JavaScript). Diese Regeln können voneinander abweichen je nachdem welches Theme gewählt wird.

Themes definieren einen Satz klar definierter Darstellungsregeln. Beispielsweise wird der Hintergrund von ein und demselben Element mit unterschiedlichen Themes verschieden eingefärbt. Ein klassisches Beispiel hierfür sind der Helle und Dunkle Modus von vielen Webseiten und Programmen. Es ist allerdings auch möglich eigene Themes festzulegen. All diese Aspekte machen das Erstellen visuell ansprechender und funktional userfreundlicher Anwendungen um einiges leichter. [40], [41]

Beispiele für die in Angular Materials verwendeten Komponenten:

- Toolbar
- Checkbox
- Expansion Panel
- Form field
- Stepper
- Pageinator
- Radio button
- Tabs

- etc. [42]

8.1.9. angular-oauth2-oidc

Das Modul angular-oauth2-oidc wird für die Kommunikation mit dem Keycloak verwendet. Es bietet mehrere Funktionen, die die Kommunikation vereinfachen und erleichtern. [43]

- Lauffähig mit allen modernen Browsern und IE
- Automatisches Aktualisieren eines Tokens, wenn/einige Zeit bevor es abläuft
- Konfigurierbare Anmelde- und Abmelde-Funktion

Anmerkungen zu den Features

Die Lauffähigkeit für alle modernen Browser ist in der modernen Zeit eigentlich ein Muss, da der Benutzer die Freiheit haben sollte, seinen präferierten Browser zu nutzen. Das automatische Aktualisieren des Tokens, sollte dieser abgelaufen sein, verbessert die Userfreundlichkeit, da der Benutzer sich nicht neu einloggen muss oder anderweitig einen neuen Token anfordern muss.

Logging out Funktion

Die Logout-Methode löscht den zugewiesenen Token-Speicher (standardmäßig sessionStorage) und leitet den Benutzer an den Logout-Endpunkt des Auth-Servers weiter, falls ein solcher konfiguriert wurde.

Logging in Funktion

Die Funktion bietet die Möglichkeit den Benutzer an die angegebene URL des Identitäts-Anbieters weiterzuleiten. Nach einer erfolgreichen Anmeldung wird der Benutzer zu einer in der Konfiguration angegebenen URL auf der Webseite umgeleitet. Alternativ kann angegeben werden, dass die Anmeldung übersprungen wird.

8.1.10. angular/localize

Das Paket angular/localize enthält Hilfestellungen und Werkzeuge für die Lokalisierung einer Angular Anwendung. Die Idee dahinter besteht darin, dass Texte, die übersetzt

werden sollen, mit speziellen Tags markiert werden. Die Übersetzung selbst kann einerseits entweder zur Laufzeit im Browser erfolgen oder andererseits durch Markierungen im Code und einem statischen Nachbearbeitungswerkzeug, der den Originaltext durch übersetzten Text ersetzt, bevor der Code bereitgestellt wird. Für jede Sprache wird eine eigene Ausgangssprachdatei bereitgestellt, die Schlüssel-Wert-Paare mit Nachrichtenbezeichnern als Schlüssel und lokalisierten Nachrichten als Werte enthält. Für das Format der Datei kann XLIFF 1.2 (Standart), XLIFF 2 oder in XML-Message Bundle (XMB) verwendet werden. Die Ausgangssprachdatei muss als Identifikation der Sprache mit einem Sprachkürzel (ISO 639-2) versehen werden um die Sprache, das Land zu spezifizieren. [44], [45], [46], [47], [48], [49]

Beispiele für Sprachkürzel

- en
- en_Us
- fr
- de

XLIFF 1.2 und XLIFF 2

XLIFF (XML Localization Interchange File Format) ist ein XML-basiertes Bitext-Format, das zur Standardisierung der Übermittlung von Lokalisierungsdaten zwischen verschiedenen Tools während eines Lokalisierungsprozesses entwickelt wurde. Im Vergleich zu XLIFF 1.2 hat XLIFF 2.0 eine anders organisierte DOM-Struktur und andere Anwendung der Modularität. [50], [51]

XML Message Bundle

Der länderspezifische Text wurde in separate XML-Dateien extrahiert. Diese werden lose als XML-Ressourcenbündel bezeichnet und je nach Gebietsschema abgerufen und durchsucht. [52], [53], [54]

8.2. Docker

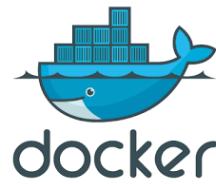


Abbildung 21.: CSS Logo von <https://www.cloudflight.io/de/blog/docker-container-die-zukunft-moderner-applikationen-und-multi-cloud-deployments/>

Docker ist eine Open-Source Softwareplattform für die Erstellung, Lieferung und Ausführung von Anwendungen. [55] Docker teilt sich in die Bestandteile:

- Docker-Container
- Dockerfile
- Container-Images
- Docker run
- Docker Hub
- Docker Engine
- Docker Compose
- Docker Desktop.

8.2.1. Docker Container

Docker Container sind kleine und leichtgewichtige Ausführungsumgebungen, die den Kernel des Betriebssystems gemeinsam nutzen, ansonsten aber isoliert voneinander laufen und über ein eigenes Dateisystem verfügen. Durch die Perfektionierung dieses Prinzips hat Docker sich schnell zu einem De-facto-Industriestandard für Container entwickelt.

Docker-Container ermöglichen Kompositionsfähigkeit. Container erleichtern es den Entwicklern, die Bausteine einer Anwendung zu einer modularen Einheit mit leicht austauschbaren Teilen zusammenzustellen, was Entwicklungszyklen, Funktionsfreigaben und Fehlerbehebungen beschleunigen kann. Docker-Container sind zustandslos und unveränderlich. Container booten und laufen von einem Image, dass ihren Inhalt beschreibt. Dieses Image ist standardmäßig unveränderlich - einmal erstellt, ändert es sich

nicht mehr. Eine Container-Instanz ist jedoch vergänglich. Wenn der Container aus dem Systemspeicher entfernt wird, ist er für immer verschwunden. [56], [57], [58], [59]

Unterschied zu Virtuellen Maschinen

Eine virtuelle Maschine (VM) ist die Virtualisierung / Emulation eines Computersystems. Virtuelle Maschinen bieten die Funktionalität eines physischen Computers. Sie sind isoliert vom Rechner, auf dem die virtuelle Maschinen laufen. Jede VM benötigt ihr eigenes Betriebssystem, was bedeutet, dass sie in der Regel viel Speicherplatz benötigen und daher langsam starten, schwierig zu bewegen und umständlich zu warten und zu aktualisieren sind. [60], [61]

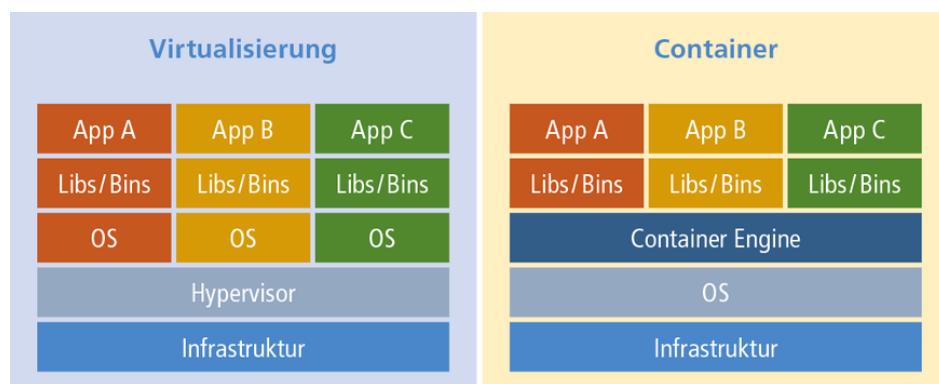


Abbildung 22.: Unterschied von Virtuellen Maschinen und Containern von <https://www.shd-online.de/fachartikel/der-app-store-fuer-server-wofuer-sie-container-brauchen/>

8.2.2. Dockerfile

Jeder Docker-Container beginnt mit einem Dockerfile. Diese Textdatei enthält eine Reihe von Anweisungen zur Erstellung eines Docker-Abbilds, einschließlich des Betriebssystems, der Sprachen, der Umgebungsvariablen, der Dateispeicherorte, der Netzwerk-Ports und aller anderen Komponenten, die zur Ausführung benötigt werden.

```

FROM registry.access.redhat.com/ubi8/ubi-minimal:8.4

ARG JAVA_PACKAGE=openjdk-headless
ARG RUN_JAVA_VERSION=1.3.8
ENV LANG='en_US.UTF-8' LANGUAGE='en_US:en'
# Install java and the run-java script
# Also set up permissions for user '1001'
RUN microdnf install curl ca-certificates ${JAVA_PACKAGE} \
    && microdnf update \
    && microdnf clean all \
    && mkdir /deployments \
    && chown 1001 /deployments \
    && chmod "g+rwx" /deployments \
    && chown 1001:root /deployments \
    && curl https://repo1.maven.org/maven2/io/fabric8/run-java-sh/${RUN_JAVA_VERSION}/run-java-sh-${RUN_JAVA_VERSION}-sh.sh -o /deployments/run-java.sh \
    && chown 1001 /deployments/run-java.sh \
    && chmod 540 /deployments/run-java.sh \
    && echo "securerandom.source=file:/dev/urandom" >> /etc/alternatives/jre/conf/security/java.security

# Configure the JAVA_OPTIONS, you can add -XshowSettings:vm to also display the heap size.
ENV JAVA_OPTIONS="-Dquarkus.http.host=0.0.0.0 -Djava.util.logging.manager=org.jboss.logmanager.LogManager"
# We make four distinct layers so if there are application changes the library layers can be re-used
COPY --chown=1001 target/quarkus-app/lib/ /deployments/lib/
COPY --chown=1001 target/quarkus-app/*.jar /deployments/
COPY --chown=1001 target/quarkus-app/app/ /deployments/app/
COPY --chown=1001 target/quarkus-app/quarkus/ /deployments/quarkus/

EXPOSE 8080
USER 1001

ENTRYPOINT [ "/deployments/run-java.sh" ]

```

Abbildung 23.: Beispiel eines Dockerfiles

8.2.3. Container-Images

Ein Docker-Image ist eine andere Möglichkeit einen Docker-Container zu erstellen. Ein Docker-Image ist eine portable, schreibgeschützte, ausführbare Datei, die die Anweisungen zur Erstellung eines Containers und die Spezifikationen dafür enthält, welche Softwarekomponenten der Container wie ausführen wird. Jeder Container ist eine Instanz eines Images, wobei mehrere Instanzen desselben Images gleichzeitig ausgeführt werden können. Diese Cluster von Containern müssen dann orchestriert werden, wozu in der Regel Kubernetes eingesetzt wird. [62]

Kubernetes

Kubernetes ist eine flexible, erweiterbare Open-Source-Plattform für die Verwaltung von containerisierten Systemen und Diensten, die sowohl eine deklarative Konfiguration als auch eine Automatisierung ermöglicht. [63], [64]

- Automatisierte Terminplanung
- Automatisierte Rollouts und Rollbacks
- Horizontale Skalierung und Lastausgleich
- Bietet eine konsistente Umgebung für Entwicklung, Tests und Produktion
- Automatisch skalierbare Infrastruktur

8.2.4. Docker run

Docker run ist der Befehl von Docker, mit dem ein Container gestartet wird. Es können Spezifikationen zum Beispiel für den Namen des Containers und den Port, auf dem der Container laufen soll, getroffen werden.

8.2.5. Docker Hub

Docker Hub ist ein Repository, in dem Container-Images gespeichert, freigegeben und verwaltet werden können. Es ist mit den GitHub-Repository für GitHub zu vergleichen, nur werden im Docker Hub Images zur Verfügung gestellt. Startet man einen Container eines Images wird mit den Standardeinstellungen dieses Image automatisch auf Docker Hub gesucht und wenn gefunden, heruntergeladen. [65]

8.2.6. Docker-Engine

Die Docker-Engine ist die zugrunde liegende Client-Server-Technologie, die Container erstellt und ausführt. Die Docker-Engine umfasst einen langlaufenden Daemon-Prozess zur Verwaltung von Containern, APIs, die es Programmen ermöglicht, mit dem Docker-Daemon zu kommunizieren, sowie eine Befehlszeilenschnittstelle bildet.

8.2.7. Docker Compose

Docker Compose ist ein Tool, das YAML-Dateien verwendet, um Multicontainer-Docker-Anwendungen zu definieren und auszuführen. Es ermöglicht Ihnen, alle Dienste Ihrer Konfiguration zu erstellen, zu starten, zu stoppen und neu zu erstellen und den Status und die Protokollausgabe aller laufenden Dienste anzuzeigen.

YAML

YAML ist eine für Sprache zur Daten-Serialisierung. Sie wird üblicherweise für Konfigurationsdateien und in Anwendungen verwendet, in denen Daten gespeichert oder übertragen werden. YAML verwendet Schlüssel-Wert-Paaren. [66], [67]

```

version: 3.7

services:
  backend:
    build: .
    ports:
      - 80:8080
    restart: unless-stopped
    depends_on:
      - postgres
      - keycloak

  keycloak:
    container_name: keycloak
    environment:
      - KEYCLOAK_USER=admin
      - KEYCLOAK_PASSWORD=admin
      - DB_VENDOR=h2
    ports:
      - '8180:8080'
    image: 'quay.io/keycloak/keycloak:15.0.2'
    volumes:
      - ./src/main/docker/certs:/etc/ssl/certs/java
      - ./src/main/docker/imports:/opt/jboss/keycloak/imports
  postgres:
    image: postgres:13
    volumes:
      - postgres-data:/var/lib/postgresql/data
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: postgres
      POSTGRES_DB: db

volumes:
  postgres-data:

```

Abbildung 24.: Beispiel für Docker Compose YAML-Datei

8.2.8. Docker Desktop

Docker Desktop vereint all diese Komponenten in einer Anwendung, die eine benutzerfreundliche Möglichkeit zur Erstellung und gemeinsamen Nutzung von containerisierten Anwendungen und Microservices bietet. [68], [69]

8.3. Keycloak

Keycloak ist ein Open-Source-Identitäts- und Zugriffsmanagement-Tool mit dem Schwerpunkt auf modernen Anwendungen



KEYCLOAK

33

Abbildung 25.: Keycloak Logo

wie Single-Page-Anwendungen, mobilen Anwendungen und REST-APIs. Mit der Verwendung von Keycloak authentifizieren sich Benutzer mittels des Keycloak und nicht mit individuellen Anwendungen. Das bedeutet, dass keine Login-Formulare, Authentifizierung von Benutzern und Speicherung von Benutzern implementiert werden müssen. Diese Login-Seite kann sogar angepasst werden. Nach einmaligen Einloggen bei Keycloak, müssen sich die Benutzer nicht erneut anmelden, um auf eine andere Anwendung zuzugreifen. Dasselbe gilt auch für die Abmeldung. Mit der Verwendung von Single-Sign-Out müssen sich Benutzer nur einmal abmelden um von allen Anwendungen, die den Keycloak benutzen, abgemeldet zu werden.

Eine zusätzliche Funktion des Keycloak ist die eingebaute Unterstützung, um sich mittels LDAP mit bestehenden Active-Directory-Servern zu verbinden. Docker stellt ein Image für Keycloak zur Verfügung. [70], [71], [72], [73], [74]

8.3.1. LDAP

LDAP oder ausgeschrieben Lightweight Directory Access Protocol ist ein Softwareprotokoll, das Daten speichert und sortiert, um sie leicht auffindbar zu machen. Bei den Daten kann es sich um beliebige Informationen über Geräte oder Benutzer handeln, die in Verzeichnissen gespeichert sind. LDAP ist das Protokoll, das von Servern verwendet wird, um mit den Verzeichnissen vor Ort zu kommunizieren.

Der Hauptnutzen von LDAP besteht darin, als zentraler Mittelpunkt für die Authentifizierung und Autorisierung zu dienen. LDAP hilft dabei Benutzernamen und Kennwörter zu speichern und später wieder abrufbar zu machen, zum Beispiel wenn ein Benutzer versucht, auf eine LDAP-fähige Anwendung zuzugreifen. Mithilfe der in LDAP gespeicherten Anmeldeinformationen wird der Benutzer authentifiziert.

In LDAP können auch Benutzerattribute gespeichert werden, die bestimmen, worauf der Benutzer zugreifen darf. Obwohl LDAP und Active-Directory (AD) häufig synonym verwendet werden, handelt es sich um zwei verschiedene Arten von Software, die jedoch zusammenarbeiten können. [75], [76],

8.3.2. Active-Directory

Active-Directory (AD) ist eine Datenbank und bietet eine Reihe von Diensten, die Benutzer mit den Netzwerkressourcen verbinden. Die Datenbank beziehungsweise das Directory enthält wichtige Informationen über Ihre Umgebung, zum Beispiel welche Benutzer und Computer es gibt und wer was tun darf. [77], [78], [79], [80]

Active-Directories bestehen aus drei Hauptebenen:

1. Domänen
2. Bäume
3. Wälder

Eine Domäne ist eine Gruppe zusammengehöriger Benutzer, Computer und anderer Objekte. Mehrere Domänen können zu einem Baum (tree) zusammengefasst werden, und mehrere Bäume können zu einer Gesamtstruktur (auch Forest genannt) gruppiert werden.

8.3.3. Login Prozess

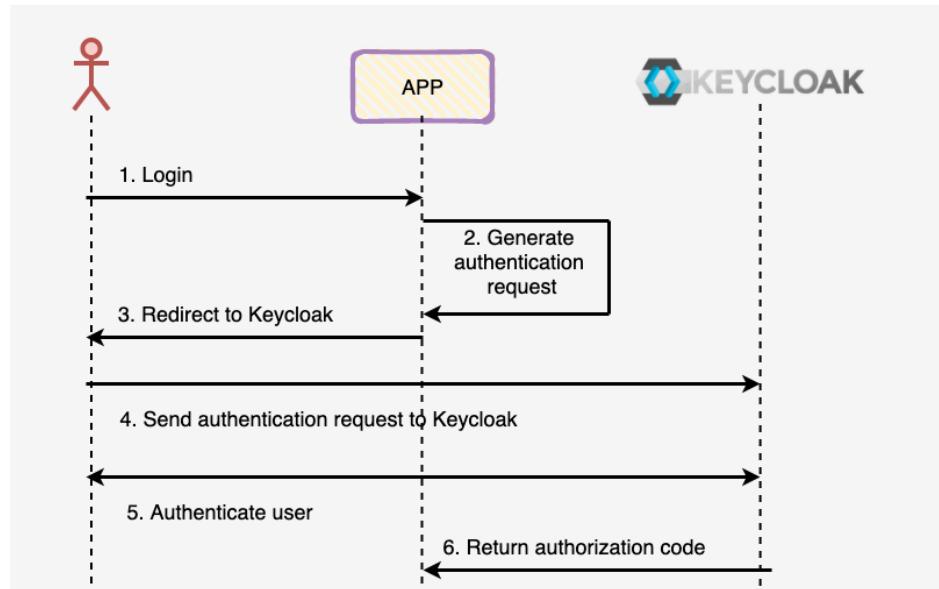


Abbildung 26.: Login Prozess von <https://medium.com/codex/introduction-to-keycloak-227c3902754a>

In der Abbildung (siehe Abb. 26) wird der Login-Prozess des KeyCloak vereinfachend dargestellt.

1. Der Benutzer klickt auf einen Login-Button

2. Die Anwendung generiert eine Authentifizierungsanfrage.
3. Die Authentifizierungsanforderung wird an den Benutzer gesendet.
4. Der Keycloak zeigt dem Benutzer die Anmeldeseite an. Der Benutzer gibt seinen Benutzernamen und sein Passwort ein und schickt das Formular ab
5. Keycloak überprüft die Anmelde Daten und erstellt daraufhin einen Autorisierungscode, der an die Anwendung zurückgeschickt wird
6. Autorisierungscode wird gegen das ID-Token und das Refresh-Token ausgetauscht

JSON-Web-Token (JWT)

Der Keycloak sendet standardmäßig ein signiertes JSON-Web-Token (JWT). JSON Web Token (JWT) ist ein offener Standard, der eine kompakte Methode zur sicheren Übertragung von Informationen zwischen Parteien in Form eines JSON-Objekts ermöglicht. Es wird hauptsächlich für Authentifizierung und Informationsaustausch verwendet. Diese Informationen können überprüft werden und sind vertrauenswürdig, da sie digital signiert werden. JWTs können aber auch verschlüsselt werden, um die Geheimhaltung zwischen den Parteien zu gewährleisten. [81], [82], [83]

8.3.4. Admin Console

Über die Admin Konsole können alle Aspekte des Keycloak-Servers zentral verwaltet werden. Die verschiedene Einstellungen können hier getroffen und Funktionen aktiviert und deaktiviert werden. Weiters können Identitäts-Brokering und Benutzer-Föderation konfiguriert werden. Zudem besteht die Möglichkeit Anwendungen und Dienste zu erstellen und zu verwalten und feinkörnige Autorisierungsrichtlinien zu definieren. Einstellungen von Benutzer, einschließlich Berechtigungen und Sitzungen können ebenfalls getroffen werden. [84], [85]

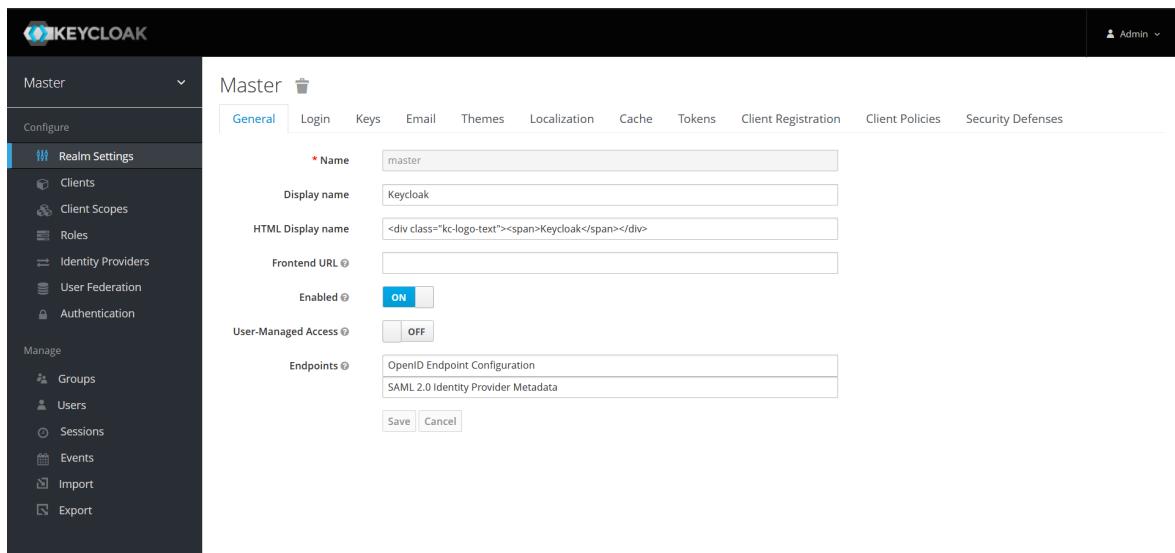


Abbildung 27.: Admin Console

8.3.5. Keycloak Konto-Konsole

Die Konsole ist für Benutzer. Dort können sie ihr Konto verwalten, z.B. ihr Profil und ihr Passwort aktualisieren.

8.4. Traefik

Traefik ist ein Open Source Edge Router, der die Veröffentlichung von Diensten vereinfacht. Der Router empfängt Anfragen stellvertretend für das System und findet heraus, welche Komponenten für deren Bearbeitung zuständig sind. Traefik ist kompatibel mit allen wichtigen Cluster-Technologien wie Kubernetes, Docker, Docker Swarm, AWS, Mesos und Marathon.

Was Traefik neben seinen vielen Funktionen auszeichnet, ist, dass es automatisch Dienste im Netzwerk erkennt, auch wenn diese nachträglich gestartet werden. Traefik inspiziert die Infrastruktur des Systems und findet heraus, welcher Dienst für welche Anfragen zuständig ist und kann diese in Echtzeit abfangen und anpassen (z.B. https auf http). Traefik benötigt keine separate Konfigurationsdatei um das System zu verwalten und zu synchronisieren: Alles geschieht automatisch und in Echtzeit ohne Neustarts und Verbindungsunterbrechungen. [86], [87]

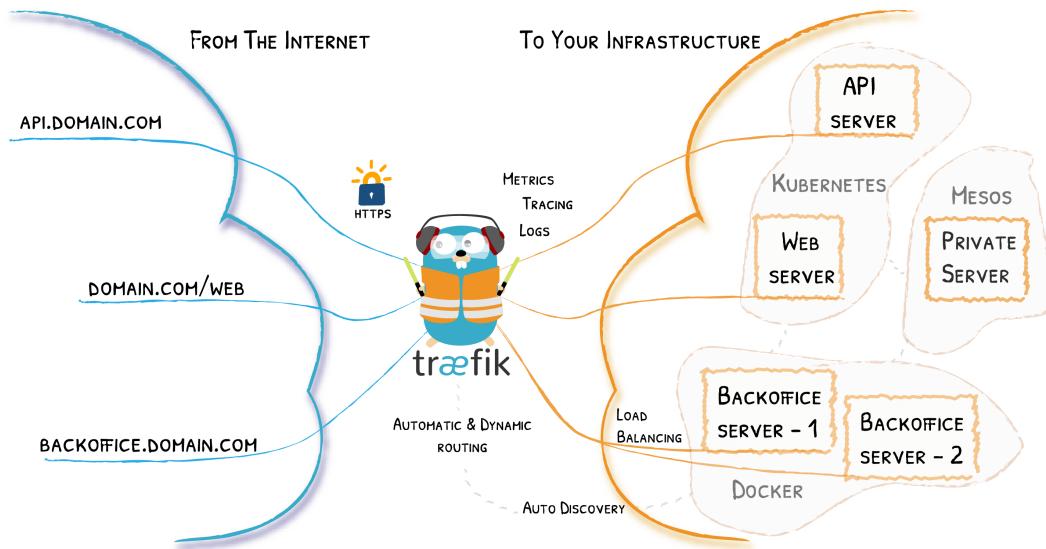


Abbildung 28.: Funktionsweise Traefik

8.5. PlantUML



Abbildung 29.: PlantUML Logo source: [88]

PlantUML ist ein Open-Source-Werkzeug, mit dem Benutzer/innen Diagramme aus einer einfachen Textsprache erstellen können. Neben verschiedenen UML-Diagrammen unterstützt PlantUML auch verschiedene andere Formate für die Softwareentwicklung. [89]

8.5.1. PlantUML integration

Das Plugin PlantUML integration wird von IntelliJ verwendet um die PlantUML-Dateien anzuzeigen und zu erstellen. Das Plugin bietet Code-Navigation und Hervorhebung. [88]

8.6. Markdown

Markdown ist eine leichtgewichtige Auszeichnungssprache, mit der Formatierungselemente zu Textdokumenten im Klartext hinzufügen werden können. [90], [91]



Abbildung 30.: Markdown Logo source: [90]

8.7. Java

Java ist eine objektorientierte Programmiersprache, die in vielen Teilen der Informatik verwendet wird. Java als Programmiersprache dient vor allem zum Formulieren von Programmen. Anfangs liegen die Programme in Quellcode, also in für Menschen verständlichen Text vor. Mithilfe des Java-Compilers wird der Quellcode in für Maschinen verständlichen Bytecode übersetzt. Dieser Quellcode wird dann über die Java Virtual Machine (JVM) ausgeführt.

[?]



Abbildung 31.: Java Logo

[?]

Ein großer Vorteil von Java ist die Plattformunabhängigkeit.

Zum Ausführen der Programme oder Anwendungen wird kein bestimmtes Betriebssystem benötigt sondern lediglich die Java-Laufzeitumgebung (JRE). Diese Laufzeitumgebung ist bereits auf vielen Computern mit verschiedenen Betriebssystemen vorinstalliert, wodurch die Plattformunabhängigkeit gegeben ist. Die Plattformunabhängigkeit wird auch durch die Java Virtuell Machine (JVM) realisiert. [?]

8.7.1. Java Runtime Environment

Die Java Runtime Environment (JRE) oder auch Java Laufzeitumgebung oder Java Runtime ist Teil des Java Development Kits (JDK).

Eine Laufzeitumgebung verhält sich wie ein kleines Betriebssystem und stellt alle notwendigen Funktionalitäten für die Ausführung eines Programmes bereit. Die Laufzeitumgebung lädt Anwendungen und lässt diese auf einer Plattform laufen, auf der alle notwendigen Ressourcen für einen betriebssystemunabhängigen Betrieb zu Verfügung.

Eine laufende Anwendung interagiert über ein Runtime System mit der Laufzeit-

umgebung. Die Laufzeitumgebung wiederum dient als Vermittler zwischen Anwendung und Betriebssystem. Die JRE stellt verschiedene Funktionen für Speicher, Netzwerk und Hardware bereit. Diese Funktionen werden von der Laufzeitumgebung ausgeführt und nicht von der Anwendung und arbeiten unabhängig vom Betriebssystem.

Ein großer Vorteil der Laufzeitumgebung ist, dass Programme Zugriff auf alle benötigten Funktionen haben, jedoch unabhängig von Betriebssystem arbeiten. [?]

8.7.2. Java Development Kit

Das Java Development Kit (JDK) bildet die Grundlage, auf der Anwendungen aufgebaut werden. JDK besteht aus verschiedenen Softwarekomponenten und enthält eine Vielzahl von Tools und Dienstprogrammen, mit denen verschiedene Aufgaben ausgeführt werden können. Die Hauptverwendung von JDK besteht darin, Code von Java-Code in Bytecode zu kompilieren, wobei die JRE verwendet wird, um den Bytecode auszuführen. [?]

8.7.3. Java Virtual Machine

Die Java Virtual Machine (JVM) ist die zentrale Komponente der Java-Laufzeitumgebung (JRE). Die JVM ermöglicht die plattformunabhängige Ausführung von Programmen im Java-Bytecode. Die JVM ist eine prozessbasierte VM oder auch ein einfacher virtueller Computer, in dem Java-Bytecode ausgeführt wird. Die VM übersetzt die Anweisungen des Bytecodes zur Laufzeit in Maschinencode.

Ein weiterer Vorteil der JVM ist die Geschwindigkeit und Sicherheit. Da die Java-Programme von den Betriebssystemen abgeschottet sind, wird die Sicherheit erhöht, da Zugriffe auf Ressourcen sich genau kontrollieren lassen. [?]

8.8. Quarkus

Quarkus ist ein Java Framework für JVMs (Java Virtual Machines) und native Kompilierung, um Java-Anwendungen für Container und Clouds zu optimieren. [?]

Quarkus unterstützt 2 Modi:



QUARKUS

Abbildung 32.: Quarkus Logo [?]

1. Optimierung des Bytecodes und Ausführen in der JVM

Java Code, der mit Quarkus geschrieben wurde, kann auf der JVM ausgeführt werden. Es gibt jedoch Vorteile in Hinsicht auf den Speicherverbrauch und die Startzeit der laufenden Anwendung. Um dies zu ermöglichen, schiebt Quarkus eine Reihe von zeitaufwendigen Schritten in den Build-Prozess. Unter anderem:

- Laden und Parsen von Konfigurationen
- Scannen des Java-Klassen-Pfades und das Auflösen von Annotationen
- gegebenenfalls das Erstellen von Entitäten-Modellen für Datenankten

Quarkus führt diese Schritte einmalig durch und speichert die Ergebnisse für einen schnellen Abruf zwischen.

Unter anderem reduziert Quarkus die Menge der zur Laufzeit dynamisch vorliegenden Informationen. Diese Informationen werden durch entsprechende Konstrukte ersetzt, was im Hinblick auf den Einsatz mit Containern sinnvoll ist.

2. Ausführen als nativer Code nach Kompilierung

Mit der Ahead-of-time-compilation (AOT) wird aus dem Java-Quelltext kein Bytecode erzeugt, sondern direkt ausführbarer Maschinencode. Dies bedeutet, dass auf der Ziel-Hardware keine JVM benötigt wird.

[?]

Ahead-of-time compilation

Bei der Ahead-of-time (AOT) compilation wird der Programmcode bereits zur Compi lezeit/Kompilierzeit in Maschinensprache übersetzt.

Mithilfe der AOT compilation wird die Startzeit der JVM von Java-Programmen verbessert. [?]

8.9. PostgreSQL

PostgreSQL ist ein leistungsstarkes, objektrelationales Open-Source-Datenbanksystem. PostgreSQL basiert auf dem Client-Server-Modell, dies bedeutet, dass ein am Server laufender Prozess die Datenbankdateien und deren Verbindungen, die von Client zum Server aufgebaut werden, und bearbeitete Anfragen verwaltet.

[?]

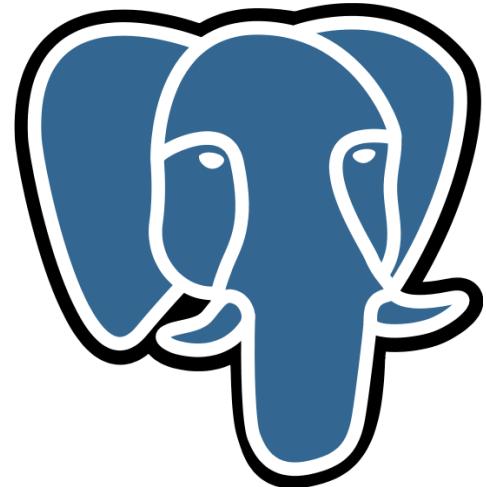


Abbildung 33.: PostgreSQL Logo
[?]

Ein Vorteil von PostgreSQL ist, dass es ein Open Source System ist. Somit ist der Code frei zugänglich, daher ist es möglich, PostgreSQL zu nutzen oder so zu verändern und implementieren, wie es aktuell für einen von Vorteil ist. Weites ist es möglich, die PostgreSQL Datenbank zu skalieren und somit kann die Datenbank auf die Größe der Anwendung angepasst werden. [?]

8.10. Rest-API

Representational State Transfer - Application Programming Interface (Rest-API) ermöglicht den Austausch von Informationen, die sich auf unterschiedlichen Systemen befinden. Eine Rest-API ist zustandslos, dies bedeutet, dass Aufrufe unabhängig voneinander erfolgen und jeder Aufruf alle Daten enthält, die erforderlich sind für eine erfolgreiche Durchführung. [?]

Außerdem baut Rest-API auf bestehende Systeme und Funktionen des Hypertext Transfer Protocol (HTTP) auf. Somit können Rest-basierte Interaktionen deren Status über numerische HTTP-Statuscodes kommunizieren. Der Vorteil besteht darin, dass REST-APIs diese HTTP-Statuscodes verwenden, um Fehler zu erkennen.

Weiters ist Rest-API Sprachunabhängig, dies bedeutet, dass man beim Erstellen von Restful-APIs oder Webdiensten jegliche Sprache verwenden kann, die HTTP verwendet. [?]

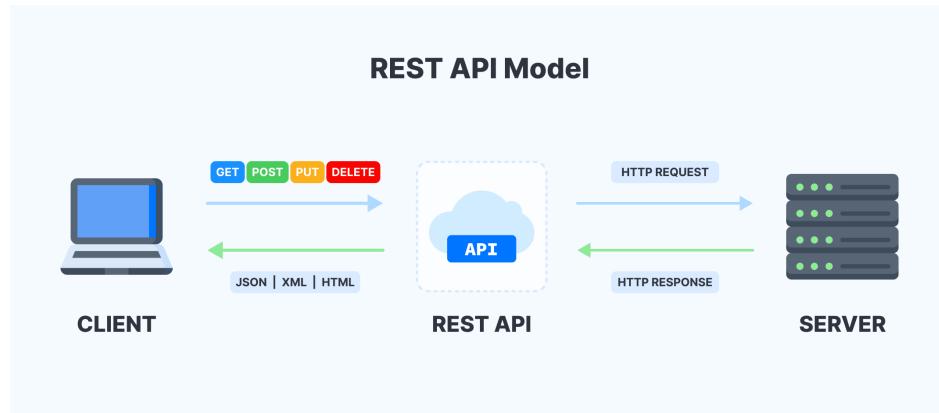


Abbildung 34.: Rest-API Model [?]

Der Client schickt mittels HTTP-Request Methoden Anfragen an die Rest-API, die die Requests an den Client weiterschicken.

Der Server sendet einen HTTP-Response an die Rest-API welche die Response an den Client als JSON, XML oder HTML Datei sendet.

8.10.1. API

Ein Application Programming Interface (API) ist eine Schnittstelle, die verschiedene Programme verbindet und die Datenübertragung und den Austausch von Anweisungen zwischen Programmteilen standardisiert.

APIs bieten verschiedene Anwendungsbereiche, über welche unterschiedliche Aufgaben ausgeführt werden können.

Es gibt vier verschiedene Arten von APIs:

- Funktionsorientierte APIs
- Dateiorientierte APIs
- Protokollorientierte APIs
- Objektorientierte APIs

Funktionsorientierte APIs sind komplexe Schnittstellen, die es ermöglichen, auf Hardware-Komponenten zuzugreifen. Dateiorientierte APIs ermöglichen die Verbindung auf Dateiebene und damit können Daten abgefragt und geschrieben werden. Protokollorientierte APIs werden zur standardisierten Kommunikation zwischen Programmen verwendet. Objektorientierte APIs sind flexibel und können in verschiedenen Bereichen eingesetzt werden. [?]

8.10.2. HTTP

Hypertext Transfer Protocol (HTTP) ist ein Protokoll, mit dem Daten in Netzwerken übertragen werden. HTTP ist standardisiert und definiert, wie Webclient und Server miteinander kommunizieren, damit vom Client angeforderte Daten geladen und angezeigt werden.

HTTP definiert zwei unterschiedliche Arten von Nachrichten

- Anfrage (Request)
- Antwort (Response)

Diese Nachrichten bestehen aus einem HTTP-Header und einem HTTP-Body. Im Header sind Meatinformationen enthalten und im Body befinden sich die Daten, die an den Client geschickt werden.

Zur Übertragung von Daten zwischen Server und Client wird das TCP/IP-Protokoll verwendet. Fordert der Client über TCP ein Dokument an, wird eine Antwort (Response) vom Server versendet, dabei wird auch ein HTTP-Statuscode mitgesendet. Mithilfe des Statuscodes wird Auskunft darüber gegeben, ob die Anfrage erfolgreich war. Je nachdem, ob es erfolgreich war oder nicht wird ein dreistelliger HTTP-Code an den Client gesendet. [?]

HTTP-Statuscodes

HTTP-Statuscodes zeigen an, ob eine HTTP-Anfrage (Request) erfolgreich abgeschlossen wurde.

Man unterscheidet zwischen folgenden Statuscodes:

- Informativne Antworten (100-199)
- Erfolgreiche Antworten (200-299)
- Umleitungen (300-399)
- Client-Fehler (400-499)
- Server-Fehler (500-599)

[?]

HTTP Status Codes



Abbildung 35.: HTTP-Statuscodes [?]

HTTP-Request

Ein HTTP-Request ist eine Anfrage, die vom Client an den Server gesendet wird. Die Requests sagen mithilfe von Methoden, Server mit dem Request durchführen soll.

Folgende Methoden können in den Requests enthalten sein:

- "**GET**": Mithilfe der GET-Methode fordert der Client Inhalte vom Server an.
- "**POST**": Mit POST sendet der Client Dokumente zum Server.
- "**HEAD- "**PUT**": Bei der PUT-Methode sendet der Client Daten zum Server. Beziehen sich die mitgeschickten Daten auf eine vorhandene Ressource, werden diese aktualisiert, ansonsten neu erstellt.
- "**DELETE**": Mithilfe der DELETE-Methode werden Daten auf dem Server gelöscht.
- "**CONNECT**": Mit der CONNECT-Methode kann ein Tunnel erstellt werden, um auf Websites, die SSL verwenden, zuzugreifen.
- **OPTIONS**: Bei der OPTIONS-Methode kann der Client Informationen über verfügbare Kommunikationsoptionen abrufen.**

- "Trace": Über die TRACE-Methode kann der Client Requests verfolgen.



Abbildung 36.: HTTP-Methoden [?]

HTTP-Response

Ein HTTP-Response ist eine Antwort oder Meldung, die von Server zum Client gesendet wird. Ein HTTP-Response besteht aus einem Header mit optionalen Antwortparametern und einem Body, der jedoch auch leer sein kann. Der Header enthält die Protokollversion, außerdem enthält er auch noch einen Statuscode und einer Statusmeldung. Der Statuscode der Antwort teilt dem Client mit, ob die gewünschte Operation vom Server ausgeführt werden konnte. [?]

TCP/IP-Protokoll

Das TCP/IP-Protokoll ist eine Gruppe von Protokollen, die die Grundlage für das Internet und andere Netzwerke bilden. Der Name TCP/IP setzt sich aus den beiden Protokollen Transmission Control Protocol (TCP) und Internetprotokoll (IP) zusammen. Wobei bei auch bei diesen Protokollen mehrere Protokollen zusammengefasst werden. Bei TCP/IP handelt es sich also um keine bestimmte Technik, sondern um eine Gruppierung von Protokollen.

Die Protokolle des TCP/IP-Models sind so standardisiert, dass es nicht von Wichtigkeit ist, welches Betriebssystem oder welches Gerät man für die Kommunikation über das Netzwerk verwendet. [?]

8.11. Software-Testing

Software-Testing ist der Prozess, um die Funktionalität einer Softwareanwendung zu bewerten. Es dient zum Feststellen, ob die entwickelte Software die Anforderungen erfüllt oder nicht. Es wird also das Ergebnis der Entwicklung getestet. [?]

8.11.1. Testtechniken

Man unterscheidet zwischen zwei Testtechniken

Statische Testtechniken

Statische Testtechniken dienen dem Überprüfen von Arbeitsergebnissen, ohne diese auf dem Rechner auszuführen.

Vorteile sind, dass es eine frühe Fehlererkennung gibt und Fehler direkt aufgedeckt werden. Außerdem verringern sie die Anzahl von dynamischen Tests, welche aufwändiger und teuer sind.

Dynamische Testtechniken

Dynamische Testtechniken werden verwendet, um Fehler in der Software zu entdecken.

[?] Es gibt verschiedene dynamische Testtechniken. Die geläufigsten sind:

- "**Black-Box-Testverfahren**" Dabei werden Testfälle nur anhand der Spezifikationen/Anforderungen erstellt, ohne die innere Struktur zu berücksichtigen.
[?]
- "**White-Box-Testverfahren**" Berücksichtigt im Gegensatz zum Black-Box-Testverfahren das innere des Testobjekts, das heißt den Code. [?]
- "**Erfahrungsbasierte Testverfahren**" Dabei werden Testfälle anhand von Erfahrungen, Wissen und Intuition der Tester erstellt. [?]

8.11.2. Testarten

Man unterscheidet zwischen verschiedenen Arten zu testen. Die Entscheidung, welcher Test verwendet werden soll, hängt von dem Ziel ab, welches der Test erfüllen soll.

Funktionale Tests

Funktionale Tests konzentrieren sich auf Funktionen in der Anwendung und überprüfen, ob diese ihre Aufgabe richtig erfüllen.

Nicht-funktionale Tests

Nicht funktionale Tests überprüfen, wie gut die Anwendung im Ganzen funktioniert.

- "**Performanz/ Effizienz**" wird durch Performanztests oder Lasttest geprüft.
- "**SZuverlässigkeit**" Zuverlässigkeitstest überprüfen, ob ein Testobjekt über einen bestimmten Zeitraum ein bestimmtes Leistungsniveau unter bestimmten Bedingungen aufrechterhält.
- "**Benutzbarkeit/ Gebrauchstauglichkeit**" Benutzbarkeitstests prüfen, wie gut und einfach ein System für den Benutzer zu bedienen ist.
- "**SSicherheitstest**" überprüfen, ob System und Daten vor unerlaubten Zugriffen und externen Bedrohungen geschützt sind.
- "**Kompatibilität**" Interoperabilitätstest bewerten die Fähigkeit des Softwareprodukts.
- "**Wartbarkeit**" wird durch Reviews und werkzeuggeschützte statische Analyse geprüft.
- "**Übertragbarkeit**" wird durch Portabilitätstests getestet.

[?]

8.11.3. Unit Test

Mithilfe von Unit Tests werden Komponenten des geschriebenen Codes überprüft. Der Code wird in einzelne Teile isoliert und diese Teile werden auf ihre Funktionalitäten überprüft.

Unit Tests sind nach den drei A's der Modultests aufgebaut.

- **Ärrange (Anordnen)**: Beim Arrange werden die Anforderungen definiert, die der Code erfüllen muss.

- **Äct (Handeln)**"": Beim Act wird der Test durchgeführt.
- **Ässert (Umsetzung)**"": Beim Assert werden die Ergebnisse überprüft. Wenn die Ergebnisse dem entsprechen, was erwartet wird, wird der Test validiert.

[?]

8.11.4. Smoke Test

Smoke Test werden als Software-Probelauf bezeichnet. In einem kurzen Zeitraum soll die Funktionalität von neuen Funktionen und Anwendungen geprüft werden. Das Ziel von Smoke Tests ist es, Fehler zu finden, die später gravierende Probleme auslösen.

[?]

8.11.5. Regressionstest

Mithilfe von Regressionstests wird überprüft, ob Änderungen an der Anwendung oder andere dazugehörige Softwarekomponenten Fehler herbeiführen und ob die Änderungen am Code die Funktionsweise beeinträchtigen oder stören.

Regressionstests können bei folgende Prozessen erforderlich sein:

- beim Einführen von neuen Features
- beim Beheben eines Fehlers
- beim Refactoring zur Steigerung der Leistung
- beim Ändern der Hosting-Umgebung einer Anwendung

Regressionstests werden auf einer der folgenden Testebene durchgeführt:

- Unit-Tests
- Integrationstests
- Systemtests
- Akzeptanztests

[?]

8.11.6. Integrationstest

Bei Integrationstests werden Units in Gruppen auf verschiedenste Weise kombiniert und getestet. Integrationstests können Probleme von Schnittstellen aufdecken.

Es gibt 2 Methoden zur Durchführung von Integrationstest:

- "**Bottom-up-Methode**": Bottom-up Integrationstests beginnen mit Unit-Test, die gefolgt werden von Tests mit Kombinationen von Units.
- "**Top-down-Methode**": Top-down Integrationstests beginnen damit, die Module höchster Ebene zu testen. Erst dann werden Module niedrigerer Ebenen getestet.

[?]

8.11.7. Systemtests

Bei Systemtests wird überprüft, wie die verschiedenen Komponenten im vollständigen System oder in der Anwendung zusammenarbeiten.

Es gibt verschiedene Arten von Systemtests, welche unterschiedliche Komponenten testen:

- "**Leistungstests**": Geschwindigkeit, Stabilität, Reaktionszeiten
- "**Lasttests**": Latenz, Anzahl der Benutzer
- "**Usability-Tests**": Aufgabenerfolgsrate, Zeit bis Aufgaben erledigt sind

[?]

8.11.8. Testframeworks

Testframeworks sind Richtlinien oder Regeln, die zum erstellen und entwerfen von Testfällen verwendet werden.

[?]

9. Tools

9.1. Adobe XD

Adobe XD ist eine kostenpflichtige vektorbasierte Design-Plattform für die Erstellung von Wireframes, Animations- und Interaktions-Design, User Interface Design und Prototypen und gilt weiters als All-in-One-Tool. Adobe XD erlaubt es Komponenten als eine Art Template anzulegen und diese zur Wiederverwendung und Synchronisierung gängiger Elemente wie zum Beispiel Buttons und Navigationssymbole zu verwenden. Verändert man nun eine erstellte Hauptkomponente im Nachhinein, werden alle zugehörigen Änderungen automatisch in allen Instanzen angezeigt. Mit Komponentenzuständen können allerdings Variationen für eine einzelne Komponente erstellt werden. Zum Beispiel kann eine andere Hintergrundfarbe im Hover-Zustand (Mauszeiger liegt über dem Element) bestimmt werden.



Eine weitere Funktion Responsive Resize erkennt Layouts und passt das Design für andere Formate automatisch an. Adobe XD bietet ebenfalls die Verwendung von Plugins für die verschiedensten Funktionen, wie zum Beispiel die Verwendung von Icons, an. Um einen funktionierenden Prototyp zu erstellen kann auf die Verwendung von Ankerlinks nicht verzichtet werden, da nahtlose Übergänge und eine Navigation von Seite zu Seite dafür unabdinglich sind. [92], [93]

9.2. Oracle SQL-Developer

Oracle SQL Developer ist eine kostenlose, integrierte Entwicklungsumgebung, für die Entwicklung und Verwaltung von Datenbanken. SQL Developer bietet eine vollständige Ende-zu-Ende-Entwicklung von PLSQL- und SQL-Anwendungen, ein Arbeitsblatt für die Ausführung von Abfragen und Skripten, eine Datenbankadministrationskonsole für die Verwaltung der Datenbank, eine Berichtsschnittstelle und eine vollständige Datenmodellierungslösung den Data Modelers. [94]

9.2.1. Data Modeler

Oracle SQL Developer Data Modeler ist ein kostenloses, grafisches Tool, dass die Datenmodellierungsaufgaben vereinfacht. Mit dem Data Modeler können Benutzer/innen logische, relationale, physische und mehrdimensionale Datentypenmodelle erstellen, durchsuchen und bearbeiten. Der Data Modeler bietet Forward- und Reverse-Engineering-Funktionen und unterstützt die gemeinsame Entwicklung durch integrierte Quellcodekontrolle. [95]

10. Ausgewählte Aspekte

10.1. Besonders gut gelöste Programmteile

10.1.1. Internationalization

Listing 1: Internationalization-Makierung im HTML

```
1 <mat-label i18n="label text|Question text">Frage text eingeben</mat-label>
2 <mat-card-title i18n="Title|card title" >Frage
   {{administration.question.sequenceNumber}}</mat-card-title>
```

Texte, die eine Übersetzung bekommen sollen, werden mit der Kennzeichnung i18n versehen, kurz für Internationalization. Nach der Kennzeichnung können noch zusätzliche Informationen in der Form von i18n="`<Bedeutung> | <Beschreibung>`" für die automatische Übersetzung angegeben werden. In diesem Beispiel wurde keine automatische Übersetzung verwendet, trotzdem wurden zusätzliche Attribute angeben um im Falle eines Umstiegs schneller wechseln zu können.

Listing 2: xliff-Datei

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <xliff version="1.2" xmlns="urn:oasis:names:tc:xliff:document:1.2">
3   <file source-language="en" datatype="plaintext" original="ng2.template">
4     <body>
5       <trans-unit id="3292363768460657245" datatype="html">
6         <source>Richtige Antwort</source>
7         <target>Right Answer</target>
8         <context-group purpose="location">
9           <context
10             context-type="sourcefile">src/app/answer-option-erstellen/answer-option-erstellen
11             <context context-type="linenumber">7</context>
12           </context-group>
13           <note priority="1" from="description">right answer</note>
14           <note priority="1" from="meaning">checkbox text</note>
15         </trans-unit>
16         <trans-unit id="878423452191694276" datatype="html">
17           <source><x id="INTERPOLATION" equiv-text="{{this.questions.length}}"/>
18             Fragen</source>
19             <target><x id="INTERPOLATION" equiv-text="{{this.questions.length}}"/>
20               Question</target>
21             <context-group purpose="location">
22               <context
23                 context-type="sourcefile">src/app/answer-survey/answer-survey.component.html</context>
24                 <context context-type="linenumber">4</context>
25               </context-group>
26               <note priority="1" from="description">question anz</note>
27               <note priority="1" from="meaning">headline text</note>
28             </trans-unit>
29             ...
30           </context-group>
31         </trans-unit>
32       </body>
33     </file>
34   </xliff>
```

Durch die Ausführung des Befehls `ng xi18n -output-path src\locale` werden alle Texte mit dem Kennzeichen ibn18 in eine XLF-Datei extrahiert. In der Datei wird zuerst angegeben in welcher Form sie geschrieben ist. In diesem Fall ist es eine XLF-Datei mit der Version 1.2 (Angular Standard). (Zeile 2)

`source-language="en"` gibt die Sprache an. (Zeile 3)

Eine `<trans-unit>` wird für jeden mit einer Kennzeichnung versehenen Text generiert. (Zeile 5)

In `<source>` steht der Originaltext, der angegeben wurde. (Zeile 6)

Der Text der nun als Übersetzung gilt muss umklammert von `<target>` `<|target>` für jede `<trans-unit>` selbst übersetzt und unter `<source>` eingefügt werden. (Zeile 7)

Listing 3: Konfigurieren der Sprachen im angular.json

```

1   "i18n": {
2     "sourceLocale": "de",
3     "locales": {
4       "en": "src/locale/language_source.eng.xlf"
5     }
6     ...
7     "en": {
8       "localize": ["en"]
9     },
10    "de": {
11      "localize": ["de"]
12    },
13    ...

```

Zum Schluss muss der Applikation noch mitgeteilt werden, wo die Ausgangssprache Datei liegt (Zeile 4) und in welcher Sprache diese angegeben ist. Zeile 7 und 10 sind Kürzel, die später für das Starten der Applikation wichtig sind.

Es ist zu beachten, dass nach diesen Änderungen beim Starten der Applikation immer anzugeben ist welche Sprache verwendet werden soll. Beispielsweise: `ng serve -configuration=de` oder `ng serve -configuration=en` Durch kleine Änderungen im package.json kann auch automatisiert eine bestimmte Sprache gestartet werden.

Listing 4: automatisches Starten der deutschen Sprache

```

1   "scripts": {
2     "ng": "ng",
3     "start": "ng serve --configuration=de",
4     ...

```

Mittels `npm start` wird nun automatisch die deutsche Version der Applikation gestartet.

10.1.2. Angular-CSS bearbeiten

Listing 5: Semi-Transparentes Eingabefeld

```

1  ::ng-deep .mat-form-field-appearance-fill .mat-form-field-flex {
2      background: rgba(255, 255, 255, 0.5);
3 }
```

Um den Hintergrund einer Material-Komponente Mat-Form-Field (eine Komponente die Formularelemente einschließt) halbtransparent zu machen wurde ng-deep verwendet.

Ng-deep ermöglicht den Zugriff auf DOM-Elemente, in nicht selbstgeschriebenen Komponenten. Im Beispiel von Angular Material (oder eine andere Bibliothek eines Drittanbieters wie Materials), befinden sich die Elemente außerhalb des Bereichs (wie zum Beispiel Eingabefelder), und lassen keinen direkten Zugriff mit CSS auf diese Elemente zu. Ng-deep ermöglicht diesen Zugriff. Mehr Infos zu Angular Material befinden sich im Kapitel 8.1.8.

10.1.3. Keycloak

Mittels des Keycloak können sich die Benutzer der Applikation mit gewohnten Anmeldedaten anmelden.

Listing 6: Konfiguration des Keycloaks

```

1  {
2      "backendApiUrl": "http://localhost:8080/api",
3      "keycloakUrl": "http://localhost:8180/auth/realms/School"
4 }
```

Für die Anbindung des Keycloaks an das Frontend muss zuerst festgelegt werden, unter welcher URL sich der KeycloakService und das Backend befinden.

Listing 7: Aufrufen des Logins

```

1  openLogin(){
2      this.config.init().then(() => this.authenticationService.initializeLogin())
3 }
```

Wird nun das Login aufgerufen wird der Authentifikations-Service initialisiert.

Listing 8: Initialisierung des AuthenticationService

```

1     init(): Promise<any> {
2       return this.httpClient
3         .get<Config>('/assets/config.json')
4         .pipe(
5           tap(config => {
6             this.config = config;
7             authModuleConfig.resourceServer.allowedUrls!.push(config.backendApiUrl);
8           })
9         )
10        .toPromise();
11      }

```

Mit der Initialisierung werden die angegebenen Konfigurationsdateien ausgelesen und die angegeben URLs zum Weiterleiten zugelassen.

Listing 9: Aufrufen des Authentifikationswächters

```

1   const routes: Routes = [
2     {path: '', component: StartseiteComponent, pathMatch: "full"},
3     {path: '', canActivateChild: [AuthGuard], children: [
4       // {path: '', redirectTo: '/dashboard', pathMatch: 'full'},
5       {path: 'dashboard', component: DashboardComponent},
6       {path: 'fragebogen-erstellen', component: FragebogenErstellenComponent},
7       {path: 'dashboard/fragebogen-editieren/:id', component:
8         EditQuestionnaireComponent}
9     ]},
10    {path: 'answerSurvey', component: AnswerSurveyComponent},
11    {path: 'startseite', component: StartseiteComponent},
12    {path: '**', component: StartseiteComponent}
13  ];

```

Man wird nicht nur mit dem Klick auf den Login-Knopf zum Anmelden auf den Keycloak weitergeleitet. Will man auf Unterseiten, die einen Login erfordern zugreifen, wird man ebenfalls automatisch zum Login weitergeleitet. Dies wird mittels Authentifikationswächtern (authenticationguards) bewerkstelligt. In diesem Fall wird der Wächter aufgerufen, wenn auf jede Unterseite (Zeile 3, 4, 5, 6 und 7) (bis auf die Startseite und die Seite zum Beantworten der Fragen (Zeile 9 und 10)) aufgerufen wird.

Listing 10: Authentifikationswächter

```

1   canActivateChild(
2     childRoute: ActivatedRouteSnapshot,
3     state: RouterStateSnapshot
4   ): Observable<boolean | UrlTree> | Promise<boolean | UrlTree> | boolean |
5     UrlTree {
6       console.log(this.authenticationService.oidcLoaded.value);
7       if (this.authenticationService.oidcLoaded.value) {
8         return true;
9       }
10      this.config.init().then(() => this.authenticationService.initializeLogin())
11      return false;
12    }

```

Der Authentifikationswächter überprüft beim Aufrufen einer Seite, die er überwacht beziehungsweise beschützt, ob der derzeitige Benutzer sich bereits einmal angemeldet

hat (Zeile 6). Sollte das der Fall sein wird der Benutzer auf die gewünschte Seite weitergeleitet. Hat sich der Benutzer noch nicht eingeloggt wird der Authentifikations-Service initialisiert und der derzeitige Aufrufen der Unterseite abgewiesen. (Zeile 9 und 10)

Listing 11: Login und Logout

```

1  username = new BehaviorSubject<string>('');
2  roles = new BehaviorSubject<string[]>([]);
3  oidcLoaded = new BehaviorSubject<boolean>(false);
4  ...
5  logOut() {
6    this.oAuthService.logOut();
7  }
8
9  async initializeLogin(): Promise<void> {
10    console.log(this.configService.config);
11    this.oAuthService.configure({
12      issuer: this.configService.config.keycloakUrl,
13      redirectUri: window.location.origin,
14      clientId: 'my-frontend-service',
15      responseType: 'code',
16      scope: 'offline_access',
17      showDebugInformation: true
18    });
19    await this.oAuthService.loadDiscoveryDocumentAndTryLogin({
20      customHashFragment: window.location.search
21    });
22
23    if (!this.oAuthService.hasValidAccessToken()) {
24      this.oAuthService.initLoginFlow();
25    } else {
26      this.oAuthService.setupAutomaticSilentRefresh();
27      const profile: any = await this.oAuthService.loadUserProfile();
28      this.username.next(profile.info.preferred_username);
29      this.roles.next(this.parseJwt(this.oAuthService.getAccessToken()).realm_access.roles);
30      this.oidcLoaded.next(true);
31    }
32  }

```

Mit dem Aufruf der Logout-Methode wird der Authentifikations-Service benachrichtigt und loggt den Benutzer aus. (Zeile 1 und 2). Will sich der Benutzer nun einloggen, werden zuerst einige wichtige Informationen gesetzt wie zum Beispiel der Name des Client des Keycloaks für die Applikation (Zeile 10) und wo der Benutzer nach erfolgreichem Einloggen, hingeleitet wird (Zeile 9). Ist der Benutzer nicht eingeloggt wird der Benutzer zum Login über den Keycloak weitergeleitet. (Zeile 20). Ist der Benutzer bereits eingeloggt wird der Authentifikations-Token erneuert und der Benutzername und die Rolle aktualisiert.

Das Erhalten der Anmeldedaten wurde mithilfe eines Observable-Patterns gelöst. Daten können hierbei in ein Observables gegeben werden. Ändern sich die Daten des Observables werden die Observer, die das Observable beobachten, benachrichtigt.

Nach erfolgreichem Login werden die Anmeldedaten in ein Observable gegeben -

konkret in ein BehaviorSubject, das als Observable fungiert. Um die Daten zu erhalten, wird ein Subscribe auf das BehaviorSubject durchgeführt.

Listing 12: Verwendung der geschickten Daten TypeScript

```

1  ngOnInit(): void {
2      this.authenticationService.username.subscribe(value => {
3          this.logged_in = (value.length > 1);
4      })
5  }

```

Listing 13: Verwendung der geschickten Daten HTML

```

1  ...
2  <button *ngIf="!this.logged_in" i18n="button text|Login button text"
3      mat-stroked-button color="primary" (click)="openLogin()">Einloggen</button>
4  <div *ngIf="this.logged_in">
5      <button mat-stroked-button color="primary"
6          routerLink="/dashboard">Dashboard</button>
7      <button i18n="button text|Logout button text" mat-stroked-button
8          color="primary" (click)="logoutUser()">Ausloggen</button>
9  </div>
10 ...

```

Der Header der Webseite führt beispielsweise ein Subscribe auf das BehaviorSubject, dass den Benutzernamen beinhaltet, aus. Ist der Benutzer nicht eingeloggt, ist beim Laden der Seite der Wert des BehaviorSubject leer. Daraufhin wird ein Login-Button angezeigt. Loggt sich der Benutzer nun ein, wird der Observer des Headers darüber informiert und statt des Login-Buttons wird nun ein Logout-Button und ein Button der zum Dashboard führt angezeigt.

10.1.4. CSV-Download

Listing 14: CSV-Download

```

1  const dialogRef =
2      this.generateTransactionCodes.open(TransactionComponent,{width:"30%"});
3
4  dialogRef.afterClosed().subscribe(result => {
5      if(typeof result == "number"){
6          this.transactionservice.generateTransactioncodes(result,
7              this.survey.id).subscribe((response : any)=>{
8                  let dataType = response.type;
9                  let binaryData = [];
10                 binaryData.push(response);
11                 let downloadLink = document.createElement('a');
12                 downloadLink.href = window.URL.createObjectURL(new Blob(binaryData,
13                     {type: dataType}));
14                 if ("transactionCodes.csv"){
15                     downloadLink.setAttribute('download', "transactionCodes_" +
16                         this.survey.title + ".csv");
17                     document.body.appendChild(downloadLink);
18                     downloadLink.click();
19                 }
20             });
21         });
22     });
23   );

```

Es wird ein Dialogfenster geöffnet (Zeile 1) in dem die Anzahl der Transaktions Codes angegeben wird. Nach dem Schließen des Dialogfensters wird überprüft, ob das Dialogfenster "richtig" geschlossen wurde - es wurde eine Zahl angegeben und *Generieren* und nicht *Zurück* ausgewählt (Zeile 2 & 3). Danach werden die Transactioncodes mittels einer HTTP-Anfrage an den Server generiert. Dieser schickt ein CSV-File an das Frontend zurück (Zeile 4). Die Daten werden von einem JSON-Objekt wieder in eine CSV-File umgewandelt (Zeile 5, 6 und 7) und danach wird das File zum Herunterladen bereit gemacht (8 und 9). Der Datei wird noch ein Name verliehen (Zeile 11) und danach automatisch heruntergeladen (DownloadLink wird geklickt Zeile 13).

10.1.5. Reaktive Formulare

Reaktive Formulare werden für die Verarbeitung und Überprüfung von Formulareingaben verwendet.

Listing 15: Formularüberprüfung mit reaktiven Formularen TypeScript

```

1  this.userForm = new FormGroup({
2    name: new FormControl('', [Validators.required, Validators.minLength(5),
3      Validators.maxLength(50)]),
4    desc: new FormControl('', [Validators.required, Validators.minLength(5),
5      Validators.maxLength(250)])
6  });

```

Um reaktive Formulare zu verwenden, muss eine *FormGroup* (stellvertretend für das Formular) mit 1 oder mehreren Einträgen mit *FormControl* (stellvertretend für eine Formulareingabe) erstellt werden. Die im *FormControl* angegebenen Validationen dienen zur Überprüfung, ob das Feld für den Gebrauch, richtig ausgefüllt worden ist. In diesem Fall sollen die Felder ausgefüllt und zwischen 5 und 50 \250 Zeichen enthalten.

Listing 16: Formularüberprüfung mit reaktiven Formularen HTML

```

1  <form [formGroup]="userForm">
2  ...
3    <input matInput formControlName="name" i18n-placeholder="placeholder
4      text|placeholder for questionnaire name" placeholder="Umfragename eingeben"
5      name="questionnaireName" [(ngModel)]="this.questionnaire.name" required>
6    <mat-error *ngIf="userForm.controls['name'].invalid">
7      <p i18n="error message|error name">Name muss zwischen 5-50 Zeichen enthalten.</p>
8    ...

```

Um die *FormGroup* und die *FormControl* mit den zugehörigen Elementen zu verknüpfen, müssen diese im HTML referenziert werden. (Zeile 1 und 3). Sollten die angegeben Validationen nicht erfüllt sein, wird ein entsprechender Fehlertext angezeigt. (Zeile 4)

10.1.6. Berechnen der anzuzeigenden Fragen

Beim Beantworten von Fragen werden nur eine gewissen Menge an Fragen auf einmal angezeigt, um langes Scrollen zu verhindern. Zudem soll es den Benutzer / die Benutzerin dabei unterstützen sich immer nur auf ein paar Fragen zu fokussieren, um nicht überwältigt zu werden.

Listing 17: Pageinator

```

1   ...
2   <div class="mat-paginator-container">
3     <mat-paginator [showFirstLastButtons]="true" [length]="this.questions.length"
4       [pageSize]="'pageSizeOptions[1]" [pageSizeOptions]="'pageSizeOptions"
5         aria-label="Select page" (page)="recalc($event)">
6     </mat-paginator>
7   </div>
8
9   <div id="questionsbox">
10    <div *ngFor="let item of this.indexArray, let i=index">
11      <app-questions-to-answer
12        [questionToAnswer]="'this.questionsToAnswer[item]"></app-questions-to-answer>
13    </div>
14  </div>
15
16  ...

```

Während der Beantwortung der Fragen wird mit einem Pageinators gearbeitet, (Zeile 2) der die gesamten Fragen auf mehrere Seiten aufteilt. Dieser hilft dabei, die Fragen nur stückchenweise anzuzeigen. Mit ihm können die gesamte Anzahl an Fragen, die angezeigt werden (3, 5 oder 10 Fragen Standardmäßig wurde 5 eingestellt) und die Seite, die man ansteuern will, eingestellt werden. Es ist auch möglich, schnell auf die erste und letzte Seite zu springen. Werden die Standardeinstellungen geändert oder wird die Seite gewechselt, werden die angezeigten Fragen neu berechnet.

Listing 18: Berechnen der anzuzeigenden Fragen TypeScript

```

1   ...
2   recalc(pageEvent: PageEvent){
3     this.fillArray(pageEvent.pageSize * pageEvent.pageIndex, pageEvent.pageSize *
4       (pageEvent.pageIndex + 1));
5   }
6   fillArray(start : number, end : number){
7     this.indexArray = [];
8     for (let index = start; index < end; index++) {
9       if(index < this.questionsToAnswer.length){
10         this.indexArray.push(index);
11       }
12     }
13   }
14   ...

```

Für die Berechnung wird ein Event ausgelöst, welches mitteilt, dass sich die Einstellungen verändert haben (Zeile 2). Danach wird ein Array, mit den anzuzeigenden Fragen, zuerst geleert und danach gefüllt.

10.1.7. Konfiguration des Traefik

Traefik wurde für die Sicherheit, genauer gesagt die Verwendung von https statt http eingesetzt. HTTPS- und HTTP-Anfragen werden innerhalb des DockerContainer auf HTTP-Anfragen umgeleitet. Zudem wurde eine festgelegte Version des Traefik-Images verwendet und ein Dashboard hinzugefügt (siehe 37).

Listing 19: Starten Traefik

```

1   version: "3.3"
2   services:
3     traefik:
4       image: "traefik:v2.6"
5       command:
6         - --entrypoints.web.address=:80
7         - --entrypoints.websecure.address=:443
8         - --providers.docker
9         - --api
10      ports:
11        - "80:80"
12        - "8080:8080"
13        - "443:443"
14      volumes:
15        - "/var/run/docker.sock:/var/run/docker.sock:ro"
16      labels:
17        # Dashboard
18        - "traefik.http.routers.traefik.rule=Host('traefik.docker.localhost')"
19        - "traefik.http.routers.traefik.tls.certresolver=leresolver"
20        - "traefik.http.routers.traefik.service=api@internal"
21        - "traefik.http.routers.traefik.entrypoints=websecure"
22
23      # middleware redirect
24      - "traefik.https.middlewares.redirect-to-https.redirectscheme.scheme=http"
25
26      # global redirect to http
27      - "traefik.https.routers.redirs.rule=hostregexp('{host:.+}')"
28      - "traefik.https.routers.redirs.entrypoints=web"
29      - "traefik.https.routers.redirs.middlewares=redirect-to-https"

```

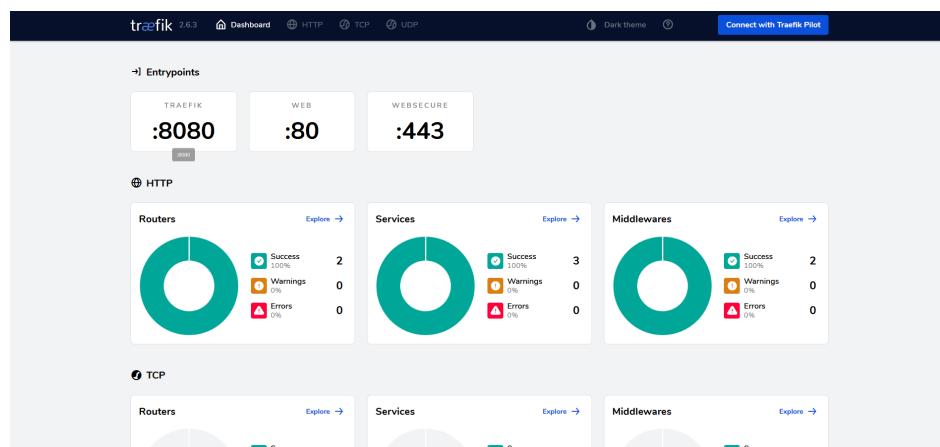


Abbildung 37.: Dashboard Traefik

10.1.8. Dockerfiles

Keycloak Docker

Listing 20: Keycloak Dockerfile

```

1  version: '3.9'
2  services:
3    keycloak:
4      # Name des Containers
5      container_name: keycloak
6      environment:
7        # Zugriffsdaten
8        - KEYCLOAK_USER=admin
9        - KEYCLOAK_PASSWORD=admin
10       # Eingebaute h2 Datenbank
11       - DB_VENDOR=h2
12       ports:
13         - '8180:8080'
14       # Image auf dem Aufgebaut wird
15       image: 'quay.io/keycloak/keycloak:15.0.2'
16       volumes:
17         # Speicherort der Zertifikate der Benutzer und Benutzerinnen
18         - ./certs:/etc/ssl/certs/java
19         # Speicherort der Keycloak Standardeinstellungen
20         - ./imports:/opt/jboss/keycloak/imports
21         # Einlesen der Standardeinstellungen und Verhalten des Einlesens bereits
22           vorhandene Objekte nicht ersetzen
23       command: [ '-b', '0.0.0.0',
24                 '-Dkeycloak.profile.feature.upload_scripts=enabled',
25                 '-Djavax.net.ssl.trustStore=/etc/ssl/certs/java/cacerts',
26                 '-Djavax.net.ssl.trustStorePassword=changeit',
27                 '-Dkeycloak.migration.action=import',
28                 '-Dkeycloak.migration.provider=singleFile', '-Dkeycloak.migration.strategy=IGNORE_EXISTING',
29                 '-Dkeycloak.migration.file=/opt/jboss/keycloak/imports/keycloak_settings.json'
30               ]
31       networks:
32         - keycloak_network
33
34   networks:
35     keycloak_network:
36       name: keycloak_network
37       driver: bridge

```

Die importierten Standardeinstellungen erstellen automatisch das Realm, Clients für Frontend und Backend, die Rollen, etc. die für die Benutzung gebraucht werden. Somit müssen die Einstellungen nicht jedesmal beim Starten manuell ausgeführt werden.

Listing 21: Auszug aus den importieren Einstellungen

```

1  {
2    "id": "School",
3    "realm": "School",
4    "notBefore": 0,
5    "defaultSignatureAlgorithm": "RS256",
6    "revokeRefreshToken": false,
7    "refreshTokenMaxReuse": 0,
8    "accessTokenLifespan": 300,
9    "accessTokenLifespanForImplicitFlow": 900,
10   "ssoSessionIdleTimeout": 1800,
11   "ssoSessionMaxLifespan": 36000,
12   "ssoSessionIdleTimeoutRememberMe": 0,
13   "ssoSessionMaxLifespanRememberMe": 0,
14   "offlineSessionIdleTimeout": 2592000,
15   "offlineSessionMaxLifespanEnabled": false,
16   "offlineSessionMaxLifespan": 5184000,
17   "clientSessionIdleTimeout": 0,
18   "clientSessionMaxLifespan": 0,
19   "clientOfflineSessionIdleTimeout": 0,
20   "clientOfflineSessionMaxLifespan": 0,
21   "accessCodeLifespan": 60,
22   "accessCodeLifespanUserAction": 300,
23   ...

```

Postges Docker

Listing 22: Postgres Dockerfile

```

1      version: '3.1'
2      services:
3          # Datenbank
4          db:
5              # Containername
6              container_name: survey_postgres
7              # Image auf das Aufgebaut wird
8              image: postgres:13.3-alpine
9
10             # Bei (manuell oder anderweitigen) stoppen der DB wird, nicht neu gestartet
11             # wird, selbst wenn der Docker-Daemon neu startet
12             restart: unless-stopped
13             # Zugriffsdaten
14             environment:
15                 POSTGRES_USER: postgres
16                 POSTGRES_PASSWORD: postgres
17                 POSTGRES_DB: db
18             ports:
19                 - 5432:5432
20             volumes:
21                 # Speicherort der permanenten Daten
22                 - ./db-postgres/db:/var/lib/postgresql/data
23                 # Importdatenspeicherort
24                 - ./db-postgres/import:/import
25             networks:
26                 - postgres
27
28             # grafische Tool zur Entwicklung und Administration
29             pgadmin:
30                 container_name: survey_pgadmin
31                 image: dpage/pgadmin4:5.5
32                 environment:
33                     PGADMIN_DEFAULT_EMAIL: ${PGADMIN_DEFAULT_EMAIL:-pgadmin4@pgadmin.org}
34                     PGADMIN_DEFAULT_PASSWORD: ${PGADMIN_DEFAULT_PASSWORD:-admin}
35                     PGADMIN_CONFIG_SERVER_MODE: 'False'
36                 volumes:
37                     - ./db-postgres/pgadmin:/root/.pgadmin
38                 ports:
39                     - 8090:80
40                 networks:
41                     - postgres
42                 restart: unless-stopped
43
44             networks:
45                 postgres:
46                     driver: bridge

```

10.1.9. Fragebogen duplizieren

In dieser Diplomarbeit besteht die Möglichkeit, Fragebögen zu duplizieren. Dies wurde mit dem Code von 23 umgesetzt. In diesem Code wird der übergebene Questionnaire ohne id in der Datenbank gespeichert. Da damit aber nur der Fragebogen und nicht die Fragen oder die Antwortmöglichkeiten dupliziert wurde, müssen die Questions und AnswerOptions gefunden werden, welche für den aktuellen Fragebogen verwendet werden. Ist dies geschehen, werden auch diese Daten ohne id in der Datenbank gespeichert. Erst wenn das alles erfüllt ist, wurde der Fragebogen richtig kopiert.

Listing 23: Fragebogen duplizieren

```

1  public Questionnaire duplicateQuestionnaire(Questionnaire questionnaire,
2                                              Interviewer interviewer) {
3      Questionnaire originalQuestionnaire = findQuestionnaire(questionnaire);
4      Questionnaire duplicateQuestionnaire;
5      duplicateQuestionnaire = new Questionnaire(
6          originalQuestionnaire.name,
7          originalQuestionnaire.description,
8          false,
9

```

```

8     interviewer);
9     duplicateQuestionnaire = save(duplicateQuestionnaire);
10    Question duplicateQuestion;
11    List<Question> questions =
12      questionRepository.findByQuestionnaire(originalQuestionnaire);
13    AnswerOption duplicateAnswerOption;
14    List<AnswerOption> answerOptions;
15    for (Question originalQuestion : questions) {
16      duplicateQuestion = new Question(
17        originalQuestion.text,
18        originalQuestion.sequenceNumber,
19        originalQuestion.questiontype,
20        duplicateQuestionnaire);
21      duplicateQuestion = questionRepository.save(duplicateQuestion);
22      answerOptions = answerOptionRepository.findByQuestionId(originalQuestion.id);
23      for (AnswerOption originalAnswerOption : answerOptions) {
24        duplicateAnswerOption = new AnswerOption(
25          originalAnswerOption.text,
26          originalAnswerOption.value,
27          originalAnswerOption.sequenceNumber,
28          originalAnswerOption.isCorrectAnswer,
29          duplicateQuestion);
30        duplicateAnswerOption = answerOptionRepository.save(duplicateAnswerOption);
31      }
32    }
33  }
34  return duplicateQuestionnaire;
35 }
```

10.1.10. Auswertung

Um herauszufinden, wie die Umfrage ausgefallen ist, gibt es die Auswertung. Bei dieser Diplomarbeit wurde sich dafür entschieden, zwei verschiedene Arten von Auswertungen zu implementieren:

1. Auswertung pro Frage
2. Auswertung pro Befragten

Auswertung pro Frage

Bei der ersten Variante wurde sich dafür entschieden, anzuzeigen, wie oft eine Antwortmöglichkeit ausgewählt wurde. Wie so eine Auswertung aussehen kann, sieht man in Abbildung38.

Wie diese Auswertung umgesetzt wurde, wird in 24 gezeigt. Mithilfe eines StringBuilders wurden alle nötigen Informationen zu einem String zusammengefügt und danach mittels PrintWriter in das evaluation.csv File geschrieben. Es werden folgende Daten benötigt:

- Name und Beschreibung vom aktuellen Fragebogen

	A	B
1	Feedback	
2	Feedbackbogen für Lehrer	
3		
4	2 Personen haben den Fragebogen ausgefüllt	
5		
6	* Welches ist dein Lieblingsfach?	
7	- Mathematik	1
8	- Deutsch	0
9	- Englisch	0
10	- Sonstige	1
11	* Gefällt die die Umfrage?	
12	- Ja	2
13	- Nein	0
14	* Noch Anmerkungen?	
15	- Mich stört das der Lehrer immer zu spät kommt	
16	- Nein, keine weiteren Anmerkungen	

Abbildung 38.: Auswertung

- wie viele Transaktionscodes verwendet wurden, mit einem Verweis auf die aktuelle Umfrage
- für jede Frage jeweils
 - den Namen
 - die Antwortmöglichkeiten
 - die Anzahl, wie oft eine Antwortmöglichkeit ausgewählt wurde

Es wird außerdem überprüft, ob eine Frage vom Typen Freetext ist. Ist dies der Fall, wird die Antwort dieser Frage am Ende des Files angehängt.

Listing 24: Auswertung

```

1 Survey survey = findById(surveyId);
2 Questionnaire questionnaire =
3   questionnaireRepository.findById(survey.questionnaire.id);
4 List<Question> questions =
5   questionRepository.getQuestionsByQuestionnaireId(questionnaire.id);
6 List<Transaction> transactions = transactionRepository.findBySurvey(survey);
7 for (Transaction transaction : transactions) {
8   if (transaction.isUsed){
9     usedTransactions++;
10 }
11 File file = new File("evaluation.csv");
12 try (PrintWriter writer = new PrintWriter("evaluation.csv")) {
13   StringScanner sb = new StringScanner();
14   sb.append(questionnaire.name);
15   sb.append(questionnaire.description);
16   sb.append(usedTransactions);
17   sb.append(" Personen haben den Fragebogen ausgefüllt");
18   for (Question question : questions) {
19     sb.append(question.text);
20     answers = answerRepository.findByQuestion(question);
21     if (question.questiontype.equals(QuestionType.FREETEXT)){
22       for (Answer answer : answers) {
23         sb.append(" - ");
24         sb.append(answer.answerText);
25       }
26     }else{
27       answerOptions = answerOptionRepository.findById(question.id);
28       for (AnswerOption answerOption : answerOptions) {
29         chosenOptions =
          chosenOptionRepository.findByAnswerOption(answerOption);
         sb.append(" - ");
       }
     }
   }
 }

```

```

30             sb.append(answerOption.text);
31             sb.append(": ");
32             sb.append(chosenOptions.size());
33         }
34     }
35 }
36 writer.write(sb.toString());
37 } catch (FileNotFoundException e) {
38     System.out.println(e.getMessage());
39 }

```

Bei der zweiten Variante wurde sich dafür entschieden, pro Befragten den Wert der ausgewählten Antwortmöglichkeit anzuzeigen. Wie so eine Auswertung aussehen kann sieht man in Abbildung 39.

	A	B	C	D
1	Feedback			
2	Feedbackbogen für Lehrer			
3				
4	2 Personen haben den Fragebogen ausgefüllt			
5				
6	Person 1	1	3	
7	Person 2	2	4	
8				
9	-Nein, keine weiteren Anmerkungen			
10	-Mich stört das der Lehrer immer zu spät kommt			

Abbildung 39.: Auswertung

Wie diese Auswertung umgesetzt wurde, wird in 25 gezeigt. Mithilfe eines StringBuilders wurden alle nötigen Informationen zu einem String zusammengefügt und danach mittels PrintWriter in das evaluation2.csv File geschrieben. Es werden folgende Daten benötigt:

- Name und Beschreibung vom aktuellen Fragebogen
- wie viele Transaktionscodes verwendet wurden, mit einem Verweis auf die aktuelle Umfrage
- pro Transaktionscode, der für diese Umfrage verwendet wurde, den Wert der ausgewählten Antwortmöglichkeit, jeder Frage

Es wird außerdem überprüft, ob eine Frage vom Typen Freetext ist. Ist dies der Fall, wird die Antwort dieser Frage am Ende des Files angehängt.

Listing 25: Auswertung

```

1 Survey survey = findById(surveyId);
2 Questionnaire questionnaire =
3     questionnaireRepository.findById(survey.questionnaire.id);
4 List<Question> questions =
5     questionRepository.getQuestionsByQuestionnaireId(questionnaire.id);
6 List<Transaction> transactions = transactionRepository.findBySurvey(survey);
7 File file = new File("evaluation2.csv");

```

```

6      try (PrintWriter writer = new PrintWriter("evaluation2.csv")) {
7          StringBuilder sb = new StringBuilder();
8          sb.append(questionnaire.name);
9          sb.append(questionnaire.description);
10         sb.append(usedTransactions);
11         sb.append(" Personen haben den Fragebogen ausgefüllt");
12
13         for (Transaction transaction : transactions) {
14             answers = answerRepository.findByTransaction(transaction);
15             count++;
16             sb.append("Person ");
17             sb.append(count);
18             sb.append(",");
19             for (Answer answer : answers) {
20                 chosenOptions = chosenOptionRepository.findByAnswer(answer);
21                 for (ChosenOption chosenOption : chosenOptions) {
22                     answerOpt = chosenOption.answerOption;
23                     if (chosenOption.question.questiontype ==
24                         QuestionType.FREETEXT){
25                         sb.append("\n");
26                         freetextAnswers.add(chosenOption.answer);
27                     }else {
28                         sb.append(" ");
29                         sb.append(answerOpt.value);
30                         sb.append(";");
31                     }
32                 }
33             }
34             if (freetextAnswers.size() != 0){
35                 for (Answer answer : freetextAnswers) {
36                     sb.append("\n");
37                     sb.append("-");
38                     sb.append(answer.answerText);
39                 }
40                 writer.write(sb.toString());
41             } catch (FileNotFoundException e) {
42                 System.out.println(e.getMessage());
43             }
44         }
45     return file;
46 }

```

10.2. Besondere Probleme, die gelöst wurden

10.2.1. Bild speichern im Backend

Hauptgrund für die schlussendliche Nichtimplementierung des optionalen Speicherns von Bildern zu Fragen war vor allem die Zeiteinteilung, da das Anliegen für das Einbinden von Bildern erst kurz vor Abgabe der Arbeit aufgekommen ist. Trotz dessen wurden Versuche für die Speicherung im Backend unternommen. Diese wurden jedoch im weiteren Verlauf eingestellt.

Um ein Bild in der Datenbank zu speichern, kann ein Objekt von Typ byte[] erstellt werden. (Siehe: 26)

Listing 26: @Lob

```

1      @Lob
2      @Column(name = "photo")
3      private byte[] photo;

```

Das übergebene Bild muss in ein byte[] umgewandelt werden, um dieses zu speichern. Um dies zu erreichen, wurden folgende Versuche unternommen:

Bild in Byte Array umwandeln

Beim ersten Versuch wurde das Bild mithilfe eines ImageIO.read() gelesen und mit einem ByteArrayOutputStream in einem Array gespeichert. Danach wurde das Array durch einen ByteArrayInputStream eingelesen und mit ImageIO.write() im File output.jpg gespeichert.

Listing 27: Bild in Byte Array umwandeln

```

1  public void saveImage(File imageFile){
2      BufferedImage bImage = null;
3      try {
4          bImage = ImageIO.read(imageFile);
5      } catch (IOException e) {
6          e.printStackTrace();
7          System.out.println(e.getMessage());
8      }
9      ByteArrayOutputStream bos = new ByteArrayOutputStream();
10     try {
11         ImageIO.write(bImage, "jpg", bos );
12     } catch (IOException e) {
13         e.printStackTrace();
14     }
15     byte [] data = bos.toByteArray();
16     Question question = new Question("hallo", data, 100,
17         QuestionType.FREETEXT,null);
18     save(question);
19 }
20
21 public File returnImage(byte[] data){
22     ByteArrayInputStream bis = new ByteArrayInputStream(data);
23     BufferedImage bImage2 = null;
24     try {
25         bImage2 = ImageIO.read(bis);
26     } catch (IOException e) {
27         e.printStackTrace();
28     }
29     try {
30         ImageIO.write(bImage2, "jpg", new File("output.jpg") );
31     } catch (IOException e) {
32         e.printStackTrace();
33     }
34     File imageFile = new File("output.png");
35     return imageFile;
}

```

Bild mit Session in DB speichern

Beim zweiten Versuch wurde mithilfe von HibernateUtil.getSessionFactory().openSession() eine Session eröffnet, in der mittels einer FileInputStream das Bild in ein byte[] umgewandelt wird. Danach wird das erzeugte byte[] in der DB gespeichert. Um das byte[] aus der DB zu laden und in ein File umzuwandeln, verwendet man den FileOutputStream.

Dieser Versuch wurde anfangs mit einem Bild, das lokal auf dem Rechner liegt, getestet. Dabei wurde das Bild in der DB gespeichert und wieder geladen. Als ausporbiert wurde,

ob dieser Versuch funktioniert, wenn das Bild vom Frontend mitgesendet wird, trat der Fehler auf, dass man ein mitgeschicktes File nicht so leicht umwandeln kann. Nach dieser Erkenntnis wurde auch dieser Versuch eingestellt.

Listing 28: Bild mit Session in DB speichern

```

1  public static void main( String[] args )
2  {
3      System.out.println("Hibernate save image into database");
4      Session session = HibernateUtil.getSessionFactory().openSession();
5
6      session.beginTransaction();
7
8      //save image into database
9      File file = new File("C:\\mavan-hibernate-image-mysql.gif");
10     byte[] bFile = new byte[(int) file.length()];
11
12     try {
13         FileInputStream fileInputStream = new FileInputStream(file);
14         //convert file into array of bytes
15         fileInputStream.read(bFile);
16         fileInputStream.close();
17     } catch (Exception e) {
18         e.printStackTrace();
19     }
20
21     Avatar avatar = new Avatar();
22     avatar.setImage(bFile);
23
24     session.save(avatar);
25
26     //Get image from database
27     Avatar avatar2 = (Avatar)session.get(Avatar.class, avatar.getAvatarId());
28     byte[] bAvatar = avatar2.getImage();
29
30     try{
31         FileOutputStream fos = new FileOutputStream("C:\\\\test.gif");
32         fos.write(bAvatar);
33         fos.close();
34     }catch(Exception e){
35         e.printStackTrace();
36     }
37
38     session.getTransaction().commit();
39 }
```

10.2.2. CORS Fehler

Nach der Implementierung einiger Hauptfunktionen im Frontend – zum Beispiel das Erstellen eines Fragebogen – wurden diese mit den bereits erstellten REST-Methoden verbunden und getestet. Im Laufe der Testungen sind CORS Fehler aufgetreten. Wie so ein Fehler aussieht wird beispielhaft in Abb. 40 dargestellt. Ein CORS Fehler entsteht beim Laden von Daten aus verschiedenen Quellen. Das Problem wurde durch das Einfügen von folgender Zeile 29 in den application.properties behoben.

The screenshot shows a browser error message. At the top, there is a red warning icon followed by the text: "Quellübergreifende (Cross-Origin) Anfrage blockiert: Die Gleiche-Quelle-Regel verbietet das Lesen der externen Ressource auf http://localhost:8080/api/interviewer. (Grund: CORS-Anfrage schlug fehl)." Below this, there is a blue link labeled "Weitere Informationen". Underneath, there is another line starting with a red error icon and the word "ERROR": "Object { headers: {...}, status: 0, statusText: "Unknown Error", url: "http://localhost:8080/api/interviewer", ok: false, name: "HttpErrorResponse", message: "Http failure response for http://localhost:8080/api/interviewer: 0 Unknown Error", error: error }". On the right side of the message, the text "core.js:6479" is visible.

Abbildung 40.: CORS Fehler

Listing 29: Lösung CORS Error

```
1     quarkus.http.cors=true
```

10.2.3. Cyclic-Object-Error

Cyclic-Object-Error (siehe Abb. 41 als Beispiel) können bei der Konvertierung von Angular-Objekten in JSON-Objekte auftreten. [96], [97]

Der Fehler tritt genau dann auf wenn:

1. ein Objekt auf sich selbst referenziert
2. zwei Objekte aufeinander referenzieren
3. mehreren Objekten, die in einen Kreis aufeinander referenzieren, konvertiert werden sollen

Das JSON-Format an sich unterstützt keine Objektreferenzen, daher versucht JSON.stringify() nicht, diese zu lösen, und schlägt entsprechend fehl. Veranschaulicht würde das JSON-Objekte unendlich lang und in die Tiefe gehen.

```
1 ► ERROR TypeError: cyclic object value
  ► Angular 2
  ► RxJS 16
  checkandsend fragebogen-erstellen.component.ts:57
  ► RxJS 11
  ► Angular 8
1 ► ERROR TypeError: cyclic object value
  ► Angular 2
  ► RxJS 16
  checkandsend fragebogen-erstellen.component.ts:57
  ► RxJS 11
  ► Angular 8
1 ► ERROR TypeError: cyclic object value
```

Abbildung 41.: Cyclic-Object-Error

In der vorliegenden Arbeit wurde versehentlich ein Cyclic-Object erstellt, um die Darstellung von Fragen und zugehörigen Antwortmöglichkeiten zu gewährleisten. Dabei referenzierte die Frage auf die Antwortmöglichkeit und die Antwortmöglichkeit auf die Frage. Die Referenzierung der Antwortmöglichkeiten auf die Frage ist durch das Backend vorgegeben worden. Die Referenzierung der Antwortmöglichkeit auf die Frage wurde für die einfachere Zuordnung und Darstellung der beiden im Frontend eingefügt. (siehe Listing 30)

Listing 30: Klassen mit Cyclic-Error

```
1 import { Interviewer } from "./interviewer";
2 import { AnswerOption } from "./answer-option"
3
4 export class Questionnaire {
5   constructor(
```

```

6     public id = 0, public name :string = "",
7     public description : string = "",
8     public isPublic: Boolean = false,
9     public interviewer : Interviewer = new Interviewer(1, "HTL Leonding"),
10    public AnswerOptions : AnswerOption[] = []
11  ){}
12 }
13
14 import { Question } from "./question";
15
16 export class AnswerOption {
17   constructor(
18     public id = 0, public text :string = "",
19     public value = 0,
20     public sequenceNumber = 0,
21     public isCorrectAnswer = true,
22     public question : Question = new Question()
23   ){}
24 }
```

Für die Übertragung von Daten zwischen Frontend und Backend wurde beschlossen JSON zu verwenden. Dieses Format kann diese Referenzierung nicht darstellen und die einmalige Verwendung von cycle.js, einer Bibliothek, die solche Referenzierung darstellen kann, wäre zu aufwändig gewesen.

Für die Lösung dieses Problem wurde eine neue Klasse erstellt und beide Objekte (Frage und Antwortmöglichkeiten-Liste) einander zugeordnet. Somit konnte die Referenzierung der Antwortmöglichkeit auf die Frage entfernt und der Cyclic-Object-Error gelöst werden.

Listing 31: Klassen ohne Cyclic-Error

```

1   import { AnswerOption } from "./answer-option"
2   import { Question } from "./question"
3
4   export class Administration {
5     constructor(
6       public question : Question = new Question(),
7       public answerOptionArray : Array<AnswerOption>,
8       public aocounter = 1
9     ){}
10 }
```

10.3. Entwurfsentscheidungen

10.3.1. Verwendung von CSS, SCSS, SASS oder Less

Die Entscheidung welches Stylesheet zur Layoutierung von Angular verwendet wird, war von Anfang an klar. Technologien wie SASS und Less (siehe Kapitel 8.1.4) kamen nicht in Frage, da dem Team diese Technologien eher unbekannt sind. Übrig blieben folgend CSS und SCSS. Die Entscheidung wurde für SCSS getroffen, da es eine Obermenge von CSS mit mehr Funktionen ist.

10.3.2. Entscheidung zwischen Angular und anderen Frameworks

Nicht nur wird Angular von vielen Unternehmen für ihre Projekte verwendet, sondern liefert es auch im Gegensatz zu anderen Frameworks eine vorgegebene Struktur, die die Verwendung erleichtern. Auch bietet Angular die benötigte Leistungsfähigkeit für die Erstellung des Frontends. Die Entscheidung wurde zudem durch längere Erfahrung mit dem Framework Angular und Mangel an Erfahrung mit anderen Frameworks wie React oder Vue verstärkt. [98], [99], [100]

10.3.3. Entscheidung Java

Wir haben uns einerseits für diese Programmiersprache entschieden, einerseits da sie uns seit der ersten Klasse an der HTL Leonding beigebracht wird und wir dadurch gewisse Grundkenntnisse in Java besitzen. Andererseits, da unser Diplomarbeitsbetreuer sehr gute Kenntnisse in Java besitzt und wir bei Fragen uns an ihn wenden können. Weiteres existiert im Internet eine große Community, die Hilfestellungen oder Problemlösungen dokumentiert und bereitstellt.

10.3.4. Entscheidung PostgreSql

Wir haben uns für PostgreSql entschieden, da PostgreSql dem SQL-Standard entspricht. Außerdem ist PostgreSql nicht an ein Betriebssystem gebunden.

11. Implementierung

11.1. Endpoints

11.1.1. Transaction

find all

GET `https://localhost:8080/api/transaction`

Rückgabewert Zurückgegeben wird eine Liste von allen Transactionen als JSON Format.

find By Id

GET `https://localhost:8080/api/transaction/id/{id}`

Parameter

- *id*: eindeutige Nummer einer Transaction

Rückgabewert Zurückgegeben wird das gefundene Transaction Objekt als JSON Format. Wenn keines mit der angegebenen *id* gefunden wird, so wird der HTTP Error *No Content (204)* zurückgegeben.

create Transaction

POST `https://localhost:8080/api/transaction`

Body

- Transaction Objekt im JSON Format
- UriInfo

Aktion In der Datenbank wird das übergebene Transaction Objekt persistiert.

Rückgabewert Zurückgegeben wird die URI, über welche das soeben erstellte Objekt über einen anderen Endpoint angefordert werden kann.

get Transaction with code

GET `https://localhost:8080/api/transaction/code/{code}`

Parameter

- `code`: String mit dem Code einer Transaction

Rückgabewert Zurückgegeben wird das gefundene Transaction Objekt als JSON Format. Wenn keines mit der angegebenen `id` gefunden wird, so wird der HTTP Error *No Content* (204) zurückgegeben.

get Survey with Transaction

POST `https://localhost:8080/api/transaction/getSurvey`

Body

- Transaction Objekt im JSON Format

Aktion Es werden alle Survey Objekte mit einer referenz auf das übergebene Transaction Objekt gesucht.

Rückgabewert Zurückgegeben wird eine Liste von Survey Objekten.

update Transaction

POST `https://localhost:8080/api/transaction/{id}`

Body

- `id`: eindeutige Nummer einer Transaction
- Transaction Objekt im JSON Format

Aktion In der Datenbank wird das wird das bereits persistierte Objekt aktualisiert.

Rückgabewert Zurückgegeben wird das persistierte Objekt. Wenn keines mit der angegebenen `id` gefunden wird, so wird der HTTP Error Bad Request (400) zurückgegeben.

delete Transaction

DELETE `https://localhost:8080/api/transaction/{id}`

Body

- *id*: eindeutige Nummer einer Transaction

Aktion Es wird das Transaction Objekt mit der übergebenen Id gesucht und anschließend aus der Datenbank gelöscht.

Rückgabewert Nach einem erfolgreichen löschen Vorgang wird der http status *ok* (*200*) zurückgegeben.

generate Transaction

GET *https://localhost:8080/api/transaction/generate/{number}/{surveyId}*

Parameter

- *surveyId*: eindeutige Nummer einer Survey
- *number*: Anzahl wie viele Transactions generiert werden müssen

Aktion Es wird die Anzahl der gewünschten Transactions erstellt und in der Datenbank persistiert

Rückgabewert Zurückgegeben wird ein File in dem alle Codes der neu generierten Transactions stehen.

11.1.2. Survey

find All

GET *https://localhost:8080/api/survey*

Rückgabewert Zurückgegeben wird eine Liste von allen Survey Objekten als JSON Format.

find By Id

GET *https://localhost:8080/api/survey/id/{id}*

Parameter

- *id*: eindeutige Nummer einer Survey

Rückgabewert Zurückgegeben wird das gefundene Survey Objekt als JSON Format. Wenn keines mit der angegebenen *id* gefunden wird, so wird der HTTP Error *No Content* (204) zurückgegeben.

create Survey

POST *https://localhost:8080/api/survey*

Body

- Survey Objekt im JSON Format
- UriInfo

Aktion In der Datenbank wird das übergebene Survey Objekt persistiert.

Rückgabewert Zurückgegeben wird die URI, über welche das soeben erstellte Objekt über einen anderen Endpoint angefordert werden kann.

find Survey by Interviewer Id

GET *https://localhost:8080/api/survey/{interviewerId}*

Parameter

- *interviewerId*: eindeutige Nummer eines Interviewers

Rückgabewert Zurückgegeben wird das gefundene Survey Objekt als JSON Format.

find Questionnaire with Survey Id

GET *https://localhost:8080/api/survey/questionnaire/{surveyId}*

Parameter

- *surveyId*: eindeutige Nummer einer Survey

Rückgabewert Zurückgegeben wird das gefundene Questionnaire Objekt als JSON Format.

update Survey

POST *https://localhost:8080/api/survey/{id}*

Body

- id: eindeutige Nummer einer Survey
- Survey Objekt im JSON Format

Aktion In der Datenbank wird das wird das bereits persistierte Objekt aktualisiert.

Rückgabewert Zurückgegeben wird das persistierte Objekt. Wenn keines mit der angegeben id gefunden wird, so wird der HTTP Error Bad Request (400) zurückgegeben.

delete Survey

DELETE `https://localhost:8080/api/survey/{id}`

Body

- id: eindeutige Nummer einer Survey

Aktion Es wird das Survey Objekt mit der übergebenen Id gesucht und anschließend aus der Datenbank gelöscht.

Rückgabewert Nach einem erfolgreichen löschen Vorgang wird der http status *ok* (200) zurückgegeben.

Auswertung

GET `https://localhost:8080/api/survey/evaluation/{surveyId}`

Parameter

- *surveyId*: eindeutige Nummer einer Survey

Rückgabewert Zurückgegeben wird ein File mit der Auswertung.

11.1.3. Question

find All

GET `https://localhost:8080/api/question`

Rückgabewert Zurückgegeben wird eine Liste von allen Question Objekten als JSON Format.

find By Id

GET `https://localhost:8080/api/question/id/{id}`

Parameter

- *id*: eindeutige Nummer einer Question

Rückgabewert Zurückgegeben wird das gefundene Question Objekt als JSON Format. Wenn keines mit der angegebenen *id* gefunden wird, so wird der HTTP Error *No Content* (204) zurückgegeben.

create Question

POST `https://localhost:8080/api/question`

Body

- Question Objekt im JSON Format
- UriInfo

Aktion In der Datenbank wird das übergebene Question Objektpersistiert.

Rückgabewert Zurückgegeben wird die URI, über welche das soeben erstellte Objekt über einen anderen Endpoint angefordert werden kann.

update Question

POST `https://localhost:8080/api/question/{id}`

Body

- *id*: eindeutige Nummer einer Question
- Question Objekt im JSON Format

Aktion In der Datenbank wird das wird das bereits persistierte Objekt aktualisiert.

Rückgabewert Zurückgegeben wird das persistierte Objekt. Wenn keines mit der angegebenen id gefunden wird, so wird der HTTP Error Bad Request (400) zurückgegeben.

delete Question**DELETE** *https://localhost:8080/api/question/{id}***Body**

- *id*: eindeutige Nummer einer Question

Aktion Es wird das Question Objekt mit der übergebenen Id gesucht und anschließend aus der Datenbank gelöscht.**Rückgabewert** Nach einem erfolgreichen löschen Vorgang wird der http status *ok (200)* zurückgegeben.

11.1.4. Questionnaire

find All**GET** *https://localhost:8080/api/questionnaire***Rückgabewert** Zurückgegeben wird eine Liste von allen Questionnaire Objekten als JSON Format.**find By Id****GET** *https://localhost:8080/api/questionnaire/id/{id}***Parameter**

- *id*: eindeutige Nummer einer Questionnaire

Rückgabewert Zurückgegeben wird das gefundene Questionnaire Objekt als JSON Format. Wenn keines mit der angegebenen *id* gefunden wird, so wird der HTTP Error *No Content (204)* zurückgegeben.**create Questionnaire****POST** *https://localhost:8080/api/questionnaire***Body**

- Questionnaire Objekt im JSON Format
- UriInfo

Aktion In der Datenbank wird das übergebene Questionnaire Objekt persistiert.

Rückgabewert Zurückgegeben wird die URI, über welche das soeben erstellte Objekt über einen anderen Endpoint angefordert werden kann.

find public Questionnaires

GET `https://localhost:8080/api/questionnaire/public`

Rückgabewert Zurückgegeben wird eine Liste von Questionnaire Objekten, bei denen das Attribut isPublic auf true gestezt ist.

get public and owned Questionnaires

GET `https://localhost:8080/api/questionnaire/public/{interviewerId}`

Body

- interviewerId: eindeutige Nummer eines Interviewers

Rückgabewert Zurückgegeben wird eine Liste von Questionnaire Objekten, bei denen das Attribut isPublic auf true gestezt ist und eine referenz auf das übergebene Interviewer Onjekt besteht.

find Questionnaire By Interviwer id

GET `https://localhost:8080/api/questionnaire/interviewer/{interviewerId}`

Body

- interviewerId: eindeutige Nummer eines Interviewers

Rückgabewert Zurückgegeben wird eine Liste von Questionnaire Objekten, bei denen eine referenz auf das übergebene Interviewer Onjekt besteht.

duplicate Questionnaire

POST `https://localhost:8080/api/duplicateQuestionnaire`

Body

- Questionnaire Objekt im JSON Format

Aktion Es wird ein duplcate des übergebenen Questionnaire Objekts in der Datenbank persistiert.

Rückgabewert Zurückgegeben wird das duplizierte Questionnaire Objekt.

update Questionnaire

POST `https://localhost:8080/api/questionnaire/{id}`

Body

- id: eindeutige Nummer einer Questionnaire
- Questionnaire Objekt im JSON Format

Aktion In der Datenbank wird das wird das bereits persistierte Objekt aktualisiert.

Rückgabewert Zurückgegeben wird das persistierte Objekt. Wenn keines mit der angegeben id gefunden wird, so wird der HTTP Error Bad Request (400) zurückgegeben.

delete Questionnaire

DELETE `https://localhost:8080/api/questionnaire/{id}`

Body

- id: eindeutige Nummer eines Questionnaire

Aktion Es wird das Questionnaire Objekt mit der übergebenen Id gesucht und anschließend aus der Datenbank gelöscht.

Rückgabewert Nach einem erfolgreichen löschen Vorgang wird der http status *ok* (200) zurückgegeben.

11.1.5. Interviewer

find All

GET `https://localhost:8080/api/interviewer`

Rückgabewert Zurückgegeben wird eine Liste von allen Interviewer Objekten als JSON Format.

find By Id

GET `https://localhost:8080/api/interviewer/id/{id}`

Parameter

- *id*: eindeutige Nummer eines Interviewers

Rückgabewert Zurückgegeben wird das gefundene Interviewer Objekt als JSON Format. Wenn keines mit der angegebenen *id* gefunden wird, so wird der HTTP Error *No Content* (204) zurückgegeben.

create Question

POST `https://localhost:8080/api/interviewer`

Body

- Interviewer Objekt im JSON Format
- UriInfo

Aktion In der Datenbank wird das übergebene Interviewer Objektpersistiert.

Rückgabewert Zurückgegeben wird die URI, über welche das soeben erstellte Objekt über einen anderen Endpoint angefordert werden kann.

update Interviewer

POST `https://localhost:8080/api/interviewer/{id}`

Body

- *id*: eindeutige Nummer einer Interviewer
- Interviewer Objekt im JSON Format

Aktion In der Datenbank wird das wird das bereits persistierte Objekt aktualisiert.

Rückgabewert Zurückgegeben wird das persistierte Objekt. Wenn keines mit der angegebenen id gefunden wird, so wird der HTTP Error Bad Request (400) zurückgegeben.

delete Interviewer**DELETE** *https://localhost:8080/api/interviewer/{id}***Body**

- *id*: eindeutige Nummer einer Interviewer

Aktion Es wird das Interviewer Objekt mit der übergebenen Id gesucht und anschließend aus der Datenbank gelöscht.

Rückgabewert Nach einem erfolgreichen löschen Vorgang wird der http status *ok (200)* zurückgegeben.

11.1.6. ChosenOption

find All**GET** *https://localhost:8080/api/chosenOption*

Rückgabewert Zurückgegeben wird eine Liste von allen ChosenOption Objekten als JSON Format.

find By Id**GET** *https://localhost:8080/api/chosenOption/id/{id}***Parameter**

- *id*: eindeutige Nummer einer ChosenOption

Rückgabewert Zurückgegeben wird das gefundene ChosenOption Objekt als JSON Format. Wenn keines mit der angegebenen *id* gefunden wird, so wird der HTTP Error *No Content (204)* zurückgegeben.

create ChosenOption**POST** *https://localhost:8080/api/chosenOption***Body**

- ChosenOption Objekt im JSON Format
- UriInfo

Aktion In der Datenbank wird das übergebene ChosenOption Objekt persistiert.

Rückgabewert Zurückgegeben wird die URI, über welche das soeben erstellte Objekt über einen anderen Endpoint angefordert werden kann.

update ChosenOption

POST `https://localhost:8080/api/chosenOption/{id}`

Body

- id: eindeutige Nummer einer ChosenOption
- ChosenOption Objekt im JSON Format

Aktion In der Datenbank wird das wird das bereits persistierte Objekt aktualisiert.

Rückgabewert Zurückgegeben wird das persistierte Objekt. Wenn keines mit der angegeben id gefunden wird, so wird der HTTP Error Bad Request (400) zurückgegeben.

delete ChosenOption

DELETE `https://localhost:8080/api/chosenOption/{id}`

Body

- id: eindeutige Nummer einer ChosenOption

Aktion Es wird das ChosenOption Objekt mit der übergebenen Id gesucht und anschließend aus der Datenbank gelöscht.

Rückgabewert Nach einem erfolgreichen löschen Vorgang wird der http status *ok* (200) zurückgegeben.

11.1.7. Answer

find All

GET `https://localhost:8080/api/answer`

Rückgabewert Zurückgegeben wird eine Liste von allen Answer Objekten als JSON Format.

find By Id

GET `https://localhost:8080/api/answer/id/{id}`

Parameter

- *id*: eindeutige Nummer einer Answer

Rückgabewert Zurückgegeben wird das gefundene Answer Objekt als JSON Format. Wenn keines mit der angegebenen *id* gefunden wird, so wird der HTTP Error *No Content* (204) zurückgegeben.

create Answer

POST `https://localhost:8080/api/answer`

Body

- Answer Objekt im JSON Format
- UriInfo

Aktion In der Datenbank wird das übergebene Answer Objekt persistiert.

Rückgabewert Zurückgegeben wird die URI, über welche das soeben erstellte Objekt über einen anderen Endpoint angefordert werden kann.

update Answer

POST `https://localhost:8080/api/answer/{id}`

Body

- *id*: eindeutige Nummer einer Answer
- Answer Objekt im JSON Format

Aktion In der Datenbank wird das wird das bereits persistierte Objekt aktualisiert.

Rückgabewert Zurückgegeben wird das persistierte Objekt. Wenn keines mit der angegebenen id gefunden wird, so wird der HTTP Error Bad Request (400) zurückgegeben.

delete Answer**DELETE** *https://localhost:8080/api/answer/{id}***Body**

- *id*: eindeutige Nummer einer Answer

Aktion Es wird das Answer Objekt mit der übergebenen Id gesucht und anschließend aus der Datenbank gelöscht.

Rückgabewert Nach einem erfolgreichen löschen Vorgang wird der http status *ok (200)* zurückgegeben.

11.1.8. AnswerOption

find All**GET** *https://localhost:8080/api/answerOption*

Rückgabewert Zurückgegeben wird eine Liste von allen AnswerOption Objekten als JSON Format.

find By Id**GET** *https://localhost:8080/api/answerOption/id/{id}***Parameter**

- *id*: eindeutige Nummer einer AnswerOption

Rückgabewert Zurückgegeben wird das gefundene AnswerOption Objekt als JSON Format. Wenn keines mit der angegebenen *id* gefunden wird, so wird der HTTP Error *No Content (204)* zurückgegeben.

create Answer**POST** *https://localhost:8080/api/answerOption***Body**

- AnswerOption Objekt im JSON Format
- UriInfo

Aktion In der Datenbank wird das übergebene AnswerOption Objekt persistiert.

Rückgabewert Zurückgegeben wird die URI, über welche das soeben erstellte Objekt über einen anderen Endpoint angefordert werden kann.

update AnswerOption

POST *https://localhost:8080/api/answerOption/{id}*

Body

- id: eindeutige Nummer einer AnswerOption
- AnswerOption Objekt im JSON Format

Aktion In der Datenbank wird das wird das bereits persistierte Objekt aktualisiert.

Rückgabewert Zurückgegeben wird das persistierte Objekt. Wenn keines mit der angegeben id gefunden wird, so wird der HTTP Error Bad Request (400) zurückgegeben.

delete AnswerOption

DELETE *https://localhost:8080/api/answerOption/{id}*

Body

- id: eindeutige Nummer einer AnswerOption

Aktion Es wird das AnswerOption Objekt mit der übergebenen Id gesucht und anschließend aus der Datenbank gelöscht.

Rückgabewert Nach einem erfolgreichen löschen Vorgang wird der http status *ok (200)* zurückgegeben.

12. Projektorganisation

12.1. Anforderungen

Um die vorliegende Arbeit bestmöglich und ohne größere projektorganisatorische Probleme bewältigen zu können, mussten Hilfsmittel gefunden werden, die die folgenden Anforderungen erfüllen können.

- Gemeinsame Codebasis
- Versionierung
- Visualisierung Arbeitsfortschritt
- Arbeitszuteilung
- Zu erledigende Aufgaben

12.2. Mögliche Hilfsmittel

12.2.1. GitHub

GitHub ist eine webbasierte Versionskontroll- und Kollaborationsplattform für Softwareentwickler. GitHub wird verwendet, um den Code für ein Projekt zu speichern und den kompletten Verlauf aller Änderungen an diesem Code zu verfolgen. Es ermöglicht Entwicklern eine reibungslose Zusammenarbeit an einem Projekt, indem es Werkzeuge für die Verwaltung möglicherweise widersprüchlicher Änderungen von mehreren Entwicklern bereitstellt. [101]

12.2.2. GitHub Desktop

GitHub Desktop ist eine Anwendung, die es ermöglicht, mit GitHub über eine grafische Benutzeroberfläche zu interagieren, statt über die Eingabeaufforderung oder einen Webbrowsert. Wie in anderen GitHub-Anbindungen zeigt der GitHub Desktop

Änderungen aller Art (Löschen, Hinzugefügt und Geändert) grafisch an und ermöglicht die Interaktion mit einfachen Klicks. [102]

12.2.3. GitHub Projektboards und Issues

Projektboards auf GitHub helfen die Arbeit zu organisieren und zu priorisieren. Projektboards können für die Arbeit an bestimmten Funktionen, umfassende Roadmaps oder sogar zur Erstellung von Release-Checklisten verwendet werden. Projektboards bieten die Flexibilität, individuelle Arbeitsabläufe zu erstellen, die den Bedürfnissen angepasst werden können. Projektboards bestehen aus Issues, Pull-Requests und Notizen, die als Cards in Spalten kategorisiert werden. GitHub bietet die Möglichkeit Meilensteine anzulegen und 1 oder mehrere Issues mit dem Meilenstein zu verlinken. [103]

12.2.4. YouTrack

YouTrack ist eine eigenständige, browserbasierte Bug-Tracker-, Problemverfolgungs- und Projektmanagementssoftware, die von JetBrains entwickelt wurde. Der Schwerpunkt liegt auf der abfragebasierten Fehlersuche mit Auto vervollständigung, der Bearbeitung von Fehlern in Stapeln, der Anpassung der Fehlerattribute und der Erstellung benutzerdefinierter Workflows.

In YouTrack lassen sich Sprints frei definieren und Aufgaben bzw. Bugs mit Prioritäten versehen. Diese können dann je nach Fertigstellungsgrad in den verschiedenen Schwimmbahnen angeordnet werden. Mithilfe eines GitHub Accounts und eines Authentifizierungstokens kann YouTrack mit einem GitHub Repository gekoppelt werden. YouTrack bietet zur Visualisierung ein Burn Down Chart an. Ein weiterer Vorteil von YouTrack ist es, dass Tasks anderen Tasks untergeordnet werden können. [104]

Kanban

Kanban ist eine Arbeitsmanagement-Methode, deren Hauptzweck die Minimierung von Aktivitäten ist, ohne dabei zu Verlusten zu führen und ohne die Produktivität zu beeinträchtigen. ([105], [106]) Auf eine gewisse Art lässt sich Scrum als eine mögliche Implementierung von Kanban ansehen. Um Kanban erfolgreich einzusetzen zu können sollten folgende Praktiken befolgt werden: [107]

- Workflow visualisieren

- Laufende Arbeit begrenzen
- Workflow-Management
- Prozessrichtlinien ausformulieren
- Laufendes Feedback

Scrum

Scrum ist ein leichtgewichtiges, interaktives und inkrementelles System für die Entwicklung, Durchführung und Aufrechterhaltung komplexer Projekte. Das System stellt die Annahmen des traditionellen, sequentiellen Ansatzes (z.B. Wasserfallmodell) in Frage. Scrum verfolgt das Ziel, die Fähigkeit des Teams zu maximieren, auf neue Anforderungen zu reagieren und sich an sich entwickelnde Technologien und veränderte Marktbedingungen anzupassen.

Der Ablauf von Scrum läuft in der Regel wie folgt ab. Zuerst werden alle Anforderungen und Aufgaben für das Projekt ermittelt. Diese können jedoch später noch erweitert oder geändert werden. Dann werden für einen Sprint bestimmte Aufgaben aufgeteilt und bearbeitet. Ist ein Sprint (Arbeitszyklus) abgeschlossen, wird das Ergebnis evaluiert. Dieser Prozess wird solange wiederholt bis das Projekt fertiggestellt ist. Als Beispiel für Scrum siehe Abb. 42. [108]

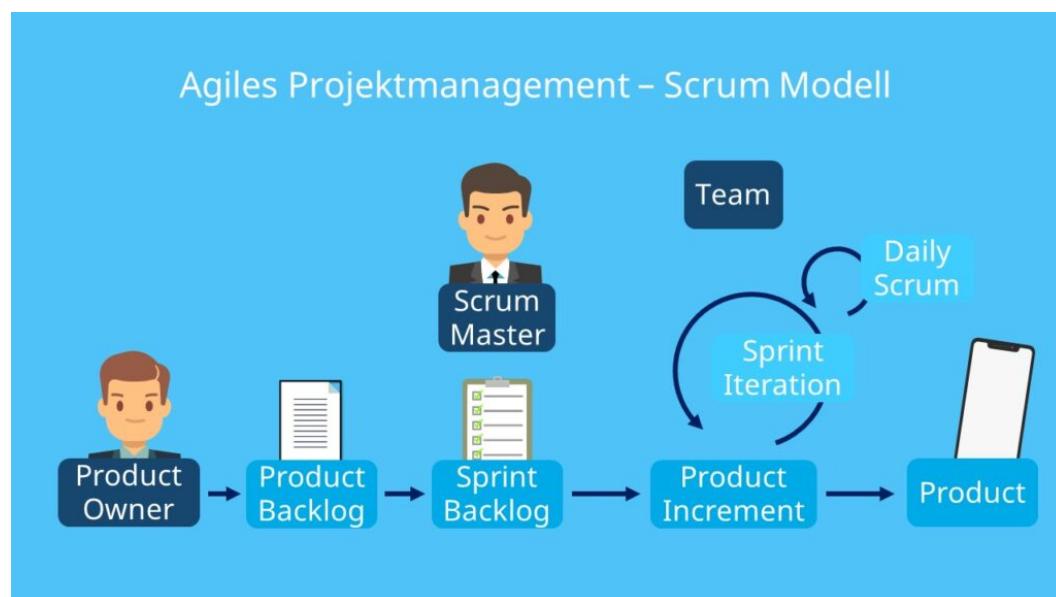


Abbildung 42.: Veranschaulichung von Scrum <https://studyflix.de/wirtschaft/scrum-methode-3426>

Product Backlog

Der Product Backlog enthält sämtliche Anforderungen an das Produkt. Diese Anforderungen werden Schritt für Schritt in Sprints bearbeitet. In einem separaten Meeting wird dann entschieden, welche Anforderungen aus dem Product Backlog im jeweiligen Sprint abgearbeitet werden. Der Backlog kann Qualitätsanforderungen, Funktionale Anforderungen, User Stories, Fehler (Bugs) und Verbesserungen enthalten. [109]

Burn Down Chart

Burn Down Charts visualisieren die Arbeit, die zum aktuellen Zeitpunkt noch zu erledigen ist. Es lässt sich anhand des Diagrammes die Arbeitsgeschwindigkeit des Teams ableiten. Das Burn Down Chart eignet sich weiters dafür, die Interaktionen bzw. Sprints zu kontrollieren und wenn nötig kontrollierend eingreifen. [110], [111]

GitHub-Tokens

Persönliche Zugangs-Token (PATs) sind eine Alternative zur Verwendung von Passwörtern für die Authentifizierung bei GitHub. Als Sicherheitsvorkehrung entfernt GitHub automatisch persönliche Zugangstoken, die seit einem Jahr nicht mehr verwendet wurden. Seit Juli 2020 sind Authentifizierungstoken statt Passwörter zu verwenden. [112], [113]

12.3. Problemlösung

12.3.1. GitHub

Die Wahl Git zur Versionskontrollierung und Kollaboration zu verwenden, ist aus offensichtlichen Gründen der einfachen Zusammenarbeit und Codezusammenführung getroffen worden. Für den Provider wurde sich für GitHub entschieden, da das Team viel Erfahrung im Umgang mit der Plattform besitzt und daher der Gebrauch naheliegend ist. In Abbildung (siehe Abb. 43) wird die Verwendung von GitHub gezeigt. In Abbildung (siehe Abb. 44) wird die gemeinsame und parallele Arbeit dargestellt. Das Zusammenfügen des Quellcodes konnte hauptsächlich durch Git automatisch durchgeführt werden. Einige Zusammenfügungen mussten allerdings manuell abgewickelt werden.

Online-Fragebogen, Diplomarb
2021/22, Weissengruber, Raffeine
[htl-leonding-project.github.io/2021-da-leo-survey](#)

About

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Contributors 4

- weissengruberNina
- chrisyraffaeiner
- turnerisabel

Abbildung 43.: GitHub Repository von Leo-Survey

Abbildung 44.: GitHub Repository von Leo-Survey Commits

12.3.2. GitDesktop

Die Abbildung (siehe Abb. 45) zeigt die Verwendung von GitHub Desktop. GitHub Desktop wurde hauptsächlich aus dem Grund verwendet, da zwei einzeln geöffnete Anwendungen (IntelliJ für das Backend) und (Visual Studio Code für das Frontend) die Änderungen des jeweils anderen Programmes nicht erkennen. Durch einen Neustart

konnten die Änderungen dann erkannt werden. Da dies allerdings mit Aufwand verbunden ist wurde drauf verzichtet und GitHub Desktop verwendet. Die Anwendung löste dieses Problem.

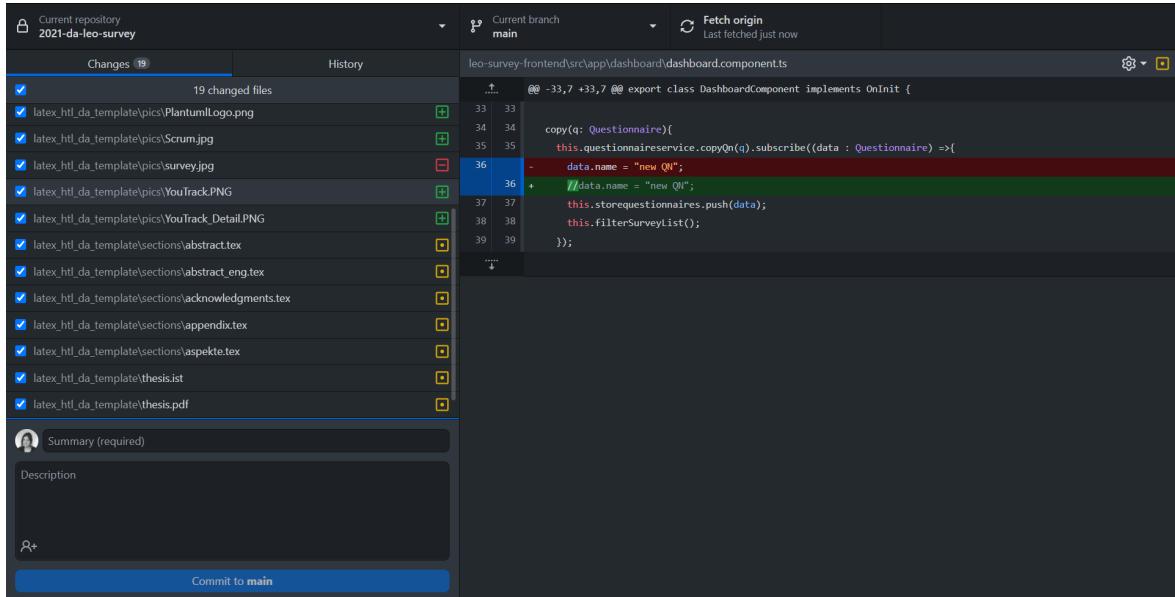


Abbildung 45.: GitHub Desktop Beispiel

12.3.3. YouTrack

Obwohl für das Projekt GitHub verwendet wurde, wurde sich gegen die Verwendung von GitHub Projektboards, Issues und Meilensteine entschieden und für den Gebrauch von YouTrack. Hauptgrund dafür war, dass im Gegensatz zu GitHub, Tasks, Sprints etc. in YouTrack auf einer Seite übersichtlicher dargestellt werden konnten und auch zusätzliche Features wie das Burn Down Chart zu Verfügung stehen. Verbindet man YouTrack außerdem mit einem GitHub Repository verfügt man über die Funktion von automatisch einzuordnenden Tasks. Die Projektboards wurden mit der YouTrack-Vorlage Kanban erstellt. Diese Vorlage erstellt automatisch 4 Schwimmbahnen (Open, In Progress, To Verify und Done). Die Vorlage kann bearbeitet werden. Es wurde allerdings auf eine Änderung verzichtet, da die Vorlage für die vorliegende Arbeit ausgereicht hat.

Die Abbildung (siehe Abb. 46) zeigt beispielhaft einen Sprint der Arbeit.

12.3.4. GitHub-Anbindung in YouTrack

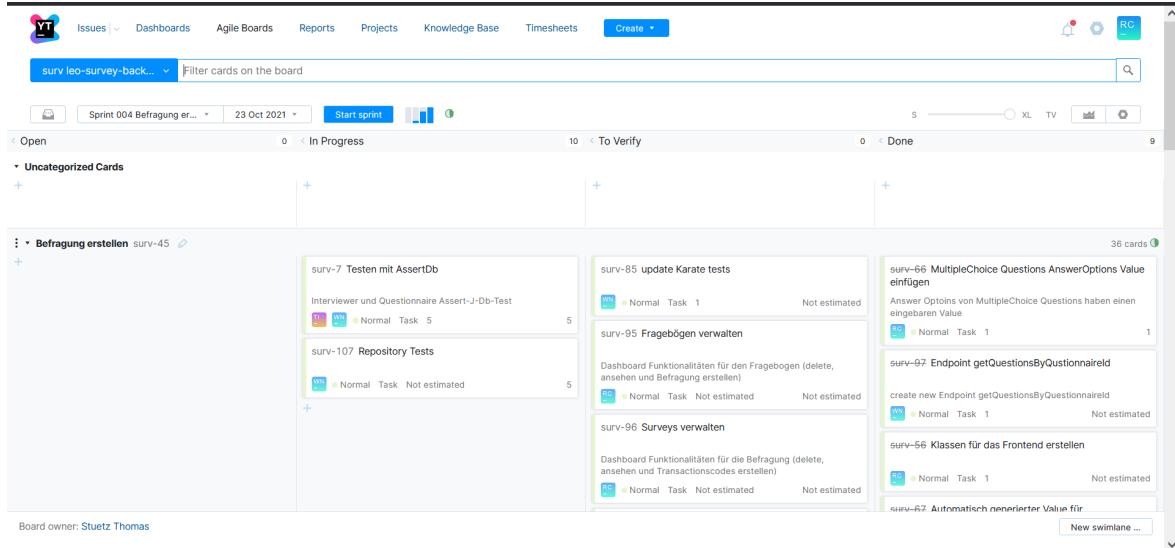


Abbildung 46.: YouTrack Sprint Beispiel

Für designtechnische Aspekte, wie die Erstellung des Logos, Hintergrundbildes und ScreenDesigns wurde ein eigenes Projektboard erstellt. (siehe Abb. 47)

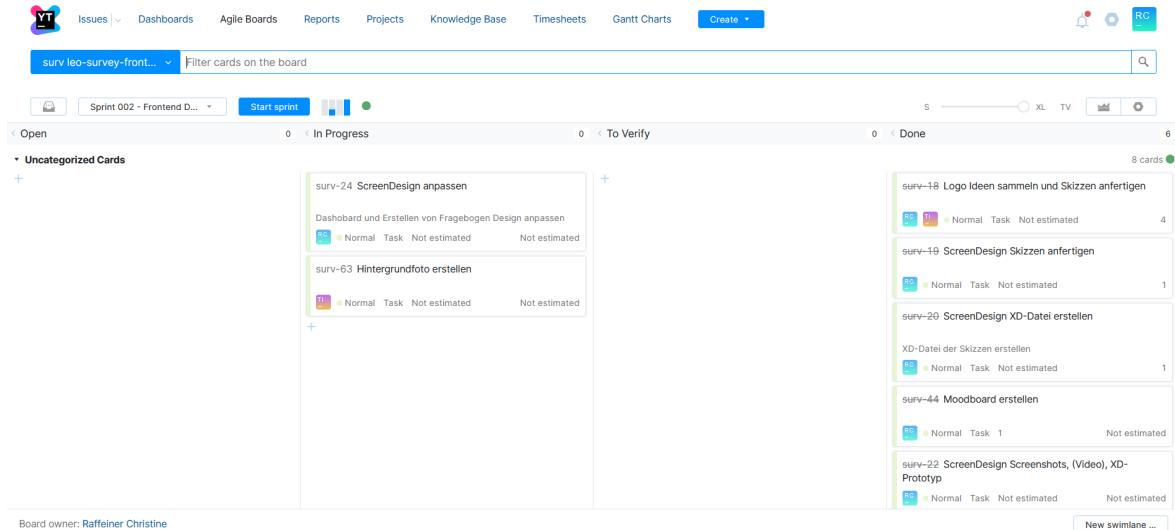


Abbildung 47.: GitHub-Anbindung in YouTrack

Abbildung (siehe Abb. 48) zeigt beispielsweise ein Burn Down Chart aus der vorliegenden Arbeit. Das Chart zeigt den Arbeitsaufwand des Deployments. Die grüne Line gibt den idealen Verlauf des noch zu tätigen Arbeitsaufwandes an, sodass am Ende des Sprints alle Aufgaben erledigt sind. Die blaue Line zeigt den tatsächlichen Verlauf des Arbeitsaufwandes und gibt Auskunft darüber, dass bis zum Ende des Sprints noch Arbeit zu erledigen ist. Zudem lässt sich daraus schlussfolgern, dass das Team die

Vorgaben, bis zu diesem Zeitpunkt, eingehalten hat. (blaue Linie verläuft unter der grünen)

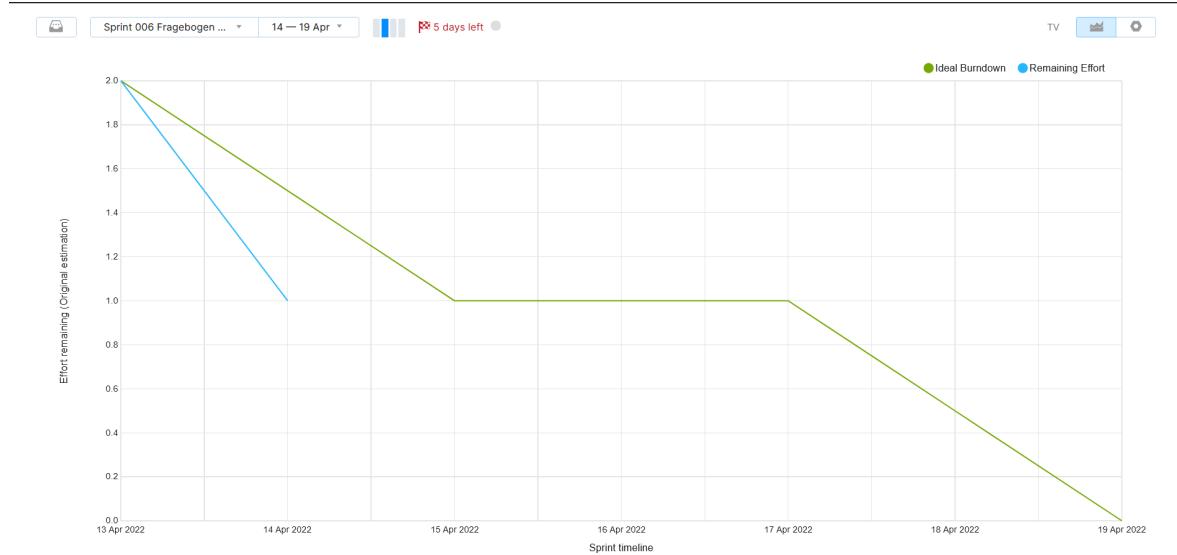


Abbildung 48.: YouTrack Burn Down Chart Beispiel

Die Abbildung (siehe Abb. 49) zeigt die Anbindungen von YouTrack an das GitHub Repository, die dazu dient, beide Programme miteinander zu vereinen. Wird ein Task in einer Commit-Message erwähnt, werden diese sonstigen Änderungen in der Detailansicht des Tasks angezeigt. Jeder angelegte Task bekommt eine Kennung in der Form #surv-<zahl> (in diesem Beispiel #surv-7) und wird automatisch bei Erwähnung in einer Commit-Nachricht erkannt. Folgt auf die Kennung noch eine Bezeichnung einer Schwimmbahn wird der Task automatisiert in die angegebene Schwimmbahn geschoben (als Beispiel #surv-7 To Verify). [114], [115]

The figure shows a screenshot of a GitHub commit history for issue surv-107 in YouTrack. The commit history includes several commits from Weissengruber Nina and others, with various annotations and links. On the right side, there is a detailed view of the issue card for surv-107, showing project information, priority, type, state, assignee, sprints, ideal days, original estimation, watchers, and boards.

Project	2021_4ahitn_da_
Priority	Normal
Type	Task
State	In Progress
Assignee	Weissengruber Nina
Sprints	Sprint 004 Befragung erstellen
Ideal days	5
Original estimation	5
Watchers	0 > Watch issue
Boards	+ Add to board

Abbildung 49.: GitHub-Anbindung in YouTrack

13. ScreenDesign

13.1. Handskizzen

Aus den Erkenntnissen der Umfeldanalyse wurden in der ersten Instanz einfache Handskizzen (siehe Abb. 50 und 51) für das Design angefertigt. Bereits in den ersten Skizzen kristallisierten sich spezielle Ideen, wie zum Beispiel das Dashboard, heraus.

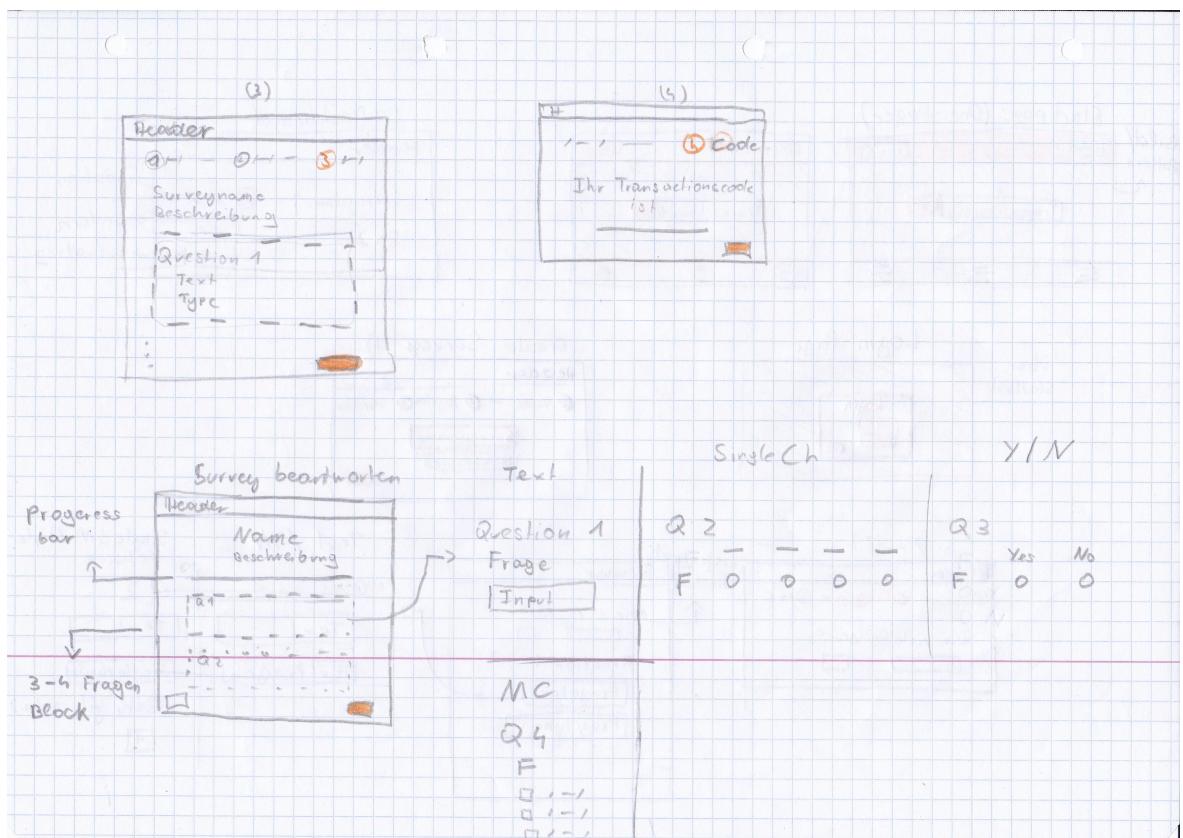


Abbildung 50.: Skizze Seite 1

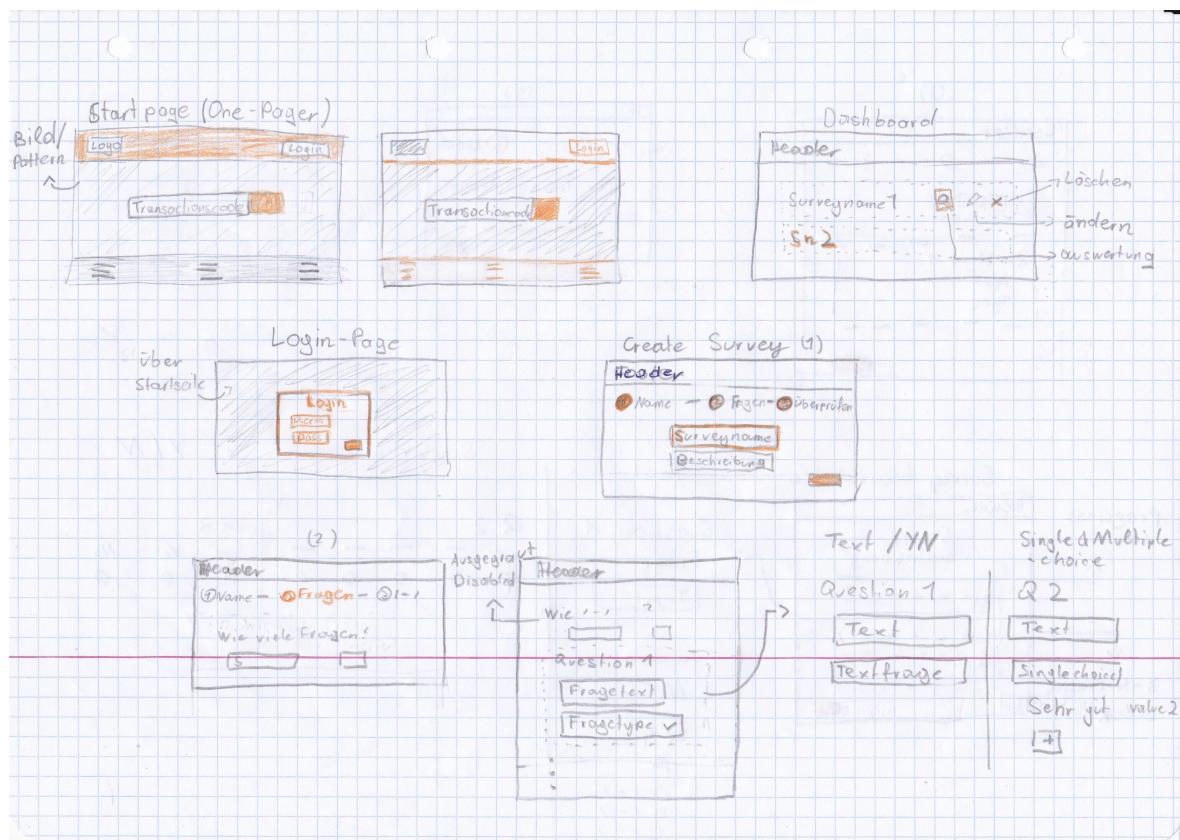


Abbildung 51.: Skizze Seite 2

13.2. Screendesign Prototyp

Als wichtiger Zwischenschritt zwischen den Skizzen und dem Prototypen wurde ein Moodboard erstellt, dass die grundlegende Stimmung der Webseite, wichtige Elemente der zukünftigen Webseite, Schriften und Farben darstellt. Im Moodboard (siehe Abb. 52) wurden die wichtigsten Elemente für die stufenweise Erstellung des Fragebogens, die Fortschrittsanzeige für die Beantwortung einer Umfrage und die wichtigsten Funktionen des Dashboard (durch die Icons symbolisiert) erstmals dargestellt.

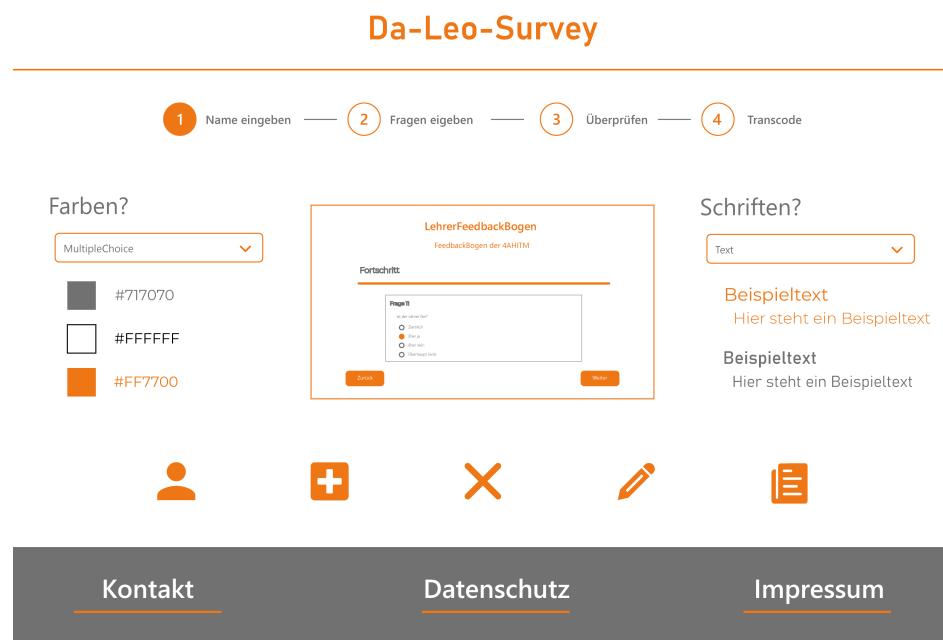


Abbildung 52.: Moodboard Leo-Survey

Nach der Erstellung des Moodboards wurden die Skizzen im Programm Adobe XD umgesetzt. Es wurde auf das zukünftige Design mit Angular Materials Rücksicht genommen und die Elemente möglichst in dem Design von Materials erstellt.

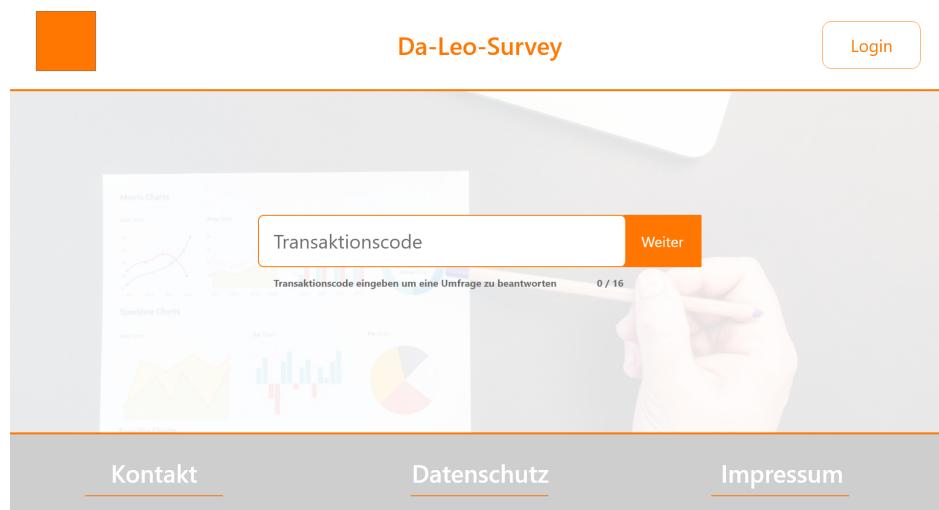


Abbildung 53.: ScreenDesign Startseite

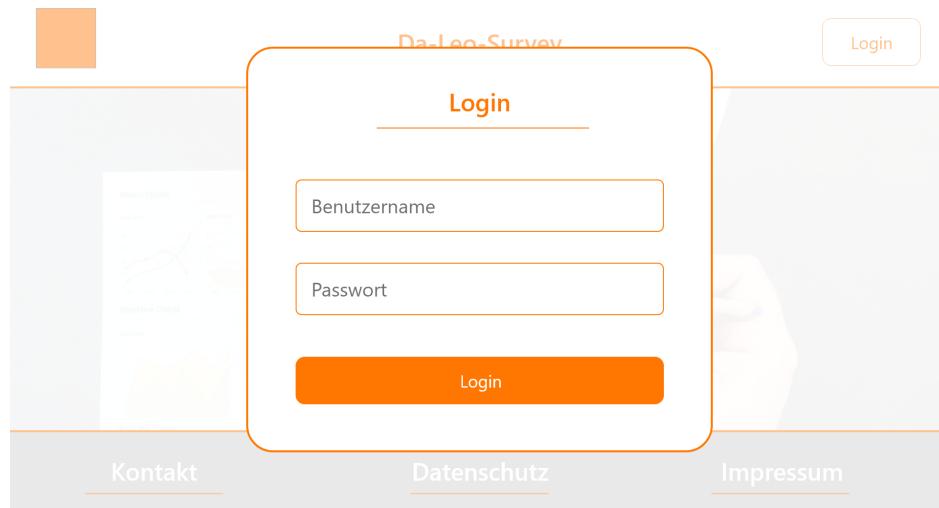


Abbildung 54.: ScreenDesign Login



Abbildung 55.: ScreenDesign Dashboard



Abbildung 56.: ScreenDesign Fragebogen erstellen



Abbildung 57.: ScreenDesign Fragebogen erstellen

Wie viele Fragen sollen erstellt werden?

Frage 1

Frage text
Frage type

Frage 2

Frage text

Abbildung 58.: ScreenDesign Fragebogen erstellen

Frage 1
Was ist zu verbessern?
Text

Frage 2
Wie gut war das Essen?
SingleChoice
Sehr gut value: 2

Abbildung 59.: ScreenDesign Fragebogen erstellen

Da-Leo-Survey

Login

1 Name eingeben —— 2 Fragen eingeben —— 3 Überprüfen —— 4 Transcode

FeedBackBogen

FeedBackbogen für Schüler

Frage 1
Was ist zu verbessern?
Text

Abbildung 60.: ScreenDesign Fragebogen erstellen

Da-Leo-Survey

Login

1 Name eingeben —— 2 Fragen eingeben —— 3 Überprüfen —— 4 Transcode

Ihr Transaktionscode für diese Umfrage:
123GUHI878575

Fertig

Abbildung 61.: ScreenDesign Fragebogen erstellen

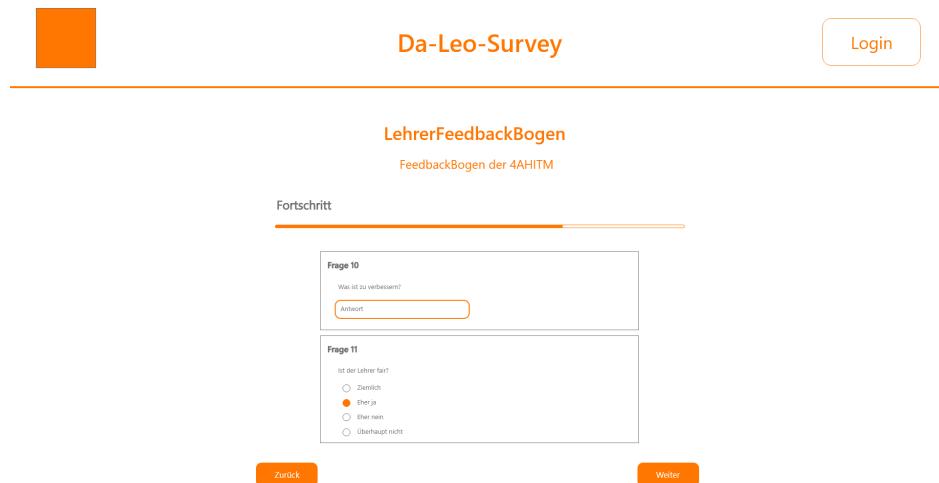


Abbildung 62.: ScreenDesign Fragebogen ausfüllen

Die Login-Seite (siehe Abb. 54) wurde im Produktivsystem mit der des Keycloak ersetzt. Auch andere kleinere Elemente wie der Footer auf der Startseite (siehe Abb. 53) wurden in der Engversion abgeändert. Die Endversion befindet sich in Kapitel 14 auf S. 103.

14. Ergebnis

14.1. Zusammenfassung

Ergebnis der Arbeit ist eine lauffähige Applikation bestehend aus Frontend, Backend, Keycloak und Datenbank. Alle Komponenten wurden gelockert und auf eine VM deployed. Für die Sicherheit sorgt eine Sperre der Ports für die VM und der Reverse-Proxy Traefik.

14.2. Frontend

Es folgen Abbildung der wichtigsten Unterseiten und Funktionalitäten der Arbeit. [116], [117], [118]

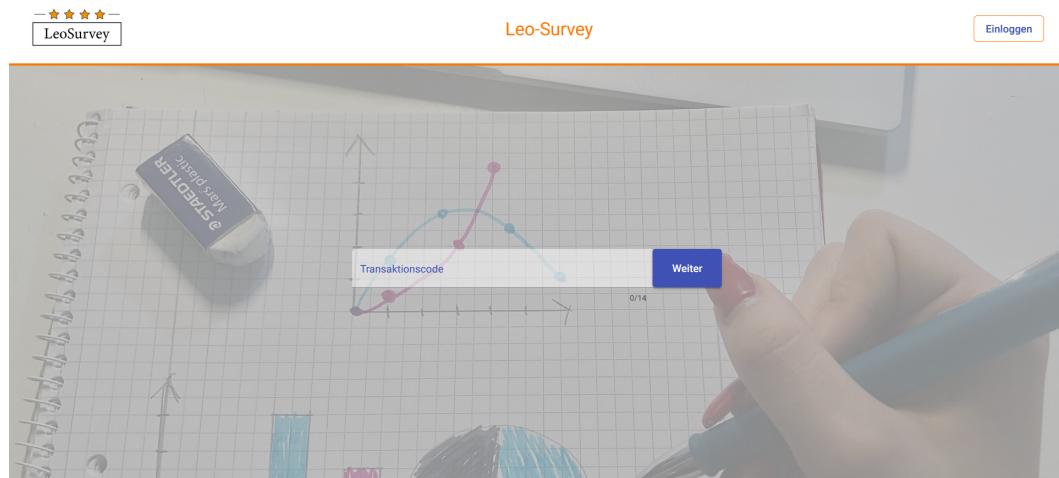


Abbildung 63.: Startseite

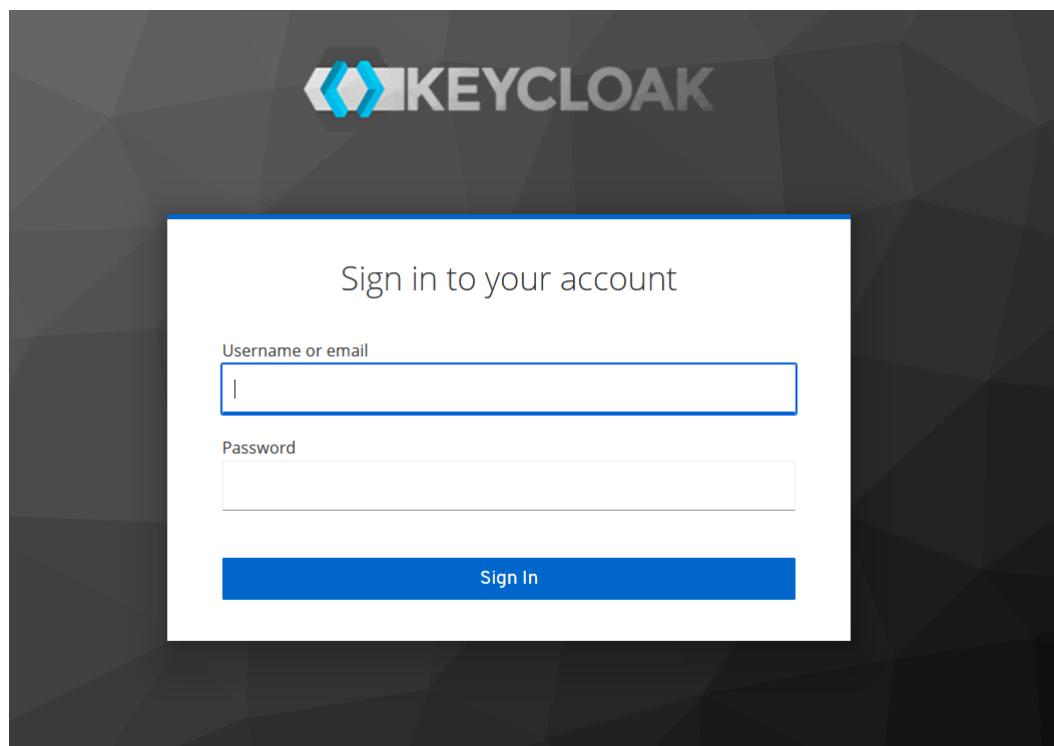


Abbildung 64.: Login mittels Keycloak

Mithilfe des Keycloak können sich die Schüler/innen und Lehrer/innen mit dem gewohnten Anmelddaten anmelden.

A screenshot of a dashboard interface. At the top left is a red 'Zurück zur Startseite' button. At the top right is a blue 'Fragebogen erstellen +' button. Below these are two sections of cards. The first section is titled 'Filter' and contains a dropdown menu. The second section displays two cards: 'Lehrer-Feedbackbogen Feedbackbogen für Lehrer' and 'Lehrerinnen-Feedbackbogen Feedbackbogen für Lehrerinnen'. Each card has a set of orange icons for edit, delete, and other actions. The overall layout is clean with a white background and light grey borders for the cards.

Abbildung 65.: Dashboard

Auf dem Dashboard werden alle für den / die Benutzer/in verfügbaren Fragebögen und Umfragen angezeigt. Alle Fragebögen und Umfragen, die im Besitz einer Person sind, können geändert und gelöscht werden. Von anderen Fragebögen können Kopien erstellt werden. Fragebögen können gefiltert werden. Umfragen können erstellt und ausgewertet werden.

Abbildung 66.: Fragebogen erstellen

Frage 1

Frage eingeben *

Wie hat euch die Presentation gefallen?

39/250

Durchsuchen... Keine Dateien ausgewählt.

Fragetyp *

YESORNO

Yes	1	<input type="checkbox"/> Richtig	X
No	2	<input type="checkbox"/> Richtig	X

Frage 2

Frage eingeben *

Irgendwelche Anmerkungen?

25/250

Durchsuchen... Keine Dateien ausgewählt.

Fragetyp *

FREETEXT

Abbildung 67.: Startseite

Fragebögen können stufenweise erstellt werden. Es kann zwischen vier verschiedenen Fragetypen gewählt werden.

Lehrerinnen-Feedbackbogen

Feedbackbogen für Lehrerinnen

46 Fragen

Abschicken

Items per page: 3 ▾ 1 – 3 of 46 |< < > >|

Frage 1

Die Lehrerin ist fair - gerecht

trifft völlig zu
 trifft eher zu
 trifft eher nicht zu
 trifft völlig eher nicht zu

Frage 2

Die Lehrerin ist humorvoll - fröhlich

trifft völlig zu
 trifft eher zu
 trifft eher nicht zu
 trifft völlig eher nicht zu

Abbildung 68.: Startseite

Das Beantworten der Umfrage erfordert keinen Login und wird mithilfe von Transaktionscodes durchgeführt. Fragen werden in kleinen Gruppen zusammengefasst, angezeigt und können optional beantwortet werden. Vor dem Abschicken wird dem/er Benutzer/in angezeigt, wie viele Fragen er / sie nicht beantwortet hat.

15. Resümee

15.0.1. Meilesteine

Für eine Evaluation der Meilesteine siehe Kapitel Meilensten Analyse D.

15.0.2. Kommunikation

Kommunikation ist ein wichtiger Punkt bei Projektarbeiten. Deshalb wurde auch bei dieser Arbeit großer Wert auf eine bestmögliche Kommunikation gelegt.

Die Kommunikation im Team ist von beiden Parteien als angenehm empfunden worden. Man konnte sich immer auf seinen Teampartner verlassen und konnte damit rechnen, dass Aufgaben und Fehler schnellstmöglich erledigt, beziehungsweise behoben werden. Bei auftretenden Problemen wurde Kritik immer objektiv formuliert und angenommen. Bei der Vereinbarung von Meetings konnte sich das Team immer rasch auf ein Datum und Uhrzeit einigen. Auch mit dem Betreuer konnte sich schnell auf einen Zeitpunkt geeinigt werden. Die Projektkommunikation im Entwicklerteam intern hat primär auf YouTrack stattgefunden, aber auch persönliche Gespräche waren ein wichtiger Bestandteil. Die wichtigsten Informationen für den Betreuer wurden per E-Mail übermittelt, aber auch hier wurde großer Wert auf das persönliche Gespräch gelegt.

15.0.3. Probleme während der Implementierung

Wir hatten große Schwierigkeiten bei der Einhaltung des Zeitplans, da wir den Umfang des Projektes falsch eingeschätzt haben. Außerdem hatten wir Probleme bei den Technologien, bei deren Umsetzung manche Aspekte davon neu erlernt werden mussten.

Anhang

A. Teamvorstellung

A.0.1. Team

Raffeiner Christine

Christine Raffeiner ist eine 20 Jahre junge Musterschülerin, die viel Zeit und Mühe investiert um ihre schulischen Ziele zu erreichen. Sie ist bereits das sechste Jahr an der HTL Leonding, jedoch ist sie in allen Fächern eine der Besten aus ihrer Klasse und ist auch gerne bereit anderen zu helfen und ihnen Fragen zu erklären. Zusätzlich zu ihrer Hilfsbereitschaft strahlt sie Fröhlichkeit aus und ist eine Person mit der man gern über alle möglichen Themen spricht, egal ob schulischer oder privater Natur.



Weissengruber Nina

Nina Weissengruber die perfekte Ergänzung für jedes Team und ein richtiges Energiepacket, die die Aufmerksamkeit auf sich zieht. Nina ist eine hervorragende Schülerin, die man sich zum Vorbild nehmen kann und muss, denn sie schafft es ohne große Hindernisse, zu bestehen. Besonders erwähnenswert sind auch ihre ausgezeichneten Kenntnisse in Mathematik und Apex.



Über Uns

Zusammenfassend ist noch zu sagen, dass wir ausgezeichnet zusammenarbeiten konnten, obwohl wir uns vor unserem Projekt nicht wirklich gut kannten. Unsere Arbeitsweise betrachtend spielten wir gut zusammen und verfolgten ohne viel Absprache die selben Ziele.

B. Wichtige Lektionen

Die offensichtlichste Lektion die wir gelernt haben, war die Handhabung von längerfristigen Projekten, die hohe Priorität und gewisse Eigenverantwortung mit sich ziehen. Um das noch weiter zu führen wurden auch die Erfahrungen gemacht, die Projektarbeit völlig außerhalb der Schulstunden neben Schul- und Lernstress mit sich bringen. Dazu zählen besonders das Einteilen von Aufgaben und Zeitmanagement. Persönliche Erfahrungen wurden besonders im Bereich der Testimplementierung und Wichtigkeit dieser erzielt. Auch wurde außerhalb der bereits erlernten Kenntnisse wissen im Bereich des Keycloak und Reverse-Proxy gesammelt. Rückschließend sind wir zu dem Entschluss gekommen, dass die stressfreie Zeit während der Sommerferien besser hätten nutzen sollen.

C. Protokolle

Folgend befinden sich sämtliche Gesprächsprotokolle der Meetings, die das Team im Laufe der Arbeit mit ihrem Betreuer Herrn Professor Thomas Stütz geführt hat. Die Protokolle stellen eine Zusammenfassung der jeweiligen Meetings dar. [119]

Besprechungsprotokoll 29.04.2021

Table 1. Teilnehmer

anwesend	Verteiler
Nina Weissengruber	
Christine Raffeiner	

Table 2. Ort und Zeit

Ort	Online (Discord)
von-bis	19:05 - 20:10
Dauer	1h 5min

Besprochene Themen

- Wir haben allgemein über die Aufgabenstellung und den Umfang unserer Diplomarbeit gesprochen, da Raffeiner Christine neu zu dem Projekt hinzugekommen ist.
- Danach haben wir in der Diplomarbeitsdatenbank für unsere Themenstellung Meilensteine und die individuellen Themenstellungen festgelegt.
- Anschließend haben wir ein Github-Repository für unsere Diplomarbeit angelegt.

Nächste Schritte

- Die Erklärung für die Diplomarbeit unterschreiben und auf die Datenbank laden.
- Den Antrag für die Diplomarbeit abgeben.

Besprechungsprotokoll 01.07.2021

Table 1. Teilnehmer

anwesend	Verteiler
Nina Weissengruber	
Christine Raffeiner	

Table 2. Ort und Zeit

Ort	In der Schule
von-bis	11:45 - 12:40
Dauer	55 min

Besprochene Themen

- Wir haben unseren Diplomarbeitsantrag umgeändert und dabei eine Zielsetzung hinzugefügt und die individuellen Themenstellungen angepasst.
- Danach haben wir den Funktionsumfang der Diplomarbeit besprochen.
 - Gestalten von Fragebögen
 - Durchführen von Fragebögen
 - Auswerten von Befragungen (exportieren als csv Datei + Standartauswertung)
 - Anmelden mittels WebUnits
 - Mehrere Fragebögen stehen für alle Benutzer (Fragebogenersteller) zur Verfügung
 - Befragungen können anonym (mit TAN und auch mit Authentisierung / nur einen Fragebogen ausfüllen) oder mit Namen (Authentisierung) oder keine Anmeldung durchgeführt werden.
- Weiters haben wir über den Ablauf der Diplomarbeit gesprochen.
 1. Datenmodel definieren
 2. Datenmodell als Java Klassen implementieren
 3. Use-Cases definieren
 4. Die einzelnen Use-Cases ohne GUI implementieren
 5. Endpoints erstellen, welche die Repository-Funktionen verwenden und die die Angular App später verwenden kann
 6. Implementieren der Authentifizierung und Authorisierung

Nächste Schritte

- Datenmodell erstellen und bei Professor Stütz melden.

Besprechungsprotokoll 15.07.2021

Table 1. Teilnehmer

anwesend	Verteiler
Nina Weissengruber	
Christine Raffeiner	
Isabel Turner	

Table 2. Ort und Zeit

Ort	Online (Discord)
von-bis	18:00 - 19:00
Dauer	1h 0min

Besprochene Themen

- Wir haben das Use-Case-Diagramm vom repo leo-survey besprochen und einige Änderungen daran vorgenommen (Use Cases hinzugefügt), um diese in unserem Use-Case-Diagramm so zu übernehmen.
- Weiters haben wir auch das Datenmodell des repos leo-survey durchgesprochen und beschlossen dieses mit leichten Änderungen (Attribute hinzugefügt und Entities umbenannt) als, unser Datenmodell zu übernehmen.
- Anschließend haben wir uns für unserer Datenbank auf eine PostgreSQL entschieden.
- Danach haben wir beschlossen erstmals noch kein ERD zu erstellen, sondern es uns später generieren zu lassen.

Nächste Schritte

- Bis morgen ein Klassen-Diagramm erstellen.

Besprechungsprotokoll 16.07.2021

Table 1. Teilnehmer

anwesend	Verteiler
Nina Weissengruber	
Christine Raffeiner	
Isabel Turner	

Table 2. Ort und Zeit

Ort	Online (Discord)
von-bis	14:00 - 15:55
Dauer	1h 55min

Besprochene Themen

- Zuerst haben wir das von uns erstellte Klassendiagramm besprochen und daran kleine Änderungen vorgenommen. (Schreibfehler korrigiert.)
- Danach versuchten wir mittels des Tools PlantUML-Generator die erstellten Klassen des CLDs in Code umzuwandeln, um uns Arbeit zu ersparen, doch wegen eines Parse-Errors funktionierte der Vorgang nicht und wir entschieden uns die Klassen doch händisch zu programmieren.
- Weiters haben wir unser Projekt erstellt und die Dependencies für unser Projekt angepasst.
- Währenddessen haben wir über die Unterschiede bzw. Vor- und Nachteile von jUnit, AssertJDB und AssertJ gesprochen, da wir diese Libraries für das Testen unseres Servers verwenden.
- Später haben wir die Klasse Questionnaire einschließlich aller Attribute, einer toString-Methode, einem Repo und einer Test-Klasse, eingeschlossen Tests für das Speichern in der Datenbank, angelegt.
- Während der Erstellung der Tests haben wir auch eine PostgreSQL Datenbank, die innerhalb eines Docker-Containers läuft, erstellt, konfiguriert und getestet.
- Da die Datenbank in einem Docker-Container läuft, haben wir kurz besprochen wie man Docker für Windows installiert.
- Als letztes haben wir noch über den neuen Ablauf des Projektes gesprochen.
 1. Model und Tests für die Entities erstellen
 2. Repositories erstellen
 3. Rest-Anbindungen umsetzen
 4. Angular Client erstellen
 5. Authentifizierung umsetzen
 6. Dockern

Nächste Schritte

- Die restlichen Entities, die dazugehörigen Repos (mithilfe von Panach) und die Test-Klassen erstellen.
- Mithilfe der Tests 3 Fragen des Lehrer-Feedbackbogens in die Datenbank speichern.

Besprechungsprotokoll 05.08.2021

Table 1. Teilnehmer

anwesend	Verteiler
Nina Weissengruber	
Christine Raffeiner	
Isabel Turner (tw.)	

Table 2. Ort und Zeit

Ort	Online (Discord)
von-bis	09:45 - 11:35
Dauer	1h 50min

Besprochene Themen

- Wir haben ein YouTrack-Project erstellt und dafür eine VCS-Anbindung angelegt.
- Danach haben wir einige Issues und einen Sprint für das Projekt angelegt und besprochen wie YouTrack allgemein funktioniert.
- Zudem haben wir auch über die Verwendung von User-Stories und Sprint-Backlogs gesprochen.
- Weiteres haben wir die richtige Weise einen Commit zu gestalten besprochen.
- Später haben wir von Herrn Professor Stütz eine neue gedockerte PostgreSQL-Datenbank erhalten, die im Gegensatz zu der alten eine Web-Oberfläche besitzt.
- Als letztes haben wir noch einige Referenzen (@JoinColumn) bei unseren Klassen ausgebessert.

Nächste Schritte

- Mit Unit-Test einen Fragebogen mit allen Fragetypen und Antworten zu den Fragen erstellen. Anschließend die Ergebnisse mit Unit-Tests testen.
- Restliche Referenzen für die Tabellen ausbessern.

Besprechungsprotokoll 09.08.2021

Table 1. Teilnehmer

anwesend	Verteiler
Nina Weissengruber	
Christine Raffeiner	

Table 2. Ort und Zeit

Ort	Online (Discord)
von-bis	20:00 - 21:50
Dauer	1h 50min

Besprochene Themen

- Zuerst haben wir Imports bei unseren Tests ausgetauscht.
- Danach haben wir unsere Probleme mit AssertJ-db-Tests angesprochen und anschließend einen solchen Test gemeinsam programmiert.
- Daraufhin haben wir uns darauf geeinigt den vorhandenen Test (Test für Repositories) nicht mehr so genau zu testen und uns auf die Testung der Rest-Schnittstellen zu konzentrieren.
- Wir haben beschlossen die Rest-Schnittstellen mit Karate-junit5 zu testen und haben diese Library daraufhin implementiert.
- Danach haben wir eine Erklärung zu Karate erhalten und angefangen einen einfachen Test zu implementieren.

Nächste Schritte

- Rest-Schnittstellen und Tests (HTTP-File) für das Speichern von einem Interviewer und einem Questionnaire erstellen. Danach bei Herr Professor Stütz melden (evtl. Änderungen vornehmen) und danach die restlichen Schnittstellen für die restlichen Klassen implementieren.
- YouTrack Tasks für jede Klasse erstellen.

Besprechungsprotokoll 10.08.2021

Table 1. Teilnehmer

anwesend	Verteiler
Christine Raffeiner	

Table 2. Ort und Zeit

Ort	Online (Discord)
von-bis	18:45 - 19:15
Dauer	0h 30min

Besprochene Themen

- Wir haben uns die 2 REST-Schnittstellen für das Erstellen eines Interviewers und eines Questionnaires angesehen.
- Bei beiden Methoden wurde die Annotation @Path entfernt und weiters auch bei den Test-Pfaden dieser Teil des Pfades herausgenommen.
- Danach haben wir noch über die Benennung von Pfaden im Web gesprochen. (z.B. nicht GetByID sondern get-by-Id).

Nächste Schritte

- Die restlichen REST-Schnittstellen überlegen und diese für die restlichen Klassen implementieren. Dazu HTTP-Test erstellen und ausprobieren.

Besprechungsprotokoll 24.08.2021

Table 1. Teilnehmer

anwesend	Verteiler
Christine Raffeiner	
Nina Weissengruber	
Isabel Turner (tw.)	

Table 2. Ort und Zeit

Ort	Online (Discord)
von-bis	20:00 - 22:25
Dauer	2h 25min

Besprochene Themen

- Zuerst haben wir ein YouTrack Problem "repariert", dass uns es nicht erlaubte, die von einer anderen Person erstellten Tasks zu bearbeiten oder zu verschieben.
- Danach haben wir einen Karate-Test für die REST-Schnittstelle des Interviewers angelegt. Dabei sind wir auf ein Problem mit der findAll-Methode gestoßen.
- Auch haben wir über das Erstellte ScreenDesign gesprochen, welches Grundsätzlich in Ordnung ist, doch müssen noch Veränderungen getätigt werden (Dashboard).

Nächste Schritte

- Bei der Klasse Questionnaire soll der Interviewer eingetragen werden.
- Für die Klassen Questionnaire, Interviewer, Question und AnswerOption sollen bei jedem Start Daten importiert werden (import.sql-File erstellen).
- Für die Klassen Questionnaire, Interviewer, Question und AnswerOption sind Karate-Tests für die REST-Schnittstellen zu implementieren. Sollten hier Fehler auftreten sollen Repository-Tests erstellt werden.

Besprechungsprotokoll 31.08.2021

Table 1. Teilnehmer

anwesend	Verteiler
Christine Raffeiner	
Nina Weissengruber	

Table 2. Ort und Zeit

Ort	Online (Discord)
von-bis	14:00 - 15:50
Dauer	1h 50min

Besprochene Themen

- Zuerst haben wir über unsere GitHub-Branches gesprochen, wobei uns Herr Professor Stütz darauf aufmerksam gemacht hat, dass es bei der jetzigen Lage eventuell zu Konflikten kommen könnte.
- Danach haben wir den Text, der beim Starten des Servers im Terminal angezeigt wird von Quarkus auf HTL Leonding geändert.
- Als Nächstes haben wir alle nicht mehr relevanten Tests mit `@Disabled` annotiert, sodass sie bei automatischem Start der Tests nicht mehr mit ausgeführt werden.
- Als letztes haben wir uns die erstellten Karate-Test für die Klassen Interviewer, Questionnaire, Question und AnswerOption angesehen. Grundsätzlich sind die Tests in Ordnung, doch die Tests sollten noch ausgebaut werden.

Nächste Schritte

- Bei den Methoden auch auf nicht mögliche Requests testen (z.B. Löschen eines nicht vorhandenen Interviewers) und daraufhin passenden Fehlermeldungen mit aussagekräftigen Fehlercodes zurückgeben.
- Für den jetzigen Teil einen Angular-Client ohne Authentifizierung bzw. Autorisierung erstellen und schrittweise mit dem Backend erweitern.
- In kleinen Schritten die weiteren Karate-Tests für die Klassen erstellen und testen.
- Frau Isabell Turner muss Tests für die Repositories der Klassen InterviewerRepository, QuestionnaireRepository und QuestionRepository erstellen.

Besprechungsprotokoll 07.10.2021

Table 1. Teilnehmer

anwesend	Verteiler
Christine Raffeiner	
Nina Weissengruber	

Table 2. Ort und Zeit

Ort	Schule
von-bis	15:30 - 16:20
Dauer	0h 50min

Besprochene Themen

- Zuerst haben wir über das Frontend gesprochen. Hier wurde angemerkt, dass bei den AnswerOptions der MultipleChoice Questions auch einen Value eingegeben werden kann. Dazu sollen die Values der AnswerOptions automatisch einen default-Value vergeben bekommen.
- Auch wurde besprochen, dass die Login-Page nicht nötig ist, da die Autorisierung/Authentifizierung über Keycloak stattfindet.
- Als Nächstes haben wir über das Backend gesprochen. Die Karate-Tests sollen nach User-Stories kategorisiert werden.

Nächste Schritte

- Die angesprochenen Anmerkungen beheben.
- Die Nächsten Sprints mit UserStories und Task vorplanen um zu sehen was noch zu tun ist und wie lange wir denken, dass wir dafür brauchen.
- Transaktionscodes in einer angegebenen Menge generieren und alle auf einer Druckbaren Seite darstellen lassen.
- Die erstellten Fragebögen auf dem Dashboard farblich unterscheidbar darstellen, je nachdem ob der Fragebogen public (für alle verfügbar ist) oder private (nur für den ersteller sichtbar).
- Github Pages erstellen. Allgemeines über unser Project, Meilensteine und Getting-Started einfügen.

Besprechungsprotokoll 22.10.2022

Table 1. Teilnehmer

anwesend	Verteiler
Christine Raffeiner	
Nina Weissengruber	
Isabel Turner	

Table 2. Ort und Zeit

Ort	Schule
von-bis	09:30 - 09:45 / 10:00 - 11:00
Dauer	1h 15min

Besprochene Themen

- Bei der Evaluierung unseres jetzigen Standes der Diplomarbeit sind wir auf einige Ablaufprobleme beim Anlegen von Transactioncodes und Surveys im Dashboard gekommen. Aufgrunddessen soll das Dashboard nochmals überarbeitet werden. Dafür wird eine hierachische Tabelle für die Fragebögen und Befragungen, die man filtern kann implementiert.
- Weiteres haben wir darüber gesprochen das für den Anwendungsfall eines Testes mit der Survey ein zusätzliches Attribut bei den AnswerOptions (isTrue) einzufügen.
- Zudem haben wir auch über die Notwendigkeit einer Löschbestätigung für Fragebögen und Befragungen gesprochen.
- Außerdem ist beschlossen worden, wir zurzeit keine Ldap-User implementieren müssen. (Login entfällt)
- Abschließend wurde Isabel die Auswertung der Befragung sowohl im Frontend als auch im Backend zugeteilt. Diese soll mithilfe von einer Library zusätzlich visualisiert werden. (z.B. Chart.js)

Nächste Schritte

- Die angesprochenen Probleme beheben.
- CLD und import.sql aktualisieren.

Besprechungsprotokoll 22.10.2022

Table 1. Teilnehmer

anwesend	Verteiler
Christine Raffeiner	
Nina Weissengruber	
Isabel Turner	

Table 2. Ort und Zeit

Ort	Schule
von-bis	13:40 - 15:50
Dauer	2h 10min

Besprochene Themen

- Beim Diplomarbeitsgespräch haben wir uns zuerst das Beantworten eines Fragebogen im Frontend angesehen und sind dort auf einige Probleme bzw. Verbesserungen gestoßen (z.B. mehr Rückmeldung an den User wie viele Fragen er schon beantwortet hat, Layout/Css-Änderungen und Bug-Fixes (man sieht nicht welche Fragen man schon beantwortet hat.))
- Zudem sollte die Diplomarbeit auf eine VM, die von der Schule zur Verfügung gestellt wird laufen und an Herrn Prof. Stütz der Link zu der Webseite geschickt werden.
- Auch sind auf den anderen Seiten kleine Änderungen nötig (Button größer etc.) und vor allem sollen Rechtschreibfehler ausgebessert werden.
- Auch soll der Login nun mittels LDAP implementiert werden und bei einer Frage soll optional ein Bild hinzugefügt werden können.
- Weiters sind wir beim Testen der Beantwortung der Fragen auf die Idee gekommen Emojis mit Zeichen auszutauschen. Das kann mit einer Library oder eines Algorithmus gelöst werden.
- Auch sind wir beim Durchbesprechen auf das Problem gestoßen, dass beim Erstellen des Files mit den Transactioncodes mehrere Files nicht voneinander unterschieden werden können. Dieses Problem soll mittels eines aussuchbaren Filenames oder eines Textes im File gelöst werden.
- Beim Frontend sind die Texte gemischt auf Englisch und Deutsch um das zu vermeiden soll mittels angular i18n gearbeitet werden.
- Als letztes haben wir noch über die eventuelle Verwendung von Gatling gesprochen, mit dessen Hilfe automatisch Antworten für die Tests angelegt werden sollen. Das kann auch mittels HTTP-Befehlen gelöst werden.

Nächste Schritte

- Alle besprochenen Änderungen und Bugs implementieren bzw. ausbessern.

- Login und die oben genannten Features implementieren.
- Die Diplomarbeit auf einer VM laufen lassen.

D. Meilensteinliste und -analyse

Die erste Version der Meilensteinliste sah vor, dass die Arbeit innerhalb des Sommers, während der Sommerferien implementiert und bis zu Schulbeginn fertiggestellt wird.

Tabelle 1.: Meilensteinliste Version 1

05.07.2021	Datenmodell fertiggestellt
19.07.2021	Backend (mit Endpoints) zum Ausfüllen eines vorhandenen Fragebogens erstellt
02.08.2021	Frontend zum Ausfüllen eines vorhandenen Fragebogens erstellt
09.08.2021	Backend (mit Endpoints) zum Erstellen eines Fragebogens erstellt
23.08.2021	Frontend zum Ausfüllen von Fragebögen erstellt
13.09.2021	Gesamtsystem mit Benutzerverwaltung und Docker-Deployment fertiggestellt

Im Laufe des Sommers stellte sich bald heraus, dass diese Meilensteine aus vielerlei Gründen nicht eingehalten werden konnten. Die Meilensteine, (siehe Tabelle 1) (1) Datenmodell fertiggestellt und (2) Backend (mit Endpoints) die zum Ausfüllen eines vorhandenen Fragebogens erstellt, wurden in den Ferien verspätet fertiggestellt. Daraus folgend hat das Team die Meilensteine angepasst und eine neue Liste erstellt (siehe Tabelle 2).

Tabelle 2.: Meilensteinliste Version 2

05.11.2021	Backend und Frontend Funktionen zum Fragebogen erstellen fertiggestellt
10.11.2021	Frontend Dashboard fertiggestellt
19.11.2021	Backend und Frontend Funktionen zum Fragebogen ausfüllen fertiggestellt
29.11.2021	Backend zum Fragebogen ausfüllen fertiggestellt
20.12.2021	Backend und Frontend Login mittels Keycloak fertiggestellt
10.01.2022	Gesamtsystem Docker-Deployment fertiggestellt

Die neue Meilensteinliste verschob die noch nicht erledigten Meilensteine nach hinten - mit dem Datum der Erreichung des letzten Meilensteines in den Winterferien. Das

Team konnte auch diese Meilensteine nicht eingehalten. Da nun die Fertigstellung der Implementierung im Vordergrund stand wurden keine neuen Meilensteine mehr definiert und es wurde ohne Meilensteinliste auf die Fertigstellung hingearbeitet.

Schlussfolgernd ist zu sagen, dass die ursprünglichen Meilensteine oder zumindest die 2. Version der Liste, eingehalten hätten werden sollen. Das Team ließ sich viel zu viel Zeit und war auch wenig konsequent mit der Einhaltung der gesetzten Ziele. Zusätzlich wurde der gesamte Arbeitsumfang der Arbeit stark unterschätzt.

E. Arbeitsverteilung

Die Arbeitsaufteilung war von Anfang an schnell und ohne Probleme entschieden. Die Programmierung des Backend und Testung fiel dabei Nina zu und Implementierung und Erstellung des Designs des Frontend wurde Christine überlassen. Außerdem wurde entschieden, dass Nina das Deployment übernimmt und Christine den Keycloak implementiert.

E.0.1. Schriftliche Arbeitsteilung

1. Danksagung, Zusammenfassung und Inhaltsverzeichnis [R]
2. Einleitung [R & W]
3. Ausgangssituation [W]
4. Umfeldanalyse [R]
 - a. Vorgänger Projekte [R]
 - b. Online Fragebogenwerkzeuge [R]
5. Problembeschreibung [W]
6. Aufgabenstellung [W]
7. Ziele [R & W]
8. Systemarchitektur [R & W]
9. a. Aktivitätsdiagramm [W]
 - b. Klassendiagramm und Entity-Relationship-Modell [R]
10. Technologien [R & W]
 - a. Angular [R]
 - b. Docker [R]
 - c. Keycloak [R]
 - d. Traefik [R]
 - e. Oracle SQL-Developer [R]
 - f. PlantUML [R]
 - g. Markdown [R]

- h. Java [W]
 - i. Quarkus [W]
 - j. PostgreSQL [W]
 - k. Rest-API [W]
 - l. Software Testing [W]
11. Ausgewählte Aspekte [R & W]
- a. Besonders gut gelöste Programmteile [R & W]
 - b. Besondere Probleme die gelöst wurden [R & W]
 - c. Entwurfsentscheidung [R & W]
12. Implementierung [W]
13. Projektorganisation [R]
- a. Anforderungen [R]
 - b. Mögliche Hilfsmittel [R]
 - c. Problemlösung [R]
14. ScreenDesign [R]
15. a. Handskizzen [R]
- b. ScreenDesign Prototyp [R]
16. Ergebnis [R]
17. a. Zusammenfassung [R]
- b. Frontend [R]
18. Resümee [R & W]
19. Anhang [R & W]
- a. Teamvorstellung [R & W]
 - b. Wichte Lektionen [R & W]
 - c. Protokolle [R]
 - d. Meilensteinliste und -analyse [R]
 - e. Arbeitsteilung [R & W]

Literaturverzeichnis

- [1] „Fragebogen Vorlage 1 questionz.” Online verfügbar: <https://github.com/htl-leonding-project/questionz>
- [2] „Fragebogen Vorlage 2 LeoSurvey.” Online verfügbar: <https://github.com/htl-leonding-project/leo-survey>
- [3] „Fragebogen Referenzobjekt 2ask.com.” Online verfügbar: <https://www.2ask.com/de/>
- [4] „Fragebogen Referenzobjekt appinio.com.” Online verfügbar: <https://www.appinio.com/de/>
- [5] „Fragebogen Referenzobjekt ingress.com.” Online verfügbar: <https://www.ingress.de>
- [6] „Fragebogen Referenzobjekt questionpro.com.” Online verfügbar: <https://www.questionpro.com/>
- [7] „Fragebogen Referenzobjekt survio.com.” Online verfügbar: <https://www.survio.com/>
- [8] „Fragebogen Referenzobjekt umfrageonline.com.” Online verfügbar: <https://www.umfrageonline.com>
- [9] „pdflatex, \includegraphics, and multi-page PDF files,” Aug. Online verfügbar: <https://tex.stackexchange.com/questions/7938/pdflatex-includegraphics-and-multi-page-pdf-files>
- [10] „Was ist Entity-Relationship-Diagramm (ERD)? - Definition von WhatIs.com.” Online verfügbar: <https://whatis.techtarget.com/de/definition/Entity-Relationship-Diagramm-ERD>
- [11] „Klassendiagramm,” Feb. 2022, Page Version ID: 220087015. Online verfügbar: <https://de.wikipedia.org/w/index.php?title=Klassendiagramm&oldid=220087015>
- [12] „Klassendiagramm Syntax und Funktionen.” Online verfügbar: <https://plantuml.com/de/class-diagram>
- [13] „Normalisierung (Datenbank),” Mar. 2022, Page Version ID: 220974383. Online verfügbar: [https://de.wikipedia.org/w/index.php?title=Normalisierung_\(Datenbank\)&oldid=220974383](https://de.wikipedia.org/w/index.php?title=Normalisierung_(Datenbank)&oldid=220974383)
- [14] „Erste Normalform (1NF) | Normalisierung von Datenbanken.” Online verfügbar: <https://www.datenbanken-verstehen.de/datenmodellierung/normalisierung/erste-normalform/>
- [15] „Angular.” Online verfügbar: <https://angular.io/>

- [16] „Angular,” Dez. 2021, Page Version ID: 217853124. Online verfügbar: <https://de.wikipedia.org/w/index.php?title=Angular&oldid=217853124>
- [17] „TypeScript,” Feb. 2022, Page Version ID: 220627084. Online verfügbar: <https://de.wikipedia.org/w/index.php?title=TypeScript&oldid=220627084>
- [18] „Softwareentwicklung: Was JavaScript von TypeScript unterscheidet.” Online verfügbar: <https://www.computerwoche.de/a/was-javascript-von-typescript-unterscheidet.3548614>
- [19] „JavaScript | MDN.” Online verfügbar: <https://developer.mozilla.org/de/docs/Web/JavaScript>
- [20] „JavaScript,” Feb. 2022, Page Version ID: 220210768. Online verfügbar: <https://de.wikipedia.org/w/index.php?title=JavaScript&oldid=220210768>
- [21] „JavaScript-Grundlagen - Lerne Webentwicklung | MDN.” Online verfügbar: https://developer.mozilla.org/de/docs/Learn/Getting_started_with_the_web/JavaScript_basics
- [22] „CSS - Ausführliche Erklärung aus dem Hosting-Lexikon.” Online verfügbar: <https://www.checkdomain.de/hosting/lexikon/css/>
- [23] „Cascading Style Sheets,” Feb. 2022, Page Version ID: 220167254. Online verfügbar: https://de.wikipedia.org/w/index.php?title=Cascading_Style_Sheets&oldid=220167254
- [24] „CSS – SELFHTML-Wiki.” Online verfügbar: <https://wiki.selfhtml.org/wiki/CSS>
- [25] „CSS Tutorial.” Online verfügbar: <https://www.w3schools.com/css/>
- [26] „World Wide Web Consortium (W3C).” Online verfügbar: <https://www.w3.org/>
- [27] „World Wide Web Consortium,” Feb. 2022, Page Version ID: 220241372. Online verfügbar: https://de.wikipedia.org/w/index.php?title=World_Wide_Web_Consortium&oldid=220241372
- [28] „Unterschied zwischen CSS, SCSS & SASS - Rechnerhaus GmbH.” Online verfügbar: <https://www.rechnerhaus.de/blog/unterschied-zwischen-css-scss-sass>
- [29] „SASS vorgestellt - SASS vs LESS vs SCSS - Das sind die Unterschiede - IONOS.” Online verfügbar: <https://www.ionos.at/digitalguide/websites/web-entwicklung/sass/>
- [30] „Sass (Stylesheet-Sprache) – Wikipedia.” Online verfügbar: [https://de.wikipedia.org/wiki/Sass_\(Stylesheet-Sprache\)](https://de.wikipedia.org/wiki/Sass_(Stylesheet-Sprache))
- [31] „Das große SASS-Tutorial: Das müssen Sie wissen!” Online verfügbar: <https://www.ionos.at/digitalguide/websites/web-entwicklung/sass-tutorial/>
- [32] „Sass: Syntactically Awesome Style Sheets.” Online verfügbar: <https://sass-lang.com/>
- [33] „Less (Stylesheet-Sprache) – Wikipedia.” Online verfügbar: [https://de.wikipedia.org/wiki/Less_\(Stylesheet-Sprache\)](https://de.wikipedia.org/wiki/Less_(Stylesheet-Sprache))
- [34] „Hypertext Markup Language,” Feb. 2022, Page Version ID: 220236919. Online verfügbar: https://de.wikipedia.org/w/index.php?title=Hypertext_Markup_Language&oldid=220236919

- [35] „HTML: HyperText Markup Language | MDN.” Online verfügbar: <https://developer.mozilla.org/de/docs/Web/HTML>
- [36] „HTML einfach und verständlich erklärt - SEO-Küche.” Online verfügbar: <https://www.seo-kueche.de/lexikon/html/>
- [37] Node.js, „Node.js.” Online verfügbar: <https://nodejs.org/en/>
- [38] „Node.js,” Mar. 2022, Page Version ID: 221490047. Online verfügbar: <https://de.wikipedia.org/w/index.php?title=Node.js&oldid=221490047>
- [39] „Node.js Introduction.” Online verfügbar: https://www.w3schools.com/nodejs/nodejs_intro.asp
- [40] „Angular Material UI component library.” Online verfügbar: <https://material.angular.io/>
- [41] „Official components for Angular,” Apr. 2022, original-date: 2016-01-04T18:50:02Z. Online verfügbar: <https://github.com/angular/components>
- [42] „Name der Material-Icons.” Online verfügbar: <https://fonts.google.com/icons>
- [43] „angular-oauth2-oidc - npm.” Online verfügbar: <https://www.npmjs.com/package/angular-oauth2-oidc>
- [44] „XML Localization Interchange File Format,” Jan. 2020, Page Version ID: 196375817. Online verfügbar: https://de.wikipedia.org/w/index.php?title=XML_Localization_Interchange_File_Format&oldid=196375817
- [45] „localization - Internationalization in JSF, when to use message-bundle and resource-bundle? - Stack Overflow.” Online verfügbar: <https://stackoverflow.com/questions/2668161/internationalization-in-jsf-when-to-use-message-bundle-and-resource-bundle>
- [46] „Angular - @angular/localize.” Online verfügbar: <https://angular.io/api/localize>
- [47] „Angular - Angular Internationalization.” Online verfügbar: <https://angular.io/guide/i18n-overview>
- [48] „@angular/localize.” Online verfügbar: <https://www.npmjs.com/package/@angular/localize>
- [49] „Angular - Localizing your app.” Online verfügbar: <https://docs.angular.lat/guide/i18n>
- [50] „XLIFF,” Jan. 2022, Page Version ID: 1068044098. Online verfügbar: <https://en.wikipedia.org/w/index.php?title=XLIFF&oldid=1068044098>
- [51] „XLIFF-Dateien (XML Localization Interchange File Format).” Online verfügbar: <https://docs.memoq.com/current/de/Places/xliff-files.html>
- [52] „An XML Language for Describing Java Resource Bundles | Jeffrey Morgan.” Online verfügbar: <https://jeffreymorgan.io/articles/java-resource-bundles/>
- [53] „Message bundles.” Online verfügbar: <https://documentation.opencms.org/opencms-documentation/localization/message-bundles/index.html>
- [54] „XML files and resource bundles.” Online verfügbar: <https://help.hcltechsw.com/commerce/7.0.0/com.ibm.commerce.customizetools.doc/concepts/ctfbundles.html>

- [55] „Home - Docker.” Online verfügbar: <https://www.docker.com/>
- [56] „Docker Installation On Windows 10 | How To Install Docker? | Docker Installation | Simplilearn.” Online verfügbar: <https://www.youtube.com/watch?v=5nX8U8Fz5S0>
- [57] „Install Docker Desktop on Windows.” Online verfügbar: <https://docs.docker.com/docker-for-windows/install/>
- [58] „Docker (Software),” Jan. 2022, Page Version ID: 219084716. Online verfügbar: [https://de.wikipedia.org/w/index.php?title=Docker_\(Software\)&oldid=219084716](https://de.wikipedia.org/w/index.php?title=Docker_(Software)&oldid=219084716)
- [59] „Was ist Docker?” Online verfügbar: <https://www.redhat.com/de/topics/containers/what-is-docker>
- [60] „Virtuelle Maschine,” Dez. 2021, Page Version ID: 217786126. Online verfügbar: https://de.wikipedia.org/w/index.php?title=Virtuelle_Maschine&oldid=217786126
- [61] „Was ist Virtuelle Maschine - Virtual Machine (VM)? - Definition von WhatIs.com.” Online verfügbar: <https://www.computerweekly.com/de/definition/Virtuelle-Maschine-Virtual-Machine-VM>
- [62] „Docker – die revolutionäre Container-Technologie.” Online verfügbar: <https://www.ionos.de/digitalguide/server/knowhow/was-ist-docker/>
- [63] „Kubernetes Tutorial: Von Docker zu Kubernetes,” July 2021. Online verfügbar: <https://info.syseleven.de/kubernetes-tutorial/>
- [64] „Was ist Kubernetes?” Section: docs. Online verfügbar: <https://kubernetes.io/de/docs/concepts/overview/what-is-kubernetes/>
- [65] „Docker Hub.” Online verfügbar: <https://hub.docker.com/>
- [66] „YAML,” Mar. 2022, Page Version ID: 221562025. Online verfügbar: <https://de.wikipedia.org/w/index.php?title=YAML&oldid=221562025>
- [67] „Einführung in YAML für Anfänger,” Jan. 2021. Online verfügbar: <https://geekflare.com/de/yaml-introduction/>
- [68] „Docker Desktop - Docker.” Online verfügbar: <https://www.docker.com/products/docker-desktop/>
- [69] „Install Docker Desktop on Windows | Docker Documentation.” Online verfügbar: <https://docs.docker.com/desktop/windows/install/>
- [70] „Keycloak.” Online verfügbar: <https://www.keycloak.org/>
- [71] „Quarkus Keycloak’s.” Online verfügbar: <https://www.youtube.com/watch?v=qNeDPgrytxE&t=1793s>
- [72] „keycloak/keycloak: Open Source Identity and Access Management For Modern Applications and Services.” Online verfügbar: <https://github.com/keycloak/keycloak>
- [73] „Keycloak,” Feb. 2022, Page Version ID: 1073770840. Online verfügbar: <https://en.wikipedia.org/w/index.php?title=Keycloak&oldid=1073770840>

- [74] „Mit Keycloak kommt unser Learning Friday Projekt auf ein neues Level,” Apr. 2020. Online verfügbar: <https://www.gepardec.com/keycloak/>
- [75] T. Bär, „Was ist LDAP (Lightweight Directory Access Protocol)?” Online verfügbar: <https://www.ip-insider.de/was-ist-ldap-lightweight-directory-access-protocol-a-581204/>
- [76] „Lightweight Directory Access Protocol – Wikipedia.” Online verfügbar: https://de.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol
- [77] „Active Directory und Domäne – einfach erklärt.” Online verfügbar: <https://www.dr-datenschutz.de/active-directory-und-domaene-einfach-erklaert/>
- [78] „Lightweight Directory Access Protocol,” Mar. 2022, Page Version ID: 221664556. Online verfügbar: https://de.wikipedia.org/w/index.php?title=Lightweight_Directory_Access_Protocol&oldid=221664556
- [79] „Active Directory – Wikipedia.” Online verfügbar: https://de.wikipedia.org/wiki/Active_Directory
- [80] „Was ist ein Active Directory?” Online verfügbar: <https://www.ip-insider.de/was-ist-ein-active-directory-a-626455/>
- [81] auth0.com, „JWT.IO - JSON Web Tokens Introduction.” Online verfügbar: <http://jwt.io/>
- [82] „JSON Web Token – Wikipedia.” Online verfügbar: https://de.wikipedia.org/wiki/JSON_Web_Token
- [83] „JSON Web Token (JWT): Funktionsweise + Beispiele - IONOS.” Online verfügbar: <https://www.ionos.at/digitalguide/websites/web-entwicklung/json-web-token-jwt-vorgestellt/>
- [84] „Keycloak — Open Source Admin-Handbuch der Linuxfabrik.” Online verfügbar: <https://docs.linuxfabrik.ch/software/keycloak.html>
- [85] „Server Administration Guide.” Online verfügbar: https://www.keycloak.org/docs/latest/server_admin/
- [86] „Traefik.” Online verfügbar: <https://doc.traefik.io/traefik/>
- [87] „traefik/traefik: The Cloud Native Application Proxy.” Online verfügbar: <https://github.com/traefik/traefik>
- [88] „PlantUML integration - IntelliJ IDEs Plugin | Marketplace.” Online verfügbar: <https://plugins.jetbrains.com/plugin/7017-plantuml-integration>
- [89] „PlantUML,” Jan. 2022, Page Version ID: 1067019984. Online verfügbar: <https://en.wikipedia.org/w/index.php?title=PlantUML&oldid=1067019984>
- [90] „Markdown,” Okt. 2021, Page Version ID: 216277920. Online verfügbar: <https://de.wikipedia.org/w/index.php?title=Markdown&oldid=216277920>
- [91] „Markdown: Anleitung für die einfache Auszeichnungssprache.” Online verfügbar: <https://www.ionos.at/digitalguide/websites/web-entwicklung/markdown/>
- [92] „Adobe XD | UI/UX-Design und Zusammenarbeit.” Online verfügbar: <https://www.adobe.com/at/products/xd.html>

- [93] „Was ist Adobe XD und was ist damit möglich?” Online verfügbar: <https://www.adobe.com/at/products/xd/learn/get-started/what-is-adobe-xd-used-for.html>
- [94] „Was ist SQL Developer | Oracle Deutschland.” Online verfügbar: <https://www.oracle.com/de/tools/technologies/whatis-sql-developer.html>
- [95] „Data Modeling Tools | Oracle SQL Developer Data Modeler.” Online verfügbar: <https://www.oracle.com/database/technologies/appdev/datamodeler.html>
- [96] „TypeError: cyclic object value - JavaScript | MDN.” Online verfügbar: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Errors/Cyclic_object_value
- [97] „Fixing the error “A possible object cycle was detected” in different versions of ASP.NET Core – Felipe Gavilán.” Online verfügbar: <https://gavilan.blog/2021/05/19/fixing-the-error-a-possible-object-cycle-was-detected-in-different-versions-of-asp-net-core/>
- [98] „Angular Alternatives | Learn The Top 8 Alternatives of Angular.” Online verfügbar: <https://www.educba.com/angular-alternatives/>
- [99] „Angular vs React 2022: Which JS Framework your Project Requires?” Online verfügbar: <https://www.simform.com/blog/angular-vs-react/>
- [100] „Vue vs React: What is The Best JavaScript Framework in 2022? | Codica.” Online verfügbar: <https://www.codica.com/blog/react-vs-vue/>
- [101] „GitHub,” Feb. 2022, Page Version ID: 219833775. Online verfügbar: <https://de.wikipedia.org/w/index.php?title=GitHub&oldid=219833775>
- [102] „GitHub Desktop.” Online verfügbar: <https://desktop.github.com/>
- [103] „GitHub Issues · Project planning for developers.” Online verfügbar: <https://github.com/features/issues>
- [104] „YouTrack: Project management for all your teams.” Online verfügbar: https://www.jetbrains.com/youtrack/?source=google&medium=cpc&campaign=10593512144&gclid=CjwKCAjwi6WSBhA-EiwA6NiokxWkS87okQ1xEorkscsy_GvHs0SI4PS6eNffKEPaPzlgZVLhtuUz6xoC17AQAvD_BwE
- [105] „Kanban.” Online verfügbar: <https://www.ionos.at/digitalguide/websites/webentwicklung/kanban/>
- [106] „Kanban (Softwareentwicklung),” Mar. 2022, Page Version ID: 221647684. Online verfügbar: [https://de.wikipedia.org/w/index.php?title=Kanban_\(Softwareentwicklung\)&oldid=221647684](https://de.wikipedia.org/w/index.php?title=Kanban_(Softwareentwicklung)&oldid=221647684)
- [107] „Was ist ein Kanban-Board – Grundlagen | Kanbanize.” Online verfügbar: <https://kanbanize.com/de/kanban-ressourcen/kanban-erste-schritte/was-ist-kanban-board>
- [108] „Scrum,” Apr. 2022, Page Version ID: 221753487. Online verfügbar: <https://de.wikipedia.org/w/index.php?title=Scrum&oldid=221753487>
- [109] „Scrum Grundlagen einfach erklärt: Der Product Backlog - InLoox.” Online verfügbar: <https://www.inlooxx.de/unternehmen/blog/artikel/scrum-grundlagen-einfach-erklaert-der-product-backlog/>

- [110] „Burn-Down-Chart,” Jan. 2021, Page Version ID: 207597995. Online verfügbar: <https://de.wikipedia.org/w/index.php?title=Burn-Down-Chart&oldid=207597995>
- [111] „Was ist ein Burn Down Chart? - Wissen online.” Online verfügbar: <https://www.microtool.de/wissen-online/was-ist-ein-burn-down-chart/>
- [112] „Token authentication requirements for Git operations,” Dez. 2020. Online verfügbar: <https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/>
- [113] „Creating a personal access token.” Online verfügbar: <http://ghdocs-prod.azurewebsites.net:80/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token>
- [114] „YouTrack Problem.” Online verfügbar: <https://youtrack-support.jetbrains.com/hc/en-us/community/posts/360009613400-Multiple-boards-one-issue>
- [115] „Agile Consulting.” Online verfügbar: <https://www.pwc.at/de/dienstleistungen/unternehmensberatung/agile-consulting.html?gclid=Cj0KCQjw0PWRBhDKARIAPKHFQjoQu3GFrkkNmPe0VozUaMMRGSFROI73ZNN39j779wcB>
- [116] „Flex-Box.” Online verfügbar: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- [117] „W3Schools.” Online verfügbar: <https://www.w3schools.com>
- [118] „Wait ... Sekunden befor man nächste zeile ausführt.” Online verfügbar: <https://stackoverflow.com/questions/14226803/wait-5-seconds-before-executing-next-line>
- [119] „AsciiDock Befehle,” Für Doku. Online verfügbar: <https://docs.asciidoc.org/asciidoc/latest/syntax-quick-reference/>

Abbildungsverzeichnis

1.	Multiple-Choice-Frage von [3]	5
2.	Beispiel für die Erstellung eines Fragebogen von [4]	5
3.	Fortschrittsanzeige bei der Beachtung der Fragebögen von [5]	6
4.	Fortschrittsanzeige bei der Beachtung der Fragebögen und Antwortoptionendesign von [6]	7
5.	Visuelle Trennung der Fragen von den Antwortoptionen von [8]	8
6.	Anpassung und Dateien zu einer Frage von [8]	8
7.	Antwortoptionendesign von [8]	9
8.	Use Case Diagramm	11
9.	Systemarchitektur	14
10.	Aktivitätsdiagramm	15
11.	Erste Version des Datenmodell (ERD)	16
12.	Erste Version des Datenmodell (CLD)	16
13.	Zweite Version des Datenmodells	18
14.	Zwischenversion des Datenmodells	19
15.	Letzte Version des Datenmodells	20
16.	Angular Logo von https://angular.io	21
17.	TypeScript Logo von https://thenewstack.io/how-typescript-helps-enterprise-developers/	22
18.	CSS Logo	23
19.	CSS Logo von https://de.wikipedia.org/wiki/HTML5	24
20.	NodeJS Logo von https://de.wikipedia.org/wiki/HTML5	25
21.	CSS Logo von https://www.cloudflight.io/de/blog/docker-container-die-zukunft-moderner-applikationen-und-multi-cloud-deployments/	29
22.	Unterschied von Virtuellen Maschinen und Containern von https://www.shd-online.de/fachartikel/der-app-store-fuer-server-wofuer-sie-container-brauchen/	30
23.	Beispiel eines Dockerfiles	31
24.	Beispiel für Docker Compose YAML-Datei	33
25.	Keycloak Logo von https://www.gepardec.com/keycloak/	33
26.	Login Prozess von https://medium.com/codex/introduction-to-keycloak-227c3902754a	35
27.	Admin Console	37
28.	Funktionsweise Traefik	38
29.	PlantUML Logo source: [88]	38
30.	Markdown Logo source: [90]	39
31.	Java Logo [?]	39
32.	Quarkus Logo [?]	40
33.	PostgreSql Logo [?]	42
34.	Rest-API Model [?]	43
35.	HTTP-Statuscodes [?]	45
36.	HTTP-Methoden [?]	46
37.	Dashboard Traefik	61

38.	Auswertung	65
39.	Auswertung	66
40.	CORS Fehler	69
41.	Cyclic-Object-Error	70
42.	Veranschaulichung von Scrum https://studyflix.de/wirtschaft/scrum-methode-3426	90
43.	GitHub Repository von Leo-Survey	92
44.	GitHub Repository von Leo-Survey Commites	92
45.	GitHub Desktop Beispiel	93
46.	YouTrack Sprint Beispiel	94
47.	GitHub-Anbindung in YouTrack	94
48.	YouTrack Burn Down Chart Beispiel	95
49.	GitHub-Anbindung in YouTrack	95
50.	Skizze Seite 1	96
51.	Skizze Seite 2	97
52.	Moodboard Leo-Survey	98
53.	ScreenDesign Startseite	99
54.	ScreenDesign Login	99
55.	ScreenDesign Dashboard	99
56.	ScreenDesign Fragebogen erstellen	100
57.	ScreenDesign Fragebogen erstellen	100
58.	ScreenDesign Fragebogen erstellen	100
59.	ScreenDesign Fragebogen erstellen	101
60.	ScreenDesign Fragebogen erstellen	101
61.	ScreenDesign Fragebogen erstellen	101
62.	ScreenDesign Fragebogen ausfüllen	102
63.	Startseite	103
64.	Login mittels Keycloak	104
65.	Dashboard	104
66.	Fragebogen erstellen	105
67.	Startseite	105
68.	Startseite	106

Tabellenverzeichnis

1.	Meilensteinliste Version 1	127
2.	Meilensteinliste Version 2	127

Quellcodeverzeichnis

1.	Internationalization-Makierung im HTML	53
2.	xliff-Datei	53
3.	Konfigurieren der Sprachen im angular.json	54
4.	automatisches Starten der deutschen Sprache	54
5.	Semi-Transparentes Eingabefeld	55
6.	Konfiguration des Keycloaks	55
7.	Aufrufen des Logins	55
8.	Initialisierung des AuthenticationService	55
9.	Aufrufen des Authentifikationswächters	56
10.	Authentifikationswächter	56
11.	Login und Logout	57
12.	Verwendung der geschickten Daten TypeScript	58
13.	Verwendung der geschickten Daten HTML	58
14.	CSV-Download	58
15.	Formularüberprüfung mit reakitven Formularen TypeScript	59
16.	Formularüberprüfung mit reakitven Formularen HTML	59
17.	Pageinator	60
18.	Berechnen der anzuzeigenden Fragen TypeScript	60
19.	Starten Traefik	61
20.	Keycloak Dockerfile	61
21.	Auszug aus den importieren Einstellungen	62
22.	Postgres Dockerfile	62
23.	Fragebogen duplizieren	63
24.	Auswertung	65
25.	Auswertung	66
26.	@Lob	67
27.	Bild in Byte Array umwandeln	68
28.	Bild mit Session in DB speichern	69
29.	Lösung CORS Error	69
30.	Klassen mit Cyclic-Error	70
31.	Klassen ohne Cyclic-Error	71