Technology
Arts Sciences
TH Köln

Technische Hochschule Köln
Fakultät für Informatik und Ingenieurwissenschaften

BERICHT ZUM PRAXISPROJEKT

Plattform für die Spielersuche

Eine mit Javascript entwickelte Plattform

Vorgelegt an der TH Köln Campus Gummersbach
im Studiengang Wirtshaftsinformatik

ausgearbeitet von:
Timo Kalter 12345678
Carlo Menjivar 11117929

Erster Prüfer: Frau Prof. Birgit Bertelsmeier
Zweiter Prüfer: <Name des 2. Prüfers>

Gummersbach, im <Monat der Abgabe>

Zusammenfassung

Platz für das deutsche Abstract...

Abstract

Platz für das englische Abstract...

Das Abstract

Bei einem Abstract handelt es sich um eine Art *Zusammenfassung* Ihrer Arbeit. Diese kann in deutscher und/oder englischer Sprache verfasst werden. Mithilfe des Abstracts kann der Leser sich zügig orientieren, in wie fern Ihre Arbeit für ihn Relevanz besitzt.
Sprechen Sie unbedingt mit Ihrer Betreuerin/Ihrem Betreuer, ob Sie für Ihre Arbeit ein Abstract benötigen.
Ein Abstract beinhaltet folgende Aspekte [a]:

- Ziel der Arbeit
- Fragestellung der Arbeit
- Herangezogener, theoretischer Ansatz ("Quellen")
- *Optional:* Methodik

[a] Vgl. [SW11], S. 249

Hinweise zu dieser Dokumentvorlage

- Es handelt sich hierbei um eine Beispiel-Vorlage für wissenschaftliche Ausarbeitungen. Über die konkrete, formale Ausgestaltung Ihrer wissenschaftlichen Arbeit sprechen Sie unbedingt mit Ihre/m Betreuer/in.
- Unabhängig, ob Sie beispielsweise eine Bachelor-, Master- oder Hausarbeit schreiben müssen. Diese Vorlage kann als eine gute Basis für Ihre Arbeit dienen. Passen Sie einfach die Vorlage Ihren Anforderungen entsprechend an.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Problemstellung

Unsere Aufgabe ist es, eine Webplattform für Spieler des Videospiels "League of Legendsßu schaffen, auf der sich Spieler kennenlernen und zu einer gemeinsamen Partie verabreden können. Um eine möglichst gute Plattform zu bieten, sind verschiedene Schritte zu erledigen.

1.1 Aufmerksamkeit/Marketing

Im Rahmen der Projektarbeit steht Marketing nicht im Fokus. Wir beschreiben daher nur kurz, worauf zu achten ist, wenn das Produkt nach dem Praxisprojekt auf den Markt kommen soll.

Wie bei sozialen Plattformen und sozialen Netzwerken üblich, erhöht sich der Nutzen durch positive Netzwerkeffekte mit steigender Nutzeranzahl. Ein Ziel ist es daher, eine möglichst große Nutzerbasis aufzubauen, die unsere Webseite möglichst oft verwendet. Zuerst müssen wir Mögichkeiten finden, damit Personen von unserer Webseite überhaupt erfahren. Auf technischer Seite ist dies mit Suchmaschinenoptimierung möglich, Marketingtechnisch haben wir verschiedene Möglichkeiten der Werbung, unter anderem Bannerwerbung, Influencermarketing und "Kunden werben Kunden".

Sobald Personen unsere Webseite besuchen, müssen diese überzeugt werden, dass unser Produkt gut ist und dass sie ein Konto erstellen sollten. Es sollte daher auf einer ansprechenden Startseite das Produkt vorgestellt werden, auf dem erklärt wird, was das Ziel unserer Plattform ist und wie wir dieses Ziel erreichen wollen.

Je besser unser Produkt ist, desto wahrscheinlicher ist es, dass ein Nutzer durch intrinsische Motivation unsere Webseite verwendet. Es ist daher wichtig, ein Produkt mit möglichst hoher Qualität zu erstellen. Ein Nutzer, der von unserer Webseite begeistert ist, wird wahrscheinlicher die Webseite öfter verwenden und Freunden von der Webseite erzählen, was wiederum neue Nutzer generieren kann.

1.2 Das Nutzerkonto

Um die Webseite vollständig nutzen zu können, sollen Besucher der Seite ein Konto erstellen. Dazu sollen sie eine E-Mail angeben sowie ein Passwort und einen freigewählten Nutzernamen aussuchen. Auch sollen sie nach Kontoerstellung Nutzerdaten wie ein Profilbild, das Alter und einen Freitext angeben können. Die angegeben Daten sollen dem Nutzer dabei helfen, Kontakte auf der Plattform zu finden und können als Gesprächsthema dienen.

Auch ist geplant, dass das Nutzerkonto mit dem Riot-Konto verbunden werden kann. Dies soll es ermöglichen, dass Spielstatistiken, wie zum Beispiel die meist gespielten Cham-

pions, die Lieblingsposition und die Elo, angezeigt werden. Ein Nutzer, der auf Kontaktsuche ist, erhält damit mehr Informationen über den Spieler und kann somit besser abschätzen, ob er Kontakt aufnehmen will.

1.3 Kontaktsuche

Um mit anderen Nutzern in Kontakt treten zu können, soll es möglich sein, andere, zufällige Nutzerprofile anzuzeigen. Profile sollen einzeln hintereinander angezeigt werden; nachdem sich der Nutzer entschieden hat, ob er dem angezeigten Nutzer einen Like [GLOSSAR!] vergeben will, wird das nächste Profil angezeigt.

Sobald zwei Nutzer gegenseitig einen Like vergeben haben, sollen diese befreundet sein. Befreundete Nutzer sollen untereinander Nachrichten austauschen können.

1.4 Filter

Um die Qualität der angezeigten Profile zu erhöhen und eine möglichst gute Nutzererfahrung zu gewährleisten, soll es dem Nutzer möglich sein, seine Suche einzugrenzen. Durch Filter werden ihm nur die Nutzer angezeigt, die für ihn relevant sind.

1.5 Regeln und Richtlinien

Wir möchten mit unserer Plattform einen sicheren Ort bieten, auf dem die Spieler unabhängig ihrer Eigenschaften respektiert werden. Wir sind der Meinung, dass die meisten Nutzer in der Lage sind, respektvoll miteinander umzugehen. Dies können wir jedoch nicht gewährleisten.

Wir wollen uns daher das Recht vorbehalten, Nutzern, die gegen unsere Regeln und Richtlinien verstoßen, den Zugang zu unserer Plattform zu verwehren. Sollte sich ein Nutzer von einem anderen Nutzer beleidigt fühlen oder andersweitig der Meinung sein, dass dieser Nutzer gegen die Regeln verstößt, soll dieser Regelbrecher gemeldet werden können. Wir als Betreiber der Plattform sollen in der Lage sein, die Meldungen zu verarbeiten und angemessen mit den Regelbrechern zu verfahren.

Unabhängig davon soll es den Nutzern möglich sein, andere Nutzer zu blockieren. Blockierte Nutzer sollen dem aktiven Nutzer nicht mehr in der Spielersuche angezeigt werden, nicht mehr Nachrichten verschicken und aktive Chats sollen aufgelöst werden. Einen anderen Spieler zu blockieren heißt nicht zwangsläufig, dass dieser gegen Regeln oder Richtlinien verstößt, es kann auch sein, dass der Nutzer einfach den Kontakt abbrechen will. Sollte ein Nutzer jedoch gemeldet werden, soll er automatisch blockiert werden.

1.6 Internationalisierung

Die Webseite soll in Englisch angeboten werden, da dies eine der weit verbreitetsten Sprachen der Welt ist und damit viele Nutzer erreicht.

1.7 Einstellungen

Um den verschiedenen Vorlieben unterschiedlichster Nutzer gerecht zu werden, sollen diese in der Lage sein, verschiedene Optionen in den Einstellungen zu verwalten. Unter anderem sollen Nutzer in der Lage sein, auszusuchen, wie und ob sie über neue Nachrichten, Freundschaften und Neuigkeiten informiert werden.

1.8 Nachrichten

Sobald zwei Nutzer befreundet sind, können sie sich gegenseitig Nachrichten schreiben. Dazu soll auf einer dafür eingerichteten Seite in ein Eingabefeld der zu versendende Text eingegeben werden. Sobald die Nachricht abgeschickt wird, soll der andere Nutzer möglichst Zeitnah von dieser erfahren. Dies soll den glatten Ablauf von Chats ermöglichen.

Zeichen sollen nach dem UTF-8 Zeichensatz erlaubt sein.

1.9 Sicherheit

Die Sicherheit der Nutzerdaten ist von hoher Priorität. Passwörter müssen besonders geschützt und nach derzeitigem kryptologischen Stand der Technik gesichert werden, um den Zugriff von unautorisierten Angreifern zu unterbinden. Identifizierende Daten wie E-Mail-Adressen dürfen nicht öffentlich einsehbar sein.

Sollte in einem späteren Schritt die Plattform veröffentlicht werden, ist auf DSGVO-Konformität zu achten.

1.10 Dokumentation

"Technische Schulden"(en. "technical dept", schlechte Umsetzung von Software, die einen erheblichen Mehraufwand in der Zukunft bedeutet) sorgen in vielen Fällen für Probleme im Laufe des Produktlebenszyklus. Um die Wahrscheinlichkeit zu verringern, dass sich technische Schulden anhäufen und, soll eine gut leserliche Dokumentation zum Quellcode geschrieben werden, welche es allen Gruppenmitgliedern ermöglichen soll, den überblick zu halten, welcher Codeschnipsel welches Problem löst. Desweiteren sollen umfangreiche Tests durchgeführt werden, welche die Qualität des Codes gewährleisten sollen.

Um die Kommunikation zwischen Backend und Frontend zu gewährleisten, sollen Schnittstellen entsprechend dokumentiert werden. Es sollte einem Frontend-Entwickler

möglich sein, nur mit der Dokumentation auf die Schnittstelle zuzugreifen und die Informationen zu erhalten, die er benötigt, um entsprechende Daten im Frontend anzeigen zu können.

1.11 Konto löschen

Es wird Benutzer geben, die aus verschiedensten Gründen unsere Webseite nicht weiter verwenden wollen und ihr Konto löschen wollen. Unsere Aufgabe ist es, eine sichere Löschung des Benutzerkontos zu gewährleisten und persönliche Daten nach Datenschutzvorschriften aus der Datenbank zu entfernen.

Optional können wir zudem den Nutzer darum bitten, ein Formular auszufüllen, in dem dieser uns Rückmeldung geben kann, aus welchen Gründen er die Webseite nicht weiter verwenden will.

2 Einleitung

WORK IN PROGRESS...

2.1 Einführung in das Thema (Motivation, zentrale Begriffe etc.)

2.2 Hinführung zu den Ergebnissen

2.3 Ggf. Angabe des Schwerpunktes

2.4 Ggf. Einschränkungen darlegen

2.5 Problemstellung

2.6 Zielstellung der Arbeit

2.7 Fragestellung der Arbeit

2.8 Struktur der Arbeit

Dieser Projektbericht ist in folgenden Kapitel unterteilt: **Kapitel 2**

Kapitel 3

Kapitel 4? erläutert, welche JavaScript-Frameworks zu Beginn des Projekts berücksichtigt wurden. Er gibt einen Überblick über die Faktoren, die zu beachten sind, wenn man sich für React entscheidet, und zeigt schließlich, wie die Daten mit Hilfe der React JavaScript-Bibliothek und ihrer Hooks für den Endbenutzer visualisiert werden.

Kapitel 5? geht es um die Interaktion zwischen dem Client und dem Server.

Anhand von zwei Beispielen wird gezeigt, wie Lese- und Schreibabfragen mit Hilfe von GraphQL und ApolloClient durchgeführt wurden.

Kapitel 6? zeigt die Relevanz von Testfällen. Sie zeigt auch die Vorteile automatisierter und eingebetteter Testfällen in der Entwicklungsumgebung.

Die Einleitung(DELETE AFTER CHECK WE COMPLETE ALL POINTS)

Die Einleitung umfasst folgende Elemente[a]:

- Einführung in das Thema (Motivation, zentrale Begriffe etc.)
- Hinführung zu den Ergebnissen
- Ggf. Angabe des Schwerpunktes
- Ggf. Einschränkungen darlegen
- Problemstellung
- Zielstellung der Arbeit
- Fragestellung der Arbeit
- Übersicht über die Kapitel geben:

> Eine Einleitung muss auch durch die Arbeit führen. Sie muss dem Leser helfen, sich in der Arbeit und ihrer Struktur zu Recht zu finden. Für jedes Kapitel sollte eine ganz kurze Inhaltsangabe gemacht werden und ggf. motiviert werden, warum es geschrieben wurde. Oft denkt sich ein Autor etwas bei der Struktur seiner Arbeit, auch solche Beweggründe sind dem Leser zu erklären[b]:.

[a] Vgl. u.a. [BBoJ], S. 5-6
[b] [BBoJ], S. 6

3 Grundlagen

TEXT FOLGT...

3.1 Unterabschnitt von Grundlagen

TEXT FOLGT...

Das Kapitel/der Abschnitt

Hierbei handelt es sich um ein Beispiel-Kapitel. Es ist zu empfehlen, dass Sie Kapitel und auch Abschnitte immer mit einer kurzen Einleitung beginnen. In dieser beschreiben Sie kurz, was den Leser in diesem Kapitel/Abschnitt erwartet. Bei einem Kapitel mit Abschnitten nehmen Sie auch inhaltlichen Bezug auf die enthaltenen Abschnitte (inklusive Referenzierung auf die Abschnittsnummerierung).

Abbildungen, Tabellen & Co.

Bei Verwendung von Tabellen und auch Abbildungen beachten Sie bitte, dass diese immer Unter-/Überschriften enthalten (inklusive einer Nummer). Im Textfluss erklären/beschreiben Sie die Abbildung bzw. die Tabelle und nehmen Bezug über einen Verweis auf die Nummer.

4 Benutzeroberfläche

Nachdem die Projektanforderungen definiert waren, wurden verschiedene Optionen für die Entwicklung der Benutzeroberfläche bewertet.

Folgende JavaScript Frameworks wurden berücksichtigt: Angular, Vue und React.

Es war möglich das Projekt auch mit Angular durchgeführt werden können, mit dem Unterschied, dass die Entwicklung mit Angular mehr Zeit und Lernaufwand gekostet hätte, außerdem ist das Projekt nicht komplex genug, um das Angular-Ökosystem zu benötigen.

4.1 Überlegungen bei der Verwendung von React

Die Entscheidung zwischen Vue und React wurde unter anderem wegen des Unternehmens hinter React getroffen, zwar Facebook. Auch für persönliche Vorlieben.

4.1.1 Vorteile

Deklarativ

Mit React ist es leicht, interaktive Benutzeroberflächen zu erstellen. Man kann einfache Ansichten für jeden Zustand der Anwendung. React aktualisiert und rendert effizient genau die richtigen Komponenten, wenn sich deren Daten ändern. Durch deklarative Ansichten wird der Code vorhersehbarer und einfacher zu debuggen.

Komponentenbasiert

Gekapselte Komponenten, die ihren eigenen Zustand verwalten. Da die Komponentenlogik in JavaScript geschrieben wird, kann man problemlos umfangreiche Daten durch die Anwendung leiten und den Zustand aus dem DOM heraushalten.

Große Entwickler-Community

React besteht aus rund 56.162 professionellen Entwicklern auf der ganzen Welt.

Laut einer StackOverFlow Umfrage hat React.js im Jahr 2021 jQuery als das am häufigsten verwendete Web-Framework überholt. [1]

[1] Vgl. [SO01]

4.1.2 Nachteile

Bei der Auswahl des Frameworks wollten wir so unvoreingenommen wie möglich sein, daher listen wir einige Aspekte auf, die bei Projekten mit React zu beachten sind.

JSX

Während dies für einige Entwickler ein Nachteil sein könnte, ist es wichtig zu beachten, dass JSX auch seine Vorteile hat und hilft, den Code vor Injektionen zu schützen.

Ein hohes Entwicklungstempo

Entwickler, die das Entwicklungstempo als Nachteil sehen, würden argumentieren, dass sie die Arbeit mit React ständig neu erlernen müssen und es schwierig ist, damit Schritt zu halten.

Es ist wichtig festzustellen, dass neue Entwicklungen des Frameworks verbessern und dazu beitragen, dass er ein höheres Leistungsniveau erreicht.

Eine zu leichte Dokumentation

Aufgrund der rasanten Entwicklung ist die Dokumentation in Bezug auf die neuesten Aktualisierungen und Änderungen oft spärlich. [2]

Das Projekt besteht aus Komponenten

In diesem Kapitel wird der Zweck der einzelnen Komponenten erläutert.... TO-DO und SOLLTE Ueberhaupt DIESES KAPITEL GESCHRIEBEN WERDEN?

4.2 React Hooks

Beginnend mit 16.8.0, enthält React eine stabile Implementierung von React Hooks.

4.2.1 useState

Der Hook useState gibt uns die Möglichkeit, den Zustand unserer Anwendung zu verwalten. Sie besteht aus mindestens einen Wert und einer Funktion, die die besagte Variable aktualisiert. Der Wert bei der Definition kann ein Zahl, ein String, ein Array oder sogar ein Objekt sein. Darüber hinaus kann bei der Definition von useState ein Anfangswert festgelegt werden.

```
const [wert, definiereWert] = useState(Anfangswert);
```

In unserem Projekt haben wir useState verwendet, um alle damit verbundenen Informationen zu manipulieren und dem Benutzer anzuzeigen.

[2] Vgl. [R01]

```
const [profile, setProfile] = useState(state)
```

Die Variable state enthält die Daten des angemeldeten Benutzers. Diese Daten sind die Antwort auf die an den Server gesendete Anfrage. Hier sind einige Beispieldaten für die Variable state.

```
name: "gilo2754"
aboutMe: "I like video code video games too"
avatar:
    "https://lol-friendfinder-dev.s3.eu-central-1.amazonaws.com/avatars/a028fd00-e767-
dateOfBirth: "1990-06-01T00:00:00.000Z"
gender: "male"
ingameRole: (2) ["Fill", "Bot"]
languages: (2) ["bm", "de"]
friends:(5) ["60eb61b33a2481451c1cd7ac", "60eb61b33a2481451c1cd7b0",
    "60eb61b33a2481451c1cd7b1", "60eb61b33a2481451c1cd7b4",
    "60eb61b33a2481451c1cd7b7"]
blocked: (1) ["60eea5eed7e23b5ef08f68af"]
```

In späteren Kapiteln wird näher erläutert, wie diese Informationen, die wir durch einen einzigen Aufruf an den Server erhalten, in den Chat- und Benutzerkomponenten genutzt werden können.

Wie man sieht, gibt es innerhalb des states nicht nur einen Wert, sondern mehrere. Es handelt sich eigentlich um ein Objekt. Objekte sind dasselbe wie Variablen in JavaScript, der einzige Unterschied ist, dass ein Objekt mehrere Werte in Form von Eigenschaften und Methoden enthält. Objekte können andere Objekte, Strings, Arrays, Zahlen und so weiter enthalten.

4.2.2 useEffect

useEffect ermöglicht es uns, verschiedene Arten von Effekten zu erzeugen, nachdem eine Komponente gerendert wurde.

Der Hook useEffect entspricht einer Kombination aus componentDidMount, componentDidUpdate und componentWillUnmount.

In unserem Projekt manipulieren wir die Anzeige der Daten abhängig der Änderungen des lokalen Zustands.
Im Zustand „profile“ werden die Änderungen, die der Benutzer an seinen Daten vornimmt, bevor sie in der Datenbank gespeichert werden.
Der globale Zustand „state“ ist in der Komponente App enthalten. In diesem wird die Antwort von der Datenbankenabfrage gespeichert.
Wir definieren den Zustand als Abhängigkeit von useEffect, so dass er bei jeder Änderung in der Benutzeroberfläche aktualisiert wird.

```
useEffect(() => {
      setProfile(state)
  }, [state])
```

Es ist möglich, mehrere useEffects in unserem Code zu haben und mit ihrer jeweiligen Abhängigkeiten separat definieren. Wenn wir wollen, dass der Code in unserem useEffect nur einmal nach dem ersten Rendering ausgeführt wird, definieren wir ein leeres Array als Abhängigkeit.

useContext hat es uns ermöglicht, Benutzerdaten auf einfache Weise in den Komponenten zu teilen, in denen diese Daten benötigt werden. In frühen Versionen der Anwendung wurde für den gleichen Zweck props verwendet. Unserer Erfahrung nach ist useContext eine elegantere und sauberere Lösung für da gleiche Ziel.

useContext Provider

```
<MyContext.Provider value={/* irgendein Wert */}>
```

Jedes Context-Objekt kommt mit einer Provider React Komponente, welche konsumierenden Komponenten erlaubt, die Veränderungen von Context zu abonnieren. [3]

In unserem Projekt haben wir den Provider in der Komponente App folgendermaße definiert.

```
import { React, useState, useEffect, createContext } from "react"

export const GlobalContext = createContext()
export default function App() {
    const [token, setToken] = useState()
    const [state, setState] = useState(null)

    ...

    return (
        <GlobalContext.Provider value={{ token, setToken, state, setState,
            refetch }}>
          <>
            <MyNavbar />

            <Switch>
              <Route exact path="/" component={Home} />
              <Route path="/login" component={() => <Login />} />
              <Route path="/signup" component={SignUp} />
              <Route exact path="/users" component={() => <Users />} />
              <Route exact path="/profile" component={() => <Profile />} />
              <Route exact path="/chat" component={() => <Chat />} />
              <Route exact path="/ChatMessage" component={() =>
                  <ChatMessage />} />
```

[3] Vgl. u.a. [R02]

```
              <Route component={NotFound} />
            </Switch>
          </>
        </GlobalContext.Provider>
      )
    }
//Code-Auszug in frontend/src/App.js
```

Mit dem obigen Code sind wir bereit, die Werte innerhalb von value in jeder Komponente innerhalb des Contexts zu verwenden.

In unserem Fall können die Werte token und state gelesen werden. Darüber hinaus sind die Funktionen setToken, setState und refetch ebenfalls verfügbar.

useContext Consumer

Nachfolgend zeigen wir, wie GlobalContext in der Komponente Chat.js verwendet wurde, um die Freunde des angemeldeten Benutzers zu ermitteln.

Die Kennungen der Freunde waren notwendig, um die Kommunikation zu ermöglichen und die zwischen den Freunden ausgetauschten Nachrichten zu laden.

```
    import { useContext, useEffect, useState, React } from "react"
    import { GlobalContext } from "../App"

    //mit Hilfe von Destrukturierende Zuweisung wurden neue Variablen und
       Funktionen definiert
    const { token, state, setState, refetch } = useContext(GlobalContext)

    ...

    return !token ? (

    ...

        <div className="user-many">
        {state?.friends &&
          state?.friends?.map((item, index) => {
            return (

//Diese Komponente gibt den Namen und den Avatar des Benutzers auf der
   Grundlage der eingegebenen ID zurueck.
              <FriendList
                setUserID={setUserID}
                setUserNameChat={setUserNameChat}
                setChatAvatar={setChatAvatar}
```

```
                userId={item.user}
                searchUser={searchUser}
                setSearchUser={setSearchUser}
                key={index + 1}
              />
            )
          })}
        </div>
      )

      \dots
//Code-Auszug in frontend/src/Chat.js
```

4.2.3 Alternative zu useContext

Eine frühere Version der kürzlich erläuterten Implementierung verwendete „props“ als Mittel zur gemeinsamen Nutzung von Daten zwischen Komponenten. Wir beschlossen, diese Idee zu ändern, da der Code schwieriger zu pflegen war, da wir gezwungen waren, jedes Mal auf die Werte der importierten Variablen zuzugreifen.

5 Serveranfragen

5.1 GraphQL

GraphQL ist eine Abfragesprache und Server-Laufzeitumgebung für APIs. Ihre Aufgabe ist es, genau die Daten zu liefern, die anfordert werden, und nicht mehr.
Mit GraphQL sind APIs schnell, flexibel und einfach für Entwickler.

Laut dem 2020 State of the API Report von Postman.com steht GrapQL an fünfter Stelle der spannendsten Technologien für 2021.

Im Hinblick auf die Art und Weise, wie Abfragen an den Server mithilfe von GraphQL behandelt werden können, sind folgende Aspekte zu beachten.

Vorteile

- GraphQL-Aufrufe werden in einem einzigen Round Trip gehandhabt. Wir bekommen genau die Daten, die angefragt haben (kein Over-Fetching).
- Stark definierte Datentypen verringern das Risiko einer Fehlkommunikation zwischen Client und Server.
- GraphQL ist introspektiv. So können wir eine Liste der verfügbaren Datentypen anfordern. Dies ist ideal für automatisch erstellte Dokumente.
- Eine Anwendungs-API kann sich mit GraphQL weiterentwickeln, ohne dass bestehende Anfragen beeinträchtigt werden.
- GraphQL schreibt keine spezifische Anwendungsarchitektur vor. Es kann auf einer vorhandenen REST-API installiert und mit aktuellen API-Management-Tools verwendet werden.
- Als Alternative zu REST ermöglicht GraphQL Entwicklern die Erstellung von Abfragen zur Extraktion von Daten aus mehreren Quellen mit einer einzigen API-Abfrage.

Nachteile

- Für Entwickler, die sich bereits mit REST-APIs auskennen, bedeutet GraphQL weiteren Lernaufwand.
- Mit GraphQL verschiebt sich die Funktionalität von Datenabfragen zur Serverseite, was zusätzliche Komplexität für Serverentwickler bedeutet.

[4]

[4] Vgl. u.a. [RH1]

Implementierung der Serveranfragen

Alle Abfragen und Mutationen wurden in einem separaten Ordner gesammelt. Damit soll eine saubere Struktur des Codes gewährleistet werden. Diese wurden für die spätere Verwendung in den React-Komponenten exportiert.

Beim Einloggen in unseren Account werden die entsprechenden Daten geladen, für die zunächst mithilfe der Apollo Client Hook useQuery Daten gelesen werden. In ähnlicher Weise wurde der Schreibprozess mit dem Hook useMutation implementiert.
In diesem Kapitel wird erläutert, wie die beiden oben genannten Abfragen implementiert wurden.

5.1.1 Leseabfrage

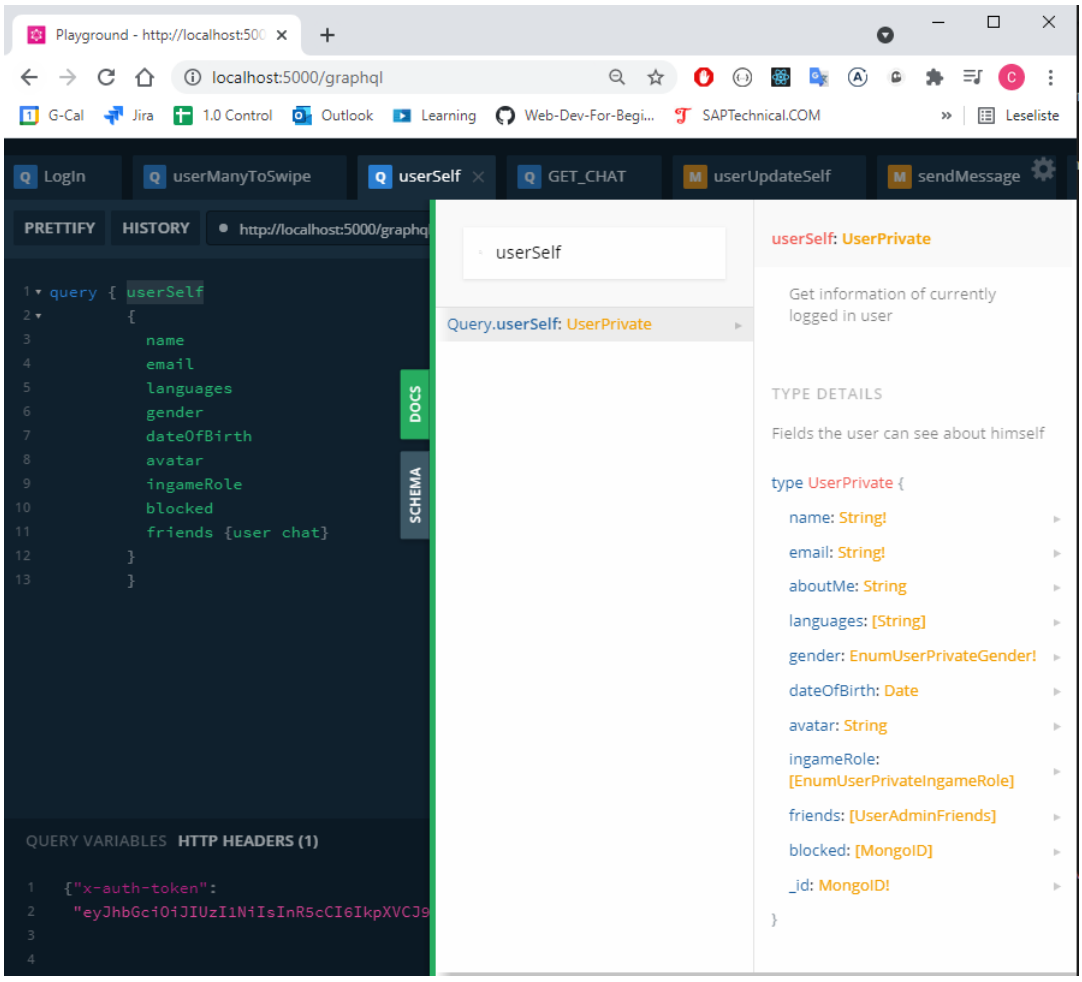


Abbildung Unterunterabschnitt 5.1.1:
Beim GraphQL Playground: alle verfügbaren Felder und deren Datentyp in Bezug auf eine Beispielabfrage auf.

Genau derselbe Code wurde für die Abfrage von Benutzerinformationen verwendet.

Apollo Client

Nachdem eine Abfrage exportiert wurde, ist sie bereit, in einer React-Komponente importiert
und angewendet zu werden.

```
import { GET_MY_INFO } from "./GraphQL/Queries"
import { useQuery } from "@apollo/client"

const { loading, error, data } = useQuery(
GET_MY_INFO,
ContextHeader(token),
)
//Code-Auszug in frontend/src/App.js
```

Die Konstante ContextHeader enthält das Token in der Struktur, die erforderlich ist, um die Abfrage nur dann stellen zu können, wenn der Benutzer dazu berechtigt ist.
Sollte das Token einen undefinierten Wert, null oder ungültig enthalten, wird der Server ein Fehler zurückgegeben.
Der useQuery Hook liefert ein Ergebnisobjekt, welches eine der folgenden Optionen zurückgibt.

loading:

Ein boolescher Wert, der angibt, ob die Abfrage in Bearbeitung ist. Wenn loading wahr ist, ist die Anfrage noch nicht abgeschlossen. Typischerweise kann diese Information verwendet werden, um einen Lade-Spinner anzuzeigen.

error:

Ein Laufzeitfehler mit den Eigenschaften von GraphQL Errors und network Error. Dieses enthält Informationen darüber, was bei der Abfrage fehlgeschlagen ist.

date:

Ein Objekt, das das Ergebnis der GraphQL-Abfrage enthält.
Es enthält die tatsächlichen Daten vom Server.

5.2 Axios

Zusätzlich zu den GraphQL-Abfragen wurde eine Post-Anfrage mit Axios bereitgestellt. Mit dieser war es möglich, Bilder auf die S3 Speicher von AWS hochzuladen.

> Axios ist ein Promise-basierter HTTP-Client für node.js und den browser. Auf der Server-Seite wir das modul http verwendet, während im Browser XML-HttpRequests (ajax) ausgeführt werden.

[5]

```
function disableBtn() {
    const uploadBtn = document.getElementById("uploadBtn")
    uploadBtn.disabled = true
    uploadBtn.style.background = "#000000"
  }

  function fileUploadHandler() {
    errored ?
     disableBtn()
     /*
    Tritt ein Fehler auf, wird die Abfrage nicht gesendet und der Knopf
        deaktiviert
    Ein Fehler tritt auf, wenn die Datei zu gross ist oder oder wenn sie
        ein ungueltiges Format hat
    */
    :
    console.log("uploading pic...", file?.name)
    const fd = new FormData()
    /* avatar ist der Name der Datei, die hochgeladen wird
      file ist der zu sendende Wert */
    fd.append("avatar", file)

    axios
      .post(urlAvatar, fd, {
        headers: {
          "x-auth-token": TOKEN,
        },
      })
      .then((res) => {
        setState((state) => ({ ...state, avatar: res?.data?.location }))
        //In location finden wir die URL fuer das gerade hochgeladende Bild
```

[5] Vgl. u.a. [AX1]

```
        })
    }
```

Das HTML-Eingabefeld „input“ wurde folgendermaßen definiert.

```
<input type="file" onChange={fileSelectedHandler} />
```

Das Format der hochzuladenden Dateien wurde limitiert, damit nur zulässige Dateien an den Server gesendet werden.

```
const admittedImageFormats = ["png", "jpg", "jpeg"]
```

Die Größe der hochzuladenden Datei wurde um 1 MB abgegrenzt. Durch die Eigenschaft „size“ der ausgewählten Datei konnten wir auf die Größe der Datei zugreifen.

```
const imageSize = e.target.files[0].size
```

6 Qualitätssicherung

Software Testing

Obwohl unser Projekt relativ klein ist, wollten wir, wo immer möglich, automatisierte Testfälle einbeziehen.

Für das Frontend haben wir End-to-End Testfälle mit Cypress geschrieben.

So können wir in Sekundenschnelle feststellen, ob etwas in unserer Anwendung defekt ist.

Ein mögliches Szenario ist, dass eine Komponente, die in anderen Komponenten verwendet wird, geändert wurde, ohne zu berücksichtigen, dass sie in einer anderen Komponente für einen anderen Zweck verwendet wurde.

Das ist der Fall bei der Komponente AvatarImage. Dies ist eine Funktionskomponente, die 3 Parameter erhält: Größe des Bildes, Bild-URL und Benutzername.

Zu Beginn des Projekts wurde nicht daran gedacht, die Größe des Bildes über einen Parameter dieser Funktion zu steuern. Im Laufe des Projekts wurde uns klar, dass wir die Logik in diesem Element wiederverwenden konnten.

Innerhalb der Komponente wird geprüft, ob eine URL existiert, und wenn ja, wird das mit dem Link verbundene Bild gezeichnet. Falls es keine URL angegeben wurde, werden die ersten beiden Buchstaben des Benutzernamens verwendet, um ein Standardsymbol zu erzeugen.

In der aktuellen Version des Codes wird diese Komponente in vier anderen Komponenten wieder verwendet. Wenn das Projekt weiter wachsen würde, würde auch die Möglichkeit von Fehlern im Code zunehmen. Und es wäre schwieriger, sie zu finden.

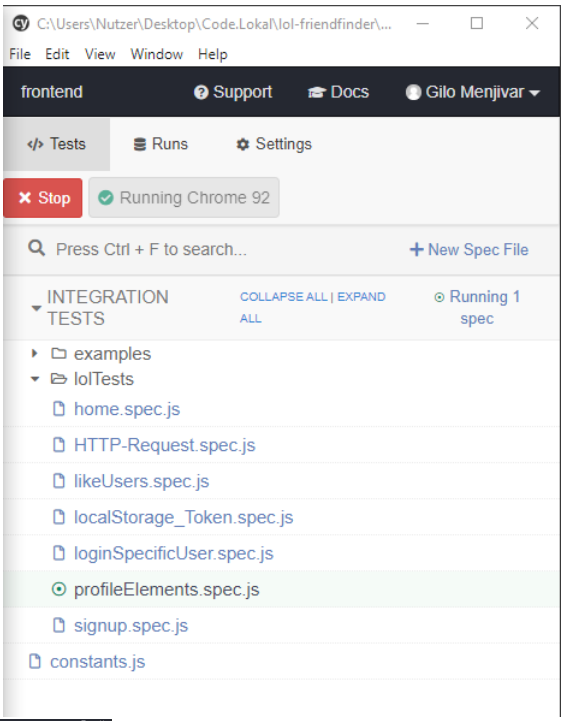
Ohne automatisierte Tests müssten Sie normalerweise manuell prüfen, ob die übergeordneten Komponenten noch wie erwartet funktionieren.

```
//in constants.js sind Standard-Testdaten
import * as c from "../constants.js"

const qtyLanguages = 101
const qtyGenders = 8
const qtyInGameRoles = 6

describe("Check if all profile elements have been rendered", () => {
  it("Login with existing account", () => {
    cy.typeLogin(c.email, c.password)
  })

  it("Username was rendered", () => {
    cy.get('#username').should("be.visible")
  })
```

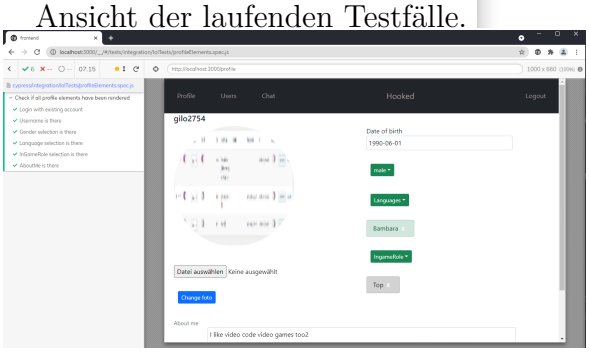
```
  it("Gender selection was rendered", () => {
    cy.get('#dropdown-gender').should(
      "be.visible")
  })

it(`Gender options is not empty and >= to ${qtyGenders}`, () => {
  cy.get('#dropdown-gender').click()
  cy.get(`.dropdown-menu > :nth-child(${qtyGenders})`).should(
      "be.visible")
  //close the dropdown
  cy.get('#dropdown-gender').click()

  })

  it("Language selection was rendered", () => {
    cy.get("#avaliableLanguages > .dropdown > #dropdown-languages").should(
      "be.visible")
  })

  it(`Language options is not empty and has at least ${qtyLanguages} options`,
      () => {
    cy.get("#dropdown-languages").click()
    cy.get(`.dropdown-menu > :nth-child(${qtyLanguages})`).should(
        "be.visible")
    //close the dropdown
    cy.get("#dropdown-languages").click()
  })

  it("InGameRole selection is there", () => {
    cy.get("#avaliableIngameRoles > .dropdown > #dropdown-ingamerole").should(
      "be.visible")
  })

it(`InGameRole options is not empty and has at least ${qtyInGameRoles}
    options`, () => {
    cy.get("#dropdown-ingamerole").click()
    cy.get(`#avaliableIngameRoles > .dropdown > .dropdown-menu >
        :nth-child(${qtyInGameRoles})`).should( "be.visible")
    //close the dropdown
    cy.get("#dropdown-ingamerole").click()
```

```
  })

 it("AboutMe was rendered", () => {
    cy.get('#aboutMe').should("be.visible")
  })

})
```

6.1 Die Testfälle für unser Projekt



Ansicht der laufenden Testfälle.



home.spec.js

Prüfen Sie, ob die Startseite „Home“ gerendert wurde.

HTTP-Request.spec.js

likeUsers.spec.js

Mit diesem Testfall wird überprüft, ob nach der Vergabe von einem „Like“ oder einem „Dislike“ ein anderer Nutzer angezeigt wird.

Testfall localStorage Token

Hier wird der Wert des JSON-Web-Tokens zu verschiedenen Zeitpunkten überprüft. Die Erwartung ist, dass das Token null ist, wenn der Benutzer nicht angemeldet ist. Dieses Token wird in localStorage gespeichert. Es besteht auch die Möglichkeit, dass das Token abgelaufen ist, wodurch alle Anfragen an den Server, die eine Authentifizierung erfordern, unmöglich werden.

profileElements.spec.js

Der Testfall prüft, ob die Elemente und Komponenten der Komponente Profil gerendert wurden und sichtbar sind. Diese Elemente sind userName, Gender und AboutMe. Die Komponenten sind Language und InGameRole.

Außerdem wird überprüft, ob Komponenten, ein Element „Dropdown“ enthalten, eine Mindestanzahl von Elementen enthalten, die angezeigt werden müssen.

signUp.spec.js

Dieser Testfall erstellt einen neuen Benutzer mit einer zufälligen E-Mail und einem zufälligen Benutzernamen.

Commands bei Cypress

In den Testfällen gibt es Aktionen, die sich immer wieder wiederholen, zum Beispiel die Anmeldung eines Nutzers. Zu diesem Zweck wurde in Cypress ein wiederverwendbarer Befehl definiert.

```
Cypress.Commands.add("typeLogin", (email, password) => {
    cy.visit("/login")

    cy.get("[id=email-input]").type(email).should("have.value", email)

    cy.get("[id=password-input]").type(password).should("have.value",
        password)
```

```
        cy.get("[id=btn-submit]").click()
    })
```

Mit anderen Worten, es handelt sich um eine Funktion, die zwei Parameter, E-Mail und Passwort, erhält. Beide Parameter werden in die entsprechenden Eingabefelder eingetragen, abschließend wird die Eingabetaste gedrückt. Diese Funktion kann in jedem anderen Testfall aufgerufen werden, ohne dass sie importiert werden muss.

7 Glossar

Hooks:

...

Framework:

...

JSX:
Es heißt JSX und ist eine Syntaxerweiterung für JavaScript. Wir empfehlen, sie mit React zu verwenden, um zu beschreiben, wie die Benutzeroberfläche aussehen soll. JSX erinnert vielleicht an eine Template-Sprache, aber es verfügt über die volle Leistungsfähigkeit von JavaScript. JSX erzeugt React-Ëlemente"

Over-Fetching:
Empfang von überschüssigen Daten durch eine Abfrage.

Web Token:
JSON-Web-Tokens sind eine dem Industriestandard RFC 7519 entsprechende Methode zur sicheren Darstellung von Forderungen zwischen zwei Parteien.

undefined:
Eine Variable, der kein Wert zugewiesen wurde oder die überhaupt nicht deklariert wurde (nicht deklariert, existiert nicht), ist undefiniert. Eine Methode oder Anweisung gibt auch undefiniert zurück, wenn der ausgewerteten Variablen kein Wert zugewiesen wurde. Eine Funktion gibt undefiniert zurück, wenn kein Wert zurückgegeben wurde.

FormData:
Die FormData-Schnittstelle bietet eine einfache Möglichkeit, eine Reihe von Schlüssel/Wert-Paaren zu erstellen, die die Felder eines Formulars und ihre Werte darstellen und mit der XMLHttpRequest.send()-Methode einfach gesendet werden können.

componentDidMount:
componentDidMount() wird unmittelbar nachdem eine Komponente (Einfügen in den Baum) montiert. Die Initialisierung, die DOM-Knoten erfordert, sollte hier erfolgen. Wenn Daten von einem Endpunkt geladen werden müssen, ist dies ein guter Ort, um die Netzwerkanfrage zu instanziieren. Diese Methode ist ein guter Ort, um Abonnements einzurichten. Wenn das der Fall ist, sollte es nicht vergessen werden, sich in componentWillUnmount() abzumelden.

componentDidUpdate:
componentDidUpdate() wird unmittelbar nach der Aktualisierung aufgerufen. Diese Methode wird beim ersten Rendering nicht aufgerufen.

componentWillUnmount:
componentWillUnmount() wird aufgerufen, unmittelbar bevor eine Komponente demontiert und zerstört wird. Man sollte in dieser Methode alle notwendigen Bereinigungen durchgeführen, wie z. B. das Ungültigmachen von Zeitgebern, das Abbrechen von Netzwerkanforderungen oder das Aufräumen von Abonnements, die in componentDidMount() erstellt wurden.

Destrukturierende Zuweisung:
Die destrukturierende Zuweisung ermöglicht es, Daten aus Arrays oder Objekten zu extrahieren, und zwar mit Hilfe einer Syntax, die der Konstruktion von Array- und Objekt-Literalen nachempfunden ist. https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment

Akronyme:
AWS
Amazon Web Services
DOM
Document Object Model
API
Application Programming Interface
AJAX
Asynchronous JavaScript And XML

8 Zusammenfassung und Ausblick

TEXT MUSS KONTROLIERT WERDEN...
Es hat sich gezeigt, dass moderne Entwicklungswerkzeuge für JavaScript auch ohne umfassende Kenntnisse der Softwareentwicklung zugänglich sind.

Das Ziel, eine echte Plattform zu schaffen, wurde im Zeitraum von Mai 2021 bis Mitte August 2021 erreicht.

Das heißt, ein Team von zwei Studenten mit grundlegenden Programmierkenntnissen war in der Lage, eine funktionelle Plattform zu schaffen, die die Registrierung, die Anmeldung der Benutzer, die Verwaltung von persönlichen Daten, die Interaktion mit anderen Benutzern auf der Grundlage ihrer Präferenzen umfasst und einen auf Textnachrichten basierenden Kommunikationskanal.

Inhalte der *Zusammenfassung und Ausblick*

Das Kapitel *Zusammenfassung und Ausblick* enthält folgende formale Aspekte[a]:

- Kapitelweise Kurzdarstellung der Inhalte (inklusive Referenzierung auf die Kapitelnummerierung) => Nach dem Motto: *Was wurde wo beschrieben?*
- Kurzdarstellung *Problem – Lösungsweg – Ergebnisse*
- Rückkopplung auf die Einleitung: Wurde die Zielstellung der Arbeit und die Fragestellung zufriedenstellend beantwortet?
- Kritische Bewertung (sofern nicht bereits im Hauptteil geschehen)
- Offene Probleme
- Richtung der zukünftigen/möglichen Arbeiten
- Erläuterung, warum welche Aspekte in der Arbeit nicht erläutert wurden

[a] Vgl. [BBoJ],S. 6

9 Quellenverzeichnis

9.1 Literatur

[SW11] Stickel-Wolf, Christine; Wolf, Joachim (2011): Wissenschaftliches Lernen und Lerntechniken. Erfolgreich studieren—gewusst wie!. Wiesbaden: Gabler.

9.2 Internetquellen

[BBoJ] Bertelsmeier, Birgit (o. J.): Tipps zum Schreiben einer Abschlussarbeit. Fachhochschule Köln-Campus Gummersbach, Institut für Informatik. http://lwibs01.gm.fh-koeln.de/blogs/bertelsmeier/files/2008/05/abschlussarbeitsbetreuung.pdf (29.10.2013).

[HR08] Halfmann, Marion; Rühmann, Hans (2008): Merkblatt zur Anfertigung von Projekt-, Bachelor-, Master- und Diplomarbeiten der Fakultät 10. Fachhochschule Köln-Campus Gummersbach.http://www.f10.fh-koeln.de/imperia/md/content/pdfs/studium/tipps/anleitungda270108.pdf (29.10.2013).

[V01] Offizielle Vue-Website: Vergleich zwischen Vue, React und Angular. https://vuejs.org/v2/guide/comparison.html#Preact-and-Other-React-Like-Libraries (unbekannte Veröffentlichung).

[R01] Offizielle React-Website: React Hooks. https://reactjs.org/docs/hooks-faq.html#which-versions-of-react-include-hooks

[R02] Offizielle React-Website: Fortgeschrittene Anleitugen, Context. https://de.reactjs.org/docs/context.html

[A01] Offizielle Website Apollo für React. https://www.apollographql.com/docs/react/

[SO01] StackOverFlow: Developer Survey 2021. https://insights.stackoverflow.com/survey/2021#section-most-popular-technologies-web-frameworks

[EE1] Elad Elrom: React and Libraries. https://link.springer.com/content/pdf/10.1007%2F978-1-4842-6696-0.pdf

[SS1] Stoyan Stefanov: Durchstarten mit React. https://content-select.com/media/moz_viewer/5d5fc360-478c-4038-ac17-246bb0dd2d03/language:de

[RH1] Red Hat: Was ist GraphQL? https://www.redhat.com/de/topics/api/what-is-graphql

[PM1] Postman: 2020 State of the API Report https://www.postman.com/state-of-api/the-future-of-apis/#the-future-of-apis

[AX1] Offizielle Website Axios https://axios-http.com/

A Anhang

A.1 Aufwandsverteilung

Hier zeigen wir, wie die Aufgaben unter den Autoren des Projekts verteilt wurden. TABELLE/Bild KOMMT...

A.2 ANHAND X

A.3 Verwendete Technologien und Werkzeuge

NodeJS
React
Bootstrap
GitHub
Cypress
VSC
Heroku
LaTeX
Cypress

Erklärung über die selbständige Abfassung der Arbeit

Ich versichere, die von mir vorgelegte Arbeit selbständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht.
Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

_____

(Ort, Datum, Unterschrift)

Hinweise zur obigen *Erklärung*

- Bitte verwenden Sie nur die Erklärung, die Ihnen Ihr **Prüfungsservice** vorgibt. Ansonsten könnte es passieren, dass Ihre Abschlussarbeit nicht angenommen wird. Fragen Sie im Zweifelsfalle bei Ihrem Prüfungsservice nach.
- Sie müssen **alle abzugebende Exemplare** Ihrer Abschlussarbeit unterzeichnen. Sonst wird die Abschlussarbeit nicht akzeptiert.
- Ein **Verstoß** gegen die unterzeichnete *Erklärung* kann u. a. die Aberkennung Ihres akademischen Titels zur Folge haben.