Technology
Arts Sciences
TH Köln

Technische Hochschule Köln
Fakultät für Informatik und Ingenieurwissenschaften

B A C H E L O R A R B E I T

Titel der Arbeit

Ggf. Untertitel

Vorgelegt an der TH Köln
Campus Gummersbach
im Studiengang
Wirtshaftsinformatik

ausgearbeitet von:
Timo Kalter
11127585
Carlo Menjivar
11117929

Erster Prüfer: <Name des 1. Prüfers>
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. [1], 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 Einleitung

TEXT FOLGT...

Die Einleitung

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. [1], S. 5-6
[b] [1], S. 6

2 Grundlagen

TEXT FOLGT...

2.1 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.

3 Projektziel

Unsere Aufgabe ist es, eine Webplattform für Spieler des Videospiels 'League of Legends' zu 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.

3.1 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 unsere Webseite durch intrinsische Motivation weiterverwendet. 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.

3.2 Das Nutzerkonto

Um die Webseite vollständig nutzen zu können, sollen Besucher der Seite ein Konto erstellen. Dazu sollen sie eine eMail 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.

3.3 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 User 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.

3.4 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.

3.5 Regeln und Richtlinien

Wir möchten mit unserer Plattform einen sicheren Ort bieten, auf dem die Spieler unabhängig ihrer Eigenschaften respektiert werden. vWir glauben, dass die meisten Nutzer in der Lage sind, respektvoll miteinander umzugehen, allerdings wird es auch Fälle geben, wo das nicht der Fall ist.

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.

3.6 Internationalisierung

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

3.7 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.

3.8 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.

3.9 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 eMail-Adressen dürfen nicht öffentlich einsehbar sein.

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

3.10 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.

3.11 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.

4 Datenbank

4.1 Einleitung

4.2 Ziel

Nutzer unserer Plattform erstellen sich ein Profil, auf dem sie etliche Daten angeben können. Diese Daten helfen anderen Nutzern bei der Kontaktsuche. Sobald sich 2 Nutzer gefunden haben, können sie miteinander schreiben und die Texte auch später noch lesen. Um all dies zu ermöglichen und verwalten zu können, benötigt es Infrastruktur in Form von einer Datenbank.

4.3 Wahl der Datenbank

In der näheren Auswahl der Datenbank standen die SQL-Datenbank PostgreSQL und MongoDB, eine NoSQL-Datenbank, zwei beliebte Datenbankmanagementsysteme mit unterschiedlichen Ansätzen. Wir werden uns im Folgenden die Gemeinsamkeiten und Unterschiede der Datenbanken anschauen und erklären, warum wir uns für MongoDB entschieden haben.

4.4 PostgreSQL

Michael Stonebreaker veröffentlichte 1974 in Berkeley unter BSD-Lizenz das RDBMS INGRES. Jahre später führte er das Projekt als Postgres (Post INGRES) weiter und fügte einen objektorientierten Ansatz hinzu. [12] Im Laufe der Jahre wurde PostgreSQL fortlaufend weiterentwickelt und hat viele neue Funktionen dazuerhalten. PostgreSQL beschreibt sich selbst als das fortschrittlichste, quelloffene, relationale Datenbankmanagementsystem und habe sich in über 30 Jahren aktiver Entwicklung einen guten Ruf für Zuverlässlichkeit, Funktionsrobusheit und Leistung verdient. [13] PostgreSQL wird auch in Zukunft unter einer quelloffenen Lizenz stehen, welche die kommerzielle Nutzung erlaubt. [14] PostgreSQL wurde exemplarisch als Option eines SQL-Systems verwendet, da der Funktionsumfang vergleichbar mit anderen beliebten SQL-Systemen ist. [15]

4.5 MongoDB

Die 2007 neu gegründete Firma 10Gen, mittlerweile bekannt als MongoDB Inc., benötigte eine Datenbank, welche den Anforderungen ihrer quelloffenen Plattform-as-a-Service Cloud-Architektur gerecht werden würde. Das Team suchte nach einer Datenbank, die elastisch, skalierbar, einfach zu verwalten und für Entwickler und Anwender einfach zu benutzen ist. Unzufrieden mit den auf dem Markt verfügbaren Datenbanksystemen wurde MongoDB, eine dokumentbasierte Datenbank entwickelt. Als das Team das Potenzial der

Datenbank realisierte, wurde die Idee der Cloud-Plattform eingestellt und die Entwicklung von MongoDB gefördert.[20] Laut eigenen Aussagen ist „MongoDB [] eine universelle, dokumentbasierte, verteilte Datenbank für die moderne Anwendungsentwicklung und die Cloud, die in puncto Produktivität höchsten Ansprüchen gerecht wird“. [21]

4.6 MongoDB und PostgreSQL im Vergleich

Kriterien	PostgreSQL
Datenbanktyp	SQL, Objektrelational
Ranking nach DB-Engines [29]	577,50 Punkte, viertbeliebteste Datenbank, viertbeliebteste rela
Architektur	
Monolithisch	
Dezentralisiert Erscheinungsjahr	1989
Datenschema	Ja
Referenzierung	Referenzierung per ID, erlaubt Foreign Keys
Datenstruktur	Tabellen bestehen aus Zeilen und Spalten
Typisierung	Erlaubt verschiedene Datentypen und nutzerdefinierte Da
Horizontale Skalierung	Repliken nach Master-Slave-System [17]
Konsistenzmodell	ACID, optional eventuelle Konsistenz [25] [26
CAP-Theorem	CA

[31] [32]

4.7 ACID

Auch wenn man NoSQL oft mit BASE verbindet, muss eine NoSQL Datenbank nicht zwangsläufig den Prinzipien von BASE folgen. MongoDB ist seit Erschaffung ACID-Konform auf Dokumentebene und unterstützt ab den Versionen 4.0 (Juni 2018 [27]) auf einem einzelnen Replik-Set bzw. ab Version 4.2 (August 2019 [27]) zwischen mehreren Sets multi-Dokument-Transaktionen mit ACID-Konformität. [25] Dies erlaubt Älles-oder-NichtsTransaktionen, bei denen es kritisch ist, dass entweder alle Teile oder kein Teil der Transaktion ausgeführt wird und löst damit Probleme, die historisch nur mit SQL-Datenbanken lösbar waren.

4.8 Referenzierung und Denormalisierung

PostgreSQL erlaubt es, andere Tabellen mit Foreign Keys (FK) gemäß SQL-Standard zu referenzieren. Der Wert eines FK muss korrekt sein, sollte der angegebene FK in der referenzierten Tabelle nicht existieren, wird ein Fehler geworfen. Auch ist es möglich, entsprechende Regeln zu definieren, was passieren soll, wenn der referenzierte Datensatz

beispielsweise gelöscht wird. MongoDB erlaubt auch, andere Dokumente per ID zu referenzieren, bietet jedoch keine direkte Option, direkt zu definieren, was beim löschen des referenzieren Datensatz passieren soll. Es ist weiterhin möglich, Trigger beim Löschen des referenzierten Datensatzes feuern zu lassen, dies ist im Vergleich zum Foreign Key Constraint jedoch aufwändig. Stattdessen setzt MongoDB darauf, alle relevanten Informationen in einem Dokument einzubetten, soweit möglich.

Denormalisierung und damit eingebettete Dokumente erlauben schnelleren Lesezugriff, da alle relevanten Informationen in einem Dokument sind und anders als in SQL nicht über mehrere Tabellen mit über Zeit immer länger werdenden JOIN-Anweisungen zusammengefasst werden müssen. Dies erleichtert Abfragen und erhöht die Geschwindigkeit von Leseabfragen. Der Nachteil ist, dass Daten über verschiedene Dokumente dupliziert werden und Schreibzugriffe langsamer werden - schließlich müssen Daten in verschiedenen Dokumenten erstellt oder aktualisiert werden. Wir glauben, dass viele Nutzer ihre Profilinformationen wie Avatarbild, Name oder Profiltext selten verändern. Jedes Mal, wenn das Profil eines Nutzers in der Kontaktsuche gezeigt werden soll, jedes Mal, wenn die Freundesliste oder ein Chat geöffnet wird, muss das Profil abgefragt werden - teilweise hunderte Male am Tag. Eine schnelle Antwort ist hier auch wichtig: Der Nutzer will nicht warten müssen, bis die Seite geladen hat. Auf der anderen Seite wird ein Nutzer vermutlich selten seine Profilinformationen ändern, höchstens ein paar Mal am Tag. Wir glauben auch, dass der Nutzer hier eher geneigt ist, eine kurze Verzögerung für die Aktualisierung seiner Daten hinzunehmen. Entsprechend sind wir der Meinung, dass die Vorteile der Denormalisierung dessen Nachteile überwiegen.

4.9 Skalierbarkeit

Sowohl PostgreSQL als auch MongoDB lassen sich vertikal skalieren - mit mehr Ressourcen läuft die Datenbank schneller. Bei horizontaler Skalierung verfolgen die beiden Datenbanken verschiedene Ansätze.

PostgreSQL nutzt ein Master-Slave-System bzw. Primary-Standby-System mit Load Balancing.[17] Die Standby-Knoten sind Kopien des Primärknotens und können Lesezugriffe verarbeiten. Schreibzugriffe werden nur vom Primärknoten angenommen. Sollte der Primärknoten versagen, bietet PostgreSQL keine automatische Lösung dieses Problems, ohne Drittsoftware muss manuell ein neuer Primärknoten gewählt werden. Es gibt nur einen Primärknoten, für Multi-Master-Systeme wird Drittsoftware benötigt. Selbst mit synchronen Repliken dauert es einen Moment, bis die Standby-Knoten die Aktualisierungen der Datenbank übernommen haben, dies kann dazu führen, dass Abfragen mit veralteten Datensätzen beantwortet werden.[19] Consistency nach CAP-Theorem ist für PostgreSQLs Master-Slave-Systeme somit nicht mehr einhundertprozentig gegeben.

Durch Sharding lässt sich die Datenbank in einzelne Knoten aufteilen, die jeweils nur einen Teil der Daten beinhalten. Dies erlaubt es, die Last und benötigte Speicherkapazität pro Knoten weiter zu verringern. Es ist möglich, mehrere Shards vom gleichen Knoten verwalten zu lassen; dies erleichtert die Skalierung, da die Anzahl der Shards flexibel an die technischen Ressourcen des verwaltenden Knoten angepasst werden kann.

MongoDB Sharding und Standbyknoten wurden erst im Laufe des Lebenszyklus von PostgreSQL entwickelt und mussten daher potenziell Suboptimale Lösungen verwenden, um die bereits bestehenden Systeme nicht zu zerstören. MongoDB got developed with scaling out in Mind.

4.10 Das CAP-Theorem

Brewers CAP-Theorem sagt aus, dass es in verteilten Systemen wie Datenbanken nicht möglich ist, gleichzeitig Consistency (Konsistenz), Availability (Verfügbarkeit) und Partition Tolerance (Partitionstoleranz) zu gewährleisten. Man muss sich für zwei entscheiden. Dies ist eine starke Simplifizierung, die den heutigen Datenbanken in der Form nicht gerecht wird und zwölf Jahre später von Brewer angepasst wurde. Viele aktuelle Datenbanken erlauben es, in unterschiedlichen Konfigurationen verschiedene Ziele zu erreichen.

Wir haben am Fallbeispiel von PostgreSQL zeigen können, dass im Multiknoten-System ein Teil der Konsistenz aufgegeben werden kann, um kürzere Antwortzeiten und Partitionstoleranz (Standby-Knoten dürfen ausfallen) zu gewährleisten. Die Einführung eines Multi-Master-Systems würde eine Erhöhung der Partitionstoleranz (beliebige Knoten dürfen ausfallen) zu Kosten von Konsistenz (gleichzeitige Schreibzugriffe auf verschiedene Knoten) erwirken und somit das System Richtung AP-Datenbank verschieben. Synchrone Replizierung auf Standby-Knoten erhöht die Konsistenz, erhöht aber gleichzeitig die benötigte Zeit einer Operation und verringert somit die Verfügbarkeit, schiebt also somit den Fokus Richtung CP-Datenbank. Diese Funktionen sind in anderen Datenbanken in ähnlicherweise zu finden und erlauben aktuellen Datenbanken, fein auf die Anforderungen des Benutzers angepasst zu werden.

Flexibilität

Gerade in der Anfangsphase von Projekten werden Datenbankstrukturen oft umgeworfen, angepasst und verworfen. Dokumente einer Collection müssen, anders als Spalten einer ORDBMS-Tabelle, nicht die gleiche Struktur aufweisen. Sollten also Änderungen an einem Dokuemtentyp vorgenommen werden, müssen vorherige Daten nicht angepasst und bereinigt werden. Dies erleichtert einen agilen Programmieransatz, kann aber die technischen Schulden des Projektes erhöhen.

BSON

MongoDB speichert Daten im "binary JSONFormat. JSON wiederum steht für 'JavaScript

Object Notation' und 'ist ein schlankes Datenaustauschformat, welches für Menschen einfach zu lesen und für Maschinen einfach zu parsen [] ist' [**?**]. JSON als semistrukturiertes Dateiformat eignet sich gut für Schnittstellendaten. Zudem wird im gewählten MERN-Techstack ausschließlich JavaScript verwendet - das JavaScript native Dateiformat JSON ist daher ohne Umwandlungen direkt verwendbar und der Umgang für unsere Entwickler bereits bekannt. Dies verringert die Gefahr möglicher Komplikationen und spart Lern- und Programmieraufwand.

Atlas

MongoDB Inc. bietet mit MongoDB Atlas eine Database-as-a-Service Lösung an, die flexibel auf die Größe und Auslastung des Projektes angepasst werden kann. Die Einrichtung des Datenbankservers scheint uns recht einfach zu sein und erlaubt uns, unseren Fokus darauf zu legen, das Projekt weiterzubringen. Sollte unser Projekt erfolgreich sein und in kurzer Zeit viele Nutzer generieren, ist es die Sache von ein paar Minuten, ein teureres Abonnement mit stärkerer Rechenleistung zu buchen.Für kleine Projekte und Projekte in der Entwicklungsphase ist das DaaS-Angebot zudem kostenlos. Dies verringert die Kosten in der Entwicklungsphase und verringert somit das Risiko einer Fehlinvestition.

Das Risiko, dass Code auf der lokalen Maschine funktioniert, aber auf der Produktionsumgebung Fehler wirft, wird durch die Verwendung von der gleichen Technologie (Atlas) in der Produktions- und Entwicklungsumgebung stark verringert. Die Dev-Prod-Vergleichbarkeit wird durch die geringere Werkzeuglücke erhöht. [34]

Beliebtheit

Mit X und Y gehört MongoDB zu einer der beliebtesten Datenbanken. Beliebte npm-Pakete wie "mongooseërleichtern die Verwendung von MongoDB weiter und das npm-Paket "graphql-compose-mongooseërmöglicht eine autogenerierte, funktionsfähige Schnittstelle durch GraphQL.

Erfahrung

Das Entwicklerteam hat in der Vergangenheit bereits Erfahrung in MongoDB sammeln können und ist gut mit der Datenbank zurecht gekommen.

Sowohl PostgreSQL als auch MongoDB eigenen sich sehr gut für unser Projekt. Uns ist bei der Recherche jedoch aufgefallen, dass MongoDB in unserem verwendeten Tech-Stack mit Express, React, und Node (MERN) deutlich mehr Verwendung findet als die Alternative mit PostgreSQL. Sollte

5 Nutzerkonto

Wenn ein Besucher ein Konto erstellt, müssen entsprechende Daten erstellt werden. Diese Daten werden in der Datenbank gespeichert, um auch in Zukunft drauf zugreifen zu können. Als Datenbank haben wir uns für MongoDB entschieden, da <GRÜNDE>.

Im Nutzerkonto-Dokument sind etliche Daten angegeben:

Feld	Öffentlich?	Beschreibung
id	ja	einzigartige ID, um den Nutzer bestimmen zu können. Automatisch generiert.
Name	ja	einzigartiger Name, mit dem der Nutzer anderen Nutzern angezeigt wird
normalisierter Name	nein	Name in Kleinbuchstaben. Wird verwendet, um die Einzigartigkeit von Namen zu gewährleisten
Email	nein	private eMail des Nutzers
Rolle	nein	Gibt an, ob der Nutzer autorisiert ist - Moderatoren und Administratorkonten haben mehr Rechte
Geburtsdatum	nein	privates Geburtsdatum des Nutzers
Alter	ja	öffentliches Alter des Nutzers
Sprachen	ja	Liste von Sprachen, die der Nutzer sprechen kann
Geschlecht	ja	gesellschaftliches Geschlecht des Nutzers.
Spielposition	ja	Position, die der Spieler in League of Legends am liebsten einnimmt. Der Nutzer kann seine zwei Lieblingspositionen angeben
Freitext	ja	kurzer Text, in dem der Nutzer sich beschreiben kann.
Avatar	ja	URI [GLOSSAR!] vom Avatarbild des Nutzers
Freunde	nein	Liste von allen Freunden des Nutzers. Beinhaltet die NutzerID und die ChatID
Geblockt	nein	Liste von NutzerIDs der Nutzer, die geblockt wurden

Alle Daten bis auf die ID und den Namen sind freiwillige Angaben. Wir möchten dem Nutzer die Entscheidung geben, welche Daten er uns und den anderen Nutzern preisgeben will.

Um die privaten Daten der Nutzer möglichst zu schützen, haben wir uns dazu entschieden, das Alter öffentlich zu halten, während das Geburtsdatum privat bleiben muss. Dies hat uns vor eine technische Herausforderung gestellt. Im ersten Versuch entschieden wir uns dafür, das Alter als virtuelles Feld zu definieren. Virtuelle Felder existieren nicht persistent auf der Datenbank, sondern werden erst bei Anfrage des Dokumentes berechnet. Dies hat leider den entscheidenden Nachteil, dass bei jeder Abfrage auch das Geburtsdatum in Erfahrung gebracht werden muss. Wir haben keine möglichkeit gefunden, in GraphQL zu definieren, dass das Geburtsdatum abgefragt werden soll, aber nicht

als öffentliches Feld bei der Abfrage zur verfügung steht. Ein weiterer großer Nachteil ist, dass, dadurch dass das Alter nicht persistent ist, nicht nach diesem gefiltert werden kann. Eine Abfrage wie ßeige mir alle Nutzer an, die zwischen X und Y Jahre alt sindïst nicht möglich, stattdessen müsste die Abfrage heißen: ßeige mir alle Nutzer an, die zwischen Datum 1 und Datum 2 geboren wurden". Damit wäre das Geburtsdatum wiederum nicht privat. Als zweites haben wir versucht, das Alter als Getter des Geburtsdatums zu definieren - wenn das Geburtsdatum erfragt wird, erhält die abfragende Partei stattdessen das berechnete Alter. Dies löst das Problem des öffentlichen Geburtsdatums zwar, lässt uns aber immer noch nicht nach dem Alter filtern. Im Endeffekt haben wir uns dafür entschieden, das Alter als persistentes Feld anzulegen. Das Alter wird durch einen CRON-Job jeden Tag für jeden Nutzer einmal neu berechnet und ist damit tagesaktuell. Durch das persistente Alters-Feld ist es möglich, nach diesem zu filtern. Die benötigte Rechenkapazität, um täglich das Alter zu erneuern, ist vernachlässigbar. Sollte das Geburtsdatum geändert werden, wird das Alter zudem neu berechnet.

Wenn man ein Profil bis zu ein Jahr lang verfolgt, ist es immer noch möglich, das Geburtsdatum anhand der Änderung des Alters am Geburtstag herauszubekommen, allerdings glauben wir, dass dieses Risiko zu gering ist, um ernsthaft Schaden anzurichten. Das Interesse der Nutzer, zu wissen, wie alt ihr gegenüber ist, überwiegt das Risiko, dass der Geburtstag von einzelnen Nutzern in Erfahrung gebracht wird.

<TODO> Geschlecht <muss weiter erklärt werden>

<TODO> Um weitere Statistiken anzeigen zu können, ist es dem Nutzer möglich, sein Konto via Riot-Schnittstelle mit seinen "League of LegendsKonto zu verbinden. Dies ermöglicht es, das Level, die Elo, die Lieblingshelden nd viele weitere Statistiken anzuzeigen.

6 Kontaktsuche

Unsere Plattform hat das Ziel, dass Spieler sich gegenseitig finden können, sich vernetzen können. Spielern werden dazu andere Spieler angezeigt und können entscheiden, ob sie mit diesen in Kontakt treten wollen.

Auf der Datenbank müssen diese Angaben entsprechend gespeichert werden. Wir nennen die dadurch erstellten Dokumente 'likes' bzw. 'dislikes'. Die Datenbankeinträge haben entsprechend folgende Felder:

Feld	Beschreibung
Anfragender	NutzerID des Nutzers, der die Entscheidung getroffen hat
Angefragter	NutzerID des Nutzers, über den die Entscheidung getroffen wurde
Status	Entscheidung, kann 'liked' oder 'disliked' sein.

Jedes Mal, wenn ein Like in die Datenbank eingetragen wird, wird geprüft, ob es bereits ein Gegenpaar gibt. Ist dies der Fall, werden die beiden Nutzer auf die Freundesliste des jeweils anderen hinzugefügt.

Beispiel: Alice hat Bob einen like gegeben. Bob gibt im Anschluss Alice einen like. Da es bereits einen Datensatz des Gegenpaar gibt (Alice mag Bob), sind die beiden nun befreundet.

Bei der Suche werden nur "neue"Nutzer angezeigt, das bedeutet, nur Nutzer, die vom aktiven Nutzer nicht bereits geliked, disliked oder geblockt wurden und sich auch nicht bereits in der Freundesliste befinden.

7 Benutzeroeberflaeche

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

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

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

Warum sollte man React berücksichtigen?

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.

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 des Frameworks in Bezug auf die neuesten Aktualisierungen und Änderungen oft spärlich.

Was sind React Hooks und wie kann man daraus profitieren?
Beginnend mit 16.8.0, enthält React eine stabile Implementierung von React Hooks.

- useState Das 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.
- useEffect
- useContext...

API Anfragen mit Rest, Axios und Apollo Client

ES FOLGT Was ist axios...

Über axios wurde eine Post-Anfrage bereitgestellt, die ermöglicht hat Bilder auf die S3 Speicher von AWS hochzuladen.

In der Benutzeroberfläche wurde ein Eingabefeld ïnput"verwendet mit type = "file"definiert. Um die Art der hochzuladenden Dateien einzuschränken, wurde eine Reihe zulässiger Dateiformate definiert, die an den Server gesendet werden dürfen.

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

8 Serveranfragen

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 mit Hilfe 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 einem 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.

[1]

[1] Vgl. u.a. [9]

Wie wurden die Anfragen in unserem Projekt verwaltet?
Alle Abfragen und Mutations wurden in einem Ordner gesammelt. Diese wurden für die spätere Verwendung in den React-Komponenten exportiert.

```
export const GET_MY_INFO = gql`
{
  userSelf {
    _id
    name
    aboutMe
    languages
    gender
    avatar
    ingameRole
    dateOfBirth
    friends { user chat }
    blocked
  }
}
`
//Beispiel einer Abfrage
```

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"

const { loading, error, data } = useQuery(
GET_MY_INFO,
ContextHeader(token),
)
```

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 undefined, null oder ungültig sein, wird der Server ein Fehler zurückgegeben.

Der useQuery Hook liefert ein Ergebnisobjekt, das Folgendes enthält.

loading:

Ein boolescher Wert, der angibt, ob die Abfrage in Bearbeitung ist. Wenn loading true 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 schief gelaufen ist.

date:

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

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. Er ist isomorphisch (= kann auf dem server und im browser verwendet weden). Auf der Server-Seite wir das modul http verwendet, während im Browser XMLHttpRequests (ajax) ausgeführt werden.

```
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
      })
  }
```

Das HTML-Eingabefeld ïnput"wurde folgendermaße 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 1Mb abgegrenzt. Durch die Eigenschaft „size“ der ausgewählten Datei konnten wir auf die Größe der Datei zugreifen.

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

9 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"

Under-Fetching:

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.

Akronyme:

AWSAmazon Web Services
DOMDocument Object Model
APIApplication Programming Interface

technische Schulden:
technical Dept...

RDBMS
ORDBMS
BSD-Lizenz
Eine Open Source Lizenz

ACID BASE CAP Sharding Load Balancing

10 Zusammenfassung und Ausblick

TEXT FOLGT...

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. [1],S. 6

11 Literaturverzeichnis

12 Quellenverzeichnis

13 Quellenverzeichnis

13.1 Literatur

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

13.2 Internetquellen

[1] 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).

[2] 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).

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

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

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

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

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

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

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

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

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

[12] Michael Stonebraker | Biography, MIT, Facts, & Turing Award | Britannica https://www.britannica.com/biography/Michael-Stonebraker#ref1245757

[13] PostgreSQL: The world's most advanced open source database https://www.postgresql.org/

[14] PostgreSQL: License https://www.postgresql.org/about/licence/

[15] Microsoft SQL Server vs. MySQL vs. Oracle vs. PostgreSQL Comparison https://db-engines.com/en/system/PostgreSQL%3BMicrosoft+SQL+Server%3BMySQL%3BOracle

[16] https://www.postgresql.org/docs/9.5/datatype.html

[17] [Literatur] Seite 51 Zeile 5-6 https://www.researchgate.net/profile/Ciprian-Octavian-Truica/publication/264416935_Asynchronous_Replication_in_Microsoft_SQL_Server_PostgreSQL_and_MySQL/links/53dbe6160cf216e4210c0375/Asynchronous-Replication-in-Microsoft-SQL-Server-PostgreSQL-and-MySQL.pdf

[18] https://cloud.google.com/community/tutorials/setting-up-postgres-hot-standby

[19] Multimaster - PostgreSQL wiki https://wiki.postgresql.org/wiki/Multimaster

[20] MongoDB daddy: My baby beats Google BigTable | The Register https://www.theregister.com/2011/05/25/the_once_and_future_mongodb/

[21] Die beliebteste Datenbank für moderne Apps | MongoDB https://www.mongodb.com/de-de

[22] BSON Types — MongoDB Manual https://docs.mongodb.com/manual/reference/bson-types/

[23] Sharding https://docs.mongodb.com/manual/sharding/

[24] Replizierung https://docs.mongodb.com/manual/replication/

[25] [Literatur] Multidocument ACID Transactions Seite 24 / Figure 6 https://webassets.mongodb.com/finalmongodb_multidocument_acid_trnasactions.pdf

[26] Häufige Fragen | MongoDB: Wie sorgt MongoDB für Konsistenz https://docs.mongodb.com/manual/core/read-isolation-consistency-recency/

[27] MongoDB Software Lifecycle Schedules https://www.mongodb.com/support-policy/lifecycles

[28] How to Scale MongoDB: What is vertical scaling in MongoDB? https://www.mongodb.com/basics/scaling

[29] https://db-engines.com/de/ranking_definition

[30] https://db-engines.com/de/ranking

[31] MongoDB vs PostgreSQL https://www.mongodb.com/compare/mongodb-postgresql

[32] MongoDB vs. PostgreSQL Vergleich https://db-engines.com/de/system/MongoDB%3BPostgreSQ

[33] PostgreSQL: Warum sollte man PostgreSQL verwenden? https://www.postgresql.org/about/

[34] The Twelve-Factor-App: X.Dev-Prod-Vergleichbarkeit https://12factor.net/de/

A Anhang

A.1 Unterabschnitt von Anhang

TEXT FOLGT...

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.