

DIPLOMARBEIT

WoSamma

Ausgeführt im Schuljahr 2025/26 von:

Jan Tiefenbacher	5BHITM-01
Laurenz Pichler	5BHITM-02

Betreuer:

Dipl.-Ing. (FH) Brandstetter Gerald
Dipl.-Ing. (FH) Brandstetter Gerald

Krems, am 03.04.2026

EIDESSTATTLICHE ERKLÄRUNG

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 und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Krems, am 03. April 2026

Verfasser/Verfasserinnen:

Jan Tiefenbacher

Laurenz Pichler

DIPLOMARBEIT

Bestätigung der Abgabe

Abgabebestätigung

Datum

Name

Unterschrift

Genehmigung der Diplomarbeit

Approbation

Datum

Prüfer*in

Abteilungsleiter*in
Direktor*in

DIPLOMARBEIT

Dokumentation

Verfasser*innen

Jan Tiefenbacher, 5BHITM

Laurenz Pichler, 5BHITM

Abteilung

Informationstechnologie

Ausbildungsschwerpunkt: Medientechnik

Schuljahr

2025/26

Thema der Diplomarbeit

Wosamma geoinformationsbasiertes Quizspiel

Kooperationspartner

Xinger Solutions GmbH

Aufgabenstellung

Ziel des Projekts WoSamma war die Entwicklung einer mobilen, geoinformationsbasierten Quiz-App, die sich auf Österreich fokussiert. Spieler sollen anhand von Street-View-Bildern erraten, an welchem Ort sie sich befinden, und dafür Punkte erhalten. Die App soll sowohl Lern- als auch Unterhaltungszwecke erfüllen und österreichisches Geographiewissen spielerisch vermitteln. Neben der Einzelspieler-Funktion war auch die Umsetzung von Mehrspielermodi, Ranglisten, Freundesystemen und einem Administrationsbereich geplant. Die Anwendung sollte plattformunabhängig lauffähig sein und insbesondere auf mobilen Endgeräten eine flüssige Nutzererfahrung bieten.

Realisierung

- **Programmiersprache / Framework:** React Native mit TypeScript
- **Backend & Datenbank:** Supabase (PostgreSQL, Auth, Storage)
- **Deployment:** Mit Apple Developer Konto auf dem Handy nutzbar
- **UI/UX-Design:** Figma (Mockups & Komponenten)
- **Versionsverwaltung:** Git & GitHub
- **Projektmanagement:** Jira (agiles Vorgehen, Sprints)
- **Besonderheiten bei der Entwicklung:**
 - Performance-Optimierung für mobile Geräte
 - API-Sicherheit & Authentifizierung
 - Plattformübergreifende Kompatibilität (iOS & Android)
 - Echtzeitfunktionen (Chat, Multiplayer)

Ergebnisse

Eine voll funktionsfähige Mobile-App mit vielfältigen Spielmodi, integriertem Freundesystem und Echtzeit-Chat für ein dynamisches und interaktives Spielerlebnis.

DIPLOMA THESIS

Documentation

Authors

Jan Tiefenbacher, 5BHITM

Laurenz Pichler, 5BHITM

Department

Informationstechnologie

Specialization: Medientechnik

Academic year

2025/26

Thesis Topic

WoSamma geo-information-based quiz game

Co-operation partners

Xinger Solutions GmbH

Task Description

The goal of the WoSamma project was to develop a mobile, geo-information-based quiz app focused on Austria. Players should guess their location based on Street View images and earn points for correct answers. The app is designed to be both educational and entertaining, helping users learn about Austrian geography in a playful way.

In addition to a single-player mode, the project also included the implementation of multiplayer modes, leaderboards, friend systems, and an administration area. The application was intended to run on multiple platforms and provide a smooth user experience, especially on mobile devices.

Implementation

- **Programming Language / Framework:** React Native with TypeScript
- **Backend & Database:** Supabase (PostgreSQL, Auth, Storage)
- **Deployment:** Usable on mobile devices via an Apple Developer account
- **UI/UX Design:** Figma (mockups & components)
- **Version Control:** Git & GitHub
- **Project Management:** Jira (agile workflow, sprints)
- **Special Aspects of Development:**
 - Performance optimization for mobile devices
 - API security & authentication
 - Cross-platform compatibility (iOS & Android)
 - Real-time features (chat, multiplayer)

Results

A fully functional mobile app featuring multiple game modes, an integrated friends system, and real-time chat for a dynamic and interactive player experience.

Inhaltsverzeichnis

1.	Präambel	10
1.1.	Zusammenfassung	10
1.2.	Abstract	10
1.3.	Team	10
1.4.	Danksagung	10
1.5.	Gendererklärung	11
2.	Einleitung	12
2.1.	Ausgangslage und Motivation der Arbeit	12
2.2.	Spezifische Ausgangslage (Muss noch erweitert werden)	12
2.3.	Spezifische Forschungsfrage	12
2.3.1.	Spezifische Forschungsfrage - Jan Tiefenbacher	12
2.3.2.	Spezifische Forschungsfrage - Laurenz Pichler	12
3.	Dokumentation der Implementierung	13
3.1.	Grundlagen der Implementierung	13
3.1.1.	Systemumgebung	13
3.1.2.	Verwendete Technologien	13
3.1.3.	Architekturüberblick	13
3.2.	Implementierung der Funktionen	13
3.2.1.	Login und Registrierung	13
3.2.2.	Benutzerverwaltung	14
3.2.3.	Einzelspieler-Modus	14
3.2.4.	Mehrspieler-Modus	14
3.2.5.	Bestenliste	14
3.2.6.	Tägliches Spiel	14
3.2.7.	Österreichweite Spielmodi	14
3.2.8.	Chat mit Freunden	15
3.2.9.	Freundesliste und Anfragen	15
3.2.10.	Profilbilder und Shop-System	15
3.3.	Datenhaltung	15
3.4.	Deployment	15
4.	Theoretische Grundlagen	16
4.1.	Jans Ausarbeitung	16
4.1.1.	Frontend-Technologien	16
4.1.2.	Trends & Weiterentwicklungen	16
4.1.3.	UI/UX Design	16
4.2.	Laurenz Ausarbeitung	16
4.2.1.	Mobile App Entwicklung	16
4.2.2.	Backend-Technologien	17
4.2.3.	Hosting- und Infrastrukturmodelle	19
4.2.4.	Datenmanagement	19
4.2.5.	Skalierbarkeit und Performanceoptimierung	19

4.2.6. Sicherheit und Authentifizierung	19
5. Zusammenfassung und Ausblick	20
5.1. Zusammenfassung	20
5.2. Ausblick	20
I. Literaturverzeichnis	21
II. Abbildungsverzeichnis	22
III. Tabellenverzeichnis	23
IV. Quellcodeverzeichnis	24
A. Anhang	25
A.1. Arbeitsteilung	25
A.2. Kapitelverzeichnis	25
A.3. Projekttagebücher	25
A.3.1. Projekttagebuch Max Mustermann	25
A.3.2. Projekttagebuch Mex Musterjuan	25
A.4. Besprechungsprotokolle	26
A.5. Besprechungsprotokolle 1	28
A.6. Datenträgerbeschreibung	29

1. Präambel

1.1. Zusammenfassung

Im Rahmen dieser Diplomarbeit wurde die mobile Anwendung WoSamma entwickelt – ein geoinformationsbasiertes Quizspiel, das es den Nutzern ermöglicht, Orte innerhalb Österreichs zu erkennen. Die App bietet verschiedene Spielmodi, darunter Einzelspieler-, Freundes- und Mehrspielervarianten. Zusätzlich wurden ein Freunds- und Chatsystem integriert, um den sozialen Aspekt des Spiels zu fördern.

- React Native App (mit TypeScript & Expo)
- Supabase (PostgreSQL, Authentifizierung, Storage)
- Eigene REST-API für Datenkommunikation

1.2. Abstract

As part of this diploma thesis, the mobile application WoSamma was developed – a geo-information-based quiz game that allows users to recognize and locate virtual places within Austria. The app combines playful learning with modern mapping technology and offers various game modes, including single-player, friends, and multiplayer modes. In addition, a friends and chat system was integrated to enhance the social aspect of the game.

- React Native App (with TypeScript & Expo)
- Supabase (PostgreSQL, Authentication, Storage)
- Custom REST API for data communication

1.3. Team

Das Projektteam besteht aus dem Projektleiter Jan Tiefenbacher und Kollegen Laurenz Pichler. Betreuer ist DI(FH) Brandstetter Gerald, der Auftraggeber ist Xinger Solutions GmbH.

1.4. Danksagung

Unser besonderer Dank gilt Dipl.-Ing. (FH) Gerald Brandstetter für seine engagierte und kompetente Betreuung im Rahmen dieser Diplomarbeit. Er unterstützte uns insbesondere im Bereich des UI-Designs mit wertvollen Anregungen und fachlicher Expertise. Auch im Backend-Bereich trug sein umfangreiches Know-how wesentlich zur erfolgreichen Umsetzung bei. Seine kontinuierliche Bereitschaft zur Unterstützung sowie seine ausführlichen und konstruktiven Rückmeldungen waren maßgebliche Faktoren für das sehr positive Endergebnis dieses Projekts.

Ebenso möchten wir dem Geschäftsführer der Xinger Solutions GmbH, Georg Kreuzinger, unseren Dank aussprechen. Seine Motivation und sein Interesse an der gemeinsamen Umsetzung dieses Projekts sowie seine langjährige Projekterfahrung waren von großem Wert, insbesondere bei der konzeptionellen Planung und strategischen Ausrichtung der Arbeit.

Darüber hinaus danken wir unseren Freunden und Familien für ihre Unterstützung während der gesamten Entwicklungsphase. Durch ihre Ermutigung, ihr Verständnis und ihre regelmäßigen Rückmeldungen haben sie wesentlich zur erfolgreichen Fertigstellung dieser Diplomarbeit beigetragen.

1.5. Gendererklärung

Zur besseren Lesbarkeit der Diplomarbeit wurde ausschließlich die männliche Form verwendet. Da Begriffe wie „Benutzerinnen und Benutzer“ den Text unleserlich machen, wurde es schlicht auf „Benutzer“ gekürzt, dies soll jedoch keine Geschlechterdiskriminierung zum Ausdruck bringen.

2. Einleitung

2.1. Ausgangslage und Motivation der Arbeit

Literaturrecherche, ähnliche Projekte, ...

2.2. Spezifische Ausgangslage (Muss noch erweitert werden)

Die zentrale Aufgabe unserer Diplomarbeit liegt darin, eine mobile Applikation zu entwickeln, welche auf mobilen iOS- sowie Android-Geräten lauffähig ist. Wir sahen uns zu Beginn unserer Arbeit mit einer Entscheidung konfrontiert. Es ging um die Fragestellung, welches Framework wir für die Realisierung unseres Projekts verwenden.

Für die inhaltliche Ausgestaltung der Applikation ließen wir uns von dem populären Spiel „GeoGuesser“ inspirieren. Obschon wir dieses Spielkonzept als sehr reizvoll empfanden, offenbarte sich rasch, dass es überaus herausfordernd ist, globale, zufällige Orte präzise zu identifizieren und zu erraten, insbesondere für Nutzer ohne geographisches Fachwissen. Wir konzipieren die Idee zu „WoSamma“, um dieses Prinzip weiterhin erfolgreich zu realisieren und es zugleich an unsere Projektgegebenheiten anzupassen.

2.3. Spezifische Forschungsfrage

2.3.1. Spezifische Forschungsfrage - Jan Tiefenbacher

2.3.2. Spezifische Forschungsfrage - Laurenz Pichler

Welche technischen und organisatorischen Skalierungsstrategien ermöglichen es, eine bestehende App-Infrastruktur kurzfristig und langfristig an stark wachsende Nutzerzahlen anzupassen?

3. Dokumentation der Implementierung

3.1. Grundlagen der Implementierung

3.1.1. Systemumgebung

Die entwickelte Anwendung ist ein mobiles Spiel für iOS, umgesetzt mit React Native und TypeScript. Für die Entwicklung wurde Visual Studio Code unter macOS verwendet. Die Anwendung kann über ein Apple-Developer-Konto direkt auf ein iPhone installiert werden und ist für eine zukünftige Veröffentlichung im Apple App Store vorgesehen.

3.1.2. Verwendete Technologien

- **React Native** – Framework zur Entwicklung plattformübergreifender mobiler Anwendungen.
- **TypeScript** – Typsichere Erweiterung von JavaScript zur Erhöhung der Codequalität.
- **Supabase (PostgreSQL)** – Backend-as-a-Service zur Bereitstellung einer REST-API, Authentifizierung und Datenbank.
- **Google Street View API** – Externe API zur Integration von Street-View-Bildern in das Spiel.
- **GitHub** – Versionsverwaltung und Projektmanagement.

3.1.3. Architekturüberblick

Die Anwendung nutzt eine klassische Client–Server-Struktur. Der Client (React Native App) kommuniziert über die Supabase-REST-API mit einer PostgreSQL-Datenbank. Zusätzlich werden externe Anfragen an die Google Street View API gesendet.

Administrator- und Benutzerrollen sind implementiert, wobei Administratoren erweiterte Berechtigungen besitzen (Freigeben von globalen Nachrichten, Einsehen von Meldungen).

3.2. Implementierung der Funktionen

3.2.1. Login und Registrierung

Die Authentifizierung erfolgt über Supabase Auth. Es werden Funktionen zur Registrierung, Anmeldung und Passwortverwaltung bereitgestellt. Alle Benutzerdaten werden sicher in der PostgreSQL-Datenbank gespeichert.

3.2.2. Benutzerverwaltung

Das System unterstützt zwei Rollen:

- **Benutzer** – reguläre Spieler.
- **Premium** – reguläre Spieler ohne Werbung.
- **Developer** – erweiterte Rechte, z. B. Verwaltung von Meldungen und Veröffentlichung globaler Nachrichten zur Testung der App.
- **Administrator** – erweiterte Rechte, z. B. Verwaltung von Meldungen und Veröffentlichung globaler Nachrichten.

Spieler können andere melden; Meldungen werden in Supabase gespeichert und durch Administratoren einsehbar.

3.2.3. Einzelspieler-Modus

Der Einzelspielermodus nutzt die Google Street View API zur Anzeige zufälliger Standorte. Der Spieler gibt eine Vermutung ab und erhält Punkte basierend auf der Distanz zum tatsächlichen Ort.

3.2.4. Mehrspieler-Modus

Spieler können eine Lobby erstellen, beitreten oder suchen. Die Kommunikation erfolgt in Echtzeit über Supabase Channels. Jeder Spieler sieht dasselbe Street-View-Bild; das System wertet anschließend die Ergebnisse aus und führt eine Rangliste.

3.2.5. Bestenliste

Die globalen Highscores werden in der Datenbank gespeichert. Ein Ranking wird dynamisch aus den Spielergebnissen generiert.

3.2.6. Tägliches Spiel

Jeden Tag steht allen Nutzern dasselbe Bild zur Verfügung. Die Ergebnisse werden global verglichen und in einer separaten Tabelle gespeichert.

3.2.7. Österreichweite Spielmodi

Es existieren zwei Varianten:

- **Ganz Österreich** – zufällige Orte im gesamten Bundesgebiet.
- **Spezifische Bundesländer** – Einschränkung auf einzelne Regionen.

3.2.8. Chat mit Freunden

Spieler können miteinander chatten. Nachrichten werden über Supabase Realtime synchronisiert und persistent gespeichert.

3.2.9. Freundesliste und Anfragen

Es besteht die Möglichkeit:

- nach Freunden zu suchen,
- Freundschaftsanfragen zu senden und zu akzeptieren,
- eine bestehende Liste von Freunden zu verwalten.

3.2.10. Profilbilder und Shop-System

Spieler können Profilbilder „kaufen“. Der Kaufvorgang ist derzeit simuliert und dient als Platzhalter für ein zukünftiges Zahlungssystem.

3.3. Datenhaltung

Die Datenbank basiert auf PostgreSQL mit Supabase als Service. Tabellen umfassen u. a.:

- Benutzer
- Freundesbeziehungen
- Chats und Nachrichten
- Meldungen
- Ergebnisse (Einzelspieler/Mehrspieler)
- tägliche Spiele
- globale Nachrichten

3.4. Deployment

Während der Entwicklungsphase wird die App lokal über ein Apple-Developer-Konto auf iOS-Geräten ausgeführt. Eine zukünftige Veröffentlichung im Apple App Store ist vorgesehen.

4. Theoretische Grundlagen

4.1. Jans Ausarbeitung

4.1.1. Frontend-Technologien

4.1.2. Trends & Weiterentwicklungen

4.1.3. UI/UX Design

4.2. Laurenz Ausarbeitung

4.2.1. Mobile App Entwicklung

4.2.1.1. Native vs. Cross-Platform Entwicklung

Quellenangabe: "<https://www.denovo.at/blog/native-vs-cross-plattform-app-welcher-weg-ist-der-richtige-fuer-ihre-idee>"

Es existieren verschiedene technische Herangehensweisen bei der Entwicklung mobiler Apps, um Anwendungen für mobile Endgeräte wie Smartphones und Tablets zu realisieren. Dabei gibt es einen wesentlichen Unterschied zwischen der Entwicklung nativer Apps und der Cross-Plattform-Entwicklung. Beide Optionen weisen Unterschiede hinsichtlich der Architektur, des Entwicklungsprozesses und der technischen Realisierung auf.

Unter nativer App-Entwicklung versteht man die Entwicklung und Umsetzung von Anwendungen, die speziell für ein bestimmtes Betriebssystem programmiert worden sind. Beispiele sind das für iOS-Anwendungen, die auf Swift basieren während Android-Applikationen typischerweise in Kotlin entwickelt werden. Die Anwendung wird direkt für dieses bestimmte System entwickelt und ist mit keinem anderen kompatibel.

Ein wesentliches Merkmal nativer Anwendungen ist der direkte Zugriff auf systemnahe Funktionen und Hardware-Komponenten des Endgeräts, wie Kamera, GPS oder Sensoren. Darüber hinaus orientieren sich native Apps stark an den Design- und Interaktionsrichtlinien des jeweiligen Betriebssystems, wodurch sie sich nahtlos in dessen Benutzeroberfläche einfügen.

Im Gegensatz dazu steht die Cross-Platform-App-Entwicklung, bei der Anwendungen mit einer gemeinsamen Codebasis für mehrere Betriebssysteme entwickelt werden. Ziel dieses Ansatzes ist es, den Entwicklungsaufwand zu reduzieren, indem große Teile des Codes plattformübergreifend wiederverwendet werden. Die Ausführung erfolgt anschließend auf verschiedenen Betriebssystemen, meist iOS und Android.

Cross-Platform-Frameworks abstrahieren die zugrunde liegenden Betriebssysteme und stellen einheitliche Schnittstellen zur Verfügung. Dadurch können Entwicklerinnen und Entwickler eine Anwendung erstellen, die auf mehreren Plattformen lauffähig ist, ohne für jedes Betriebssystem eine vollständig eigene Implementierung zu benötigen. Dennoch erfolgt die Interaktion mit gerätespezifischen Funktionen häufig über zusätzliche Abstraktionsschichten oder spezielle Erweiterungen.

4.2.1.2. React Native Framework

Quellenangabe: "<https://brainhub.eu/de/library/was-ist-react-native>"

React Native ist ein plattformübergreifendes Framework zur Entwicklung mobiler Anwendungen, das von Meta Platforms (ehemals Facebook) und der Open-Source-Community entwickelt wird. Es ermöglicht, Anwendungen für verschiedene mobile Betriebssysteme wie iOS und Android zu erstellen, indem es die Programmiersprache JavaScript in Kombination mit nativen UI-Elementen nutzt.

Im Gegensatz zu klassischen nativen Entwicklungsansätzen, bei denen separate Codebasen für unterschiedliche Betriebssysteme notwendig sind, verfolgt React Native den Ansatz einer gemeinsamen Codebasis. Dabei werden Benutzeroberflächen nicht als Webview-Elemente gerendert, sondern über native Komponenten dargestellt, sodass die resultierenden Anwendungen sich in Look and Feel deutlich näher an nativen Apps orientieren.

React Native basiert auf dem populären JavaScript-Framework React, das ursprünglich für die Entwicklung von Benutzeroberflächen im Web entwickelt wurde. Die Integration dieser Technologie ermöglicht es Entwicklern, UI-Komponenten modular zu strukturieren und wiederverwendbar zu machen, was die Wartung und Weiterentwicklung von Anwendungen erleichtert.

Ein technisches Charakteristikum von React Native ist der Einsatz der JavaScriptCore-Laufzeit sowie von Babel-Transpilern, die moderne JavaScript-Funktionen (z. B. ES6-Features wie Arrow-Funktionen oder `async/await`) unterstützen und zugleich die Kompatibilität mit unterschiedlichen Zielplattformen sicherstellen. Dadurch können Entwickler aktuelle Sprachstandards verwenden, ohne auf traditionelle plattformspezifische Sprachen wie Swift (für iOS) oder Kotlin/Java (für Android) angewiesen zu sein.

4.2.2. Backend-Technologien

4.2.2.1. APIs

Quellenangabe: "<https://www.talend.com/de/resources/was-ist-eine-api/>"

APIs (Application Programming Interfaces) sind Programmierschnittstellen, die es unterschiedlichen Softwaresystemen ermöglichen, miteinander zu kommunizieren. Sie definieren, wie Anfragen gestellt und wie Daten oder Funktionen zwischen Anwendungen ausgetauscht werden können, ohne dass die internen Abläufe der beteiligten Systeme bekannt sein müssen.

Eine API fungiert dabei als Vermittlungsschicht zwischen verschiedenen Softwarekomponenten. Sie legt fest, welche Funktionen oder Daten zur Verfügung stehen und in welcher Form diese genutzt werden dürfen. Durch diese klar definierten Schnittstellen wird eine strukturierte und kontrollierte Interaktion zwischen Frontend- und Backend-Systemen ermöglicht.

In der Softwareentwicklung bilden APIs eine wesentliche Grundlage für den Aufbau modularer Systeme. Sie unterstützen die Trennung von Zuständigkeiten zwischen einzelnen Systemkomponenten und tragen zur Wartbarkeit sowie Erweiterbarkeit von Anwendungen bei.

4.2.2.2. Datenbanken

Quellenangabe: "<https://www.ibm.com/think/topics/database>"

Datenbanken sind digitale Speichersysteme zur organisierten Verwaltung von Informationen. Sie dienen dazu, große Mengen strukturierter und unstrukturierter Daten so zu speichern, dass Anwendungen, Benutzer und Automatisierungsprozesse effizient darauf zugreifen, diese verwalten und aktualisieren können. Eine Datenbank besteht dabei aus einem Repository zur Speicherung der Daten sowie aus Software-Komponenten, die den Zugriff, die Strukturierung und die Sicherheit dieser Daten steuern.

Die grundlegende Funktion einer Datenbank besteht darin, Daten nicht nur passiv zu speichern, sondern sie in einer Form bereitzustellen, die schnelle Abfragen, konsistente Verwaltung und kontrollierten Zugriff ermöglicht. Datenbanken bilden damit eine essentielle Grundlage moderner Anwendungen, da sie die zentrale Grundlage für die Verwaltung und Bereitstellung von Daten in Informationssystemen darstellen.

Dabei umfasst der Begriff „Datenbank“ nicht nur die gespeicherten Daten selbst, sondern auch die zugehörige Infrastruktur, die physische Speicherung und die Software-Komponenten einschließt, die Datenbankoperationen steuern und ausführen. Durch diese Struktur ermöglichen Datenbanken eine systematische Organisation großer Datenbestände, wodurch diese für Anwendungen adressierbar und nutzbar werden.

Datenbanken werden in vielfältigen Kontexten verwendet und sind integraler Bestandteil zahlreicher Softwarelösungen, da sie die grundlegende Datenverwaltung für Anwendungen unterstützen. Ohne Datenbanken wäre die zentrale Verwaltung großer Informationsmengen sowie deren strukturierter Zugriff, wie er für Web-Anwendungen, mobile Anwendungen und Backend-Systeme heute notwendig ist, nicht realisierbar.

4.2.2.3. Serverless vs. klassisches Backend

4.2.3. Hosting- und Infrastrukturmodelle

4.2.3.1. On-Premise, Cloud und Hybrid-Architekturen

4.2.3.2. IaaS, PaaS, FaaS und Serverless

4.2.3.3. Containerisierung und Orchestrierung (Docker, Kubernetes)

4.2.4. Datenmanagement

4.2.4.1. Relationale Datenbanken vs. NoSQL

4.2.4.2. Datenmodellierung und Normalisierung

4.2.4.3. Echtzeit-Daten

4.2.5. Skalierbarkeit und Performanceoptimierung

4.2.5.1. Vertikale vs. horizontale Skalierung

4.2.5.2. Autoscaling und Lastverteilung

4.2.5.3. Latenzoptimierung und Durchsatzsteigerung

4.2.6. Sicherheit und Authentifizierung

4.2.6.1. Auth-Systeme

4.2.6.2. Verschlüsselung

4.2.6.3. Datenschutz

5. Zusammenfassung und Ausblick

5.1. Zusammenfassung

Zusammenfassend war diese Diplomarbeit ein sehr lehrreiches Projekt, bei dem wir viele neue Erfahrungen gemacht haben. ...

5.2. Ausblick

I. Literaturverzeichnis

- [1] abc: *DB-Engine Ranking*, März 2016. Online in Internet: URL: <http://db-engines.com/de/ranking>.

II. Abbildungsverzeichnis

III. Tabellenverzeichnis

A.1. Kapitelverzeichnis	25
A.2. Arbeitstagebuch Mustermann	25
A.3. Arbeitstagebuch Musterjuan	25

IV. Quellcodeverzeichnis

A. Anhang

A.1. Arbeitsteilung

Kurze Beschreibung, wer was gemacht hat (Überblick).

A.2. Kapitelverzeichnis

Kapitel	Editor
2.2 Spezifische Ausgangslage (Muss noch erweitert werden)	Max Mustermann
?? ??	Mex Musterjuan

Tabelle A.1.: Kapitelverzeichnis

A.3. Projekttagebücher

A.3.1. Projekttagebuch Max Mustermann

Tag	Zeit	kumulativ	Fortschritt
Mo 28.11.16	2h	2h	Besprechung der Programmanforderungen
Di 29.11.16	3h	5h	Datenbankmodell erstellt
Mi 30.11.16	1h	6h	Datenbankmodell überarbeitet
Do 01.12.16	3h	9h	Pflichtenheft erstellt

Tabelle A.2.: Arbeitstagebuch Mustermann

A.3.2. Projekttagebuch Mex Musterjuan

Tag	Zeit	kumulativ	Fortschritt
Mo 28.11.16	2h	2h	Besprechung der Programmanforderungen

Tabelle A.3.: Arbeitstagebuch Musterjuan

A.4. Besprechungsprotokolle

... Hier können auch pdf Dateien eingebunden werden!

Betreuungsprotokoll zur Diplomarbeit**Ifd. Nr.:**

Themenstellung:

Kandidaten/Kandidatinnen:

Jahrgang:

Betreuer/in:

Ort:

Datum:

Zeit:

Besprechungsinhalt:

Name	Notiz

Aufgaben:

Name	Notiz	zu erledigen bis

A.5. Besprechungsprotokolle 1

htl <small>bildung mit zukunft</small>	HTL Krems Höhere Lehranstalt für Informationstechnologie Ausbildungsschwerpunkt	Reife- und Diplomprüfung
--	--	-------------------------------------

Betreuungsprotokoll zur Diplomarbeit**Ifd. Nr.:**

Themenstellung:

Kandidaten/Kandidatinnen:

Jahrgang:

Betreuer/in:

Ort:

Datum:

Zeit:

Besprechungsinhalt:

Name	Notiz

Aufgaben:

Name	Notiz	zu erledigen bis

A.6. Datenträgerbeschreibung