

# **Nochba – App für Nachbarschaftshilfe**

## **DIPLOMARBEIT**

verfasst im Rahmen der

**Reife- und Diplomprüfung**

an der

**Höheren Abteilung für Informatik**

Eingereicht von:

Arsham Edalatkhah  
Sandin Habibovic  
Martin Hausleitner

Betreuer:

Rainer Stropek

Projektpartner:

IKT Linz GmbH - Open Common Linz

Leonding, März 2023

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

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

Die vorliegende Diplomarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Leonding, April 2022

S. Schwammal & S. Schwammal

# Abstract

Brief summary of our amazing work. In English. This is the only time we have to include a picture within the text. The picture should somehow represent your thesis. This is untypical for scientific work but required by the powers that are. Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.



# Zusammenfassung

Zusammenfassung unserer genialen Arbeit. Auf Deutsch. Das ist das einzige Mal, dass eine Grafik in den Textfluss eingebunden wird. Die gewählte Grafik soll irgendwie eure Arbeit repräsentieren. Das ist ungewöhnlich für eine wissenschaftliche Arbeit aber eine Anforderung der Obrigkeit. *Bitte auf keinen Fall mit der Zusammenfassung verwechseln, die den Abschluss der Arbeit bildet!* Suspendisse vel felis.

Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Aufgabenstellung . . . . .	1
1.3	Zusammenfassung des Projektergebnisse . . . . .	1
1.4	Projektvorgehensmodell . . . . .	1
<b>2</b>	<b>Technologien</b>	<b>2</b>
2.1	Technologieevaluierung . . . . .	2
2.2	Fazit . . . . .	3
<b>3</b>	<b>Systemarchitektur</b>	<b>5</b>
3.1	Flutter . . . . .	5
3.2	Firebase . . . . .	5
3.3	Algolia Search . . . . .	8
<b>4</b>	<b>Umsetzung</b>	<b>9</b>
4.1	Continuous Integration/Delivery . . . . .	9
4.2	Mobile Anwendung . . . . .	9
4.3	UI/UX Design . . . . .	13
4.4	Backend . . . . .	14
<b>5</b>	<b>Evaluating getroffener Entscheidungen</b>	<b>16</b>
<b>6</b>	<b>Zusammenfassung</b>	<b>17</b>
	<b>Literaturverzeichnis</b>	<b>VI</b>
	<b>Abbildungsverzeichnis</b>	<b>VII</b>
	<b>Tabellenverzeichnis</b>	<b>VIII</b>

**Quellcodeverzeichnis**

**IX**

**Anhang**

**X**

# **1 Einleitung**

## **1.1 Motivation**

## **1.2 Aufgabenstellung**

## **1.3 Zusammenfassung des Projektergebnisse**

## **1.4 Projektvorgehensmodell**

# 2 Technologien

## 2.1 Technologieevaluierung

Im Rahmen dieser Diplomarbeit wurden umfangreiche Anforderungen an das Frontend und Backend der Anwendung gestellt, um eine hochwertige und funktionale Anwendung zu entwickeln. Für das Frontend wurden iOS- und Android-Apps gefordert, wobei die Möglichkeit einer zukünftigen Web-App berücksichtigt werden sollte. Das Backend sollte eine Echtzeit-Datenbank für den Chat und eine Datenbank mit Geofunktionen für den Radios-Filter bereitstellen. Dabei war es wichtig, dass das Backend einfach zu implementieren ist und eine gute SDK-Unterstützung für das Frontend bietet.

Neben diesen technischen Anforderungen war auch die Umsetzung der Business-Logik ein wichtiger Aspekt. Hierbei wurden Aspekte wie die Registrierung und Verifizierung von Nutzern berücksichtigt, um eine sichere und benutzerfreundliche Anwendung zu gewährleisten. Darüber hinaus war es wichtig, dass die Anwendung mit neuen und effizienten Technologien entwickelt wird, um eine schnelle und agile Anpassung an die sich ändernden Anforderungen des Marktes und der Nutzer zu ermöglichen.

### 2.1.1 Frontend

Nach ausführlicher Recherche wurden die folgenden plattformübergreifenden Frameworks in Betracht gezogen[1]: Flutter, React Native, Ionic und Xamarin. Jede Plattform hat ihre Vor- und Nachteile.

Nach langer Recherche und Evaluierung aller Faktoren wurde Flutter als Plattform für das Frontend ausgewählt. Insbesondere die schnelle Entwicklung und das einfache Debugging sowie die Integration in VS Code waren entscheidende Faktoren. Jeder aus unserem Team hat eine eigene Flutter Test App programmiert [2] um die Entscheidung zu festigen.

Insgesamt waren wir mit unserer Entscheidung, Flutter zu verwenden, sehr zufrieden. Während der Entwicklungszeit stellte sich heraus, dass die Klammerverwendung in Flutter am Anfang etwas ungewohnt war, aber dank der hervorragenden Unterstützung



durch die Flutter-Community und der umfangreichen Dokumentation auf Stack Overflow konnten wir alle Herausforderungen bewältigen. Außerdem wurde die Entwicklung durch die ständigen Verbesserungen und Updates von Flutter, wie der Einführung der Impeller-Rendering-Engine, beschleunigt. [3]

Insgesamt war die Wahl von Flutter für das Frontend der App eine gute Entscheidung.

### 2.1.2 Backend

Für unser Projekt kamen drei Technologien in Frage: Supabase [4], Firebase[5] und Appwrite[6]. Sie alle bieten SDKs für Flutter und verfügen über eine benutzerfreundliche Authentifizierungsschicht. Darüber hinaus verfügen sie alle über eine Datenbank, die Geoqueries für das Hauptfeature des Beitragsradius unterstützt, sowie einen Speicherplatz für Beitragsfotos.

Allerdings bietet Supabase zum Zeitpunkt der Evaluierung im Februar 2022 noch keine Cloud-Funktionen wie Firebase und Appwrite, was es schwierig macht, eine robuste Business-Logik umzusetzen. Obwohl Supabase am 1. April 2022 experimentell Edge Functions eingeführt hat, ist dies für eine Produktions-App nicht empfehlenswert.

Daher wurden Appwrite und Firebase als die beiden besten Optionen identifiziert. Firebase bietet bessere Flutter-SDKs und verfügt über mehr Features, einschließlich der Unterstützung von Google für Flutter, weshalb wir uns schließlich für Firebase entschieden haben.

In der folgenden Tabelle sind die Vor- und Nachteile von Appwrite und Firebase aufgeführt:

Insgesamt bietet Firebase aufgrund der umfangreichen Features und der einfacheren Integration mit anderen Google-Tools Vorteile gegenüber Appwrite.

## 2.2 Fazit

Die Wahl von Flutter als Frontend-Plattform und Firebase als Backend-Plattform erwies sich als gute Entscheidung für die Anforderungen dieser Diplomarbeit. Die Einfachheit der Syntax, das schnelle Debugging und die umfangreiche Dokumentation und Unterstützung durch die Community machten Flutter zu einer einfachen und

effizienten Plattform für die Entwicklung der mobilen Apps. Firebase hat eine gute Lösung für die Echtzeit-Datenbank und die Geodatenbank und eine einfache Integration mit Flutter.

Insgesamt haben die gewählten Technologien dazu beigetragen, dass die Entwicklung der Anwendung schnell und effizient verlief und die Anforderungen der Diplomarbeit erfüllt wurden.

# 3 Systemarchitektur

sys diagram

## 3.1 Flutter

Hier kommt die Beschreibung der Technologie und deren Vorteile/Nachteile, auf Basis von wissenschaftlichen Studien und Erfahrungen aus der Praxis.

Quellen:

<https://docs.flutter.dev/resources/architectural-overview>

<https://flutter.dev/docs/resources/technical-overview>, <https://pub.dev/packages/flutter>

## 3.2 Firebase

Hier wird die Vorstellung der Firebase Technologie inklusive deren Funktionen und Vorteile erklärt.

Quellen:

<https://firebase.google.com/docs/guides>

<https://medium.com/flutter-community/flutter-firebase-from-scratch-28c8ba7d98b5>

### 3.2.1 Firebase Authentication

Beschreibung der Firebase Authentication Technologie, inklusive Sicherheitsregeln und Best Practices.

Quellen:

<https://firebase.google.com/docs/auth>

<https://medium.com/flutter-community/firebase-authentication-in-flutter-752d14209a8a>

### Security Rules

In unserer App ist die Sicherheit ein sehr wichtiger Faktor, um die Vertraulichkeit und Integrität der Daten und der Nutzer zu gewährleisten. Firebase bietet für seine Cloud-basierten Datenbank- und Speicherlösungen - Cloud Firestore und Cloud Storage - die sogenannten Security Rules[7][8][9], um den Zugriff auf die Daten und Ressourcen zu kontrollieren. Diese Regeln definieren, wer auf welche Art und Weise auf welche Daten zugreifen darf.

Die Security Rules für Cloud Firestore und Cloud Storage sind in einer eigenen Sprache geschrieben und werden serverseitig auf Firebase-Servern ausgeführt. Die Syntax basiert auf einer ähnlichen Struktur wie JSON und erlaubt komplexe Abfragen. Die Regeln können für eine bestimmte Sammlung oder einen bestimmten Pfad definiert werden und erlauben es, bestimmte Bedingungen für Lese- oder Schreibzugriffe zu definieren.

#### Listing 1: Security Rules Beispiel

```
1      match /posts/{postId} {  
2          allow read: if request.auth.uid != null;  
3          allow write: if request.auth.uid == resource.data.author;  
4      }
```

Diese Regel definiert, dass jeder Nutzer Lesezugriff auf alle Posts hat, aber nur der Autor des Posts ihn auch ändern darf.

Insgesamt bieten die Security Rules für Cloud Firestore und Cloud Storage eine leistungsstarke und flexible Möglichkeit, die Zugriffsrechte auf die Daten und Ressourcen in einer Firebase-App zu steuern. Mit Hilfe der Security Rules können wir sicherstellen, dass die Daten ihrer Nutzer geschützt und nur von berechtigten Personen abgerufen oder geändert werden können.

### 3.2.2 Cloud Firestore

Vorstellung der Datenbanktechnologie Firestore

Quellen:

<https://firebase.google.com/docs/firestore>

<https://medium.com/flutter-community/firebase-cloud-firestore-in-flutter-26c6e8c6f90c>

### Datenmodell

Hier kommt die Erklärung des Datenmodells in Firebase Firestore und dessen Auswirkungen auf die App-Architektur.

Quellen:

<https://firebase.google.com/docs/firestore/data-model>

<https://www.raywenderlich.com/6628345-cloud-firestore-for-flutter-getting-started>

Weitere wichtige punkte:

- Präsentation des eigenen Datenmodells in der Arbeit.
- Diagramme werden verwendet, um das Modell zu präsentieren und Entscheidungen zu erläutern und zu begründen.
- Anforderungen der App-Architektur werden dabei berücksichtigt werden.
- Performance, Skalierbarkeit und Strukturierung können thematisiert werden.
- Zur Veranschaulichung unserer eigenen Datenmodell-Entwicklung kann auf ein Beispiel-Datenmodell-Diagramm auf der offiziellen Firebase-Website verwiesen werden, welches uns als Orientierungshilfe diene.

Beispiel-Datenmodell-Diagramm Quelle:

[https://firebase.google.com/docs/firestore/data-model#structure\\_your\\_data](https://firebase.google.com/docs/firestore/data-model#structure_your_data)

### 3.2.3 Cloud Storage

Beschreibung der Cloud Storage Technologie in Firebase und deren Einsatz in der App.

Quellen:

<https://firebase.google.com/docs/storage>

<https://medium.com/flutter-community/firebase-cloud-storage-in-flutter-flutter-an-firebase-tutorial-c5de7835c6cd>

### 3.2.4 Firebase Cloud Functions

Erklärung der Cloud Functions Technologie in Firebase, inklusive Beispiele für deren Einsatz in der App.

Quellen:

<https://firebase.google.com/docs/functions>

<https://medium.com/codeburst/organizing-your-firebase-cloud-functions-67dc17b3b0da>

## 3.3 Algolia Search

Vorstellung der Algolia Search Technologie und deren Integration in die App-Architektur.

Algolia AI

Quellen:

<https://www.algolia.com/doc/>

<https://www.algolia.com/doc/guides/sending-and-managing-data/send-and-update-your-data/tutorials/firebase-algolia/>

### Firestore Cloud Function

Beschreibung der Firestore Cloud Functions und deren Rolle in der Algolia Integration.

Quellen:

<https://firebase.google.com/docs/functions>

<https://www.algolia.com/doc/guides/sending-and-managing-data/send-and-update-your-data/tutorials/firebase-algolia/>

# 4 Umsetzung

## 4.1 Continuous Integration/Delivery

### 4.1.1 GitHub Actions

allg. actions warum kosten

**Build IOS**

**Build Android**

### 4.1.2 Fastlane

**Build Number increment**

### 4.1.3 Firebase App Distribution

## 4.2 Mobile Anwendung

### 4.2.1 Dateistruktur

datei benennung folder structure

### 4.2.2 State Management

entscheidung vergleich warum getx

**GetX**

### 4.2.3 Authentifizierung

**Anmelde Flow**

Diagram erklärung screenshots

### Registrierungs Flow

Diagram erklärung screenshots

### Firebase Authentifizierung

allg.

#### 4.2.4 Feed

foto aufbau

#### 4.2.5 Beiträge

Beiträge sind das Hauptkommunikationsmittel auf der App. Jedem Beitrag muss ein Titel, eine Beschreibung und eine Reichweite, unter der, der Beitrag sichtbar ist, angegeben werden. Weiters ist es möglich einem Beitrag ein Bild und Tags anzuhängen.

### Kategorien

Um Beiträge besser zuordnen zu können, muss der User den Beitrag vor dem Veröffentlichen in eine bestimmte Kategorie einteilen. Diese Kategorien ermöglichen es Usern, die Art Ihrer Anfrage ihm vorhinein besser zu spezifizieren und die Suche nach Beiträgen einer bestimmten Art zu vereinfachen. Bestimmte Kategorien werden weiters in Unterkategorien aufgeteilt, da diese ein zu weit gefächertes Genre an Anfragen umfassen.

Es existieren folgende Kategorien bzw. Unterkategorien:

- Mitteilung
  - Frage
  - Appell
  - Warnung
  - Empfehlung
  - Gefunden
- Suche
  - Hilfe
  - Verloren
- Ausleihen



- Event

Mitteilung: Die Kategorie der Mitteilung dient dazu die Nachbarn über ein bestimmtes Ereignis oder Meldung zu informieren oder zu befragen.

Suche: Die Kategorie der Suche dient dazu mit den Nachbarn im Falle einer Hilfesuche oder eines verloren gegangenen Objekts in Kontakt zu treten.

Ausleihen: Die Kategorie des Ausleihens dient dazu die Nachbarn nach der Erlaubnis, sich ein bestimmtes Werkzeug oder Objekt ausborgen zu dürfen, zu bitten.

Event: Die Kategorie des Events dient dazu die Nachbarn auf eine bestimmte Veranstaltung aufmerksam zu machen.

### **Tags**

Als Tag wird ein Schlüsselwort beschrieben, was man an ein Informationsgut anhängen kann, um es besser beschreiben zu können und/oder besser auffindbar zu machen. In der App werden Tags als eine Erweiterung der Kategorien verwendet, um es Usern zu ermöglichen Ihren Beitrag einem selbstdefinierten Typ zuzuordnen.

### **Info**

Jeder Beitrag hat eine eigene Sektion, wo wichtige Entscheidungsinformationen angegeben werden, wie der Stadtteil und die ungefähre Entfernung zum gegebenen Nachbarn und das Erstelldatum des Beitrags.

### **Kommentare**

Die Kommentarfunktion ermöglicht es den Usern unter einem Beitrag Ihre Meinung, Feedback oder sonstiges zu hinterlassen.

### **Beitrag oder Kommentar Melden**

Um auf unangebrachte Beiträge oder Kommentare schnell reagieren zu können, gibt es die Möglichkeit Beiträge oder Kommentare zu melden. Diese Meldungen werden auf Firestore gespeichert und können dann im Einzelnen überprüft werden. Fürs Melden muss ein Grund ausgewählt und eine genauere Beschreibung angegeben werden. Gründe fürs Melden eines Beitrags oder Kommentars:

- Unangebrachter Inhalt
- Belästigung
- Betrug
- Spam
- Sonstiges

### 4.2.6 Filter

Filtern nach Kategorien Ordnen nach Likes oder Datum Absteigend oder aufsteigend sortieren

### 4.2.7 Suche

### 4.2.8 Typesense

### 4.2.9 Algolia

diagram

### Firestore Sync

### Algolia SDK

### 4.2.10 Chat

package genommen warum

### Flyer Package

### 4.2.11 Profil

Was wird angezeigt? Name, User Public Info, Posts von User

### Profil Melden

### 4.2.12 Benachrichtigungen

Grund und Funktionsweise von Benachrichtigungen diagram beschreibung

### 4.2.13 Einstellungen

Email/Passwort ändern, Konto löschen, Sprache einstellen, Benachrichtigungen ein/aus-schalten

### 4.2.14 Feedback

feedback feature beschreiben

## 4.3 UI/UX Design

### 4.3.1 Inspiration

was waren meine vorbilder: nebenan dribbble twitter...

### 4.3.2 Prototyping

#### Framer

fotos von prototyp erklärung warum framer

#### Adobe XD

fotos von prototyp beschreibung warum adobe xd

### 4.3.3 Design

design patterns

#### Farben

farben history warum orange

#### Icons

foto von den benutzen icons welche icons warum

### **Fonts**

fotos von fonts welche fonts wie ist die typographie aufgebaut

### **Logo**

logo foto anforderungen design ideen logo history

### **4.3.4 App Design**

eingehen mit fotos und beschreibung auf die haupt screens der app

### **4.3.5 Websites Design**

2-3 nochba.at design foto beschreibung

## **4.4 Backend**

### **4.4.1 Cloud**

### **4.4.2 Firebase**

#### **Firebase SDKs**

Firebase-Tools

#### **Firebase Authentication**

Email und Passwort authentication

#### **Cloud Firestore**

NoSQL-Database, Dokument-basierte Speicherung, Subcollections

#### **Cloud Storage**

Speicherung von Profilbilder und Post-Bilder

### **Firebase Cloud Functions**

jeder schreibt über seine CF

### **4.4.3 Algolia Search**

### **4.4.4 Algolia SDK**

**Typesense Search**

### **4.4.5 Typesense SDK**

## **5 Evaluating getroffener Entscheidungen**

# 6 Zusammenfassung

Aufzählungen:

- Itemize Level 1
  - Itemize Level 2
    - Itemize Level 3 (vermeiden)
- 1. Enumerate Level 1
  - a. Enumerate Level 2
    - i. Enumerate Level 3 (vermeiden)

**Desc** Level 1

**Desc** Level 2 (vermeiden)

**Desc** Level 3 (vermeiden)





# Literaturverzeichnis

- [1] S. Srivastava, „Top 10 Best Cross-Platform App Development Frameworks,” <https://appinventiv.com/blog/cross-platform-app-frameworks/>, February 2023.
- [2] M. Hausleitner, „Creating Test Apps with Flutter and Firebase,” <https://github.com/Martin-Hausleitner/flutter-firebase-test-01>, 2022.
- [3] Flutter, „Impeller,” <https://github.com/flutter/flutter/wiki/Impeller>, 2022.
- [4] „Supabase,” <https://supabase.io/>.
- [5] „Firebase,” <https://firebase.google.com/>.
- [6] „Appwrite,” <https://appwrite.io/>.
- [7] Firebase. (2021) Security Rules | Firebase. Online verfügbar: <https://firebase.google.com/docs/rules>
- [8] M. Hausleitner. (2021) Firestore Security Rules Nochba. Online verfügbar: <https://github.com/Martin-Hausleitner/Nochba/blob/main/firestore.rules>
- [9] ——. (2021) Storage Security Rules Nochba. Online verfügbar: <https://github.com/Martin-Hausleitner/Nochba/blob/main/storage.rules>

# Abbildungsverzeichnis

# Tabellenverzeichnis

# Quellcodeverzeichnis

1	Security Rules Beispiel . . . . .	6
---	-----------------------------------	---

# Anhang