

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

| | | |
|----------|---|-------------|
| 1 | Einleitung | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Aufgabenstellung | 1 |
| 1.3 | Zusammenfassung des Projektergebnisse | 1 |
| 1.4 | Projektvorgehensmodell | 1 |
| 2 | Technologien | 2 |
| 2.1 | Technologieevaluierung | 2 |
| 3 | Systemarchitektur | 4 |
| 3.1 | Flutter | 4 |
| 3.2 | Firebase | 4 |
| 3.3 | Algolia Search | 4 |
| 4 | Umsetzung | 5 |
| 4.1 | Continuous Integration/Delivery | 5 |
| 4.2 | Mobile Anwendung | 5 |
| 5 | Zusammenfassung | 6 |
| | Literaturverzeichnis | V |
| | Abbildungsverzeichnis | VI |
| | Tabellenverzeichnis | VII |
| | Quellcodeverzeichnis | VIII |
| | Anhang | IX |

1 Einleitung

1.1 Motivation

1.2 Aufgabenstellung

1.3 Zusammenfassung des Projektergebnisse

1.4 Projektvorgehensmodell

2 Technologien

2.1 Technologieevaluierung

Im Zuge meiner Diplomarbeit musste ich mich entscheiden, welche Technologien wir für die Entwicklung einer mobilen Anwendung verwenden werden, sowohl für das Frontend als auch das Backend.

2.1.1 Frontend

Zu Beginn hatten wir keine Erfahrung in der Entwicklung von mobilen Apps und nur begrenzte Erfahrung in JavaScript, jedoch viel Erfahrung in C-Sharp aufgrund unserer Ausbildung an der HTL. Nach langer Recherche kamen folgende plattformübergreifende Technologien in Frage: Flutter, React Native und Xamarin.

Flutter wurde aufgrund der schöneren Syntax, die an C-Sharp erinnert, und der Einfachheit der Entwicklung ausgewählt. Weitere Gründe waren die Hot-Reload-Funktion, die einfache Integration von Drittanbieterpaketen, das gute Debugging und die schnelle Entwicklung von Prototypen. Jeder von uns hat einen Prototypen mit Flutter erstellt.

Quellen wie [1] haben gezeigt, dass Flutter im Vergleich zu React Native und Xamarin in Bezug auf die Entwicklungszeit, Leistung und App-Größe am besten abschneidet.

Während der Entwicklung wurden wir von der sehr aktiven Flutter-Community unterstützt und fanden zu jedem Problem, das wir hatten, eine Lösung auf Stack Overflow oder in GitHub-Issues. Obwohl die Klammerverwendung am Anfang etwas gewöhnungsbedürftig war, hat uns die Flutter-Erweiterung in VS Code [2] mit Funktionen wie Quick Refactoring und der einfachen Organisation von Widgets in eigene Klassen sehr geholfen.

Wir haben mit Flutter 2.10 begonnen und arbeiten jetzt nach etwas mehr als einem Jahr mit Flutter 3.7. In dieser Zeit hat sich viel verändert, einschließlich der Einführung einer schnelleren Rendering-Engine [3], die die Entwicklung beschleunigt hat.

2.1.2 Backend

Unsere Anforderungen an das Backend waren die einfache Integration mit dem Flutter-SDK, Schnelligkeit, eine Realtime-Datenbank für Chats und Geofencing für das Radios-Feature.

Firebase wurde als Backend-Plattform ausgewählt, da es sich als zuverlässig und einfach zu integrieren erwies. Es bietet eine Realtime-Datenbank und eine einfache Authentifizierung. Die Integration mit Flutter war ebenfalls sehr einfach, da Firebase eine umfangreiche Dokumentation und eine sehr aktive Community hat. Wir konnten das Backend schnell entwickeln und es bot alle Funktionen, die wir benötigten.

Quellen:

1. M. Kucukkaldirim, S. Zengin, and H. Yalcin, "Performance Comparison of Flutter and React Native for Cross-Platform Mobile Development," 2020 International Artificial Intelligence and Data Processing Symposium (IDAP), 2020, pp. 1-6, doi: 10.1109/IDAP49677.2020.9349785.
2. Visual Studio Marketplace, "Flutter," <https://marketplace.visualstudio.com/items?itemName=Code.flutter>, abgerufen am 12. Februar 2023.
3. Flutter, "Impeller," <https://github.com/flutter/flutter/wiki/Impeller>, abgerufen am 12. Februar 2023.

3 Systemarchitektur

3.1 Flutter

3.2 Firebase

3.2.1 Firebase Authentication

Security Rules

3.2.2 Cloud Firestore

Datenmodel

3.2.3 Cloud Storage

3.2.4 Firebase Cloud Functions

3.3 Algolia Search

Firebase Cloud Function

4 Umsetzung

4.1 Continuous Integration/Delivery

4.1.1 GitHub Actions

4.1.2 Firebase App Distribution

4.1.3 Fastlane

4.2 Mobile Anwendung

5 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

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

Anhang