

Gymnasium Nepomucenum Coesfeld
Besondere Lernleistung im Abitur 2022

DMergency - App zur Alarmierung und Verwaltung von Sanitätsdiensten

Entwicklung einer App zur effizienten und benutzerfreundlichen
Alarmierung, sowie Verwaltung von Sanitätsdiensten

vorgelegt von

Mattis Rinke

Fachbereich Informatik

Herr Brumma
Herr Willenbring

Inhaltsverzeichnis

1	Einleitung	3
2	Wahl der Entwicklungsweise	3
2.1	Version 1 - Native Entwicklung	3
2.2	Version 2 - Xamarin Forms	3
2.3	Version 3 - Flutter	3
3	Das Projekt	3
3.1	Zielsetzung	3
3.2	Abgrenzung zur Server Ausarbeitung	4
3.3	Funktionen der App	4
3.3.1	Rollen und Registrierung	4
3.3.2	Alarmauslösung	4
3.3.3	Alarmempfang	4
3.3.4	Vertretungen	4
3.3.5	News	4
3.3.6	Notfallnummern	4
4	Kommunikation mit dem Server	4
4.1	API-Nutzung	4
4.2	Nutzung von Firebase-Messaging	4
5	Speicherung der Daten	4
5.1	Umsetzung	4
5.2	Aufbau der Datenbank	5
6	Design Umsetzung	5
7	Schultests	5
8	Fazit	5
8.1	Was wurde erreicht	5
8.2	Wie geht es weiter	5
9	Abbildungsverzeichnis	6
10	Literaturverzeichnis	6
11	Selbstständigkeitserklärung	6
12	Anhang	6

1 Einleitung

In der heutigen Zeit wird der Drang nach Digitalisierung immer stärker. So auch in Sanitätsdiensten, die darauf angewiesen sind schnell und effizient alarmiert zu werden. Diesen Drang habe ich selbst in meiner Funktion als Schulsanitätsdienst-Leiter und als Mitglied beim DRK mitbekommen. Da ich bereits das ein oder andere kleinere Projekt selbst programmiert hatte, habe ich mir überlegt selber eine App zum Alarmieren und Verwalten von Sanitätsdiensten zu programmieren, was sich am Ende als deutlich schwieriger und komplexer rausstellte als am Anfang erwartet.

2 Wahl der Entwicklungsweise

2.1 Version 1 - Native Entwicklung

2.2 Version 2 - Xamarin Forms

2.3 Version 3 - Flutter

Erklärung warum Flutter genutzt wird

3 Das Projekt

3.1 Zielsetzung

Ziel des Projekts ist das effiziente und sinnvolle Alarmieren, sowie Verwalten von Sanitätsdiensten. Hierbei soll zum einen die Nutzerfreundlichkeit, als auch die Effizienz und Kompatibilität im Vordergrund stehen.

Hierbei geht es grundlegend um die Funktionen der Alarmierung, der Dienstregelungen (Wann hat welcher Sanitäter Dienst) und den Empfang des Alarms. Dabei soll die Alarmierung, so intuitiv wie möglich und komplex wie nötig gestaltet werden um den Sanitätern so viele Informationen wie möglich zu geben, den Alarmierungsprozess für den Alarmierenden jedoch nicht unnötig kompliziert zu gestalten. Dies ist vor allem unter dem Punkt zu sehen, dass größtenteils von Laien alarmiert werden soll, da sich das Projekt auf kleine Sanitätsdienste bzw. Schulsanitätsdienste beschränkt.

Um dies Umzusetzen soll eine App programmiert werden, die sich schnell öffnen lässt und von den Alarmierenden schnell genutzt werden kann um die nötigsten Informationen einzugeben und dann schnell zu Alarmieren. Des weiteren soll es in der gleichen App möglich sein die gesendeten Alarme zu empfangen und dann eine Rückmeldung an den betroffenen Alarmierenden zu senden. Damit der Sanitäter nicht dauerhaft im Dienst ist soll es außerdem einen Dienstplan geben, aus dem man sich im Notfall austragen kann oder aber auch den Dienst eines anderen Sanitäters zu übernehmen, falls dieser verhindert ist.

3.2 Abgrenzung zur Server Ausarbeitung

In dieser Ausarbeitung geht es um die App "Dmergency". Nicht um die Webanwendung, bzw. den Server. Es wird jedoch auf die Kommunikation mit dem Server eingegangen um den Datenfluss darzustellen und die Funktionsweise der App zu verdeutlichen.

3.3 Funktionen der App

In den folgenden Abschnitten werden jetzt die Funktionen der App dargestellt und erklärt. Dazu werden beispielhaft einzelne Methodenimplementationen herausgenommen, erörtert und im Kontext der jeweiligen Funktion erklärt.

3.3.1 Rollen und Registrierung

3.3.2 Alarmauslösung

3.3.3 Alarmempfang

3.3.4 Vertretungen

3.3.5 News

3.3.6 Notfallnummern

4 Kommunikation mit dem Server

4.1 API-Nutzung

4.2 Nutzung von Firebase-Messaging

5 Speicherung der Daten

5.1 Umsetzung

Es gibt mehrere Möglichkeiten auf mobilen Endgeräten appspezifische Daten zu speichern. Zum einen gibt es die so genannten SharedPreferences bzw. UserDefaults¹ dies sind einfache Schlüssel, mit denen ein Wert verknüpft wird. Eine weitere Möglichkeit ist eine lokale Datei, in welche alle wichtigen Daten geschrieben werden, oder als letzte, dritte Möglichkeit gibt es die Datenbank.

Im Fall der App Dmergency habe ich zunächst versucht die Platform-Nativen Speichermethoden, also die SharedPreferences bzw. die UserDefaults, zu nutzen. Dies habe ich gemacht, da ich in der App selbst eigentlich kaum Daten speichern muss, was im Laufe der Ausarbeitung noch deutlich wird.

¹SharedPreferences (Android), bzw. UserDefaults(iOS) ist plattformspezifischer Langzeitspeicher für einfache Daten (String, Integer)

Da die Platform-Nativen Speichermethoden so einfach gehalten sind und in der App nur Strings und Integer gespeichert werden müssen bieten sich diese also sehr gut an.

Dieser Gedanke musste jedoch schnell verworfen werden, da die Platform-Nativen Speichermethoden unterschiedliche Instanzen in unterschiedlichen Threads haben.

Dies wirft zwei Probleme auf:

1. Die Effizienz der App wird stark beeinträchtigt
2. Daten können in unterschiedlichen Threads nicht mit den Platform-Nativen Speichermethoden abgerufen werden.

Um Alarme zu empfangen muss ein Hintergrundprozess laufen, welcher in einem anderen Thread arbeitet als der Rest der App. Da jedoch auf die Alarmdaten auch im Rest der App zugegriffen werden muss und auch im Hintergrundprozess auf die gespeicherten Nutzerdaten zurückgegriffen werden muss, sind die plattformspezifischen Speichermethoden für die App nicht in Frage gekommen.

Daher habe ich mich dazu entschieden eine Datenbank anzulegen, welche als lokale Datei auf dem Handy abgelegt ist. Dadurch ist es möglich jederzeit auf alle Daten zuzugreifen und die Probleme, der Effizienz und der Speicherung der Daten sind behoben.

5.2 Aufbau der Datenbank

6 Design Umsetzung

7 Schultests

8 Fazit

8.1 Was wurde erreicht

8.2 Wie geht es weiter

[1]

9 Abbildungsverzeichnis

10 Literaturverzeichnis

Literatur

- [1] Carbogning et. al. Global change and relative sea level rise at venice: what impact in term of flooding. *Climate Dynamics Observational, Theoretical and Computational Research on the Climate System*, 2009.

11 Selbstständigkeitserklärung

12 Anhang