

## Einreichung 2c – Dokumentation

### Emergency Hospital Locator Web-App

Jannes Dallmann, Matrikelnummer: 920932

Ioannis Svolos, Matrikelnummer: 906758

Entwurfsmethoden und -muster-WiSe24

Datum der Abgabe: 18.03.2025

#### Abstract

Die Emergency Hospital Locator App ist eine React-basierte Anwendung zur schnellen Ermittlung nahegelegener Krankenhäuser und Apotheken. Die Standortbestimmung erfolgt über die Geolocation API und wird auf einer interaktiven Karte dargestellt. Eine integrierte Liste medizinischer Einrichtungen in Berlin und Umgebung ermöglicht eine gezielte Suche. Basierend auf dem aktuellen Standort berechnet die App automatisch eine geradlinige Entfernung zu den nächstgelegenen Krankenhäusern und hebt diese hervor. Zusätzliche Funktionen umfassen die Anpassung des Suchradius, das Filtern nach Apotheken, die direkte Anwahl verfügbarer Telefonnummern und das Teilen von Standorten. Eine integrierte Notruffunktion ermöglicht schnelles Handeln im Ernstfall. Durch Antippen einer Einrichtung können detaillierte Informationen abgerufen werden.

## 1. TECHNISCHE UMSETZUNG

### 1.1 Technologien und Frameworks

Die Anwendung Emergency Locator wurde mit modernen Webtechnologien und Frameworks entwickelt, um eine leistungsfähige und benutzerfreundliche Lösung bereitzustellen.

Das Frontend basiert auf React, einer JavaScript-Bibliothek zur Erstellung interaktiver Benutzeroberflächen.

Für die Kartendarstellung wird Leaflet genutzt, um Karten einzubinden, Marker zu platzieren und Benutzerinteraktionen wie Zoomen und Verschieben zu ermöglichen. Die Kartendaten stammen von OpenStreetMap (OSM).

Die Standortermittlung erfolgt über die Geolocation API. Diese API ermöglicht es der Anwendung, den aktuellen Standort des Nutzers zu bestimmen, sofern die entsprechende Berechtigung erteilt wird.

Für ein modernes und konsistentes Design der Benutzeroberfläche wurde Material-UI (MUI) verwendet. Dieses beliebte React-Framework stellt vorgefertigte Komponenten und Designelemente bereit, die eine ansprechende und einheitliche Gestaltung unterstützen.

Die Suche nach Krankenhäusern und Apotheken erfolgt über die Overpass API. Diese API ermöglicht gezielte Abfragen in OpenStreetMap und liefert

strukturierte Daten zu medizinischen Einrichtungen innerhalb eines definierten Radius. Diese Informationen werden in der Anwendung verarbeitet und auf der Karte dargestellt.

### 1.2 Hauptkomponenten

Die Anwendung setzt sich aus mehreren Hauptkomponenten zusammen, die jeweils spezifische Funktionen übernehmen. Im Zentrum steht die Datei *EmergencyLocator.jsx*, die die Kernlogik für die Standortermittlung, die Kartenanzeige sowie die Interaktion mit den Krankenhaus- und Apothekendaten enthält. Sie koordiniert die verschiedenen Funktionen und sorgt dafür, dass die Benutzeroberfläche dynamisch auf Nutzeraktionen reagiert.

Ein essenzieller Bestandteil der Anwendung ist die Service-Klasse *HospitalService.js*. Sie übernimmt die Suche nach Krankenhäusern und Apotheken, indem sie Abfragen an die Overpass API sendet, die zurückgegebenen Daten verarbeitet und für die Anzeige in der Benutzeroberfläche aufbereitet.

Die Standortermittlung erfolgt über den benutzerdefinierten Hook *useGeolocation.js*. Dieser nutzt die Geolocation API des Browsers, um den aktuellen Standort des Nutzers zu ermitteln. Er stellt Funktionen für das Abrufen des Standorts, das Einholen von Berechtigungen und die Fehlerbehandlung bereit.

Abgerundet wird die Struktur der Anwendung durch die Datei *constants.js*, die eine zentrale Konstante „Notrufnummern“, enthält. Diese Konstante dient der Gestaltung der Benutzeroberfläche und erweitert die Funktionalität der Anwendung.

## 2. FUNKTIONSWEISE

### 2.1 Standortermittlung

Die Anwendung nutzt die Geolocation API, um den aktuellen Standort des Nutzers zu bestimmen. Dabei fragt der Browser die Gerätestandortdaten ab und gibt die entsprechenden Koordinaten (Breiten- und Längengrad) zurück. Falls der Nutzer die Standortfreigabe nicht erteilt, erscheint ein Dialogfeld mit einer entsprechenden Berechtigungsanfrage. Ohne diese Zustimmung kann die App den Standort nicht bestimmen und somit keine nahegelegenen Krankenhäuser oder Apotheken anzeigen.

### 2.2 Suche nach Krankenhäusern und Apotheken

Auf Basis des ermittelten Standorts sucht die Anwendung nach nahegelegenen Krankenhäusern und Apotheken. Hierfür wird die Overpass API genutzt, die OpenStreetMap-Daten mittels einer leistungsfähigen Abfragesprache durchsucht. Die API filtert nach Objekten mit den Tags **amenity=hospital** (Krankenhäuser) und **amenity=pharmacy** (Apotheken) innerhalb eines definierten Suchradius, z. B. 10 Kilometer. Anschließend liefert sie eine Liste der gefundenen Einrichtungen zurück.

### 2.3 Verarbeitung der Daten

Die von der Overpass API gelieferten Daten werden in der Methode **processOverpassResults** aufbereitet. Diese extrahiert wesentliche Informationen aus den OpenStreetMap-Tags, darunter Name, Telefonnummer, Website, Notaufnahme-Verfügbarkeit (**emergency=yes**), Barrierefreiheit (**wheelchair=yes**) und eine Beschreibung.

### 2.4 Umkreissuche

Die Anwendung ermöglicht es dem Nutzer, den Suchradius individuell anzupassen (z. B. 5 km, 10 km oder 20 km). Nur Einrichtungen innerhalb des gewählten Radius werden angezeigt. Diese Filterfunktion erleichtert es dem Nutzer, die Suchergebnisse auf relevante Standorte einzugrenzen.

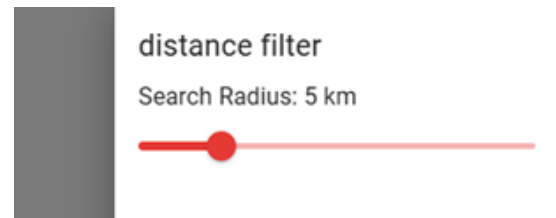


Abbildung 1: Suchradius Funktion

### 2.5 Kartenanzeige

Die ermittelten Krankenhäuser und Apotheken werden mithilfe der Leaflet-Bibliothek auf einer interaktiven Karte dargestellt. Diese zeigt sowohl den aktuellen Standort des Benutzers als auch die Standorte der medizinischen Einrichtungen an. Zur besseren Orientierung sind die Marker farblich gekennzeichnet:

Blau: Der aktuelle Standort des Benutzers.

Rot: Krankenhäuser.

Grün: Apotheken.

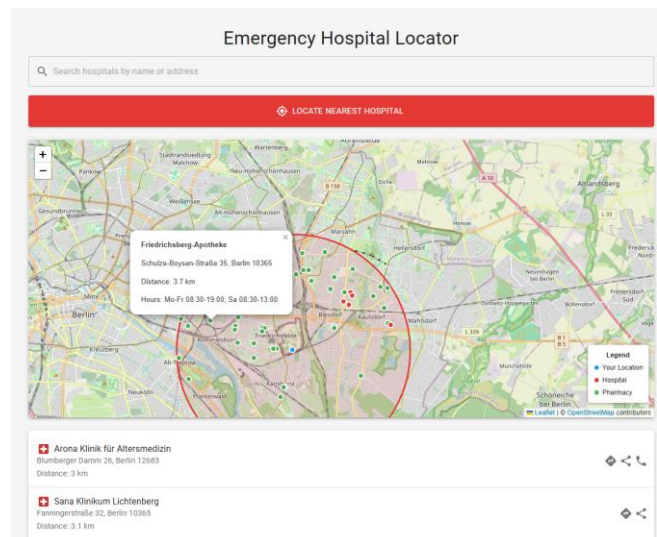


Abbildung 2: Kartenanzeige

Die Karte unterstützt interaktive Funktionen wie Zoomen und Scrollen. Beim Klicken auf einen Marker werden zusätzliche Informationen, darunter Name, Adresse und Entfernung, angezeigt.

### 3. WICHTIGE CODEBESTANDTEILE UND DATEIEN

#### 3.1 EmergencyLocator.jsx

Diese Datei ist eine zentrale Komponente der Anwendung für die Standorterkennung, Kartenanzeige sowie die Interaktion mit den Daten zu Krankenhäusern und Apotheken.

Wichtige Funktionen:

**handleLocateMe:** Ermittelt den Standort des Benutzers über die Geolocation API und startet die Suche nach nahegelegenen Krankenhäusern.

**handleHospitalClick:** Wird ausgelöst, wenn der Benutzer auf ein Krankenhaus in der Liste oder auf der Karte klickt. Sie öffnet eine Detailansicht mit Adresse, Telefonnummer und verfügbaren Diensten.

#### 3.2 HospitalService.js

Diese Datei dient für die Suche nach medizinischen Einrichtungen und übernimmt die Kommunikation mit der Overpass API, um relevante Daten zu beziehen.

Wichtige Funktionen:

**findNearbyHospitals:** Sendet eine API-Anfrage, um nahegelegene Krankenhäuser basierend auf dem Benutzerstandort zu ermitteln.

**processOverpassResults:** Verarbeitet die API-Daten und extrahiert relevante Informationen wie Name, Adresse, Telefonnummer und verfügbare Dienste, um sie in der App nutzbar zu machen.

#### 3.3 useGeolocation.js

Die Datei ist ein benutzerdefinierter React-Hook, der die Geolocation API des Browsers verwendet, um den Standort des Benutzers zu ermitteln.

Wichtige Funktionen:

**getLocation:** Holt den aktuellen Standort des Benutzers und liefert die Koordinaten.

**requestPermission:** Fordert die Standortfreigabe an, falls noch keine Berechtigung erteilt wurde.

#### 3.4 Constants.js

Die Datei enthält eine Konstante und dient als zentraler Ort für die Definition von Werten wie die Notrufnummern.

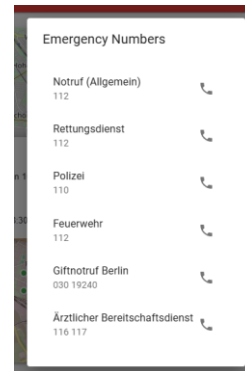


Abbildung 3: Notrufnummern-Ansicht in der Web-App

### 4. APP AUSFÜHREN (ANLEITUNG)

#### 4.1 Starten der App

Um die Anwendung zu testen, muss die Datei public.zip in ein geeignetes Verzeichnis abgelegt und entpackt werden. Anschließend öffnet man die Eingabeaufforderung (CMD) und navigiert zu dem Ordner, in dem die Datei abgelegt wurde. Dies erfolgt mit dem Befehl

1. Anschließend wird in das Verzeichnis navigiert, in dem der Quellcode der Anwendung gespeichert werden soll. Dies erfolgt mit dem Befehl `cd` (Change Directory). Beispielsweise:

```
cd C:\Benutzer\public
```

2. Anschließend werden die benötigten Abhängigkeiten installiert. Dieser Befehl lädt alle notwendigen Pakete und Bibliotheken herunter, die für die Anwendung erforderlich sind:

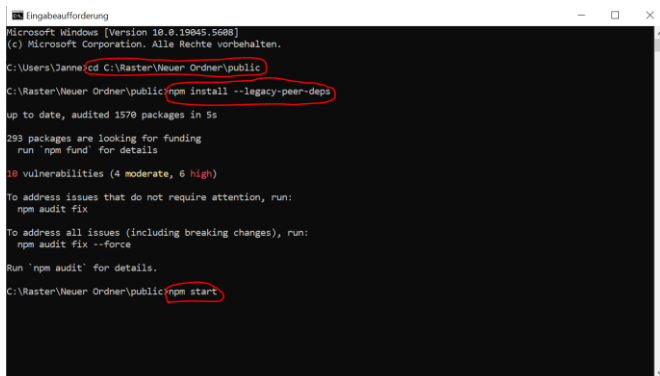
```
npm install --legacy-peer-deps
```

3. Sobald die Installation abgeschlossen ist, wird die Anwendung im Entwicklungsmodus gestartet. Dies erfolgt mit dem Befehl:

```
npm start
```

Die Anwendung wird nun gebaut und automatisch im Standardbrowser unter der Adresse `http://localhost:3000` geöffnet. Die Benutzeroberfläche der App ist nun sichtbar und einsatzbereit.

Zur Abgabe des Projekts gehört außerdem auch ein kurzes Demonstrationsvideo, das die Funktionsweise der App zeigt.



```
Microsoft Windows [Version 10.0.19045.5688]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Janne>cd C:\Raster\Neuer Ordner\public
C:\Raster\Neuer Ordner\public>npm install --legacy-peer-deps
up to date, audited 1570 packages in 5s
293 packages are looking for funding
  run 'npm fund' for details
10 vulnerabilities (4 moderate, 6 high)
To address issues that do not require attention, run:
  npm audit fix
To address all issues (including breaking changes), run:
  npm audit fix --force
Run 'npm audit' for details.
C:\Raster\Neuer Ordner\public>npm start
```

Abbildung 4: cmd-Ansicht beim Ausführen der App

## 5. DOWNLOADLINK

### 5.1 GitHub

<https://github.com/loannisSvolosBht/Emergency-Hospital-Locator-App-.git>

### 5.2 Google Drive

<https://drive.google.com/drive/folders/1EuZ2YtLeRCXpxZnq5KiKYgBHme3F6MgN?usp=sharing>