Wetter-App

Projekt M335

Nils Rothe, Alejandro wenzler, aywin nay

2023

Inhaltsverzeichnis

[Planung 2](#_Toc135209760)

[Funktionen 2](#_Toc135209761)

[Verwendete Elemente 2](#_Toc135209762)

[Testen 2](#_Toc135209763)

[Mockup 2](#_Toc135209764)

[Umsetzung 3](#_Toc135209765)

[Tools 3](#_Toc135209766)

[Tatsächlich Verwendete Elemente 3](#_Toc135209767)

[App.js 3](#_Toc135209768)

[Favoriten Screen 3](#_Toc135209769)

[Weather Screen 3](#_Toc135209770)

[Search Screen 3](#_Toc135209771)

[Publizieren 4](#_Toc135209772)

[Testen 4](#_Toc135209773)

[Test Schritte 4](#_Toc135209774)

# Planung

## Funktionen

Wir möchten eine Wetter App mit Erweiterungen realisieren.

Die App soll den Aktuellen Standort auslesen und das Wetter ausgeben.

Der Standort ebenfalls auf einer Karte dargestellt werden.

Der User kann Standorte als Favoriten speichern.

Der User kann in der App Orte suchen.

Als App Erweiterung werden an den jeweiligen Standorten noch Camping Plätze in der Nähe ausgegeben.

## Verwendete Elemente

GPS für Standortbestimmung

Wetter-API für Wetterdaten

API für Camping Plätze

## Testen

Stimmt der Automatisches Standort, wird er korrekt auf der Karte ausgegeben.

Funktioniert die Standortsuche.

Können Standorte gespeichert werden.

Werden Campingplätze in der Nähe ausgegeben.

## Mockup

Ein Bild, das Text, Wolke, Himmel, Screenshot enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Handy, Schrift enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Favoriten Übersicht

Ort Ansicht

Ort suchen

# Umsetzung

## Tools

Auflistung der Verwendeten Werkzeugen.

GitHub: Wir haben ein GitHub Repository erstellt, um gemeinsam an dem Projekt zu arbeiten.

Link: <https://github.com/deepspace1000/Wetterapp.git>

WebStorm: Als Programmierumgebung haben wir Webstorm verwendet.

React Native: Wir haben unsere App mit React Native programmiert.

## Tatsächlich Verwendete Elemente

Expo Location

Wetter API

Lokaler Speicher

Karte

## App.js

In der App.js Datei haben wir einen TabNavigator Implementiert mit welchem man zwischen den drei Hauptseiten Navigieren kann.

## Favoriten Screen

Im Favoriten Screen kann ein User Standorte abspeichern, um diese Später schnell wieder zu finden.   
Durch das Auswählen eines Gespeichertes Standortes wird die Weather Screen Komponente aufgerufen und der Ortsname mit den Navigation props übergeben.

Async Storage: Um Daten zu speichern verwenden wir Async Storage um Daten/Standorte Lokal auf dem Gerät zu speichern. Diese werden bei Programm Start geladen. Gespeicherte Standorte können auch auf dem Screen wieder gelöscht werden.

Expo Location: Es wird als erstes Element der Aktuelle Standort angezeigt dieser wird mit Expo Location abgerufen. Die abgerufenen Koordinaten werden dann ebenfalls mit Expo Location in einen Ortsnamen zurück übersetzt durch Reverse Geocoding.

## Weather Screen

Im Weather Screen werden die Wetterdaten angezeigt. Der Ortsname wird von den Anderen Screens per props an diesen Screen übergeben und es wird mir der API die Daten des Ortes abgerufen.

Wetter-API: Die Wetterdaten werden von der API von Open Weather Map abgerufen. Wir konnten uns dort als Studenten Registrieren und kostenlos einen Developer Plan Abonnement abschliessen wodurch wir mehr abfragen durchführen konnten und Zugriff auf mehr Daten hatten.

Async Storage: In diesem Screen wurde ebenfalls einen Zugriff auf den Lokalen Speicher eingerichtet, um Standorte in die Favoriten speichern zu können.

## Search Screen

Im Search Screen hat es ein eine Suche, in welcher der User nach einem Ortsnamen suchen kann. Wenn der User dann das Such Symbol drückt, wird der Eingegebene Ortsnamen als prop an Weather Screen übergeben.   
Im Screen hat es auch eine Karte, auf welcher der Automatisch der Standort angezeigt wird, welcher der User ins Suchfeld eingegeben hat.

Map View: Um die Karte darzustellen wird Map View von React-Native-Maps verwendet.

Expo Location: Um den Standort auf der Karte anzuzeigen werden Koordinaten benötigt. Um diese zu bekommen, wird mit Expo Location mittels Geo Coding der eingegebene Ortsname in Koordinaten umgewandelt.

# Publizieren

# Testen

Wir haben unser Projekt manuell getestet.

## Test Schritte

**Stimmt der Automatisches Standort, wird er korrekt auf der Karte ausgegeben.**

Erweiterung: Der Aktuelle Standort wird nie auf der Karte ausgegeben.  
Angepasste Definition: Der Aktuelle Standort wird richtig ausgelesen und die Karte zeigt einen Gesuchten Standort richtig an.  
Test:

Aktueller Standort stimmt: Positiv

Karte zeigt den Gesuchten Ort an: Positiv

**Funktioniert die Standortsuche.**

Test:

Es kann im Textfeld etwas eingegeben werden: Positiv

Beim Suchen wird im Weather Screen die Wetter Daten des Ortes ausgegeben: positiv

**Können Standorte gespeichert werden.**

Test:

Im Weather Screen können Standorte als Favoriten markiert werden: positiv

Markierte Standorte werden abgespeichert: positiv

Gespeicherte Standorte werden im Favoriten Screen angezeigt: positiv

**Werden Campingplätze in der Nähe ausgegeben.**

Erweiterung: Dieser Funktion wurde nicht implementiert.