HTWG Konstanz

Fakultät für Informatik

DeepRain

Hilfsdokumentation zur einarbeitung für Entwickler in das Projekt

MSI AS - Master Informatik, Autonome Systeme

Autoren: Simon Christofzik
Paul Sutter
Till Reitlinger

Version vom: 7. September 2020

Betreuer: Prof. Dr. Oliver Dürr

Inhaltsverzeichnis

1 Einleitung

Dieses Dokument soll dabei helfen, sich in den aktuell bestehenden Projektstand möglichst einfach einzuarbeiten. Daher wurden hier einige Informationen, die keinen Platz in der offiziellen Dokumentation gefunden haben, aber dennoch wichtig sind, gesammelt.

2 Getting Started

Im ersten Schritt den Master aus dem Git auschecken (DeepRain2). So bekommt Ihr den finalen Stand des Projektes.

3 deep_rain_app Ordnerstruktur

In diesem Ordner befindet sich der gesamte Code, der für jegliche Funktionalitäten der App benötigt wird. Dazu gehört der Eigentliche App Code (deep_rain), die Cloudfunction, welche für die Pushbenachrichtigung benötigt wird (FlutterCloudFunction), sowie der Serversimulator (deeprainFireBase).

3.1 deep_rain

Die relevanten Ordner sind lib und assets.

3.1.1 assets

Hier befinden sich alle in der App benötigten Bilder, sowie andere statische Daten. Dazu gehören zum Beispiel die Listen mit hilfe welcher der Pixel in den Vorhersage PNGs berechnet werden kann (data).

3.1.2 lib

In diesem Ordner befindet sich der komplette, selbst geschriebene, Flutter Code.

DataObjects

In diesem Ordner befinden sich die Klasse, welche die Vorhersage PNGs speichert (DataHolder), sowie die Klasse die das Objekt beschreibt, welches in der Liste angezeigt wird (ForecastListItem).

global

In diesem Ordner befinden sich zwei Datenklassen welche von der gesamten App verwendet werden. In der Datei GlobalValues befinden sich alle global gespeicherten

Variablen. In der Datei UIText befinden sich alle angezeigten Texte in verschiedenen Sprachen.

screens

Es gibt für jeden Screen eine Datei in dem Ordner screens. Diese werden über die Datei main.dart verwaltet.

Services

Database.dart kümmert sich um die Kommunktion mit der Datenbank und ist die einzige Klasse, welche direkt mit der Datenbank kommuniziert. Alle Datenbank aktionen laufen über diese Klasse. FindPixel.dart stellt eine Funktion zur Verfügung, mit welcher es möglich ist, den eigenen Pixel im Vorhersage PNG zu finden. ProvideForecastData stellt die aktuellen ForecastListItems für die Forecast List zur verfügung. In PushNotification.dart wird festgelegt, wie sich die App bei dem öffnen und erhalten einer Pushbenachrichtigung verhalten soll.

Widgets

Die beiden .dart Dateien in diesem Ordner legen fest, wie die ForecastListe aussehen soll. forecast_list_widget stellt dabei die gesamte Liste dar, forecast_tile nur eine Zeile aus der Liste.

3.2 deepRainFireBase

Database.py simuliert den Server, wenn dieser nicht aktiv ist. Es werden Regenvorhersagen in die Datenbank geladen und Push Benachrichtigungen getriggert. Viele Teile dieser Klasse sind später auch Teil der verwendeten Pipeline geworden. Diese Klasse ist nur ein Werkzeug, um die weiterentwicklung der App zu vereinfachen.

3.3 FlutterCloudFunction

In dem Ordner functions, befindet sich in der Datei index.js die eigentliche CloudFunktion. Die Funktionsweise dieser Funktion ist in der Hauptdokumentation ausgeführt. Um die änderungen an der CloudFunktion zu veröffentlichen und anzuwenden, muss diese zuerst über die Konsole deployed werden.

4 DeepRain App Getting Started

Prof. Dr. Oliver Duerr kann euch Zugriff auf das Projekt in Firebase geben. Von dort aus habt Ihr auf die gesamte Firebase zugriff und könnt die Datenbanken, CloudFunktionen, Verschlüsselung ect. verwalten.

4.1 Die App zum laufen bekommen(deep_rain)

Um den Code auf einem Endgerät auszuführen, muss Flutter installiert sein. Siehe dazu folgender Link: https://flutter.dev/docs/get-started/install Dann das Projekt in einer beliebigen IDE öffnen, Smartphone per USB verbinden und Code ausführen. Wir haben immer mit IntelliJ Ultimate Idea gearbeitet.

4.2 Server Simulator zum laufen bekommen (deepRainFireBase)

Es müssen die benötigten Pakete installiert werden. Im Anschluss kann der Server ohne weiteres verwendet werden.