Python Projekt

Wetter-App zur Wiedergabe/Visualisierung von Standortspezifischen Wetter-Daten

– Jannic Meier, Luc Weishar, Elias Peter

Inhalt

[1. Projektbeschreibung 1](#_Toc75508005)

[2. Module /- externe Module 2](#_Toc75508006)

[3. Zielplatform 2](#_Toc75508007)

[4. Ergebnis 2](#_Toc75508008)

# Projektbeschreibung

Ziel des Projektes ist die Entwicklung und Implementierung einer Wetter-App (Datencrawler). Dabei soll der Benutzer eine grafische Oberfläche haben auf welcher Wetterdaten angezeigt werden sowie Vorhersagen für die nächsten 7 Tage erscheinen. Dabei soll der Standort auch gewählt werden und abhängig davon Daten gesammelt werden. Das Programm soll abhängig von den anstehenden Wetterbedingungen für den Tag Benachrichtigungen versenden (z.B. „nehme heute den Regenschirm mit, es soll Regnen“)

**In-Scope:**

* Benachrichtungen zu bedingten Wetterbedingungen
* GUI zur Anzeige der Wettervorhersage (nächste 8 Tage)
* Wetterstatistik Niederschlag/etc...
* Standortbezogene Wettervorhersage
* Benutzer kann zwischen Standorten wechseln

**Nice-to-have:**

* Uhrzeit und Datum wird angezeigt
* Temperatur für den gewählten Standort
* Temperaturumrechnung von C° in F°
* Wetter-Historie (letzten 5 Tage, wie war das Wetter)
* Es kann ein Favoriten-Standort angelegt werden, welchem beim Start sofort alle Daten anzeigt
* Es wird die Flagge des jeweiligen Lands angezeigt, damit man Orte wie Paris - Frankreich und Paris - USA unterscheiden kann auf den ersten Blick
* Das User-Interface passt sich der lokalen Zeit des Users an - Tagsüber wird ein Sonnen-Logo angezeigt und bei Nacht ein Mond-logo

**Out-of-Scope:**

* Einbindung mit Alexa/Smart Home
* Bluetooth Einbindung

# Module / externe Module

Module

|  |  |
| --- | --- |
| Modul | Beschreibung |
| Main.py | In Main.py werden die Wetter-Daten von einer externen Seite abgefragt und dann in nächsten Schritten verarbeitet und in dem User-Interface übersichtlich und leicht einsehbar dargestellt |

Externe Module

|  |  |
| --- | --- |
| * certifi * chardet * datetime * geojson * idna * json * numpy * Pillow * pip * pyowm * PySocks * requests | * setuptools * time * io * tkinter * urllib3 * cycler * kiwisolver * matplotlib * pyparsing * python-dateutil |

# Zielplatform

Als Zielplatform für dieses Projekt ist ein Windows 10 Laptop oder Desktop, jedoch würde sich die Anwendung auch für Tablets eignen

# Ergebnis

Das Ergebnis des Projektes war sehr erfolgreich und alles funktioniert so wie wir es geplant haben. Es wurden fast alle Features umgesetzt, die wir zu Beginn festgelegt haben und es kamen noch einige Zusätzlich dazu, jedoch sind auch Features weggefallen, das aus begrenzter Verfügbarkeit der Daten durch den gewählten Provider nicht möglich war, ohne weitere Kosten zu verursachen.

* Uhrzeit und Datum wird angezeigt
* Es kann ein Favoriten-Standort angelegt werden, welchem beim Start sofort alle Daten anzeigt
* Es wird die Flagge des jeweiligen Lands angezeigt, damit man Orte wie Paris - Frankreich und Paris - USA unterscheiden kann auf den ersten Blick
* Das User-Interface passt sich der lokalen Zeit des Users an - Tagsüber wird ein Sonnen-Logo angezeigt und bei Nacht ein Mond-logo
* Wetterstatistik – in Form von Statistiken/Graphen
* Wettervergleiche – Durchschnittstemperatur Vergleich letzter Monat/Jahr
* Wetterstatistik – in Form von einer Heatmap

(in Grün die neuen Features und in Rot das Feature, welches weggefallen ist)

Die Wetterbedingten Benachrichtigungen lassen sich auch noch erweitern/anpassen, bis jetzt gibt es nur eine Warnung, wenn es Regen sollte.

Die Wetterstatistik in Form von einer Heatmap, konnte nicht umgesetzt werden aufgrund nicht verfügbarer Daten. Alternativen, die stets eine Heatmap beinhalteten, konnten ebenfalls nicht umgesetzt werden, da diese entweder Javascript oder html benötigten. Die Arbeit mit Tkinter war gewöhnungsbedürftig.

