Planung

|  |  |  |
| --- | --- | --- |
| **Deadline** | **Titel** | **Beschreibung** |
| 10.10.2021 | Erste Verbindung zur Schnittstelle, CSV und Daten von API einlesen | Jeder soll auf seinem Rechner das Programm ausführen können |
| 17.10.2021 | Ausgabe der aktuellen Temperatur auf der Webseite | Es soll nicht schön formatiert sein, soll nur ein erster Prototyp sein. |
| 24.10.2021 | Programm auf dem Raspberry PI installieren und zum Laufen bringen | Vorgang dokumentieren, jeder soll dies auf seinem Raspberry PI machen |
| 31.10.2021 | UI-Sketch der Flask-Webseite | Zeichnung welche Werte wo angezeigt werden. (aktuelle und historische Daten sowie Voraussagen) |
| 07.11.2021 | UI implementieren |  |
| 21.11.2021 | Autostart des Programmes und Update Software auf RPI einrichten. | InfluxDB als Service installieren. Chrome im Kiosk Modus starten. |
| 28.11.2021 | Schritt-für-Schritt Anleitung erstellen, wie man einen neuen RPI einrichtet | Zum Testen: OS auf SD-Karte neu installieren. |
| 12.12.2021 | Error handling definieren und implementieren |  |
| 5.12.20201 | Optional: Forecast implementieren |  |
| 19.12.2021 | Dokumentation für Programmierer erstellen.  Anwendung und Anleitung testen |  |
| 02.01.2022 | Präsentation erstellen / Zeitreserve für Unvorhergesehenes |  |
| 11.01.2022 | Abgabe |  |
| 17.01.2022 | Verteidigung? |  |

Bemerkungen von Lucas:

* Möglichst schnell ein Konzept was und wie angezeigt wird.
* Struktur der DB ist schon vorgegeben, Strukturbearbeiten (wenn überhaupt nötig, falls eine weitere Spalte hinzukommt) erst später machen.
* Daten importieren möglichst früh machen.
* Weniger um DB kümmern. Mehr was und wie anzeigen (UI-Sketch).
* Bis in einem Monat aktuelle Werte ausgeben (nur Temperatur und Wasser, nicht schön formatiert)
* Chrome Kioskmodus benutzen
* Trello benutzen

Ziel neuen der Planung. Durchbrüche am Anfang machen, um allfällige Hindernisse früh zu erkennen.