

Wetterstation



# Wetterstation

Smart Home Praktikum

Niklas Schildhauer (ns162)  
Carina Szkudlarek (cs324)  
Nicolas Wyderka (nw073)  
Alexander Merker (am206)  
Kim Bastiaanse (kb139)

CSM SoSe 2021  
Hochschule der Medien

Prof. Dr. Gottfried Zimmermann  
Tobias Ableitner

## Abstract

Im Rahmen der Vorlesung "Smart-Home Praktikum" im Studiengang Computer Science and Media an der Hochschule der Medien wurde im Sommersemester 2021 die Wetterstation WeatherIO entwickelt.

Mithilfe von verschiedenen Sensoren erfasst WeatherIO sowohl im Außen- als auch im Innenbereich Messdaten und speichert diese in einer Datenbank.

Die Ergebnisse werden anschließend in einer personalisierbaren Web Applikation grafisch dargestellt. Mit den Personalisierungsfunktionen Self-Voicing, Reduce Motion, anpassbare Schriftgröße und High Contrast Theme werden Personen mit unterschiedlichen Bedürfnissen und Einschränkungen unterstützt.

## Keywords

Wetterstation; Smart Home;  
Personalisierung; Anpassung; OpenApe;  
Accessibility; Adaptable; Adaptive

## Einleitung

Das Projekt Wetterstation wurde als Teil der Vorlesung "Smart-Home Praktikum" von einem interdisziplinären Team aus fünf Studenten des Studiengangs Computer Science and Media (CSM) im Sommersemester 2021 entwickelt. Die Wetterstation wurde mit embedded Komponenten gebaut und trägt den Namen WeatherIO. Die Anzeige der Messdaten lässt sich an die individuellen Bedürfnisse unterschiedlichster Personen anpassen, um somit ein zufriedenstellendes Nutzungserlebnis zu bieten.

## State of the Art

### Allgemein

Meistens werden Produkte spezifisch für eine bestimmte Zielgruppe entwickelt. Dabei werden Personengruppen mit Beeinträchtigungen häufig vernachlässigt, was dazu führt, dass diese ein Produkt nur eingeschränkt oder gar nicht nutzen können (Wisniewska, 2013). Die von uns erstellte Wetterstation nutzt den Ansatz einer adaptiven Benutzeroberfläche. Sie lässt sich an Menschen mit unterschiedlichen Bedürfnissen anpassen, um so ein angenehmes und barrierefreies Nutzungserlebnis zu bieten. Die Funktionsweise und weitere Informationen über adaptable und adaptive UIs werden von Loitsch (2018, S.18) näher beschrieben. Um uns über die unterschiedlichen Benutzergruppen und deren Bedürfnisse im Klaren zu sein, wurden im Voraus Benutzerprofile entwickelt.

Ein weiteres Ziel war es, die Wetterstation so kostengünstig wie möglich zu entwickeln, sodass eine Alternative zu teuren Markenprodukten entsteht.

### Benutzerprofile

Da der Fokus der Wetterstation auf der Entwicklung einer adaptable Benutzeroberfläche lag, wurden die drei Benutzerprofile David, Ingrid und Nadine entwickelt. Hierfür wurden jeweils Personalisierungseinstellungen implementiert.

## David

Das Benutzerprofil von David beschreibt einen Nutzer mit einer Pollenallergie. Außerdem leidet er an einer Achromatopsie, durch diese ist seine Farbwahrnehmung stark eingeschränkt. Farben können nur als Grauabstufungen wahrgenommen werden. (Kohl & Wissinger, 2005).

## Ingrid

Das Profil Ingrid beschreibt eine Person höheren Alters. Dies bedeutet unter anderem, dass sie Schwierigkeiten mit kleiner Schriftgröße hat. Oftmals vergisst sie, zu lüften was für schlechte Luft im Innenraum sorgt. Eine Anzeige würde ihr dabei sehr helfen. Außerdem bevorzugt sie Oberflächen mit reduzierten Animationen, um Ihrer vestibulären Störung (Craig, 2018) entgegenzukommen.

## Nadine

Das Dritte Profil beschreibt Nadine, eine blinde Person. Diese würde von einer integrierten Sprachausgabe profitieren.

# Umsetzung

## Embedded

Für die Hardware Umsetzung wurden zwei Sensorpakete erstellt, eines zum Messen von Wetterdaten im Außenbereich und eines zur Messung von Temperatur und Luftqualität im Innenraum. Das Paket für den Außenbereich besteht aus einem Controller und einem BME280 Sensor, der Temperatur, Luftfeuchtigkeit und Luftdruck misst. Das Paket für den Innenraum enthält ebenfalls einen Controller, einen BME280

Sensor und zusätzlich einen MQ135 Sensor für die Messung der Innenraumluftqualität.

## Wettervorhersage

Die Wettervorhersage wurde unter Einbeziehung unserer selbst gemessenen Daten implementiert. Konkret wurde diese Vorhersage auf Basis des Zambretti Forecast/Algorithmus umgesetzt, welcher das Wetter für ein Zeitfenster von 12h prognostizieren kann (Beteljuices Zambretti Algorithm Based Forecaster, 2013). Als Hauptmessgröße für die Zambretti-Vorhersage wird der Luftdruck und dessen Veränderung über die Zeit verwendet (Meteormetrics Limited: Precision Instruments for Measurement & Detection, 2021). Die nötigen Informationen hierfür erhalten wir durch die Messungen unseres Außensensors. Die konkrete Art der Berechnung der Vorhersage basiert dabei auf dem von Andonov et. al (2018) vorgestellten Verfahren. Dieses stellt vor, wie der Zambretti Algorithmus zur Vorhersage des Wetters in einer Smart-Home Applikation verwendet werden kann. Außerdem behandelt das Verfahren von Andonov et al. eine Zusammenfassung der feingranularen Ausgabewerte des Zambretti-Verfahrens in gröbere, besser nutzbare Kategorien zur Wettervorsage.

## Architektur

Die Umsetzung des Webservers erfolgte in Form einer Microservice-Architektur. Dies hat den Vorteil, dass die Funktionen des Servers unabhängig voneinander implementiert werden konnten. Für jede Funktion (Wetterdaten, Datenbank, Authentifizierung, Personalisierung) des

Servers wurde somit ein eigener Service implementiert.

## Web Applikation

Für das Frontend wurde eine Angular Progressive Web App (PWA) entwickelt. Diese wird auf einem Raspberry Pi im lokalen Netzwerk gehostet und ist somit im Heimnetz für alle Geräte verfügbar. Es wurde der “mobile first”-Ansatz verfolgt. Für den Nutzenden sollte der Zugriff auf Informationen so schnell und unkompliziert wie möglich erfolgen. Außerdem haben wir uns somit auf den relevantesten Inhalt begrenzt (Xia, 2017).

Ohne Anmeldung kann so ein Nutzer auf die Informationen der äußeren Witterungsverhältnisse, wie die Temperatur, Luftfeuchtigkeit und Vorhersage sowie auf verschiedene Messwerte aus dem Innenraum (Temperatur, Luftfeuchtigkeit und Innenraumluftqualität) und den aktuellen Pollenflug zugreifen.

Für die Startseite wurde das Dashboard UI-Pattern verwendet (“User Interface Design Patterns”, n.d.). Dadurch soll der Nutzenden auf einen Blick alle wichtigen Informationen einsehen können. Hierfür wurden Widgets eingesetzt, welche die Informationen darstellen und auf eine Detailseite verweisen.



Abb. 01 Dashboardansicht

In Abbildung 01 sind zwei Screenshots des Dashboards zu sehen. Die Darstellung auf der rechten Seite der Abbildung zeigt einen unteren Ausschnitt des Bildschirms. Hierbei ist erkennbar, dass nun die Außentemperatur und das aktuelle Wetter Icon verkleinert über den anderen Widgets schwebt. Das in Abbildung 02 präsentierte Historie Widget zeigt eine graphische Darstellung des Verlaufs der Temperatur über einen Zeitraum. Des Weiteren können in der Detailansicht der Historie weitere Daten über einen größeren Zeitraum, bspw. einen Monat abgerufen werden. Neben der Temperatur ist auch eine Historie zur Luftfeuchtigkeit verfügbar.

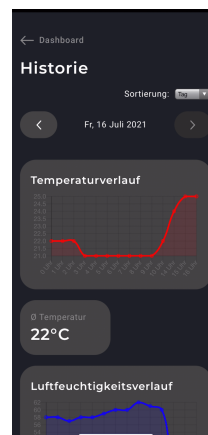


Abb. 02 Detailansicht Historie.

Die Widgets im Dashboard können adaptiv angeordnet werden. Dies hat den Vorteil, dass eine Priorisierung anhand der Präferenzen des Nutzers möglich ist. Beispielsweise erscheint das Innenraumluftqualitäts-Widget ganz oben, sollte die Luftqualität schlecht sein.

Dieses Vorgehen wurde für die Anzeige des Pollenflugs in gleicher Weise umgesetzt. Hierbei werden ausgewählte Pollentypen im Dashboard abhängig von der aktuellen Belastung stets an oberster Stelle angezeigt.

## Personalisierung

Mithilfe der Einbindung von OpenApe kann der Nutzer seine globalen Personalisierungseinstellungen auf die Wetterstation synchronisieren. Dies vereinfacht die Einrichtung der Wetterstation für den Nutzer.

### Themes

Eine der Personalisierungsoptionen ist die Auswahl zwischen drei unterschiedlichen Themes. Ein Light-, ein Dark- und ein High-Contrast-Theme wurden hierfür implementiert, siehe Abbildung 03.

Das High-Contrast-Theme soll Nutzer mit Sehbeeinträchtigung unterstützen, indem hohe Kontraste zwischen Text und Hintergrund eingesetzt werden und die Komplexität der Oberfläche reduziert wird (Microsoft Edge Team, 2020). Damit soll David die Bedienung der Wetterstation erleichtert werden. Die beiden anderen Themes wurden eingebaut, um auf persönliche Präferenzen eingehen zu können.

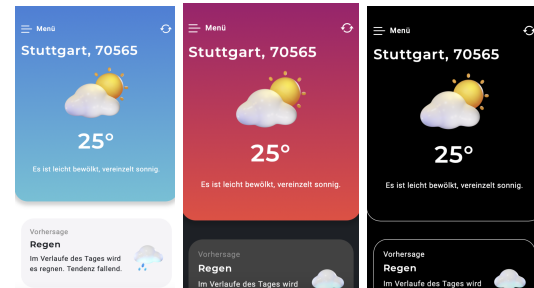


Abb. 03 Themes: Light, Dark, High-Contrast (links nach rechts)

Um die Dynamik und die allgemeine User Experience zu verbessern ändert die Anwendung je nach Tageszeit oder auch Wetterverhältnissen den Farbverlauf auf dem Dashboard. So erscheint die Oberfläche bei Dämmerung beispielsweise in einem dunklen Orange, bei Regen wird sie dunkler.

### Pollen

Ein User hat die Möglichkeit, Pollenallergien, die ihn betreffen, anzugeben. Daraufhin werden nur die ausgewählten, relevanten Pollendaten angezeigt. Wenn zur Zeit eine hohe Belastung für einen bestimmten, vom User ausgewählten Pollentyp vorliegt, wird das Pollen-Widget ganz oben im Dashboard angezeigt.

Damit ist die im Benutzerprofil "David" referenzierte Anforderung, Pollenallergien anzuzeigen und personalisieren zu können, erfüllt.

### Schriftgröße

Für ältere Nutzergruppen wie Ingrid oder Personen, die allgemein eine größere Schriftart bevorzugen wurde die Option implementiert, auch diese individuell über einen Slider anzupassen.

## Reduce motion

Für Personen, wie das Benutzerprofil Ingrid wurde eine Reduce Motion Funktion implementiert. Im Falle, dass der Nutzende an einer vestibulären Störung leidet können damit störende Animationen deaktiviert werden können. Somit besteht die Möglichkeit Trigger abzustellen (Craig, 2017).

## Self-Voicing

Die Self-Voicing Funktion unterstützt vor allem Personen mit Sehbeeinträchtigung, indem sie den Inhalt der auf der Seite dargestellten Informationen laut vorliest. Damit ist die im Benutzerprofil "Nadine" referenzierte Anforderung, eine Sprachausgabe für blinde Personen zur Verfügung zu stellen, erfüllt.

## Ergebnis

Im Rahmen des Projekts wurde unter Verwendung von zwei Sensorentypen, einem Microcontroller und einem RaspberryPi erfolgreich eine Wetterstation entwickelt. Es wurde eine Benutzeroberfläche entwickelt, welche adaptable und adaptive ist. Diese passt sich an den Benutzerkontext und den Umweltkontext an. Die Kacheln der UI werden basieren auf den aktuellen Wetterdaten angeordnet.

Eine Lüftungserinnerung per Sprachausgabe wurde nicht umgesetzt. Geplant war es, einen Lautsprecher an den Raspberry Pi anzuschließen, welcher per Sprachausgabe an das Lüften erinnert.

## Ausblick

Im folgenden Abschnitt werden mögliche Weiterentwicklungen und Vorschläge zur Verbesserung der WeatherIO Wetterstation vorgestellt.

Eine mögliche Erweiterung ist die Implementierung der Lüftungserinnerung über einen Lautsprecher. Weiter könnte man den Lautsprecher dazu nutzen, um die Self-Voicing-Funktion zu erweitern und Sprache über denselben Lautsprecher auszugeben. Dadurch würde die im Nutzerprofil Nadine referenzierte Person noch mehr von der Sprachausgabe profitieren.

Beim Frontend wäre der nächste Schritt eine native App zu entwickeln, um Push-Notifications zu ermöglichen. Dies würde ermöglichen, Lüftungserinnerungen auch per Push-Notification direkt auf Mobilgeräten zu erhalten.

Um die Wettervorhersage zu verbessern kann der Wetterstation ein Sensor zur Messung des Windes hinzugefügt werden. Dies ermöglicht präzisere Wettervorhersagen unter Verwendung von Messwerten der Windrichtung.

## Quellen

### A

Andonov & Petrov (2018). Weather station for smart home applications.

### B

Bernd Wissinger, Susanne Kohl (2005). Genetische Ursachen der Farbenblindheit.

Abgerufen 19. Juli 2021 von

<http://www.apickert.ch/farbenblindheit.pdf>

*Beteljuices Zambretti Algorithm based Forecaster.* (2013, 26. Mai). Abgerufen am 19. Juli 2021 von <http://www.beteljuice.co.uk/zambretti/forecast.html>

## C

Craig, J. (2017). Responsive design for motion. Abgerufen 19. Juli 2021 von <https://webkit.org/blog/7551/responsive-design-for-motion/>

## L

Loitsch, C. (2018). *Designing accessible user interfaces for all by means of adaptive systems.*

## U

"User Interface Design patterns" (n.d.). Dashboard Design pattern. Abgerufen 19. Juli 2021 von <http://ui-patterns.com/patterns/dashboard>

## M

Meteormetrics Limited: Precision Instruments for Measurement & Detection. Abgerufen am 19. Juli 2021 von <https://web.archive.org/web/20110610213848/http://www.meteormetrics.com/zambretti.htm>

Microsoft Edge Team. (2020). Styling for windows high contrast with new standards for forced colors. *Microsoft Blog*. Verfügbar 19. Juli 2021 unter <https://blogs.windows.com/msedgedev/2020/09/17/styling-for-windows-high-contrast-with-new-standards-for-forced-colors/>

## W

Wisniewska, Iwona (2013). Entwickler schrecken vor barrierefreien Apps zurück <https://www.derstandard.at/story/1373514414194/entwickler-schrecken-vor-barrierefreien-apps-zurueck>. Abgerufen 19. Juli 2021

## X

Xia, Vincent (2017). What is Mobile First Design? Why It's Important & How To Make It? Aufgerufen am 19. Juli 2021 von <https://medium.com/@Vincentxia77/what-is-mobile-first-design-why-its-important-how-to-make-it-7d3cf2e29d00>