**Use-Case „Wetterdatenanalyse“**

**Überblick**

Der Benutzer möchte Wetterstationen im Umkreis bestimmter Koordinaten finden, um später deren Messdaten für einen angegebenen Zeitraum tabellarisch und/oder grafisch anzuzeigen.

**Akteure**

**Primärakteur:** Benutzer der Anwendung (Benutzer) -> Möchte sich die gemessenen Temperaturminima und -maxima je Jahr oder nach meteorologischen Jahreszeiten im Umkreis von einem bestimmten Längen- und Breitengrad anzeigen lassen.

**Sekundärakteur**: Website für Datenabruf (Global Historical Climatology Network) -> Stellt die Daten zu den Stationen und den gemessenen Temperaturen bereit.

**Tertiärakteur:** Dozenten sind daran interessiert, dass die Studenten das notwendige Wissen erwerben.

**Stakeholder**

**Benutzer:** Möchten gemessene Temperaturen suchen und die Messdaten auswerten. **H**aben primär das Ziel, schnell und zuverlässig an Daten zu kommen und diese auch zu visualisieren.

**Vorbedingungen**

* Netzwerkverbindung ist vorhanden.
* Die Datenquelle (Daily Global Historical Climatology Network) ist verfügbar.
* Browser ist kompatibel.

**Hauptzweig**

1. Benutzer öffnet die Anwendung im Browser.
2. System zeigt Eingabemasken an.
3. Benutzer gibt Breitengrad und Längengrad in das entsprechende Eingabefeld ein.
4. Benutzer gibt den Suchradius ein.
5. Benutzer gibt die maximale Anzahl der angezeigten Stationen ein.
6. Benutzer gibt den Zeitraum ein.
7. Benutzer löst Suche nach Stationen aus.
8. System zeigt die gefundenen Stationen an.
9. Benutzer wählt eine gewünschte Station aus.
10. Benutzer wählt den Anzeigetyp (Beides/Tabellarisch/Grafisch).
11. Benutzer legt fest, für welche meteorologischen Jahreszeiten (alle oder ausgewählte) Werte angezeigt werden sollen.
12. Benutzer löst Anzeige der Auswertung aus.
13. System zeigt die Mittelwerte für Temperaturminima und -maxima an.
14. Benutzer sichtet die angezeigten Informationen.

**Erweiterung**

\*a. Container stürzt ab.

* Container wird neu gestartet.

3-6. Benutzer macht fehlerhafte Eingabe (z.B. Text anstelle von einer Zahl, Pflichtfelder nicht ausgefüllt).

* Benutzer wird aufgefordert die Eingabe zu korrigieren auf das passende Format.
* Benutzer korrigiert fehlerhafte Eingabe.

8a. Keine Stationen werden gefunden.

* Benutzer bekommt die Mitteilung, dass im gewählten Radius keine Messtationen liegen und der Suchradius erweitert werden muss.
* Benutzer wählt neuen Radius.

12a. Für ein bestimmtes Jahr oder eine Jahreszeit liegen keine Messwerte vor.

* Es soll eine Fehlermeldung eingeblendet werden, wenn keine Messdaten angezeigt werden können.

**Besondere Anforderungen**

**Grundsätzliche Anforderungen**

* Die Anwendung hat eine grafische Benutzeroberfläche.
* Die Anwendung wird in einer Client-Server-Architektur ausgeführt.
* Beim dem Client kann sich um eine Desktop-Anwendung oder eine Web-Anwendung handeln.
* Der Server wird in einem oder mehreren Containern aufgesetzt.
* Die Anwendung ist lauffähig unter Windows 11, optional in einer WSL

(Windows Subsystem for Linux) Virtual Machine (VM).

* Als Datenbasis wird das Daily Global Historical Climatology Network eingesetzt.

**Nicht funktionale Anforderungen**

Code

* Freie Wahl der Programmiersprachen und Frameworks.
* Testautomatisierung mit Unit- und Service-Tests, TDD wird empfohlen, Testabdeckung über 80 %.
* Manuelle oder automatisierte Systemtests.
* Zur Strukturierung der Services wird eine Schichtenarchitektur empfohlen.
* Beim Erstellen von Serviceschnittstellen soll REST als Schnittstellenformat genutzt werden.
* Absicherung der Schnittstelle ist optional.
* Datenbank ist für das Projekt optional, begründete Wahl einer Datenbank, die Datenbank darf maximal 5 GB Festplattenspeicher verbrauchen.

Betriebsumgebung

* Zur Containerisierung ist Docker oder Podman zu verwenden.
* Ein Container soll jeweils maximal 2 CPUs und 1 GB RAM nutzen.
* Zur Orchestrierung mehrerer Container ist Docker Compose oder Podman Compose zu verwenden.

DevOps

* Verwendung von GitHub zur Versionskontrolle.
* Das GitHub-Repository soll öffentlich sein oder die Dozenten müssen in ein privates Repository eingeladen werden.
* Verwendung von GitHub Actions zur Umsetzung von CI/CD Pipelines.
* Verwendung der GitHub Container Registry zur Veröffentlichung der Container-Images.

Anwendung

* Single-Page-Webanwendung.
* Unterstütze Browser: Firefox, Chrome.
* Lokales Puffern der Messwerte, der letzten abgerufenen Station.
* Lokale Speicherung der Stationen.
* Maximal eine Sekunde Reaktionszeit der Oberfläche.
* Wenn nach 5 Sekunden keine Messdaten angezeigt werden konnten, wird eine Fehlermeldung angezeigt.

**Softwarearchitektur**

Die Architektur besteht aus einem **Client**, einem **Server** (Webserver und Backend) und einer **externen Datenquelle** (Global Historical Climatology Network). Ob die Trennung von Frontend und Backend erfolgt, hängt von der erzielten Performance ab.  
Der **Client** kommuniziert über HTTP mit dem **Webserver** und zeigt empfangene Daten in der Benutzeroberfläche an.  
Im **Backend** wird die Haupt-Geschäftslogik ausgeführt, inklusive dem Zugriff auf eine lokale **Stationsliste**.  
Zur Anforderung historischer Klimadaten ruft das **Backend** die externe Datenquelle auf und bereitet die empfangenen Informationen auf.  
Der **Webserver** übermittelt schließlich die aufbereiteten Daten als HTTP-Response an den Client.

**Ein Bild, das Diagramm, Plan, technische Zeichnung, Rechteck enthält.

KI-generierte Inhalte können fehlerhaft sein.**