

Dokumentation zur Anwendungsentwicklung eines Wetterdaten-Analysetools

Dokumentation Projekt mit Anwendungsentwicklung

Studienjahrgang 2022

Kurs C

Fakultät Wirtschaft

Studiengang Wirtschaftsinformatik

DUALE HOCHSCHULE BADEN-WÜRTTEMBERG

VILLINGEN-SCHWENNINGEN

Bearbeiter:

N. Diehl, S. Dubiel,

Q. Großhauser, A. Iacona, M. Paffen

Betreuer Dozent:

Prof. Dr. Wolfgang Funk



Inhaltsverzeichnis

Inhaltsverzeichnis	1
1 Projektdefinition	1
1.1 Projektsteckbrief	1
1.2 Projektziel	1
1.3 Rahmenbedingungen	1
2 Projektorganisation	2
2.1 Stakeholder	2
2.2 Verantwortlichkeiten	2
2.3 Kommunikationsplan	3
3 Projektplanung	4
3.1 Projektmeilensteine	4
3.2 Vorgangsliste	4
3.3 Netzplan	5
3.4 Risikomanagement	5
3.5 Qualitätssicherung	9
4 Projektdurchführung	13
4.1 Funktionale Anforderungen	13
4.2 Nicht-funktionale Anforderungen	16
4.3 Use-Case „Wetterdaten analysieren“	18
4.4 Handbuch	20
4.4.1 Grafische Benutzeroberfläche (Mockup)	21
4.4.2 Architektur	21
4.4.3 Bedienungsanleitung	24
4.5 Test der Anwendung	28
4.6 Eingesetzte Werkzeuge	31
5 Projektabschluss	34
5.1 Projektabschlussbericht	34
5.2 Erfahrungsbericht	34
6 Anhang	35
7 Selbstständigkeitserklärung	46

1 Projektdefinition

1.1 Projektsteckbrief

Zum Start des Projektes liefert der Projektsteckbrief eine kompakte Übersicht der Projektdaten und dient allen Beteiligten als zentrale Informationsquelle.

Projektname	Anwendungsentwicklung eines Wetterdaten-Analysetools
Auftragnehmer	Gruppe 1
Projektteam	Nikolas Diehl (ND), Simon Dubiel (SD), Quirin Großhauser (QG), Angelo Iacona (AI), Malte Paffen (MP)
Projektleiter	Malte Paffen
Auftraggeber	Prof. Dr.-Ing. Wolfgang Funk
Projektzeitraum	08.01.2025 – 14.03.2025

1.2 Projektziel

Ziel des Projektes ist die Entwicklung einer Softwareanwendung zur Analyse historischer Wetterdaten auf Basis des Daily Global Historical Climatology Network (GHCN). Die Anwendung soll dabei eine benutzerfreundliche grafische Oberfläche bieten, um Messdaten von Wetterstationen zu durchsuchen, zu analysieren und zu visualisieren.

1.3 Rahmenbedingungen

Die folgenden Rahmenbedingungen definieren die zentralen Vorgaben und Einschränkungen des Projektes und legen somit den Projektrahmen fest.

Datenbasis: GHCN

Anwendung: Lauffähig unter Windows 11 (optional WSL)

Architektur: Client-Server-Modell; Server wird in Containern bereitgestellt

Technologien: Python inkl. Flask (Backend), JavaScript und HTML (Frontend)

Systembereitstellung: Docker Containerisierung

2 Projektorganisation

2.1 Stakeholder

Die Stakeholder des Projektes sind alle Personen oder Gruppen, die ein Interesse am Projekt haben oder direkt betroffen sind. Die folgende Tabelle umfasst alle Stakeholder inklusive ihrer Rolle/Funktion innerhalb des Projektes. Der Grad der Betroffenheit sowie die Art der Kommunikation sind ebenfalls dargestellt.

Name	Funktion	Betroffenheit	Kommunikation
ND, SD, QG, AI, MP	Projektteam	hoch	Tägliche Abstimmung des Teams
Malte Paffen	Projektleitung	sehr hoch	Wöchentliche Updates
Prof. Dr. Wolfgang Funk	Kunde	sehr hoch	Wöchentlicher Statusbericht durch Projektleitung
Julian Gommlich	Interner Consultant	mittel	Bei Bedarf (wöchentlich)

2.2 Verantwortlichkeiten

Die klare Zuweisung von Verantwortlichkeiten stellt sicher, dass Aufgaben und Zuständigkeiten innerhalb des Projektes eindeutig definiert sind, womit die Zusammenarbeit langfristig gestärkt wird. Die folgende Tabelle gibt Auskunft über die Verantwortlichkeiten der einzelnen Projektbeteiligten.

Nikolas Diehl	Backend-Entwicklung und Entwicklungscoordination
Simon Dubiel	Projektkoordination, Projektdokumentation und Testing
Quirin Großhauser	Dokumentation und Testunterstützung
Angelo Iacona	Design, Frontend-Entwicklung und technische Dokumentation
Malte Paffen	Technische Frontend Entwicklung und Projektmanagement

2.3 Kommunikationsplan

Für die erfolgreiche Durchführung eines Projektes mit mehreren Beteiligten ist die Kommunikation innerhalb des Teams sowie mit externen Stakeholdern ein entscheidender Erfolgsfaktor. Im Rahmen des Projektes wurden die folgenden Kommunikationswege gewählt.

Microsoft Teams

- Ablage und Synchronisation aller Dokumente und Unterlagen
- Regelmäßiger Austausch über aktuellen Status und Next Steps
- Visuelle Darstellungen über die Aufgabenverteilung im Team

E-Mail

- Externe Kommunikation mit dem Kunden
- Interne Kommunikation mit dem Consultant
- Weiterleitung relevanter Informationen an das Projektteam

Meetings

- Besprechung wichtiger Fortschritte und Hindernisse
- Beratungsgespräche mit internem Consultant über Optimierungen
- Projektabnahme durch Kunden innerhalb eines Meetings

Ad-hoc Kommunikation

- Kurzfristige und spontane Absprachen

3 Projektplanung

3.1 Projektmeilensteine

Projektmeilensteine markieren wichtige Etappen im Projektverlauf und dienen zur kontinuierlichen Fortschrittskontrolle. Die folgende Tabelle gibt einen Überblick über die zentralen Meilensteine von Beginn bis Ende des Projektes.

08.01.2025	Projektbeginn
Woche 0-2	Projektinitialisierung
Woche 2-4	Basis Setup & Prototyping
Woche 4-6	Funktionale Integration
Woche 6-8	Datenvisualisierung
Woche 8-9	Finalisierung und Deployment
17.03.2025	Projektende

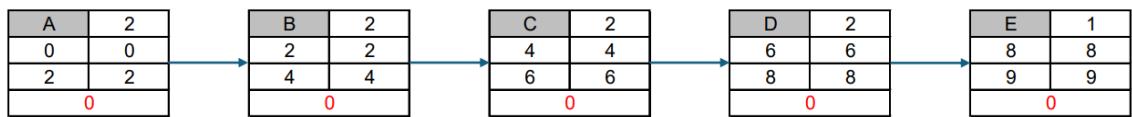
3.2 Vorgangsliste

Die zuvor definierten Meilensteine werden in eine Vorgangsliste überführt, die somit alle notwendigen Aufgaben enthält und für eine strukturierte Planung sorgt. Die folgende Tabelle zeigt die Vorgangsliste, die in Anbetracht des Umfangs des Projektes sehr verständlich und trivial dargestellt ist.

Tätigkeit	Beschreibung	Dauer in Wochen	Vorgänger
A	Projektinitialisierung	2	-
B	Basis Setup & Prototyping	2	A
C	Funktionale Integration	2	B
D	Datenvisualisierung	2	C
E	Finalisierung und Deployment	1	D

3.3 Netzplan

Die Vorgangsliste mit den zentralen Tätigkeiten des Projektes kann in einen Netzplan überführt werden, um den Ablauf der Aktivitäten zu visualisieren. Der folgende Netzplan impliziert, dass aufgrund des engen zeitlichen Projektrahmens, keine Aktivität einen möglichen Puffer erlaubt.



3.4 Risikomanagement

Ein effektives Risikomanagement identifiziert potenzielle Risiken frühzeitig und entwickelt angemessene Maßnahmen, um diesen entgegenzuwirken. Zunächst werden innerhalb einer Risikoübersicht potenzielle Risiken identifiziert, die den Projektverlauf maßgeblich beeinflussen können.

Risikoübersicht

Risikokategorie	Risiko-ID	Beschreibung
Organisation	R1	Verzögerungen bei der Kommunikation mit dem Kunden (MP).
	R2	Missverständnisse oder Unklarheiten bei den Anforderungen.
Team & Ressourcen	R3	Ausfall eines Schlüsselmitglieds (z. B. Krankheit von SD oder ND).
	R4	Überlastung einzelner Teammitglieder aufgrund enger Zeitplanung.
	R5	Koordinationsprobleme im Team (z. B. unklare Verantwortlichkeiten).
Technik (Frontend)	R6	Technische Schwierigkeiten bei der Implementierung der Weltkugel (Three.js).
	R7	Probleme bei der Integration von Backend-Daten ins Frontend.
Technik (Backend)	R8	Verzögerungen bei der Integration der NOAA-API (ND).
	R9	Probleme bei der Datenaufbereitung und Optimierung.
	R10	Unzureichende Testabdeckung im Backend aufgrund des verkürzten Zeitplans.
Technik (CI/CD)	R11	Fehlerhafte CI/CD-Konfiguration oder Docker-Setup.

Dokumentation	R12	Fehlende oder fehlerhafte Dokumentation aufgrund von Zeitdruck.
Deployment	R13	Probleme bei der Veröffentlichung des Docker-Images auf der GitHub Container Registry.

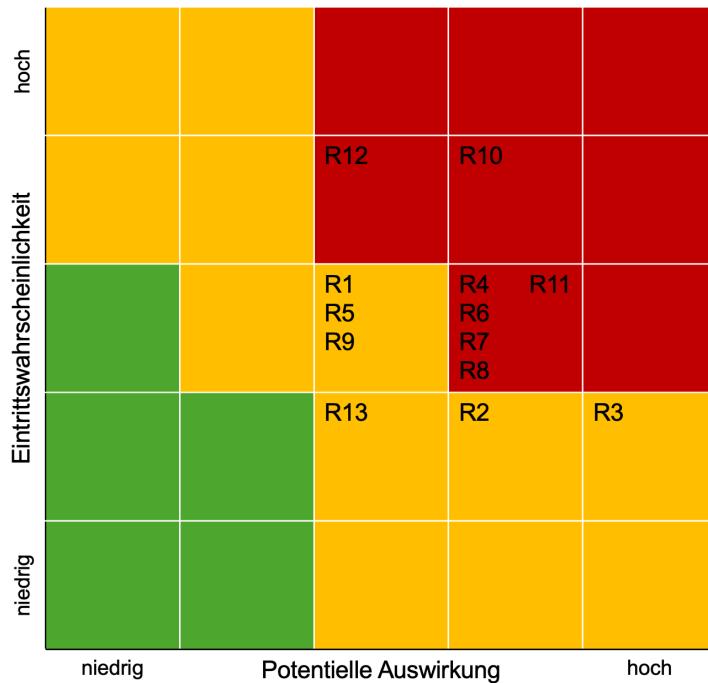
Nach der Identifikation werden die Risiken hinsichtlich ihrer Eintrittswahrscheinlichkeit und potenzieller Auswirkung bewertet. Das Risiko-Level ergibt sich aus der Multiplikation dieser Faktoren. Somit liefert die folgende Risiko-Level-Tabelle die Basis für die Entwicklung von Maßnahmen zur Risikominimierung.

Risiko-Level

ID	Eintrittswahrscheinlichkeit	Potenzielle Auswirkung	Risiko-Level
R1	3	3	9 (Mittel)
R2	2	4	8 (Mittel)
R3	2	5	10 (Hoch)
R4	3	4	12 (Hoch)
R5	3	3	9 (Mittel)
R6	3	4	12 (Hoch)
R7	3	4	12 (Hoch)
R8	3	4	12 (Hoch)
R9	3	3	9 (Mittel)
R10	4	4	16 (Kritisch)
R11	3	4	12 (Hoch)
R12	4	3	12 (Hoch)
R13	2	3	6 (Mittel)

Die Einschätzung über das Risiko-Level der einzelnen Risiken ermöglicht es, die Risiken visuell in einer Risikomatrix darzustellen.

Risikomatrix



Risikomanagementplan

Der Risikomanagementplan sieht Maßnahmen für den Umgang mit den identifizierten Risiken für den Projektverlauf vor. Risikobewältigende Maßnahmen verringern die Eintrittswahrscheinlichkeiten bzw. Auswirkungen oder vermeiden kritische und hohe Risiken. Überwachungsmaßnahmen stellen die regelmäßige Prüfung des Status mittlerer Risiken sicher. Die Beschreibung des Umgangs mit niedrigen Risiken entfällt mangels Risiken mit entsprechendem Risiko-Level. Die Verantwortlichkeiten für einzelne Risiken sind im Risikomanagementplan aufgeführt.

Maßnahmen für kritische und hohe Risiken (Risiko-Level >= 12)

ID	Maßnahme	Ziel	Verant-wortlich	Frist
R4	Aufgabenverteilung überprüfen und Zeitpuffer für kritische Aufgaben einplanen.	Überlastung vermeiden	MP	Laufend

R6	Proof-of-Concept für kritische Three.js-Funktionen frühzeitig umsetzen.	Technische Risiken minimieren	AI	Woche 2
R7	Tests für Schnittstellen zwischen Frontend und Backend planen und frühzeitig durchführen.	Integration sicherstellen	AI, SD, ND	Woche 4
R8	API-Integration priorisieren und mit Mock-Daten arbeiten, um frühzeitig Daten aufzubereiten.	Verzögerungen vermeiden	ND	Woche 3
R10	Teststrategien für kritische Backend-Funktionen priorisieren und regelmäßig prüfen.	Qualität des Backends sicherstellen	SD, ND	Woche 5
R11	Manuelle Tests der CI/CD-Pipeline vor Automatisierung durchführen, um Fehler zu vermeiden.	Fehler in der Pipeline beheben	MP, SD	Woche 6
R12	Zwischenstände der Dokumentation regelmäßig überprüfen und Feedback einholen.	Dokumentationsqualität sicherstellen	QG	Laufend

Maßnahmen für mittlere Risiken (Risiko-Level 6-11)

ID	Maßnahme	Verantwortlich	Frequenz
R1	Regelmäßige Rückmeldungen an das Team über den Status der Kundenkommunikation.	MP	Wöchentlich
R2	Anforderungen schriftlich festhalten und frühzeitig vom Kunden bestätigen lassen.	MP	Vor jedem Sprint

R3	Dokumentation und Übergabepläne erstellen, um Wissensverlust bei Krankheit zu vermeiden.	Alle	Laufend
R5	Regelmäßige Abstimmungen zwischen den Gruppenmitgliedern sichern das rechtzeitige Erkennen von Missverständnissen und Abstimmungsbedarf.	MP bzw. alle	Mindestens wöchentlich
R9	Aufgaben mit geringerer Priorität verschieben, um Zeit für Tests zu gewinnen.	SD, ND	Nach jedem Sprint
R13	Parallelisierung der Veröffentlichung mit Testumgebungen sicherstellen.	MP, SD	Vor Veröffentlichung

Niedrige Risiken (Risiko-Level <= 5)

Risiken mit einem Risiko-Level kleiner oder gleich 5 werden im Projektverlauf zunächst akzeptiert. Die Auswirkungen werden hingenommen. Da keines der 13 ermittelten Risiken ein Risiko-Level kleiner oder gleich 5 hat sind dementsprechend Maßnahmen für niedrige Risiken zu vernachlässigen.

3.5 Qualitätssicherung

Die Qualitätssicherung stellt sicher, dass das Projekt die funktionalen und nicht-funktionalen Anforderungen vollständig erfüllt. Es gewährleistet die besondere Qualität der Dokumentation und Kommunikation im Projektverlauf sowie vor der Übergabe an den Kunden. Das Qualitätsmanagement zielt auf die reibungslose Übergabe des fertigen Produkts an den Kunden. Dazu stellt es die Qualität der Lieferobjekte sowie die Testabdeckung in geforderter Höhe sicher.

Qualitätsziele

Bereich	Qualitätsziel
Funktionale Anforderungen	Alle definierten funktionalen Anforderungen (ID 0.x) werden vollständig und korrekt implementiert.
Nicht-funktionale Anforderungen	Mindestens 80 % Testabdeckung, Einhaltung von Resourcenlimits (Docker: 2 CPUs, 1 GB RAM).
Dokumentation	Vollständige, konsistente und DIN-konforme Projektdokumentation (DIN 69901).
Codequalität	Sauber strukturierter, dokumentierter und durch Tests validierter Code.
Deployment	Stabiles CI/CD mit GitHub Actions und erfolgreich veröffentlichte Docker-Images.

Qualitätsmanagementprozesse

1. Qualitätsplanung

- Ziel: Definition von Qualitätsanforderungen und Standards für das Projekt.
- Maßnahmen:
 - Festlegung der Anforderungen anhand der funktionalen und nicht-funktionalen Anforderungen (ID 0.x und 1.x).
 - Entwicklung eines Testplans für Unit-, Service- & Integrationstests.
 - Definition von Akzeptanzkriterien für jedes Arbeitspaket.

2. Qualitätssicherung

- Ziel: Laufende Überprüfung, ob die Qualitätsziele eingehalten werden.
- Maßnahmen:
 - Code-Reviews durch Teammitglieder (SD und ND überprüfen gegenseitig).
 - Wöchentliche Statusmeetings zur Besprechung von Fortschritten und Qualitätsproblemen (MP moderiert).
 - Automatisierte Tests (GitHub Actions) zur Sicherstellung der Funktionalität nach jedem Push.

3. Qualitätskontrolle

- Ziel: Überprüfung der Arbeitsergebnisse, um sicherzustellen, dass sie den Anforderungen entsprechen.
- Maßnahmen:
 - Abnahmeprotokolle für abgeschlossene Arbeitspakete (z.B. API-Endpunkte, Frontend-Visualisierungen).
 - Vergleich der Testergebnisse mit den Akzeptanzkriterien.
 - Dokumentationsprüfung durch QG vor Meilenstein- und Projektabschluss.

Qualitätsprüfmethoden

Methode	Beschreibung	Verantwortlich
Code-Review	Überprüfung des Codes auf Lesbarkeit, Einhaltung der Standards und potenzielle Fehler.	SD, ND
Automatisierte Tests	Durchführung von Unit-, Service- und Integrationstests über GitHub Actions.	SD, ND
Manuelle Tests	Testen der Anwendung im Frontend (Three.js-Weltkugel, Tabellen, Diagramme).	AI
Dokumentationsprüfung	Überprüfung der Vollständigkeit und Konsistenz der Dokumentation anhand einer Checkliste.	QG
Review-Meetings	Gemeinsame Besprechung der Ergebnisse und Qualitätsfragen im Team.	MP, alle

QM-Rollen und -Verantwortlichkeiten

Rolle	Verantwortung
MP	Überwachung der Qualitätssicherung und Moderation von Review-Meetings.
QG	Sicherstellung der Dokumentationsqualität und Konsistenz.
SD & ND	Umsetzung der Backend-Qualität (API, Tests) und gegenseitige Code-Reviews.
AI	Sicherstellung der Frontend-Qualität (Three.js, D3.js) und Tests der Visualisierungen.

Qualitätsprüfplan

Meilenstein	Qualitätsmaßnahme	Verantwortlich	Frist
Projekt-initialisierung	Festlegung der Qualitätsziele und Erstellung des Testplans.	MP, QG	Woche 2
Basis Setup & Prototyping	Überprüfung der grundlegenden API- und Frontend-Funktionen durch automatisierte und manuelle Tests.	SD, ND, AI	Woche 4
Funktionale Integration	Durchführung von End-to-End-Tests und Sicherstellung der Backend-Frontend-Kommunikation.	SD, ND, AI	Woche 6
Daten-visualisierung	Überprüfung der korrekten Darstellung und Performance der Visualisierungen durch Tests und Feedback.	SD, ND	Woche 8
Finalisierung & Deployment	Durchführung einer Dokumentationsprüfung und Abnahme des gesamten Systems.	QG, MP	Woche 9

QM-Werkzeuge und -Hilfsmittel

Werkzeug	Zweck
GitHub Actions	Automatisierung der Tests und CI/CD-Prozesse.
Docker	Sicherstellung einer einheitlichen Laufzeitumgebung.
pytest	Durchführung von Unit- und Integrationstests.
Puppeteer, Mocha, und Chai	Unitests für die Logik-Bestandteile im Frontend.

Überwachung und Bericht

Qualitätsmetriken:

- Testabdeckung ($\geq 80\%$).
- Anzahl der offenen und geschlossenen Fehler (Bugs).
- Anzahl der durchgeföhrten Code-Reviews.

Berichtswesen:

- Wöchentliche Statusberichte durch MP und QG.
- Abschlussbericht zur Einhaltung der Qualitätsziele am Projektende

4 Projektdurchführung

4.1 Funktionale Anforderungen

Die folgenden Anforderungen beschreiben zentrale Funktionen des Systems.

Anforderung 0.1: Eingabe von Koordinaten und Radius

Schlüsselwort		Information
ID	0.1	
Art	Funktionale Anforderung	
Anforderungsträger	Auftraggeber	
Anforderung	Nutzer sollen in der Lage sein, geografische Koordinaten (Breite, Länge) und einen Radius einzugeben, um damit eine Eingrenzung der relevanten Wetterstationen vorzunehmen.	
Beschreibung und Kontext	Die Anwendung erfasst die eingegebenen Parameter und leitet diese an das Backend weiter, wo eine Filterung der Daten vorgenommen wird.	
Eingaben und Quellen	Nutzereingaben (Koordinaten, Radius).	
Ausgaben und Ziele	Validierte Weitergabe der Eingaben an das Backend zur Verarbeitung.	
Begründung	Diese Eingabeparameter bilden die Grundlage für die Datenverarbeitung und Filterung relevanter Wetterstationen.	
Klassifikation	Muss	
Priorität	Hoch	
Konflikte	Keine	

Anforderung 0.2: Filterung von Wetterstationen

Schlüsselwort		Information
ID	0.2	
Art	Funktionale Anforderung	
Anforderungsträger	Auftraggeber	
Anforderung	Basierend auf den Eingaben des Nutzers sollen Wetterstationen innerhalb des spezifizierten Radius identifiziert werden.	
Beschreibung und Kontext	Das Backend interagiert mit der NOAA-API, um Wetterstationsdaten zu beziehen, und nutzt Algorithmen wie die Haversine-Formel zur Distanzberechnung, um Stationen im spezifizierten Radius zu ermitteln. Die Verfügbarkeit von	

	Wetterdaten im gewünschten Zeitraum ist anhand des sog Inventories zu ermitteln.
Eingaben und Quellen	Nutzerangaben, NOAA-API („/stations“-Endpunkt).
Ausgaben und Ziele	Eine Liste relevanter Wetterstationen mit ID, Name und Koordinaten.
Begründung	Diese Funktion dient der Identifikation von Wetterstationen, die auf Grundlage geografischer Daten für den Nutzer relevant sind.
Klassifikation	Muss
Priorität	Hoch
Konflikte	Keine

Anforderung 0.3: Auswahl einer Wetterstation

Schlüsselwort	Information
ID	0.3
Art	Funktionale Anforderung
Anforderungsträger	Auftraggeber
Anforderung	Nutzende sollen eine Wetterstation aus der zuvor generierten Liste auswählen können.
Beschreibung und Kontext	Die ausgewählte Station wird genutzt, um weitere Datenabfragen durchzuführen und diese im Kontext historischer Wetterdaten aufzubereiten.
Eingaben und Quellen	Nutzerauswahl (Stations-ID).
Ausgaben und Ziele	Weiterleitung der Stations-ID an das Backend.
Begründung	Diese Auswahl erlaubt eine gezielte Datenanalyse für eine spezifische Station.
Klassifikation	Muss
Priorität	Hoch
Konflikte	Keine

Anforderung 0.4: Abfrage von historischen Wetterdaten

Schlüsselwort	Information
ID	0.4
Art	Funktionale Anforderung
Anforderungsträger	Auftraggeber
Anforderung	Das Backend muss historische Wetterdaten der ausgewählten Wetterstation über die NOAA-API abrufen und diese Daten aggregieren.

Beschreibung und Kontext	Die abgerufenen Rohdaten werden verarbeitet, um Jahresmittelwerte von Temperaturminima und -maxima zu berechnen.
Eingaben und Quellen	Stations-ID, NOAA-API („/data“-Endpunkt).
Ausgaben und Ziele	Strukturierte Daten, die Jahresmittelwerte für die Nutzer visualisieren.
Begründung	Ermöglicht Nutzenden die Analyse von historischen Wetterdaten auf Basis einer spezifischen Station.
Klassifikation	Muss
Priorität	Hoch
Konflikte	Keine

Anforderung 0.5: Darstellung der Wetterdaten

Schlüsselwort	Information
ID	0.5
Art	Funktionale Anforderung
Anforderungsträger	Auftraggeber
Anforderung	Historische Wetterdaten sollen sowohl tabellarisch als auch grafisch dargestellt werden.
Beschreibung und Kontext	<p>Die Datenvisualisierung ermöglicht es Nutzenden, Trends und historische Entwicklungen in Temperaturminima und -maxima intuitiv zu erfassen. Es werden Temperaturmaxima sowie -minima im gewählten Zeitraum jährlich sowie je Jahreszeit ausgegeben.</p> <p>Die Jahreszeiten sind für Standorte auf der Nordhalbkugel wie folgt anzunehmen (je + sechs Monate für Standorte auf der Südhalbkugel): Winter – November des Vorjahres bis Januar, Frühling – Februar bis April, Sommer – Mai bis Juli, Herbst – August bis Oktober.</p>
Eingaben und Quellen	Daten vom Backend
Ausgaben und Ziele	Diagramme und Tabellen zur Visualisierung
Begründung	Eine visuelle Darstellung fördert das Verständnis und die Analyse der historischen Wettertrends.
Klassifikation	Muss
Priorität	Hoch
Konflikte	Keine

4.2 Nicht-funktionale Anforderungen

Die folgenden Anforderungen legen die Qualitätsmerkmale des Systems fest.

Anforderung 1.1: Testautomatisierung und Testabdeckung

Schlüsselwort		Information
ID	1.1	
Art	Nicht-funktionale Anforderung	
Anforderungsträger	Auftraggeber	
Anforderung	Es wird eine Testabdeckung von mindestens 80% gefordert. Zusätzliche entspricht der Code den Anforderungen des sog. Clean Codes.	
Beschreibung und Kontext	Durch die Implementierung von Tests wird sichergestellt, dass die Funktionalität der Anwendung stabil bleibt.	
Eingaben und Quellen	Quellcode, Testframeworks (z. B. pytest)	
Ausgaben und Ziele	Automatisierte Berichte über Testergebnisse	
Begründung	Eine hohe Testabdeckung reduziert die Wahrscheinlichkeit von Fehlern und verbessert die Qualität des Codes.	
Klassifikation	Muss	
Priorität	Hoch	
Konflikte	Keine	

Anforderung 1.2: Nutzung von Docker zur Containerisierung

Schlüsselwort		Information
ID	1.2	
Art	Nicht-funktionale Anforderung	
Anforderungsträger	Auftraggeber	
Anforderung	Die Anwendung wird in Docker Containern zur Verfügung gestellt.	
Beschreibung und Kontext	Die Containerisierung stellt sicher, dass die Anwendung konsistent auf verschiedenen Umgebungen läuft und einfach zu deployen ist.	
Eingaben und Quellen	Dockerfile, Docker-Compose	
Ausgaben und Ziele	Lauffähige Docker-Container	
Begründung	Ermöglicht eine einfache Bereitstellung und Wartung der Anwendung.	
Klassifikation	Muss	
Priorität	Hoch	
Konflikte	Keine	

Anforderung 1.3: Nutzung von GitHub und CI/CD-Pipeline

Schlüsselwort		Information
ID	1.3	
Art	Nicht-funktionale Anforderung	
Anforderungsträger	Auftraggeber	
Anforderung	GitHub wird als Versionskontrollsystem verwendet. Eine automatisierte CI/CD-Pipeline stellt sicher, dass Tests und Builds bei jedem Push ausgeführt werden.	
Beschreibung und Kontext	GitHub Actions wird genutzt, um den Entwicklungsprozess zu automatisieren und die Codequalität kontinuierlich zu überprüfen.	
Eingaben und Quellen	GitHub, GitHub Actions	
Ausgaben und Ziele	Automatisierte Builds und Testergebnisse	
Begründung	Die Automatisierung verbessert die Effizienz und Qualitätssicherung des Entwicklungsprozesses.	
Klassifikation	Soll	
Priorität	Mittel	
Konflikte	Keine	

Anforderung 1.4: Ressourcenbegrenzung in Containern

Schlüsselwort		Information
ID	1.4	
Art	Nicht-funktionale Anforderung	
Anforderungsträger	Auftraggeber	
Anforderung	Jeder Docker-Container ist so zu konfigurieren, dass er maximal 2 CPUs und 1 GB RAM verwendet.	
Beschreibung und Kontext	Die Begrenzung der Ressourcen stellt sicher, dass die Anwendung ressourcenschonend arbeitet und die Stabilität der Betriebsumgebung nicht beeinträchtigt.	
Eingaben und Quellen	Docker-Compose Konfiguration	
Ausgaben und Ziele	Container, die die spezifizierten Ressourcenbegrenzungen einhalten.	
Begründung	Stellt sicher, dass die Systemressourcen effizient genutzt werden.	
Klassifikation	Muss	
Priorität	Hoch	
Konflikte	Keine	

4.3 Use-Case „Wetterdaten analysieren“

Ein Use-Case (Anwendungsfall) ist eine Ablaufbeschreibung um Anforderungen strukturiert zu erfassen. Er dient als Input für Analyse und Design. Im Folgenden wird der Use-Case des Wetterdaten-Analysetools vorgestellt.

Überblick:

Ziel ist die Abfrage von Wetterdaten und Statistiken mittels einer Web-Anwendung. Ein Benutzer kann nach Eingabe beliebiger Koordinaten sowohl die in der Nähe liegenden Wetterstationen auf einer Karte erkennen, sowie die entsprechenden Wetterdaten der Stationen einsehen.

Geltungsbereich:

Der Anwendungsfall gehört zur Entwicklung einer Software innerhalb des Kurses „Projekt mit Anwendungsentwicklung“ und zielt auf die Auswertung meteorologischer Daten auf Basis von eingegebenen Koordinaten ab.

Anwendungsschicht:

Benutzer

Primärer Akteur:

Benutzer (Endnutzer der Web-Anwendung)

Weitere Akteure:

- System
- Wetterdatenanbieter (GHCN)
- Professor (Kunde)
- Consultant (interner Consultant)

Stakeholder und ihre Belange:

- Wissenschaftler und Meteorologen: zur Erhebung der Wetterdaten
- Umweltforscher: Überwachung klimatischer Veränderungen
- Endnutzer: Abruf aktueller Wetterdaten aus verschiedenen Regionen
- Professor: Möchte als Kunde die Anwendung funktionsfähig erhalten
- Consultant: Arbeitskollege mit Interesse an Erfolg des Projektes

Vorbedingungen:

- Der Benutzer hat Zugriff auf die Anwendung
- Die Wetterdatenbank ist erreichbar und erhält die korrekten Daten
- Der Server ist funktionsfähig und stellt die Daten in richtiger Weise dar

Nachbedingungen:

- Die angeforderten Wetterdaten wurden erfolgreich abgerufen und angezeigt

Hauptzweig:

1. Der Benutzer gibt geografische Koordinaten (Länge und Breite) ein
2. Das System sucht in der GHCN-Datenbank nach passenden anliegenden Wetterstationen und zeigt diese an
3. Der Benutzer wählt eine der angebotenen Stationen aus
4. Das System ruft die historischen Wetterdaten der gewählten Stationen aus
5. Die Stadt wird mithilfe der Karte angezeigt
6. Die Wetterdaten der Station werden unten in tabellarischer Form angezeigt
7. Der Benutzer kann eine weitere Wetterstation auswählen oder neue Koordination auswählen

Erweiterungen:

1. Es wird keine Station in der Umgebung gefunden: Es werden keine Wetterstationen angezeigt und der Benutzer kann neue Koordinaten eingeben.
2. Der Nutzer kann anstelle von Koordinaten Städtenamen oder Postleitzahlen eingeben und bekommt passende Stationen zur Auswahl gestellt.

Besondere Anforderungen:

- Die Benutzeroberfläche muss intuitiv und responsiv sein
- Die Suchparameter (Radius, Zeitraum) müssen anpassbar sein

- Die Anwendung muss eine effiziente Datenverarbeitung und Darstellung
- Die Anwendung muss containerisiert bereitgestellt werden, um eine einfache Deployment- und Skalierbarkeit zu ermöglichen

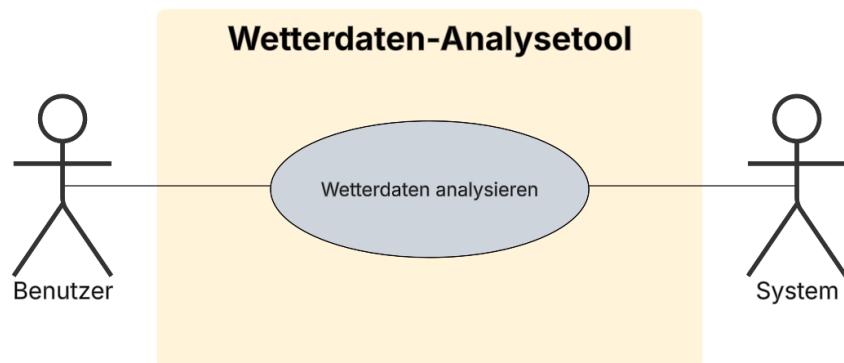
Variationen in Bezug auf Technologie und Daten:

- Die Regionen der Wetterstationen werden mithilfe einer Karte angezeigt

Häufigkeit des Auftretens:

- Erwartet wird eine regelmäßige (tägliche) Nutzung des Benutzers
- Wissenschaftliche Nutzer könnten periodisch größere Datenmengen abrufen und für weiter Zwecke nutzen

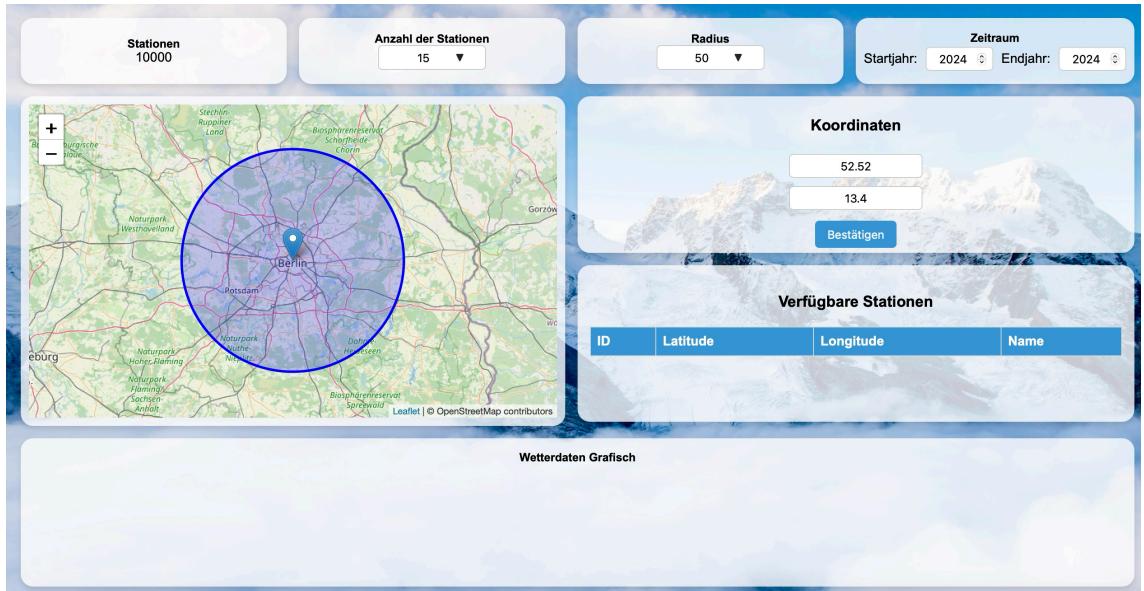
Use-Case Diagramm



4.4 Handbuch

Das Handbuch spielt eine elementare Rolle im Projekt, da es alle wichtigen Aspekte der Anwendung dokumentiert und eine klare Orientierung für Entwickler und Nutzer schafft. Zunächst enthält es ein Mockup der Software, das die Benutzeroberfläche veranschaulicht. Die Beschreibung der Systemarchitektur führt zu einer verständlichen Darlegung der technischen Struktur. Eine ausführliche Bedienungsanleitung erleichtert die Nutzung und ermöglicht eine schnelle Einarbeitung. Abschließend werden mögliche zukünftige Erweiterungen vorgestellt. Die Code-Dokumentation, in der alle Funktionen beschrieben werden, ist als readme-Datei im Github Repository abgelegt.

4.4.1 Grafische Benutzeroberfläche (Mockup)



4.4.2 Architektur

Die Architektur folgt einer Client-Server-Struktur, bestehend aus:

1. Frontend (Client)
2. Backend (Server)
3. Externe API (NOAA API)
4. CI/CD und Containerisierung (GitHub Actions + Docker)

Komponenten und Schichtenarchitektur

Im Folgenden findet sich ein Überblick der wesentlichen funktionalen Bestandteile dieses Produktes. Im Repository sind detaillierte Erläuterungen sämtlicher Bestandteile zu finden, s. „./users_guide“ und „readme.md“.

Die Architektur ist in vier Hauptkomponenten unterteilt:

1. Frontend (Benutzeroberfläche)
 - Technologien: HTML, CSS, JavaScript (D3.js), OpenStreetMap.
 - Verantwortlich für:
 - Darstellung der Karte mit Wetterstationen.
 - Benutzeroberfläche zur Eingabe von Koordinaten und Radius.
 - Anzeige von Wetterstationsdaten in Tabellen & einem Diagramm.

- Kommunikation mit Backend:
 - Sendet API-Anfragen an das Flask-Backend.
 - Erhält Antworten z.B. in Form von Wetterdaten oder La-destatus.
- Empfang von Wetterdaten zur Visualisierung.
- Berechnung und Verarbeitung aggregierter Wetterdaten zur Dar-stellung von Mittelwerten.

2. Backend (Flask API-Server)

- Technologien: Python (Flask), Flask-CORS, Requests, Pandas, io, math und threading.
- Verantwortlich für:
 - Empfang und Verarbeitung der Nutzereingaben.
 - Kommunikation mit der NOAA API, um Wetterstations- und Wet-terdaten abzurufen.
 - Datenverarbeitung mit Pandas: Filterung und Formatierung der Wetterstations- und Wetterdaten
 - Bereitstellung von REST-APIs für das Frontend - API-Endpunkte:
 - **/preload_status:** Dieser Endpunkt liefert eine JSON-Ant-wort, die anzeigt, ob das Vorladen (Preloading) der Sta-tions- und Inventardaten abgeschlossen ist. Er gibt entwe-dter {"status": "loading"} oder {"status": "done"} zurück.
 - **/:** Der Root-Endpunkt rendert das Template index.html und übergibt die Anzahl der geladenen Stationen (sta-tion_count) als Kontext.
 - **/get_stations:** Ein GET-Endpunkt, der über Query-Para-meter (wie latitude, longitude, radius_km, station_count, start_year und end_year) die Stationen räumlich sowie an-hand des Inventory-Vezeichnisses filtert. Er liefert eine JSON-Liste der Stationen, die den Kriterien entsprechen.
 - **/get_weather_data:** Ein GET-Endpunkt, der die Wetterda-ten einer bestimmten Station abruft. Er erwartet Parameter

- wie station_id, start_year und end_year und gibt die gefilterten Wetterdaten im JSON-Format zurück.
 - **/error** (nur verfügbar, wenn app.config.get("TESTING") wahr ist): Ein Test-Endpunkt, der absichtlich eine Ausnahme auslöst, um einen Fehler zu simulieren.
 - Datenfluss:
 - Nutzereingaben → Backend verarbeitet Koordinaten → NOAA API Anfrage → Aufbereitung → Rückgabe ans Frontend.
3. Externe API (NOAA National Climate Data Center API)
- Basis-URL: <https://www.ncdc.noaa.gov/cdo-web/api/v2/>
 - Genutzte API-Endpunkte:
 - ghcnd-stations.txt
 - URL:
<https://www.ncdc.noaa.gov/pub/data/ghcn/daily/ghcnd-stations.txt>
 - Liefert Wetterdaten für eine ausgewählte Station.
 - ghcnd-inventory.txt
 - URL:
<https://www.ncdc.noaa.gov/pub/data/ghcn/daily/ghcnd-inventory.txt>
 - Dieses Inventar enthält Informationen zu den verfügbaren Messgrößen (wie TMIN und TMAX) und deren Zeiträumen.
 - by_station/{station_id}.csv
 - URL:
https://www.ncdc.noaa.gov/pub/data/ghcn/daily/by_station/{station_id}.csv
 - Hier werden die Wetterdaten einer spezifischen Station im CSV-Format abgerufen.
 - all//{station_id}.dly
 - URL:
https://www.ncdc.noaa.gov/pub/data/ghcn/daily/all/{station_id}.dly

- Dieser Endpunkt wird als Fallback genutzt, wenn die CSV-Daten nicht verfügbar sind; er liefert die Wetterdaten im DLY-Format.
- Datenfluss:
 - Flask sendet HTTP-Requests → NOAA API liefert Rohdaten → Backend verarbeitet die Daten → Frontend zeigt die Ergebnisse an.

4. CI/CD und Containerisierung (GitHub Actions + Docker)

- Containerisierung mit Docker
- Flask-Backend wird in einem Docker-Container ausgeführt.
- Das Frontend wird direkt vom Flask-Backend gehostet.
- CI/CD-Pipeline mit GitHub Actions:
 - Automatischer Build & Push des Docker-Images nach GHCR.
 - Build & Push nach GHCR
 - Automatischer Test der Funktionsfähigkeit der App bei Deployment durch den Docker Container.
 - Automatisierter Testlauf mit:
 - Python Unitests und
 - Formatchecks bzw. automatischer Formatierung via Black und flake8 sowie
 - Javascript Unitests via virtuellen Browsers: Es werden ausschließlich die logikrelevanten Kernfunktionen "process-WeatherData" und "fillMissingYears" getestet.

4.4.3 Bedienungsanleitung

Diese Anleitung zeigt in einfachen Schritten, wie ein Nutzer das Wetterdaten-Analysetool (Anwendung) einfach und unkompliziert bedienen kann. Ergänzend findet sich eine Vorführung der Anwendung im Bewegtbildformat im Repository unter „/users_guide“.

1. Aufruf der Anwendung als Docker-Image

Hat der Nutzer Docker Desktop installiert, kann er sich im Terminal innerhalb des Docker-Fensters beim github Container Registry (ghcr) anmelden. Dazu dient dieser Terminal-Befehl (zu ergänzen um username und pat): `{"docker login ghcr.io -u “[github-username]” -p “[personal access token]”}`.

Nach erfolgreicher Anmeldung kann das Image abgerufen werden. Dazu dient der folgende Terminal-Befehl: `{curl -sL https://raw.githubusercontent.com/simondubiel/projekt_funk/e31090df59572f9212903ee1172e5d1dd11de777/docker-compose.yml -o docker-compose.yml && docker compose up --build -d}`.

Das Image wird unter „Images“ angezeigt. Mit einem Click auf den Pfeil in der entsprechenden Zeile (Play-Symbol) öffnet sich ein Fenster. Hier genügt ein weiterer Click auf „Run“.

Der Container wird gestartet und wird unter „Container“ angezeigt. In der entsprechenden Zeile wird der Port „8080:8080“ angezeigt. Mit einem Click öffnet sich dieser sog. Localhost. Die Anwendung öffnet sich in einem Browserfenster. Das Neuladen der Seite führt immer zum Zurücksetzen der erfolgten Eingaben in der App (Neustart der Benutzeroberfläche). Für einen Neustart der Anwendung insgesamt muss der Port oder respektive der Container gestoppt und erneut gestartet werden.

2. Standort festlegen

Der Nutzer gibt die gewünschten Koordinaten (Breiten- und Längengrad) ein oder wählt einen Ort mit Klick auf die Karte aus. Im zweiten Fall werden die Koordinaten des Standorts automatisch für die Auswahl übernommen.

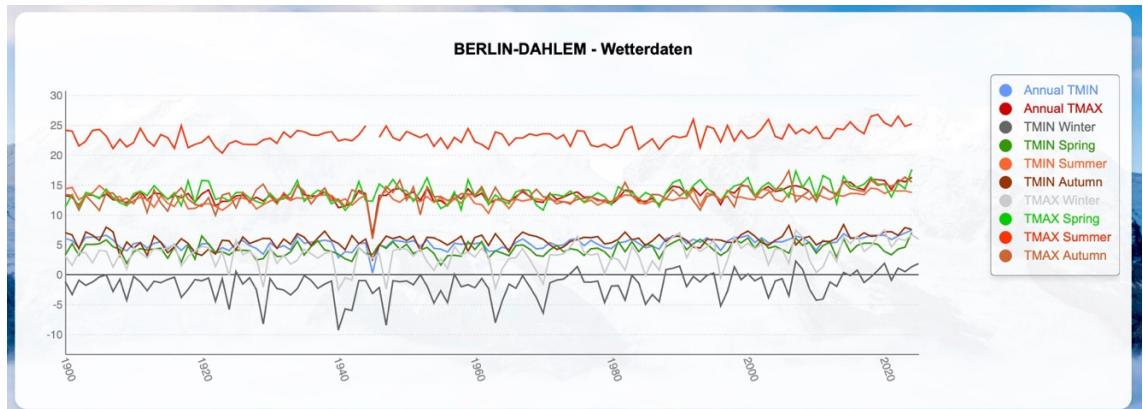
3. Filter einstellen

Der Benutzer kann mehrere Filterkriterien definieren:

- Wie viele Wetterstationen sollen höchstens angezeigt werden (0-10)?
- Wie groß soll der Suchbereich um den Standort sein (Radius in Kilometern 0-100)?
- Für welchen Zeitraum sollen angezeigte Wetterstationen Daten bereitstellen (Start- und Endjahr)?

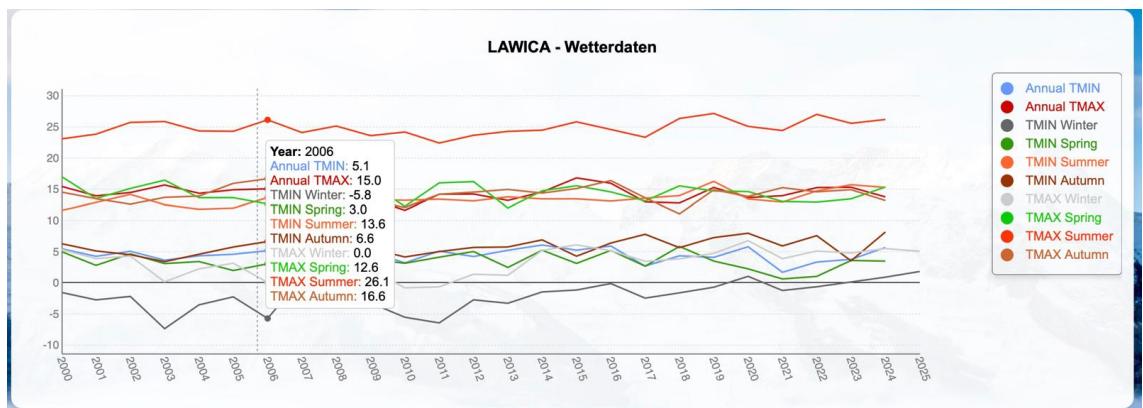
4. Daten anzeigen

Mit dem Klick auf „Bestätigen“ kann der Nutzer die Wetterdaten laden und schließlich anzeigen lassen. Die Daten erscheinen in Tabellen (mit Durchschnittswerten) und einem Diagramm (mit Temperaturverläufen).

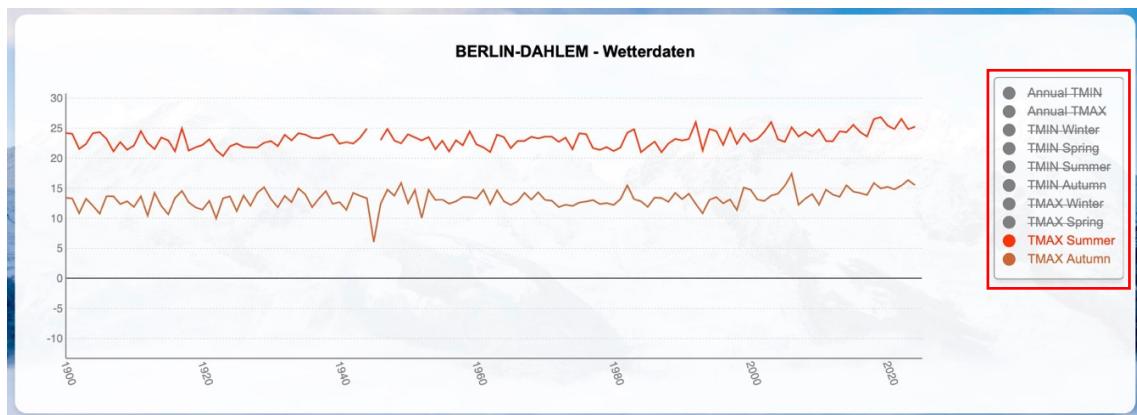


5. Das Diagramm nutzen

Durch Bewegen der Maus können einzelne Details eingesehen werden.



Per Klick in der Legende können einzelne Linien ein- oder ausgeblendet werden. Jeder Klick auf einen Eintrag schaltet die Sichtbarkeit der entsprechenden Temperaturdaten (wie beispielsweise TMIN, TMAX oder saisonale Werte) um.



Mögliche zukünftige Erweiterungen

Caching (Redis/Memcached)

→ API-Anfragen zwischenspeichern, um Ladezeiten zu reduzieren.

Separate Frontend-Container mit Nginx

→ Falls das Frontend komplexer wird.

Datenbank hinzufügen

→ Falls Wetterdaten länger gespeichert werden sollen.

Erweiterte API-Funktionen

→ Mehr Wetterdaten (z. B. Niederschlag, Windgeschwindigkeit).

4.5 Test der Anwendung

In den bisherigen Unit-Tests haben wir mittels Mocking-Techniken sichergestellt, dass die Logik unseres Codes korrekt funktioniert, indem wir API-Antworten simuliert und verschiedene Szenarien überprüft haben. Um jetzt auch die tatsächliche Anbindung an das Internet und die NOAA-API zu validieren, führen wir nun drei Live-Tests durch, die den Echtbetrieb unter realen Bedingungen testen.

Zum Testen der Funktionalität der Anwendung wurden drei verschiedene Standorte gewählt. Grundlage für den Test ist das Starten der Anwendung. Im Folgenden wird die Durchführung der Tests ausführlich beschrieben. Screenshots der Anwendung, sowie zur Überprüfung der Werte anhand der Originaldaten sind dem Anhang beigefügt.

Die heruntergeladenen Wetterdaten aus der GHCN-Datenbank enthalten zahlreiche Wetterstationen mit sehr umfassenden Datensätzen. Um nur die relevanten Daten für eine spezifische Station wie "GME00102380" für das Jahr 1950 herauszufiltern, wurde der folgende Befehl im Terminal (MacOS) ausgeführt:

```
grep "GME00102380" 1950.csv > GME00102380_1950.csv
```

Dadurch entsteht eine neue Datei (GME00102380_1950.csv), welche ausschließlich die benötigten Wetterdaten dieser genauen Station enthält und somit eine gezielte Auswertung ermöglicht. Um die Ergebnisse mit der Anwendung vergleichen zu können, wurden die Daten aus der Datenbank in einer Excel aufbereitet (siehe Screenshots im Anhang). Diese wurden anschließend strichprobenartig auf TMIN, TMAX und verschiedenen Jahreszeiten gefiltert, um die Werte mit der Anwendung zu vergleichen.

1. Berlin

Testschritt	Beschreibung
Auswahl der Koordinaten	Latitude: 52.51350 Longitude: 13.40036 <i>(ausgewählt durch Klick auf interaktiver Karte)</i>
Auswahl der Parameter	Anzahl der Stationen: 10 Radius: 100 Zeitraum: 1920-2024
Ausgabe der Stationen	GM000003319: BERLIN-DAHLEM GM000003342: POTSDAM GM000010393: LINDENBERG
Anhang 2.1: Zeigt die Ausgabe der oben genannten Koordinaten, Parameter und Stationen.	
Anhang 2.2: Die ausgegebenen Werte der Wetterstation GM000003319 (Berlin-Dahlem) im Jahr 1920 wurden auf ihre Korrektheit überprüft. In den beigefügten Excel-Screenshots ist ersichtlich, dass die aus der GHCN-Datenbank extrahierten TMIN- und TMAX-Werte mit den Werten der Anwendung übereinstimmen.	
Anhang 2.3: Die ausgegebenen Werte der Wetterstation GM000003319 (Berlin-Dahlem) im Jahr 1945 wurden auf ihre Korrektheit überprüft. In den beigefügten Excel-Screenshots ist ersichtlich, dass die aus der GHCN-Datenbank extrahierten	

TMIN- und TMAX-Werte für Frühling und Herbst mit den Werten der Anwendung übereinstimmen.

Anhang 2.4:

Die ausgegebenen Werte der Wetterstation GM000003319 (Berlin-Dahlem) im Jahr 2008 wurden auf ihre Korrektheit überprüft. In den beigefügten Excel-Screenshots ist ersichtlich, dass die aus der GHCN-Datenbank extrahierten TMIN- und TMAX-Werte für Frühling und Herbst mit den Werten der Anwendung übereinstimmen.

2. Nürnberg

Testschritt	Beschreibung
Auswahl der Koordinaten	Latitude: 49.44277 Longitude: 11.07602 <i>(manuelle Eingabe der Koordinaten)</i>
Auswahl der Parameter	Anzahl der Stationen: 10 Radius: 100 Zeitraum: 1920-2024
Ausgabe der Stationen	GME00102380: NURNBERG GMM00010761: WEISSENBURG GM000004063: BAMBERG

Anhang 3.1:

Zeigt die Ausgabe der oben genannten Koordinaten, Parameter und Stationen.

Anhang 3.2:

Die ausgegebenen Werte der Wetterstation GME00102380 (Nürnberg) im Jahr 1920 wurden auf ihre Korrektheit überprüft. In den beigefügten Excel-Screenshots ist ersichtlich, dass die aus der GHCN-Datenbank extrahierten, TMIN- und TMAX-Werte mit den Werten der Anwendung übereinstimmen.

Anhang 3.3:

Die ausgegebenen Werte der Wetterstation GME00102380 (Nürnberg) im Jahr 1950 wurden auf ihre Korrektheit überprüft. In den Screenshots der Anwendung ist zu erkennen das für diesen Zeitraum keine Daten zur Verfügung stehen. Die beigefügten Excel-Screenshots zeigen ebenso, dass keine Werte für diesen Zeitraum in der GHCN-Datenbank vorhanden sind. Dies beweist, dass die Werte der Anwendung mit der GHCN-Datenbank übereinstimmen.

Anhang 3.4:

Die ausgegebenen Werte der Wetterstation GME00102380 (Nürnberg) im Jahr 2003 wurden auf ihre Korrektheit überprüft. In den beigefügten Excel-Screenshots ist ersichtlich, dass die aus der GHCN-Datenbank extrahierten, TMIN- und TMAX-Werte für Frühling und Herbst mit den Werten der Anwendung übereinstimmen.

3. Windhoek

Testschritt	Beschreibung
Auswahl der Koordinaten	Latitude: -22.57851 Longitude: 17.07297 <i>(ausgewählt durch Klick auf interaktiver Karte)</i>
Auswahl der Parameter	Anzahl der Stationen: 10 Radius: 100 Zeitraum: 1980-2024
Ausgabe der Stationen	WA007401240: WINDHOEK

Anhang 4.1:
Zeigt die Ausgabe der oben genannten Koordinaten, Parameter und Stationen.

Anhang 4.2:
Die ausgegebenen Werte der Wetterstation WA007401240 (Windhoek) im Jahr 1980 wurden auf ihre Korrektheit überprüft. In den beigefügten Excel-Screenshots ist ersichtlich, dass die aus der GHCN-Datenbank extrahierten, TMIN- und TMAX-Werte mit den Werten der Anwendung übereinstimmen.

Anhang 4.3:
Die ausgegebenen Werte der Wetterstation WA007401240 (Windhoek) im Jahr 2020 wurden auf ihre Korrektheit überprüft. In den beigefügten Excel-Screenshots ist ersichtlich, dass die aus der GHCN-Datenbank extrahierten, TMIN- und TMAX-Werte für Frühling und Herbst mit den Werten der Anwendung übereinstimmen.

Anhang 4.4:
Test des Graphen; über die Legende wird nur TMAX und TMIN ausgegeben.

Anhang 4.5:
Test des Graphen; es werden alle Werte außer TMIN und TMAX gezeigt.

4.6 Eingesetzte Werkzeuge

Im Rahmen des Projekts wurden mehrere Werkzeuge wie Programmiersprachen und Technologien eingesetzt, um eine leistungsfähige, interaktive und datengetreue Web-Applikation zu erstellen. Für das Backend wurde Python mit Unterstützung des Flask-Frameworks eingesetzt, um Daten abzurufen, zu verarbeiten und als API bereitzustellen. Auf der Clientseite wurde JavaScript genutzt, wobei Bibliotheken wie Leaflet und D3.js verwendet wurden, um interaktive Karten,

Diagramme und dynamische Benutzeroberflächen zu erstellen. HTML und CSS bilden das Grundgerüst für die Gestaltung der Weboberfläche. HTML sorgt für die Struktur und CSS für das Layout und Styling.

Python

Technik/Bibliothek	Beschreibung
Flask	Web-Framework zur Erstellung von Routen, Rendering von Templates und Bereitstellung von JSON-Files.
Flask-CORS	Erweiterung zur Unterstützung von Cross-Origin Resource Sharing (CORS) für externe Requests.
Requests	Bibliothek zum Durchführen von HTTP-Anfragen (z. B. zum Abrufen von NOAA-Daten).
Pandas	Datenanalyse und -manipulationsbibliothek zur Verarbeitung und Filterung von Station- und Inventardaten.
io.StringIO	Ermöglicht das Einlesen von Textdaten als file-like Objekt, um diese mit Pandas zu verarbeiten.
math (radians, sin, cos, sqrt, atan2)	Standard-Python-Modul für mathematische Berechnungen.
threading	Ermöglicht das parallele Ausführen von Aufgaben (z. B. Vorladen der Daten im Hintergrund).

Java Script Code

Technik/Bibliothek	Beschreibung
DOM-Methoden	Nutzung von document.getElementById, querySelector, createElement zur Manipulation des HTML-DOM.
Fetch API	Asynchrone HTTP-Anfragen (AJAX) zum Abrufen von Daten vom Server.
Asynchrone Programmierung (async/await, Promises)	Ermöglicht den nicht-blockierenden Abruf und die Verarbeitung von Daten.
Leaflet	JavaScript-Bibliothek für interaktive Karten, Marker, Kreise und Kartenfokus.
D3.js	Bibliothek zur Erstellung von datengetriebenen Diagrammen und Visualisierungen (SVG-basierte Charts).

Event Handling	Nutzung von addEventListener zur Reaktion auf Klicks, Input-Events und andere Benutze rinteraktionen.
Template Literals	Ermöglicht das Einfügen von Variablen in Strings (z. B. für dynamische Überschriften).

5 Projektabschluss

5.1 Projektabschlussbericht

Das Projekt zur Anwendungsentwicklung eines Wetterdaten-Analysetools wurde im Zeitraum vom 08.01.2025 bis zum 14.03.2025 erfolgreich durchgeführt und abgeschlossen. Ziel war die Bereitstellung einer Web-Applikation zur Analyse historischer Wetterdaten basierend auf der GHCN-Datenbasis.

Alle funktionalen Anforderungen aus Kapitel 4.1 wurden erfolgreich umgesetzt, darunter insbesondere die Filterung von Wetterstationen, die Abfrage historischer Wetterdaten und deren grafische Darstellung. Ebenfalls wurden die aus Kapitel 4.2 nicht funktionalen Anforderungen, wie die Testabdeckung von über 80% und eine begrenzte Ressourcennutzung erfüllt.

Dennoch sind während der Projektdurchführung einige der in Kapitel 3.4 ermittelten Risiken aufgekommen. So führte der kurzzeitige krankheitsbedingte Ausfall des Projektleiters Malte Paffen zu organisatorischen Schwierigkeiten wie Missverständnissen innerhalb des Teams, verzögerte Kundenkommunikation und Überlastung der Projektmitglieder. Auch technische Schwierigkeiten, wie die Implementierung der Weltkugel oder unzureichende Testabdeckung im Backend konnten zunächst nicht vermieden werden. Durch umfassende Qualitätssicherheitsmethoden und Maßnahmen zur Risikobehandlung konnten die Risiken jedoch rechtzeitig abgedeckt werden. Das finale Produkt wurde dem Kunden zeitgerecht überreicht.

5.2 Erfahrungsbericht

Das Projekt hat allen Projektbeteiligten gezeigt, wie essenziell klare Kommunikation und strukturierte Planung sind. Regelmäßige Meetings und eine zentrale Dokumentation ermöglichen eine effiziente Zusammenarbeit. Die flexible Aufgabenverteilung half bei dem Auffangen von Engpässen. Frühzeitiges Testen und iterative Optimierungen waren entscheidend für die technische Umsetzung. Insgesamt waren die klaren Verantwortlichkeiten, regelmäßige Abstimmungen und ein strukturierter Entwicklungsprozess die wichtigsten Erfolgsfaktoren.

6 Anhang

Anhang 1: Meeting Protokolle

Datum	Dauer	Kategorie	Ort	Teilnehmer	Themen
08.01.25	0:20h	Interne Absprache	Virtuell	Alle	Gruppenorganisierung, Besprechung der Aufgabenstellung und Setup aller relevanten Anwendungen (VSC, Github usw.).
20.01.25	2:30h	Gemeinsame Gruppenarbeit	Kursraum	Nikolas, Malte	Allgemeine Projektplanung und Besprechung der Vorgehensweise. Definieren von Meilensteinen und Einteilung von Verantwortlichen. Arbeit an Frontend und Backend.
22.01.25	1:00h	Gemeinsame Gruppenarbeit	Kursraum	Alle	Festlegung der Grundstruktur für den Call mit dem Kunden und Erstellung von Slides zu den entsprechenden Aufgaben.
29.01.25	3:00h	Gemeinsame Gruppenarbeit	Zuhause Nikolas	Alle	Finalisieren der Präsentation mit dem Kunden. Austausch über die Umsetzung funktionaler und nicht funktionaler Anforderungen.
02.02.25	1:00h	Interne Absprache	Virtuell	Alle	Vorbereitung auf die Kundenpräsentation.
03.02.25	0:50h	Extern	Virtuell	Alle + Kunde	Vorstellung des aktuellen Stands des Projektes und Einholung von Feedback und Verbesserungsvorschlägen. Klärung von Fragen.
05.02.25	0:30h	Beratung	Virtuell	Alle + Consultant	Beratung durch internen Consultant über die technische Umsetzung des Projektes.
10.02.25	0:20h	Interne Absprache	Virtuell	Alle	Allgemeine Absprache zum aktuellen Stand, den Next Steps und aktuellen Problemen.
12.02.25	1:30h	Gemeinsame Gruppenarbeit	Kursraum	Nikolas, Malte	Implementierung der Tests und Optimierung der CI/CD-Pipeline
17.02.25	0:20h	Interne Absprache	Virtuell	Alle	Allgemeine Absprache zum aktuellen Stand, den Next Steps und aktuellen Problemen.
19.02.25	1:30h	Gemeinsame Gruppenarbeit	Virtuell	Nikolas, Malte	Grafische Darstellung der Wetterdaten im Frontend mittels d3.js.
24.02.25	0:20h	Interne Absprache	Virtuell	Alle	Allgemeine Absprache zum aktuellen Stand, den Next Steps und aktuellen Problemen.
26.02.25	1:30h	Gemeinsame Gruppenarbeit	Kursraum	Alle	Weiterführung der Dokumentation auf Basis des aktuellen technischen Standes und diversen Projektmanagement Methodiken.
03.03.25	0:20h	Interne Absprache	Virtuell	Alle	Allgemeine Absprache zum aktuellen Stand, den Next Steps und aktuellen Problemen.
06.03.25	0:45h	Extern	Hybrid	Alle	Abnahmetermin mit dem Kunden. Testung der Anwendung. Klärung von Fragen hinsichtlich der Projektmanagement-Dokumentation.
12.03.25	1:00h	Interne Absprache	Virtuell	Alle	Finale Anpassungen an der Anwendung und der Dokumentation. Vorbereitung zur Abgabe.

Anhang 2: Screenshots des Systemtests für Berlin

2.1 Testprotokoll (1. Standort Berlin: Konfiguration)

Verfügbare Stationen: 40177

Anzahl der Stationen: 10

Radius: 100

Zeitraum: Startjahr: 1920 Endjahr: 2024

ID	Latitude	Longitude	Name
GM000003319	52.4639	13.3017	BERLIN-DAHLEM
GM000003342	52.3833	13.0639	POTSDAM
GM000010393	52.2094	14.1203	LINDBERG

2.2 Testprotokoll (1. Standort Berlin: Datenanzeige als Tabelle)

BERLIN-DAHLEM - Jährliche Durchschnittswerte		
Year	Annual TMIN	Annual TMAX
1920	5.1	13.5
1921	5.2	14.2
1922	3.7	11.5
1923	4.5	12.2
1924	4.0	12.3
1925	5.2	13.1
1926	5.7	13.3
1927	4.8	12.3

BERLIN-DAHLEM - Saisonale Durchschnittswerte

Year	TMIN Winter	TMIN Spring	TMIN Summer	TMIN Autumn	TMAX Winter	TMAX Spring	TMAX Summer	TMAX Autumn
1920	-0.3	6.5	11.8	3.2	5.8	15.7	22.1	11.4
1921	-0.6	5.0	11.8	4.2	4.5	15.6	23.1	12.8
1922	-4.5	3.6	11.4	3.4	1.3	12.6	21.3	9.9
1923	-0.7	3.8	10.7	5.7	3.8	12.9	20.3	13.3

BERLIN-DAHLEM - Saisonale Durchschnittswerte		BERLIN-DAHLEM - Jährliche Durchschnittswerte	
Year	TMIN Winter	TMIN Spring	TMIN Summer
1920	-0.3	6.5	11.8
1921	-0.6	5.0	11.8
1922	-4.5	3.6	11.4
1923	-0.7	3.8	10.7

Automatisches Speichern Start Einfügen Zeichnen Seitenlayout Formeln Daten Überprüfen Ansicht

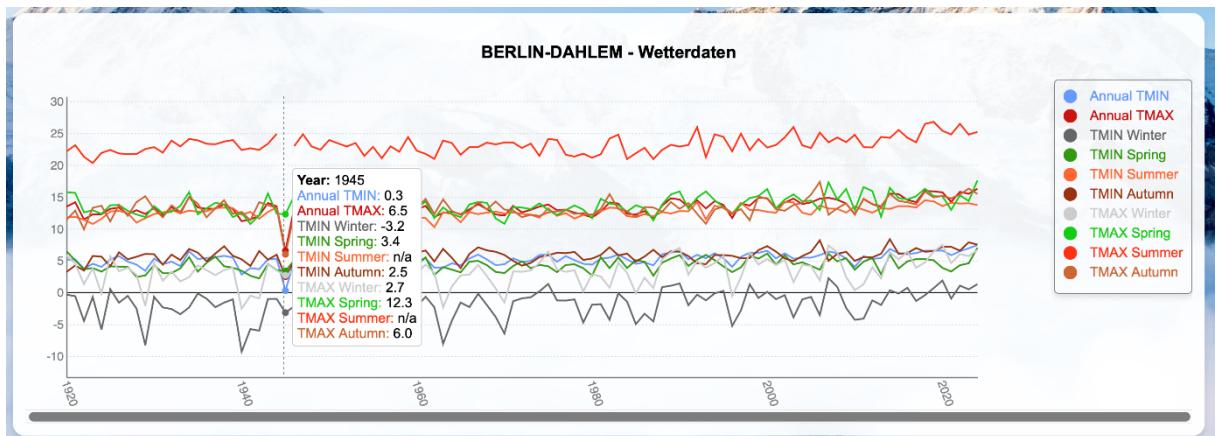
Suchen (Befehl + STRG + U) Kommentare Freigeben

E1 Spalte D/10

	A	B	C	D	E	F	G	H	I	J	K
1	Station	Jahr	Temperatur	Zehntel-Grad Celsius	Spalte D/10						
48243	GM000003319	19200101	TMIN	15	1,5	I					
48244	GM000003319	19200102	TMIN	-28	-2,8	I					
48245	GM000003319	19200103	TMIN	-38	-3,8	I					
48246	GM000003319	19200104	TMIN	-30	-3	I					
48247	GM000003319	19200105	TMIN	-36	-3,6	I					

Bereit 366 von 67016 Datensätzen gefunden. Bedienungshilfen: nicht verfügbar Mittelwert: 5,136065574 Anzahl: 366 Summe: 1879,8

2.3 Testprotokoll (1. Standort Berlin: Datenanzeige als Diagramm für 1945)



Automatisches Speichern Start Einfügen Zeichnen Seitenlayout Formeln Daten Überprüfen Ansicht

Suchen (Befehl + STRG + U) Kommentare Freigeben

E1 Spalte D/10

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Station	Jahr	Temperatur	Zehntel-Grad Celsius	Spalte D/10								
2	GM000003319	19450101	TMAX	6	0,6								
5	GM000003319	19450102	TMAX	-5	-0,5								
8	GM000003319	19450103	TMAX	26	2,6								
11	GM000003319	19450104	TMAX	33	3,3								
14	GM000003319	19450105	TMAX	18	1,8								

Bereit Filter-Modus Bedienungshilfen: nicht verfügbar Mittelwert: 6,549714286 Anzahl: 176 Summe: 1146,2

Automatisches Speichern Start Einfügen Zeichnen Seitenlayout Formeln Daten Überprüfen Ansicht

Suchen (Befehl + STRG + U) Kommentare Freigeben

E1 Spalte D/10

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Station	Jahr	Temperatur	Zehntel-Grad Celsius	Spalte D/10								
3	GM000003319	19450101	TMIN	-46	-4,6								
6	GM000003319	19450102	TMIN	-96	-9,6								
9	GM000003319	19450103	TMIN	-14	-1,4								
12	GM000003319	19450104	TMIN	14	1,4								
15	GM000003319	19450105	TMIN	-34	-3,4								

Bereit 175 von 526 Datensätzen gefunden. Bedienungshilfen: nicht verfügbar Mittelwert: 0,284 Anzahl: 176 Summe: 49,7

Automatisches Speichern Start Einfügen Zeichnen Seitenlayout Formeln Daten Überprüfen Ansicht

Suchen (Befehl + STRG + U) Kommentare Freigeben

E1 Spalte D/10

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Station	Frühling März - Mai Temperatur	Zehntel-Grad Celsius	Spalte D/10									
2	GM0000033	19450301	TMIN	12	1,2								
3	GM0000033	19450302	TMIN	11	1,1								
4	GM0000033	19450303	TMIN	-3	-0,3								
5	GM0000033	19450304	TMIN	-22	-2,2								
6	GM0000033	19450305	TMIN	-21	-2,1								

Bereit 1945, 1945_Frühling_MIN, 1945_Herbst_MAX, 2008 + Markieren Sie den Zielbereich, und drücken Sie die Eingabetaste. Mittelwert: 3,367272727 Anzahl: 56 Summe: 185,2

E1	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Station	Herbst September - November	Temperatur	Zehntel-Grad	Spalte D/10								
2	GM00000331		20080901	TMIN	124	12,4	E						
3	GM00000331		20080902	TMIN	121	12,1	E						
4	GM00000331		20080903	TMIN	119	11,9	E						
5	GM00000331		20080904	TMIN	91	9,1	E						
6	GM00000331		20080905	TMIN	96	9,6	E						

Bereit Barrierefreiheit: Keine Probleme Mittelwert: 6,057142857 Anzahl: 92 Summe: 551,2

1945 1945_Frühling_MIN 1945_Herbst_MAX 2008 2008_Frühling_MAX 2008_Herbst_MIN +

150 %

Anhang 3: Screenshots des Systemtests Nürnberg

3.1 Testprotokoll (2. Standort Nürnberg: Konfiguration)

ID	Latitude	Longitude	Name
GME00102380	49.5042	11.0567	NURNBERG
GMM00010761	49.0203	10.9617	WEISSENBURG
GM000004063	49.8753	10.9217	BAMBERG

3.2 Testprotokoll (2. Standort Nürnberg: Datenanzeige als Tabelle)

NURNBERG - Jährliche Durchschnittswerte		
Year	Annual TMIN	Annual TMAX
1920	5.3	14.1
1921	5.1	14.2
1922	3.8	11.8
1923	4.6	12.9
1924	3.9	12.2
1925	5.0	13.0
1926	5.2	14.0
1927	5.0	13.4

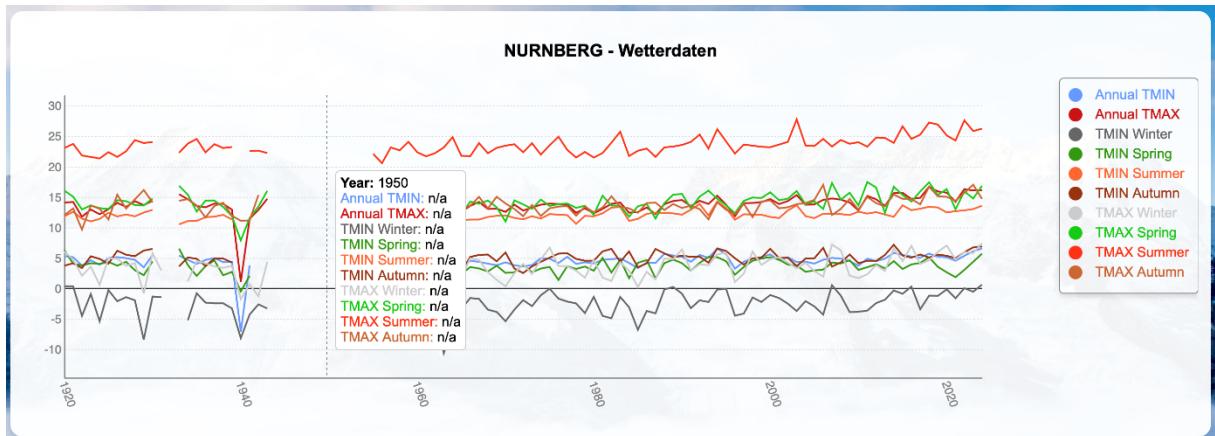
NURNBERG - Saisonale Durchschnittswerte								
Year	TMIN Winter	TMIN Spring	TMIN Summer	TMIN Autumn	TMAX Winter	TMAX Spring	TMAX Summer	TMAX Autumn
1920	0.4	6.3	11.8	3.7	6.1	16.1	23.0	12.0
1921	0.3	4.1	12.6	4.1	4.7	15.1	23.7	13.1
1922	-4.5	3.8	11.4	3.3	2.1	13.0	21.8	9.7
1923	-0.9	4.1	11.0	5.3	3.6	13.6	21.6	13.6
1924	-5.4	4.0	11.4	4.9	0.6	13.2	21.3	13.1

Screenshot of an Excel spreadsheet showing seasonal temperature data for Nürnberg. The data includes columns for Station, Year, Temperature Type (TMAX or TMIN), and Celsius values. The spreadsheet also displays summary statistics at the bottom: Mittelwert: 14,08606557, Anzahl: 367, Summe: 5155,5, and a zoom level of 144%.											
A	B	C	D	E	F	G	H	I	J	K	L
1	Station	Jahr	Temperatur	Zehntel-Grad Celsius	Spalte D/10						
2	GME00102380	19200101	TMAX	60	6	E					
4	GME00102380	19200102	TMAX	22	2,2	E					
6	GME00102380	19200103	TMAX	8	0,8	E					
8	GME00102380	19200104	TMAX	0	0	E					
10	GME00102380	19200105	TMAX	10	1	E					

Bereit 366 von 723 Datensätzen gefunden. Barrierefreiheit: Keine Probleme Mittelwert: 14,08606557 Anzahl: 367 Summe: 5155,5

Nürnberg_GME00102380 — Gespeichert									
Einfügen	Zeichnen	Seitenlayout	Formeln	Daten	Überprüfen	Ansicht	Kommentare	Freigeben	Suchen (Befehl + STRG + U)
A	B	C	D	E	F	G	H	I	J
1	Station	Jahr	Temperatur	Zehntel-Grad Celsius	Spalte D/10				
3	GME00102380	19200101	TMIN	10	1	E			
5	GME00102380	19200102	TMIN	4	0,4	E			
7	GME00102380	19200103	TMIN	-32	-3,2	E			
9	GME00102380	19200104	TMIN	-35	-3,5	E			
11	GME00102380	19200105	TMIN	-7	-0,7	E			

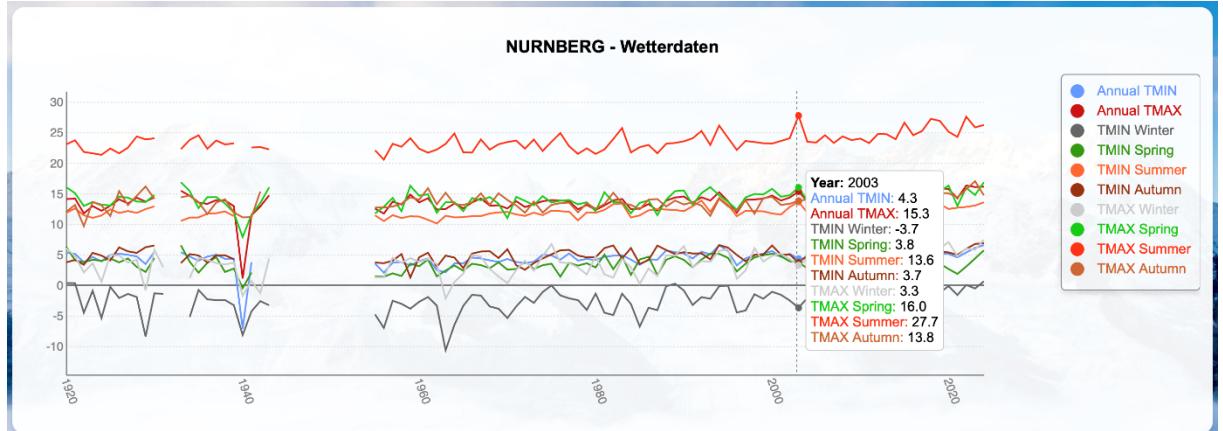
3.3 Testprotokoll (2. Standort Nürnberg: Datenanzeige als Diagramm für 1950)



1950.csv									
Q~ GME00102380									
USC00320136,19500101,TMIN,-200,,,0,1900									
USC00320136,19500101,TOBS,-194,,,0,1900									
USC00320136,19500101,PRCP,0,,,0,1900									
USC00320136,19500101,SNOW,0,,,0,									
USC00320149,19500101,PRCP,0,,,0,1900									
USC00320149,19500101,SNWD,102,,,0,									
USC00320196,19500101,TMAX,-67,,I,0,									
USC00320196,19500101,TMIN,-189,,,0,									
USC00320196,19500101,TOBS,-183,,,0,									
USC00320196,19500101,PRCP,0,,,0,									
USC00320196,19500101,SNOW,0,,,0,									
USC00320196,19500101,SNWD,127,,,0,									
USC00320196,19500101,WT11,1,,,0,									
USC00320209,19500101,TMAX,-67,,,0,0800									
USC00320209,19500101,TMIN,-206,,,0,0800									
USC00320209,19500101,TOBS,-206,,,0,0800									
USC00320209,19500101,PRCP,0,,,0,0800									
USC00320209,19500101,SNOW,0,,,0,									
USC00320209,19500101,SNWD,25,,,0,									
USC00320369,19500101,TMAX,-117,,,0,1800									
USC00320369,19500101,TMIN,-272,,,0,1800									
USC00320369,19500101,TOBS,-194,,,0,1800									
USC00320369,19500101,PRCP,0,,,0,1800									
USC00320382,19500101,TMAX,-78,,,0,1800									
USC00320382,19500101,TMIN,-172,,,0,1800									
USC00320382,19500101,TOBS,-172,,,0,1800									
USC00320382,19500101,PRCP,0,,,0,1800									
USC00320382,19500101,SNOW,0,,,0,									

->N/A es wurden keine Daten gefunden

3.4 Testprotokoll (2. Standort Nürnberg: Datenanzeige als Diagramm für 2003)



Automatisches Speichern Nürnberg_GME00102380 — Gespeichert

Start Einfügen Zeichnen Seitenlayout Formeln Daten Überprüfen Ansicht

E1 Spalte D/10

1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Station	Jahr	Temperatur	Zehntel-Grad Celsius	Spalte D/10										
2	GME00102380	20030101	TMAX	77	7,7	E									
6	GME00102380	20030102	TMAX	120	12	E									
10	GME00102380	20030103	TMAX	92	9,2	E									
14	GME00102380	20030104	TMAX	60	6	E									
18	GME00102380	20030105	TMAX	-19	-1,9	E									

Bereit 365 von 1509 Datensätzen gefunden. Barrierefreiheit: Keine Probleme Mittelwert: 15,32986301 Anzahl: 366 Summe: 5595,4

Automatisches Speichern Nürnberg_GME00102380 — Gespeichert

Start Einfügen Zeichnen Seitenlayout Formeln Daten Überprüfen Ansicht

E1 Spalte D/10

1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Station	Jahr	Temperatur	Zehntel-Grad Celsius	Spalte D/10										
3	GME00102380	20030101	TMIN	-24	-2,4	E									
7	GME00102380	20030102	TMIN	77	7,7	E									
11	GME00102380	20030103	TMIN	55	5,5	E									
15	GME00102380	20030104	TMIN	-20	-2	E									
19	GME00102380	20030105	TMIN	-46	-4,6	E									

Bereit 365 von 1509 Datensätzen gefunden. Barrierefreiheit: Keine Probleme Mittelwert: 4,320821918 Anzahl: 366 Summe: 1577,1

Automatisches Speichern Nürnberg_GME00102380 — Gespeichert

Start Einfügen Zeichnen Seitenlayout Formeln Daten Überprüfen Ansicht

E1 Spalte D/10

1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
2	GME0010238	20030301	TMAX	94	9,4	E									
3	GME0010238	20030302	TMAX	113	11,3	E									
4	GME0010238	20030303	TMAX	60	6	E									
5	GME0010238	20030304	TMAX	85	8,5	E									
6	GME0010238	20030305	TMAX	115	11,5	E									

Bereit 365 von 1509 Datensätzen gefunden. Barrierefreiheit: Keine Probleme Mittelwert: 15,99565217 Anzahl: 93 Summe: 1471,6

Automatisches Speichern Nürnberg_GME00102380 — Gespeichert

Start Einfügen Zeichnen Seitenlayout Formeln Daten Überprüfen Ansicht

E1 Spalte D/10

1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
2	GME001	20030901	TMIN	76	7,6	E									
3	GME001	20030902	TMIN	51	5,1	E									
4	GME001	20030903	TMIN	19	1,9	E									
5	GME001	20030904	TMIN	23	2,3	E									
6	GME001	20030905	TMIN	53	5,3	E									

Bereit Barrierefreiheit: Keine Probleme Mittelwert: 3,671428571 Anzahl: 92 Summe: 334,1

Anhang 4: Screenshots des Systemtests Windhoek

4.1 Testprotokoll (3. Standort Windhoek: Konfiguration)

ID	Latitude	Longitude	Name
WA007401540	-22.567	17.1	WINDHOEK

4.2 Testprotokoll (3. Standort Windhoek: Datenanzeige als Tabelle)

Year	Annual TMIN		Annual TMAX	
1980	12.8		27.3	
1981	13.0		27.4	
1982	13.6		27.1	
1983	13.8		27.2	
1984	14.1		27.7	
1985	13.5		27.2	
1986	13.2		27.1	
1987	14.0		24.9	

Year	TMIN Summer	TMIN Autumn	TMIN Winter	TMIN Spring	TMAX Summer	TMAX Autumn	TMAX Winter	TMAX Spring
1980	17.3	12.9	7.5	13.6	31.5	26.2	22.2	29.3
1981	18.3	12.7	7.2	14.2	32.8	26.1	21.5	29.3
1982	18.1	13.1	8.2	15.1	31.6	26.0	21.7	29.1
1983	18.1	14.1	8.3	14.9	30.7	27.1	21.8	29.1

Bereit 366 von 1098 Datensätzen gefunden. Barrierefreiheit: Keine Probleme

Mittelwert: 27,26338798 Anzahl: 367 Summe: 9978,4

Windhoek_WA007401540 — Gespeichert

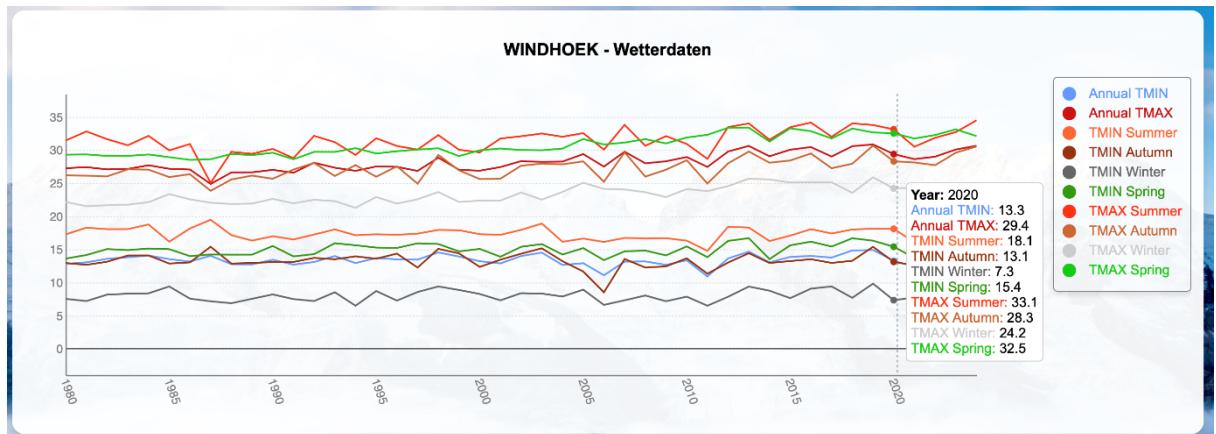
Start Einfügen Zeichnen Seitenlayout Formeln Daten Überprüfen Ansicht

E1 Spalte D/10

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Station	Jahr	Temperatur	Zehntel-Grad Celsius	Spalte D/1														
3	WA007401540	19800101	TMIN		200	20													
6	WA007401540	19800102	TMIN		221	22,1													
9	WA007401540	19800103	TMIN		166	16,6													
12	WA007401540	19800104	TMIN		188	18,8													
15	WA007401540	19800105	TMIN		171	17,1													

Bereit 366 von 1098 Datensätzen gefunden. Barrierefreiheit: Keine Probleme Mittelwert: 12,82404372 Anzahl: 367 Summe: 4693,6

4.3 Testprotokoll (3. Standort Windhoek: Datenanzeige als Diagramm für 2020)



Windhoek_WA007401540 — Gespeichert

Start Einfügen Zeichnen Seitenlayout Formeln Daten Überprüfen Ansicht

E1 Spalte D/10

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Station	Jahr	Temperatur	Zehntel-Grad Celsius	Spalte D/1										
2	WA007401540	20200101	TMAX		347	34,7									
4	WA007401540	20200102	TMAX		340	34									
8	WA007401540	20200103	TMAX		328	32,8									
12	WA007401540	20200104	TMAX		328	32,8									
16	WA007401540	20200105	TMAX		322	32,2									

Bereit 325 von 1006 Datensätzen gefunden. Barrierefreiheit: Keine Probleme Mittelwert: 29,39569231 Anzahl: 326 Summe: 9553,6

Windhoek_WA007401540 — Gespeichert

Start Einfügen Zeichnen Seitenlayout Formeln Daten Überprüfen Ansicht

E1 Spalte D/10

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Station	Jahr	Temperatur	Zehntel-Grad Celsius	Spalte D/1										
5	WA007401540	20200102	TMIN		175	17,5									
9	WA007401540	20200103	TMIN		177	17,7									
13	WA007401540	20200104	TMIN		186	18,6									
17	WA007401540	20200105	TMIN		178	17,8									
21	WA007401540	20200106	TMIN		170	17									

Bereit 277 von 1006 Datensätzen gefunden. Barrierefreiheit: Keine Probleme Mittelwert: 13,27292419 Anzahl: 278 Summe: 3676,6

The image shows two screenshots of Microsoft Excel spreadsheets. Both have the title bar "Windhoek_WA007401540 — Gespeichert" and the ribbon menu "Start", "Einfügen", "Zeichnen", "Seitenlayout", "Formeln", "Daten", "Überprüfen", "Ansicht".

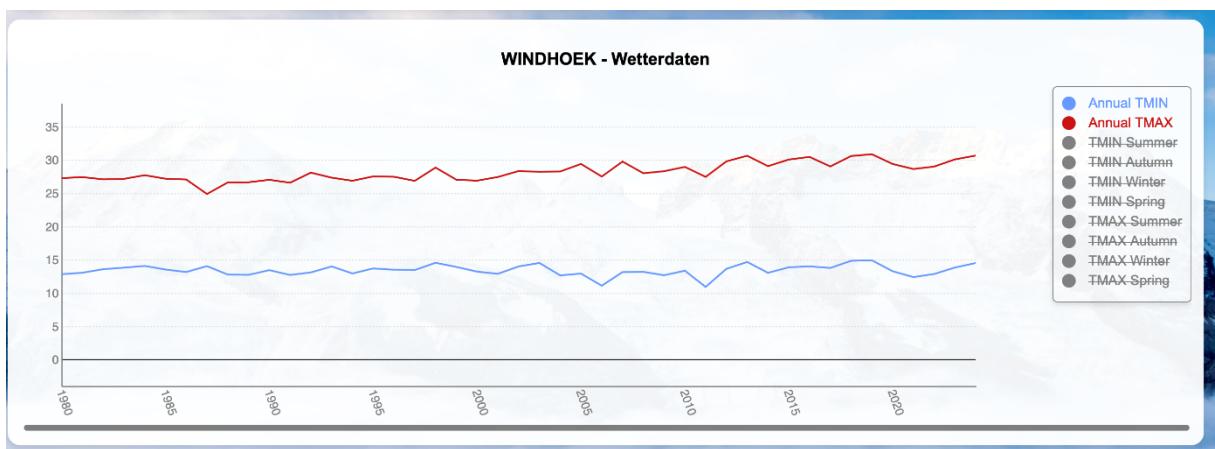
Table 1 (Top):

E1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Station	Herbst März - Mai	Temperatur	Zehntel-Grad Celsius	Spalte D/10											
2	WA0074	20200301	TMIN	165	16,5			S								
3	WA0074	20200303	TMIN	146	14,6			S								
4	WA0074	20200304	TMIN	147	14,7			S								
5	WA0074	20200306	TMIN	149	14,9			S								
6	WA0074	20200307	TMIN	146	14,6			S								

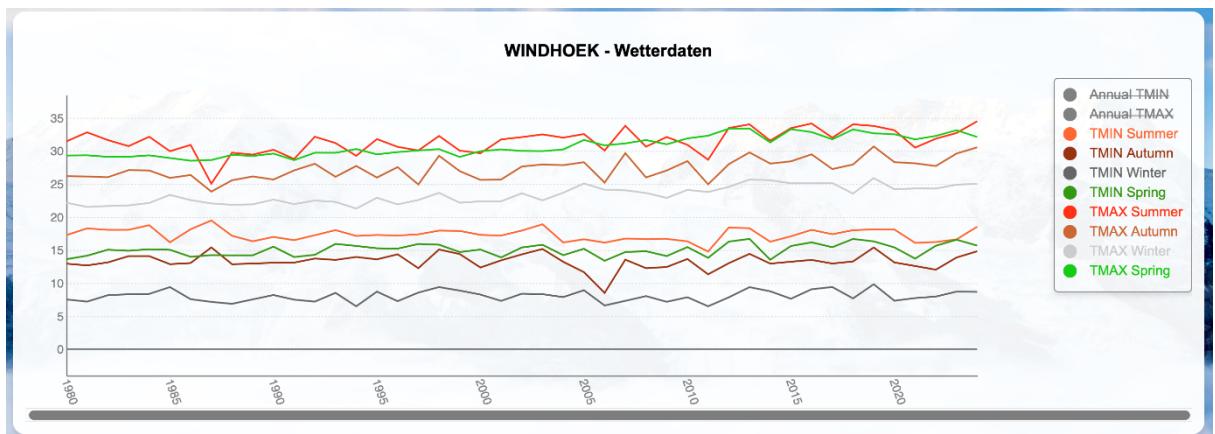
Table 2 (Bottom):

E1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
1	Station	Frühling September - November	Temperatur	Zehntel-Grad Celsius	Spalte D/10										
2	WA0074	20200901	TMAX	257	25,7			S							
3	WA0074	20200902	TMAX	258	25,8			S							
4	WA0074	20200903	TMAX	293	29,3			S							
5	WA0074	20200904	TMAX	293	29,3			S							
6	WA0074	20200905	TMAX	296	29,6			S							

4.4 Testprotokoll (3. Standort Windhoek: Einblendung von TMIN, TMAX)



4.5 Testprotokoll (3. Standort Windhoek: Ausblendung von TMIN, TMAX)



7 Selbstständigkeitserklärung

Wir versichern hiermit, dass wir die vorliegende Arbeit mit dem Thema: Dokumentation zur Anwendungsentwicklung eines Wetterdaten-Analysetools selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben.

Villingen-Schwenningen, 14.03.2025

N. Diehl S. Dubiel Q. Großhauser A. Iacona M. Paffen

Nikolas Diehl, Simon Dubiel, Quirin Großhauser, Angelo Iacona, Malte Paffen