



Anforderungsspezifikation Klima-Viewer

Projektteam KlimaViewer

Ivo Kozina
Ohran Mujkic

Version 1.0
Bern, 30. November 2018

Inhaltsverzeichnis

1	ZWECK DES DOKUMENTS	3
2	VISION	3
3	PROJEKTZIELE	3
3.1	STAKEHOLDER	3
3.2	ZIELE	3
3.2.1	<i>Alle Stakeholder</i>	3
3.2.2	<i>Auftraggeber</i>	3
3.2.3	<i>Privatpersonen</i>	3
3.3	NICHT-ZIEL	3
4	SYSTEMABGRENZUNG	4
4.1	PROZESSUMFELD	4
4.2	SYSTEMUMFELD	4
4.3	RAHMENBEDIENUNG	5
5	ANFORDERUNGEN	5
5.1	QUELLEN UND VORGEHEN	5
5.1.1	<i>Quellen</i>	5
5.1.2	<i>Vorgehen</i>	5
5.2	FUNKTIONALE ANFORDERUNGEN	6
5.2.1	<i>Use Case F1</i>	6
5.2.2	<i>Use Case F2</i>	7
5.2.3	<i>Use Case F3</i>	7
5.2.4	<i>Use Case F4</i>	8
5.3	MOCKUP	9
5.4	QUALITÄTSANFORDERUNGEN	11
5.5	RANDBEDINGUNGEN	11
5.6	DATENMODELL	12
5.6.1	<i>Datenbankmodellierung</i>	12
5.6.2	<i>Datenarchivierung</i>	12
5.6.3	<i>Klassendiagramm</i>	12
6	GLOSSAR	13
7	ABILDUNGSVERZEICHNIS	14
8	ANHANG	15
8.1	ABSTIMMUNG DER ANFORDERUNGEN	15
8.2	DEFINITION OF READY – CHECKLIST	15
8.3	PROJEKTBESCHREIBUNG	16
9	VERSIONSKONTROLLE	17

1 Zweck des Dokuments

Der Hauptzweck dieses Dokument ist es die Ziele sowie die Anforderungen für das Projekt „KlimaViewer“ aufzuführen.

2 Vision

Mit dem Projekt wird eine Webapplikation entwickelt, die es möglich macht Klimaveränderungen über einen bestimmten Zeitraum für die Nutzer verständlich, informativ sowie attraktiv darzustellen. Dabei soll der Verlauf der Klimaveränderung in grafischen Darstellungen erfolgen. Die grafischen Darstellungen können ohne bestimmte Vorkenntnisse interpretiert werden. Die Nutzer dieser Webapplikation können mit echten Klimadaten verschiedene Auswertungen und Trends erstellen. Zudem ermöglicht die Webapplikation das Aufrufen von aktuellen Wetterdaten. Somit wird ein neues Tool entwickelt, welches auf den bisherigen gesammelten Daten aufbaut. Das Tool soll als Prototyp dienen und keine produktive und vollendete Software darstellen.

3 Projektziele

3.1 Stakeholder

- Auftraggeber
- Privatpersonen

3.2 Ziele

3.2.1 Alle Stakeholder

Die Applikation soll die Kennzahlen, Auswertungen und Verläufe im Wetterbereich anhand einer sehr einfachen Bedienung schnell und verständlich darstellen. Vor- und Fachkenntnisse sollen nicht notwendig sein. Dabei soll das Tool offen für alle und aufrufbar für die gängigsten Browser sein.

3.2.2 Auftraggeber

Die Applikation soll nach den dokumentierten funktionalen Zielen dem Auftraggeber überwiesen werden. Die Aufrufbarkeit, Qualität und Brauchbarkeit sollen in einem zu erwartenden Niveau sein. Es soll möglich sein, die Applikation auf den gängigsten Systemen nutzen zu können. Ausserdem soll der Lerneffekt durch diese praktische Arbeit gegeben sein sowie sollte das Projekt-Team in der Lage sein mit dem erworbenen Know-how ein ähnliches Projekt wieder umzusetzen.

3.2.3 Privatpersonen

Die Applikation soll ohne grosse Installationen auf das Privatgerät funktionieren. Ausserdem soll das UI sehr benutzerfreundlich sein, damit Daten auch ohne Fachkompetenzen angezeigt und interpretiert werden können.

3.3 NICHT-Ziel

Es sollen keine Prognosen erstellt werden, da für ein solches Vorhaben weitere Kompetenzen benötigt werden, welche den Rahmen des Projektes deutlich sprengen.

4 Systemabgrenzung

Als Datenlieferant wurde in der Projektbeschreibung WorldClim als Empfehlung angegeben, jedoch wurde gemeinsam mit dem Stakeholder abgesprochen, dass OpenWeatherMap gebraucht wird. Der Grund dazu ist, dass für die aktuellen Projektzwecke WorldClim zu aufwändig ist, da die Daten erst noch relativ kompliziert weiterverarbeitet müssen. Mit OpenWeatherMap ist die Abfrage und das Sammeln der Daten deutlich einfacher und praktischer. Diese können über eine API abgefragt werden.

4.1 Prozessumfeld

Die folgende Grafik beschreibt den Geschäftsprozess:

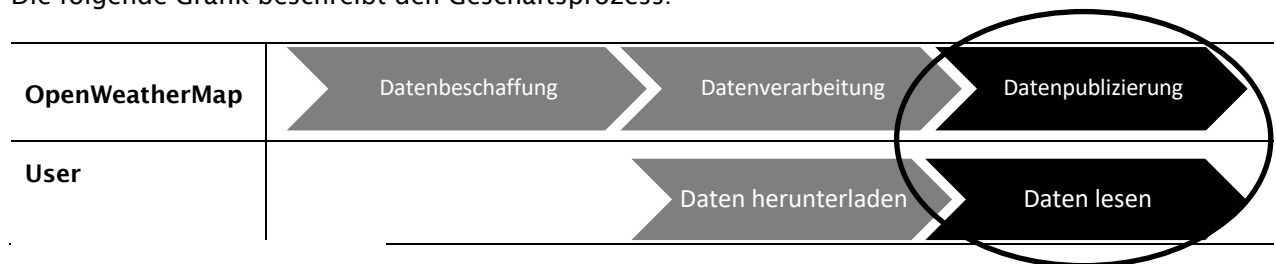


Abbildung 1: Prozessumfeld

Es gibt grundsätzlich zwei Hauptprozesse. Nachdem OpenWeatherMap die Daten beschafft und entsprechend verarbeitet, werden sie auf ihrer Website zum Herunterladen oder Abfragen bereitgestellt. Der Benutzer kann diese Daten herunterladen und lesen.

Mit dem KlimaViewer werden die publizierten Daten von OpenWeatherMap verarbeitet und benutzerfreundlich für den Benutzer dargestellt. Somit muss der Benutzer keine Daten herunterladen und selbst verarbeiten. Jedoch hat er trotzdem die Möglichkeit Wetterdaten aus dem KlimaViewer zu exportieren.

4.2 Systemumfeld

Das Umfeld des Systems wird in der folgenden Grafik dargestellt:

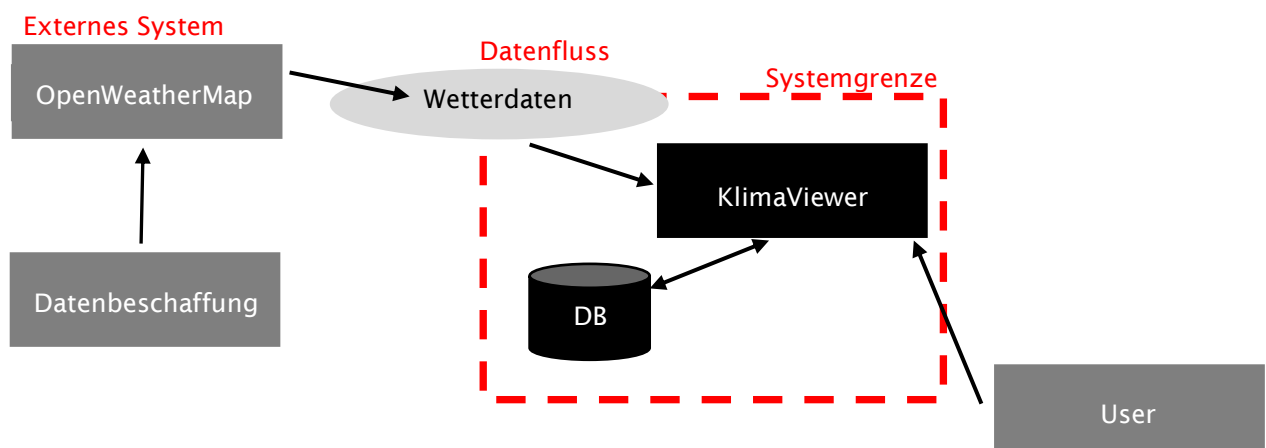


Abbildung 2: Systemumfeld

Die Daten werden von OpenWeatherMap (<https://openweathermap.org/>) zu Verfügung gestellt. Diese Daten sind nicht nur wahrheitsgetreu, sondern auch für die Nutzung freigegeben. Der KlimaViewer wird sich diese Leistung auf zwei Arten zu nutzen machen: Einerseits werden historische Daten in einer Datenbank gesammelt, damit Wetterveränderung über einen bestimmten Zeitraum gemacht werden können. Andererseits werden über eine API Wetterdaten abgefragt, um sie dann dem Nutzer im KlimaViewer anzuzeigen.

4.3 Rahmenbedienung

Die Realisierung für dieses Projekt erfolgt im Rahmen des Moduls „Projekt 1“. Eine Fortsetzung des Projektes ausserhalb des genannten Moduls steht noch offen. Die Umsetzung wird hauptsächlich mit der Technologie JavaScript erfolgen, da wir uns auf das MEAN Stack stützen. Die Lösung ist auf einer virtuellen Maschine angesiedelt und steht über eine URL zur Nutzung.

5 Anforderungen

5.1 Quellen und Vorgehen

5.1.1 Quellen

Für die Anforderungsermittlung stützen wir uns auf die folgenden Quellen:

Quelle	Bemerkung
Initialer Projektbeschrieb	Ausgangslage für dieses Projekt
Wetterdaten-Webseite	OpenWeatherMap bietet uns einen Überblick, wie wir die gewünschten Wetterdaten erhalten. - https://openweathermap.org/
Wetter Apps/ Webseiten	Mithilfe von anderen Wetter-Apps werden wir uns ein Bild machen, wie eine gute Wetter App oder Webseite aussehen könnte und welche Daten sie alles liefert. - https://www.accuweather.com/en/ch/switzerland-weather - https://weather.com/
Joachim Kaltz	Wir erhalten viele technische Inputs. Des Weiteren wird der Funktionsumfang der Webapplikation mit ihm abgesprochen.
Team	Gegeben durch der Freiheit vom Projekt können die Auswertungen sowie die grafischen Darstellungen vom Team gewählt werden. Das Team bildet im Projekt die grösste Quelle an Informationen.

5.1.2 Vorgehen

Zu Beginn werden wir uns mit dem Projektbeschrieb auseinandersetzen, um diesen zu verstehen und die darin enthaltenen Anforderungen notieren. Somit können wir sicher sein, dass wir die Grundanforderung bestimmt nicht übersehen und uns ein erstes Bild ausmalen können, wie das Endprodukt aussehen könnte.

Als nächstes werden wir uns mit verschiedensten Wetterseiten auseinandersetzen, vor allem mit denen, welche Wetterdaten liefern. Dadurch werden wir sehen, welche Möglichkeiten wir haben und für welche Datenquelle wir uns schlussendlich entscheiden werden.

Bevor wir mit der Umsetzung starten, schauen wir uns andere Wetter Apps/ Webseiten an, um zu sehen, welche Gemeinsamkeiten sie haben und wie unsere Webseite aussehen könnte. Ausserdem können wir so sehen, was unserer Webseite nicht fehlen darf.

Zum Schluss kommt die Umsetzung des Produktes. Dabei haben wir uns für MEAN entschieden. MEAN ist ein kostenloser und Open-Source-JavaScript-Software-Stack zum Erstellen dynamischer Websites und Webanwendungen. Der MEAN-Stack besteht aus MongoDB, Express.js, AngularJS und Node.js. Es ist noch unklar, ob eine MongoDB benötigt wird, da eventuell Daten direkt aus einer API gelesen werden.

5.2 Funktionale Anforderungen

Attribute:

ID: eindeutige Identifikation

Status: Entwurf / Geprüft / Freigegeben

Priorität: Muss / Optional P1, P2, P3 / Wunsch (Nice to have)

ID	Status	Priorität	Beschreibung
F1	Freigegeben	M	Der Benutzer kann die Applikation in einem Webbrowser öffnen.
F2	Freigegeben	M	Der Benutzer kann für verschiedene Kennzahlen (Temperatur, Windgeschwindigkeit, Niederschlag sowie Luftdruck) über einen bestimmten Zeitraum in der Vergangenheit eine graphische Darstellung erhalten, welche sich hauptsächlich auf die Veränderung fokussiert. Diese Daten kann er auch als CSV-Datei exportieren.
F3	Freigegeben	M	Der Benutzer kann für verschiedene Städte die aktuellen Wetterdaten (Temperatur, Windgeschwindigkeit, Niederschlag sowie Luftdruck) erfahren.
F4	Freigegeben	M	Der Benutzer kann eine neue Stadt hinzufügen und diese wird in der Webapplikation hinterlegt und weiter getrackt.

5.2.1 Use Case F1

Nr.	F1
Kurzbeschreibung	Der Benutzer ruft die Internetseite über den URL auf, damit er Zugriff auf die Webapplikation KlimaViewer erhält.
Beteiligte Akteure	Benutzer, System
Auslöser/ Vorbedingung	Der Benutzer gibt in seinem Internetbrowser die URL von Klimaviewer an.
Ergebnis/ Nachbedingung	Der Benutzer gelangt erfolgreich auf die Seite von der Webapp KlimaViewer

Ablauf

Nr.	Wer?	Was
1.1	Benutzer	Der Benutzer kommt mit der URL sowie seinem gewöhnlichen Internetbrowser auf die Webapplikation KlimaViewer.
1.2	System	Das System gibt eine valide Internetseite zurück

5.2.2 Use Case F2

Nr.	F2
Kurzbeschreibung	Der Benutzer kann über die Rubrik "Historical" vergangene Kennzahlen im Vergleich zu einander von einer bestimmten Stadt anschauen.
Beteiligte Akteure	Benutzer, System
Auslöser/ Vorbedienung	Der Benutzer wählt eine Stadt und einen Zeitraum aus.
Ergebnis/ Nachbedienung	Der Benutzer erhält ein Diagramm in Form eines Linien- oder Balkendiagramm mit den gewünschten Informationen.

Ablauf

Nr.	Wer?	Was
2.1	Benutzer	Der Benutzer wählt die Rubrik "Historical".
2.2	Benutzer	Der Benutzer wählt eine Stadt und einen Zeitraum aus.
2.3	System	Das System gibt ihm alle Daten aus in Form eines Diagramms, welches sehr verständlich und einfach gehalten ist.

Ausnahme / Varianten

Nr.	Wer?	Was
2.3.1	System	Der Benutzer hat eine nicht-valide Stadt oder einen nicht-validen Zeitraum ausgewählt. Also kann das System ausser einer Fehlermeldung nicht zurückgeben.

5.2.3 Use Case F3

Nr.	F3
Kurzbeschreibung	Der Benutzer kann über die Rubrik "Live Weather" aktuelle Kennzahlen zu bestimmten Städten anschauen.
Beteiligte Akteure	Benutzer, System
Auslöser/ Vorbedienung	Der Benutzer wählt die Rubrik "Live Weather" aus.
Ergebnis/ Nachbedienung	Der Benutzer erhält die Kennzahlen grafisch verpackt.

Ablauf

Nr.	Wer?	Was
3.1	Benutzer	Der Benutzer wählt die Rubrik "Live Weather".
3.2	System	Das System gibt ihm alle Städte aus, die getrackt werden auf der Webapplikation, mit den aktuellen Wetterdaten aus.

5.2.4 Use Case F4

Nr.	F4
Kurzbeschreibung	Der Benutzer kann über die Rubrik "Live Weather" weitere Städte hinzufügen.
Beteiligte Akteure	Benutzer, System
Auslöser/ Vorbedienung	Der Benutzer tippt eine Stadt ein und die aktuellen Wetterdaten in dieser Stadt erscheinen auf dem Dashboard.
Ergebnis/ Nachbedienung	Der Benutzer sieht auf dem Dashboard die gesuchte Stadt mit den aktuellen Wetterdaten.

Ablauf

Nr.	Wer?	Was
4.1	Benutzer	Der Benutzer wählt unter der Rubrik "Live Weather" die Funktion "Add City" aus. Und tippt den Namen der gewünschten Stadt aus.
4.2	System	Das System fügt die Stadt ins Backend hinzu und gibt dem Benutzer auf dem Dashboard die gewünschte Stadt mit den aktuellen Wetterdaten zurück.

Ausnahme / Varianten

Nr.	Wer?	Was
4.1.1	System	Der Benutzer hat eine nicht-valide Stadt eingegeben. Also kann das System ausser einer Fehlermeldung nicht zurückgeben.

5.3 Mockup

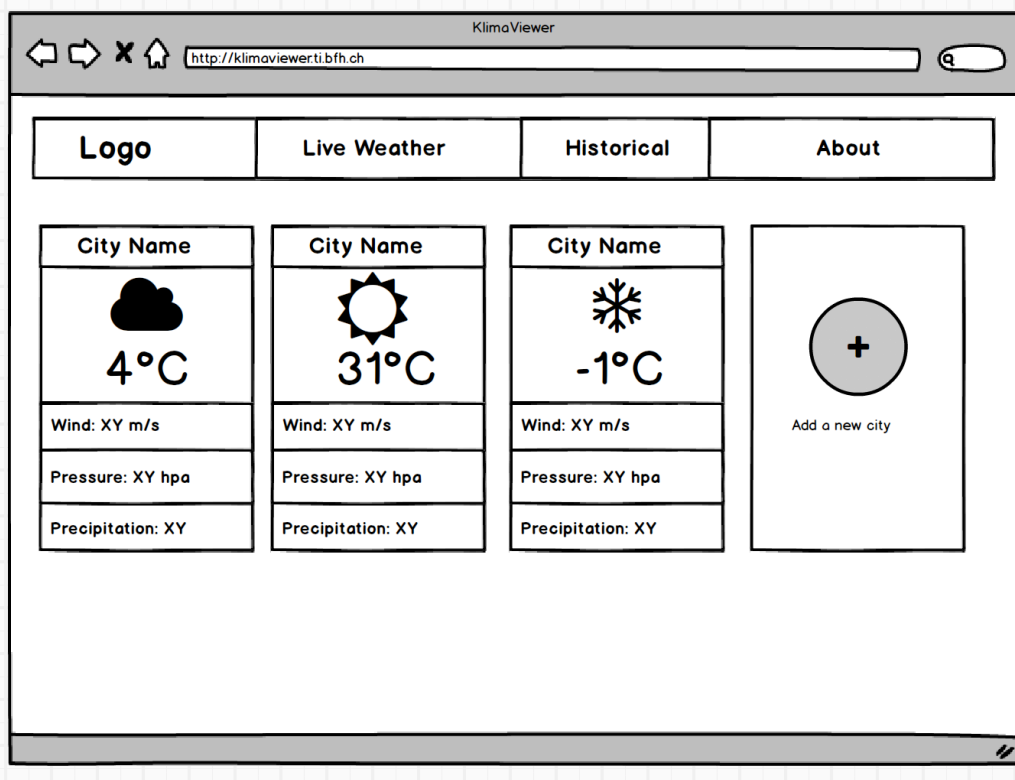


Abbildung 3: Mockup Dashboard

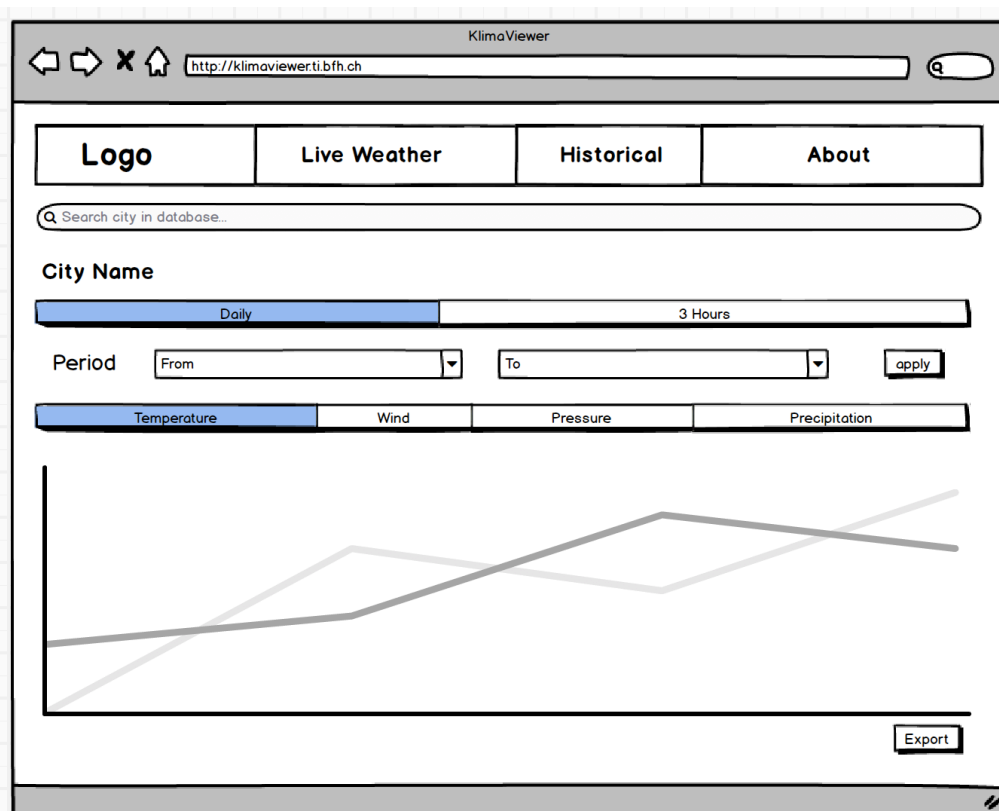


Abbildung 4: Mockup Live-Weather

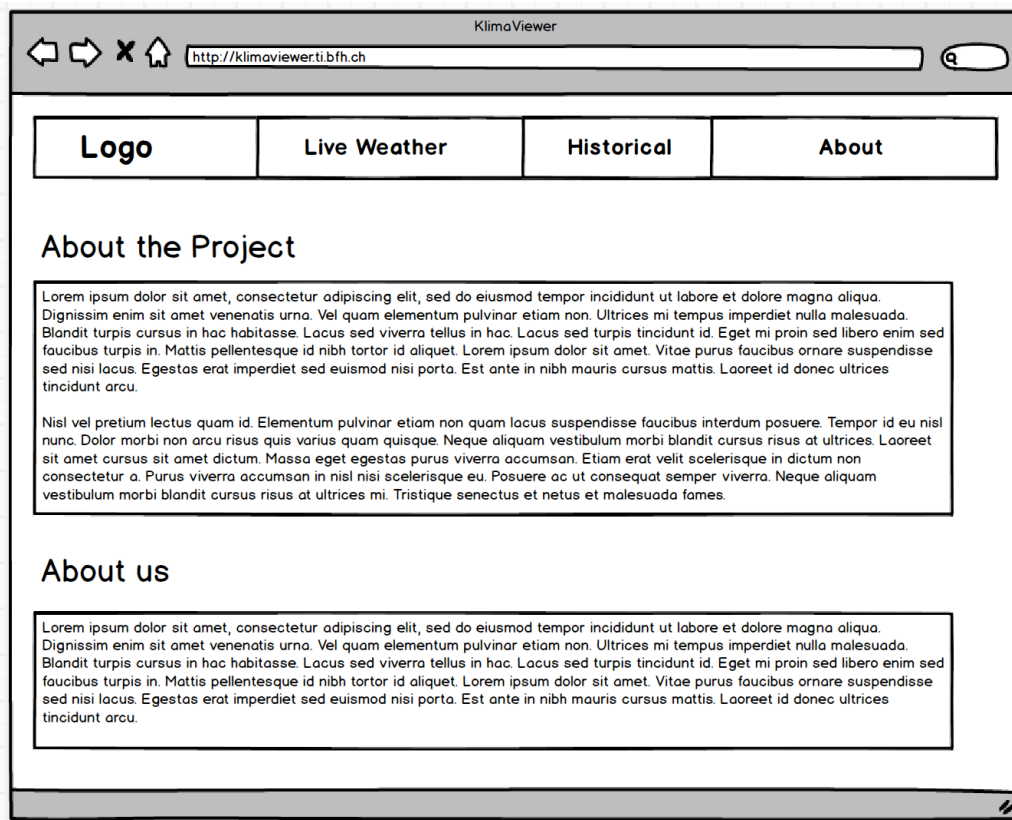


Abbildung 5: Mockup About

5.4 Qualitätsanforderungen

Attribute:

ID: eindeutige Identifikation

Status: Entwurf / Geprüft / Freigegeben

Priorität: Muss / Optional P1, P2, P3 / Wunsch (Nice to have)

ID	Status	Prio	Beschreibung
Q1	Freigegeben	M	Die Benutzer müssen sich nicht authentifizieren.
Q2	Freigegeben	M	Das Design soll ansprechend und leicht zu bedienen sein.
Q3	Freigegeben	M	Die Verfügbarkeit der Wetterdaten wird nicht garantiert.
Q4	Freigegeben	M	Die Wetterdaten werden nicht verändert oder verfälscht.

5.5 Randbedingungen

Attribute:

ID: eindeutige Identifikation

Status: Entwurf / Geprüft / Freigegeben

Priorität: Muss / Optional P1, P2, P3 / Wunsch (Nice to have)

ID	Status	Prio	Beschreibung
R1	Freigegeben	M	Die Wetterdatenquelle ist verfügbar und kostenlos.
R2	Freigegeben	M	Der Sourcode ist auf Github zu finden.
R3	Freigegeben	M	Als Entwicklungsumgebung wird IntelliJ verwendet.
R5	Freigegeben	M	Die Applikation und Datenbank laufen auf einer VM.
R6	Freigegeben	M	Es werden keine Wetterprognosen erstellt.
R7	Freigegeben	M	Das Projekt wird im definierten Zeitraum realisiert und abgeschlossen.

5.6 Datenmodell

5.6.1 Datenbankmodellierung

Die folgende Abbildung zeigt einen Entwurf für die Datenbank. Es handelt sich hier um ein Entity-Relationship-Diagramm, was üblich als Basis für die erste Initialisierung der Datenbank gebraucht wird. Dieses Diagramm kann sich im Verlauf der Realisierung noch verändern.

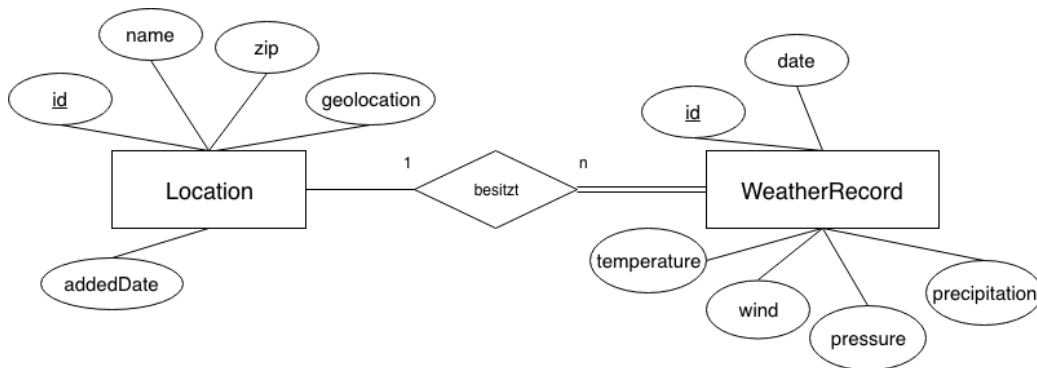


Abbildung 6: ERD

5.6.2 Datenarchivierung

Die Daten über den REST-API von OpenWeatherMap abgefragt. Diese kommen als JSON wieder zurück, welche dann in die MongoDB gespeichert. Da leider OpenWeatherMap nur die aktuellen Wetterdaten kostenlos zu Verfügung stellt, sind wir gezwungen selbst die Daten zu archivieren. Durch einen Cronjob kann alle drei Stunden eine Abfrage gemacht werden auf den REST-API. Somit wächst die eigene Datensammlung. Dabei werden folgende Daten je nach Stadt abgefragt:

- Temperatur
- Niederschlag
- Luftdruck
- Windgeschwindigkeit

5.6.3 Klassendiagramm

Die folgende Abbildung zeigt einen Entwurf für die Software. Es handelt sich hierbei um ein reduziertes Klassendiagramm, welches nur die wichtigste Beziehung aufweist. Während der Realisierung kann es durchaus sein, dass es Veränderungen geben kann.

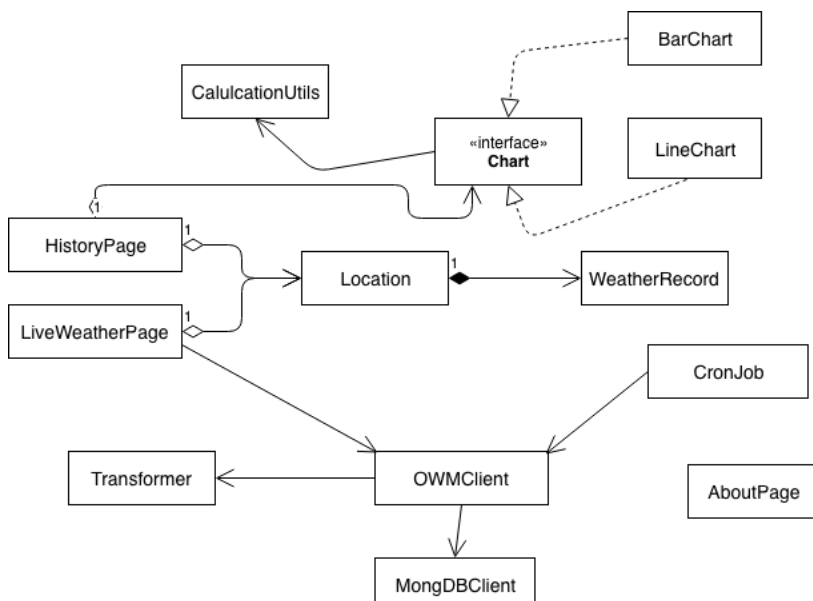


Abbildung 7: Klassendiagramm

6 Glossar

Begriffe	Erklärung
Prototyp	Muster
Browser	Programm um nach Webseiten zu suchen, lesen oder verwalten.
UI	User Interface, Benutzeroberfläche
WorldClim	Plattform, welche Wetterdaten aus der ganzen Welt als Datei zu Verfügung stellt.
OpenWeatherMap	Plattform, welche Wetterdaten aus der ganzen Welt über einen API zu Verfügung stellt.
API	Programmierschnittstelle
REST	Representational state transfer – Software-Architekturstil meistens für Webservices.
JSON	JavaScript Object Notation – Dateiformat meistens im Gebrauch für Webservices.
DB	Datenbank – Sammlung und Ablage von Daten.
Mean Stack	Kostenloser, Open-Source JavaScript Library zum Erstellen von dynamischen Webseiten.
URL	Adressierung einer Webseite
Entity-Relationship-Model	Zeigt anhand von einem Diagramm das Grobkonzept einer Datenbank. Hierbei spielen die Tabellen sowie deren Beziehung zueinander die Schlüsselrolle.
Cronjob	Ein Skript, welches im Hintergrund regelmässig zu einer vordefinierten Zeit, immer sich ausführt.
MongoDB	Das ist eine Datenbankprogramm.
CSV	Comma-separated Values – Ist ein Dateityp, dass sich besonders gut für den Export von Daten eignet.

7 Abbildungsverzeichnis

Abbildung 1: Prozessumfeld	4
Abbildung 2: Systemumfeld	4
Abbildung 3: Mockup Dashboard	9
Abbildung 4: Mockup Live-Weather	9
Abbildung 5: Mockup About	10
Abbildung 6: ERD	12
Abbildung 7: Klassendiagramm.....	12

8 Anhang

8.1 Abstimmung der Anforderungen

Ein Konflikt bei der Abstimmung der Anforderungen war WorldClim. Da WorldClim in der Projektbeschreibung als Datenlieferant empfohlen wurde, plante wir unser Projekt zuerst damit. Jedoch stellten wir fest, dass Daten von WorldClim erst noch relativ kompliziert weiterverarbeitet werden müssen um diese zu nutzen. Also haben wir mit unserem Stakeholder Joachim Wolfgang Kaltz die Situation besprochen und sind zum Schluss gekommen, dass wir in unserem Projekt OpenWeatherMap verwenden werden. Jedoch stellte sich dann der zweite Konflikt. Bei OpenWeatherMap sind Daten in weiter Vergangenheit kostenpflichtig. Also haben wir abgemacht, dass wir für eine kürzere Periode Daten von OpenWeatherMap sammeln und diese als Vorlage verwenden. Da es beim Projekt rein um die Umsetzung geht und WorldClim nur als Beispiel vorgeschlagen wurde, können beide Konflikte als gelöst betrachtet werden.

8.2 Definition of Ready – Checklist

ID	Status	Definition of Ready
DoR1	Erfüllt	Projektrahmen ist für die Teamgrösse angepasst.
DoR2	Erfüllt	Eine Aufwandschätzung ist gemacht.
DoR3	Erfüllt	Ressourcen wurden ermittelt und sind verplant mit Reserve-Spielraum.
DoR4	Erfüllt	Alle Mittel für die Umsetzung sind vorhanden.
DoR5	Erfüllt	Externe Abhängigkeiten wurden erkannt.
DoR6	Erfüllt	Die Applikation und Datenbank laufen auf einer VM.

8.3 Projektbeschreibung

Modul BTI7301

Projekt 1

Herbstsemester 2018

Projektname:	Klima Viewer	
Firma:		
Studierende:		
Betreuer:	Firma:	BFH: J. Wolfgang Kaltz
Kurzbeschreibung:	<p>In diesem Projekt soll eine Applikation „Klima Viewer“ entwickelt werden, die Klimaveränderungen über die letzten Jahre sichtbar macht. Der Benutzer soll für verschiedene Kennzahlen (Temperatur, Windgeschwindigkeit usw.) eine graphische Darstellung über einen bestimmten Zeitraum erhalten, samt Veränderungen über diesen Zeitraum.</p> <p>Als Datenbasis kann z.B. WorldClim (http://worldclim.org) verwendet werden.</p>	
Technologie:	Java, Web Applikation, oder App	

9 Versionskontrolle

Version	Datum	Beschreibung	Autor
X0.1	26.09.2018	Dokument erstellt	Ohran Mujkic
X0.2	27.09.2018	Zweck des Dokuments erstellt	Ohran Mujkic
X0.3	28.09.2018	Vision erstellt	Ivo Kozina
X0.4	28.09.2018	Projektziele ermittelt	Ohran Mujkic
X0.5	10.10.2018	Prozessumfeld	Ivo Kozina
X0.6	10.10.2018	Systemumfeld	Ohran Mujkic
X0.7	10.10.2018	Randbedienung	Ohran Mujkic
X0.8	17.10.2018	Anforderung angefangen	Ivo Kozina
X0.9	07.11.2018	Anforderung erstellt	Ohran Mujkic
X0.91	14.11.2018	Überarbeitung der Anforderungen	Ivo Kozina
V1.0	30.11.2018	Abschlussarbeiten am ganzen Dokument und freigabe	O. Mujkic und I. Kozina