|  |
| --- |
| ildergebnis fÃ¼r klima |
| Abschlussdokumentation  Klima-Viewer  **Projektteam KlimaViewer**  Ivo Kozina  Ohran Mujkic  Version 1.0  Bern, 23. Dezember 2018 |
|  |

Inhaltsverzeichnis

[1 Zweck des Dokuments 3](#_Toc533354821)

[2 Kontext 3](#_Toc533354822)

[3 Anforderungsspezifikation 3](#_Toc533354823)

[3.1 Funktionale Anforderungen 3](#_Toc533354824)

[**3.1.1 Use Case F1** 3](#_Toc533354825)

[**3.1.2 Use Case F2** 4](#_Toc533354826)

[**3.1.3 Use Case F3** 4](#_Toc533354827)

[**3.1.4 Use Case F4** 5](#_Toc533354828)

[3.2 Design 6](#_Toc533354829)

[3.3 Datenmodell 6](#_Toc533354830)

[**3.3.1 Datenbankmodellierung** 6](#_Toc533354831)

[**3.3.2 Datenarchivierung** 6](#_Toc533354832)

[**3.3.3 Klassendiagram** 7](#_Toc533354833)

[3.4 Namenkonventionen 8](#_Toc533354834)

[4 Ausblick 9](#_Toc533354835)

[6 Glossar 10](#_Toc533354836)

[7 Abildungsverzeichnis 11](#_Toc533354837)

[8 Anhang 12](#_Toc533354838)

[8.1 Projektbeschreibung 12](#_Toc533354839)

[9 Versionskontrolle 13](#_Toc533354840)

# Zweck des Dokuments

Der Hauptzweck dieses Dokument ist es die erreichten Ziele sowie die Anforderungen für das Projekt „KlimaViewer“ aufzuführen und zu vergleichen, sowie die verschiedenen Arbeitsschritte und Entscheidungen näher zu bringen.

# Kontext

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.

# Anforderungsspezifikation

## 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. |

### **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/ Vorbedienung | Der Benutzer gibt in seinem Internetbrowser die URL von Klimaviewer an. |
| Ergebnis/ Nachbedienung | 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 |

### **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 Diagram in Form eines Linien- oder Balkendiagram 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. |

### **Use Case F3**

|  |  |
| --- | --- |
| Nr. | F3 |
| Kurzbeschreibung | Der Benutzer kann über die Rubrik "Live Weather" akutelle 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. |

### **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. |

## Design

SCREENSHOTS

## Datenmodell

### **Datenbankmodellierung**

Die folgende Abbildung zeigt ein einfaches Modell von der Datenbank. Es handelt sich hier um ein Entitiy-Relationship-Diagram, was üblich als erster. Dieses Diagram kann sich im Verlauf der Realisierung noch verändern.



Abbildung 6: ERD

Beschreibung zu den Tabellen:

Location : Tabelle, welche die verschiedenen Ortschaften beinhaltet

* id : Primary Key, welcher von openweathermap.org genommen wurde
* name : Name der Ortschaft
* zip : Postleitzahl der Ortschaft
* geolocation : Die geografische X- und Y-Koordinate der Messstation innerhalb der Ortschaft
* addedDate : Datum der ersten Initialisierung

WeatherRecord : Tabelle, welche die Wetterdaten der Ortschaften von gemesse Zeitpunkte beinhaltet

* id : Foreign Key, welcher von openweathermap.org genommen wurde
* date : Datum der Messung
* temperature : Die gemessene Temperatur in Kelvin
* wind: Die gemessene Windstärke in m/s
* pressure : Der gemessene Luftdruck in hPa
* precipitation : Der gemessene Niederschlag mm pro Quadratmeter
* response : Die komplette Antwort auf die Abfrage zur aktuellen Wetterlage.

### **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

### **Klassendiagram**

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

Abbildung 7: Klassendiagramm

Projektstruktur

Backend

* Api
  + node\_modules : Speicherort ganzen Library-Package
  + DB .js : URL zur Datenbank
  + package.json : Projekt-Konfiguration
  + server.js : Backend, das eine Verbindung mit der Datenbank herstellt und die Abfragen erhält
* App
  + node\_modules : Speicherort ganzen Library-Package
  + package.json : Projekt-Konfiguration
  + e2e : End-to-End Testing
  + models : Enthält die Models, welche eine Abbildung der Tabellen in der Datenbank sind
  + routes : Der API-Service, welcher die Abfragen empfängt und weiter bearbeitet. Aufgeliedert nach Tabellen
  + src : Source Code der Webapp
    - app
      * services : Services beinhalten verschiedene Funktione, die immer wieder verwendetet werden.
      * Live-weather : Logik und View von der LiveWeather-Seite
      * Historical : Logik und View von der Historical-Seite
      * About: Logik und View von der About-Seite
      * Location.ts : Interface für Location-Objekt
      * WeatherRecord.ts : Interface für WeatherRecord-Objekt
      * App-routing.module.ts : Routing der Web-App
      * App.component.ts : Logik Appübergreifend
      * App.component.html : View Appübegreifend
      * App.module.ts : Alle importierten Libraries
      * App.component.scss : Stylesheet Appübergreifend
      * App.comonent.spec.ts : Unit-Test Appübergreifend
    - Assets : Ablage für Bilder
    - Environments : Konfiguration für verschiedene Enviroments
    - Tests.ts : Tests-Config

Screenshot von Package-Explorer von Visual Studio

## Namenkonventionen

Wir haben in der Namenskonvention an den üblichen Konventionen des Frameworks gehalten. Dazu haben wir die Styling Guides befolgt, welche von Angular zu Verfügung stehen <https://angular.io/guide/styleguide>

# Ausblick

# 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-Architekturestyle meistens für Webservices. |
| JSON | JavaScript Object Notation – Dateiformat meisens 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. |

# Abildungsverzeichnis

[Abbildung 1: Prozessumfeld 4](file:////Users/ohranmujkic/Desktop/Anforderungsspezifikation_KlimaViewer.docx#_Toc531383432)

[Abbildung 2: Systemumfeld 4](file:////Users/ohranmujkic/Desktop/Anforderungsspezifikation_KlimaViewer.docx#_Toc531383433)

[Abbildung 3: Mockup Dashboard 9](#_Toc531383434)

[Abbildung 4: Mockup Live-Weather 9](#_Toc531383435)

[Abbildung 5: Mockup About 10](#_Toc531383436)

[Abbildung 6: ERD 12](#_Toc531383437)

[Abbildung 7: Klassendiagramm 12](#_Toc531383438)

# Anhang

## Projektbeschreibung



# 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 |
| V1.0 | 30.11.2018 | Abschlussarbeiten am ganzen Dokument und Freigabe | O. Mujkic und I. Kozina |