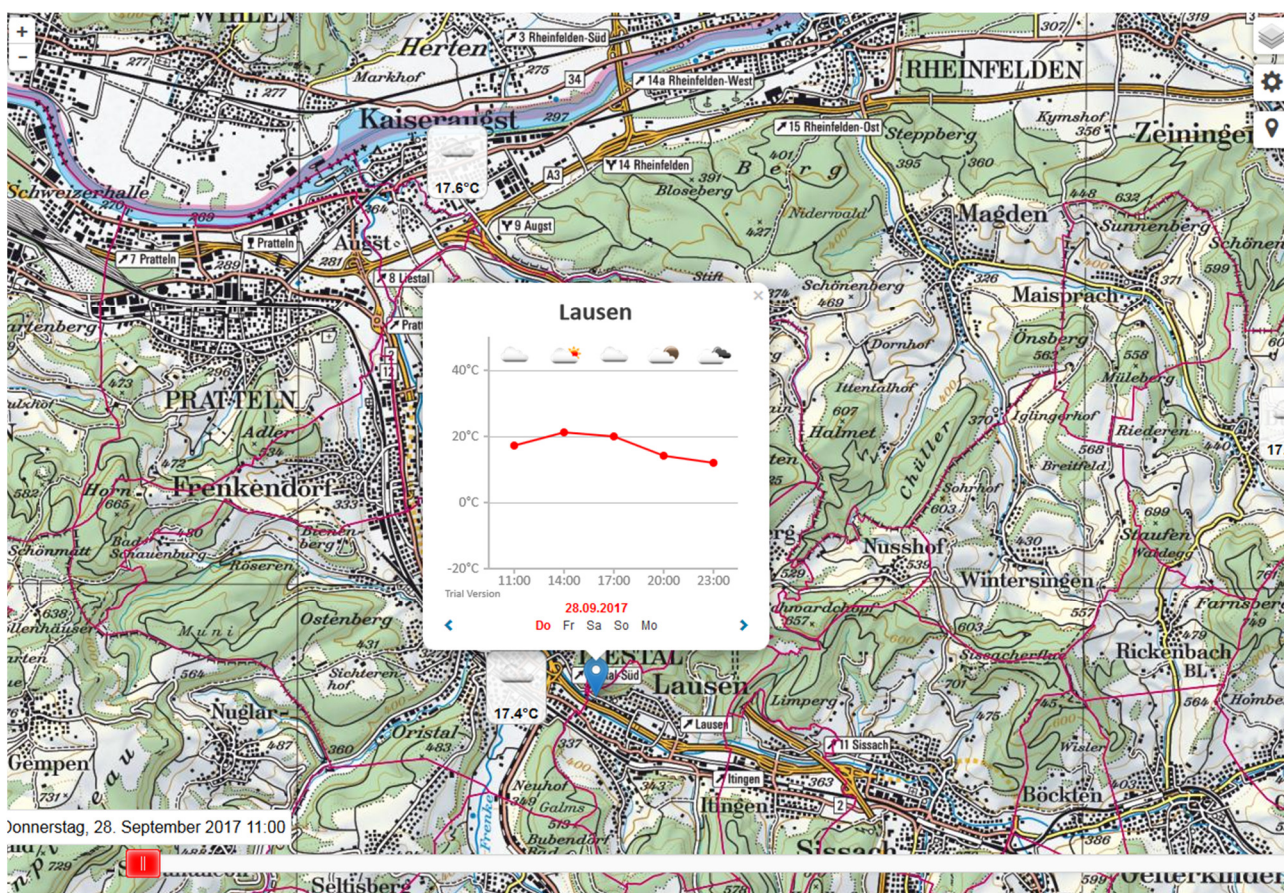


Liestal, Datum / Timo Schlatter  
 Registernummer: 12.07

## WetterMap



Änderungen am Dokument:

Datum:	Version:	Änderung:	Bearbeiter:
23.05.2017	0.1	Start	Sc
29.05.2017	0.1	Mockups / Konzept / Zeitplan	Sc
30.05.2017	0.2	Überarbeitungen, Zusätze	Sc
02.06.2017	0.2	Überarbeitungen, Weather map Layers	Sc
23.08.2017	0.5	Begonnen mit der Doku über die Realisierung	Sc
1.9 – 7.9.2017	0.9	Realisierungsabschnitt	Sc
10.9. - 21.9.2017	1.0	Tests / Fazit / Letzte Korrekturen / Titelbild	Sc

Dokumentpfad:

H:\agi\12\_Geoinformationssystem\07\_Praktika\_und\_Berufsbildung\Informatik\_Lehrlinge\Spielwiese\Timo\P12 - WetterMap\Dokumentation.docx

## Inhaltsverzeichnis

<b>1. Einleitung .....</b>	<b>4</b>
<b>2. Ausgangslage .....</b>	<b>4</b>
2.1 Aufgabe .....	4
2.2 Ressourcen .....	4
<b>3. Grundlagenbeschaffung .....</b>	<b>5</b>
3.1 Datenherkunft.....	5
3.2 Analyse der API .....	5
3.2.1 Current weather data (aktuelle Wetterdaten) .....	5
3.2.2 5 day / 3 hour forecast (5 Tage / alle 3 Stundenprognose).....	7
3.2.3 Weather map layers (Wetterkartenschichten) .....	8
3.2.4 Weitere APIs .....	8
3.3 API URL-Parameter .....	9
3.3.1 Notwendige Parameter .....	9
3.3.2 Zusätzliche Parameter .....	9
3.4 Darstellung der Wetterinformationen (Leaflet).....	10
3.5 Anpassungen der Ausgabewerte.....	10
3.5.1 Windgeschwindigkeit .....	10
3.5.2 Windrichtung .....	11
3.5.3 Zeitstempel.....	11
3.5.4 Wetterstationen.....	11
<b>4. Lösungsvorschlag.....</b>	<b>11</b>
4.1 Startseite .....	11
4.2 Wetterkarte.....	12
4.2.1 Darstellung Wetterangaben per Klick.....	13
4.2.2 Darstellung Wetterprognose per Klick .....	14
4.2.3 Wettericons .....	14
4.3 Optionsmenü / Wettermodiauswahl.....	15
4.3.1 Aktuelles Wetter .....	15
4.3.2 Wetterprognose .....	16
4.4 Vordefinierte Städte / Gemeinden .....	16
4.4.1 Basel-Landschaft .....	16
4.4.2 Welt .....	16

4.5	Benutzerinteraktion .....	16
4.6	Zeitplan - Soll .....	17
<b>5.</b>	<b>Realisierung .....</b>	<b>18</b>
5.1	Startseite .....	18
5.2	Wetterkarte .....	18
5.2.1	Allgemein .....	18
5.2.2	Modus Wetter .....	20
5.2.3	Modus Wetterprognose .....	20
5.3	Optionsmenü / Wettermodi .....	21
5.4	Benutzerinteraktion .....	21
5.5	Zeitplan - Ist .....	22
5.6	Repository .....	22
<b>6.</b>	<b>Tests .....</b>	<b>23</b>
6.1	Startseite .....	23
6.2	Wetterkarte .....	23
<b>7.</b>	<b>Fazit .....</b>	<b>23</b>

## 1. Einleitung

Im Projekt „WetterMap“ geht es um die Erstellung einer Applikation, welche eine Wetterkarte darstellen soll. Dies soll hauptsächlich eine übersichtliche Karte für aktuelle Wettergeschehnisse werden, aber auch eine Prüfung auf die Verlässlichkeit der Wetterdaten von [openweathermap](http://openweathermap.org). Die Applikation wird für Smartphones, Tablets und als Webseite zur Verfügung stehen.

## 2. Ausgangslage

### 2.1 Aufgabe

Es ist eine App zu erstellen, die das aktuelle Wettergeschehen, sowie die Prognose darstellt auf der Grundlage von <http://openweathermap.org/>. Dargestellt wird das Wetter auf einer Karte mit Symbolen, welche die Wettersituationen (schön, bedeckt, Regen, etc.) anzeigen. Als Hintergrundkarte sollen die farbige Grundkarte von GeoView BL und die weltweite OSM-Grundkarte zur Auswahl stehen. Der Benutzer soll in der App wählen können, ob er das aktuelle Wetter oder eine Prognose angezeigt bekommen möchte.

### 2.2 Ressourcen

Die App soll mit dem Phonegap Framework erstellt werden. Dadurch kann mit HTML/CSS/JS gearbeitet werden. Mithilfe des Phonegap Tools kann ein Webserver mit dem aktuellen Projekt gehostet werden. Das Phonegap Developer Tool ermöglicht den Zugriff auf den Webserver via IP-Adresse und Portangabe, dadurch kann das Projekt sehr einfach und immer aktuell auf einem Smartphone getestet werden (Vorausgesetzt man befindet sich im gleichen Netzwerk).

Die Hintergrundkarte wird mithilfe der JavaScript-Bibliothek Leaflet (<http://leafletjs.com/>) dargestellt. Die Karten werden per WMS geladen:

- OSM: <http://openstreetmap.org>
- GeoView: <http://geowms.bl.ch>

Die Wetter- und Wetterprognosedaten können durch einen Aufruf der *openweathermap*- API abgefragt werden.

**API** API steht für *Application Programming Interface* (Anwendungsprogrammierschnittstelle). Mithilfe dieser Schnittstelle können Daten aus einer Anwendung genutzt werden.

Zur Entwicklung stehen der Laptop und das Entwickler-Tablet Google Nexus 7 zur Verfügung.

Da eine Startseite notwendig für die Auswahl der Karte ist, kann dafür die Library von Bootstrap verwendet werden. Diese eignet sich sehr gut, da alle Objekte grösstenteils *responsive* sind und eigenes Vorwissen ebenfalls vorhanden ist.

Für die Verwendung von Bootstrap ist die JavaScript-Library jQuery notwendig.

## 3. Grundlagenbeschaffung

Für den Anfang werden Informationen über die API benötigt. Damit man diese API nutzen darf, wird ein API-Key benötigt, welcher die Abfragen ermöglicht. Ohne einen Key werden keine Daten zurückgesendet. Diesen erhält man bei der Registrierung auf der *openweathermap* Webseite. Dieser wird unter den **API-Keys** des Accounts dargestellt.

**Achtung:** Unter gewissen Kapiteln befinden sich Links zu Beispielen für die *openweathermap* API. Der API-Key wurde bei allen entfernt. Damit die Links funktionieren, muss der eigene API-Key verwendet werden. Diesen fügt man in der URL hinter dem *&appid=* hinzu.

### 3.1 Datenherkunft

Die Daten kommen von verschiedenen Wetterstationen, verteilt auf der ganzen Welt. Leider gibt es bislang keine Möglichkeit, die Ortschaft der Wetterstationen herauszufinden. Es ist ebenfalls unklar, wie die Prognosen berechnet werden.

Wetterstationen können mit einer weiteren API selbst hinzugefügt werden und tragen zur besseren Wettervorhersage bei.

### 3.2 Analyse der API

Bei einem Aufruf der API werden verschiedene Daten im JSON-Format zurückgesendet. Diese gilt es zu analysieren.

**JSON** JSON steht für *JavaScript Object Notation*. Es ist ein schlankes Datenaustauschformat, welches für Menschen einfach zu schreiben ist und für Maschinen einfach zu analysieren. Ausserdem ist es unabhängig der Programmiersprache anwendbar, was es zum idealen für den Datenaustausch macht.

In dieser Dokumentation werden die folgenden drei APIs betrachtet:

- *Current weather data*
- *5 day / 3 hour forecast*
- *Weather map layers*

**Hinweis** In den folgenden zwei Kapiteln wird des Öfteren „Interner Parameter“ verwendet. Diese Daten werden von *openweathermap* intern genutzt und es sind keine weiteren Nutzungsangaben vorhanden.

#### 3.2.1 Current weather data (aktuelle Wetterdaten)

Auf den folgenden Bildern werden die Resultate eines *By Coordinates* (Koordinaten) -Abrufes für die Region Liestal zu den aktuellen Wetterdaten betrachtet:

**Link** <http://api.openweathermap.org/data/2.5/weather?lat=47.48&lon=7.73&units=metric&lang=de&appid=>

```

▼ coord:
  lon:      7.73
  lat:      47.48

```

Koordinaten des Bereiches:

- lon: Longitude-Wert
- lat: Latitude-Wert

```

weather:
  0:
    id: 800
    main: "Clear"
    description: "clear sky"
    icon: "01d"
    base: "stations"

```

Wetterinformationen:

- id: Eindeutige Identifikation (Wettertyp)
- main: Hauptanzeige Wetter
- description: Beschreibungen Wetter
- icon: Icon ID
- base: Interner Parameter der API

```

main:
  temp: 296.97
  pressure: 1020
  humidity: 47
  temp_min: 296.15
  temp_max: 298.15
  visibility: 10000

```

Temperaturangaben:

- temp: Temperatur (Kelvin)
- pressure: Luftdruck
- humidity: Luftfeuchtigkeit
- temp\_min: Minimaltemperatur (Abweichung)
- temp\_max: Maximaltemperatur (Abweichung)
- visibility: Sichtweite in Meter

```

wind:
  speed: 5.1
  deg: 250

```

Wind:

- speed: Windgeschwindigkeit in m/s (Meter pro Sekunde)
- deg: Windrichtung in Gradmassen (meteorologisch)

```

clouds:
  all: 0
  dt: 1495537200

```

Bewölkung:

- all: Bewölkung in Prozent
- dt: Zeit der Datenberechnung im Unix Format (UTC)

**UTC** UTC steht für *Universal Time Coordinated* (Koordinierte Weltzeit) und ist die heute gültige Weltzeit.

```

▼ sys:
  type:      1
  id:        5646
  message:   0.0068
  country:   "CH"
  sunrise:   1495510965
  sunset:    1495566582
  id:        6458854
  name:      "Bezirk Liestal"
  cod:       200
  
```

Systeminformationen:

- type: interner Parameter
- id: interner Parameter
- message: interner Parameter
- country: Landeskürzel
- sunrise: Sonnenaufgang, Zeitangabe im Unix Format (UTC)
- sunset: Sonnenuntergang, Zeitangabe im Unix Format (UTC)
- id: Eindeutige Identifikation der Stadt
- name: Stadt - oder Bezirksname
- cod: interner Parameter

### 3.2.2 5 day / 3 hour forecast (5 Tage / alle 3 Stundenprognose)

Der Aufruf für das aktuelle Wetter sieht so aus:

<http://api.openweathermap.org/data/2.5/weather?lat=47.48&lon=7.73&units=metric&lang=de&appid=>

Um die API der Wetterprognose zu benutzen, muss in der URL des aktuellen Wetters das Schlüsselwort **weather** durch **forecast** ersetzt werden:

<http://api.openweathermap.org/data/2.5/forecast?lat=47.48&lon=7.73&units=metric&lang=de&appid=>

Die Antwort ist aufgeteilt in eine Liste, welche mehrere Einträge besitzt, die das Wetter alle drei Stunden für fünf Tage voraussagt.

Die Daten in den Einträgen sind fast die gleichen wie bei den aktuellen Wetterdaten, allerdings gibt es hier einige mehr. Diese wurden rot markiert:

```

▼ main:
  temp:      297.63
  temp_min:  295.397
  temp_max:  297.63
  pressure:  970.81
  sea_level: 1033.35
  grnd_level: 970.81
  humidity:  56
  temp_kf:   2.24
  
```

- sea\_level: Luftdruck auf Seehöhe in hPa (Hektorpascal)
- grnd\_level: Luftdruck auf Bodenhöhe in hPa
- temp\_kf: interner Parameter

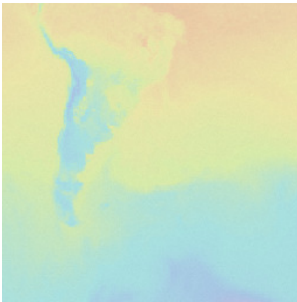
```
sys:
  pod: "d"
  dt_txt: "2017-05-23 15:00:00"
```

- pod: interner Parameter
- dt\_txt: Zeitpunkt der Datenberechnung

### 3.2.3 Weather map layers (Wetterkartenschichten)

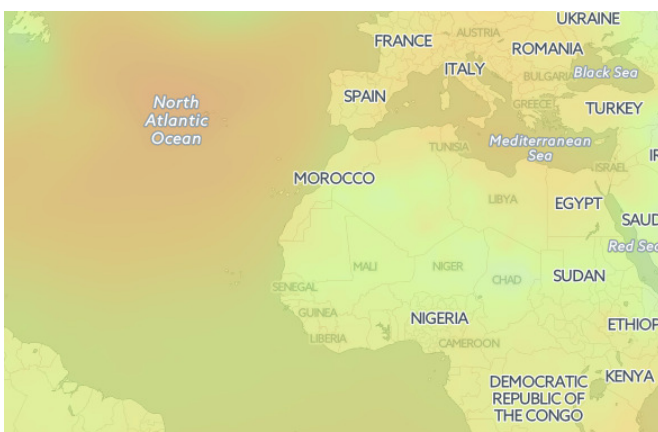
Mithilfe der API *Weather map layers* können verschiedene *Weatherbaselayer* (Wetterbasisschichten) und *Weatheroverlays* (Wetterüberlagerungsschichten) auf einer Karte hinzugefügt werden. Diese sind in verschiedene, quadratische, sogenannte *tiles* (Kacheln) unterteilt. Ein einzelnes *tile* könnte so aussehen:

Link [http://tile.openweathermap.org/map/temp\\_new/3/5,4342/1,234.png?appid=7ee4d709dbeeebc029df559fddf4e146](http://tile.openweathermap.org/map/temp_new/3/5,4342/1,234.png?appid=7ee4d709dbeeebc029df559fddf4e146)



Hier ist ein kleiner Ausschnitt einer ganzen Karte, welche die Temperatur durch verschiedene Verfärbungen ausdrückt:

Link <http://openweathermap.org/weathermap?basemap=map&cities=false&layer=temperature&lat=30&lon=-20&zoom=3>



Diese API ist ausserdem darauf ausgelegt, mit Leaflet zu funktionieren.

### 3.2.4 Weitere APIs



Openweathermap besitzt zu den oben beschriebenen APIs noch viele weitere. Allerdings sind jene nur mit einer kostenpflichtigen Mitgliedschaft verfügbar oder sind in diesem Projekt nicht weiter von Nutzen. Aufgrund dessen wird hier nicht weiter darauf eingegangen.

### 3.3 API URL-Parameter

Die Abfragen der API können in der URL angepasst werden. Dafür gibt es verschiedene Parameter, welche die Ausgaben verändern können.

Diese können nach dem „?“ in der URL hinzugefügt werden. Bei mehreren Parametern müssen sie mit dem „&“ getrennt werden. Bei nur einem ist dies nicht notwendig.

**Beispiel** <http://api.openweathermap.org/data/2.5/forecast?lat=47.48&lon=7.73&lang=de&appid=&units=metric>

Gewisse Parameter **müssen** gegeben sein. Die notwendigen Parameter können nicht miteinander kombiniert werden. Diese Parameter sind notwendig, damit eine korrekte Antwort vom Server gegeben wird. Jedoch können die zusätzlichen Parameter beliebig mit den notwendigen Parametern kombiniert werden.

#### 3.3.1 Notwendige Parameter

**Stadtname** <http://api.openweathermap.org/data/2.5/forecast?q=Liestal,ch&appid=>

**Stadt-ID** <http://api.openweathermap.org/data/2.5/forecast?id=524901&appid=>  
Stadt-ID Liste von *openweathermap*: <http://bulk.openweathermap.org/sample/>

**Koordinaten** <http://api.openweathermap.org/data/2.5/weather?lat=47.48&lon=7.73&appid=>

**ZIP-Code** <http://api.openweathermap.org/data/2.5/weather?zip=4410,ch&appid=>

#### 3.3.2 Zusätzliche Parameter

**Format** Standardmässig JSON, zusätzlich kann XML (Extensible Markup Language) ausgewählt werden.  
<http://api.openweathermap.org/data/2.5/weather?zip=4410,ch&mode=xml&appid=>

**Temperatur** Standardmässig in Kelvin, zusätzlich in Celsius(*metric*) oder Fahrenheit(*imperial*) möglich.  
<http://api.openweathermap.org/data/2.5/weather?zip=4410,ch&units=imperial&appid=>  
<http://api.openweathermap.org/data/2.5/weather?zip=4410,ch&units=metric&appid=>

**Sprache** Die Sprachausgabe bezieht sich allein auf die *description* - Ausgabe.  
<http://api.openweathermap.org/data/2.5/weather?zip=4410,ch&lang=de&appid=>

### 3.4 Darstellung der Wetterinformationen (Leaflet)

**Vordefinierte Städte** Die JavaScript-Bibliothek Leaflet wird für die Darstellung der Karte verwendet. Um eine schöne Übersicht bei Applikationsstart anzuzeigen, können verschiedene Textblasen von Leaflet verwendet werden. Diese Textblasen sollen eine kleine, bündige Angabe über das Wetter bei wichtigen Ortschaften anzeigen. Dies kann mithilfe der API für die Wetterkartendarstellung ermöglicht werden.

Falls die Angaben der API nicht ausreichen sollten, müssen weitere Informationen selbst definiert werden.

**„On-Click Methode“** Zusätzlich kann ebenfalls per Klick auf ein Gebiet eine Textblase erscheinen, welche die benötigten Informationen über das Gebiet darstellt. Dies kann einfach erarbeitet werden, da sowohl Leaflet als auch openweathermap mit Longitude - und Latitude Werten arbeitet. Zusätzlich sollen bei Applikationsstart bereits Angaben zu wichtigen Ortschaften vorhanden sein.

Bei weitem Hinauszoomen wäre es schön, wenn die Informationen in den Textblasen verschwinden und stattdessen nur noch das Wettericon angezeigt wird. Leaflet arbeitet mit verschiedenen Zoomstufen, aufgrund derer die Anzeige für die verschiedenen Zoomstufen abgeändert werden kann.

### 3.5 Anpassungen der Ausgabewerte

Gewisse Daten, welche von der API zurückgesendet werden, können in dieser Art nicht für den Benutzer verwendet werden. Die Rede ist von Angaben wie die Windgeschwindigkeit in Meter pro Sekunde oder der Windrichtung 170 Grad. Solche Daten müssen benutzerfreundlich und verständlich dargestellt werden. Deshalb werden diese Daten umgewandelt, mithilfe der Skalen welche in den nächsten Kapiteln beschrieben werden.

#### 3.5.1 Windgeschwindigkeit

Für die Windgeschwindigkeitsskala wird von der **Beaufort-Skala**<sup>1</sup> Gebrauch gemacht.

Windgeschwindigkeit	Bezeichnung der Windstärke
0.0 - < 0.3 m/s	Windstille
0.3 - < 1.6 m/s	Leichter Zug
1.6 - < 3.4 m/s	Leichte Brise
3.4 - < 5.5 m/s	Schwache Brise
5.5 - < 8.0 m/s	Mässige Brise
8.0 - < 10.8 m/s	Frische Brise
10.8 - < 13.9 m/s	Starker Wind
13.9 - < 17.2 m/s	Steifer Wind
17.2 - < 20.8 m/s	Stürmischer Wind
20.8 - < 24.5 m/s	Sturm
24.5 - < 28.5 m/s	Schwerer Sturm
28.5 - < 32.7 m/s	Orkanartiger Sturm
> 32.7 m/s	Orkan

<sup>1</sup> <https://de.wikipedia.org/wiki/Beaufortskala>

### 3.5.2 Windrichtung

Die Windrichtungen werden in 16 Richtungen unterteilt.

Abkürzung	Windrichtung	Grad
N	Nord	348.75 - < 11.25
NNE	Nordnordost	11.25 - < 33.75
NE	Nordost	33.75 - < 56.25
ENE	Ostnordost	56.25 - < 78.75
E	Ost	78.75 - < 101.25
ESE	Ostsüdost	101.25 - < 123.75
SE	Südost	123.75 - < 146.25
SSE	Südsüdost	146.25 - < 168.75
S	Süd	168.75 - < 191.25
SSW	Südsüdwest	191.25 - < 213.75
SW	Südwest	213.75 - < 236.25
WSW	West südwest	236.25 - < 258.75
W	West	258.75 - < 281.25
WNW	Westnordwest	281.25 - < 303.75
NW	Nordwest	303.75 - < 326.25
NNW	Nordnordwest	326.25 - < 348.75

### 3.5.3 Zeitstempel

Die Zeitausgabe der API wird nur im Unix-Zeitformat ausgegeben. Die Unixzeit läuft seit dem 1. Januar 1970 und zählt die Zeit in Sekunden, da dies für Computerprogramme viel einfacher zu lesen ist. Allerdings stiftet dieses Format beim Menschen nur Verwirrung. Die Zeit muss in ein Menschenlesbares Format gestellt werden.

Diese Umwandlung kann mithilfe von JavaScript und den Datumsobjekten durchgeführt werden.

Link <https://stackoverflow.com/questions/847185/convert-a-unix-timestamp-to-time-in-javascript>

### 3.5.4 Wetterstationen

Momentan werden keine Listen über die gebrauchten Wetterstationen an Benutzer herausgegeben. Laut Angabe eines Mitarbeiters sollte dies im Verlauf dieses Jahres noch möglich sein.

Link <https://openweathermap.de/customer/portal/questions/16771503-is-there-a-list-of-stations->

## 4. Lösungsvorschlag

### 4.1 Startseite

Mit der Startseite beginnt die ganze Applikation.

Im oberen Bereich befindet sich ein Panel mit dem Logo bzw. dem Titel der App.

Danach folgt die Kartenauswahl. Zwischen den Karten kann hin – und hergezogen (Swipen) werden. Zur Auswahl stehen momentan nur die Basel-Landschaftliche Karte von GeoView und die Weltkarte von OpenStreetMap. Das Laden der Karte wird mit einem Klick auf die Karte erzielt.

Zusätzlich befindet sich ein Button für die Bedienungsanleitung und einen für das Impressum unterhalb der Auswahl.

Ganz unten befindet sich die Fusszeile mit dem Copyright.

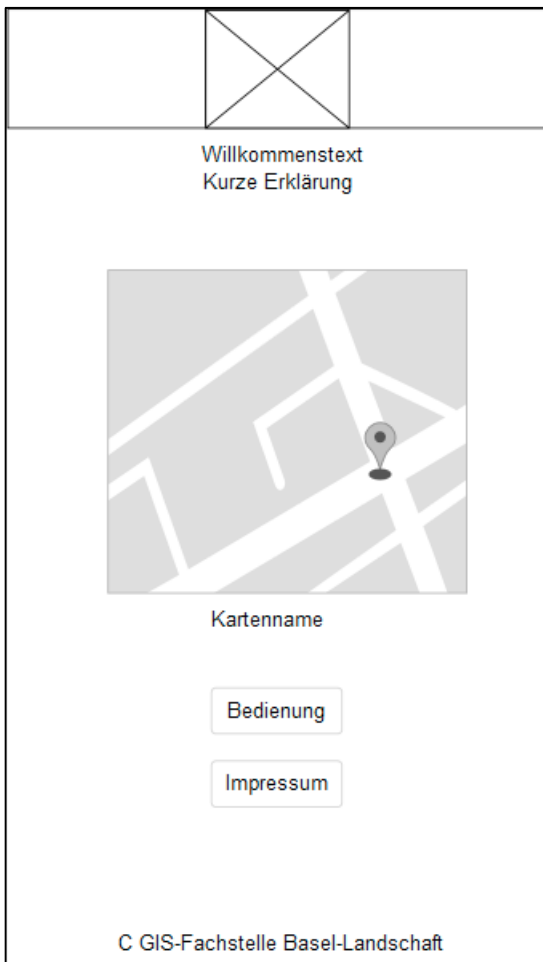


Abbildung 1: Startseite WetterMap

## 4.2 Wetterkarte

**Vordefinierte Städte** Während dem Laden der Wetterkarte sollen die vordefinierten Städte (Siehe 3.4) bereits angezeigt werden.

Diese werden in einer JSON-Liste eingetragen und vor dem Laden der Karte laufen die Abfragen durch oder sie werden mithilfe der „*weather map layer*“ - API dargestellt.

**„On-Click“ Methode** Jegliche weiteren Informationen zum aktuellen Gebiet können mit einem Klick angezeigt werden. Mit einer Funktion, welche auf den Klick auf die Karte reagiert, werden die Latitude und Longitude Werte abgerufen und für die jeweiligen Werte wird eine Abfrage gestartet, welche die Wetterdaten antwortet.

**Wetter-Daten** Zur besseren Übersicht werden nur die Grunddaten dargestellt. Da aber weitere Informationen notwendig sind, können diese mithilfe der Checkboxes im Optionsmenü aktiviert oder auch deaktiviert werden.

**Wetter-modi** Standardmässig wird der Modus „aktuelles Wetter“ beim Start angezeigt. Die Wettermodi können im Optionsmenü geändert werden (Das Optionsmenü und die Wettermodi werden im Kapitel 4.3 ausführlicher beschrieben). Oben Links befindet sich ein Symbol, um das Optionsmenü zu öffnen.

**Weather map layers** Die Wetteroverlays (siehe 3.2.3) werden ebenfalls eingebunden. Folgende Overlays werden zur Verfügung stehen: Städte / Bewölkung / Temperatur / Wind / Niederschlag. Diese lassen sich in der rechten, oberen Ecke ein – und ausschalten.

**Link** <https://github.com/buche/leaflet-openweathermap>

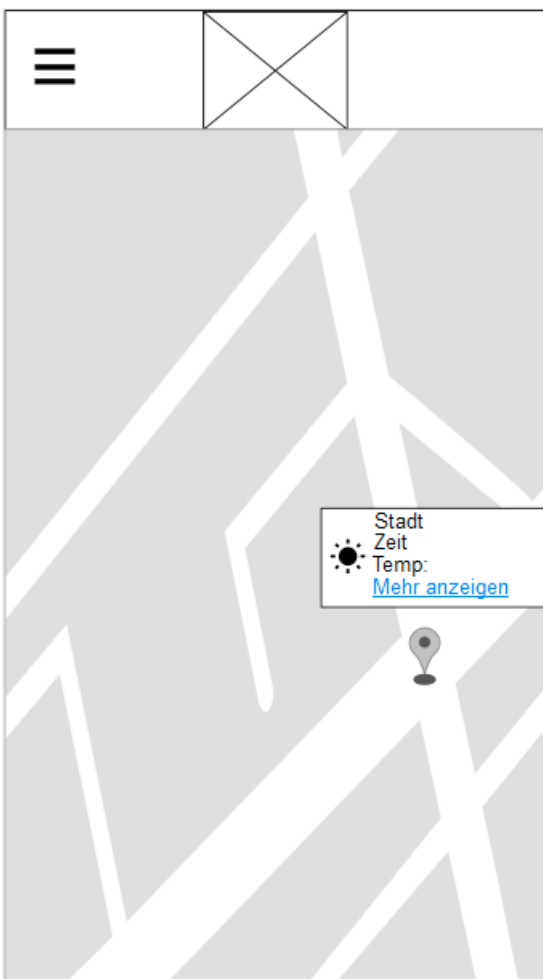


Abbildung 2: Wetterkarte im Modus „aktuelles Wetter“

### 4.2.1 Darstellung Wetterangaben per Klick

Im Bild des Kapitel 4.2 ist bereits ein Ausschnitt der Wetterangabe. Die Wetterangabe wird sich zentriert, oberhalb der Markierung befinden. Links in der Textblase befindet sich das Icon, gefolgt von den verschiedenen Angaben.

Mit dem Aktivieren einer weiteren Information (Siehe 4.3.1) aus dem Optionsmenü vergrößert sich die Textblase und verschiebt zugleich den bisherigen Text nach oben, damit die zusätzlichen Informationen darunter Platz finden.

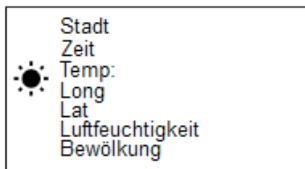


Abbildung 3: Textblase mit weiteren Informationen

### 4.2.2 Darstellung Wetterprognose per Klick

Die Wetterprognose muss anders dargestellt werden, da die Übersichtlichkeit ansonsten beträchtlich leiden würde. Dies soll mithilfe eines Zeit-Temperatur-Graphen gelöst werden. Die Wetterprognose bestimmt für die nächsten fünf Tage alle drei Stunden die Temperatur und das Wetter. Der Graph besteht aus der Zeit und der Temperaturangabe. Im Abstand von drei Stunden wird jeweils ein Balken angezeigt, welcher so hoch ist wie die aktuelle Temperatur. Das aktuelle Wetter Icon wird direkt darüber angezeigt. Mithilfe eines Schiebereglers kann man den angezeigten Tag wechseln. Die Angabe für den angezeigten Tag soll sich immer unter dem Regler befinden.

Der Graph wird ebenfalls per Klick aufrufbar sein und die vordefinierten Städte werden ebenfalls als Graphen angezeigt.

Um diesen Graphen zu zeichnen, kann Canvas verwendet werden, ein Plugin für jQuery. Dies eignet sich besonders, da jQuery in diesem Projekt ebenfalls verwendet wird. Canvas kann für Entwicklungszwecke 30 Tage lang kostenlos benutzt werden, danach wird es kostenpflichtig.

Link <http://canvasjs.com/jquery-charts/>

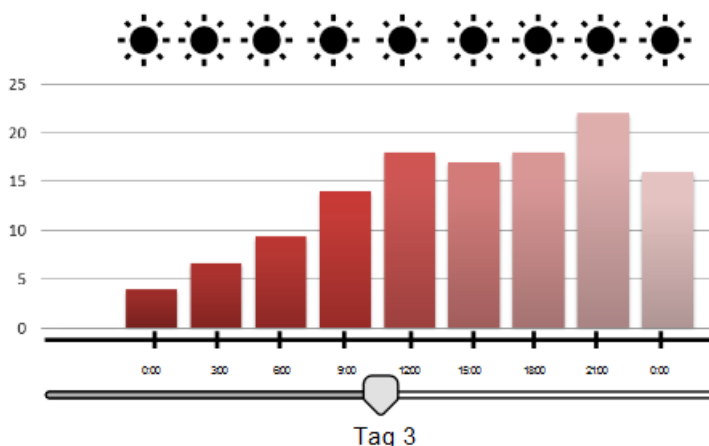


Abbildung 4: Darstellung Wetterprognose inklusive Graph

### 4.2.3 Wettericons

Da die Ausgabe der API auch immer die ID (Siehe 3.2.1) für das entsprechende Icon mitgibt, werden die Icons der openweathermap API verwendet.

Link <https://openweathermap.org/weather-conditions>

## 4.3 Optionsmenü / Wettermodiauswahl

Im Optionsmenü kann der Wettermodus mit speziellen Symbolen gewechselt werden und zusätzlich können weitere Wetterdaten ausgewählt werden, welche bei einem Klick angezeigt werden sollen. Man kann ebenfalls zurück zur Startseite wechseln. Das Optionsmenü wird in einer *Modalbox* dargestellt.

Die Informationen der Wettermodi werden in den folgenden Unterkapiteln beschrieben.

**Modalbox** Eine Modalbox agiert als eine neue Seite, jedoch ohne die aktuelle Seite zu verlassen. Sie erscheint in der Form eines Popups.

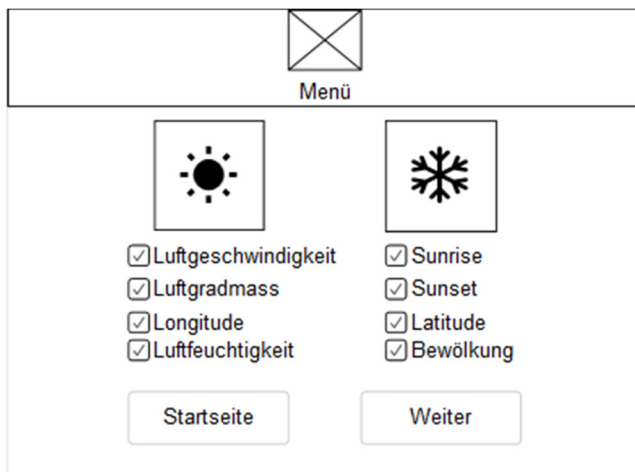


Abbildung 5: Optionsmenü mit den Wettermodi-Symbolen und den Checkboxes für die Zusatzinformationen

**Beschreibung** Für diese Applikation steht eine Auswahl von zwei verschiedenen Wetterspezifischen Datenpaketen zur Verfügung. Die Grunddaten werden sofort bei einem Klick angezeigt und die weiteren Daten können mittels Aktivierung der Checkboxes dargestellt werden.

### 4.3.1 Aktuelles Wetter

In diesem Modus werden die folgenden Informationen angezeigt:

#### Grunddaten:

- Das Icon
- Temperatur
- Stadt-/Bezirksname
- Zeitpunkt
- Wetteroverlay Temperatur / Wetteroverlay Bewölkung / Wetteroverlay Wind / Wetteroverlay Städte / Wetteroverlay Niederschlag

In der rechten, oberen Ecke befindet sich ein Layer-Menü, welches das Wechseln zwischen den Wetterlayern ermöglicht.

#### Mehr anzeigen:

Diese zusätzlichen Daten müssen im Optionsmenü ausgewählt werden, damit diese auch angezeigt werden.

- Longitude
- Latitude

- Luftfeuchtigkeit
- Bewölkung
- Windgeschwindigkeit
- Luftgradmass
- Sunrise
- Sunset

### 4.3.2 Wetterprognose

Im Wetterprognose-Modus stehen nur folgende Grunddaten zur Verfügung:

#### Grunddaten:

- Das Icon
- Graph (Temperatur-Zeit Graph)
- Stadt-/Bezirksname
- Zeitpunkt

## 4.4 Vordefinierte Städte / Gemeinden

Wie bereits beschrieben, werden einige Wetterangaben zu gewissen Städten / Gemeinden bei Applikationsstart angezeigt.

### 4.4.1 Basel-Landschaft

Bei der farbigen Grundkarte des Kantons Basel-Landschaftes werden bei allen Gemeinden die Wetterangaben angezeigt.

Hier könnte man in eine JSON-Datei alle Gemeinden mit den jeweiligen Koordinaten auflisten und daraufhin in der Applikation integrieren. Die Abfrage wird ausgeführt, bevor die Karte fertiggeladen ist.

### 4.4.2 Welt

Bei der Weltkarte sollen bei allen Hauptstädten die Wetterangaben bereits angezeigt werden. Dies wurde im Kapitel 3.4 bereits genauer beschrieben.

## 4.5 Benutzerinteraktion

Der Benutzer hat die Möglichkeit, die Karte oder den Wettermodi selber auszuwählen. Eine neue Karte oder einen anderen Wettermodi kann nicht vom Benutzer hinzugefügt werden.

Auf der Karte kann man sich frei bewegen. Allerdings besitzen die Karten Ränder, welche als Grenzen zu betrachten sind. Darüber hinauszugehen ist nicht möglich.

Der Bereich, bei welchem die Wetterinformationen dargestellt werden sollen, kann ebenfalls frei gewählt werden.

Der Zoom besitzt ebenfalls eine maximale – und minimale Zoomstufe, welche je nach Karte verschieden ist.



## 4.6 Zeitplan - Soll

Planung WetterMap		Entwicklung							
		Juli				August			
		W1	W2	W3	W4	W1	W2	W3	W4
Projektstart	<i>Projekt aufsetzen</i>								
Startseite	<i>Startseite</i>								
Karte	<i>Wetter Overlays</i>								
	<i>Webdienst Abfragen</i>								
	<i>Output (Sprechblasen / Graph) Forecast (Tag blättern)</i>								
Optionsmenü	<i>Modalbox</i>								
	<i>Modiauswahl</i>								
	<i>Weitere Informationen (Checkbox)</i>								
Letzte Korrekturen									
Präsentation									
Dokumentation	<i>(laufend)</i>								
Fazit									

Leitung:  Timo

## 5. Realisierung

### 5.1 Startseite

Die Startseite wurde fast exakt so erstellt, wie es im Lösungsvorschlag (siehe 4.1) angegeben wurde. Allerdings ist die Karte von der Startseite aus nicht mehr wählbar und es wird zusätzlich auf der Karte eine aktuelle Wetterangabe für Liestal angezeigt.

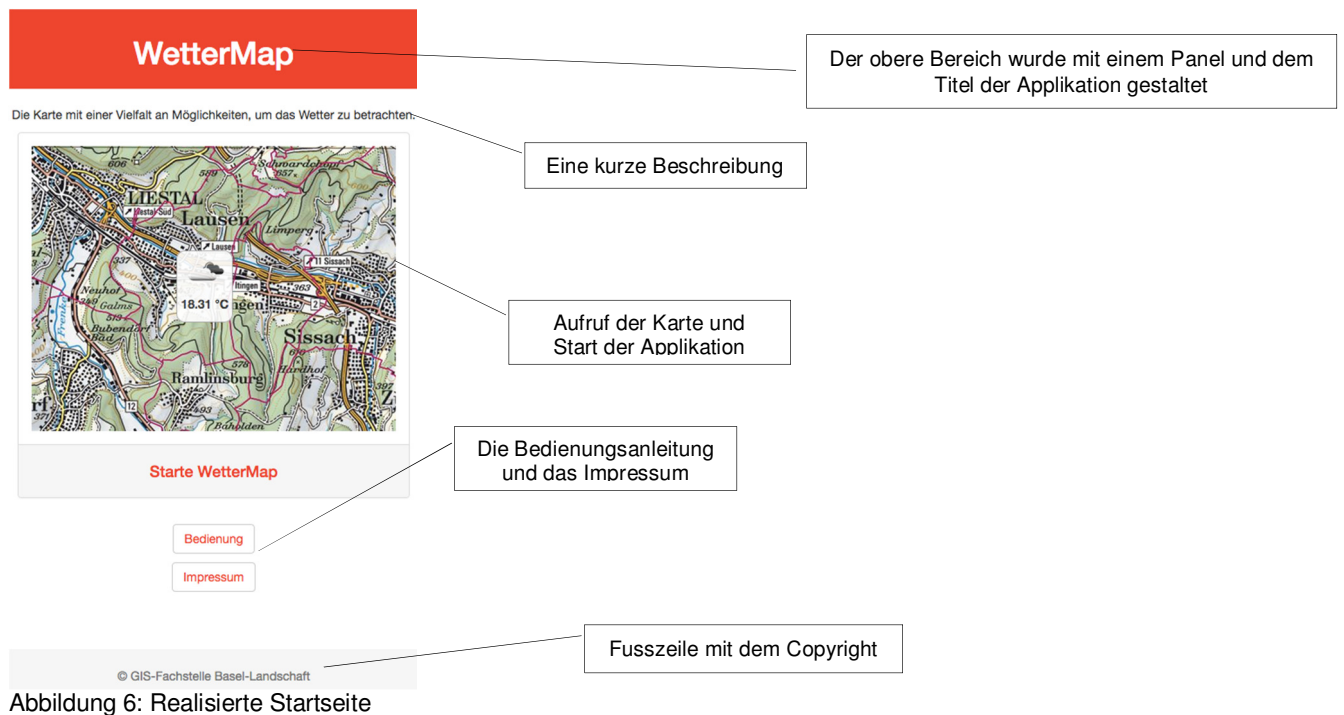


Abbildung 6: Realisierte Startseite

### 5.2 Wetterkarte

#### 5.2.1 Allgemein

Die Wetterkarten wurden grösstenteils nach dem Lösungsvorschlag erstellt. Allerdings konnten manche Punkte nicht wie gewünscht erfüllt werden bzw. es wurde eine *für-besser-empfundene* Lösung eingesetzt.

**Vordefinierte Städte** Die vordefinierten Städte bzw. Gemeinden wurden wunschgemäss eingebunden. Auf der Weltkarte wurde dies mit der „weather map layer“-API erledigt. Diese lädt je nach Zoomstufe andere Städte im Bild. Für die Basellandschaftliche Karte wurde eine JSON-Liste erstellt. Das Ganze ist dynamisch – Neue Gemeinden können mit Angabe des Namens, des Latitude - und des Longitude-Wertes der Liste hinzugefügt werden. Beim nächsten Aufruf der Karte werden die neuen Gemeinden gleich angezeigt. Dies steht dem Benutzer allerdings nicht zur Verfügung. Bei beiden Karten werden diese beim Aufruf bereits angezeigt und können mithilfe des „Layer-Control-Menü“ ab – und wieder angezeigt werden.

**„On-Click Methode“**

Bei einem Klick können einige Basisinformationen über das Wetter bzw. über die Wetterprognose angezeigt werden. Je nach ausgewähltem Modus unterscheidet sich die Ausgabe der Informationen.

Diese werden in einem Pop-up dargestellt, welches zusammen mit einem Marker auf der Karte erscheint. Das Pop-Up kann mit einem Klick auf den Marker geschlossen, aber auch wieder geöffnet werden.

**Karte(n)**

Wie bereits unter 5.1 beschrieben, steht keine Kartenauswahl auf der Startseite zur Verfügung. Die Weltkarte und die Karte des Kantons Basel-Landschaftes wurden kombiniert. Grundsätzlich startet man auf der Karte des Kantons Basel-Landschaftes. Entfernt man sich zu weit aus dem Baselland oder zoomt man zu weit hinaus, wechselt es automatisch auf die Weltkarte. Befindet man sich in der definierten Zoomstufe und im Bildausschnitt vom Baselland, wechselt man wieder zurück.

Bei dem automatischen Kartenwechsel wechselt es ebenfalls zwischen den Gemeinden – und Städteoverlays. Der Städteoverlay ist eine externe API, welche das Ganze regelt. Der Gemeindeoverlay wurde selbst entwickelt.

**Weather map layer**

Die *Weather map layer* lassen sich in der rechten, oberen Ecke aktivieren bzw. deaktivieren.

Durch den automatischen Wechsel der Karten ist ein manuelles Umstellen der *Weatherbaselayer* Welt und Baselland nicht möglich. Allerdings kann man auf die Orthofoto-Ansicht des Kantons Basel-Landschaftes wechseln. Vorausgesetzt man befindet sich nicht in der Weltkarte. Das Orthofoto ist ein massstabsgetreues Luftbild, welches zusätzlich implementiert wurde.

Die beschriebenen *Weatheroverlays* (Siehe 4.3.1 ) wurden nicht nur für den Wettermodus verfügbar geschaltet, sondern sind nun für beide verfügbar.

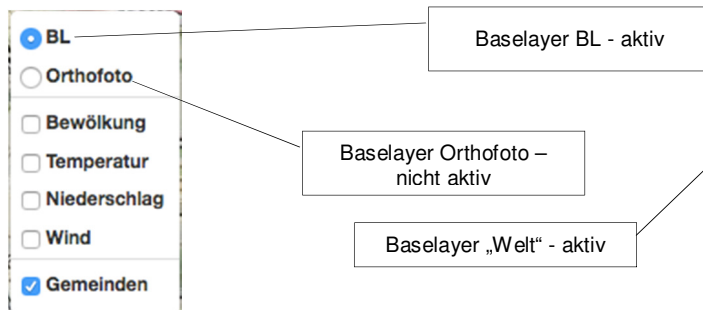


Abbildung 7: Layer-Controlmenü (BL)

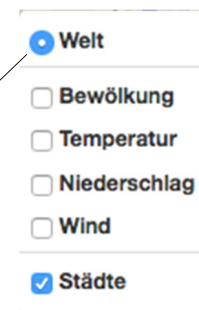


Abbildung 8: Layer-Controlmenü (Weltkarte)

**Legenden**

Für die Weatheroverlayer „Temperatur“ und „Wind“ werden Farben für die Darstellung der Stärke verwendet. Dafür gibt es zusätzliche Legenden, welche unten Links beim Aktivieren erscheinen und Aufschluss über die Farben geben. Auf dem Mobiltelefon sind die Legenden deaktiviert, da sie zu viel Platz einnehmen.

**LocateMe**

Der *LocateMe* - Button zeigt die aktuelle Position des Benutzers in der Applikation an. Der Button befindet sich oben rechts, direkt unter dem Optionsmenü - Button. Da diese Lokalisierungen nicht ganz auf den Meter genau stimmen, wird die Markierung zusätzlich umkreist. Dies gibt den ungefähren Standort an, in dem man sich befindet. Ein PopUp gibt den zugehörigen Radius in Metern an.

**„Hinauszoomen“**

Das gewünschte Verkleinern der Informationen beim Hinauszoomen (siehe 3.4) wurde nicht eingebaut. Die Textblasen mit den Informationen passen sich immer der Zoomstufe an und die vordefinierten Städte / Gemeinden besitzen nur das Icon und die Temperatur. Somit wäre dies nicht wirklich sinnvoll.

## 5.2.2 Modus Wetter

**Darstellung Wetter** Die Darstellung des Wetters wurde exakt dem Lösungsvorschlag (siehe 4.2.1) entsprechend umgesetzt.

**Wetterdaten** Zusätzlich zu den angegebenen Grunddaten wurde noch eine kleine Beschreibung des Wetters hinzugefügt. Ansonsten wurde dies nach den Vorgaben (siehe 4.3.1) umgesetzt.



Abbildung 9: „On-Click“ - Wetterinformationen



Abbildung 10: „On Click“ – mit zusätzlichen Informationen

## 5.2.3 Modus Wetterprognose

**Darstellung Wetterprognose** Die Prognose wurde wie beschrieben mit einem Graphen gelöst (siehe 4.2.2). Allerdings stellte sich ein Punkt-Linien-Graph als die bessere Option heraus. Der Graph besteht aus Temperatur und Zeit, allerdings wurde aus Platzgründen nur alle sechs Stunden ein Punkt für die Temperaturangabe dargestellt. Das aktuelle Wetter Icon wird direkt darüber angezeigt. Der Schieberegler wurde nicht beim Graphen eingesetzt, da es einfach zu klein wäre, um wirklich davon zu profitieren.

Der Tag kann mithilfe des Pfeiles nach links oder nach rechts gewechselt werden. Zwischen den Pfeilen werden das aktuelle Datum und die fünf Wochentage, welche momentan betroffen sind, angezeigt. Mit dem Wechseln des Graphen wird auch automatisch das Datum angepasst und der passende Wochentag wird fett markiert. Der Graph wird allerdings nicht für die vordefinierten Städte angezeigt.

Der Graph wird mit Canvas gezeichnet. Da es sich nicht um eine öffentliche Applikation handelt, werden wir trotz der bereits abgelaufenen Probezeit von 30 Tagen weiterhin Canvas verwenden.

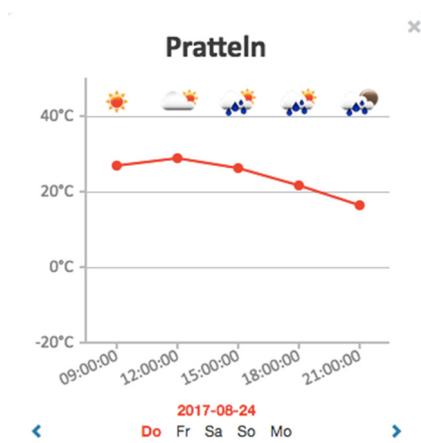


Abbildung 11: Realisierter Wetterprognose-Graph

**Schiebe-  
regler**

Anstatt des beschriebenen Graphen (siehe 4.2.2) für die vordefinierten Städte, wurde hier ein Schieberegler eingesetzt. Dieser befindet sich in der unteren Hälfte des Bildschirms und ist nur im Wetterprognose Modus und auf der Basellandschaftlichen Karte verfügbar. Ein Tooltip, welcher sich oberhalb des Reglers befindet, zeigt das Datum und die Zeit an. Hier kann nun im drei Stunden Takt der Regler verschoben werden und alle Anzeigen der vordefinierten Städte zeigen das Icon und die Temperatur für den jeweiligen Zeitraum an.



Abbildung 12: Realisierter Schieberegler im Wetterprognosemodus

**Wetter-  
Icons**

Dies wurde den Vorgaben (siehe 4.2.3) entsprechend ausgeführt.

## 5.3 Optionsmenü / Wettermodi

**Options-  
Menü**

Das Optionsmenü wurde dem Lösungsvorschlag (siehe 4.3) entsprechend umgesetzt. Allerdings befindet sich der Button in der rechten, oberen Ecke und nicht in der linken Ecke (siehe 4.2) und die Wettermodi werden nicht nebeneinander dargestellt wie es auf dem Foto zu sehen ist, sondern müssen per „Swipe“ gewechselt werden. Die zusätzlichen Informationen für den Modus „aktuelles Wetter“ können per Checkbox aktiviert bzw. deaktiviert werden.



## 5.4 Benutzerinteraktion

Der Benutzer hat alle Möglichkeiten, wie sie im Lösungsvorschlag beschrieben wurde. Allerdings wechselt die Karte automatisch beim Zoomen oder beim Darüberfahren mit der richtigen Zoomstufe.

## 5.5 Zeitplan - Ist

Planung WetterMap		Entwicklung														
		Juli				August				September						
		W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4			
Projektstart	Projekt aufsetzen															
Startseite	Startseite															
Karte	Wetter Overlays															
	Webdienst Abfragen															
	Output (Sprechblasen / Graph)															
Optionsmenü	Forecast (Tag blättern)															
	Modalbox															
	Modiauswahl															
Forecast mit Slider	Weitere Informationen (Checkbox)															
Diverse Korrekturen / Verbesserungen																
Dokumentation	(laufend)															
Fazit																
Präsentation																
Stand: 07.09.2017																

Leitung: ■ Timo  
Ferien: ■

**Probleme** Am Anfang konnte genau nach Zeitplan gearbeitet werden. Allerdings wurde die Dokumentation zu diesem Zeitpunkt vernachlässigt und auf den Schluss verschoben. Der Zeitplan funktionierte unter anderem wegen dem „Forecast mit Slider“ nicht, da dieser gar nicht einberechnet wurde. Ausserdem gab es bei diversen Objekten Korrekturen oder bessere Ideen, welche noch mehr Zeit benötigten als einberechnet war. Dies führte zu der späteren Vollendung des Projektes.

## 5.6 Repository

Der Sourcecode des Projekts ist öffentlich auf GitLab zugänglich: <https://gitlab.com/gf-bl/WetterMap>  
Die Software wird hier publiziert: PFAD auf <http://geoforum.bl.ch/azubi/wettermap/> und <https://www.geo.bl.ch/index.php?id=100>

## 6. Tests

Die fertige Applikation wurde auf verschiedenen Geräten getestet. Die Applikation wurde als Build für die folgenden Plattformen erstellt:

- iOS\*
- Android
- Browser

Getestet wurde die Applikation auf einem iOS-Gerät, einem Android-Smartphone, einem Android-Tablet und als Webseite auf einem Laptop und dem Verwaltungscomputer getestet. In den folgenden Kapiteln werden die Testresultate beschrieben.

\*iOS wurde über die Entwicklungsumgebung XCode getestet. Ein eigentlicher iOS-Build kann zurzeit nicht getestet werden, da kein Developer-Account zur Verfügung steht. Einen solchen braucht es, um den Build mit Apple Hilfsmitteln (Provisioning Profile) zu signieren.

### 6.1 Startseite

Die Startseite funktionierte auf allen Geräten einwandfrei. Das einzige was aufgefallen ist, dass auf dem Tablet das Bild nicht zentriert im „panel-body“ ist (wurde als Issue im GitLab erfasst).

Zusätzlich sind die jeweiligen Texte auf den Mobilgeräten auf Grund des Zeilenumbruchs in die Länge gezogen. Dies macht es eher unschön zum Lesen. Man sollte dafür den Quermodus des Mobilgerätes verwenden.

### 6.2 Wetterkarte

Die Wetterkarte machte am Anfang auf den Mobilgeräten Probleme. Die Kartentiles wurden nicht geladen und die vordefinierten Gemeinden wurden ebenfalls nicht angezeigt. Es stellte sich heraus, dass bei beiden Problemen die Content Security Policy (CSP) Probleme verursachte. Durch die korrekten Einstellungen der CSP konnten diese Probleme behoben werden.

**CSP** Die Content Security Policy ist ein Sicherheitskonzept, welches Webübergreifende Angriffe, sogenanntes Cross-Site-Scripting, verhindern soll. Dabei muss festgelegt werden, von welchen Seiten Daten verwendet werden dürfen.

## 7. Fazit

Das Projekt konnte im Grossen und Ganzen den Wünschen entsprechend umgesetzt werden. Dabei wurden verschiedene Libraries verwendet, welche zum jetzigen Ziel führten.

**Bootstrap** Bootstrap war mir schon von den vorherigen Projekten und aus der Schule bekannt. Es ist eine geeignete Library, um die Struktur einer Webseite „responsive“ darzustellen und ich arbeite sehr gerne damit.

**Leaflet** Leaflet war mir ebenfalls bereits aus dem vorherigen Projekt bekannt und konnte dementsprechend auch schnell integriert werden. Damit kann sehr einfach eine Karte dargestellt werden und auch die zusätzlichen Möglichkeiten, z.B. das Layer-Control-Menü, können mit kurzem Lesen der Dokumentation einfach verwendet werden und der Effekt ist enorm.

**Openweathermap** Das Integrieren der openweathermap-API wurde bereits schon vor dem Projekt ausprobiert und auch im eigentlichen Projekt selbst stellte es keine Herausforderung dar. Die API ist sehr leicht zu integrieren und die Daten können per AJAX-Call einfach implementiert und verwendet werden.

**Daten openweathermap** Die Daten von openweathermap unterscheiden sich aber in gewissen Bereichen stark von Angaben von <http://www.meteoschweiz.admin.ch/home.html> oder ähnlichen Seiten. Deshalb würde ich die Daten als eher ungenau einschätzen und mich nicht voll und ganz darauf verlassen.

**Canvas** Mit Canvas lässt sich jeder nur denkbare Graph zeichnen. Es benötigt ein bisschen Zeit, bis man versteht wie es funktioniert aber danach stellte es keine Probleme mehr dar. Allerdings sollte es nicht als open-Source weiterverwendet werden, möchte man damit eine öffentliche Applikation entwickeln. Eine Alternative zu Canvas wäre die chartjs Library. Damit können ebenfalls schöne Graphen gezeichnet werden.

Das Projekt lief alles in Allem gut. Mithilfe der Planung und den Vorbereitungen konnten Anfangsschwierigkeiten überwunden werden.

Mit den grössten Schwierigkeiten bei den AJAX-Calls, ging das verwenden der einzelnen Libraries ohne grössere Probleme.

Aufgrund der abweichenden Daten würde ich die App eher nicht verwenden. Aber ich finde die Idee mit der Click-Methode sehr nützlich.

Das Projekt hat Spass gemacht, aber ich freue mich nun auf kommende Projekte.