

# Sunaware-Webanwendung



Praktikumsbericht im Fach Datenbanken und verteilte Systeme  
am ISYS des Fachbereichs Informatik  
an der Universität Hamburg im Sommersemester 2017  
eingereicht von Anne Kunstmann (6651481), Marcel Repenning (6681658),  
Tim Pietz (6808046) und Benjamin Schmidtke (6811012)

## Inhaltsverzeichnis

1	Entwurf der Sunaware-Plattform (Entwurfsphase)	1
1.1	Anwendungsidee	1
1.1.1	Lightning Talk	1
1.2	Ideen für Anwendungsfunktionalitäten	2
1.2.1	Angabe von Wetterlagen durch Nutzer	2
1.2.2	Berechnung des UV-Belastungsgrades eines Standortes	2
1.2.3	Empfehlungen für die Dauer eines Aufenthalts im Freien	3
1.2.4	Konsolidierung von Wetterinformationen einer Region	3
1.2.5	Approximation der UV-Belastungsgrade für Regionen/Standorte	3
1.2.6	Ausgabe von Sonnenschutzempfehlungen	3
1.2.7	Ideen für zusätzliche Funktionalitäten	3
1.2.7.1	Tracken des Vitamin D3 Haushaltes eines Nutzers	4
1.2.7.2	Produktempfehlungen	4
1.3	Einzusetzende Technologien	4
1.3.1	Baqend	4
1.3.2	Framework Bulma	4
1.3.3	Framework React	4
1.3.4	Google-Maps	5
1.3.5	Mögliche zusätzliche Technologien	5
1.3.5.1	Private Wetterstationen	5
1.3.5.2	Webservice für UV- und Wetterdaten	5
1.4	Entwurf der ersten Softwarearchitektur	5
1.5	Teilaufgaben und Verantwortlichkeiten innerhalb der Projektgruppe	7
1.5.1	Arbeitspaket: Frontend	8
1.5.2	Arbeitspaket: Backend und Fachlogik	8
1.5.3	Arbeitspaket: Projektmanagement und Projektdokumentation	9
2	Version zur ersten Evaluation im Plenum	10
2.1	Zusammenfassung der erreichten Ziele (Entwicklungsphase 1)	10
2.1.1	Realisierte Funktionalitäten	10
2.1.1.1	Angabe von Wetterdaten durch Nutzer	10
2.1.1.2	Berechnung des UV-Indexes eines Standortes	10
2.1.1.3	Berechnung von Zeitdauern für den Aufenthalt im Freien	11
2.1.1.4	Approximation von Wetterinformationen für eine Region	11
2.2	Vorführung der realisierten Anwendung	11

2.2.1	Interaktionsdesign der ersten Entwicklungsphase .....	11
2.3	Aktualisierung des Entwurfes nach Zwischenpräsentation .....	13
2.3.1	Noch zu realisierende Funktionalitäten und Anpassungen .....	13
3	Die fertige Anwendung (Abschluss) .....	15
3.1	Zusammenfassung der erreichten Ziele (Entwicklungsphase 2) .....	15
3.1.1	Verbesserung der Approximationen.....	15
3.1.2	Anpassung der Konsolidierung von Wetterdaten .....	15
3.1.3	Interaktionsdesign der zweiten Entwicklungsphase .....	16
3.1.4	Frontend und Backend Validation .....	16
3.1.5	Nutzerprofile erweitern .....	16
3.1.6	Optimierung für Mobilgeräte.....	17
3.1.7	Abschließende Softwarearchitektur .....	17
4	Evaluation des Praktikums .....	19
4.1	Tutorial-Phase .....	19
4.2	Entwurfsphase.....	19
4.2.1	Arbeitspaket Frontend.....	19
4.2.2	Arbeitspaket Backend .....	20
4.2.3	Arbeitspaket Projektmanagement und Projektdokumentation.....	20
4.3	Entwicklungsphase 1 .....	20
4.4	Entwicklungsphase 2.....	21
4.5	Das Arbeiten in der Gruppe .....	21
5	Abbildungsverzeichnis .....	22

# **1 Entwurf der Sunaware-Plattform (Entwurfsphase)**

Die Zielsetzung des Praktikums „Datenbanken und verteilte Systeme“ am ISYS des Fachbereichs Informatik stellt die Entwicklung einer datenbankbasierten Webanwendung unter Einbindung der am ISYS entwickelten Caching-as-a-Service Plattform „Baqend“ dar. Von der Entwicklung einer Anwendungs idee bis hin zur Implementierung dieser, beschreibt der vorliegende Praktikumsbericht das Vorgehen und die erreichten Ergebnisse der Gruppe „Sunaware“.

Dieses erste Kapitel beschreibt hierzu die Ergebnisse der ersten Projektwoche. Analog beschreiben die Kapitel 2 und 3 die zweite und dritte Projektwoche. Das abschließende vierte Kapitel dient der Evaluation des Praktikums und beschreibt zudem den Ablauf und Probleme des Projektmanagements.

## **1.1 Anwendungsidee**

Die Sunaware-Plattform ist eine Web-Applikation mit welcher Nutzer die UV-Belastung, der diese in einem gewissen Zeitraum ausgesetzt waren, tracken können. Darüber hinaus können Benutzer der Applikation Wetterinformationen teilen, welche genutzt werden, um die UV-Bestrahlung eines Standortes auf der Grundlage eines Basis UV Indexes zu approximieren. Die Funktionalität der Plattform wird durch die Nutzer getrieben und ermöglicht damit eine dezentralisierte Abbildung von Wetterlagen, welche unabhängig von durch Institutionen fest installierten Wettermessstationen ist. Aus den generierten Wetterinformationen werden durch die Plattform Empfehlungen für Schutzmaßnahmen vor UV-Strahlen ausgesprochen. Im ersten Entwurf dieser Web-Anwendung werden die UV-Belastungsstufen aus Nutzerangaben zu Wetterlagen und Hauttypen berechnet. Im späteren Lebenszyklus der Web-Anwendung ist die Anbindung von privaten Wettermessstationen denkbar, um die Genauigkeit der Approximationen der UV-Belastungsgrade zu verbessern. Hiermit bleibt die Kernidee einer nutzergetriebenen und dezentralisierten Applikation erhalten. Darüber hinaus kann diese Plattform über Empfehlungsmarketingmaßnahmen monetarisiert werden. Denkbar sind an dieser Stelle konkrete Empfehlungen für Produkte, welche zum UV-Schutz für den individuellen Benutzer geeignet sind (Cremes, Brillen, Textilien etc.). Diese Empfehlungen werden auf Grundlage der für jedes Nutzerprofil gespeicherten UV-Belastungsgrade generiert. Des Weiteren werden pro Nutzer Belastungsprofile (Historie des Umfangs der absorbierten Strahlung) erstellt, sodass dieser einsehen kann, welchem Maß an Strahlung dieser in einer gewissen Zeitperiode ausgesetzt worden ist.

### **1.1.1 Lightning Talk**

Im Rahmen der Entwurfsphase ist die zuvor geschilderte Anwendungsidee kurz im Plenum vorzustellen gewesen. Im Anschluss an die kurze Präsentation entfiel eine Feedbackrunde, sodass die Entwicklung der vorgestellten Idee zu diesem Zeitpunkt ohne weitere Anpassungen weiterverfolgt wurde.

## **1.2 Ideen für Anwendungsfunktionalitäten**

Nachfolgend werden die Kernfunktionalitäten in der Reihenfolge ihrer Priorisierung, beginnend mit der wichtigsten und endend mit optionalen Funktionen, beschrieben.

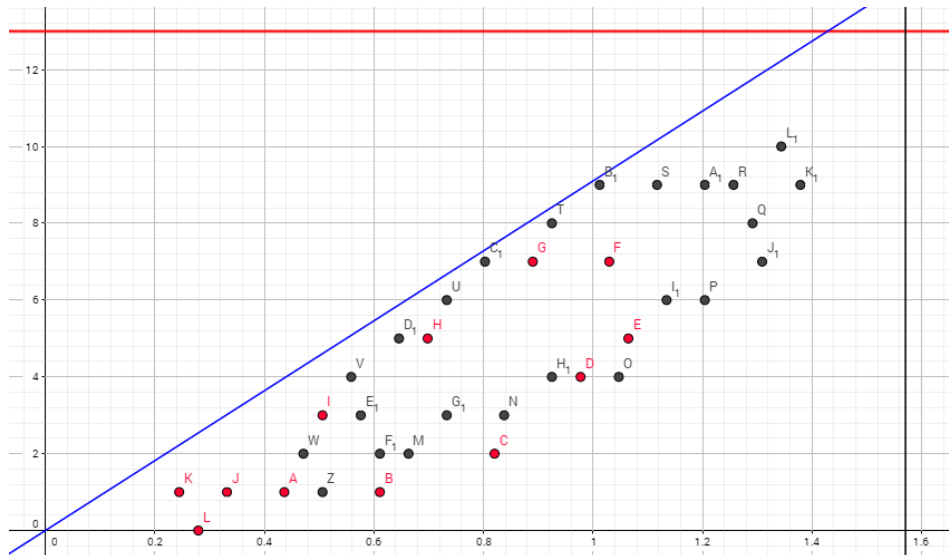
### **1.2.1 Angabe von Wetterlagen durch Nutzer**

Nutzer bekommen eine Funktion bereitgestellt mit der Informationen zu Wetterlagen gespeichert und geteilt werden können. Diese Informationen werden benötigt um die Werte für die UV-Belastung zu approximieren. Konkret ist an dieser Stelle der Bewölkungsgrad für die Funktionalität der Plattform relevant.

### **1.2.2 Berechnung des UV-Belastungsgrades eines Standortes**

Um diese grundlegende Funktionalität gewährleisten zu können, wird zuerst der Sonnenstand (Winkel) aus der Uhrzeit, dem Datum und der Standort des Nutzers berechnet. Auf dieser Grundlage wird dann die Basis-UV-Bestrahlung für eine wolkenlose Wetterlage berechnet. Anschließend wird aus Informationen zur Wetterlage für einen Standort und dem Basis-UV-Bestrahlungswert der approximierte UV-Bestrahlungswert berechnet.

Um eine Funktion für die vereinfachte Berechnung eines Basis-UV-Indexes in Abhängigkeit von Ort und Tageszeit zu erstellen, soll eine externe Node-Bibliothek (SunCalc) verwendet und serverseitig in Baqend eingebunden werden, um den Sonnenstand aus Tageszeit und Ort zu berechnen. Anschließend wird der Basis-UV-Index in einer Funktion berechnet, welche den Sonnenstand als Parameter übergeben bekommt. Bei der einfachen Formel zur Berechnung des Indexes handelt es sich um eine lineare Funktion. In Abhängigkeit von der Höhe des Sonnenstandes steigt der UV-Index linear an. Die Funktion liefert angenäherte Werte für Basis-UV-Indexe. Die Funktion ist näherungsweise entstanden. Hierzu sind Sonnenwinkel und zugehörige UV-Indexe in Relation zueinander gesetzt worden. In einem Koordinatensystem visualisiert, ist die lineare Funktion ermittelt worden. Hierbei sind obere Schranken der UV-Indexe betrachtet worden, um die näherungsweise Berechnung für Werte unter unterschiedlichen Wetterbedingungen nicht zu stark zu verfälschen. Die folgende Abbildung zeigt das Vorgehen zum Finden der beschriebenen Funktion. Die x-Achse zeigt den Sonnenwinkel bis zum Wert  $\pi/2$ . Die y-Achse beschreibt die Größe des UV-Indexes, wobei dieser beim Wert 13 gedeckelt worden ist, da höhere Werte in der Regel nicht auftreten (Abbildung 1: Finden einer Funktion zur Approximation des Basis-UV-Indexes).



### 1.2.3 Empfehlungen für die Dauer eines Aufenthalts im Freien

Dem Plattformnutzer wird eine Funktion bereitgestellt mit welcher dieser Zeitdauern tracken kann in denen dieser UV-Belastungen ausgesetzt gewesen ist. Mit einer Start- und Stopfunktion dieses Timers werden Zeitintervalle erstellt auf deren Grundlage individuelle UV-Belastungswerte berechnet werden und Empfehlungen ausgesprochen werden können. Diese Intervalle können aufaddiert werden, um ein UV-Belastungsprofil über einen gewissen Zeitraum zu erstellen (z.B. ein paar Tage, eine Woche etc.). Der Timer zeigt zudem die restliche empfohlene Aufenthaltszeit im Freien an während dieser läuft.

### 1.2.4 Konsolidierung von Wetterinformationen einer Region

Für diese Funktion werden bereits gespeicherte Wetterinformationen für einen gewissen Umkreis erfasst und auf einen Durchschnitt abgebildet. Hiermit wird die Wetterlage für größere Gebiete aus Wetterinformationen für einzelne konkrete Standorte berechnet.

### 1.2.5 Approximation der UV-Belastungsgrade für Regionen/Standorte

Auf der Grundlage von Basis-UV-Belastungswerten und mehreren Nutzerangaben über Wetterlagen an unterschiedlichen Standorten kann die UV-Belastung für ein größeres Gebiet (z.B. Stadt oder Bundesland) approximiert werden.

### 1.2.6 Ausgabe von Sonnenschutzempfehlungen

Auf der Grundlage von individuellen Belastungsprofilen für Benutzer können Handlungsempfehlungen für Sonnenschutzmaßnahmen durch die Plattform ausgegeben werden.

### 1.2.7 Ideen für zusätzliche Funktionalitäten

Im Folgenden werden mögliche zusätzliche Funktionalitäten präsentiert, welche die Sunaware-Plattform sinnvoll ergänzen können.

#### **1.2.7.1 Tracken des Vitamin D3 Haushaltes eines Nutzers**

Diese Funktion kann den gesundheitsorientierten Nutzen der Plattform steigern, indem dem Nutzer vor Augen geführt wird, in welchem Maße dessen Körper dieses Vitamin im Zuge der Sonnenaufenthalte produziert hat. Der Nutzer kann somit für dieses gesundheitsrelevante Thema, vor dem Hintergrund einer chronischen Unterversorgung in der Bevölkerung, sensibilisiert werden.

#### **1.2.7.2 Produktempfehlungen**

Gekoppelt an die Ausgabe von Sonnenschutzempfehlungen kann eine Produktempfehlungsfunktion genutzt werden, um die Plattform zu monetarisieren (Empfehlungsmarketing).

### **1.3 Einzusetzende Technologien**

Im Rahmen der Entwicklung der Web-Anwendung „Sunaware“ sollen verschiedene Technologien zum Einsatz kommen. Die ausgewählten Technologien werden in diesem Abschnitt vorgestellt und begründet.

#### **1.3.1 Baqend**

Für das Hosting der Anwendungsdaten, der Anwendungslogik sowie der Realisierung des Backend der Sunaware-Webanwendung wird die Caching-as-a-Service Plattform Baqend verwendet. Der Vorteil bei der Verwendung der Baqend Plattform liegt in der Verbesserung von Ladezeiten der Sunaware-Plattform durch die Verwendung global verteilter Caches bei gleichzeitiger Beibehaltung von Konsistenzeigenschaften gecachter Daten. Hierbei werden die Datenbasis und die Anwendungslogik der Plattform automatisch durch Baqend skaliert und gekapselt. Weitere Vorteile in der Verwendung von Baqend liegen in der garantierten Verfügbarkeit des Service sowie der standardmäßig enthaltenen Verschlüsselungstechniken und der Compliance mit deutschen Datenschutzrichtlinien. Hierdurch werden Nutzerdaten optimal vor dritten abgeschirmt.

#### **1.3.2 Framework Bulma**

Zur Unterstützung der Entwicklung des Frontend wird das CSS Framework Bulma verwendet. Dadurch, dass Bulma nur CSS enthält, muss kein JavaScript Code angepasst werden. Diese auf der Website von Bulma aufgeführten Keyfacts wurden zur Entscheidungsfindung herangezogen.

Aufgrund dessen, dass die Entscheidung für ein CSS-Framework wenig Einfluss auf Realisierung der funktionalen Anforderungen der Sunaware-Plattform nimmt, ist die Entscheidung schnell zugunsten von Bulma ausgefallen, um rasch produktiv werden zu können.

#### **1.3.3 Framework React**

Zur Unterstützung der Entwicklung des Frontends wird die JavaScript Bibliothek React verwendet. Zur Auswahl für eine Bibliothek standen Vue.js und React.

Die funktionalen Anforderungen an das Frontend der Sunaware-Plattform werden mit Hilfe des React Frameworks umgesetzt. Aufgrund dessen, dass die funktionalen Anforderungen an das Frontend der Plattform mit beiden Frameworks umsetzbar

sind, ist die Wahl auf React gefallen, da ein Teil der Projektgruppe dieses Framework gerne kennenlernen wollte. Zudem unterstützt die JavaScript API von Baqend die Verwendung dieser Bibliothek zur Entwicklung des Frontend.

### **1.3.4 Google-Maps**

Der Kartenservice von Google ist in zweifacher Hinsicht geeignet, um in der Sunaware-Webanwendung genutzt zu werden.

Dieser Kartenservice dient dazu die Userpositionierung zu visualisieren, um dem Benutzer der Plattform rasche eine Orientierung auf der Karte zu ermöglichen. Des Weiteren wird die Kartenansicht genutzt, um die relevanten Wetterreports einer Region der Nutzer an der Benutzerschnittstelle zu visualisieren. Somit wird dem Benutzer ein Teil des Funktionsumfangs der Sunaware-Plattform auf der Karte zur Verfügung gestellt.

### **1.3.5 Mögliche zusätzliche Technologien**

Nachfolgend werden zwei weitere Technologien beschrieben, welche die Anwendung sinnvoll unterstützen können. Jedoch wird nur eine Technologie hiervon ernsthaft vom Projektteam für den Einsatz in einem potenziellen späteren Lebenszyklus der Plattform in Betracht gezogen.

#### **1.3.5.1 Private Wetterstationen**

Private Wetterstationen, welche durch Nutzer installiert werden, könnten an die Sunaware-Plattform angebunden werden. Hiermit könnte das Wettergeschehen exakt für einen Standort abgebildet werden. Aus den exakten Daten für mehrere Standorte kann das Wettergeschehen in einer Region besser näherungsweise abgebildet werden. Die Nutzung privater Wetterstationen fördert den, bereits zuvor skizzierten, Grundgedanken einer dezentralisierten und nutzergetriebenen Web-Applikation. Hierdurch wäre auch eine genauere Approximation von UV-Belastungsgraden möglich.

#### **1.3.5.2 Webservice für UV- und Wetterdaten**

Ein Web-Service für UV- und Wetterdaten kann verwendet werden, um mit einer geeigneten Funktion die Approximationen für die UV-Belastungswerte abzugleichen und zu validieren. Als hauptsächliche Datenquelle kommt die Abfrage von Webservices im Rahmen des Projektes allerdings nicht in Frage, da dieser dem Grundgedanken einer dezentralisierten und nutzergetriebenen Web-Applikation widerspricht.

## **1.4 Entwurf der ersten Softwarearchitektur**

Im Rahmen der Entwurfsphase der ersten Projektwoche ist die Software-Architektur der geplanten Anwendung entworfen worden. Hierbei ist ein UML-Klassendiagramm der fachlichen Klassen erarbeitet worden. Des Weiteren sind Funktionen identifiziert worden, welche objektübergreifend die Funktionalität der Sunaware-Plattform gewährleisten sollen. Im Folgenden wird diese erste grundlegende Architektur vorgestellt.



Die Objekte der Klasse „User“ enthalten Felder für alle relevanten Informationen, welche benötigt werden, um dem Nutzer die beschriebenen Funktionalitäten zur Verfügung stellen zu können.

Die Klasse „WeatherInfo“ enthält Felder zur Abbildung von Wetterlagen. Objekte dieser Klasse werden für die Berechnung von UV-Werten sowie dem Mitteilen von neuen Wetterinformationen benötigt. Des Weiteren werden Objekte dieser Klasse dazu verwendet, um Wetterlagen für Regionen zu approximieren.

Die Klasse „UserReport“ wird benötigt, um Reports zu Nutzern zu generieren. Hierbei wird einem Report ein Objekt vom Typ WeatherInfo zugeordnet.

Die Klasse „Timer“ wird benötigt, um dem Nutzer Zeitintervalle zuordnen zu können, in denen dieser UV-Strahlung ausgesetzt gewesen ist.

Die Klasse „WeatherService“ dient, in Anlehnung an den WAM-Ansatz, als ein Service, welcher mehrere Objekte vom Typ WeatherInfo bearbeitet und ein Objekt vom Typ WeatherInfo berechnet, welches die konsolidierten Wetterinformationen für einen bestimmten Umkreis um eine bestimmte Position enthält.

Die Klasse „UVService“ dient ebenfalls als ein Service. Dieser Service bietet eine Operation an, welche aus Angaben zur geografischen Lage, der Tageszeit und Informationen zur Wetterlage den UV-Index approximiert. Des Weiteren enthält der Service private Hilfsmethoden zu diesem Zweck.

Die Klasse „UserService“ ist der umfangreichste Service der geplanten Architektur und bietet Operationen zur Berechnung der verbleibenden Restzeit eines Nutzers, in welcher dieser gefahrlos UV-Strahlung absorbieren kann. Hierzu werden auch Operationen angeboten, welche das Maß an absorbierter Strahlung für gewisse Zeitspannen berechnen.

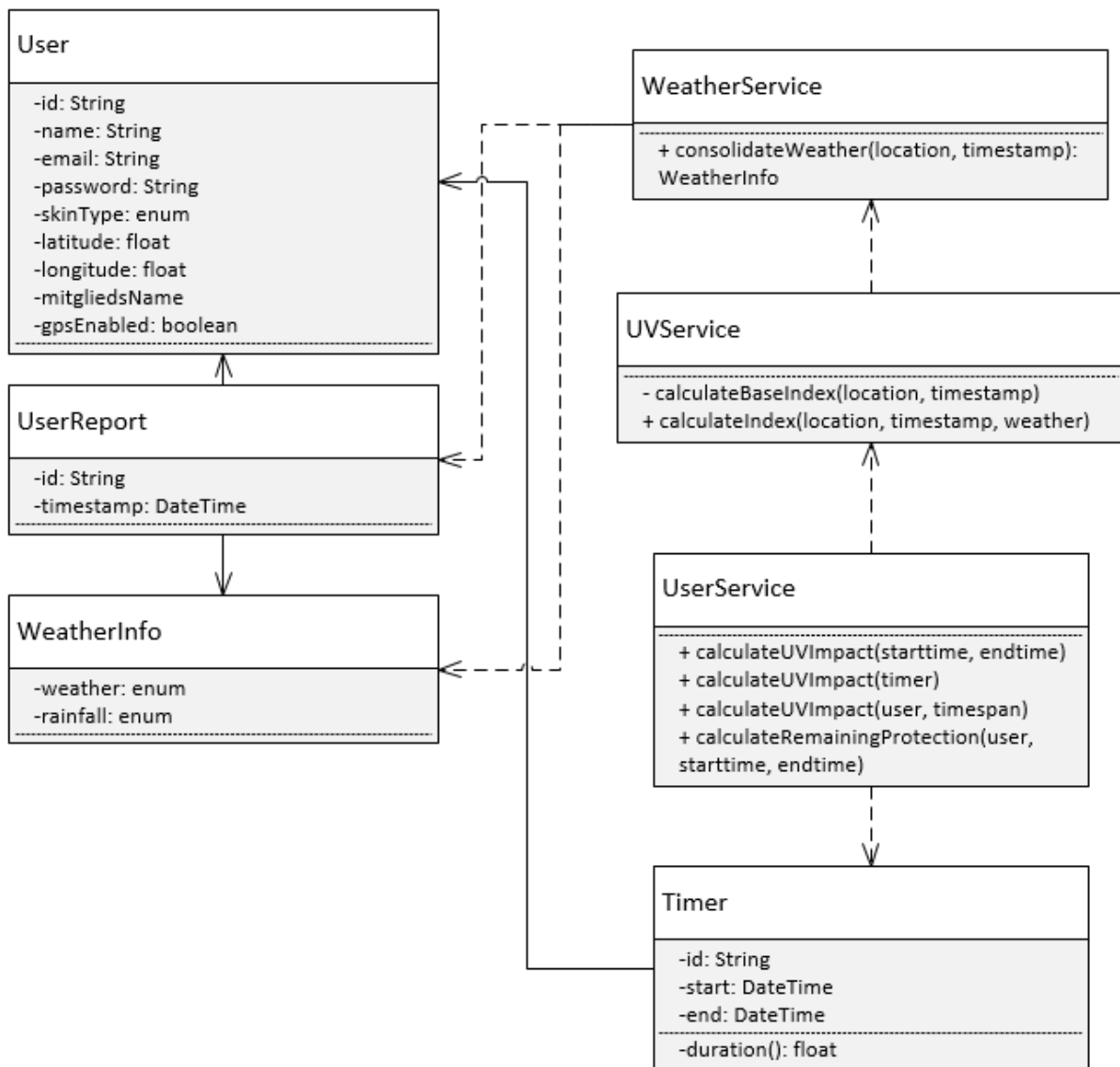


Abbildung 2: UML Diagramm der Entwurfsphase

## 1.5 Teilaufgaben und Verantwortlichkeiten innerhalb der Projektgruppe

Zur Umsetzung der Anwendungs-idee der Sunaware-Webapplikation werden drei Arbeitspakete definiert, welche innerhalb der drei Projektwochen abzarbeiten sind. Hierzu wird das Projekt in zwei Phasen (Sprints) mit fest definierten Zielen unterteilt.

Die erste Projektphase umfasst die ersten zwei Projektwochen (07.08.2017 - 20.08.2017) und hat zum Ziel eine erste Version der Sunaware-Plattform zu implementieren.

Die zweite Projektphase umfasst die letzte Projektwoche (21.08.2017 – 24.08.2017). Die Zielsetzung dieser zweiten Phase zur finalen Ausarbeitung der Anwendung stellt die Umsetzung von Verbesserungsvorschlägen und die Umsetzung potenzieller Erweiterungen der Plattform dar.

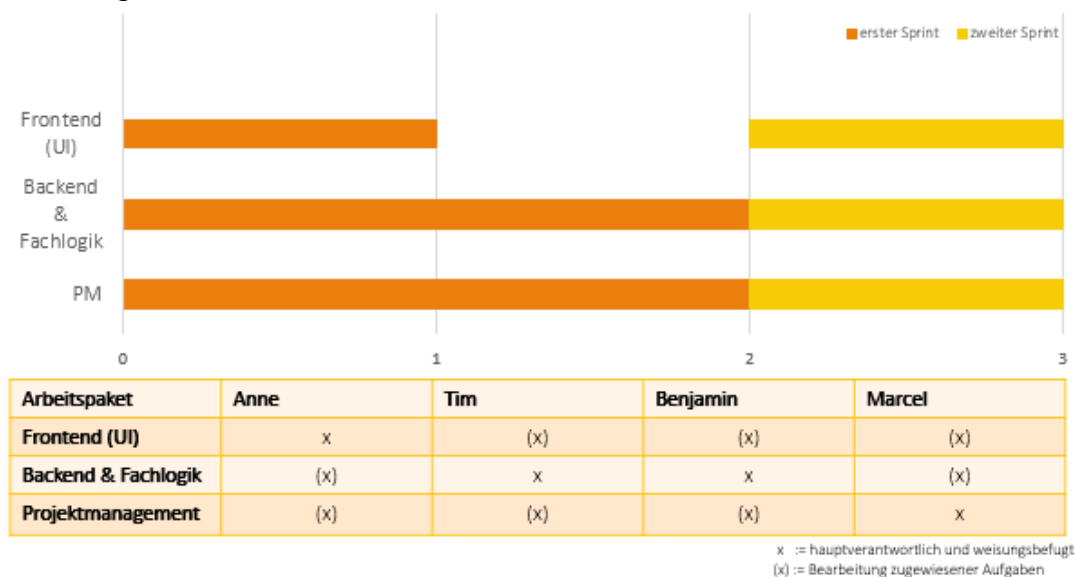


Abbildung 3: Projektstruktur

### 1.5.1 Arbeitspaket: Frontend

Die Zielsetzung dieses Arbeitspaketes stellt die Entwicklung eines Interaktionsdesigns für die Anwendung sowie die Umsetzung dessen in eine Benutzerschnittstelle (User Interface) dar.

Im Zuge dieses Arbeitspaketes sind in der ersten Projektwoche der ersten Projektphase alle nötigen Seiten der Anwendung erstellt worden.

Frau Anne Kunstmann hat dieses Arbeitspaket hauptverantwortlich betreut und Aufgaben an das restliche Projektteam delegiert.

Im Anschluss an die Entwicklung und Evaluierung eines Konzeptes zur Benutzerinteraktion ist dieses prototypisch bis zum Ende der ersten Projektwoche umgesetzt worden.

Im Zuge der zweiten Projektphase soll das Interaktionsdesign zu finalisiert werden.

### 1.5.2 Arbeitspaket: Backend und Fachlogik

Die Zielsetzung dieses Arbeitspaketes stellt die Implementierung des Backends sowie der Anwendungslogik unter der Zuhilfenahme der bereits vorgestellten Technologien dar. Im Kontext dieses Arbeitspaketes sollen innerhalb der ersten Projektphase die definierten Funktionalitäten der Anwendung zu implementiert werden.

Herr Tim Pietz und Herr Benjamin Schmidtke haben dieses Arbeitspaket hauptverantwortlich als technische Leiter betreut und Entwicklungsaufgaben an das übrige Projektteam delegiert.

Im Rahmen der zweiten Projektphase sollen die bereits implementierten Funktionalitäten finalisiert und gegebenenfalls um weitere Funktionalitäten ergänzt werden.

### **1.5.3 Arbeitspaket: Projektmanagement und Projektdokumentation**

Die Zielsetzung des Arbeitspakets „Projektmanagement und Projektdokumentation“ stellt die projektbegleitende Erstellung des Praktikumsberichtes, die Zeitplanung, die Vorbereitung von Präsentationen sowie die Ergebnissicherung in allen Projektphasen dar.

Herr Marcel Repenning hat dieses Arbeitspaket hauptverantwortlich geführt und Aufgaben bezüglich der Projektdokumentation an das Projektteam delegiert.

## **2 Version zur ersten Evaluation im Plenum**

Im Zuge des Abschlusses der ersten Entwicklungsphase ist die Anwendung in ihrer ersten Version am 21.08.2017 im Plenum zu präsentieren gewesen. An dieser Stelle galt es zudem die erreichten Ziele zu reflektieren und noch zu erreichende Ziele zu identifizieren.

### **2.1 Zusammenfassung der erreichten Ziele (Entwicklungsphase 1)**

Die Ergebnisse der ersten Entwicklungsphase stellen eine schlanke und übersichtliche Benutzerschnittstelle, ein funktionierendes Frontend und ein funktionstüchtiges Backend der Sunaware-Plattform dar, sodass das Ziel der Präsentation einer funktionierenden Applikation am Ende der ersten Entwicklungsphase erreicht worden ist.

#### **2.1.1 Realisierte Funktionalitäten**

Nachfolgend werden die realisierten Funktionalitäten der ersten Entwicklungsphase vorgestellt und näher beschrieben. Alle zuvor erwähnten Anwendungsideen sind jedoch nicht umgesetzt worden. Beispielsweise ist anstatt einer Funktion, welche den UV-Index für eine ganze Region approximiert, eine Funktion entwickelt worden, welche das Wettergeschehen in einem gewissen Umkreis näherungsweise bestimmt. Objektübergreifende Funktionen sind hierbei nach dem Service-Konzept des WAM-Ansatzes implementiert worden. Die „baqendseitige“ Implementation der fachlichen Logik schirmt diese vor dritten ab und sorgt im Sinne der Kapselung für die Sicherheit der entwickelten Funktionen, da diese clientseitig nur über eine Schnittstelle angesprochen werden.

Um regelmäßige Updates für Wetterdaten zu erhalten, ist ein zudem ein Bot geschrieben worden, welcher in einem gewissen Zeitabstand neue Wetterreports aus den Daten eines Webservices für Wetterdaten generiert und somit eine Menge an Nutzern simuliert.

##### **2.1.1.1 Angabe von Wetterdaten durch Nutzer**

Das Mitteilen von Wetterinformationen durch Benutzer steht im Zentrum der Anwendung. Es werden Informationen zur Bewölkungslage, dem Ausmaß von Niederschlag sowie die Temperatur erfasst. Die Daten zur Bewölkung sind neben der Position des Nutzers essentiell für die Berechnung von UV-Indizes. Unter der Verwendung von Baqend-Realtime werden Wetterreports von anderen Nutzern der Plattform dem eingeloggten User angezeigt, sodass dieser das Wettergeschehen verfolgen kann.

##### **2.1.1.2 Berechnung des UV-Indexes eines Standortes**

Die Bestimmung der UV-Indizes wird im UVService der Anwendung serverseitig in mehreren Schritten durchgeführt. Zuerst wird der Sonnenwinkel mit Hilfe der externen Node-Bibliothek „SunCalc“ berechnet. Hierzu werden Angaben zur Position des Nutzers in Längen- und Breitengrad sowie die Tageszeit benötigt. Dieser Winkel wird als Parameter einer selbst erstellten Funktion zur Annäherung des UV-Indexes übergeben, um einen Basiswert (Wetterlage ohne Bewölkung) zu bestimmen. Aus

diesem Basiswert wird der individuelle UV-Index unter Berücksichtigung von Wetterlagen mit einer weiteren selbst erstellten Annäherungsfunktion berechnet.

### 2.1.1.3 Berechnung von Zeitdauern für den Aufenthalt im Freien

Beim Einrichten eines Profils wird der Nutzer nach dessen Hauttyp gefragt. Diese Information wird für die Berechnung der individuellen Sonnenzeit benötigt. Der Berechnung von individuellen Sonnenzeiten (ohne Sonnenschutz) liegt der MED-Wert zu Grunde, welcher die Widerstandsfähigkeit von Hauttypen gegenüber UV-Strahlung beschreibt. Jedem Hauttyp der Anwendung ist ein MED-Wert zugeordnet. Die Sonnenzeit in Minuten ergibt sich hierbei aus der Division des MED mit dem eineinhalbfachen Wert des UV-Indexes. Diese Funktionalität wird serverseitig im UserService zur Verfügung gestellt. Das Ergebnis wird dem Nutzer im bereits skizzierten Timer angezeigt.

### 2.1.1.4 Approximation von Wetterinformationen für eine Region

Zur Annäherung von Wetterlagen werden mehrere WeatherReport-Objekte aus der Datenbank angefragt und zu einem WeatherReport-Objekt zusammengefasst. Hierbei werden alle Wetterinformationen, welche als Number erfasst werden, aufaddiert und durch ihre Anzahl geteilt. Somit wird das Wettergeschehen für einen Umkreis angenähert.

## 2.2 Vorführung der realisierten Anwendung

Im Sinne des Abschlusses der ersten Entwicklungsphase ist die realisierte Anwendung im Plenum vorgestellt worden. Hierbei ist die Funktionalität der Anwendung präsentiert worden. Anschließend sind die noch zu erreichenden Ziele kurz präsentiert und begründet worden. Im Rahmen des Praktikumsberichtes werden an dieser Stelle die Ergebnisse des Interaktionsdesigns der ersten Projektphase präsentiert.

### 2.2.1 Interaktionsdesign der ersten Entwicklungsphase

Sobald die Anwendung aufgerufen wird, erhält der Nutzer die Optionen sich mit einem bestehenden Account einzuloggen (Abbildung 3: Login) oder sich einen neuen Account anzulegen (Abbildung 4: Sign Up).

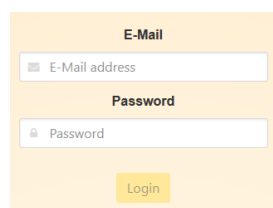
The login form is a vertical rectangle with a light orange background. It contains two input fields: the top one is labeled 'E-Mail' with a placeholder 'E-Mail address' and an envelope icon; the bottom one is labeled 'Password' with a placeholder 'Password' and a lock icon. Below these fields is a yellow button labeled 'Login'.

Abbildung 4: Login

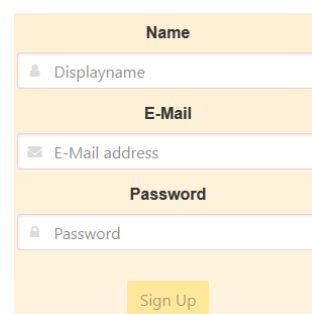
The sign up form is a vertical rectangle with a light orange background. It contains three input fields: the top one is labeled 'Name' with a placeholder 'Displayname' and a person icon; the middle one is labeled 'E-Mail' with a placeholder 'E-Mail address' and an envelope icon; the bottom one is labeled 'Password' with a placeholder 'Password' and a lock icon. Below these fields is a yellow button labeled 'Sign Up'.

Abbildung 5: Sign Up

In den User-Settings (Abbildung 5: User-Settings) werden Informationen zum Hauttyp und der geografischen Position des Nutzers erfasst. Diese Informationen sind grundlegend, um dem User die bereits beschriebenen Funktionalitäten anbieten zu können.

Abbildung 6: User-Settings

Auf einer Karte werden dem User Wetterinformationen für dessen Region angezeigt (Abbildung 6: Karte und Wetterinformationen). Die Informationen werden als „Sonnen“ und „Wolken“ aufbereitet und an der Oberfläche präsentiert. Am rechten Rand werden von anderen Nutzern geteilten Wetterinformationen in einem Feed angezeigt und in Echtzeit aktualisiert.

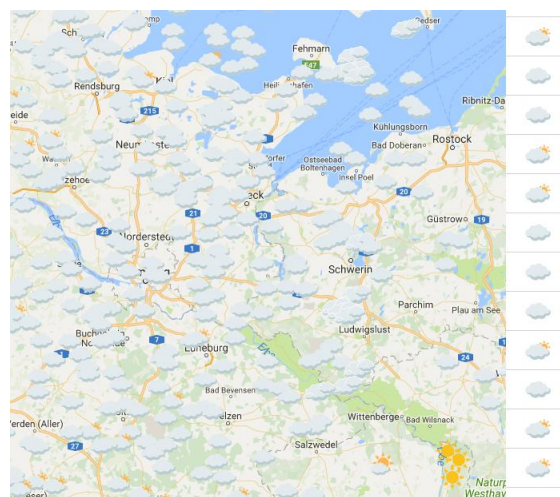


Abbildung 7: Karte und Wetterinformationen

Jeder Nutzer hat die Möglichkeit das Wetter an dessen Standort regelmäßig zu teilen. Hierzu gibt der Nutzer Informationen zum Regen und zur Bewölkung an dessen Standort an. Abschließend kann der Nutzer noch die Temperatur in Grad-Celsius angeben (Abbildung 8: Wetterinformationen teilen). Mit Baqend Realtime ist

hierbei realisiert worden, dass sich die von Nutzern geteilten Wetterberichte in Echtzeit auf der Karte und im Feed aktualisieren.

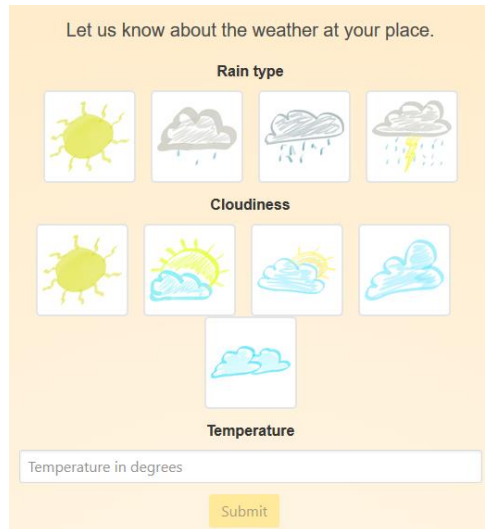


Abbildung 8: Wetterinformationen Teilen

Der oberhalb der Karte, rückwärts zählende, eingblendete Timer gibt an wie viel Zeit der Nutzer bis zur ersten Hautrötung im Freien verbringen kann. Unterhalb wird die bereits verbrauchte Sonnenzeit angezeigt. Der Timer verwendet Baqend Realtime um zu gewährleisten, dass der Timer weiterläuft wenn sich der Nutzer ausloggt oder das Browserfenster schließt. Somit wird gewährleistet, dass dem Nutzer auch nach abermaligem starten der Anwendung, z.B. nach einem Aufenthalt im Freien, die aktuelle Sonnenzeit angezeigt wird.

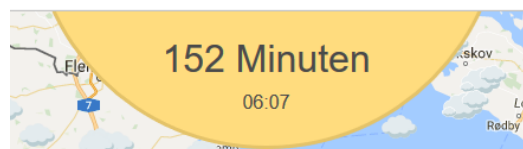


Abbildung 9: Timer

Die vorgestellte Nutzeroberfläche ist unter der Nutzung des JavaScript Frameworks React und des CSS-Frameworks Bulma implementiert worden. Vorteile zur Verwendung von React stellen die „baqendseitige“ Unterstützung dieses Frameworks sowie dessen Performancevorteile gegenüber anderen Frameworks zur Implementation von Webanwendungen dar.

## 2.3 Aktualisierung des Entwurfes nach Zwischenpräsentation

Im Kontext des Abschlusses der ersten Entwicklungsphase sind vorzunehmende Anpassungen und gewünschte Erweiterungen des entstandenen Artefakts zu identifizieren gewesen. Nachfolgend sollen diese genannt werden.

### 2.3.1 Noch zu realisierende Funktionalitäten und Anpassungen

Im Rahmen der zweiten Entwicklungsphase wird der Fokus von der Implementation der Fachlogik hin zur Verbesserung der Gebrauchstauglichkeit, der technischen Eigenschaften sowie der Behebung von Fehlern der Plattform gelegt. Hierzu sind die im Folgenden genannten (Teil-) Arbeitspakete identifiziert und priorisiert worden.



1. Frontend & Backend Validation
2. Spezialfälle behandeln (z.B. keine vorliegenden Wetterdaten, Tagesgrenzen, Approximationsfunktionen überarbeiten)
3. Positionierung/ Positionsangabe auf GPS umstellen
4. Nutzerprofile ergänzen
5. Wetterberichte anzeigen
6. ACL (Zugriffskontrolle)
7. Optimierung für Mobilgeräte
8. Kartenstil anpassen

Diese identifizierten Arbeitspakete werden im Verlauf der dritten Projektwoche im Projektteam aufgeteilt und ihrer zugewiesenen Priorität gemäß bearbeitet.

### 3 Die fertige Anwendung (Abschluss)

Im Zuge des Abschlusses der zweiten Entwicklungsphase ist die Anwendung in ihrer zweiten Version am 24.08.2017 im Plenum zu präsentieren gewesen. An dieser Stelle galt es zudem die erreichten Ziele zu reflektieren.

#### 3.1 Zusammenfassung der erreichten Ziele (Entwicklungsphase 2)

Das Ergebnis der zweiten Entwicklungsphase stellt eine überarbeitete Version der Sunaware-Plattform dar. Hierbei sind unter anderem Anpassungen an der Nutzerschnittstelle sowie der Anwendungslogik vorgenommen worden.

##### 3.1.1 Verbesserung der Approximationen

Zur Berechnung genaueren Annäherungen an Werte für die UV-Indizes ist eine neue Funktion entwickelt worden, welche den UV-Index in Abhängigkeit von der Bewölkungslage ohne die vorher aufgetretenen abrupten Sprünge annähert. Hierzu sind UV-Indizes und Bewölkungslagen aus Beispieldaten entnommen und in Relation zueinander gesetzt worden. Hierbei ist der maximal erreichbare Index auf 12 festgelegt worden, da höhere Werte nur selten auftreten. Als niedrigsten UV-Index ist der Wert 1 bestimmt worden, da zur Berechnung der Sonnenzeit der bereits erwähnte MED-Wert durch den eineinhalbfachen Wert des UV-Indexes geteilt wird. Diese Festlegung auf den Wert 1 ist also notwendig, um zu vermeiden, dass der Nenner dieses Bruchs den Wert 0 erhält. Die Folgende Abbildung beschreibt den Verlauf der Funktion. Die x-Achse stellt hierbei die Bewölkungsgrade und die y-Achse die Werte für den UV-Index dar (Abbildung 10: Verbesserung der UV-Index Approximation).

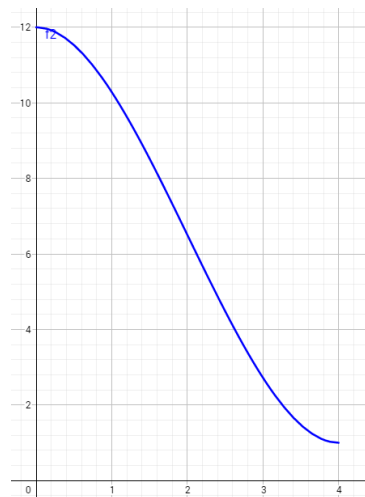


Abbildung 10: Verbesserung der UV-Index Approximation

##### 3.1.2 Anpassung der Konsolidierung von Wetterdaten

Die Konsolidierungsfunktion des WeatherService ist dahingehend überarbeitet worden, als dass für die Konsolidierung der Wetterdaten keine Umkreissuche mit fest definiertem Radius mehr genutzt wird, sondern eine Suche, welche die 31 nächst gelegenen Wetterreports aus der Datenbank abfragt. Hiermit wird eine aufgetretene Ausnahme behandelt. Falls keine Wetterreports im Umkreis vorhanden gewesen sind, ist die Konsolidierungsfunktion nicht nutzbar gewesen. Durch die Nutzung der

31 nächstgelegenen Wetterreports sind auf diese Weise immer Wetterdaten für die Konsolidierungsfunktion verfügbar.

### 3.1.3 Interaktionsdesign der zweiten Entwicklungsphase

Während der zweiten Entwicklungsphase ist die Benutzeroberfläche überarbeitet und um weitere Elemente ergänzt worden. Somit ist es nun möglich die geteilten Wetterberichte durch ein Klicken auf ein Wolkensymbol auf der Karte oder ein Wettersymbol des Feeds anzuzeigen zu lassen. Zudem wird dem Nutzer nun dessen Position auf der Karte angezeigt. Dies kann die Position des Browsers oder eine beliebig gewählte Position auf der Welt sein, welche durch eine einfache Ortsangabe über Google Maps angegeben wird. Damit entfällt die Angabe von Längen- und Breitengraden der ersten Version. In der zweiten Version der Anwendung entfällt zudem der Button unterhalb der Karte zum Teilen von Wetterinformationen und wird durch ein gelbes Pluszeichen oberhalb des Wetterfeeds ersetzt (Abbildung 11: Positionierung, Berichte, Kartenstil, Teilen).

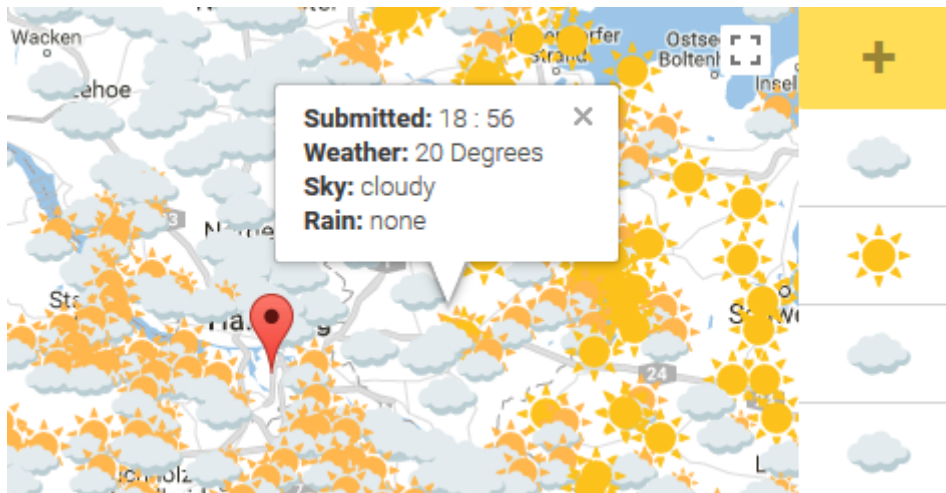


Abbildung 11: Positionierung, Berichte, Kartenstil, Teilen

### 3.1.4 Frontend und Backend Validation

Auf der Seite des Frontends ist die Validierung bei allen Submit-Buttons realisiert worden. Zudem wird beim Eintragen von Wetterdaten und beim Signup geprüft, ob alle Felder gültige Werte haben. Dies wird live bei jedem Input über den State geprüft. Zur Prüfung wird u.a. Regex verwendet.

Im Backend Code wird beim Anlegen eines Nutzers geprüft, ob der Username verfügbar ist und ob der Hauttyp angegeben wurde. Zudem darf der Hauttyp nicht verändert werden beim Profilupdate. Beim Anlegen eines Wetterreports wird geprüft, ob allen Feldern gültige Werte übergeben werden. Bei fehlerhaften Angaben werden entsprechende Nachrichten ausgegeben.

### 3.1.5 Nutzerprofile erweitern

Im Zuge der zweiten Entwicklungsphase sollten die Nutzerprofile, um Statistiken ergänzt werden. Diese, in der Entwurfsphase definierte, Funktionalität der Plattform

sollte im Anschluss an die Abschlusspräsentation entwickelt werden. Dem Nutzer sollen für gewisse Zeiträume Informationen bereitgestellt werden, die beschreiben in welchem Maße dieser sich UV-Bestrahlung ausgesetzt hat. Die Statistiken sollen mit einem Klick auf den Reiter „Statistics“ angezeigt werden können.

### **3.1.6 Optimierung für Mobilgeräte**

Die Optimierung für Mobilgeräte wird durch den mobile first Ansatz des gewählten CSS-Framework Bulma standardmäßig unterstützt. An dieser Stelle galt es somit noch Anpassungen der Funktionalitäten und des UI vorzunehmen.

### **3.1.7 Abschließende Softwarearchitektur**

Den Abschluss der Präsentation des Erreichten bildet die Softwarearchitektur in ihrem Ist-Zustand am Ende der zweiten Entwicklungsphase (Tag der Präsentation). Diese Architektur der fachlichen Klassen unterscheidet sich von der initialen Softwarearchitektur insoweit, als dass im Verlauf des Projektes Felder angepasst und ergänzt wurden und Operationen ergänzt, umbenannt oder gestrichen wurden. Die grundsätzliche Struktur der Klassen ist jedoch nicht verändert worden, sodass die zuvor beschriebenen hauptsächlichen Verantwortlichkeiten dieser für das System unverändert geblieben sind. Die Services sind weiterhin als Baqend-Module beibehalten und überarbeitet worden. Die Services bieten asynchrone Funktionalitäten. Operationen, welche in diesem Sinne arbeiten sind beispielsweise mit dem Ergebnistyp „Promise<number>“ gekennzeichnet.

Die Klassen „User“, „UserReport“, „WeatherInfo“ und „Timer“ bilden die in der Baqend-Datenbank enthaltenen Objekte mit ihren Feldern und werden durch Services verarbeitet.

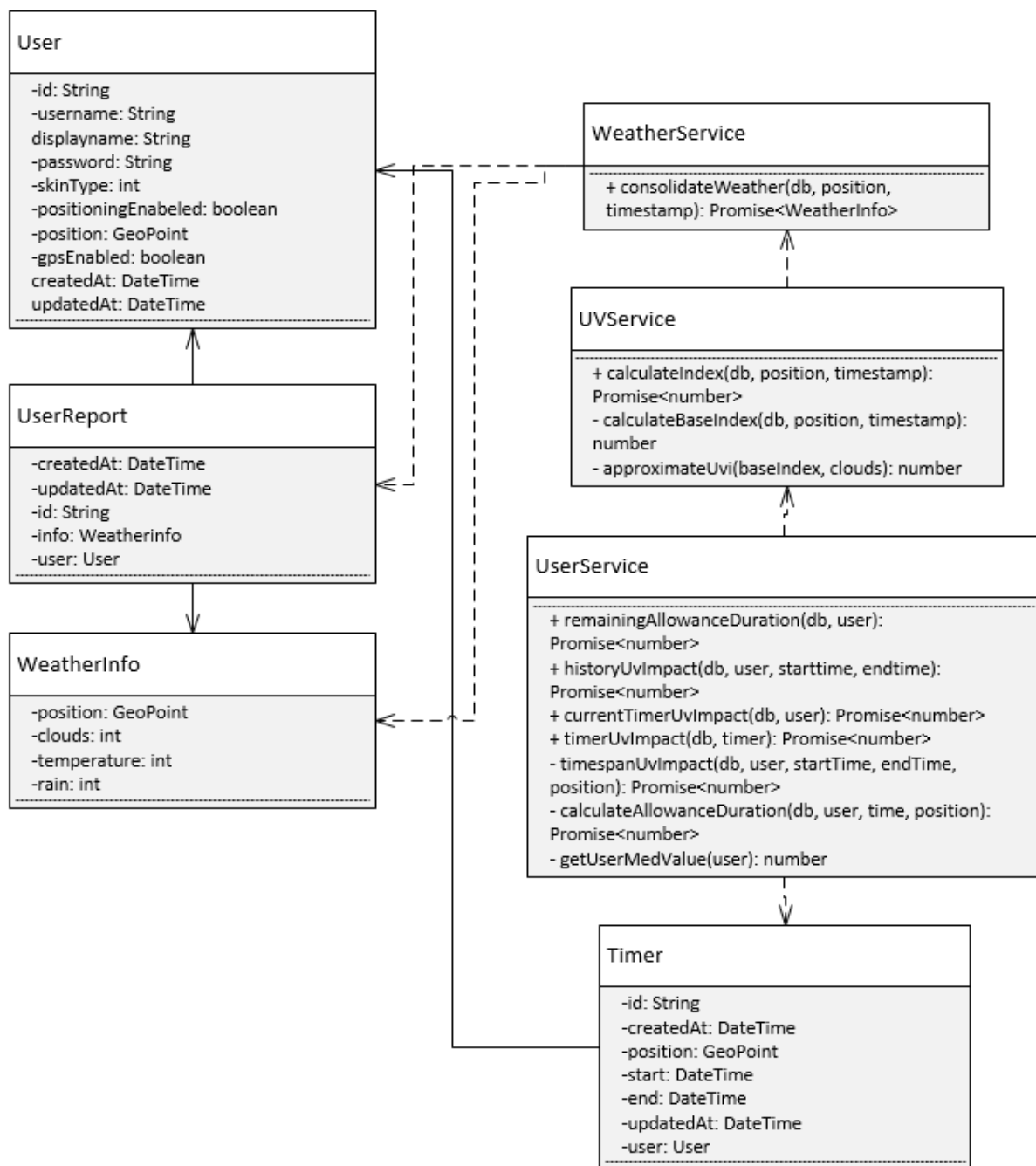


Abbildung 12: UML Klassendiagramm der zweiten Entwicklungsphase

## **4 Evaluation des Praktikums**

Den Abschluss des Praktikumsberichtes bildet die Evaluation der einzelnen Projektphasen. Hierzu wird der Ablauf dieser in den folgenden Unterkapiteln beschrieben. Hierbei wird auch auf Probleme und Herausforderungen eingegangen, die im Projektmanagement beachtet werden mussten.

### **4.1 Tutorial-Phase**

Die Zielsetzung der Tutorial-Phase stellte die Einarbeitung in „Baqend“ dar, um das Backend-as-a-Service-Konzept zu erlernen. Hierzu ist in der Gruppe der ausgehändigte Leitfaden bis auf den Punkt der optionalen Vertiefung durchgearbeitet worden.

Darüber hinaus sind die ersten drei Projektstage genutzt worden, um sich in die neuen Technologien einzuarbeiten und ein grundlegendes Verständnis dieser zu erlangen. Parallel hierzu hat das Projektteam sich um erste Ansätze für den Softwareentwurf gekümmert, um ein gemeinsames Verständnis für die Zielsetzung zu erlangen.

### **4.2 Entwurfsphase**

Die Zielsetzung der Entwurfsphase stellte die Entwicklung der Anwendungs idee sowie die Vorbereitung eines kurzen Lightning Talks zur Präsentation dieser am vierten Projekttag dar. Weiterhin ist die Anwendungs idee in Entwürfen konkretisiert worden und in einer Präsentation am Montag den 14.08.2017 vorgestellt worden.

Die entstandenen Artefakte dieser Phase sind die Beschreibung der Anwendungs idee, die Beschreibung der gewünschten Funktionalitäten, der Entwurf einer grundlegenden Softwarearchitektur, Entwürfe für das gewünschte User-Interface und die begründete Auswahl zu nutzender Technologien.

Des Weiteren sind Arbeitspakete definiert und Verantwortlichkeiten auf die Gruppenmitglieder aufgeteilt worden, um die anschließende Entwicklungsphase zu gliedern.

#### **4.2.1 Arbeitspaket Frontend**

Die Zielsetzung dieses Arbeitspaketes stellt die Entwicklung der Nutzerschnittstellen, welche im Webbrowser aufgerufen werden, dar.

Hierbei sind im ersten Schritt Wireframes auf Papier erstellt worden, um ein gemeinsames Verständnis für das Interaktionsdesign der Plattform zu schaffen. Auf der Grundlage dieser Ergebnisse sind weitere Ideen für fachliche Funktionen der Anwendung identifiziert worden.

Im nächsten Schritt ist die angestrebte grafische Benutzeroberfläche unter der Verwendung von React, HTML und des CSS-Frameworks Bulma implementiert worden.

Bis zum Ende der ersten Projektwoche ist die erste geplante Iteration dieses Arbeitspaketes erfolgreich abgeschlossen worden, wobei sich weitere Anforderungen ergeben haben, sodass sich im Zuge der zweiten Projektwoche ebenfalls ein Bedarf zur Bearbeitung dieses Arbeitspaketes ergab.

Die geplante zweite Iteration und die Finalisierung der Nutzeroberfläche ist für die dritte Projektwoche im Anschluss an die Zwischenpräsentation vom 21.08.2017 eingeplant gewesen.

#### **4.2.2 Arbeitspaket Backend**

Die Zielsetzung dieses Arbeitspaketes stellt die Entwicklung der Anwendungslogik sowie die Realisierung der Datenhaltung dar.

Hierzu ist im ersten Schritt ein UML-Klassendiagramm zur Beschreibung der Fachlichen Komponenten der Plattform erstellt worden. Hierbei sind alle anwendungsfachlichen Funktionen in ihren Signaturen festgelegt worden. Zusätzlich sind in Baqend die benötigten Schemata der Datenhaltung angelegt worden.

Im zweiten Schritt ist die Anwendungslogik in auf Baqend-Seite in JavaScript implementiert worden.

Dieses Arbeitspaket ist ebenfalls in zwei Phasen aufgeteilt worden. Die ersten beiden Projektwochen (Phase 1) hatten zum Ziel die Anwendung zu Demonstrationszwecken fertig zu implementieren. Die letzte Projektwoche sollte verwendet werden, um Verbesserungsvorschläge, welche sich aus der Zwischenpräsentation am 21.08.2017 ergaben, für die Anwendung zu beurteilen und gegebenen Falls zu implementieren. Eine weitere Zielsetzung der zweiten Phase stellte die Implementation von zusätzlichen Funktionalitäten dar.

#### **4.2.3 Arbeitspaket Projektmanagement und Projektdokumentation**

Die Zielsetzung dieses projektbegleitenden Arbeitspaketes stellt das Projektmanagement, die Projektdokumentation, Präsentationsvorbereitung sowie die Ergebnissicherung dar. Dieses Arbeitspaket wird entwicklungsbegleitend abgearbeitet. Das Ergebnis dieses Paketes stellt den umfassenden Praktikumsbericht dar, welcher alle erreichten Ergebnisse beschreibt und das Projekt zusammenfasst.

### **4.3 Entwicklungsphase 1**

Die Zielsetzung der ersten Entwicklungsphase stellte die Umsetzung der erarbeiteten Anwendungsidee in ein erstes funktionsfähiges Softwareartefakt dar. Abschließend sollte die Anwendung im Plenum präsentiert werden.

Während der ersten Entwicklungsphase hat sich herausgestellt, dass für die Entwicklung eines abnahmefertigen Frontend mehr Zeit, als zunächst eingeplant gewesen ist, aufgewendet werden musste. Dies begründete sich aufgrund einer längeren Einarbeitungsphase in die ausgewählten Technologien, sowie Hürden bei der Einarbeitung in das Modulkonzept von Baqend. Aufgrund dessen, dass die Implementierung der Anwendungslogik in Baqend rascher verlief, als erwartet, konnte der Zeitverzug bei der Entwicklung des Frontend ausgeglichen werden. Somit konnte das Projektteam den Zeitplan für die Entwicklungsphase anpassen und diesen umsetzen.

Die entstandenen Artefakte dieser Phase stellen die erste Version der Sunaware-Anwendung sowie die Projektdokumentation dieser Phase dar.

## **4.4 Entwicklungsphase 2**

Die Zielsetzung der zweiten Entwicklungsphase stellte die Umsetzung der in Phase 1 identifizierten nötigen Anpassungen und Ergänzungen an der Anwendung dar. Hierzu sind diese ihrer Priorität nach geordnet worden und dieser Reihenfolge nach bearbeitet worden. Aufgrund dessen, dass für das Beheben kleinerer Fehler in der Anwendung und die zu realisierenden Anpassungen mehr Zeit aufgewendet werden musste, als zunächst gedacht, hat das Projektteam durch die Verschiebung der Deadline nach Hinten hinaus auf Freitag den 25.08.2017 profitiert. Die hierdurch gewonnenen zeitlichen Ressourcen haben dazu beigetragen die Arbeit angenehmer verteilen zu können.

Die entstandenen Artefakte dieser Phase stellen die zweite Version der Sunaware-Anwendung sowie die Projektdokumentation dieser Phase dar.

## **4.5 Das Arbeiten in der Gruppe**

Abschließend sollen noch einige Punkte bezüglich der Gruppenarbeit genannt werden.

Aufgrund dessen, dass wenig bis keine Vorerfahrung in der Webentwicklung im Projektteam vorhanden war, ist das Einschätzen von Fähigkeiten und der notwendigen Dauer zur Einarbeitungszeit in unbekannte Technologien nicht ohne weiteres intuitiv möglich gewesen, sodass der Aspekt der Einarbeitung in der kurzen Projektzeit nicht zu unterschätzen gewesen ist. Dadurch, dass die Einarbeitung in neue Technologien natürlicher Weise bei jeder Person unterschiedlich schnell geht, ist es nicht selten zu Unterbrechungen im Arbeitsablauf gekommen, um Fragen zu klären und gemachte Fehler zu beheben. Dies und die ursprünglich angesetzte Deadline für die funktionierende Anwendung und den Bericht (24.08.2017, 10:15 Uhr) hat dazu beigetragen, dass zum Teil siloartig gearbeitet wurde. Gemeint ist hiermit, dass alle Mitglieder ihren mitgebrachten Fähigkeiten entsprechend versucht haben in möglichst kurzer Zeit möglichst produktiv zu sein, um das Gruppenziel zu erreichen und die Fristen einzuhalten. Hierunter hat das Einarbeiten in unbekannte Technologien zum Teil gelitten. Für das nächste Projekt sollte jedes Mitglied der Gruppe sich individuelle Lernziele setzen, die aufeinander abgestimmt sind, sodass zielgerichteter gearbeitet werden kann und der Lerneffekt für alle Mitglieder ähnlich positiv ausfällt.

Diese Tatsachen haben nicht zuletzt auch zu Unstimmigkeiten und Unzufriedenheiten im Projektteam geführt mit denen es umzugehen galt.

Dessen ungeachtet ist das Team stolz auf das entwickelte Artefakt und hat auf technischer und organisatorischer Ebene dazugelernt und neue Fähigkeiten entwickelt.



## 5 Abbildungsverzeichnis

Abbildung 1: Finden einer Funktion zur Approximation des Basis-UV-Indexes .....	3
Abbildung 2: UML Diagramm der Entwurfsphase .....	7
Abbildung 3: Projektstruktur .....	8
Abbildung 4: Login .....	11
Abbildung 5: Sign Up .....	11
Abbildung 6: User-Settings .....	12
Abbildung 7: Karte und Wetterinformationen .....	12
Abbildung 8: Wetterinformationen Teilen .....	13
Abbildung 9: Timer .....	13
Abbildung 10: Verbesserung der UV-Index Approximation .....	15
Abbildung 11: Positionierung, Berichte, Kartenstil, Teilen .....	16
Abbildung 12: UML Klassendiagramm der zweiten Entwicklungsphase .....	18