

## WRIP

### Fachübergreifendes Projekt

#### Gruppenteilnehmer:

Ahmed Abouismail: 565952

Leon Kling: 561099

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>2</b>
<b>2. Allgemeines</b>	<b>2</b>
2.1 Hauptziel	2
2.2 Ausgangssituation	2
2.3 Zielgruppe	2
<b>3. Konzeption</b>	<b>3</b>
3.1 Benutzeroberfläche	3
3.1 Workflow	4
<b>4. Funktionsbeschreibung</b>	<b>6</b>
4.1 Route eingeben	6
4.2 Reisezeiten eingeben	8
4.3 Karte Anzeigen	9
4.4 Wetter Bewertung	10
4.5 Alternative Zeiten vorschlagen	12
4.6 Route speichern	14
<b>5. Rahmenbedingungen</b>	<b>15</b>
5.1 Zeitplan	15
5.2 Technische Anforderungen	15

# 1. Einleitung

Aus dem vorliegenden Lastenheft werden folgenden Anforderungen in diesem Projekt nachgegangen:

Es wird ein Web Anwendung gebaut, die eine all in one Lösung sein soll, um einfach und schnell einen Roadtrip zu planen und Entscheidungen aufgrund des vorliegenden Wetters zu treffen. Die Web Anwendung soll die Eingabe von mehreren Zielen ermöglichen und dem dazugehörigen Ankunftsdatum.

Dem User soll eine Karte mit der Route angezeigt werden und dem jeweiligen Wetter für jeden Standort. Die Web Anwendung soll dann in der Lage sein dieses Wetter zu bewerten und bei Bedarf Alternativen vorzuschlagen. Am Ende steht dem User die Möglichkeit frei, sich die Karte zu speichern.

WRIP steht für Weather Roadtrip und ist der offizielle Name der Anwendung.

## 2. Allgemeines

### 2.1 Hauptziel

Hauptziele der Entwicklung sind folgende:

1. Implementierung der Routeneingabe
2. Visualisierung der Route auf einer Karte
3. Abrufen und Bewertung von Wetterdaten
4. Speicherung der Karte

### 2.2 Ausgangssituation

Für den einwandfreien Betrieb der Web Anwendung ist eine stabile Internetverbindung nötig, ein aktueller Browser und ein aktueller PC / Mobile Gerät.

### 2.3 Zielgruppe

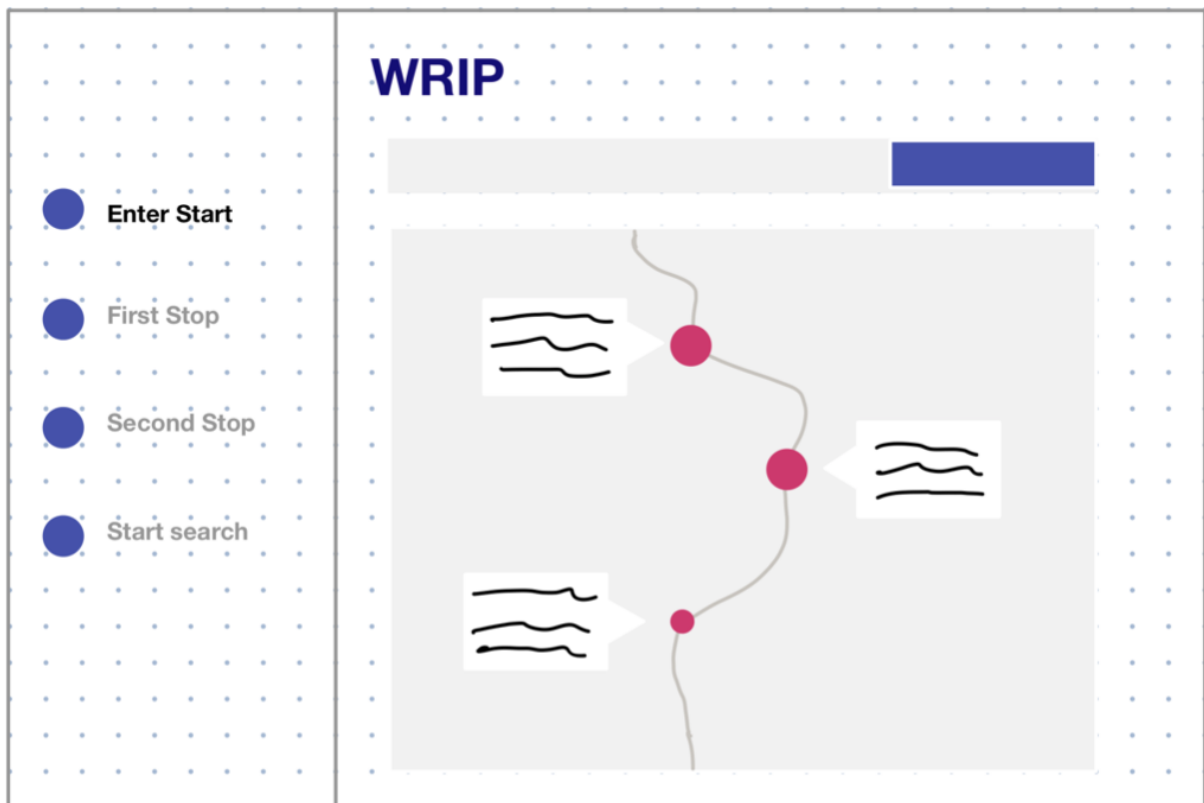
Die Anwendung ist für sämtliche Nutzer vorgesehen und ohne Vorwissen bedienbar. Sämtliche Schritte werden dem Nutzer erklärt und es gibt keinerlei Beschränkungen.

## 3. Konzeption

### 3.1 Benutzeroberfläche

Die Skizze war ein erstes Wireframe aus dem Lasterhaft. Sie beinhaltet alle Hauptelemente: die Karte, die Suche und die Navigationsleiste.

Auf der Karte wird die Information zu dem Wetter angezeigt.



### 3.1 Workflow

Zuerst ist es nötig, die Webseite aufzurufen. Es wird kein Login benötigt und auch auf einem Mobile Gerät ist die Website im Browser voll funktionstüchtig. Der User wird nun begrüßt und über die möglichen Features aufgeklärt bevor er seine Suche starten kann. Auf der Startseite wird nun auf der linken Seite die Routeneingabe angezeigt und auf der rechten die Karte. Alle Funktionen laufen innerhalb dieser Seite und der Nutzer wird durch jeden Schritt geführt. Am Ende kann der User die Funktion nutzen eine Karte seines Trips in einem vordefinierten Format zu speichern. Fehlerhafte Eingaben oder Systemfehler werden als Fehlermeldungen angezeigt.

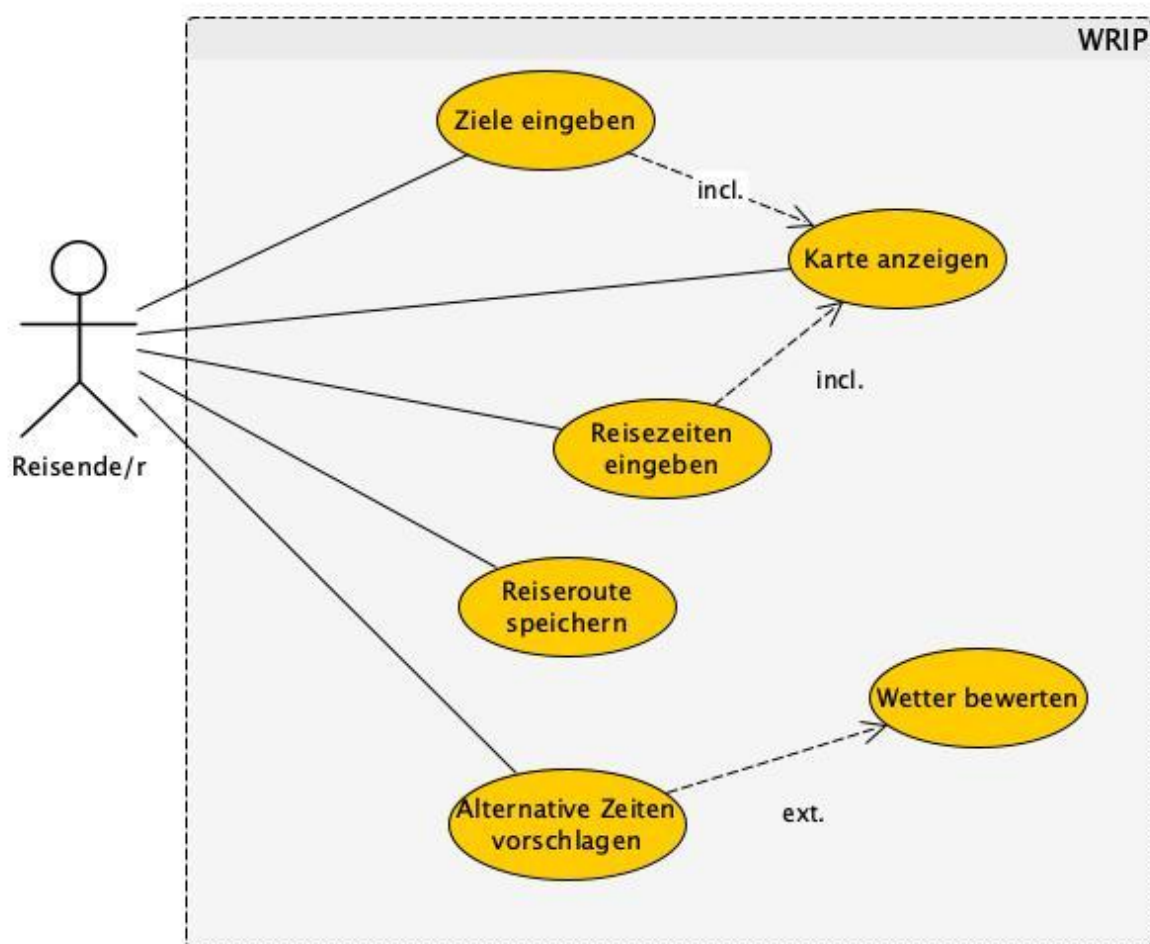


Abbildung 1. Use Case Diagramm: Beschreibt die verschiedenen Use Cases der Webanwendung

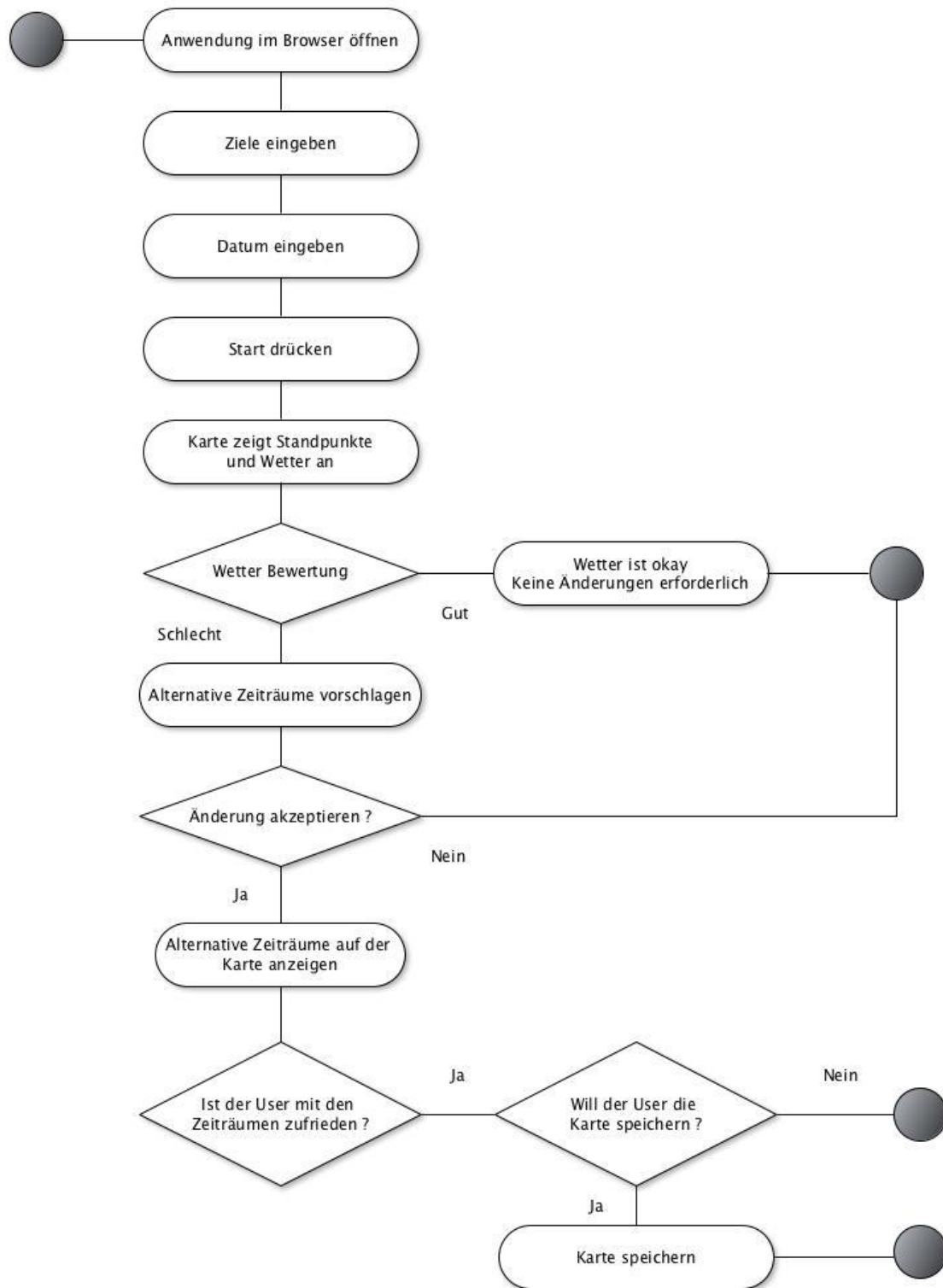


Abbildung 2. Aktivitätsdiagramm: Zeigt den groben Ablauf der Anwendung

## 4. Funktionsbeschreibung

### 4.1 Route eingeben

Beschreibung	Der User gibt seine Reiseziele ein, um seine seine Adressen festzulegen
Akteur	User
WF-Referenz	Abbildung 3.
Vorbedingung	<ul style="list-style-type: none"><li>• Erfolgreiche Verbindung mit der Webseite</li><li>• Stabile Internetverbindung</li></ul>
Daten Input	Eingabe in das Eingabefeld für die Adresse
Verarbeitungsschritte	<ol style="list-style-type: none"><li>1. Eingegebene Orte werden an die Google Maps API gesendet</li><li>2. Orte werden auf der Karte angezeigt</li><li>3. Koordinaten werden gespeichert</li></ol>
Ergebnis	Erfolgreiche Eingabe der Adressen, bedeutet das die Anwendung nun die Standorte ermitteln kann
Fehlermeldung	Erneute Aufforderung bei fehlerhafter Eingabe
Anforderungen	Lastenheft 1.2
Test Case	<ul style="list-style-type: none"><li>• das Eingeben mit korrekten Daten muss funktionieren</li><li>• bei falschen Daten muss eine erneute Eingabe erfordern</li></ul>

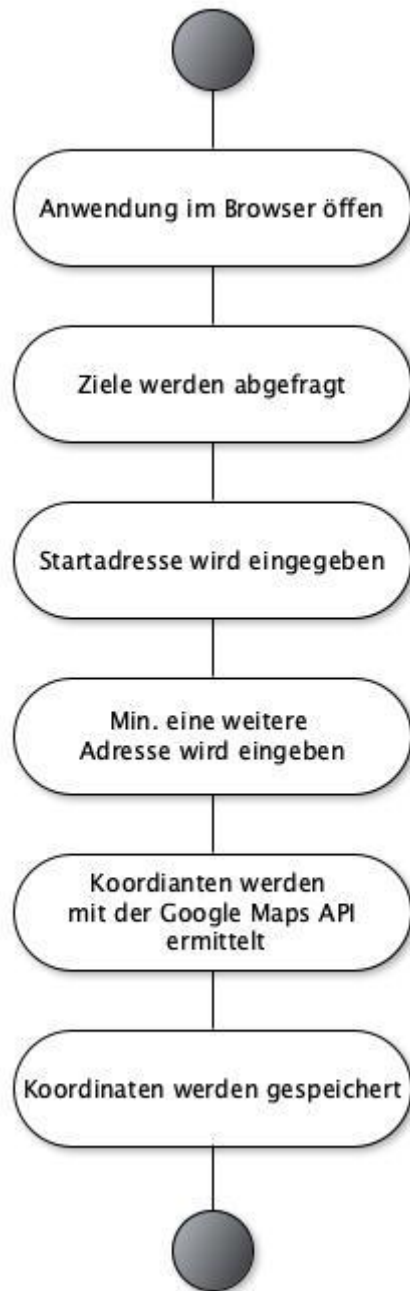


Abbildung 3. Aktivitätsdiagramm: Ablauf der Routeneingabe



## 4.2 Reisezeiten eingeben

Beschreibung	Die gewünschten Reisezeiten müssen eingegeben werden, damit das Wetter ermittelt werden kann
Akteur	User
WF-Referenz	Abbildung 2.
Vorbedingung	<ul style="list-style-type: none"><li>• Erfolgreiche eingabe von min. zwei Adressen</li></ul>
Daten Input	Eingabe in das Eingabefeld für die Reisezeiten
Verarbeitungsschritte	<ol style="list-style-type: none"><li>1. Eingebene Zeiten werden gespeichert</li><li>2. Zeiten werden auf der Karte angezeigt</li></ol>
Ergebnis	Erfolgreiche Eingabe der Zeiten, bedeutet das die Anwendung nun das Wetter ermitteln kann
Fehlermeldung	Erneute Aufforderung bei fehlerhafter Eingabe
Anforderungen	Lastenheft 1.7
Test Case	<ul style="list-style-type: none"><li>• das Eingeben mit korrekten Daten muss funktionieren</li><li>• bei falschen Daten muss eine erneute Eingabe erfordern</li></ul>

## 4.3 Karte Anzeigen

Beschreibung	Die Karte zeigt dem User jederzeit wo sich seine eingegebenen Ziele befinden.
Akteur	User
WF-Referenz	Abbildung 2.
Vorbedingung	<ul style="list-style-type: none"><li>• Erfolgreiche eingabe von min. zwei Adressen</li><li>• Erfolgreiche eingabe von Reisezeiten</li></ul>
Verarbeitungsschritte	<ol style="list-style-type: none"><li>1. Adressen werden an die Google Maps API gesendet</li><li>2. Karte zeigt die Adressen an</li><li>3. Wetterdaten werden über die OpenWeather API ermittelt</li><li>4. Wetterdaten werden für jeden Standpunkt angezeigt</li></ol>
Ergebnis	Die angegebenen Ziele werden mit Informationen zu dem Wetter von jedem Standort auf der Karte angezeigt
Fehlermeldung	Warnung erscheint, dass die Daten möglicherweise falsch sein könnten oder nicht aktualisiert wurden
Anforderungen	Lastenheft 1.3
Test Case	<ul style="list-style-type: none"><li>• Beispielwerte senden nach erstmaligem Starten des Systems</li><li>• Bei falschen Daten wird ein Fehler angezeigt</li><li>• Test für den Request an die API</li></ul>

## 4.4 Wetter Bewertung

Beschreibung	Das System prüft ob das Wetter als gut befunden wird oder nicht
Akteur	System
WF-Referenz	Abbildung 4.
Vorbedingung	<ul style="list-style-type: none"> <li>• Erfolgreiche Eingabe von Adressen</li> <li>• Erfolgreiche Eingabe von Reisezeiten</li> <li>• Erfolgreiche verbindung mit Google Maps</li> <li>• Erfolgreiches verbindung mit OpenWeather API</li> </ul>
Verarbeitungsschritte	<ol style="list-style-type: none"> <li>1. Die Daten werden an die OpenWeather API gesendet.</li> <li>2. Die Api liefert die Wetterinformationen zu jedem Standort zurück.</li> <li>3. Die gelieferte Daten werden verarbeitet und mit gewünschten Parametern verglichen</li> </ol>
Ergebnis	Die Bewertung von dem Wetter für jeden Standort wird angezeigt.
Fehlermeldung	Warnung erscheint, dass die Daten möglicherweise falsch sein könnten oder nicht aktualisiert wurden
Anforderungen	Lastenheft 1.4
Test Case	<ul style="list-style-type: none"> <li>• Beispielwerte senden nach erstmaligem Starten des Systems</li> <li>• Bei falschen Daten wird ein Fehler angezeigt</li> <li>• Test für den Request an die API</li> </ul>

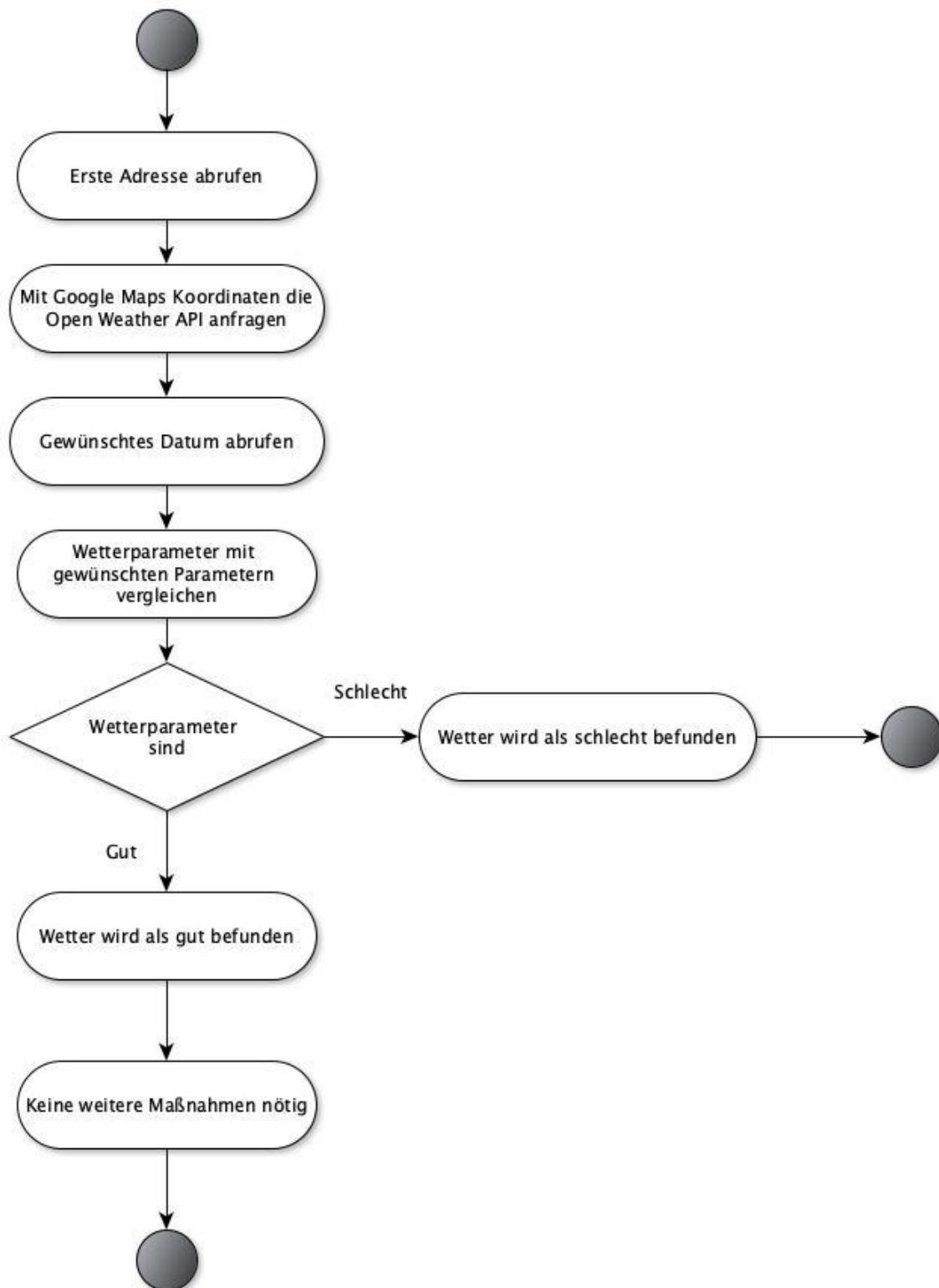


Abbildung 4. Aktivitätsdiagramm: Ablauf der Wetterbewertung

## 4.5 Alternative Zeiten vorschlagen

Beschreibung	Das System schlägt alternative Zeitpunkt vor, an denen das Wetter besser ist
Akteur	System
WF-Referenz	Abbildung 5.
Vorbedingung	<ul style="list-style-type: none"><li>• Wetter muss für Schlecht empfunden worden sein</li><li>• Erfolgreiche verbindung mit Google Maps</li><li>• Erfolgreiches verbindung mit OpenWeather API</li></ul>
Verarbeitungsschritte	<ol style="list-style-type: none"><li>1. Wetterdaten für die kommenden Tage für jeden Standort prüfen</li><li>2. bei übereinstimmung mit Wunschparametern Wetter für gut empfinden</li></ol>
Ergebnis	Die alternative Zeiten und Wetter werden auf der Karte angezeigt
Fehlermeldung	Warnung erscheint, dass die Daten möglicherweise falsch sein könnten oder nicht aktualisiert wurden
Anforderungen	Lastenheft 1.6
Test Case	<ul style="list-style-type: none"><li>• Beispielwerte senden nach erstmaligem Starten des Systems</li><li>• Bei falschen Daten wird ein Fehler angezeigt</li><li>• Test für den Request an die API</li></ul>

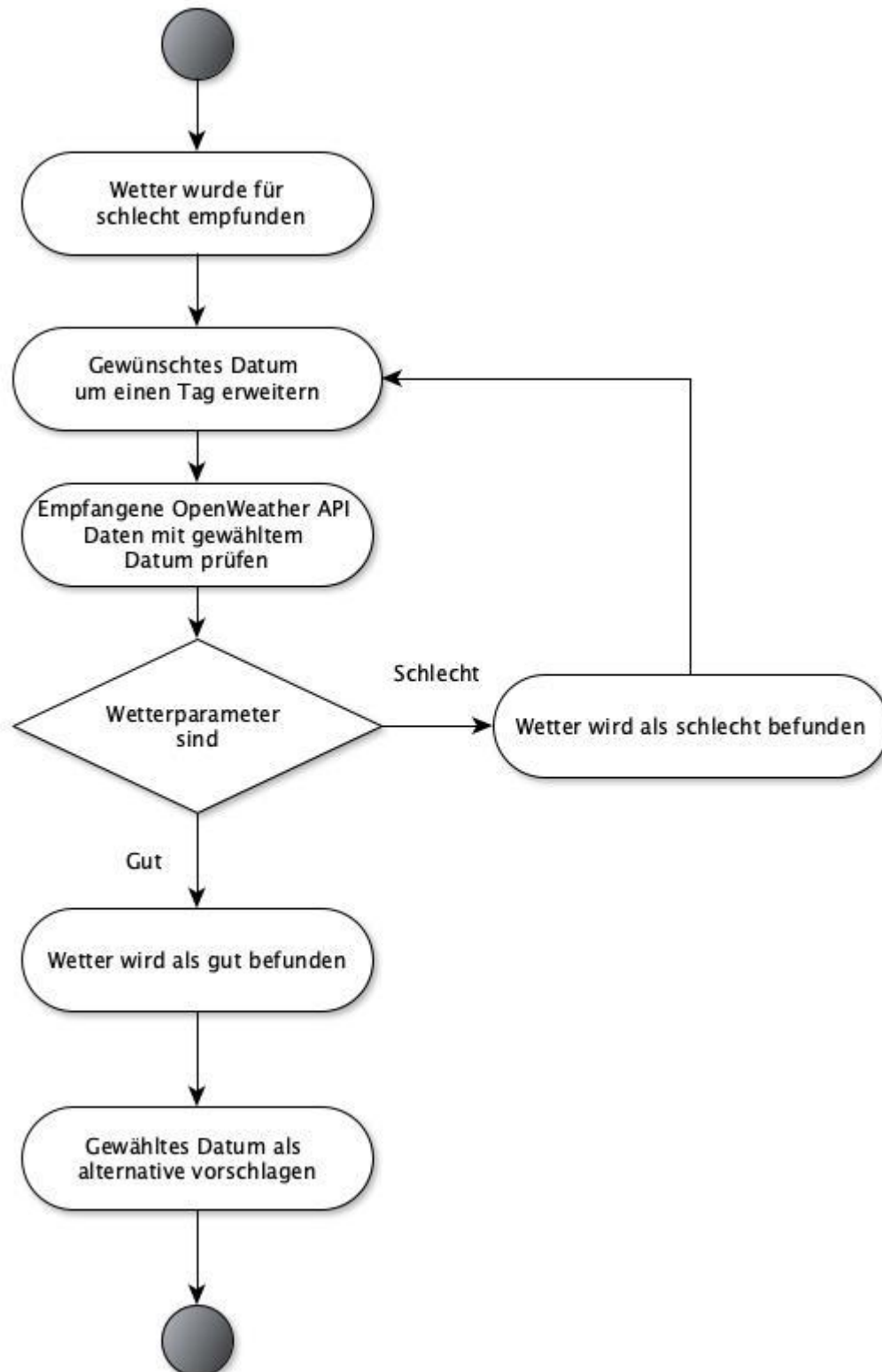


Abbildung 5. Aktivitätsdiagramm: Ablauf von Alternative Zeiten vorschlagen

## 4.6 Route speichern

Beschreibung	Der User kann seine Route speichern und herunterladen
Akteur	User
WF-Referenz	Abbildung 2.
Vorbedingung	<ul style="list-style-type: none"> <li>Die Karte samt Wetter wird angezeigt</li> <li>Der User möchte keine veränderungen vornehmen</li> </ul>
Daten Input	<ul style="list-style-type: none"> <li>Es wird auf den Speicher Button von dem User geklickt</li> <li>Es wird ein Speicherpfad ausgewählt</li> </ul>
Verarbeitungsschritte	<ol style="list-style-type: none"> <li>Der User hat die Wahl zwischen verschiedenen Speicherpfäden</li> <li>Ein Screenshot der Karte wird aufgenommen und als JPEG in dem angegebenen Pfad gespeichert</li> </ol>
Ergebnis	Eine Bilddatei, die auf dem Gerät des Users gespeichert wird
Fehlermeldung	<ul style="list-style-type: none"> <li>Die Bilddatei konnte nicht in dem angegebenen Pfad gespeichert werden</li> <li>Es konnte keine Bilddatei erstellt werden</li> </ul>
Anforderungen	Lastenheft 1.9
Test Case	<ul style="list-style-type: none"> <li>Bei fehlerhaftem speichern wird ein Fehler angezeigt</li> </ul>

## 5. Rahmenbedingungen

### 5.1 Zeitplan

Angaben zum Zeitplan befinden sich in der beiliegenden Project MS Datei.

### 5.2 Technische Anforderungen

Das Projekt wird mit VSCode entwickelt und auf GitLab Synchronisiert. Als SCRUM Tool wird ebenfalls GitLab verwendet und ebenfalls wird die gesamte Sprint und Projektplanung in GitLab stattfinden. Gehostet soll am Ende die Anwendung auf einem HTW Server.