VVETTER aPP

## App-Name

### Version 1.0.0, Datum: 3. Juni 2023

Hier kannst du einen Printscreen deiner App einfügen.

Abstract (Kurzbeschreibung)

Wir haben eine Wetter App umgesetzt, die Hilfe mit einer API verschiedene Daten aus verschiedenen Regionen anzeigt. So wie z.B. Grad Celcius, Lufthochdruck, oder Windgeschwindigkeit.

Der Benutzer kann Orte hinzufügen, welche dann auf einer separaten Seite gespeichert werden. Mithilfe eines Buttons (siehe Screenshot), kann der Benutzer auf der Settings Page, den aktuellen Ort anwählen, der auf der Home-Page angezeigt werden soll.

Auf der Home-Page werden die aktuellen Daten Luftdruck, Temperatur und Windgeschwindigkeit angezeigt. Ebenfalls auch eine Prognose für die nächsten Tage angezeigt.

user stories

**User Story 1**

* **Als** Benutzer einer Wetter-App
* **möchte ich** aktuelle Wetterdaten wie Temperatur, Luftfeuchtigkeit, und Luftdruck für meinen Standort angezeigt bekommen,
* **damit** ich meine Tagesplanung besser gestalten kann.

**User Story 2**

**Als** Reisender

**möchte ich** Wetterinformationen für einen spezifischen Zielort abfragen können,

**damit** ich mich entsprechend auf die Reise vorbereiten kann (z.B. Kleidung und Aktivitäten).

Mockups

Die Folgenden Mockups stellen den ursprünglich geplanten Aufbau der Applikation dar.  
zum Zeitpunkt des Erstellens war noch nicht klar welche API verwendet werden sollt weshalb diese vom end Resultat abweichen.

Ein Bild, das Text, Screenshot, Design enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Handy, mobiles Gerät enthält.

Automatisch generierte Beschreibung

Home Screen

Auf der Home-Page sieht der Benutzer die ausgegebenen Daten von seinem Ort, den er auf der Settings-Page angewählt hat bzw. vorher selbst hinzugefügt hat.

Man kann nun die Prognose für die nächsten Tage sehen, dies beinhaltet die Temperatur, die Windgeschwindigkeit und ein entsprechendes Icon.

Location Page

Bei der Location Page, kann der Benutzer neue Orte hinzufügen bzw. Locations (English). Es gibt INPUT Fields. In der realisierten App kann der Benutzer anstatt Region und Kanton, den Breiten und Längengrad des Ortes eingeben. Nachdem er die Daten eingegeben hat, auf den Button gedrückt hat, kommt ein Pop Up das Ihn nochmals fragt, ob er den Ort wirklich hinzufügen will.

Settings Page

Auf dieser Page kann der Benutzer eine Übersicht aller Orte sehen, die einmal auf der Location-Page eingegeben hatte. Mithilfe eines Buttons kann er auswählen, welchen Ort er gerade auf der Home-Page angezeigt bekommen will.

Technische Realisierung

Komponenten

# ****HomeScreen****

**Beschreibung**: Der Startbildschirm der App, der Informationen über das Wetter beim Ort anzeigt und Navigation zu anderen Bildschirmen Locations und Settings ermöglicht.

**Funktionalitäten**:

* Anzeige von Wetterdaten und Sensorinformationen.
* Navigation zu anderen Bildschirmen wie der Locations Page.

# ****LocationScreen****

**Beschreibung**: Ermöglicht dem Benutzer, neue Orte hinzuzufügen mithilfe Input Felder. Er kann direkt den Breitengrad und Längengrad eingeben.

**Funktionalitäten**:

* Eingabefelder für Ort, Längengrad und Breitengrad.
* Ein Button zum Hinzufügen neuer Orte.
* Modale Dialoge zur Bestätigung der Aktionen.

# ****SettingsScreen****

**Beschreibung**: Hier werden die hinzugefügten Orte verwaltet. Der angewählte Ort mittels Buttons, wird auf der Homepage angezeigt.

**Funktionalitäten**:

* Anzeige der hinzugefügten Orte. Datenspeicherung.

**Lokale Speicherung**

* + **AsyncStorage**: Die App verwendet AsyncStorage zur lokalen Speicherung der Orte. Dies ermöglicht es, Daten auch nach dem Schliessen der App zu behalten.

**Datenquellen**

* + **API-Anfragen**: Für die Wetterdaten wird eine Wetter-API verwendet.   
    wir verwenden die API von <http://www.7timer.info/doc.php?lang=en>

**Datenstruktur**

* + **Ortsdaten**: Die Datenstruktur für die Orte umfasst Felder wie Ort, Längengrad und Breitengrad. Diese werden als JSON-Objekte in AsyncStorage gespeichert.

# Modals

In unserer App wird ein Modal verwendet, um den Benutzer zu fragen, ob er einen neuen Ort wirklich hinzufügen möchte. Dies stellt sicher, dass Benutzer ihre Eingaben bestätigen können, bevor die Aktion tatsächlich ausgeführt wird

Testing

In diesem Kapitel definiert ihr die manuellen Tests, die durchgeführt werden. Es müssen mind. 4 manuelle Tests definiert werden.

Hier ein Beispiel eines Tests:

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | 1 |
| Anforderungen | Man sollte auf der Locations Page neue Orte hinzufügen können. |
| Vorbedingungen | Die App muss funktionieren und man sollte auf die Locations Page kommen |
| Ablauf | Der Benutzer muss einen Input machen, um eine neue Location hinzuzufügen |
| Erwartetes Resultat | Der Ort ist auf der Settings Page ersichtlich. |

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | 2 |
| Anforderungen | Man möchte das Wetter am aktuellen Ort wissen |
| Vorbedingungen | Der richtige Ort muss angewählt werden |
| Ablauf | Der Tester geht auf die Home-Page und schaut ob alle Daten, wie Prognosen, Grad Celsius, Windstärke und Luftdruck angezeigt wird |
| Erwartetes Resultat | Man sollte nun wissen man anziehen sollte ⁠Grinsen |

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | 3 |
| Anforderungen | Man sollte auf der Settings-Page anwählen, welchen Ort man angezeigt haben möchte, |
| Vorbedingungen | Die App muss funktionieren und man sollte auf die Settings Page kommen |
| Ablauf | Der Tester muss einen Button drücken, damit die Daten des Ortes angezeigt werden können |
| Erwartetes Resultat | Der Ort sollte auf der Home-Page zu sehen sein. |

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | 4 |
| Anforderungen | Man sollte auf der Locations Page bestehende Orte entfernen können. |
| Vorbedingungen | Die App muss funktionieren, man sollte auf die Locations Page kommen und es muss mindestens ein Ort hinzugefügt worden sein. |
| Ablauf | Der Benutzer muss einen bestehenden Ort auswählen und auf einen "Ort entfernen" Button klicken. |
| Erwartetes Resultat | Der Ort wird von der Liste der Orte auf der Settings Page entfernt. |

Testprotokoll

In diesem Kapitel wird die Testausführung protokolliert. Welche Tests wurden wann von wem mit welchem Resultat durchgeführt.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Person / Datum | Erfolgreich | Bemerkungen |
| 01 | T.Brasser  05.07.2024 | Ja | Funktioniert einwandfrei |
| 02 | D.Meister  05.07.2024 | Ja | Alles tiptop |
| 03 | L.Ljubisavljevic  05.07.2024 | Ja | Funktionert |
| 04 | L.Roth  05.07.2024 | Ja | Noch nie so gut geklappt |

fazit

Wir hatten ein paar Probleme bei der Speicherung der Daten (Async Storage) und anfangs mit der API, da sie anfangs die Daten nicht direkt auf der Home-Page angezeigt hatte.

Auf der anderen Seite gelangte es uns sehr gut die verschiedenen Pages mit dem Design und Input zu programmieren. Auch die generelle Einbindung der API, funktionierte ziemlich gut.

Wir haben unser Mockup gut umgesetzt, also haben unser Ziel erreicht. Auch in der entsprechenden Zeit.