

GEOREMINDER

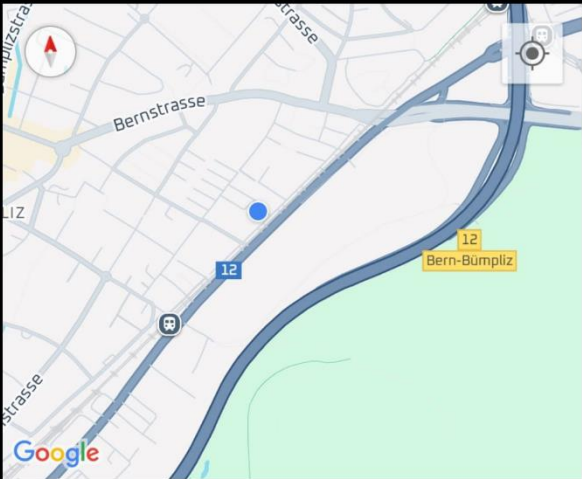
Deine Aufgaben erinnern dich genau dort, wo du sie brauchst.

Version 1.0.0, Datum: 4. Juli 2025

Jonas Schären, Aaron Mettler, Loris Stahlberg

Erinnerung erstellen

Erstelle eine neue Erinnerung







Koordinaten

Radius (in Metern)

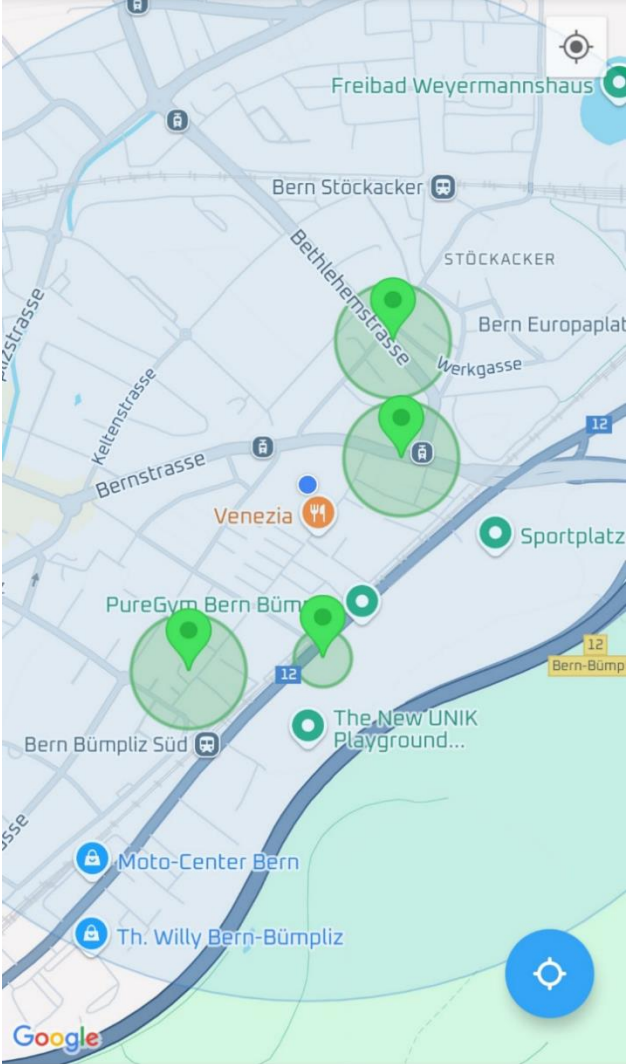
Titel





Inhalt

Karte

Alle Erinnerungen auf der Karte



Georeminderr

1

ABSTRACT

An dieser Stelle kommt eine kurze Einführung, die in 5 – 10 Sätzen das Projekt beschreibt.

GeoReminder ist eine innovative App, die standortbasierte Erinnerungen ermöglicht, um das Problem vergessener Aufgaben an bestimmten Orten zu lösen. Nutzer können auf einer Karte einen Ort auswählen, einen Umkreis festlegen (z. B. 500 Meter) und eine Erinnerung mit Titel erstellen, die beim Betreten des Bereichs ausgelöst wird. Im Gegensatz zu herkömmlichen zeitbasierten Reminder-Apps, die oft zum falschen Zeitpunkt benachrichtigen, weil der genaue Aufenthaltszeitpunkt unvorhersehbar ist, bindet GeoReminder Erinnerungen direkt an den Ort. Dadurch wird sichergestellt, dass Nutzer genau in dem Moment erinnert werden, in dem sie handeln können. Die App adressiert das häufige Problem, dass ortsabhängige Aufgaben vergessen werden, weil die Erinnerung nicht mit dem tatsächlichen Aufenthaltsort verknüpft ist. GeoReminder bietet somit eine praktische Lösung für eine präzise, kontextbezogene Aufgabenverwaltung.

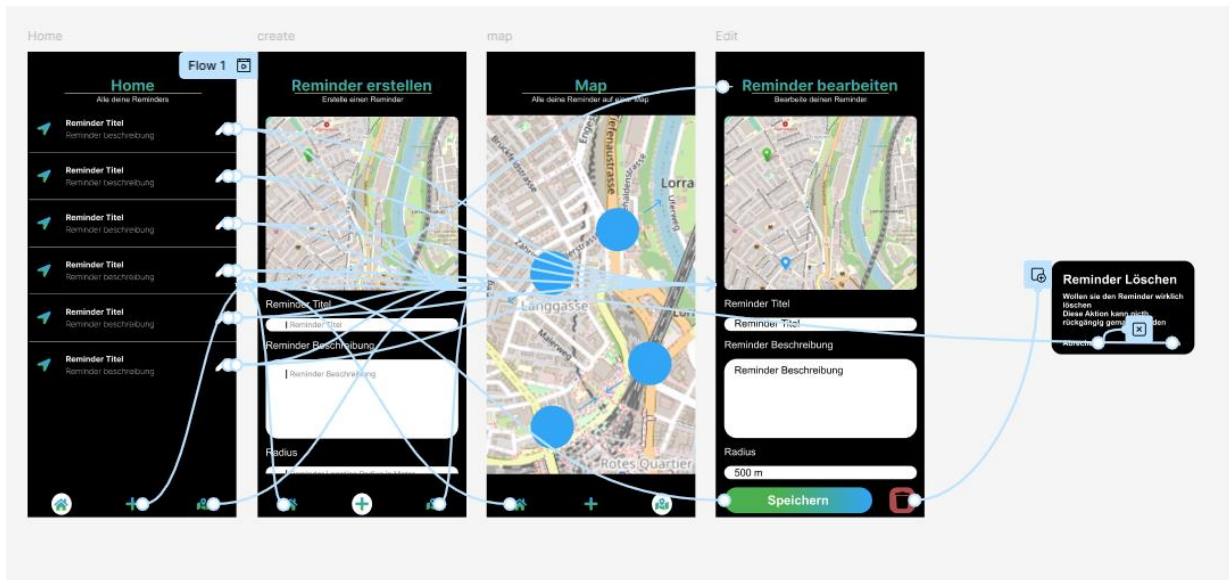
USER STORIES

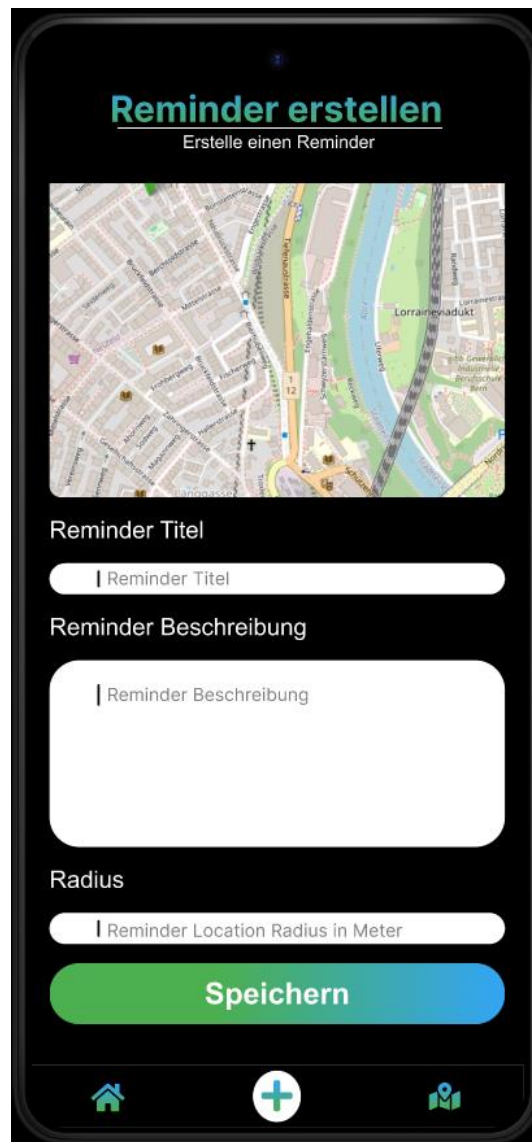
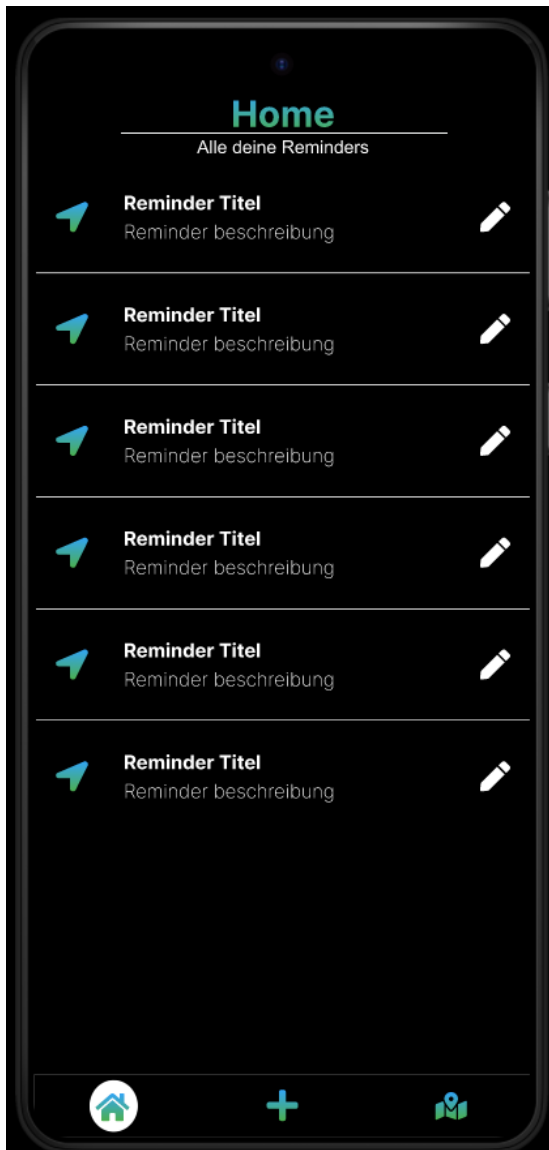
- **Als Nutzer** möchte ich eine neue Erinnerung erstellen,
damit ich Dinge genau dann erledige, wenn ich am richtigen Ort dafür bin.
- **Als Nutzer** möchte ich eine Erinnerung löschen können,
damit ich veraltete oder nicht mehr benötigte Erinnerungen entfernen kann.
- **Als Nutzer** möchte ich alle meine Erinnerungen in einer Liste sehen,
damit ich den Überblick behalte, welche Erinnerungen ich bereits erstellt habe.
- **Als Nutzer** möchte ich eine Erinnerung antippen können,
damit ich die genaue Position der Erinnerung auf einer Karte sehen kann.
- **Als Nutzer** möchte ich auf eine Edit-Icon klicken können,
damit ich direkt zur Bearbeitungsansicht gelange und sie bei Bedarf ändern oder löschen kann.
- **Als Nutzer** möchte ich eine Karte sehen, auf der alle Erinnerungen angezeigt werden,
damit ich visuell erkennen kann, wo überall Erinnerungen gesetzt wurden.
- **Als Nutzer** möchte ich, dass jede Erinnerung auf einer Karte angezeigt wird,
damit ich schnell sehe, welche Erinnerungen an welchen Orten gespeichert sind.
- **Als Nutzer** möchte ich, dass meine Erinnerungen lokal gespeichert werden,
damit sie auch nach dem Schliessen der App noch vorhanden sind.

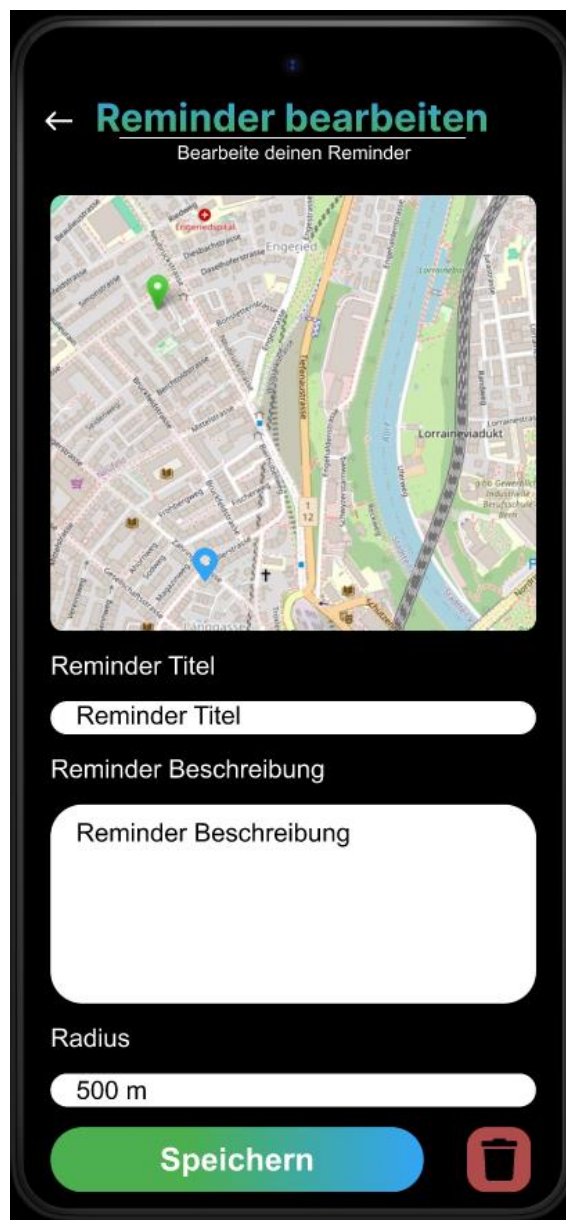
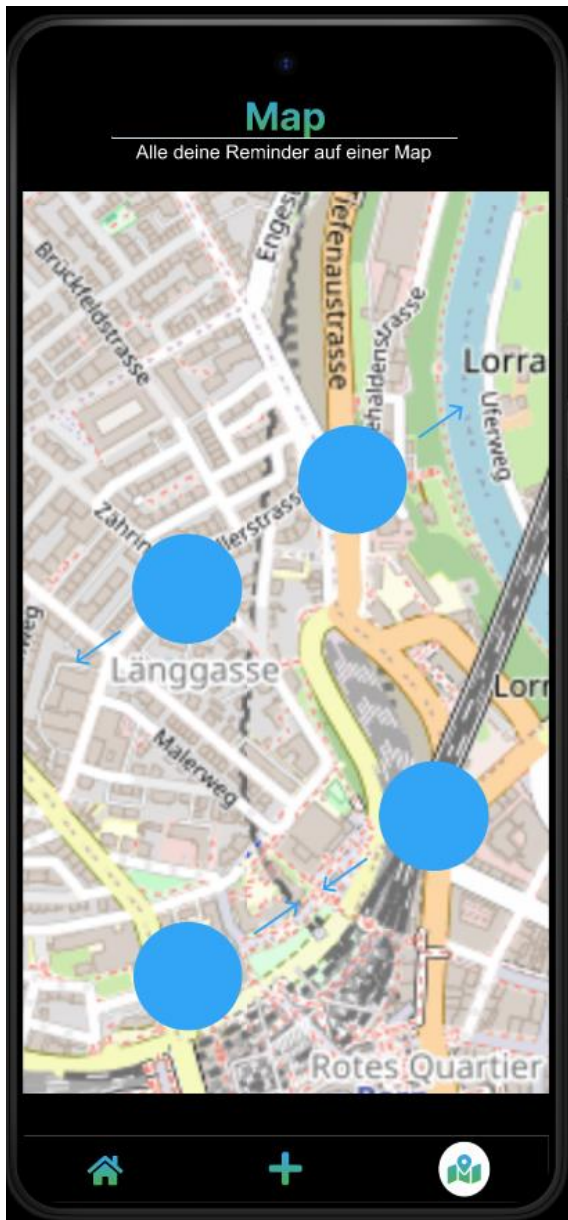
MOCKUPS

An dieser Stelle den Prototypen deiner App beschreiben. Füge einen Printscreen deiner Screens ein und beschreibe jeden Screen einzeln.

Figmalink: [Figmalink](#)









ABWEICHUNGEN VOM MOCKUP

Durch das hinzufügen der Onlinesynchronisation, wurde ebenfalls ein neuer Tab Profil hinzu gefügt der im Mockup nicht ersichtlich ist. Die Weissen punkte im Bottomnav haben wir aus Design gründen entfernt. Aus gleichen Gründen haben wir auch bei Buttons zum Teil die Borderradius angepasst.

TECHNISCHE REALISIERUNG

Technische Dokumentation – GeoReminder

1. Projektstruktur & Architektur

Die GeoReminder-App ist als modulare, plattformübergreifende Mobile-App mit React Native und Expo entwickelt. Die Verzeichnisstruktur ist folgendermassen gegliedert:

- **components/**: Wiederverwendbare UI-Komponenten wie MemoizedMapView, FormField, ReminderListItem, SearchBar, CoordinateInput usw.
- **app/**: Enthält die Hauptlogik und die verschiedenen Screens, organisiert nach Tabs im Unterordner (tabs) (z. B. home, map, create).
- **assets/**: Statische Ressourcen wie Bilder und Icons.
- **global.css** und **tailwind.config.js**: Für das Styling und die Definition des Designsystems.

Ein explizites Architekturpattern wie MVVM oder Clean Architecture wird nicht verwendet, jedoch ist die Trennung von UI (components) und Logik (app) klar umgesetzt. Die Modularisierung sorgt für gute Wartbarkeit und Erweiterbarkeit.

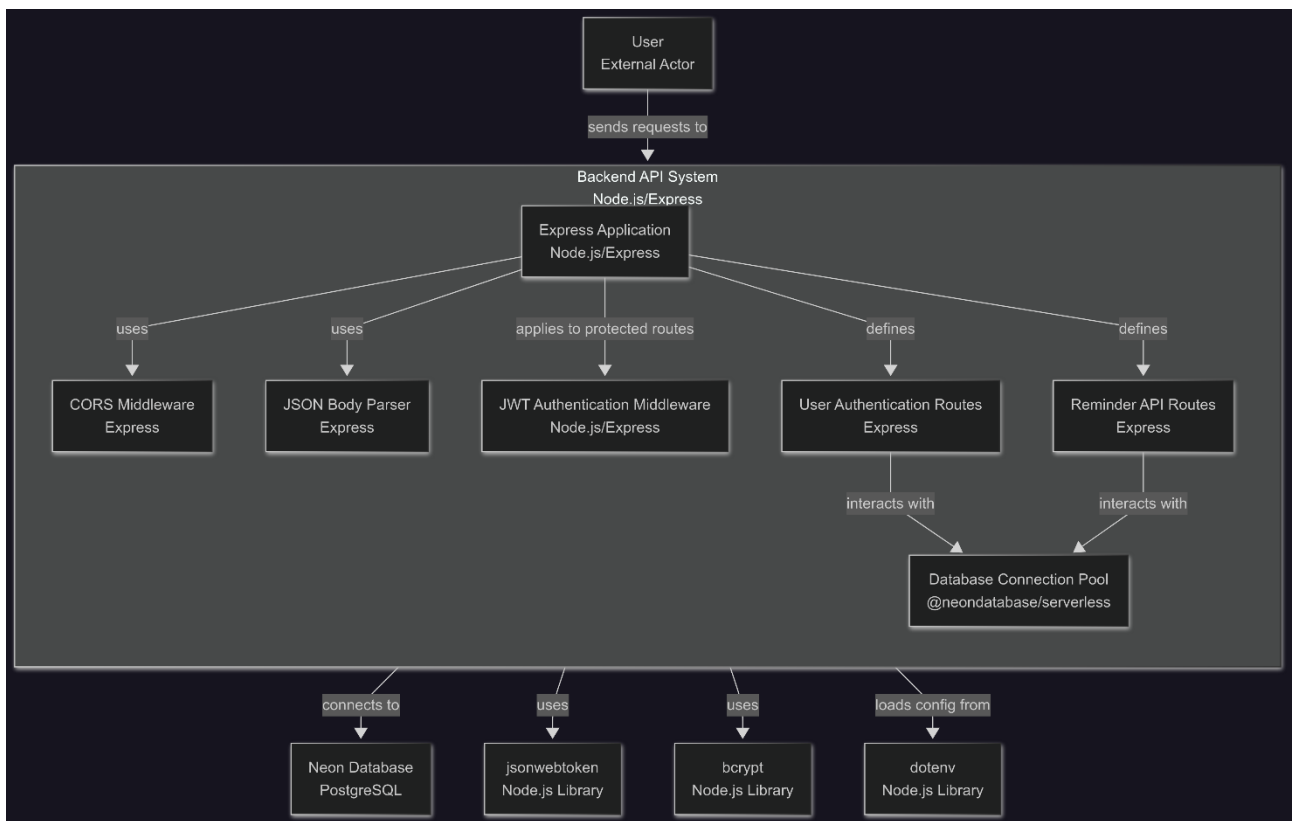


Abbildung 1: Backend

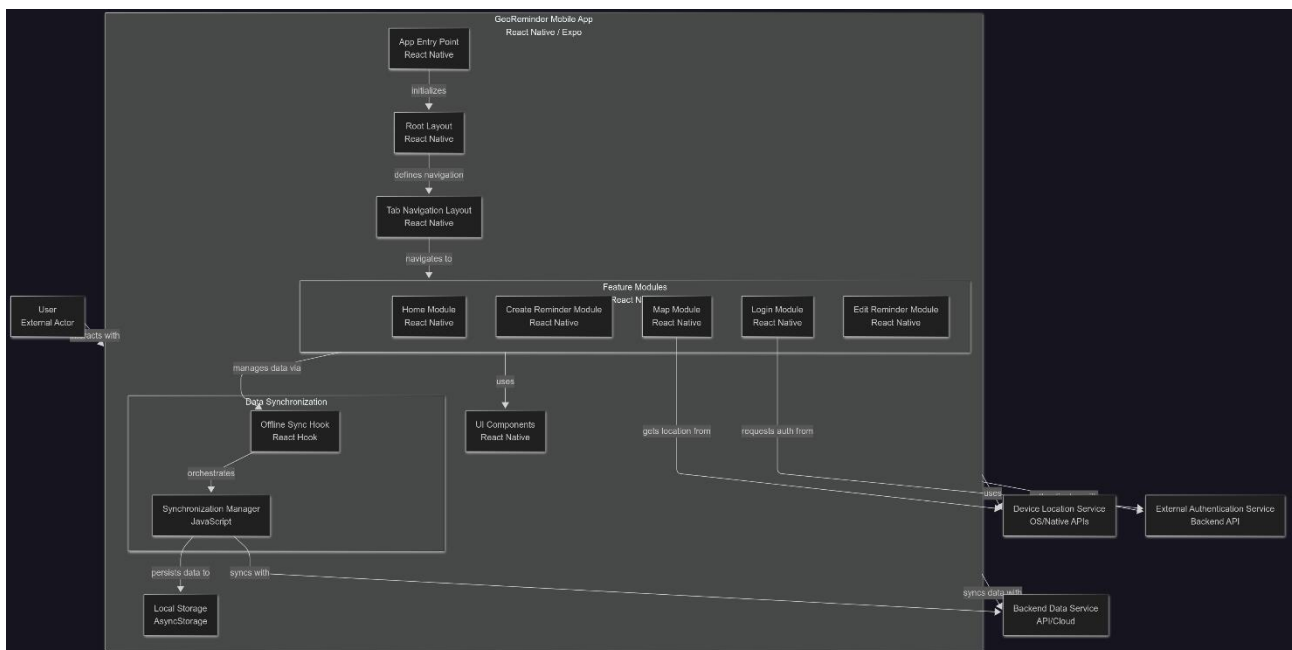


Abbildung 2: Frontend

2. State Management

Das State Management erfolgt komponentenbasiert mit React Hooks (useState, useEffect). Der Zustand (z. B. Erinnerungen, Benutzerposition) wird lokal in den jeweiligen Komponenten verwaltet. Für persistente Daten wird AsyncStorage genutzt.

3. Standortfunktionen & Geofencing

- Die Bibliothek **react-native-maps** wird für die Kartenanzeige und Marker verwendet.
- Die aktuelle Nutzerposition wird mit **expo-location** ermittelt.
- Die Überprüfung, ob sich der Nutzer in einem Zielgebiet befindet, erfolgt durch Vergleich der aktuellen Koordinaten mit gespeicherten Marker-Positionen.
- Geofencing und Hintergrund-Location-Tracking können (je nach Plattform und Berechtigung) mit **expo-location** erweitert werden.

4. Benachrichtigungen

- Lokale Benachrichtigungen werden mit **expo-notifications** (oder einer vergleichbaren Bibliothek) realisiert.
- Eine Benachrichtigung wird ausgelöst, wenn der Nutzer einen gespeicherten Ort erreicht (z. B. durch ein Geofencing-Event).

5. Plattformübergreifende Besonderheiten

- Unterschiede bei Berechtigungen und Hintergrundverhalten zwischen iOS und Android werden durch plattformspezifische Abfragen und Hinweise gelöst.
- Die App prüft, ob die notwendigen Berechtigungen erteilt wurden, und informiert den Nutzer bei Problemen.
- UI-Elemente wie Header oder Modals werden plattformgerecht angepasst (z. B. zusätzlicher Zurück-Button für iOS).

6. Offline-Fähigkeit & Datenspeicherung

- Erinnerungen werden lokal auf dem Gerät mit **AsyncStorage** gespeichert.
- Die App funktioniert auch ohne Internetverbindung, insbesondere beim Erstellen, Anzeigen und Verwalten von Erinnerungen.
- Für Kartendaten und Standortdienste ist eine Internetverbindung erforderlich.
- Für die Online Synchronisation ist ebenfalls eine Internetverbindung notwendig

7. Berechtigungs-Handling

- Der Zugriff auf Standortdienste wird beim Start der App abgefragt.
- Für iOS muss der Nutzer ggf. die Freigabe manuell in den Einstellungen anpassen.

8. UI & Designsystem

- Das Styling erfolgt mit **Tailwind-RN** für ein konsistentes und modernes Design.
- Die Styles sind zentral in `tailwind.config.js` definiert.

9. Build-Tools

- Die App wird mit **Expo** entwickelt und gebaut.
- Der Build und das Deployment erfolgen über **EAS Build**.

10. Komponentenstruktur

Die wichtigsten Komponenten sind:

- **MemoizedMapView**: Kapselt die Kartenanzeige und Marker.
- **FormField, CoordinateInput, SearchBar, SearchResults**: Für die Eingabe und Suche von Orten sowie die Verwaltung der Erinnerungen.
- **ReminderListItem**: Stellt einzelne Erinnerungen in einer Liste dar.

- **BottomnavIcon, Header, LoadingView, SubmitButton, Trash:** Zusätzliche UI-Komponenten für Navigation, Layout und Interaktion.

Die Navigation erfolgt über ein Tab-basiertes Layout im Verzeichnis (tabs).

11. Datenhaltung und -verarbeitung

- Erinnerungsdaten (Name, Beschreibung, Koordinaten) werden lokal gespeichert.
- Marker auf der Karte werden aus diesen Daten geladen und als Marker-Komponenten angezeigt.
- Die App arbeitet ausschließlich mit lokal gespeicherten Daten; externe Datenquellen werden nicht genutzt.
- Die aktuelle Nutzerposition wird über die Standortdienste des Geräts ermittelt.
- Für angemeldeten Benutzer werden die Daten zusätzlich Online gesichert

12. Ablauf und Interaktion

- Beim Starten der App wird die aktuelle Position des Nutzers ermittelt und die Karte initialisiert.
- Nutzer können neue Erinnerungen anlegen, indem sie einen Ort auswählen und die zugehörigen Daten eingeben.
- Alle gespeicherten Erinnerungen werden als Marker auf der Karte angezeigt.
- Die Navigation zwischen den Hauptfunktionen erfolgt über die Tab-Navigation.

13. Erweiterbarkeit

Die modulare Struktur ermöglicht eine einfache Erweiterung, z. B. um neue Ansichten, Datentypen oder Speicheroptionen zu integrieren.

TESTING

Abschnitt	Inhalt
ID	ST_01
Anforderungen	Als Nutzer möchte ich eine neue Erinnerung erstellen, damit ich Aufgaben genau dann erledige, wenn ich mich am richtigen Ort befinde.
Vorbedingungen	Der Tester befindet sich in der geöffneten App auf dem Home Screen.
Ablauf	<ol style="list-style-type: none">1. Klicke auf das Plus-Icon unten in der Navbar.2. Wähle einen Standort auf der Karte aus und setze einen Pin, indem du auf die gewünschte Position tippst.3. Gib in den Feldern darunter einen Radius, einen Titel und eine Beschreibung ein.4. Drücke den „Erstellen“-Button.
Erwartetes Resultat	Du wirst zum Home Screen zurückgeführt und siehst die soeben erstellte Erinnerung in der Liste.
ID	ST_02
Anforderungen	Als Nutzer möchte ich eine Erinnerung löschen können, damit ich veraltete oder nicht mehr benötigte Erinnerungen entfernen kann.
Vorbedingungen	Der Tester befindet sich in der geöffneten App auf dem Home Screen.
Ablauf	<ol style="list-style-type: none">1. Klicke auf das Edit-Symbol eines Reminders.2. Scrolle nach unten und drücke den Button mit dem Mülleimer-Symbol.3. Bestätige die Löschung.
Erwartetes Resultat	Die gelöschte Erinnerung wird auf dem Home Screen nicht mehr angezeigt.
ID	ST_03
Anforderungen	Als Nutzer möchte ich alle meine Erinnerungen in einer Liste sehen, damit ich den Überblick behalte, welche Erinnerungen ich erstellt habe.
Vorbedingungen	Der Tester befindet sich in der geöffneten App auf dem Home Screen.

Ablauf	1. Auf dem Home Screen werden alle erstellten Erinnerungen angezeigt. 2. Streiche mit dem Finger von oben nach unten, um die Liste zu aktualisieren.
--------	---

Erwartetes Resultat	Ein Ladebalken erscheint und verschwindet nach dem Laden wieder.
---------------------	--

ID	ST_04
----	-------

Anforderungen	Als Nutzer möchte ich eine Erinnerung antippen können, damit ich die genaue Position der Erinnerung auf einer Karte sehen kann.
---------------	---

Vorbedingungen	Der Tester befindet sich in der geöffneten App auf dem Home Screen.
----------------	---

Ablauf	1. Klicke auf einen Reminder in der Liste. 2. Du wirst zum Map-Tab weitergeleitet.
--------	---

Erwartetes Resultat	Der ausgewählte Pin wird im Fokus auf der Karte angezeigt.
---------------------	--

ID	ST_05
----	-------

Anforderungen	Als Nutzer möchte ich auf ein Edit-Symbol klicken können, damit ich direkt zur Bearbeitungsansicht gelange und die Erinnerung bei Bedarf ändern oder löschen kann.
---------------	--

Vorbedingungen	Der Tester befindet sich in der geöffneten App auf dem Home Screen.
----------------	---

Ablauf	1. Klicke auf das Edit-Symbol eines erstellten Reminders. 2. Ändere die Position, den Radius, den Titel und die Beschreibung. 3. Klicke auf „Speichern“.
--------	--

Erwartetes Resultat	Auf dem Home Screen wird die aktualisierte Erinnerung angezeigt.
---------------------	--

ID	ST_06
----	-------

Anforderungen	Als Nutzer möchte ich eine Karte sehen, auf der alle Erinnerungen angezeigt werden, damit ich visuell erkenne, wo überall Erinnerungen gesetzt wurden.
---------------	--

Vorbedingungen	Der Tester befindet sich in der geöffneten App auf dem Home Screen.
----------------	---

Ablauf	1. Klicke auf das Map-Icon unten in der Navbar. 2. Zoome heraus, um die gesamte Karte zu betrachten.
--------	---

Erwartetes Resultat	Alle erstellten Pins werden auf der Karte angezeigt.
------------------------	--

ID	ST_07
----	-------

Anforderungen	Als Nutzer möchte ich, dass jede Erinnerung auf einer Karte angezeigt wird, damit ich schnell sehe, welche Erinnerungen an welchen Orten gespeichert sind.
---------------	--

Vorbedingungen	Der Tester befindet sich in der geöffneten App auf dem Home Screen.
----------------	---

Ablauf	1. Klicke auf das Map-Icon unten in der Navbar. 2. Navigiere zu den Orten, an denen Pins gesetzt wurden.
--------	---

Erwartetes Resultat	Alle erstellten Pins werden auf der Karte angezeigt.
------------------------	--

ID	ST_08
----	-------

Anforderungen	Als Nutzer möchte ich, dass meine Erinnerungen lokal gespeichert werden, damit sie auch nach dem Schliessen der App noch vorhanden sind.
---------------	--

Vorbedingungen	Der Tester befindet sich in der geöffneten App auf dem Home Screen.
----------------	---

Ablauf	1. Erstelle einen Reminder. 2. Schliesse die App. 3. Öffne die App erneut.
--------	--

Erwartetes Resultat	Der erstellte Reminder wird weiterhin auf dem Home Screen angezeigt.
------------------------	--

TESTPROTOKOLL

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

Hier ein Beispiel eines Eintrages:

ID	Person / Datum	Erfolgreich	Bemerkungen
ST-01	Jeremy Ritter 04.7.2025	Ja	-
ST_02	Jeremy Ritter 04.7.2025	Ja	-
ST_03	Jeremy Ritter 04.7.2025	Ja	-
ST_04	Jeremy Ritter 04.7.2025	Ja	-
ST_05	Jeremy Ritter 04.7.2025	Ja	-
ST_06	Jeremy Ritter 04.7.2025	Ja	-
ST_07	Jeremy Ritter 04.7.2025	Ja	-
ST_08	Jeremy Ritter 04.7.2025	Ja	-

FAZIT

Hier kommt eure Reflexion zum Projekt.

Was lief gut/schlecht?

- Gut
 - o Umsetzung der App.
 - o Wir konnten alle features implementieren
 - o Wir konnten auch noch weitere features dazutun (cloud storage, verschiedene users, suchfunktion in create und edit, und noch weitere)
- Schlecht:
 - o Probleme mit der Integration der online-Synchronisation, dies ist aber ein Zusatz Feature und wir konnten auch alle Fehler beheben.
 - o

Wie seid ihr mit dem Endergebnis zufrieden?

Ja, Wir sind mit dem Endergebnis sehr zufrieden. Die wichtigsten Funktionen konnten wir wie geplant umsetzen, und wir haben wir sogar noch zusätzliche Features eingebaut. Die App funktioniert stabil und funktioniert wie gewollt.

Was habt ihr gelernt?

Wir haben gelernt, wie man mit react-native eine App entwickeln kann. Wir wissen jetzt, wie man ein React-native Projekt aufsetzen soll und wie man Expo-routing verwendet. Wir haben gelernt, wie man Daten mit AsyncStorage lokal speichern kann, wie man diese daten wieder aufrufen kann und auch wie man diese daten dann löschen kann. Wir haben gelernt, wie man Elemente in einer List ausgeben kann und wie man dann eine «detail-ansicht» zu diesen einzelnen Elementen machen kann, bei uns ist das jetzt mit dem Edit-screen gemacht. Dort wird der Inhalt vom korrektem ReminderElement gezeigt und man kann diese dann bearbeiten.

War alles vorhanden oder was fehlte noch?

Die Exportierte App funktioniert noch nicht ganz da API-key für die Google Maps noch nicht vorhanden ist. Der rest funktioniert alles. Wir konnten alle Funktionen und user-stories die wir umsetzten wollten gut realisieren.