

React Native

Thema: ÜK-332

Dokumentinformationen

Dateiname: uk335-22-BT1_Journal_documentation_Anton_Alessio_Remo.docx
Speicherdatum: 12.05.2022

Autoreninformationen

Autor: Venturini Alessio, Detken Anton, Aeberli Remo

Inhaltsverzeichnis

Änderungsgeschichte	2
1 Einleitung	3
1.1 Sinn und Zweck	3
1.2 Ausgangslage	3
1.3 Ziele	3
1.4 Ausblick	3
1.5 Arbeitsaufteilung	3
2 User Storys	4
2.1 Template	4
2.2 User Storys	4
3 Architektur / Design	7
3.1 Architektur	7
3.2 Verhalten	8
3.3 Design	9
3.4 Endergebnis vergleich	14
4 Technischer Bericht	19
4.1 Testing	19

Änderungsgeschichte

Version	Datum	Autor	Details
1.0	11.05.2022	Alessio Venturini	Dokument erstellt

1 Einleitung

1.1 Sinn und Zweck

Im ÜK 332 geht es darum, dass wir mit React Native in einer Gruppe eine Applikation mit Vorgaben programmieren. Jede Applikation enthält einen Reminder, welcher dem User nach Wahl eine Nachricht schickt, welche den Benutzer daran erinnern sollte, etwas zu erledigen. Unser Projekt war es einen Reminder in einer App zu programmieren, welcher dich daran erinnert, dass du dein Arbeitsjournal abgeben sollst.

1.2 Ausgangslage

Wir erhalten eine Einführung in React Native. Das Projekt wird nach dem Design gestartet. Wir erstellen zuvor ein Design für unsere Applikation. Das Design deckt alle Ziele ab. Wir starten in das Projekt mit leichtem Vorwissen. Das Wissen wird während dem Projekt immer erweitert, mit der Hilfe von Inputs.

1.3 Ziele

Die Ziele an die Applikation.

- Umsetzung des Bewertungsraster
- Die Applikation bietet einen Screen, der das Bearbeiten und Erstellen der Erinnerung ermöglicht.
- Die Erinnerung soll wöchentlich oder monatlich stattfinden können.
- Ein Tag in der Woche soll an der Erinnerung gegeben werden.
- Die Applikation zeigt Standard mässig einen Screen, mit der Erinnerung (wenn eine existiert).
- Die Einstellungen werden persistiert.

1.4 Ausblick

Das Endprodukt sollte alle Ziele beinhalten und unserem Mockup entsprechen.

1.5 Arbeitsaufteilung

In unserem Team gab es eine Arbeitsaufteilung aber trotzdem haben alle immer am gleichen Bereich gearbeitet. Im Projekt haben wir mit dem Design begonnen. Wir haben uns dazu entschieden, dass jeder ein einfaches Mockup macht und wir am Ende entscheiden, welches wir umsetzen. Das Mockup sollte aber noch nicht komplett ausgearbeitet sein. Als wir uns für eines entscheiden haben, haben wir damit begonnen dieses Design zu verbessern und einige Dinge hinzuzufügen. Als wir mit dem Projekt begonnen haben, haben wir zuerst alle zusammen die Struktur unseres Projektes gestaltet und implementiert. Danach haben wir alle angefangen zu coden. Wir haben aber nicht einfach wild drauflos programmiert, sondern wir sind es taktisch angegangen. Wir haben mit den Atomen begonnen, weil es die kleinste Gruppe ist. Jeder hat sich einige Atome ausgesucht und diese implementiert. Das gleiche, haben wir auch mit den Molekülen, Organismen und Pages gemacht.

Am Ende hat jeder seine Sachen verbessert. Anton hat am Ende mehr im Code programmiert und die Fehler verbessert. Remo hat am Anfang auch noch den Code verbessert und am Ende die Präsentation gemacht. Alessio hat am Ende die Dokumentation gemacht.

Schlussendlich war unsere Aufteilung gut. Wir hatten nie arbeitslose Minuten und wir haben uns gegenseitig nie behindert.

2 User Storys

2.1 Template

Prioritäten:

- A. Muss implementiert/vorhanden sein
- B. Sollte implementiert/vorhanden sein
- C. Kann implementiert/vorhanden sein
- D. Freiwillig
- E. Unwichtig

Priorität	Bezeichnung	Grösse
Ich als:	Wer?	
möchte:	Was?	
um:	Warum?	
Die User Story ist fertig, wenn: Akzeptanzkriterien in Form einer Checkliste		

2.2 User Storys

A	Reminder erstellen	5
Ich als:	Reminder Benutzer	
möchte:	Einen neuen Reminder erstellen	
um:	Einen neuen Reminder erstellen, um nicht zu vergessen, dass ich an einem gewissen Tag das Arbeitsjournal abgeben muss.	
Die User Story ist fertig, wenn: <ul style="list-style-type: none"> [1] Eine Page mit einem Create-Button vorhanden ist. [2] Eine Page um die Werte für den Reminder zu setzen. [3] Die Werte für den Reminder gespeichert sind. [4] Mit einem Button die Daten des Reminders zu speichern. [5] Mit einem Button die Erstellung abubrechen. 		

A	Reminder anschauen	5
Ich als:	Reminder Benutzer	
möchte:	Erstellte Reminder anschauen	
um:	Mein erstellten Reminder auf einer Page anschauen (übersichtlich gestaltet).	
Die User Story ist fertig, wenn:		
<div>[1] Eine Page mit dem Reminder.</div> <div>[2] Eine Karte mit den Daten von dem Reminder.</div> <div>[3] Auf der Karte des Reminders, sind zwei Buttons Edit und Delete.</div> <div>[4] Wenn kein Button vorhanden ist, wird eine Benutzer freundliche darauf hingewiesen.</div> <div>[5] Wenn man den Button löscht, wird er nicht mehr angezeigt und der Benutzer wird freundlich darauf hingewiesen.</div> <div>[6] Die Page enthält eine Navbar.</div> <div>[7] Die Page enthält einen Button, womit man neue Reminder erstellen kann.</div> <div>[8] Die Page deaktiviert den Create-button, wenn schon ein Reminder vorhanden ist.</div>		

A	Reminder bearbeiten	5
Ich als:	Reminder Benutzer	
möchte:	Einen schon erstellten Reminder bearbeiten	
um:	Auf der Übersichtsseite kann der Reminder bearbeitet werden. Die neuen Daten werden gespeichert.	
Die User Story ist fertig, wenn:		
<div>[1] Die Karte für den Reminder enthält einen Button, um den Reminder zu editieren.</div> <div>[2] Wenn man den Edit-Button drückt, kommt man auf die Editierpage.</div> <div>[3] Die geänderten Daten werden in dem Storage gespeichert.</div> <div>[4] Die neuen Daten können auch angezeigt werden.</div> <div>[5] Die Editierpage enthält eine Navbar.</div> <div>[6] Die Editierpage enthält ein Button, womit man die neuen Änderungen speichern kann.</div> <div>[7] Die Editierpage enthält einen Button, womit man die den Vorgang abbrechen kann.</div>		

B	Benachrichtigung erhalten	3
Ich als:	Applikation benutzen	
möchte:	Der Reminder sendet eine Nachricht	
um:	Wenn die ausgewählte Zeit des Reminders eintritt, erhält der Benutzer eine Nachricht.	
Die User Story ist fertig, wenn:		
<div>[1] Wenn der Benutzer einen Reminder erstellt hat, bekommt er auch eine Nachricht.</div> <div>[2] Die Nachricht tauch auf dem Smartphone auf.</div> <div>[3] Die Daten des Reminders werden an die Benachrichtigung gesendet.</div> <div>[4] Die Benachrichtigung kann auch in einer Loop gesendet werden.</div> <div>[5] Die Benachrichtigung erhält der Benutzer um die Zeit, die der Benutzer beim Reminder erstellen angegeben hat.</div> <div>[6] Die Benachrichtigung enthält eine Benutzer freundliche Nachricht.</div> <div>[7] Die Nachricht in der Benachrichtigung enthält einen Hinweis, dass er das Arbeitsjournal abgeben muss.</div> <div>[8] Die Benachrichtigung kann an einem gewünschten Tag abgesendet werden.</div>		

B	Reminder löschen	2
Ich als:	Reminder Benutzer	
möchte:	Ein erstellter Reminder wird gelöscht	
um:	Wenn ein Benutzer einen Reminder erstellt, der nicht mehr verwendet wird, hat er die Möglichkeit diesen zu löschen.	
Die User Story ist fertig, wenn:		
<div>[1] Der Benutzer löscht einen erstellten Reminder.</div> <div>[2] Der Reminder wird auf der Page angezeigt.</div> <div>[3] Die Karte des Reminders enthält einen Button, um den Reminder zu löschen.</div> <div>[4] Der Button, um den Reminder zu löschen ist verständlich.</div> <div>[5] Wenn der Reminder gelöscht wird, wird er nicht mehr angezeigt.</div> <div>[6] Wenn der Reminder gelöscht wird, ist er nicht mehr abgespeichert.</div> <div>[7] Wenn kein Reminder mehr vorhanden ist, ist es möglich wieder einen neuen Reminder zu erstellen.</div>		

3 Architektur / Design

3.1 Architektur

Wir haben eine klare Struktur in unserem Projekt. Wir haben von beginn an mit Atomen, Molekülen, Organismen, Pages und Services gearbeitet. Wir haben unsere Elemente des Programmes in die einzelnen Teile aufgeteilt.

3.1.1 Service

In dem Service-Ordner befindet sich nur eine Klasse. Diese Klasse beinhaltet nur Logik und ist nicht dafür zuständig, dass etwas auf dem Bildschirm angezeigt wird. In dem Service-Ordner befindet sich die Klasse, welche die Logik für die Benachrichtigung enthält.

3.1.2 Types

Der nächste Ordner ist der Types-Ordner. In diesem Ordner befinden sich die verschiedenen Typen. Der Reminder wurde erstellt, um alle Daten des Reminders in dem Typen Reminder zu packen und diesen abzuspeichern. Somit wird alles zusammen gespeichert und es wird nichts einzeln abgespeichert

3.1.3 UI

Der UI-Ordner enthält alle Elemente, die für den Benutzer ersichtlich sind. Dieser Ordner wurde erstellt, weil alle Elemente zusammengehören und dies einfacher ersichtlich ist.

3.1.4 Atoms

In dem Atoms-Ordner befinden sich alle kleinen Elemente, die im späteren Verlauf zusammengesetzt werden. Es vereinfacht den ganzen Code und verhindert dadurch Redundanz.

3.1.5 Molecules

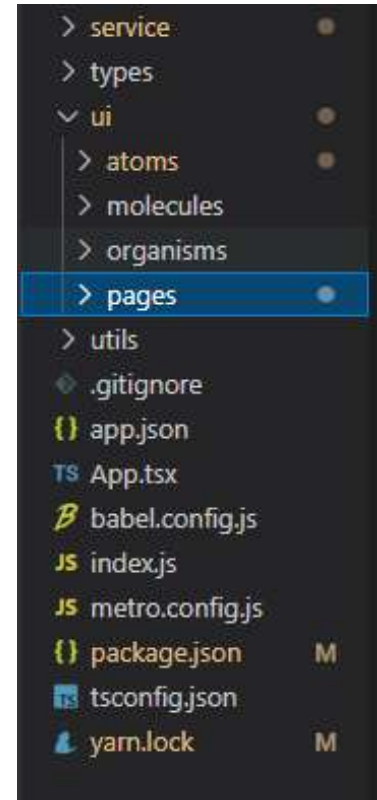
In dem Molecules-Ordner befinden sich die Klassen, welche die Atome zusammensetzt. Es enthält ebenfalls Klassen, welche Elemente beinhalten, welche nicht in Atome aufteilbar sind.

3.1.6 Organisms

Organismen sind zusammengesetzte Moleküle. In den Klassen werden die Moleküle zu einem Organismus zusammengebracht. Diese Organismen werden schlussendlich für den Benutzer sichtbar. Im Organismus ist alles perfekt und kann nur noch auf die Page gebracht werden.

3.1.7 Pages

Pages beinhaltet die verschiedenen Ansichten, die der Benutzer sehen kann. Auf den Pages werden die Organismen angezeigt.



3.2 Verhalten

Unsere App ist demnach nach dieser Struktur auch im Code aufgebaut. Der User sieht schlussendlich nur Pages welche Organismen enthalten. Die anderen Komponenten werden nicht spezifisch im Code der Page aufgerufen. Wir arbeiten mit mehreren Pages. Wenn der User ein Reminder hat, ist es die gleiche Page, wie wenn er keinen Reminder hat. Wenn einer verwendet wird, wird er zu der Page hinzugefügt aber es existiert keine einzelne Page dafür.

Bei der Editpage ist es dasselbe Prinzip. Wenn ein Button nicht verfügbar sein soll, wird keine neue Page mit dem deaktivierten Button angezeigt, sondern die gleiche Page, nur das die Buttons auf der Page deaktiviert sind.

3.3 Figma

Link: <https://www.figma.com/file/p8mo8NH4AVk8JC6JSKW0Ho/Arbeitsjournal-App?node-id=204%3A1008>

3.4 Farbkonzept

Alle Farben in unserem Design:

- Hauptfarbe: #6200EE
- Hintergrund: #FFFFFF
- FAB: #03DAC5
- Button: #01A299
- Chips enabled: #EEEEEE
- Chips active: #6200EE
- Schrift: #000000

3.5 Design

Wir wollten bei der Homepage nicht viel Spezielles. Die Homepage sollte dem Benutzer seinen Reminder anzeigen. Wenn kein Reminder vorhanden ist, sollte der Benutzer auch eine freundliche Nachricht sehen, welche ihn darauf hinweist, dass er momentan keinen Reminder hat. Auf dieser Seite ist es demnach auch möglich einen neuen Reminder zu erstellen. Die Daten für den neuen Reminder werden aber auf einer anderen Page aufgenommen.

3.5.1 Homeseite (1)

Auf dem Bild rechts sieht man die erste Seite, welche man als Benutzer direkt sieht, wenn man die App (Applikation) startet. Ganz oben befindet sich eine einfache Navbar, welche die App etwas aufgeteilter aussehen lässt. In der Navbar befindet sich ebenfalls noch einen Titel welche dem Benutzer sagt, auf welcher Seite er sich gerade befindet.

Unter der Navbar kommt die Benutzer freundliche Nachricht, welche den Benutzer darauf hinweist, dass er noch keinen Reminder erstellt hat. Wir haben für diesen Text und allgemein für die Applikation die Sprache Englisch gewählt, damit ist es einfacher, wenn man später damit international gehen wollen würde.

Am Ende der Seite, sieht man einen grünen Button. Dieser Button hat die Farbe Grün, um den Benutzer darauf hinzuweisen, dass man sehr einfach unten rechts einen neuen Reminder hinzufügen kann. Der Button ist die einzige Möglichkeit in dieser Lage eine Interaktion mit dem Programm zu machen. Das ist ebenfalls ein Grund, warum dieser sich von den anderen Elementen der App abhebt.



3.5.2 Homeseite (2)

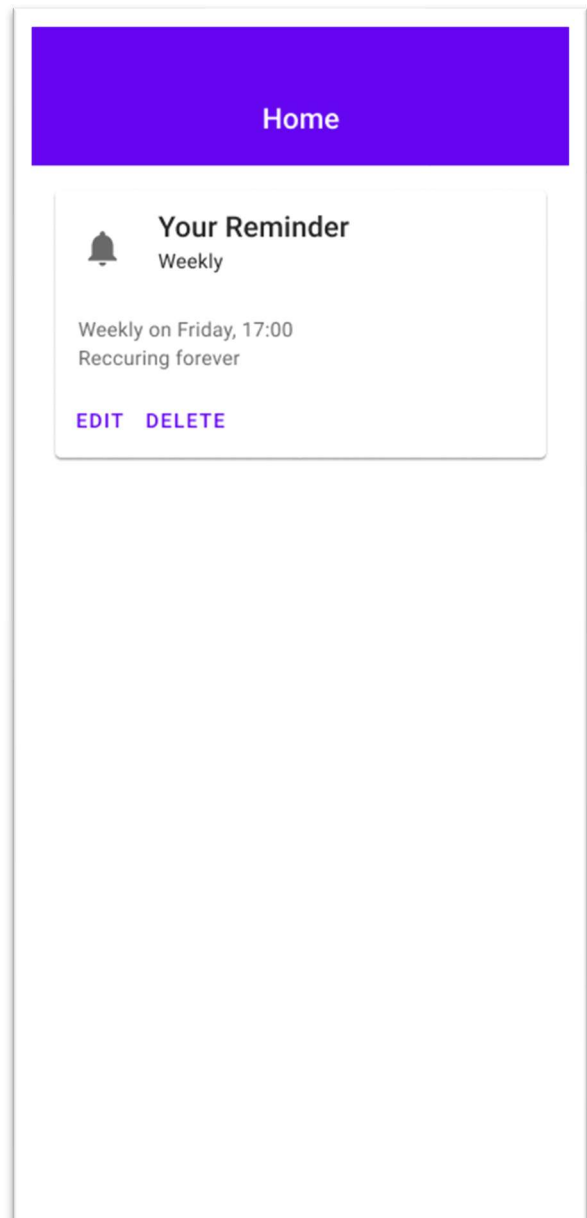
Die zweite Seite kommt dann in den Einsatz, wenn der Benutzer einen neuen oder schon einen Reminder hat.

Ganz oben befindet sich wieder die Navbar, welche auf allen Seiten einheitlich ist. Wie man im Titel in der Navbar auch schon lesen kann, befindet man sich nicht auf einer neuen Seite, sondern immer noch auf der Homepage.

Darunter sieht man, dass die Nachricht nicht mehr vorhanden ist und stattdessen eine Karte aufgetaucht ist, mit den Daten des Reminders. Das Icon sollte dem Benutzer recht schnell sagen, dass dies ein Reminder ist ohne, dass er den Text lesen muss. Trotzdem befindet sich rechts davon einen Titel, welcher vom Benutzer gesetzt werden kann. Darunter stehen noch weitere Informationen zu dem Reminder, welche aber nicht sehr relevant für den Benutzer sind, da er sie selber gesetzt hat und diese schon weiss. Der Titel sticht sehr heraus, damit der Benutzer schnell weiss welcher seiner Reminder aktuell benutzt wird.

Am Ende der Karte befinden sich zwei neue Buttons. Diese Buttons sind dafür da, dass man den aktuellen Reminder bearbeiten kann oder auch löschen kann. Wenn man den Edit-Button drückt, kommt man auf eine andere Seite, welche dem Benutzer ermöglicht, seinen Reminder zu bearbeiten. Der Delete-Button ermöglicht es dem Benutzer seinen aktuellen Reminder zu löschen, wenn der nicht mehr verwendet wird oder der Benutzer einen neuen Reminder erstellen möchte.

Am Ende der Seite befindet sich jetzt kein Create-Button mehr. Der Button ist verschwunden, weil schon ein Reminder vorhanden ist. Der Button verschwindet, nachdem man einen Reminder erstellt und erscheint wieder, wenn kein Reminder mehr vorhanden ist.



3.5.3 Reminderseite (1)

Diese Seite bekommt der Benutzer zu Gesicht, wenn er einen neuen Reminder auf der Homeseite erstellen möchte oder er mit dem aktuellen Reminder nicht zufrieden ist und ihn bearbeiten möchte.

Ganz oben, befindet sich wieder die Standard Navbar. Der Unterschied zu der Homeseite ist aber, dass der Titel anders ist, damit der Benutzer nun weiss, dass er sich auf einer neuen Seite befindet.

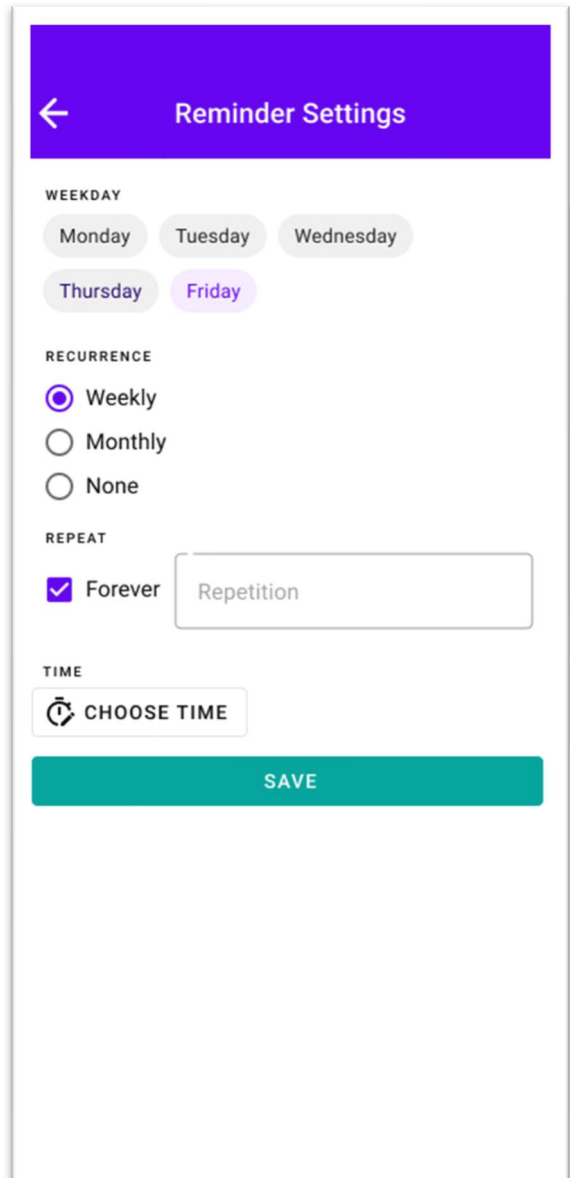
Zuoberst befinden sich die Radio-Buttons. Mit dem Titel oberhalb der Buttons sieht man recht schnell, dass es sich um die Wochentage handelt. Es ist immer nur möglich, einen der Buttons anzuwählen. Der Benutzer kann die Benachrichtigung immer nur an einem Tag erhalten. Der aktuelle angewählte Button wird mit der Farbe Violett angezeigt. Der Button ist Violett, weil er nicht das wichtigste auf der Seite ist, aber ein wichtiges Element ist. Wir haben die Farbe Violett für den Button gewählt, weil es die Standardfarbe unserer App ist und es genug ersichtlich ist, dass dieser Button aktuell angewählt ist.

Darunter befinden sich wieder Radio-Buttons. Oberhalb befindet sich wieder einen kleinen Titel, welcher dem Benutzer direkt sagt, um was es geht. Die Farbe ist wieder Violett, weil es wieder nicht das wichtigste ist und wir mit der Standardfarbe arbeiten. Es ist nur möglich eines der Felder anzuwählen. Es können nicht alle angewählt sein.

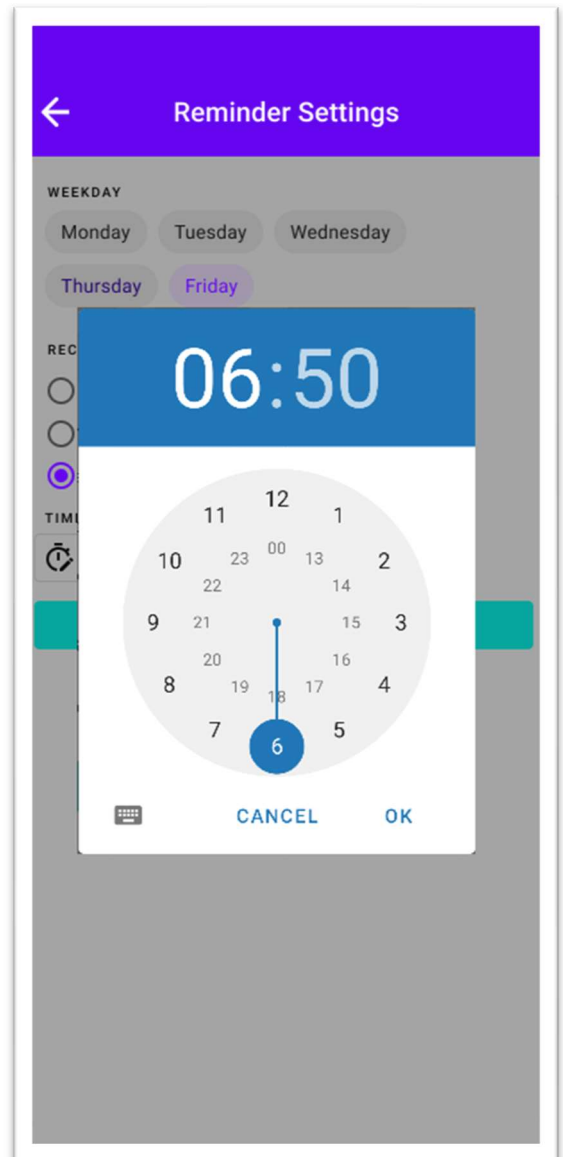
Es befindet sich einen Button darunter, welcher möglich ist anzuwählen, wenn man möchte, dass die Benachrichtigung immer um diese Zeit kommt. Wenn der Button nicht angewählt ist, ist es aber auch möglich, eine Zahl in das Inputfeld zu schreiben. Jedoch ist nicht beides möglich.

Am Ende der Seite befinden sich ein Button. Laut der Farbe ist es einfach zu sehen, dass der Button das wichtigste auf der Seite ist. Wenn dieser Button benutzt wird, werden alle Daten, die der Benutzer angegeben hat, gespeichert.

Oben links gibt es noch einen Pfeil, welcher eigentlich das Gegenstück von dem grünen Save-Button ist. Wenn dieser Button gedrückt wird, wird man wieder auf die Page davor geschickt. Die Daten gehen dabei verloren und der Benutzer kann es nicht mehr wiederholen.



Das letzte Feld, welches Daten entgegennimmt, ist das Time Element. Es ist möglich, den Button anzuklicken und danach, kann man an einem Rad, die Zeit angeben. Rechts davon kann der Benutzer die Angegebene Zeit sehen, falls er vergessen hat, was er angegeben hat.



3.5.4 Reminderseite (2)

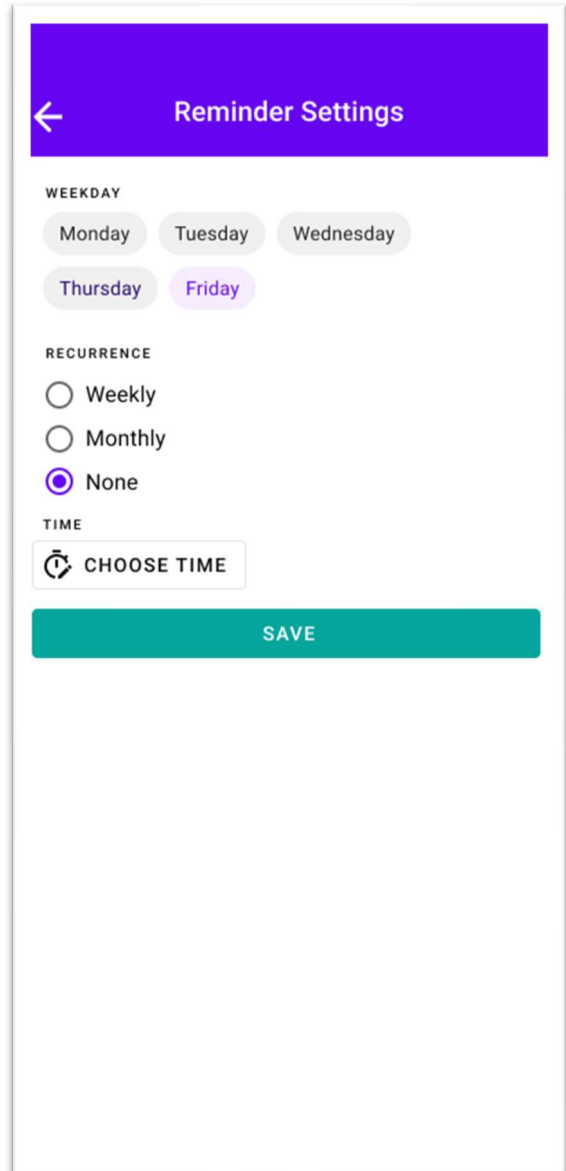
Diese Seite sieht der Benutzer, wenn er einen Reminder bearbeiten oder einen neuen erstellen möchte. Wenn er bei der Daten Angaben einige Dinge anwählt, ändert sich das Layout und es ist möglich, dass sich einige Buttons nicht mehr benutzen lassen, weil sie deaktiviert wurden.

Zuoberst befindet sich wieder die einfache Navbar, mit dem gleichen Titel wie die andere Seite, weil es keine neue Seite ist, sondern die aktuelle angepasst wird.


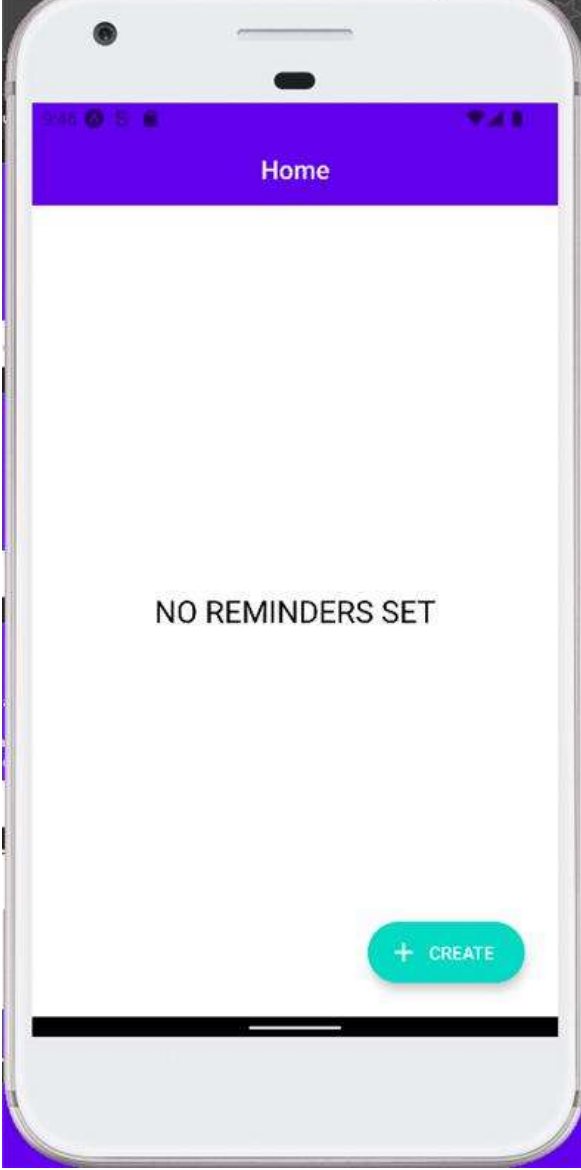
Bei der Weekday angabe ändert sich nicht viel. Wenn man einen anderen Tag anwählt, wird dieser einfach angewählt und der alte abgewählt. Es wird aber nicht am Design geändert. Es ist einfach nicht möglich mehrere Buttons gleichzeitig anzuwählen.

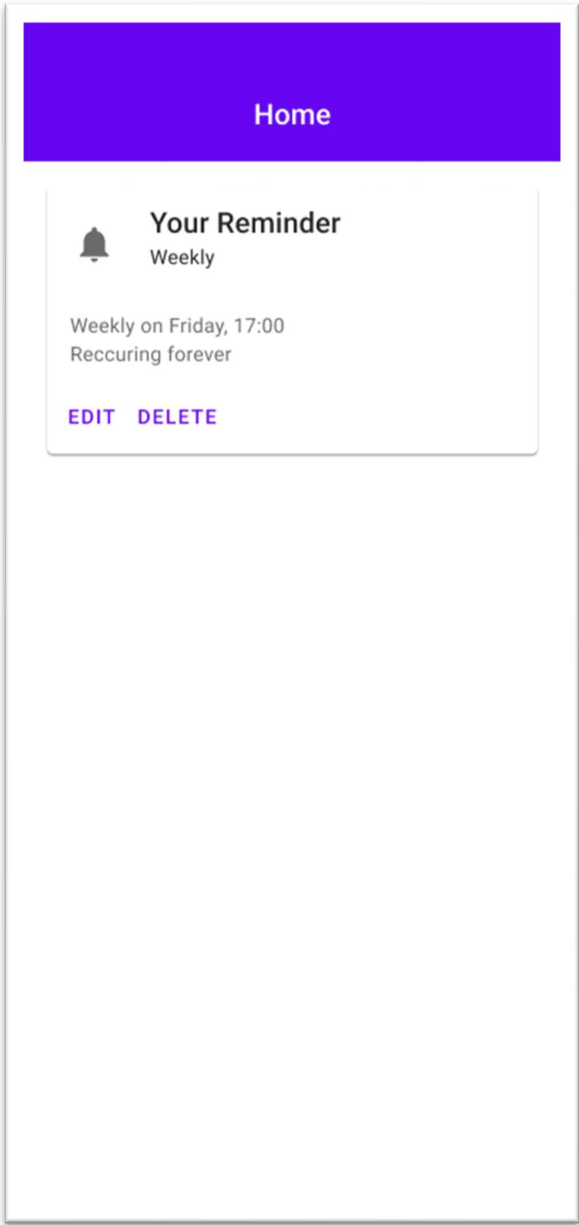
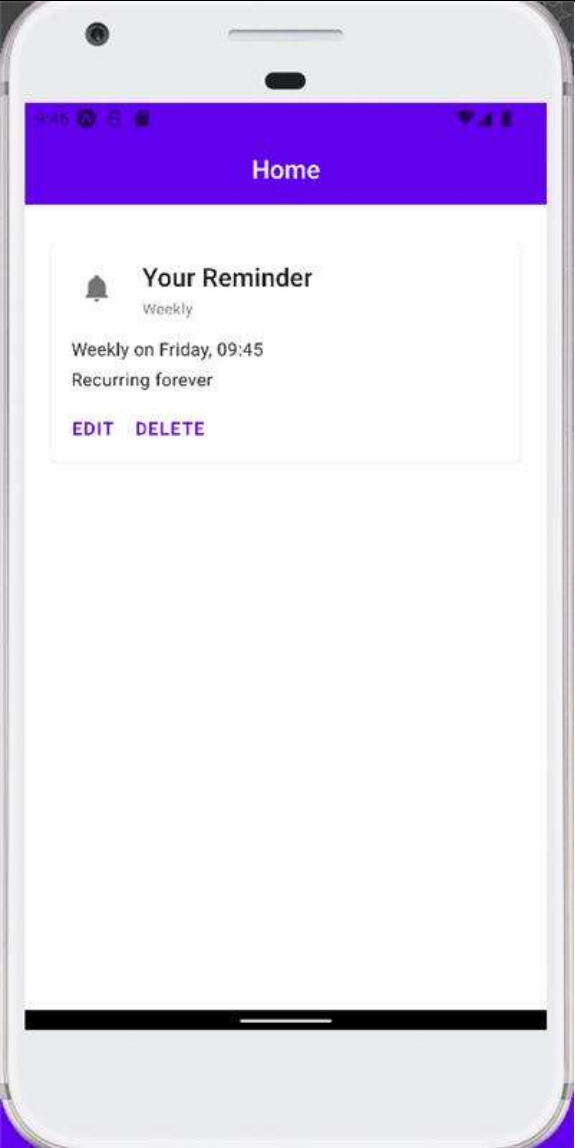
Bei der Recurrence ändert sich aber etwas, wenn man einen bestimmten Button anwählt. Wenn der Benutzer den None-Button anwählt, ist es nicht mehr möglich Repeat anzugeben. Das liegt daran, dass es keinen Sinn macht, wenn der benutzer angeben kann, wie viel mal er ausgeführt werden soll, wenn oben angegeben wird, dass er nur einmal ausgeführt wird. Mit None sagt der Benutzer, dass er die Benachrichtigung nur einmal an diesem Tag und um diese Zeit erhalten möchte. Dabei wird ebenfalls auch der Forever-Button verschwunden.

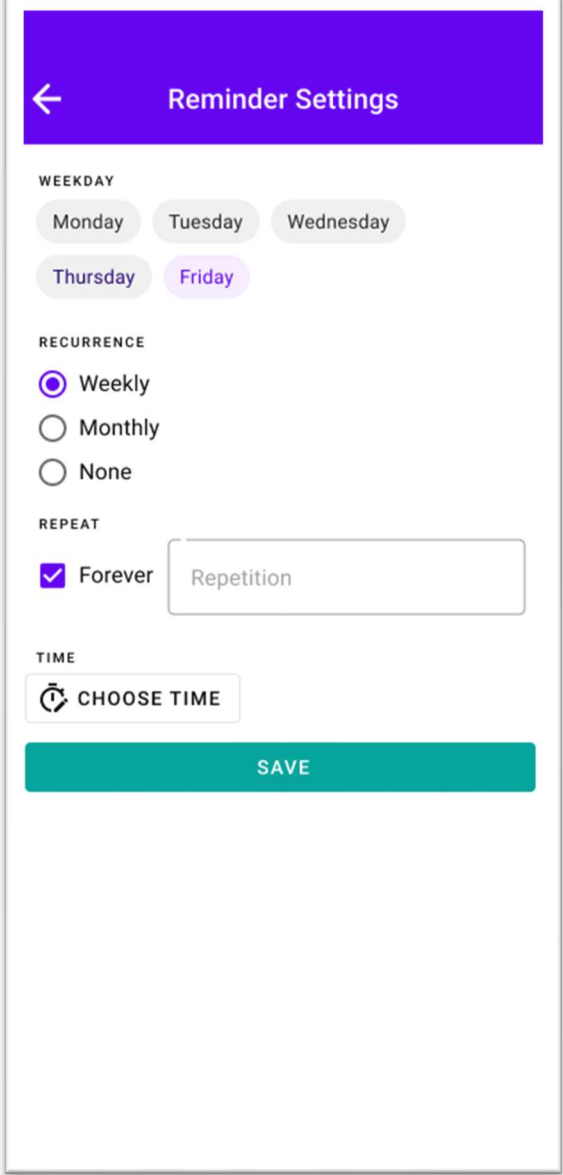
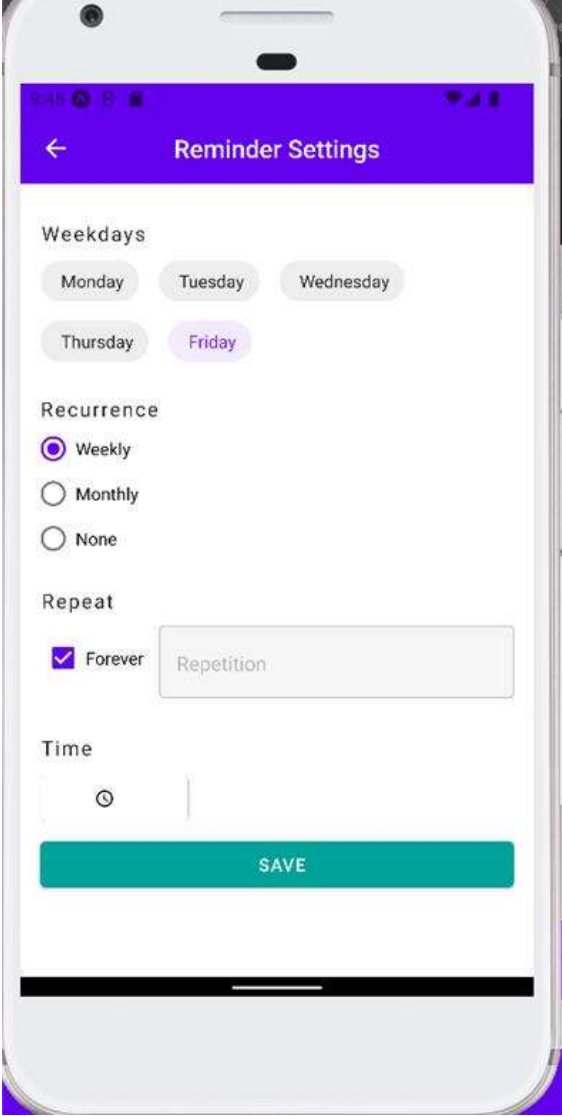
Sonst ändern isch keine Dinge an dieser Seite. Der Benutzer kann nur, mit einigen Wahlen einige Buttons deaktivieren. Wenn er das Ganze jedoch doch nicht mehr will, kann er den Button einfach abwählen.

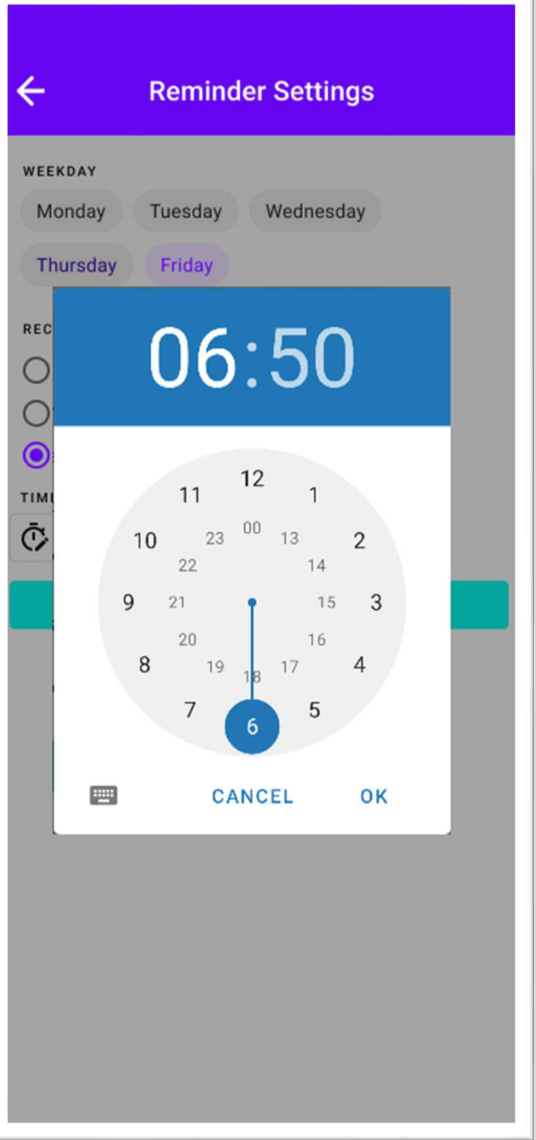
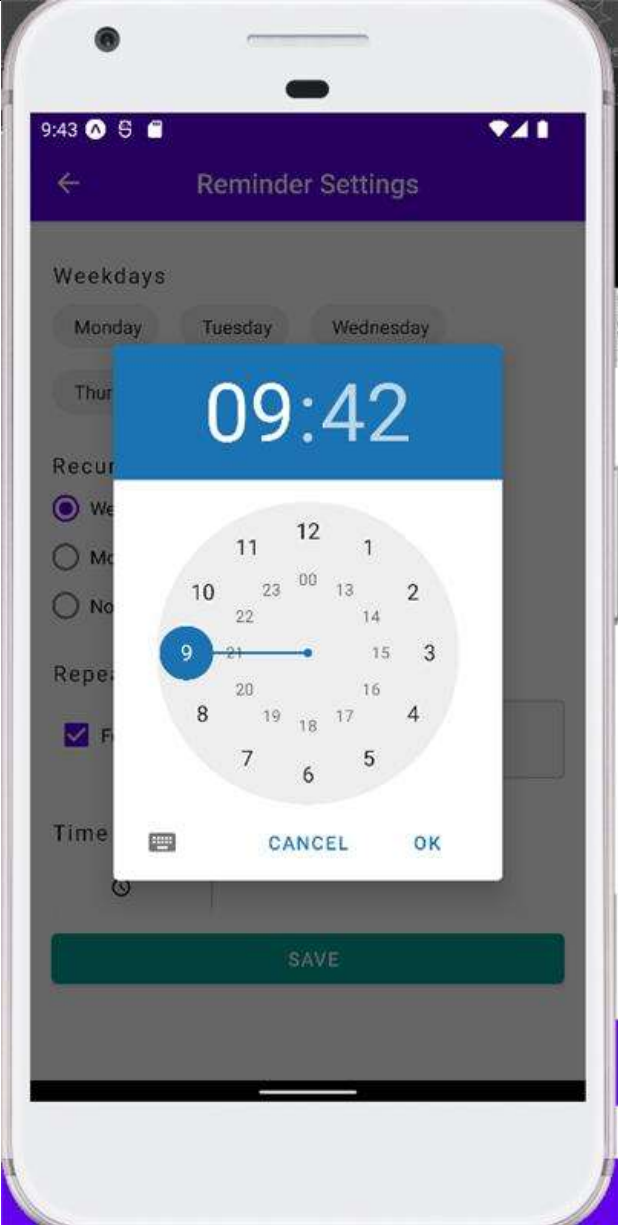


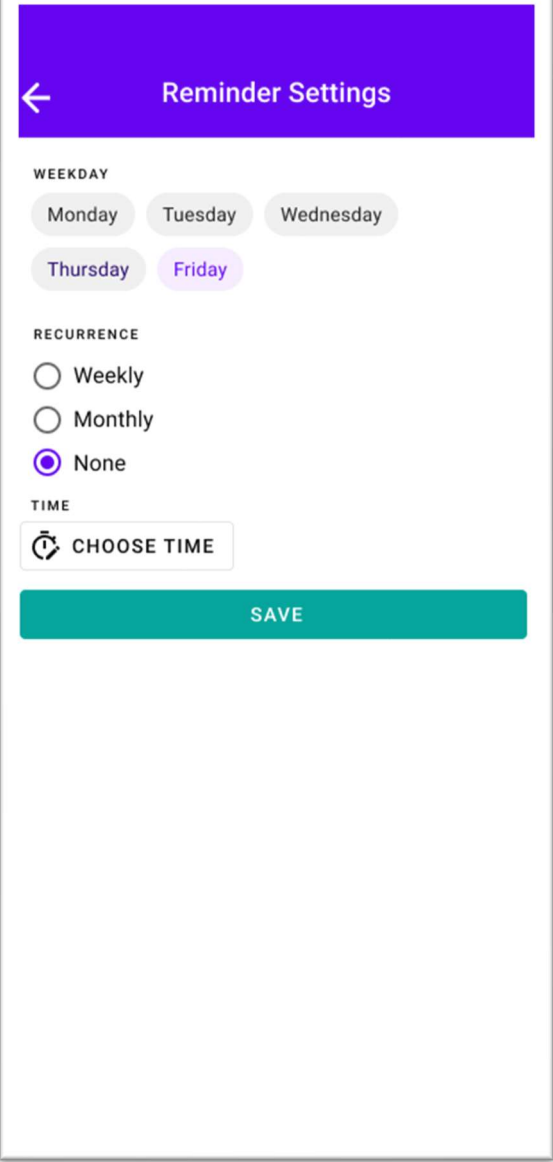
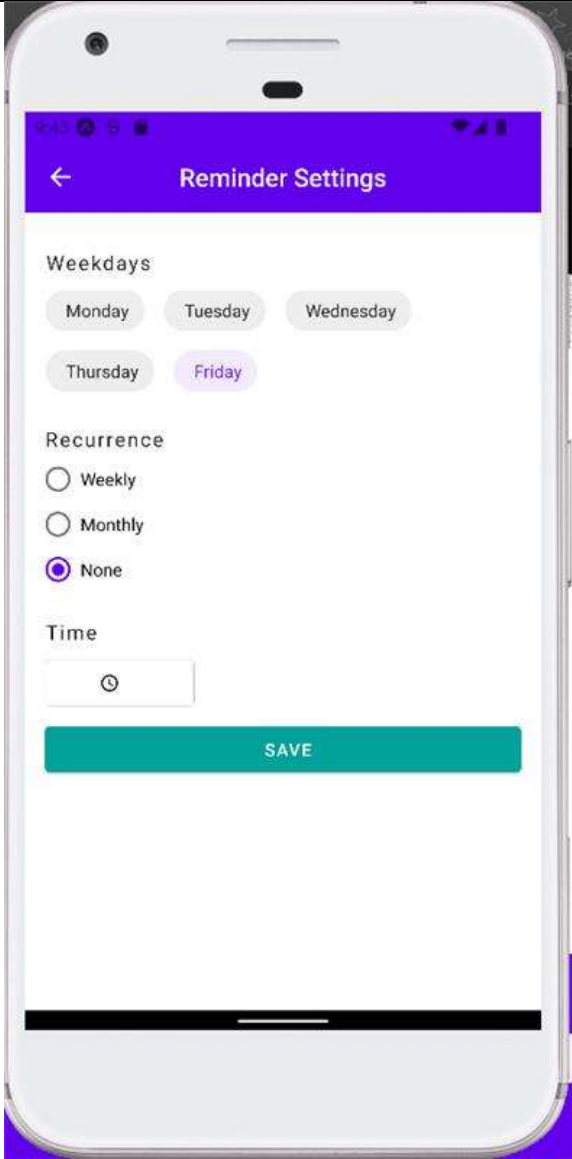
3.6 Endergebnis vergleich

Mockup (Figma)	Implementation (React Native)
	
<p>Bei der ersten Homepage haben wir keine Unterschiede, was unser UI angeht. Man sieht auf dem rechten sieht man unten eine kleine Navbar. Diese kommt aber nicht vom UI, sondern vom Smartphone.</p>	

Mockup (Figma)	Implementation (React Native)
 <p>The Figma mockup shows a mobile app interface. At the top is a purple header bar with the text 'Home' in white. Below this is a white card with a bell icon and the title 'Your Reminder'. Under the title, it says 'Weekly' in a smaller font. The main text of the card reads 'Weekly on Friday, 17:00' and 'Reccuring forever' (note the typo). At the bottom of the card are two links: 'EDIT' and 'DELETE' in purple.</p>	 <p>The React Native implementation shows the same app interface on a mobile device. It features a purple header bar with 'Home' in white. Below is a white card with a bell icon, the title 'Your Reminder', and the text 'Weekly'. The main text says 'Weekly on Friday, 09:45' and 'Recurring forever'. At the bottom of the card are 'EDIT' and 'DELETE' links in purple. The device's status bar at the top shows the time as 9:45 and various icons.</p>
Auf der zweiten Homepage sieht man ebenfalls keinen Unterschied vom UI und dem Figma.	

Mockup (Figma)	Implementation (React Native)
	
<p>Bei der ersten Reminderpage sieht man einen kleinen Unterschied. Wenn man sich den Timer anschaut, merkt man das die Uhr nicht bei beiden Identisch ist. Wir haben ebenfalls den Text abgeändert. Nun wird die Zeit angezeigt, welche man angegeben hat.</p>	

Mockup (Figma)	Implementation (React Native)
	
Wenn man auf den Timer klickt, kommt ein kleines Fenster hervor. Wie man sieht, gibt es hier keinen Unterschied vom Figma zum UI.	

Mockup (Figma)	Implementation (React Native)
	
<p>Auf der zweiten Reminderpage sehen wir wieder den gleichen kleinen Unterschied. Das Uhr Icon ist nicht bei beiden Timer identisch. Es wurde ebenfalls wieder der Text im Timer geändert. Der Text zeigt nun an, was der Benutzer angegeben hat.</p>	

4 Technischer Bericht

4.1 Testing

4.1.1 Template

Testprotokoll	
TestCaseid	Id des Testes
Component	Name des Component
Priority	Die Priorität des Testes
Description	Was wird getestet
Pre-requisites	Voraussetzungen die getroffen werden müssen
Test Steps	Ablauf des Testes
Expected result	Was wird erwartet
Actual Result	Was ergab der Test wirklich
Status	Status des Testes (Passed, Failed)
Test Executed By	Wer hat es getestet

4.1.2 Positive Tests

Testprotokoll	
TestCaseid	CreateReminder_1
Component	ButtonElement
Priority	A
Description	Wenn ein Benutzer noch keinen Reminder hat, sollte er auf der Homepage einen grünen Button sehen, welcher ihn auf die Reminderpage bringt.
Pre-requisites	Kein Reminder ist vorhanden.
Test Steps	<ol style="list-style-type: none"> 1. Schauen, ob ein Reminder vorhanden ist. Wenn einer vorhanden ist soll er gelöscht werden. 2. Den Create-Button drücken. 3. Auf die Reminderpage kommen.

Expected result	Der Benutzer wird umgeleitet auf eine andere Page. Auf dieser Page sollte es möglich sein, einen Reminder zu erstellen.
Actual Result	Der Benutzer gelangt auf die Reminderpage und kann einen neuen Reminder erstellen.
Status	Pass
Test Executed By	Alessio

Testprotokoll	
TestCaseid	DeleteReminder_1
Component	ReminderCard
Priority	A
Description	Wenn ein benutzer einen Reminder erstellt hat, diesen aber nicht mehr möchte, sollte er diesen löschen könne. Nach dem er gelöscht wurde sollte er nicht mehr angezeigt werden.
Pre-requisites	Ein Reminder ist vorhanden.
Test Steps	<ol style="list-style-type: none"> 1. Schauen, ob ein Reminder auf der Page angezeigt wird. 2. Den Delete-Button auf der Card drücken. 3. Die Card sollte verschwinden. 4. Neuer Reminder erstellen sollte möglich sein
Expected result	Der Reminder verschwindet von der Page und wird gelöscht. Dem Benutzer ist es wieder erlaubt einen neuen zu erstellen.
Actual Result	Wenn man den Button anklickt, verschwindet die Card und es wird kein Reminder mehr angezeigt. Der Reminder wird gelöscht. Der Create-Button erscheint wieder und die Nachricht.
Status	Pass
Test Executed By	Alessio

Testprotokoll	
TestCaseid	EditReminder_1
Component	ReminderCard
Priority	A
Description	Wenn an dem Benutzer der erstellte Reminder nicht mehr gefällt, sollte es ihm gestattet sein, seinen aktuelle Reminder zu bearbeiten.
Pre-requisites	Ein Reminder ist vorhanden.
Test Steps	<ol style="list-style-type: none"> 1. Schauen, ob ein Reminder vorhanden ist. Wenn kein Reminder vorhanden ist soll kurz einer erstellt werden. 2. Auf der Card den Edit-Button anklicken. 3. Der Benutzer gelangt auf eine neue Page. 4. Auf der neuen Page sieht er seine Daten. 5. Die Daten können bearbeitet werden.
Expected result	Der Benutzer kommt auf eine neue Page und kann den reminder bearbeiten.
Actual Result	Der Benutzer gelangt auf die Reminderpage und kann einen neuen Reminder bearbeiten.
Status	Pass
Test Executed By	Alessio

Testprotokoll	
TestCaseid	DatenPersistierung_1
Component	AsyncStorage
Priority	B
Description	Wenn ein Benutzer ein Reminder erstellt hat, sollte er auch abgespeichert werden.
Pre-requisites	Kein Reminder vorhanden.
Test Steps	<ol style="list-style-type: none"> 1. Schauen, ob kein Reminder vorhanden ist. 2. Den Create-Button drücken. 3. Auf der Reminderpage einen neuen Reminder erstellen. 4. Schauen, ob der neue Reminder angezeigt wird.

	5. Die Daten im Reminder mit den Daten davor vergleichen.
Expected result	Ein neuer Reminder wird erstellt und angezeigt. Die Daten des Reminders werden abgespeichert und angezeigt.
Actual Result	Der Reminder wird mit den Daten angezeigt. Die Daten werden gespeichert.
Status	Pass
Test Executed By	Alessio

Testprotokoll	
TestCaseid	DatenPersistierung_2
Component	AsyncStorage
Priority	B
Description	Wenn ein Benutzer seinen Reminder bearbeiten möchte, sollten die gespeicherten Daten des Reminders angezeigt werden.
Pre-requisites	Ein vorhandener Reminder.
Test Steps	<ol style="list-style-type: none"> 1. Schauen, ob ein Reminder vorhanden ist. 2. Auf den Edit-Button in der Card drücken. 3. Auf die Reminderpage gelangen. 4. Die Daten von der Erstellung sehen.
Expected result	Die Daten von der Erstellung wurden gespeichert und werden beim Öffnen des Reminder angezeigt.
Actual Result	Auf der Reminderpage werden die Daten von der Erstellung angezeigt.
Status	Pass
Test Executed By	Alessio

Testprotokoll	
TestCaseid	RadioButtonFunction_1
Component	RecurrenceGroupe
Priority	B
Description	Wenn ein Benutzer bei der Erstellung oder bei der Bearbeitung den Radio-Button None anwählt, wird der Forever und Repeat deaktiviert.
Pre-requisites	Ein Reminder vorhanden oder nicht.
Test Steps	<ol style="list-style-type: none"> 1. Einen neuen Reminder erstellen. 2. Auf den Radio-Button None klicken 3. Schauen, ob die Buttons Forever und das Input Feld Repeat verschwinden
Expected result	Wenn der Benutzer der Recurrence den Button None anwählt, verschwindet der Forever-button und das Input Feld Repeat.
Actual Result	Der Forever-Button und das Input Feld Repeat sind verschwunden und nicht mehr benutzbar.
Status	Pass
Test Executed By	Alessio

Testprotokoll	
TestCaseid	InputFeldfunction_1
Component	RepeatInputGroupue
Priority	B
Description	Wenn ein Benutzer einen neuen Reminder erstellt oder einen existierenden Reminder bearbeitet und er den Forever-Button anwählt, sollte das Repeat Input Feld deaktiviert sein.
Pre-requisites	Ein Reminder vorhanden oder nicht.
Test Steps	<ol style="list-style-type: none"> 1. Auf die Reminderpage gehen, um einen neuen Reminder zu erstellen oder einen existierenden zu bearbeiten. 2. Zu dem Forever-Button gehen und diesen auswählen. 3. Schauen, ob das Repeat Input Feld grau ist. 4. Prüfen, ob das Input Feld noch funktioniert. 5. Den Forever-Button abwählen.

	6. Schauen, ob das Input Feld wieder brauchbar ist.
Expected result	Das Repeat Input Feld ist grau und kann keinen Input mehr erhalten. Das Input Feld ist nicht mehr brauchbar, bis er wieder ausgewählt wird.
Actual Result	Das Input Feld wird unbrauchbar, wenn man den Forever-Button drückt. Wenn der Button wieder ausgewählt wird, wird das Input Feld wieder brauchbar.
Status	Pass
Test Executed By	Alessio

4.1.3 Negative Tests

Testprotokoll	
TestCaseid	RepeatGroupe_1 (Nur im Web möglich auf dem Handy nicht)
Component	RepeatInputGroupue
Priority	C
Description	Es sollte dem Benutzer nicht erlaubt sein, Buchstaben in das Repeat Input Feld einzugeben.
Pre-requisites	Leeres Input Feld
Test Steps	<ol style="list-style-type: none"> 1. Auf die Reminderpage gehen, um einen neuen Reminder zu erstellen oder einen existierenden zu bearbeiten. 2. Schauen, dass der Forever-Button nicht angewählt ist. 3. In das Input Feld eine Text schreiben, der nur Buchstaben enthält. 4. Den Reminder speichern. 5. Nachschauen, ob die Daten gespeichert wurden.
Expected result	Der Text sollte ihn dem Reminder nicht gespeichert werden. Wenn man die Daten des Reminders anschaut, sollte der Repeat keinen Wert haben.
Actual Result	Die Daten mit dem Text werden nicht abgespeichert. Wenn man den Reminder anschaut, befinden sich keine Daten im Repeat.
Status	Pass
Test Executed By	Alessio

Testprotokoll	
TestCaseid	RepeatGroupe_2
Component	RepeatInputGroupue
Priority	C
Description	Ein Benutzer versucht eine Zahl in das Repeat Input Feld zu schreiben und danach den Forever-Button zu drücken. Der Text befindet sich im Input Feld, darf aber nicht abgespeichert werden.
Pre-requisites	Leeres Input Feld
Test Steps	<ol style="list-style-type: none"> 1. Auf die Reminderpage gehen, um einen neuen Reminder zu erstellen oder einen existierenden zu bearbeiten.

	<ol style="list-style-type: none">2. Schauen, dass der Forever-Button nicht angewählt ist.3. In das Input Feld eine Zahl eingeben.4. Den Forever-Button drücken.5. Schauen, ob die Zahl grau im Input Feld steht.6. Die Erstellung/Änderung speichern.7. Im Reminder nachschauen, ob die Daten gespeichert wurden.
Expected result	Die zahl im Input Feld dürfen nicht gespeichert werden. Wenn man den Reminder öffnet, sollte nur der Forever-button angewählt sein.
Actual Result	Die Daten aus dem Input Feld sind nicht gespeichert worden. Wenn man den Reminder anschaut, sieht man den Forever-Button angewählt.
Status	Pass
Test Executed By	Alessio