

# RemindMe!

---

## Reminder app

A React-native app for making reminders,  
using react-native-paper and expo

00

00

## ÜK-335 Project

A Project for ÜK-355

17

00

## Done by

Amanda Walser  
Danny Nguyen  
Cedric Zollinger

23

59



# Inhaltsverzeichnis

|   |           |
|---|-----------|
| <b>Anforderungen.....</b>                     | <b>2</b>  |
| Personas.....                                 | 2         |
| Lernende/r im Basislehrjahr.....              | 2         |
| Lernende/r im Praxislehrjahr.....             | 2         |
| Ausbildner.....                               | 2         |
| User Stories.....                             | 3         |
| User Story Basislehrjahr Lernender.....       | 3         |
| User Story Praxislehrjahr Lernender.....      | 4         |
| User Story Berufsbildner.....                 | 5         |
| Funktionale Anforderungen.....                | 7         |
| Nicht-funktionale Anforderungen.....          | 7         |
| <b>Architektur und Design.....</b>            | <b>8</b>  |
| Optische Anforderungen an das Produkt.....    | 8         |
| Homepage.....                                 | 8         |
| Erstellseite.....                             | 9         |
| Bearbeitungsseite.....                        | 10        |
| Funktionale Anforderungen an das Produkt..... | 10        |
| Anforderungen an die Struktur.....            | 10        |
| Anforderungen an das Verhalten.....           | 11        |
| Vergleich Mockup und GUI.....                 | 12        |
| Homepage.....                                 | 12        |
| Erstellseite.....                             | 13        |
| Bearbeitungsseite.....                        | 14        |
| <b>Technischer Bericht.....</b>               | <b>15</b> |
| Testing.....                                  | 15        |
| Ausblick.....                                 | 20        |
| <b>Auswertung.....</b>                        | <b>20</b> |
| Erreichung der Ziele.....                     | 20        |
| Ziele der Struktur.....                       | 20        |
| Ziele des Verhaltens.....                     | 20        |

# Anforderungen

## Personas

### Lernende/r im Basislehrjahr

Alter: 15-19

Technische Affinität: Durchschnittlich

Probleme: Vergesslich und unsicher

Bedürfnisse: Eine App, welche einen Reminder speichert, den Badge morgens nicht zu vergessen

### Lernende/r im Praxislehrjahr

Alter: 16-20

Technische Affinität: Überdurchschnittlich

Probleme: Gestresst, keine Terminplanung / Zeiteinteilung

Bedürfnisse: Eine App, welche Reminder verschickt, das Lernjournal zu erledigen

### Ausbildner

Alter: 24-65

Technische Affinität: Erfahren

Probleme: Kein Gefühl für Zeit, gestresst

Bedürfnisse: Eine App, welche Reminder verschickt, Pausen einzulegen oder Feierabend zu machen. Ausserdem soll es möglich sein, verschiedene Erinnerungen an Tasks zu erstellen.

## User Stories

### User Story Basislehrjahr Lernender

|  |   |          |
|--|---|----------|
| <b>A</b>   | <b>BLJ Badge mitnehmen Reminder erstellen</b> | <b>8</b> |
| <p>Ich als: <b>Basislehrjahr Lernender,</b></p> <p>möchte: <b>jeden Mittwoch, Donnerstag und Freitag um 06:00 Uhr einen Reminder bekommen,</b></p> <p>um: <b>eine Erinnerung zu bekommen, damit ich meinen Badge einpacke, da ich diesen für die elektronische Eintrittskontrolle bei der Noser Young benötige.</b></p>  |   |          |
| <p>Die User Story ist fertig, wenn:</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Der User die App öffnen kann.</li> <li><input type="checkbox"/> Der User dem Reminder den Titel "Badge mitnehmen" geben kann.</li> <li><input type="checkbox"/> Der User die Zeit des Reminders auf 06:00 Uhr stellen kann.</li> <li><input type="checkbox"/> Der User Auswählen kann, dass der Reminder sich wöchentlich wiederholen soll.</li> <li><input type="checkbox"/> Der User beim Erstellen des Reminders die Tage Mittwoch, Donnerstag und Freitag auswählen kann.</li> <li><input type="checkbox"/> Der User den Reminder speichern kann.</li> <li><input type="checkbox"/> Der User am Mittwoch, Donnerstag und Freitag um 06:00 Uhr einen Reminder mit dem Titel "Badge mitnehmen" bekommt.</li> </ul> |   |          |
| <b>B</b>   | <b>BLJ Badge mitnehmen Reminder löschen</b>   | <b>2</b> |
| <p>Ich als: <b>Basislehrjahr Lernender,</b></p> <p>möchte: <b>den Reminder den ich jeden Mittwoch, Donnerstag und Freitag um 06:00 Uhr bekomme löschen,</b></p> <p>um: <b>keine Erinnerung für das Einpacken des Badges mehr zu erhalten, da ich diesen in meiner neuen Firma nicht mehr brauche.</b></p>  |   |          |
| <p>Die User Story ist fertig, wenn:</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Der User die App öffnen kann.</li> <li><input type="checkbox"/> Der User die Reminder mit dem Titel "Badge mitnehmen" finden kann.</li> <li><input type="checkbox"/> Der User die Bearbeitungsansicht dieses Reminders finden und öffnen kann.</li> <li><input type="checkbox"/> Der User den Reminder löschen kann.</li> <li><input type="checkbox"/> Der Reminder gelöscht wird und nicht mehr in der Reminder-Übersicht zu finden ist. Zudem soll der User keine Reminder-Notification mehr bekommen.</li> </ul>  |   |          |

## User Story Praxislehrjahr Lernender

|  |   |           |
|--|---|-----------|
| <b>A</b>   | <b>PLJ Pause beginnen Reminder erstellen</b>    | <b>8</b>  |
| <p>Ich als: Praxislehrjahr Lernender,</p> <p>möchte: jeden Mittwoch, Donnerstag und Freitag um 09:50 Uhr einen Reminder bekommen,</p> <p>um: eine Erinnerung zu haben, damit ich meine Pause beginne und vor den Basislehrjahr Lernenden im Coop sein kann.</p>  |   |           |
| <p>Die User Story ist fertig, wenn:</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Der User die App öffnen kann.</li> <li><input type="checkbox"/> Der User dem Reminder den Titel "Pause beginnen" geben kann.</li> <li><input type="checkbox"/> Der User die Zeit des Reminders auf 09:50 Uhr stellen kann.</li> <li><input type="checkbox"/> Der User Auswählen kann, dass der Reminder sich wöchentlich wiederholen soll.</li> <li><input type="checkbox"/> Der User beim Erstellen des Reminders die Tage Mittwoch, Donnerstag und Freitag auswählen kann.</li> <li><input type="checkbox"/> Der User den Reminder speichern kann.</li> <li><input type="checkbox"/> Der User am Mittwoch, Donnerstag und Freitag um 09:50 Uhr einen Reminder mit dem Titel "Pause beginnen" bekommt.</li> </ul> |   |           |
| <b>C</b>   | <b>PLJ Pause beenden Reminder Zeit anpassen</b> | <b>13</b> |
| <p>Ich als: Praxislehrjahr Lernender,</p> <p>möchte: jeden Mittwoch, Donnerstag und Freitag um 10:05 Uhr anstatt um 10:10 Uhr einen Reminder bekommen,</p> <p>um: eine Erinnerung zu haben, damit ich meine Pause beende und rechtzeitig, also nach 15 statt 20 Minuten, wieder an meinem Projekt weiterarbeiten kann.</p>   |   |           |
| <p>Die User Story ist fertig, wenn:</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Der User die App öffnen kann.</li> <li><input type="checkbox"/> Der User den Reminder mit dem Titel "Pause beenden" finden kann.</li> <li><input type="checkbox"/> Der User die Zeit des Reminders von 10:10 Uhr auf 10:05 Uhr ändern kann.</li> <li><input type="checkbox"/> Der User den bearbeiteten Reminder speichern kann.</li> <li><input type="checkbox"/> Der User am Mittwoch, Donnerstag und Freitag um 10:05 Uhr einen Reminder mit dem Titel "Pause beenden" bekommt.</li> </ul>  |   |           |

|   |  |          |
|---|--|----------|
| <b>A</b>  | <b>PLJ Arzttermin Reminder erstellen</b> | <b>3</b> |
| <p>Ich als: <b>Praxislehrjahr Lernender,</b></p> <p>möchte: <b>am 23.06.2023 um 11:00 Uhr einen Reminder bekommen,</b></p> <p>um: <b>mich daran zu erinnern, dass ich dann einen Arzttermin habe und die Noser Young verlassen muss.</b></p>  |  |          |
| <p>Die User Story ist fertig, wenn:</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Der User die App öffnen kann.</li> <li><input type="checkbox"/> Der User den Reminder mit dem Titel "Arzttermin" erstellen kann.</li> <li><input type="checkbox"/> Der User die Zeit des Reminders auf 11:00 Uhr setzen kann.</li> <li><input type="checkbox"/> Der User bestimmen kann, dass der Reminder sich nicht wiederholt und am 23.06.2023 sein soll.</li> <li><input type="checkbox"/> Der User den Reminder speichern kann.</li> <li><input type="checkbox"/> Der User am 23.06.2023 um 11:00 Uhr einen Reminder mit dem Titel "Arzttermin" bekommt.</li> </ul> |  |          |

## User Story Berufsbildner

|  |   |          |
|--|---|----------|
| <b>A</b>   | <b>Berufsbildner Feierabend machen Reminder erstellen</b> | <b>5</b> |
| <p>Ich als: <b>Berufsbildner,</b></p> <p>möchte: <b>jeden Tag einen Reminder um 17:00 Uhr bekommen,</b></p> <p>um: <b>eine Erinnerung zu haben, dass ich in den Feierabend gehen sollte, da ich mich mehr auf meine Work-Life Balance fokussieren möchte.</b></p>  |   |          |
| <p>Die User Story ist fertig, wenn:</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Der User die App öffnen kann.</li> <li><input type="checkbox"/> Der User dem Reminder den Titel "Feierabend machen" geben kann.</li> <li><input type="checkbox"/> Der User die Zeit des Reminders auf 17:00 Uhr stellen kann.</li> <li><input type="checkbox"/> Der User Auswählen kann, dass der Reminder sich täglich wiederholen soll.</li> <li><input type="checkbox"/> Der User den Reminder speichern kann.</li> <li><input type="checkbox"/> Der User täglich um 17:00 Uhr einen Reminder mit dem Titel "Feierabend machen" bekommt.</li> </ul> |   |          |

|  |  |          |
|--|--|----------|
| <b>A</b>   | <b>Berufsbildner Wasser trinken Reminder erstellen</b> | <b>5</b> |
| <p>Ich als: <b>Berufsbildner,</b></p> <p>möchte: <b>stündlich um XX:15 Uhr einen Reminder bekommen,</b></p> <p>um: <b>eine Erinnerung zu haben, dass ich ein Glas Wasser trinken gehen sollte, da ich mich mehr auf meine Gesundheit achten möchte.</b></p>  |  |          |
| <p>Die User Story ist fertig, wenn:</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Der User die App öffnen kann.</li> <li><input type="checkbox"/> Der User dem Reminder den Titel "Wasser trinken" geben kann.</li> <li><input type="checkbox"/> Der User Auswählen kann, dass der Reminder sich stündlich wiederholen soll.</li> <li><input type="checkbox"/> Der User Auswählen kann, zu welcher Minute der Reminder ist.</li> <li><input type="checkbox"/> Der User den Reminder speichern kann.</li> <li><input type="checkbox"/> Der User stündlich einen Reminder mit dem Titel "Wasser trinken" bekommt.</li> </ul> |  |          |

|   |  |           |
|---|--|-----------|
| <b>C</b>  | <b>Berufsbildner "BLJ Lernende kommen" Reminder Tag ändern</b> | <b>13</b> |
| <p>Ich als: <b>Berufsbildner,</b></p> <p>möchte: <b>anstatt jeden Mittwoch, Donnerstag und Freitag um 07:30 Uhr nun jeden Dienstag, Mittwoch und Donnerstag um 07:30 Uhr einen Reminder bekommen,</b></p> <p>um: <b>eine Erinnerung zu haben, dass bald die Lernenden vom Basislehrjahr kommen.</b></p>   |  |           |
| <p>Die User Story ist fertig, wenn:</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Der User die App öffnen kann.</li> <li><input type="checkbox"/> Der User den Reminder mit dem Titel "BLJ Lernende kommen" finden kann</li> <li><input type="checkbox"/> Der User die Erinnerung am Freitag durch eine Erinnerung am Dienstag ersetzen kann. Dies tut er durch das Auswählen vom Dienstag und das Entfernen des Häkchens vom Freitag.</li> <li><input type="checkbox"/> Der User den Reminder mit der Änderung speichern kann.</li> <li><input type="checkbox"/> Der User nun jeden Dienstag, statt jeden Freitag, um 07:30 Uhr einen Reminder mit dem Titel "BLJ Lernende kommen" bekommt.</li> </ul> |  |           |

## Funktionale Anforderungen

Das Ziel unseres Projektes ist eine Reminder Applikation. Diese soll eine Mobile App sein, die mit React Native in TypeScript implementiert wird. Man soll die CRUD Methoden, also Create, Read, Update und Delete verwenden können. In der Praxis bedeutet das, dass man einen Reminder anschauen, erstellen, bearbeiten und löschen kann. Die Reminder sollen auch nach dem Beenden der App gespeichert bleiben, dadurch kann man die App neu starten und hat die vorher erstellten Reminder noch. Das Speichern haben wir mithilfe von AsyncStorage gelöst. Der User unserer App soll durch Notifications an seine Reminder erinnert werden. Die Notification wird zur Zeit, zu der der Reminder ist, versendet.

Beim Erstellen eines Reminders muss man auswählen, wann der Reminder stattfindet und ob er sich wiederholt oder nicht. Wenn der Reminder sich wiederholt, wählt man aus, ob er sich stündlich, täglich oder wöchentlich wiederholt. Falls die Repetition wöchentlich ist, kann man entscheiden, an welchen Tagen der Reminder ist, dabei kann man auch mehrere Tage auswählen, zwingend ist jedoch einer. Der Text, welcher auch der Titel des Reminders ist, soll frei wählbar sein. Zudem soll man eine Notiz hinzufügen können.

## Nicht-funktionale Anforderungen

Die Darstellung und das Design sind ein wichtiger Faktor, um viele User zu erreichen. Das Design soll sich hauptsächlich an Leute im Alter von 15 bis 20 richten, aber auch den Geschmack 24-65-Jähriger treffen. Der Fokus liegt hier bei der Intuition der App, also dass sie logisch und einfach zu bedienen ist.

In der Praxis sieht das Design wie folgt aus: Um die Reminder anzuschauen, haben wir eine Übersichtsseite, auf der alle Reminder mit ihren Konfigurationen zu sehen sind - diese Seite ist unsere Homepage. Wenn man auf einen Reminder klickt, kann man diesen bearbeiten und durch einen Knopf auf dieser Bearbeitungsseite auch löschen. Um einen neuen Reminder zu erstellen, gibt es auf der Homepage einen Erstellknopf, dieser leitet auf die Erstellseite weiter.



# Architektur und Design

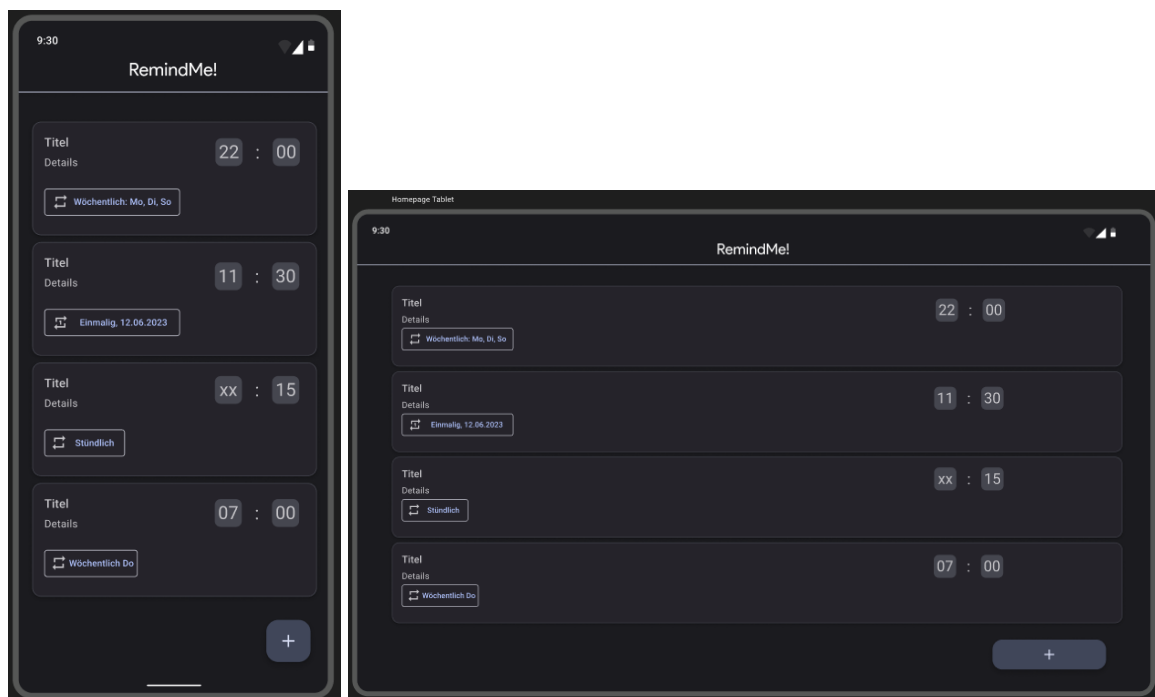
Link zum Figma: [Figma Mockup](#)

## Optische Anforderungen an das Produkt

Grundanforderungen an das Aussehen sind, dass die App responsive ist, also dass das Design für verschiedene Geräte mit verschiedenen Display Dimensionen verwendet werden kann. Da wir eine Mobile Applikation entwickeln, fokussieren wir uns darauf, dass das Design auf einem Smartphone und einem Tablet gut aussieht. Wie bereits bei den nicht-funktionalen Anforderungen erwähnt, ist es uns wichtig, dass das Design intuitiv und einfach zu bedienen ist. Das Design soll modern gestaltet sein und wir halten es so minimalistisch wie möglich. Zudem soll es unsere Zielgruppe optisch ansprechen, damit wir möglichst viele Kunden erreichen können. Dies berücksichtigen wir zum Beispiel, indem wir unsere App in einem dunklen Farbschema gestalten, da dieses bei unserer Zielgruppe beliebter ist als ein helles Farbschema.

## Homepage

Die Homepage soll verschiedene Karten mit den Remindern haben. Wir haben den Titel unserer App oben auf dem Bildschirm, dadurch kann man sie klar von anderen Apps unterscheiden. Die Homepage ist einfach gehalten, damit man sich auf die Reminder fokussieren kann und einen besseren Überblick hat.



## Erstellseite

Unsere Erstellseite soll Eingabefelder für alle benötigten Daten haben. Dabei haben wir, je nachdem ob und wie oft sich ein Reminder wiederholt, ein anderes Eingabefeld für die Definition des Datums oder Tages, an dem der Reminder wiederholt wird.

The image displays three mobile app screens for creating reminders, each with a dark theme and a top status bar showing 9:30. All screens have a back arrow, a checkmark, and a title bar.

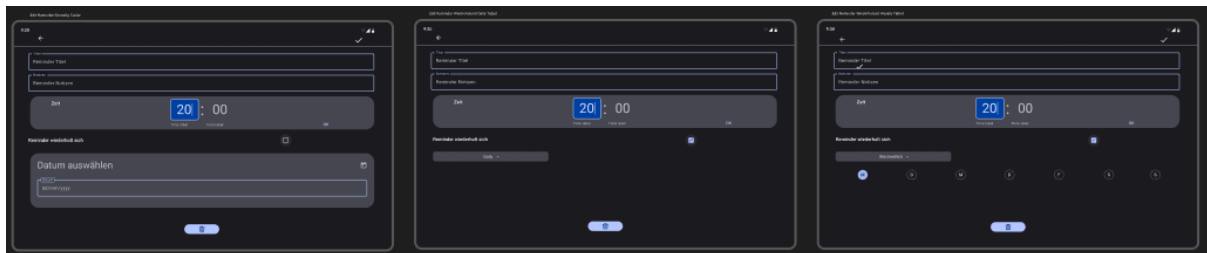
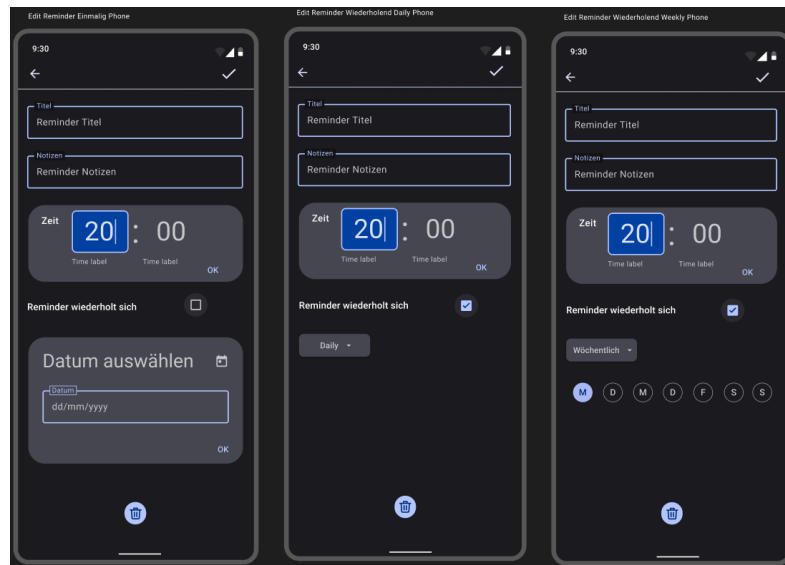
- Create Reminder Wiederholend Weekly Phone:** Features a 'Titel' field with a placeholder, a 'Notizen' field with a placeholder, a 'Zeit' picker set to 20:00, and a 'Reminder wiederholt sich' checkbox that is checked. Below the checkbox is a 'Wöchentlich' dropdown and a row of seven day selection buttons (M, D, M, D, F, S, S).
- Create Reminder Wiederholend Daily Phone:** Features the same 'Titel' and 'Notizen' fields, a 'Zeit' picker set to 20:00, and a 'Reminder wiederholt sich' checkbox that is checked. Below the checkbox is a 'Täglich' dropdown.
- Create Reminder Einmalig Phone:** Features the same 'Titel' and 'Notizen' fields, a 'Zeit' picker set to 20:00, and an unchecked 'Reminder wiederholt sich' checkbox. Below the checkbox is a 'Datum auswählen' section with a date picker icon and a text input field showing 'dd/mm/yyyy'.

The image displays three mobile app screens for editing reminders, each with a dark theme and a top status bar showing 9:30. All screens have a back arrow, a checkmark, and a title bar.

- Wiederholend Weekly Phone:** Features a 'Titel' field with a placeholder, a 'Notizen' field with a placeholder, a 'Zeit' picker set to 20:00, and a 'Reminder wiederholt sich' checkbox that is checked. Below the checkbox is a 'Wöchentlich' dropdown and a row of seven day selection buttons (M, D, M, D, F, S, S).
- Wiederholend Daily Phone:** Features the same 'Titel' and 'Notizen' fields, a 'Zeit' picker set to 20:00, and an unchecked 'Reminder wiederholt sich' checkbox. Below the checkbox is a 'Datum auswählen' section with a date picker icon and a text input field showing 'dd/mm/yyyy'.
- Einmalig Phone:** Features the same 'Titel' and 'Notizen' fields, a 'Zeit' picker set to 20:00, and an unchecked 'Reminder wiederholt sich' checkbox. Below the checkbox is a 'Datum auswählen' section with a date picker icon and a text input field showing 'dd/mm/yyyy'.

## Bearbeitungsseite

Die Bearbeitungsseite ist grundsätzlich gleich aufgebaut wie die Erstellseite. Doch sie hat einen zusätzlichen Knopf, der für das Löschen eines Reminders verantwortlich ist. Zudem werden bei der Bearbeitungsseite die Daten des Reminders angezeigt, bei der Erstellseite hingegen werden Platzhalter angezeigt.



## Funktionale Anforderungen an das Produkt

### Anforderungen an die Struktur

Der Code der App wird, soweit es sinnvoll ist, in Components aufgeteilt. Dies ist zum Beispiel bei der Anzeige der Zeit der Fall, da dieser Component gut wiederverwendet werden kann. Jede Seite unserer App hat eine eigene Page, diese enthält gegebenenfalls Components. Für das Umsetzen des Designs im Code verwenden wir React Paper als Design Library.

Unser Code wird mit TSDoc dokumentiert, wobei wir uns auf die wichtigen Funktionen begrenzen. Das bedeutet, dass wir zum Beispiel keine Dokumentation für Getter und Setter haben wollen, da diese auch so verständlich sind und deren Dokumentation den Code nur schlechter lesbar macht.

Das Konzept von Clean Code liegt uns bei unserem Projekt am Herzen, damit der Code gut verständlich ist. Zudem hat man so die Möglichkeit in Zukunft an diesem Projekt einfach weiterzuarbeiten und es zu erweitern.

## Anforderungen an das Verhalten

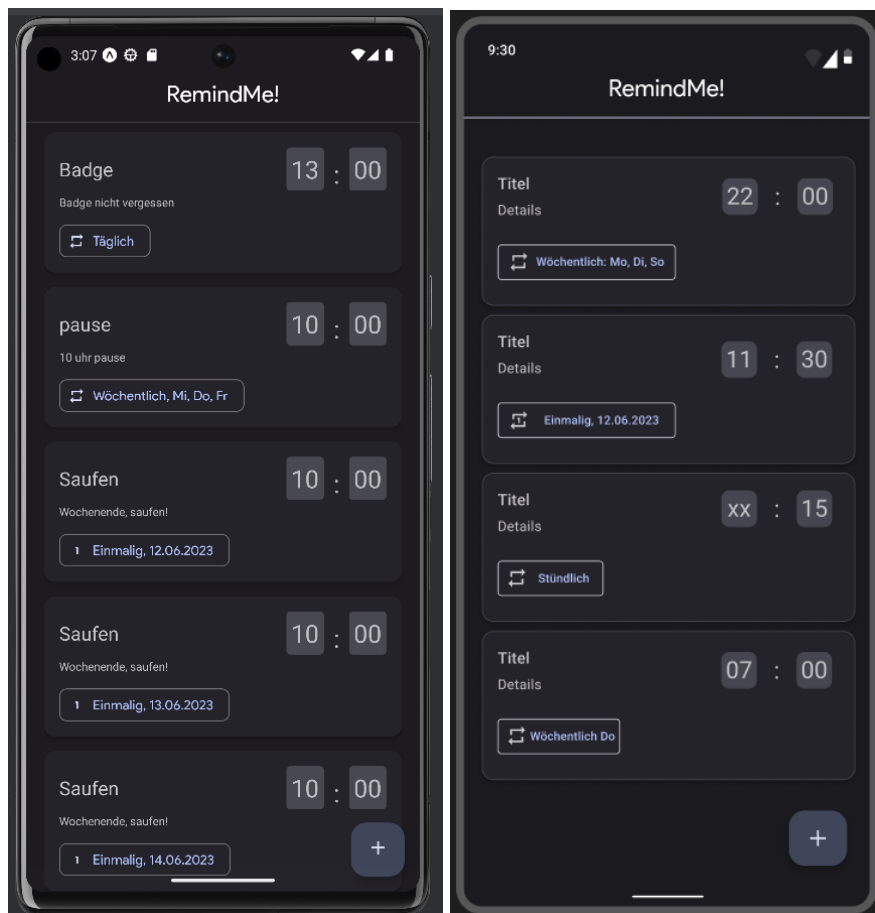
Wie bei den funktionalen Anforderungen erwähnt, ist das Ziel unseres Projekts eine mobile Applikation mit React Native in TypeScript. Mit unserer App sind alle CRUD Methoden ausführbar. Also das Erstellen, Anschauen, Bearbeiten und Löschen von Remindern. Die Daten der Reminder sind persistent. Dies bedeutet, dass man die App neu starten kann und die erstellten Reminder immer noch vorhanden sind. Dafür verwenden wir den AsyncStorage. Aufgrund dessen, wann ein Reminder ausgeführt werden soll und wie oft er sich wiederholt, werden Notifications versendet und zwar jedes Mal, wenn die Erinnerung ist.

Unsere App ist für das Erstellen von Remindern, für das Einpacken des Badges und das Einhalten der Pausenzeiten ausgelegt. Doch man kann den Text und den Titel eines Reminders selbst bestimmen, somit ist der Reminder universell anwendbar. Zudem kann man einstellen, ob und wie oft sich der Reminder wiederholt. Dort kann man zwischen stündlich, täglich und wöchentlich auswählen. Wenn der Reminder sich stündlich wiederholt, muss man nur die Zeit des Reminders eingeben und er wird dann stündlich zur gleichen Minute ausgelöst. Bei der täglichen Repetition wählt man auch eine Zeit aus und der Reminder wird jeden Tag zur selben Stunde und Minute ausgeführt. Zur wöchentlichen Wiederholung wählt man die Tage aus, an denen der Reminder ausgeführt wird. Dabei muss man mindestens einen Tag wählen und kann maximal alle Tage nehmen.

## Vergleich Mockup und GUI

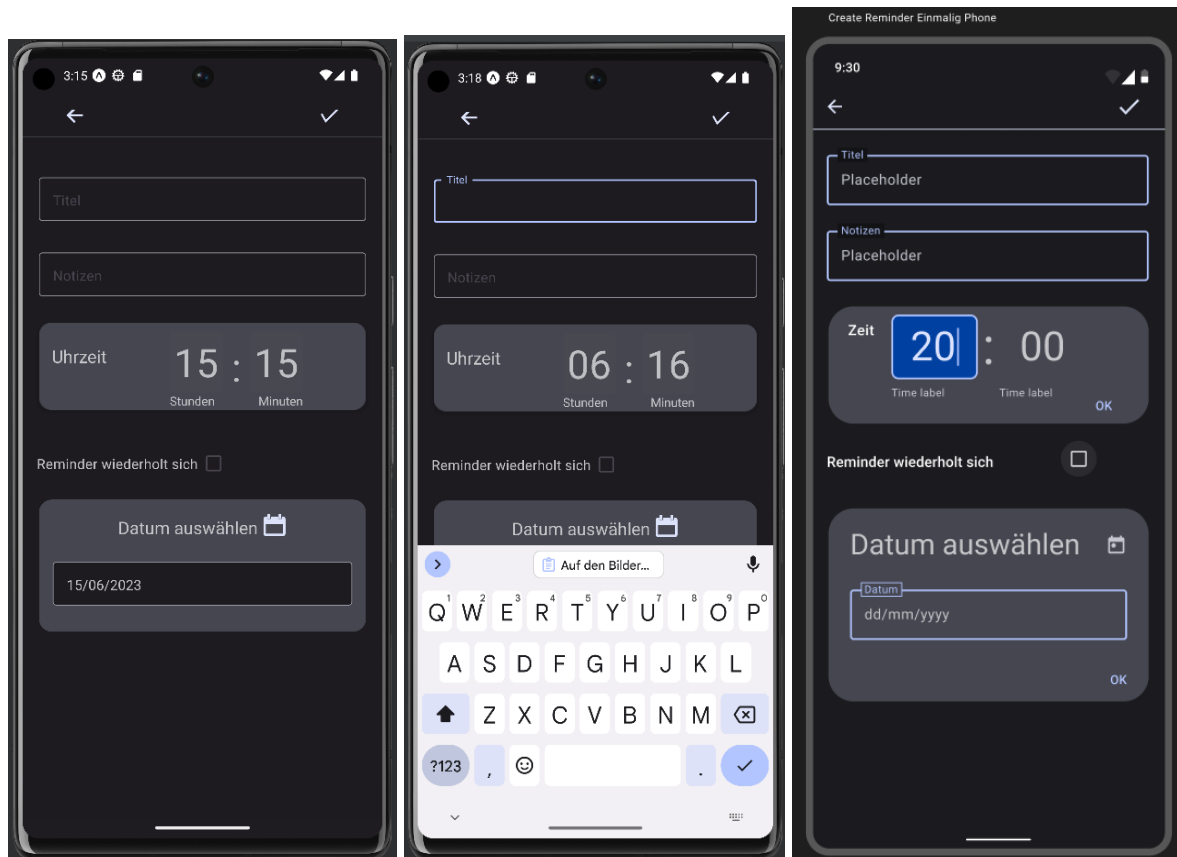
### Homepage

Auf den Bildern unten sehen wir links unsere Implementation und rechts unser Mockup. Deren Design ist sehr ähnlich, jedoch nicht exakt gleich. Wir haben beim Umsetzen gemerkt, dass wir gewisse Details wie die Grenze unter unserem Logo feiner gestalten sollten, da es auf dem Gerät besser aussieht. Im Figma sieht man die Details jedoch besser, wenn sie dominanter sind, deshalb haben wir diesen kleinen Unterschied beibehalten. Das Design, die Farben und auch die Proportionen sind bei der Implementation gleich, das heisst, dass unsere Planung in Figma gut und umsetzbar ist und wir mit dem Design zufrieden sind.



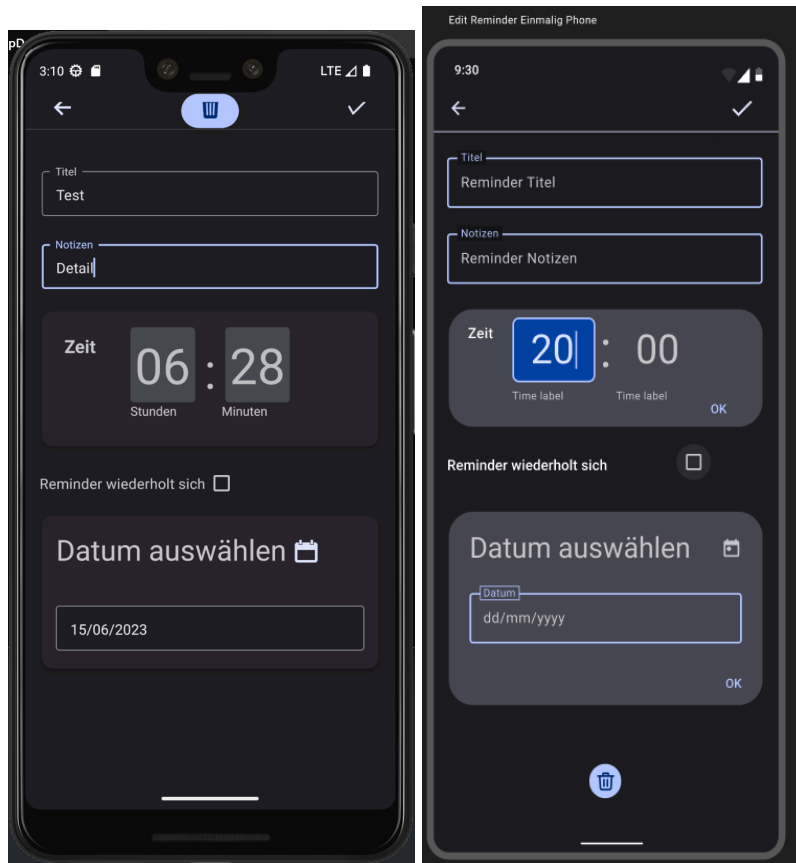
## Erstellseite

Auf den Bildern unten sehen wir links und in der Mitte unsere Implementation und rechts unser Mockup. Wie bei der Homepage haben wir hier bei der Implementation auch wieder gewisse Details etwas dezenter gestaltet. Die Textfelder werden erst beim Bearbeiten blau gefärbt und sind sonst grau. Dies ermöglicht es, dass der User sieht, welches Feld er gerade bearbeitet. Das ist zwar eine kleine Abweichung vom Figma, aber aus praktischen Gründen haben wir uns entschieden, dies so umzusetzen.



## Bearbeitungsseite

Auf den Bildern unten sehen wir links unsere Implementation und rechts unser Mockup. Wie bei der Erstellseite werden die Eingabefelder hier nur bei der Bearbeitung blau, ansonsten sind sie grau. Die Begründung ist auch wieder, dass es so nutzerfreundlicher ist. Zudem haben wir uns dazu entschieden, bei der Implementation den Delete Button oben auf der Seite zu machen, da so alle Bedienungsknöpfe nahe beieinander sind.



# Technischer Bericht

## Testing

Das Testing unserer Applikation ist manuell. Wir haben zu Beginn des Projektes das unten stehende Testprotokoll erstellt, mit dem wir die Anforderungen an unsere App anhand unserer User Stories überprüfen können. Diese Testfälle werden im Verlauf des Projekts von einem von uns durchgeführt und das Ergebnis wird protokolliert.

| Testprotokoll       |                |  |  |   |  |  |        |                    |
|---------------------|----------------|--|--|---|--|--|--------|--------------------|
| TestCaseID          | Komponente     | Beschreibung   | Voraussetzungen                                | Test Schritte   | Erwartetes Resultat  | Resultat beim Test   | Status | Durchgeführt durch |
| CreateRepeat Weekly | CreateReminder | Einen Reminder, mit dem Titel "Badge mitnehmen" und der Notiz "Den Badge einpacken" erstellen, der sich wöchentlich am Mittwoch, Donnerstag und Freitag um 06:00 Uhr wiederholt. | App ist auf dem Mobilgerät und der Code läuft. | 1) App öffnen<br>2) Auf den hinzufügen Knopf drücken<br>3) Titel "Badge mitnehmen" und Notiz "Den Badge einpacken" schreiben<br>4) 06:00 Uhr als Zeit für den Reminder auswählen<br>5) Reminder wiederholt sich auswählen<br>5) Wöchentlich wiederholen auswählen<br>6) Mittwoch, Donnerstag und Freitag auswählen bei den Tagen an denen sich der Reminder wiederholt<br>7) Reminder speichern | Reminder mit dem Titel "Badge mitnehmen" und der Notiz "Den Badge einpacken" wird erstellt. Dieser Reminder wiederholt sich jeden Mittwoch, Donnerstag und Freitag um 06:00 Uhr. | Reminder mit dem Titel "Badge mitnehmen" und der Notiz "Den Badge einpacken" wird erstellt. Dieser Reminder wiederholt sich jeden Mittwoch, Donnerstag und Freitag um 06:00 Uhr. |        | Cedric Zollinger   |



|                           |                    |   |  |   |   |  |  |                  |
|---------------------------|--------------------|---|--|---|---|--|--|------------------|
| CreateRepeat<br>Daily     | CreateRe<br>minder | Einen Reminder, mit dem Titel "Feierabend machen" und der Notiz "Nach Hause gehen und entspannen" erstellen, der sich täglich um 17 Uhr wiederholt. | App ist auf dem Mobilgerät und der Code läuft. | 1) App öffnen<br>2) Auf den hinzufügen Knopf drücken<br>3) Titel "Feierabend machen" und Notiz "Nach Hause gehen und entspannen" schreiben<br>4) 17:00 Uhr als Zeit für den Reminder auswählen<br>5) Reminder wiederholt sich auswählen<br>5) Täglich wiederholen auswählen<br>6) Reminder speichern  | Reminder mit dem Titel "Feierabend machen" und der Notiz "Nach Hause gehen und entspannen" wird erstellt. Dieser Reminder wiederholt sich jeden Tag um 06:00 Uhr. | Reminder mit dem Titel "Feierabend machen" und der Notiz "Nach Hause gehen und entspannen" wird erstellt. Dieser Reminder wiederholt sich jeden Tag um 06:00 Uhr.  |  | Cedric Zollinger |
| CreateRepeat<br>EveryHour | CreateRe<br>minder | Einen Reminder, mit dem Titel "Wasser trinken" und der Notiz "Ein Glas Wasser trinken" erstellen, der sich stündlich wiederholt.                    | App ist auf dem Mobilgerät und der Code läuft. | 1) App öffnen<br>2) Auf den hinzufügen Knopf drücken<br>3) Titel "Wasser trinken" und Notiz "Ein Glas Wasser trinken" schreiben<br>4) 00:15 Uhr als Zeit für den Reminder auswählen, damit der Reminder sich ab dem nächsten Tag stündlich wiederholt<br>5) Reminder wiederholt sich auswählen<br>6) Stündlich wiederholen auswählen<br>7) Reminder speichern | Reminder mit dem Titel "Wasser trinken" und der Notiz "Ein Glas Wasser trinken" wird erstellt. Dieser Reminder wiederholt sich jede Stunde um XX:15 Uhr.          | Reminder mit dem Titel "Wasser trinken" und der Notiz "Ein Glas Wasser trinken" wird erstellt. Dieser Reminder wiederholt sich jede Stunde um XX:15 Uhr. Jedoch versendet der Reminder keine Notification. |  | Cedric Zollinger |
| UpdateRepeat<br>Weekly    | UpdateR<br>eminder | Ein Reminder, mit dem Titel "BLJ Lernende kommen", der  | - App ist auf dem Mobilgerät und der Code      | 1) App öffnen<br>2) Den Reminder mit dem Titel "BLJ Lernende kommen" finden   | Der Reminder mit dem Titel "BLJ Lernende  | Der Reminder mit dem Titel "BLJ Lernende   |  | Danny Nguyen     |

|                |                |  |  |   |   |   |  |               |
|----------------|----------------|--|--|---|---|---|--|---------------|
|                |                | sich wöchentlich jeden Mittwoch, Donnerstag und Freitag wiederholt, soll sich von nun an wöchentlich jeden Dienstag, Mittwoch und Donnerstag wiederholen.                  | läuft.<br>- Ein Reminder mit dem Titel "BLJ Lernende kommen" und den Einstellungen aus der Beschreibung existiert.               | 3) Auf den Bearbeitungs-Button klicken<br>4) Den Freitag auswählen<br>5) Den Dienstag auswählen<br>6) Reminder speichern  | kommen" wiederholt sich nun am Dienstag, Mittwoch und Donnerstag und nicht mehr am Freitag.   | kommen" wiederholt sich nun am Dienstag, Mittwoch und Donnerstag und nicht mehr am Freitag.   |  |               |
| CreateNoRepeat | CreateReminder | Einen Reminder erstellen, der einmalig am 23.06.2023 um 11:00 ist, dieser soll den Titel "Arzttermin" haben.   | App ist auf dem Mobilgerät und der Code läuft.   | 1) App öffnen<br>2) Auf den hinzufügen Knopf klicken<br>3) Titel "Arzttermin" und schreiben<br>4) 11:00 Uhr als Zeit für den Reminder auswählen<br>5) 23.06.2023 als Datum für den Reminder eingeben<br>6) Reminder speichern | Ein Reminder mit dem Titel "Arzttermin" wird erstellt, dieser ist um 11:00 Uhr am 23.06.2023. | Ein Reminder mit dem Titel "Arzttermin" wird erstellt, dieser ist um 11:00 Uhr am 23.06.2023. |  | Amanda Walser |
| Delete         | UpdateReminder | Einen Reminder löschen. Dabei nehmen wir den Reminder, den der BLJ Lernende benutzt, um den Badge einzupacken, da dieser nun in seine Firma geht, wo er den Reminder nicht | - App ist auf dem Mobilgerät und der Code läuft.<br>- Ein Reminder mit dem Titel "Badge mitnehmen" und den Einstellungen aus der | 1) App öffnen<br>2) Den Reminder mit dem Titel "Badge mitnehmen" finden<br>3) Auf den Bearbeitungs-Button klicken<br>4) In der Bearbeitungsansicht auf den Delete-Button drücken  | Der Reminder mit dem Titel "Badge mitnehmen" wird gelöscht und ist nicht mehr vorhanden.      | Der Reminder mit dem Titel "Badge mitnehmen" wird gelöscht und ist nicht mehr vorhanden.      |  | Danny Nguyen  |

|                                   |                |  |  |   |  |  |  |               |
|-----------------------------------|----------------|--|--|---|--|--|--|---------------|
|                                   |                | mehr braucht.  | Beschreibung existiert.                        |   |  |  |  |               |
| ErrorHandlingInvalidHour          | CreateReminder | Beim Erstellen eines Reminders wird eine Zeit eingegeben, die nicht existiert. Ein Beispiel hierfür wäre 26:00 Uhr | App ist auf dem Mobilgerät und der Code läuft. | 1) App öffnen<br>2) Auf den hinzufügen Knopf klicken<br>3) Titel "Reminder" und schreiben<br>4) 26:00 Uhr als Zeit für den Reminder auswählen | Die Zeit kann nicht ausgewählt werden und geht auf 24:00 Uhr zurück. | Die Zeit kann nicht ausgewählt werden und geht in der Zeitauswahl auf 0:00 Uhr zurück. Wenn man aus dem Zeit-Auswahl menü geht, wird es von 0:00 Uhr zu 24:00 Uhr.   |  | Amanda Walser |
| ErrorHandlingInvalidMinuteAndHour | CreateReminder | Beim Erstellen eines Reminders wird eine Zeit eingegeben, die nicht existiert. Ein Beispiel hierfür wäre 26:30 Uhr | App ist auf dem Mobilgerät und der Code läuft. | 1) App öffnen<br>2) Auf den hinzufügen Knopf klicken<br>3) Titel "Reminder" und schreiben<br>4) 26:30 Uhr als Zeit für den Reminder auswählen | Die Zeit kann nicht ausgewählt werden und geht auf 24:00 Uhr zurück. | Die Zeit kann nicht ausgewählt werden und die Stunde geht in der Zeitauswahl auf 0:00 Uhr zurück. Doch die Auswahl der Minuten bleibt bestehen. Wenn man aus dem Zeit-Auswahl menü geht, wird die Stunde von 0:00 Uhr zu |  | Amanda Walser |

|  |  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  | 24:00 Uhr.<br>Doch die 30<br>Minuten<br>bleiben<br>bestehen. In<br>unserer App<br>wird dies dann<br>als 24:30 Uhr<br>abgespeichert<br>und nicht als<br>00:30 Uhr,<br>was eine Zeit<br>ist, die nicht<br>existiert. |  |  |
|--|--|--|--|--|--|--|--|--|

## Ausblick

Aufgrund des Testings und der Wahrnehmung der App gibt es Sachen, die man für die Zukunft noch verändern kann, um die App zu optimieren. Der nächste Schritt wäre es, die Validation und das Error Handling zu verbessern respektive zu erweitern. In unserer App sind momentan noch Aktionen ausführbar, die nicht möglich sein sollten. Den Test, welchen ich damit anspreche, ist das Eingeben einer Zeit, die nicht existiert. Das sollte erweitert werden, um auszuschliessen, dass der User einen Reminder erstellt, der nicht funktional ist.

Zudem wäre es ein guter Schritt, als Nächstes User Feedback einzuholen. Dies im Bezug auf das Design und auch die Funktionalitäten. Denn das Wichtigste an einer App ist die Nutzerzufriedenheit, welche so gut verbessert werden könnte.

## Auswertung

### Erreichung der Ziele

#### Ziele der Struktur

Unsere App hat Components, die wir wiederverwenden können. Dies nutzen wir zum Beispiel beim Setzen oder Bearbeiten der Zeit. Zudem verwenden wir Pages, wie wir es geplant haben. Für das Design nutzen wir die React Paper Library, womit dieses Ziel auch erfüllt ist.

Die wichtigsten Funktionen in unserem Code haben wir mit TSDoc dokumentiert. Ausserdem ist unser Code verständlich. Jedoch haben wir die Clean Code Richtlinien nicht ausnahmslos eingehalten, dieses Ziel ist also nur teilweise erfüllt und wenn wir noch an dem Projekt weiterarbeiten würden, sollte man das noch verbessern.

#### Ziele des Verhaltens

Das Hauptziel, also das Erstellen einer Mobile App mit React Native, welche die CRUD Methoden möglich macht, ist erreicht. Die Konfigurationsmöglichkeiten, welche unser Ziel waren, sind auch alle möglich. Wir haben unser Ziel also erreicht. Das Design unserer App ist unserem Figma Design sehr nahe und wir haben nur kleine Anpassungen vorgenommen, die das Benutzen der App erleichtern und somit die App Nutzerfreundlicher machen. Die Persistierung von Daten ist ebenfalls ausnahmslos möglich und somit ein erreichter Punkt unseres Zieles.

Doch es gibt auch Ziele, die wir nicht vollständig erfüllt haben, wie das Testing zeigt. Dabei beziehen wir uns vor allem auf die Validierung der Eingaben, diese wäre für eine fertiggestellte App noch nicht ausreichend. Doch diese fehlende Validierung bringt unsere App nicht zum Abstürzen, was bedeutet, dass die App trotzdem gut nutzbar ist. Bei den Notifications ist unser Ziel leider auch nur teilweise erreicht, da wir Notifications versenden können und diese auch auf die täglichen oder wöchentlichen Zeiten und Daten anpassen

können. Jedoch haben wir es nicht geschafft, die Reminder stündlich zu einer bestimmten Minute versenden zu können.

Als Fazit können wir also sagen, dass unsere App funktioniert, aber noch einige Verbesserungen zu machen sind, bis sie voll und ohne Fehlerrisiko nutzbar ist. Trotzdem sind die Grundanforderungen und somit unser Hauptziel erfüllt.