



UEK 335

MOBILE-APPLIKATION REALISIEREN DOKUMENTATION

NGOC-PHUC NGUYEN

GIAN-LUCA KUNFERMANN

INHALT

1.	Einleitung.....	3
1.1	Beschreibung.....	3
1.2	Ausgangslage.....	3
1.3	Ziele.....	3
1.4	Ausblick.....	3
2.	Anforderungen	5
2.1	User-Stories.....	5
2.2	Personas.....	7
3.	Architektur / Design.....	8
3.1	Figma-Mockup	8
	LandingPage.....	8
	AddingPage / Editpage	9
3.2	Effektives Design	10
	Landingpage.....	10
	New Reminder Page / Editpage.....	11
3.3	Architektur, Frameworks & Coding Guidelines.....	12
4.	Technischer Bericht	13
4.1	Testing.....	13
	Teilnehmer des Teams.....	13
	Ziel der Tests.....	13
	Art der Tests	13
	Testvorgaben.....	13
	Testvalidierung	13
	Testprotokoll.....	13
5.	Fazit	16
5.1	Gian-Luca Kunfermann.....	16
5.2	Ngoc-Phuc Nguyen.....	16
6.	Quellenverzeichnis	17

1. EINLEITUNG

1.1 BESCHREIBUNG

Im Rahmen des überbetrieblichen Kurses 335 müssen wir eine Mobile-Applikation realisieren. Der Auftrag liegt darin, eine Erinnerungs-App zu erstellen, mit der ein Nutzer bestimmte Termine festlegen kann, bei der die App ihn an ein Ereignis erinnern soll. Die Art der Ereignisse ist gruppenspezifisch, haben aber auf die Anforderungen der App keinen besonderen Einfluss. In unserem Fall sollen wir einen Reminder für die Einnahme von Medikamenten kreieren.

1.2 AUSGANGSLAGE

Für die Entwicklung der App wird React Native verwendet aufgrund unseres Vorwissens in React TS/JS. Folgende Elemente sind uns bereits gegeben:

- Projekt aufgesetzt von Gianluca mit Beispielen zu den jeweiligen Aspekten der Applikation
- Verschiedene Libraries (notifee, paper, etc.) welche uns bereits Code zur Verfügung stellen

Für die Funktionalitäten wurde uns ein Beurteilungsraster gegeben an welchem wir uns orientiert haben.

Der Prototyp, beziehungsweise das Mockup wurde mit Figma erstellt und soll dem Endprodukt eins zu eins entsprechen.

1.3 ZIELE

Es soll ein Produkt erstellt werden, welches einem Benutzer die Möglichkeit gibt, eine Erinnerung zu erstellen, welche ihm per Notification daran erinnert, die Medikamente einzunehmen. Dabei soll auf mehrere Dinge geachtet werden.

- Verwirklichung anhand des Bewertungsrasters
- Applikation bietet zwei Screens, der Erste zum Setzen der Reminder, der Zweite um alle bereits erstellten anzeigen zu lassen
- Einer bis alle Tage der Woche sollen als Checklist auswählbar sein
- Die Zeit der Erinnerung ist Standortabhängig und wird in Stunden und Minuten angezeigt
- Man soll die Möglichkeit haben, die gewählten Tage wiederholen zu lassen, so dass die Erinnerung über mehrere Wochen läuft
- Änderungen jeglicher Art werden auf dem lokalen Speicher des Mobiltelefons gespeichert
- Beim Zeitpunkt der Erinnerung soll dem Benutzer eine Notification gesendet werden

Zusätzlich zur Applikation sind zwei Dokumente abzugeben. Einerseits die Hauptdokumentation, in welcher wir uns befinden und einem Read.me welches die Installationsanleitung enthält. Ebenso muss die Applikation in einer zwanzigminütigen Präsentation veranschaulicht werden.

1.4 AUSBLICK

Aufgrund unvorhergesehener Aspekte mussten wir mit einigen Problem kämpfen die es uns verhinderten, alle Punkte der praktischen Umsetzung zu erfüllen. Diese werden kurz erläutert:

- Translation: Wir vermuten, dass unser Time Picker, welcher ebenfalls etwas ähnliches wie Translation Files verwendet, mit der Translation von i18n interagiert oder sogar blockiert. Wir haben vier Verschiedene Varianten versucht, um das Ganze zu kombinieren, sind aber auch mit der Hilfe von Gianluca fehlgeschlagen. Deshalb befinden sich lediglich die Überreste des letzten Versuches

weiterhin im Code, sollten wir doch noch eine Lösung finden in naher Zukunft, so dass die bereits geleistete Arbeit nicht umsonst war.

- Notifications: Die Implementation von Benachrichtigungen hat bereits zu Beginn einige Probleme bereitet, wobei nach anfänglichen Schwierigkeiten die Integration funktioniert hat. Die Implementation und das Benutzen hat leider auch nach zwei Tagen herumprobieren nicht funktioniert. Wir können zwar mit Erstellen eines neuen Reminders, oder mit der Anpassung des Switches eines bestehenden Reminders, einen Trigger setzen, welcher die Notifikation auslösen sollte, jedoch erschien bis heute keine dieser Benachrichtigungen.

Somit kann abschliessend erwähnt werden, dass die Applikation leider nicht hundertprozentig unsere Wünschen entspricht. Da wir beide viel von uns selbst halten haben wir auch immer hohe Ansprüche an uns selbst. Diesen sind wir leider nicht ganz gerecht geworden, auch wenn die Gründe dafür nicht bei uns zu suchen sind. Sollte sich in den nächsten Wochen die Zeit ergeben, dass wir an diesem Projekt weiterarbeiten würden, so können wir aber doch auf einer sehr soliden Basis weitermachen, die schon viele Funktionen enthält.

2. ANFORDERUNGEN

2.1 USER-STORIES

A	Reminder erstellen	8
<p>Ich als: App-Nutzer</p> <p>möchte: Reminder erstellen können,</p> <p>um: mich an das Einnehmen von Medikamenten zu erinnern</p>		
<p>Die User Story ist fertig, wenn:</p> <ul style="list-style-type: none"> ○ Der Nutzer kann eine beliebige Menge an Reminder erstellen ○ Der Nutzer kann Reminder als aktiv oder inaktiv setzen ○ Folgende Eigenschaften können beim Reminder gesetzt werden <ul style="list-style-type: none"> • Titel des Reminders • Beschreibung des Reminders • Uhrzeit im 24h-Format • An welchen Tagen der Reminder eingesetzt wird • Wie oft der Reminder wiederholt werden soll 		

B	Reminder bearbeiten	4
<p>Ich als: App-Nutzer</p> <p>möchte: Reminder bearbeiten können,</p> <p>um: die Zeiten der Medikamenteneinnahme anzupassen</p>		
<p>Die User Story ist fertig, wenn:</p> <ul style="list-style-type: none"> ○ Der Nutzer kann einen bestimmten Reminder auswählen und anklicken ○ Der Nutzer wird auf eine Seite geführt wo man die Zeiten / Daten anpassen kann ○ Folgende Eigenschaften können beim Reminder gesetzt werden <ul style="list-style-type: none"> • Titel des Reminders • Beschreibung des Reminders • Uhrzeit im 24h-Format • An welchen Tagen der Reminder eingesetzt wird • Wie oft der Reminder wiederholt werden soll 		

C	Reminder löschen	7
<p>Ich als: App-Nutzer</p> <p>möchte: Reminder löschen können,</p> <p>um: nicht mehr gebrauchte Reminder zu entfernen</p>		
<p>Die User Story ist fertig, wenn:</p> <ul style="list-style-type: none">○ Der Nutzer kann einen bestimmten Reminder lange anklicken○ Der Nutzer erhält einen Knopf, um den Reminder löschen zu können○ Der Reminder wird aus der Liste gelöscht und kann nicht mehr angezeigt werden		

2.2 PERSONAS



Hans Peter Geerdes, 58

Musiker

Bio

Herr Geerdes ist ein alteingesessener Musiker, der schon seit Jahrzehnten die Technoszene mit seiner Musik bereichert. Nun hat das Alter ihn eingeholt und benötigt regelmässige Einnahmen von lebenswichtigen Medikamenten.

Wünsche, Bedürfnisse

- Reminder, der bei der Benachrichtigung einen lauten Ton abspielt
- Sollte einfach zu bedienen sein
- Schriften sollten gross genug und lesbar sein

Fazit

Ein Programm, das übersichtlich und einfach zu bedienen ist.

Frust

Frühere Programme sind oft zu kompliziert aufgebaut. Er möchte ein Programm haben, das seine Medizineinnahmen übersichtlich darstellt.

Lieblingsmarken





Lea Himmelsläufer, 25

Kleinkinderzieherin

Bio

Frau Himmelsläufer hat ihr Hobby zum Beruf gemacht und möchte Kindern eine Bezugsperson geben, nachdem sie früh im Leben ihre Eltern verloren hat.

Wünsche, Bedürfnisse

- Da sie in ihrem Alltag viel mit beeinträchtigten Kindern arbeitet, möchte sie gerne eine Reminder App haben, die viele Reminder fassen kann
- Die Benachrichtigungen sollte eine Beschreibung liefern, die von ihr verfasst wurde, damit sie die Benachrichtigung einfach unterscheiden kann

Fazit

Möchte eine vernünftige App zu vernünftigen Preis.

Frust

Komplexe Programme sind ihr in der Regel zu teuer. Gratis-Programme limitieren ihre Funktionen oder sind voller Werbungen.

Lieblingsmarken



3. ARCHITEKTUR / DESIGN

3.1 FIGMA-MOCKUP

In einem ersten Schritt galt es mit Figma ein Mockup zu erstellen, was letztendlich dem Endprodukt nicht nur nahe, sondern identisch sein soll. Grundsätzlich sind wir mit unserem ersten Konzept sehr weit gekommen. Die Idee, welche wir hatten und wie die Umsetzung davon auszusehen hatte, formierte sich bei uns schnell. Unserer Meinung nach sollte die Applikation zwei, beziehungsweise drei verschiedenen Seiten bestehen, welche nun im Detail erläutert werden.

LANDINGPAGE

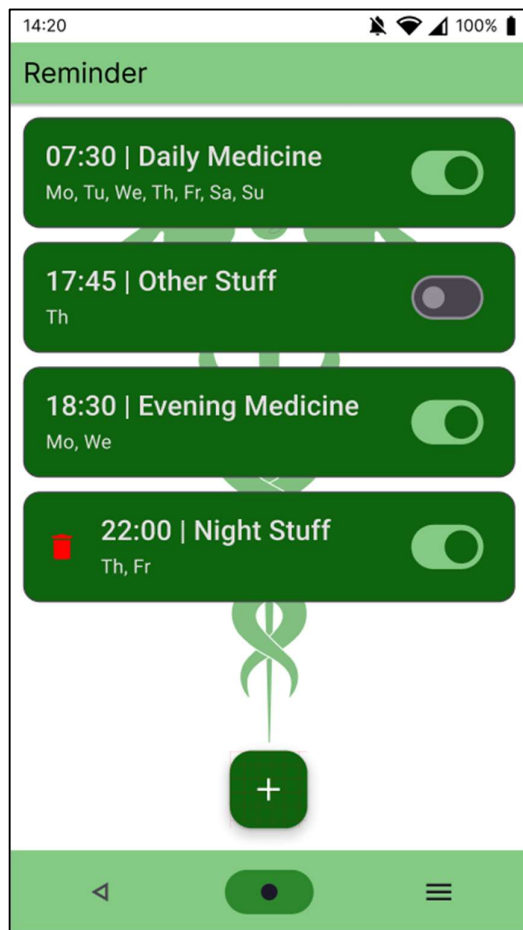


Abbildung 1 Startseite der App

Sollte der Benutzer aber länger auf die Card drücken, erscheint ein weiterer Knopf mit einem Kübelsymbol, woraus ersichtlich wird, dass der Reminder so gelöscht werden kann.

Zuerst vorneweg: Die Leisten ganz oben und ganz unten stellen unsere Interpretation des Android Betriebssystems dar. Diese können von Model zu Model variieren. Ebenso sehen die Switches, welche hier zu sehen sind und direkt von der React Native Paper Figma Library genommen wurden, im Android Betriebssystem anders aus, diese hier scheinen von iOS zu stammen. Nichts destotrotz, versuchten wir das ganze so simpel wie möglich zu halten. Ganz oben erhält die Seite einen prägnanten Namen (Reminder), welcher die Seite einleitet und dem Benutzer direkt Auskunft gibt was er zu erwarten hat. Anschliessend gibt es eine Auflistung der bereits erstellten Reminder, diejenigen welche aktiv sind und eine Notification senden sollen wenn der Zeitpunkt des Reminders eintrifft und auch diejenigen welche inaktiv sind. Über der unteren Android Leiste befindet sich zentriert ein Floating Action Button, welcher ein Plus Symbol enthält. Dadurch wird auch ohne weitere Informationen ersichtlich, dass man anhand dieses Knopfes einen neuen Reminder erstellen kann. Auch wenn die Seite ansonsten keine anklickbaren Stellen wie weitere Knöpfe oder dergleichen enthält, gibt es trotzdem zusätzliche Funktionalitäten. Intuitiv wird einem normalen Smartphone User bewusst sein, dass auch die einzelnen Reminder, welche als Cards dargestellt werden, anklickbar sind. Dabei zeigen sich zwei verschiedene Funktionalitäten, anhand verschiedener Druckweisen. Einerseits kann mit einem einzelnen kurzen Klick der Reminder in einer weiteren Seite angepasst werden.

ADDINGPAGE / EDITPAGE

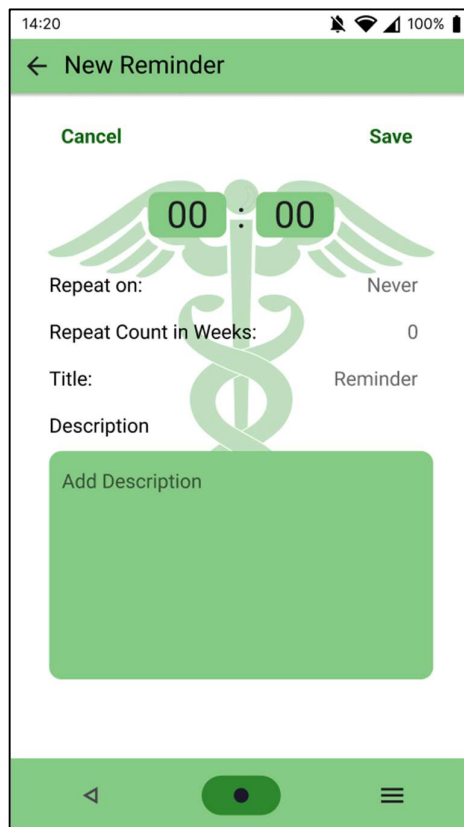


Abbildung 2 Mockup: Erstellen eines neuen Reminders

Nachdem der Plus Knopf gedrückt wurde, wird der Benutzer auf Seite geführt, in der ein neuer Reminder erstellt werden kann. Dabei entspricht das Design dem Schema welches wir bereits in der vorherigen Seite aufgebaut haben. Die Beiden Knöpfe in diesem Fenster sind so benannt, dass direkt klar wird was ihre Funktion ist. Abgesehen davon, kann der User auch mit allem anderen interagieren. Dabei erscheint beim Klicken auf «Description», «Title» und «Repeat Count in Weeks» jeweils ein Text Fenster, in welchem der gewünschte Eintrag erfasst werden kann. Für «Repeat on» und für die Erfassung der Zeit, werden dem Benutzer direkt die korrekten Auswahlmöglichkeiten angezeigt. Diese werden unten aufgeführt. Beim Klicken auf «Repeat on» erscheint ein kleines Pop-up Fenster, in welchem der Benutzer einen, oder mehrere Tage der Woche auswählen kann. Diese werden anschliessend vermerkt und angezeigt, sowohl auf dieser Seite, wie auch auf der Landingpage, wenn man den Reminder gespeichert hat. Beim Klicken auf eine der beiden Zahlen erscheint das unten aufgeführte Zahlenrad. Mit dessen Hilfe wird in einem ersten Schritt die Stunde ausgewählt, also eine Zahl zwischen 1 – 24 und direkt im Anschluss die Minuten zwischen 1-60. Auch diese werden direkt persistiert. Nur beim Klicken von «Save» wird der Reminder auch im Local Storage des Mobiltelefons abgespeichert. Sollte die Applikation geschlossen werden, oder entweder auf «cancel» oder den Pfeil zurück geklickt werden, wird der Reminder nicht gemerkt.

Beim Anpassen eines bestehenden Reminders wird man indirekt auf die gleiche Seite geführt, wie wenn man einen neuen erstellen würde, die jeweiligen Werte sind aber schon ausgefüllt mit dem das bereits vorhanden war.

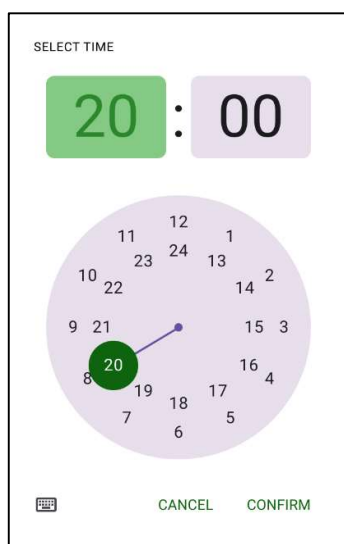


Abbildung 4 Mockup: Dialogfenster für die Zeiteinstellung

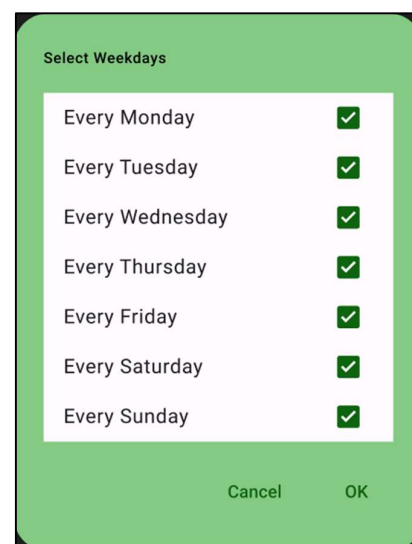


Abbildung 3 Mockup: Dialogfenster für Auswahl der Wochentage

3.2 EFFEKTIVES DESIGN

LANDINGPAGE



Abbildung 5 Startseite unserer App in Android

Bei der Landingpage konnten wir uns eins zu eins an unsere Vorlage halten. Dabei ist zu erwähnen, dass mit der Implementation der Navigation zwischen den einzelnen Seiten uns von der Komponente direkt ein Header gegeben wird, welchem man sehr simpel den korrekten Titel geben kann. Dies war die grösste Veränderung des ursprünglichen Mockups nachdem wir bereits mit dem Programmieren begonnen haben. Dort haben wir uns einfach an dem orientiert, was uns zur Verfügung gestellt wurde. Abgesehen davon erkennt man, dass auf Grund einer leicht anderen Variante von Android in diesem Falle die untere Leiste anders aussieht, wobei dies keine Änderung an unserer Applikation voraussetzt. Eine weitere Änderung war die Implementation eines eigenen Themes mit einem Farbschema, welches vom gegebenen abweicht. Da wir zu Beginn nicht wussten wie man das in Figma integrieren kann, erstellten wir unsere Vorlage mit den Farben die uns gegeben wurden. Dies konnte letztendlich doch auch noch auf beiden Seiten angepasst werden.

NEW REMINDER PAGE / EDITPAGE

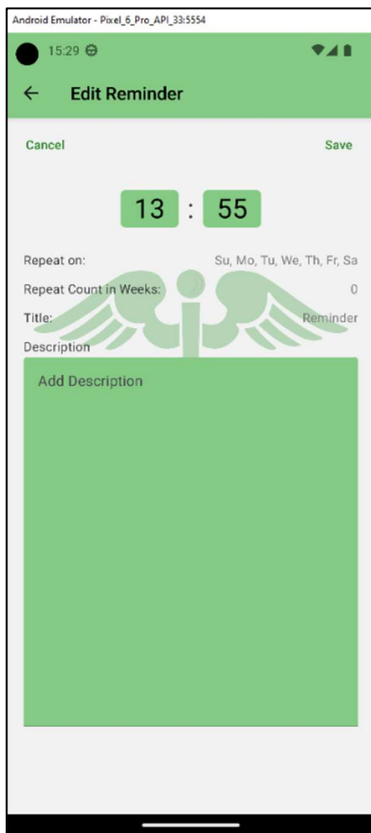


Abbildung 6 Bearbeiten eines Reminders in der App

Auch bei der New Reminder beziehungsweise der Editpage konnten wir uns ziemlich genau an die Vorlage halten. Wir würden sogar so weit gehen um zu sagen, dass wir alles genau so implementiert haben, wie wir es selbst designt haben. In diesen Seiten und den dazugehörigen Pop-ups mussten wir an unserer ursprünglichen Idee nichts verändern. Auch wenn es viel Arbeit und Zeit gekostet hat um alles so genau wie möglich darzustellen haben wir nicht das Ziel aus den Augen verloren. Dafür sind wir umso zufriedener mit dem Endergebnis.

Dialogfenster:

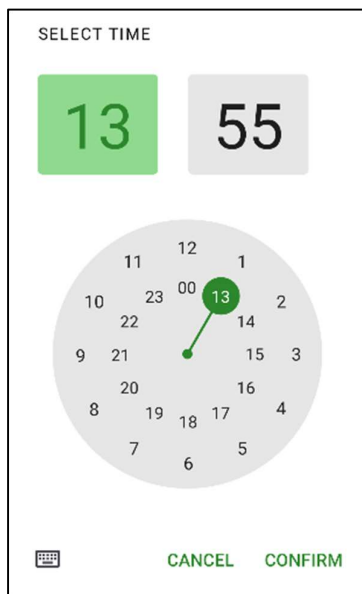


Abbildung 8 Zeiteinstellung in der App

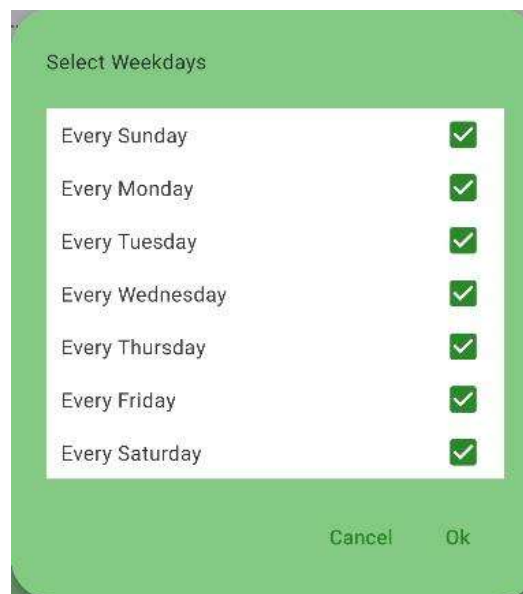


Abbildung 7 Auswahl der Wochentage in der App

3.3 ARCHITEKTUR, FRAMEWORKS & CODING GUIDELINES

Im Gegensatz zu der üblichen Drei Lagen Struktur, welcher wir bisher verwendet haben, wird in dieser Applikation alles im Frontend gelöst. Das heisst Datenmanagement und Service muss auf eine andere Art und Weise gelöst werden. Anstelle der Datenbank, übernehmen wir das Zepter über den lokalen Speicher des Telefons und sprechen direkt von unserem Service im Frontend diesen Speicher an. Dabei verwenden wir ein «async storage system», bei welchem wir anhand eines Key-Value Paars alle Informationen in einem einzelnen String abspeichern.

Damit wir all diese Funktionalitäten brauchen können, kommen verschiedene Frameworks zum Zug. Beginnend mit react-native und dem dazugehörigen react-native-paper. Diese beiden sind die Grundpfeiler, auf denen unsere Applikation liegt. Sie geben uns alle grundsätzlichen Komponenten, wie Knöpfe oder Karten, wie dies bspw. auch MUI macht. Zusätzlich versuchten wir mit Hilfe von i18n eine Übersetzung einzubauen, aber das hat wie bereits erwähnt leider nicht funktioniert. Dazu kommt noch die Einbindung von notifee, welche auch nicht hundertprozentig funktioniert.

Da die Zeit in einem üK grundsätzlich knapp ist, haben wir uns von Beginn an entschieden, keine eigenen Coding Guidelines zu verfassen und zu verfolgen. Dazu kommt noch, dass die allgemeinen Style Guides, wie bspw. von Google für Typescript, vorhanden und weit verbreitet sind. Aber wir haben uns auch an die SonarLint Meldungen gehalten, welche dem Schema der Noser Young folgen.

4. TECHNISCHER BERICHT

4.1 TESTING

TEILNEHMER DES TEAMS

Ngoc-Phuc Nguyen, Gian-Luca Kunfermann

ZIEL DER TESTS

Ziel des manuellen Testing ist es, herauszufinden ob alle Funktionalitäten der Applikation gewährleistet sind. Dabei werden die bereits erwähnten User Stories durchgespielt und noch zusätzliche Aspekte genauer beleuchtet. Damit soll eine grösstmögliche Abdeckung gewährleistet werden.

ART DER TESTS

In diesem Falle werden nur Whitebox Tests durchgeführt da die Tests von uns durchgeführt wurden und wir selbstverständlich mit dem Code bekannt sind.

TESTVORGABEN

Das Programm wird entweder in einem Emulator auf dem Computer oder über das Mobiltelefon gestartet. Anschliessend soll nach den jeweiligen Vorgaben der einzelnen Tests vorgegangen werden. Laut Vorgaben die im Dokument «335_erwartungen-dokumentation_v2.docx»

TESTVALIDIERUNG

Projektname	Medizin Reminder Applikation
Ansprechpartner	Gianluca Daffré
Autor des Testprotokolls	Gian-Luca Kunfermann
Tester	Ngoc-Phuc Nguyen, Gian-Luca Kunfermann
Test Datum	22.12.2022

TESTPROTOKOLL

Testcase ID	Applikationsnutzung_Happy_1
Komponenten	FAB, Timepicker, Texteingabe, Checkbox
Priorität	10/10
Beschreibung / Test Zusammenfassung	Verifizierung, dass ein Benutzer einen neuen Reminder erstellen kann, welcher auf der Hauptseite angezeigt wird.
Nötige Vorgaben	Die Applikation läuft lokal auf einem Emulator oder auf einem Android Mobil Telefon
Testschritte	<ol style="list-style-type: none"> 1. Klicken Sie auf den FAB (Plus Knopf am unteren Rand der Applikation) 2. Nachdem Sie auf die neue Seite weitergeleitet worden sind, soll bei der Auswahl der Tage der Montag ausgewählt werden.

	3. Anschliessende soll die gewünschte Häufigkeit (Anzahl Wiederholungen) auf drei gesetzt werden. 4. Geben Sie dem Reminder den Titel «Test» und die Beschreibung «Ibuprofen, 500 mg» 5. Klicken sie auf den «save» Knopf in der oberen rechten Ecke
Erwartetes Resultat	Auf der Übersichtsseite findet sich der erstellte Reminder mit den eingegebenen Informationen vor, mit einem Switch der eingeschaltet ist.
Effektives Resultat	''
Status	Bestanden
Test durchgeführt von	Gian-Luca Kunfermann

Testcase ID	Applikationsnutzung_Error_1
Komponenten	FAB, Timepicker, Texteingabe, Checkbox
Priorität	8/10
Beschreibung / Test Zusammenfassung	Verifizierung, dass ein Benutzer bei der Eingabe der Anzahl Wiederholungen keine Buchstaben und nur Zahlen eingeben kann.
Nötige Vorgaben	Die Applikation läuft lokal auf einem Emulator oder auf einem Android Mobil Telefon, ein Reminder wurde bereits anhand dem Test «Applikationsnutzung_Happy_1» erstellt.
Testschritte	1. Klicken/Drücken Sie auf den bereits erstellten Reminder. 2. Wählen Sie «Repeat Count in Weeks» aus. 3. Versuchen Sie anstelle einer Zahl, den Buchstaben a einzugeben und zu speichern.
Erwartetes Resultat	Es ist dem Benutzer nicht möglich etwas anderes als eine Zahl einzufügen.
Effektives Resultat	''
Status	Bestanden
Test durchgeführt von	Ngoc-Phuc Nguyen

Testcase ID	Applikationsnutzung_Error_2
Komponenten	FAB, Timepicker, Texteingabe, Checkbox
Priorität	7/10
Beschreibung / Test Zusammenfassung	Verifizierung, dass ein Benutzer auch ohne Eingabe einen neuen Reminder speichern kann. Dem User ist es freigestellt, ob er zusätzliche Informationen zu

	seinem Reminder festlegen möchte, oder ob ihm die Uhrzeit ausreicht.
Nötige Vorgaben	Die Applikation läuft lokal auf einem Emulator oder auf einem Android Mobil Telefon.
Testschritte	<ol style="list-style-type: none">1. Klicken Sie auf den FAB (Plus Knopf am unteren Rand der Applikation) .2. Drücken Sie auf die Zeiteingabe und geben Sie die Uhrzeit 08:00 ein.3. Anschliessend drücken Sie auf «save».
Erwartetes Resultat	Der Reminder wird ohne Titel, ohne Titel und ohne Wiederholung erstellt, wird demzufolge nur einmal ausgeführt.
Effektives Resultat	''
Status	Bestanden
Test durchgeführt von	Ngoc-Phuc Nguyen

5. FAZIT

5.1 GIAN-LUCA KUNFERMANN

Zu Beginn des üKs war die Vorfreude gross. Die Arbeit an einer Mobilen Applikation stellte ich mir als erfüllend und spannend vor. Leider war es aber wie oftmals in den üKs das die Zeit aufgrund unerwarteter Begebenheiten wieder knapp war, um alles korrekt zu implementieren. Das hat uns nicht nur Zeit und Nerven gekostet, sondern auch Funktionalitäten die wir gerne eingebaut hätten. Beginnend mit der Übersetzung. Leider konnten wir die Übersetzung, auch wenn wir dies probiert haben, nicht integrieren, da es, wie wir vermuten, mit unserem «Time Picker» interferiert und so nicht eingebaut bzw. gebraucht werden kann. Trotzdem verlasse ich den üK mit einem positiven Gefühl, auch wenn die negativen Aspekte das Ganze getrübt haben. Hätten wir anstelle des Zeitdrucks unlimited Zeit gehabt, wären diese Probleme auch nicht ansatzweise so schlimm gewesen, da die Challenge so ein Hindernis zu überwinden eigentlich sehr belohnend ist.

5.2 NGOC-PHUC NGUYEN

Es hat unheimlich viel Spass gemacht, diese App zu entwickeln. Für mich bedeutete dieser üK, dass wir einen weiteren Schritt Richtung Arbeitsweil gemacht haben. Es war undenkbar, dass ich eine App schreiben kann, die ich auf meinem eigenen Smartphone abspielen lassen konnte. Die unzähligen, schnellen Feedbacks beim Erstellen der App bereiteten mir stets Freude, ganz gleich, ob die Feedbacks positiv oder negativ war.

Leider dauerte der üK nur fünf Tage, daher mussten wir uns auf das Wesentliche fokussieren. Das bedeutete jedoch, dass wir nicht alle Funktionalitäten implementieren konnten, die wir uns gewünscht hatten. Die Problematik mit uns unbekannten Libraries bedeutete, dass wir nicht so schnell oder einfach die Fehler entdecken konnten, falls es zu Problemen bei der Implementierung gekommen ist. Sich in neue Libraries einzulesen, bedarf oft eine Menge Zeit. Nichtsdestotrotz bin ich sehr zufrieden mit der App, die wir abgeliefert hatten.

6. QUELLENVERZEICHNIS

Titelbild:

<https://pixy.org/154282/>

Libraries:

- react-native-keyboard-aware-scroll-view:
<https://www.npmjs.com/package/react-native-keyboard-aware-scroll-view>
- React Native:
<https://reactnative.dev/docs/typescript>
- React Native Paper
<https://callstack.github.io/react-native-paper/>
- Internationalization:
<https://www.i18next.com/>
- Notifee
<https://notifee.app/>