

ulm university universität **UU**

Fakultät für Ingenieurwissenschaften, Informatik und Psychologie Institut für Datenbanken und Informationssysteme

Bachelorarbeit

Im Studiengang Informatik

Konzeption und technische Realisierung einer mobilen Feedback-App zur Unterstützung schwangerer Frauen am Beispiel des Android Betriebssystems

Verfasser: Fabian Haug

Matrikelnummer: 843435

1. Prüfer: Prof. Dr. Manfred Reichert

Betreuer: Dr. Rüdiger Pryss

September 2017

Kurzfassung

Immer mehr Menschen nutzen Apps auf ihren Smartphones nicht nur um Spiele zu spielen oder für diverse Kommunikations- und Social-Media-Dienste. Es kommen immer mehr Apps auf den Markt, um bestimmte Alltagssituationen zu erleichtern oder auch zur Förderung der Gesundheit. Diese Arbeit soll zeigen, wie eine solche App zur Unterstützung und Begleitung von Schwangeren während ihrer Schwangerschaft entwickelt werden kann.

Jede Schwangerschaft ist eine besondere Phase im Leben einer Frau. Dabei hat sie mit vielen neuen Situationen zu kämpfen und muss mit vielen körperlichen Veränderungen und Umstellungen zurechtkommen. Die heutige Forschung zeigt, dass bestimmte psychosoziale Einflüsse, wie zum Beispiel Lebenssituation, Stress oder Beziehungsprobleme sich auf die Schwangerschaft und die Entwicklung des Kindes auswirken können.

In dieser Arbeit wird gezeigt, wie es möglich ist diese verschiedenen Umstände im Leben einer Schwangeren zu erfassen und ihr dann konstruktives Feedback zu geben für eine positive Beeinflussung ihrer Schwangerschaft.

Dabei wird gezeigt, wie eine solche App architektonisch zu realisieren ist und es wird auf verschiedene Bereiche der Implementierung genauer eingegangen, wie z.B. die Benachrichtigungen für das Beantworten der Fragebögen.

<u>Inhaltsverzeichnis</u>

1.	Einleitung	5
	1.1 Überblick	5
	1.2 Motivation	5
	1.3 Vision	5
	1.4 Projektkontext	5
	1.5 Aufbau der Arbeit	6
2.	Existierende Projekte	7
	2.1 MyKind	7
	2.2 Track Your Tinnitus	7
	2.3 Assess Your Stress	7
	2.4 Weitere Android Apps	8
3. /	Anforderungen	9
	3.1 Funktionale Anforderungen	9
4. /	Architektur	11
	4.1 Architekturübersicht	11
	4.2 Genereller Ablauf	11
	4.3 Datenstruktur	15
	4.4 Architektur der App	17
	4.4.1 Model-View-Controller	17
	4.4.2 Model	19
	4.4.3 ListAdapter, Receiver und Services	20
5. ՝	Vorstellung des Track Your Pregnancy Rahmenwerks	22
	5.1 Vorstellung der App	22
	5.1.1 Anmeldung mit Benutzername und Passwort	22
	5.1.2 Registrierung in der App	23
	5.1.3 Passwort vergessen	23
	5.1.4 Reset Passwort	24
	5.1.5 Hauptmenü	25
	5.1.6 Fragebögen	25
	5.1.7 Fragebogen Struktur	26
	5.1.8 Studien	27
	5.1.9 Benachrichtigungseinstellungen	28
	5.1.10 Sprachauswahl	29
	5.1.11 Antwortsätze	30
	5.1.12 Ergebnisse	30

6. Implementierung	32
6.1 Benachrichtigungen	32
6.2 Slider ohne initialen Wert	35
7. Anforderungsabgleich	37
7.1 Funktionale Anforderungen	37
8. Fazit	38
8.1 Zusammenfassung	
8.2 Ausblick	
8.2.1 Verbesserung der Visualisierung der Ergebnisse	
8.2.2 Verbesserung am User-Interface und Hilfe	38
<u>Abbildungsverzeichnis</u>	
Abbildung 1: Ablaufdiagramm	12
Abbildung 2: Studien bearbeiten	13
Abbildung 3: Fragebogen beantworten	14
Abbildung 4: Ergebnisse abrufen	15
Abbildung 5: Lokale Datenbank	16
Abbildung 6: Model-View-Controller	17
Abbildung 7: Model	19
Abbildung 8: ListAdapter, Receiver und Services	20
Abbildung 9: Login Ansicht	22
Abbildung 10: Registrierung Ansicht	23
Abbildung 11: Passwort vergessen Ansicht	24
Abbildung 12: Reset Passwort Ansicht	24
Abbildung 13: Hauptmenü Ansicht	25
Abbildung 14: Fragebögen Ansicht	26
Abbildung 15: Fragebogen Struktur Ansicht	27
Abbildung 16: Studien Ansicht und Code Dialog	28
Abbildung 17: Benachrichtigungseinstellungen Ansicht und Bearbeitungsdialog	
Abbildung 18: Sprachauswahl Ansicht	
Abbildung 19: Antwortsätze Ansicht	30
Abbildung 20: Ergebnisse Ansicht	31
Abbildung 21: Benachrichtigungen	32

Listings

Listing 1: MyNotificationManager: GetNotificationTime	. 33
Listing 2: MyNotificationManager: scheduleNotification und getPendingIntent	. 34
Listing 3: NotificationPublisher: onReceive	. 35
Listing 4: Input Slider	. 36

1. Einleitung

1.1 Überblick

Es soll eine mobile Anwendung für Android Geräte erstellt werden, mit der Nutzer (Schwangere) durch das Ausfüllen eines einmaligen demographischen und eines regelmäßig wiederkehrenden und mehrmals zu beantwortenden kontinuierlichen Fragenbogens ihre Schwangerschaft dokumentieren können und durch die Teilnahme an einer Studie auch Rückmeldungen zu ihren Antworten erhalten sollen. Dies ist ein Versuch der Schwangeren grundlegende Tipps zu geben und Sie bei ihrer Schwangerschaft zu unterstützen.

1.2 Motivation

Dieses Softwareprojekt ist Teil meiner Bachelorarbeit an der Uni Ulm. Die derzeitig vorhandene App (MyKind) ist nicht mehr für alle neuen Betriebssystemversionen kompatibel und soll so wieder aufgefrischt bzw. neu aufgebaut, umstrukturiert und erweitert werden.

1.3 Vision

Im Idealfall ist die Anwendung einfach aufgebaut und benutzerfreundlich zu bedienen. Mit dieser Anwendung soll der Nutzer leicht verständlich, benutzerfreundlich und am besten in seiner ausgewählten Sprache die Fragebögen ausfüllen können und ein Feedback zu seinen Angaben bekommen. Dadurch wird der Nutzer motiviert die Fragebögen regelmäßig auszufüllen, was zu einer aussagekräftigeren Studie führt. Des Weiteren wäre eine Freigabe des Nutzers durch einen Arzt wünschenswert, da so auch Probleme mit dem Arzneimittelgesetz gelöst wären.

1.4 Projektkontext

In der heutigen Zeit herrscht gerade in ländlichen Gebieten ein Mangel an Ärzten [1]. Somit ist der regelmäßige Besuch beim Frauenarzt dort manchmal sehr umständlich. Diese Anwendung soll dabei helfen, dass bei kleineren Fragen und Bedenken der Schwangeren der Weg zum Arzt nicht unbedingt erforderlich ist. Ein weiterer Aspekt ist auch, dass die App auf mögliche Risikofaktoren aufmerksam machen soll, die möglicherweise nicht für jeden Benutzer offensichtlich sind. Allerdings kann sie auf keinen Fall die regelmäßigen Untersuchungen durch einen Arzt ersetzen.

1.5 Aufbau der Arbeit

Dieser Abschnitt stellt eine Übersicht der Struktur dieser Arbeit dar. Diese Arbeit besteht aus acht Kapiteln. Nach der Einleitung folgt ein kurzer Überblick über bereits existierende "Tracking-Projekte" und Apps, die eine ähnliche Funktionalität wie die App des Track Your Pregnancy Projektes haben. Darauf folgt in Kapitel 3 die Darstellung der Anforderungen an das Track Your Pregnancy Projekt. Danach wird in Kapitel 4 die Architektur des Projektes beschrieben. Dabei wird eine generelle Übersicht der Architektur gegeben und die Datenstruktur und der allgemeine Aufbau der App erklärt. In Kapitel 5 wird ein kleiner Einblick in die App gegeben und die Möglichkeiten und Funktionen, die die App dem Benutzer bietet, gezeigt und beschrieben. Im Anschluss folgen in Kapitel 6 Einblicke in die Implementierung der App, die zur Erfüllung der Anforderungen beitragen. Daraufhin werden in Kapitel 7 die Anforderungen mit dem Stand der Entwicklung abgeglichen und in Kapitel 8 folgt als Abschluss eine Zusammenfassung und ein Ausblick auf die mögliche Weiterentwicklung des Projekts.

2. Existierende Projekte

2.1 MyKind

Die aktuell im GooglePlay Store verfügbare App MyKind liefert für das Track Your Pregnancy Projekt die Grundidee, eine Frau während ihrer Schwangerschaft mit Hilfe einer App zu begleiten und zu beraten. Dabei sollen ebenfalls durch die Beantwortung eines wiederkehrenden Fragebogens (KINDEX [2] [3]) die Veränderungen der Befindlichkeit, der Stimmung und des Umfeldes der Schwangeren beobachtet werden. Durch die systematische Erfassung von Belastungen und möglichen Risikofaktoren kann gezielt auf diese hingewiesen werden und die Schwangere erhält wertvolle Tipps für die positive Beeinflussung ihrer Schwangerschaft.

Die MyKind App ist ein Gemeinschaftsprojekt der Universität Ulm und der Universität Konstanz und ist im GooglePlay Store [4] und im Apple AppStore [5] erhältlich.

Allerdings ist es bei der MyKind App noch nicht möglich durch das Beantworten des Fragebogens bewusst an einer ausgewählten oder sogar vom Arzt vorgegebenen Studie teilzunehmen. Dies soll beim Track Your Pregnancy Projekt durch das Teilnehmen an Studien, die auch privat sein können und an denen nur durch ein Passwort teilgenommen werden kann, realisiert werden [6].

2.2 Track Your Tinnitus

Das Track Your Tinnitus Projekt¹ wurde bereits 2014 gestartet und versucht Menschen mit Tinnituserkrankung zu unterstützen und dient gleichzeitig zur weiteren Erforschung der Krankheit, speziell beim Verlauf der Schwankungen der Lautstärke und deren Ursachen. Menschen die an Tinnitus erkrankt sind, hören Töne oder Geräusche ohne physikalische Ursache. Dabei kommt es bei 60% der Patienten zu Schwankungen der Lautstärke dieser Geräusche. Die App ist darauf ausgerichtet, dass der Benutzer immer wieder, zu unterschiedlichen Zeiten, die Lautstärke des Tinnitus mit Hilfe der App dokumentiert. Dadurch ist es möglich einen detaillierten zeitlichen Ablauf der Lautstärke nachzuvollziehen und auch dem Benutzer zu visualisieren. Des Weiteren ist es auch möglich diese Schwankungen mit verschiedenen Alltagssituationen in Verbindung zu bringen und so zu überprüfen, wie sich z.B. Stress oder die Umgebungslautstärke auf den Tinnitus auswirken. Das Track Your Tinnitus Projekt ist im Rahmen einer Diplomarbeit von Jochen Herrmann an der Universität Ulm 2014 [7] entstanden und wird derzeit durch weitere Abschlussarbeiten weiterentwickelt. Die App für das Track Your Tinnitus Projekt ist im GooglePlay Store [4] und im Apple AppStore [5] verfügbar.

2.3 Assess Your Stress

Das Assess Your Stress Projekt ist ebenfalls ein Projekt an der Universität Ulm [8]. Dabei soll das schon früher erstellte Track Your Tinnitus Projekt überarbeitet und neu aufgebaut werden. Des Weiteren soll es nicht auf Tinnitus Symptome fixiert bleiben, sondern konzentriert sich jetzt auf die Messung des Stresslevels des Benutzers und die Ursachen dafür. Dieses Projekt wurde ins Leben gerufen nachdem der Deutsche Bundestag das Präventionsgesetz [9], das zur Stärkung der Gesundheitsförderung und der Prävention führen soll, verabschiedet hat.

Das Projekt wird derzeit in mehreren Abschlussarbeiten an der Universität Ulm entwickelt und ist im Moment noch nicht in den gängigen AppStores verfügbar.

¹ Vgl. und für weitere Infos [34] [35] [36] [37] [38] [39] [40] [41]

2.4 Weitere Android Apps

Es gibt im GooglePlay Store [4] auch noch weitere Apps, die sich mit dem Thema Schwangerschaft beschäftigen. So sind als Top Suchergebnisse die Apps "Schwangerschaft +" [10], "Pregnancy Week By Week" [11] und "Meine Schwangerschaft & Baby" [12] zu nennen.

Dabei liegt der Hauptaspekt bei diesen Apps aber nicht in der Bewertung der Lebenssituation der Schwangeren und deren Risikofaktoren für das Baby, sondern das Hauptaugenmerk liegt hier in der Aufzeichnung von Bildern und Gewicht des Babys oder auch der Organisation von Arztterminen. Im Unterschied zum Track Your Pregnancy Projekt, das eher auf dem medizinischen Aspekt der Schwangerschaft konzentriert ist, werden bei den meisten im GooglePlay Store angebotenen Apps, eher Lifestyle Aspekte in den Vordergrund gerückt.

3. Anforderungen

Dieses Kapitel definiert die Anforderungen an das Track Your Pregnancy Projekt. Dabei sind diese Anforderungen in funktionale und nichtfunktionale Anforderungen unterteilt.

3.1 Funktionale Anforderungen

Dieser Abschnitt zeigt die funktionalen Anforderungen an die App. Dabei werden die wichtigsten Funktionen der App gezeigt, die diese einem Benutzer bieten sollte. Die folgende Tabelle zeigt eine Aufstellung der funktionalen Anforderungen.

Nr.	Beschreibung	Problembeschreibung
1.	Registrierung in der App	Die Benutzung der App kann ohne Benutzerkonto
		nicht erfolgen. Es sollte daher möglich sein, direkt auf
		dem Endgerät ein solches Benutzerkonto zu erstellen.
2.	App auch ohne	Eine funktionierende Internetverbindung auf dem
	Internetverbindung bedienen.	Smartphone sollte keine Voraussetzung für das
		Benutzen der App sein, da ein Benutzer evtl. nur
		schlechten oder gar keinen Empfang hat. Eventuell
		gespeicherte Werte werden anschließend bei
		funktionierender Internetverbindung im Hintergrund
		übermittelt.
	St	udienteilnahme
3.	An verschiedenen Studien	Ein Benutzer sollte die Möglichkeit haben, sich in
	teilnehmen	mehrere Studien einzuschreiben und deren
		Fragebögen auszufüllen. Ein Benutzer soll die
		Studienteilnahme zudem jederzeit beenden können.
4.	Studieneinladungen	Der Benutzer soll die Möglichkeit haben, persönliche
	annehmen	Einladungen zur Studienteilnahme in der App
		anzusehen und diese direkt anzunehmen.
5.	An einer privaten Studie	Der Benutzer kann sich per App in eine private Studie
	teilnehmen	einschreiben, sofern er das Passwort kennt.
		Fragebögen
6.	Statistische Fragebögen	Für Studienteilnehmer können durch bestimmte
	innerhalb der	Benutzer (Studienleitung) beliebige Fragebögen
	Studienteilnahme ausfüllen	erstellt werden, um das jeweilige Studienziel zu
		erreichen. Die Fragebögen werden dem
		Studienteilnehmer in der App angezeigt.
7.	Fragebögen deaktivieren	Deaktivierte Fragebögen können von den Benutzern
		nicht ausgefüllt werden.
8.	Zustand eines Fragebogens	Fragebögen sollen je nach Konfiguration vom
	ändern	Benutzer einmal oder mehrmals (wiederholt)
		ausgefüllt werden können. Wurde ein einmaliger
		Fragebogen bereits ausgefüllt, darf er dem Benutzer
		nicht noch einmal präsentiert werden.
9.	Ergebnisse synchronisieren	Zur Visualisierung der Ergebnisse aus den Fragebögen
		und für Forschungszwecke, sollten die Ergebnisse aus
		den Apps an den Server übertragen werden.

10.	Slider ohne initialen Wert	Ein Benutzer lässt sich beim Ausfüllen eines Fragebogens davon beeinflussen, welcher Wert voreingestellt ist. Daher darf der Slider in einem Fragebogen in der Track Your Pregnancy App keinen initialen Wert haben.
11.	An auszufüllende Fragebögen erinnern	Ein Benutzer sollte von der App benachrichtigt werden, wenn ein neuer Fragebogen auf Basis eines hinterlegten Terminplans ausgefüllt werden soll.
12.	Erinnerungsfunktion anpassen	Ein Benutzer sollte die Benachrichtigungshäufigkeit und den Zeitraum der Benachrichtigungen in der App frei einstellen können, wenn dies vom Studienleiter für den Fragebogen vorgesehen ist.
13.	Ergebnisse in der App anzeigen	Um die zeitliche Entwicklung in der App direkt anzeigen zu können, sollten die Ergebnisse aus dem Fragebogen visualisiert werden.

4. Architektur

Dieses Kapitel beschreibt die Architektur des Track Your Pregnancy Projekts. Zuerst wird eine kurze Übersicht über die Architektur gegeben (Kapitel 4.1). In Kapitel 4.2 wird ein typischer Ablauf gezeigt, gefolgt von der detaillierten Beschreibung der Datenstruktur in Kapitel 4.3. Zum Abschluss wird in Kapitel 4.4 die Architektur der App erklärt.

4.1 Architekturübersicht

Track Your Pregnancy besteht aus zwei Komponenten: Server und App für ein Android Smartphone. Der Server, sowie die API wurden vom Institut für Datenbanken und Informationssysteme bereitgestellt. Die App wurde nativ mit Java auf Android entwickelt. Die App greift nicht direkt auf die Daten auf dem Server zu. Jegliche Kommunikation zwischen App und Server läuft über eine REST-ähnliche JSON-API.

4.2 Genereller Ablauf

Die Benutzung der Track Your Pregnancy App setzt ein Benutzerkonto voraus. Die App bietet einem Benutzer die Möglichkeit sich in der App ein solches Benutzerkonto zu erstellen. Die folgenden Abbildungen zeigen den typischen Ablaufprozess bei Benutzung der App. Bei erstmaliger Öffnung der App muss der Benutzer sich zuerst registrieren. Dabei muss er Daten wie E-Mail Adresse, Benutzername und Passwort festlegen. Bei erfolgreicher Registrierung erhält der Benutzer eine E-Mail zur Bestätigung und Verifizierung seiner E-Mail Adresse. Durch das Klicken auf den in der E-Mail enthaltenen Link aktiviert er sein Benutzerkonto und kann sich ab diesem Moment in der App einloggen. Nach dem Einloggen kann der Benutzer über das Menü zu verschiedenen Bereichen navigieren. Diese sind die Verwaltung der Studien, seine Fragebögen, die Benachrichtigungseinstellungen, die Ergebnisansicht, die Spracheinstellung und durch Ausloggen wieder der Loginbereich.

Im Bereich der Studienverwaltung kann der Nutzer alle aktiven Studien einsehen und gleichzeitig wird ihm angezeigt, ob er schon an einer Studie teilnimmt, eingeladen ist oder für eine Teilnahme angefragt hat. Im Bereich seiner Fragebögen werden dem Benutzer die Fragebögen angezeigt, die zu seinen aktuell ausgewählten Studien gehören. Dort kann er dann den gewünschten Fragebogen auswählen und beantworten. Dabei werden ihm zunächst, wenn vorhanden und noch nicht beantwortet, nur statistische Fragebögen angezeigt, da diese vor den kontinuierlichen Fragebögen beantwortet werden sollen. Bei den Benachrichtigungseinstellungen werden dem Benutzer die Zeitpläne der Benachrichtigungen von Fragebögen angezeigt, sofern diese auch verändert werden dürfen. Diese kann der Benutzer dann beliebig verändern. In der Ergebnisansicht kann der Benutzer alle seine bis jetzt an den Server gesendeten Antwortsätze sehen, für die auch schon ein Ergebnisset vorhanden ist. Durch Auswahl eines Antwortsatzes wird dem Benutzer dann das erstellte Feedback angezeigt.

In der Spracheinstellung werden dem Benutzer die vom Server bereitgestellten Sprachen angezeigt und er kann wählen in welcher Sprache die App angezeigt werden soll.

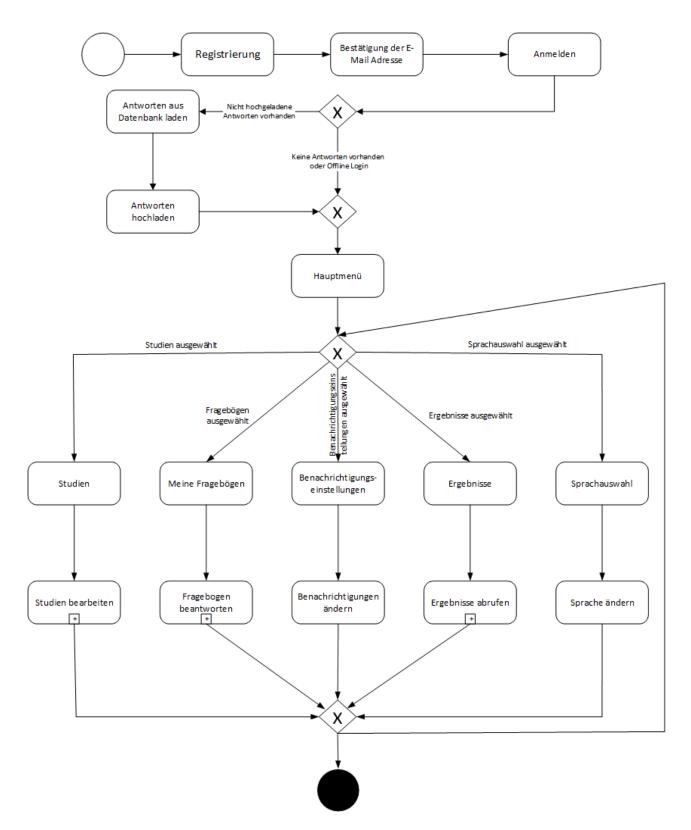


Abbildung 1: Ablaufdiagramm

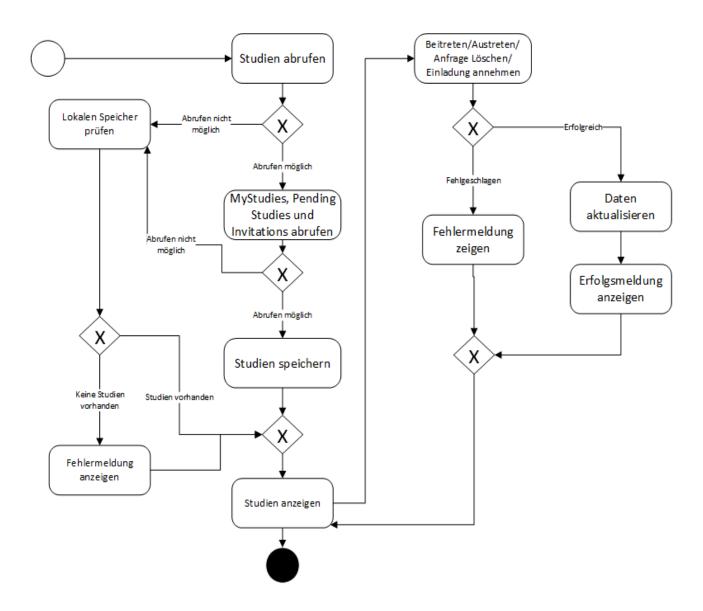


Abbildung 2: Studien bearbeiten

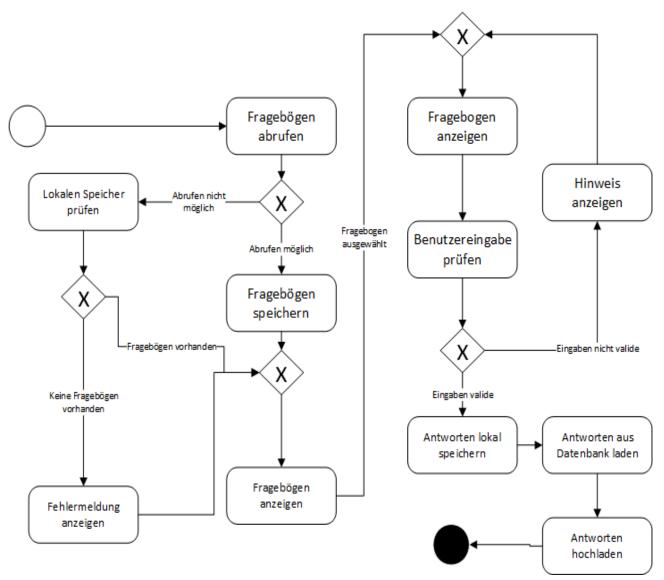


Abbildung 3: Fragebogen beantworten

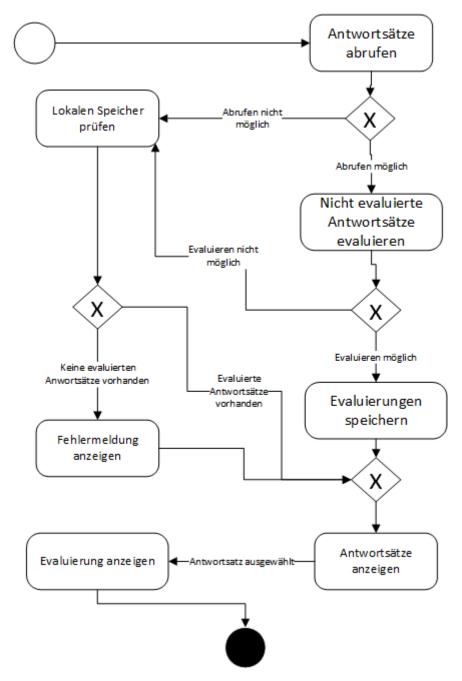


Abbildung 4: Ergebnisse abrufen

4.3 Datenstruktur

Die Datenstruktur dieser App basiert auf dem durch die vom Server und der API vorgegeben Datenmodell. Die genauere Nutzung dieses Datenmodells und ihre Umsetzung in der App, folgt in Kapitel 4.4. Hier wird nur eine Beschreibung der lokalen Datenbank gegeben, die nicht das komplette Datenmodell auf dem Server spiegelt, sondern nur den Offline-Betrieb sicherstellen soll. Dadurch wird das vorgegebene Datenmodell vereinfacht und verkleinert. Die folgende Abbildung zeigt den Aufbau der in der App implementierten SQLite Datenbank.

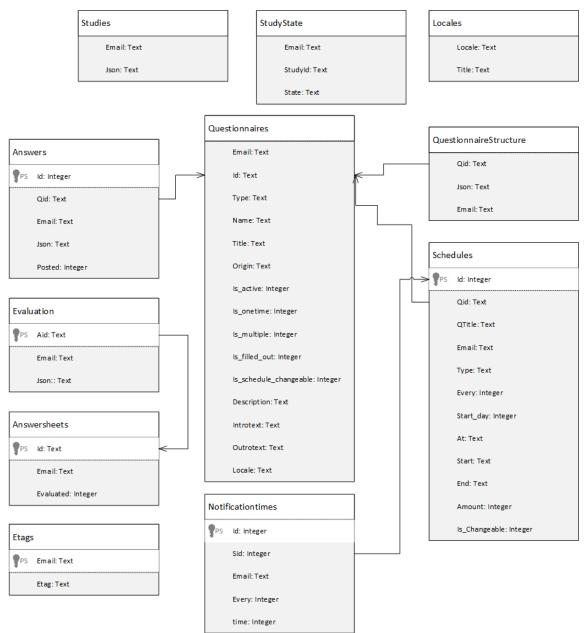


Abbildung 5: Lokale Datenbank

Die Spalte Email referenziert überall den Benutzer und garantiert damit eine Offline-Nutzung auch von mehreren Benutzern auf dem gleichen Endgerät.

In der Tabelle *Studies* wird der vom Benutzer zuletzt geladene Json-String, der alle verfügbaren Studien enthält, gespeichert. Um die verschiedenen Status der Studien für den Benutzer abzuspeichern wird der Status mit Verweis auf Nutzer und ID der Studie in der *StudyState* Tabelle gespeichert.

Die Tabelle *Locales* sichert die bei der letzten Onlineverbindung verfügbaren Sprachen, um auch die Sprachauswahl im Offline Modus zu sichern. Die Texte für die Bedienelemente in den verschiedenen Sprachen werden dabei nicht in der Datenbank gespeichert, sondern in den *SharedPreferences* [13] der App.

Die Tabelle *Questionnaires* enthält alle Infos zu den Fragebögen, die vom Server bei der Abfrage der Fragebögen für den jeweiligen Nutzer kommen. Die Aufspaltung des Json-Strings in eine klassenähnliche Tabelle ist hier nötig, um bei dem Austreten aus einer Studie die Offline Daten konsistent halten zu können. Somit werden Fragebögen die zur Studie gehören automatisch gelöscht ohne eine erneute Anfrage an den Server, um die Fragebögen zu aktualisieren. Des Weiteren wird die

Tabelle um ein Locale ergänzt, um zu speichern, in welcher Sprache die Fragebögen für den Benutzer vorliegen.

Die *Schedules* Tabelle lagert dabei die in den Fragebögen enthaltenen Schedules aus. Da diese zum Teil auch verändert werden können und dabei nur lokal gespeichert werden sollen.

Die Benachrichtigungszeiten, die aus den Schedules erstellt werden und später dem *AlarmManager* [14] übergeben werden, werden in der Tabelle *Notificationtimes* gespeichert.

In der *QuestionnaireStructure* Tabelle wird lediglich der zum Fragebogen gehörende Json-String mit den Elementen des Fragebogens gespeichert.

Die Answers Tabelle enthält den zum Hochladen bereiten Json-String eines Antwortsets, das von einem Nutzer ausgefüllt wurde. Die Verweise auf die Id des Fragebogens und auf den Benutzer sind nötig, um alles richtig hochladen zu können. In der Posted Spalte wird vermerkt ob das Antwortset schon erfolgreich an den Server weitergeleitet wurde oder noch nicht.

Die *Answersheets* und *Evaluation* Tabellen sind für die Anzeige der Ergebnisse nötig. Dabei wird die Evaluated Spalte auch benötigt, um zu speichern ob ein Antwortset schon evaluiert wurde. In der *Etag* Tabelle werden die Etags für jeden Benutzer gespeichert die bei der letzten Abfrage der Fragebögen vom Server zurückgegeben wurden, um leichter eine Veränderung bzw. keine Veränderung der Fragebögen abfragen zu können.

4.4 Architektur der App

4.4.1 Model-View-Controller

Die folgende Abbildung zeigt den Aufbau der Track Your Pregnancy App nach dem Model View Controller Design. Pfeile stellen dabei eine Vererbung dar und "+" eine Referenz auf ein Objekt der anderen Klasse.

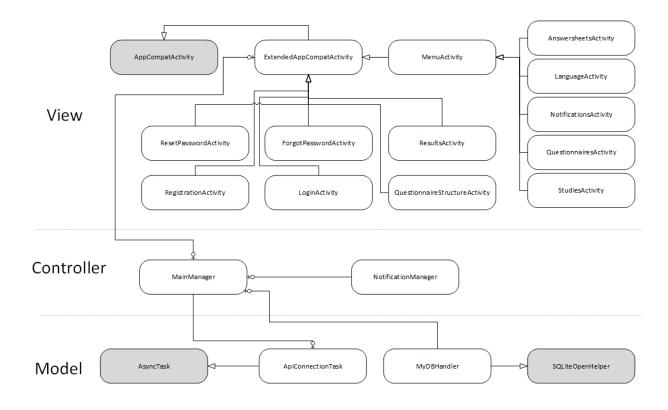


Abbildung 6: Model-View-Controller

Die Oberklasse für alle Views ist *AppCompatActivity* [15] bzw. die davon erbende von mir erweiterte *ExtendedAppCompatActivity*. Diese enthält die Funktionen, um die Views nach dem Ausführen von Lade-Operationen zu aktualisieren oder Rückmeldung zu geben. Um dies zu realisieren hat jede ExtendedAppCompatActivity einen *MainManager*, dem sie sich selber als View mitgibt. Alle Views die zusätzlich ein Menü erhalten sollen, um im Hauptmenübereich nach dem Einloggen navigieren zu können, erben von der *MenuActivity*, die dieses Menü zum Navigieren bereitstellt. Die Views bieten folgende Ansichten und Interaktionsmöglichkeiten:

LoginActivity: Einloggen, zur RegistrationActivity oder ForgotPasswordActivity navigieren

RegistrationActivity: Registrierung, zurück zum Login

ForgotPasswordActivity: Passwort-Reset-Email anfordern, zurück zum Login oder weiter zur Eingabe eines neuen Passworts.

ResetPasswortActivity: Neues Passwort festlegen

MenuActivity: Willkommensmeldung und Menü zum navigieren

AnswersheetActivity: Auflistung der abgegebenen Antwortsätze, die evaluiert wurden, zur Auswahl

ResultsActivity: Anzeige der Evaluierung des ausgewählten Antwortsatzes.

LanguageActivity: Auflistung der Sprachauswahl

NotificationsActivity: Auflistung der Benachrichtigungen mit Bearbeitungsdialog bei Auswahl

Questionnaires Activity: Auflistung der zu beantwortenden Fragebögen zur Auswahl

QuestionnaireStructureActivity: Anzeige des ausgewählten Fragebogens mit Ausfüll- und Absendemöglichkeit

StudiesActivity: Auflistung aller Studien mit Interaktionsmöglichkeit (Beitreten, Austreten, Anfragen)

Der MainManager fungiert für alle Views als Controller und verwaltet die Daten zur Laufzeit. So wird zum Beispiel der Anmelde-Token vom Server oder auch die Liste der Studien im MainManager als statische Variablen gehalten. In ihm sind Funktionen enthalten, die eine Kommunikation mit dem Server durch einen neuen ApiConnectionTask ausführen und die Antworten vom Server verarbeiten. Des Weiteren spricht er den MyDBHandler an und sorgt so für das Speichern und Laden in der lokalen Datenbank.

Der *MainManager* hat auch einen *NotificationManager*, um ihm Meldungen über geänderte Schedules (Benachrichtigungszeiten) mitzuteilen, die entweder durch den Nutzer über die View oder über den *ApiConnectionTask* vom Server eingegangen sind.

Der *NotificationManager* beinhaltet dann Funktionen durch die dem in Android verfügbaren *AlarmManager* [14] die Benachrichtigungszeitpunkte übergeben werden.

Der ApiConnectionTask erbt von der Klasse AsyncTask [16] und stellt eine variable Verbindung mit dem Server her. Er kann verschiedene Eingabevariablen (URL-Ende, http-Methode, Header, usw.) erhalten und so unterschiedlichste Verbindungen aufbauen, die zur Kommunikation mit dem Server und der API nötig sind.

Der MyDBHandler erbt vom SQLiteOpenHelper [17] und liefert alle Funktionen, die nötig sind, um die vom Server und dem Benutzer erhaltenen Daten zu speichern und wieder zu laden.

4.4.2 Model

Das Datenmodell wird in Klassen umgesetzt. Hierbei wird noch eine Unterteilung in Model und ViewModel gemacht, da manche Klassen lediglich für die Anzeige und Eingabe zuständig sind und mit der internen Datenhaltung nichts zu tun haben. Auch hier stehen Pfeile für Vererbung und "+" für eine Referenz auf ein Objekt der anderen Klasse.

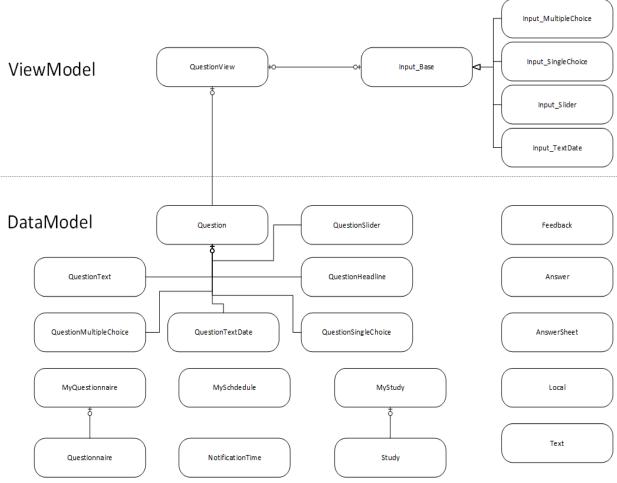


Abbildung 7: Model

Die Klasse *QuestionView* wird verwendet, um den Fragebogen anzuzeigen und ausfüllen zu können. In ihr werden während der Eingabe die Antworten gespeichert und sie enthält natürlich eine Frage und eine Eingabemöglichkeit (nur bei Headline und Text ist diese natürlich null). Diese Eingabemöglichkeit wird durch die Klasse *Input_Base* realisiert von der die Klassen *Input_MultipleChoice*, *Input_SingleChoice*, *Input_Slider* und *Input_TextDate* erben. Diese Klassen stellen View [18] Elemente zur Eingabe von bestimmten Fragetypen bereit. *Input_TextDate* kann hier je nach Fragetyp verschiedene Eingabearten für das Eingabefeld haben (z.B. ein Datum, Multiline oder nur einen String).

Die Klasse *Question* enthält als Object [19] Variable eine der verschiedenen Fragetypen, die sich in ihrem Aufbau unterscheiden. So wird aber ermöglicht, dass die unterschiedlichen Fragetypen doch in einer Liste gehalten werden. Der Aufbau der verschiedenen Fragetypenklassen (*QuestionHeadline*, QuestionText, *QuestionMultipleChoice*, *QuestionSingleChoice*, *QuestionSlider*, *QuestionTextDate*) richtet sich nach den JSON Objekten, die vom Server übermittelt werden. So können sie mit Hilfe von *Gson* [20] direkt aus dem JSON-String in Java-Objekte gewandelt werden.

So ist es auch bei den Klassen *Questionnaire* und *Study*. Die Extraklassen *MyQuestionnaire* und *MyStudy*, die jeweils ein *Study* bzw. einen *Questionnaire* enthalten, dienen nur dazu zusätzliche Extrainfos, die so nicht direkt vom Server kommen zu speichern (z.B. Status der Studie für den Benutzer).

Die Klasse *MySchedule* dient zur Verarbeitung der Benachrichtigungen, die für einen Fragebogen möglich sind. Die daraus entstehenden Benachrichtigungszeiten werden durch die Klasse *NotificationTime* verwaltet.

Die Klasse *Answer* beinhaltet alle Antworten von einem Benutzer für einen Fragebogen und dient zum Hochladen dieser Antworten an den Server.

Die Klasse *Answershee*t wird benötigt, um alle Antwortsätze, die der Benutzer an den Server übermittelt hat zu organisieren und deren Evaluation anzufragen oder dem Benutzer zu zeigen, dass sie schon evaluiert wurden und er die Ergebnisse ansehen kann.

Diese Ergebnisse werden über die Klasse *Feedback* verwaltet und dem Benutzer bei Bedarf angezeigt. Die Klassen *Local* und *Text* dienen dazu dem Nutzer eine Sprachauswahl anzubieten, die vom Server bereitgestellt wird. Dabei repräsentieren Objekte der Klasse *Local* die verschiedenen Sprachen und die Klasse *Text* wird genutzt, um einen einzelnen Text (z.B. für den Login Button) zu verwalten. Die eigentliche Speicherung dieser Texte findet durch *SharedPreferences* [13] statt.

4.4.3 ListAdapter, Receiver und Services

Die folgenden ListAdapter sind für *ListViews* [21] in den namentlich dazugehörigen Activitys, die schon in Abbildung 6 zu sehen waren. Die Klasse *JsonParser* wird hier auch noch genannt, ist aber lediglich eine Hilfsklasse, in der Funktionen enthalten sind, die entweder Informationen aus einem JSON-String entnehmen und zurückgeben oder Informationen in einen JSON-String verpacken, um die Informationen an den Server senden zu können.

Der *NotificationPublisher* ist eine Klasse, die von *BroadcastReceiver* [22] erbt und auf die vorher dem *AlarmManager* [14] übergebenen Broadcasts reagiert. Er erstellt dabei eine Benachrichtigung mit Text und Titel und meldet es dann dem Benutzer.

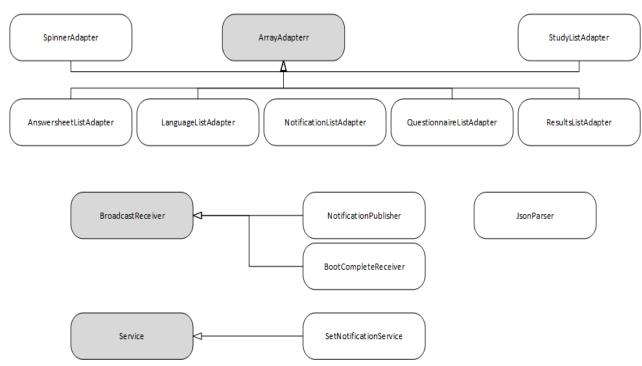


Abbildung 8: ListAdapter, Receiver und Services

Da die Zeiten im *AlarmManager* nach einem Neustart nicht mehr vorhanden sind wird auch ein *BootCompleteReceiver*, der ebenfalls von *BroadcastReceiver* erbt, benötigt, der bei einem Neustart des Gerätes aktiviert wird und einen von *Service* [23] erbenden *SetNotificationService* startet, der dann die in der Datenbank gespeicherten Benachrichtigungszeitpunkte im *AlarmManager* neu anlegt.

Die folgenden Adapter erben alle von ArrayAdapter [24].

Der LanguageListAdapter, QuestionnaireListAdapter und AnswersheetListAdapter sind dabei vom Aussehen identisch, bis auf den Unterschied, dass bei der Auswahl eines Elements beim QuestionnaireListAdapter und AnswersheetListAdapter die Activity gewechselt wird und der ausgewählte Fragebogen bzw. die Ergebnisse angezeigt werden. Beim LanguageListAdapter wird nach der Auswahl die Sprache auf die gewählte Sprache umgestellt und dem Benutzer auch angezeigt welche Sprache ausgewählt ist.

Der *StudyListAdapter* zeigt alle Studien an mit Titel und Beschreibungstext an. Daneben ist dann ein Button, an dem der Status der Studie für den Benutzer zu sehen ist und durch den er dann direkt agieren und z.B. der Studie beitreten kann.

Beim *ResultsListAdapter* werden für die Feedbacks für jedes Element ein Titel mit Beschreibungstext angezeigt.

Der NotificationListAdapter zeigt immer ein Element der MySchedule Klasse an und repräsentiert die Klassendefinition wie ein Schedule vom Server definiert ist. Neben jedem Element gibt es einen Edit Button, durch den dieser Schedule beliebig verändert werden kann. Dabei sind die Auswahlelemente für den Scheduletype und den Starttag mit Spinnern [25] realisiert. Für diese Spinner wird der SpinnerAdapter genutzt.

5. Vorstellung des Track Your Pregnancy Rahmenwerks

Dieses Kapitel stellt das Track Your Pregnancy Rahmenwerk aus Sicht eines Benutzers vor. Dabei werden die einzelnen Funktionen und Ansichten gezeigt, die dem Benutzer beim Ausführen der App bereitstehen und angezeigt werden.

5.1 Vorstellung der App

Die Teilnahme an Studien, das Ausfüllen von dazugehörigen Fragebögen und die Ansicht der Ergebnisse sind die Hauptfunktionen dieser App. Die App unterstützt jegliche Android Geräte ab *Android Version 4.0.3*, wurde aber für die Darstellung auf Smartphones optimiert.

5.1.1 Anmeldung mit Benutzername und Passwort

Ohne eine gültige Anmeldung mit Benutzername und Kennwort lässt sich die App nicht benutzen. Daher ist die Anmeldung (Abbildung 9) die erste Ansicht, mit der ein Benutzer interagieren kann. Ein Benutzer kann sich entweder direkt anmelden, oder erreicht über den Button "Registrieren" die Ansicht, in der er ein Benutzerkonto auf dem Server erstellen kann (Abbildung 10). Durch den Button "Passwort Vergessen?" gelangt der Benutzer zu einer Ansicht, um sein Passwort zurücksetzen zu können (Abbildung 11). Nach der Eingabe von Benutzername und Passwort meldet sich der Benutzer mit dem Button "Login" am Server an. Sollte das Gerät aktuell keine funktionierende Internetverbindung haben wird die Eingabe mit dem zuletzt eingeloggten Benutzer abgeglichen und bei Übereinstimmung wird ein Login im Offline-Modus durchgeführt. Wenn der Benutzername oder das Passwort falsch sein sollten, wird eine entsprechende Fehlermeldung ausgegeben. Nach einer erfolgreichen Online-Anmeldung werden automatisch noch vorhandene und nicht an den Server hochgeladene Antwortsätze des Benutzers an den Server hochgeladen. Allgemein folgt auf eine erfolgreiche Anmeldung die Weiterleitung zur Hauptmenü Ansicht (Abbildung 13) mit Willkommensmeldung.

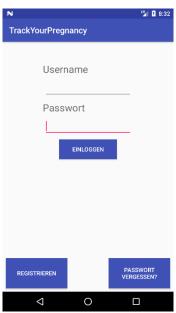


Abbildung 9: Login Ansicht

5.1.2 Registrierung in der App

Für die Registrierung (Abbildung 10) bei Track Your Pregnancy muss ein Benutzer einen Benutzernamen, seine E-Mail Adresse und ein Passwort angeben. Um falsche Eingaben zu verhindern muss das Passwort zwei Mal eingegeben werden. Der Button "Registrieren" überprüft, ob alle Daten eingegeben wurden und überträgt diese Daten an den Server. Bei erfolgreicher Übermittlung der Daten und korrekten Eingaben, wird eine Erfolgsmeldung angezeigt und eine E-Mail zur Bestätigung des Benutzerkontos wird an die E-Mail Adresse des Benutzers gesendet. Wenn Daten fehlen oder falsch eingegeben wurden, der Benutzername oder die E-Mail Adresse bereits registriert sind oder bei fehlender Internetverbindung, wird eine Fehlermeldung ausgegeben.

Durch das Betätigen des "Abbrechen" Buttons gelangt der Benutzer zurück zur Login Ansicht (Abbildung 9). Dies ist natürlich auch durch das Drücken der in jeden Android Gerät integrierten "Zurück"- Taste möglich.

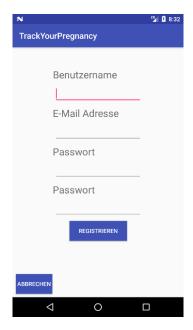


Abbildung 10: Registrierung Ansicht

5.1.3 Passwort vergessen

Falls ein Benutzer sein Passwort vergessen haben sollte besteht hier die Möglichkeit für ihn eine Anfrage für das Zurücksetzen seines Passworts an den Server zu senden (Abbildung 11). Durch das Eingeben seiner E-Mail Adresse und das Drücken des "Abschicken" Buttons wird diese Anfrage an den Server gesendet. Bei erfolgreicher Anfrage erscheint eine Erfolgsmeldung, dass eine E-Mail mit Anweisungen für den Passwort-Reset an seine E-Mail Adresse gesendet wurde. Bei der Eingabe einer nicht registrierten oder noch nicht verifizierten E-Mail Adresse, sowie fehlender Internetverbindung wird eine Fehlermeldung angezeigt.

Mit dem Button "Einloggen", sowie dem integrierten "Zurück"-Button von Android gelangt der Benutzer zurück zur Login Ansicht (Abbildung 9).

Durch das Betätigen des "Password Ändern" Buttons kommt der Benutzer zur Reset Passwort Ansicht (Abbildung 12), in der er dann sein neues Passwort eingeben und festlegen kann.

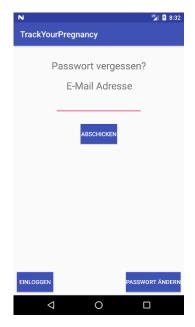


Abbildung 11: Passwort vergessen Ansicht

5.1.4 Reset Passwort

Nach der Passwort-Vergessen-Anfrage kann der Benutzer in der Reset Passwort Ansicht (Abbildung 12) sein neues Passwort festlegen. Dazu muss er den in der E-Mail enthaltenen Reset-Token eingeben und sein neues Passwort zur Sicherung vor Fehleingaben zwei Mal eingeben. Durch das Drücken des "Abschicken" Buttons werden die eingegebenen Daten an den Server gesendet. Bei Erfolg erscheint eine Erfolgsmeldung und der Benutzer kann sich ab diesem Moment mit dem neuen Passwort einloggen. Bei falschen Eingabedaten oder fehlender Internetverbindung wird eine Fehlermeldung angezeigt.

Durch den "Abbrechen" Button oder den integrierten "Zurück" Button kommt der Benutzer zurück zur Passwort Vergessen Ansicht (Abbildung 11).

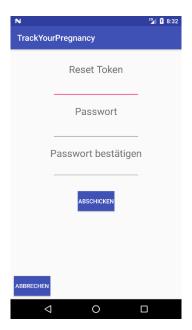


Abbildung 12: Reset Passwort Ansicht

5.1.5 Hauptmenü

Das Hauptmenü (Abbildung 13) dient dem Benutzer, um zu den verschiedenen Bereichen der App zu navigieren. Dieses Menü wird durch das Betätigen des Buttons in der rechten oberen Ecke des Bildschirms geöffnet und zeigt die verschiedenen Bereiche der App und zusätzlich einen "Abmelden" Button für den Logout an. Nach der Auswahl eines Menüpunktes wird das Menü wieder geschlossen und es wird zur ausgewählten Ansicht navigiert.

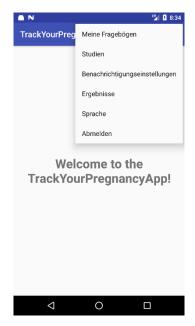


Abbildung 13: Hauptmenü Ansicht

5.1.6 Fragebögen

Die Track Your Pregnancy App bietet dem Benutzer die Möglichkeit an mehreren Studien gleichzeitig teilzunehmen und dadurch auch mehrere unterschiedliche Fragebögen beantworten zu können. In der Fragebögen Ansicht (Abbildung 14) werden dem Benutzer seine aktuellen und aktiven Fragebögen zur Auswahl angezeigt.

Dabei werden One-time Fragebögen, die noch nicht ausgefüllt wurden, vorrangig behandelt und nur diese angezeigt. Sobald es keine unausgefüllten One-time Fragebögen mehr für den Benutzer gibt, werden die sich wiederholenden Fragebögen zur Auswahl angezeigt.

Es wird bei jedem Öffnen dieser Ansicht versucht, die Fragebögen des Nutzers durch Verbindung zum Server zu aktualisieren. Wenn sich die Fragebögen beim Server nicht geändert haben oder die Verbindung fehlschlägt wird der Stand aus der lokalen Datenbank geladen und angezeigt. Durch einen Klick auf einen der angezeigten Fragebögen gelangt man zu der Fragebogen Struktur Ansicht (Abbildung 15), in der der ausgewählte Fragebogen angezeigt wird.

Die Navigation zu den anderen Bereichen funktioniert dabei wieder, wie schon im Hauptmenü (Abbildung 13) beschrieben.



Abbildung 14: Fragebögen Ansicht

5.1.7 Fragebogen Struktur

Die Fragebogen Struktur Ansicht (Abbildung 15) zeigt dem Nutzer den zuvor ausgewählten Fragebogen mit seinen Frageelementen an. Dies stellt die eigentliche Hauptaufgabe der App dar. Mit dieser Ansicht kann der Benutzer den ausgewählten Fragebogen ausfüllen und mit dem "Abschicken" Button an den Server senden bzw. bei fehlender Internetverbindung zwischenspeichern, damit er später hochgeladen werden kann. Danach wird der Benutzer auf die Hauptmenü Ansicht (Abbildung 13) zurückgeleitet.

Die unterschiedlichen Fragetypen, die die App anzeigen kann:

- 1. Headline: Überschrift des Fragebogens.
- 2. Text: Text der zwischen zwei Frageelementen stehen soll.
- 3. Frage mit Textfeldeingabe: Dabei gibt es verschiedene Typen von Textfeldern, wie z.B. eine Datumseingabe, ein Multiline Textfeld oder ein ganz normales Textfeld.
- 4. Frage mit SingleChoice: Auswahl einer Antwortmöglichkeit aus mehreren möglichen Antworten durch Auswählen eines RadioButtons. Auch Ja/Nein Fragen werden durch eine solche SingleChoice Frage angezeigt.
- 5. Frage mit MultipleChoice : Auswahl mehrerer Antwortmöglichkeiten aus mehreren möglichen Antworten durch Auswählen der entsprechenden Checkboxen.
- 6. Frage mit Slider Eingabe: Antworteingabe durch ein beschriftetes Sliderelement.

Durch den "Abbrechen" Button oder den integrierten "Zurück" Button gelangt der Benutzer zurück zur Fragebögen Ansicht (Abbildung 14).



Abbildung 15: Fragebogen Struktur Ansicht

5.1.8 Studien

Die Track Your Pregnancy App bietet dem Benutzer die Möglichkeit sich bei mehreren Studien einzuschreiben und sich von diesen Studien auch wieder auszuschreiben. Bei jedem Öffnen der Studien Ansicht (Abbildung 16) wird versucht den aktuellen Stand der Studien und den Status der Studien für den Benutzer zu laden. Bei Erfolg wird sofort die Liste der Studien angezeigt. Bei fehlender Internetverbindung wird der letzte Stand aus der Datenbank geladen und eine Meldung angezeigt, dass man sich im Offline-Modus befindet.

Der Status der Studie für den Benutzer wird durch den danebenliegenden Button angezeigt. Dabei gibt es folgende verschiedenen Möglichkeiten für die Darstellung des Buttons und seiner Funktionsweise:

Studienstatus	Farbe des Buttons	Buttontext	Status bei Erfolg	Funktionsweise
Beigetreten	Rot	Studie verlassen	Nicht beigetreten	Beim Klick auf den Button wird versucht aus der Studie auszutreten
Nicht beigetreten	Grün	Studie beitreten	Nicht Privat: Beigetreten	Beim Klick auf den Button wird versucht der Studie beizutreten.
			Privat: 1)Invite: Aufnahme angefragt	Wenn die Studie privat ist wird je nach Typ unterschieden: 1) Beim Typ Invite wird eine Anfrage auf Aufnahme gemacht. 2) Beim Typ Password muss über den
			2)Passwort: Beigetreten	Code Dialog noch das richtige Passwort eingegeben werden.
Eingeladen	Gelb	Einladung annehmen	Beigetreten	Beim Klick auf den Button wird versucht der Studie beizutreten.
Aufnahme angefragt	Grau	Anfrage löschen	Nicht beigetreten	Beim Klick auf den Button wird versucht die Anfrage auf Aufnahme zu löschen.



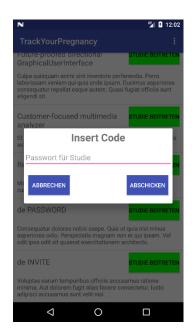


Abbildung 16: Studien Ansicht und Code Dialog

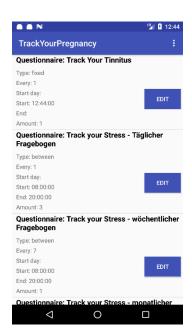
Wenn die Anfrage an den Server erfolgreich war, erscheint eine Erfolgsmeldung und der Status der Studie wird geändert. Bei Fehlschlagen der Anfrage wegen fehlender Internetverbindung, falschem Passwort oder Anderem kommt eine Fehlermeldung.

Die Navigation zu den anderen Bereichen funktioniert dabei wieder, wie schon im Hauptmenü (Abbildung 13) beschrieben.

5.1.9 Benachrichtigungseinstellungen

Der Benutzer soll in der Track Your Pregnancy App die Möglichkeit bekommen für bestimmte Fragebögen die Benachrichtigungszeiten nach seinen eigenen Wünschen ändern zu können. Dafür werden in der Benachrichtigungseinstellungen Ansicht (Abbildung 17) die veränderbaren Benachrichtigungszeitpunkte angezeigt. Neben jedem Benachrichtigungszeitraum ist ein "Edit" Button. Durch das Klicken dieses Buttons öffnet sich ein Bearbeitungsdialog für den Benachrichtigungszeitraum. In diesem kann der Benutzer durch verschiedene Spinner und Textfeldeingaben den Benachrichtigungszeitraum neu festlegen. Durch einen Klick auf den "Save" Button wird, wenn die Eingaben alle korrekt sind, der neue Zeitraum in der lokalen Datenbank gespeichert und erst wieder verändert, wenn der Benutzer ihn ändert oder vom Server neue Daten für die Fragebögen kommen.

Durch einen Klick auf den "Abbrechen" Button wird die Bearbeitung abgebrochen. Die Navigation zu den anderen Bereichen funktioniert dabei wieder, wie schon im Hauptmenü (Abbildung 13) beschrieben.



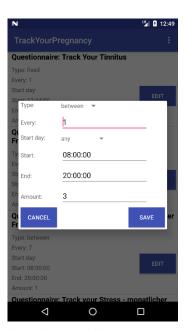


Abbildung 17: Benachrichtigungseinstellungen Ansicht und Bearbeitungsdialog

5.1.10 Sprachauswahl

Beim Track Your Pregnancy Projekt war auch die Mehrsprachigkeit der App eine wichtige Eigenschaft. In der Sprachauswahl Ansicht (Abbildung 18) werden dem Benutzer alle vom Server bereitgestellten Sprachen angezeigt. Die Default Sprache der App ist Englisch. Durch die Auswahl einer Sprache wird diese grün markiert und die App wird von diesem Zeitpunkt an in der ausgewählten Sprache angezeigt.

Die Navigation zu den anderen Bereichen funktioniert dabei wieder, wie schon im Hauptmenü (Abbildung 13) beschrieben.



Abbildung 18: Sprachauswahl Ansicht

5.1.11 Antwortsätze

Nach dem Beantworten von Fragebögen gibt es für den Benutzer die Möglichkeit sich in der Antwortsätze Ansicht (Abbildung 19) anzeigen zu lassen, welche Antwortsätze er bis jetzt an den Server gesendet hat und diese vom Server evaluiert wurden. Dabei wird bei jedem Öffnen der Ansicht versucht vom Server neue Antwortsätze und deren Evaluierungen zu laden. Bei Erfolg wird eine Liste der Antwortsätze zur Auswahl angezeigt. Dabei wird dem Benutzer auch der Zeitpunkt, an dem der Antwortsatz hochgeladen wurde, angezeigt, damit er erkennen kann welcher Antwortsatz der aktuellste ist. Bei Misserfolg durch fehlende Internetverbindung erscheint eine Offline-Modus Meldung und die bisher in der Datenbank gespeicherten und evaluierten Antwortsätze werden geladen und angezeigt.

Durch die Auswahl eines Antwortsatzes gelangt man zur Ergebnisansicht (Abbildung 20). Die Navigation zu den anderen Bereichen funktioniert dabei wieder, wie schon im Hauptmenü (Abbildung 13) beschrieben.

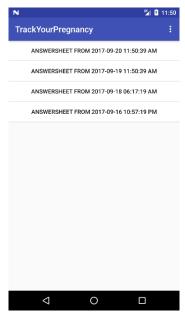


Abbildung 19: Antwortsätze Ansicht

5.1.12 Ergebnisse

Um dem Benutzer die Ergebnisse seiner Antworten anzuzeigen, gibt es die Ergebnisansicht (Abbildung 20). Dabei wird dem Benutzer das Feedback zu seinem zuvor in der Antwortsätze Ansicht (Abbildung 19) ausgewählten Antwortsatz gezeigt. Die verschiedenen Feedbackblöcke werden in einer Liste mit Titel und dazugehörigem Text angezeigt.

Zuerst werden dem Nutzer nur die Titel der Feedbacks angezeigt, um eine bessere Übersichtlichkeit zu gewährleisten. Durch einen Klick auf einen Titel wird der dazugehörige Text angezeigt. Dieser Text verschwindet durch einen weiteren Klick auf den Titel wieder.

Durch die integrierte "Zurück"- Taste gelangt der Benutzer wieder zurück zur Antwortsätze Ansicht.



Abbildung 20: Ergebnisse Ansicht

6. Implementierung

In diesem Kapitel werden bestimmte Bereiche der Implementierung der Track Your Pregnancy App, die für die Erfüllung der Anforderungen nötig sind, genauer beleuchtet. Dafür wird zuerst gezeigt, wie die App die automatischen oder auch vom Benutzer geänderten Benachrichtigungen umsetzt. Danach wird noch gezeigt, wie ein Slider (in Android Seekbar [26]) ohne initialen Wert erzeugt werden kann.

6.1 Benachrichtigungen

Da die Hauptfunktion der Track Your Pregnancy App das regelmäßige Ausfüllen eines Fragebogens ist, sollte die App eine Funktion bereitstellen, den Benutzer zu genau dieser Aufgabe aufzufordern und ihn daran zu erinnern. Dabei gibt es die Möglichkeit diese Benachrichtigungen von einem Server über eine Internetverbindung an das Gerät zu senden. Dafür gäbe es den *Google Cloud Messaging* Service [27]. Da aber eine der Anforderungen an die App die Bedienbarkeit und Funktionalität ohne eine dauerhafte Internetverbindung ist, werden die Benachrichtigungen lokal über das jeweilige Gerät erstellt. Ein weiterer Punkt der für diese *local notifications* spricht ist, dass die vom Benutzer veränderten Benachrichtigungszeiträume auch nur in der lokalen Datenbank gespeichert und nicht wieder an den Server übermittelt werden.

In Android werden diese Benachrichtigungen mit Hilfe eines *AlarmManager* [14] implementiert, der einen Broadcast zu dem gewünschten Zeitpunkt sendet, der dann wiederum von einem Receiver empfangen wird. In diesem Receiver wird dann erst die eigentliche Benachrichtigung erstellt und dem Benutzer auf dem Gerät angezeigt. Da die gespeicherten Zeitpunkte im *AlarmManager* einen Neustart des Gerätes nicht überdauern, müssen die Benachrichtigungszeiten nach einem Neustart erneut beim *AlarmManager* registriert werden. Dies wird durch einen Receiver erreicht, der durch das Neustarten des Gerätes aktiviert wird und in seiner *onReceive* Methode den *SetNotificationService* startet, der wiederum den *MyNotificationManager* aufruft, um die Benachrichtigungszeitpunkte wieder neu beim *AlarmManager* zu registrieren. Dieser Vorgang wird in der folgenden Abbildung 21 gezeigt.

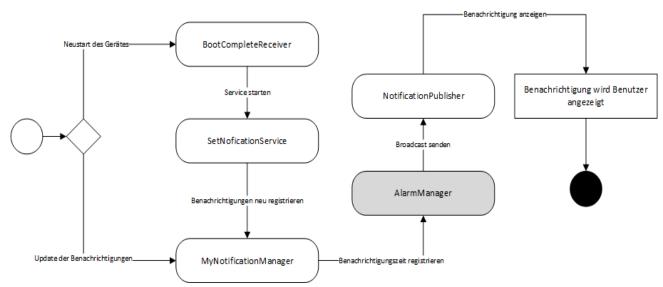


Abbildung 21: Benachrichtigungen

Das Erstellen eines Benachrichtigungszeitraumes für den *AlarmManager* wird in der Klasse *MyNotficationManager* gemacht.

Dafür wird zuerst für eine Liste von MySchedules eine Liste von NotificationTimes erstellt, die die Zeitpunkte der Benachrichtigung repräsentieren. Dies wird mit der Methode in Listing 1 gemacht. Zuerst wird eine Liste von NotificationTime erstellt (Linie 2). Danach wird aufgrund des Schedule-Typs unterschieden wie die Calendar [28] Objekte, die benötigt werden, um die Zeit einzustellen, erstellt werden sollen, da es beim Typ "fixed" nur einen Zeitpunkt gibt und bei "between" mehrere möglich sind. Bei "fixed" wird dann von Linie 6 bis 13 ein Calendar erstellt mit dem angegebenen Wochentag und der Zeit. Aus diesem Calendar wird dann ein NotificationTime Objekt erstellt und der Liste hinzugefügt (Linie 14-15). Bei "between" werden zuerst zwei Calendar erstellt. In Linie 21-27 mit dem Startzeitpunkt und in Linie 30-36 mit dem Endzeitpunkt. In Linie 28 und 37 werden diese Zeitpunkte in Millisekunden gespeichert, um dann, falls amount größer als 1 ist (Linie 38), in Linie 39 die zeitliche Entfernung zwischen zwei Benachrichtigungen zu ermitteln. Damit werden dann in Linie 42-46 die restlichen NotificationTime Objekte mit Hilfe eines Calendar erstellt und der Liste hinzugefügt. In Linie 40 und 41 wurden bereits der Start- und Endzeitpunkt der Liste hinzugefügt. Falls in Linie 38 amount nicht größer als 1 ist, wird lediglich ein NotificationTime, der zwischen Startund Endzeit liegt, der Liste hinzugefügt (Linie 50). Am Ende wird in Linie 57 dann die Liste mit den NotificationTimes zurückgegeben.

Listing 1: MyNotificationManager: GetNotificationTime

```
1 public List<NotificationTime> getNotificationtime(List<MySchedule>
                                                            mySchedules) {
          List<NotificationTime> returnList = new ArrayList<>();
2
3
          for (MySchedule mySchedule:mySchedules) {
4
              switch (mySchedule.schedule.type) {
5
                  case "fixed":
6
                      Calendar c = Calendar.getInstance();
7
                      if (mySchedule.schedule.start day != 0) {
8
                          c.set (Calendar.DAY OF WEEK,
                                          getCalendarDayFromDayOfWeek(
                                          mySchedule.schedule.start day));
9
10
                      String[] time = mySchedule.schedule.at.split(":");
11
                    c.set(Calendar.HOUR OF DAY, Integer.parseInt(time[0]));
12
                      c.set(Calendar.MINUTE, Integer.parseInt(time[1]));
13
                      c.set(Calendar.SECOND, Integer.parseInt(time[2]));
14
                      NotificationTime nTimef = new
                              NotificationTime (mySchedule.id,
                        c.getTimeInMillis(), mySchedule.schedule.every);
15
                      returnList.add(nTimef);
16
                      break;
17
                  case "between":
18
                 String[] starttime = mySchedule.schedule.start.split(":");
19
                 String[] endtime = mySchedule.schedule.end.split(":");
20
                      Calendar cstart = Calendar.getInstance();
21
22
                      if (mySchedule.schedule.start day != 0) {
23
                          cstart.set(Calendar.DAY OF WEEK,
                                          getCalendarDayFromDayOfWeek(
                                          mySchedule.schedule.start day));
24
                      }
25
          cstart.set(Calendar.HOUR OF DAY, Integer.parseInt(starttime[0]));
          cstart.set(Calendar.MINUTE, Integer.parseInt(starttime[1]));
26
27
          cstart.set(Calendar.SECOND, Integer.parseInt(starttime[2]));
28
                      long startTime = cstart.getTimeInMillis();
29
30
                      Calendar cend = Calendar.getInstance();
```

```
if (mySchedule.schedule.start day != 0) {
31
                        cend.set (Calendar.DAY OF WEEK,
                                           getCalendarDayFromDayOfWeek(
                                           mySchedule.schedule.start day));
33
                       }
            cend.set(Calendar.HOUR OF_DAY, Integer.parseInt(endtime[0]));
34
35
            cend.set(Calendar.MINUTE, Integer.parseInt(endtime[1]));
            cend.set(Calendar.SECOND, Integer.parseInt(endtime[2]));
36
37
                       long endTime = cend.getTimeInMillis();
38
                       if (mySchedule.schedule.amount>1) {
39
                           long step = (endTime - startTime) /
                                           (mySchedule.schedule.amount - 1);
40
                         returnList.add(new
                        NotificationTime (mySchedule.id, cstart.
                        getTimeInMillis(), mySchedule.schedule.every));
41
                         returnList.add(new NotificationTime(mySchedule.id,
                        cend.getTimeInMillis(), mySchedule.schedule.every));
42
              for (int i = 1; i \le (mySchedule.schedule.amount - 2); <math>i++) {
43
                               Calendar cAmount = Calendar.getInstance();
44
                            cAmount.setTimeInMillis(startTime + (step * i));
45
                          returnList.add (new NotificationTime (mySchedule.id,
                  cAmount.getTimeInMillis(), mySchedule.schedule.every));
46
                       }
48
                       else
49
50
                         returnList.add (new NotificationTime (mySchedule.id,
                   (cstart.getTimeInMillis()+cend.getTimeInMillis()) / 2,
                                           mySchedule.schedule.every));
51
52
                       break;
53
                   default:
54
                      break;
55
              }
56
57
          return
                   returnList;
58
      }
```

Danach werden für jede *NotificationTime* ein Benachrichtigungszeitpunkt beim *AlarmManager* registriert.

Dies geschieht mit den Funktionen in Listing 2.

Dabei wird zuerst in Linie 3 vom System eine Instanz von AlarmManager angefragt. Danach wird in Linie 4 ein sich wiederholender Zeitpunkt erstellt, da die Benachrichtigung sich nach einer gewissen Anzahl von Tagen wiederholen soll. Dabei wird die Funktion getPendingIntent genutzt, die einen *Intent* [29] erstellt, der die *NotificationPublisher* Klasse als Receiver des Broadcast kennzeichnet (Linie 7-10).

Listing 2: MyNotificationManager: scheduleNotification und getPendingIntent

Als Letztes wird dann die *onReceive* Methode des *NotificationPublishers* gebraucht. Diese wird in Listing 3 genauer beschrieben.

Zuerst werden in Linie 2 die Texte für die Benachrichtigung aus den *SharedPreferences* [13] geladen. Danach wird vom System eine Instanz von *NotificationManager* [30] angefragt (Linie 3). Dann wird in Linie 4-6 ein *Intent* [29]für eine Activity erstellt, die beim Klicken auf die Benachrichtigung aufgerufen werden soll (hier LoginActivity). Dieser wird dann in Linie 12 dem in Linie 7 erstellten *NotificationCompat.Builder* [31] übergeben. In Linie 8-11 werden noch weitere Informationen der Benachrichtigung festgelegt, wie Titel, Text, Icon und AutoCancel (die Benachrichtigung verschwindet nachdem sie geklickt wurde). Zuletzt wird dann mit dem *NotificationManager* die Benachrichtigung angezeigt (Linie 14).

Listing 3: NotificationPublisher: onReceive

```
public void onReceive(Context context, Intent intent) {
         SharedPreferences sharedTexts =
                  context.getSharedPreferences("Texts" + MainManager.local,
                                                Context.MODE PRIVATE);
3
          NotificationManager notificationManager =
                              (android.app.NotificationManager) context.
                        getSystemService(Context.NOTIFICATION SERVICE);
4
          Intent repeating intent= new Intent(context,LoginActivity.class);
          repeating intent.setFlags(intent.FLAG ACTIVITY CLEAR TOP);
5
6
          PendingIntent pendingIntent =PendingIntent.getActivity(context,0,
                       repeating_intent,PendingIntent.FLAG_UPDATE_CURRENT);
7
          NotificationCompat.Builder builder = new
                                    NotificationCompat.Builder(context);
8
    builder.setContentTitle(sharedTexts.getString("app.notification.text",
                                    "Please fill in a new questionnaire"));
   builder.setContentText(sharedTexts.getString("app.notification.title",
                                          "Track your Pregnancy now"));
10
         builder.setSmallIcon(R.mipmap.ic launcher);
11
          builder.setAutoCancel(true);
12
         builder.setContentIntent(pendingIntent);
13
         Calendar cal = Calendar.getInstance();
14
         notificationManager.notify((int)cal.getTimeInMillis(),
                                                      builder.build());
15
      }
```

6.2 Slider ohne initialen Wert

Eine der Anforderungen der Track Your Pregnancy App war, dass Slider-Eingabe Objekte keinen initialen Wert haben. Dies könnte sonst zur Beeinflussung des Benutzers führen und so seine Antworten manipulieren. Da das *Seekbar* [26] Objekt in Android dies nicht standardmäßig macht, müssen in der *Input_Slider* Klasse, die das Slider-Eingabe Objekt in der App anlegt, ein paar Anpassungen bei dem *Seekbar* Objekt gemacht werden.

Diese Anpassungen werden in Listing 4 genauer gezeigt. Dabei werden aber nur die für die Anforderung wichtigen Zeilen hier gezeigt.

Die Klasse Input_Slider legt in ihrem Konstruktor in Linie 4 zuerst eine neue Instanz eines Seekbar Objektes an. Danach wird in Linie 5 der Progress auf 0 gesetzt damit kein Progressbalken mehr zu sehen ist. In Linie 6 wird dann ein transparentes Drawable [32] Objekt erstellt, dass in Linie 7 der Seekbar als Thumb übergeben wird. Somit ist die Seekbar komplett leer und hat keinen initialen Wert

Damit der Benutzer aber wieder eine normale *Seekbar* zur Verfügung hat, wenn er die Frage beantworten möchte, wird der *OnSeekBarChangeListener* [33] bei der *onStartTrackingTouch* Methode in Linie 11 überschrieben.

Dabei wird ein neuer Thumb, durch eine Vorlage, die im Projekt als *Drawable Resource* in der Datei *seekbar_thumb.xml* gespeichert ist, erstellt (Linie 12). Dieser neue Thumb wird dann der *Seekbar* wieder als Thumb in Linie 13 übergeben und dadurch hat der Benutzer wieder eine ganz normale *Seekbar*, um seine Antwort einzugeben.

Listing 4: Input_Slider

```
1 public class Input Slider extends Input Base {
    SeekBar seekBar;
    public Input Slider (Question View question View,
                        List<QuestionSlider.Attributes.Content.Answers>
                        answers, QuestionSlider.Attributes.Content.Values
                        values, Context c) {
        seekBar = new SeekBar(c);
5
        seekBar.setProgress(0);
6
        Drawable transparentDrawable = newColorDrawable(Color.TRANSPARENT);
        seekBar.setThumb(transparentDrawable);
        final Resources res = c.getResources();
9
        seekBar.setOnSeekBarChangeListener(new
                                          SeekBar.OnSeekBarChangeListener()
10
              @Override
              public void onStartTrackingTouch(SeekBar seekBar) {
11
12
                Drawable thumb = res.getDrawable(R.drawable.seekbar thumb);
13
                seekBar.setThumb(thumb);
14
              1
15
          });
16
      }
17}
```

7. Anforderungsabgleich

In diesem Kapitel werden die Anforderungen, die in Kapitel 3 definiert wurden, mit der Implementierung und den Funktionen der App abgeglichen.

7.1 Funktionale Anforderungen

Hier werden die funktionalen Anforderungen an die App abgeglichen. Dabei wird überprüft ob in der aktuellen Implementierung alle funktionalen Anforderungen erfüllt sind.

Nr.	Beschreibung	Anforderungsanalyse
1.	Registrierung in der App	Anforderung erfüllt (siehe Kapitel 5.1.2 Registrierung in der App).
2.	App auch ohne Internetverbindung bedienen	Anforderung erfüllt. Alle zur Bedienung benötigten Daten werden in der lokalen Datenbank auf dem Gerät gespeichert. So ist, natürlich bis auf die Studienverwaltung, eine volle Funktionalität auch ohne funktionierende Internetverbindung garantiert.
	St	udienteilnahme
3.	An verschiedenen Studien teilnehmen	Anforderung erfüllt (siehe Kapitel 5.1.8 Studien).
4.	Studieneinladungen annehmen	Anforderung erfüllt (siehe Kapitel 5.1.8 Studien).
5.	An einer privaten Studie teilnehmen	Anforderung erfüllt (siehe Kapitel 5.1.8 Studien).
		Fragebögen
6.	Statistische Fragebögen innerhalb der Studienteilnahme ausfüllen	Anforderung erfüllt (siehe Kapitel 5.1.6 Fragebögen und Kapitel 5.1.7 Fragebogen Struktur).
7.	Fragebögen deaktivieren	Anforderung erfüllt (siehe Kapitel 5.1.6 Fragebögen).
8.	Zustand eines Fragebogens ändern	Anforderung erfüllt (siehe Kapitel 5.1.6 Fragebögen).
9.	Ergebnisse synchronisieren	Anforderung erfüllt (siehe Kapitel 5.1.7 Fragebogen Struktur).
10.	Slider ohne initialen Wert	Anforderung erfüllt (siehe Kapitel 6.2 Slider ohne initialen Wert).
11.	An auszufüllende Fragebögen erinnern	Anforderung erfüllt (siehe Kapitel 6.1 Benachrichtigungen).
12.	Erinnerungsfunktion anpassen	Anforderung erfüllt (siehe Kapitel 5.1.9 Benachrichtigungseinstellungen).
13.	Ergebnisse in der App anzeigen	Anforderung erfüllt (siehe Kapitel 5.1.11 Antwortsätze und Kapitel 5.1.12 Ergebnisse).

8. Fazit

Die erste Phase der Entwicklung des Track Your Pregnancy Projekts ist zur Zeit dieser Arbeit abgeschlossen. Die Funktionalität der App ist derart umgesetzt, dass ein Benutzer seine Studien verwalten, Fragebögen beantworten und auswerten lassen, Benachrichtigungseinstellungen ändern und die Sprache auswählen kann. Der erste Teil dieses Kapitels fasst die Erkenntnisse aus dieser Arbeit zusammen. Im zweiten Teil werden Ideen behandelt, die möglicherweise in einer weiteren Entwicklung der App umsetzbar wären, um die Funktionsweise der App zu erweitern bzw. zu verbessern.

8.1 Zusammenfassung

Im Rahmen dieser Bachelorarbeit ist eine mobile Anwendung für Smartphones und Tablets mit Android Betriebssystem entstanden.

Die Anforderungen wurden in mehreren Meetings definiert. Auch während der Arbeit kamen einige neue Anforderungen und Ideen hinzu, die in dieser Arbeit berücksichtigt wurden. Dabei wurde gezeigt, wie lokale Benachrichtigungen an den Benutzer gesendet werden, da dies, durch die Anforderung für eine Benutzung ohne funktionierende Internetverbindung, nötig war. Auch die Anpassungen der View Elemente von Android (hier SeekBar), die aufgrund wissenschaftlicher Erkenntnisse für eine unbeeinflusste Beantwortung der Fragebögen nötig sind, wurden in dieser Arbeit näher beleuchtet.

Des Weiteren wurde gezeigt, wie beliebig unterschiedliche Fragebögen mit unterschiedlichen Elementen angezeigt und beantwortet werden können und zur gleichen Zeit an mehreren Studien teilgenommen werden kann.

8.2 Ausblick

In diesem Abschnitt werden Ideen behandelt, die während der Entwicklung dieser App aufgekommen sind, um das Projekt zu verbessern.

8.2.1 Verbesserung der Visualisierung der Ergebnisse

Die Visualisierung der Ergebnisse beschränkt sich im Moment noch auf die schlichte Anzeige von Ergebnistexten mit deren Hinweistiteln. Dabei wäre eine Idee diese verschiedenen Feedbackblöcke farblich zu markieren, je nachdem ob sie positive oder negative Auswirkungen auf den Schwangerschaftsverlauf haben könnten. Des Weiteren wäre es auch möglich das gesamte Feedback nach positiv und negativ zu sortieren oder es zusammen mit der ausschlaggebenden Frage darzustellen, um dem Benutzer noch ein genaueres Feedback zu geben.

8.2.2 Verbesserung am User-Interface und Hilfe

Da bei dieser Bachelorarbeit das Hauptaugenmerk auf dem Funktionsumfang lag, sind manche GUIs offen für Verbesserungen, um dem Benutzer manche Informationen noch deutlicher zu machen und so die intuitive Bedienbarkeit der App zu verbessern. So könnte man in der Studienansicht schon

deutlich machen welche Studien privat und welche öffentlich sind oder auch bei den Benachrichtigungseinstellungen, die Anzeige eines Benachrichtigungszeitraumes schöner gestalten. Des Weiteren gibt es in der App momentan noch keinerlei Hilfe- oder Infotexte. Dies würde dem Benutzer sehr helfen, wenn doch größere Verständnisprobleme mit der App auftauchen.

Literaturverzeichnis

- [1] Baden-Würrtemberg, Landesärztekammer, "Ärztemangel? Ärztliche Versorgung in Baden-Württemberg," [Online]. Available: http://www.aerztekammer-bw.de/news/2017/2017-04/pm-daet/index.html. [Zugriff am 11 September 2017].
- [2] M. Ruf-Leuschner, N. Brunnemann, M. Schauer, R. Pryss, E. Barnewitz, M. Liebrecht, M. Reichert und T. Elbert, "Die KINDEX-App ein Instrument zur Erfassung und unmittelbaren Auswertung von psychosozialen Belastungen bei Schwangeren in der täglichen Praxis bei Gynäkologinnen, Hebammen und in Frauenkliniken," *Verhaltenstherapie*, August 2016.
- [3] M. Ruf-Leuschner, R. Pryss, M. Liebrecht, J. Schobel, A. Spyridou, M. Reichert und M. Schauer, "Preventing further trauma: KINDEX mum screen assessing and reacting towards psychosocial risk factors in pregnant women with the help of smartphone technologies," in XIII Congress of European Society of Traumatic Stress Studies (ESTSS) Conference, 2013, pp. 70--70.
- [4] Google Inc., "GooglePlay Store," [Online]. Available:
 https://play.google.com/store?utm_source=emea_Med&utm_medium=hasem&utm_content=
 May1115&utm_campaign=Evergreen&pcampaignid=MKT-EG-emea-de-all-Med-hasem-pyEvergreen-May11151%7cONSEM_kwid_43700007181511317&referrer=gclid%3DEAlalQobChMI5sWOtoud1glVg5U
 bCh1jv. [Zugriff am 11 September 2017].
- [5] Apple Inc., "Apple AppStore," [Online]. Available: https://www.apple.com/de/ios/app-store/. [Zugriff am 11 September 2017].
- [6] J. Herrmann, R. Dr. Pryss und J. Schobel, "MyKind," Universität Ulm (Institut für Datenbanken und Informationssysteme) und Universität Konstanz (Klinische Psychologie und Neurowissenschaften), [Online]. Available: https://www.mykind.info/. [Zugriff am 11 September 2017].
- [7] J. Herrmann, Konzeption und technische Realisierung eines mobilen Frameworks zur Unterstützung tinnitusgeschädigter Patienten, Universität Ulm, 2014.
- [8] B. Klecina, "Assess Your Stress: Conceptual re-design of the Track Your Tinnitus system for measuring stress at the workplace," 22 Juli 2016. [Online]. Available: http://dbis.eprints.uniulm.de/1395/1/MA_KL_2016.pdf. [Zugriff am 11 September 2017].
- [9] Bundesministerium für Gesundheit, "Bundestag verabschiedet Präventionsgesetz," [Online]. Available: https://www.bundesgesundheitsministerium.de/ministerium/meldungen/2015/praeventionsgesetz.html. [Zugriff am 11 September 2017].
- [10] Google Inc. und Health & Parenting Ltd, "Schwangerschaft +," [Online]. Available: https://play.google.com/store/apps/details?id=com.hp.pregnancy.lite. [Zugriff am 11 September 2017].

- [11] Google Inc. und Amila, "Pregnancy Week by Week," [Online]. Available: https://play.google.com/store/apps/details?id=com.easymobs.pregnancy. [Zugriff am 11 September 2017].
- [12] Google Inc. und BabyCenter, "Meine Schwangerschaft & Baby," [Online]. Available: https://play.google.com/store/apps/details?id=com.babycenter.pregnancytracker. [Zugriff am 11 September 2017].
- [13] Google Inc., "SharedPreferences," [Online]. Available: https://developer.android.com/reference/android/content/SharedPreferences.html. [Zugriff am 3 August 2017].
- [14] Google Inc., "AlarmManager," [Online]. Available: https://developer.android.com/reference/android/app/AlarmManager.html. [Zugriff am 8 August 2017].
- [15] Google Inc., "AppCompatActivity," [Online]. Available: https://developer.android.com/reference/android/support/v7/app/AppCompatActivity.html. [Zugriff am 10 Juni 2017].
- [16] Google Inc., "AsyncTask," [Online]. Available: https://developer.android.com/reference/android/os/AsyncTask.html. [Zugriff am 15 August 2017].
- [17] Google Inc., "SQLiteOpenHelper," [Online]. Available: https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html. [Zugriff am 10 August 2017].
- [18] Google Inc., "View," [Online]. Available: https://developer.android.com/reference/android/view/View.html. [Zugriff am 24 Juni 2017].
- [19] Google Inc., "Object," [Online]. Available: https://developer.android.com/reference/java/lang/Object.html. [Zugriff am 28 Juni 2017].
- [20] Google Inc., "Gson," [Online]. Available: https://google.github.io/gson/apidocs/com/google/gson/Gson.html. [Zugriff am 24 Juli 2017].
- [21] Google Inc., "ListView," [Online]. Available: https://developer.android.com/guide/topics/ui/layout/listview.html. [Zugriff am 2 August 2017].
- [22] Google Inc., "BroadcastReceiver," [Online]. Available: https://developer.android.com/reference/android/content/BroadcastReceiver.html. [Zugriff am 20 August 2017].
- [23] Google Inc., "Service," [Online]. Available: https://developer.android.com/reference/android/app/Service.html. [Zugriff am 20 August 2017].

- [24] Google Inc., "ArrayAdapter," [Online]. Available: https://developer.android.com/reference/android/widget/ArrayAdapter.html. [Zugriff am 20 Juli 2017].
- [25] Google Inc., "Spinners," [Online]. Available: https://developer.android.com/guide/topics/ui/controls/spinner.html. [Zugriff am 28 Juli 2017].
- [26] Google Inc., "Seekbar," [Online]. Available: https://developer.android.com/reference/android/widget/SeekBar.html. [Zugriff am 30 Juli 2017].
- [27] Google Inc., "Google Cloud Messaging," [Online]. Available: https://developers.google.com/cloud-messaging/gcm. [Zugriff am 28 Juni 2017].
- [28] Google Inc., "Calendar," [Online]. Available: https://developer.android.com/reference/java/util/Calendar.html. [Zugriff am 2 August 2017].
- [29] Google Inc., "Intent," [Online]. Available: https://developer.android.com/reference/android/content/Intent.html. [Zugriff am 13 Juli 2017].
- [30] Google Inc., "NotificationManager," [Online]. Available: https://developer.android.com/reference/android/app/NotificationManager.html. [Zugriff am 15 Juli 2017].
- [31] Google Inc., "NotificationCompat.Builder," [Online]. Available: https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder .html. [Zugriff am 16 Juli 2017].
- [32] Google Inc., "Drawable," [Online]. Available: https://developer.android.com/reference/android/graphics/drawable/Drawable.html. [Zugriff am 20 Juli 2017].
- [33] Google Inc., "Seekbar.OnSeekBarChangedListener," [Online]. Available: https://developer.android.com/reference/android/widget/SeekBar.OnSeekBarChangeListener. html. [Zugriff am 20 Juli 2017].
- [34] T. Probst, R. Pryss, B. Langguth, M. Spiliopoulou, M. Landgrebe, M. Vesala, S. Harrison, J. Schobel, M. Reichert, M. Stach und W. Schlee, "Outpatient Tinnitus Clinic, Self-Help Web Platform, or Mobile Application to Recruit Tinnitus Study Samples?," *Frontiers in Aging Neuroscience*, Bd. 9, pp. 113--113, April 2017.
- [35] R. Pryss, T. Probst, W. Schlee, J. Schobel, B. Langguth, P. Neff, M. Spiliopoulou und M. Reichert, "Mobile Crowdsensing for the Juxtaposition of Realtime Assessments and Retrospective Reporting for Neuropsychiatric Symptoms," in 30th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2017), I. C. S. Press, Hrsg., 2017.
- [36] R. Pryss, W. Schlee, B. Langguth und M. Reichert, "Mobile Crowdsensing Services for Tinnitus Assessment and Patient Feedback," in *6th IEEE International Conference on AI* \& *Mobile Services (IEEE AIMS 2017)*, I. C. S. Press, Hrsg., 2017.

- [37] W. Schlee, R. Pryss, T. Probst, J. Schobel, A. Bachmeier, M. Reichert und B. Langguth, "Measuring the Moment-to-Moment Variability of Tinnitus: The TrackYourTinnitus Smart Phone App," *Frontiers in Aging Neuroscience*, Bd. 8, pp. 294--294, Dezember 2016.
- [38] R. Pryss, M. Reichert, J. Herrmann, B. Langguth und W. Schlee, "Mobile Crowd Sensing in Clinical and Psychological Trials? A Case Study," in *28th IEEE Int'l Symposium on Computer-Based Medical Systems*, I. C. S. Press, Hrsg., 2015, pp. 23--24.
- [39] R. Pryss, M. Reichert, B. Langguth und W. Schlee, "Mobile Crowd Sensing Services for Tinnitus Assessment, Therapy and Research," in *IEEE 4th International Conference on Mobile Services* (MS 2015), I. C. S. Press, Hrsg., 2015, pp. 352--359.
- [40] T. Probst, R. Pryss, B. Langguth und W. Schlee, "Emotional states as mediators between tinnitus loudness and tinnitus distress in daily life: Results from the TrackYourTinnitus application," *Scientific Reports*, Bd. 6, February 2016.
- [41] T. Probst, R. Pryss, B. Langguth, J. Rauschecker, J. Schobel, M. Reichert, M. Spiliopoulou, W. Schlee und J. Zimmermann, "Does tinnitus depend on time-of-day? An ecological momentary assessment study with the TrackYourTinnitus application," *Frontiers in Aging Neuroscience*, Bd. 9, pp. 253--253, 2017.

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Ark die angegebenen Hilfsmittel verwendet habe. Sinn als solche kenntlich gemacht und mit genauer Que versehen.	gemäße Übernahmen aus anderen Werken sind
Ulm, den	
Oiii, deii	Fabian Haug