



Konzeption und Realisierung eines Konzepts zur mobilen Unterstützung Tinnitus-geschädigter Patienten am Bei- spiel des mobilen Betriebssystems Win- dows Phone

Bachelorarbeit an der Universität Ulm

Vorgelegt von:

Michael Klaus Meyer
michael.meyer@uni-ulm.de

Gutachter:

Prof. Dr. Manfred Reichert

Betreuer:

Rüdiger Pryss

2015

Fassung 27. Mai 2015

Inhaltsverzeichnis

1	Einleitung	1
1.1	Tinnitus	1
1.2	Motivation	2
1.3	Aufbau der Arbeit	2
2	Track Your Tinnitus	5
2.1	Vorstellung der Webseite	5
2.2	Vorstellung der App	6
3	Verwandte Arbeiten	7
4	Anforderungen	9
4.1	Funktionale Anforderungen	9
4.2	Nichtfunktionale Anforderungen	11
5	Architektur	13
5.1	Ablauf	13
5.2	Datenstruktur	14
5.3	Libraries	17
5.3.1	Windows Phone Toolkit	17
5.3.2	Windows Phone Calender Control	18
5.4	Klassen der App	19
6	Implementierung und Implementierungsaspekte	21
6.1	Benachrichtigungen in der App	21
6.1.1	Remote oder Local Notification	21
6.1.2	Alarm und Reminder	22
6.1.3	Implementierung der Reminder	22
6.1.4	Erzeugen der Benachrichtigungszeit	24
	Zufällige Benachrichtigung	24
	Benutzerdefinierte Benachrichtigung	25

6.2	Speichernutzung in der App	26
6.2.1	Lokale SQL-Datenbank	26
6.2.2	IsolatedStorageSettings	28
7	Vorstellung der Anwendung	31
7.1	Anmeldung mit Benutzername und Kennwort	31
7.2	Registrierung	32
7.3	Statistische Fragebögen	33
7.4	Hauptmenü	34
7.5	Fragebogen zur Überwachung der Tinnituswahrnehmung	35
7.6	Benachrichtigungseinstellungen und Einstellungen	36
8	Abgleich der Anforderungen	39
8.1	Funktionale Anforderungen	39
8.2	Nichtfunktionale Anforderungen	40
9	Zusammenfassung und Ausblick	41
9.1	Zusammenfassung	41
9.2	Ausblick	42
	Literaturverzeichnis	43

1 Einleitung

Im Rahmen dieses Kapitels soll zunächst geklärt werden, was ein Tinnitus ist und wie weit er in der Bevölkerung verbreitet ist. Danach folgt eine Beschreibung des Zusammenhangs mit dem Track Your Tinnitus Projekt. Im Anschluss soll ein kurzer Überblick über den Inhalt und Aufbau dieser Arbeit gegeben werden.

1.1 Tinnitus

Mit dem Begriff Tinnitus wird ein Symptom bezeichnet, bei dem der Betroffene Töne und Geräusche wahrnimmt. Diese Töne und Geräusche haben keine erkennbare Ursache. Dabei sind zwei Arten des Tinnitus zu unterscheiden. Beim objektiven Tinnitus nimmt die betroffene Person Töne und Geräusche wahr, welche in ihrem eigenen Körper entstehen und dadurch messbar sind. Diese Art des Tinnitus tritt im Gegensatz zum subjektiven Tinnitus eher selten auf.

Vom subjektiven Tinnitus wird gesprochen, sollte der Betroffene Töne und Geräusche wahrnehmen, bei denen keine physikalische Ursache in der direkten Umgebung zum Betroffenen festgestellt werden kann. Dabei kann er in den unterschiedlichsten Formen auftreten. Diese Formen gehen von kaum wahrnehmbaren bis zu Lebens beeinflussenden Tönen und Geräuschen. Diese Dauerbelastung kann die Betroffenen in Extremfällen bis zum Selbstmord treiben. Tinnitus tritt häufig auf, wenn betroffene einen Gehörverlust erleiden, zum Beispiel durch Knalltraumata oder zunehmendes Alter. In vielen Fällen ist es jedoch nicht möglich eine konkrete Ursache festzustellen [11].

Die Herausforderung beim subjektiven Tinnitus stellt dabei vor allem das Messen dar. Da der subjektive Tinnitus nicht mit einer konkreten Quelle verbunden werden kann, ist es nur dem Betroffenen möglich über den Tinnitus zu berichten.

1.2 Motivation

Da der Tinnitus und dessen Zustand nur vom Betroffenen wahrgenommen und festgehalten werden kann, sind genaue Messungen schwierig. Gleichzeitig kann die Wahrnehmung des Tinnitus im Verlauf des Tages Schwankungen unterliegen, welche auch bei chronischem Tinnitus auftreten können. Diese Schwankungen können durch verschiedene Umstände, wie zum Beispiel Stress und Umgebungsgeräusche, stärker oder schwächer werden. Betroffenen ist es jedoch meistens möglich eine relativ akkurate Einschätzung dieser Schwankungen zu liefern.

Aufgrund dieser Tatsachen ist das Track Your Tinnitus Projekt entstanden. Dieses soll es ermöglichen diese Schwankungen zu dokumentieren und sie mit der Umgebung und der Tätigkeit des Betroffenen in Verbindung zu bringen. Die hierbei erfassten Daten zudem können mit einem behandelnden Arzt oder Therapeuten besprochen werden. Im Rahmen dieses Projekts, stehen dem Betroffenen eine Webseite und eine App für iOS und Android zur Verfügung, welche in Kapitel 2.1 und 2.2 vorgestellt werden. Ein ähnliches Prinzip, wie in diesem Projekt, wurde bereits in einer Studie verwendet [6]. In dieser wurde den Teilnehmern ein PDA mit einer Anwendung zu Verfügung gestellt, welcher die Teilnehmer viermal am Tag einen Fragebogen ausfüllen ließ.

Bisher wurde im Projekt Track Your Tinnitus das Betriebssystem *Windows Phone* noch nicht berücksichtigt. Da jedoch immer mehr Nutzer von Smartphones auch zu Smartphones mit dem Betriebssystem *Windows Phone* wechseln, soll das Projekt nun um eine *Windows Phone* App erweitert werden. Die Entwicklung dieser App ist das Ziel dieser Arbeit.

1.3 Aufbau der Arbeit

Im zweiten Kapitel wird nun ein genauerer Einblick in das Track Your Tinnitus Projekt und dessen Möglichkeiten gegeben. Hierbei stehen vor allem die Webseite und die iOS und Android App im Vordergrund.

Kapitel 3 beschreibt verwandte Arbeiten, welche sich ebenfalls mit Apps zur Unterstützung von Tinnituspatienten beschäftigen. Dazu wurden bereits vorhandene Arbeiten, sowie bisher nur theoretische Arbeiten betrachtet.

Anschließend werden in Kapitel 4 die konkreten Anforderungen an die *Windows Phone* App vorgestellt. Es sollen dabei sowohl die funktionalen, als auch die nicht-funktionalen Anforderungen genau definiert werden.

Kapitel 5 beschreibt die Architektur der entwickelten App. Hierzu werden sowohl die Ablaufdiagramme, als auch Informationen über die verwendeten Libraries gegeben.

In Kapitel 6 wird die Implementierung der App erläutert. Bei diesem Teil der Arbeit wurde der Schwerpunkt auf die Implementierung der Benachrichtigung und der Speicherung der Daten gelegt.

Im Anschluss erfolgt eine Vorstellung der App in Kapitel 7. Hier werden die Funktionen der App erklärt und die verschiedenen Ansichten gezeigt.

Kapitel 8 beinhaltet den Abgleich der Anforderungen mit dem aktuellen Entwicklungsstand der App. Der Abgleich findet in tabellarischer Form statt und ist in funktionale und nicht-funktionale Anforderungen gegliedert.

Am Ende wird in Kapitel 9 eine Zusammenfassung der Arbeit geliefert. Diese gibt einen Überblick über die einzelnen Aspekte der Arbeit und liefert einen Ausblick auf mögliche Verbesserungen.

2 Track Your Tinnitus

In diesem Kapitel wird das Track Your Tinnitus Projekt näher beleuchtet. Hierbei werden zunächst die Webseite und die vorhandenen Funktionen erläutert. Im zweiten Teil sollen die Möglichkeiten der App veranschaulicht werden.

2.1 Vorstellung der Webseite

Track Your Tinnitus ist ein Forschungsprojekt, welches aus verschiedenen Teilen besteht. Hierfür wurde zunächst eine Webseite erstellt, die als Informationsquelle für interessierte Tinnituspatienten dient. Diese ist die Anlaufstelle für neue Benutzer, wenn sie sich registrieren wollen und bietet darüber hinaus die Möglichkeit Benutzer, Gruppen und statistische Fragebögen zu verwalten. Zudem können auf der Webseite statistischen Fragebögen ausgefüllt und deren Ergebnisse betrachtet werden. Des Weiteren werden auf ihr detaillierte Informationen über das Projekt zur Verfügung gestellt.

Auf der Startseite werden dem Benutzer die ersten Schritte erklärt, um mit der Überwachung seiner Tinnituswahrnehmung und deren Schwankungen zu beginnen. Hierzu wird dem Benutzer der Link zur Registrierung, sowie ein Link zu der App auf den unterschiedlichen Betriebssystemen angezeigt. Des Weiteren bietet es ihm die Möglichkeit einfach auf Informationen über das Projekt und die Datenschutzbestimmungen zuzugreifen.

Der Benutzerbereich bietet Einsicht in die statistischen Fragebögen, die der Benutzer beantwortet hat und noch ausfüllen kann. Diese werden in einer Übersicht mit zusätzlichen Fortschrittsbalken angezeigt, welche durch ihre Füllmenge und Farbe Auskunft über die Menge an bereits beantworteten Fragen liefern. Zusätzlich können in einem weiteren Reiter Therapien, sowie wie deren Form und Dauer eingetragen werden. Weiter können hier die Daten der App betrachtet, sowie Benachrichtigungen eingesehen werden. Außerdem besteht die Möglichkeit verschiedene Einstellungen wie Benutzername und Newsletter zu ändern oder den gesamten Account zu löschen.

2.2 Vorstellung der App

Um die Schwankungen der Tinnituswahrnehmung zu überwachen, wurde eine App für iOS und Android entwickelt. Diese setzt sich aus verschiedenen Funktionen zusammen, die im Folgenden näher erläutert werden.

Nach der Anmeldung wird dem Benutzer der Zugriff auf seine statistischen Fragebögen ermöglicht. Diese können zusätzlich zur Webseite auch innerhalb der App beantwortet werden. Dem Benutzer bleibt es hierbei überlassen, ob er die Fragebögen in der App oder auf der Webseite beantworten möchte, jedoch müssen diese Fragebögen vollständig ausgefüllt sein, bevor mit der Überwachung begonnen werden kann.

Den Hauptteil der App stellt der Fragebogen zur Überwachung dar. Dieser wird beim Öffnen der App oder durch Interaktion mit einer Benachrichtigung angezeigt. Durch diesen Fragebogen wird, zu unterschiedlichen Zeiten, der Zustand des Tinnitus abgefragt. Die Fragen werden vorwiegend durch Regler beantwortet, mit denen der Betroffene zum Beispiel Auskunft über Intensität des Tinnitus geben kann.

Als weitere Option ist es dem Benutzer möglich Benachrichtigungen zu konfigurieren. Diese können entweder fest über einen bereitgestellten Kalender eingestellt werden oder der Benutzer wird zufällig innerhalb von eigens festgelegten Zeiträumen dazu aufgefordert den Fragebogen auszufüllen, um möglichst verschiedene Situationen bei der Ausfüllung des Fragebogens zu haben. Innerhalb der Einstellungen kann der Benutzer einen Klingelton für die Benachrichtigungen festlegen. Außerdem wird innerhalb der Einstellungen die Geräuschemessung während des Ausfüllens des Fragebogens deaktiviert oder aktiviert.

Im Ergebnisbereich der App werden die bisher gesammelten Daten aus den Fragebögen mit Hilfe von Diagrammen visualisiert. Hierbei erhält der Benutzer eine Übersicht über die einzelnen Abschnitte wie beispielsweise Stärke des Tinnitus und Belastungsfaktoren. Des Weiteren werden die Ergebnisse in einer Timeline dargestellt, bei der man jeweils die Ergebnisse, pro Fragebogen zusammengefasst, betrachten kann.

3 Verwandte Arbeiten

In diesem Kapitel sollen andere Arbeiten kurz vorgestellt werden, welche sich ebenfalls mit dem Thema Tinnitus beschäftigen. Hierbei werden jedoch nur Arbeiten und Projekte aufgeführt, die dabei auch versuchen das Betriebssystem *Windows Phone* in ihre Hilfe für Tinnitus Patienten mit einzubeziehen.

Der Anteil an Anwendungen auf dem *Windows Phone* Betriebssystem, die sich mit dem Tinnitus beschäftigen, ist sehr gering. Zum Zeitpunkt des Verfassens dieser Arbeit beläuft sich die Anzahl auf ungefähr 15 Anwendungen. Der größte Teil dieser Anwendungen konzentriert sich vorwiegend auf das Therapieren des Tinnitus. Hierbei setzen sie auf die Klangtherapie, bei welcher mit akustischen Schwingungen, wie zum Beispiel Tönen, Geräuschen und Musik, gearbeitet wird. Einige Beispiele, die ausschließlich für Tinnitusbetroffene entwickelt wurden, sind *White Noise* [3], *Tinnitus* [4], *NoizeMaker* [28] und *Tinnitus Cure* [10].

Die einfachsten Anwendungen bieten dem Benutzer eine ausführliche und wohl geordnete Bibliothek an Informationen über den Tinnitus. Die Informationen reichen dabei von der Grundfrage, was ein Tinnitus überhaupt ist, bis hin zu Möglichkeiten der Eigenbehandlung. Andere einfach gehaltene Anwendungen bieten die Möglichkeit verschiedene Hintergrundgeräusche abzuspielen. Hierbei wird versucht für den Benutzer eine beruhigende Umgebung durch Vogelgezwitscher oder Meeresrauschen zu erzeugen. Dadurch soll der Benutzer ein ausgeprägtes Entspannungsgefühl, sowie eine Abschwächung des Stresslevels erfahren.

Weiterführende Anwendungen bieten zusätzlich zu verschiedenen Hintergrundgeräuschen die Möglichkeit einen Therapieplan auszuwählen und diesen zu verwalten. Hierbei wird der Benutzer mehrmals täglich aufgefordert sich in einer ruhigen Umgebungen zu entspannen, um dann über eine festgesetzte Zeitspanne verschiedenen beruhigenden Klängen zu lauschen. Die Häufigkeit und Dauer werden dabei von der Anwendung festgelegt und festgehalten, wodurch dem Benutzer ein Überblick über seinen Therapieverlauf gegeben werden kann. Einige Anwendungen bieten dabei zusätzlich die Möglichkeit eigene Musik für die Therapie zu nutzen. Diese wird dann vor dem Abspielen

3 Verwandte Arbeiten

durch verschiedene Filter bearbeitet, um störende oder stresserzeugende Töne und Tonlagen zu entfernen.

Die im vorherigen Abschnitt behandelten Anwendungen stellen einen Überblick über die bisher vorhandenen Möglichkeiten für Tinnitus-Betroffene mit einem *Windows Phone* Smartphone dar. Im Anschluss an Nachforschungen über die vorhandenen Möglichkeiten wurde nach bisher nur theoretischen bzw. schriftlich festgehaltenen Arbeiten gesucht. Diese beschränkten sich zum Zeitpunkt des Verfassens dieser Arbeit jedoch auf ein Minimum. Abgesehen von Projekten und Forschungen zum Tinnitus in rein medizinischer Herangehensweise wurden weder Projekte mit Unterstützung von Apps noch solche veröffentlicht, die ausschließlich auf dem Betriebssystem *Windows Phone* Anwendungen basieren. Es gibt jedoch verschiedene Arbeiten, welche sich mit klinisch und psychologischen Studien auf dem Smartphone befassen. In diesen werden unter anderem verschiedene Methoden zur Datenerfassung mit Hilfe von Smartphones in Studien beschrieben. [24, 26, 20, 25, 8, 2, 27, 23, 5, 9, 1]

4 Anforderungen

Dieses Kapitel beschreibt die gestellten Anforderungen an die Track Your Tinnitus App unter dem Betriebssystem *Windows Phone*. Hierbei wurden die Anforderungen in funktionale und nicht-funktionale Anforderungen aufgeteilt.

4.1 Funktionale Anforderungen

Funktionale Anforderungen beschreiben die wichtigsten Funktionen einer Anwendung, die dem Benutzer zur Verfügung stehen sollen. Diese werden in der folgenden Tabelle aufgeführt und näher erläutert.

Nr.	Anforderung	Beschreibung
1	Ausfüllen der Fragebögen zur Überwachung der Schwankung der Tinnituswahrnehmung	Es sollte dem Benutzer ein Fragebogen angezeigt werden, mit welchem er die Schwankungen der Tinnituswahrnehmung dokumentieren kann.
2	Registrieren und Anmelden in der App	Die App kann ohne ein Benutzerkonto nicht genutzt werden. Deshalb sollte es die Möglichkeit geben sich direkt auf dem Gerät ein solches Konto zu erstellen. Außerdem sollte man sich mit einem bereits bestehend Konto anmelden können.
3	Nutzbarkeit ohne Internetverbindung	Für das Benutzen der App sollte eine funktionierende Internetverbindung keine Voraussetzung sein. Es ist nicht davon auszugehen, dass der Benutzer zu jedem Verwendungszeitpunkt einen ausreichenden Internetempfang haben kann.

Tabelle 4.1: Tabellenübersicht der funktionalen Anforderungen

Nr.	Anforderung	Beschreibung
4	Festlegen des Klingeltons einer Benachrichtigung	Da die Möglichkeit besteht, dass der Klingelton einer Benachrichtigung in einigen Fällen vom Tinnitus überdeckt werden könnte, sollte der Benutzer die Option haben den Klingelton der Benachrichtigung zu verändern.
5	Einstellung der Benachrichtigungszeiten (zufällig)	Der Benutzer sollte die Möglichkeit haben den Zeitraum für zufällige Benachrichtigungen festlegen zu können, so dass sie den Wünschen des Benutzers entsprechen.
6	Einstellung der Benachrichtigungszeiten (planmäßig)	Die Benachrichtigungen sollten zu einem konkreten eingestellten Zeitpunkt stattfinden können, falls der Benutzer keine zufälligen Benachrichtigungen wünscht.
7	Ausfüllen der statistischen Fragebögen	Da vollständig ausgefüllte statistische Fragebögen Voraussetzung für das Benutzen der App sind, sollten diese auch über die App ausgefüllt werden können.
8	Synchronisierung der Antworten von statistischen Fragebögen	Es sollte die Möglichkeit bestehen, während des Ausfüllens eines Fragebogens zwischen der Webseite und der App zu wechseln.
9	Messung der Lautstärke von Umgebungsgeräuschen	Um festzustellen, ob der Tinnitus durch Umgebungsgeräusche beeinflusst oder durch diese überdeckt wird, sollte während des Ausfüllens des Fragebogens zur Überwachung der Schwankung der Pegel der Hintergrundgeräusche durch die App gemessen werden.
10	Slider ohne initialen Wert	Der Slider eines Fragebogens darf keinen initialen Wert besitzen, da sich Benutzer von voreingestellten Werten beeinflussen lassen.

Tabelle 4.2: Tabellenübersicht der funktionalen Anforderungen

4.2 Nichtfunktionale Anforderungen

Der folgende Abschnitt umfasst die nicht-funktionalen Anforderungen der Anwendung zusammengefasst in einer Tabelle. Diese definieren die Anforderung an die Anwendung hinsichtlich des Aussehens und der Handhabung.

Nr.	Anforderung	Beschreibung
1	Darstellung der App und des Fragebogens	Die Track Your Tinnitus App sollte auf dem Windows Phone ähnlich bis identisch zur bereits vorhanden iOS und Android App sein, um dem Benutzer auch beim Wechseln des Smartphones eine gewohnte Bedienung zu gewährleisten.
2	Farbgebung und Bedienungselemente	Die App sollte bereits vorhandene Farbschemata und Bedienungselemente benutzen, um einen einfachen Umstieg zwischen verschiedenen Smartphones zu ermöglichen.
3	Ausfüllen des Fragebogens zur Überwachung der Schwankungen in kurzer Zeit	Der Fragebogen zur Überwachung der Schwankungen zur Tinnituswahrnehmung muss mehrmals täglich vom Benutzer ausgefüllt werden. Deshalb sollte das Ausfüllen in möglichst kurzer Zeit zu erledigen sein.

Tabelle 4.3: Tabellenübersicht der nicht-funktionalen Anforderungen

5 Architektur

In diesem Kapitel wird die Architektur der Track Your Tinnitus App für *Windows Phone* beschrieben. Zuerst wird ein typischer Ablauf der App und ihrer Funktionen gezeigt, auf welchen eine Beschreibung der Datenstruktur folgt. Im Anschluss daran werden die verwendeten Libraries erklärt, bevor am Ende des Kapitels das Klassendiagramm der App vorgestellt wird.

5.1 Ablauf

Im folgenden Abschnitt wird ein typischer Ablauf bei der Benutzung der App erläutert und in Abbildung 5.1 gezeigt. Zur Überwachung der Schwankung der Tinnituswahrnehmung mit Hilfe der App wird ein Benutzerkonto vorausgesetzt. Hierzu wird dem Benutzer die Möglichkeit geboten ein Benutzerkonto zu registrieren, mit welchem er sich auch nach der Bestätigung der E-Mail Adresse anmelden kann. Sollte der Benutzer bereits ein Benutzerkonto besitzen ist es auch möglich sich direkt einzuloggen. Nach dem erfolgreichen Login überprüft die App, ob alle statistischen Fragebögen vollständig beantwortet wurden. Falls sie noch nicht beantwortet wurden, müssen diese zwingend ausgefüllt werden, da sonst keine Überwachung der Schwankungen möglich ist. Im Anschluss kann der Benutzer auswählen, ob er die Benachrichtigungseinstellungen überprüfen will. Sollte er diese überprüfen wollen, werden die Benachrichtigungseinstellungen geöffnet. Falls er dies nicht möchte wird für den Benutzer direkt der Fragebogen zur Überwachung der Schwankungen der Tinnituswahrnehmung geöffnet.

Vom Fragebogen zur Überwachung der Tinnituswahrnehmung aus lässt sich das Hauptmenü öffnen, mit welchem man in alle Bereiche der App navigieren kann. Dadurch lassen sich die Benachrichtigungszeiten einstellen, sowie generelle Einstellungen, wie Privatsphäreinstellungen, vornehmen. Sollten Benachrichtigungen eingestellt worden sein, wird der Benutzer beim Klicken der Benachrichtigung direkt zum Fragebogen zur Überwachung der Schwankungen der Tinnituswahrnehmung geleitet.

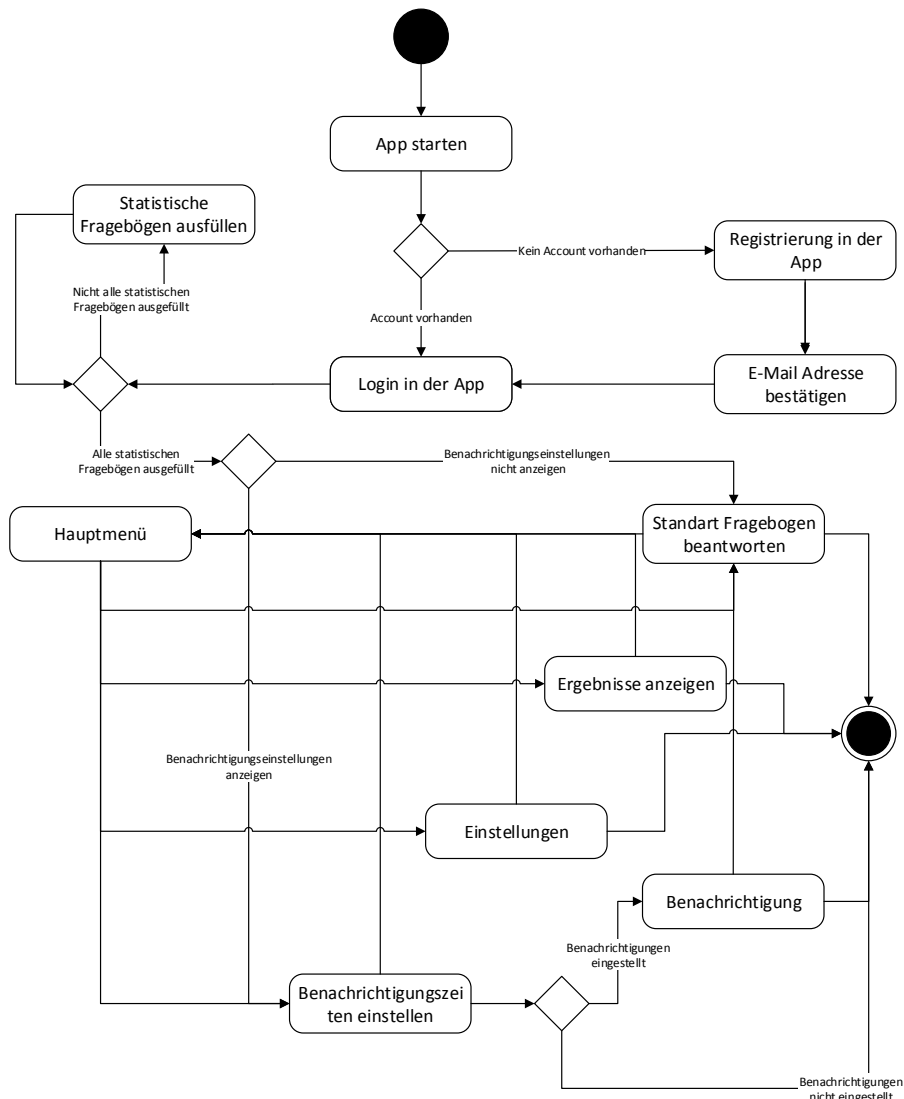


Abbildung 5.1: Ablaufdiagramm

5.2 Datenstruktur

Für die lokale Datenstruktur der Track Your Tinnitus App wurde ein *SQL Server Compact* [16] verwendet. Die Implementierung der App benutzt zusätzlich den *IsolatedStorageSettings* [14], um benutzerspezifische Daten zu speichern (siehe Kapitel 6.2). In Abbildung 5.2 ist die Datenstruktur des Fragebogens zur Überwachung der Tinnituswahrnehmung (vgl. Abbildung 5.2a) und die Datenstruktur für die statistischen Fragebögen (vgl. Abbildung 5.2b und c) dargestellt.

Die Datenstruktur des Fragebogens zur Überwachung wurde der Datenstruktur der Webseite angepasst (vgl. [7]). Jeder abgespeicherte Fragebogen enthält einen Eintrag für jede Frage und der dazugehörigen Antworten, welche in *Question1-Question8*

gespeichert werden. Mit *Autosaved* wird festgehalten, ob die Antworten des Fragebogens vom Benutzer oder automatisch gespeichert wurden. Zusätzlich zum Zeitstempel für das Erzeugen des Fragebogens *Created_at* und dem Zeitpunkt der Aktualisierung *Updated_at*, wird ein Zeitstempel *Save_date* gespeichert. Dieser gibt an, wann der Benutzer die Antworten gespeichert hat. Falls ein Benutzer den Fragebogen aufgrund einer Benachrichtigung ausfüllt, wird der Zeitpunkt der Benachrichtigung *Notification_date* festgehalten. Des Weiteren wird in *Notification_fixed* abgespeichert, ob der Benutzer den Benachrichtigungszeitpunkt festgelegt hat. In *Soundlevel* wird die Lautstärke der Umgebungsgeräusche gespeichert, falls dies vom Benutzer erlaubt wurde. Das Feld *User_agent* beinhaltet Informationen über das Geräte (Hersteller und Modellbezeichnung), sowie die Version des Betriebssystems und Versionsnummer der App. Für die Zuordnung wird aus dem *IsolatedStorageSetting* die Benutzer-ID in den Feldern *user_id* und *user_id_reference* gespeichert.

Die Datenstruktur der statistischen Fragebögen ist in einem ähnlichen Format gehalten, hier werden die Antworten der Fragen in den Feldern *Question0-Question14* bzw. *Question0-Question10* abgespeichert. Die Zeitstempel *Save_date* und *Updated_at* halten fest, wann der Fragebogen gespeichert bzw. zuletzt aktualisiert wurde. Auch bei der Datenstruktur des Fragebogens nach dem schlimmsten Symptom wurde dieses Schema benutzt (vgl. Abbildung 5.3b).



Abbildung 5.2: Ansicht der Datenstruktur

Die Datenstruktur für das Speichern der benutzerdefinierten Benachrichtigungen wurde in zwei Felder aufgeteilt, *Id* und *Date*. Beide Felder wurden als Primärschlüssel deklariert, um Eindeutigkeit der Termine, sowie einen schnellen Zugriff auf die Termine zu gewährleisten. Dabei wird das Feld *Date* als Zeitstempel gespeichert, welcher den Benachrichtigungszeitpunkt darstellt.

ReminderData		
PS	Id	int
PS	Date	datetime

(a) Benutzerdefinierte Benachrichtigung

SymptomFormAnswers		
PS	Id	int
	Question0	int
	Save_date	datetime
	Updated_at	datetime
	User_id	int
	User_id_reference	int

(b) Fragebogen zum schlimmsten Symptom

Abbildung 5.3: Ansicht der Datenstruktur

5.3 Libraries

Während der Entwicklung der App traten aufgrund der spezifischen Anforderungen verschiedene Problemstellungen auf, welche nicht durch die vorhandenen Möglichkeiten vollständig gelöst werden konnten. Aufgrund dessen wurden verschiedenen Libraries eingebunden, welche im folgenden Abschnitt näher erläutert werden.

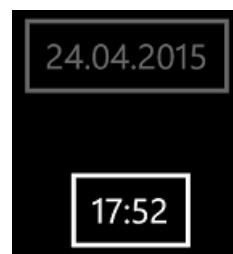
5.3.1 Windows Phone Toolkit

Das *Windows Phone Toolkit* stellt Entwicklern eine große Sammlung an Controllern zur Verfügung, mit deren Hilfe eine konsistente Oberfläche bei *Windows Phone* Apps erzeugt werden kann.

Die wichtigsten beiden Controller aus dieser Library, welche bei der Entwicklung der App verwendet wurden, sind der *Time-* bzw. *DatePicker* und der *ToggleSwitch*. Diese sind in der Abbildung 5.4 (a) und (b) zu sehen.



(a) ToggleSwitch



(b) Date-/TimePicker

Abbildung 5.4: Verwendeter ToggleSwitch und Date-/TimePicker

Der Picker bildet dabei eine einfache und übersichtliche Ansicht zur Auswahl der Zeit bzw. des Datums. Hierbei werden die Uhrzeit und das Datum zunächst in einer auswählbaren Standardansicht dargestellt und eine übersichtliche Methode bei der Auswahl einer Uhrzeit bzw. des Datums ermöglicht, siehe Abbildung 5.5.

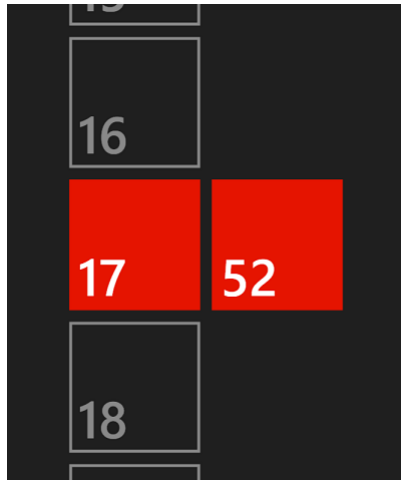


Abbildung 5.5: Ansicht der Zeitauswahl

Der ToggleSwitch bietet eine für den Benutzer einfache und schnelle Darstellung eines einfach An/Aus-Buttons, welcher bei der App benutzt wurde, um eine möglichst ähnliche Darstellung zur iOS und Android App zu erzielen.

5.3.2 Windows Phone Calender Control

Mit der *Windows Phone Calender Control* wird dem Entwickler eine einfache Möglichkeit zur Bereitstellung eines Kalenders innerhalb einer App gegeben.

Der durch diese Library bereitgestellte Kalender bietet eine Auswahl des Datums an, sowie das farbliche Markieren von bestimmten Daten, siehe Abbildung 5.6. Jedoch besitzt der Kalender Restriktionen, weshalb er nicht vollständig den Anforderung entspricht. Der Kalender bietet zum Zeitpunkt der Arbeit keine Möglichkeit einer Wochen- oder Tagesansicht. Des Weiteren ist eine Änderung der Größe und Ansicht des Kalenders nicht möglich. Eine Alternative wurde zum Zeitpunkt der Arbeit nicht gefunden.

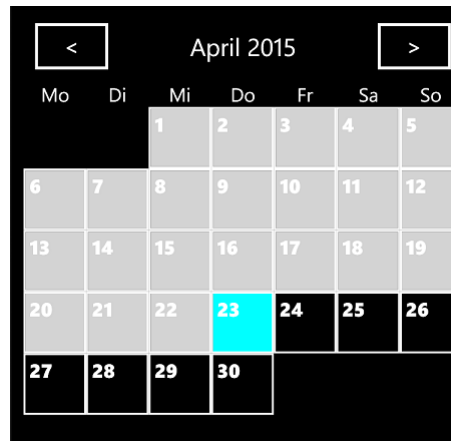


Abbildung 5.6: Ansicht des Kalenders mit verschiedenen Farben

5.4 Klassen der App

Durch die Möglichkeiten des *Model View ViewModel* Konzepts [29] besteht jede Ansicht der Track Your Tinnitus App aus einer *XAML*-Datei und der dazugehörigen Klasse. Aus diesem Grund wurde bei der Implementierung keine große Vererbungsstruktur zwischen den Klassen verwendet. In Abbildung 5.7 sind alle Klassen der App dargestellt. Die Klasse *MainPage* ist für die Darstellung des Fragebogens zur Überwachung der Schwankungen der Tinnituswahrnehmung verantwortlich und wird als erstes aufgerufen, wenn die App geöffnet wird.

Sollte sich der Benutzer noch nicht eingeloggt oder nicht alle statistischen Fragebögen beantwortet haben, wird er direkt zu den entsprechenden Klassen weitergeleitet. Die Klasse *Login* zeigt ein Formular zur Anmeldung in der App und bietet dem Benutzer die Möglichkeit zum Registrierungsformular zu wechseln. Die Klasse *Registration* bietet dem Benutzer ein Formular zum Erstellen eines Benutzerkontos an und umfasst Methoden zur Überprüfung der eingegebenen Daten, sowie zum Übertragen der Daten an den Server.

Die Klassen *StaticForm*, *PersonalForm* und *SymptomForm* sind für die Darstellung der statistischen Fragebögen verantwortlich. Des Weiteren bieten sie Methoden zur Verarbeitung und Synchronisierung der Antworten der statistischen Fragebögen.

Mit der Klasse *Countdown* wird der Countdown für das Schließen der App nach erfolgreichem Speichern des Antworten des Fragebogens zur Überwachung der Tinnituswahrnehmung dargestellt. Zusätzlich beinhaltet Sie eine Methode zum Schließen der App.

Die Klassen *Options* und *NotificationOptions* erzeugen die Ansicht für die Einstellungen und Benachrichtigungseinstellungen. Außerdem werden Methoden bereitgestellt, mit denen die aktuellen Einstellungen gespeichert und geladen werden. Zusätzlich werden Einstellungen für anderen Ansichten, falls nötig, aktualisiert und bei Bedarf zufällige Benachrichtigungen erzeugt.

Der Kalender für die benutzerdefinierten Benachrichtigungen wird von der Klasse *Calender* dargestellt. Die Klasse beinhaltet nur eine Methode zum Auswählen eines Tages für eine Benachrichtigung. Mit Hilfe der Klasse *Colorconverter* werden innerhalb des Kalenders die Tage farblich voneinander abgesetzt.

Die Klasse *UserNotification* ist für das Speichern und Anzeigen von benutzerdefinierten Benachrichtigungen zuständig und bietet zusätzlich Methoden zum Löschen von bereits erstellten Benachrichtigungen.

Die Klassen *Answers* und *ReminderData* sind Teilklassen der Klasse *DatabaseContext*. Sie bieten Methoden zum Interagieren mit den jeweiligen Tabellen innerhalb der lokalen SQL-Datenbank an. Die Klasse *DatabaseContext* bildet die Verbindung zwischen App und Datenbank. Sie beinhaltet Methoden zur Überprüfung der Verbindung, sowie zur Verwaltung der Datenbank an.

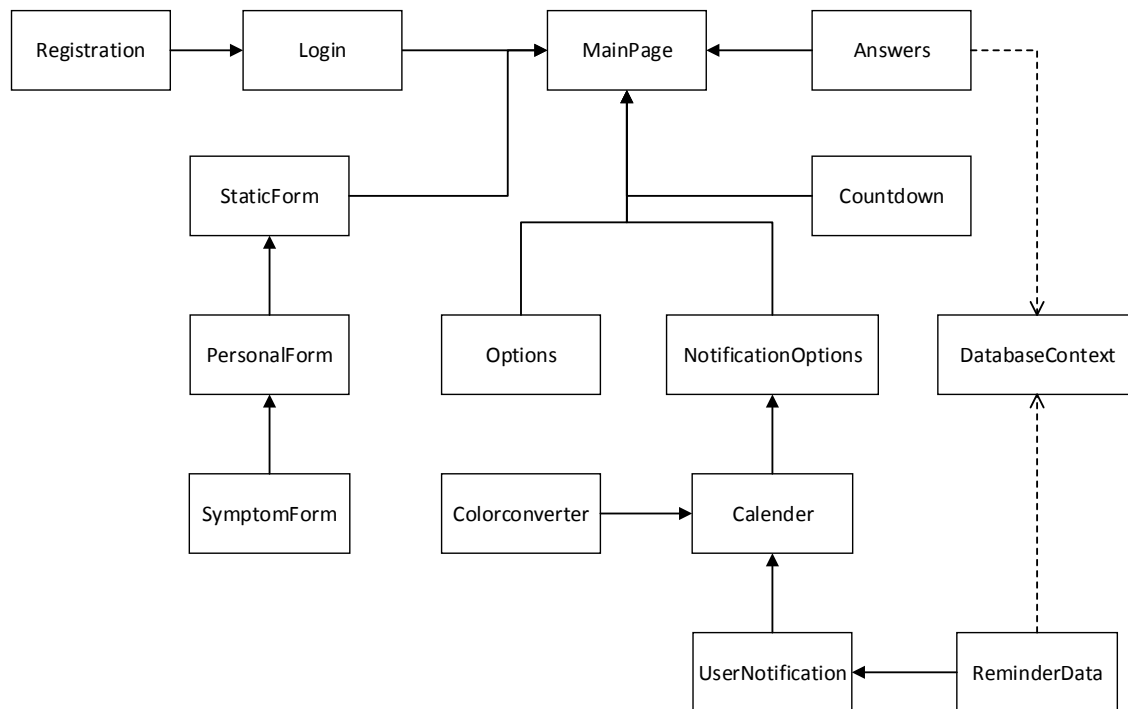


Abbildung 5.7: Klassendiagramm

6 Implementierung und Implementierungsaspekte

Im folgenden Kapitel werden verschiedene Aspekte der Implementierung bei der Entwicklung der Track Your Tinnitus App näher betrachtet. Hierbei soll zunächst die Implementierung der Benachrichtigungen näher erläutert werden. Anschließend wird die Speichernutzung innerhalb der App beschrieben. Dazu gehören die Ergebnisse der Fragebögen, sowie die vom Benutzer gewählten Einstellungen.

6.1 Benachrichtigungen in der App

Die wichtigste Funktion der Track Your Tinnitus App stellt das Ausfüllen der Fragebögen zur Überwachung der Schwankungen der Tinnituswahrnehmung dar. Dazu soll der Benutzer möglichst in unregelmäßigen Abständen zum Ausfüllen des Fragebogens aufgefordert werden. Das bietet die Möglichkeit die Schwankungen über einen großen Zeitraum umfassend aufzuzeichnen. Zur Benachrichtigung des Benutzers, während die App im Hintergrund ausgeführt wird, bietet das Betriebssystem *Windows Phone*, wie auch iOS und Android, verschiedenen Möglichkeiten. Diese Möglichkeiten sollen in den folgenden Abschnitten näher erläutert werden.

6.1.1 Remote oder Local Notification

Windows Phone bietet dem Entwickler unterschiedliche Möglichkeiten zur Benachrichtigung des Benutzers. Diese können generell in Remote Notifications und Local Notifications eingeteilt werden.

Die Remote Notifications werden über das Internet über einen Server an das jeweilige Geräte gesendet und erfordern daher eine funktionierende Internetverbindung. Der Vorteil besteht dabei darin, dass die Informationen auf dem Server den Inhalt der Benachrichtigung bestimmen. Diese Benachrichtigungen werden hierbei auf *Windows Phone* durch den *Microsoft Push Notification Service* bereitgestellt. Diese können dabei zusätz-

lich in ihrer Darstellung entweder als Push- oder Tile Notifications präsentiert werden. [15]

Die Local Notifications werden im Gegensatz zu den Remote Notifications auf dem Geräte geplant und bereitgestellt, auf welchem sich die Anwendung befindet. Daher werden für diese keine Internetverbindung benötigt, jedoch müssen die Inhalte der Benachrichtigungen bereits bei der Entwicklung weitestgehend festgelegt werden. Auf dem *Windows Phone* werden hierfür *Alarm* und *Reminder* bereitgestellt, welche im folgenden Abschnitt näher erläutert werden sollen. [13]

Da eine funktionierende Internetverbindung keine Voraussetzung für das Nutzen der Track Your Tinnitus App darstellen soll, wurde bei der Entwicklung eine Local Notification genutzt.

6.1.2 Alarm und Reminder

Ein Alarm ist eine Benachrichtigungsfunktion, bei welcher einem Benutzer eine Benachrichtigung mit einem variablen Titel geboten wird. Zum Benachrichtigungszeitpunkt wird dabei gleichzeitig zum Anzeigen des Alarms ein über die App einstellbarer Klingelton abgespielt. Dieser wird mit zunehmender Dauer immer lauter. Außerdem hat der Benutzer die Möglichkeit den Alarm auszuschalten oder die Schlummerfunktion zu aktivieren. Diese ist dabei bei dem Alarm immer auf 9 Minuten festgesetzt.

Ein Reminder bietet neben einem variablen Titel zusätzlich die Möglichkeit der Benachrichtigung einen kurzen Text hinzuzufügen. Dieser wird dann zum Benachrichtigungszeitpunkt ebenfalls angezeigt. Im Gegensatz zum Alarm kann bei einem Reminder der Klingelton jedoch nicht über die App festgelegt werden. Für Reminder ist dieser durch das Gerät festgelegt. Der Klingelton hat dabei Ähnlichkeit mit der Funktionsweise eines SMS-Klingeltons. Auch der Reminder bietet ein Ausschalten, sowie eine Schlummerfunktion an. Die Zeit für die Schlummerfunktion ist dabei aber variabel einstellbar.

Aufgrund der Eigenschaften des Reminders wurde dieser als Benachrichtigungsfunktion für die App gewählt.

6.1.3 Implementierung der Reminder

In den folgenden Abschnitten wird die Implementierung des Reminders vorgestellt. Dessen Implementierung in der Track Your Tinnitus App wird in Listing 6.1 dargestellt. Der Code wird im Folgenden Schritt für Schritt erläutert:

1. Mit der Methode *removeReminder* (Zeile 3) wird überprüft, ob ein alter Reminder vorhanden ist und dieser dann entfernt.
2. Damit gesichert ist, dass der Benutzer auch benachrichtigt werden will, wird dies zusätzlich durch die *if*-Abfrage (Zeile 4) überprüft.
3. Zunächst wird dann ein Reminder-Objekt erstellt (Zeile 6).
4. Durch die Eigenschaft *BeginTime* wird dem Reminder eine Anfangszeit zugewiesen, welche durch die Methode *createReminderTime* erzeugt wird. Die Methode *createReminderTime* erzeugt anhand der Benachrichtigungseinstellungen die Anfangszeit und wird in Abschnitt 6.1.4 näher erläutert. Die Eigenschaft *ExpirationTime* legt fest, nach welchem Zeitraum die Benachrichtigung ausläuft, sollte der Benutzer nicht reagieren.
5. Der Titel und Inhalt der Benachrichtigung werden durch die Eigenschaften *Title* und *Content* festgelegt (Zeile 10 - 11).
6. Mit der Eigenschaft *NavigationUri* (Zeile 12) wird festgelegt, zu welchem Abschnitt der App der Benutzer weitergeleitet wird, wenn er die Benachrichtigung berührt.
7. Zum Schluss wird der Reminder zu den geplanten Aktionen hinzugefügt (Zeile 13), durch den der Reminder automatisch zur festgelegten Zeit aktiviert wird.

Listing 6.1: Erzeugen eines Reminder

```

1 public void setReminder()
2 {
3     removeReminder();
4     if (isToNotify)
5     {
6         Reminder tmpReminder = new Reminder("currentReminder");
7
8         tmpReminder.BeginTime = createReminderTime();
9         tmpReminder.ExpirationTime = tmpReminder.BeginTime.AddMinutes(30);
10        tmpReminder.Title = "Track Your Tinnitus";
11        tmpReminder.Content = "Bitte füllen sie den Fragebogen aus! \n" +
12                               tmpReminder.BeginTime.ToString();
13        tmpReminder.NavigationUri = new Uri("/MainPage.xaml", UriKind.Relative);
14        ScheduledActionService.Add(tmpReminder);
15        showReminder();
16    }
17 }

```

6.1.4 Erzeugen der Benachrichtigungszeit

In den folgenden beiden Abschnitten wird erläutert wie die *createReminderTime*-Methode die Startzeit für eine Benachrichtigung festlegt. Dazu wird zunächst die Methode zum Erzeugen der zufälligen Benachrichtigungen vorgestellt, bei welcher vom Benutzer die Benachrichtigungsart „Standard“ ausgewählt wurde. Im Anschluss daran wird die Implementierung der benutzerdefinierten Benachrichtigungen vorgestellt.

Die Auswahl, welche der Benachrichtigungsarten verwendet werden soll, ist in Listing 6.2 implementiert. Hierbei entspricht der *Boolean*-Wert der Variable *notificationTypeStandard* (Zeile 4) der ausgewählten Benachrichtigungsart. **True** entspricht dabei der Benachrichtigungsart „Standard“ und **False** der Benachrichtigungsart „Benutzerdefiniert“.

Listing 6.2: Überprüfen der Benachrichtigungsart

```

1 public DateTime createReminderTime()
2 {
3     DateTime tmpDateTime = new DateTime();
4     if (notificationTypeStandard)
5     {
6         tmpDateTime = setReminderBeginTime();
7     }
8     else
9     {
10        tmpDateTime = getNextListedReminder();
11    }
12    return tmpDateTime;
13 }

```

Zufällige Benachrichtigung

Listing 6.3 zeigt die Implementierung für die erzeugte Zeit der zufälligen Benachrichtigungen. In den Zeilen 3 bis 6 werden temporäre Variablen zum Speichern der Zeiten erzeugt, wobei die Variable *day* den Index des Tages in der Woche enthält. Zeile 8 zeigt wie der Endzeitpunkt des aktuellen Tages aus dem Speicher geladen wird. Im Folgenden wird in den Zeilen 10 bis 17 überprüft, ob an diesem Tag noch eine Benachrichtigung stattfinden soll. Nachdem der Tag für die nächste Benachrichtigung gesichert wurde, werden die für den Tag entsprechenden Anfangs- und Endzeitpunkte geladen (Zeile 19 bis 21). Anschließend wird in den Zeilen 23 bis 26 der Zeitraum für die zufällige Benachrichtigungszeit konkretisiert. Am Ende wird mit der Methode *createRandomTime* ein Zeitpunkt festgelegt (Zeile 28) und dann zurückgegeben.

Listing 6.3: Erzeugen der zufälligen Zeit

```

1 public DateTime setReminderBeginTime ()
2 {
3     DateTime tmpDateTimeReminder;
4     DateTime tmpStart;
5     DateTime tmpEnd;
6     int day = (int)DateTime.Now.DayOfWeek;
7     day = day + 7;
8     tmpEnd = (DateTime)notificationSettings [day.ToString ()];
9
10    if ((notificationCounterTmp < notificationCounter) && (DateTime.Now.TimeOfDay <
        tmpEnd.TimeOfDay))
11    {
12        day = (int)DateTime.Now.DayOfWeek;
13    }
14    else
15    {
16        day = (int)DateTime.Now.DayOfWeek + 1;
17    }
18
19    tmpStart = (DateTime)notificationSettings [day.ToString ()];
20    day = day + 7;
21    tmpEnd = (DateTime)notificationSettings [day.ToString ()];
22
23    if ((tmpStart.TimeOfDay < DateTime.Now.TimeOfDay)&&(day ==
        (int)DateTime.Now.DayOfWeek))
24    {
25        tmpStart = DateTime.Now;
26    }
27
28    tmpDateTimeReminder = createRandomTime (tmpStart.TimeOfDay, tmpEnd.TimeOfDay);
29    return tmpDateTimeReminder;
30 }
31

```

Benutzerdefinierte Benachrichtigung

In dem Listing 6.4 wird gezeigt wie eine benutzerdefinierte Benachrichtigung ausgelesen wird. Hierzu werden zunächst alle vom Benutzer festgelegten Termine für Benachrichtigung aus der lokalen Datenbank in eine Liste geladen (Zeile 4). Diese werden danach anhand der Zeitpunkte so sortiert, dass das erste Element der Liste die aktuellste Benachrichtigung darstellt (Zeile 5). Anschließend wird der Zeitpunkt noch einmal überprüft und aus der lokalen Datenbank gelöscht (Zeilen 7 bis 11), bevor er zurückgegeben wird (Zeile 13).

Listing 6.4: Auslesen der benutzerdefinierten Benachrichtigung

```

1 public DateTime getNextListedReminder()
2 {
3     DateTime tmpNextDateTime = new DateTime();
4     List<ReminderData> listOfAllReminder =
        connector.GetTable<ReminderData>().ToList<ReminderData>();
5     listOfAllReminder.Sort((rd1, rd2) => DateTime.Compare(rd1.Date, rd2.Date));
6
7     do
8     {
9         tmpNextDateTime = listOfAllReminder.First<ReminderData>().Date;
10        connector.ReminderDatas.DeleteOnSubmit(listOfAllReminder.First<ReminderData>());
11        connector.SubmitChanges();
12    } while (tmpNextDateTime < DateTime.Now);
13
14    return tmpNextDateTime;
15 }

```

6.2 Speichernutzung in der App

Ein weiterer Teil der Implementierung der Track Your Tinnitus App ist die Speicherung der Daten. Da die App auch funktionieren soll, während keine Internetverbindung vorhanden ist oder diese eventuell instabil ist, müssen die Antworten des Fragebogen zur Überwachung der Schwankung der Tinnituswahrnehmung auf dem Handy gesichert werden. Diese müssen dann bis zur nächste Verbindung zum Server auf dem Handy gespeichert werden. Zusätzlich ist es notwendig die Einstellungen des Benutzers zu speichern. Die Einstellungen sind vom Benutzer jederzeit veränderbar und sollten möglichst schnell in den Speicher übernommen werden.

Aus diesen Gründe wurde der Speicher in der Track Your Tinnitus App aufgeteilt. So wird für die Ergebnisse des Fragebogens eine lokale SQL-Datenbank genutzt, während für die Speicherung der Einstellungen der *IsolatedStorageSetting* genutzt wird. Die beiden Speicherarten werden in den folgenden Abschnitten vorgestellt und deren Implementierung erläutert.

6.2.1 Lokale SQL-Datenbank

Um eine große Menge an Daten zu speichern, bietet sich auf dem *Windows Phone* Betriebssystem eine lokale SQL-Datenbank an. Diese bietet mehr Möglichkeiten als

beispielsweise eine Speicherung der Daten durch Textfiles oder andere Speichermethoden. Hierbei wird ein *Microsoft SQL Server Compact* benutzt, ein relationales eingebettetes Datenbanksystem. Dadurch wird einer Anwendung ermöglicht einen Großteil der Möglichkeiten einer SQL-Datenbank zu nutzen, ohne auf einen externen Dienst oder Installation angewiesen zu sein. Zusätzlich wird von der SQL-Datenbank LinQ unterstützt, welches die Möglichkeit bietet Objekte und Werte ohne komplexe SQL-Queries abzurufen oder zu speichern. Hierzu lässt sich mit Hilfe von *Visual Studio* eine Datenbank-Klasse erzeugen, welche dann alle Methoden und Verbindungsdaten zur Nutzung der Datenbank und LinQ enthält.

Der folgende Abschnitt soll nun die Verbindung und die Nutzung der Datenbank anhand der benutzerdefinierten Benachrichtigungen erläutern. Bei diesen werden vom Benutzer ein Datum und eine Uhrzeit für eine Benachrichtigung festgelegt, welche dann in der Datenbank abgespeichert werden. In Listing 6.5 wird der hierzu implementierte Quelltext gezeigt.

Zunächst wird die Verbindung zur Datenbank initialisiert, indem ein *DatabaseContext*-Objekt erzeugt wird (Zeile 1). Dieses Objekt muss immer mit einem *ConnectionString* erzeugt werden. Dazu stehen in der *DatabaseContext*-Klasse zwei Eigenschaften zur Verfügung, der *ConnectionString* und der *ConnectionStringReadOnly*. Der erste String bietet dabei dem *DatabaseContext*-Objekt die Möglichkeit sowohl lesend als auch schreibend auf die Datenbank zuzugreifen. Während der zweite String nur eine lesende Berechtigung zur Verfügung stellt. Danach wird ein *ReminderData*-Objekt erzeugt, dessen Aufbau und Eigenschaften der Tabelle *ReminderData* in der Datenbank entsprechen (Zeile 2). Anschließend werden die Daten für die benutzerdefinierte Benachrichtigung in das Objekt geschrieben, bevor es mit dem Befehl *connector.ReminderData.InsertOnSubmit* an den DataContext angeheftet wird und in der Datenbank in die Tabelle *ReminderData* mit dem *SubmitChanges*-Befehl eingetragen wird (Zeilen 6 bis 8). Zum Schluss werden das Datum und die Uhrzeit wieder aus dem Settingsspeicher gelöscht und der Benutzer zur vorherigen Seite zurück geleitet (Zeilen 9 bis 11).

Listing 6.5: Abspeichern der benutzerdefinierten Benachrichtigung

```

1 DatabaseContext connector = new DatabaseContext(DatabaseContext.ConnectionString);
2 ReminderData tmpReminder = new ReminderData();
3
4 private void Save_Click(object sender, RoutedEventArgs e)
5 {
6     tmpReminder.Date = new DateTime(tmpDate.Year, tmpDate.Month, tmpDate.Day,
        tmpTime.Hour, tmpTime.Minute, 0);

```

```
7 connector.ReminderData.InsertOnSubmit(tmpReminder);
8 connector.SubmitChanges();
9 tmpCal.userDateSettings.Remove("Date");
10 tmpCal.userDateSettings.Remove("Time");
11 NavigationService.GoBack();
12 }
```

Das Auslesen und Löschen von Daten aus der Datenbank ist in Listing 6.4 gezeigt. Hier werden in Zeile 4 mit dem Befehl *GetTable* alle Zeilen der Tabelle *ReminderData* ausgelesen und in eine Liste aus *ReminderData*-Objekten eingefügt. Das Löschen der Daten erfolgt, wie auch das Einfügen in die Datenbank, in zwei Schritten. Zunächst wird die zu löschende Zeile als Objekt dem *DataContext* übergeben. Dabei muss das Objekt, wie auch beim Einfügen, der Tabelle entsprechen, also ein *ReminderData*-Objekt sein (Zeile 10). Anschließend wird die Aktion an die Datenbank mit dem *SubmitChanges*-Befehl übertragen (Zeile 11).

6.2.2 IsolatedStorageSettings

Mit Hilfe des *IsolatedStorageSettings* lassen sich viele Einstellung einer Anwendung schnell und einfach abspeichern und auslesen. Dabei werden alle Daten in einer lokalen Datei gesichert, auf die ausschließlich die dazugehörige Anwendung Zugriff hat. Sie werden in einem *Dictionary* als Paare gespeichert, welche aus einem Namen und einem Wert bestehen. Diese können dann später wieder abgerufen werden. Diese Paaren bleiben unabhängig vom Zustand der Anwendung gespeichert, also egal ob die Anwendung gestartet oder gestoppt wird. Selbst beim Ausschalten des Gerätes, auf dem sich die Anwendung befindet, bleiben die Daten erhalten und werden erst dann gelöscht, wenn die Anwendung vom Gerät deinstalliert wird.

Beim Umgang mit dem *IsolatedStorageSettings* gibt es jedoch verschiedene Dinge, welche beachtet werden müssen:

- Sollte versucht werden Daten aus dem Speicher zu lesen, zu denen noch kein Name eingespeichert wurde oder gelöscht wurde, erzeugt dies einen Error. Deshalb muss immer, wenn Daten aus dem *IsolatedStorageSettings* gelesen werden, deren Existenz überprüft werden.
- Mit dem *IsolatedStorageSettings* können sämtliche Daten gespeichert werden, das bedeutet einzelne Werte, sowie Objekte. Dazu gehören ebenfalls vom Benutzer erzeugte Objekte, welche ohne Umstände direkt gespeichert werden können.

- Alle im `IsolatedStorageSettings` befindlichen Daten sind immer nur Daten und müssen beim Auslesen als entsprechendes Objekt gecastet werden.

Der letzte Punkt, der beachtet werden muss, stellt dabei auch den größten Schwachpunkt des `IsolatedStorageSettings` dar. Alle Objekte müssen gecastet werden. Daher kann es schnell zu Fehlern kommen, wenn die ausgelesenen Daten nicht ordnungsgemäß überprüft werden.

Der folgende Abschnitt behandelt die Implementierung des `IsolatedStorageSettings` innerhalb der App. Dazu wird in dem Listing 6.6 der Code für die Benachrichtigungseinstellungen gezeigt, welche als Beispiel für die Implementierung dienen sollen.

Zunächst muss das `IsolatedStorageSettings`-Objekt erzeugt werden, welches die Verbindung zum Speicher ist (Zeile 1). Mit dessen Hilfe werden die Daten in dem Speicher bearbeitet und gelöscht. Danach werden innerhalb der *try-catch*-Blöcke in den Zeilen 7 bis 15 und 18 bis 26 die verschiedenen Einstellungen abgespeichert. Im *try*-Abschnitt der Blöcke wird zunächst versucht die Daten aus dem Speicher zu laden. Dabei werden unter Berücksichtigung des Typs die Daten aus dem Speicher ausgelesen, gecastet und den passenden Variablen zugewiesen. In Zeile 7 wird dabei der *notificationCounter* als *Integer* gecastet und in Zeile 20 der *Boolean*-Wert für die Benachrichtigungsart. Sollte das Abrufen der Variablen fehlschlagen, weil sie entweder gelöscht wurden oder noch nicht initialisiert, werden diese im *catch*-Abschnitt in den Speicher geschrieben. Hierzu werden die Variablen mit ihren Standardwerten initialisiert (Zeile 12 und 24). Anschließend werden die Werte der Variablen mit einem Namen für den Speicher verbunden. Der *notificationCounter* wird dem Speicher neu hinzugefügt (Zeile 14), während beim *notificationType* der Wert dem bereits vorhandenen Name im Speicher zugewiesen wird (Zeile 25). Zum Schluss werden in beiden Fällen die Daten in den Speicher geschrieben, indem die *Save*-Methode aufgerufen wird (Zeile 16 und 27)

Listing 6.6: Erzeugen der Benachrichtigungseinstellungen

```

1 private IsolatedStorageSettings notificationSettings =
    IsolatedStorageSettings.ApplicationSettings;
2
3 private void loadPageSettings()
4 {
5     try
6     {
7         notificationCounter = (int)notificationSettings["notificationCounter"];
8         notificationCounter_Text.Text = notificationCounter.ToString();
9     }
10    catch
11    {

```

```
12     notificationCounter = 5;
13     notificationCounter_Text.Text = notificationCounter.ToString();
14     notificationSettings.Add("notificationCounter", notificationCounter);
15 }
16 notificationSettings.Save();
17
18 try
19 {
20     notificationTypeStandard = (bool)notificationSettings["notificationType"];
21 }
22 catch
23 {
24     notificationTypeStandard = true;
25     notificationSettings["notificationType"] = notificationTypeStandard;
26 }
27 notificationSettings.Save();
28 }
```

7 Vorstellung der Anwendung

Nachdem im vorherigen Kapitel die Implementierung der Anwendung erläutert wurde, sollen in diesem Kapitel die Darstellung und Benutzung der Anwendung im Vordergrund stehen. Da die App zur Überwachung der Schwankungen der Tinnituswahrnehmung dient und dies ausschließlich über die App möglich ist, ist der Fragebogen deren Hauptfunktion. Die App unterstützt dabei alle Geräte mit dem Betriebssystem *Windows Phone* 8 oder höher.

7.1 Anmeldung mit Benutzername und Kennwort

Da die App sich nur benutzen lässt, wenn eine gültige Anmeldung mit Benutzername und Kennwort durchgeführt wurde, ist die Anmeldung (vgl. Abbildung 7.1) die erste Seite, die der Benutzer sieht. Auf dieser Seite kann sich der Benutzer entweder direkt anmelden oder über den Button „Registrierung“ die Ansicht zur Erstellung eines neuen Benutzerkontos auf dem Server erreichen. Hat der Benutzer Benutzername und Kennwort eingegeben, meldet er sich mit dem Button „Login“ am Server an. Im Falle einer fehlenden Internetverbindung oder bei Eingabe eines falschen Benutzernamens oder Kennworts wird eine entsprechende Fehlermeldung ausgegeben.

Nachdem sich der Benutzer erfolgreich angemeldet hat, werden automatisch die aktuellen statistischen Fragebögen heruntergeladen. Sollte der Benutzer diese Fragebögen schon auf der Webseite beantwortet haben, werden auch die entsprechenden Antworten heruntergeladen.

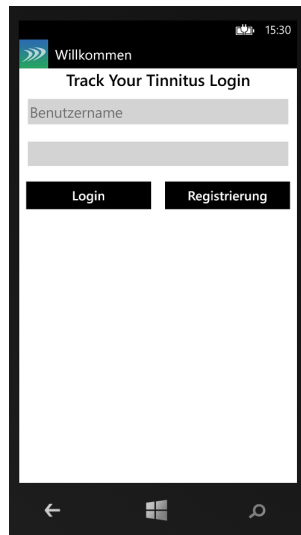


Abbildung 7.1: Login Ansicht der App

7.2 Registrierung

Die Registrierung (vgl. Abbildung 7.2) bei Track Your Tinnitus erfordert vom Benutzer einen Benutzernamen, sowie seine E-Mail Adresse und ein Kennwort. Um sicherzugehen, dass der Benutzer keine falschen Eingaben tätigt, müssen E-Mail Adresse und Kennwort jeweils zwei Mal eingegeben werden. Sollten Daten bei der Eingabe fehlen oder der Benutzername oder die E-Mail Adresse bereits registriert sein, wird eine Fehlermeldung ausgegeben. Eine Fehlermeldung wird auch ausgegeben, sollte die Registrierung aufgrund einer fehlenden Internetverbindung fehlschlagen. Im Falle einer erfolgreichen Registrierung wird dem Benutzer gemeldet, dass eine Bestätigung per E-Mail an den Benutzer gesendet wird. Die Überprüfung der Daten erfolgt sobald der Benutzer den Button „Registrieren“ drückt. Anschließend werden die Daten an den Server übertragen.

Aufgrund der Restriktionen der Universität Ulm werden E-Mail Adressen nicht automatisch in der Datenbank gespeichert. Daher wird die E-Mail Adresse direkt nach dem Versand der Bestätigung per E-Mail aus der Datenbank gelöscht. Es wird dem Benutzer jedoch auch die Möglichkeit geboten explizit der Speicherung seiner E-Mail Adresse zuzustimmen (1). Die Nutzung einer *checkbox* zum Zustimmung der Speicherung der E-Mail Adresse orientiert sich dabei an der Android Version. Im Gegensatz zur iOS und Android Version bietet die Anwendung auf *Windows Phone* keinen zusätzlichen „Zurück“-Button an, sondern nutzt die physikalische „Zurück“-Taste, welche jedes Windows Phone Gerät besitzt.

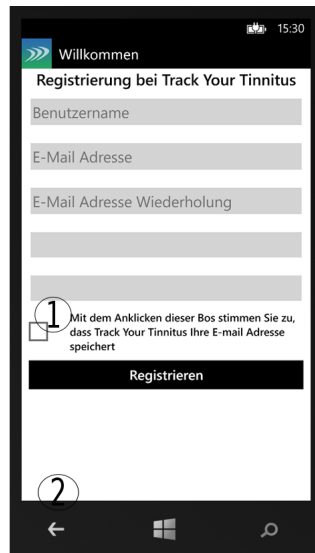


Abbildung 7.2: Registrierung in der App

7.3 Statistische Fragebögen

Hat sich der Benutzer angemeldet, wird er zum ersten statistischen Fragebogen geleitet. Die Fragebögen sind immer nach dem gleichen Farbschema aufgebaut: Die Fragen sind grau hinterlegt, während die Antwortmöglichkeiten weiß hinterlegt sind (vgl. Abbildung 7.3). Die Konfiguration der Fragebögen wird dabei vom Server übernommen. Nachdem der Benutzer den Fragebogen ausgefüllt hat, werden die Antworten mit dem „Speichern“-Button gespeichert und an den Server gesendet. Im Anschluss wird der nächste Fragebogen geladen. Der Benutzer muss alle statistischen Fragebögen beantwortet haben, bevor er mit der Überwachung der Schwankung der Tinnituswahrnehmung beginnen kann. Sollten nicht alle notwendigen Fragen beantwortet sein, wird eine Fehlermeldung ausgegeben.

Da der Benutzer beim Beantworten der statistischen Fragebögen jederzeit zwischen App und Webseite wechseln können soll, wurde ein zusätzlicher „Aktualisierungs“-Button in der oberen rechten Ecke bereitgestellt.

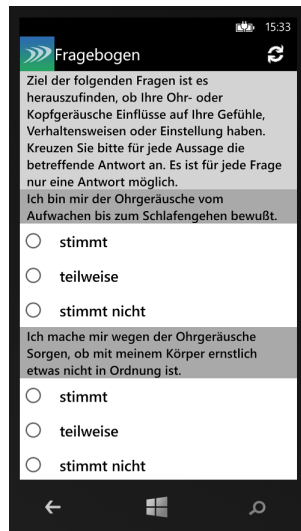


Abbildung 7.3: Statistischer Fragebogen Ansicht in der App

7.4 Hauptmenü

Mit dem Hauptmenü (vgl. Abbildung 7.4) hat der Benutzer die Möglichkeit zu den verschiedenen Bereichen, wie beispielsweise den Benachrichtigungseinstellungen, zu gelangen. Die Menüpunkte sind in der *Windows Phone* App identisch zu denen der iOS und Android Version. Jedoch unterscheidet sich die *Windows Phone* App in der Darstellung und der Art, wie das Hauptmenü geöffnet wird.

Die Darstellung wurde in der *Windows Phone* Version dahingehend verändert, dass die Icons zu den jeweiligen Menüpunkten weggelassen wurden. Die Darstellung der Menüpunkte mit Icons ist zwar auf dem Betriebssystem *Windows Phone* möglich, jedoch sind die Menüpunkte dann verhältnismäßig klein. Das Hauptmenü in der *Windows Phone* Version wird durch Berühren des unteren Abschnitts von unten nach oben geöffnet. Beim Auswählen eines Menüpunktes verschwindet das Menü und die ausgewählte Ansicht wird geöffnet. Zum Schließen der App wird die physikalische „Zurück“-Taste genutzt, jedoch muss der Benutzer dazu den Fragebogen zur Überwachung der Tinnituswahrnehmung geöffnet haben. Sollte sich der Benutzer in einer anderen Ansicht befinden, führt die „Zurück“-Taste zunächst immer zurück zur Ansicht des Fragebogens.

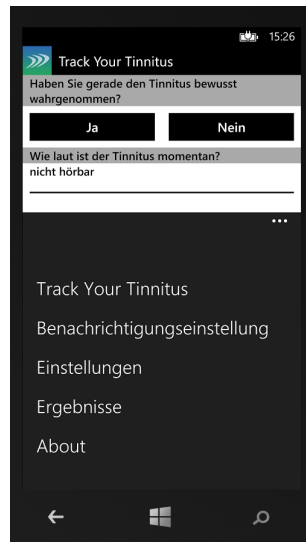


Abbildung 7.4: Hauptmenü Ansicht in der App

7.5 Fragebogen zur Überwachung der Tinnituswahrnehmung

Der Fragebogen zur Überwachung der Tinnituswahrnehmung (vgl. Abbildung 7.5) ist der Hauptbestandteil der App. Dieser wird dem Benutzer beim Start der App angezeigt, wenn er angemeldet ist, oder bei einer Interaktion mit einer Benachrichtigung. Mit Hilfe des Fragebogens kann der Benutzer zu jedem Zeitpunkt seine Wahrnehmung des Tinnitus speichern. In der Abbildung ist der obere Teil des Fragebogens dargestellt. Der Fragebogen ist aus sieben Fragen zusammengesetzt, welche verschiedene Antworttypen besitzen:

1. Haben Sie gerade den Tinnitus bewusst wahrgenommen? - Antworttyp: Ja/Nein
2. Wie laut ist der Tinnitus momentan? - Antworttyp: Slider
3. Wie belastend empfinden Sie den Tinnitus im Moment? - Antworttyp: Slider
4. Wie ist Ihre aktuelle Stimmungslage? - Antworttyp: Radio Button
5. Wie aufgeregt sind Sie gerade? - Antworttyp: Radio Button
6. Wie gestresst fühlen Sie sich gerade? - Antworttyp: Slider
7. Wie sehr haben Sie sich auf das konzentriert, was Sie gerade tun? - Antworttyp: Slider

Zusätzlich wird eine Frage angezeigt, bei welcher das „Schlimmste Symptom“ abgefragt wird, sollte diese Frage im statistischen Fragebogen ausgefüllt worden sein. Hat der Benutzer den Fragebogen ausgefüllt, werden die Antworten mit dem „Speichern“-Button

gesichert. Anschließend, wenn die Antworten gespeichert sind, beginnt ein Countdown von drei Sekunden. Sollte der Benutzer das Hauptmenü innerhalb dieser Zeit öffnen, wird der Countdown unterbrochen. Falls der Benutzer nichts weiter unternimmt, wird die App automatisch nach Ablauf der Zeit geschlossen. Vergisst der Benutzer das Speichern der Antworten, werden diese automatisch gespeichert.

Abbildung 7.5: Fragebogen zur Überwachung der Tinnituswahrnehmung der App

7.6 Benachrichtigungseinstellungen und Einstellungen

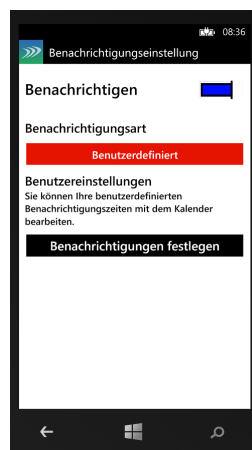
Die Track Your Tinnitus App bietet verschiedene Einstellungsmöglichkeiten (vgl. Abbildung 7.6). Bei den Einstellungsmöglichkeiten gibt es zwei unterschiedliche Ansichten, die Benachrichtigungseinstellungen (vgl. Abbildung 7.6a und b) und die allgemeinen Einstellungen (vgl. Abbildung 7.6c). Bei der App auf *Windows Phone* lässt sich im Gegensatz zur iOS und Android Version nur auswählen, ob während des Ausfüllens des Fragebogens die Umgebungsgeräusche mit aufgenommen werden dürfen oder nicht. Die Benachrichtigungseinstellungen unterscheiden sich nur in der Darstellung von der iOS und Android Version. Mit diesen Benachrichtigungseinstellungen kann der Benutzer die Zeiten für die Benachrichtigungen festlegen.

Zunächst kann der Benutzer allgemein festlegen, ob er benachrichtigt werden soll (1). Die Benachrichtigungsart wählt der Benutzer mit dem dazugehörigen Button (2), wobei er zwischen zwei Möglichkeiten wählen kann:

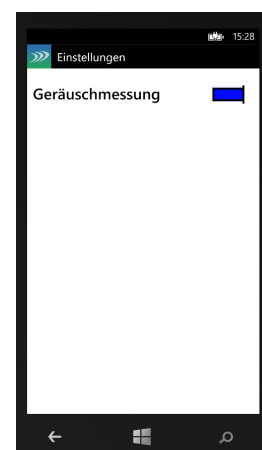
1. Standard: Bei dieser Benachrichtigungsart wird der Benutzer zufällig innerhalb eines festgelegten Zeitraums benachrichtigt. Hierzu kann der Benutzer die Anzahl an Benachrichtigungen pro Tag auswählen (3). Im Anschluss daran kann der Benutzer für jeden Tag der Woche ein Zeitintervall festlegen (4). Innerhalb des Zeitintervalls werden dann entsprechend viele Benachrichtigungen gesendet.
2. Benutzerdefiniert: Bei der benutzerdefinierten Einstellung erscheint der Button „Benachrichtigungen festlegen“, welcher den Benutzer zur Kalenderübersicht weiterleitet. Diese unterscheidet sich von der iOS und Android Version dahingehend, dass nur eine Monatsansicht vorhanden ist (vgl. Abbildung 7.7a). Wird in dieser ein Tag vom Benutzer berührt, öffnet sich ein *DatePicker* mit dem Datum des berührten Tages, sowie einer ungefähren Uhrzeit. Hier kann der Benutzer nachdem er eine Uhrzeit festgelegt hat, den Zeitpunkt für die Benachrichtigung speichern. Gleichzeitig werden dem Benutzer für den ausgewählten Tag alle bisher gespeicherten Benachrichtigungen angezeigt. Sollte der Benutzer eine Benachrichtigung löschen wollen, muss er die zu löschende Benachrichtigung doppelt berühren (vgl. Abbildung 7.7b).



(a) Benachrichtigungseinstellung: Standard

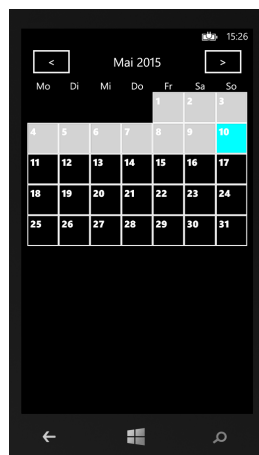


(b) Benachrichtigungseinstellung: Benutzerdefiniert

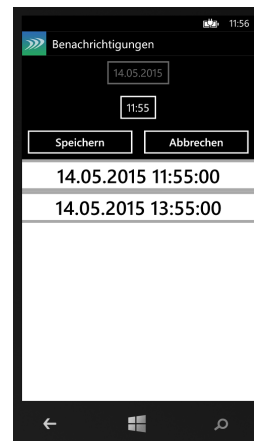


(c) Einstellungen

Abbildung 7.6: Ansicht der Einstellungen



(a) Monatsansicht



(b) DatePicker

Abbildung 7.7: Benachrichtigungsauswahl der App

8 Abgleich der Anforderungen

In diesem Kapitel werden alle Anforderungen, die in Kapitel 4 vorgestellt wurden, mit der Implementierung und den Funktionen der App abgeglichen. Der Abgleich der Anforderung ist wie in Kapitel bereits 4 in funktionale und nichtfunktionale Anforderungen unterteilt.

8.1 Funktionale Anforderungen

Der folgende Abschnitt zeigt den Abgleich für die funktionalen Anforderungen. Hierzu wird überprüft, ob alle funktionalen Anforderungen in der aktuellen Implementierung erfüllt sind bzw. deren aktueller Stand angegeben.

Nr.	Anforderung	Beschreibung
1	Ausfüllen der Fragebögen zur Überwachung der Schwankung der Tinnituswahrnehmung	Anforderung erfüllt (siehe Kapitel 7.5)
2	Registrieren und Anmelden in der App	Anforderung erfüllt (siehe Kapitel 7.1 und 7.2)
3	Nutzbarkeit ohne Internetverbindung	Anforderung erfüllt (siehe Kapitel 6.2.1)
4	Festlegen des Klingeltons einer Benachrichtigung	Anforderung teilweise erfüllt. Das Festlegen des Klingeltons ist nicht direkt in der App möglich, jedoch über das Gerät. (siehe Kapitel 6.1.2)
5	Einstellung der Benachrichtigungszeiten (zufällig)	Anforderung erfüllt (siehe Kapitel 7.6 und 6.1.3)
6	Einstellung der Benachrichtigungszeiten (planmäßig)	Anforderung erfüllt (siehe Kapitel 7.6 und 6.1.3)
7	Ausfüllen der statistischen Fragebögen	Anforderung erfüllt (siehe Kapitel 7.3)
8	Synchronisierung der Antworten von statistischen Fragebögen	Anforderung erfüllt (siehe Kapitel 7.3)

Tabelle 8.1: Abgleich der funktionalen Anforderungen

Nr.	Anforderung	Beschreibung
9	Messung der Lautstärke von Umgebungsgeräuschen	Anforderung erfüllt (siehe Kapitel 7.6)
10	Slider ohne initialen Wert	Anforderung erfüllt (siehe Kapitel 7.5)

Tabelle 8.2: Abgleich der funktionalen Anforderungen

8.2 Nichtfunktionale Anforderungen

In der folgenden Tabelle werden die nichtfunktionale Anforderungen mit dem aktuellen Zustand der App abgeglichen.

Nr.	Anforderung	Beschreibung
1	Darstellung der App und des Fragebogens	Anforderung teilweise erfüllt. Die App wurde möglichst identisch zu den bereits existierenden Versionen gestaltet, jedoch gibt es einige Unterschiede, wie zum Beispiel bei der Kalenderansicht.
2	Farbgebung und Bedienungselemente	Anforderung teilweise erfüllt. In der App wurden möglichst identisch Bedienungselemente genutzt, jedoch unterscheiden sich diese in der Darstellung von den anderen Versionen.
3	Ausfüllen des Fragebogens zur Überwachung der Schwankungen in kurzer Zeit	Anforderung erfüllt. Nach verschiedenen Tests konnte der Fragebogen unter einer Minute ausgefüllt werden.

Tabelle 8.3: Abgleich der nicht-funktionalen Anforderungen

9 Zusammenfassung und Ausblick

Am Ende dieser Arbeit erfolgt im letzten Kapitel eine Zusammenfassung der Erkenntnisse, die aus der Entwicklung der *Windows Phone App* (Kapitel 9.1) gewonnen werden konnten. Anschließend werden in Kapitel 9.2 weitere Ideen und Möglichkeiten beschrieben, um die Funktionen der App in Zukunft zu verbessern und zu erweitern.

9.1 Zusammenfassung

Im Rahmen dieser Bachelorarbeit entstand eine Version der Track Your Tinnitus App für das Betriebssystem *Windows Phone* 8 und höher. Um die Anforderungen für die *Windows Phone App* zu definieren, dienten die in [7, 21, 22, 18, 19] ausgearbeiteten Anforderungen als Vorlage. Zusätzlich fand während der Entwicklung ein kurzer Vergleich mit der Android Version der Track Your Tinnitus App statt. Aufgrund der Möglichkeiten des Betriebssystems *Windows Phone*, welches bei der Entwicklung benutzt wurde, konnten einige Anforderungen nur bedingt erfüllt werden. Daher war zum Beispiel ein Hauptmenü mit Icons in einer vertikalen Darstellung nicht möglich, da von *Windows Phone* nur eine horizontale Darstellung mit geringerer Schriftgröße unterstützt wird. Bei der Implementierung des Kalenders musste auf eine zusätzliche Library zurückgegriffen werden (siehe Kapitel 5.3.2). Mit dieser Library konnte ein Kalender für die benutzerdefinierten Benachrichtigungen implementiert werden, welcher jedoch nur eine Monatsansicht bietet, was ein deutlicher Unterschied zur Möglichkeit einer Wochen- und Tagesansicht der iOS und Android Version darstellt. Auch die Benachrichtigungen entsprechen nicht genau den Anforderungen aufgrund der Beschränkungen durch die Nutzung von Remindern (siehe Kapitel 6.1.2). Daher muss der Benutzer in der *Windows Phone App* den Klingelton für die Benachrichtigungen außerhalb der App einstellen.

9.2 Ausblick

Die aktuelle App, welchem im Zuge dieser Arbeit entstanden ist, wurde für das Betriebssystem *Windows Phone* 8 und höher entwickelt, da diese den größten Marktanteil an Smartphones mit dem Betriebssystem *Windows Phone* darstellen [12]. Da Microsoft mit der Entwicklung von *Windows Phone* 8.1 den App-Entwicklern neue Möglichkeiten bot, unter anderem durch die Unterstützung von Silverlight, kann das Interface verbessert werden, wie zum Beispiel das Hauptmenü. Des Weiteren ist durch die Nutzung von Silverlight eine Portierung für das Betriebssystem *Windows* möglich. Bisher wurde die Track Your Tinnitus App vorwiegend für Smartphones entwickelt, jedoch wird durch Microsoft die Entwicklung von Apps, die sowohl auf *Windows Phone* als auch auf *Windows* funktionieren, erleichtert. Zusätzlich gibt es immer neue Möglichkeiten durch die Tablet-PCs, wie das Microsoft Surface, seine Arbeit unterwegs zu erledigen. Eine Tablet Version der Track Your Tinnitus App wäre zwar nur bedingt geeignet, jedoch könnte eine App, welche es dem Benutzer ermöglicht, den Fragebogen auszufüllen, während er zum Beispiel an einem Rechner arbeitet, die Überwachung der Tinnituswahrnehmung vereinfachen. An dieser Stelle sei geschlossen:

That's the thing about people who
think they hate computers ... What
they really hate is lousy programmers.

Larry Niven [17]

Literaturverzeichnis

- [1] COOK, A. J. ; ROBERTS, D. A. ; HENDERSON, M. D. ; VAN WINKLE, L. C. ; CHASTAIN, D. C. ; HAMILL-RUTH, R. J.: Electronic pain questionnaires: a randomized, crossover comparison with paper questionnaires for chronic pain assessment. In: *Pain* 110 (2004), Nr. 1, S. 310–317
- [2] CROMBACH, A. ; NANDI, C. ; BAMBONYE, M. ; LIEBRECHT, M. ; PRYSS, R. ; REICHERT, M. ; ELBERT, T. ; WEIERSTALL, R. : Screening for mental disorders in post-conflict regions using computer apps - a feasibility study from Burundi. In: *XIII Congress of European Society of Traumatic Stress Studies (ESTSS) Conference*, 70–70
- [3] DEVELOPMENT, P. M.: *White Noise*. Windows Phone Store, . – <http://www.windowsphone.com/de-de/store/app/white-noise/6e0f14c7-39ef-df11-9264-00237de2db9e>; Version: Mai 2015
- [4] DEVELOPMENT, R. : *Tinnitus*. Windows Phone Store, . – <http://www.windowsphone.com/de-de/store/app/tinnitus/406edd31-20f4-4821-85b0-739f5f4d18eb>; Version: Mai 2015
- [5] FANNING, J. ; MCAULEY, E. : A Comparison of Tablet Computer and Paper-Based Questionnaires in Healthy Aging Research. In: *JMIR research protocols* 3 (2014), Nr. 3
- [6] HENRY, J. A. ; GALVEZ, G. ; TURBIN, M. B. ; THIELMAN, E. J. ; McMILLAN, G. P. ; ISTVAN, J. A.: Pilot study to evaluate ecological momentary assessment of tinnitus. In: *Ear and hearing* 32 (2012), Nr. 2, S. 179
- [7] HERRMANN, J. : *Konzeption und technische Realisierung eines mobilen Frameworks zur Unterstützung tinnitusgeschädigter Patienten*, University of Ulm, Diplomarbeit, 2014
- [8] ISELE, D. ; RUF-LEUSCHNER, M. ; PRYSS, R. ; SCHAUER, M. ; REICHERT, M. ; SCHOBEL, J. ; SCHINDLER, A. ; ELBERT, T. : Detecting adverse childhood experi-

- ences with a little help from tablet computers. In: *XIII Congress of European Society of Traumatic Stress Studies (ESTSS) Conference*, 69–70
- [9] KIM, J. H. ; KWON, S.-S. ; SHIM, S. R. ; SUN, H. Y. ; KO, Y. M. ; CHUN, D.-I. ; YANG, W. J. ; SONG, Y. S.: Validation and reliability of a smartphone application for the international prostate symptom score questionnaire: a randomized repeated measures crossover study. In: *Journal of medical Internet research* 16 (2014), Nr. 2
- [10] KOOLAPPZ: *Tinnitus Cure*. Windows Phone Store, . – <http://www.windowsphone.com/de-de/store/app/tinnitus-cure/7cc0ef24-b2fe-40d6-814b-a2d6faf986d5>; Version: Mai 2015
- [11] LANGGUTH, B. ; HAJAK, G. ; KLEINJUNG, T. ; CACACE, A. ; MOLLER, A. : *Tinnitus: Pathophysiology and Treatment Preface*. 2007
- [12] LIJANA JOUZAITYTÉ, M. L. ; MENDELEVICH, A. : *AdDuplex Windows Phone Statistics, November 2014*. Blogpost, . – <http://blog.adduplex.com/2014/11/adduplex-windows-phone-statistics.html>; Version: Mai 2015
- [13] MICROSOFT: *How to create alarms and reminders for Windows Phone 8*. Windows Dev Center, . – [https://msdn.microsoft.com/en-us/library/windows/apps/hh202965\(v=vs.105\).aspx](https://msdn.microsoft.com/en-us/library/windows/apps/hh202965(v=vs.105).aspx); Version: Mai 2015
- [14] MICROSOFT: *How to: Store and Retrieve Application Settings Using Isolated Storage*. MSDN Dev Center, . – [https://msdn.microsoft.com/en-us/library/cc221360\(v=vs.95\).aspx](https://msdn.microsoft.com/en-us/library/cc221360(v=vs.95).aspx); Version: Mai 2015
- [15] MICROSOFT: *Sending push notifications for Windows Phone 8*. Windows Dev Center, . – [https://msdn.microsoft.com/en-us/library/windows/apps/hh202945\(v=vs.105\).aspx](https://msdn.microsoft.com/en-us/library/windows/apps/hh202945(v=vs.105).aspx); Version: Mai 2015
- [16] MISHRA, A. : *Introducing SQL Server Compact 4.0, the Next Gen Embedded Database from Microsoft*. MSDN Blog, . – <http://blogs.msdn.com/b/sqlservercompact/archive/2010/07/07/introducing-sql-server-compact-4-0-the-next-gen-embedded-database-from-microsoft.aspx>; Version: Mai 2015
- [17] NIVEN, L. ; POURNELLE, J. : *Oath of fealty*. 1981
- [18] PRYSS, R. ; REICHERT, M. ; HERRMANN, J. ; LANGGUTH, B. ; SCHLEE, W. : Mobile Crowd Sensing in Clinical and Psychological Trials - A Case Study. In: *28th IEEE Int'l Symposium on Computer-Based Medical Systems*, 2015

- [19] PRYSS, R. ; REICHERT, M. ; LANGGUTH, B. ; SCHLEE, W. : Mobile Crowd Sensing Services for Tinnitus Assessment, Therapy and Research. In: *IEEE 4th International Conference on Mobile Services (MS 2015)*, 2015
- [20] RUF-LEUSCHNER, M. ; PRYSS, R. ; LIEBRECHT, M. ; SCHOBEL, J. ; SPYRIDOU, A. ; REICHERT, M. ; SCHAUER, M. : 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*, 70–70
- [21] SCHLEE, W. ; HERRMANN, J. ; PRYSS, R. ; REICHERT, M. ; LANGGUTH, B. : How dynamic is the continuous tinnitus percept? In: *11th International Tinnitus Seminar*
- [22] SCHLEE, W. ; HERRMANN, J. ; PRYSS, R. ; REICHERT, M. ; LANGGUTH, B. : Moment-to-moment variability of the auditory phantom perception in chronic tinnitus. In: *13th Int'l Conf on Cochlear Implants and Other Implantable Auditory Technologies*
- [23] SCHOBEL, J. ; PRYSS, R. ; REICHERT, M. : Using Smart Mobile Devices for Collecting Structured Data in Clinical Trials: Results From a Large-Scale Case Study. In: *28th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2015)*, IEEE Computer Society Press
- [24] SCHOBEL, J. ; RUF-LEUSCHNER, M. ; PRYSS, R. ; REICHERT, M. ; SCHICKLER, M. ; SCHAUER, M. ; WEIERSTALL, R. ; ISELE, D. ; NANDI, C. ; ELBERT, T. : A generic questionnaire framework supporting psychological studies with smartphone technologies. In: *XIII Congress of European Society of Traumatic Stress Studies (ESTSS) Conference*, 69–69
- [25] SCHOBEL, J. ; SCHICKLER, M. ; PRYSS, R. ; MAIER, F. ; REICHERT, M. : Towards Process-Driven Mobile Data Collection Applications: Requirements, Challenges, Lessons Learned. In: *10th Int'l Conference on Web Information Systems and Technologies (WEBIST 2014), Special Session on Business Apps*, 371–382
- [26] SCHOBEL, J. ; SCHICKLER, M. ; PRYSS, R. ; NIENHAUS, H. ; REICHERT, M. : Using Vital Sensors in Mobile Healthcare Business Applications: Challenges, Examples, Lessons Learned. In: *9th Int'l Conference on Web Information Systems and Technologies (WEBIST 2013), Special Session on Business Apps*, 509–518
- [27] SCHOBEL, J. ; SCHICKLER, M. ; PRYSS, R. ; REICHERT, M. : Process-Driven Data Collection with Smart Mobile Devices. In: *Web Information Systems and Techno-*

logies - 10th International Conference, WEBIST 2014, Barcelona, Spain, Revised Selected Papers. Springer (LNBIP)

- [28] SHOOTERSOFTWARE: *NoizeMaker*. Windows Phone Store, . – <http://www.windowsphone.com/de-de/store/app/noizemaker/65354759-a949-e011-854c-00237de2db9e>; Version: Mai 2015
- [29] SMITH, J. : *WPF Apps With The Model-View-ViewModel Design Pattern*. MSDN Magazin, . – <https://msdn.microsoft.com/en-us/magazine/dd419663.aspx>; Version: Mai 2015

Name: Michael Klaus Meyer

Matrikelnummer: 747940

Erklärung

Ich erkläre, dass ich die Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den

Michael Klaus Meyer