ulm university universität
uulm

Universität Ulm | 89069 Ulm | Germany

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

Konzeption und Realisierung einer mobilen Anwendung zur Bestimmung der Tinnitusfrequenz am Beispiel von Android

Bachelorarbeit an der Universität Ulm

Vorgelegt von:
Annika Sophie Stampf
annika.stampf@uni-ulm.de

Gutachter:
Prof. Dr. Manfred Reichert

Betreuer:
Dr. Rüdiger Pryss

2018

Fassung 4. Juli 2018

Satz: PDF-LaTeX 2$_\varepsilon$

Kurzfassung

Tinnitus ist eine weitverbreitete Hörstörung. Gegenwärtig beschäftigen sich eine Vielzahl an Forschungsgruppen damit wirksame Therapien zu entwickeln. Eine der daraus entstandenen Studien evaluiert die Wirkung einer Klangtherapie, die stark auf der Musiktechnologie basiert. Um diese oder vergleichbare Therapieform anwenden zu können, muss zuvor die Tinnitus-Frequenz des Patienten möglichst präzise bestimmt werden können.

In der vorliegenden Arbeit wird die Konzeption und Realisierung einer mobilen Android Applikation vorgestellt, die Patienten die Möglichkeit bietet eine solche Frequenzbestimmung selbständig durchzuführen. Zusätzlich wird ein auf mehreren Leveln basierendes Hörtraining bereitgestellt, welches betroffene Personen in der Erkennung und Bewertung differenzierter Tonhöhenunterschiede schult, um so eine möglichst präzise Bestimmung zu ermöglichen.

Dafür werden zu Beginn einige Hintergrundinformationen geliefert und Anforderungen an die Applikation identifiziert, auf deren Basis eine geeignete Architektur bestimmt wird. Anschließend werden die Funktionalitäten der prototypischen Realisierung gezeigt und ein Ausblick auf eventuelle Erweiterungen gegeben.

Der in diesem Rahmen entstandene Prototyp soll dabei helfen neue Erkenntnisse darüber zu gewinnen, inwieweit eine eigenständige Messung der Patienten sinnvoll ist und wie eingebaute Hörtraining-Einheiten diese verbessern können.

Inhaltsverzeichnis

1

Einleitung

Tinnitus ist eine Hörstörung von der vermutlich 10-15% der erwachsenen Bevölkerung betroffen ist. Gegenwärtig wird viel geforscht, um wirksame Therapien zu entwickeln. Diese Therapien können im Grunde in zwei Klassen geteilt werden: Behandlungen, die die Wahrnehmung des Tinnitus betreffen, also versuchen die Lautstärke zu senken und welche, die mit Hilfe psychologischer Maßnahmen den Umgang mit der Hörstörung erleichtern sollen [1].

Eine Therapie aus erstgenannter Kategorie ist zum Beispiel das Tailor-Made Notched Music Training. Hierbei wird der Patient beliebiger Musik ausgesetzt, bei der die Energie aus dem Frequenzbereich gefiltert wird, welcher die individuelle Tinnitusfrequenz umgibt [2].

Eine Studie des europäischen Förderprojekts TINNET[1] evaluiert die Wirkung einer solchen Klangtherapie [5]. Diese zeigte, dass die Lautstärke des Tinnitus der Patienten nach einer Kurzzeitbehandlung temporär um durchschnittlich 7% verringert werden konnte.

Um diese oder ähnliche Therapien anwenden zu können, muss zuvor jedoch die Tinnitusfrequenz des Patienten möglichst präzise bestimmt werden können.

[1] Das europäische Förderprojekt TINNET ist eine COST Aktion [3] und arbeitet an dem Verständnis der Heterogenität von Tinnitus zur Verbesserung und Entwicklung neuer Behandlungen [4].

1.1 Problemstellung

Eine solche Bestimmung der Tinnitusfrequenz wird üblicherweise in einer Klinik durchgeführt. Sie ist zeitlich aufwendig, kostenintensiv und führt nicht selten zu falschen Werten, da die meisten Patienten keinen musikalischen Hintergrund aufweisen und in der Tonhöhenbestimmung nicht geübt sind. Außerdem kann sich die Frequenz des Tinnitus im Laufe der Zeit ändern [5]. Das bedeutet, dass eine Bestimmung in regelmäßigen Abständen wiederholt werden muss, um einen bestmöglichen Therapieerfolg gewährleisten zu können.

Mit Hilfe einer mobilen Anwendung kann diese Bestimmung vom Patienten selbst übernommen werden. Mit einem zusätzlichen Hörtraining könnte der Patient so sein Hören schulen und Frequenzbestimmungen beliebig oft wiederholen. Damit erlangt der Patient in der Tonhöhenbestimmung Routine, weshalb eine Bestimmung in der Klinik entweder schneller und präziser durchgeführt werden kann oder im besten Fall unnötig wird. Außerdem könnte der Patient nach regelmäßiger Verwendung der mobilen Applikation in der Lage sein selbst einzuschätzen, wann eine erneute Frequnzbestimmung aufgrund einer Änderung seines Tinnitus sinnvoll ist.

1.2 Zielsetzung

Ziel ist die Konzeption und prototypische Realisierung einer mobilen Android Anwendung zur Bestimmung der Tinnitusfrequenz. Um eine präzise Bestimmung erreichen zu können, soll außerdem ein Hörtraining implementiert werden, das dem Betroffenen spielerisch hilft verschiedene Tonhöhen abschätzen und vergleichen zu können. Des Weiteren soll die Lautstärke für verschiedene Frequenzintervalle, der im Hörtraining verwendeten Töne, angepasst werden können, um die Anwendung durch einen eventuellen Hörsturz nicht zu beeinträchtigen.

1.3 Struktur der Arbeit

Diese Arbeit gliedert sich in acht thematische Bereiche, welche jeweils durch ein Kapitel repräsentiert werden. Im nachfolgenden Kapitel werden Hintergrundinformationen zum Thema geliefert. Dabei handelt es sich um eine Hinführung zur verwendeten Methode der Frequenzbestimmung und eine Erläuterung des Musical Instrument Digital Interface. In Kapitel 4 werden die Anforderungen an die Applikation beschrieben. Diese sind in funktionale und nichtfunktionale Anforderungen geteilt, wobei die Erstgenannten zusätzlich nach Hauptfunktionalitäten der Anwendung gegliedert sind. Nachfolgend wird anhand der erarbeiteten Anforderungen eine geeignete Architektur bestimmt und erläutert. Dazu werden die Prozesse der Anwendung modelliert, das Systemmodell erläutert und das daraus resultierende Datenbankschema gezeigt. Zuletzt werden außerdem verwendete Bibliotheken vorgestellt. In Kapitel 5 werden anschließend einige ausgewählte Code-Abschnitte erklärt. Die Funktionalitäten des Prototyps werden in Kapitel 6 aus Sicht des Benutzers und anhand von Bildschirmaufnahmen gezeigt. Schließlich werden in Kapitel 7 die prototypische Realisierung mit den in Kapitel 3 erarbeiteten Anforderungen abgeglichen. Als Abschluss wird eine Zusammenfassung der Arbeit und ein Ausblick auf eventuelle Erweiterungen und Anpassungen gegeben.

2

Hintergrund

In diesem Kapitel werden einige Hintergrundinformationen geliefert, die die vorliegende Arbeit in einen größeren Kontext bringen und verwendete Methoden und Werkzeuge erläutern.

2.1 TrackYourTinnitus

Eine verwandte Arbeit ist die mobile Anwendung TrackYourTinnitus [6]. Diese wurde von der Tinnitus Research Initiative (TRI) in Kooperation mit dem Institut für Datenbanken und Informationssysteme der Universität Ulm (DBIS) entwickelt. Sie bietet Patienten speziell entworfene Fragebögen zur Erfassung derer aktuellen Tinnitus Wahrnehmung. Sie dient damit als eine Art Assistent, der die Betroffenen im Umgang mit dem Tinnitus unterstützt [7]. Anhand von Analysen, der dadurch erlangten Daten, konnten im Rahmen dieses Projektes außerdem einige neue Erkenntnisse über den Verlauf und die Abhängikeiten innerhalb der Tinntus-Symptomatik gewonnen werden [8] [9] [10]. Dieses Wissen wird wiederum für die Entwicklung neuer Therapien genutzt.

Die Applikation beinhaltet momentan noch keine Frequenzbestimmung, obgleich die Kenntnis der präzisen Tinnitusfrequenz Voraussetzung für die Anwendung einiger erfolgversprechender Therapieformen ist.

Der im Rahmen dieser Arbeit zu realisierende Prototyp soll die Funktionalität der TrackYourTinnitus Applikation in einer eigenständigen Anwendung um diese genannte Tinnitus-Frequenzbestimmung erweitern.

2.2 Methoden der Frequenzbestimmung

Es gibt verschiedene Möglichkeiten die Tinnitusfrequenz zu bestimmen. Dazu zählen zum Beispiel die *Method of Adjustment*, die *Forced Choice Double Staircase* Methode und das *Likeliness Rating* [11].

Einzelne Forschungsarbeiten wie die von Richard S. Tyler und David Conrad-Armes aus dem Jahr 1983 [12] und eine noch unveröffentlichte Studie des europäischen Förderprojekts TINNET vergleichen einige dieser Methoden, um eine bestmögliche Frequenzbestimmung zu ermöglichen. Faktoren, die diese Vergleiche stark erschweren und Resultate nicht zwingend verallgemeinerbar machen sind zum Beispiel die Umgebung (wird die Messung in einer Klinik oder Zuhause durchgeführt), der Grad der Hilfestellung (erfolgt die Messung mit ärztlicher Unterstützung oder selbstständig) oder der persönliche Hintergrund des Patienten (wird über musikalisches Vorwissen verfügt oder nicht).

Da bei der Implementierung der Match Your Tinnitus Applikation eine Methode benötigt wird, die selbstständig, schnell und ohne großes musikalisches Vorwissen verwendet werden können soll, ohne die Präzision der Bestimmung zu beeinträchtigen, wird die bereits genannte Method of Adjustment verwendet. Bei dieser Methode kann der Patient die Tonhöhe des abgespielten Tones direkt über einen Drehknopf oder Ähnliches an die wahrgenommene Frequenz des Tinnitus anpassen. Dabei gibt es die Möglichkeit zwischen einer groben und feinen Regelung hin und her wechseln zu können. Diese Methode ist in ihrer Verwendung sehr intuitiv und leicht verständlich, weil sie nach dem Prinzip *What You See Is What You Get* arbeitet. Des Weiteren ist die Methode sehr schnell, da sie nur aus einem Messdurchgang besteht. Nicht zu vernachlässigen ist außerdem, dass hinter dieser Methode kein aufwendiger Algorithmus steht, was die Implementierung stark vereinfacht.

2.3 Musical Instrument Digital Interface

Das Musical Instrument Digital Interface MIDI ("Digitale Schnittstelle für Musikinstrumente") wurde 1983 als Standard eingeführt und ist für die Kommunikation elektronischer Klangerzeuger zuständig. Für den Informationsaustausch der Tonhöhe zwischen der Anwendung und Pure Data (siehe Abschnitt 5.1.1), wird die sogenannte MIDI Tastennummer n verwendet [13]. Diese liegt zwischen 0 und 127 und wird durch folgende Formel mit Hilfe der Frequenz f berechnet:

$$n = \left(12 * \log_2\left(\frac{f}{440}\right)\right) + 69$$

Der Abstand zwischen zwei nebeneinanderliegenden Tastennummern beträgt einen Halbton. 12 Tastennummern ergeben also eine Oktave. Diese kann auch als Intervall zwischen zwei Tönen definiert werden, deren Frequenzen sich wie 2:1 verhalten [14]. Würde man für die Implementierung der Frequenzregler eine lineare Frequenzverteilung verwenden, wäre die wahrgenommene Tonhöhenverteilung logarithmisch. Um dies zu vermeiden werden die vorgestellten MIDI Tastennummern verwendet. Da die meisten Benutzer mit dieser Notation nicht vertraut sind, werden die Tastennummern für die Darstellung in Frequenzen, angegeben in Hertz, umgerechnet.

3

Anforderungsanalyse

Dieses Kapitel beschreibt die Anforderungen, die an die Android Anwendung Match Your Tinnitus gestellt wurden. Dabei sind die Anforderungen in funktionale und nichtfunktionale Anforderungen unterteilt.

3.1 Funktionale Anforderungen

In diesem Abschnitt werden die Funktionalen Anforderungen tabellarisch beschrieben. Der Leistungsumfang der Anwendung richtet sich dabei nach diesen Anforderungen. Für eine bessere Gliederung sind die geforderten Funktionen in mehrere Kategorien unterteilt.

3.1.1 Allgemeine Funktionale Anforderungen

Alle funktionalen Anforderungen, die die allgemeine Benutzung betreffen und unabhängig von der Frequenzmessung und des Hörtrainings sind, werden nachfolgend erläutert.

Nr.	Titel und Beschreibung
1	**Nutzer müssen ihren Namen angeben, um die Anwendung das erste Mal starten zu können** Um dem Nutzer individuellen Inhalt liefern zu können, muss dieser beim erstmaligen Öffnen der Anwendung angeben wie er heißt.

2	**Die Sprache wird automatisch und abhängig von den Geräteeinstellungen angepasst werden** Abhängig von den Geräteeinstellungen soll der Inhalt der Anwendung entweder auf deutsch oder auf englisch angezeigt werden.
3	**Töne werden nur dann ausgegeben, wenn Kopfhörer mit dem Gerät verbunden sind** Töne sollen ausschließlich dann abgespielt werden, wenn Kopfhörer mit dem Gerät verbunden sind. Werden diese entfernt während ein Ton gespielt wird, soll dieser Ton gestoppt und der Nutzer darüber informiert werden.

Tabelle 3.1: Allgemeine funktionale Anforderungen

3.1.2 Funktionale Anforderungen an die Frequenzmessung

Im Folgenden werden die Anforderungen beschrieben, die die Frequenzmessung und damit die Messung an sich und die Übersicht aller Messungen betreffen.

Nr.	Titel und Beschreibung
4	**Der Ton wird bei der Messung abhängig von der Position an dem der Nutzer seinen Tinnitus wahrnimmt ausgegeben** Der Nutzer muss wählen, ob er seinen Tinnitus 'links', 'rechts', 'links und rechts' oder 'im Kopf' hört. Abhängig davon soll der Ton bei der Messung bei der Auswahl von 'links' oder 'rechts' jeweils contralateral und bei den beiden anderen Wahlmöglichkeiten rechts abgespielt werden.
5	**Die Startfrequenz bei der Messung wird abhängig davon gewählt, in welchem Frequenzbereich der Nutzer seinen Tinnitus subjektiv wahrnimmt** Der Nutzer hat die Wahlmöglichkeiten 'tief', 'mittel', 'hoch' und 'sehr hoch'. Abhängig davon soll der Startwert bei der Messung auf 1000Hz, 2000Hz, 4000Hz oder 80000Hz gesetzt werden.

6	**Der Benutzer kann zwischen einer groben und einer feinen Frequenzbestimmung wählen** Dem Benutzer soll eine grobe und eine feine Frequenzbestimmung bereitgestellt werden, zwischen denen der Nutzer während der Messung frei wechseln kann.
7	**Die Messung deckt die Frequenzen von 50Hz bis 20.000Hz ab** Tinnitusfrequenzen zwischen 50Hz und 20.000Hz sollen auf einen Halbton genau gemessen werden können.
8	**Die Lautstärke und die Balance des gemessenen Tones kann an den Tinnitus angepasst werden** Im Anschluss an die Messung soll der Nutzer die Balance und die Lautstärke des gemessenen Tones an die seines Tinnitus anpassen können.
9	**Der Nutzer kann die Messung bewerten** Die Messung soll durch ein Rating vom Nutzer mit 0-5 Sternen bewertet werden können.
10	**Dem Nutzer wird eine Übersicht über alle Messungen geboten** Dem Nutzer soll auf einer von der Messung unabhängigen Seite eine Übersicht über alle Messungen angezeigt werden. Eine Messung soll dabei das Datum und die Wertung der Messung, die gemessene Frequenz, Lautstärke und Balance beinhalten.
11	**Der Benutzer hat die Möglichkeit seine Messungen zu teilen** Die Messungen sollen vom Benutzer als PDF-Datei geteilt werden können.

Tabelle 3.2: Funktionale Anforderungen an die Frequenzmessung

3.1.3 Funktionale Anforderungen an das Hörtraining

Alle Anforderungen, die im Rahmen des Hörtrainings gestellt werden, darunter das Hörtraining an sich, die Einstellungen der Tonausgaben und die Übersicht der Trainingsresultate, sind im Nachfolgenden beschrieben.

Nr.	Titel und Beschreibung
12	**Der Benutzer kann zwischen verschiedenen Levels wählen** Dem Nutzer werden verschiedene Level mit steigendem Schwierigkeitsniveau angeboten, zwischen denen er wählen kann. Jedes Level kann dabei beliebig oft wiederholt werden.
13	**Neue Levels können freigeschaltet werden** Zu Beginn des Training soll nur ein Level spielbar sein. Schließt der Benutzer ein Level erfolgreich ab, so soll ein neues Level freigeschaltet werden.
14	**Das Training beinhaltet mehrere Aufgabentypen** Das Training soll mehrere unterschiedlichen Aufgabentypen beinhalten, die das erfolgreiche Abschätzen und Vergleichen von Frequenzen auf jeweils unterschiedliche Art ausbilden.
15	**Die Reihenfolge der Aufgabentypen ist zufällig** Um dem Nutzer ein abwechslungsreiches Training zu bieten, sollen die Aufgabentypen in jedem Training neu und zufällig angeordnet werden.
16	**Die Frequenzen der Töne im Hörtraining sind zufällig** Um das Hören des Benutzers möglichst umfangreich trainieren zu können und das Training trotz beschränkter Anzahl an Aufgabentypen abwechslungsreich gestalten zu können, soll in jeder Aufgabe die Frequenz eines beliebigen Tones zufällig und die Frequenzen der anderen Töne in Abhängigkeit des Schwierigkeitsgrades gewählt werden.
17	**Dem Nutzer wird eine Übersicht über die Resultate seines Hörtrainings geboten** Dem Nutzer soll auf einer vom Hörtraining unabhängigen Seite eine Übersicht über die Resultate seines Hörtrainings angezeigt werden. Diese Übersicht beinhaltet jeweils das Level, Datum und die erreichte Punktzahl des Trainings.
18	**Der Benutzer hat die Möglichkeit die Resultate seines Trainings zu teilen** Die Resultate seines Trainings sollen vom Benutzer als PDF-Datei geteilt werden können.

19	**Die Lautstärke verschiedener Frequenzintervalle und die Balance der Töne im Training sind anpassbar** Der Nutzer soll die Möglichkeit haben die Lautstärke einzelner Frequenzintervalle und die Balance, der im Training abgespielten Töne, einstellen zu können.

Tabelle 3.3: Funktionale Anforderungen an das Hörtraining

3.2 Nichtfunktionale Anforderungen

Nr.	Titel und Beschreibung
1	**Technische Anforderungen** Die Anwendung soll ab Android 5.0 (API Level 21) lauffähig sein.
2	**Bedienbarkeit** Die Anwendung soll für den Benutzer intuitiv und verständlich sein.
3	**Erweiterbarkeit** Die Anwendung soll in der Anzahl der Level und der Aufgabentypen und in dem Einbau neuer Funktionalitäten einfach erweiterbar sein.
4	**Aussehen** Die Anwendung soll auf Grund ihres klinischen Einsatzes ein schlichtes Design aufweisen. Designelemente sollen vor allem zur besseren Bedienbarkeit eingesetzt werden.

Tabelle 3.4: Nichtfunktionale Anforderungen

4

Architektur

In diesem Kapitel wird die Architektur der Match Your Tinnitus Anwendung vorgestellt. Es enthält Prozessmodelle, das Klassendiagramm und das unterliegende Datenmodell.

4.1 Ablaufbeschreibung

Anhand der Anforderungsanalyse in Kapitel 3 wurden auf Basis der Spezifikationssprache BPMN (Business Process Model and Notation) Prozessdiagramme erstellt, um den Ablauf der Anwendung graphisch darstellen zu können. Diese erstellten Diagramme werden im Folgenden näher erläutert.

In Abbildung 4.1 ist die Navigation innerhalb der Anwendung dargestellt. Öffnet der Nutzer die Anwendung zum ersten Mal, soll eine Texteingabe erscheinen, die ihn auffordert seinen Namen anzugeben. Im Anschluss daran wird dem Nutzer die Seite zur Auswahl des Trainings und der Messung bereitgestellt. Über das Menü gelangt der Benutzer zu den Übersichten und den Einstellungen.

Über die Seite *Training und Messung* lässt sich optional eine Messung oder das Training starten. Dieser Vorgang lässt sich in Abbildung 6.6 erkennen.

In Abbildung 4.3 ist der Trainingsprozess dargestellt. Startet der Benutzer das Training wird erneut die Levelauswahl dargestellt. Ein Level ist nur dann auswählbar, wenn Kopfhörer mit dem Gerät verbunden sind. Während des Trainings wird die Lautstärke eines gespielten Tones in Abhängigkeit der Frequenz aus der Datenbank geladen. Nach dem Beenden eines Levels wird eine Levelübersicht angezeigt und das Trainingsresultat in der Datenbank gespeichert. Von hier aus gelangt der Nutzer zurück zur Levelauswahl

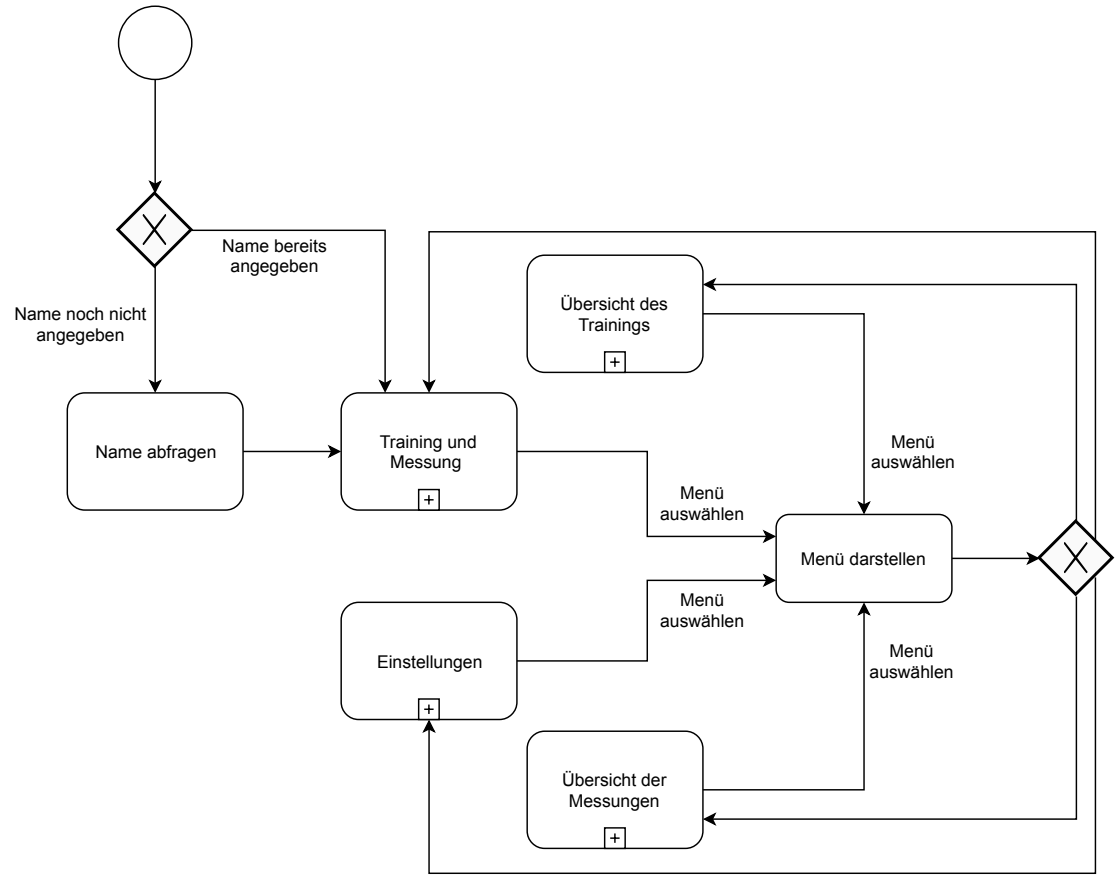

Abbildung 4.1: Navigation

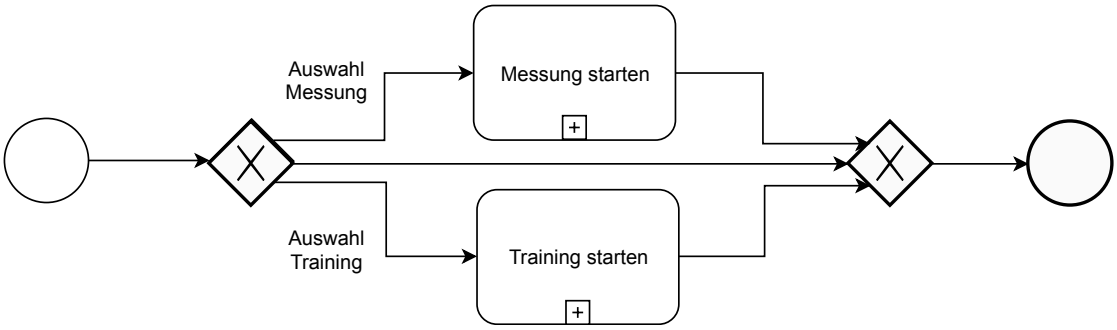

Abbildung 4.2: Training und Messung

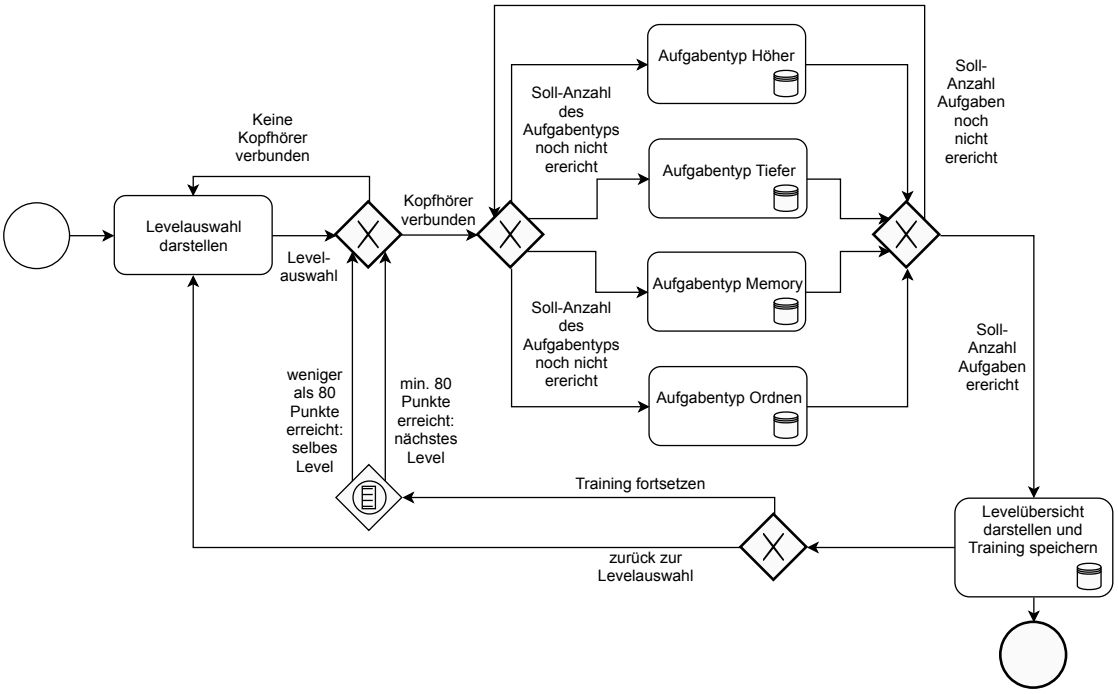

Abbildung 4.3: Training

oder abhängig von der erreichten Punktzahl erneut in das eben gespielte oder in das nächste Level.

In Abbildung 4.4 ist der Messprozess dargestellt. Zu Beginn der Messung wird die Position und die Höhe des Tinnitus abgefragt. Wurden diese Werte vollständig angegeben, wird eine Oberfläche für die Frequenzmessung bereitgestellt. Parallel dazu können Lautstärke und Frequenz angepasst und der Ton abgespielt werden.

Wie in Abbildung 4.5 ersichtlich, kann ein Ton nur dann abgespielt werden, wenn Kopfhörer mit dem Gerät verbunden sind. Ist dies nicht der Fall, wird ein Warnhinweis angezeigt. Der Ton kann außerdem beliebig oft abgespielt und pausiert werden. Drückt der Nutzer auf *weiter*, wird eine Oberfläche für die Lautstärke- und Balance-Anpassung bereitgestellt. Diese Werte können parallel angepasst werden und auch hier kann der Ton abgespielt werden. Wurden alle Anpassungen getroffen, kann der Nutzer wiederum auf *weiter* klicken und es wird eine Übersicht über die Messung angezeigt. Diese kann hier zusätzlich bewertet werden, bevor sie beendet und in der Datenbank gespeichert werden kann.

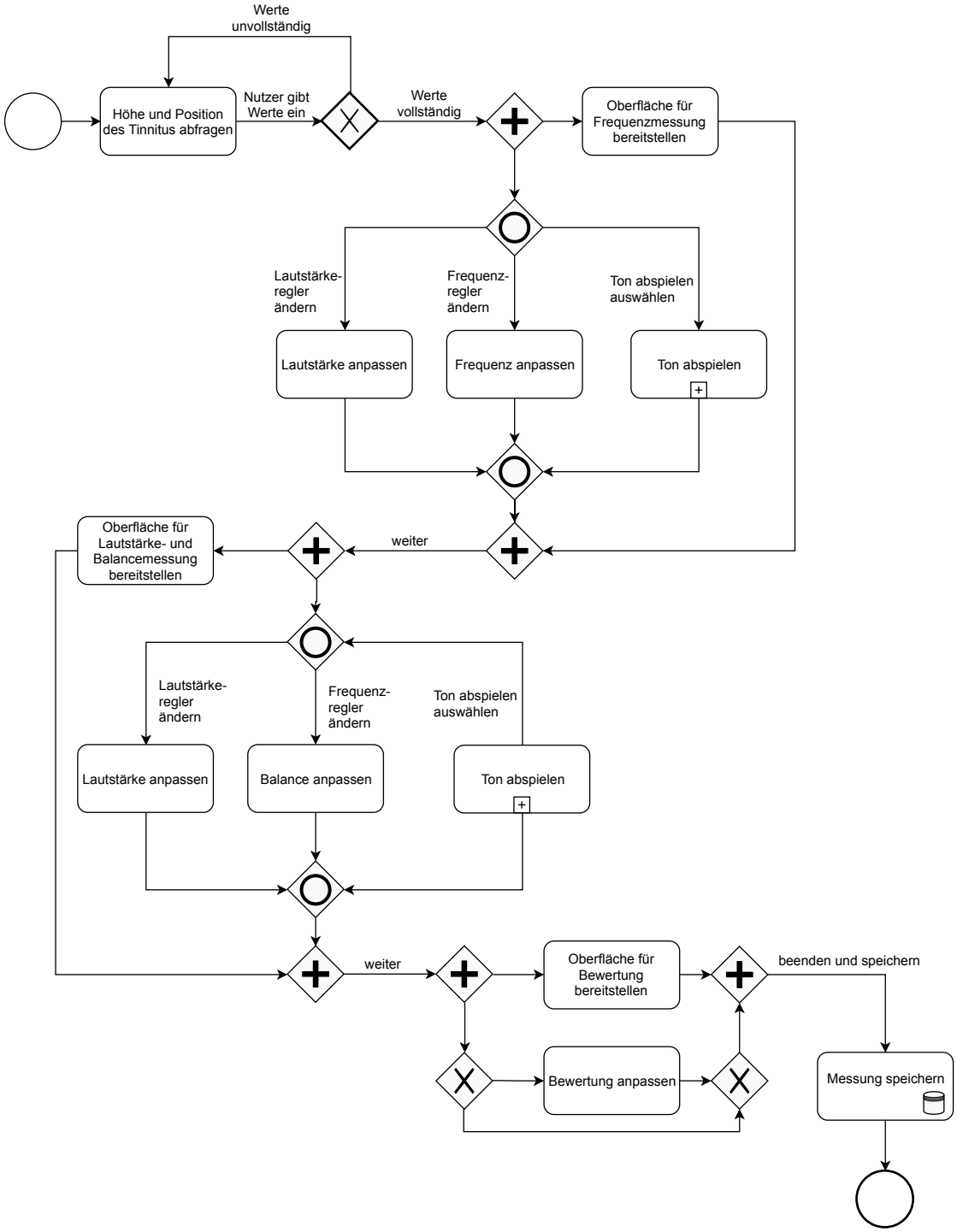

Abbildung 4.4: Messung

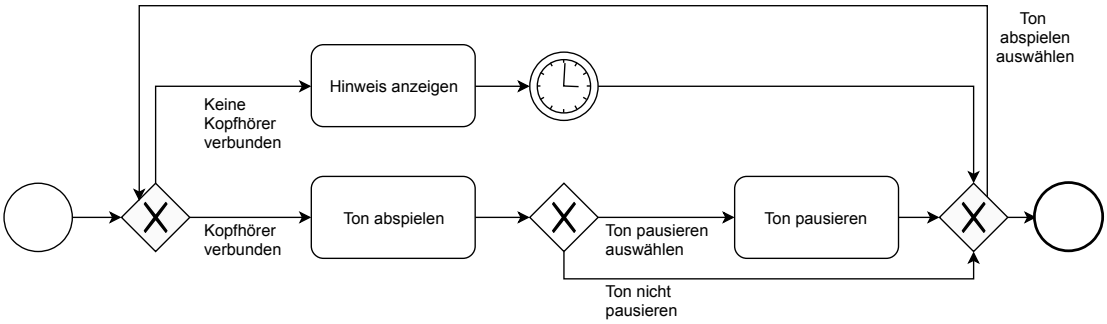

Abbildung 4.5: Ton abspielen

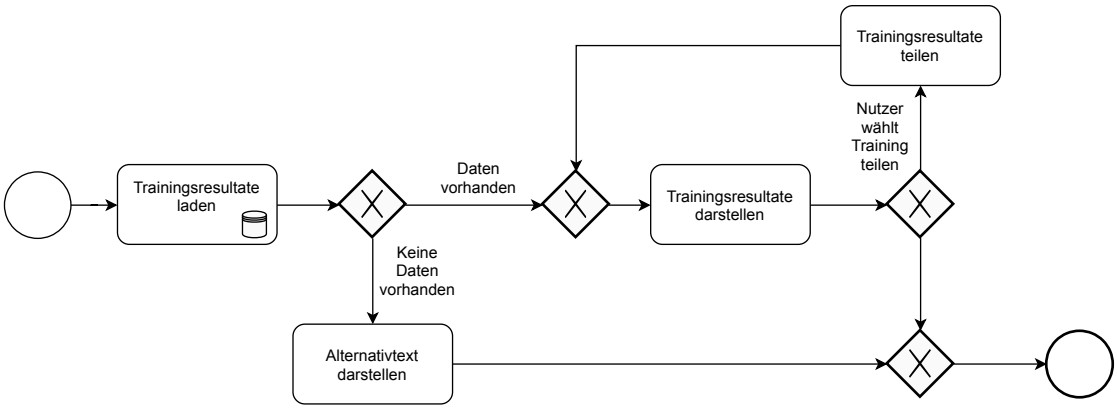

Abbildung 4.6: Übersicht Training

Abbildung 4.6 erläutert die Darstellung der Trainingsresultate. Gelangt der Benutzer über das Menü auf diese Seite, werden die Daten aus der Datenbank geladen und dargestellt. Liegen bisher noch keine Trainingsdaten vor, wird dem Benutzer alternativ ein Hinweis dargestellt. Wählt der Nutzer *Training teilen*, wird eine PDF-Datei mit den Daten erstellt, welche dann zum Beispiel per E-Mail weitergeleitet werden kann.

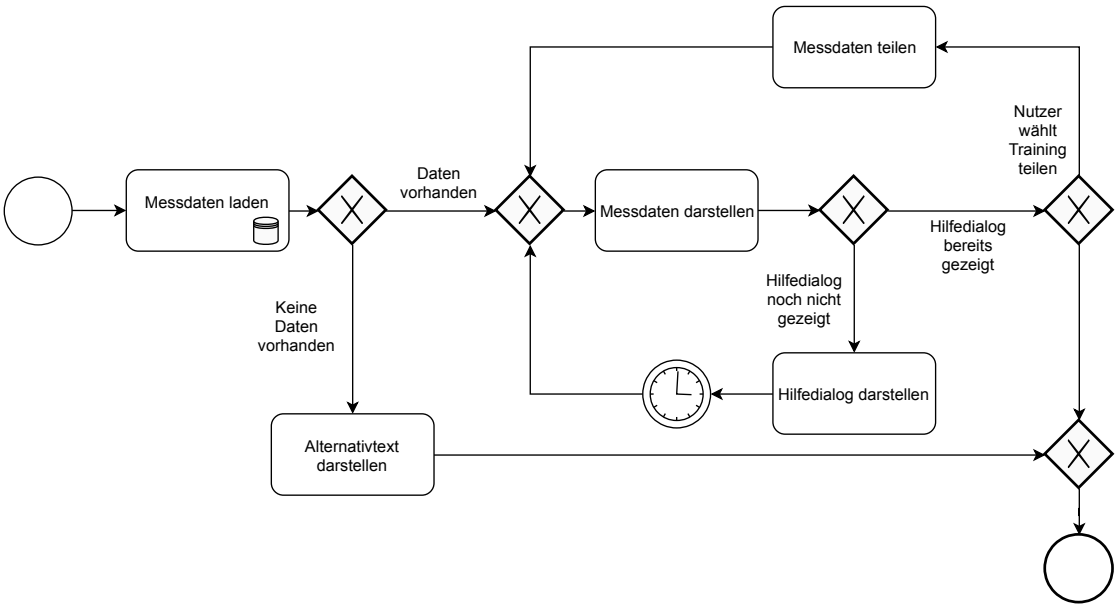

Abbildung 4.7: Übersicht Messungen

In Abbildung 4.7 erkennt man, dass der Prozessablauf bei der Übersicht der Messung derselbe ist, wie der, der Übersicht des Trainings. Der einzige Unterschied besteht darin, dass, sobald Daten vorhanden sind, ein Hilfsdialog bereitgestellt wird, der einmalig und so lange angezeigt wird, bis ihn der Benutzer schließt.

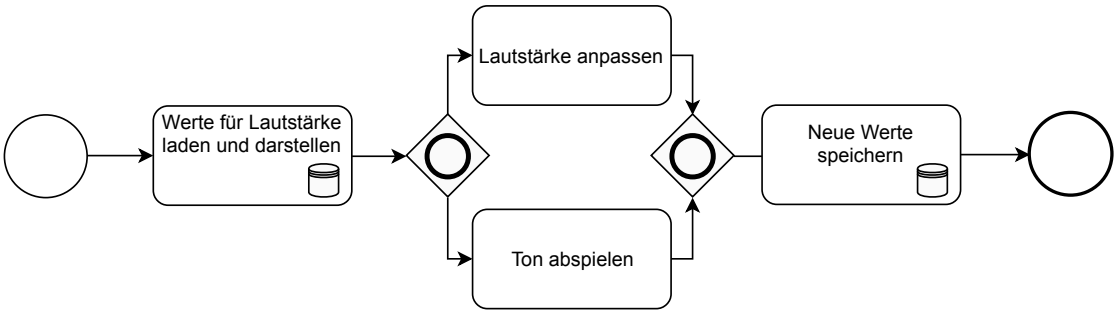

Abbildung 4.8: Einstellungen

In Abbildung 4.8 ist der Einstellungsprozess dargestellt. Wird die Seite von dem Nutzer geöffnet, werden die aktuellen Lautstärke-Werte aus der Datenbank geladen und angezeigt. Diese können dann angepasst werden und gleichzeitig können die verschiedenen Töne abgespielt werden. Klickt der Benutzer auf *speichern* werden die angepassten Werte in der Datenbank gespeichert.

4.2 Architekturkonzept

Im Folgenden wird das Architekturkonzept der Applikation erläutert. Dadurch wird die Hierarchie der Klassen, sowie die Verknüpfung der einzelnen Einheiten verdeutlicht. Das Architekturkonzept dient außerdem als Grundlage für den späteren Datenbankentwurf. Zur Darstellung wird das UML-Klassendiagramm verwendet.

Die Anwendung besteht aus Komponenten, welche das Datenmodell beinhalten (Model), die Benutzerschnittstelle implementieren (View) und das Programm steuern (Controller).

Abbildung 4.9 beschreibt die Beziehungen der zu implementierenden Klassen. Wie man sehen kann, erben alle Activities[1] von der `AppCompatActivity` [16]. Die `MainActivity` beinhaltet die Navigation, weshalb alle Seiten, die vom Menü aus erreichbar sind, durch Fragments[2] realisiert sind. Alle anderen Funktionalitäten sind durch eigene Activities implementiert. Die `LevelSelectionActivity` enhält außerdem ein `LevelSlideFragment`, um die Slidefunktion der Levelkarten realisieren zu können. Für Dialoge wird eine Instanz von `DialogHelper` erzeugt. Diese Hilfsklasse erbt von dem `DialogFragment` [18]. Das `TrainingOverviewFragment` erzeugt einen `ExpandableListAdapter`, der von der Klasse `BaseExpandableListAdapter` [19] erbt, um die Trainingsresultate als dynamische Liste anzeigen zu können. Um den Fall abfangen zu können, dass die Kopfhörer während einer Soundausgabe vom Gerät getrennt werden, wird in den Activities `Measurement1Activity` und `Measurement2Activity` eine private Klasse `HeadsetTask` implementiert. Diese erbt von von der Klasse

[1] Eine Activity ist der Einstiegspunkt für die Interaktion mit dem Benutzer. Sie stellt einen einzelnen Bildschirm mit einer Benutzerschnittstelle dar [15].

[2] Ein Fragment kann als modularer Abschnitt einer Activity gesehen werden. Es enthält eigene Eingabeereignisse und kann während der Ausführung einer Activity hinzugefügt oder entfernt werden [17].

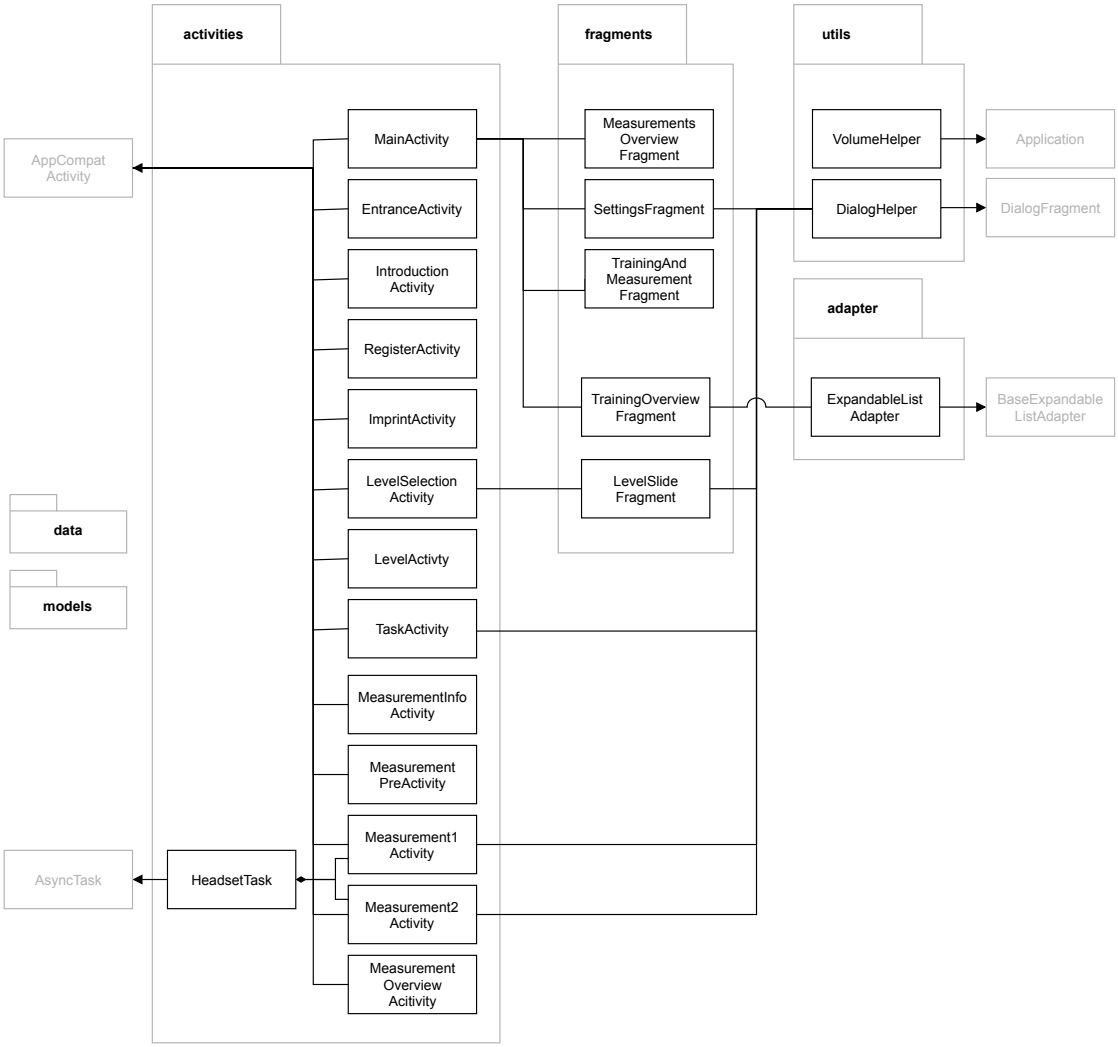

Abbildung 4.9: Klassendiagramm Allgemein

`AsyncTask` [20] und kontrolliert in einem parallel zum UI-Thread laufenden Prozess, ob die Kopfhörer noch verbunden sind. Das Package `utils` beinhaltet außerdem die Klasse `VolumeHelper`. Diese erweitert die Klasse `Application` [21] und wird beim Start der Anwendung angelegt. Sie kontrolliert, ob die Gerätelautstärke während der Benutzung geändert wurde und setzt diese auf den vorherigen Wert zurück, sobald die Anwendung in den Hintergrund rückt. Tritt die Anwendung anschließend wieder in den Vordergrund, stellt sie sicher, dass der Nutzer über die Lautstärke-Änderung informiert wird.

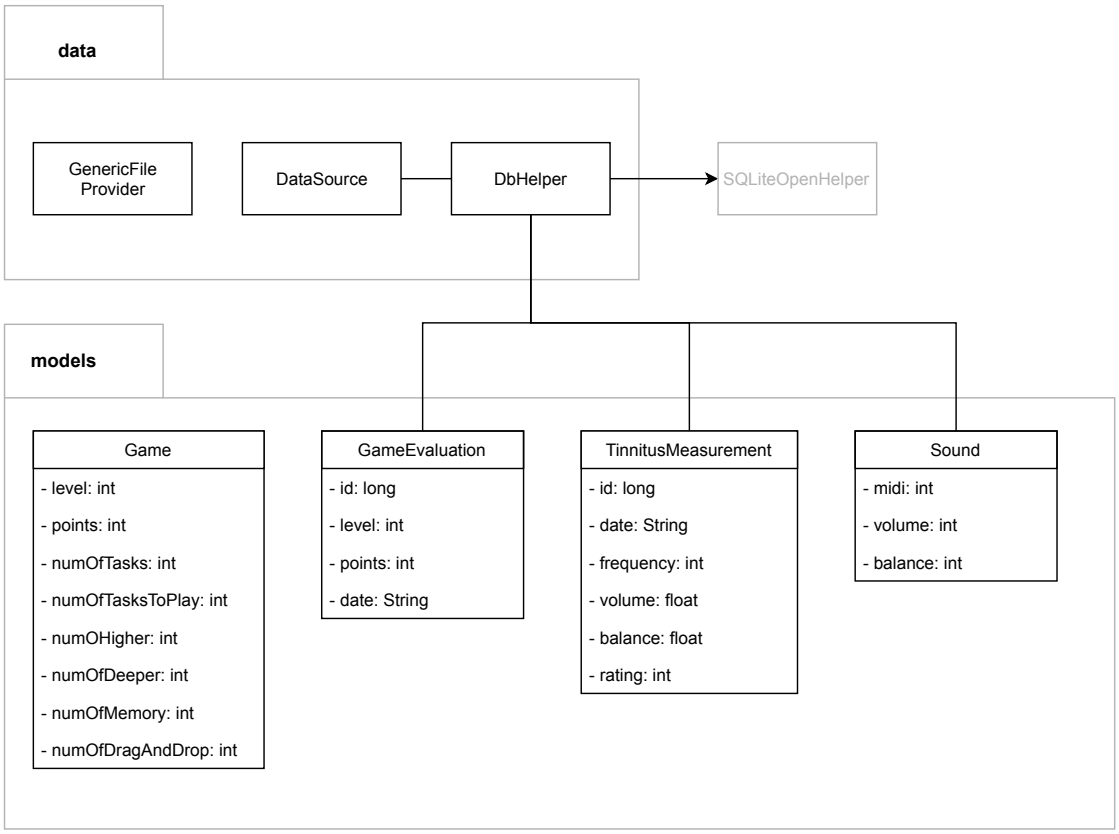

Abbildung 4.10: Klassendiagramm Models und Data Packages

In Abbildung 4.10 sind hauptsächlich die Klassen dargestellt, die für das SQLite Datenbanksystem benötigt werden. Instanzen, der sich in dem Package `models` befindlichen Klassen, können die Daten eines SQLite-Datensatzes aufnehmen. Mit Hilfe des `DbHelpers`, einer Subklasse des `SQLiteOpenHelper` [22], wird die Datenbank erstellt. Sie enthält wichtige Konstanten, die für die Arbeit mit der Datenbank benötigt

werden. Die Verwaltung der Daten übernimmt die `DataSource`. Über eine Instanz dieser Klasse kann eine Verbindung zur Datenbank erstellt und getrennt werden. Außerdem ist sie zuständig für das Auslesen, Hinzufügen, Ändern und Löschen von Datensätzen.

4.3 Datenbankschema

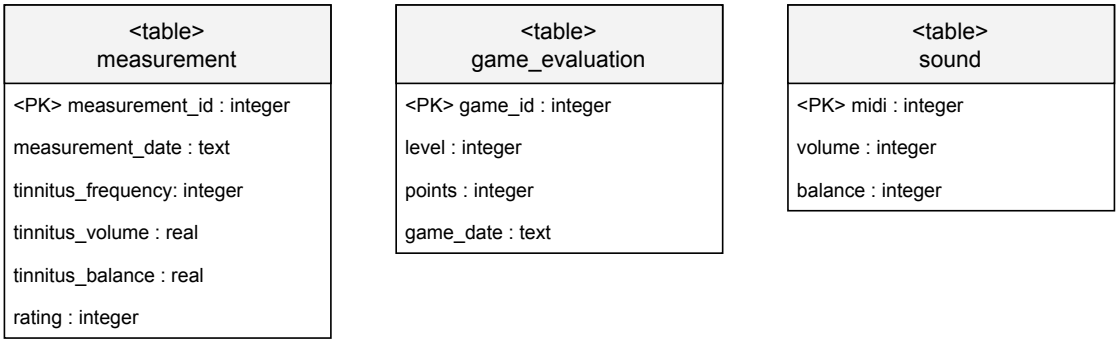

Abbildung 4.11: Datenbankschema

Das Datenbankmodell in Abbildung 4.11 beschreibt die Relationen der lokalen SQLite-Datenbank. Die Tabelle `measurement` speichert die Daten der Frequenzmessungen, `game_evaluation` beinhaltet die Trainingsresultate und `sound` enthält die Werte der einzelnen Frequenz-Intervalle, welche in den Einstellungen geändert werden.

4.4 Verwendete Bibliotheken

Im Folgenden werden die Bibliotheken erläutert, die verwendet wurden.

4.4.1 pd-for-android

Die Pure Data Bibliothek pd-for-android [23] für Android wurde zur Soundgenerierung genutzt. Näheres zur Verwendung kann dem Kapitel 5.1.1 entnommen werden. Die Bibliothek ist veröffentlicht unter der 3-Clause BSD License [24].

4.4.2 GraphView

Um die Übersicht der Frequenzmessung anschaulich darstellen zu können, wurde die Bibliothek GraphView [25] verwendet. Durch diese können einfach Diagramme erstellt und angepasst werden. Die Bibliothek steht unter der Apache v2 Lizenz [26].

4.4.3 SeekArc

Die Bibliothek SeekArc [27] wurde verwendet um runde SeekBars erstellen zu können, wie sie bei der Frequenzmessung zum Einsatz kommen. Dabei haben sie annähernd die selbe Funktionalität wie SeekBars. Die Bibliothek wurde von Neil Davies geschrieben und veröffentlicht unter der MIT-Lizenz [28]. Das Copyright besitzt Triggertrap Ltd..

5

Ausgewählte Implementierungsdetails

Im Folgenden werden einige Implementierungsdetails aus der Match Your Tinnitus Applikation vorgestellt. Dabei wird insbesondere auf die Tongenerierung mit Pure Data und die Frequenzmessung als Hauptfunktionalität der Anwendung eingegangen. Bei dem eingefügten Code handelt es sich um Codeausschnitte, welche alleinstehend nicht unbedingt lauffähig sind.

5.1 Tongenerierung

Für die Tongenerierung der Match Your Tinnitus App wurde Pure Data [29] verwendet. Im Nachfolgenden wird eine kurze Einführung zu Pure Data gegeben, um im Anschluss die realisierte Pure Data Datei erläutern zu können.

5.1.1 Pure Data

Pure Data ist eine datenstromorientierte Programmiersprache und eine visuelle Open-Source Entwicklungsumgebung zur Klangerzeugung in Echtzeit.
Eine Pure Data Datei wird *Patch* genannt.
Ein Patch besteht wiederum aus verschiedenen Elementen, dargestellt als Boxen, die miteinander verbunden sind [29].

Ein Pure Data Element kann Eingänge (siehe 1 in Abbild 5.1) und Ausgänge (siehe 2 in Abbild 5.1) besitzen.

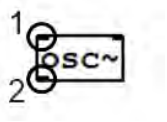

Abbildung 5.1: Pure Data Element

Über diese können die einzelnen Elemente verbunden und Signale oder Werte übertragen werden.

Für die Erstellung des Patches wurden drei verschiedene Elementtypen verwendet. Diese kann man in Abbildung 5.2 erkennen. Objekte sind gekennzeichnet durch rechteckige Boxen, Nachrichten durch Fahnen ähnelnden Boxen und Zahlen durch rechteckige Boxen mit einer abgeschnittenen Ecke.

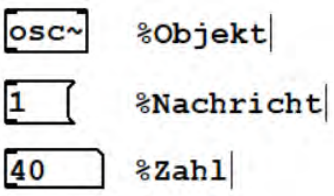

Abbildung 5.2: Pure Data Elementtypen

Zu erläuternde Objekte sind **osc**∼, **dac**∼ und **line**∼.

Das Objekt **osc**∼ (Oszillator) wandelt eine übergebene Frequenz in eine Kosinus-Welle um [30].
Das Objekt **dac**∼ (Digital Audio Converter) spricht die Soundkarte auf dem Gerät an, es werden damit also die Töne erzeugt. Der linke Eingang ist verantwortlich für die Ausgabe auf dem linken Kanal und umgekehrt ist der rechte Eingang verantwortlich für die Ausgabe auf dem rechten Kanal.
Durch das Objekt **line**∼ wird von dem momentanen Wert auf den ersten Wert der eingehenden Nachricht gewechselt. Dies geschieht über ein bestimmtes Zeitintervall, das durch den zweiten Wert der Nachricht festgelegt wird. Gewechselt bedeutet dabei, dass zwischen zwei Werten linear interpoliert wird, sodass große Sprünge zwischen Werten vermieden werden [31].

Pure Data Realisierung

Öffnet man das entworfene Pure Data Patch als graphische Repräsentation, erhält man ein Fenster mit der in Abbildung 5.3 dargestellten Ansicht.

Die Objekte, die mit einem **r** beginnen, sind sogenannte Empfänger-Objekte. Sie erhalten Nachrichten, die von den jeweiligen Android Activities gesendete werden (nähere Informationen darüber in Kapitel 5.1.2). Zum besseren Verständnis wurde das entwor-

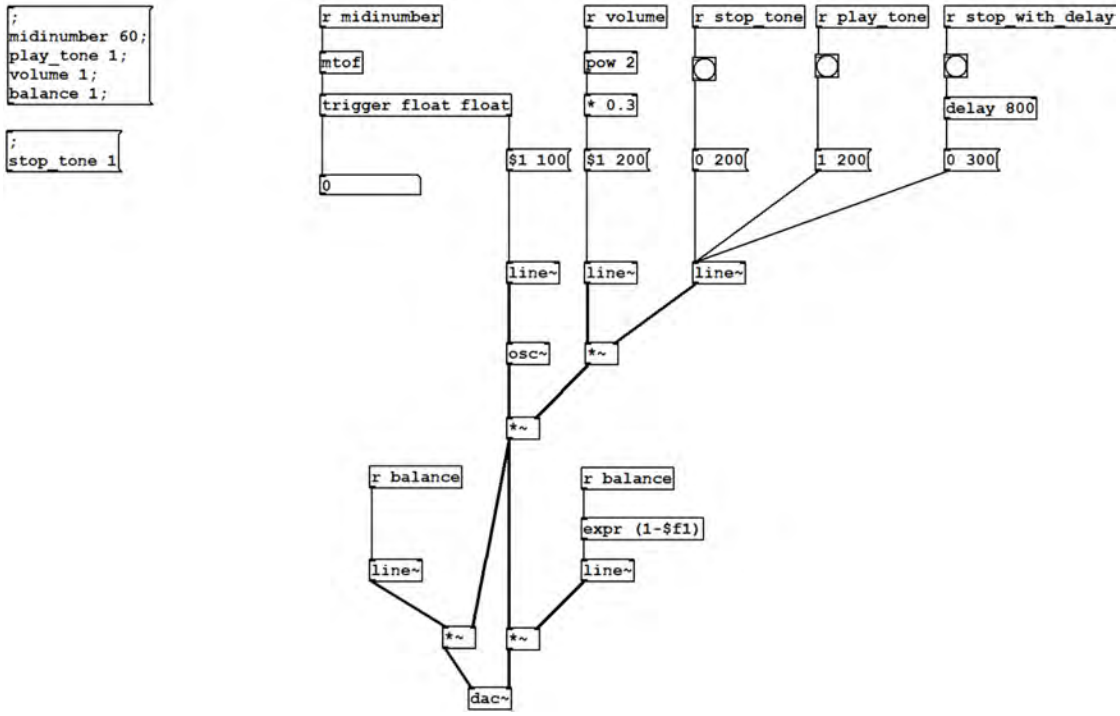

Abbildung 5.3: Pure Data Patch

fenen Patch durch zwei beispielhafte Nachrichten ergänzt (siehe links, oben), wie sie auch durch Android Activities geschickt werden könnten.

Diese Empfänger-Objekte leiten die Werte der empfangenen Nachricht weiter.

Die empfangene MIDI Tastennummer wird dabei durch **mtof** zuerst in eine Frequenz und dann durch den Oszillator in eine Kosinus Wellenform umgewandelt.

Bei einer linearen Skalierung der Lautstärkeregler würde das Wachstum der Lautstärke als logarithmisch empfunden werden. Aus diesem Grund wird der empfangene Lautstärke-Wert mit e potenziert, um so eine logarithmische Skalierung und damit ein lineares Wachstum der Lautstärkeempfindung zu ermöglichen. Außerdem wird der Wert mit 0.3 multipliziert, um das allgemeine Lautstärke-Niveau auf einen angemessenen Wert zu verringern. Im Anschluss wird eine Interpolation in einem Zeitintervall von 200 ms durchgeführt.

Welche Werte *stop_tone*, *play_tone* und *stop_tone* empfangen ist nicht von Belangen, da das Weiterleiten des Wertes einen sogenannten *bang* wirft. Dies führt dazu, dass die

momentane relative Lautstärke mit 1 oder 0 interpoliert wird, sodass ein Ton entweder innerhalb von 200 ms angeschaltet, oder nach 800 ms innerhalb von 300 ms ausgeschaltet wird. Man kann damit einen Ton erzeugen, der solange spielt bis *play_ton* eine Nachricht empfängt. Dies wird verwendet bei der Tinnitusfrequenzmessung und bei den Einstellungen. Es kann jedoch auch ein auf 800 ms begrenzt spielender Ton erzeugt werden, indem den Empfänger-Objekten *play_tone* und *stop_with_delay* gleichzeitig eine Nachricht geschickt wird.

Die Kosinuswelle wird nun noch mit der relativen Lautstärke multipliziert und, abhängig von der empfangenen Balance, auf den linken und rechten Kanal verteilt. Anschließenden wird der Ton erzeugt.

5.1.2 Kommunikation mit Pure Data

Um Pure Data in Android nutzen zu können, muss die *pd-for-android*-Bibliothek zu den Gradle Dependencies hinzugefügt werden, siehe Listing 5.1.

build.gradle

```
dependencies {
        [...]
        implementation 'org.puredata.android:pd-core:1.0.2'
}
```

Listing 5.1: Einbindung der pd-for-android Bibliothek in build.gradle

Um einen Ton spielen zu können, muss zunächst eine Verbindung zu Pure Data aufgebaut und das Pure Data Patch geladen werden. Über textttPdBase können dann ganz einfach Nachrichten verschickt werden, die dann vom Patch verarbeitet werden. Das Listing 5.2 zeigt, wie dies genau implementiert ist.

```
import org.puredata.android.io.AudioParameters;
import org.puredata.android.io.PdAudio;
import org.puredata.android.utils.PdUiDispatcher;
import org.puredata.core.PdBase;
import org.puredata.core.utils.IoUtils;

```

```
7  public class TaskActivity extends AppCompatActivity{
8
9      PdUiDispatcher pdUiDispatcher;
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_task);
15
16         try {
17            initPD();
18            loadPDPatch();
19         } catch (IOException e) {
20            e.printStackTrace();
21         }
22
23         playTone(60);
24
25     }
26
27     // Initialise Pure Data
28     private void initPD() throws IOException {
29         int sampleRate = AudioParameters.suggestSampleRate();
30         PdAudio.initAudio(sampleRate, 0, 2, 8, true);
31         pdUiDispatcher = new PdUiDispatcher();
32         PdBase.setReceiver(pdUiDispatcher);
33     }
34
35     // Load Pure Data Patch
36     private void loadPDPatch() throws IOException {
37         File dir = getFilesDir();
38         IoUtils.extractZipResource(getResources()
39             .openRawResource(R.raw.pdpatch), dir, true);
40         File pdPatch = new File(dir, "pdpatch.pd");
41         PdBase.openPatch(pdPatch.getAbsolutePath());
42     }
43
44     private void playTone(int midiNumber) {
```

```
45          // Get Volume and Balance from Database
46          volume = getVolumeFromDb(midiNumber);
47          balance = getBalanceFromDb();
48          // Send messages to the patch via PdBase
49          PdBase.sendFloat("midinumber", midiNumber);
50          PdBase.sendFloat("play_tone", 1);
51          PdBase.sendFloat("volume", volume);
52          PdBase.sendFloat("balance", balance);
53          PdBase.sendFloat("stop_with_delay", 1);
54      }
55  }
```

Listing 5.2: Kommunikation mit dem Pure Data Patch

5.1.3 Tongenerierung nur mit verbundenen Kopfhörern

Da der Nutzer nur dann Töne abspielen können soll, wenn das verwendete Android Gerät mit Kopfhörern verbunden ist, muss in allen Activities, die Nachrichten an das Pure Data Patch schicken, kontrolliert werden, ob diese Bedingung erfüllt ist. Dadurch soll gewährleistet werden, dass das Hörtraining und die Frequenzmessung immer unter gleichen Bedingungen stattfindet und die Balance Einstellungen auch tatsächlich berücksichtigt werden.

Dabei ist es wichtig diese Bedingung nicht nur zu überprüfen, wenn eine solche Activity geöffnet bzw. ein Ton abgespielt wird, sondern auch die ganze Zeit über während der Ton spielt, da es offensichtlich jederzeit passieren kann, dass der Benutzer aktiv oder aus Versehen die Kopfhörer vom Gerät trennt.

Dadurch ergibt sich mit Hilfe der `HeadsetTask` folgende Implementierung:

```
1  private class HeadsetTask extends AsyncTask<String, Integer, String> {
2
3          @Override
4          protected void onPreExecute() {
5              super.onPreExecute();
6              // Create new AlertDialog
```

```
7               AlertDialog alertDialog = new
                    AlertDialog.Builder(Measurement1Activity.this)
8                       .setMessage(R.string.headset_sound_error)
9                       .setPositiveButton(getString(R.string.ok),
10                          new DialogInterface
11                                  .OnClickListener() {
12                                          public void onClick(
13                                          DialogInterface dialog, int id) {
14                                  // User clicked OK button
15                                  dialog.dismiss();
16                          }
17                      });
18          }
19
20          @Override
21          protected String doInBackground(String ...params) {
22                  // While sound is playing check if headset is disconnected
23              while(soundIsPlaying){
24                  if(!audioManager.isWiredHeadsetOn()){
25                      headsetError = true;
26                      return null;
27                  }
28              }
29              headsetError = false;
30              return null;
31          }
32          protected void onPostExecute(String result) {
33                  // Show AlertDialog and stop playing sound if a headsetError
                        has occured
34              if(headsetError) {
35                  alertDialog.show();
36                  stopSound();
37                  playSwitch.setChecked(false);
38              }
39          }
40      }
```

Listing 5.3: Asychron laufendene HeadsetTask

5.2 Bestimmung der MIDI Tastennummern im Training

Für die Aufgaben im Training soll pro Aufgabe eine Liste mit MIDI Nummern erzeugt werden. Die Anzahl der Elemente ist abhängig von der Aufgabe und muss deshalb an die Methode übergeben werden. Außerdem soll die Differenz zwischen den einzelnen MIDI Nummern mit zunehmendem Level abnehmen. Wichtig ist des weiteren, dass die Töne zufällig gemischt werden, um ein sinnvolles Training zu erstellen. So kann jede Aufgabe beliebig oft wiederholt werden, ohne, dass sich eine sich wiederholende Struktur ergibt.

Um später erkennen zu können, ob eine Aufgabe richtig gelöst wurde, muss zusätzlich zu der Liste mit zufälligen MIDI Nummern eine Liste erstellt und zurückgegeben werden, die die Indizes der MIDI Nummern in geordneter Reihenfolge enthält. Die Implementierung dazu kann Listing 5.4 entnommen werden.

```
1  private HashMap<String, List<Integer>> getRandomMidis(int number) {
2      List<Integer> midis = new ArrayList<>();
3      List<Integer> midisSequence = new ArrayList<>();
4      HashMap<String, List<Integer>> midisWithSequence = new HashMap<>();
5      List<Integer> helper = new ArrayList<>();
6      int midiDifference;
7      switch (game.getLevel()) {
8          case 1:
9              midiDifference = 5;
10             break;
11         case 2:
12             midiDifference = 4;
13             break;
14         case 3:
15             midiDifference = 3;
16             break;
17         case 4:
18             midiDifference = 2;
19             break;
20         default:
21             midiDifference = 1;
22             break;
```

```
23      }
24      // Get a random value between 55 and 120
25      int helperMidi = ThreadLocalRandom.current().nextInt(55, 120);
26      for (int i = 0; i < number; i++) {
27          midis.add(helperMidi + i * midiDifference);
28          helper.add(helperMidi + i * midiDifference);
29          }
30      }
31      // Shuffle the list to get a random sequence
32      Collections.shuffle(midis);
33      // Save sequence to the midiSequence list
34      for (int i = 0; i < number; i++) {
35         for (int j = 0; j < number; j++) {
36             if (Objects.equals(midis.get(i), helper.get(j))) {
37                 midisSequence.add(j);
38             }
39         }
40      }
41      midisWithSequence.put("midis", midis);
42      midisWithSequence.put("sequence", midisSequence);
43      return frequenciesWithSequence;
44  }
```

Listing 5.4: Erzeugung zufälliger MIDI Tastennummern

5.3 Hilfestellung

Dem Benutzer soll bei der erstmaligen Benutzung der Messdaten-Übersicht eine Hilfestellung anhand eines Dialogs geboten werden. Um sicherzustellen, dass diese auch tatsächlich nur einmal angezeigt wird, wird ein Schlüssel in `SharedPreferences` [32] gespeichert, sobald er angezeigt wurde. Dies ist wie folgt implementiert:

```
1 SharedPreferences sharedPreferences =
      getActivity().getSharedPreferences(getString(R.string.preferences_key),
      Context.MODE_PRIVATE);
2 SharedPreferences.Editor editor = sharedPreferences.edit();
3 editor.putString(getString(R.string.explanation_measurement_key), "true");
```

Listing 5.5: Speichern eines Schlüssels

Öffnet der Benutzer dann erneut die Übersicht, wird abgefragt, ob der Schlüssel bereits existiert:

```
1 private boolean explanationShowed() {
2         // Get private SharedPreferences
3         SharedPreferences sharedPreferences =
              getActivity().getApplicationContext().getSharedPreferences(
4         getString(R.string.preferences_key), Context.MODE_PRIVATE);
5        // If no value is stored in explanation_measurement_key yet return
             true, else return false
6        sharedPreferences.getString(getString(
7        R.string.explanation_measurement_key),null) != null;
8 }
```

Listing 5.6: Methode zum Erkennen, ob bereits eine Hilfestellung gegeben wurde

Falls kein Schlüssel existiert, wird die Hilfestellung angezeigt, ansonsten nicht.

6

Vorstellung der Anwendung

In diesem Kapitel werden die Funktionalitäten der prototypischen Anwendung aus Sicht des Benutzers und anhand von Bildschirmaufnahmen erläutert.

6.1 Allgemeines

Wann immer der Benutzer die Anwendung neu öffnet, erscheint ihm das *Match Your Tinnitus* Logo, dargestellt in Abbildung 6.1, bevor er zur eigentlichen Anwendung weitergeleitet wird.

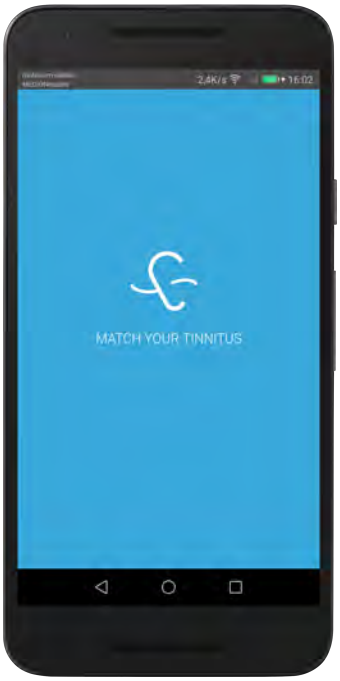

Abbildung 6.1: Logo Ansicht

Abhängig von der Geräteeinstellung wird außerdem der Inhalt der Anwendung auf deutsch oder auf englisch angezeigt.

Da der Nutzer während der Benutzung aufgefordert wird die Lautstärke voll aufzudrehen, um eine einheitliche und vergleichbare Nutzung zu ermöglichen, wird die Lautstärke, sobald die Anwendung in den Hintergrund rückt, auf den vorherigen Wert zurückgesetzt. Rückt die Anwendung im Anschluss erneut in den Vordergrund, wird der Benutzer über die Änderung informiert.

Des Weiteren ist die Ausgabe von Tönen und das Spielen eines Levels nur dann gestattet, wenn Kopfhörer mit dem Gerät verbunden sind. Ist dies nicht der Fall, wird

der Benutzer darauf hingewiesen. Werden die Kopfhörer beim Spielen eines Tones vom Gerät getrennt, wird der Ton gestoppt und der Nutzer informiert.

6.2 Erstmaliges Öffnen der Anwendung

Abbildung 6.2: Begrüßung

Abbildung 6.3: Angabe des Namens

Bei der erstmaligen Nutzung der Anwendung wird dem Benutzer eine Einführung angezeigt, die die Funktionalitäten von *Match Your Tinnitus* erläutert. Diese kann man in Abbildung 6.2 sehen. Klickt der Nutzer auf *weiter* wird er auf aufgefordert seinen Namen einzugeben (siehe Abbildung 6.3). Wurde ein Name angegeben, kann der Nutzer auf *jetzt starten!* drücken und wird daraufhin zur eigentlichen Anwendung weitergeleitet.

6.3 Navigation

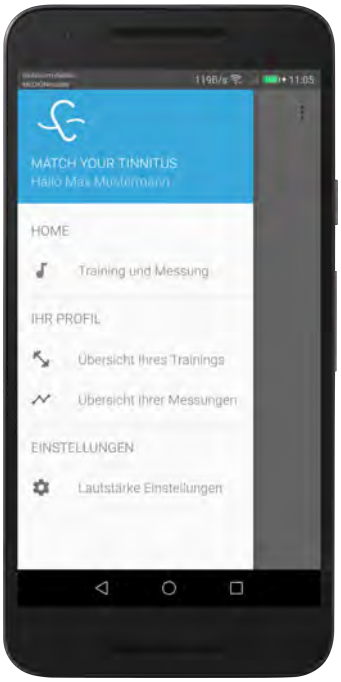

Abbildung 6.4: Hauptmenü

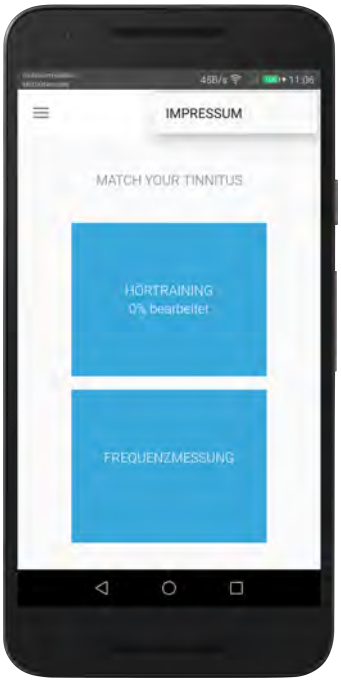

Abbildung 6.5: Menü

Die Navigation innerhalb der Anwendung erfolgt über das Hauptmenü, siehe Abbildung 6.4. Dieses Hauptmenü kann durch einen Klick auf das Menüzeichen in der Toolbar links oder durch eine Wischgeste nach rechts über den linken Bildschirmrand geöffnet und durch ein Tippen in dem Bereich neben dem Menü oder wiederum durch eine Wischgeste nach links geschlossen werden. Das Menü ist in die Kategorien *Home*, *Ihr Profil* und *Einstellungen* gegliedert. Dabei sind jeweils nur deren Unterpunkte klickbar.

Beim Öffnen der App befindet sich der Benutzer auf der Seite *Training und Messung*, siehe Abschnitt 6.4. Durch einen Klick auf *Übersicht Ihres Trainings* gelangt der Nutzer auf eine Übersichtsseite, auf der die Resultate seines Trainings aufgelistet sind. Diese ist erläutert in Abschnitt 6.5. Über den Punkt *Übersicht Ihrer Messungen* wird der Benutzer auf eine Seite mit einer graphischen und textuellen Darstellung seiner Messungen geleitet, die in Abschnitt 6.6 beschrieben ist. Es gibt außerdem die Möglichkeit Einstellungen, siehe Abschnitt 6.7, vorzunehmen. Diese sind über *Lautstärke Einstellungen* zugänglich.

Zum Hauptmenü gibt es zudem ein weiteres Menü, das in Abbildung 6.5 dargestellt ist. Dieses lässt sich durch einen Klick auf die drei Punkte rechts in der Toolbar öffnen. Über dieses Menü kann man durch Klicken von *Impressum* zum Impressum gelangen.

6.4 Training und Messung

Von der Seite Training und Messung, siehe Abbildung 6.6, hat der Nutzer die Möglichkeit mit einem Klick auf die obere Karte zum Hörtraining zu gelangen, siehe Abschnitt 6.4.1, oder mit einem Klick auf die untere Karte eine neue Frequenzmessung, siehe Abschnitt 6.4.2, zu starten.

Dem Benutzer wird auf dieser Seite außerdem dargestellt welchen Anteil des Trainings er bereits absolvierte und wann seine letzte Frequenzmessung stattfand.

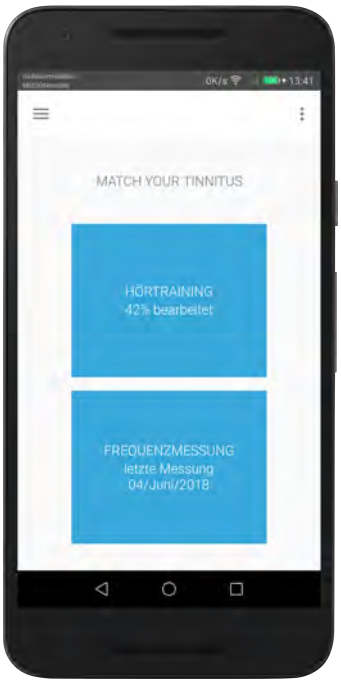

Abbildung 6.6: Training und Messung

6.4.1 Hörtraining

Trainingsaufbau

Zu Beginn des Trainings hat der Benutzer nur die Möglichkeit Level 1 zu spielen. Wurde ein Level erfolgreich absolviert, wobei erfolgreich bedeutet, dass mindestens 80 Punkte erzielt wurden, wird ein neues Level freigeschaltet.

Abbildung 6.7 zeigt, dass ein Level drei verschiedene Zustände haben kann:
1. Freigeschaltet und bereits absolviert. Dieses Level kann man *wiederholen*.
2. Freigeschaltet und noch nicht absolviert. Dieses Level kann man *beginnen*.
3. *Noch nicht freigeschaltet*. Dieses Level kann noch nicht gespielt werden. Versucht der Nutzer ein solches Level durch Klick auf die Karte zu öffnen, bekommt er den

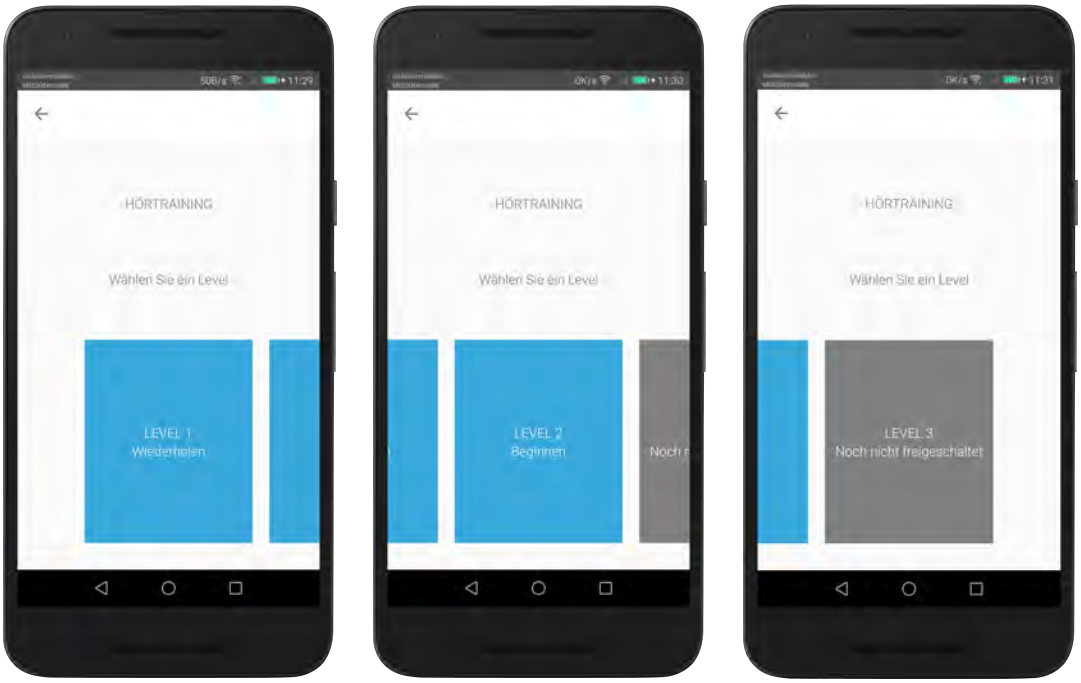

Abbildung 6.7: Levelauswahl mit jeweiligem Trainingsstatus

Hinweis, dass das Level erst dann freigeschaltet wird, wenn er mindestens 80 Punkte im vorherigen Level erzielen konnte.
Ein Level lässt sich durch einen Klick auf die jeweilige Levelkarte öffnen. Dies ist nur dann möglich, wenn Kopfhörer im Gerät angeschlossen sind. Ist dies nicht der Fall wird dem Benutzer ein Hinweis angezeigt.

Levelaufbau

Ein Level besteht aus einer pro Level festgelegten Anzahl an Aufgaben aus vier verschiedenen Aufgabentypen. Alle Aufgabentypen kann man in Abbildung 6.8 erkennen. Die Reihenfolge, in der dem Nutzer diese Aufgaben vorgelegt werden, ist dabei zufällig. Der Benutzer hat zu Beginn 100 Punkte. Bei jedem gemachten Fehler werden davon fünf Punkte abgezogen.

Abbildung 6.8: Aufgabentypen, v.l.n.r Höher, Tiefer, Memory, Ordnen

Am Ende eines Levels wird dem Benutzer die erreichte Punktzahl angezeigt (siehe Abbildung 6.9).

Außerdem wird der Benutzer darüber benachrichtigt, wenn ein neues Level freigeschaltet wurde. Ist dies der Fall oder ist das nächste Level bereits freigeschaltet, hat der Nutzer die Möglichkeit gleich mit dem nächsten Level fortzufahren. Hat es der Benutzer nicht geschafft mindestens 80 Punkte zu erreichen, gibt es die Möglichkeit das Level zu wiederholen. Möchte er das Training beenden oder ein anderes Level auswählen, kann er das Training auch beenden und wird daraufhin zurück zur Levelauswahl geleitet.

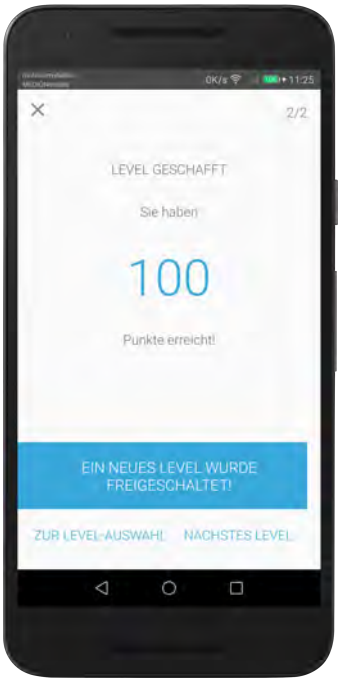

Abbildung 6.9: Level Übersicht

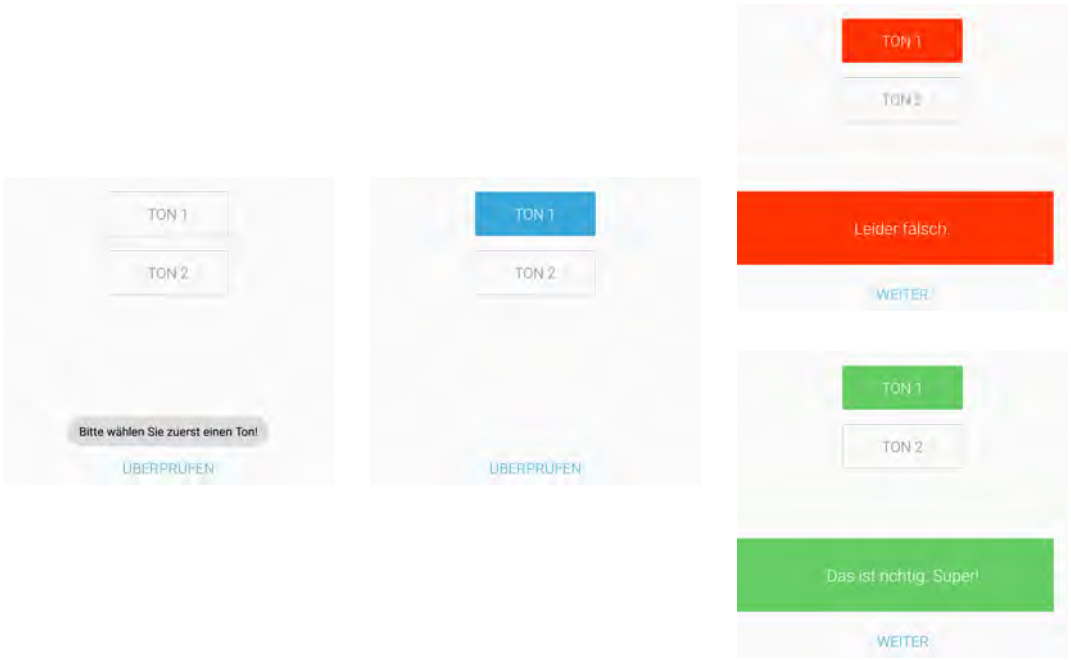

Abbildung 6.10: Spielablauf Aufgabentyp Höher/Tiefer

Aufgabentypen Höher und Tiefer

Bei den Aufgabentypen Höher und Tiefer kann der Benutzer beide Buttons klicken und hört daraufhin jeweils einen zeitlich begrenzten Ton. Wird ein Button geklickt, gilt er gleichzeitig als ausgewählt und wird blau gefärbt. Drückt der Nutzer danach den *überprüfen*-Button, bekommt er die Auflösung eingeblendet und der *weiter*-Button erscheint. Diesen Spielverlauf kann man in Abbildung 6.10 erkennen. Will der Benutzer die Aufgabe aufzulösen, bevor ein Button ausgewählt wurde, wird der Hinweis angezeigt, dass erst ein Ton gewählt werden muss.

Aufgabentyp Memory

Ein weiterer Aufgabentyp ist das Ton-Memory. Hier muss der Benutzer Ton-Paare mit derselben Frequenz finden. Da der Nutzer höchstens zehn Versuche benötigt, um alle Paare finden zu können, hat er genauso viele Versuche. Für jedes weitere aufgedeckte Paar werden ihm fünf Punkte abgezogen. Dem Benutzer wird angezeigt, ob er ein

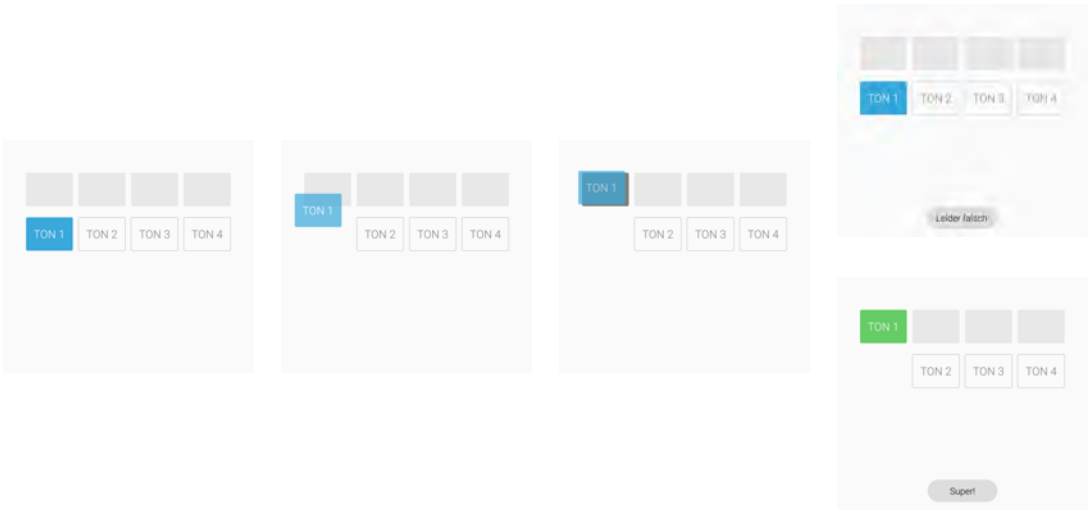

Abbildung 6.11: Spielablauf Aufgabentyp Ordnen

richtiges Paar gefunden hat, indem dieses grün gefärbt wird. Werden zwei Töne gewählt, die nicht zusammenpassen, färben sich diese kurz rot, um dann wieder grau und damit erneut auswählbar zu werden.

Aufgabentyp Ordnen

Bei dem Aufgabentyp Ordnen muss der Benutzer die Töne abhängig von ihrer Frequenz in die richtige Reihenfolge bringen. Ein beispielhafter Verlauf ist in Abbildung 6.11 ersichtlich. Klickt der Benutzer auf einen Ton-Button wird der Ton für einen Moment gespielt. Hält der Nutzer einen Ton gedrückt, so löst sich der Button vom Hintergrund und kann verschoben werden. Wird der Button auf einem Feld losgelassen und er befindet sich nicht auf der richtigen Position, bewegt sich der Button zurück zu seinem Ausgangspunkt. Wurde der Button jedoch auf das richtige Feld bewegt, heftet er sich dort an und färbt sich grün. Der Nutzer bekommt jeweils zusätzliches textuelles Feedback angezeigt.

6.4.2 Frequenzmessung

Abbildung 6.12: Information zur Messung

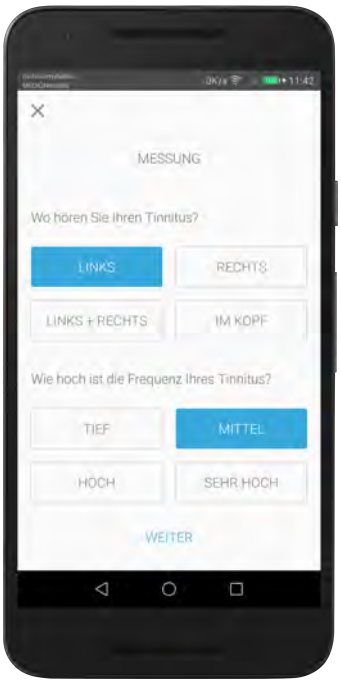

Abbildung 6.13: Vorauswahl zur Messung

Die eigentliche Hauptfunktion der Anwendung ist die Frequenzmessung. Wählt der Benutzer diese aus, wird ihm zuallererst eine Information über die Messung angezeigt (Abbildung 6.12). Nach dieser muss die Lautstärke des Geräts voll aufgedreht und das Gerät mit Kopfhörern verbunden haben.

Drückt der Nutzer auf *jetzt starten!*, wird er auf eine Seite weitergeleitet auf der er angeben muss wo er seinen Tinnitus hört und wie hoch er die Frequenz seines Tinnitus subjektiv einschätzt (Abbildung 6.13). Dabei gibt es für Ersteres die Auswahlmöglichkeiten *links*, *rechts*, *links + rechts* und *im Kopf* und für die Höhe des Tinnitus die Möglichkeiten *tief*, *mittel*, *hoch* und *sehr hoch*. Wurden nicht durch einen Klick auf den jeweiligen Button für beide Kategorien eine Auswahl getroffen, wird der Nutzer aufgefordert dies nachzuholen. Ist dies nicht der Fall, kann der Nutzer fortfahren und gelangt zur eigentlichen Messung.

Die Messung (Abbildung 6.14) ist in 3 Schritte geteilt:
Schritt 1: Der Nutzer spielt einen Ton ab. Dies wird durch Klick auf den An-/Ausschalter

Abbildung 6.14: Frequenzbestimmung

Abbildung 6.15: Lautstärke- und Balancebestimmung

ausgeführt. Durch den Schalter kann der Vergleichston jederzeit gestoppt und gestartet werden, um beispielsweise den Tinnitus besser hören und vergleichen zu können.
Schritt 2: Der Nutzer dreht die Lautstärke auf einen für ihn angenehmen Wert. Die Lautstärke kann zusätzlich jederzeit angepasst werden.
Schritt 3: Über den runden Regler kann der Benutzer nun die Frequenz des spielenden Tons so anpassen, dass sie mit der Frequenz des Tinnitus übereinstimmt. Die Frequenz lässt sich ändern, indem der blaue Kreis auf dem Regler gedrückt und gleichzeitig verschoben wird. Sind die Abstände zwischen den einzelnen Frequenzen zu groß, sodass die passende Frequenz nicht getroffen werden kann, hat der Benutzer zusätzlich die Möglichkeit durch einen Klick auf den *fein*-Button neben dem Regler die Abstände zwischen den einzelnen Frequenzen zu vergrößern. Im Fein-Modus kann der Ton, der beim Klick auf den Button eingestellt war, jeweils um eine halbe Oktave nach oben bzw. nach unten angepasst werden. Der Benutzer kann jederzeit zwischen den Modi Fein und Grob wechseln.

Ist der Benutzer zufrieden mit seiner Messung, gelangt er über den *weiter*-Button auf die nächste Seite (Abbildung 6.15). Hier kann der Benutzer zusätzlich die relative Lautstärke und Balance seines Tinnitus messen. Über den An-und Ausschalter kann der Ton, dessen Frequenz auf der vorherigen Seite gemessen wurde, abgespielt werden. Die Anpassung der Lautstärke und der Balance geschieht wiederum über einen Regler. Ist der Benutzer mit diesen Anpassungen zufrieden kann er auch hier wieder über den *weiter*-Button auf die nächste Seite gelangen.

Nun befindet sich der Nutzer auf der Übersichtsseite (Abbildung 6.16). Hier wird die gemessene Frequenz dargestellt. Des Weiteren kann die Messung dahingehend bewertet werden, wie zufrieden die betreffende Person mit ihrer Messung ist. Diese Bewertung geht von 0 bis 5 und kann durch einen Klick auf den n-ten Stern getroffen werden. Bis zu diesem Zeitpunkt kann der Benutzer die Messung durch Drücken des *X*-Buttons oben links jederzeit beenden. Versucht er dies, wird er über einen Dialog darauf hingewiesen, dass der Fortschritt beim Verlassen nicht gespeichert wird und gefragt, ob er die Messung wirklich abbrechen möchte. Ist dies der Fall, wird er auf die Auswahlseite von Training und Messung zurück geleitet. Möchte er die Messung jedoch nicht abbrechen, sondern speichern, kann er dies über den Button *Messung speichern und beenden* bestätigen. Daraufhin wird die Messung gespeichert und der Nutzer wird auf die Auswahlseite von Training und Messung geleitet.

Abbildung 6.16: Messung

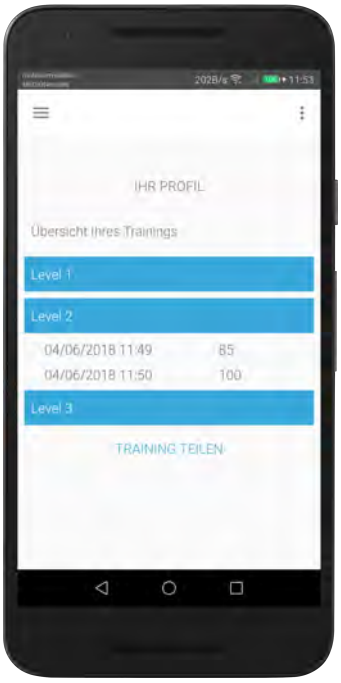

Abbildung 6.17: Übersicht des Trainings

Abbildung 6.18: Übersicht der Messungen

6.5 Übersicht des Trainings

In Abbildung 6.17 kann man die Übersicht der Trainingsresultate erkennen. Klickt der Benutzer auf eine Levelkarte, klappen sich die Resultate des Levels aus/ein. Der Benutzer sieht daraufhin, wann er dieses Level absolviert hat und wie viele Punkte er jeweils erreichen konnte. Außerdem kann der Nutzer die Resultate mit Freunden oder seinem Arzt teilen, indem er auf *Training teilen* klickt. Diese Resultate können dann als PDF-Datei verschickt werden.

6.6 Übersicht der Messungen

Öffnet der Benutzer diese Seite das erste Mal nachdem er eine Messung durchgeführt hat, bekommt er einen Dialog angezeigt, der ihm textuell und durch ein zusätzliches Video den Umgang mit dem Graphen erklärt. Anschließend werden ihm seine Messungen, wie in Abbildung 6.18, dargestellt. Werden die Datenpunkte im Graphen angeklickt, so

erhält der Benutzer weitere Informationen zu der Messung. Diese weiteren Informationen bestehen aus dem Datum und der Wertung der Messung, sowie der gemessenen Frequenz, Lautstärke und Balance. Öffnet der Nutzer diese Seite, ist standardmäßig die letzte Messung ausgewählt. Der Benutzer kann außerdem mit zwei Fingern in den Graphen zoomen und mit einer Wischgeste nach rechts oder links scrollen.

6.7 Lautstärke Einstellungen

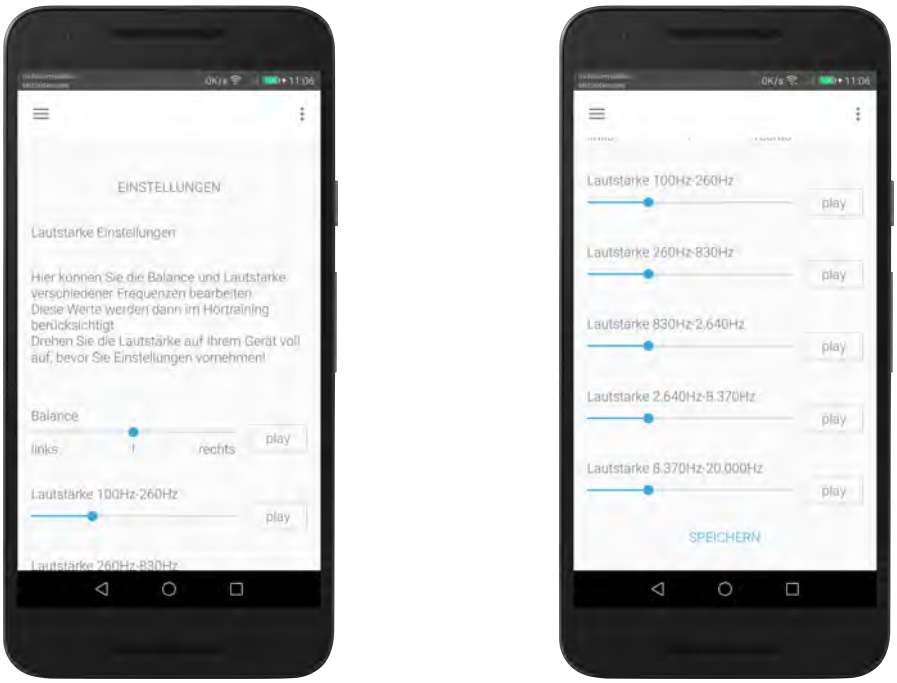

Abbildung 6.19: Lautstärke Einstellungen

In Abbildung 6.19 kann man die Einstellungsseite sehen. Der Benutzer hat hier die Möglichkeit die Lautstärke für verschiedene Frequenzbereiche individuell anzupassen und zu speichern. Diese Werte werden im Hörtraining berücksichtigt.

7

Anforderungsabgleich

In diesem Kapitel werden die einzelnen Punkte der Anforderungsanalyse aufgegriffen und beurteilt inwieweit sie in der Anwendung umgesetzt werden konnten.

7.1 Funktionale Anforderungen

In diesem Abschnitt wird überprüft, ob die Anwendung alle Anforderungen bezüglich ihrer Funktionalität dem Nutzer gegenüber erfüllt.

7.1.1 Allgemeine Funktionale Anforderungen

Alle funktionalen Anforderungen, die die allgemeine Benutzung betreffen und unabhängig von der Frequenzmessung und des Hörtrainings sind, werden nachfolgend überprüft.

Nr.	Titel und Beschreibung
1	**Nutzer müssen ihren Namen angeben, um die Anwendung das erste mal starten zu können** *Anforderung erfüllt.* Benutzer müssen ihren Namen beim erstmaligen Öffnen angeben (siehe Abschnitt 6.1).
2	**Die Sprache wird automatisch und abhängig von den Geräteeinstellungen angepasst** *Anforderung erfüllt.* Abhängig von den Geräteeinstellungen wird der Inhalt der Anwendung entweder auf deutsch oder auf englisch angezeigt (siehe Abschnitt 6.1).

3	**Töne werden nur dann ausgegeben, wenn Kopfhörer mit dem Gerät verbunden sind** *Anforderung erfüllt.* Töne werden ausschließlich dann abgespielt, wenn Kopfhörer mit dem Gerät verbunden sind. Werden diese entfernt während ein Ton gespielt wird, wird dieser Ton gestoppt und der Nutzer darüber informiert (siehe Abschnitt 6.1).

Tabelle 7.1: Abgleich allgemeiner funktionaler Anforderungen

7.1.2 Funktionale Anforderungen an die Frequenzmessung

Im Folgenden werden die Anforderungen abgeglichen, die die Frequenzmessung und damit die Messung an sich und die Übersicht aller Messungen betreffen.

Nr.	Titel und Beschreibung
4	**Der Ton wird bei der Messung, abhängig von der Position an dem der Nutzer seinen Tinnitus wahrnimmt, ausgegeben** *Anforderung erfüllt.* Der Nutzer muss wählen, ob er seinen Tinnitus 'links', 'rechts', 'links und rechts' oder 'im Kopf' hört. Abhängig davon wird der Ton bei der Auswahl von 'links' oder 'rechts' jeweils contralateral und bei den beiden anderen Wahlmöglichkeiten rechts abgespielt (siehe Abschnitt 6.4.2).
5	**Die Startfrequenz bei der Messung wird abhängig davon gewählt, in welchem Frequenzbereich der Nutzer seinen Tinnitus subjektiv wahrnimmt** *Anforderung erfüllt.* Der Nutzer hat die Wahlmöglichkeiten 'tief', 'mittel', 'hoch' und 'sehr hoch'. Abhängig davon wird der Startwert bei der Messung auf 1.000Hz, 2.000Hz, 4.000Hz oder 8.000Hz gesetzt (siehe Abschnitt 6.4.2).
6	**Der Benutzer kann zwischen einer groben und einer feinen Frequenzbestimmung wählen** *Anforderung erfüllt.* Dem Benutzer wird eine grobe und eine feine Frequenzbestimmung bereitgestellt, zwischen denen der Nutzer während der Messung hin und her springen kann (siehe Abschnitt 6.4.2).

7	**Die Messung deckt die Frequenzen von 50Hz bis 20.000Hz ab** *Anforderung erfüllt.* Tinnitusfrequenzen zwischen 50Hz und 20.000Hz können auf einen Halbton genau gemessen werden (siehe Abschnitt 6.4.2).
8	**Die Lautstärke und die Balance des gemessenen Tones kann an den Tinnitus angepasst werden** *Anforderung erfüllt.* Im Anschluss an die Messung hat der Nutzer die Möglichkeit die Balance und die Lautstärke des gemessenen Tones an die seines Tinnitus anzupassen (siehe Abschnitt 6.4.2).
9	**Der Nutzer kann die Messung bewerten** *Anforderung erfüllt.* Die Messung kann durch ein Rating vom Nutzer mit 0-5 Sternen bewertet werden (siehe Abschnitt 6.4.2).
10	**Dem Nutzer wird eine Übersicht über alle Messungen geboten** *Anforderung erfüllt.* Dem Nutzer werden auf der Seite *Übersicht Ihrer Messungen* alle Messungen angezeigt. Diese Anzeige geschieht über eine graphische und eine textuelle Darstellung. Eine Messung beinhaltet dabei das Datum und die Bewertung der Messung, sowie die gemessene Frequenz, Lautstärke und Balance (siehe Abschnitt 6.6).
11	**Der Benutzer hat die Möglichkeit seine Messungen zu teilen** *Anforderung erfüllt.* Die Messungen können vom Benutzer als PDF-Datei geteilt werden. Er kann dabei selbst entscheiden über welche Applikation die Datei geteilt werden soll (siehe Abschnitt 6.6).

Tabelle 7.2: Abgleich funktionaler Anforderungen an die Frequenzmessung

7.1.3 Funktionale Anforderungen an das Hörtraining

Alle Anforderungen, die im Rahmen des Hörtrainings gestellt wurden, werden im Nachfolgenden abgeglichen.

Nr.	Titel und Beschreibung
12	**Der Benutzer kann zwischen verschiedenen Levels wählen** *Anforderung erfüllt.* Dem Nutzer werden verschiedene Level mit steigendem Schwierigkeitsniveau angeboten, zwischen denen er wählen kann. Jedes Level kann dabei beliebig oft wiederholt werden (siehe Abschnitt 6.4.1).
13	**Neue Levels können freigeschaltet werden** *Anforderung erfüllt.* Zu Beginn des Trainings ist nur ein Level spielbar. Schließt der Benutzer ein Level erfolgreich ab, so wird ein neues Level freigeschaltet (siehe Abschnitt 6.4.1).
14	**Das Training beinhaltet mehreren Aufgabentypen** *Anforderung erfüllt.* Das Training beinhaltet mehrere unterschiedliche Aufgabentypen (siehe Abschnitt 6.4.1).
15	**Die Reihenfolge der Aufgabentypen ist zufällig** *Anforderung erfüllt.* Um dem Nutzer ein abwechslungsreiches Training zu bieten, sind die Aufgabentypen in jedem Training neu und zufällig angeordnet (siehe Abschnitt 6.4.1).
16	**Die Frequenzen der Töne im Hörtraining sind zufällig** *Anforderung erfüllt.* Die Frequenz eines Tones ist in jeder Aufgabe zufällig und die Frequenzen der anderen Töne in Abhängigkeit des Schwierigkeitsgrades gewählt (siehe Abschnitt 6.4.1 und Abschnitt 5.2).
17	**Dem Nutzer wird eine Übersicht über die Resultate seines Hörtrainings geboten** *Anforderung erfüllt.* Dem Nutzer wird eine Übersicht über die Resultate seines Hörtrainings angeboten. Diese Übersicht beinhaltet jeweils das Level, Datum und die erreichte Punktzahl des Trainings (siehe Abschnitt 6.5).
18	**Der Benutzer hat die Möglichkeit die Resultate seines Trainings zu teilen** *Anforderung erfüllt.* Die Resultate seines Trainings können vom Benutzer als PDF-Datei geteilt werden (siehe Abschnitt 6.5).

19	**Die Lautstärke verschiedener Frequenzintervalle und die Balance der Töne im Training sind anpassbar** *Anforderung erfüllt.* Der Nutzer hat die Möglichkeit die Lautstärke einzelner Frequenzintervalle und die Balance, der im Training abgespielten Töne, einzustellen (siehe Abschnitt 6.7).

Tabelle 7.3: Abgleich funktionaler Anforderungen an das Hörtraining

7.2 Nichtfunktionale Anforderungen

Nr.	Titel und Beschreibung
1	**Technische Anforderungen** Die Anwendung ist ab Android 5.0 (API Level 21) lauffähig.
2	**Bedienbarkeit** Die Anwendung ist in der Bedienung und der Sprache sehr simpel gehalten und es wurde darauf geachtet, dass dem Benutzer immer nur eine geringe Menge an Inhalt bereitgestellt wird.
3	**Erweiterbarkeit** Die Anzahl und der Aufbau der Level kann durch wenige Zeilen Code angepasst werden.
4	**Aussehen** Bei der Implementierung der Anwendung wurde sehr darauf geachtet, dass Design Prinzipien eingehalten werden und dass Design Elemente vor allem zur besseren Bedienbarkeit eingesetzt werden.

Tabelle 7.4: Abgleich nichtfunktionaler Anforderungen

8

Fazit

In diesem abschließenden Kapitel wird eine Zusammenfassung der Arbeit gegeben und ein Ausblick auf mögliche Verbesserungen und Erweiterungen geschildert.

8.1 Zusammenfassung

Ziel dieser Arbeit war die Konzeption und prototypische Realisierung einer mobilen Android Anwendung zur Bestimmung der Tinnitusfrequenz.

Dazu wurden zunächst einige für die Umsetzung relevante Hintergrundinformationen gegeben und Anforderungen an die Anwendung definiert und beschrieben. Anhand der erarbeiteten Anforderungen wurde im Anschluss darauf ein geeignetes Architekturkonzept bestimmt. Des Weiteren wurden einige ausgewählte Implementierungsdetails erläutert und die daraus resultierenden Funktionalitäten aus Sicht des Benutzers beschrieben, welche auch als eine Art Benutzerdokumentation fungieren können. Als Abschluss wurde ein Anforderungsabgleich vorgenommen und so Soll- und Ist-Zustand der Applikation verglichen.

8.2 Ausblick

Der Prototyp wurde auf verschiedenen Endgeräten getestet und könnte im nächsten Schritt anhand von Nutzerstudien evaluiert werden.

Mögliche Erweiterungen könnten in der Einheit des Hörtrainings vorgenommen werden. So wären weitere ergänzende Aufgabentypen und eine Erhöhung der Level-Anzahl vorstellbar.

Außerdem könnten dem Patienten in der Anwendung weitere Informationen zum Thema Tinnitus dargestellt und seine Trainingsresultate und Frequenzbestimmungen in einen erweiterten Kontext gebracht werden. Ein Beispiel dafür wäre zum Beispiel die Tinnitusfrequenz des Patienten in Relation zur durchschnittlichen prozentualen Verteilung der empfundenen Frequenz zu zeigen.

Eine weitere denkbare Änderung könnte sein, die vorliegende Arbeit in die bereits vorhandene TrackYourTinnitus Applikation einzubinden. Im Rahmen dieser Verschmelzung wäre eine Verknüpfung mit einem serverseitigem Backend, das über eine REST-API definierte Operationen zulässt, möglich. Im Backend könnten damit Messungen der Patienten hinterlegt und im nächsten Schritt verarbeitet werden. Dadurch könnten Aussagen darüber getroffen werden, wie sich die gemessenen Tinnitusfrequenzen in Abhängigkeit des Trainingsstandes ändern und, falls man zusätzlich die Dauer einer Messung berechnet, wie sich diese dazu verhält. Veränderungen der Tinnitusfrequenz könnten außerdem in einem Zusammenhang mit Gegebenheiten im persönlichen Umfeld des Patienten betrachtet werden.

Literaturverzeichnis

[1] McMillan, G.P., Thielman, E.J., Wypych, K., Henry, J.A.: A Bayesian Perspective on Tinnitus Pitch Matching. Ear and hearing **35** (2014) 687

[2] Okamoto, H., Stracke, H., Stoll, W., Pantev, C.: Listening to tailor-made notched music reduces tinnitus loudness and tinnitus-related auditory cortex activity. Proceedings of the National Academy of Sciences **107** (2010) 1207–1210

[3] European Cooperation in Science & Technology (COST): About COST. `http://www.cost.eu/about_cost` (2017) Abruf: 03.07.2018.

[4] University Hospital Regensburg: TINNET. `https://tinnet.tinnitusresearch.net/` (2018) Abruf: 02.07.2018.

[5] Serquera, J., Schlee, W., Pryss, R., Neff, P., Langguth, B.: Music Technology for Tinnitus Treatment within Tinnet. In: Audio Engineering Society Conference: 58th International Conference: Music Induced Hearing Disorders. (2015)

[6] Ulm, U.: Track your Tinnitus. `https://www.trackyourtinnitus.org/` (2013) Abruf: 03.07.2018.

[7] Pryss, R., Schlee, W., Langguth, B., Reichert, M.: Mobile crowdsensing services for tinnitus assessment and patient feedback. In: 6th IEEE International Conference on AI & Mobile Services (IEEE AIMS 2017), IEEE Computer Society Press (2017)

[8] Probst, T., Pryss, R., Langguth, B., Rauschecker, J., Schobel, J., Reichert, M., Spiliopoulou, M., Schlee, W., Zimmermann, J.: Does tinnitus depend on time-of-day? an ecological momentary assessment study with the trackyourtinnitus application. Frontiers in Aging Neuroscience **9** (2017) 253–253

[9] Probst, T., Pryss, R., Langguth, B., Schlee, W.: Emotional states as mediators between tinnitus loudness and tinnitus distress in daily life: Results from the trackyourtinnitus application. Scientific Reports **6** (2016)

[10] Pryss, R., Probst, T., Schlee, W., Schobel, J., Langguth, B., Neff, P., Spiliopoulou, M., Reichert, M.: 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), IEEE Computer Society Press (2017)

[11] Henry, J.A., Meikle, M.B., et al.: Psychoacoustic Measures of Tinnitus. Journal of the American Academy of Audiology **11** (2000) 138–155

[12] Tyler, R.S., Conrad-Armes, D.: Tinnitus pitch: A comparison of three measurement methods. British Journal of Audiology **17** (1983) 101–107 PMID: 6626779.

[13] The MIDI Manufacturers Association: The Complete MIDI 1.0 Detailed Specification. `https://www.midi.org/specifications/item/the-midi-1-0-specification` (1996) Abruf: 24.06.2018.

[14] Wolfe, J.: Note Names, MIDI Numbers and Frequencies. `https://newt.phys.unsw.edu.au/jw/notes.html` (2005) Abruf: 24.06.2018.

[15] Google Inc.: App Fundamentals. `https://developer.android.com/guide/components/fundamentals` (2018) Abruf: 24.06.2018.

[16] Google Inc.: AppCompatActivity. `https://developer.android.com/reference/android/support/v7/app/AppCompatActivity` (2018) Abruf: 03.07.2018.

[17] Google Inc.: Fragment. `https://developer.android.com/reference/android/support/v4/app/Fragment` (2018) Abruf: 25.06.2018.

[18] Google Inc.: DialogFragment. `https://developer.android.com/reference/android/support/v4/app/DialogFragment` (2018) Abruf: 25.06.2018.

[19] Google Inc.: BaseExpandableListAdapter. `https://developer.android.com/reference/android/widget/BaseExpandableListAdapter` (2018) Abruf: 25.06.2018.

[20] Google Inc.: AsyncTask. https://developer.android.com/reference/android/os/AsyncTask (2018) Abruf: 25.06.2018.

[21] Google Inc.: Application. https://developer.android.com/reference/android/app/Application (2018) Abruf: 25.06.2018.

[22] Google Inc.: SQLiteOpenHelper. https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper (2018) Abruf: 25.06.2018.

[23] Brinkmann, P.: pd-for-android. https://github.com/libpd/pd-for-android (2018) Abruf: 25.06.2018.

[24] Open Source Initiative: The 3-Clause BSD License. https://opensource.org/licenses/BSD-3-Clause (n.d.) Abruf: 25.06.2018.

[25] Gehring, J.: GraphView. http://www.android-graphview.org/ (2018) Abruf: 25.06.2018.

[26] Apache Software Foundation: Apache v2 Licens. https://www.apache.org/licenses/LICENSE-2.0 (2004) Abruf: 25.06.2018.

[27] Davies, N.: SeekArc. https://github.com/neild001/SeekArc (2013) Abruf: 25.06.2018.

[28] Open Source Initiative: The MIT License. https://opensource.org/licenses/MIT (n.d.) Abruf: 25.06.2018.

[29] Institut für Elektronische Musik und Akustik - IEM: Pure Data. https://puredata.info/ (n.d.) Abruf: 02.07.2018.

[30] Holzer, D.: Oscillators. http://write.flossmanuals.net/pure-data/oscillators/ (2010) Abruf: 02.07.2018.

[31] Kreidler, J.: Programming Electronic Music in Pd. http://www.pd-tutorial.com/ (2013) Abruf: 02.07.2018.

[32] Google Inc.: SharedPreferences. https://developer.android.com/reference/android/content/SharedPreferences (2018) Abruf: 02.07.2018.

Abbildungsverzeichnis

Tabellenverzeichnis

Name: Annika Sophie Stampf Matrikelnummer: 879146

Erklärung

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

Ulm, den .....

Annika Sophie Stampf