

The present work was submitted to Lehr- und Forschungsgebiet Educational Technologies at DIPF

Entwicklung einer Applikation zum assoziativen Medien Lernen

Bachelor-Thesis

Presented by

Sebastian Rödling

6268671

First examiner: Prof. Dr. Hendrik Drachsler

Supervisor: Hr. Daniel Biedermann

Frankfurt, 14. Oktober 2019

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Zudem versichere ich, dass die von mir abgegebenen schriftlichen (gebundenen) Versionen der vorliegenden Arbeit mit der abgegebenen elektronischen Version auf einem Datenträger inhaltlich übereinstimmt.

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed.

Frankfurt, 14. Oktober 2019

Sebastian Rödling

Abstract

Ziel der vorliegenden Bachelorarbeit ist es, den Einfluss von zusätzlicher am Handgelenk wahrgenommener Vibration in Verbindung mit der visuellen Darstellung eines Lerninhaltes auf den Lernerfolg zu messen. Der Lernerfolg wird hierbei durch die Lerngeschwindigkeit sowie den Umfang der Wissenskonsolidierung über die Testreihe definiert. Zu diesem Zweck wurde eine Experimentalstudie zum *Assoziativen Lernen* durchgeführt. Für die Studie verwendeten 33 Probanden eine App, die für die vorliegende Arbeit entwickelt wurde. Im Mittel aller Studienergebnisse wurden sowohl für die Lerngeschwindigkeit als auch für die Wissenskonsolidierung bessere Werte erzielt, wenn die Probanden die Möglichkeit hatten, den Lerninhalt sowohl visuell als auch haptisch zu erfahren. Die festgestellten Unterschiede des Lernerfolges erreichten jedoch keine statistische Signifikanz. Die Abweichungen der Ergebnisse nach der Umsetzung der vorgeschlagenen Änderungen am Studiendesign sind abzuwarten. Die Bachelorarbeit ist vor allem für den Bildungsbereich interessant.

The present bachelor thesis aims to measure the influence of vibration perceived at the wrist in connection with the visual representation of learning content on the learning success. The learning success is defined by the learning speed and the extent of knowledge consolidation over the test series. For this purpose, an experimental study on *Associative Learning* was conducted. For the study, 33 test persons used an app, which was developed for the present work. On average of all study results better values were achieved for both learning speed and knowledge consolidation, if the test persons could experience the learning content both visually and haptically. However, the differences in learning outcomes did not reach statistical significance. The results of the deviations after the implementation of the proposed changes to the study design must be awaited. The Bachelor's thesis is particularly interesting for the education sector.

Inhaltsverzeichnis

1	Einführung	1
2	Grundlagen	3
2.1	Grundlagen Lernpsychologie	3
2.1.1	Assoziatives Lernen	3
2.1.2	Passives haptisches Lernen	7
2.1.3	Zusammenfassung	8
2.2	Grundlagen Applikations Entwicklung	9
2.2.1	Vorwort	9
2.2.2	Manifest.xml Datei	10
2.2.3	Activities in Android	11
2.2.4	Fragment	12
2.2.5	Intent-Filter und Intents	13
2.2.6	Broadcast Receiver	14
2.2.7	Services	14
2.2.8	Zusammenfassung	15
3	Studienkonzept	17
3.1	Studienablauf	17
3.2	Dokumentation der Studie	18
3.2.1	Phase 1 - 1. Testlauf	19
3.2.2	Phase 2 - 2. Testlauf	19
3.2.3	Phase 3 - 3. Testlauf	19
4	Entwicklung der Applikation	21
4.1	Vorwort	22
4.2	Mobile Applikation	25
4.2.1	LoginActivity und RegistrationActivity	26
4.2.2	MainActivity und Fragmente	27

4.3	Wear Applikation	34
4.4	Umgang mit der Applikation	35
5	Evaluation	37
5.1	Vorwort	37
5.2	Lerneinheiten	38
5.3	Testeinheiten	39
5.4	Herzfrequenzen	41
5.5	Auswertung Lerneinheiten	42
5.6	Auswertung Testergebnisse	45
5.7	Auswertung Herzfrequenzen	46
6	Diskussion und Fazit	49
	Tabellenverzeichnis	53
	Abbildungsverzeichnis	55
	Literatur	57
	Appendices	61
	Anhang 1: Datenbanktabellen	63
	Anhang 2: Layouts der Vibning Applikation	65
	Anhang 3: Evaluierung der Studie	67

1 Einführung

“Try to learn something about everything and everything about something.”

—Thomas Huxley

"Für den Großteil der Bevölkerung ist die Nutzung des Internets im beruflichen und privaten Alltag selbstverständlich geworden. Vor diesem Hintergrund sowie dem bildungspolitischen etablierten Paradigma des 'Lebenslangen Lernens' spielt der Einsatz computer- und netzbasierter Lernentechnologien -kurz E-Learning- in der Aus- und Weiterbildung eine zunehmende Rolle "[33]. Dieses Zitat und die stetig steigende Zahl der Endnutzergeräte zeigt deren Relevanz für den Bildungsbereich. Eine sich hieraus ergebende Frage ist, inwieweit man diese Technologien nutzen kann, um eine Förderung des Lernprozesses zu erzielen. Unter dem Begriff Förderung des Lernprozesses wird im Kontext dieser Arbeit sowohl eine Zeitersparnis beim aktiven Lernen oder aber auch eine bessere Konsolidierung der Lerninhalte in Verbindung mit verbesserten Wiedergabeleistungen verstanden. Die vorliegende Arbeit beschreibt die Umsetzung psychologischer Lernmodelle, wie die des Assoziativen Lernens in einer Applikation für Smartphone und Smartwatch. Anhand einer Studie wird der Forschungsfrage nachgegangen, inwiefern man mit Hilfe der Assoziation eines Lerninhaltes mit einer passenden, auf der Smartwatch generierten Vibration, den Lernerfolg beeinflussen kann. Smartwatches bieten den Vorteil, dass diese bereits konzipierte Sensoren besitzen, um Reize auszulösen oder Daten, wie die Herzfrequenz, zu erfassen. Für die durchgeführte Studie wurden Testpersonen in zwei Gruppen aufgeteilt und gebeten, Morsezeichen mit Hilfe der Applikation zu lernen. Beide Probandengruppen lernten denselben Inhalt unter gleichen Rahmenbedingungen. Während die Experimentalgruppe das Lernen von Morsezeichen sowohl mit Hilfe von Vibration, als auch mit visuellem Reiz durchführte, erhielt die Kontrollgruppe nur den visuellen Reiz. Einen Tag sowie drei Tage nach der Lerneinheit wurden die Probanden gebeten, sich einem Wissenstest zu unterziehen. Anhand der erfassten Lernzeiten und der Testergebnisse konnte anschließend auf die Forschungsfrage eingegangen werden. Eine weitere exploratorische Auswertung der aufgezeichneten Herzfrequenzen konnte ebenfalls durchgeführt werden.

In den kommenden Kapiteln werden zunächst einige Grundlagen zu Teilbereichen der Lernpsychologie und der Entwicklung einer Applikation für mobile Endgeräte geschaffen. Darunter fällt zum einen das Assoziative Lernen, welches Aufschluss über das Lernverhalten des Menschen gibt. Die dazu vorgestellte Studie erklärt anhand der Assoziation von Pseudowörtern mit Alltagsgegenständen exemplarisch, wie das Assoziative Lernen beim Menschen funktioniert. Des Weiteren wird das Passive haptische Lernen (PHL) näher betrachtet. Hier vorgestellte Studien liefern erste Anhaltspunkte für das Lernen von Morsezeichen mit Hilfe der Vibration einer Smartwatch und auditiven Reizen. Anders als beim PHL lernen Probanden der Studie dieser Arbeit aktiv und mit Hilfe der Wahrnehmung von visuellen oder visuohaptischen Reizen. Bevor die Umsetzung und Entwicklung der Applikation besprochen wird, erfolgt eine nähere Betrachtung des Studiendesign. Hierbei wird auch auf das Lernkonzept der Applikation namens Vibning eingegangen. In den letzten beiden Kapiteln wird eine Evaluierung der Studie und deren Ergebnisse durchgeführt. Angeknüpft an den Studienergebnissen werden im letzten Kapitel ein Fazit und die daraus abgeleiteten Verbesserungsvorschläge für das Studiendesign und die Applikation diskutiert.

2 Grundlagen

In diesem Kapitel werden dem Leser grundlegende Modelle, auf denen diese Arbeit aufbaut, vermittelt. Der erste Teil beschäftigt sich dabei mit dem lernpsychologischen Hintergrund und ausgewählten Lernformen, mit denen die Studie im Kapitel 3 arbeitet. Der zweite Teil behandelt Grundlagen zur Applikationsentwicklung für Android Geräte und bereitet somit Kapitel 4 vor, in dem es um die Umsetzung der Applikation 'Vibning' geht.

2.1 Grundlagen Lernpsychologie

'Es gibt drei grundlegende Arten von Lernprozessen' [26, p.316]. Nicht assoziatives Lernen, assoziatives Lernen und Beobachtungslernen. Im Folgenden werden wir uns von den drei Lernprozessen lediglich auf das assoziative Lernen beschränken. Der Grund hierfür ist, dass Beobachtungslernen den Erwerb oder die Veränderung eines Verhaltens erzielen will [26]. Nicht assoziatives Lernen versucht eine 'Veränderung in der Reaktion auf *einen* Reiz' [26, p.317] hervorzurufen. Diese beiden Lernformen passen nicht mit dem Ziel dieser Arbeit, zwei Reize miteinander zu verknüpfen, zusammen. Die Lernform des assoziativen Lernens wird im folgenden Unterkapitel beschrieben. Im zweiten Unterkapitel, *Passives haptisches Lernen 2.1.2*, wird eine Art des Lernens näher betrachtet, die in Verbindung mit dem Lernkonzept der Studie dieser Arbeit gebracht werden kann.

2.1.1 Assoziatives Lernen

Gazzaniga et al. beschreibt assoziatives Lernen als einen Lernprozess, welcher Reize aneinander oder mit einem bestimmten Verhalten koppelt (vgl. 'Assoziatives Lernen ist die Verbindung von Reizen oder von Reizen und Verhaltensweisen, die in räumlicher oder zeitlicher Nähe zueinander auftreten[...]. Assoziationen entwickeln sich der traditionellen Lerntheorie zufolge durch Konditionierung' [26, p.317]). Dabei werden Konditionierungen in das Paradigma der Klassischen Konditionierung und das Paradigma der Operanten Konditionierung geteilt [31]. Zu den Pionierarbeiten der klassischen Konditionierung gelten 'Der pavlowsche Hund' des russi-

schen Physiologen Ivan P. Pavlov und die klassische Konditionierung des Patellarsehnenreflex von Twitmyer (1905) [31][26]. Als Vorreiter der Operanten Konditionierung werden in der Literatur des Öfteren der Versuch 'Die Thorndikesche Katze' von Edward Lee Thorndike [31] oder die Versuche von Skinner mit Mäusen und der Skinner-Box [26] erwähnt. Bei der Operanten Konditionierung wird versucht eine neue Stimulus-Reaktion Verbindung zu erschaffen. 'Das operante Verhalten erfolgt *instrumentell*, da das Verhalten zu einer Veränderung bzw. Konsequenz in der Umwelt führt' [31]. So werden Mäuse beispielsweise darauf konditioniert bestimmte Mechanismen zu betätigen, um an Essen zu gelangen. Da diese Art der Konditionierung ebenfalls nicht mit dem Ziel dieser Arbeit einhergeht, werden wir auf dieses Paradigma nicht weiter eingehen. Bei der klassischen Konditionierung wird eine bereits bestehende Verbindung zwischen einem unkonditionierten Stimulus und einer unkonditionierten Reaktion verwendet, um einen neuen neutralen Stimulus zu einem Auslöser der Reaktion zu bilden [31]. In der Lernphase werden zunächst der unkonditionierte Stimulus (US) mit dem neutralen Stimulus (NS) zusammen dargeboten, dies löst die unkonditionierte Reaktion (UR) aus. Nach einigen Lerndurchgängen wird, so die Theorie, der neutrale Stimulus zum konditionierten Stimulus (CS) mit einer konditionierten Reaktion (CR). In der Abbildung 2.1 ist die schematische Darstellung der klassischen Konditionierung der Arbeit vom *pavlovschen Hund* zu sehen. Wie in der Abbildung dargestellt, wurde ein Hund darauf konditioniert bei der Wahrnehmung eines Tons, der mit Futter assoziiert wird, Speichel zu produzieren (konditionierte Reaktion), unabhängig davon ob der Ton (CS) mit oder ohne Futtergabe erfolgte. Das hier vorgestellte Lernprinzip gilt ebenfalls für das menschliche Lernen [31].

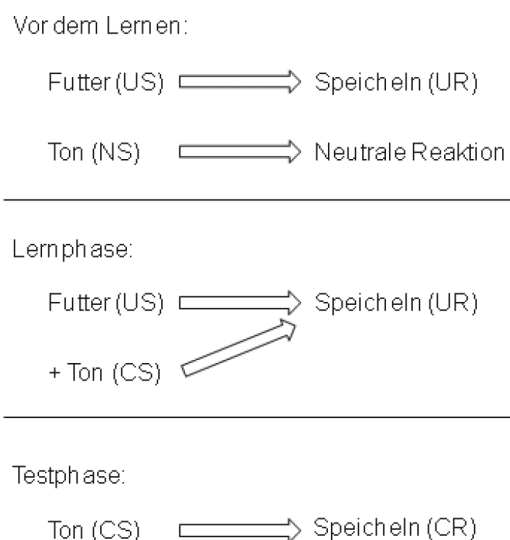


Abbildung 2.1: Schematische Darstellung der klassischen Konditionierung anhand des Beispiels 'der pavlovsche Hund' [31]

Weitergehende Beschreibungen des assoziativen Lernens in dieser Arbeit beziehen sich ausschließlich auf die Wirkung beim Menschen. In dem Zeitschriftartikel 'Lernen aus neurowissenschaftlicher Sicht: von der Assoziation zur Kognition' [21] von Caterina Breitenstein wird das assoziative Lernen anhand des Fremdsprachenerwerbs genauer beschrieben. Die in diesem Artikel aufgeführten Punkte, welche im Zusammenhang mit dieser Arbeit stehen, werden im Folgenden dargelegt. Zunächst definiert Breitenstein den Begriff *Assoziatives Lernen* als einen automatisierten Lernprozess von motorischen Bewegungsabfolgen, Faktenwissen und sozialen Konventionen. Im Gegensatz zu bewusstem Lernen würde unbewusstes Lernen die häufige Wiederholung von Reizkonstellationen in unterschiedlichen Situationen erfordern. Als Beispiel führt sie weiter an: 'Ein Beispiel für assoziatives Lernen *en passant* ist der Fremdsprachenerwerb in einem Land mit für den Besucher unbekannter Sprache. Initial werden die neuen Sprachlaute der Umgebung für den Besucher keinen Sinn ergeben; bei längerer Verweildauer erklingen bestimmte Lautäußerungen wiederholt in Zusammenhang mit bestimmten Alltagsobjekten oder Ereignissen.' [21]. Die Wiederholung helfe deshalb, weil bestimmte Laut-Ereignis-Kopplungen häufiger auftreten würden als andere, dabei würden diese Kopplungen gestärkt. Sobald das Lernen erfolgreich ist, würden feste Verbindungen zwischen Lauten und semantischen Konzepten bestehen. Der Funktionsweise des Assoziativen Lernens unterliege dabei einem statistischen System im menschlichen Gehirn [21]. 'Dabei werden aus der Flut eingehender Sinneserfahrungen probabilistische Regelmäßigkeiten ermittelt. Dies geschieht, ohne dass dem Lernenden das zugrunde liegende probabilistische Prinzip bewusst ist' [32].

Die folgende Studie unterlegt das probabilistische Prinzip des Menschen anhand eines Beispiels mittels sogenannter *Pseudowörter* und Alltagsgegenständen. Die Studie 'Development and validation of a language learning model for behavioral and functional-imaging studies' [22] beschreibt das Assoziative Lernen beim Menschen mittels des Fallbeispiels von Bildern von Alltagsgegenständen und auditiven Pseudowörtern. Die Vorgehensweise dieser Studie ist in Abbildung 2.1.1 dargestellt. Die Pseudowörter wurden durch 40 Studenten ausgewählt, zu dessen Kriterien gehörte, dass diese nur wenig bestehende Assoziationen haben sollten und von neutraler emotionaler Wertigkeit seien (bspw. 'enas'). Während der fünf zehnminütigen Lerneinheiten wurden den Probanden in 1500ms Abständen für eine Dauer von 1000ms Bild-Wort Paare gezeigt. Nach jeder dieser Phasen hatten die Probanden weitere 1000ms um zu entscheiden, ob die Bild-Wortpaarung korrekt oder inkorrekt ist. Das Lernkonzept bestand dabei darin, dass richtige Bild-Wort Paarungen von Lerneinheit zu Lerneinheit immer öfter, während inkorrekte Darbietungen immer gleich oft dargeboten wurden. Die erste Lerneinheit startete mit einem korrekt: inkorrekt Verhältnis von 4: 2 und endete in der letzten Lerneinheit mit einem Verhältnis von 20: 2. Das einzige Feedback, welches den Probanden gegeben wurde, war eine Prozentzahl an richtigen und falschen Antworten gegenüber den gesamten Antworten der zehnminütigen Lerneinheit. Im Anschluss an die, an verschiedenen Tagen stattgefundenen, Lerneinheiten wurde

die Assoziation zwischen den Pseudowörtern und den alltäglichen Namen der Alltagsobjekte erfolgreich geprüft. Weitere Überprüfungen nach einer Woche und einem Monat bestätigten die vorhandene Assoziation und dessen Stabilität. Diese Art des assoziativen Lernens wird in der Literatur als latentes *Stimulus-Stimulus-Konditionierung* beschrieben [21, 33].

Breitenstein beschreibt weiter, dass für den Lernerfolg nicht wichtig sei während der Aufgabendurchführung Rückmeldung zu erhalten [23]. Weiterhin seien die geringen kognitiven Anforderungen sowie die hohe Stabilität des Lernerfolgs die Stärken des assoziativen Lernens [23].

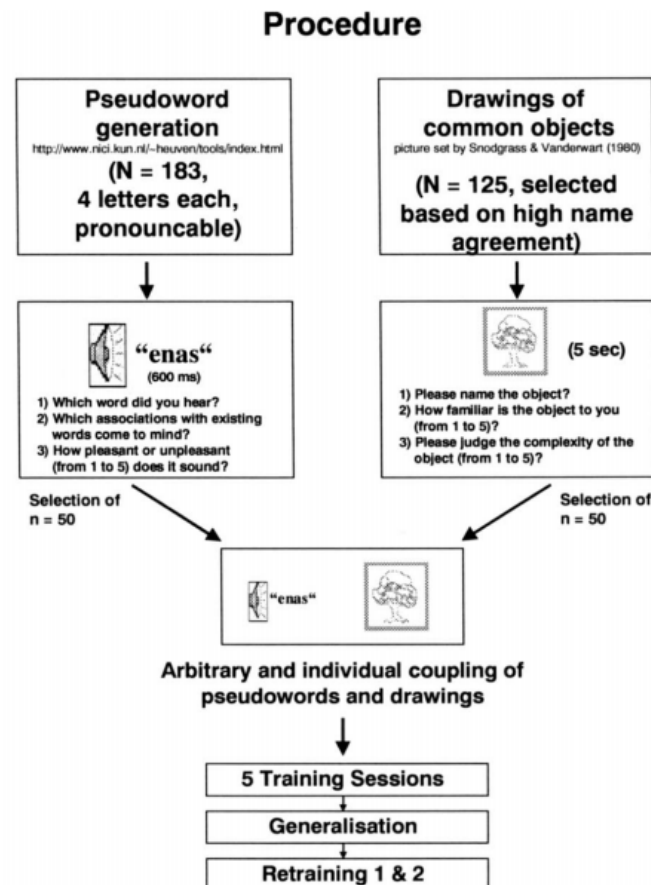


Fig. 1. Procedure of stimulus generation and selection as well as development of the word learning program.

Abbildung 2.2: Screenshot der Studienprozedur aus der Studie [22] zum Assoziativen Lernen

2.1.2 Passives haptisches Lernen

Dieses Unterkapitel befasst sich mit dem Thema des Passiven haptischen Lernen (PHL). Dabei wird zunächst eine Begriffserklärung gegeben und anschließend auf für diese Arbeit interessante Studien eingegangen.

Passives haptisches Lernen (PHL) beschreibt eine Lernform für sensomotorische Fähigkeiten, welche keine oder wenig aktive Aufmerksamkeit erfordert [49]. Das bedeutet, dass Lernende sich während des PHLs auf eine andere nicht themenbezogenen Aufgabe konzentrieren können und passiv den Lerninhalt vermittelt bekommen. Dabei wird anders als der ursprüngliche Gedanke von Vibration eines Endgeräts (das Signalisieren von Alarmen) [36], diese verwendet, um passive Stimulation bei den Probanden auszulösen, Feedback zu geben oder die Bewegung der Benutzer zu führen [50, 49]. Weiter schreibt Seim, 'Haptic Learning can be used as a tool for learning new skills [50]'. Verschiedene Studien haben gezeigt, dass der Anwendungsbereich für PHL sehr vielfältig sein kann. Beispielsweise wurden in den vergangenen Jahren Probanden angelernt, kleine Musiksequenzen auf dem Klavier zu spielen, über Vibration von der Haut Lesen zu lernen, oder das Morsezeichenalphabet zu erlernen und zu beherrschen [49, 28, 37, 43].

Huang und Kohlsdorf haben in ihrem Artikel 'Mobile music touch: mobile tactile stimulation for passive learning' [29] und 'Mobile Music Touch: The Effect of Primary Tasks on Passively Learning Piano Sequences' [34] gezeigt, dass passive Stimulation ausreicht, um kleine Musiksequenzen auf dem Klavier spielen zu können. Für diese Studie wurde ein mit Vibratoren versehener Handschuh angefertigt, um Stimulation beim Probanden zu erzeugen [29]. Dabei wurden Probanden angelernt die richtigen Tasten zu drücken, jedoch ohne einen vorgegebenen Rhythmus. Aufbauend auf den Studien von Huang hat Kohlsdorf et al. gezeigt, dass die Art der ablenkenden Aufgabe (Fokus der Konzentration) wenig bis gar keinen Einfluss auf das PHL hat [34]. So schreibt er, 'Contrary to expectations, all subjects were able to learn the note sequences in all conditions, and no condition seems significantly worse than any other as to its effect on passive haptic learning.' [34].

Für diese Arbeit von größerer Bedeutung sind die Studien von Seim et al. [50, 49], in denen als Lerninhalt Morsezeichen gewählt wurden. Mit diesen Studien wurde gezeigt, dass es mittels PHL möglich ist ein rhythmisch basiertes Textsystem [20] zu lernen. In der ersten Studie von Seim et al. [49], welche wir näher betrachten wollen, wurde mittels der Google Glass den Probanden Morsezeichen beigebracht. Die Google Glass ist eine von Google entwickelte Brille, welche Eingabe über ein Trackpad erfahren, Vibrationen auslösen und über eingebaute Lautsprecher auditive Signale ausgeben kann [49]. Während des passiven Lernens wurde

zur haptischen Stimulation auditive Stimulation dargeboten. Die Eingabe der Morsezeichen erfolgte über das Trackpad auf der Google Glass. Seim schreibt, dass bereits nach 4 Stunden des passiven Lernens Probanden der Experimentalgruppe eine Genauigkeit von 94% beim kodieren eines Pangramms¹ hatten, wohingegen die Kontrollgruppe, welche lediglich auditive Stimulation erfahren haben, nur 54% richtig hatten. Noch vor Abschluss der Studie hatten PHL Probanden eine Genauigkeit von 100% erreicht. In schriftlichen sowie Enkodierungsaufgaben wurden ähnliche Diskrepanzen zwischen Kontroll- und Experimentalgruppe gemessen [49].

Die zweite Studie von Seim et al. 'Towards Haptic Learning on a Smartwatch' [50] beschäftigt sich weitergehend mit PHL und verwendet statt der Google Glass eine Smartwatch. Eine Smartwatch hat in der Regel einen schwachen Vibrationsmechanismus, da dieser lediglich signalisieren soll, dass eine Nachricht empfangen wurde, ohne dabei viel Akku zu verbrauchen [50]. Mit dieser Studie haben sie gezeigt, dass die schwache Vibration von Smartwatches ausreicht, um passives haptisches Lernen zu fördern. Dazu wurden 6 Probanden passiv angelehrt, 10 Morsezeichen (T, H, E, L, A, Z, Y, D, O, G) in 8 oder 16 Minuten Konditionen zu lernen. Die Ergebnisse dieser Studie zeigten, dass zwischen dem Vorwissenstest und dem abschließenden Test deutliche Fortschritte seitens der Probanden gemacht wurden, 'All users showed significant improvement from pre- to post-test knowledge of Morse code' [50]. Im Anschluss an den abschließenden Test wurden Probanden gebeten am darauffolgenden Tag und den dritten Tag nach dem Test, erneut einen Wissenstest zu machen. Während der Zeiten zwischen den Tests wurden die Probanden angewiesen, sich nicht weiter mit Morsezeichen zu beschäftigen. Die Ergebnisse der beiden Tests stimmten ungefähr mit den Abschlusstests der Lerneinheit überein [50].

2.1.3 Zusammenfassung

Für die in Kapitel 3 beschriebene Studie können sowohl das Assoziative Lernen als auch das Passive haptische Lernen eine Rolle spielen. Aus diesem Grund werden hier einige Gemeinsamkeiten des Assoziativen Lernens und PHL zusammengetragen. Beiden 'Lernformen' wird eine gewisse Stabilität bezüglich der Konsolidierung des Lerninhaltes zugesprochen. Ebenso benötigen diese 'Lernformen' keine explizite aktive Aufmerksamkeit seitens der Lernenden. So beschrieb Breitenstein das *Assoziative Lernen en passant* anhand des Beispiels des Fremdsprachenerwerbs [21] und die Studien zum *Passiven haptischen Lernen* eine ablenkende Aufgabe, welche die Probanden während der Lerneinheiten ausführen mussten. Eine weitere Gemeinsamkeit ergibt sich daraus, dass in beiden Fällen die Häufigkeit der Paarungen wiederholt dargeboten wurden. Weiterhin wird in den Artikeln [49, 50] die Aussage getroffen, dass aktives

¹ Ein Pangramm ist definiert als ein Satz, welcher alle Buchstaben des Alphabets beinhaltet [49].

Lernen die gleichen Testergebnisse wie aus den Studien schneller produzieren könne als das passive Lernen. Des Weiteren zeigt die Studie [50], dass Menschen durch eine Smartwatch Stimulation zum passiven Lernen erfolgreich ein rhythmisch basiertes Textsystem (Morsezeichen) lernen können. Ebenso ermöglicht die Studie [50] einen ersten Vergleich zwischen einer Experimental- und Kontrollgruppe. Dieser bezieht sich zwar lediglich auf das passive Lernen, bietet aber für die Studie dieser Arbeit einen ersten Anhaltspunkt. Eine weitere Studie von Morrison zeigt, dass die visuohaptische Stimulation sowohl als auch der rein visuellen oder der rein haptischen Stimulation in Bezug auf Lernen überlegen ist [42]. Dabei ist jedoch zu beachten, dass der Begriff Haptik nicht ausschließlich nur auf Vibration zurückzuführen ist wie bei den angeführten Studien im Unterkapitel 2.1.2. Diese Gegebenheit wird sich später in der Studie dieser Arbeit wiederfinden, da eine Gruppe mit rein visuellen Reizen lernte, während die andere Teilnehmergruppe mit visuohaptischen Reizen lernen konnte.

2.2 Grundlagen Applikations Entwicklung

In diesem Kapitel wird auf notwendigen Grundlagen der Applikationsentwicklung eingegangen. Eine detaillierte Beschreibung und Ausarbeitung aller Komponenten einer Applikation ist im Rahmen einer Bachelorarbeit nicht möglich. Detailliertere Beschreibungen befinden sich auf der Android Developers Homepage [17] und im Buch 'Android-Apps entwickeln für Einsteiger' [45].

2.2.1 Vorwort

Für die Applikation '*Vibning*' wurde die Plattform Android gewählt. Android ist gegenüber anderen Plattformen weiter verbreitet [53] und bietet die Möglichkeit der Entwicklung auf allen größeren Betriebssystemen wie macOS, Linux und Windows. Des Weiteren standen für die Entwicklung bereits ein Huawei Honor 7X und die TicWatch C2 zur Verfügung. Mit Hilfe der Entwicklungsumgebung Android Studio 3.0 in der Version 3.4.2 [17] und der Programmiersprache Java [46] lässt sich unkompliziert eine kompilierbare und installierbare Applikation für ausgewählte Endgeräte erstellen. Dabei wird automatisch ein Verzeichnis erstellt, wie man es in Abbildung 2.3 sehen kann. Das auf Abbildung 2.3 automatisch erstellte Projekt wurde für die Programmiersprache Java [46] und die ausgewählten Endgeräte 'mobile' und 'wear' erstellt. Beide Verzeichnisse sind nahezu identisch aufgebaut. Wir beschäftigen uns weiterhin nur mit dem Verzeichnis 'mobile' und deren Komponenten näher.

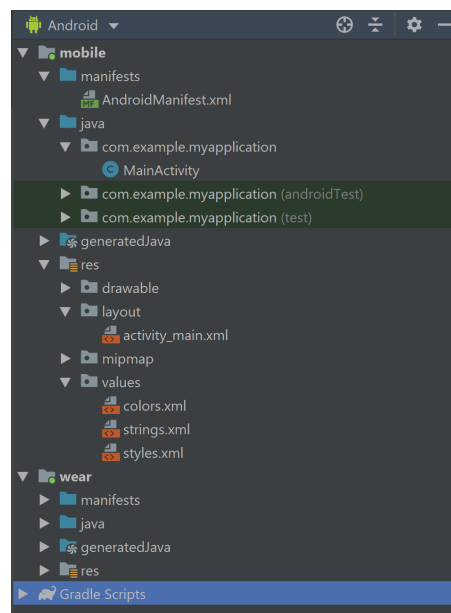


Abbildung 2.3: Von Android Studio automatisch erstellte Projekt

2.2.2 Manifest.xml Datei

"Das Android-Manifest ist die zentrale Beschreibung ihrer App [...] [45, p.99]. In dieser sind verschiedene Angaben über die App gemacht, die das Android-System benötigt, um die App installieren und ausführen zu können [19]".

Listing 2.1: Beispiel Manifest Datei

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3 package="com.example.myapplication">
4 <uses-permission android:name="android.permission.WAKE_LOCK" />
5 <application
6 android:allowBackup="true"
7 android:icon="@mipmap/ic_launcher"
8 android:label="@string/app_name"
9 android:roundIcon="@mipmap/ic_launcher_round"
10 android:supportsRtl="true"
11 android:theme="@style/AppTheme">
12 <activity android:name=".MainActivity">
13 <intent-filter>
14 <action android:name="android.intent.action.MAIN" />
15 <category android:name="android.intent.category.LAUNCHER" />
16 </intent-filter>

```

```
17 </activity>
18 </application>
19 </manifest>
```

Dem Listing 2.1 ist entnehmbar, dass eine Reihe von Eigenschaften der Applikation im Manifest festgehalten werden. Zunächst wird in Zeile 3 ein Paketname für die Applikation festgelegt. Im nächsten Abschnitt hat man die Möglichkeit dem Projekt verschiedene *'uses-permission'* einzutragen. Mittels diesen wird der Zugriff auf Hardwarekomponenten des Gerätes ermöglicht. Genauer zu *'uses-permission'* und eine Liste aller Sicherheitsberechtigungen sind unter dieser Referenz zu finden [16]. Für die Applikation *'Vibning'* sind beispielsweise zunächst Zugang zum *android.permission.INTERNET*, *android.permission.BLUETOOTH* und der *android.permission.VIBRATE* wichtig. Der Abschnitt *<application>* beschäftigt sich rein mit der Applikation, ohne Hardware Komponenten einzubeziehen. Hier werden weitere, sich selbst erklärende, Attribute der Applikation festgelegt [5]. Ein passendes Beispiel dazu ist *android:usesCleartextTraffic="true"*, welches der Applikation erlaubt unverschlüsselte HTTP-Anfragen zu bearbeiten. Dabei ist zu beachten, dass alle Werte welche mit @ anfangen Verlinkungen zu den Dateien im 'res' Ordner sind. Dieser ist in Figure 2.3 wiederzufinden. Von weiterem Interesse für die Applikation *'Vibning'* ist ebenfalls die *Theme* Wahl, welche das grundlegende Design der Applikation festlegt. Das bedeutet, dass alle eingesetzten Widgets auf dem User-Interface von diesem *Theme* erben und ihr Aussehen von ihm definiert werden. Hierfür findet man im Pfad *'res/values/style.xml'* ein *style* welches den Namen *'AppTheme'* trägt und von einem bestimmten *parent* erbt. In diesem Fall ist das *Theme.AppCompat.Light.DarkActionBar* [1]. Desweiteren hat man die Möglichkeit weitere Programmkomponenten anzumelden, wie beispielsweise *Activities*, *Intent-Filter*, *Services* oder *Receiver*. Diese werden in den kommenden Kapiteln genauer betrachtet.

2.2.3 Activities in Android

Activities werden von der Android Developer Homepage beschrieben als eine einzelne fokussierte Sache, die der Benutzer machen kann [8]. Sie dienen dazu, mit dem Benutzer zu interagieren und bieten die Möglichkeit ein Layout anzuzeigen [4]. Alle verwendeten Activities müssen in der Manifestdatei der Applikation angemeldet sein. Zudem kann man in der Manifestdatei festlegen, welche Activity beim Start der Applikation zuerst aufgerufen wird (Vgl. Listing 2.1 Zeile 16) [45, p.99-100]. Im Hinblick auf die erstellte Applikation spielt der Lebenszyklus einer Activity eine große Rolle (siehe Abbildung 2.4).

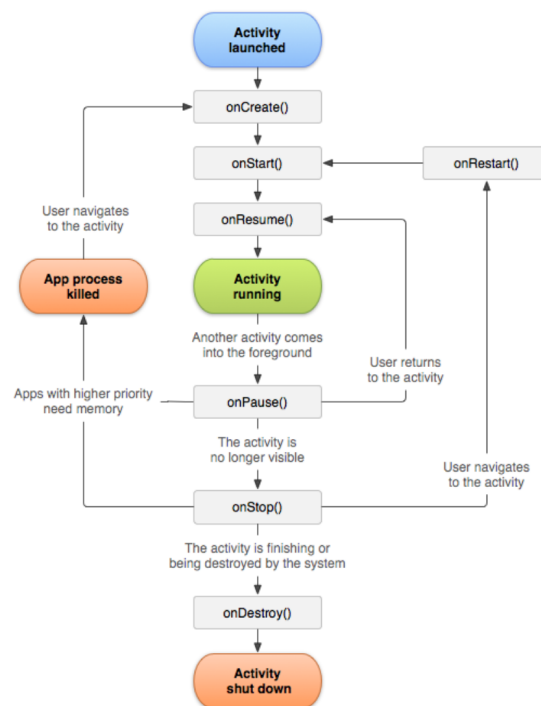


Abbildung 2.4: Lebenszyklus einer Android Activity [4]

Die Abbildung 2.4 zeigt, dass vier Hauptzustände einer Activity existieren. Eine Activity kann sich in einem der folgenden Stadien befinden, *Activity launched*, *App process killed*, *Activity shut down* oder *Activity running*. Beim Start einer Activity werden ausgehend von dem in Abbildung 2.4 blau gezeichneten Status alle Funktionen in der Reihenfolge der Pfeile ausgeführt. Das Layout der Activity wird in der Regel in der `onCreate()` Methode festgelegt. In der Applikation Vibning wird je nach Eingabe und Verhalten des Benutzers am Smartphone, die korrespondierende SmartWatch-Applikation in ein anderes Stadium gesetzt, um für einzelne Studienteildurchgänge keine Seiteneffekte zu riskieren. Beispiele für einzelne Activities können eine Login- oder eine Registration-Activity sein.

2.2.4 Fragment

Fragmente können dazu verwendet werden ein Verhalten oder einen Teil der Benutzeroberfläche einer Activity zu repräsentieren. Dabei ist es möglich, mehrere Fragmente in eine Activity einzubinden und diese auch wiederzuverwenden [7]. Ähnlich wie Activities haben Fragmente einen eigenen Lebenszyklus und sind ebenfalls eine Android Programmkomponente, welche mit dem Benutzer der Applikation interagieren kann [7]. Dabei ist der Lebenszyklus eines Fragments stets an den Lebenszyklus der aufrufenden Activity gebunden und verhält sich beim

Pausieren oder Zerstören der Activity dementsprechend gleich [7].

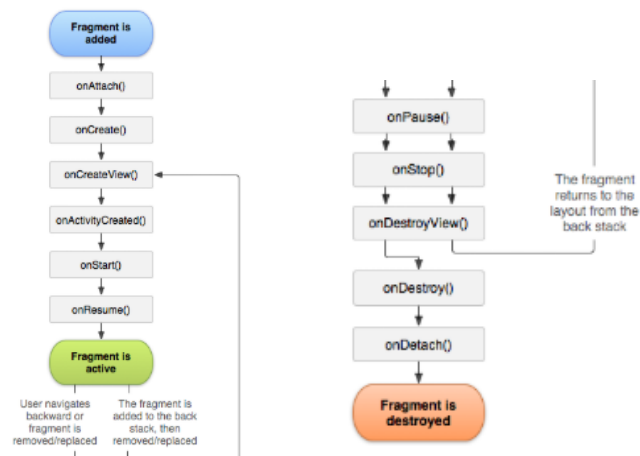


Abbildung 2.5: Lebenszyklus eines Android Fragments [7]

Die Abbildung 2.5 zeigt die Ähnlichkeit zwischen dem Lebenszyklus eines Fragments und einer Activity (vgl. Abbildung 2.4). Auf die einzelnen Verwendungen der in den Abbildungen sichtbaren Methoden wird genauer im Kapitel zur 'Applikations- Entwicklung' 4 eingegangen. Je nach Leistungsorientierung des Fragments kann man mit dem Lebenszyklus Daten speichern oder bei einem neuen Aufruf von der *onAttach()* Methode ein neues Objekt erstellen. Des Weiteren kann man Fragmente dazu verwenden, um einer Activity über ein Menü mehrere zusammenhängende Layouts zu geben. Beispielsweise mit Hilfe einer BottomAppBar [6] oder (wie in der Applikation 'Vibning') mit einem Navigation Drawer [11]. Unabhängig von der Vibning Applikation sind Fragmente ebenfalls das Mittel um für verschieden große Displays Teile der Activity nebeneinander oder einzeln darzustellen [7].

2.2.5 Intent-Filter und Intents

Intents (Absichten) werden in der Programmierung von Android verwendet, um zwischen einzelnen Programmkomponenten asynchrone Nachrichten zu senden, um beispielsweise Aktionen zu starten [18]. Sie können auch dazu verwendet werden, Activities zu starten, Absichten an Receiver zu senden oder mit Hintergrundprozessen zu kommunizieren [10]. Dabei wird zwischen den impliziten und expliziten Intents unterschieden. Explizite Intents benötigen eine Programmkomponente und beinhalten oft keine weiteren Informationen, sie bieten dem Nutzer nur die Möglichkeit zwischen verschiedenen Aktivitäten zu wechseln [10]. Implizite Intents benötigen mehrere Informationen, damit das System für das Intent geeignete Komponenten zur Verfügung stellen kann [10]. Ein gutes Beispiel für ein impliziten Intent wäre das Senden einer E-Mail in der das System verfügbare E-Mail Applikationen zur Verfügung stellt. Wie dem

Listing 2.1 aus Kapitel 2.2.2 entnehmbar werden bereits per Default Intents implementiert. So sorgt das Intent `'android.intent.action.MAIN'` dafür, dass beim Öffnen des Launchers der Applikation diese Activity aufgerufen wird [45, p.100]. Das Intent `android.intent.category.LAUNCHER` erstellt ein dazugehöriges Icon im Android Menü [45, p.100]. Intent-Filter werden von der Applikation dabei verwendet, um implizite Intents zu akzeptieren oder zu verwerfen [14]. Jeder Intent-Filter legt dabei fest welche Aktion, Daten und Kategorien er annimmt (Vgl Listing 2.1 im Kapitel 2.2.2) und bekommt auch nur solche vom System geliefert [14].

2.2.6 Broadcast Receiver

Broadcast Receiver können eine weitere Programmkomponente einer Applikation sein. Sie geben der Applikation die Möglichkeit Intents zu empfangen und zu verarbeiten. Dabei ist es nicht notwendig, dass die Activity gerade den Stadium *Activity running* hat [15]. Folgt man der Android Developers Homepage kann man Receiver entweder in der Manifestdatei festlegen oder programmatisch im Code selber [15]. Die Applikation *'Vibning'* verwendet Receiver in beiden Formen. Beispielsweise werden Receiver eingesetzt um zwischen *'mobile'* und *'wear'* Informationen zu senden und zu empfangen. Weitere Receiver werden lokal verwendet, um die empfangenen Informationen an die gerade aktive Activity zu senden und den Applikationsprozess weiter zu führen.

2.2.7 Services

Services sind die letzte Programmkomponente, die in diesem Grundlagenkapitel betrachtet wird. Services sind Programmkomponente, die im Hintergrund der Applikation laufen und keine Benutzeroberfläche bieten [9]. Der Lebenszyklus eines Service ist nicht vollständig abhängig von der laufenden Activity. Services werden gestartet und gestoppt [9]. Ein Service kann Netzwerktransaktionen abhandeln, ohne den laufenden Prozess der Activity zu unterbrechen [9]. Die Applikation *Vibning* verwendet beispielsweise jeweils für das *'mobile'* und das *'wear'* einen Service, um Nachrichten zwischen den beiden Geräten auszutauschen. Ein weiterer Service wird verwendet, um über die *'wear'* Herzfrequenzen zu messen und Vibrationen auszulösen. Ein Beispiel zu deren Implementation ist im Kapitel *'Applikations- Entwicklung'* beschrieben.

2.2.8 Zusammenfassung

Dieses Kapitel fasst die wichtigen Punkte zu einigen grundlegenden Komponenten der Applikationsentwicklung zusammen. Die Manifestdatei dient als Beschreibung von sämtlichen größeren Komponenten der Applikation (wie die einer Activity oder eines Service). Die Zugriffsberechtigungen zu Hardwarekomponenten des Gerätes werden ebenfalls hier eingetragen. Einige Berechtigungen lassen sich ohne manuelle Zustimmung des Benutzers eintragen, für andere Berechtigungen muss der Benutzer diese erst bestätigen. Activities bieten eine Benutzeroberfläche, mit der ein Benutzer der Applikation interagieren kann. Dabei kann die Benutzeroberfläche über eine Activity durch verschiedene Fragmente in vielen möglichen Konstellationen unterteilt sein. Intents, Receiver und Services können dazu verwendet werden, um zwischen verschiedenen internen Stellen einer Applikation oder nicht intern mit anderen Applikationen zu kommunizieren.

3 Studienkonzept

Im Rahmen dieser Bachelorarbeit wurde eine Studie zum Thema assoziativen Lernen mit einem Handy und einer Smartwatch durchgeführt. Das geplante Vorgehen dieser Studie, sowie das theoretische Lernkonzept und die daraus resultierenden notwendigen Mittel werden in diesem Kapitel besprochen. Änderungen, welche im Verlauf der Studie unternommen wurden, sind hier ebenfalls festgehalten worden. Im Unterkapitel *Dokumentation der Studie* 3.2 wird näher auf Probleme und Korrekturen der Studie eingegangen. Eine Evaluierung und Fazit der Studienergebnisse befinden sich in den Kapiteln 5 und 6. Ziel der Studie ist es herauszufinden, ob zu jedem Lerninhalt verschieden generierte und ausgelöste Vibrationsmuster am Handgelenk den Lernenden dabei unterstützen kann, seine Lernleistung zu verbessern. Darunter versteht diese Arbeit zum einen eine kürzere Lernzeit oder eine bessere Konsolidierung der Lerninhalte an sich. Eine graphische Darstellung des Studienverlaufs befindet sich in der Abbildung 3.1. Dabei steht die Abkürzung EG für Experimentalgruppe. Die Abkürzung KG für Kontrollgruppe und LE für Lerneinheit.

3.1 Studienablauf

Das Studiendesign dieser Arbeit gleicht dem einer experimentellen Studie. Dazu wurden alle teilnehmenden Probanden in zwei Gruppen aufgeteilt. Die erste Gruppe, die Experimentalgruppe, lernte anhand eindeutiger Vibrationsmuster zu jedem Lerninhalt zu dem visuell wahrnehmbaren Reiz. Die zweite Gruppe, die Kontrollgruppe, lernte die gleichen Lerninhalte in der gleichen Test- und Lernreihenfolge ohne Vibration. Auswahlkriterium für die Probanden war ein Alter zwischen 15 und 30 Jahre und der Status Schüler oder Student. Das Alter und die aktuelle Beschäftigung der zutreffenden Person können eine Rolle bezüglich deren derzeitigen kognitiven Bereitschaft spielen [35, 26]. Beide Gruppen sollten bezüglich Alter und Beschäftigung vergleichbar sein. Um einen Lerneffekt messen zu können wurde zunächst ein Einstiegstest (Vorwissenstest) durchgeführt. Die Anzahl der abgefragten Morsezeichen wurde entsprechend der Studie [50] zunächst auf dieselben 10 Morsezeichen festgelegt. Über die drei, in Kapitel 3.2, beschriebenen Testdurchläufe wurde die Anzahl von 10 auf 20 bis hin zu 26 Morsezeichen als Lerninhalt erhöht. Die Studienteilnehmer sollten dabei eigenständig lernen und wurden nicht

speziell angewiesen auf die Vibration zu achten (Vgl Kapitel 2.1.3). Die Lerneinheiten, welche im direkten Anschluss an den Vorwissenstest stattfanden, bestanden jeweils aus einer Lern- und einer Testsequenz. In der Lernsequenz erfolgte eine einmalige Darbietung des Lerninhalts und den dazugehörigen lateinischen Buchstaben. In der Testsequenz wurde der Lerninhalt gezeigt und in einem Multiple Choice Test mit zunächst 4 Antwortmöglichkeiten abgefragt. Die Anzahl der Antwortmöglichkeiten wurde ebenfalls aufgrund der in Kapitel 3.2 beschriebenen Gegebenheiten auf 6 erhöht. Die Lerneinheiten wurden so lange wiederholt, bis der Proband in einer Testsequenz alle Paarungen richtig erkannte oder das Lernen über eine Stunde hinaus ging. Probanden stand es frei, die Studie frühzeitig zu verlassen oder mit dem Lernen vor dem Erreichen aller korrekten Paarung aufzuhören. Ausgangspunkt für die zwei folgenden Tests, einen und drei Tage nach dem Lernen, war das Ergebnis der letzten Testsequenz. Der erste Test war äquivalent zum Vorwissenstest und den Testsequenzen der Lerneinheiten aufgebaut. Für den zweiten Test wurden die Antwortmöglichkeiten auf 27 erhöht (Vgl. Kapitel 3.2). Dabei wurden die Probanden angewiesen nicht zu raten und wenn nötig eine Multiple Choice Frage zu überspringen. Im Anschluss an jede Testsequenz sollte zunächst Feedback über die korrekten und inkorrekten Antworten gegeben werden. Aufgrund der Probedurchläufe wurde dies auf eine prozentuale Anzahl an korrekten Antworten reduziert. Die Reihenfolge der abgefragten Morsezeichen sowie die Zusammenstellung der Multiple Choice Antworten wurden dabei zufällig generiert, um zu verhindern, dass Reihen und Folgen auswendig gelernt wurden.

Ausgehend vom Kapitel zum Assoziativen Lernen 2.1.1 und Kapitel 2.1.2 ist das Lernkonzept ähnlich zu den Studien [22, 50] aufgebaut. Das Lernkonzept für diese Studie sieht so aus, dass die Kontrollgruppe die visuelle Präsentation des Morsezeichens mit dem lateinischen Buchstaben assoziieren kann. Die Experimentalgruppe kann die visuelle Präsentation zuzüglich der haptischen Wahrnehmung mit dem lateinischen Buchstaben assoziieren. Probanden lernen anders als bei diesen Studien aktiv. Die nötigen Wiederholungen zum Lernen werden durch die einzelnen Lerneinheiten ermöglicht. Dabei ist den Probanden nicht vorgegeben, wie lange sie welches Morsezeichen wiederholen. Die Studie wird unter der Hypothese durchgeführt, dass Personen welche zusätzlich haptische Reize erfahren besser abschneiden als Probanden, die sich auf die visuellen Reize beschränken müssen. Der haptische Reiz kann hierbei als Hinweisreiz oder als zusätzlich konditionierter Stimulus fungieren.

3.2 Dokumentation der Studie

Dieses Kapitel befasst sich mit der Dokumentation der Studie. Veränderungen am Studiendesign und damit verbundene Änderung an der Applikation sind hier dokumentiert.

3.2.1 Phase 1 - 1. Testlauf

Bei dem ersten Testlauf der Studie wurde festgestellt, dass 10 Morsezeichen auf 4 Antwortmöglichkeiten für das Lernen nicht angemessen sind. Die gemessene Lernzeit war zu gering, um tatsächliche Unterschiede zwischen den beiden Kontrollgruppen sammeln zu können. Der erste Proband hatte die Lerneinheit in ungefähr 10 Minuten abgeschlossen und berichtet, dass bei 4 Antwortmöglichkeiten auf 10 Morsezeichen das Ausschlussverfahren bei den Multiple Choice Tests sehr gut funktioniert. Zudem würde das Feedback am Ende jeder Lerneinheit das Lernen weiter vereinfachen. Diese Kritikpunkte würden ungeachtet dessen aufkommen, dass Reihenfolgen und Antwortmöglichkeiten zufällig generiert werden. In den beiden anschließenden Tests hatte der gleiche Proband alle Antworten richtig ausgewählt. Die Studie und die Applikation wurden für die kommenden Tests dahingehend verändert, dass doppelt so viele Morsezeichen gelernt werden sollten. Dies sollte zunächst die Wahrscheinlichkeit des öfteren Auftretens der gleichen lateinischen Buchstaben in den Antwortmöglichkeiten über die gesamte Lerneinheit verringern.

3.2.2 Phase 2 - 2. Testlauf

Für den zweiten Testlauf wurde die Anzahl der Morsezeichen auf 20 erhöht. Die Anzahl der Antwortmöglichkeiten wurde gleich gelassen und das Feedback am Ende jeder Lerneinheit entfernt. Das einzige Feedback, das die Probanden noch bekamen, war eine prozentuale Angabe zu deren richtigen Antworten im Vergleich zu den gesamten Antworten. Eine deutliche Veränderung der Anzahl der benötigten Lerneinheiten oder der konsumierten Lernzeit war nicht festzustellen.

3.2.3 Phase 3 - 3. Testlauf

Für den dritten Testlauf wurde die Anzahl der Morsezeichen auf 26 erhöht, was dem gesamten lateinischen Alphabet entspricht. Die Antwortmöglichkeiten für den Vorwissenstest, die Testsequenz und den ersten Test wurden auf 6 erhöht. Dies erschwert erneut das im Kapitel 3.2.1 beschriebene Ausschlussverfahren. Will der Proband nun durch ein Ausschlussverfahren die korrekte Antwort ermitteln, muss er vier oder fünf der Antwortmöglichkeiten kennen. Da diese von Lerninhalt zu Lerninhalt (Morsezeichen) unterschiedlich sind, muss der Proband, um auf alle richtigen Antworten zu kommen, ein breites Wissen über die Codierung von Morsezeichen

abrufen können. Um zum Schluss zu validieren, dass die Probanden tatsächlich alle Morsezeichen gelernt hatten, wurde der letzte Test, welcher drei Tage nach dem Lernen stattfand, so verändert, dass nun alle 26 lateinischen Buchstaben als Antwortmöglichkeiten gegeben sind.

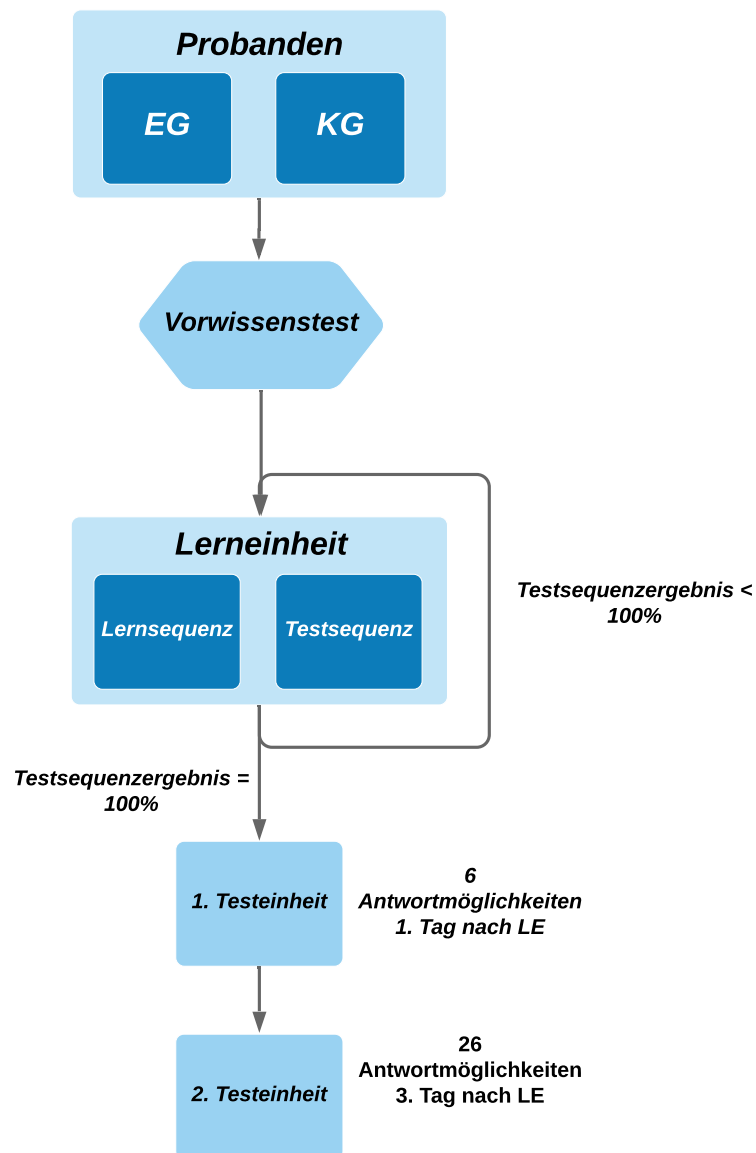


Abbildung 3.1: Schematische Darstellung des Studienablaufs.

4 Entwicklung der Applikation

“Any fool can write code that a computer can understand. Good programmers write code that humans can understand.”

—Martin Fowler

Dieses Kapitel bietet einen Einblick in die Entwicklung der Applikation 'Vibning'. Die Architektur der Applikation und die Intentionen der wichtigsten Programmkomponenten werden hier besprochen. Näher betrachtet werden Programmkomponenten, die für die im Rahmen dieser Bachelorarbeit durchgeführten Studie besonders wichtig, sowie Komponenten der Applikation die für eine Abänderung derselben für künftige Arbeiten von Bedeutung sind. Um dem Kapitel eine Struktur zu geben, werden zunächst im Vorwort Programmkonventionen die das Verstehen des Programmcodes erleichtern und die für die Datensammlung erstellte Datenbank betrachtet. Anschließend werden die Komponenten der Applikation in der Reihenfolge erklärt, wie sie der Benutzer erfährt. Dies entspricht in etwa der Reihenfolge, in der die Applikation entwickelt wurde. Die Applikation 'Vibning' besteht zunächst aus zwei Teilen. Der erste Teil ist für die 'mobile' Version zuständig und der zweite für die 'wear' Version. Beide Programmteile kommunizieren über eine Datenschicht, welche jeweils von beiden Seiten beschrieben und gelesen werden kann. Diese Programmschnittstelle wird im Kapitel zum 'mobile' Programmteil genauer beschrieben. Ausgangspunkt der gesamten Applikationsentwicklung ist die im Grundlagenkapitel 2 beschriebene Struktur des Projektes und deren Komponenten. Das fertige Projektverzeichnis für die *mobile* Applikation und seine Komponenten ist in Abbildung 4.1 dargestellt. Die Grafik dient zunächst dazu dem Leser einen Überblick über das gesamte Projekt zu geben. Weitere Beschreibungen zu den Funktionalitäten einzelner Klassen folgen in diesem Kapitel.

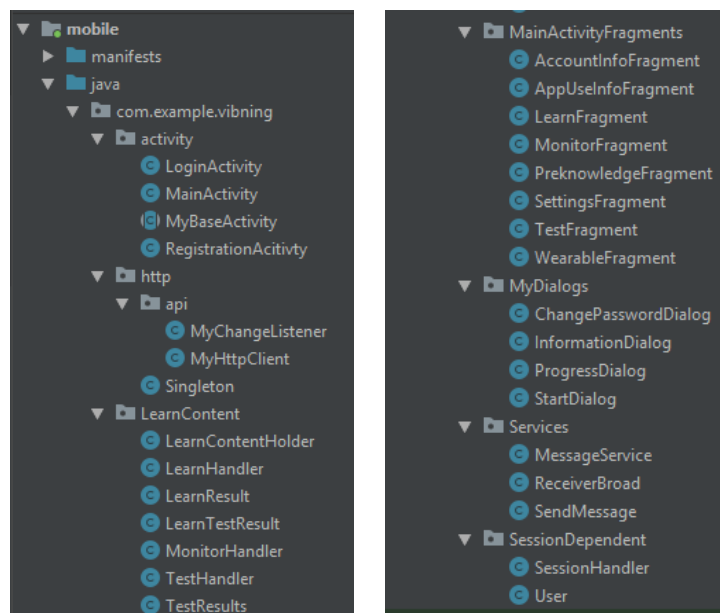


Abbildung 4.1: Fertige Projektübersicht der Vibning Applikation

4.1 Vorwort

Um die Dokumentation zur Entwicklung der Applikation so einfach wie möglich zu halten werden zunächst der Datenbankaufbau und einzelne Programmkonventionen erklärt. Die *'mobile'* Applikation kommuniziert mit einer auf einem Online-Speicher hinterlegten Datenbank, welche sowohl benutzerspezifische Daten als auch Daten der Test- und Lerndurchgänge speichert. Im digitalen Anhang befinden sich die dafür notwendigen PHP Skripte und die einzelnen Tabellen der Datenbank als CSV-Dateien. In der Datenbank werden insgesamt 4 Tabellen gehalten. Die erste im *Benutzerflow* angesprochene Tabelle ist die Tabelle *UserData* (Vgl. Anhang 1). Sie speichert für jeden registrierten Benutzer Vornamen, Namen, UserID, Zeitpunkt der Registrierung, Geburtstag, Teilnahmebedingung, Login, Passwort und Teilnehmerlevel. Die UserID wird zunächst als PrimaryKey der Datenbank festgelegt und dient als Bindeglied zwischen den einzelnen Tabellen. Als Teilnahmebedingungen existieren *with Vibration* und *without Vibration*, welche die Kontrollgruppe von der Experimentalgruppe unterscheidet. Das Teilnehmerlevel beschreibt den aktuellen Status des Probanden der Studie. Insgesamt existieren 5 verschiedene Teilnehmerlevel, welche den Zugang zu verschiedenen Fragmenten der *'mobile'* Applikation ermöglichen und für einen geregelten *Flow* sorgen. Der vorgesehene *Applikationsbenutzungsflow* ergibt sich aus folgenden Level.

- Level 0 - Benutzer ist registriert
- Level 1 - Benutzer hat den Vorwissenstest absolviert

- Level 2 - Benutzer hat die Lerneinheiten absolviert
- Level 3 - Benutzer hat den ersten Test absolviert
- Level 4 - Benutzer hat den zweiten Test absolviert

Die Tabellen PreknowledgeSession (Vgl. Anhang 2) und TestSession (Vgl. Anhang 4) sind gleich aufgebaut und halten die erfassten Daten des Startzeitpunktes und Endzeitpunktes des durchgeführten Tests, alle richtigen und falschen Antworten und die über die Testzeit gesammelten Herzfrequenzen und deren Timestamps. Die letzte Tabelle LearnSessions (Vgl. Anhang 3) hält die in den PreknowledgeSession und TestSession genannten Daten zusammen mit der Herzfrequenz sowie Start- und Endzeitpunkten des reinen Lernens fest. Die für die Kommunikation mit der Datenbank zuständige Schnittstelle in der 'mobile' Applikation heißt MyHttpClient aus dem Package *com.example.vibring.http.api*. Ausgehend von dieser kann jedes Fragment oder jede Activity durch eine Instanziierung dieser Klasse auf der Datenbank lesen, schreiben oder Tabelleneinträge updaten. Dabei wird mittels der erstellten Klasse Singleton¹ dafür gesorgt, dass Datenbankabfragen geordnet und nacheinander erfolgen. Intern kommuniziert die Applikation mittels Broadcast Receiver und Intents (Vgl. Kapitel 2.2.6) oder über Interfaces wie in den Listings 4.1 und 4.2 exemplarisch dargestellt.

Listing 4.1: Beispielcode Implementation und Aktivierung eines Interface

```

1  // any code
2  if (mListener != null) mListener.onStateChange();
3  // any code...
4
5  // Variabel change Listener
6  private ReceiverBroad.Listener mListener = null;
7  public void registerListener(ReceiverBroad.Listener listener) {
8      mListener = listener;
9  }
10 public interface Listener{
11     void onStateChange();
12 }
```

Im Listing 4.1 wird in den Zeilen 6–12 ein Interface vorbereitet. Auslösefunktion für das Interface ist die *void onStateChange()* Funktion in Zeile 11. Hat eine andere Klasse, wie in diesem Beispiel, ein Objekt der ReceiverBroad Klasse instanziiert und auf dem Objekt ein ReceiverBroad.Listener registriert, bemerkt dieser mit, wenn das Objekt der ReceiverBroad Klasse die Codezeile 2 ausführt und kann Daten abfragen oder Methoden starten.

Listing 4.2: Beispielcode Interface onStateChange() Listener

```

1  ReceiverBroad receiverBroad;
2  receiverBroad = new ReceiverBroad(context);
```

¹ Singleton Klassen werden dadurch ausgezeichnet, dass diese nur eine Instanziierung zulassen [51].

```

3 receiverBroad.registerListener(new ReceiverBroad.Listener() {
4     @Override
5     public void onStateChange() {
6         // ToDo Code
7     }
8 }

```

In diesem Listing 4.2 wird exemplarisch ein Listener auf dem Interface registriert. Sobald in der ReceiverBroad Klasse die Methode *onStateChange()* ausgeführt wird, können in dieser Klasse Daten abgefragt oder weitere Methoden ausgeführt werden.

Die eigens erstellten Klassen der Applikation 'Vibning' wurden so abstrakt wie möglich programmiert. Der Grund hierfür liegt darin, dass der gleiche Programmcode an verschiedenen Stellen wieder verwendet werden kann (Vgl. Abb. 4.6). Ebenso wurde versucht, Codes für Logik und Eingriffe in das Userinterface in verschiedenen Klassen zu trennen. Die Benennung von Variablen und Layouts wurde möglichst klar und kurz gehalten. In der Android Programmierung gibt es die Möglichkeit, Layouts in einer XML Datei anzufertigen und mittels einer *id* programmatisch darauf zuzugreifen. Die gesetzte *id* wurde im Programmcode verwendet, um der dazugehörigen Variable den gleichen Namen zu geben. Der Name jedes Widgets besteht aus drei durch Unterstriche getrennte Teile (XX_YYY_ZZZ). Dabei kürzt XX die Art des Widgets ab, YYY kürzt das dazugehörige Fragment oder die dazugehörige Activity ab. ZZZ kürzt die Intention des Widgets ab, also ob beispielsweise eine Antwort eingeblendet oder eine Lösung bereitgehalten wird.

Listing 4.3: Beispiel Widget Konvention

```

1 <TextView
2     android:id="@+id/tv_learn_learnContent"
3     android:layout_width="match_parent"
4     android:layout_height="wrap_content"
5     android:gravity="center"
6 />

```

Hier wird ein Textfeld (TextView) mit *tv* abgekürzt, welches sich im Fragment *learn* (LearnFragment) mit der Intention *learnContent* (Lerninhalt) anzuzeigen, erstellt.

Listing 4.4: Programmatische Textänderung einer TextView

```

1 tv_learn_learnContent.setText(learnContentHolder.getRightMorseCode(element));

```

Die programmatische Textänderung kann man dem Listing 4.4 entnehmen. Hier wird dem TextView ein neuer String als Inhalt (Morse Code) hinzugefügt. Zur besseren Übersicht im Programmcode wurden die Instanziierungen von verschiedenen Klassen und Widgets größtenteils aus den *onCreate()* Funktionen in die *init()* Funktion für logische Instanziierungen und die *initWidget()* Funktion für Widgets verlagert. Das Design der *mobile* Applikation und somit die

meisten verwendeten *Widgets* erben von dem durch Material Design [27] bereitgestellten *Theme MaterialComponents.Light.NoActionBar*. Das dazu passende Logo entstammt der Homepage FreelogoDesign.org [25].

4.2 Mobile Applikation

Dieses Kapitel befasst sich mit der Entwicklung der *'mobile'* Applikation. Zunächst werden dem *Benutzerflow* gemäß LoginActivity und RegistraionActivity behandelt. Im Anschluss wird näher auf die MainActivity und somit auf den Dreh- und Angelpunkt der Applikation *'Vibning'* eingegangen. Bei der Besprechung der MainActivity werden sämtliche implementierte Fragmente näher betrachtet. Das Hauptaugenmerk liegt dabei auf der exemplarisch detaillierten Beschreibung des PreknowledgeFragments und dessen Architektur. Weitere wichtige Fragmente der Applikation, wie das TestFragment oder das LearnFragment, werden im Anschluss auf deren Unterschied zum PreknowledgeFragment beschrieben. Eine Übersicht über alle Activities und Fragmente der *'mobile'* Applikation ist der Abbildung 4.2 zu entnehmen.

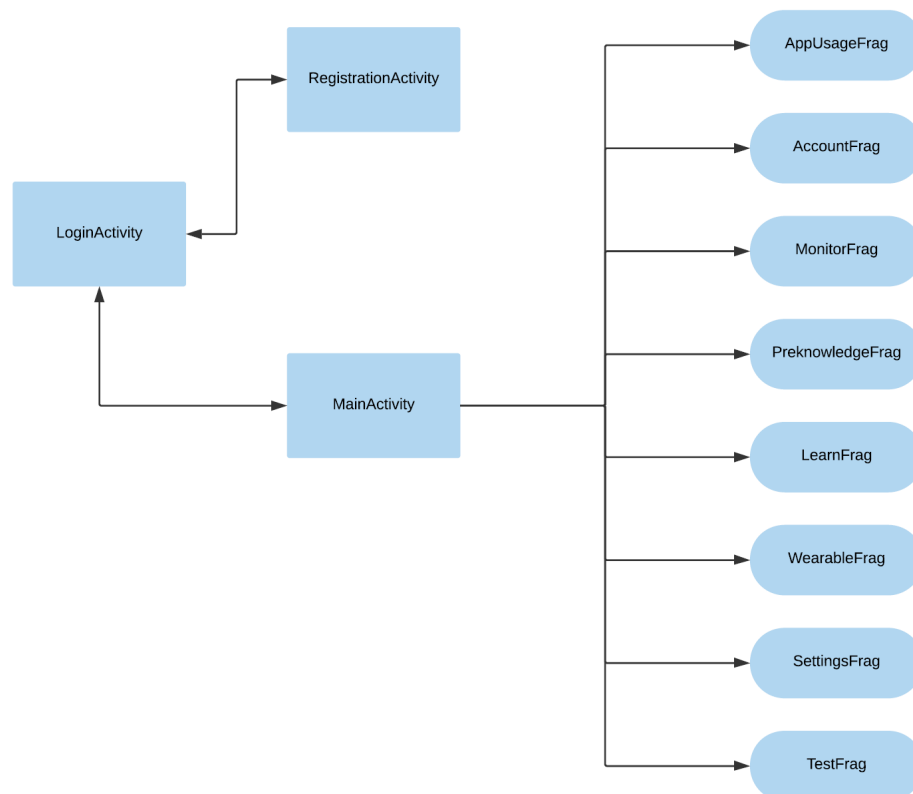


Abbildung 4.2: Activities und Fragmente der *'mobile'* Applikation

Die Graphik 4.2 beschreibt ausgehend von der LoginActivity welche Features wie zu erreichen sind. Von der LoginActivity kann man zur RegistrationsActivity und wieder zurück gelangen. Über die MainActivity erreicht man, insofern das Fragment für das Teilnehmerlevel zugänglich ist, die rechts aufgeführten Fragmente.

4.2.1 LoginActivity und RegistrationActivity

Startpunkt der gesamten Applikation ist die LoginActivity mit ihrem Layout für den Login (Abbildung 4.3). Von der LoginActivity ausgehend kann der Benutzer durch eine erfolgreiche Anmeldung zur MainActivity gelangen oder ansonsten zur RegistrationActivity wechseln. Bei einer erfolgreichen Anmeldung werden aus der Datenbank UserData alle vorhandenen Daten gelesen. Mit Hilfe der erstellten Klassen SessionHandler, User und der importierten Klasse SharedPreferences [24] werden diese Daten zugänglich für jede Activity (Kapitel 2.2.3) oder jedes Fragment (Kapitel 2.2.4) dieser Applikation. Ebenso hält die LoginActivity ein Feature des *SessionHandlings*, mit dem man den Login bei bereits kürzlich erfolgreicher Anmeldung automatisch überspringt. Dem Listing 4.5 entnehmbar ist ein automatisches Wiederlogin innerhalb von drei Stunden möglich.

Listing 4.5: Session handling

```
1 // Adding an expire Date to the SharedPreferences
2 Date date = new Date();
3 long milliseconds = date.getTime() + 10800000; // ~ 3 Stunden
4 cEditor.putLong(KEY_EXPIRES, milliseconds);
5 cEditor.commit();
```

Die Klasse *SharedPreferences* [24] bietet zusammen mit dem *SharedPreferences.Editor* die Möglichkeit Daten unabhängig vom Lebenszyklus 2.2.3 der Applikation und den Activities zu speichern. Die RegistrationActivity bietet dem Benutzer die Möglichkeit, bei vollständiger Eingabe des Formulars, sich zu registrieren und der Tabelle UserData einen neuen Eintrag hinzuzufügen. Die Abbildung 4.3 zeigt links das Layout der LoginActivity und rechts die der RegistrationActivity. Die Radio Button Auswahl *Group 1* und *Group 2* des Registrationslayouts bestehen für beide Studiengruppen, *mit Vibration* und *ohne Vibration*. Wenn eine Anmeldung oder Registrierung nicht erfolgreich ist, wird im Layout eine TextView mit einem Fehlerhinweis eingeblendet. Diesen, über die PHP Skripte beschriebenen, Fehlerhinweis empfängt das Handy mittels der Klasse *MyHttpClient*. Sowohl die RegistrationActivity als auch die LoginActivity interagieren über ein Interface auf deren instanziierten *MyHttpClient*-Objekt. Mögliche Fehlercodes können beispielsweise das Fehlen einer Internetverbindung, Login bzw Passwort inkorrekt eingegeben, sein.

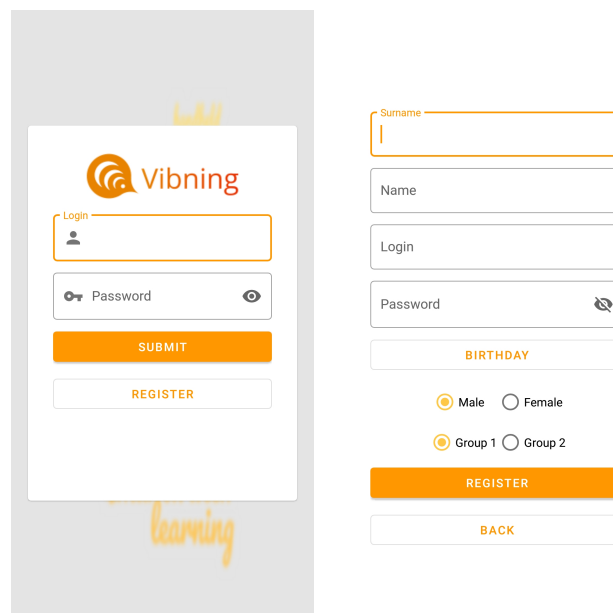


Abbildung 4.3: Screenshots Login & Registrierungs Layout

4.2.2 MainActivity und Fragmente

Die MainActivity dient in der Applikation als NavigationDrawer [11] und bündelt sämtliche zu erreichende Fragmente in einem Menü (linker Teil der Abbildung 4.6). Beim Start der MainActivity wird zunächst das AppUsageFragment in ein FrameLayout geladen und für den Benutzer sichtbar gemacht. Dieses Fragment beinhaltet lediglich einen Text, welches den Probanden oder die Benutzer dieser Applikation für die Studie instruiert (Vgl. rechter Teil der Abbildung 4.6). Alle restlichen Fragmente, welche für das aktuelle Teilnehmerlevel bestimmt sind, sind über die Menüleiste erreichbar (Abbildung 2). Das Menü kann sowohl mittels des Floating Action Button als auch mit einer wischenden Bewegung (von der linken Seite des Bildschirm nach rechts ziehen) geöffnet werden. Bei einigen Fragmenten wird die Eigenschaft des Wischens und der Floating Action Button übernommen und deren Funktionalität abgeändert. So kann in fast allen Test- und Lernfragmenten nicht vor Abschluss der Einheit ins Menü zurück gewechselt werden. Bei Fragmenten, in denen diese Funktionalität noch offen steht, lässt sich dies programmatisch mittels der implementierten Beispiele ändern. Dies soll im Allgemeinen ein Wiederholen der Einheit ohne vorheriges Absenden der gesammelten Daten verhindern. Das Pausieren der Applikation durch Wechsel in den Homescreen oder zu einer anderen Applikation beendet das laufende Fragment und setzt erneut das AppUsageFragment in das FrameLayout der MainActivity (R.layout.activity_main). Speziell für die MainActivity und seine Fragmente wurde in der

Manifest-Datei ein `MessageService` angelegt. Dieser ist im Listing 2.1 beschrieben.

Listing 4.6: Manifest `MessageService`

```

1 <service
2   android:name=". Services . MessageService "
3   android:enabled="true"
4   android:exported="true">
5   <intent-filter>
6     <action android:name="com.google.android.gms.wearable.MESSAGE_RECEIVED" />
7     <data android:scheme="wear" android:host="*" android:pathPrefix="/response" />
8   </intent-filter>
9 </service>

```

Das `PathPrefix` dieses `MessageServices` bestimmt den Pfad auf der Datenschicht, in denen geschriebene Nachrichten über einen `Listener` abgehört werden. Schreibt ein anderes Gerät auf diesen Pfad oder einen Pfad mit dem gesetzten Präfix wird es von der Applikation mittels der Klasse `MessageService` registriert. Die `MessageService` Klasse sendet anschließend an alle lokal registrierten `BroadcastReceiver` Instanziierungen den empfangenen Pfad samt Präfix und die hinterlegte Nachricht als Intent. Jede Instanziierung der Klasse `BroadcastReceiver` registriert in dessen Konstruktor einen Receiver für lokale Intents. Damit das Fragment mitbekommt, dass das instanziierte Objekt einen Intent geschickt bekommen hat, bietet die Klasse `BroadcastReceiver` die Möglichkeit an, ein Interface auf das erstellte Objekt zu registrieren. Die

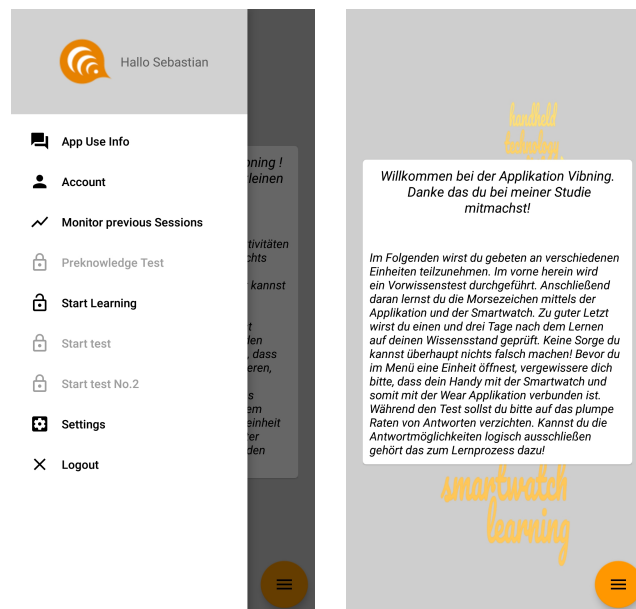


Abbildung 4.4: Screenshot des Applikations Menüs der MainActivity und dessen `AppUsageFragment`

Architektur der Fragmente zum Lernen und Testen wird anhand des `PreknowledgeFragments` genauer beschrieben (Layouts der Fragmente Abb. 4.5). Eine schematische Darstellung der Ar-

chitektur ist der Abbildung 4.6 zu entnehmen. Insgesamt haben drei Klassen der Abbildung 4.6 Zugriff auf das Userinterface, welches durch das unterste Symbol der Darstellung präsentiert wird. Das *PreknowledgeFragment* beinhaltet eine Reihe von Widgets, die dafür sorgen, dass der Benutzer sowohl Lerninhalt, verschiedene Antwortmöglichkeiten in Form von sechs Buttons und den Fortschritt des aktuellen Tests sehen kann. Die Klasse *StartDialog* wird beim Erstellen des Fragments aufgerufen. Dieser Dialog überprüft, ob eine Verbindung zur Smartwatch hergestellt werden kann und bereitet den Test für den Benutzer vor. Hierdurch wird auch der Dialog die Smartwatch Activity und die Aufnahme der Herzfrequenz gestartet, anschließend wird eine Test Vibration durchgeführt. Der *InformationDialog* übernimmt das Userinterface am Ende des Tests und gibt dem Benutzer Feedback über seine erbrachte Leistung (Prozentzahl der korrekten Antworten), die benötigte Zeit und alternativ, durch Rückgängigmachen der Kommentare in der Klasse, auch weitere Daten des Tests (Herzfrequenz, korrekte und nicht korrekte Antworten). Der Lerninhalt für den Test wird von der Klasse *LearnContentHolder* bestimmt. Die Klasse liefert auf Anfrage des *PreknowledgeFragments* neue Morsezeichen mit passendem Buchstaben in oder ohne Reihenfolge. Ebenso werden für einen Buchstaben ein Set zufällig gewählter anderer Buchstaben übergeben, so dass alle sechs *Buttons* beschriftet werden können. Der Fortschritt, für die *Progressbar*, welcher über den Lerninhalt bestimmt wird, und weitere Funktionen wie das Mischen von Antwortmöglichkeiten werden ebenfalls von dieser Klasse ausgeführt. Die Klasse *TestResult* wird verwendet, um gesammelte Testdaten zu einem Objekt zu verpacken. Darunter befinden sich alle Einträge der Tabellen *PreknowledgeSession* und *TestSession* mit Ausnahme des PrimaryKeys. Im Anschluss an den Test wird ein Objekt dieser Klasse an den *MyHttpClient* übergeben, um einen neuen Eintrag in der Datenbank zu erstellen. Benötigte Nutzerdaten für das *TestResult* Objekt werden dabei über die Klassen *SessionHandler* und *User* gelesen. Die Klasse *User* speichert ebenso wie die Klasse *TestResult* lediglich Daten und bietet ausschließlich die Funktionalität des Schreiben und Lesen von Daten. Die Klasse *TestHandler* beschreibt den Teil der Logik, welche für die Synchronisation des Handys mit der Smartwatch verantwortlich ist. Ebenso gibt diese Klasse das fertige *TestResult* Objekt an den *MyHttpClient*. Sie kommuniziert mit der Smartwatch über ein Objekt der Klasse *SendMessage* und ein Objekt der Klasse *ReceiverBroad*. Sie ist dafür zuständig, dass gesendete Aufforderungen zum Vibrieren oder zum Zurücksenden der Testdaten von der Smartwatch geregelt ablaufen. Der *TestHandler* wartet beim Senden einer Nachricht über die Klasse *SendMessage* bis ihr Interface auf einem Objekt der *ReceiverBroad* Klasse die Bestätigung dafür erhalten hat. Erst nach dieser Bestätigung sorgt ein Interface vom *PreknowledgeFragment* auf dem *TestHandler* dafür, dass ein neues Element im Lerninhalt Widget angezeigt wird oder ein anderer Prozess weitergeführt wird.

Listing 4.7: ReceiverBroad Listener für die sequenzielle Abarbeitung des Lerninhalts

```
1 public void receiverInit() {  
2     receiverBroad = new ReceiverBroad(context);
```

```

3 receiverBroad.registerListener(new ReceiverBroad.Listener() {
4
5     @Override
6     public void onStateChange() {
7         if (receiverBroad.getMessage().equals("started Vibration")){
8             if (vibrationTmp != vibrationActive
9                 && vibrationActive == false){
10                 vibrationTmp = vibrationActive;
11                 vibrationActive = true;
12                 if (mListener != null) mListener.onStateChange();
13             }
14         } else if (receiverBroad.getPath()
15             .equals("/response/stoppedVibration")
16             && (receiverBroad.getMessage().equals(sendedPattern)
17                 || receiverBroad.getMessage().equals("default") )){
18             if (vibrationTmp != vibrationActive
19                 && vibrationActive == true){
20                 vibrationTmp = vibrationActive;
21                 vibrationActive = false;
22                 if (mListener != null) mListener.onStateChange();
23             }
24         } else if (receiverBroad.getPath()
25             .equals("/response/stopSensorAndAnswer")){
26             heartLogs = receiverBroad.getHeartRatesLogs();
27             timeLogs = receiverBroad.getTimeLogs();
28             sendResults(timeLogs, heartLogs);
29         }
30     }
31 });
32 }

```

Die Funktion *receiverInit()* aus der Klasse *TestHandler* implementiert den Listener für die *ReceiverBroad* Klasse. Die *ReceiverBroad* Klasse bekommt vom *MessageService* über einen *LocalBroadcast* übermittelt, wenn auf den Pfad mit Prefix */response* geschrieben wurde. Über das Interface kann der *TestHandler* bei Empfangen einer Nachricht (bestehend aus String path und String message) überprüfen, ob eine Bestätigung für die gesendete Aufforderung erhalten wurde. Die Zeilen 8 bis 24 handeln die gesendeten Vibrationsmuster ab, indem nach dem Senden eines Vibrationsmusters hier eine Bestätigung eingeht. Um keine Unstimmigkeiten zwischen den gemessenen Zeiten der Experimentalgruppen zu erzeugen werden Aufforderungen zum Vibrieren für beide Gruppen gesendet. Beim Wechseln eines Vibrationsmusters zu einem anderen muss zunächst eine Aufforderung zum Stoppen der Vibration gesendet und auf dessen Bestätigung gewartet werden. Erst dann kann ein neues Vibrationsmuster gesendet werden. Das Prozedere soll verhindern, dass einzelne Vibrationthreads auf der Smartwatch parallel verschiedene Vibrationsmuster vibrieren. Die darauffolgende *if* Abfrage in Zeilen 25 bis 30

bearbeitet die Bestätigung des Abschluß des Tests und die Übermittlung gesammelter Herzfrequenzen und deren Aufnahmezeitpunkte. Die letzte aufgerufene Funktion *sendResults(String[] timeLogs, String[] heartLogs)* in dieser Klausel sendet das vom PreknowledgeFragment übermittelte TestResults Objekt mit den letzten fehlenden Daten über den MyHttpClient an die Datenbank. Ebenfalls löst der TestHandler ein Update des User Eintrags in der Tabelle UserData aus, um deren Teilnehmerlevel nach Beendigung des Tests zu erhöhen. Die Klasse SendMessage bietet alle nötigen Funktionen, um mittels dem Pfadpräfix */msgHandler* Nachrichten an die Smartwatch zu senden. Dabei wird in einer inneren Klasse von SendMessage ein neuer Thread für das Senden der Daten geöffnet.

Listing 4.8: Ausschnitt der inner Klass NewThread von der Klasse SendMessage

```

1 public void run () {
2     Task<List<Node>> wearableList =
3         Wearable.getNodeClient(context).getConnectedNodes();
4     try {
5         List<Node> nodes = Tasks.await(wearableList);
6         for(Node node : nodes){
7             Task<Integer> sendMessageTask =
8                 Wearable.getMessageClient(context)
9                     .sendMessage(node.getId(), path,
10                        message.getBytes());
11         };

```

Die Klasse NewThread aus dem Listing 4.2.2 sendet an alle mit dem Smartphone über Bluetooth verbundenen Smartwatches die Nachricht (Variable message Zeile 10) an den Pfad (Variable path in Zeile 9). Die Klasse MyHttpClient importiert für jegliche Http Requests die *Library Volley* [54]. So muss der MyHttpClient nur die zu übermittelnden Daten in ein JSON Objekt packen und mittels der Http Post Methode eine Abfrage senden. Alle noch nicht in diesem Kapitel vorgekommen Klassen der Architektur, welche sich über der Datenschicht befinden, werden im Kapitel 4.3 näher erläutert, da diese für alle Fragmente identisch sind.

Die Architektur des TestFragments ist bis auf einen Unterschied identisch zum PreknowledgeFragment aufgebaut. Das TestFragment wird für die Teilnehmerlevel 2 und 3 verwendet. Während der Benutzer mit Teilnehmerlevel 2 exakt den gleichen Testaufbau wie das PreknowledgeFragment als Testsequenz erhält, öffnet sich das TestFragment für Benutzer mit Teilnehmerlevel 3 mit einem NumberPicker Widget [12] versehen mit allen vorhandenen Antwortmöglichkeiten (gesamte lateinische Alphabet + 'keine Antwort' Möglichkeit, Vgl. 4.5). Das LearnFragment verwendet anstatt einem TestResults Objekt ein LearnResults Objekt. Der Unterschied dieser beiden Klassen besteht darin, dass das LearnResults Objekt zu den Daten des TestResults Objekt zusätzlich Zeiten und Herzfrequenzen während des Lernens spei-

chert. Ebenfalls wird statt dem TestHandler ein Objekt der Klasse LearnHandler instanziiert. Der Grund hierfür sind die zusätzlichen Daten des LernResult Objekts und das Abhandeln der Lern- und Testsequenzen. Eine Einheit des LearnFragments besteht aus einer Lerneinheit über alle Lerninhalte mit einem anschließenden gleichen Test wie dem des PreknowledgeFragments. Das LearnFragment wiederholt seine Funktionalität solange bis der Benutzer bei einem Lerntest alle Paarungen richtig wählt.

Das Fragment *AccountInfoFragment* zeigt die in der Tabelle *UserData* aktuell gespeicherten Daten über das Konto an (Vgl. Anhang 6). Das Fragment bietet dem Nutzer ebenfalls die Möglichkeit über ein Popup Dialog sein Passwort zu ändern. Das MonitorFragment gibt den Benutzer die Möglichkeit seine gesammelten Testdaten einzusehen. Dabei werden Herzfrequenz, falsche und richtige Antworten und Zeitpunkt des Tests angezeigt. Das letzte Fragment des Menüs aus der Abbildung 2 ist das *WearableFragment* (Settings). Dieses Fragment dient lediglich zu Entwicklungszwecken. In diesem Fragment sind nützliche Codebeispiele zum Fernsteuern der Smartwatch hinterlegt. Keines dieser Funktionen sind in der abgegebenen Version aktiv, sie lassen sich jedoch programmatisch reaktivieren. Die Layouts der zuletzt genannten Fragmente befinden sich im Anhang 6.

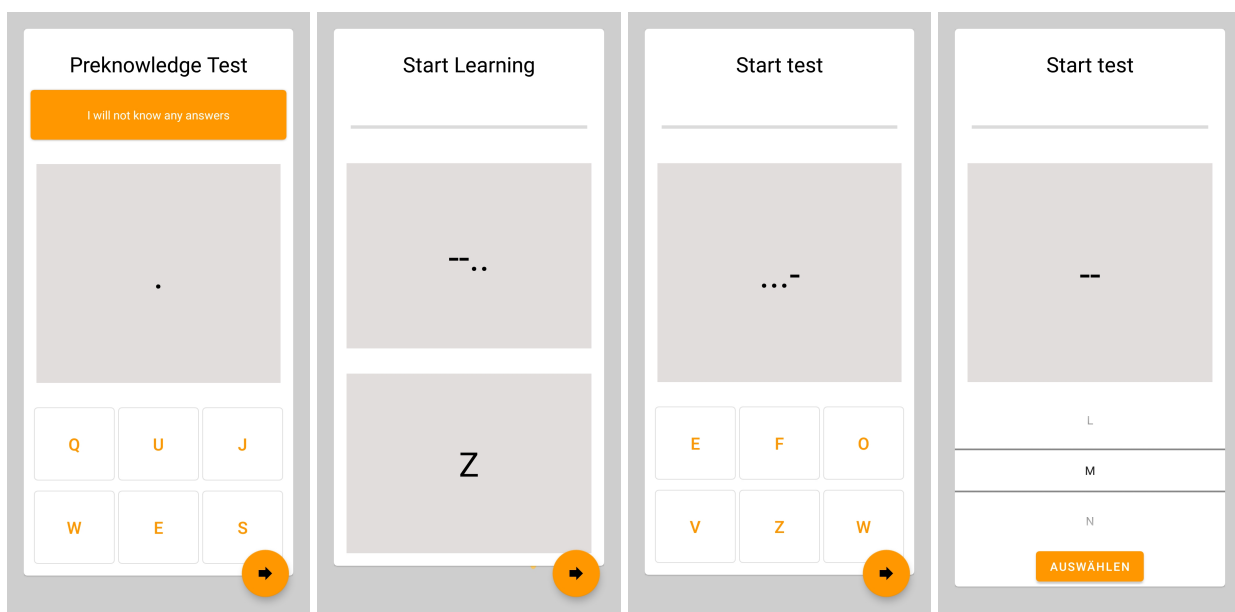


Abbildung 4.5: Layouts der Vibning Applikation für Test und Lerneinheiten

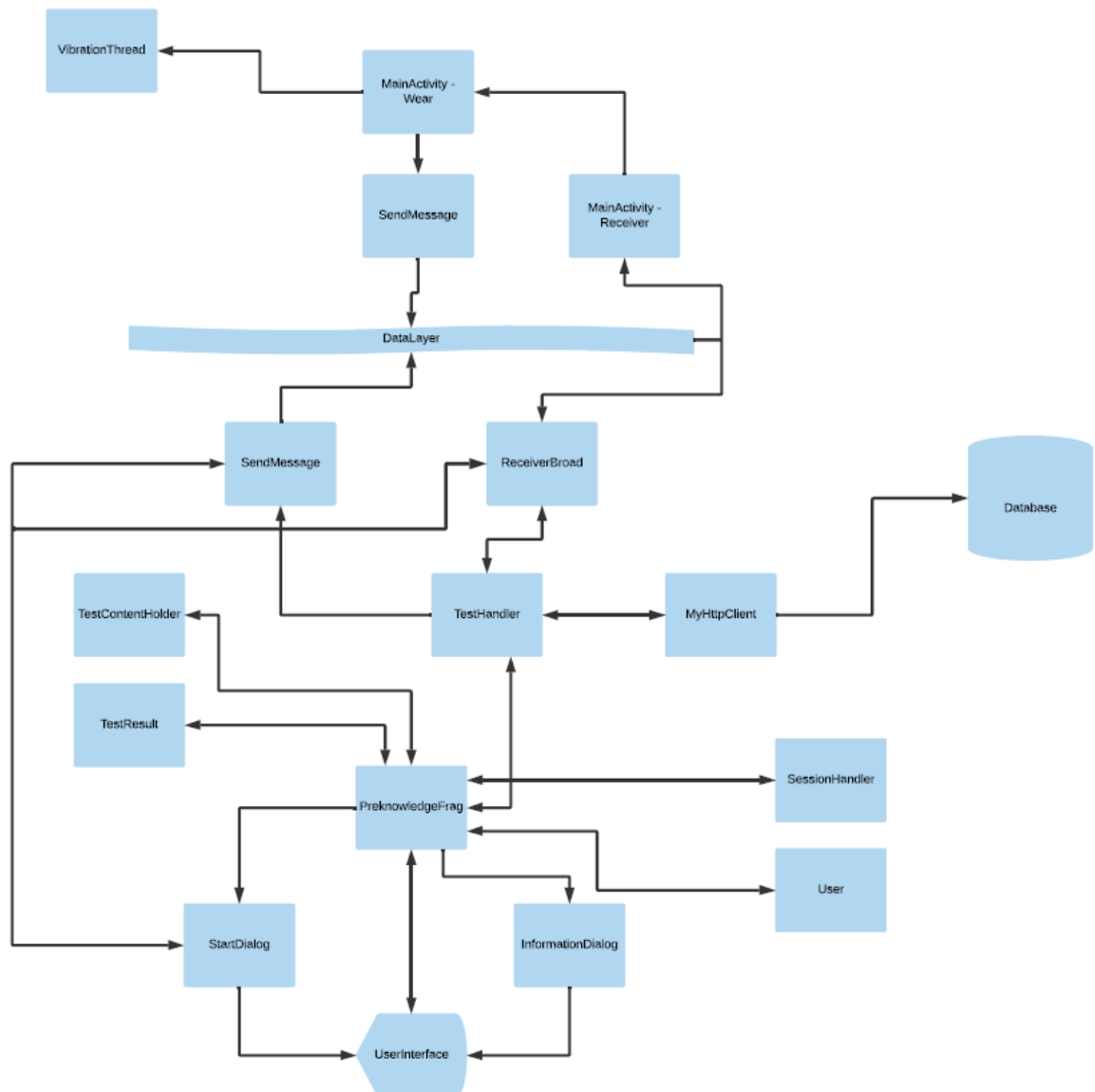


Abbildung 4.6: Screenshot der schematischen Architektur des PreknowledgeFragments

4.3 Wear Applikation

Die Wear Applikation besteht aus einer MainActivity mit zwei inneren Klassen, einem `MessageService` für das Empfangen von Nachrichten der Mobile Applikation, einer Klasse für die Vibration der Smartwatch und einer Klasse um Nachrichten zum Smartphone zu senden. Der Aufbau des `MessageService` und der `SendMessage` Klasse entspricht der Implementation bei der Mobile Applikation. Anders als die Mobile Applikation schreibt die Wear Applikation auf den Pfadpräfix `/response` und liest Nachrichten vom `/msgHandler` Pfad. Die innere Klasse `Receiver` empfängt, ähnlich wie die Klasse `ReceiverBroad`, lokale Broadcasts vom `MessageService`. Die zweite innere Klasse `trackHeartRate` öffnet einen neuen Thread, um Herzfrequenzen des Benutzers zu messen. Die Wear Applikation hat kein Interface mit dem der Benutzer interagieren muss. Verwendete Widgets sind mit Ausnahme des Buttons lediglich zum Debuggen eingebaut. Der Button mit der Aufschrift `onErrorClick` kann im Fall, dass das Smartphone eine Bestätigungsnachricht nicht empfängt, eine wiederholte Bestätigung senden, um den Applikationsprozess fortzuführen. Damit die Applikation funktionieren kann, wird zunächst bei Neuinstallation der Applikation nach Berechtigungen für die Sensoren gefragt. Die innere Klasse `Receiver` funktioniert, indem sie einen Intent mit zwei Parametern als LocalBroadcast vom `MessageService` übergeben bekommt. Mittels des Strings `path` wird die Methodenzuweisung ausgewählt. Mit dem String `message` werden, wenn nötig, Vibrationspattern ausgelesen. Die zweite innere Klasse `trackHeartRate` greift mit Hilfe der importierten Klassen `SensorManager` und `Sensor` [13] auf den Sensor `Sensor.TYPE_HEART_RATE` zu. Dabei speichert ein Objekt der Klasse alle Wertänderungen der Herzfrequenz in ein Array. Dieses Array wird zusammen mit den dazugehörigen Zeitpunkten auf Anfrage des Smartphones als Nachricht zurückgesendet. Um Vibrationen auf der Smartwatch auszulösen instanziiert die MainActivity ein Objekt der Klasse `VibrationThread`, welche von der Klasse `Thread` erbt. Übergeben wird dabei das Vibrationspattern und der Vibrationssensor. Der `VibrationThread` ist so programmiert, dass sich Vibrationen mit einer Pause von 1500ms solange wiederholen bis die MainActivity den neuen Thread unterbricht. Die Klasse `VibrationThread` setzt Vibrationen für Punkte (Dots) des Morsezeichens auf 100ms und die für Striche (Dashes) auf 300ms. Eine Amplitudenkontrolle der Vibration konnte aufgrund der TicWatch C2 nicht eingebaut werden. Damit die Wear Applikation einwandfrei funktionieren kann, muss sie nach einem Neustart der Smartwatch einmal manuell ausgeführt werden. Anschließende Reinigung der Testdaten übernimmt die mobile Applikation per Fernsteuerung. Die Smartwatch TicWatch C2 ist per *Default* so eingestellt, dass sie den Akku so effizient wie möglich verbraucht. Dabei kann sie in einen Ruhemodus schalten, bei dem einzelne Funktionalitäten, wie die des Vibrierens, nicht mehr funktionieren. Aus diesem Grund wurden Funktionen der Applikation, welche das *Ambient* der Uhr registrieren,

überschrieben. Das dazugehörige Tutorial findet sich auf der Android Homepage [2]. Alternativ kann man die Klasse `PowerManager` [3] dazu verwenden, um ein Fehlverhalten der Applikation zu verhindern.

4.4 Umgang mit der Applikation

Um sicherzustellen, dass die Applikation funktioniert, wird in diesem Unterkapitel der Umgang mit der Applikation erläutert. Die *mobile* Version lässt sich wie jede andere Applikation über das *Launcher-Icon* starten. Das *Smartphone* benötigt sowohl eine aktive Internetverbindung als auch eine aktive Bluetooth Verbindung zur *Smartwatch*. Die *wear* Version muss bei jeder Neuinstallation oder Neustart der Uhr einmal manuell geöffnet werden. Beim Start einer Lerneinheit sollte man, um sicherzugehen, einmal auf den Bildschirm der Smartwatch drücken, so dass dieser von einem dunkel beleuchteten auf einen hell beleuchteten Hintergrund wechselt. Anschließend kann die Lern- oder Testeinheit ausgeführt werden. Im StartDialog der *mobile* Version wird auf eine korrekte Verbindung überprüft. Ohne eine korrekte Verbindung kann nicht durch die Lerninhalte navigiert werden. Eine Überprüfung der Bluetooth Verbindung und ein Neustart der *mobile* Applikation kann hier Abhilfe schaffen. Sowohl die *Wear* Applikation als auch die *Mobile* Applikation haben derzeit kein eingebautes Feature, um einen Datenverlust zu verhindern. Dies kann beispielsweise mittels der Klasse `SharedPreferences` nachgerüstet werden. Aus diesem Grund empfiehlt es sich vor Beginn einer Einheit beide Geräte auf genügend Akkukapazität zu überprüfen. Beide Geräte kann man dahingehend einstellen, dass diese bei sehr geringen Akkustand nicht in einen Energiesparmodus wechseln. Der Energiesparmodus der Smartwatch beeinflusst die Funktionalität der Sensoren zum Vibrieren und Messen.

5 Evaluation

Dieses Kapitel beschäftigt sich mit der Evaluierung der Studie. Die geplante Vorgehensweise und das Ziel dieser Studie wurden im Kapitel 3 beschrieben. In den ersten drei Unterkapiteln wird zunächst die Ausgangslage nach allen Lern- und Testeinheiten der Probanden besprochen. Eine weiterführende Auswertung der Studienergebnisse erfolgt in den anschließenden Unterkapiteln. In diesem Kapitel und dem kommenden Kapitel wird der Begriff Kontrollgruppe mit KG abgekürzt. Die Experimentalgruppe mit EG. Das Kürzel n steht für eine Teilnehmeranzahl, AVG steht für den Durchschnitt beider Gruppen. Der Begriff Korrektheit beschreibt den prozentualen Anteil der richtig gewählten Paarungen gegenüber der Anzahl aller möglichen Paarungen von Morsezeichen. Hilfsmittel für dieses Kapitel sind neben den erhobenen Datensätzen durch die Applikation die Programmiersprache Python [47] mit den importierten Modulen von SciPy [48], Numpy [44] und Matplotlib [38] für Grafiken und Rechnungen. Des Weiteren wurde für die Auswertung, insbesondere für die Signifikanztests, die IBM SPSS-Software herangezogen [30].

5.1 Vorwort

An der im Rahmen dieser Bachelorarbeit durchgeführten Studie haben 33 Probanden teilgenommen. Davon waren 22 Probanden männlich und 11 weiblich. Aufgrund der im Verlauf der Studie notwendigen Änderungen (siehe Kapitel 3.2) und weiterer technischen Schwierigkeiten konnten die Ergebnisse von zwei Personen überhaupt nicht berücksichtigt werden. Weitere zwei Personen mussten aufgrund technischer Schwierigkeiten während des Versuchs aus der Analyse der Lerneinheiten entfernt werden. Die Daten dieser Probanden können für die Tests verwendet werden. Alles in allem können 29 Datensätze für die Analyse der Lerneinheiten verwendet werden. Für den ersten Test können zuzüglich der zwei zunächst ausgeschlossenen Probanden 28 Personen, beim zweiten Test 25 Personen berücksichtigt werden. Ausgangspunkt der gesamten Studie war der Vorwissenstest der Probanden, welche für beide Gruppen ergab, dass kein Wissen über Morsezeichen vorhanden war.

5.2 Lerneinheiten

Von den 33 Probanden können für die Analyse der Lerneinheiten 29 Datensätze verwendet werden. Vier von diesen Probanden (drei Personen aus der KG und eine Person aus der EG) haben die Lerneinheiten nicht mit einer Korrektheit von 100% abgeschlossen. Zwei der vier Probanden haben die Lerneinheit mit einer Korrektheit von 96% beendet (jeweils ein Proband pro Gruppe). Die anderen beiden Teilnehmer haben die Studie aus zeitlichen Gründen während des Lernens verlassen. In der Abbildung 5.1 sind alle gesammelten Testsequenzen der

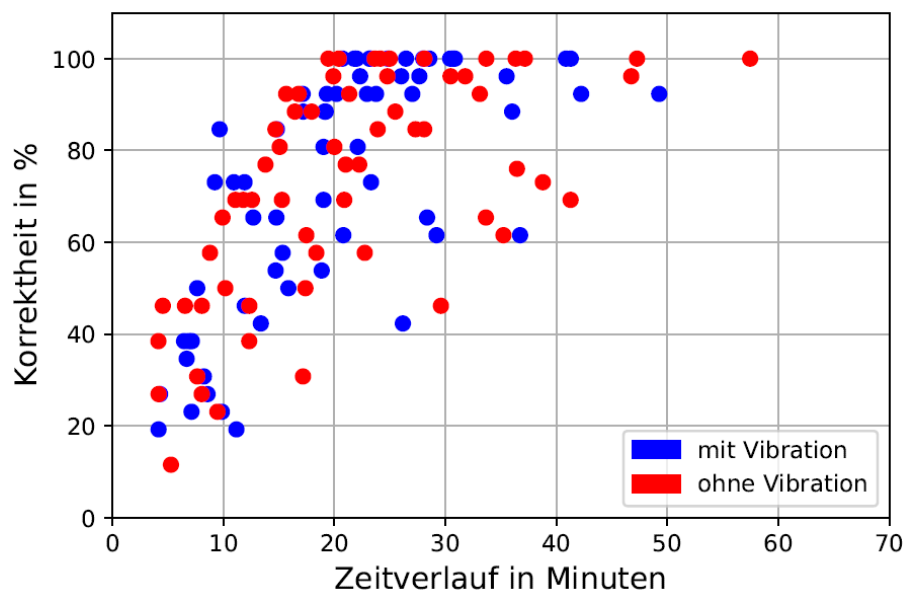


Abbildung 5.1: Zusammenstellung aller Testsequenzen der Lerneinheiten

Lerneinheiten dargestellt. Die x-Achse beschreibt die benötigte Zeit beim Lernen bis zum Ende der jeweiligen Testsequenz. Die y-Achse gibt die gemessene Korrektheit der einzelnen Testsequenzen wieder. Probanden wurden beim Lernen nicht angewiesen, wie lange eine Lerneinheit zu sein hat. Daraus resultierend können abhängig vom Probanden beliebig viele Punkte der Abbildung einer Person zugesprochen werden. Im Durchschnitt haben Probanden der EG 4.42 Lerneinheiten absolviert, wohingegen die andere Gruppe im Schnitt 4.26 Lerneinheiten gebraucht hat. Farblich rot markierte Punkte gehören zu Personen der KG. Die blauen Punkte beschreiben die Lerneinheiten der EG. Graphen zu den einzelnen Teilnehmergruppen und weitere Graphen, welche nicht fertig gelernte Probanden exkludieren, befinden sich im Anhang 6.

Die durchschnittliche Lernzeit für die Probanden mit einer Korrektheit von 96% oder mehr ist in folgender Tabelle dargestellt. Die angegebenen Zeiten ergeben sich als Mittelwert über alle gemessenen Zeiten der Lerneinheiten.

Gruppen	Zeiten
Lernzeit EG	29.47 Minuten
Lernzeit KG	33.27 Minuten
Lernzeit AVG	31.16 Minuten

Tabelle 5.1: Durchschnittliche Lernzeiten der Probandengruppen für eine Korrektheit von 96% oder mehr

Exkludiert man Probanden, welche keine Korrektheit von 100 % beim Lernen erreicht haben verändern sich die durchschnittlichen Lernzeiten zu folgenden.

Gruppen	Zeiten
Lernzeit EG	28.06 Minuten
Lernzeit KG	32.05 Minuten
Lernzeit AVG	29.81 Minuten

Tabelle 5.2: Durchschnittliche Lernzeiten der Probandengruppen, welche 100% der Paarungen richtig hatten

Den Tabellen 5.1 und 5.2 entnehmbar haben Probanden, welche mit Vibration gelernt haben, im Schnitt weniger Zeit gebraucht um (fast) alle Paarungen richtig zu wählen. Die durchschnittliche Zeit für einen Lerndurchgang betrug für die KG 7.37 Minuten und für die EG 6.90 Minuten.

5.3 Testeinheiten

An der ersten Testeinheit haben 26 Personen zuzüglich der zwei zunächst für die Lerneinheit ausgeschlossenen Probanden teilgenommen. Im Folgenden werden die Testergebnisse mit und ohne den zwei Probanden beschrieben. Die erste Testeinheit für Partizipanten der Studie erfolgte einen Tag nach den Lerneinheiten. Der zweite Test erfolgte drei Tage nach dem Lernen. Die Abbildung 5.2 zeigt in blau die Testergebnisse der EG vom ersten Test (No.1) und vom zweiten Test (No.2). Die Testergebnisse der KG sind in rot dargestellt.

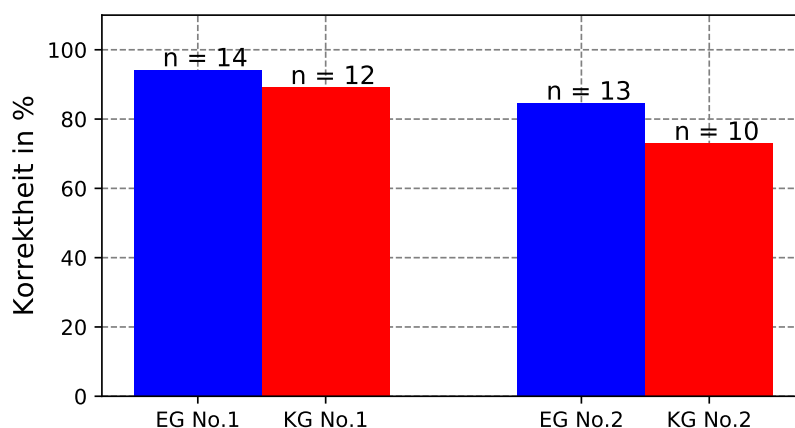


Abbildung 5.2: Zusammenstellung der Testsequenzen der verschiedenen Teilnehmergruppen

Die y-Achse dieser Grafik beschreibt die Korrektheit der Testergebnisse. Die x-Achse beschreibt die Testdurchläufe und die dazugehörigen Gruppen für den ersten und den zweiten Test. Die Testergebnisse der Abbildung 5.2 beziehen sich auf die beiden folgenden Tabellen auf der linken Seite. Tabellen auf der rechten Seite inkludieren weiterhin die Tests der zwei Probanden, welche aus technischen Schwierigkeiten bei den Lerneinheiten entnommen wurden.

Gruppen	Korrektheit
1. Test EG	94.23%
1. Test KG	89.10%
1. Test AVG	91.86%

Tabelle 5.3: Testergebnisse des ersten Tests

Gruppen	Korrektheit
1. Test EG	94.61%
1. Test KG	89.94%
1. Test AVG	92.44%

Tabelle 5.5: Testergebnisse des ersten Tests zuzüglich der zwei zunächst ausgeschlossenen Probanden

Gruppen	Korrektheit
2. Test EG	84.62%
2. Test KG	72.98%
2. Test AVG	79.56%

Tabelle 5.4: Testergebnisse des zweiten Tests

Gruppen	Korrektheit
2. Test EG	85.44%
2. Test KG	74.39%
2. Test AVG	80.58%

Tabelle 5.6: Testergebnisse des zweiten Tests zuzüglich der zwei zunächst ausgeschlossenen Probanden

Die benötigte Zeit um die Tests zu absolvieren unterschied sich ebenfalls. Für die Tests der Ta-

bellern auf der linken Seite benötigten Probanden der EG im Schnitt 3.4 Minuten für den ersten Test und 8.19 Minuten für den zweiten Test. Probanden der KG benötigten für den ersten Test durchschnittlich 2.95 Minuten und für den darauffolgenden Test 5.66 Minuten. Für die Tests auf der rechten Seite wurde für die EG ein Durchschnitt 3.22 Minuten und 7.67 Minuten gemessen. Die KG hingegen schnitt mit 2.86 Minuten und 5.67 Minuten ab.

5.4 Herzfrequenzen

Die gesammelten Daten zur Herzfrequenz dienen einem rein explorativen Zweck. Der Einfluss der Vibration am Handgelenk wird im Folgenden nicht berücksichtigt. Im Verlauf der Studie wurde zu jeder Lerneinheit und jeder Testeinheit die Herzfrequenzen aufgezeichnet. Für die Auswertung der Herzfrequenzen wurden die EG und die KG zusammengelegt. Der Grund hierfür ist, dass bereits beim Erheben der durchschnittlichen Herzfrequenzen für EG und KG fast keine Unterschiede festgestellt wurden. Die durchschnittliche Herzfrequenz für die Lerneinheiten sind 77.75 Schläge pro Minute für die EG und 81.34 für die KG. Bei den anschließenden Tests lagen die Werte dichter zusammen. So wurde für den ersten Test ein Durchschnitt von 84.71 Schläge pro Minute für die EG und 82.87 Schläge pro Minute für die KG gemessen. Der zweite Test hat mit 80.68 Schlägen pro Minute für die KG zu 80.85 Schlägen pro Minute für die EG einen ungefähren Unterschied von 0.18 Schlägen pro Minute. Für den explorativen Zweck werden die Probanden der Lerneinheiten, welche über 95% Korrektheit in einer Testsequenz hatten, in zwei Gruppen eingeteilt. Dabei werden sie abhängig von ihren Lernzeiten zugeordnet. Die erste Gruppe (Gruppe 1) enthält das Drittel der Lernenden mit dem geringsten Zeitaufwand. Die Gruppe 2 die übrig gebliebenen Teilnehmer.

Gruppen	Durchschnittliche Herzfrequenz	Anzahl
Gruppe 1	85,23	n = 9
Gruppe 2	76,56	n = 18

Tabelle 5.7: Testergebnisse des zweiten Tests zuzüglich zwei zunächst ausgeschlossenen Probanden

Der Tabelle 5.7 sind die durchschnittlichen Herzfrequenzen der Gruppen 1 und 2, sowie deren Teilnehmergrößen entnehmbar. Für den ersten und zweiten Test wurden in Abhängigkeit zu den Testergebnissen jeweils die gleichen Gruppeneinteilungen zu neuen Gruppen durchgeführt. Deren durchschnittliche Herzfrequenz und die Gruppengröße sind der folgenden Tabellen entnehmbar.

Testnummer	Gruppen	Durchschnitt. Herzfrequenz	Anzahl
1. Test	1. Gruppe	87,75	n = 9
1. Test	2. Gruppe	81.97	n = 18
2. Test	1. Gruppe	83.48	n = 8
2. Test	2. Gruppe	79.48	n = 17

Tabelle 5.8: Testergebnisse des ersten Tests zuzüglich zwei zunächst ausgeschlossenen Probanden

Die Anzahl der Probanden für dieses Unterkapitel ist für den ersten Test im Vergleich zum Kapitel 5.3 um eine Person gesunken. Das liegt daran, dass bei diesen Probanden nach Abschluss des Tests von der Applikation ausgehend keine HTTP Anfrage an die Datenbank gesendet werden konnte. Das Testergebnis wurde manuell ohne begleitende Daten in die Datenbank eingetragen.

5.5 Auswertung Lerneinheiten

Dem Kapitel 5.2 entnehmbar, haben Probanden der EG schneller gelernt als Probanden der KG. Im Folgenden werden wir uns auf die Tabelle 5.1 beziehen. Der Grund hierfür ist, dass zwei der vier nicht fertig angelernten Probanden der KG zeitlich sehr früh aus der Studie austreten mussten. Die übrig gebliebenen zwei Probanden, welche mit 96% Korrektheit abgeschlossen haben, hatten nach knapp 50 Minuten etwa die gleiche Prozentzahl an korrekten Paarung erwähnen können. Der Tabelle 5.1 zufolge haben Probanden der EG im Schnitt 3.80 Minuten weniger (12,89%) gebraucht, um dem (fast) gesamten Lerninhalt zu verinnerlichen.

Die Graphik 5.1 lässt sich dazu verwenden mit verschiedenen Modellen eine 'Lernkurve' zu generieren. Im Folgenden ist eine lineare Regression als Lernkurve für die einzelnen Teilnehmergruppen durchgeführt worden. Beispiele zur linearen Regression mit polynomiellen Features 3. Grades und zu einer logarithmischen Regression befinden sich im Anhang 6. Die Bedeutung der x und y-Achsen sind dieselben wie die der Graphik 5.1. Zu sehen sind zwei sich schneidende Geraden, welche sich an einem Punkt, nach circa 23 Minuten, kreuzen. Anhand der Steigung jeder einzelnen Gerade lässt sich die Lerngeschwindigkeit herleiten. Entstandene unterschiedliche Startpunkte, innerhalb der Grafik, lassen sich durch die ungeregelten Lerneinheitenzeiten erklären. Probanden der KG haben teilweise die ersten Lerneinheiten früher abgegeben als Probanden der EG (Vgl. Abb. 1 anhand der x-Achse). Der Graph zeigt für die blaue Gerade eine größere Steigung an, welche man sich durch die durchschnittlichen kürzeren Lernzeiten erklären kann.

Als Signifikanztest wurde für die Lerneinheiten zunächst der Mann-Whitney-U-Test gewählt. Dieser Test wird als das nichtparametrische Äquivalente zum t-Test für unabhängige Stichproben gesehen [41]. Der Vorteil hierbei ist, dass keine Normalverteilung der Stichprobe nach-

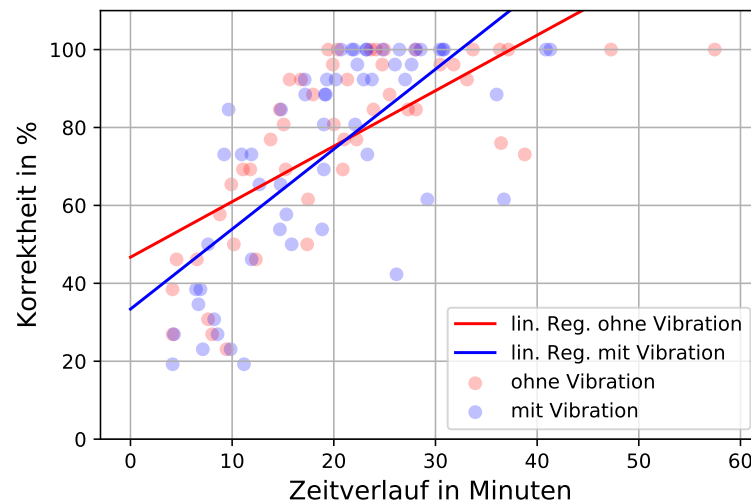


Abbildung 5.3: Lineare Regression der Testsequenzen

gewiesen werden muss. Weiterhin wird dem Mann-Whitney-U-Test zugesprochen, dass dieser auch bei kleinen Stichproben gut funktioniert und mit Ausreißern umgehen kann [41].

Der Mann-Whitney-U-Test wurde für beide Tabellen des Kapitel 5.2 angefertigt. So sind einmal alle Personen, welche mit einer Korrektheit von 100% die Lerneinheiten absolviert haben inkludiert und einmal alle die eine Korrektheit von über 95% haben. Als Signifikanzniveau wurde $\alpha = 0.05$ festgelegt. Die Hypothesen beider Tests waren identisch, $H_0 :=$ Die Verteilung der Lernzeiten ist über die Gruppen EG und KG identisch und die H_1 Hypothese als genaues Gegenteil, $H_1 :=$ Die Verteilung der Lernzeiten ist über die Gruppen EG und KG nicht identisch. Die Stichprobenanzahl n unterschied sich bei den Tests um 2 Einträge. Der Test zur 100% Kondition hatte $n = 25$ Stichproben. Die 95 % Kondition hatte 27 Stichproben. Beide Tests zeigten ähnliche Testergebnisse. Die 100% Kondition verwarf die Nullhypothese mit einer Signifikanz von 0,572 als exakte Signifikanz nicht. Die 95% Kondition tat das Gleiche mit einer Signifikanz von 0,548.

Des Weiteren wurde versucht für die verschiedenen Datensätze t-Tests für unabhängige Stichproben [40] zu rechnen. Um einen aussagekräftigen t-Test durchzuführen, müssen die Voraussetzungen auf Normalverteilung sowie auf Varianzgleichheit der Stichproben erfüllt sein [52]. Zum Testen von Stichproben auf Normalverteilung kann man sowohl den Kolmogorov-Smirnov Test als auch den Shapiro-Wilk Test verwenden [39]. Die Nullhypothese und die Alternativhypothese beider Tests sind identisch. Das Signifikanzniveau wird auch hier auf $\alpha = 0.05$ gesetzt. Die H_0 wird definiert durch $H_0 =$ Die Verteilung der Gruppe ist normalverteilt und die H_1 Hypothese definiert durch $H_1 =$ Die Verteilung der Gruppe ist nicht normalverteilt. Bei der Überprüfung mittels SPSS auf Normalverteilung der Lernzeiten für einzelne Probandengruppen wurden folgende Signifikanzen gefunden:

Tests auf Normalverteilung							
		Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Gruppen	Statistik	df	Signifikanz	Statistik	df	Signifikanz
Lernzeiten	EG	,234	15	,026	,861	15	,025
	KG	,170	12	,200*	,915	12	,248

*. Dies ist eine untere Grenze der echten Signifikanz.
a. Signifikanzkorrektur nach Lilliefors

Abbildung 5.4: Screenshot aus der SPSS Umgebung mit Tests auf Normalverteilung für die 95% Kondition einzelner Gruppen

Dem Testergebnis aus der Abbildung 5.4 zufolge ist der Datensatz der KG mit der 95% Kondition normalverteilt. Der Datensatz der EG ist nicht normalverteilt.

Tests auf Normalverteilung							
		Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Gruppen	Statistik	df	Signifikanz	Statistik	df	Signifikanz
Lernzeiten	EG	,191	14	,176	,870	14	,042
	KG	,178	11	,200*	,894	11	,157

*. Dies ist eine untere Grenze der echten Signifikanz.
a. Signifikanzkorrektur nach Lilliefors

Abbildung 5.5: Screenshot aus der SPSS Umgebung mit Tests auf Normalverteilung für die 100% Kondition einzelner Gruppen

Die Grafik 5.5 zeigt die Ergebnisse der gleichen Tests für die 100% Kondition an. Dabei ist erkenntlich, dass der Kolmogorov-Smirnov Test sowohl der EG als auch der KG zuweist, eine Normalverteilung zu sein. Der Shapiro-Wilk Test hingegen belegt eine Normalverteilung für den Datensatz der Lernzeiten der KG, nicht aber den der EG. Auf Basis der kleinen Stichprobengröße und dem Ergebnis des Kolmogorov-Smirnov Testes, kann man einen t-Test durchführen [39].

Der t-Test für die 100% Kondition mit einem Signifikanzniveau von $\alpha = 0.05$ ergibt bei einer Signifikanz der Varianzgleichheit (erfasst durch Levene-Test [40]) von 0.05 eine zweiseitige Signifikanz von 0.295. Für eine ungleiche Varianz ergibt sich eine Signifikanz von 0.33. Damit kann die Nullhypothese $H_0 = \text{Die Verteilung der Lernzeit ist über die Probandengruppen identisch}$ in beiden Fällen nicht verworfen werden. Der benannte Levene-Test testet zwei Verteilungen auf Varianzgleichheit. Seine Nullhypothese lautet $H_0 = \text{Die Verteilung der Varianzen ist über die Gruppen identisch}$. Mit einer Signifikanz von 0.05 muss bei einem Signifikanzniveau von $\alpha = 0.05$ die H_0 Hypothese nicht verworfen werden.

5.6 Auswertung Testergebnisse

Die Auswertung der Testergebnisse des ersten und des zweiten Tests bezieht sich auf die Tabellen 5.5 und 5.6. Den Tabellen zufolge haben Probanden der EG beim ersten Test 4.67% mehr Wissen wiedergeben können, beim zweiten Test waren es 11.05% mehr Wissen. In

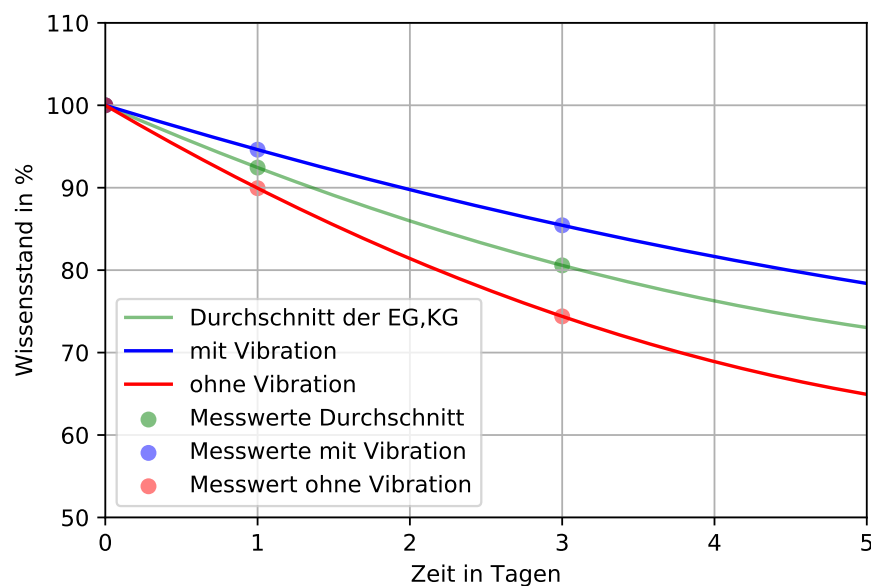


Abbildung 5.6: Zusammenstellung der Testergebnisse einen und drei Tage nach den Lerneinheiten

der Abbildung 5.6 ersichtlich sind drei Graphen, welche die durchschnittlichen Testergebnisse der einzelnen Teilnehmergruppen mittels drei Testwerten zeichnen. Als Ausgangslage wurde der Punkt (0,100) hinzugefügt. Die restlichen Punkte entstammen den besagten Tabellen. Der Graph dient hier nur als visuelle Darstellung der Testergebnisse, die gezeichneten Funktionen geben als Werte außerhalb der Messpunkte keine gesammelten Daten wieder.

Für den ersten Test wurde ähnlich wie für die Lerneinheiten ebenfalls der Mann-Whitney-U-Test mit den ähnlichen Hypothesen und dem gleichen Signifikanzniveau aufgestellt. An diesem Test haben 28 Personen teilgenommen, davon waren 13 aus der KG und 15 aus der EG. Als Nullhypothese gilt hier $H_0 = \text{Die Verteilung von Testergebnissen ist über die Probandengruppen identisch}$. Die Alternativhypothese ist aufgestellt als $H_1 = \text{Die Verteilung von Testergebnissen ist über die Probandengruppen nicht identisch}$. Daraus ergibt sich für den Mann-Whitney-U-Test für unabhängige Stichproben eine exakte Signifikanz von 0.13. Somit kann die H_0 Hypothese nicht verworfen werden.

Auf einen t-Test wurde aufgrund der nicht normalverteilten Datensätze verzichtet (Vgl. Abb. 5.7).

Tests auf Normalverteilung							
Gruppen	Kolmogorov-Smirnov ^a			Shapiro-Wilk			
	Statistik	df	Signifikanz	Statistik	df	Signifikanz	
Lernzeiten	EG	,241	15	,019	,854	15	,020
	KG	,280	13	,006	,808	13	,009

a. Signifikanzkorrektur nach Lilliefors

Abbildung 5.7: Screenshot aus der SPSS Umgebung mit Tests auf Normalverteilung für den ersten Test

Das gleiche Prozedere wie für den ersten Test ergibt für den zweiten Test bei einer Gesamtanzahl an Testergebnissen von $n = 25$ eine exakte Signifikanz von 0.149 für den Mann-Whitney-U-Test. Damit muss für den zweiten Test ebenfalls bei der Nullhypothese geblieben werden. Ebenso kann aufgrund der nur für den KG normalverteilten Datensatz kein t-Test durchgeführt werden (Vgl. Abb. 5.8).

Tests auf Normalverteilung							
Gruppen	Kolmogorov-Smirnov ^a			Shapiro-Wilk			
	Statistik	df	Signifikanz	Statistik	df	Signifikanz	
Lernzeiten	EG	,243	14	,024	,878	14	,054
	KG	,133	11	,200*	,925	11	,365

*. Dies ist eine untere Grenze der echten Signifikanz.

a. Signifikanzkorrektur nach Lilliefors

Abbildung 5.8: Screenshot aus der SPSS Umgebung mit einzelnen Gruppentests auf Normalverteilung für den zweiten Test

5.7 Auswertung Herzfrequenzen

Bezüglich der exploratorischen Auswertung der Herzfrequenzen wird an dieser Stelle ebenfalls ein Mann-Whitney-U-Tests und ein t-Test durchgeführt. Das Signifikanzniveau wird weiterhin auf $\alpha = 0.05$ gehalten. Die Größe der Stichproben für diese Tests können den Tabellen 5.7 für die Lerneinheiten und 5.8 für die anschließenden Wissenstests entnommen werden. Geprüft wird auf die Nullhypothese $H_0 = \text{Die Verteilung der Herzfrequenzen ist über die Probandengruppen}$

pen identisch mit der Alternativhypothese $H_1 = \text{Die Verteilung der Herzfrequenz ist über die Probandengruppen nicht identisch}$. Die Signifikanzen der drei Tests lassen sich der Tabelle 5.9 entnehmen.

Testart	exakte Signifikanz
Lerneinheit	0.106
1.Test	0.275
2.Test	0.374

Tabelle 5.9: Testergebnisse des Mann-Whitney-U-Tests für die Testeinheiten

Für alle drei Tests musste bei der H_0 Hypothese geblieben werden. Ein signifikanter Unterschied konnte nicht gezeigt werden.

Ein Nachweis für die Normalverteilung der Datensätze zu den Lerneinheiten, dem ersten und den zweiten Test, wurde erneut mittels Kolmogorov-Smirnov Test und Shapiro-Wilk Test erbracht.

Tests auf Normalverteilung							
	Gruppe	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
		Statistik	df	Signifikanz	Statistik	df	Signifikanz
Lernzeit	1	,138	9	,200*	,942	9	,606
	2	,152	18	,200*	,907	18	,077

*. Dies ist eine untere Grenze der echten Signifikanz.

a. Signifikanzkorrektur nach Lilliefors

Abbildung 5.9: Screenshot aus der SPSS Umgebung mit einzelnen Gruppentests auf Normalverteilung für die Lerneinheiten

Tests auf Normalverteilung							
	Gruppe	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
		Statistik	df	Signifikanz	Statistik	df	Signifikanz
Lernzeit	1	,174	9	,200*	,932	9	,502
	2	,198	18	,060	,897	18	,050

*. Dies ist eine untere Grenze der echten Signifikanz.

a. Signifikanzkorrektur nach Lilliefors

Abbildung 5.10: Screenshot aus der SPSS Umgebung mit einzelnen Gruppentests auf Normalverteilung für den ersten Test

Tests auf Normalverteilung							
Lernzeit	Gruppe	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
		Statistik	df	Signifikanz	Statistik	df	Signifikanz
Lernzeit	1	,161	8	,200*	,958	8	,789
	2	,176	17	,170	,891	17	,048

*. Dies ist eine untere Grenze der echten Signifikanz.

a. Signifikanzkorrektur nach Lilliefors

Abbildung 5.11: Screenshot aus der SPSS Umgebung mit einzelnen Gruppentests auf Normalverteilung für den zweiten Test

Ausgehend von den drei Abbildungen 5.9, 5.10, 5.11 kann man für alle Datensätze die Normalverteilung voraussetzen. Dabei sind für die Lerneinheiten beider durchgeführter Tests auf Normalverteilung dafür ausgefallen, dass die Nullhypothese, $H_0 = \text{Die Verteilung ist normalverteilt}$, nicht verworfen werden kann (Vgl. Abb 5.9). Für den ersten Test konnte die Nullhypothese mit der gleichen Begründung ebenfalls nicht verworfen werden. Der zweite Test kann aufgrund der Testergebnisse des Kolmogorov-Smirnov Test durchgeführt werden (Vgl. 5.11). Weiterhin werden über den t-Test zunächst ein Test auf Varianzgleichheit durchgeführt. Dieser Test wird durch den Levene-Test mit der Nullhypothese $H_0 = \text{Die Varianzen der Verteilungen sind gleich}$ durchgeführt. Alle Testergebnisse sind in der Tabelle 5.10 festgehalten.

Testart	Levene Signifikanz	t-Test (2-seitig) Signifikanz
Lerneinheit	0.651	0.122
Lerneinheit	ohne Varianzgleichheit	0.146
1.Test	0.505	0.385
1.Test	ohne Varianzgleichheit	0.352
2.Test	0.949	0.488
2.Test	ohne Varianzgleichheit	0.471

Tabelle 5.10: Testergebnisse des t-Tests für die Herzfrequenzen

6 Diskussion und Fazit

In diesem Kapitel werden wichtige Erkenntnisse dieser Arbeit zusammengefasst. Probleme oder Schwierigkeiten, welche während der Applikationsentwicklung oder der Studie aufgetreten sind, werden hier ebenfalls festgehalten. Angeknüpft an die Studienergebnisse wird zum Schluss Stellung zur Forschungsfrage genommen.

Aus der Literatur zum *Assoziativen Lernen* und zum *Passiven haptischen Lernen* wurde deutlich, dass für diese Art des Lernens keine auf den Lerninhalt fokussierte Aufmerksamkeit benötigt wird. Viel mehr entsteht der Lernprozess und die stabile Konsolidierung der Lerninhalte durch wiederholte Darbietung der zu assoziierenden Reize. Auf dieser Basis wurden zur Bearbeitung der Forschungsfrage ein Lernkonzept für eine Experimentalstudie angefertigt. Dieses misst den Einfluss von generierten Vibrationsmustern zu Lerninhalten auf den Lernerfolg (Lernzeit und Konsolidierung). Zur Überprüfung wurden sowohl Lerneinheiten, als auch an verschiedenen Tagen stattfindende Testeinheiten entworfen. Für die Durchführung der Studie wurde eine Applikation entwickelt, welche die gesamte Studiendurchführung abdeckt. Sowohl bei der Entwicklung der Studie als auch bei der Applikationsentwicklung sind vereinzelt Schwierigkeiten aufgetreten.

Ausgehend von den Studien zum *Passiven haptischen Lernen* wurden die Lerninhalte zunächst identisch festgelegt. Im Verlauf der drei Testdurchgänge wurde schnell sichtbar, dass diese für aktives Lernen zu schnell zu erlernen sind. Dies macht aufgrund der Tatsache Sinn, dass aktives Lernen schneller zu sein scheint als passives Lernen, so dass bei allgemein geringerem Lernaufwand keine Unterschiede zwischen EG und KG entstehen können. Die fortlaufende Veränderung zur Erschwerung des Lernprozesses konnte diese bis auf zeitlich erkennbare Unterschiede der Probandengruppen ändern. Die leichten Abänderungen des Studiendesigns waren programmatisch nicht schwer umzusetzen. Viel mehr lagen hier die Schwierigkeiten darin das Smartphone mit der Smartwatch ausreichend zu synchronisieren. Der implementierte MessageService neigte dazu, mehr gesendete Nachrichten zu registrieren als tatsächlich abgesendet wurden. Auch musste aufgrund der synchron laufenden Prozesse auf beiden Endgeräten vermehrt auf Multithreading und deren parallelen Abarbeitung von Methoden gesetzt werden. Ein

weiteres Problem, welches bereits bei der Entwicklung festgestellt wurde, war die TicWatch C2. Dieses Modell stellt zwar einen Sensor für Vibration bereit, jedoch konnte aufgrund der Hardware oder der Version des Betriebssystems keine Amplitudenkontrolle der Vibration bestimmt werden. Dies hat bezüglich des assoziativen Lernens zwei Nachteile. Erstens ist die Reizdiskriminierung [26] lediglich der Länge der einzelnen Vibrationen unterlegen. In diesem Kontext konnten nur zwei verschiedene Reize erstellt werden (Punkt und Strich des Morsezeichens). Eine Veränderung zu mehreren Reizen, die sich in ihrer Vibrationslänge unterscheiden, birgt das Risiko der Reizgeneralisierung. Da bereits in den Studien zum PHL vermehrt über dieses Problem berichtet und eine Lösung für die Darbietung der Reize eines Morsezeichens erarbeitet wurde, wurde ein Versuch für mehr als zwei unterschiedliche Reize nicht weiter ausgeführt. Der zweite Nachteil wird in der Literatur durch den Begriff Assoziationsstärke beschrieben [26]. Mittels dieses Begriffes wird erklärt, dass nicht allein die Kontiguität der Reize ausreicht eine erfolgreiche Assoziation zu bilden, sondern die Assoziationsstärke ebenfalls eine Rolle spielt. Dabei hat jeder wahrgenommene Reiz eine eigene Assoziationsstärke. Eine Hypothese zu diesem genannten Nachteil könnte sein, dass die Assoziationsstärke der Vibration beim Lernen nicht nur von dessen Dauer, sondern auch von dessen Stärke abhängig ist. Die Verwendung einer Smartwatch mit Amplitudenkontrolle hätte demnach zwei Vorteile. Erstens, die Assoziationsstärke könnte für die Vibration größer sein, wenn diese schwach und stark empfunden werden. Zweitens, die Reizdiskriminierung der Vibration bekäme eine zusätzliche Dimension an Möglichkeiten.

Ungeachtet der genannten Schwierigkeiten konnten im Mittel aller Studienergebnisse stets bessere Resultate für die EG festgehalten werden (siehe Kapitel 5). Probanden dieser Gruppe haben im Schnitt schneller gelernt und konnten in den anschließenden Wissenstests bessere Ergebnisse erzielen. Die Auswertung der Lerneinheiten haben ergeben, dass die EG im Schnitt mit 29.47 Minuten 12.89% weniger Lernzeit benötigt haben um (fast) den gesamten Lerninhalt zu lernen. Bei den Testeinheiten haben diese im Schnitt mit 4.67% (1.Test) und 11.05% (2.Test) mehr Wissen als die KG besser abgeschlossen. Ausgehend von den erhobenen Datensätzen konnte zu keinen der angefertigten Signifikanztests eine statistisch signifikant messbare Verbesserung festgestellt werden ($\alpha = 0.05$). Die niedrigsten Signifikanzen für die nicht exploratorische Auswertung wurden für die Testeinheiten festgestellt. Hierbei wurde für den ersten Test eine exakte Signifikanz von 0.13 und für den zweiten Test eine exakte Signifikanz von 0.149 berechnet. Die Nullhypothese zu diesem Tests lautete: $H_0 = \text{Die Verteilung von Testergebnissen ist über die Probandengruppen identisch}$. In anderen Worten bedeutet dies, dass die Wahrscheinlichkeit sich zu irren, wenn man die Nullhypothese verwirft, so hoch ist wie die ausgerechnete Signifikanz. Bei den Ergebnissen gilt zu berücksichtigen, dass diese für kleine Stichproben ausgerechnet wurden. Kleine Stichproben können dazu neigen höhere Signifikanzen anzugeben als größere Stichproben. Die aufgezeigten Verbesserungen sind je-

doch nicht zwangsläufig auf das Lernen mit Vibration zurückzuführen. Es wäre hier durchaus möglich, dass die Personen durch ihr Wissen über die Gruppenzugehörigkeit in ihrer Entscheidung und Motivation und dem damit verbundenen Lernerfolg beeinflusst waren. Eine interessante Erkenntnis aus Kapitel 5.3 ist die beobachtete Zeit, welche die Nutzer für die einzelnen Tests benötigt haben. Hier ist gerade im zweiten Test deutlich zu sehen, dass die Personen mit Vibration deutlich mehr Zeit investiert haben als Ihre korrespondierenden Teilnehmer der Kontrollgruppe. Dies wäre durch eine erhöhte Affektion, aufgrund der zusätzlichen Reize in Form der Vibration, welche den Nutzer eher anregen, sich mit dem gegebenen Input zu beschäftigen, als Hypothese zu begründen.

Eine weitere Erkenntnis ergibt sich aus den gemessenen Herzfrequenzen. Hier wurde unabhängig der Kontrollgruppen beobachtet, dass Probanden mit erhöhter Herzfrequenz mit einer Signifikanz von 0.106 (vgl. Tabelle 5.9) einen besseren Lernerfolg (geringere Lernzeit) gegenüber den Personen mit niedrigerem Puls erzielt haben. Dadurch zeigt sich ein gewisses Potenzial für die Anwendung von Learning Analytics, die weitere Evaluierung erfordert.

Um die Forschungsfrage nun im Hinblick auf die gewonnenen Erkenntnisse der verschiedenen Einheiten zu beantworten, kann festgehalten werden, dass im Mittel aller Studienergebnisse eine Verbesserung erkennbar war, wenn Probanden mit Vibration am Handgelenk gelernt haben. Es soll nicht unerwähnt bleiben, dass ein erneutes Ausführen der Studie unter erweiterten Rahmenbedingungen und mittels einer anderen Smartwatch ratsam wäre. Ebenfalls wäre eine erhöhte Anzahl von Probanden, um für die Signifikanztests zuverlässig die Normalverteilung verwenden zu können, sinnvoll. Des Weiteren wäre es bei einer erneuten Durchführung wichtig, die Tests für alle Teilnehmer zur gleichen Zeit durchzuführen, um eine gleiche Leistungsbereitschaft der Probanden zu gewährleisten. Die Leistungsbereitschaft könnte sowohl die Konsolidierung der Lerninhalte, als auch die Geschwindigkeit beim Lernen beeinflussen. Eine letzte Überlegung, welche in Bezug auf eine neue Studie bedacht werden sollte ist die Körperregion an der die Vibration wahrgenommen wird. Studien zum PHL haben gezeigt, dass Vibrationen an verschiedenen Körperregionen verwendet werden können, um das Lernen zu fördern. Beispielsweise könnte man den Einfluss der Vibration am linken bzw rechten Handgelenk prüfen. In der durchgeführten Studie sowie in anderen Studien konnte gezeigt werden, dass Lernen unter Verwendung von Vibration ein Verfahren mit viel Potenzial für die Bildungsforschung ist.

Tabellenverzeichnis

5.1	Durchschnittliche Lernzeiten der Probandengruppen für eine Korrektheit von 96% oder mehr	39
5.2	Durchschnittliche Lernzeiten der Probandengruppen, welche 100% der Paarungen richtig hatten	39
5.3	Testergebnisse des ersten Tests	40
5.4	Testergebnisse des zweiten Tests	40
5.5	Testergebnisse des ersten Tests zuzüglich der zwei zunächst ausgeschlossenen Probanden	40
5.6	Testergebnisse des zweiten Tests zuzüglich der zwei zunächst ausgeschlossenen Probanden	40
5.7	Testergebnisse des zweiten Tests zuzüglich zwei zunächst ausgeschlossenen Probanden	41
5.8	Testergebnisse des ersten Tests zuzüglich zwei zunächst ausgeschlossenen Probanden	42
5.9	Testergebnisse des Mann-Whitney-U-Tests für die Testeinheiten	47
5.10	Testergebnisse des t-Tests für die Herzfrequenzen	48

Abbildungsverzeichnis

2.1	Schematische Darstellung der klassischen Konditionierung anhand des Beispiels 'der pavlovsche Hund' [31]	4
2.2	Screenshot der Studienprozedur aus der Studie [22] zum Assoziativen Lernen .	6
2.3	Von Android Studio automatisch erstellte Projekt	10
2.4	Lebenszyklus einer Android Activity [4]	12
2.5	Lebenszyklus eines Android Fragments [7]	13
3.1	Schematische Darstellung des Studienablaufs.	20
4.1	Fertige Projektübersicht der Vibning Applikation	22
4.2	Activities und Fragmente der 'mobile' Applikation	25
4.3	Screenshots Login & Registrierungs Layout	27
4.4	Screenshot des Applikations Menu der MainActivity und dessen AppUsageFragment	28
4.5	Layouts der Vibning Applikation für Test und Lerneinheiten	32
4.6	Screenshot der schematischen Architektur des PreknowledgeFragments	33
5.1	Zusammenstellung aller Testsequenzen der Lerneinheiten	38
5.2	Zusammenstellung der Testsequenzen der verschiedenen Teilnehmergruppen .	40
5.3	Lineare Regression der Testsequenzen	43
5.4	Screenshot aus der SPSS Umgebung mit Tests auf Normalverteilung für die 95% Kondition einzelner Gruppen	44
5.5	Screenshot aus der SPSS Umgebung mit Tests auf Normalverteilung für die 100% Kondition einzelner Gruppen	44
5.6	Zusammenstellung der Testergebnisse einen und drei Tage nach den Lerneinheiten	45
5.7	Screenshot aus der SPSS Umgebung mit Tests auf Normalverteilung für den ersten Test	46

5.8	Screenshot aus der SPSS Umgebung mit einzelnen Gruppentests auf Normalverteilung für den zweiten Test	46
5.9	Screenshot aus der SPSS Umgebung mit einzelnen Gruppentests auf Normalverteilung für die Lerneinheiten	47
5.10	Screenshot aus der SPSS Umgebung mit einzelnen Gruppentests auf Normalverteilung für den ersten Test	47
5.11	Screenshot aus der SPSS Umgebung mit einzelnen Gruppentests auf Normalverteilung für den zweiten Test	48
1	Screenshot der UserData Tabellenstruktur	63
2	Screenshot der PreknowledgeSession Tabellenstruktur	63
3	Screenshot der LearnSession Tabellenstruktur	64
4	Screenshot der TestSession Tabellenstruktur	64
1	AccountFragment und SettingsFragment Layouts der Vibning Applikation	65
2	StartDialog, InformationDialog und MonitorFragment Layouts der Vibning Applikation	65
1	Zusammenstellung aller Testsequenzen der Lerneinheit für einzelne Teilnehmergruppen	67
2	Zusammenstellung aller Testsequenzen der Lerneinheiten exkludiert nicht fertig gewordenene Partizipanten	67
3	Zusammenstellung aller Testsequenzen der Lerneinheiten exkludiert nicht fertig gewordenene Partizipanten für einzelne Teilnehmergruppen	68
4	Lineare Regression mit polynomiellen Features 3. Grades, Logarithmische Regression der Testsequenzen	68

Literatur

- [1] *Android Beispiel Theme*. Eingesehen am 15.08.2019. URL: <https://developer.android.com/guide/topics/ui/look-and-feel/themes>.
- [2] *Android Developer Homepage - Keep your app visible on Wear*. Eingesehen am 21.09.2019. URL: <https://developer.android.com/training/wearables/apps/always-on>.
- [3] *Android Developer Klasse PowerManager*. Eingesehen am 21.09.2019. URL: <https://developer.android.com/training/scheduling/wakelock>.
- [4] *Android Developers Activity*. Eingesehen am 12.08.2019. URL: <https://developer.android.com/reference/android/app/Activity>.
- [5] *Android Developers Application*. Eingesehen am 15.08.2019. URL: <https://developer.android.com/guide/topics/manifest/application-element>.
- [6] *Android Developers BottomAppBar*. Eingesehen am 16.08.2019. URL: <https://developer.android.com/reference/com/google/android/material/bottomappbar/BottomAppBar>.
- [7] *Android Developers Fragment*. Eingesehen am 13.08.2019. URL: <https://developer.android.com/guide/components/fragments>.
- [8] *Android Developers Homepage*. Eingesehen am 14.08.2019. URL: <https://developer.android.com/>.
- [9] *Android Developers Homepage Services*. Eingesehen am 16.08.2019. URL: <https://developer.android.com/guide/components/services>.
- [10] *Android Developers Intent*. Eingesehen am 13.08.2019. URL: <https://developer.android.com/guide/components/intents-filters>.
- [11] *Android Developers Navigation Drawer*. Eingesehen am 16.08.2019. URL: <https://developer.android.com/guide/navigation/navigation-ui>.
- [12] *Android Developers NumberPicker*. Eingesehen am 08.09.2019. URL: <https://developer.android.com/reference/android/widget/NumberPicker>.
- [13] *Android Developers SensorManager*. Eingesehen am 08.09.2019. URL: <https://developer.android.com/reference/android/hardware/SensorManager>.

- [14] *Android Homepage Intent-Filter und Intent*. Eingesehen am 15.08.2019. URL: <https://developer.android.com/guide/components/intents-filters.html>.
- [15] *Android Homepage Receiver*. Eingesehen am 15.08.2019. URL: <https://developer.android.com/guide/topics/manifest/receiver-element>.
- [16] *Android Permissions Overview*. Eingesehen am 15.08.2019. URL: <https://developer.android.com/guide/topics/permissions/overview>.
- [17] *Android Studio Entwicklungsumgebung*. Eingesehen am 12.08.2019. URL: <https://developer.android.com>.
- [18] *Android Wiki Intents*. Eingesehen am 15.08.2019. URL: <https://www.droidwiki.org/wiki/Intent>.
- [19] *AndroidWiki Homepage Manifest Artikel*. Eingesehen am 14.08.2019. URL: https://www.droidwiki.org/wiki/Manifest#cite_note-1.
- [20] Andras Arato, Norbert Markus und Zoltan Juhasz. „Teaching morse language to a deaf-blind person for reading and writing SMS on an ordinary vibrating smartphone“. In: *International Conference on Computers for Handicapped Persons*. Springer. 2014, S. 393–396.
- [21] Caterina Breitenstein. „Lernen aus neurowissenschaftlicher Sicht: von der Assoziation zur Kognition“. In: *Diskurs Kindheits-und Jugendforschung/Discourse. Journal of Childhood and Adolescence Research* 7.4 (2012).
- [22] Caterina Breitenstein und Stefan Knecht. „Development and validation of a language learning model for behavioral and functional-imaging studies“. In: *Journal of neuroscience methods* 114.2 (2002), S. 173–179.
- [23] Caterina Breitenstein u. a. „Word learning can be achieved without feedback: implications for aphasia therapy“. In: *Restorative neurology and neuroscience* 22.6 (2004), S. 445–458.
- [24] *Dokumentation SharedPreference Klasse*. Eingesehen am 05.09.2019. URL: <https://developer.android.com/reference/android/content/SharedPreferences>.
- [25] *Free Logo Design Homepage*. Eingesehen am 29.09.2019. URL: <https://de.freelogodesign.org/>.
- [26] Halpern Gazzaniga Heatherton. *Psychologie*. ISBN: 978-3-621-28295-6. 2017.
- [27] *Homepage Material Design Android*. Eingesehen am 19.09.2019. URL: <https://material.io/develop/android/>.
- [28] Kevin Huang, Ellen Yi-Luen Do und Thad Starner. „PianoTouch: A wearable haptic piano instruction system for passive learning of piano skills“. In: *2008 12th IEEE International Symposium on Wearable Computers*. IEEE. 2008, S. 41–44.

-
- [29] Kevin Huang u. a. „Mobile music touch: mobile tactile stimulation for passive learning“. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM. 2010, S. 791–800.
- [30] IBM SPSS-Software. Eingesehen am 28.09.2019. URL: <https://www.ibm.com/de-de/analytics/spss-statistics-software>.
- [31] Andrea Kiesel und Iring Koch. „Lernen“. In: *Grundlagen der Lernpsychologie*. Wiesbaden (2012).
- [32] Robyn Kim u. a. „Testing assumptions of statistical learning: is it long-term and implicit?“ In: *Neuroscience letters* 461.2 (2009), S. 145–149.
- [33] Simone Kimpeler. „Lernen mit Online-Medien–E-Learning“. In: *Handbuch Online-Kommunikation*. Springer, 2010, S. 364–384.
- [34] Daniel Kohlsdorf und Thad Starner. „Mobile music touch: The effect of primary tasks on passively learning piano sequences“. In: *International Symposium on Wearable Computers (ISWC) 2010*. IEEE. 2010, S. 1–8.
- [35] Jutta Kray und Ulman Lindenberger. „3.1 Fluide Intelligenz“. In: *Entwicklungspsychologie der Lebensspanne: Ein Lehrbuch* (2007), S. 194.
- [36] Seungyon Claire Lee und Thad Starner. „BuzzWear: alert perception in wearable tactile displays on the wrist“. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 2010, S. 433–442.
- [37] Granit Luzhnica, Eduardo Veas und Caitlyn Seim. „Passive haptic learning for vibrotactile skin reading“. In: *Proceedings of the 2018 ACM International Symposium on Wearable Computers*. ACM. 2018, S. 40–43.
- [38] Matplotlib Python. URL: <https://matplotlib.org/>.
- [39] Methodenberatung Uni Zürich, Kolmogorov-Smirnov-Test auf Normalverteilung. Eingesehen am 30.09.2019. URL: https://www.methodenberatung.uzh.ch/de/datenanalyse_spss/unterschiede/proportionen/pearsonuntersch.html#36.
- [40] Methodenberatung Uni Zürich, t-Test für unabhängige Stichproben. Eingesehen am 30.09.2019. URL: https://www.methodenberatung.uzh.ch/de/datenanalyse_spss/unterschiede/zentral/ttestunabh.html.
- [41] Methodenberatung Universität Zürich, Mann-Whitney-U-Test. Eingesehen am 28.09.2019. URL: https://www.methodenberatung.uzh.ch/de/datenanalyse_spss/unterschiede/zentral/mann.html.
- [42] Dan Morris u. a. „Haptic feedback enhances force skill learning“. In: *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*. IEEE. 2007, S. 21–26.

- [43] Hugo Nicolau u. a. „UbiBraille: designing and evaluating a vibrotactile Braille-reading device“. In: *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM. 2013, S. 23.
- [44] *Numpy Dokumentation Python*. URL: <https://numpy.org/>.
- [45] Uwe Post. *Android-Apps entwickeln für Einsteiger*. Rheinwerk Computing, 2018.
- [46] *Programmiersprache Java*. Eingesehen am 12.08.2019. URL: <https://www.java.com/de/>.
- [47] *Programmiersprache Python*. Eingesehen am 28.09.2019. URL: <https://www.python.org/>.
- [48] *Scipy Dokumentation Python*. URL: <https://docs.scipy.org/doc/>.
- [49] Caitlyn Seim u. a. „Tactile taps teach rhythmic text entry: passive haptic learning of morse code“. In: *Proceedings of the 2016 ACM International Symposium on Wearable Computers*. ACM. 2016, S. 164–171.
- [50] Caitlyn Seim u. a. „Towards haptic learning on a smartwatch“. In: *Proceedings of the 2018 ACM International Symposium on Wearable Computers*. ACM. 2018, S. 228–229.
- [51] *Singleton Klasse Beschreibung für Java*. Eingesehen am 07.09.2019. URL: <https://www.theserverside.de/singleton-pattern-in-java/>.
- [52] *Statistischer Vergleich von zwei Gruppen*. URL: <https://statistik-und-beratung.de/2013/07/statistischer-vergleich-von-zwei-gruppen/>.
- [53] Laampertswalde (2018) Stoba Druck. „D21 Digital Index 2018/2019: Jährliches Lagebild zur Digitalen Gesellschaft“. In: ().
- [54] *Volley Http Requests Android*. Eingesehen am 08.09.2019. URL: <https://developer.android.com/training/volley/simple>.

Appendices

Anhang 1: Datenbanktabellen

#	Name	Typ	Kollation	Attribute	Null	Standard	Kommentare	Extra	Aktion
<input type="checkbox"/>	1	userID	int(255)		Nein	kein(e)		AUTO_INCREMENT	Bearbeiten Löschen Mehr
<input type="checkbox"/>	2	surname	varchar(255)	utf8_unicode_ci	Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	3	name	varchar(255)	utf8_unicode_ci	Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	4	birthday	varchar(255)	utf8_unicode_ci	Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	5	gender	varchar(255)	utf8_unicode_ci	Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	6	conditionterm	varchar(255)	utf8_unicode_ci	Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	7	login	varchar(255)	utf8_unicode_ci	Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	8	password	varchar(255)	utf8_unicode_ci	Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	9	participationlevel	varchar(255)	utf8_unicode_ci	Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	10	regTime	datetime		Nein	CURRENT_TIMESTAMP			Bearbeiten Löschen Mehr

☐ Alle auswählen
 markierte: Anzeigen Bearbeiten Löschen Primärschlüssel Unique Index Volltext Volltext

Abbildung 1: Screenshot der UserData Tabellenstruktur

#	Name	Typ	Kollation	Attribute	Null	Standard	Kommentare	Extra	Aktion
<input type="checkbox"/>	1	PrelID	int(11)		Nein	kein(e)		AUTO_INCREMENT	Bearbeiten Löschen Mehr
<input type="checkbox"/>	2	userID	int(255)		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	3	timeStart	mediumtext	utf8_unicode_ci	Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	4	timeEnd	mediumtext	utf8_unicode_ci	Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	5	wrongItems	mediumtext	utf8_unicode_ci	Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	6	rightItems	mediumtext	utf8_unicode_ci	Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	7	correctness	mediumtext	utf8_unicode_ci	Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	8	heartRate	mediumtext	utf8_unicode_ci	Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	9	heartRateTimeLogs	mediumtext	utf8_unicode_ci	Nein	kein(e)			Bearbeiten Löschen Mehr

☐ Alle auswählen
 markierte: Anzeigen Bearbeiten Löschen Primärschlüssel Unique Index Volltext Volltext

Abbildung 2: Screenshot der PreknowledgeSession Tabellenstruktur

#	Name	Typ	Kollation	Attribute	Null	Standard	Kommentare	Extra	Aktion
<input type="checkbox"/>	1 learnID	int(255)			Nein	kein(e)		AUTO_INCREMENT	Bearbeiten Löschen Mehr
<input type="checkbox"/>	2 userID	int(255)			Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	3 timeStartLearn	mediumtext	latin1_swedish_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	4 timeEndLearn	mediumtext	latin1_swedish_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	5 timeStartTest	mediumtext	latin1_swedish_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	6 timeEndTest	mediumtext	latin1_swedish_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	7 heartRateLearn	mediumtext	latin1_swedish_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	8 heartRateTimeLogsLearn	mediumtext	latin1_swedish_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	9 heartRateTest	mediumtext	latin1_swedish_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	10 heartRateTimeLogsTest	mediumtext	latin1_swedish_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	11 correctnessTest	mediumtext	latin1_swedish_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	12 rightItemTest	mediumtext	latin1_swedish_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	13 wrongItemTest	mediumtext	latin1_swedish_ci		Nein	kein(e)			Bearbeiten Löschen Mehr

☐ Alle auswählen
 markierte: Anzeigen Bearbeiten Löschen Primärschlüssel Unique Index Volltext Volltext

Abbildung 3: Screenshot der LearnSession Tabellenstruktur

#	Name	Typ	Kollation	Attribute	Null	Standard	Kommentare	Extra	Aktion
<input type="checkbox"/>	1 testID	int(255)			Nein	kein(e)		AUTO_INCREMENT	Bearbeiten Löschen Mehr
<input type="checkbox"/>	2 userID	int(255)			Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	3 testSessionNumber	varchar(255)	utf8_unicode_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	4 timeStart	mediumtext	utf8_unicode_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	5 timeEnd	mediumtext	utf8_unicode_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	6 wrongItems	mediumtext	utf8_unicode_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	7 rightItems	mediumtext	utf8_unicode_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	8 correctness	mediumtext	utf8_unicode_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	9 heartRate	mediumtext	utf8_unicode_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
<input type="checkbox"/>	10 heartRateTimeLogs	mediumtext	utf8_unicode_ci		Nein	kein(e)			Bearbeiten Löschen Mehr

☐ Alle auswählen
 markierte: Anzeigen Bearbeiten Löschen Primärschlüssel Unique Index Volltext Volltext

Abbildung 4: Screenshot der TestSession Tabellenstruktur

Anhang 2: Layouts der Vibning Applikation

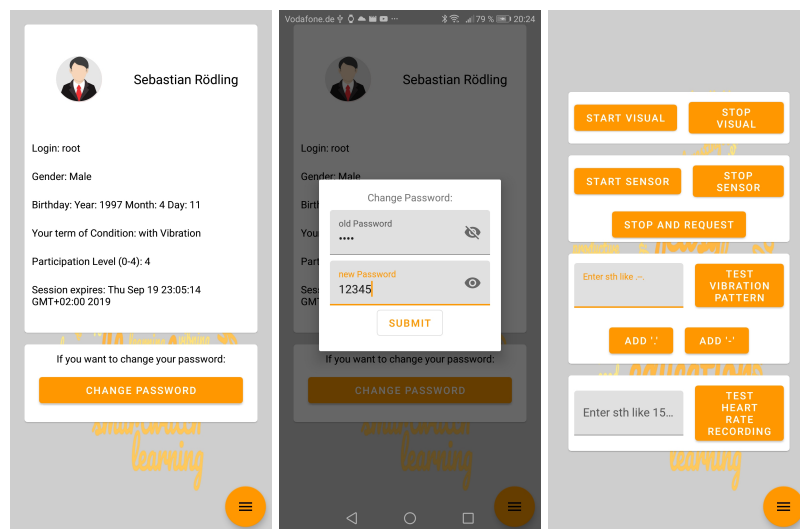


Abbildung 1: AccountFragment und SettingsFragment Layouts der Vibning Applikation

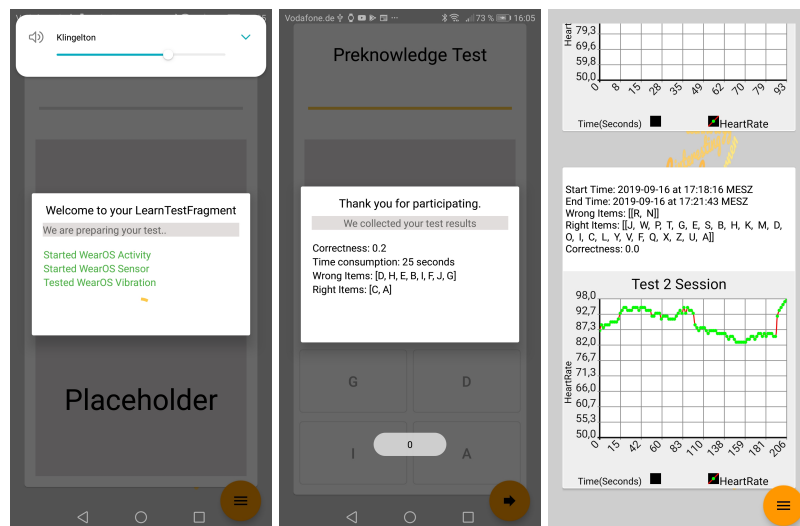


Abbildung 2: StartDialog, InformationDialog und MonitorFragment Layouts der Vibning Applikation

Anhang 3: Evaluierung der Studie

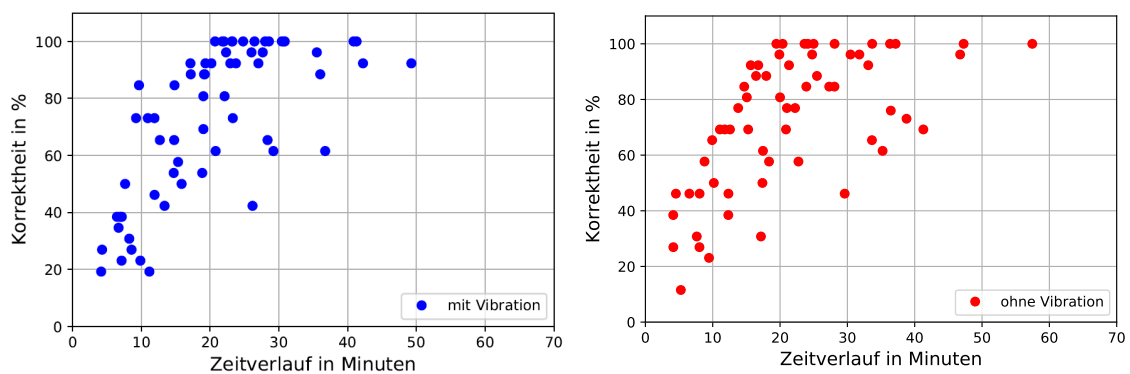


Abbildung 1: Zusammenstellung aller Testsequenzen der Lerneinheit für einzelne Teilnehmergruppen

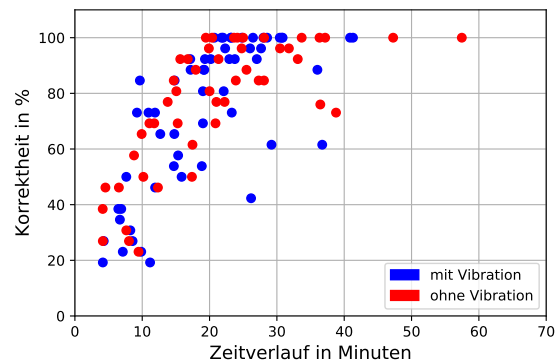


Abbildung 2: Zusammenstellung aller Testsequenzen der Lerneinheiten exkludiert nicht fertig gewordenene Partizipanten

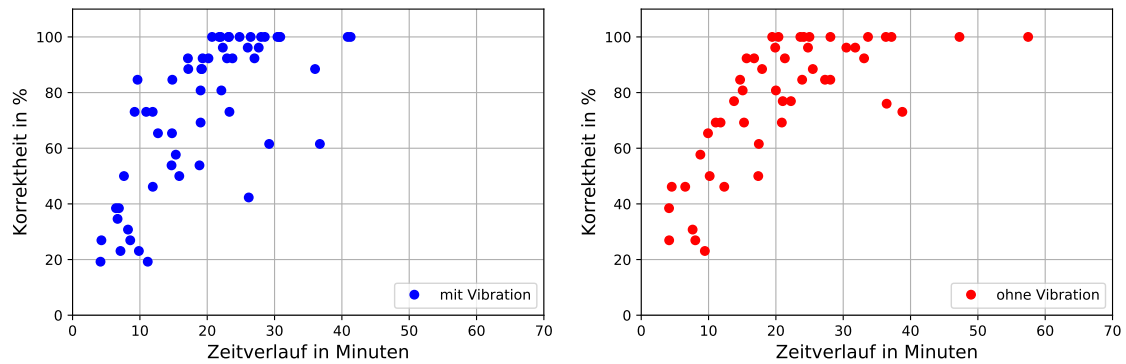


Abbildung 3: Zusammenstellung aller Testsequenzen der Lerneinheiten exkludiert nicht fertig gewordenene Partizipanten für einzelne Teilnehmergruppen

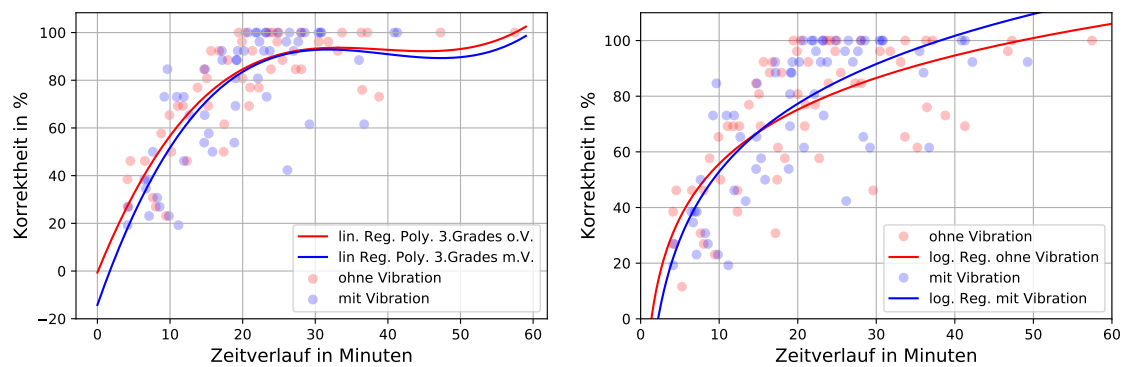


Abbildung 4: Lineare Regression mit polynomiellen Features 3. Grades, Logarithmische Regression der Testsequenzen