

Bachelorarbeit

Entwicklung eines Seniorenhandys auf der Basis einer Android App

Daniel Siebrecht

11.12.2015

Albert-Ludwigs-Universität Freiburg im Breisgau

Technische Fakultät
Institut für Informatik

Bearbeitungszeitraum

11.09.2015 – 11.12.2015

Gutachter

Prof. Dr. Hannah Bast

Betreuer

Dr. Harald Fischer

Erklärung

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen/Hilfsmittel verwendet habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, bereits für eine andere Prüfung angefertigt wurde.

Ort, Datum

Unterschrift

Kurzfassung

Die vorliegende Arbeit beschäftigt sich damit, existierende Smartphones durch eine Anwendung (App) auf ein Minimum an Funktionalität zu reduzieren, um die Bedienung für Senioren und Menschen mit Einschränkungen zu vereinfachen. Dazu wurde untersucht, welche sensorischen, körperlichen und geistigen Einschränkungen die Bedienung eines Smartphones beeinflussen können und welche Designaspekte diesen bei der Gestaltung der App entgegenwirken können. Außerdem wurde betrachtet, welche Anforderungen Senioren an ein Handy haben.

Bezugnehmend auf diese Ergebnisse wurde eine App entwickelt, welche die Benutzeroberfläche von Android durch eine einfache Darstellung ersetzt. Die App verfügt außerdem über einige Assistenzfunktionen für den Nutzer, auf welche eine autorisierte Person über ihr eigenes Handy und eine separate App zugreifen kann. Zur Evaluation der App wurde überprüft, inwieweit die herausgearbeiteten Designaspekte umgesetzt wurden. Abschließend wurde getestet, ob Senioren ohne Smartphone-Kenntnisse die App bedienen können.

Danksagung

Mein besonderer Dank gilt Frau Prof. Dr. Hannah Bast, die mir ermöglicht hat, meine Bachelorarbeit in der Lohmann & Birkner Mobile Services GmbH (L&B) zu schreiben.

Weiterhin möchte ich mich bei Dr. Harald Fischer für die tolle Unterstützung und die gute Betreuung meiner Arbeit bedanken. Dies gilt auch allen Mitarbeitern der L&B, die bei Fragen jederzeit ein offenes Ohr für mich hatten.

Nicht zu vergessen gilt mein Dank allen, die an dieser Arbeit mitgewirkt haben, ob als Testperson, Korrekturleser oder als Motivator, und bei allen, die mich während meines kompletten Studiums unterstützt und motiviert haben.

Vielen Dank!

Inhaltsverzeichnis

1. Einleitung	1
1.1. Ziel der Arbeit	1
1.2. Aufbau der Arbeit.....	2
2. Stand der Forschung und Technik	3
2.1. App Entwicklung für Senioren	3
2.1.1. Altersbedingte Einschränkungen	3
2.1.2. Erwartungen von Senioren an Handys	5
2.1.3. User Interface für Senioren.....	6
2.2. Vorhandene Seniorenhandys	8
2.2.1. Großtastenhandys	8
2.2.2. Notruftelefone	10
2.2.3. Senioren-Smartphones	11
2.3. Android Launcher für Senioren	13
2.3.1. Simple Senior Phone	13
2.3.2. Koala Phone Launcher	14
2.3.3. Fontrillo - The easy launcher	16
2.3.4. Phonotto Easy Phone Launcher	18
2.4. Zusammenfassung der Analyse	19
3. App Entwicklung für Android	21
3.1. Aufbau von Android Projekten.....	21
3.2. Activities	23
3.3. Intent-Filter und Intents	25
3.4. Broadcast Receiver	26
3.5. Services	27
4. Entwurf und Implementierung	29
4.1. Seniorenhandy-App.....	29
4.1.1. Launcher Activity.....	30
4.1.2. Konfigurationsmenü.....	34
4.1.3. Bildschirme für ein- und ausgehende Anrufe.....	35

4.1.4. Sperrbildschirm.....	38
4.1.5. Fernwartung.....	40
4.1.6. Freisprechen.....	42
4.1.7. Eigene Rufnummer wählen	42
4.1.8. Funktion „Anruf vorbereiten“	43
4.2. Konfigurations-App.....	45
4.2.1. Main Activity und Einstellungen	45
4.2.2. Konfigurations-SMS.....	46
5. Evaluation	48
5.1. Umsetzung der Anforderungen	48
5.2. Verhalten auf verschiedenen Geräten.....	50
5.3. Betriebstest	54
5.3.1. Testperson 1, Ruth, 83 Jahre.....	55
5.3.2. Testperson 2, Edith, 83 Jahre.....	57
5.3.3. Zusammenfassung des Betriebstests.....	58
6. Schlussbetrachtung	59
6.1. Ausblick	60
Literaturverzeichnis	61

1. Einleitung

Smartphones sind aus unserem Alltag nicht mehr wegzudenken. Ihre Leistungsfähigkeit und ihr Funktionsumfang wachsen in rasender Geschwindigkeit, gleichzeitig aber auch ihre Komplexität. Touchscreens, wie sie in Smartphones verbaut sind, bieten durch die direkte Eingabe auf dem Display viele Vorteile gegenüber älteren Handys mit einem kleinen Display und mechanischen Tasten. Insbesondere Senioren oder Menschen mit Einschränkungen fällt die Benutzung eines Smartphones aufgrund des großen Funktionsumfangs und der hohen Komplexität der Bedienung sehr schwer, oder sie wollen sich nicht in die Bedienung eines Handys einarbeiten.

1.1. Ziel der Arbeit

In der vorliegenden Arbeit soll eine Seniorenhandy-App entwickelt werden, welche die komplexe Benutzeroberfläche eines Smartphones durch eine vereinfachte Darstellung ersetzt und dem Nutzer nur die wichtigen Funktionen übersichtlich bereitstellt. Dafür wird untersucht, welche Erwartungen Senioren an ein Handy haben und wie eine App für Senioren gestaltet werden sollte. Die fertige App soll älteren und eingeschränkten Menschen die Verwendung eines Smartphones ermöglichen, ohne sich mit der Bedienung auseinandersetzen zu müssen.

Ergänzt wird die Seniorenhandy-App durch diverse Assistenzfunktionen, auf welche eine autorisierte Person von einer separaten App auf einem anderen Gerät aus Zugriff hat. Die Kommunikation soll dabei ohne Internetverbindung stattfinden, damit eine einfache SIM-Karte mit Guthaben verwenden werden kann.

1.2. Aufbau der Arbeit

Die Arbeit ist in sechs Kapitel aufgeteilt. Das zweite Kapitel (Stand der Forschung und Technik) beschäftigt sich mit den altersbedingten Einschränkungen und den Erwartungen von Senioren an Handys. Außerdem werden Anforderungen an die Benutzeroberfläche einer App für Senioren aufgezeigt und verschiedene vorhandene Lösungen dargestellt, zu denen sowohl komplette Geräte, wie auch App-basierte Lösungen zählen.

Im dritten Kapitel sollen verschiedenen Grundlagen der Entwicklung für Android vorgestellt werden, bevor im vierten Kapitel auf den Entwurf und die Implementierung der Apps eingegangen wird.

Das vorletzte Kapitel der Arbeit beschreibt die Evaluation der Seniorenhandy-App. Abschließend folgt mit dem sechsten Kapitel die Schlussbetrachtung, die auch einen Ausblick auf mögliche Erweiterungen gibt.

2. Stand der Forschung und Technik

Den Problemen, die Senioren und Menschen mit Einschränkungen bei der Bedienung von Smartphones haben, haben sich bereits viele Forscher, Unternehmen und Entwickler angenommen und stellen verschiedene Lösungen bereit, um den Umgang mit Smartphones zu vereinfachen. Nachfolgend wird ein Überblick über die Forschung und Lösungen geboten.

2.1. App Entwicklung für Senioren

Dadurch, dass viele Menschen mit zunehmendem Alter Einschränkungen beim Sehen, Hören, Tasten oder auch der Motorik haben [1], sollten diese Einschränkungen im Entwurfsprozess einer Seniorenapp beachtet werden. Ebenso wichtig ist die Frage, welche Funktionen eines Handys Senioren überhaupt benötigen.

Mit diesen Vorüberlegungen und den daraus folgenden Konsequenzen beschäftigen sich die folgenden Unterkapitel.

2.1.1. Altersbedingte Einschränkungen

Ab einem Alter von 65 Jahren steigt die Häufigkeit von visuellen Einschränkungen signifikant [2]. Ältere Personen haben Schwierigkeiten Farben zu unterscheiden und ihre Augen reagieren empfindlich auf Spiegelungen. Insgesamt dauert es länger, das visuelle Geschehen zu verarbeiten oder Objekte zu fokussieren [3]. Es ist deshalb von Vorteil, dass Geräte für ältere Menschen über helle Displays mit hohem Kontrastverhältnis verfügen. Außerdem sollten sie wenig spiegeln. Bei der Entwicklung der Software ist die Reduzierung auf wenige große

Elemente mit guten Farbkontrasten sehr wichtig. Animationen sollten nicht verwendet werden.

Weiterhin wird unter anderem durch den Verlust von Haarzellen im Ohr das Hörvermögen eingeschränkt [1]. Es wird demzufolge schwerer, Geräusche zu lokalisieren und leise Geräusche wahrzunehmen. Geräte für Senioren sollten deshalb über eine Freisprechfunktion verfügen oder auf eine hohe Lautstärke beim Telefonieren eingestellt sein.

Da im Alter die Flexibilität einzelner Körperteile nachlässt [1] und somit auch die Fingerfertigkeit, sollte auf Wischgesten und andere schwierige Eingabemethoden verzichtet werden. Nicht zu vergessen ist eine Verlangsamung der Informationsverarbeitung [1]. Neue Informationen zu speichern und gespeicherte Informationen abzurufen, benötigt mehr Zeit und es fällt schwer, sich auf eine Aufgabe zu konzentrieren. Dies verhindert das schnelle Erlernen einer neuen Bedienung. Gleichzeitig wird diese schneller wieder vergessen. Die Bedienung sollte darum selbsterklärend und einfach zu erlernen sein.

Viele der Probleme bei der Benutzung eines Smartphones lassen sich wie oben beschrieben softwareseitig lösen. Diese Lösungsvorschläge sind in Tabelle 2.1 aufgeführt.

Problem	Lösung
Farben unterscheiden	hohe Kontraste, kräftige Farben
Spiegelung	helle Displaybeleuchtung
Fokussierung	wenige große Elemente, keine Animationen
Hörvermögen	Freisprechfunktion, hohe Lautstärkeeinstellung
Fingerfertigkeit	keine Wischgesten, große Bedienelemente
Informationsverarbeitung	selbsterklärende Bedienung

Tabelle 2.1: Probleme von Senioren und mögliche Lösungen

2.1.2. Erwartungen von Senioren an Handys

Die Entwicklung im Bereich von mobilen Geräten erfolgt überaus schnell. Ständig gibt es neue Anwendungen, die den Funktionsumfang der Geräte erweitern. Bei der Entwicklung eines Seniorenhandys ist daher angebracht zu betrachten, welche Funktionen Senioren überhaupt benötigen.

K. Chen (2013) [4] hat dazu in einer Umfrage mit 100 Personen ab einem Alter von 60 Jahren die wichtigsten Mobiltelefon-Anwendungen für ältere Menschen ermittelt. Die Ergebnisse sind in Tabelle 2.2 aufgeführt. Dabei gaben 97 Prozent der Personen an, dass ihnen die Telefoniefunktion wichtig ist. 76 Prozent fanden eine Datums- und Uhrzeitanzeige nützlich und 69 Prozent nutzten das Adressbuch. 45 Prozent gaben an, dass ein Notrufknopf wichtig ist. Andere Funktionen wie E-Mail, Taschenrechner, Kamera oder Textnachrichten (SMS) wurden eher als unwichtig angegeben.

Mobiltelefon-Anwendung	Häufigkeit
Anrufen	97
Datum und Uhrzeit anzeigen	76
Adressbuch	69
Notfalltaste	45
Kamera	31
Wecker	29
Taschenrechner	24
Textnachrichten	20
Minispiele	15
Kalender	15

Tabelle 2.2: Top 10 der wichtigsten Mobiltelefonanwendungen für Personen über 60 Jahren, 100 Personen wurden befragt. Quelle: [4]

Nach einer Studie von Sri Kurniawan (2006) [5] fühlen sich ältere Menschen sicherer, wenn sie ein Handy bei sich tragen. Ein Hauptargument für ein Handy sei, dass in dringenden Fällen oder Notsituation sofort um Hilfe gerufen werden kann. Die meiste Kommunikation über das Handy fand mit der Familie statt. Zu den Funktionen, die sich die Teilnehmer wünschten, gehörte unter anderem eine Notfalltaste, eine über einen einzigen Knopf zu aktivierende Bildschirmsperre und ein einfacher Home-Bildschirm mit nur vier Befehlen. Da sich die Studie von Kurniawan auf die Nutzung von klassischen Handys mit mechanischen Tasten bezieht, lässt sie sich nur in Teilen auf den heutigen Stand der Technik übertragen. Dennoch bietet sie gute Anhaltspunkte und ähnelt den Ergebnissen von Chen (2013).

2.1.3. User Interface für Senioren

Tanid Phiriyapokanon (2011) [6] hat in seiner Masterarbeit mögliche Richtlinien für User Interfaces für Senioren erarbeitet, die für die hier vorgestellte Arbeit interessant sind. Er schlägt vor, wenig genutzte Funktionen zu entfernen und die Elemente auf etwa fünf pro Seite zu reduzieren, um das Interface weniger komplex zu gestalten. Damit die Bedienung klar verständlich ist, soll jede Taste nur eine einzige Funktion ausführen, nicht mehrere. Außerdem soll so oft wie möglich eine Bestätigung angezeigt werden oder der Nutzer anderweitige Rückmeldung erhalten. Des Weiteren wird vorgeschlagen Farben zu nutzen, wenn dem Benutzer Inhalte präsentiert werden. Ein konsistentes Interface, auch über verschiedene Versionen der Software sollte erarbeitet werden.

Phiriyapokanon (2011) schlägt außerdem vor, die Richtlinien, welche von vielen Soft- und Hardwareherstellern für die Interface Gestaltung bereitgestellt werden, für Senioren leicht abzuändern. Die Größe eines Buttons soll nach ihm nicht kleiner als 16,5mm sein, der Abstand zwischen Buttons soll zwischen 3,15mm und 12,7mm liegen. Da Text ein sehr wichtiges Element ist, sollte dieser möglichst gut lesbar dargestellt werden. Dazu wird am Besten eine serifenlose Schrift mit Schriftgröße 14 (etwa 5mm auf einem 72dpi Bildschirm) oder größer verwendet. Als Farbe sollte

Schwarz auf weißem Hintergrund gewählt werden. Schräg platzierte oder sich bewegende Texte sollten ebenso vermieden werden wie Scrollbalken.

Viele der Punkte stimmen mit den Richtlinien von Panayiotis Zaphiris et al. (2005) [7] überein, die sich allerdings auf Webdesign beziehen. Auch hier werden größere Schaltflächen und klare Rückmeldungen verlangt. Auf Animationen und Scrollbalken soll ebenso verzichtet werden, wie auf sich bewegende Texte. Auch Zaphiris et al. schlagen für die Textgestaltung hohe Kontraste zwischen Hintergrund und Text, sowie eine serifenlosen Schrift vor, die für den Fließtext auf Webseiten 12-14 Punkte betragen soll.

Steven Hoober (2013) [8] hat beobachtet, wie Nutzer ihr Handy im Alltag in der Hand halten. Abbildung 2.1 zeigt einige dieser Handpositionen. Die grün, gelb und rot markierten Bereiche stellen jeweils dar, welche Bereiche des Display bei der Handhaltung gut (grün), weniger gut (gelb) und schlecht (rot) erreicht werden können. Damit auch Personen mit motorischen Einschränkungen die Benutzeroberfläche gut bedienen können, sollten Bedienelemente daher nur in den grün dargestellten Bereichen liegen. Die oberen 30 Prozent der Displayfläche, sowie der rechte Bereich und vor allem das Eck rechts unten sollten gemieden werden.

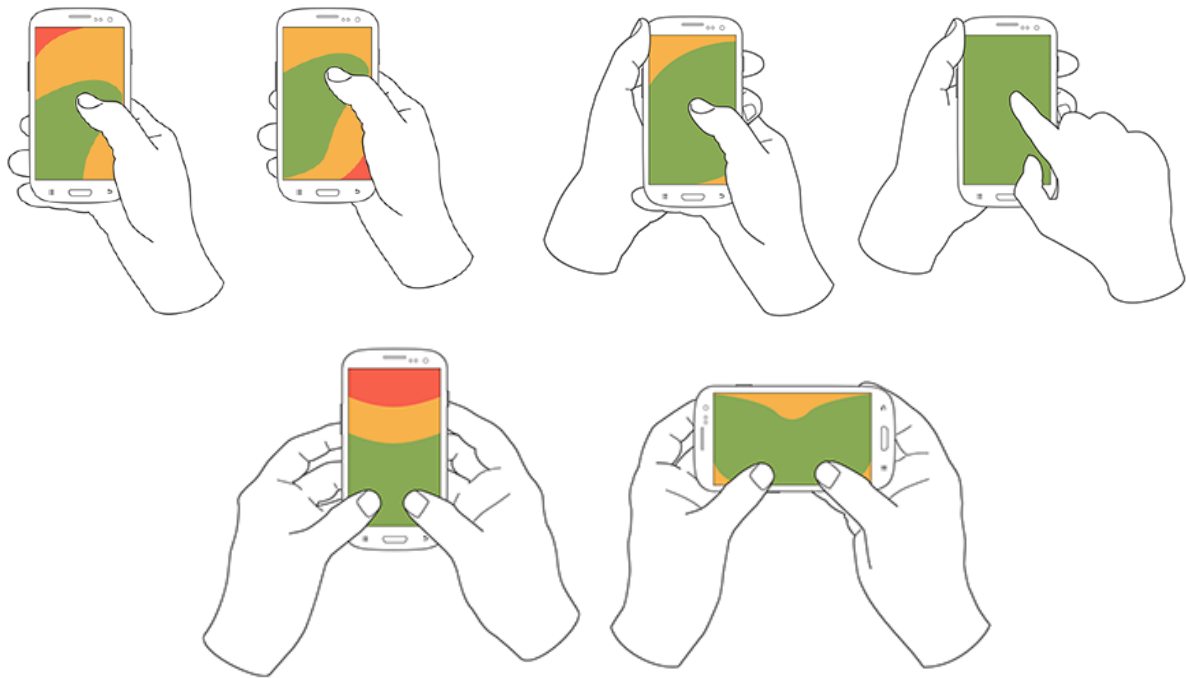


Abbildung 2.1: Verschiedene Möglichkeiten, ein Smartphone zu halten und die Erreichbarkeit der verschiedenen Displaybereiche (grün: gut, gelb: mittel, rot: schlecht). Quelle: [8]

2.2. Vorhandene Seniorenhandys

Einige Hersteller bieten speziell für Senioren entwickelte Handys oder Smartphones an. Diese werden nachfolgend in die drei Kategorien ‚Großtastendhandys‘, ‚Notruftelefone‘ und ‚Senioren-Smartphones‘ eingeteilt und kurz vorgestellt.

2.2.1. Großtastendhandys

Bei Großtastendhandys handelt es sich um Geräte für Senioren, die den klassischen Handys mit mechanischen Tasten sehr nahe kommen. Sie weisen allerdings einige wichtige Unterschiede auf, um sie seniorentauglich zu machen. Neben vergrößerten Tasten zur Bedienung, welche gut lesbar beschriftet sind, beherbergen sie hochwertige Lautsprecher für die Freisprechfunktion und besitzen

Displays mit hohem Kontrast und großem Blickwinkel. Die Menüführung ist übersichtlich gehalten und es gibt Schnellwahlmöglichkeiten wichtiger Kontakte. Die meisten Handys besitzen außerdem eine Notruf Funktion zur Benachrichtigung eingespeicherter Kontakte, die über einen Knopf an der Rückseite gestartet werden. In einigen Fällen besteht die Möglichkeit der Fernkonfiguration, sowie der Ortung des Handys über GPS [9].



Abbildung 2.2: Swisstone BBM 410d

Bildquelle: http://ecx.images-amazon.com/images/I/81zB1amnEzL._SL1500_.jpg



Abbildung 2.3: emporiaPURE (Notrufknopf auf der Rückseite)

Bildquelle: http://www.neckermann.at/out/pictures/generated/product/2/900_900_90/emporia-pure-v25-schwarz-inkl-b-free-startpaket-4856803.jpg

2.2.2. Notruftelefone

Notruftelefone teilen die meisten Eigenschaften mit den Großtastentelefonen. Auch sie verfügen über starke Lautsprecher, gut ablesbare Displays und eine übersichtliche Menüführung. Während Großtastentelefone allerdings große Wahltasten für Telefonnummern aufweisen, besitzen Notruftelefone einige wenige große Knöpfe, auf denen feste Rufnummern hinterlegt sind.

Ein Vertreter dieser Kategorie ist das Doro Secure 580IUP [10]. Dieses besitzt vier Knöpfe (A, B, C, D), welche sich belegen und beschriften lassen, sowie einen Assistenzbutton auf der Rückseite. Aufgrund der fehlenden Tasten ist es nicht möglich mit diesem Handy Nachrichten (SMS) zu schreiben. Der Fokus liegt viel mehr auf den Sicherheits- und Assistenzfunktionen. So bietet das Handy automatische und vom Nutzer aktivierte Benachrichtigungen. Fällt das Handy zu Boden oder wird lange Zeit nicht bewegt, kann es so beispielsweise einen bestimmten Kontakt alarmieren. Die Ortung per GPS ist ebenfalls möglich.



Abbildung 2.4: Doro Secure 580IUP

Bildquelle: <http://www.doro.co.uk/mobile-devices/easy-phones/doro-secure-580iup.html>

2.2.3. Senioren-Smartphones

Seit 2013 werden jährlich mehr Smartphones als klassische Handys (sogenannte Feature Phones) verkauft [11]. Auch im Bereich von Seniorenhandys folgen Hersteller diesem Trend und entwickeln spezielle Smartphones für diese Zielgruppe. Diese übernehmen viele Funktionen der bereits behandelten Großtasten- und Notruftelefone, wie zum Beispiel den Notrufknopf auf der Rückseite, und versuchen mit einer gut ablesbaren, großen Touchscreen-Anzeige zu überzeugen. Der Funktionsumfang ist deutlich größer, da auch Anwendungen wie E-Mail oder Internet genutzt werden können. Außerdem ermöglicht die Internetverbindung eine Fernwartung über einen Webbrowser oder eine separat erhältliche App von einem anderen Handy aus. Über ein GPS Modul können Angehörige den Aufenthaltsort des Smartphones jederzeit erfahren.

Der Hersteller Doro hat mit dem PhoneEasy 740 [12] ein Smartphone auf den Markt gebracht, das zusätzlich zum großen Touchscreen eine ausziehbare Tastatur mit großen Tasten bietet. Dadurch kann das Smartphone über physische Tasten bedient werden, anstatt nur über den Touchscreen [13]. Das PhoneEasy hebt sich damit etwas von der Masse der Senioren-Smartphones ab, da die meisten Geräte keine zusätzliche Tastatur besitzen. Die Entwicklung von Senioren-Smartphones orientiert sich am allgemeinen Smartphone-Markt, sodass die Geräte immer weniger Einschränkungen im Vergleich zu herkömmlichen Smartphones haben und zusätzlich mit einigen Vorteilen wie einer einfacheren Bedienung, Fernwartung, Ortung oder Notruf- und Assistenzfunktionen zu überzeugen versuchen. Ein Vertreter eines solchen Smartphones ist das Doro Liberty 825 [14].



Abbildung 2.5: Doro PhoneEasy 740 Smartphone mit ausziehbarer Tastatur
 Bildquelle: <http://www.doro.ch/doro-phoneeasy-740.html>



Abbildung 2.6: Doro PhoneEasy 825, ein aktuelles Senioren-Smartphone
 Bildquelle: <http://www.dorodeutschland.de/doro-liberto-825.html>

2.3. Android Launcher für Senioren

Neben den bereits vorgestellten Geräten für Senioren, besteht auch die Möglichkeit, normale Smartphones seniorenfreundlich zu machen, indem man eine Launcher App installiert. In der Zwischenzeit gibt es eine Vielzahl an Launcher-Apps, welche alle dasselbe Ziel haben, aber auf unterschiedlichen Wegen dorthin gelangen. Auf einige dieser Apps wird im Folgenden näher eingegangen.

2.3.1. Simple Senior Phone

Der Launcher Simple Senior Phone [15] kann kostenlos heruntergeladen werden und vereinfacht die Benutzeroberfläche von Android. Zu Beginn erscheint eine kleine Einführung. Die Einführung kann jederzeit beenden werden, sodass die App zum Startbildschirm wechselt (Abbildung 2.8a). Dieser beinhaltet im oberen Teil Statussymbole für Batterie, WLAN und ähnliches, außerdem die Anzeige von Uhrzeit und Datum, sowie einen Button zu den hinterlegten Anwendungen.

Direkt darunter befindet sich ein Benachrichtigungsfeld, gefolgt von Buttons zu den vier Funktionen Telefon, SMS, Finden und SOS. Die Finden-Funktion ermöglicht es, eingetragenen Betreuern den aktuellen Standort zu übermitteln. Über den SOS-Button kann eine SOS-Nachricht an die Betreuer gesendet werden. Die App verfügt auch über ein integriertes Adressbuch, in dem sich Name, Telefonnummer und Bilder der Kontakte speichern lassen. Über das Anwendungsmenü kann die App um einige praktische Funktionen erweitert werden: einen Taschenrechner, Kamera, Lupe, Musik, Radio und Taschenlampe.

Das Design ist sehr schlicht und durchgehend gut verständlich. Es wurden Farben eingesetzt, um die Nutzung zu erleichtern. Die App beinhaltet eine eigene Bildschirmtastatur mit ausreichend großen Tasten (Abbildung 2.8b). Es werden allerdings immer nur zehn Buchstaben in alphabetischer Reihenfolge angezeigt, sodass beim Tippen ständig zwischen drei Tastaturansichten gewechselt werden muss. Für die Zahlen kommt eine vierte Ansicht hinzu. Als kleine Tipphilfe schlägt

die Nachrichtenfunktion vordefinierte Sätze wie „Wie geht es dir?“ vor. Insgesamt dauert das Tippen aber durch den Wechsel der Ansichten sehr lange.

In der App wurde auf Scrollbalken verzichtet, sodass nur durch Drücken eines „Weiter“-Buttons durch längere Nachrichten oder Listen navigiert werden kann (siehe Abbildung 2.8d). Wischgesten sind somit nicht erforderlich. Die angepassten Anrufbildschirme (Abbildung 2.8c) sind auf das Minimum reduziert. Leider fehlt der App ein angepasster Sperrbildschirm, sodass der Nutzer nach dem Sperren des Gerätes immer mit dem Android Sperrbildschirm und der nötigen Wischgeste zum Entsperren konfrontiert wird.

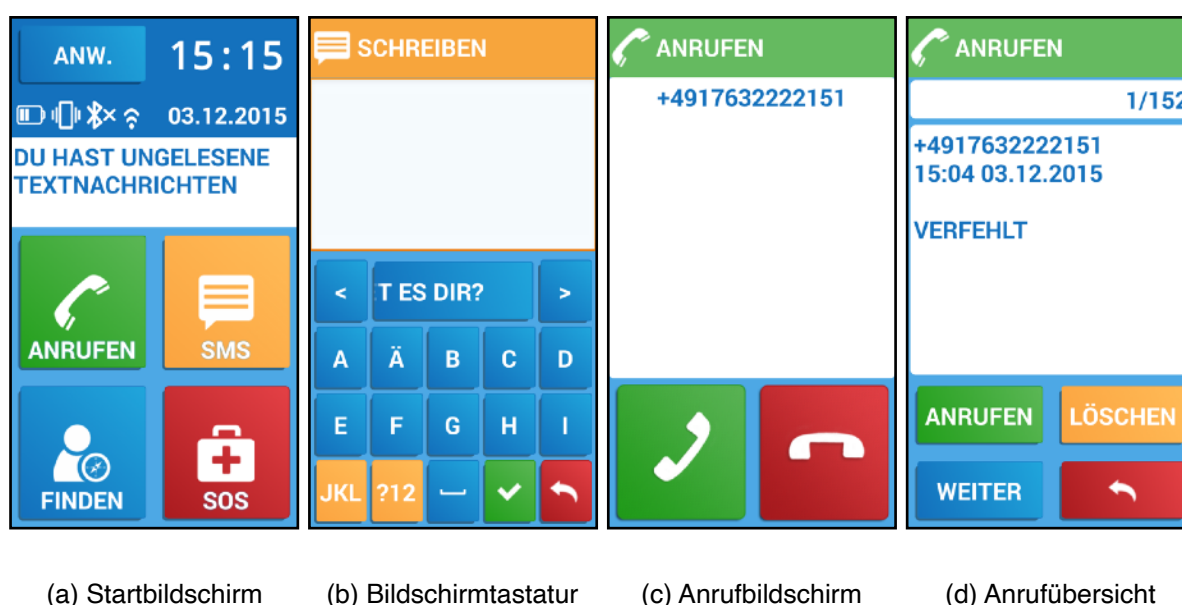


Abbildung 2.7: Simple Senior Phone, Bilder: eigene Screenshots

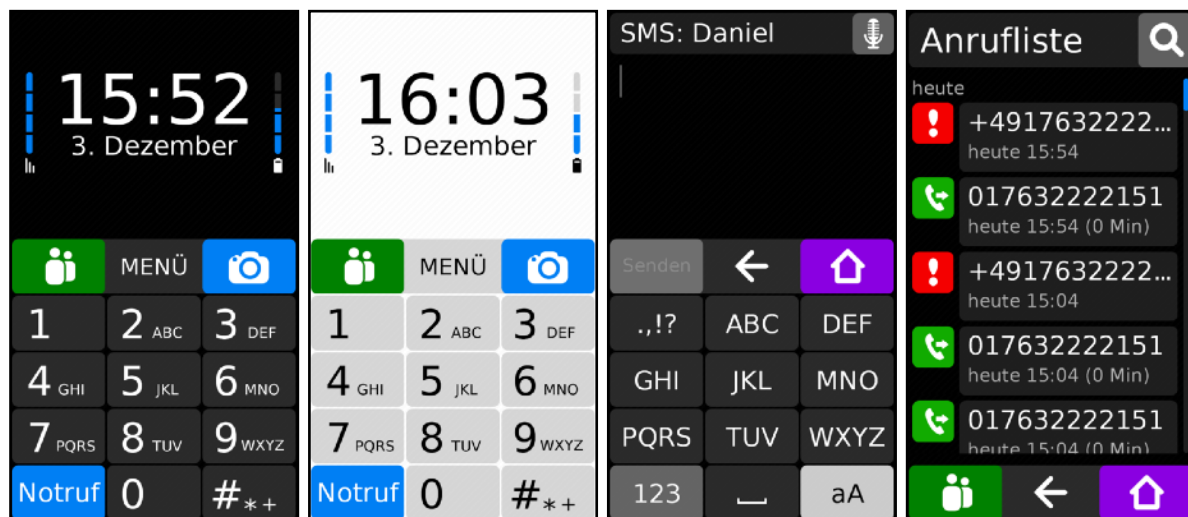
2.3.2. Koala Phone Launcher

Der Koala Phone Launcher [16] ist sowohl in einer kostenlosen, wie auch einer kostenpflichtigen Version erhältlich. Der Nutzer landet voreingestellt auf einem Startbildschirm (Abbildung 2.9a), der vom Layout her einem alten Handy nachempfunden ist: am oberen Bildschirmrand befinden sich Anzeigen für Uhrzeit, Datum, Empfang und Batterie, darunter gibt es drei Knöpfe und eine Tastatur zum

Wählen der Nummer. Die beiden oberen Knöpfe können frei belegt werden. Am unteren Rand gibt es zudem noch einen Notrufknopf, über den entweder eine Notrufnummer angerufen werden kann oder der Nummer eine SMS mit dem aktuellen Ort des Handys gesendet wird. Im Menü befinden sich weitere Funktionen wie Nachrichten, Kontakte, Kamera, Wecker oder Taschenlampe. Außerdem können beliebige Apps gestartet werden. Der Launcher verfügt über ein großes Einstellungsmenü, das unter anderem verschiedene ‚Themes‘ (siehe Abbildung 2.9b), Eingabehilfen und Belegungsmöglichkeit der Knöpfe des Startbildschirmes beinhaltet.

Koala Phone ist sehr leicht verständlich, übersichtlich und gleichzeitig schön gestaltet. Die Verwendung von Farben verbessert die Bedienung. Ästhetisch ist vor allem das Design der verschiedenen Funktionen wie der Kamera. Auch die integrierte Bildschirmtastatur (Abbildung 2.9c) ist übersichtlich und gut zu bedienen. Jede Taste ist wie bei alten Handys mit drei Buchstaben belegt, zwischen denen man durch Antippen wechseln kann. Das Tippen fällt damit leichter als mit der Tastatur von 2.3.1. Nebenbei kann per Spracherkennung geschrieben werden. Sehr gut gelöst ist außerdem, dass jeder Bildschirm der App am unteren Bildschirmrand einen einheitlichen Balken besitzt, über den man zum Startbildschirm oder zum vorherigen Bildschirm zurück kommt (siehe Abbildung 2.9d). Alles wird mit sehr angenehmen Animationen leicht überblendet. Die Animationen können aber auch deaktiviert werden.

Der einzige negative Punkt ist, dass der Launcher leider nicht über angepasste Anrufbildschirme oder einen Sperrbildschirm verfügt. So muss jedes Mal nach Sperren des Gerätes erst die Android Bildschirmsperre mit einer Wischgeste deaktiviert werden und der Nutzer wird bei Telefonaten ebenfalls mit dem normalen Design des Handys konfrontiert.



(a) Startbildschirm (b) Startbildschirm hell (c) Bildschirmtastatur (d) Anrufübersicht

Abbildung 2.8: Koala Phone Launcher, Bilder: eigene Screenshots

2.3.3. Fontrillo - The easy launcher

Fontrillo [17] ist etwas anders aufgebaut, als die meisten Launcher für Senioren. Nach einer kurzen Einführung, in der beispielsweise verschiedene Wischgesten erklärt werden, befindet sich der Nutzer im Startbildschirm (Abbildung 2.10a). Die App ist einfach aufgebaut: am oberen Bildschirmrand befindet sich ein großer SOS-Button, darunter ist die Übersicht der verschiedenen Funktionen durch die sich der Nutzer per Wischgeste nach rechts oder links bewegen kann. Etwa 80% der Fläche nimmt die aktuell aktive Funktion auf. Im Startbildschirm ist das die Anzeige von Datum und Uhrzeit, sowie ein Button, der die App beendet. Ganz unten befindet sich ein schmaler Balken mit der Anzeige der Uhrzeit und der Batterieladung.

Die App verfügt über einige integrierte Funktionen wie ein Notizbuch, eine Taschenlampe und eine gut gestaltete App-Übersicht. Es wurde eine praktische Bildschirmtastatur integriert (Abbildung 2.10b), welche das gesamte Alphabet anzeigt. Mehrfachklicks wie bei den Lösungen der anderen Launcher entfallen somit. Der Nachteil der großen Tastatur besteht darin, dass wenig Platz für die Anzeige des eingegebenen Textes bleibt. Fontrillo kann die Android

Bildschirmsperre deaktivieren, besitzt aber keinen eigenen Sperrbildschirm, sodass der Nutzer nach dem Aktivieren des Gerätes direkt im Launcher landet. Es besteht zwar in den Einstellungen die Möglichkeit einen angepassten Bildschirm für eingehende Anrufe festzulegen, auf dem Testgerät ließ sich diese Einstellung aber nicht aktivieren. Insgesamt ist die App durchgehend sehr übersichtlich und verständlich gestaltet. Die nötige Wischgeste nach rechts und links im Startbildschirm, um bestimmte Funktionen aufzurufen ist hingegen eher unpraktisch. Um eine eigene Telefonnummer zu wählen, muss beispielsweise vom Startbildschirm (Abbildung 2.10a) dreimal nach links gewischt werden, um in der Nummerneingabe (Abbildung 2.10c) zu landen.

Praktisch ist der überall verfügbare SOS-Knopf am oberen Bildschirmrand. Dieser kann durch langes Anklicken aktiviert werden und ruft dann drei eingestellte Kontakte an oder sendet diesen eine SMS mit dem aktuellen Aufenthaltsort zu. Die SOS-Funktion kann ebenfalls durch Schütteln des Gerätes aktiviert werden.

Besitzen Angehörige ebenfalls die Fontrillo App, können diese außerdem den aktuellen Aufenthaltsort des Handys erfahren. Sie benötigen dazu ein Passwort (PIN) und erhalten kurze Zeit später eine SMS mit dem Aufenthaltsort. Eine Internetverbindung ist somit nicht nötig.



(a) Startbildschirm

(b) Bildschirmtastatur

(c) Nummerneingabe

(d) Anrufübersicht

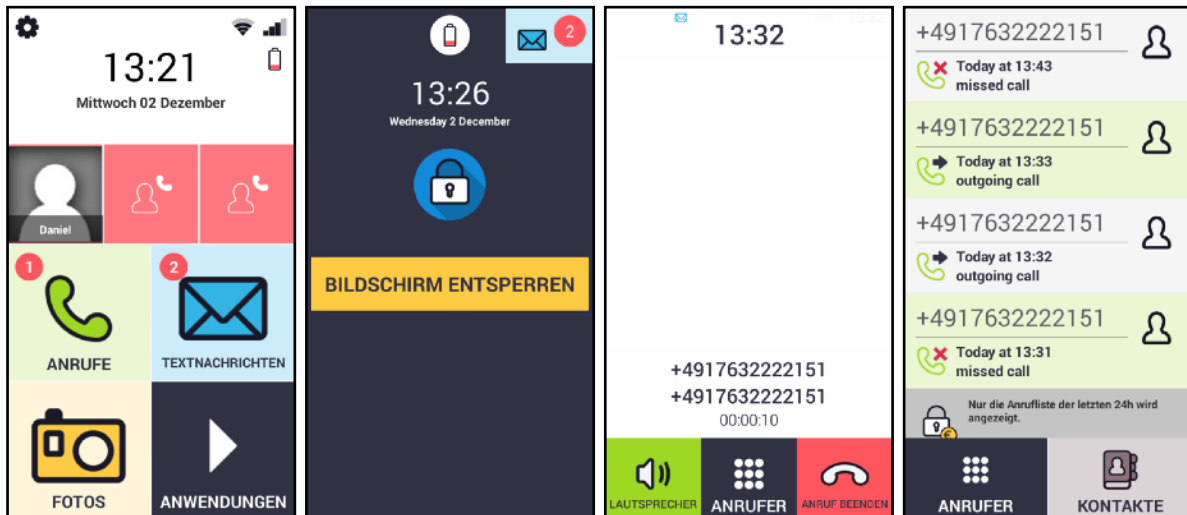
Abbildung 2.9: Fontrillo, Bilder: eigene Screenshots

2.3.4. Phonotto Easy Phone Launcher

Phonotto [18] ist eine Launcher App für ältere Menschen, die sehr übersichtlich die wichtigsten Telefonfunktionen präsentiert. Nach der Installation wird durch eine kurze Ersteinrichtung geführt, bevor die App den schlanken Startbildschirm (Abbildung 2.11a) präsentiert. Dieser beinhaltet eine große Anzeige der Uhrzeit und des Datums, sowie Statussymbole für die Datenverbindung, WLAN, Bluetooth und die Batterie, sowie einen Knopf, um in die Einstellungen zu gelangen. Darunter sind drei Schnellwahlknöpfe aufgeführt, gefolgt von vier Kacheln für die Telefon- und SMS-Funktion, Fotos und Anwendungen. Außerdem beinhaltet die App ein Adressbuch, auf das über die Telefonapp zugegriffen werden kann. Beliebige Apps können über den Knopf für Anwendungen aufgerufen werden. Voreingestellt kann man dort sechs Apps hinterlegen, wobei in der kostenlosen Version nur zwei Apps freigeschaltet sind.

Das Design ist sehr gut durchdacht und alle Funktionen sind leicht erreichbar. Die Größe der Buttons ist mehr als ausreichend. Leider nutzt die App die Standard-Android Tastatur zum Schreiben, welche sich durch sehr kleine Tasten auszeichnet. Hier könnte durch Installation einer anderen Tastatur Abhilfe geschaffen werden. In Listen ist schwer auszumachen, wo ein Element beginnt und das nächste anfängt. Hier hätte eine deutlichere visuelle Trennung vorgenommen werden können (siehe Abbildung 2.11d).

Sehr schön ist, dass die App ein einheitliches Erscheinungsbild hat. Sie verfügt neben dem Launcher und den angesprochenen Funktionen Telefon, SMS, Fotos und Adressbuch auch über einen angepassten Sperrbildschirm (Abbildung 2.11b), sowie angepasste Bildschirme für ein- und ausgehende Anrufe (Abbildung 2.11c). Anzumerken ist außerdem, dass kritische Funktionen wie die Schnellwahltasten, das Öffnen der Einstellungen, sowie das Entsperren des Gerätes ein langes Drücken des jeweiligen Knopfes erfordern.



(a) Startbildschirm

(b) Sperrbildschirm

(c) Anrufbildschirm

(d) Anrufübersicht

Abbildung 2.10: Phonotto Easy Phone, Bilder: eigene Screenshots

2.4. Zusammenfassung der Analyse

Aus den Kapiteln 2.1 bis 2.3 lassen sich Anforderungen an die Entwicklung einer Seniorenhandy-App ableiten. Diese wurden in Tabelle 2.3 in die drei Kategorien ‚Layout/Gestaltung‘, ‚Interaktion‘ und ‚Funktionsumfang‘ eingeteilt. Da alle Anforderungen bereits in den vorhergehenden Kapiteln beschrieben wurde, wird an dieser Stelle nicht mehr näher darauf eingegangen. Noch nicht genannt wurde das Vermeiden von Fehlerdialogen, was eng mit der konsistenten Benutzeroberfläche und der selbsterklärenden Bedienung zusammenhängt. Es soll vermieden werden, dass der Nutzer mit Fehlerdialogen des Systems konfrontiert wird, und stattdessen auf selbsterklärende Weise dargestellt wird, warum etwas nicht funktioniert.

Anforderungen an die Entwicklung einer Seniorenhandy-App		
Layout / Gestaltung	Interaktion	Funktionsumfang
4-5 Elemente pro Seite	keine Animationen	Telefonieren
Verwendung von Farben	keine Gestensteuerung	Datum/Uhrzeit-Anzeige
hohe Kontraste	keine Scrollbalken	Adressbuch
serifenlose Schrift, min. 14pt	selbsterklärende Bedienung	Notfalltaste
große Schaltflächen	nur eine Funktion pro Taste	Assistenzfunktionen
obere 30% und rechte Displayseite meiden	Rückmeldung auf Interaktion	Ortung per GPS
konsistente Benutzeroberfläche	Fehlerdialoge vermeiden	

Tabelle 2.3: Anforderungen an die Entwicklung einer Seniorenhandy-App

3. App Entwicklung für Android

Die Seniorenhandy-App wurde für die Android-Plattform entwickelt. Einen kleinen Überblick über die Entwicklung von Android Apps bieten die nachfolgenden Kapitel.

3.1. Aufbau von Android Projekten

Android Apps werden in der Programmiersprache *Java* [19] entwickelt. Für die Entwicklung der Seniorenhandy-App wurde *Eclipse* [20] als Entwicklungsumgebung verwendet und durch die *Android Development Tools (ADT)* [21] für Eclipse erweitert.

Eclipse und die ADT ermöglichen das einfache Erstellen eines neuen Android Projektes. Dieses besteht zu Beginn aus einer Vielzahl an Dateien und Ordnern und kann direkt als leere Android App kompiliert und auf einem Gerät installiert werden. Der Aufbau eines leeren Android Projektes ist in Abbildung 3.1 dargestellt. Eine Beschreibung der einzelnen Elemente liefert T. Allevato auf seiner Webseite [22]. Der Ordner „src“ beinhaltet den Java-Quellcode der App. Der Ordner kann verschiedene Typen von Java-Dateien beinhalten. Einige gängige Typen in Android sind Activities, Receiver, Services und Klassen. Auf die ersten drei wird in den folgenden Kapiteln kurz eingegangen.

Neben dem App-eigenen Quellcode beinhaltet die App die Android Programmierschnittstelle (in der Abbildung „Android 5.1.1“), sowie einen Ordner „libs“, in welchem vorkompilierte JAR-Archive von Drittanbieter-Libraries abgelegt werden können. Der Ordner „gen“ beinhaltet automatisch generierten Quellcode der ADT. Weiterhin existieren im Projekt die Ordner „res“ und „assets“. Unter „res“ lassen sich verschiedene Ressourcen wie Layout Dateien, Grafiken, Menüs und Sprachdateien hinterlegen, während der „assets“-Ordner für andere Mediendateien

gedacht ist, wie beispielsweise Videos, Töne oder große Bilder, welche nicht direkt im Layout der App verwendet werden.

Der „res“-Ordner beinhaltet neben den Grafiken in den „drawable...“-Unterordnern hauptsächlich XML-Dateien, in welchen die Layouts, Styles, Menüeinträge oder Text-Strings der verschiedenen Sprachen hinterlegt sind.

Eine der wichtigsten Dateien der Android App befindet sich im Hauptordner und trägt den Namen „AndroidManifest.xml“ [23]. Sie beinhaltet alle wichtigen Infos zur App, wie den Namen und die Version, die verwendeten Android SDK Versionen und benötigten Berechtigungen des Android Systems. Außerdem definiert Sie alle Programmteile der App, wie Activities, Receiver und Services.

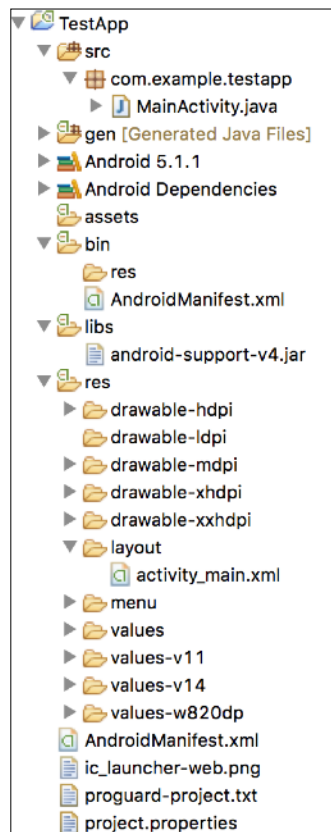


Abbildung 3.1: Aufbau einer neu erstellten Android App

Bildquelle: Screenshot aus Eclipse mit installierten Android Developer Tools

3.2. Activities

Die Android Developer Reference [24] beschreibt eine Activity als „single, focused thing that the user can do [25]“. Es sind normalerweise Teile einer App, welche mit dem Nutzer interagieren. Sie stellen daher ein Fenster bereit, welches die Bedienoberfläche darstellt. Die meisten Activities werden im Vollbild angezeigt, es besteht aber auch die Möglichkeit, sie in andere Activities zu integrieren oder sie auf einer Ebene überhalb einer anderen Activity zu platzieren.

Eine Activity besitzt vier Hauptzustände *Activity launched*, *Activity running*, *App process killed* und *Activity shut down*. Die verschiedenen Zustände, sowie die aufgerufenen Methoden bei Zustandsübergängen sind in Abbildung 3.2 dargestellt.

Beim Start einer Activity werden nachfolgend die Methoden *onCreate()*, *onStart()*, und *onResume()* der Activity ausgeführt und die Activity befindet sich danach im Zustand *Activity running*. Wird die Activity von einer nicht im Vollbild ausgeführten Activity überlagert, beispielsweise von einem Popup, ändert sich ihr Zustand auf *paused*, indem die Methode *onPause()* ausgeführt wird. Von diesem Zustand aus kann die Activity wieder aktiviert werden, sobald der Nutzer zu ihr zurückkehrt, wobei die Methode *onResume()* aufgerufen wird. Wird eine Activity vollständig von einer anderen verdeckt, wird sie durch die Methode *onStop()* in den Zustand *stopped* versetzt. Navigiert der Nutzer zurück zur Activity, wird sie auch hier wiederhergestellt und dabei die Methode *onRestart()* gefolgt von *onStart()* und *onResume()* ausgeführt.

Activities in den Zuständen *paused* oder *stopped* können vom System jederzeit beendet werden, um Speicherplatz freizulegen. Dazu kann das System entweder die *onDestroy()* Methode der Activity aufrufen oder ihren Prozess direkt beenden. In beiden Fällen muss die Activity neu gestartet werden, wenn der Nutzer sie wieder aufrufen möchte.

Listing 3.1 stellt eine einfache Definition einer Activity im *AndroidManifest.xml* dar. Neben dem Namen der Activity wird festgelegt, dass die Activity im Hochformat dargestellt wird (*android:screenOrientation*) und das Android Theme *Light* im Vollbildmodus (*Fullscreen*) ohne Titelleiste (*NoTitleBar*) verwendet wird.

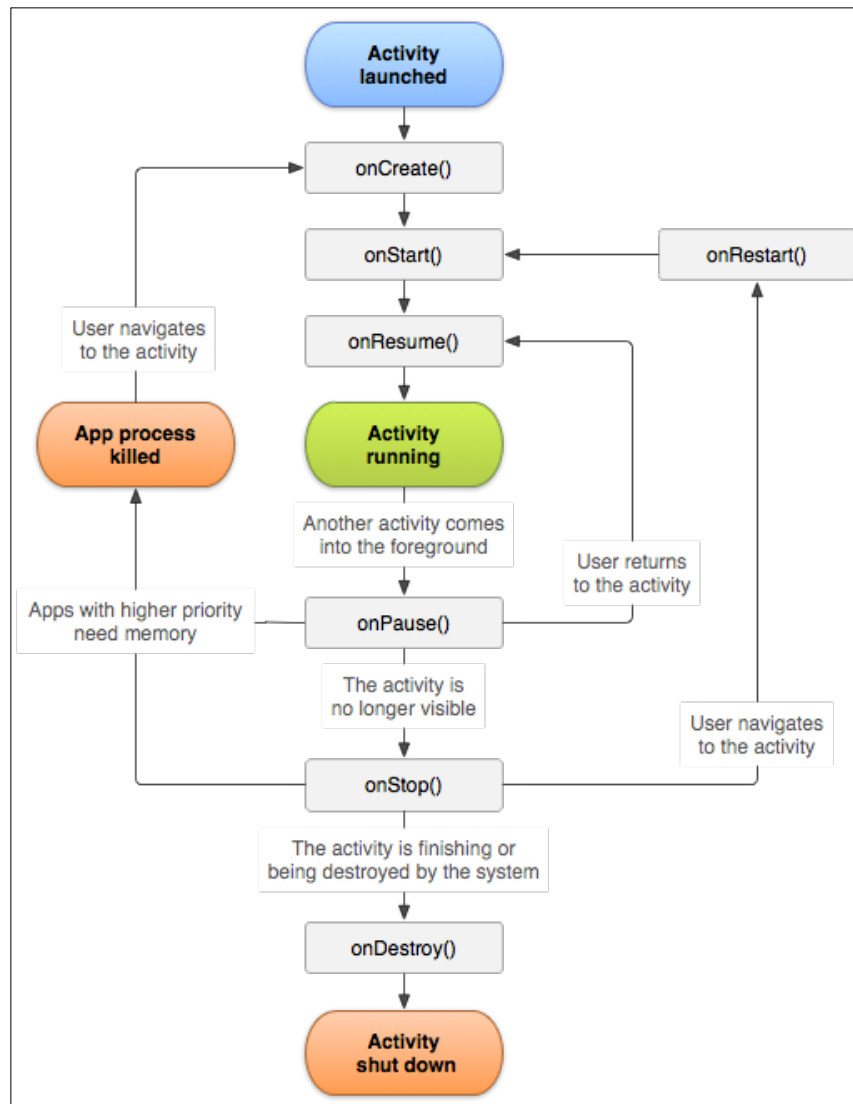


Abbildung 3.2: Activity Lifecycle

Bildquelle: <http://developer.android.com/reference/android/app/Activity.html>

```

<activity
    android:name=".RemoteCallActivity"
    android:screenOrientation="portrait"
    android:theme="@android:style/Theme.Light.NoTitleBar.Fullscreen"
    android:label="RemoteCall" >
</activity>
  
```

Listing 3.1: Definition einer Activity im AndroidManifest

3.3. Intent-Filter und Intents

Der Definition einer Activity, und der eines Broadcast Receivers (siehe 3.4) im *AndroidManifest* können Intent-Filter [26] beigefügt werden. Diese legen fest, welche Zwecke (Intents) die Activity erfüllt. Einige mögliche Intents, die im Rahmen der Arbeit verwendet wurden sind das Wählen von Rufnummern (*DIAL*), das Empfangen von Daten-SMS (*DATA_SMS_RECEIVED*) und das Starten neuer Anrufe (*NEW_OUTGOING_CALL*). Das System weiß somit, welche Apps und Activities eine bestimmte Aufgabe erledigen können und kann dem Nutzer so diverse Apps zur Auswahl bieten, wenn er beispielsweise einen E-Mail versenden möchte [27]. Listing 3.2 stellt beispielhaft die Definition des Intent-Filters einer Activity dar, die Rufnummern wählen kann. Außerdem kann über einen Intent-Filter festgelegt werden, dass die Activity durch Drücken des Homebuttons des Gerätes gestartet werden kann. Dazu muss der Activity die Intent Kategorie *HOME* zugewiesen werden [28]. Drückt der Nutzer den Homebutton, bekommt er Apps aufgelistet, die als solch eine Launcher App definiert wurden und kann festlegen, welche er standardmäßig verwenden möchte. Bei zukünftigem Drücken des Homebuttons wird dann immer direkt die festgelegte App gestartet. Eine Activity mit dieser Eigenschaft wird nachfolgend als Launcher Activity oder Launcher bezeichnet. Die Definition des Intent-Filters einer Launcher Activity im *AndroidManifest* ist unter Listing 3.3 dargestellt, wobei der relevante Teil farbig markiert ist.

Bei dem vorgestellten Seniorenhandy handelt es sich um eine solche Launcher App, die sich im Homebutton des Gerätes verankert und als Mittelpunkt der Nutzerinteraktion dient.


```
<intent-filter>
    <action android:name="android.intent.action.DIAL" />
    <category android:name="android.intent.category.DEFAULT" />
</intent-filter>
```

Listing 3.2: Definition eines Intent-Filters für eine Activity, die Rufnummern wählen kann

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.HOME" />
    <category android:name="android.intent.category.DEFAULT" />
    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

Listing 3.3: Definition einer Activity, die über den Homebutton gestartet werden kann

3.4. Broadcast Receiver

Eine weitere mögliche Komponente einer Android App ist ein Broadcast Receiver. Broadcast Receiver erlauben der App, Intents vom System oder anderen Apps zu empfangen, auch wenn andere Komponenten der App nicht aktiv sind [29].

Ähnlich wie Activities kann ein Broadcast Receiver im Android Manifest definiert werden. Welche Intents er empfangen kann, wird ebenfalls über Intent-Filter festgelegt. Listing 3.4 zeigt die Definition eines Receivers, der die Intents *PHONE_STATE* und *NEW_OUTGOING_CALL* empfangen kann. Er wird somit gestartet, wenn sich der *PHONE_STATE* des Gerätes ändert, oder wenn ein neuer ausgehender Anruf gestartet wird. Dem dargestellten Receiver wurde außerdem die höchste Priorität 999 zugewiesen, damit dieser als Erstes ausgeführt wird, bevor andere Apps den Intent empfangen. Intents werden an alle möglichen Receiver weitergeleitet, die für den Empfang des speziellen Intents vorgesehen sind. Die Reihenfolge ist dabei von der Priorität des Receivers abhängig. Ein Broadcast kann durch die Methode *AbortBroadcast()* abgebrochen werden. Das verhindert die Weiterleitung an die weiteren Receiver.

Ein Broadcast Receiver muss nicht im Android Manifest definiert werden, sondern kann auch dynamisch im Code erstellt und im System registriert werden. Dies geschieht über die Methode *Context.registerReceiver()* [29]. Für das Seniorenhandy wurden beide Varianten der Registrierung eines Receivers verwendet.

```
<receiver android:name=".PhoneListener">
  <intent-filter android:priority="999">
    <action android:name="android.intent.action.PHONE_STATE" />
    <action android:name="android.intent.action.NEW_OUTGOING_CALL" />
  </intent-filter>
</receiver>
```

Listing 3.4: Definition eines Receivers mit Intent-Filter im AndroidManifest

```
private BroadcastReceiver batteryInfoReceiver= new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        // ...
    }
};
registerReceiver( this.batteryInfoReceiver,
    new IntentFilter(Intent.ACTION_BATTERY_CHANGED) );
```

Listing 3.5: Definition eines Receivers mit Intent-Filter aus einer Activity heraus

3.5. Services

Die dritte Komponente einer App, die kurz vorgestellt werden soll, ist ein Service. Ein Service verfügt im Gegensatz zu einer Activity nicht über eine Nutzeroberfläche und wird verwendet, um dauerhaft im Hintergrund ablaufende Funktionen zu implementieren [30]. Die Definition eines Service im *AndroidManifest* ist in Listing 3.6 dargestellt. Ein Service verfügt über einen reduzierten Lifecycle im Vergleich zu einer Activity (siehe Abbildung 3.2). Sobald ein Service gestartet wird, ist er solange aktiv bis er über *Context.stopService()* oder *stopSelf()* gestoppt wird.

Ein nicht aktiver Service kann vom System ähnlich wie eine Activity beendet werden.

Das Seniorenhandy verwendet einen Service, der die Bildschirmsperre von Android deaktiviert.

```
<service android:name=".LockScreenService" />
```

Listing 3.6: Definition eines Services im AndroidManifest

4. Entwurf und Implementierung

Den größten Teil der vorgestellten Arbeit macht die Seniorenhandy-App aus, die alle Abläufe auf dem Seniorenhandy steuert. Sie besteht aus vielen verschiedenen Komponenten wie Activities, Receivern und Services. Daneben gibt es die Konfigurations-App, die es einer autorisierten Person von ihrem Handy aus ermöglicht, die Einstellungen auf dem Seniorenhandy zu ändern. In den nachfolgenden beiden Unterkapiteln wird die Entwicklung der beiden Apps beschrieben.

4.1. Seniorenhandy-App

Die Seniorenhandy-App stellt nach der Installation auf dem Endgerät den Mittelpunkt der Nutzerinteraktion dar und ersetzt die normale Android-Benutzeroberfläche. Abbildung 4.1 stellt eine vereinfachte Architektur der Seniorenhandy-App dar. Dort wird deutlich, welche Intents (rot) vom System durch die App abgefangen werden. In den nachfolgenden Kapiteln werden die einzelnen Komponenten der App beschrieben.

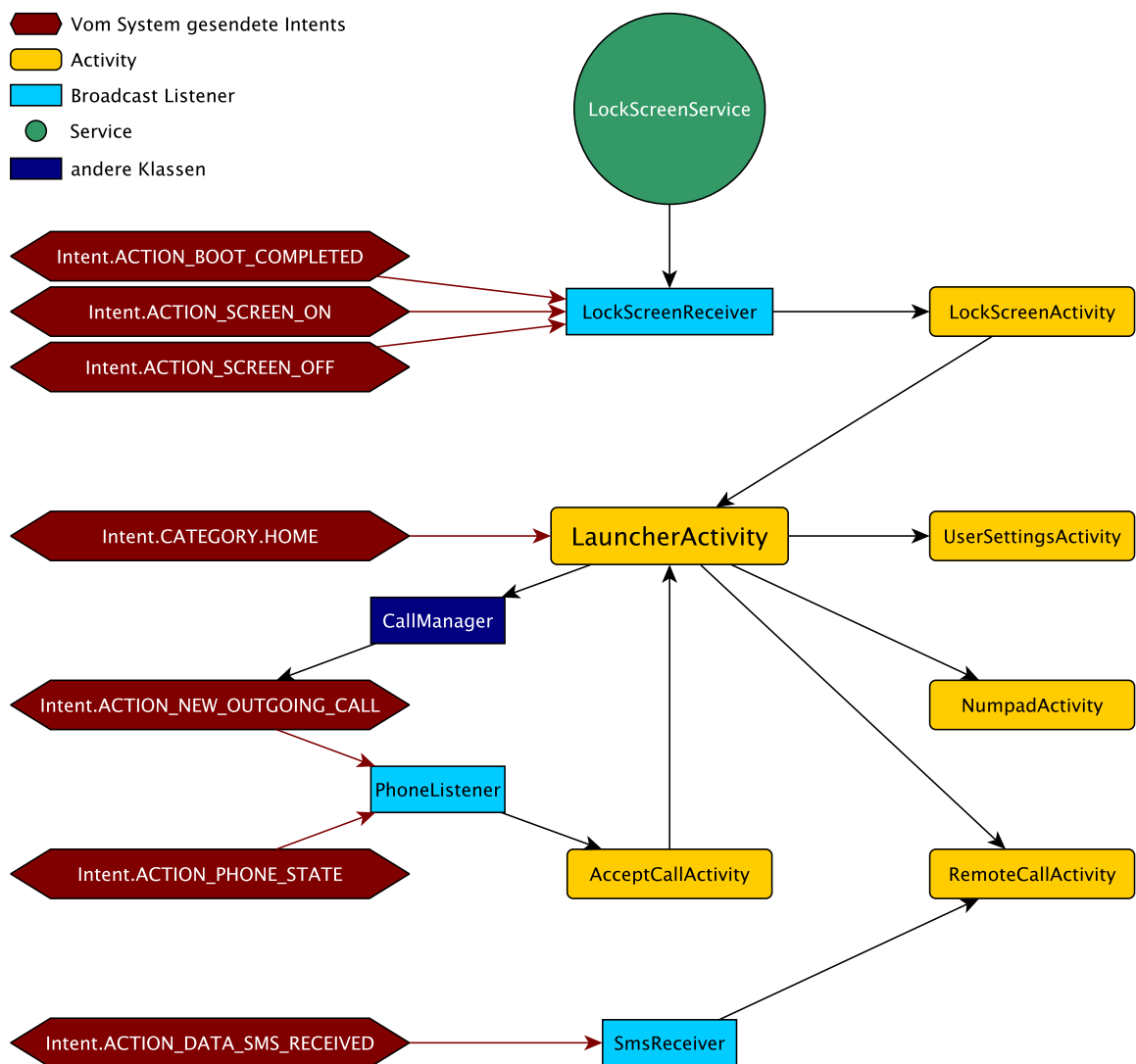


Abbildung 4.1: Vereinfachte Architektur der Seniorenhandy-App

4.1.1. Launcher Activity

Die LauncherActivity ersetzt den Startbildschirm des Android Gerätes und kann über den Homebutton aufgerufen werden. Die Grundidee ihres Layouts kam von der App Easy Home (Abbildung 4.2), die von der factis research GmbH zusammen mit der Lohmann & Birkner Mobile Services GmbH entwickelt wurde [31]. Die Activity sollte große Knöpfe über die komplette Bildschirmbreite besitzen, davon aber nur vier bis fünf untereinander. Darüber sollte als wichtiges Element die

aktuelle Uhrzeit erscheinen. Diese Idee deckt sich mit den unter 2.4 erarbeiteten Anforderungen an die Seniorenhandy-App.

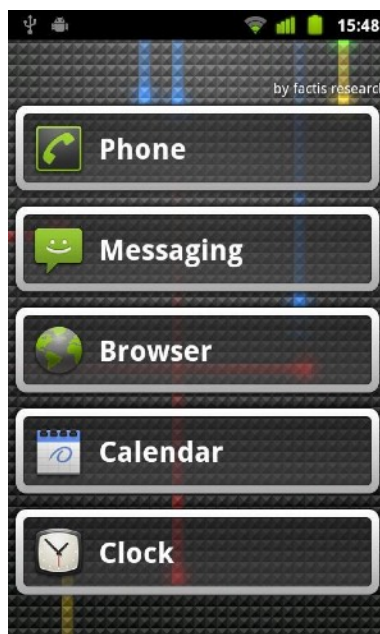


Abbildung 4.2: App Easy Home der factis research GmbH

Bildquelle: <https://play.google.com/store/apps/details?id=com.factisresearch.easyhome>

Um die Buttons einfacher zu unterscheiden, sollte jedem eine eigene Farbe gegeben werden. Die Helligkeit der Farbtöne wurde so gewählt, dass die Buttons mit dem hellen Hintergrund und einer weißen Schrift einen guten Kontrast abbilden. Der erste Entwurf (Abbildung 4.3) beinhaltete außerdem ein Hörersymbol im Button, um dem Nutzer mitzuteilen, dass mit diesem Button der entsprechende Kontakt angerufen wird. Die Belegung der Buttons ist vom Nutzer oder einer von ihm ausgewählten Person frei wählbar, die Reihenfolge der Farben ist fest. Es kann beispielsweise wie in Abbildung 4.2 der rote Button mit einer Notruf-Nummer belegt werden.



Abbildung 4.3: Erster Entwurf des Launchers

Um die *LauncherActivity* nach dem Entwurf umzusetzen, wurde eine angepasste Uhrzeitanzeige (*CustomDigitalClock*) und eine Button-Klasse erstellt, über welche die fünf farbigen Buttons verwaltet werden. Jedem Button kann eine Telefonnummer und ein Kontaktnamen zugewiesen werden. Durch einen *OnClickListener* initiiert der Button nach dem Antippen einen Anruf der gespeicherten Nummer über einen *CallManager*. Laut den Anforderungen unter 2.4 sollen Fehlerdialoge vermieden werden. Der *CallManager* prüft daher vor Beginn des Telefonats, ob ein Telefonat mit der übermittelten Nummer möglich ist. Ist kein Telefonat möglich, beispielsweise weil der Flugzeugmodus aktiviert, die SIM-Karte gesperrt oder gerade keine Netzverbindung vorhanden ist, werden alle Buttons deaktiviert und erscheinen grau. Anstatt der Anzeige der Uhrzeit liest der Nutzer „Kein Empfang“ (siehe Abbildung 4.4b). Da deaktivierte Buttons in einem Grauton erscheinen, musste die Farbe des letzten Buttons, welcher im Entwurf dunkelgrau war, angepasst werden. Die Wahl fiel auf einen Lila-Farbtönen, da dieser die Farbpalette ästhetisch erweitert und trotzdem einen guten Kontrast und somit eine gute Unterscheidbarkeit zu den anderen Buttons bieten. Das Hörersymbol wurde

nicht in die App aufgenommen, um mehr Platz für eine große Darstellung des Namens zu bieten.

Testweise wurden Hintergrund- und Textfarbe invertiert, sodass weiße Schrift, sowie die farbigen Buttons auf schwarzem Hintergrund erschienen. Diese dunkle Variante des Interfaces wurde in die App aufgenommen und lässt sich über das Einstellungsmenü oder per Fernkonfiguration aktivieren. Sie wird im Folgenden als *Dark Mode* bezeichnet. Nutzer, die Probleme beim Sehen haben, können selbst entscheiden, welche Darstellung sie als angenehmer empfinden.

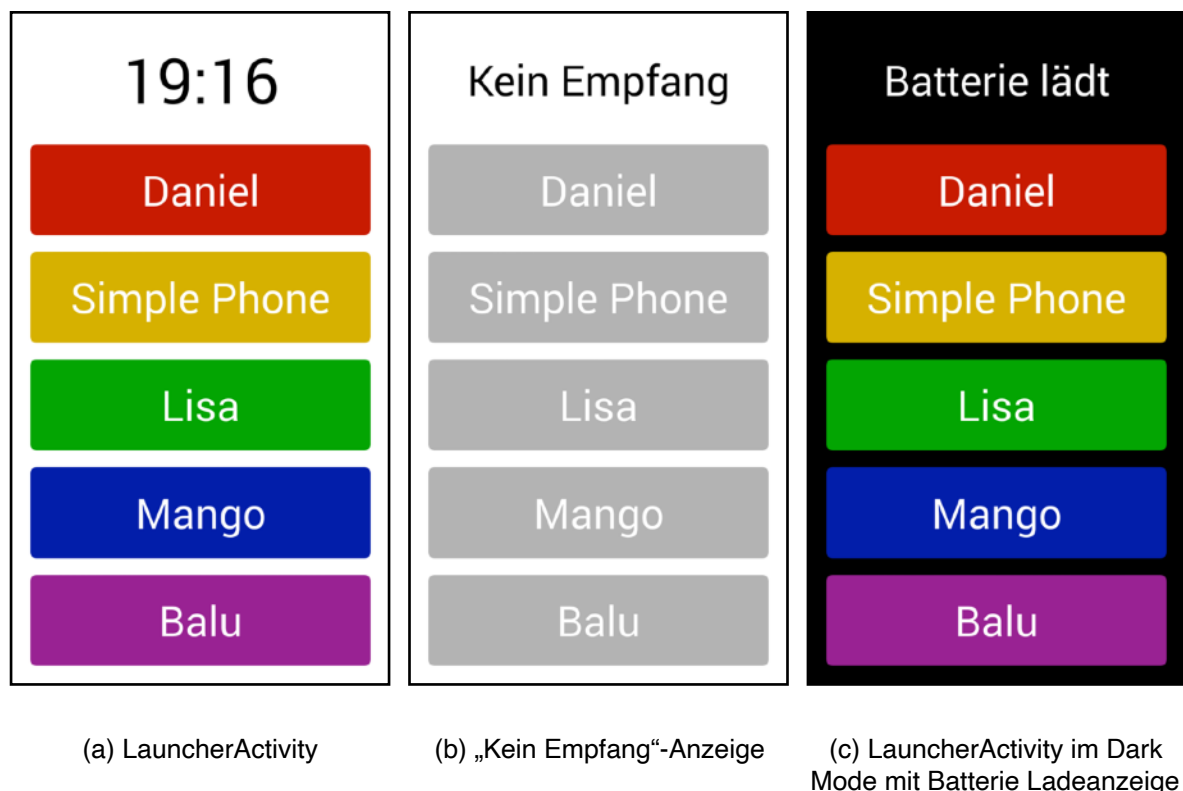


Abbildung 4.4: Verschiedene Darstellungen der LauncherActivity

4.1.2. Konfigurationsmenü

Wichtige Einstellungen des Seniorenhandys sollen nicht nur per Fernkonfiguration, sondern auf dem Seniorenhandy selbst vorgenommen werden können. Da das Menü weniger für den Nutzer selbst, sondern für die Person gedacht ist, welche die App einmalig konfiguriert und anschließend dem Senior das Smartphone übergibt, sollte es dem Nutzer nur schwer gelingen, dass er selbst in das Menü gelangt. Ältere Android Geräte verfügen meist über einen Optionen-Knopf, über den in vielen Apps ein Einstellungsmenü geöffnet werden kann. Da der Nutzer diesen Knopf sehr leicht ungewollt betätigen kann, sollte das Menü nicht über diesen Knopf aufgerufen werden können.

Die Entscheidung zur Unterbringung des Menüs fiel auf eine Wischgeste, da sich diese bei der Anwendung *MediaGuide* der Lohmann & Birkner Mobile Services GmbH bereits erwiesen hatte. Dabei handelt es sich um eine App für Museen, die bewusst keinen sichtbaren Einstellungsknopf haben soll. Es wurde somit die Geste eines ‚S‘ wie ‚Settings‘ und ein *GestureOverlayView* in den Launcher implementiert. Durch Zeichnen eines ‚S‘ in der *LauncherActivity* kann somit direkt zum Konfigurationsmenü gewechselt werden. Die Entscheidung für das ‚S‘ fiel, da es eine leichte Geste ist, die mit einer geringen Wahrscheinlichkeit zufällig gezeichnet wird. Buchstaben wie ein ‚C‘ wie ‚Config‘ wurden beispielsweise schon erkannt, wenn eine minimal gebogene Linie auf das Display gezeichnet wurde. Buchstaben wie ein ‚E‘ wie ‚Einstellungen‘ oder ‚K‘ wie ‚Konfiguration‘ waren hingegen zu schwer zu zeichnen und vom System zu erkennen.

Das fertige Konfigurationsmenü beinhaltet die Option, die Handynummer der autorisierten Person zu ändern, die das Handy aus der Ferne verwalten kann. Außerdem ermöglicht es die Umstellung auf den *Dark Mode* der App und bietet die Möglichkeit den Launcher zurückzusetzen, da die App auf keine andere Weise beenden werden kann. Hierfür wurde eine leere Launcher Activity *FakeHome* erstellt. Deren Aufruf über den entsprechenden Knopf im Konfigurationsmenü führt dazu, dass der ausgewählte Launcher zurückgesetzt wird und dem Nutzer alle installierten Launcher, nicht aber *FakeHome* selbst zur Auswahl angeboten werden.

Als weitere Einstellungen beinhaltet das Konfigurationsmenü das Aktivieren der Freisprechfunktion (4.1.6) und des Buttons zum Wählen eigener Rufnummern (4.1.7).

Da es sich bei dem Konfigurationsmenü wie beschrieben um ein optionales Feature handelt, welches nicht für den Nutzer des Handys gedacht ist, wurde hier ein Standard-Android Menü genutzt, das visuell nicht angepasst wurde (Abbildung 4.5).

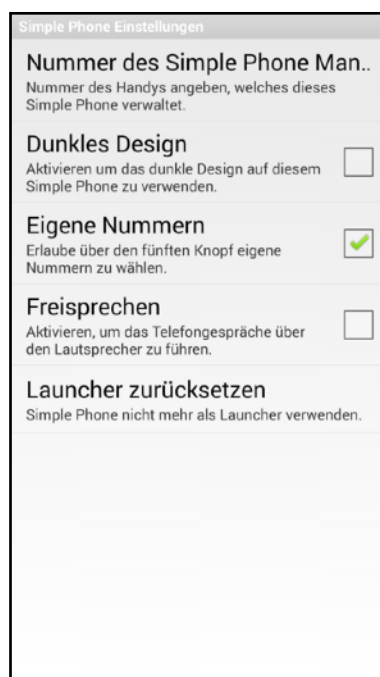
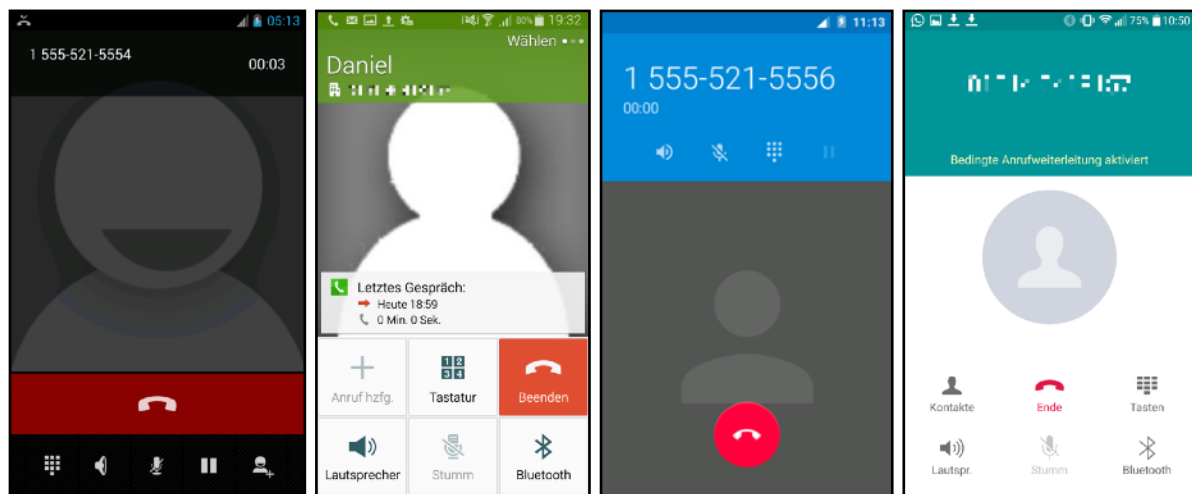


Abbildung 4.5: UserSettingsActivity

4.1.3. Bildschirme für ein- und ausgehende Anrufe

Nahezu jede Android Version beherbergt eine eigene Anzeige für ein- und ausgehende Anrufe. Smartphone Hersteller überschreiben diese systemeigenen Anzeigen, damit sich ihre eigenen Geräte nicht nur hardwaretechnisch, sondern auch bei der Darstellung der Software von anderen Geräten unterscheiden. Deshalb gibt es eine Vielzahl an unterschiedlichen Anzeigen für ein- und ausgehende Anrufe. Einige davon sind unter Abbildung 4.6 dargestellt.



(a) Android Virtual Device (Android 4.3.1)

(b) Samsung Galaxy A5 (Android 4.4.4)

(c) Android Virtual Device (Android 5.1.1)

(d) LG G4 (Android 5.1)

Abbildung 4.6: Anrufbildschirme verschiedener Android Versionen

Um dem Nutzer eine reduzierte und zugleich einheitliche Bedienung zu ermöglichen, wurden eigene Anzeigen für ein- und ausgehende Anrufe konzipiert. Diese sollten nicht mehr als die Infos zum Anrufer und ein bis zwei Buttons zum Auflegen, Annehmen oder Ablehnen des eingehenden Anrufs enthalten. Android bietet die Möglichkeit, Änderungen des Telefonstatus, also ein- und ausgehende Anrufe, abzufangen und daraufhin eine bestimmte App oder Activity zu starten (siehe 3.3). Allerdings legt sich die systemeigene Anzeige für Anrufe über die aktuell geöffnete App. Um die eigene Anzeige in den Vordergrund zu bringen, wurden zwei verschiedene Lösungen erarbeitet.

Wie in Abbildung 4.1 dargestellt, wurde ein Broadcast Receiver *PhoneListener* implementiert, der bei Änderungen des *PHONE_STATE*, sowie bei einem *NEW_OUTGOING_CALL* Intent aktiviert wird und dann anhand der übermittelten Informationen zum Anrufstatus die *AcceptCallActivity* startet. Der *PHONE_STATE* ändert sich beispielsweise durch das Starten oder Beenden eines Telefonates, wenn das Gerät zu einem anderen Netzwerktyp wechselt oder die Datenverbindung abgebrochen wird [32]. Im oben genannten Fall überprüft der *PhoneListener* über

den Status des aktuellen *CALL_STATE*, ob sich am Anrufstatus etwas geändert hat. Damit die *AcceptCallActivity* die systemeigene Anzeige für Anrufe überlagern kann, wurde die Activity zuerst verzögert vom *PhoneListener* aufgerufen. Der dadurch erreichte Übergang ist leider nicht schön, da kurzzeitig die systemeigene Anzeige erscheint, bevor die in Abbildung 4.7 dargestellte Anzeige diese überlagert. Ein weiteres Problem war, dass neuere Android Versionen ab 5.0, und vereinzelt auch schon frühere Versionen, statt einer Vollbild-Darstellung für eingehende Anrufe nur ein kleines Popup anzeigen. Dieses Popup-Fenster erschien im Vordergrund und lies sich nicht durch die *AcceptCallActivity* überlagern. Weiterhin konnte die Activity durch Drücken des Homebuttons oder Sperren und Entsperren des Gerätes einfach beendet werden, sodass sich das Gerät noch in einem Anruf befinden konnte, diesen aber nicht anzeigte und dem Nutzer somit nicht die Möglichkeit gab, den Anruf beenden zu können.

Der zweite Lösungsansatz war, die *AcceptCallActivity* ein Layout als *TYPE_SYSTEM_ALERT* zur aktuellen Displayanzeige hinzufügen zu lassen. Das dadurch angezeigte Layout liegt auf einer Ebene oberhalb von sämtlichen Anzeigen und kann auch nicht durch Drücken des Home-Buttons, oder durch Sperren und Entsperren des Gerätes entfernt werden. Nach dem Beenden des Telefonates muss die Anzeige durch die *AcceptCallActivity* wieder vom Bildschirm entfernt werden, da sie sonst dauerhaft angezeigt werden würde. Dieser zweite Lösungsansatz erwies sich als deutlich besser und kommt ohne Verzögerung aus.

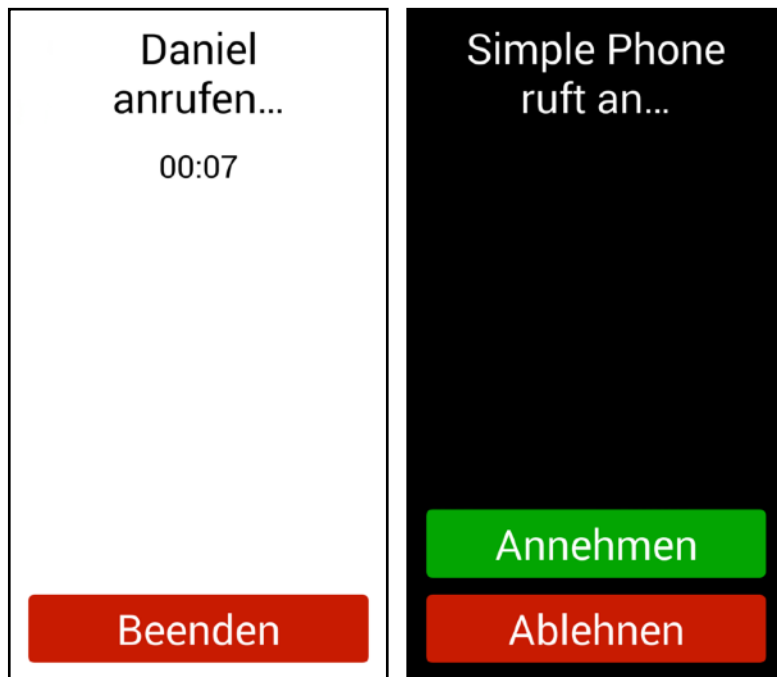


Abbildung 4.7: AcceptCallActivity

4.1.4. Sperrbildschirm

Unter 2.3 wurde festgestellt, dass viele Launcher Apps für Senioren den Nutzer mit dem standardmäßigen Sperrbildschirm des Handys konfrontieren, sofern dieser nicht in den Systemeinstellungen manuell deaktiviert wurde. Wie auch bei den Telefonbildschirmen, gilt für den Sperrbildschirm, dass dieser auf jedem Gerät anders aussehen kann. Hinzu kommt, dass er meist eine Wischgeste zum Entsperren des Gerätes erfordert, was aufgrund der Zielgruppe der App vermieden werden sollte.

Es wurde daraufhin ein schlanker Lockscreen konzipiert, der sich ins Konzept der schon existierenden Activities einreicht. So sollte der Lockscreen lediglich die von der *LauncherActivity* bereits bekannte *CustomDigitalClock* beinhalten und einen großen Button zum Entsperren des Gerätes. Von der Idee, die gesamte Bildschirmfläche als Knopf zu verwenden wurde schnell Abstand genommen, da dadurch zu leicht ungewollt das Gerät entsperrt werden kann, wenn es sich

beispielsweise in der Hosentasche befindet. Es wurde deshalb ein runder Knopf mit einem Schlosssymbol mittig auf dem Display platziert.

Um die *LockScreenActivity* zu starten musste ein *LockScreenReceiver* implementiert werden, welcher aufgerufen wird, sobald der Bildschirm an- oder ausgeschaltet wird, und außerdem direkt nach dem Einschalten des Gerätes (siehe Abbildung 4.1). Da die standardmäßige Bildschirmsperre von Android dauerhaft entfernt werden musste, um die eigene Anzeige zu verwenden, wurde zudem ein *LockScreenService* implementiert. Dieser ist wie unter 3.5 beschrieben dauerhaft im Hintergrund aktiv und übernimmt neben dem Deaktivieren der Android Bildschirmsperre über den systemeigenen *KeyguardManager* [33] auch die Registrierung des Receivers im System.

Da der Sperrbildschirm durch Drücken des Homebuttons umgangen werden konnte, wurde der von 4.1.3 bekannte zweite Lösungsansatz über den Parameter *TYPE_SYSTEM_ALERT* für die Darstellung des Layouts der *LockScreenActivity* übernommen. In diesem Fall bleibt die Darstellung des Lockscreens so lange sichtbar, bis der grüne Entsperren-Button gedrückt wird. Durch Aktivieren der Bildschirmsperre wird der Sperrbildschirm wieder aktiviert.



Abbildung 4.8: LockScreenActivity

4.1.5. Fernwartung

Um die Konfiguration von einem anderen Smartphone zu ermöglichen, wurde ein *BroadcastReceiver* für SMS (*SmsReceiver*) implementiert. Die erste Implementierung basierte auf einem einfachen Receiver, der beim Empfang einer SMS aufgerufen wird. Nach Überprüfung, ob die Absendernummer mit der Nummer des Konfigurationshandys übereinstimmt, wird der Inhalt der SMS an einen *ConfigSmsParser* weitergeleitet, der die neue Konfiguration aus dem Text der SMS ausliest und zurückgibt. Um die Weiterleitung der SMS an die Nachrichtenapp von Android zu verhindern, wird die Methode *AbortBroadcast()* im *BroadcastReceiver* aufgerufen. Auf den Aufbau einer solchen Konfigurations-SMS wird im Kapitel 4.2.2 (Konfigurations-SMS) näher eingegangen.

Als Verbesserung der Implementierung wurde der Receiver auf Binär-SMS (Daten-SMS) umgestellt. Binär-SMS können an einen bestimmten Port gesendet werden und erreichen somit die Nachrichtenapp des Handys nicht, weshalb der Aufruf von *AbortBroadcast()* entfällt. Ein weiterer Vorteil von Binär-SMS besteht darin, dass der Entwickler selbst die Kodierung (Encoding) der Nachricht festlegen kann und nicht an eine bestimmte Datenstruktur gebunden ist. So könnten beispielsweise auch Fotos der versendeten Kontakte in die Konfigurations-SMS eingebunden werden.

Eine SMS kommt erst dann als Konfigurations-SMS in Frage, wenn es sich dabei um eine Binär-SMS handelt und diese an den richtigen Port gesendet wurde. Ist das der Fall, überprüft der *SmsReceiver*, ob die Absendernummer autorisiert ist die Konfiguration des Handys zu ändern. Anschließend wird geprüft, ob die SMS den richtigen Aufbau hat, also mit dem richtigen Präfix beginnt und mit dem richtigen Suffix endet. Der Aufbau einer Konfigurations-SMS ist unter 4.2.2 beschrieben.

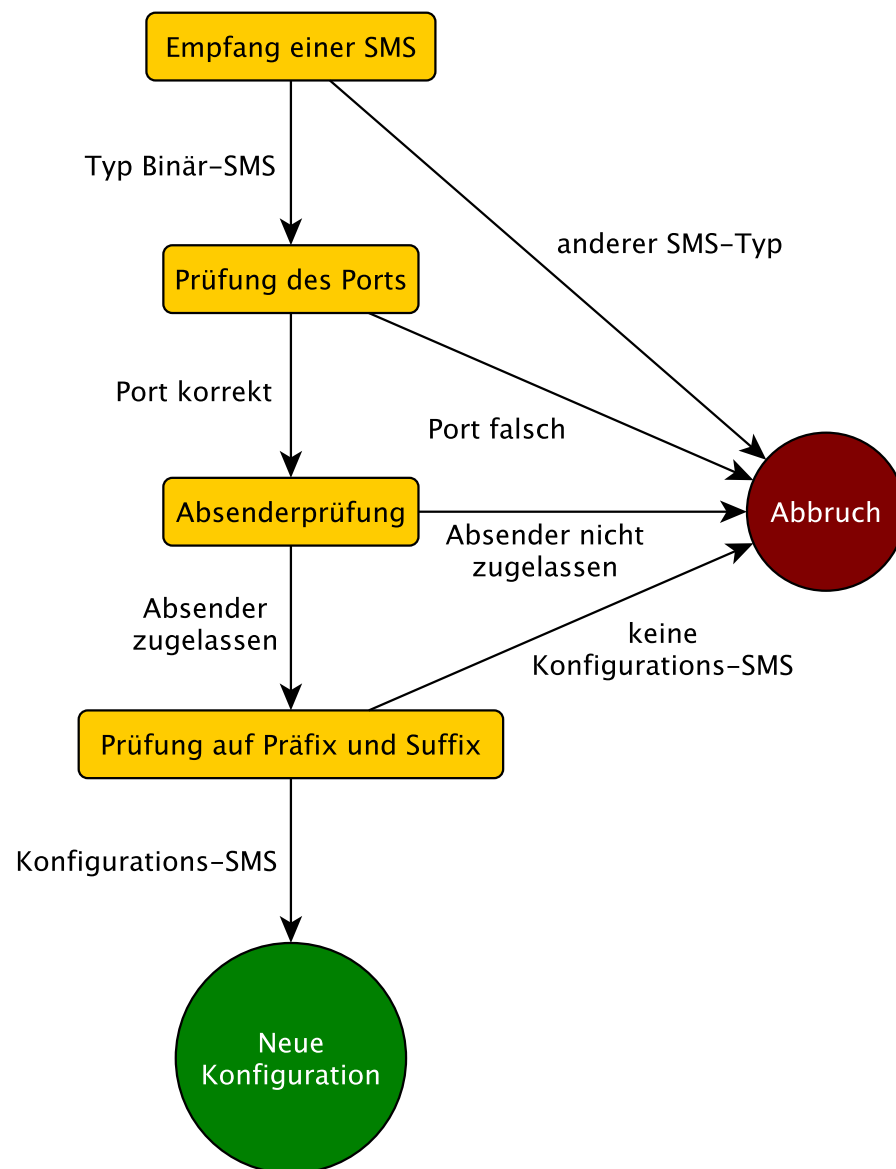


Abbildung 4.9: Vier Stufen zur Erkennung korrekter Konfigurations-SMS

4.1.6. Freisprechen

Unter 2.1.1 wurde auf altersbedingte Einschränkungen bei Senioren eingegangen. Um das Seniorenhandy auch für Personen mit eingeschränktem Hörvermögen nutzbar zu machen, ist eine Freisprechfunktion nötig. Die Funktion sollte jedoch nicht dazu führen, dass die Bedienung der App dadurch komplexer wird. Die Freisprechen-Funktion wurde deswegen als Option in den versteckten Einstellungen, sowie in der Konfigurations-App hinzugefügt. Der Nutzer kann sich somit entscheiden, ob er über die Hörmuschel des Handys oder über den eingebauten Lautsprecher des Handys telefonieren möchte. Die Einstellung wird dann im Seniorenhandy hinterlegt und kann von der Konfigurations-App jederzeit auf Wunsch des Nutzers aus der Ferne geändert werden.

4.1.7. Eigene Rufnummer wählen

Um dem Nutzer zu ermöglichen, mehr als fünf eingespeicherte Nummern anrufen zu können, wäre denkbar, dass mehrere Bildschirme mit Anrufknöpfen angeboten werden, auf die per Wischgeste nach rechts oder links zugegriffen werden kann, oder dass die Liste der Anrufknöpfe vertikal scrollbar ist. Da laut Anforderungen an die Seniorenhandy-App (2.4) aber auf Wischgesten und somit auch das Scrollen verzichtet werden soll, wurde im Rahmen der vorgelegten Arbeit keine der beiden Möglichkeiten implementiert.

Es wurde stattdessen eine Activity mit einer Tastatur zur Nummerneingabe implementiert, die es dem Nutzer ermöglicht, neben den fünf eingespeicherten Nummern auch beliebige eigene Nummer zu wählen. Diese *NumpadActivity* wird im Konfigurationsmenü des Seniorenhandys oder über die externe Konfigurations-App freigeschaltet. Ist die Funktion aktiv, wird der fünfte Button der *LauncherActivity* neu belegt und öffnet durch Antippen die Nummerneingabe.

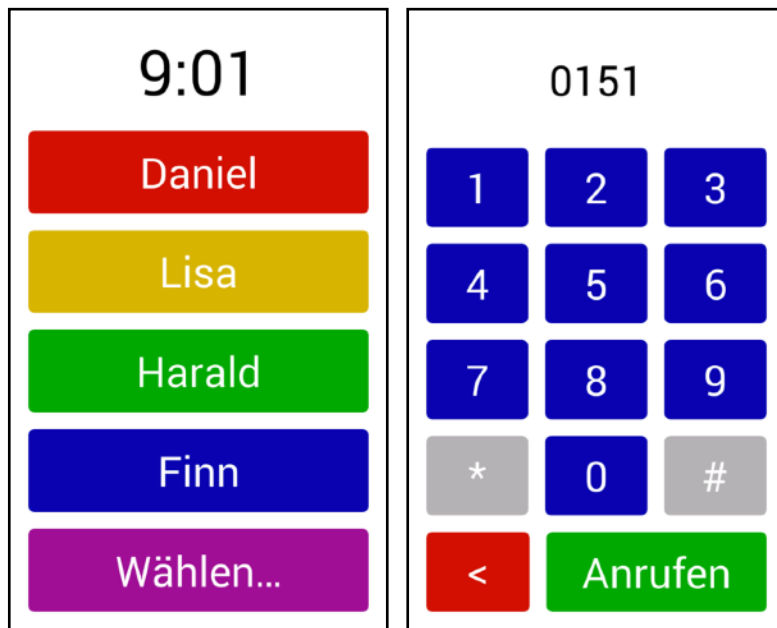


Abbildung 4.10: „Wählen...“-Button und NumpadActivity

4.1.8. Funktion „Anruf vorbereiten“

Neben der Verwaltung der Anrufbuttons durch eine autorisierte Person, wurde noch eine weitere Assistenzfunktion implementiert. Über die Konfigurations-App besteht die Möglichkeit, eine Nummer für das Seniorenhandy vorzuwählen. Diese Nummer wird dann per Binär-SMS an das Seniorenhandy gesendet. Auf diesem erscheint eine Activity *RemoteCallActivity*, welche dem Nutzer anzeigt, dass eine neue Nummer empfangen wurde. Der Nutzer kann die Meldung entweder schließen oder den Anruf direkt starten. Ein Anwendungsfall wäre, dass der Senior bei seinem Kind anruft, welches für die Verwaltung seines Seniorenhandys zuständig ist, und eine Nummer erfragt, die im Telefonbuch nicht auffindbar ist. Das Kind recherchiert im Internet und sendet seinem Vater oder seiner Mutter die entsprechende Nummer zu. Der Senior kann den Anruf direkt starten, wodurch das Diktieren und Aufschreiben der Nummer entfällt.

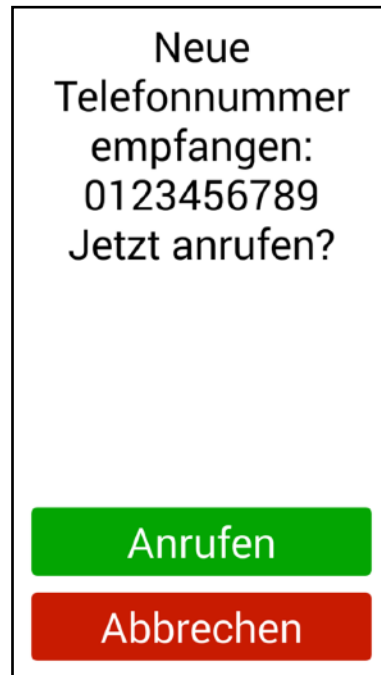


Abbildung 4.11: RemoteCallActivity

4.2. Konfigurations-App

Als Erweiterung zur Seniorenhandy-App gibt es eine eigenständige Konfigurations-App, welche auf einem anderen Gerät installiert wird. Sie dient dazu, dass ein Angehöriger des Seniors dessen Seniorenhandy aus der Ferne warten kann.

4.2.1. Main Activity und Einstellungen

Die Konfigurations-App sollte sich vom Erscheinungsbild her an der Seniorenhandy-App orientieren. Die Hauptfunktion, das Aktualisieren der fünf Wahlknöpfe, sollte klar im Mittelpunkt stehen. Dazu wurden die fünf farbigen Knöpfe vom Seniorenhandy übernommen, die sich durch Anklicken neu belegen lassen. Damit der Nutzer sofort sehen kann, ob die Buttons richtig belegt sind, zeigen diese neben dem Namen auch die Rufnummer an.

Oberhalb der Nummernknöpfe gibt es neben dem Namen der App ein kleines Zahnrad-Symbol, über das zum Einstellungsmenü gewechselt werden kann. Eine weitere Möglichkeit zum Menü zu navigieren, besteht über den Optionen-Knopf des Android-Handys. Das Einstellungsmenü bietet dieselben Einstellungen, welche auch im versteckten Einstellungsmenü des Seniorenhandys (Kapitel 4.1.2) auftauchen. So lässt sich neben dem Ändern der Rufnummer des Seniorenhandys die Freisprechfunktion, das dunkle Erscheinungsbild, sowie der Wahlknopf für eine eigene Rufnummer aktivieren.

Unterhalb der Nummernknöpfe befinden sich in der Konfigurations-App noch zwei schwarze Knöpfe. Der Erste dient dem Absenden der geänderten Einstellungen an das Seniorenhandy, über den Zweiten lässt sich ein Anruf für das Seniorenhandy vorbereiten (siehe 4.1.8).

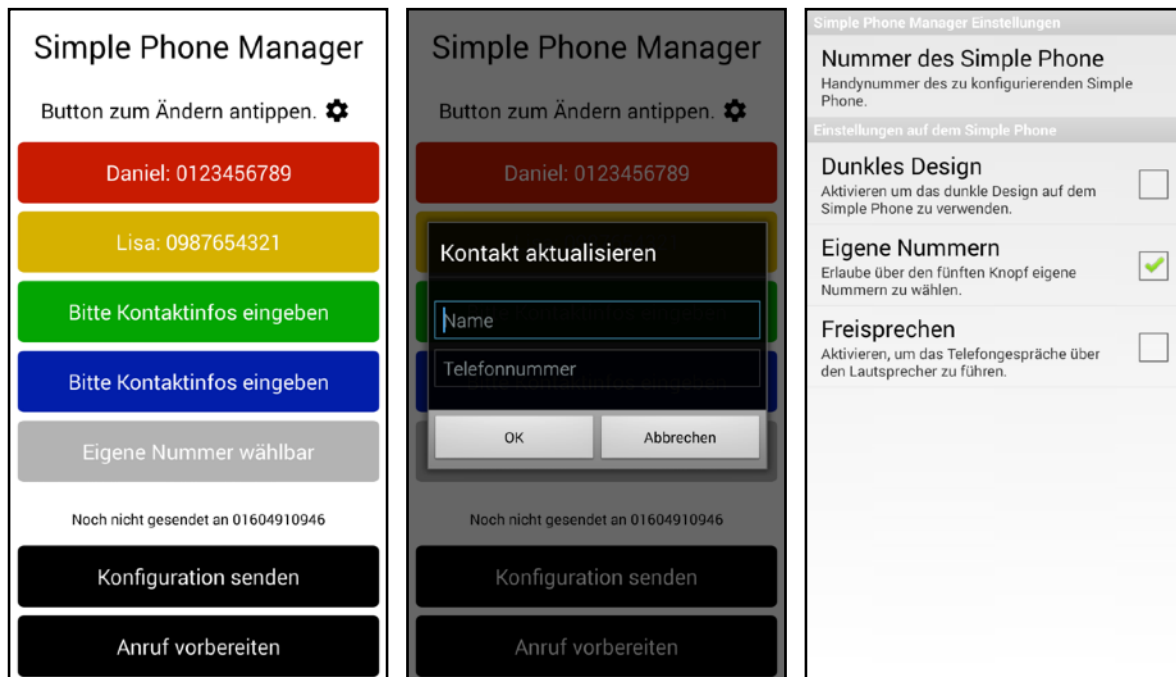


Abbildung 4.12: Konfigurations-App, „Kontakt aktualisieren“-Dialog und Einstellungen

4.2.2. Konfigurations-SMS

Konfigurations-App und Seniorenhandy kommunizieren über Binär-SMS, ohne Internetverbindung miteinander. Der Aufbau einer solchen SMS wird nachfolgend kurz behandelt. Die Sicherheitsmechanismen, welche die Kommunikation vor der Manipulation Dritter schützen, wurden bereits in 4.1.5 (Abbildung 4.9.) behandelt.

Um sicherzustellen, dass die SMS vollständig übertragen wurde, beginnt sie mit einem festen Präfix und endet mit einem festen Suffix. In der vorgelegten Arbeit wurde als Präfix „###SP“ (SP für Simple Phone) und als Suffix „###“ verwendet.

Der mittlere Teil der SMS besteht aus den verschiedenen Einstellungen, die versendet werden. Jede Einstellung beginnt mit einem Doppelpunkt, gefolgt von der Kurzbezeichnung der Einstellung. Die Bezeichnung und die verschiedenen Parameter sind jeweils durch ein Komma getrennt. Es gibt je nach Einstellung null bis zwei Parameter, die in Tabelle 4.1 dargestellt sind. Einige mögliche Konfigurations-SMS sind in der darauffolgenden Tabelle 4.2 dargestellt.

	Präfix	Bezeichnung	Parameter 1	Parameter 2
Präfix	###SP			
Button i ändern	:	bi (i von 1 bis 5)	,Name	,Nummer
Freisprechen	:	handsfree	,boolean	
Eigene Nummer	:	dialer	,boolean	
Dunkler Modus	:	dark		
Heller Modus	:	light		
Remote Call	:	call	,Nummer	
Suffix	###			

Tabelle 4.1: Aufbau von Konfigurations-SMS

Ersten Button ändern:
###SP:b1,Daniel,0123456789###
Buttons zwei und fünf neu belegen, Freisprechen aktivieren:
###SP:b2,Eugen,23456789:b5,Lisa,0987654321:handsfree,true###
Dritten Button ändern, eigene Nummerneingabe und Dunklen Modus aktivieren:
###SP:b3,Harald,75317531:dialer,true:dark###
Remote Call starten (wird alleine versendet):
###SP:call,0123456789###

Tabelle 4.2: Beispiele für Konfigurations-Nachrichten

5. Evaluation

Zur Evaluation der Seniorenhandy-App wurde die Umsetzung der unter 2.4 aufgestellten Anforderungen in der App geprüft, um die Seniorentauglichkeit festzustellen. Weiterhin wurde die App auf einigen Android Geräten mit unterschiedlichen Software-Versionen getestet. Die Funktion wichtiger Methoden wurden zudem über JUnit-Tests [34] bestätigt. Als letzter Schritt wurde ein Betriebstest durchgeführt, bei dem zwei Testpersonen die App für einige Tage testen konnten und anschließend interviewt wurden. Alle Ergebnisse sind in den folgenden Unterkapiteln aufgeführt.

5.1. Umsetzung der Anforderungen

In Tabelle 2.3 unter Kapitel 2.4 wurden Anforderungen an die Entwicklung einer Seniorenhandy-App zusammengestellt. Diese Tabelle ist nachfolgend nochmals aufgeführt, wobei markiert wurde, welche Anforderungen ganz oder nur teilweise umgesetzt wurden, sowie welche nicht umgesetzt wurden.

Die Spalte ‚Layout/Gestaltung‘ wurde fast vollständig umgesetzt. Die App verfügt über nicht mehr als fünf bedienbare Elemente pro Seite, nutzt Farben und hohe Kontraste zur Darstellung der Inhalte. Es wurde eine serifenlose Schrift mit hohem Kontrast verwendet, wobei die Schriftgröße auch auf kleinen Displays ausreichend groß ist. Schaltflächen haben auf kleinen Displays eine Höhe von einem Zentimeter, auf größeren Displays sind sie entsprechend größer. Es wurde versucht, die Schaltflächen so zu positionieren, dass sie in verschiedenen Handpositionen gut erreicht werden können. Dazu wurde der obere Teil des Display für die Anzeige von Informationen verwendet und Schaltflächen erst darunter platziert. Die meisten Schaltflächen nutzen zudem die volle Displaybreite aus. Eine Ausnahme bildet die Nummerneingabe, bei der Schaltflächen in den schlecht

erreichbaren Displaybereichen liegen. Die Benutzeroberfläche über alle implementierten Funktionen ist konsistent.

Was die Interaktion angeht, wurde auf Wischgesten zur Bedienung und Scrollbalken verzichtet. Durch die Reduktion der Funktionen und wenige beschriftete Elemente pro Seite, ist die Bedienung selbsterklärend. Jede Taste führt nur eine Aktion aus und dem Nutzer werden während der Bedienung keine Fehlerdialoge angezeigt. Die App nutzt für Übergänge zwischen Activities Systemanimationen. Diese wurden nicht als störend empfunden, könnten aber in zukünftigen Versionen der App optional auch deaktiviert werden. Um dem Nutzer eine Rückmeldung über Tastenklicks zu geben, vibriert das Gerät kurz.

Der Funktionsumfang eines Smartphones wird durch die App stark reduziert. Dabei wurden nur die Funktionen implementiert, die laut der Recherche in Kapitel 2 für Senioren wichtig sind. Die App kann zum Telefonieren genutzt werden und zeigt dem Nutzer die aktuelle Uhrzeit an. Eine Anzeige des Datums wurde in der Grundversion der App nicht implementiert. Ein Adressbuch wurde ebenfalls implementiert, beschränkt sich aber auf vier oder fünf Kontakte, je nach Einstellung. Als Assistenzfunktionen wurde die Fernkonfiguration, sowie die Funktion „Anruf vorbereiten“ (Kapitel 4.1.8) eingebaut. Die Notfalltaste und die Ortung per GPS wurde in der Arbeit nicht umgesetzt, könnten aber leicht nachgerüstet werden.

Anforderungen an die Entwicklung einer Seniorenhandy-App		
Layout / Gestaltung	Interaktion	Funktionsumfang
4-5 Elemente pro Seite ✓	keine Animationen ✗	Telefonieren ✓
Verwendung von Farben ✓	keine Gestensteuerung ✓	Datum/Uhrzeit-Anzeige ✗
hohe Kontraste ✓	keine Scrollbalken ✓	Adressbuch ✗
serifenlose Schrift, min. 14pt ✓	selbsterklärende Bedienung ✓	Notfalltaste ✗
große Schaltflächen ✓	nur eine Funktion pro Taste ✓	Assistenzfunktionen ✓
obere 30% und rechte Displayseite meiden ✗	Rückmeldung auf Interaktion ✗	Ortung per GPS ✗
konsistente Benutzeroberfläche ✓	Fehlerdialoge vermeiden ✓	

Tabelle 5.1: Anforderungstabelle 2.3, in der markiert wurde, welche Anforderungen im Rahmen der Arbeit umgesetzt wurden. Ein gelbes Kreuz bedeutet dabei, dass die Anforderung nur teilweise umgesetzt wurde.

5.2. Verhalten auf verschiedenen Geräten

Im Rahmen der Evaluation wurde ein Funktionstest der Seniorenhandy-App auf neun unterschiedlichen Geräten durchgeführt. Die neun Testgeräte basierten auf sieben verschiedenen Versionen des Android Betriebssystems. Der Schwerpunkt des Funktionstests lag auf den kritischen Programmteilen, welche durch das Abfangen von System Events (Intents) die systemeigenen Anzeigen durch die Angepassten ersetzen. Das trifft auf den Sperrbildschirm, sowie die Darstellungen für ein- und ausgehende Anrufe zu. Andere Funktionen wie das Öffnen des Einstellungsmenü oder das Empfangen einer Konfigurations-SMS wurden dennoch kurz getestet.

Auf jedem der Geräte wurde die App installiert und einmalig eingerichtet. Es wurde eine SIM-Karte mit deaktivierter PIN-Nummer eingelegt und das Gerät neu gestartet. Nach dem Neustart wurde beobachtet, ob der angepasste Sperrbildschirm wie gewünscht startet. Anschließend wurden über die Kontakt-Wahlknöpfe, sowie über die Nummerneingabe mehrere Anrufe gestartet, sowie im

gesperrten und auch im aktiven Zustand des Handys Anrufe entgegen genommen. Dabei wurde untersucht, ob sich die App auf dem Gerät wie erwartet verhält.

Die Sperrbildschirme funktionierten dabei einwandfrei und wurden sowohl nach dem Neustart des Gerätes, wie auch durch Drücken des An/Aus-Knopfes des Gerätes aktiviert und konnte nur durch Bestätigen des grünen Entsperren-Buttons auf dem Display entfernt werden. Lediglich auf dem LG G4 wurde im Test die Bildschirmsperre des Systems nicht deaktiviert und man musste somit das Gerät doppelt entsperren. Da das betroffene Testgerät im privaten Einsatz war und daher eine Vielzahl an verschiedenen Apps installiert waren, besteht die Möglichkeit, dass der angepasste Sperrbildschirm im Zusammenspiel mit den anderen Apps oder Einstellungen des Gerätes nicht funktionierte.

Weniger zuverlässig schienen die angepassten Anrufbildschirme zu funktionieren. Bei fünf von neun Geräten kam es zu einem Problem, das hauptsächlich die Darstellung von eingehenden Anrufen betraf. Seit Android 5.0 werden eingehende Anrufe vom System nicht mehr im Vollbild dargestellt, sondern als kleines Popup-Fenster am oberen Bildschirmrand. Der angepasste Anrufbildschirm der App überlagerte zwar die Anzeigen im Vollbild, scheint aber eine niedrigere Priorität zu haben, als die systemeigenen Popups. Somit wird die Anzeige ab Android 5.0 im oberen Teil durch die Systemmeldung überlagert, das Annehmen oder Ablehnen des eingehenden Anrufs ist aber über die unten dargestellten Buttons des angepassten Anrufbildschirms trotzdem möglich. Auf einem der Testgeräte, dem Samsung Galaxy A3, war ein von Samsung angepasstes Android 5.0.2 installiert. Während Android 5.0.2 wie beschrieben normalerweise ein kleines Popup für einen eingehenden Anruf anzeigt, hat der Hersteller Samsung dieses Popup in seiner Version von Android durch eine Vollbildanzeige ersetzt. Auf dem Samsung Galaxy A3 funktionierte die Darstellung des angepassten Anrufbildschirmes deshalb auch unter Android 5.0.2. Testweise wurden die unter 2.3 vorgestellten Launcher, welche ebenfalls über angepasste Anrufbildschirme verfügen, auf den problematischen Geräten mit Android 5 installiert. Es zeigte sich, dass auch diese keine Lösung für das Problem

bereitstellen, denn beide Apps wurden durch das angesprochene Popup-Fenster überlagert.

Zwei der Testgeräte zeigten beim Telefonieren am Ohr außerdem das Problem, dass das Display während dem Telefonat nicht ausgeschaltet wird. Da die App keine solche Funktionalität implementiert, sondern auf dem Verhalten des Systems beruht, ist eine mögliche Erklärung, dass durch das Überlagern der Systemapp durch die Seniorenhandy-App bei den beiden Gerätekonfigurationen auch das Systemverhalten während dem Telefonat zumindest teilweise überschrieben wird. Eine mögliche Lösung wäre, in der für die angepassten Anrufbildschirme zuständigen *AcceptCallActivity* ein eigenes Verhalten zu implementieren, welches das Display aus- und wieder anschaltet.

Bei einem der Geräte schaltete sich das Display im gesperrten Zustand bei einem eingehenden Anruf nicht ein, sodass nur der Klingelton ertönte und das Gerät vibrierte. Dieses Problem kann leicht behoben werden, indem der *PhoneListener* oder die *AcceptCallActivity* das Display zu Beginn eines eingehenden Anrufs aktiviert.

Alle weiteren Funktionen, wie das Öffnen des Konfigurationsmenüs durch die Wischgeste „S“, der Empfang einer Konfigurations-SMS oder das Wechseln zum Nummerneingabefeld funktionierten auf allen Geräten problemlos. Auf dem vier Jahre alten Samsung Galaxy Xcover erschienen eingegebene Ziffern einer eigenen Rufnummer allerdings erst nach kurzer Verzögerung. Hier könnte eventuell der Programmcode noch optimiert werden.

Android Geräte verfügen über einen Button, mit dem zur Übersicht aller geöffneten Apps gewechselt werden kann. Dieser Button lässt sich nicht deaktivieren. In vorangegangener Betrachtung sowie der nachfolgenden Tabelle wird diese Einschränkung vernachlässigt.

Die Testgeräte und Auffälligkeiten im Sperr- und Anrufbildschirm sind in der folgenden Tabelle 5.1 aufgeführt.

	Display Zoll & Pixel	Android Version	Anrufbildschirm	Sperrbildschirm
Samsung XCover	3,65 Zoll 480x320	2.3.6 (Samsung)	✓	✓
Motorola Defy	3,7 Zoll 480x854	2.2.2	✓	✓
Samsung Galaxy S2	4,3 Zoll 480x800	4.1.2 (Samsung)	✗ Display geht beim Telefonieren nicht aus	✓
Samsung Galaxy A3	4,5 Zoll 540x960	5.0.2 (Samsung)	✓	✓
LG G4	5,5 Zoll 2560x1440	5.1	✗ oberer Teil wird von System- Popup überlagert	✗ Systemsperr nicht deaktiviert
Motorola Moto G (2. Gen)	5 Zoll 720x1280	5.0.2	✗ oberer Teil wird von System- Popup überlagert, im gesperrten Zustand geht Display bei einem eingehenden Anruf nicht an	✓
Samsung Galaxy S5 mini	4,5 Zoll 720x1280	4.4.2 (Samsung)	✓	✓
Samsung Galaxy A5	5 Zoll 1280x720	4.4.4 (Samsung)	✓	✓
Samsung Galaxy Note 3	5,7 Zoll 1920x1080	4.4.2 (Samsung)	✗ Display geht beim Telefonieren nicht aus	✓

Tabelle 5.2: Testgeräte und Auffälligkeiten im angepassten Anruf- und Sperrbildschirm

5.3. Betriebstest

Um das Seniorenhandy im Alltag zu testen, wurden zwei Testpersonen mit einem Testgerät ausgestattet. Beide Testpersonen waren weiblich, 83 Jahre alt und verwendeten bereits im Alltag ein Handy. Die Testpersonen wurden zu ihrer Handynutzung befragt. Anschließend wurde ihnen als Testgerät ein Motorola Defy überreicht, auf dem die App vorinstalliert war. Entsprechend der vorausgegangenen Befragung wurde die App eingerichtet und der Testperson nach kurzer Testphase für vier Tage überlassen. Die Testpersonen wurden beauftragt, das Testgerät anstelle ihres eigenen Handys im Alltag zu verwenden. Nach Ablauf der Testperiode wurden die Testpersonen nochmals befragt. Alle Ergebnisse sind den folgenden Unterkapiteln dokumentiert.



Abbildung 5.1: Motorola Defy

Bildquelle: <http://www.cellunlocker.net/blog/wp-content/uploads/2011/09/motorola-defy.jpg>

5.3.1. Testperson 1, Ruth, 83 Jahre

Testperson 1 besitzt zwei verschiedene Handys: ein Klapphandy von Samsung, sowie ein „Handy für alle“ (Abbildung 5.1). Da das Samsung Handy nach Aussage der Testperson über zu kleine Tasten und kleine Tastenbeschriftungen verfügt, nutzt sie ausschließlich das andere Handy. Das Handy wird hauptsächlich genutzt, um Verwandte anzurufen. Die Testperson kennt alle Nummern von Kontakten auswendig, die sie über ihr Handy anruft, oder hat deren Nummern auf Kärtchen in der Handtasche immer griffbereit. Der Testperson wurden deshalb drei Buttons mit Kontakten der engen Verwandten belegt, der vierte Button mit der Nummer der für die Konfiguration zuständigen Person. Auf dem fünften Button wurde das Wählen eigener Rufnummern freigeschaltet. Außerdem wurde die Freisprechfunktion aktiviert. Die Person hatte Zeit, sich mit dem Handy auseinanderzusetzen und Fragen zu stellen. Dabei fiel auf, dass die Person oft lange auf einen Button drückte oder schnell hintereinander zwei Mal das Display berührte. Während Ersteres zu gar keiner Aktion führte, wurden bei der doppelten Berührung auch zwei Aktionen ausgeführt, sofern an der angetippten Stelle auf dem Folgebildschirm wieder ein Button platziert war.

Nach Abschluss der Testphase wurde das Testgerät mit dem Handy verglichen, welches die Testperson sonst verwendet. Die Testperson bemängelte, dass es umständlich sei, nach dem Einschalten des Displays noch auf das Schloss-Symbol zum Entsperren des Gerätes drücken zu müssen. Sie wünschte sich weiterhin als Startbildschirm nicht eine Übersicht von Kontakten zu sehen, sondern nach dem Entsperren des Gerätes direkt in der Nummerneingabe zu landen. Das Wählen einer eigenen Rufnummer dauere durch das Entsperren und Wechseln zur Nummerneingabe für sie zu lange. Wichtige Rufnummern von vier oder fünf Kontakten würde sie sich dennoch wünschen, diese sollten aber lieber über einen Knopf aufgerufen werden können, anstatt gleich zu erscheinen.

Die Bedienung empfand die Testperson als gut verständlich. Alle Beschriftungen seien ausreichend groß dargestellt, die Buttons ebenfalls. Die Testperson hatte während der Testphase nicht bemerkt, dass am oberen Bildschirmrand die Nummer

erscheint, die sie im Nummerneingabefeld wählt. Nachdem sie darauf hingewiesen wurde, bemängelte sie, dass diese größer und fettgedruckt dargestellt werden könnte. Eine Möglichkeit, einzelne Ziffern einer eingegebenen Rufnummer zu löschen, anstatt die komplette Nummer nochmals eingeben zu müssen, empfand sie als wichtig. Auch die Möglichkeit, den Freisprechmodus manuell während einem Gespräch zu deaktivieren, fehlte ihr. Als sehr gut empfand sie die Anrufbildschirme, sowie die Möglichkeit, dass gespeicherte Nummern von einer Person aus der Ferne geändert werden können. Den auf die Telefonie beschränkten Funktionsumfang empfand sie für ihre Anforderungen als ausreichend.

Insgesamt stellte die Testperson fest, dass sie ihr eigenes Handy bevorzuge, da dieses über mechanische Tasten verfügt und sich Nummern schneller wählen lassen, als bei der Seniorenhandy App.



Abbildung 5.2: „Handy für alle“

Bildquelle: https://www.weltbild.de/styx/pdf/manual/5161922_Handy_fuer_alle.pdf

5.3.2. Testperson 2, Edith, 83 Jahre

Die zweite Testperson nutzt im Alltag ein klassisches Handy von Nokia mit mechanischen Tasten, die deutlich kleiner sind, als die von Seniorenhandys. Sie gab an, das Handy hauptsächlich zu nutzen, um Freunde und Verwandte auf deren Handy anzurufen. Es wurden deshalb die fünf wichtigsten Kontakte hinterlegt, der Wahlknopf für eigene Rufnummern wurde nicht aktiviert. Die Testperson testete das Starten und Empfangen von Anrufen, was ihr keine Probleme bereitete. Es fiel auf, dass sie öfters beim Halten des Gerätes versehentlich die kapazitiven Touch-Buttons unterhalb des Displays berührte. Ein kurzes Drücken dieser Buttons führt zwar zu keiner Aktion des Gerätes, da die Testperson aber länger mit dem Finger einen der Buttons berührte, wurde ihr zwei Mal die Sprachsteuerung eingeblendet und einmal die Übersicht der gestarteten Apps. Da sich das lange Drücken der Buttons nicht deaktivieren lässt, kann hier nur durch ein Gerät mit mechanischen Tasten Abhilfe geschafft werden, bei dem die Funktionen nicht durch alleiniges Berühren der Buttons gestartet wird. Ein Problem, welches bei Testperson 1 bereits beobachtet wurde, trat auch bei Testperson 2 auf. Teilweise wurde zweimal schnell hintereinander auf einen Button gedrückt, sodass das Gerät zwei unterschiedliche Aktionen erkannte. Wurde beispielsweise der letzte Kontakt angerufen und dann noch ein zweites Mal an der Stelle auf das Display getippt, wurde der Anruf direkt wieder beendet, da sich der „Beenden“-Button des Anrufbildschirmes an der gleichen Position befindet wie der unterste Kontakt-Button.

Die Testperson kam während der Testphase sehr gut mit dem neuen Handy zurecht. Obwohl die Testperson über ein gutes Sehvermögen verfügt und auch ohne Brille die deutlich kleinere Schrift ihres Nokia Handys lesen kann, empfand sie die Textgröße auf dem Testgerät als sehr angenehm. Die Buttons waren gut zu bedienen und alle Abläufe und Meldungen klar verständlich. Die Testperson empfand als sehr positiv, nur einen Kontakt auf dem Display des Gerätes berühren zu müssen, anstatt die Nummer wählen zu müssen oder umständlich über das Menü des Handys zu den Kontakten zu navigieren. Sie fand das Verhalten deutlich einfacher als auf dem Vergleichshandy von Nokia. Auf die Frage, ob sie die

Kontakte gerne selbst ändern wollte, entgegnete sie, dass sie sich mit technischen Geräte zu wenig auskenne und diese Aufgabe deshalb lieber einer anderen Person überlasse. Die Möglichkeit der Fernkonfiguration gefiel ihr daher sehr gut, ebenfalls die Möglichkeit auf Anfrage eine Nummer zugesendet zu bekommen, die sie dann direkt anrufen kann. Da die Testperson nach ihrer eigenen Aussage mit einem Handy nur telefoniere, genügte ihr außerdem der Funktionsumfang der App.

5.3.3. Zusammenfassung des Betriebstests

Der durchgeführte Betriebstest und die Befragung der Testpersonen decken sich zu großen Teilen mit den unter 2.1.2 festgestellten Erwartungen von Senioren an Handys. Insgesamt bestätigen die Ergebnisse die unter 2.1 aufgestellten und in der App umgesetzten Anforderungen an die Seniorenhandy-App.

Aus den Testergebnissen lassen sich allerdings unterschiedliche Anforderungen der beiden Testpersonen an ein Handy ableiten. Während Testperson 1 gerne möglichst schnell eine Nummer wählen möchte, fand Testperson 2 es sehr komfortabel nur den Namen der Person antippen zu müssen. Unter Berücksichtigung der Vergleichsgeräte beider Personen, fällt auf, dass Testperson 1 mit ihrem Vergleichsgerät sehr zufrieden ist und sich das vom Vergleichsgerät bekannte Verhalten für die Seniorenhandy-App wünscht. Testperson 2 hingegen nutzte das Testgerät lieber, weil es über größere Schaltflächen verfügt und leichter zu bedienen ist als das Vergleichshandy. Zukünftige Versionen der Seniorenhandy-App könnten somit die Möglichkeit bieten, den Startbildschirm von der Kontaktübersicht auf die Nummerneingabe umzustellen.

6. Schlussbetrachtung

In dieser Arbeit wurde eine App für Smartphones entwickelt, die den Anspruch hat, Senioren oder Menschen mit Einschränkungen die Verwendung eines Smartphones zu ermöglichen, ohne sich mit der Bedienung auseinandersetzen zu müssen. Das Ergebnis ist eine Launcher App, die den Startbildschirm von Android Geräten ersetzt und dem Nutzer ermöglicht, über Schnellwahltasten oder die Nummerneingabe, schnell Anrufe zu tätigen oder entgegenzunehmen. Die App verfügt dabei über einige Assistenzfunktionen, die es einer autorisierten Person erlauben, das Seniorenhandy aus der Ferne über eine Konfigurations-App zu warten. Um die Bedienung des Seniorenhandys konsistent zu halten, fängt die App Events des Systems ab und überlagert die systemeigenen Anwendungen wie den Sperrbildschirm oder die Anrufbildschirme durch eigene Ansichten. Neuere Android-Versionen bereiten dabei Probleme, da sie bei eingehenden Anrufen ein kleines Popup-Fenster anzeigen, welches nicht von anderen Apps überlagert werden kann. Die Buttons zum Annehmen oder Ablehnen des Telefonates werden dabei jedoch nicht überlagert und können somit trotzdem bedient werden. Android Geräte verfügen über einen Button, über den zur Ansicht der aktuell geöffneten Apps gewechselt werden kann. Dieser Button lässt sich nicht deaktivieren, sodass der Nutzer bei versehentlicher Berührung in der App-Übersicht landet.

Insgesamt ist eine App entstanden, die es nahezu vollständig schafft, die Benutzeroberfläche von Android zu ersetzen. Auf anderen mobilen Plattformen wie iOS oder Windows Phone wäre das nicht möglich gewesen, da die Plattformen den Funktionsumfang von Apps deutlich einschränken.

In der Evaluation wurde abschließend festgestellt, dass die App ihrem Anspruch gerecht wird und sehr leicht zu bedienen ist.

6.1. Ausblick

Der Fokus der entwickelten Seniorenhandy-App lag auf einer selbsterklärenden Bedienung und der Beschränkung der Funktionalität auf die wichtige Grundfunktion ‚Telefonie‘ und einige Assistenzfunktionen wie die Fernkonfiguration. Da Smartphone-Apps nahezu unbegrenzte Möglichkeiten bieten, sind viele Erweiterungen der App möglich.

Weitere Assistenzfunktionen wie ein Notrufbutton, die Ortung über GPS oder die Benachrichtigung der autorisierten Person über einen niedrigen Akkustand oder die Unterschreitung einer bestimmten Guthabengrenze der Prepaid-SIM-Karte können ohne großen Aufwand ergänzt werden. Denkbar wäre auch ein Falldetektor, der einen automatischen Notruf absetzt, falls das Handy zu Boden fällt.

Im Hinblick auf weitere Funktionen könnte die Möglichkeit ergänzt werden, Textnachrichten zu schreiben. Der Einbau einer Erinnerungsfunktion oder ein Wecker könnte Personen ansprechen, die ihr Handy über das Telefonieren hinaus verwenden wollen. Interessant wäre auch, die App um eine Sprachausgabe und Spracherkennung zu erweitern, damit sie auch von blinden Menschen genutzt werden kann.

Literaturverzeichnis

Alle gelisteten Webseiten wurden zuletzt am 11.12.2015 besucht.

- [1] J. Morgan Morris. „User interface design for older adults“. *Interacting with Computers*, 6(4), 373-393 (1994). URL: <http://iwc.oxfordjournals.org/content/6/4/373.full.pdf>
- [2] Wen-Ming Hsu, Ching-Yu Cheng, Jorn-Hon Liu, Su-Ying Tsai, und Pesus Chou. „Prevalence and causes of visual impairment in an elderly chinese population in taiwan: The shihpai eye study“. *Ophthalmology*, 111(1):62 – 69 (2004). URL <http://www.sciencedirect.com/science/article/pii/S0161642003011552>.
- [3] William Kosnik, Laura Winslow, Donald Kline, Kenneth Rasinski, und Robert Sekuler. „Visual changes in daily life throughout adulthood“. *Journal of Gerontology*, 43(3):63–70 (1988). URL <http://geronj.oxfordjournals.org/content/43/3/P63.abstract>.
- [4] K. Chen, Alan H. S. Chan, und Steve N. H. Tsang. „Usage of Mobile Phones amongst Elderly People in Hong Kong“. *Proceedings of the International MultiConference of Engineers and Computer Scientists 2013*, vol. 2 (2013). URL: http://www.iaeng.org/publication/IMECS2013/IMECS2013_pp1016-1019.pdf
- [5] Sri Kurniawan, Yanuar Nugroho, und Murni Mahmud. „A Study of the Use of Mobile Phones by Older Persons“. *CHI 06 Extended Abstracts on Human Factors in Computing Systems*, pp. 989 - 994 (2006). URL: <http://dl.acm.org/citation.cfm?id=1125641>
- [6] Tanid Phiriyapokanon. „Is a big button interface enough for elderly users?“. Masterarbeit, Mälardalen University (2011). URL: <http://www.idt.mdh.se/utbildning/exjobb/files/TR1091.pdf>
- [7] Panayotis Zaphiris, Mariya Ghiawadwala, Shabana Mughal. „Age-centered Research-Based Web Design Guidelines“. *CHI 05 Extended Abstracts on Human Factors in Computing Systems*, pp. 1897 - 1900 (2005). URL: <http://dl.acm.org/citation.cfm?id=1057050>
- [8] Steven Hoober. *How do Users Really Hold Mobile Devices?* (2013) URL: <http://www.uxmatters.com/mt/archives/2013/02/how-do-users-really-hold-mobile-devices.php>
- [9] Emporia. *emporiaCONNECT*. URL: <http://www.emporia.de/produkte/uebersicht/emporiaconnect>
- [10] Doro. *Doro Secure 580IUP*. URL: <http://www.doro.co.uk/mobile-devices/easy-phones/doro-secure-580iup.html>

- [11] Gartner. *Gartner Says Annual Smartphone Sales Surpassed Sales of Feature Phones for the First Time in 2013* (2013). URL: <http://www.gartner.com/newsroom/id/2665715>
- [12] Doro. *Doro PhoneEasy 740*. URL: <http://www.doro.ch/doro-phoneeasy-740.html>
- [13] handytarife.de. *Doro PhoneEasy 740 im Test: Smartphone für Senioren*. URL: <http://www.handytarife.de/?aid=3276>
- [14] Doro. *Doro Liberto 825*. URL: <http://www.dorodeutschland.de/doro-liberto-825.html>
- [15] Seniors Phone. URL: <http://www.seniorsphone.mobi>
- [16] KoalaPhone. URL: <http://www.koalaphone.com/>
- [17] Fontrillo. URL: <http://www.fontrillo.com>
- [18] Phonotto: Simple Phone Launcher. URL: <http://www.phonotto.com>
- [19] Java. URL: <http://www.java.com>
- [20] Eclipse. URL: <http://www.eclipse.org>
- [21] Android. *Installing the Eclipse Plugin*. URL: <http://developer.android.com/sdk/installing/installing-adt.html>
- [22] Sofia. *Structure of an Android App*. URL: <http://sofia.cs.vt.edu/sofia-2114/book/chapter2.html>
- [23] Android. *App Manifest*. URL: <http://developer.android.com/guide/topics/manifest/manifest-intro.html>
- [24] Android. *Package Index*. URL: <http://developer.android.com/reference/packages.html>
- [25] Android. *Activity*. URL: <http://developer.android.com/reference/android/app/Activity.html>
- [26] Android. *<intent-filter>*. URL: <http://developer.android.com/guide/topics/manifest/intent-filter-element.html>
- [27] Android. *Intents and Intent Filters*. <http://developer.android.com/guide/components/intents-filters.html>
- [28] Arnab Chakraborty. *How to Write Custom Launcher App in Android*. URL: <http://arnab.ch/blog/2013/08/how-to-write-custom-launcher-app-in-android/>
- [29] Android. *<receiver>*. URL: <http://developer.android.com/guide/topics/manifest/receiver-element.html>
- [30] Android. *<service>*. URL: <http://developer.android.com/guide/topics/manifest/service-element.html>
- [31] Google Play. *Easy Home*. URL: <https://play.google.com/store/apps/details?id=com.factisresearch.easyhome>
- [32] Android. *TelephonyManager*. URL: <http://developer.android.com/reference/android/telephony/TelephonyManager.html>

- [33] Android. *KeyguardManager.KeyguardLock*. URL: <http://developer.android.com/reference/android/app/KeyguardManager.KeyguardLock.html>
- [34] JUnit. URL: <http://www.junit.org>