NearbyChat

Tim Lock, Linus Kurze

Hochschule Osnabrück  
Fakultät Ingenieurwissenschaften und Informatik  
Barbarastr. 16, D-49076 Osnabrück  
tim.lock@hs-osnabrueck.de  
linus.kurze@hs-osnabrueck.de

17. Januar 2023

**Zusammenfassung**

Die App NearbyChat bietet eine verbindungslose Kommunikation in Form von Chats zwischen mehreren Nutzern in der Nähe an. Nutzer haben Profile und getätigte Chats können gespeichert werden. Für die Realisierung der Kommunikation zwischen den Geräten wird ein Mesh, basierend auf Bluetooth Low Energy Advertisements aufgebaut. Dadurch wird die geringere Reichweite von Bluetooth deutlich erweitert.

# Einleitung

[TODO]

* Aufbau der Arbeit

# Konzept

Das Konzept lässt sich in mehrere Teilkonzepte unterteilen: Das Basiskonzept, welches die App dem Nutzer bieten soll, das Design mit Logo und Namen, das Interaktionskonzept, welches die Interaktion des Nutzers mit der App beschreibt, das Konzept der Bluetooth-Kommunikation zwischen den Geräten und das Datenmodell zur Abbildung der zu persistierenden Daten.

## Basiskonzept

Das Basiskonzept von „NearbyChat“ sieht vor, dass mehrere Smartphones über Bluetooth Low Energy Advertisements ein Mesh bilden. Über dieses Mesh soll ein Austausch von Nachrichten geschehen. Diese Nachrichten sollen zur Einordnung den Zeitpunkt des Absendens enthalten. Darüber hinaus sollen ebenfalls Profile ausgetauscht werden. Diese Profile sollen erste Informationen über den Nutzer bieten. Sie werden genutzt, um verfügbare Nutzer oder auch gespeicherte Chats anzuzeigen. Ein Profil beinhaltet einen Namen und eine kurze Beschreibung. Aufgrund der Tatsache, dass sich über Bluetooth Low Energy Advertisements nur wenige Daten in ausreichend schneller Zeit übertragen lassen, ist der Inhalt von Nachrichten auf 64 Zeichen begrenzt. Der Name im Profil ist auf 16 und die Beschreibung auf 32 Zeichen begrenzt. In den meisten Profilen würde ein Profilbild verwendet werden. Da dies aber aufgrund der geringen Übertragungsrate nur sehr langsam übertragen werden könnte, ist eine Profilfarbe eingeführt worden. Diese bietet bei geringem Übertragungsaufwand eine hohe Individualität.

Neben den oben beschriebenen Hauptfunktionen soll die App Ergonomisch sein: Die Bedienung soll einfach und intuitiv von statten gehen. Außerdem soll zwischen einem hellen und einem dunklen Design, sowie den Sprachen Deutsch und Englisch gewählt werden können.

Ein diesem Basiskonzept ähnliches Konzept ist den Autoren zum Zeitpunkt nicht bekannt. Es gibt zwar viele Dating-Apps, wie zum Beispiel Tinder, die auch den Standort beschränken können, allerdings handelt es sich nicht um eine vom Gerät aus lokale Suche nach verfügbaren Nutzern [1]. Darüber hinaus nutzt Tinder zur Übertragung der Daten das Internet und nicht Bluetooth [1]. In der Funktion etwas ähnlicher sind Apples AirDrop und das Pendant Nearby Share von Android. Sie verwenden Bluetooth und WLAN zur Übertragung von Daten, wie Fotos oder Videos [2] [3]. Sie können theoretisch auch zur Übertragung von Nachrichten genutzt werden, jedoch ist die Übertragung von Daten das eigentliche Ziel.

## Design, Logo & Name

Das Design der App umfasst ein grundlegendes Farbschema und ein Logo. Das Farbschema umfasst ein helles Rot und ein dunkles Rot. Neben diesen Primärfarben sind im Design nur Weiß, Schwarz und Grautöne zugelassen. Mit Ausnahme der Profilfarben werden auch keine weiteren Farben verwendet. Rot als Designfarbe hat sich aus mehreren Gründen als vorteilhaft erweisen: Zum einen nutzt kein bekannter großer Messenger die Farbe Rot: Facebook Messenger und Signal sind blau [4] [5], WhatsApp ist grün [6], Threema ist schwarz/grün [7] und Snapchat ist gelb [8]. Zum anderen bietet die Farbe Rot eine Vielzahl passender Assoziationen: Rot ist unter anderem die Farbe der Liebe und Leidenschaft [9]. Dies passt gut in den Kontext des Kennenlernens in beispielsweise einem Club oder auf einer Party.

Die Primärfarben finden sich auch im Logo der App wieder. Es besteht aus zwei Sprechblasen, die in den beiden Primärfarben eingefärbt sind. Die eine Sprechblase ist von links ausgerichtet, die andere von rechts. Dies symbolisiert den Austausch zweier Personen und damit exakt die Kernfunktion der App. Der Hintergrund ist weiß und das Logo weist keine weiteren Elemente oder Verzierungen auf. Das Logo ist daher auch in kleineren Größen leicht zu erkennen und bietet trotzdem ausreichende alleinstellungsmerkmale gegenüber den Logos oben erwähnten Messengern.

Der Name „NearbyChat“ beschreibt direkt die Hauptfunktion der App: Das kommunizieren mit sich in der Nähe befindlichen Menschen. Er ist simpel und kurz und daher trotz der Sprache Englisch auch für Deutschsprachige leicht verständlich. Darüber hinaus bietet der Name einen guten Wiedererkennungswert und könnte aufgrund der Internationalität auch in anderen Ländern eingesetzt werden.

## Interaktionskonzept

Die Interaktion des Nutzers mit der App beginnt mit dem Starten dieser. Dabei wird eine Art Splash-Screen angezeigt, während dem zunächst sämtliche Berechtigungen eingeholt werden. Darüber hinaus werden, wenn nicht bereits geschehen, Bluetooth und die Ortungsdienste eingeschaltet, damit die App vollständig funktionsfähig ist. Das Einschalten und der Zugriff auf die Ortungsdienste wird, ganz ähnlich zur Corona-App, benötigt [10]. Android fordert diese Berechtigungen, da über Bluetooth prinzipiell Informationen über den Standort erhoben werden können [11]. Falls die App feststellt, dass sie nicht mit dem Gerät kompatibel ist, da es kein Bluetooth Low Energy Advertising unterstützt, wird dies ebenfalls auf dem Splash-Screen angezeigt.

Wenn das Gerät mit der App kompatibel ist, werden nach dem Splash-Screen in einer Navigationsleiste am unteren Ende des Bildschirms vier Tabs angezeigt: Der Tab „Verfügbar“, welcher die aktuell verfügbaren Nutzer anzeigt, der Tab „Chats“, welcher die gespeicherten Chats mit anderen Nutzern anzeigt, der Tab Profil, in dem sich das eigene Profil konfigurieren lässt und der Tab „Einstellungen“, in dem sich die Einstellungen der App tätigen lassen. Die Wahl Art der Navigation zwischen diesen verschiedenen Ansichten der App ist auf Tabs gefallen, da diese gut geeignet sind, um unabhängige Informationen und Interaktionsmöglichkeiten, die sich inhaltlich in entsprechende Tabs unterteilen lassen abzubilden [12, p. 522]. Des Weiteren sind Tabs gut geeignet, wenn die Verwendungsreihenfolge der Inhalte variieren kann, was in diesem Fall ebenso gegeben ist [12, p. 522].

Der Inhalt der Tabs wird jeweils auf dem Bildschirm zwischen der Navigationsleiste und einer sich am oberen Bildschirmrand befindlichen Informationsleiste angezeigt. Neben der farblichen Hervorhebung des aktuell ausgewählten Tabs ist in der Navigationsleiste, unter dem App-Namen eine Kurzbeschreibung des angewählten Tabs zu sehen.

Der Tab „Verfügbar“ zeigt eine sich ständig aktualisierende Liste der aktuell erreichbaren Profile, jeweils mit Namen, Farbe und Beschreibung an. Die Profilfarbe wird mithilfe eines Symbols, einem generischen Avatar auf einer Standortmarkierung angezeigt. Diese sind nach Empfangsstärke sortiert, welche darüber hinaus jeweils durch ein Symbol an der rechten Seite angezeigt wird.

Der Tab „Chats“, der die gespeicherten Chats anzeigt, ist hingegen nach dem Zeitpunkt der letzten Interaktion sortiert. Hier wird, neben dem Profilnamen und dem Zeitpunkt der letzten Interaktion, die Empfangsstärke des jeweiligen Profils angezeigt. Zur einfachen Unterscheidung von dem Tab „Verfügbar“ ist die Profilfarbe auf einem anderen Symbol, nämlich einer Sprechblase, die mit Text gefüllt ist, angezeigt. Am linken Rand der Profile wird ein roter Punkt angezeigt, wenn eine neue empfangene Nachricht mit diesem Profil ungelesen ist. In diesem Tab können einzelne Chats durch Ziehen von links nach rechts gelöscht werden. Da dies eine kritische Aktion ist, wird dem Nutzer durch einen Banner am unteren Bildschirmrand eine Bestätigung angezeigt. Dieser Banner beinhaltet darüber hinaus eine Schaltfläche, um das Löschen rückgängig zu machen.

Der Tab „Profil“ beinhaltet die Konfiguration des eigenen Profilnamens und der Beschreibung durch Texteingabefelder. Diese sind zur leichten Verständlichkeit mit „Name“ und „Beschreibung“ betitelt. Darunter befindet sich die Konfiguration der Profilfarbe. Ein großes Feld unter der Überschrift Farbe zeigt die aktuell ausgewählte Farbe an. Darunter befinden sich 10 Farbfelder, aus denen sich die Farbe auswählen lässt. In diesem Tab werden vor Änderungen immer die aktuell gespeicherten Profileigenschaften angezeigt. Ein großer Button mit dem Text „Speichern“ ermöglicht das Speichern des eigenen Profils. Nach dem Speichern wird, durch einen Banner am unteren Bildschirmrand, eine Bestätigung angezeigt.

Der Tab „Einstellungen“ beinhaltet Einstellungen zur Darstellung und zur Sprache. Unter der Überschrift „Darstellung“ werden die Optionen „Dunkler Modus“, „Heller Modus“ und „Systemstandard“ als Optionsfelder angezeigt. Unter „Sprache“ sind es die Optionsfelder „Englisch“, „Deutsch“ und „Systemstandard“. Es kann für jede Einstellung immer nur ein Optionsfeld angewählt sein. Darüber hinaus wird immer die aktuell angewandte Einstellung angezeigt. Unter der Spracheinstellung folgt noch ein Hinweis, dass die Spracheinstellungen erst nach einem Neustart der App angewendet werden.

Aus den Tabs „Verfügbar“ und „Chats“ lässt sich durch Antippen eines Profils der Chat mit diesem Profil öffnen. Durch Verwendung des „Zurück“-Buttons von Android gelangt man zurück. In der Informationsleiste am oberen Bildschirmrand befindet sich auch hier, neben dem App-Namen eine Kurzerklärung der aktuellen Ansicht. Direkt darunter befindet sich eine weitere Leiste, die das Profil des Users, deren gemeinsamer Chat geöffnet ist, mit Namen, Farbe, Beschreibung und Empfangsstärke anzeigt. Am unteren Ende des Bildschirms befindet sich ein Eingabefeld, für neue Nachrichten. Direkt rechts neben dem Eingabefeld befindet sich ein Button zum Senden einer Nachricht. Der verbleibende Platz wird genutzt, um den Nachrichtenverlauf anzuzeigen. Die Nachrichten werden von neu, am unteren Ende nach alt, am oberen Ende sortiert. Dabei werden eingehende Nachrichten links, mit der Uhrzeit rechts und ausgehende Nachrichten rechts, mit der Uhrzeit links angezeigt. Die Farbe des Nachrichtenfelds variiert ebenfalls, wenn es sich um eine ausgehende oder eingehende Nachricht handelt. Zwischen den Nachrichten wird, nur wenn es sich geändert hat, das Datum angezeigt. Bei ausgehenden Nachrichten wird, wenn sie empfangen worden sind, unten rechts ein kleiner Haken angezeigt. Standardmäßig sind die Nachrichten nach ganz unten, zu den neusten Nachrichten gescrollt. Wenn der Nutzer weiter nach oben gescrollt hat, wird rechts unten ein Button zum Herunterscrollen angezeigt. Des Weiteren wird neben diesem Button, wenn eine neue Nachricht ankommt und der Nutzer nicht ganz heruntergescrollt hat, ein roter Punkt als Hinweis angezeigt.

Im Großen und Ganzen orientiert sich das Design stark an etablierten Designs von Messengern, wie WhatsApp [6], Signal [4], Facebook Messenger [5] oder Threema [7]. Das Design ist responsiv und funktioniert daher unter verschiedenen Auflösungen. Auch das Wechseln zwischen Portrait und Landscape ist ohne Verlust eingegebener Daten oder Verlust der Scroll-Position möglich.

## Konzept der Bluetooth-Kommunikation

[TODO]

## Datenmodell

Für die App sollen Nachrichten und das eigene Profil, sowie fremde Profile persistiert werden. Die Entitäten sind daher „Message“, „OwnProfile“ und „Profile“.

Eine Nachricht enthält, neben dem Inhalt einen Zeitstempel und eine Adresse. Die Adresse ist immer die des Fremden Profils, an das oder von dem aus Nachrichten gesendet werden. Darüber hinaus wird gespeichert, ob die Nachricht angekommen ist und ob sie selbst verfasst oder vom Gegenüber geschrieben wurde. Der Primärschlüssel setzt sich aus der Adresse, dem Zeitstempel und ob die Nachricht selbstgeschrieben ist, zusammen. Somit können sämtliche Nachrichten eindeutig identifiziert werden, da kein Nutzer mehrere Nachrichten zum exakt selben Zeitpunkt versenden kann. Nachrichten werden vollständig in der Datenbank persistiert.

Das eigene Profil enthält die eigene Adresse, den Namen, die Beschreibung und die Farbe. Die zehn wählbaren Farben werden durch eine Zahl zwischen null und neun einschließlich repräsentiert. Der Primärschlüssel ist die eigene Adresse, da diese immer eindeutig ist. Darüber hinaus wird allerdings nie mehr als ein eigenes Profil gespeichert. Genauso, wie Nachrichten wird das eigene Profil vollständig in der Datenbank gespeichert.

Fremde Profile enthalten, genauso wie das eigene, die Adresse, den Namen die Beschreibung und die Farbe. Darüber Hinaus enthalten sie den Zeitstempel der letzten Interaktion, also einer gesendeten oder empfangenen Nachricht und einen Wert, ob es noch ungelesene Nachrichten gibt oder nicht. Ein fremdes Profil umfasst allerdings noch zwei weitere Attribute, die nicht in der Datenbank persistiert werden: Der Hop-Count, also die Anzahl der Geräte auf dem Weg durch das Mesh zum Ziel, und die Received Signal Strength Indication, kurz RSSI, zur Beschreibung der Empfangsstärke. Diese Informationen werden nur zur Laufzeit persistent gehalten.

# Realisierung

Zur Realisierung werden viele Funktionen von Android Jetpack verwendet. Android Jetpack ist eine Menge an Libraries für die Entwicklung von Android. Es bietet viele Funktionen zum einfachen und schnellen Umsetzen einer App, die mit mehreren Android Versionen kompatibel ist. Darüber hinaus lässt sich mit Jetpack der Code schlanker halten und Best Practices besser umsetzen [13].

Im Folgenden wird auf die Besonderheiten der Realisierung im Bereich Softwaredesign, User Interface, Bluetooth-Kommunikation und Datenhaltung eingegangen.

## Softwaredesign

[TODO]

## Realisierung des User Interfaces

Die Besonderheiten bei der Realisierung des User Interfaces beginnen bereits beim Splash-Screen. Dieser wird durch eine dezidierte Aktivität umgesetzt. Er muss die Besonderheiten von Android zur Anforderung von Permissions und zur Initiierung von Änderungen an den Einstellungen beachten: Wenn die App Permissions über das System anfordern lässt, werden diese nur einmalig durch das System abgefragt und bei Ablehnung durch den Benutzer auch bei erneuter Initiierung nicht erneut abgefragt. Daher muss die App nach dem Versuch der Anforderung von Permissions über das System eine alternative Behandlung für fehlende Permissions aufweisen. Diese wird konkret durch eine Popup-Benachrichtigung mit Verlinkung zu den entsprechenden Einstellungen umgesetzt. Die Aktivierung von Bluetooth erfolgt durch eine Anfrage an das System, welches nach Bestätigung des Nutzers Bluetooth aktiviert. Für die Ortungsdienste hingegen bietet das System keine Möglichkeit der Abfrage in dieser Form. Daher muss auch hier eine eigen Popup-Benachrichtigung erstellt werden, die auf die entsprechende Stelle in den Einstellungen verweist. Allgemein werden Permissions und Änderungen von Einstellungen nur gefordert, wenn sie nicht bereits erteilt bzw. angepasst sind. Sie werden allerdings auch so lange gefordert, bis sie erteilt bzw. angepasst sind, da die App ohne sie nicht funktionsfähig ist. Die Popup-Benachrichtigungen werden durch den „AlertDialog“ von Android Jetpack realisiert.

Die Ansicht mit den vier Tabs „Verfügbar“, „Chats“, „Profil“ und „Einstellungen“ wird in einer Aktivität umgesetzt. Hierbei ist hervorzuheben, dass die Navigation durch die „BottomNavigationView“ von Googles Material Design realisiert wird. Material Design bietet, unter Anderen für Android, viele Komponenten, Guidelines und Best Practices im Bereich User Interface Design [14]. Die Änderung des Inhalts und somit der eigentlichen Tabs wird durch Fragmente aus Android Jetpack umgesetzt. Sie bieten die Möglichkeit andere Inhalte beziehungsweise Layouts zu laden, ohne eine neue Aktivität aufrufen zu müssen. Des Weiteren wird die „ActionBar“ von Android Jetpack zur Realisierung der Informationsleiste am oberen Bildschirmrand verwendet.

Die Tabs „Verfügbar“ und „Chats“ verwenden in ihren Fragmenten sogenannte „RecyclerViews“ zur Darstellung der Listen von Profilen. „RecyclerViews“ sind ebenfalls ein Teil von Android Jetpack. Für die beiden Tabs ist jeweils ein Adapter, der auf Änderungen an den Daten reagiert, indem er diese visualisiert, implementiert worden. Des Weiteren bietet der Adapter einen Click-Listener an. Dadurch kann bei Klicken auf ein Profil der Chat mit dem entsprechenden User geöffnet werden.

Das Fragment des Tabs „Chats“ implementiert einen Listener auf Zieh-Gesten reagiert. Dieser wird durch den „ItemTouchHelper“ von Android Jetpack zur Verfügung gestellt. Durch diesen Listener kann das Entfernen von Chats realisiert werden. Ein Banner, realisiert durch die „SnackBar“ des Material Design, bestätigt das Löschen. Die „SnackBar“ ermöglicht darüber hinaus durch Antippen eines Textes das Rückgängig-Machen des Löschens auszulösen.

Die Fragmente zu den Tabs „Profil“ und „Einstellungen“ verwenden die Android eigenen „ScrollViews“, damit die Inhalte auch bei verschiedenen Displayhöhen und im Portrait- sowie Landscape-Modus angezeigt werden können.

Im Fragment des Tabs „Profil“ wird der „SavedInstanceState“ verwendet, um die Profilfarbe bei erneutem Laden des Fragments anzeigen zu können. Dies geschieht beispielsweise bei dem Wechsel von Portrait- zu Landscape-Modus oder andersherum. Andere Änderungen des Nutzers, wie zum Beispiel eingaben von Texten in die Textfelder werden automatisch beibehalten. Die Zeichenbegrenzung des Namens und der Beschreibung ist durch eine entsprechende Eigenschaft der Textfelder umgesetzt. Das Speichern des Profils wird mit einer „SnackBar“ aus dem Material Design bestätigt.

Die Einstellungen, die im gleichnamigen Tab getätigt werden können, werden durch das Fragment an die darunter befindliche Aktivität weitergeleitet. Diese nutzt die Android eigenen „SharedPreferences“, um die Einstellungen jeweils als Key-Value-Paar zu speichern und wendet diese auch an. Die Anwendung geschieht bei den Sprachen durch „Locale“ und bei dem Farbschema durch „AppCompatDelegate“.

Die Chats mit einem User werden in einer eigenen Aktivität geöffnet. Diese verwendet ein „ConstraintLayout“ von Android Jetpack, um sowohl die zweite Leiste am oberen Bildschirmrand als auch das Texteingabefeld, den Senden-Button und den Herunterscroll-Button korrekt positionieren zu können. Auch hier ist die Zeichenanzahl der Nachricht durch eine entsprechend gesetzte Eigenschaft des Texteingabefelds beschränkt. Das Profil des anderen Users wird durch die Aktivität bei Änderung aktualisiert. Die Chatnachrichten werden, wie in den Tabs „Verfügbar“ und „Chats“ durch einen „RecyclerView“ realisiert. Der dazugehörige Adapter übernimmt neben Visualisieren von Änderungen, wenn Nachrichten dazu kommen, auch die Aufgabe, die Nachricht, je nachdem, ob sie ein- oder ausgehend ist, auf der richtigen Seite, mit der richtigen Farbe anzuzeigen. Außerdem wird durch Prüfen des Zeitstempels der vorangegangenen Nachricht das Datum nur bei einer Änderung angezeigt. Wenn das Datum nicht angezeigt werden muss, kollabiert das Textfeld für das Datum, sodass es keinen Platz verbraucht. Die Aktivität kann durch einen Scroll-Listener, beim Scrollen die aktuelle Scroll-Position abfragen, und dadurch Prüfen, ob der Button zum Herunterscrollen und bei Bedarf auch der Punkt als Indikator für eine neue Nachricht angezeigt werden soll. Darüber hinaus behält die Aktivität die Scroll-Position immer bei, es sei denn es wird eine neue Nachricht empfangen und die Scroll-Position war vorher ganz unten. In diesem Fall scrollt die Aktivität erneut nach ganz unten, sodass die neue Nachricht zu sehen ist. Darüber hinaus hebt die Aktivität, wenn die neusten Nachrichten gesehen wurden, den Ungelesen-Vermerk des Profils auf.

## Realisierung der Bluetooth-Kommunikation

[TODO]

## Realisierung der Datenhaltung

Die Realisierung der Datenhaltung arbeitet auf zwei Ebenen: Zum einen werden die Daten, die in der Datenbank persistiert werden sollen mit der Library „Room“ aus Android Jetpack in eine lokale Datenbank geschrieben. Zum anderen werden die zur Laufzeit zu persistierenden Daten mit „LiveData“, ebenfalls ein Teil von Android Jetpack, persistent gehalten. „LiveData“ ermöglicht es, einfach auf Änderungen zu reagieren. Da „LiveData“ auch mit Room kompatibel ist, werden auch die Daten aus der Datenbank in der App in „LiveData“ verwaltet. Änderungen an diesen Daten müssen allerdings durch gesonderte Funktionen auf der Datenbank ausgeführt werden und können nicht an den „LiveData“ Objekten vorgenommen werden. „LiveData“ bietet für die Daten der Datenbank darüber hinaus den Vorteil, dass die Zugriffe auf die Datenbank lose gekoppelt sind, sodass die Blockade von Android, dass nicht vom UI-Thread synchron auf die Datenbank zugegriffen werden darf, nicht greift.

Besonders ist, bei der Umsetzung der Datenhaltung, das Repository hervorzuheben. Es verwaltet mit Hilfe sogenannter Data Access Objekt, kurz DAOs, die Daten aus der Datenbank und führt Änderungen asynchron über die DAOs auf der Datenbank aus. Darüber hinaus verwaltet es aber auch die zur Laufzeit persistenten Daten und aktualisiert sämtliche Daten, wenn es nötig ist: Kommt beispielsweise eine Nachricht an, oder wird gesendet, so wird geprüft, ob das zugehörige Profil bereits gespeichert ist. Wenn dies nicht der Fall ist, wird es aus den verfügbaren Profilen übernommen und gespeichert, oder wenn auch dies nicht möglich ist, wird ein leere Platzhalterprofil erstellt. Ist die Nachricht eine eingehende Nachricht, so wird die Markierung für ungelesene Nachrichten auf dem Profil gesetzt. Sämtliche gespeicherten Profile werden, wenn sie verfügbar sind und die verfügbaren Profile unterschiede gegenüber den gespeicherten aufweisen, durch das Repository aktualisiert. Dies betrifft nur die Eigenschaften Name, Beschreibung, Farbe, RSSI und Anzahl der Hops, nicht jedoch den letzten Interaktionszeitpunkt oder die Markierung für ungelesene Nachrichten. Ändert sich ein gespeichertes Profil, so wird dieses, wenn verfügbar, immer um die RSSI und die Anzahl der Hops ergänzt, sodass auch in gespeicherten Chats die aktuelle Empfangsstärke angezeigt werden kann.

Das ViewModel erfüllt ebenfalls besondere Aufgaben: Zum einen kapselt es die Logik des Repositorys durch einfache Methoden für die verschiedenen Views. Zum anderen ist es für die Verwaltung des Chat-Services verantwortlich. Es stellt sicher, dass der Service gestartet wird und reagiert auf Änderungen durch den Service, beispielsweise wenn ein neues Profil entdeckt wird oder eines verschwindet. Diese Änderungen reicht das ViewModel an die Datenbank weiter.

# Tests

[TODO]

## Tests des User Interfaces und der Datenhaltung

Die Tests des User Interfaces wurden mit Tests der Datenhaltung kombiniert: Es wurden Methoden, zum Laden von Testdaten implementiert. Diese Testdaten wurden durch das UI dargestellt und konnten auch bearbeitet, ergänzt oder gelöscht werden. Durch diese Testdaten war ein Testen bereits vor Fertigstellung der Bluetooth-Kommunikation möglich.

Die konkreten Tests wurden dann mit mehreren Personen an einem Smartphone mit aufgespielter App und den Testdaten umgesetzt. Dabei haben die Personen sich zunächst allein zurechtfinden und ein paar Aufgaben erfüllen sollen. Beispiele für solche Aufgaben sind: Das eigene Profil anpassen, eine Nachricht an einen Nutzer versenden, oder einen Chat mit einem Nutzer löschen. Beim Erfüllen dieser Aufgaben haben die Personen alles erwähnt, dass ihnen aufgefallen ist. Dadurch konnte unter Anderem identifiziert werden, dass die ursprünglich verwendete Markierung von ungelesenen Chats durch einen roten Hintergrund, als unschön und schlechter lesbar empfunden wurde, weshalb der kleine rote Punkt am Rand eingeführt wurde. Darüber hinaus viel der Bedarf nach einem Butten zum Herunterscrollen in den Chats auf und dass das eigene Profil ohne das Ermöglichen des Scrollens im Landscape-Modus nicht auf jedem Gerät vollständig bearbeitet werden kann, da der Speichern-Button abgeschnitten war.

Auch nach Fertigstellung der Bluetooth-Kommunikation wurden Tests durch Anwendung durchgeführt. Hierbei haben zwei Personen über die App kommuniziert oder ihr Profil angepasst und es wurde geprüft, ob die Änderungen oder neuen Nachrichten, wie erwartet dargestellt werden. Auch die Markierung für ungelesene Chats wurde getestet. Bei diesen Tests viel allerdings nur kleiner Korrekturbedarf an der Datenhaltungslogik auf.

## Tests der Bluetooth-Kommunikation

[TODO]

# Installation

[TODO]

* Beachten: min API, Kompatibilität (meldet die App aber auch)

# Zusammenfassung & Ausblick

[TODO]

# Literaturverzeichnis

|  |  |
| --- | --- |
| [1] | Tinder, „Tinder,“ 12 Januar 2023. [Online]. Available: https://play.google.com/store/apps/details?id=com.tinder&hl=de&gl=US. [Zugriff am 16 Januar 2023]. |
| [2] | N. Hery-Moßmann, „So funktioniert AirDrop - einfach erklärt,“ Chip, 23 Juli 2022. [Online]. Available: https://praxistipps.chip.de/so-funktioniert-airdrop-einfach-erklaert\_99963. [Zugriff am 16 Januar 2023]. |
| [3] | I. Bauer, „Nearby Share auf Android - so funktioniert's,“ Heise, 16 Mai 2022. [Online]. Available: https://www.heise.de/tipps-tricks/Nearby-Share-auf-Android-so-funktioniert-s-7095779.html. [Zugriff am 16 Januar 2023]. |
| [4] | Signal Foundation, „Signal - Sicherer Messenger,“ 11 Januar 2023. [Online]. Available: https://play.google.com/store/apps/details?id=org.thoughtcrime.securesms&gl=DE. [Zugriff am 16 Januar 2023]. |
| [5] | Meta Platforms Inc., „Messenger,“ 13 Januar 2023. [Online]. Available: https://play.google.com/store/apps/details?id=com.facebook.orca&gl=DE. [Zugriff am 16 Januar 2023]. |
| [6] | WhatsApp LLC, „WhatsApp Messenger,“ 5 Dezember 2022. [Online]. Available: https://play.google.com/store/apps/details?id=com.whatsapp&gl=DE. [Zugriff am 16 Januar 2023]. |
| [7] | Threema GmbH, „Threema,“ 5 Januar 2023. [Online]. Available: https://play.google.com/store/apps/details?id=ch.threema.app&gl=DE. [Zugriff am 16 Januar 2023]. |
| [8] | Snap Inc, „Snapchat,“ 12 Januar 2023. [Online]. Available: https://play.google.com/store/apps/details?id=com.snapchat.android&gl=DE. [Zugriff am 16 Januar 2023]. |
| [9] | A. Soboth, „Farbe Rot: Bedeutung und Wirkung der Farbe Rot,“ Focus Online, 5 Februar 2019. [Online]. Available: https://praxistipps.focus.de/farbe-rot-bedeutung-und-wirkung-der-farbe-rot\_107802. [Zugriff am 16 Januar 2023]. |
| [10] | Stiftung Warentest, „Corona-App - Stand­ort-Zugriff auf Android-Handys,“ 17 Juni 2020. [Online]. Available: https://www.test.de/Corona-App-Standort-Zugriff-auf-Android-Handys-5624423-0/. [Zugriff am 16 Januar 2023]. |
| [11] | Google Inc., „Bluetooth Permissions,“ 12 Januar 2023. [Online]. Available: https://developer.android.com/guide/topics/connectivity/bluetooth/permissions. [Zugriff am 16 Januar 2023]. |
| [12] | W. O. Galitz, The Essential Guide to User Interface Design, Hoboken: Wiley, 2007. |

# Arbeitsaufteilung

|  |  |
| --- | --- |
| *Beitrag* | *Verantwortliche(r)* |
|  |  |
|  |  |
|  |  |
|  |  |

# Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne Verwendung anderer als den angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

*oder*

Hiermit erklären wir, dass wir die vorliegende Arbeit selbstständig und ohne Verwendung anderer als den angegebenen Hilfsmittel angefertigt haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Tim Lock)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Linus Kurze)

Osnabrück, den 17. Januar 2023