

Follow My Friend Android-App

Erleichterung der Navigation zu einem Freund
anhand der Positionsübertragung in Echtzeit und Routeneinzeichnung

Projekt zum Thema Fortgeschrittene Mensch-Maschine-Interaktion von Martin Heller und Farah Tahir
an der Technischen Hochschule Mittelhessen im Sommersemester 2014

1. Einleitung

1.1 Motivation und Problemstellung

Es gibt unterschiedliche Android-Apps, die auf Wunsch Standorte von Freunden liefern. Ein Beispiel dafür sei die populäre Chat-Anwendung "Viber": Mit einer eigenen Nachricht kann man zusätzlich den eigenen Standort teilen. Der Standort wird auf Google Maps angezeigt¹.

Bis zum heutigen Stand stellt keine mobile Anwendung eine Funktion bereit, mit der eine Route auf Google Maps eingezeichnet wird, die sich aus den Standortdaten von dem Anwender und seinem Freund bildet. Somit würde man dem Anwender die Navigation zu seinem Freund erleichtern. Eine solche Anwendung könnte beispielsweise nützlich sein, wenn man sich an einem Treffpunkt verabredet hat, aber am vereinbarten Ort nicht findet.

1.2 Zielsetzung

Das Ziel der Android-Anwendung Follow My Friend soll sein, dem Anwender eine Navigation zum Freund anzubieten und ihm dadurch zu helfen leicht zu Fuß an sein Ziel zu gelangen. Dabei sollen die Standortdaten des Anwenders und des Freundes aktuell gehalten werden, sodass man auch bei minimaler Standortänderung zu dem Freund gelangt.

1.3 Vorüberlegungen

Die App Follow My Friend soll als eine Art der Kommunikation und Austausch zwischen Freunden dienen. Daher muss eine Kontaktliste als Grundlage für die Anwendung dienen. Damit der Benutzer die eigene Kontaktliste mit möglichst geringem Aufwand beziehen kann, sollte dieser Vorgang vorzugsweise automatisch ablaufen, sodass er nach der Installation sofort die App verwenden kann.

Es bietet sich in dem Sinne Folgendes an: Jeder Freund, der die Anwendung installiert hat und in einem Freundeskreis (z.B. soziales Netzwerk o.ä.) eingetragen ist, wird automatisch zu der Kontaktliste der App hinzugefügt.

Den Freundeskreis könnte man umsetzen, indem man entweder auf ein soziales Netzwerk zurückgreift oder die lokal abgespeicherten Telefonkontakte als Grundlage nimmt. Damit man mehr Freiheiten in der technischen Umsetzung hat, wird die zweite Variante der lokalen Kontaktliste bevorzugt.

Außerdem ist zu berücksichtigen, dass eine Anwendung dieser Art nur dann genutzt werden kann, wenn sichergestellt ist, dass der Benutzer mit dem Internet verbunden ist und seinen Standortzugriff aktiviert hat.

1.4 Anforderungsanalyse

Aus der Zielsetzung und den Überlegungen kann die Anwendung in drei wesentliche Bestandteile aufgeteilt werden: Registrierung der eigenen Telefonnummer, Kontaktliste und Kartenanzeige.

Außerdem können folgende Anforderungen abgeleitet werden:

- Man muss eine Lösung finden eine Verbindung zwischen zwei Benutzern der Anwendung aufzubauen, damit sie ihre Standortdaten austauschen können.
- Die Anwendung soll im Hintergrund für den Benutzer herausfinden, wer von seinen lokalen Kontakten die App installiert hat, um sie in der Kontaktliste der App anzuzeigen.
- In der Kontaktliste sollte erkennbar sein, wer momentan erreichbar ist.
- Der Benutzer sollte in einer geeigneten Form Benachrichtigungen erhalten, wenn er eine neue Positionsanfrage eines Freundes erhält, auch wenn seine App inaktiv ist.
- Man benötigt eine Karte, auf der man zwei Positionen markieren und die Route anzeigen lassen kann.
- Man muss die Route ermitteln und einzeichnen können.

2. Verwandte Arbeiten

Eine ähnliche Anwendung für Android-Geräte bietet Family Safety Production unter dem Namen Finde Meine Freunde². Diese App zeigt den genauen Standort von Freunden an. Es lassen sich u.a. Gruppen bilden und sowohl die aktuellen als auch die veralteten Standorte (Historie) aller

¹ Viber - Location Sharing:
http://support.viber.com/customer/portal/articles/1415447-what-is-location-sharing-#.VEjs_vmsVW8

² Finde Meine Freunde - App:
<https://play.google.com/store/apps/details?id=com.fsp.android.friendlocator>

Gruppenmitglieder anzeigen. Zahlreiche weitere Apps bieten den Austausch von Positionsdaten unter Freunden an, sowohl für Android als auch iOS.

Jedoch bietet bisher keine Anwendung eine Echtzeitnavigation mit Einzeichnung einer Route zu einem bestimmten Freund an.

3. Methodik

In diesem Abschnitt wird erläutert, wie man die technische Lösung für das geschilderte Problem entwickelt hat.

3.1 Methodenauswahl

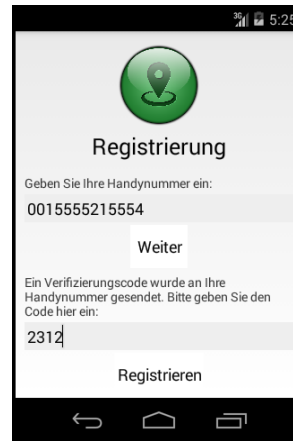
Aus der Anforderungsanalyse wurden folgende Methoden bzw. Komponenten ausgewählt:

- die Android-XMPP-Bibliothek aSmack³, mit der man die Kommunikation zwischen den Benutzern und die Präsenzanzeige (online oder offline) realisiert hat.
- einen Service, der die Kommunikation und die Notifications im Hintergrund verwaltet, damit die App auch dann funktioniert, wenn sie inaktiv ist.
- eine Datenbank, in der man festhält, welcher Benutzer sich in der App registriert hat. Mithilfe dieser erfolgt die automatische Kontaktlistengenerierung.
- Notifications und Dialogfragmente, um den Benutzer in geeigneter Form über Positionsanfragen bzw. die Annahme/Ablehnung der Anfragen zu benachrichtigen.
- ein Google Maps API V2 Fragment, um die Positionen mit Markern und die Routen mit Polygonlinien anzuzeigen.
- einen Google Directions API Parser⁴, um die Routen und die entsprechenden Polygonlinien ermitteln zu können.

Für die Entwicklung der App wurde die IDE Android Studio von Google in der Version 0.8.9 Beta verwendet. Außerdem wurde eine Versionierung mit git auf der Plattform GitHub durchgeführt⁵, um die Entwicklung im Team zu ermöglichen. In der IDE ist eine entsprechende Versionierungsunterstützung integriert.

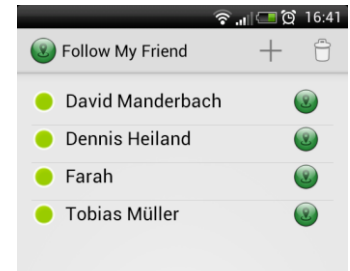
3.2 Entwicklung des Bedienkonzepts

Bei der Entwicklung des Bedienkonzepts hat man den Fokus darauf gelegt dem Benutzer möglichst viel Arbeit abzunehmen, ohne ihn aber dabei einzuschränken.



Schon bei der Registrierung wird versucht die Telefonnummer auszulesen und automatisch einzutragen, sofern diese im Telefonmanager gespeichert wurde. Auch der zugesendete Verifizierungscode wird aus der SMS ausgelesen und in das entsprechende Feld eingetragen.

In der Kontaktliste reicht ein Klick auf den Kontakt, um eine Anfrage an den Freund zu senden.



In der Map wird anfangs automatisch so herangezoomt, dass beide Standortpositionen gerade sichtbar sind. Außerdem wird die Map mit jedem Positionsupdate so rotiert, dass die eigene Position unten ist und die Position des Freundes oben ist, sodass der Benutzer das Gefühl hat, dass er von sich aus in Richtung des Freundes schaut. Dem Benutzer bleibt dabei die Freiheit rein und raus zu zoomen und die Map zu verschieben.

Um die Benutzerschnittstelle möglichst effektiv zu gestalten wurden folgende UI-Komponenten verwendet:

- Menü-Icons in einer ActionBar
- Notifications
- Dialogfragmente

Falls z.B. Einstellungen, wie die Aktivierung eines Internet- oder Standort-Providers notwendig sind, wird dies dem Benutzer über Dialogfragmente angezeigt. Bei Bestätigung eines solchen Dialogs öffnet sich die entsprechende Einstellungs-Activity, damit der Benutzer die Einstellung nicht selbst suchen muss. Wenn der Benutzer sich bei diesen beiden

³ aSmack Version 4.0.3 von: <http://asmack.freakempire.de/4.0.3/>

⁴ Google Maps Directions API Parser von: <http://wptrafficanalyzer.in/blog/driving-route-from-my-location-to-destination-in-google-maps-android-api-v2/>

⁵ Link zum Repository: <https://github.com/Corvax-Fisher/FindMyFriend>

Dialogen allerdings dagegen entscheidet, wird die App geschlossen, weil sie ohne Internet und Standortdaten nicht funktioniert.

3.3 Accountregistrierung und Generierung der Kontaktliste

Bei der Registrierung wird im Hintergrund die Telefonnummer zum einen in eine angelegte Datenbank eingetragen, damit später festgestellt werden kann, wer die App installiert hat. Das ist für die Generierung der Kontaktliste wichtig. Zum anderen wird die Nummer in einer SharedPreferences-Datei gespeichert, sodass die Anwendung sich automatisch einloggt und die Kontaktliste lädt.

Außerdem wird ein XMPP(chem. Jabber)-Account mit einem zufallsgenerierten Passwort registriert, um die Kommunikation über den Server zu ermöglichen. Die benötigte Jabber-ID (Benutzername) setzt sich aus der Telefonnummer und den App-Initialen zusammen, sodass diese mit Leichtigkeit abgeleitet werden kann.

Die abgespeicherten Nummern in der Datenbank werden mit der lokalen Telefon-Kontaktliste auf dem Smartphone verglichen. Die Übereinstimmungen werden in einen ListAdapter eingetragen und mittels einer ListView als Kontaktliste angezeigt.

3.4 Notifications

Um den Benutzer über bestimmte Ereignisse zu benachrichtigen, hat man Notifications verwendet. Es handelt sich um folgende Ereignisse:

- Positionsanfrage wurde erhalten
- Positionsanfrage wurde abgelehnt
- Positionsanfrage wurde angenommen

Die Notifications wurden so implementiert, dass sie sich nicht gegenseitig überschreiben, sondern ggf. separat angezeigt werden. Der Benutzer wird durch Vibration und Ton auf diese aufmerksam gemacht.

Da der Benutzer auch auf die Idee kommen könnte, nicht auf eine Notification zu klicken, sondern die App selbst zu öffnen, musste mehr Aufwand betrieben werden, um die logische Navigation der App aufrecht zu erhalten. Dafür werden die Notifications verwaltet und ggf. aktualisiert oder abgebrochen.

3.5 Implementierung des Kommunikationsprotokolls

Für die Implementierung des Kommunikationsprotokolls wurde die erwähnte XMPP-Bibliothek für Android namens aSmack verwendet. Die Nachrichten wurden in Gruppen aufgeteilt:

- Positionsanfrage wurde erhalten
- Positionsanfrage wurde abgelehnt
- Positionsanfrage wurde angenommen

- Positionsdaten

Für jede Nachrichtengruppe gibt es ein bestimmtes Format. Die ersten Drei bestehen lediglich aus einem Buchstaben, um den Nachrichten-Overhead so gering wie möglich zu halten. Für eine Anfrage wird ein "P" (Positionsanfrage) gesendet, für eine Ablehnung ein "N" (Not Acknowledged) und für eine Annahme ein "A" (Acknowledged). Die Positionsdaten werden im Format Längengrad:Breitengrad (z.B. 50.3306803:8.7585043) gesendet und beim Empfänger entsprechend decodiert. Auch hier wird der Overhead möglichst klein gehalten.

Die Positionsdaten werden in einem Mindestabstand von 5 Sekunden ermittelt und verschickt, sodass eine Echtzeitnavigation ermöglicht werden kann. Die Positionen werden nur verschickt, wenn der Benutzer sich in der Map-Activity befindet. Um festzustellen, ob dies der Fall ist, werden Präsenznachrichten versendet. Wechselt also ein Benutzer in die Map-Activity, wird eine Präsenznachricht mit dem Status "Available:Map" gesendet. Somit weiß die Gegenseite, dass die Positionsdaten gesendet werden sollen.

Mit den Präsenznachrichten soll auch festgestellt werden, ob ein Benutzer in der Kontaktliste gerade verfügbar ist für eine Positionsanfrage. Um dies zu erkennen wird der Status einer Präsenznachricht darauf überprüft, ob er "Available" enthält. Der Status wird nur dann "Available" enthalten, wenn ein Benutzer eine aktive Internetverbindung hat und die Standortdaten aktiviert hat.

3.6 Zusätzliche Features

Damit die Anwendung möglichst ergonomisch für den Benutzer ist, wurden zusätzliche Erweiterungen eingebaut:

- Hilfestellung: Da die Anwendung nicht für alle Benutzer selbsterklärend sein könnte, wird anfangs eine Hilfestellung angeboten. Diese kann auf Wunsch deaktiviert werden.
- Freund einladen: Die Applikation kann mit anderen Kontakten geteilt werden. Dabei wird die öffentliche Google Play-Store URL zur App versendet.
- Account löschen: Mit dieser Option kann der Benutzer seinen XMPP-Account und seinen Datenbankeintrag löschen lassen.
- Alternativrouten: Es werden dem Anwender auf Wunsch weitere Routen zum Freund angezeigt.

4. Evaluation und Ergebnisse

Zum Testen der Usability der App wurde ein Evaluationsbogen erstellt und ausgewertet. Es wurden etwa 40 Teilnehmer angefragt, davon stellten sich 11 Personen für eine Evaluation bereit. Im Folgenden werden die Ergebnisse der Evaluation präsentiert und anschließend ausgewertet.

- **Alter der Probanden**
Zwischen 20 und 28 Jahren
- **Android-Versionen der Probanden**
1x Ice Cream Sandwich, 3x Jelly Bean, 7x KitKat
- **Nutzungsverhalten der Probanden**
Oft/täglich/stündlich
- **Besitz einer Internet-Flatrate**
Haben alle Probanden bejaht
- **Aufgetretene Probleme**
Bei 8 Probanden traten Probleme auf, bei 3 Probanden nicht.
Probleme: keinen Standort des Freundes erhalten, Absturz der App in der Map, Fehler bei der Registrierung, keine Server-Verbindung, eingezeichnete Route ist verschwunden, Kontaktliste ist verschwunden
- **Erfüllt die App ihren Zweck?**
Haben alle Probanden bejaht
- **Ist die App für dich nützlich?**
8x Ja, 3x Nein
- **Würdest du die App herunterladen, wenn sie öffentlich ist?**
6x Ja, 2x Vielleicht, 3x Nein
- **Hast du Verbesserungswünsche?**
Die Verbesserungswünsche sind in das Kapitel 6 (Ausblick) eingeflossen
- **Bewertung der App als Ganzes**
4 von 5 Sternen

Es lassen sich aus dem Ergebnis der Evaluation viele Schlussfolgerungen für die weitere Entwicklung der Anwendung ziehen. Die Grundidee der Anwendung hat allen Probanden gefallen und mehr als die Hälfte wären bereit, eine solche Anwendung in ihrem Alltag zu nutzen.

Es traten jedoch auch viele Probleme während der Tests auf. Das liegt vor allem auch daran, dass die Anwendung noch nicht ausgereift ist und nicht vollständig in allen möglichen Testfällen benutzt wurde.

Aus der Evaluation konnte erfasst werden, wie wichtig es ist eine solche Anwendung testen zu lassen und nach dem Test genug Zeit einzuräumen, die Ergebnisse in die Fehlerbehebung und weitere Verbesserung der Anwendung einfließen zu lassen.

5. Zusammenfassung

Ziel und Zweck der Anwendung war es, einem Benutzer zu ermöglichen, mit möglichst wenig Aufwand zu Fuß zu einem Freund zu navigieren. Im Großen und Ganzen kann man aus den Evaluationsergebnissen und Tests ableiten, dass der Zweck der Anwendung erfüllt wurde und die Grundidee gut angekommen ist. Die App ist allerdings in vielen kleinen Hinsichten ausbaufähig.

6. Ausblick

Bei der Evaluation wurden die Probanden nach Verbesserungsvorschlägen gefragt und einige haben auch welche angegeben. Mit den Vorschlägen und eigenen Ideen lässt sich die App in den Aspekten Usability, Effektivität und Zufriedenheit entsprechend verbessern. Diese sind im Folgenden, gruppiert in die genannten Aspekte, aufgelistet:

- **Usability**
 - Map/Navigation sollte sich nicht schließen, wenn das Display aus ist oder man kurz die App verlässt
 - Positionsladialog sollte nach bestimmter Zeit abbrechen und ein Feedback geben
 - Internationalisierung der Anwendung
 - Einheitliche Nummerneingabe/Ländercode
 - Auswahl der Navigation zu Fuß/mit dem Auto
- **Effektivität**
 - Anzeige der Entfernung des Freundes in Weg und Zeit
 - Kontrolle der eigenen Standortfreigabe, nachdem man einmal die Position freigegeben hat
- **Zufriedenheit**
 - Aktion/Dialog bei Erreichen des Ziels
 - Zweite Map mit allen Freunden in der Kontaktliste, die Ihren Standort freigegeben haben
 - Positionshistorie

Durch die Evaluation hat man erkannt, wie wichtig ein Test der Anwendung mit verschiedenen Benutzern sein kann. Man sollte sich die Zeit so einplanen, dass man nach einer Evaluation noch genügend Zeit hat, um die Ergebnisse bestenfalls noch umsetzen zu können.

Nachdem die aufgezählten Verbesserungen implementiert sind, kann man von einer gelungenen Anwendung sprechen und die erste Version im Google Play Store veröffentlichen.

Die Probanden, die die Anwendung getestet haben, waren alle im Alter von 20 bis 28 Jahren. In Zukunft könnte man durch weiteres Evaluieren herausfinden, welche weiteren Zielgruppen (Jüngere und/oder Ältere, etc.) an dieser Anwendung interessiert sind.