

ESA - Proximety

Projektdokumentation



Fernfachhochschule Schweiz
Zürich | Basel | Bern | Brig

Mitglied der SUPSI

Jonas Alder, Patrick Bösch
Sandro Dallo, Andy Villiger
FFHS Bsc Inf 2011
Version 0.6

17.09.2014

Dokumentenverlauf

Version	Wer	Bemerkungen	Datum
0.1	DAL	Initialer Entwurf	17.09.2014
0.2	DAL	UseCase Diagramm hinzugefügt	19.09.2014
0.3	BOE	Kapitel Allgemeines ausformuliert	21.09.2014
0.4	BOE	GUI Prototypes Einleitung hinzugefügt, Flow Chart hinzugefügt und Fehler gemäss Rückmeldung Dozent korrigiert	19.10.2014
0.5	DAL	GUI Prototypes Variante 1 hinzugefügt, Korrekturen am Dokument vorgenommen	19.10.2014
0.6	ALO	GUI Prototypes Variante 2 und Vergleich hinzugefügt	20.10.2014

Inhaltsverzeichnis

Inhaltsverzeichnis	3
1 Allgemeines	4
1.1 Einleitung	4
2 Anforderungen	4
2.1 Funktional	4
2.1.1 Use Case Diagramm	4
2.1.2 Use Case Beschreibungen	4
2.1.2.1 UC01 - Account registrieren	4
2.1.2.2 UC02 - Freundschaftsanfrage versenden	5
2.1.2.3 UC03 - Freundschaft auflösen	6
2.1.2.4 UC04 - Aktueller Standort eines Freundes abfragen	7
2.1.2.5 UC05 - Proximity Alarm	7
2.1.2.6 UC06 - Freund kontaktieren	8
2.1.2.7 UC07 - Freund-Spezifische Einstellungen anpassen	9
2.1.2.8 UC08 - Allgemeine App Einstellungen vornehmen	9
3 GUI Prototypes	10
3.1 Einleitung	10
3.2 Variante 1	10
3.3 Variante 2	16
3.4 Vergleich und Entscheid	20
A Abbildungsverzeichnis	21
B Tabellenverzeichnis	21
Quellenverzeichnis	22

1 Allgemeines

1.1 Einleitung

Dieses Dokument beinhaltet die Beschreibung und Analyse des Projektes Proximety, welches im Rahmen des Moduls *Embedded Systems mit Android (ESA)*, *BSc INF 2011.ZH1, HS14/15, Seeliger Karl* an der Fernfachhochschule Schweiz (FFHS) erstellt wird. Das Projektteam besteht aus Jonas Alder, Patrick Bösch, Sandro Dallo und Andy Villiger.

Proximety ist eine auf der von Google entwickelten Plattform Android lauffähige Applikation, welche sich hauptsächlich den Informationen des im Smartphone befindlichen GPS-Chips bedient.

Der Benutzer kann in der Applikation Freunde einer Liste hinzufügen und das Programm meldet ab diesem Zeitpunkt, wenn sich ein solcher Freund in der Umgebung aufhält.

2 Anforderungen

2.1 Funktional

2.1.1 Use Case Diagramm

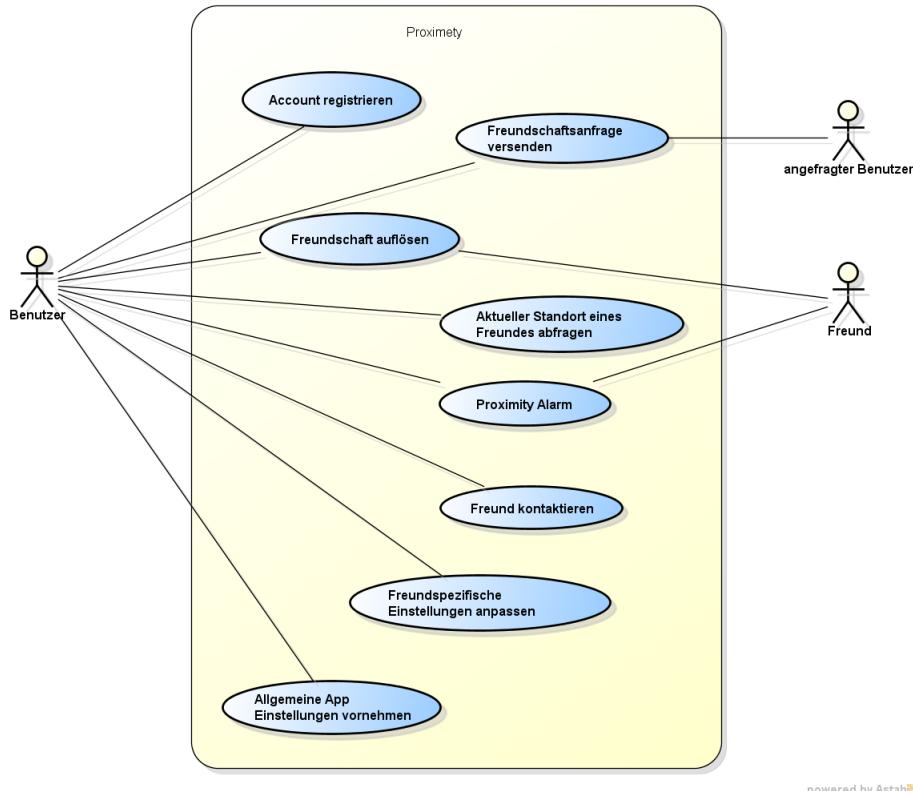


Abbildung 1: Basis UseCase Diagramm

2.1.2 Use Case Beschreibungen

2.1.2.1 UCo1 - Account registrieren

ID	UC01
Name	Account registrieren
Kurzbeschreibung	Der Benutzer registriert sich am System und erhält Logindaten.
Akteur	Benutzer
Auslöser	Der Akteur beginnt mit der Registrierung
Vorbedingungen	
Eingehende Informationen	E-Mailadresse, Benutzername, Passwort
Ergebnis	Registrierungsbestätigung
Nachbedingungen	Benutzerkonto wurde erstellt
Ablauf	<ol style="list-style-type: none"> 1. Akteur startet Registrierungsprozess 2. Akteur gibt E-Mailadresse und Passwort ein (inkl. Bestätigung) 3. Akteur gibt Benutzernamen ein 4. Akteur bestätigt die Registrierung. Es wird explizit darauf verzichtet eine Aktivierungs-E-Mail zu versenden.
Alternativer Ablauf	4a. Benutzername oder E-Mail werden bereits verwendet. Benutzer wird aufgefordert diese zu ändern.
Priorität	Hoch
Aufwand	?
Status	Offen
Änderungshistorie	?

2.1.2.2 UC02 - Freundschaftsanfrage versenden

ID	UC02
Name	Freundschaftsanfrage versenden
Kurzbeschreibung	Der Benutzer sendet eine Anfrage an einen anderen Benutzer um ihn seiner persönlichen Freundesliste hinzuzufügen.
Akteur	Benutzer, angefragter Benutzer
Auslöser	Der Akteur beginnt mit der Suche nach einem Freund
Vorbedingungen	Benutzer besitzt einen Account
Eingehende Informationen	E-Mailadresse oder Benutzername eines Benutzers
Ergebnis	Freundesanfrage an anderen Benutzer
Nachbedingungen	

Ablauf	<ol style="list-style-type: none"> 1. Akteur sucht einen Benutzer anhand Benutzername oder E-Mail 2. Akteur wählt Benutzer aus Ergebnisliste aus um eine Anfrage zu versenden 3. Angefragter Benutzer entscheidet über die Anfrage (Annahme/Ablehnen) 4. Akteur wird über das Ergebnis informiert.
Alternativer Ablauf	
Priorität	Hoch
Aufwand	?
Status	Offen
Änderungshistorie	?

2.1.2.3 UCo3 - Freundschaft auflösen

ID	UCo3
Name	Freundschaft auflösen
Kurzbeschreibung	Der Benutzer entfernt einen anderen Benutzer von seiner Freundesliste
Akteur	Benutzer, Freund
Auslöser	Der Akteur möchte einen Freund aus seiner Liste entfernen.
Vorbedingungen	<ul style="list-style-type: none"> • Benutzer besitzt einen Account • Benutzer besitzt mind. einen Freund
Eingehende Informationen	Der zu entfernende Freund
Ergebnis	Benachrichtigung an Benutzer und Freund
Nachbedingungen	Akteur und ausgewählter Benutzer sind keine Freunde mehr.
Ablauf	<ol style="list-style-type: none"> 1. Akteur sucht einen Benutzer in seiner Freundesliste 2. Akteur wählt Benutzer aus und wählt "Freund entfernen" 3. Akteur bestätigt seine Wahl (Annahme/Ablehnen) 4. Beide Benutzer werden über das Ereignis informiert
Alternativer Ablauf	
Priorität	Mittel
Aufwand	?

Status	Offen
Änderungshistorie	?

2.1.2.4 UCo4 - Aktueller Standort eines Freundes abfragen

ID	UCo4
Name	Aktueller Standort eines Freundes abfragen
Kurzbeschreibung	Der Benutzer fragt den aktuellen Standort eines Freun- des ab. Dieser wird auf einer Karte angezeigt
Akteur	Benutzer
Auslöser	Der Akteur möchte einen Freund lokalisieren.
Vorbedingungen	<ul style="list-style-type: none"> • Benutzer besitzt einen Account • Benutzer besitzt mind. einen Freund
Eingehende Informationen	Der zu lokalisierende Freund
Ergebnis	Standort des Freundes
Nachbedingungen	
Ablauf	<ol style="list-style-type: none"> 1. Akteur sucht einen Benutzer in seiner Freundesliste 2. Akteur wählt Benutzer aus und wählt "Freund lo- kalisieren" 3. Aktueller Standort wird von Freund abgerufen und auf einer Karte dargestellt.
Alternativer Ablauf	3a. Standort kann nicht abgerufen werden: Entspre- chende Fehlermeldung wird angezeigt.
Priorität	Hoch
Aufwand	?
Status	Offen
Änderungshistorie	?

2.1.2.5 UCo5 - Proximity Alarm

ID	UCo5
Name	Proximity Alarm
Kurzbeschreibung	Benachrichtigung falls sich zwei Benutzer in definierter (oder weniger) Distanz zueinander befinden.
Akteur	Benutzer, Freund

Auslöser	System ermittelt Distanz zwischen Benutzern unter definiertem Schwellwert.
Vorbedingungen	<ul style="list-style-type: none"> • Beide Benutzer sind Freunde • Aktueller Standort von beiden Benutzern bekannt • Proximity Alarm für Freund aktiviert
Eingehende Informationen	Standort der Benutzer
Ergebnis	Benachrichtigung der beiden Benutzer
Nachbedingungen	
Ablauf	<ol style="list-style-type: none"> 1. System prüft periodisch die Distanz zwischen Freunden 2. System versendet Benachrichtigung an die Benutzer
Alternativer Ablauf	
Priorität	Hoch
Aufwand	?
Status	Offen
Änderungshistorie	?

2.1.2.6 UCo6 - Freund kontaktieren

ID	UCo6
Name	Freund kontaktieren
Kurzbeschreibung	Benutzer kann einen Freund aus der App mit dritt Apps kontaktieren
Akteur	Benutzer
Auslöser	Benutzer möchte Freund kontaktieren
Vorbedingungen	Beide Benutzer sind Freunde
Eingehende Informationen	E-Mail des Freundes (z.Z. keine weiteren Kontaktinformationen vorhanden)
Ergebnis	Dritt-App wird mit Kontaktinformationen (E-Mail) gestartet
Nachbedingungen	
Ablauf	<ol style="list-style-type: none"> 1. Akteur sucht einen Benutzer in seiner Freundesliste 2. Benutzer wählt Kontakt App aus (z.B. Gmail) 3. Dritt-App wird gestartet

Alternativer Ablauf	
Priorität	Tief
Aufwand	?
Status	Offen
Änderungshistorie	?

2.1.2.7 UCo7 - Freund-Spezifische Einstellungen anpassen

ID	UCo7
Name	Freund-Spezifische Einstellungen anpassen
Kurzbeschreibung	Benutzer kann für jeden Freund Einstellungen anpassen (Tracking ein/aus, Distanz)
Akteur	Benutzer
Auslöser	Benutzer möchte Einstellungen ändern
Vorbedingungen	
Eingehende Informationen	Gewünschte Einstellungen
Ergebnis	Einstellungen sind angepasst
Nachbedingungen	
Ablauf	<ol style="list-style-type: none"> 1. Akteur sucht einen Benutzer in seiner Freundesliste 2. Benutzer ändert Einstellungen (Tracking ein/aus, Distanz) 3. Benutzer bestätigt Eingaben
Alternativer Ablauf	
Priorität	Mittel
Aufwand	?
Status	Offen
Änderungshistorie	?

2.1.2.8 UCo8 - Allgemeine App Einstellungen vornehmen

ID	UCo8
Name	Allgemeine App Einstellungen vornehmen
Kurzbeschreibung	Der Benutzer kann allgemeine Einstellungen an der App vornehmen

Akteur	Benutzer
Auslöser	Der Akteur möchte Einstellungen ändern
Vorbedingungen	Benutzer besitzt einen Account
Eingehende Informationen	
Ergebnis	Einstellungen aktualisiert und gespeichert
Nachbedingungen	
Ablauf	<ol style="list-style-type: none"> 1. Akteur wählt in der Naviagtion "Einstellungen" aus 2. Akteur kann Einstellungen anpassen (Standard Distanz für Alarm, Alarm Ton/Typ) 3. Akteur bestätigt seine Einstellungen
Alternativer Ablauf	
Priorität	Mittel
Aufwand	?
Status	Offen
Änderungshistorie	?

3 GUI Prototypes

3.1 Einleitung

Das GUI vom Proximety ist so aufgebaut, dass nur die für den aktuellen Context unerlässlichen Informationen dargestellt werden. So wird sichergestellt, dass der Benutzer nicht durch Informationen verwirrt wird, welche er für die aktuelle Aktion nicht gebrauchen kann.

Durch das Betätigen des Zurück-Knopfes kommt man im Applikationsfluss zurück auf die letzte Maske. Sollte sich der Benutzer auf dem Main Screen befinden, löst ein Drücken des Zurück-Knopfes das Schliessen der Applikation aus.

Eingaben des Benutzers werden immer auf die Gültigkeit überprüft und mit einer Hinweis-Meldung (Snackbar / Toast) behandelt. Ein erfolgreiches vorwärts gerichtetes Verlassen der Screens ist erst möglich, wenn alle Eingaben gültig sind. Zusätzlich zur Meldung wird noch das Eingabe-Feld markiert, in welchem der Fehler aufgetreten ist.

Im den Flow Charts wird der Fluss der Applikation durch die einzelnen Masken visualisiert.

3.2 Variante 1

Der allgemeine Ablauf der Anwendung kann in der Abbildung 2 entnommen werden. Anschliessend folgen alle bisher definierten Fenster der Variante 1.

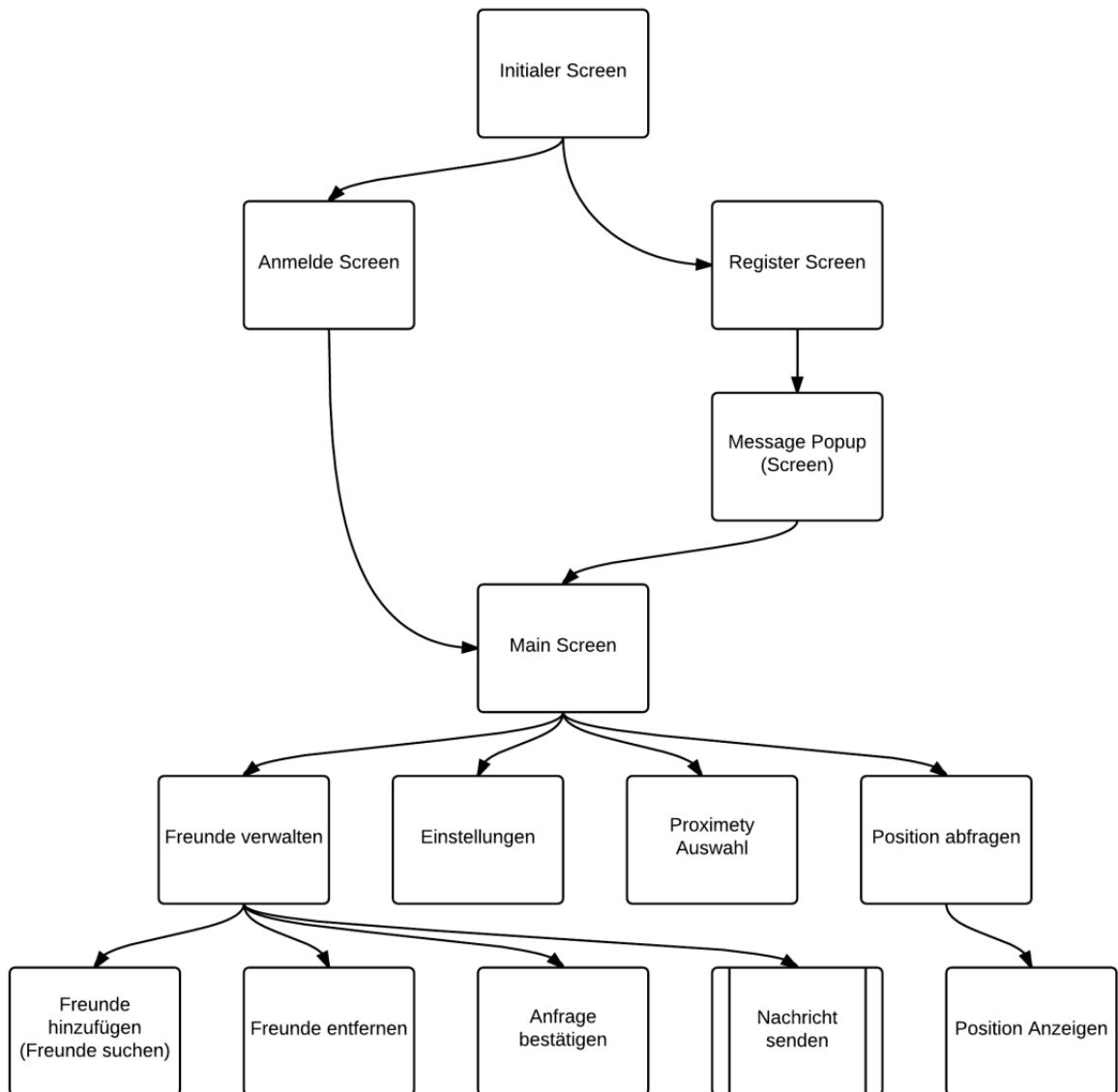


Abbildung 2: Design Flow Diagramm



(a) Initiales Fenster

(b) Registrierungsfenster

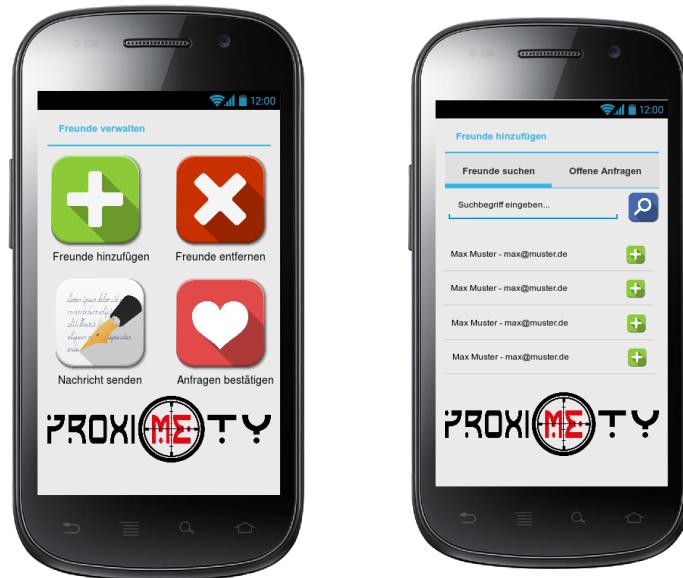
Abbildung 3: Initialer und Registrierungsfenster



(a) Benachrichtigungsfenster

(b) Hauptfenster

Abbildung 4: Benachrichtigungs- und Hauptfenster



(a) Freunde Verwaltungsfenster

(b) Freunde hinzufügen (Freunde suchen)

Abbildung 5: Freunde Verwaltungs- und Hinzufügefensster (Freunde suchen)



(a) Freunde hinzufügen (Offene Anfragen)

(b) Freunde entfernen

Abbildung 6: Freunde Entfernen- und Hinzufügefensster (Offene Anfragen)



(a) Proximety Auswahl

(b) Position abfragen Auswahl

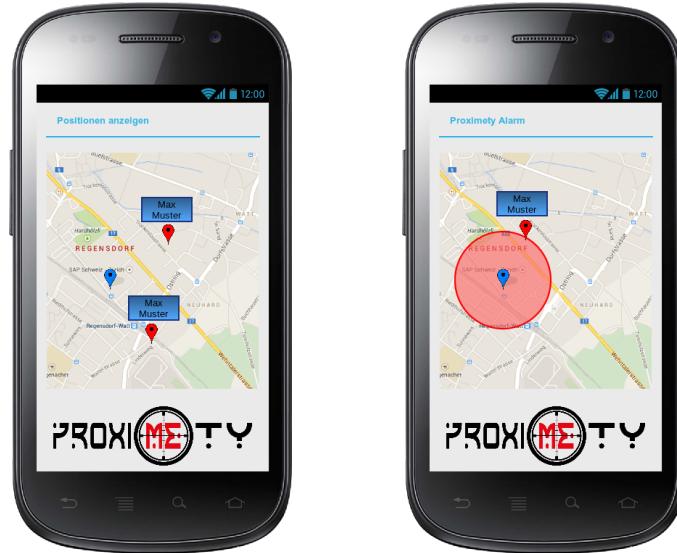
Abbildung 7: Proximety Auswahl- und Position Abfragefenster



(a) Anfrage bestätigen

(b) Anmeldefenster

Abbildung 8: Anfrage Bestätigungs- und Anmeldefenster



(a) Position anzeigen

(b) Proximity Alarm

Abbildung 9: Positions- und Proximity Alarm Anzeigefenster



(a) Details Freunde

(b) Einstellungen

Abbildung 10: Freunde Detail- und Einstellungsfenster

3.3 Variante 2

Der allgemeine Ablauf der Anwendung kann in der Abbildung 11 entnommen werden. Anschliessend folgen alle bisher definierten Fenster der Variante 2.

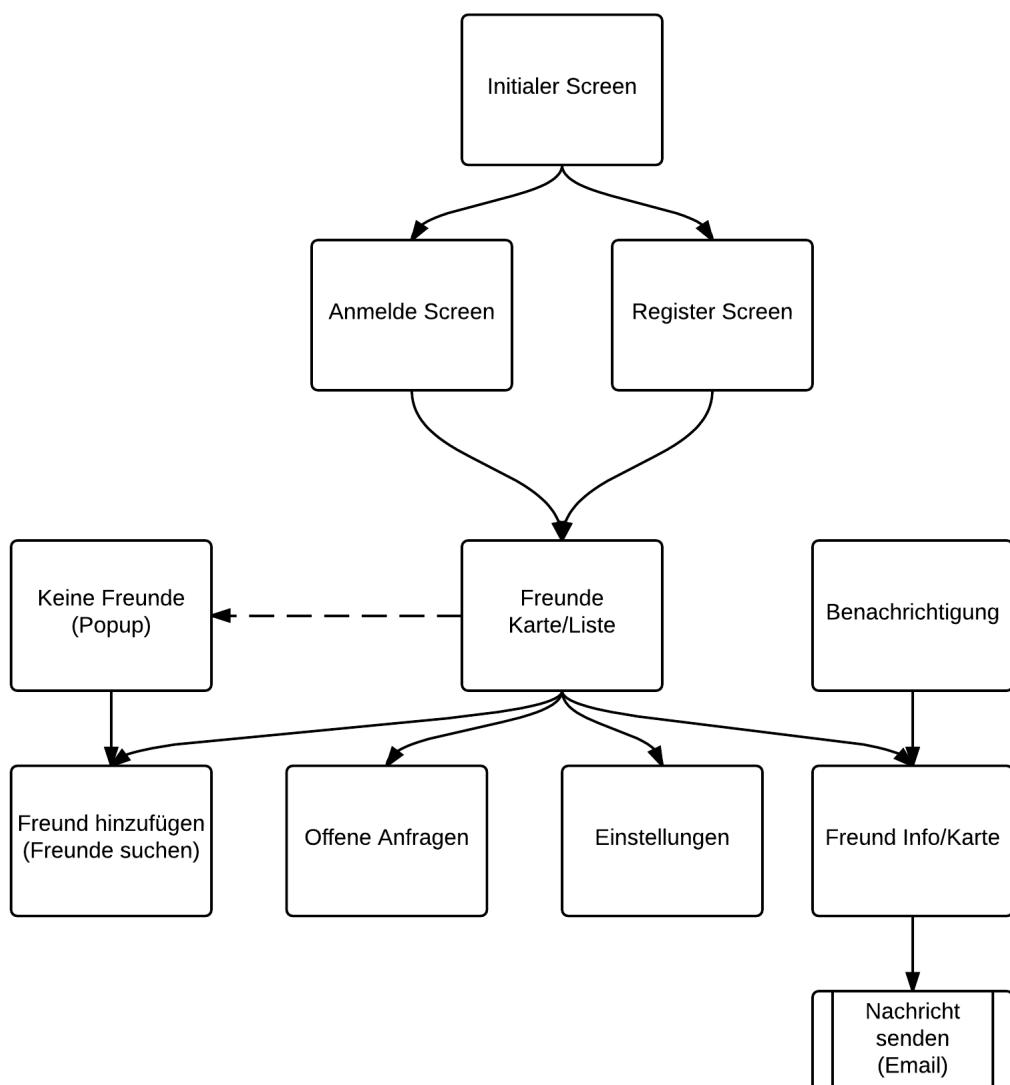
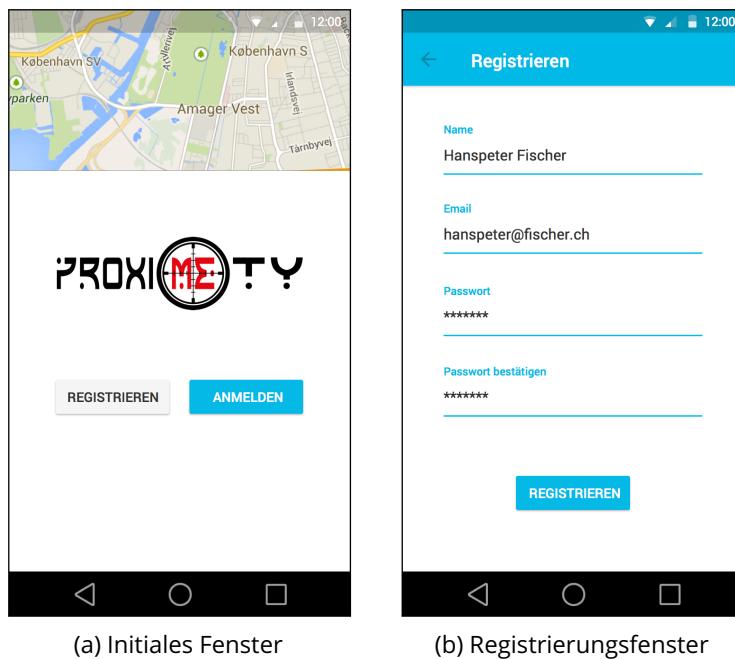


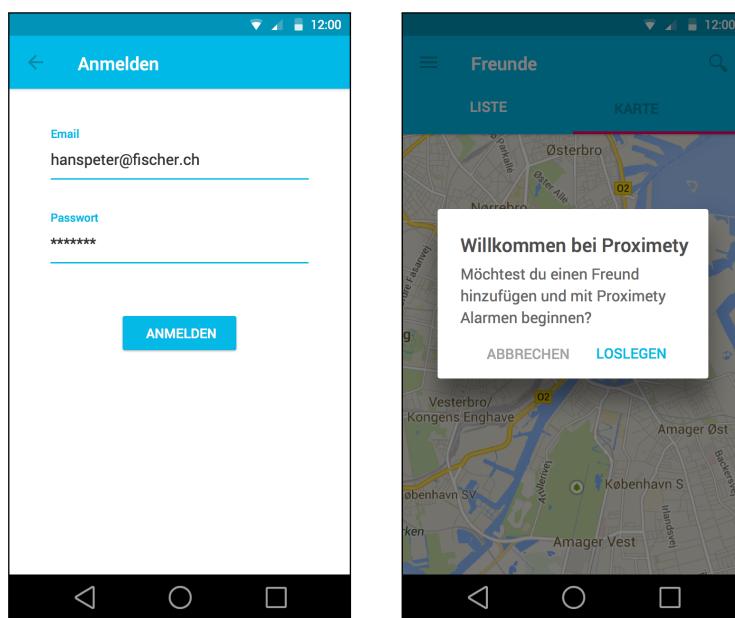
Abbildung 11: Design Flow Diagramm



(a) Initiales Fenster

(b) Registrierungsfenster

Abbildung 12: Initialer und Registrierungsfenster



(a) Anmeldung

(b) Keine Kontakte

Abbildung 13: Anmeldungs- und Hauptfenster (Ohne Kontakte)

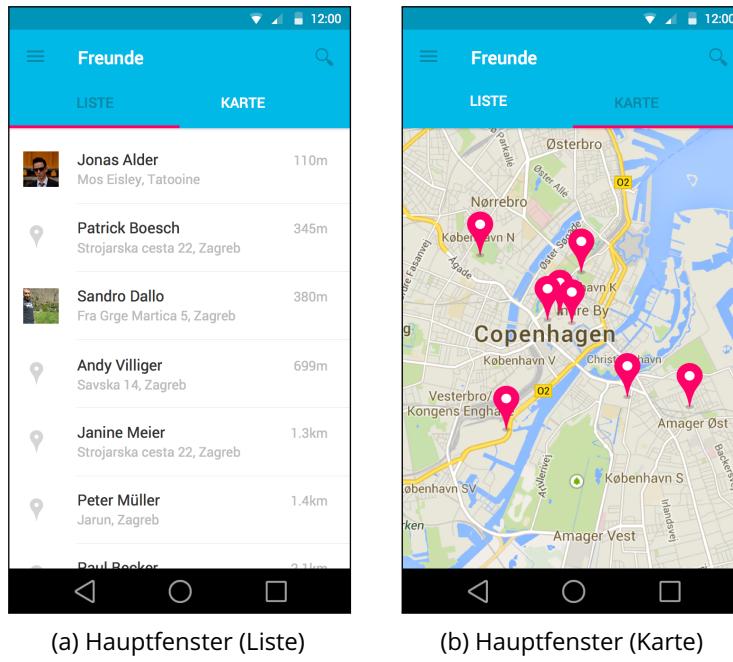


Abbildung 14: Hauptfenster in Listen- und Kartenansicht

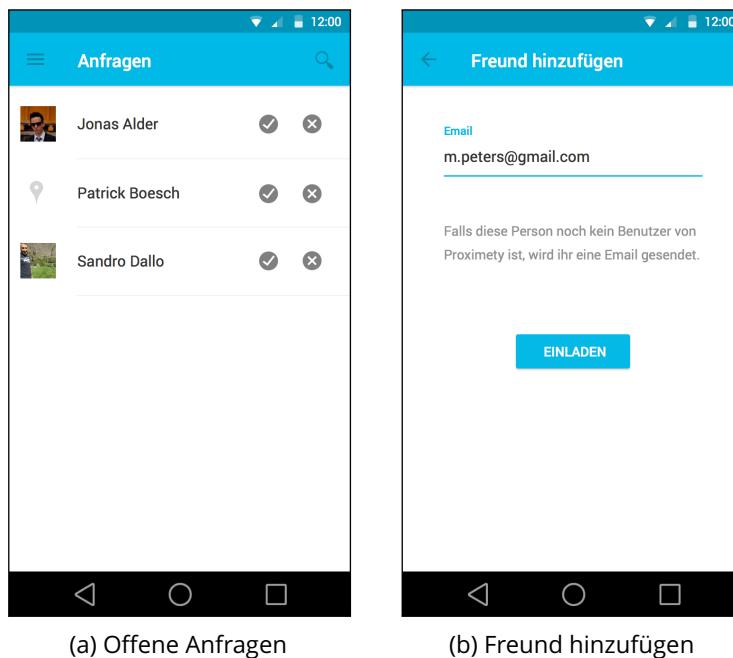


Abbildung 15: Freundschaftsanfragen stellen und beantworten

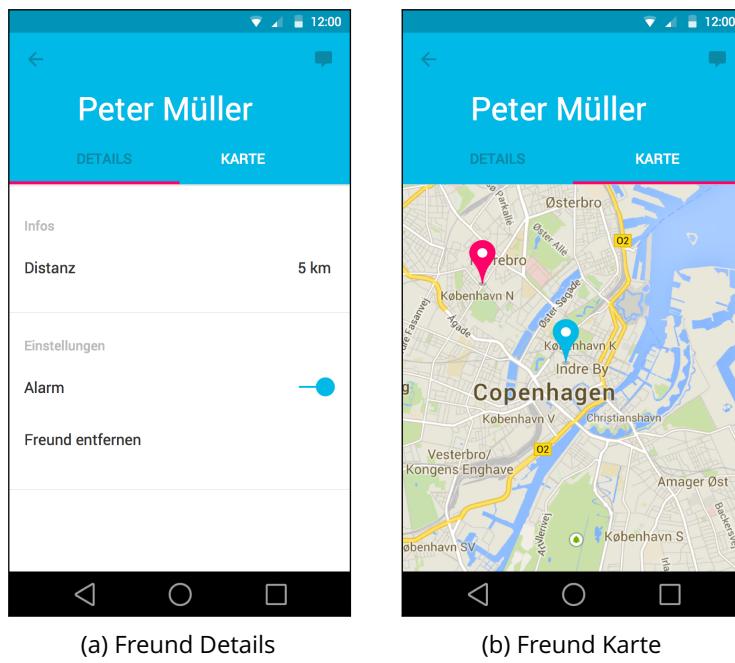


Abbildung 16: Detail- und Kartenansicht für Freund

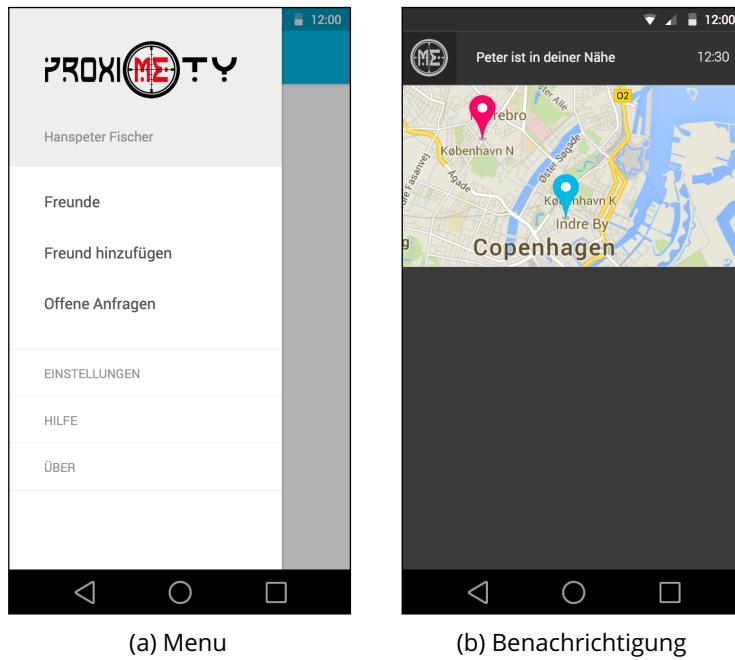


Abbildung 17: Menu und Benachrichtigung

3.4 Vergleich und Entscheid

Da es das Ziel des Designs war, dass die Screens mit möglichst wenig Informationen und Elementen - sprich schlicht - daher kommen, haben wir uns für die zweite Variante entschieden. Des Weiteren ist die Navigation im zweiten Vorschlag viel einfacher, da die Anzahl der Screens reduziert wurde. Für den Benutzer ist es so einfacher sich in der Applikation zu Recht zu finden. Das Menu wird im zweiten Vorschlag von jeder Maske mit einem Wisch zu erreichen sein – was den Android Konventionen entspricht und der Simplifizierung des Programms beiträgt.

Auch sticht sofort ins Auge, dass der Aufbau der einzelnen Masken dem "Material Design" von Android Lollipop (Android 5.0) und daher dem aktuellen Trend entspricht.

A Abbildungsverzeichnis

1	Basis UseCase Diagramm	4
2	Design Flow Diagramm	11
3	Initialer und Registrierungsfenster	12
4	Benachrichtigungs- und Hauptfenster	12
5	Freunde Verwaltungs- und Hinzufügefufenster (Freunde suchen)	13
6	Freunde Entfernen- und Hinzufügefufenster (Offene Anfragen)	13
7	Proximety Auswahl- und Position Abfragefenster	14
8	Anfrage Bestätigungs- und Anmeldefenster	14
9	Positions- und Proximety Alarm Anzeigefenster	15
10	Freunde Detail- und Einstellungsfenster	15
11	Design Flow Diagramm	16
12	Initialer und Registrierungsfenster	17
13	Anmeldungs- und Hauptfenster (Ohne Kontakte)	17
14	Hauptfenster in Listen- und Kartenansicht	18
15	Freundschaftsanfragen stellen und beantworten	18
16	Detail- und Kartenansicht für Freund	19
17	Menu und Benachrichtigung	19

B Tabellenverzeichnis

Quellenverzeichnis