

# 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 0.5

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

## Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>3</b>
<b>1 Allgemeines</b>	<b>4</b>
1.1 Einleitung . . . . .	4
<b>2 Anforderungen</b>	<b>4</b>
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
<b>3 GUI Prototypes</b>	<b>10</b>
3.1 Einleitung . . . . .	10
3.2 Variante 1 . . . . .	10
3.3 Variante 2 . . . . .	15
<b>A Abbildungsverzeichnis</b>	<b>20</b>
<b>B Tabellenverzeichnis</b>	<b>20</b>
<b>Quellenverzeichnis</b>	<b>21</b>

# 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

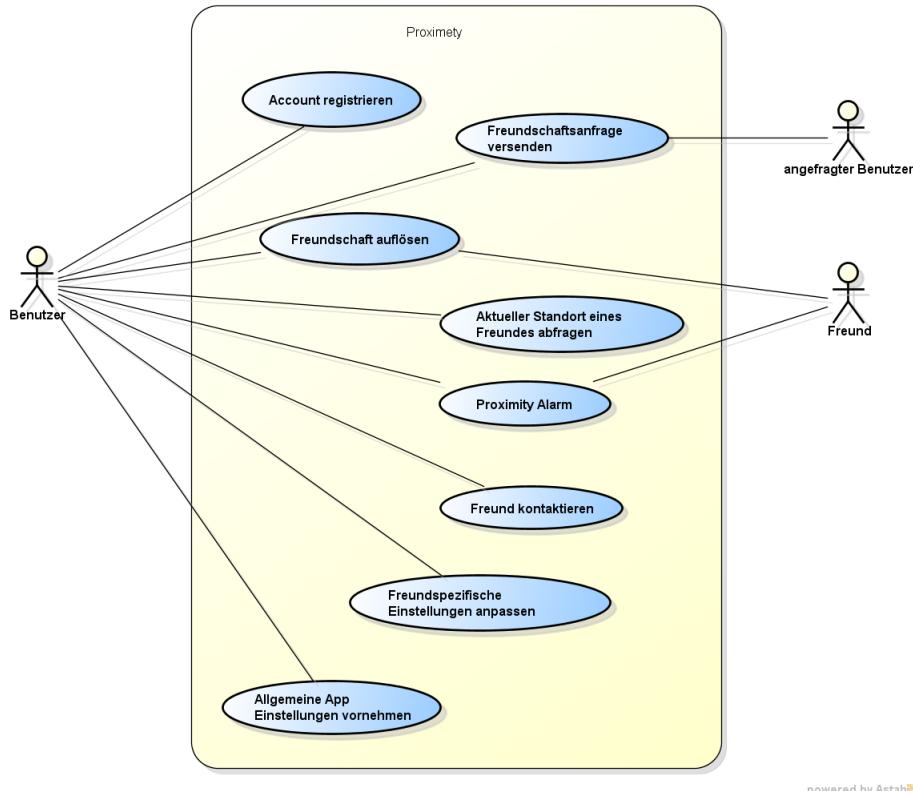


Abbildung 1: Basis UseCase Diagramm

### 2.1.2 Use Case Beschreibungen

#### 2.1.2.1 UCo1 - Account registrieren

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

### 2.1.2.2 UC02 - Freundschaftsanfrage versenden

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

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

### 2.1.2.3 UCo3 - Freundschaft auflösen

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

<b>Status</b>	Offen
<b>Änderungshistorie</b>	?

#### 2.1.2.4 UCo4 - Aktueller Standort eines Freundes abfragen

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

#### 2.1.2.5 UCo5 - Proximity Alarm

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

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

#### 2.1.2.6 UCo6 - Freund kontaktieren

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

<b>Alternativer Ablauf</b>	
<b>Priorität</b>	Tief
<b>Aufwand</b>	?
<b>Status</b>	Offen
<b>Änderungshistorie</b>	?

#### 2.1.2.7 UCo7 - Freund-Spezifische Einstellungen anpassen

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

#### 2.1.2.8 UCo8 - Allgemeine App Einstellungen vornehmen

<b>ID</b>	UCo8
<b>Name</b>	Allgemeine App Einstellungen vornehmen
<b>Kurzbeschreibung</b>	Der Benutzer kann allgemeine Einstellungen an der App vornehmen

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

## 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 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 nachfolgenden Flow Chart 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.

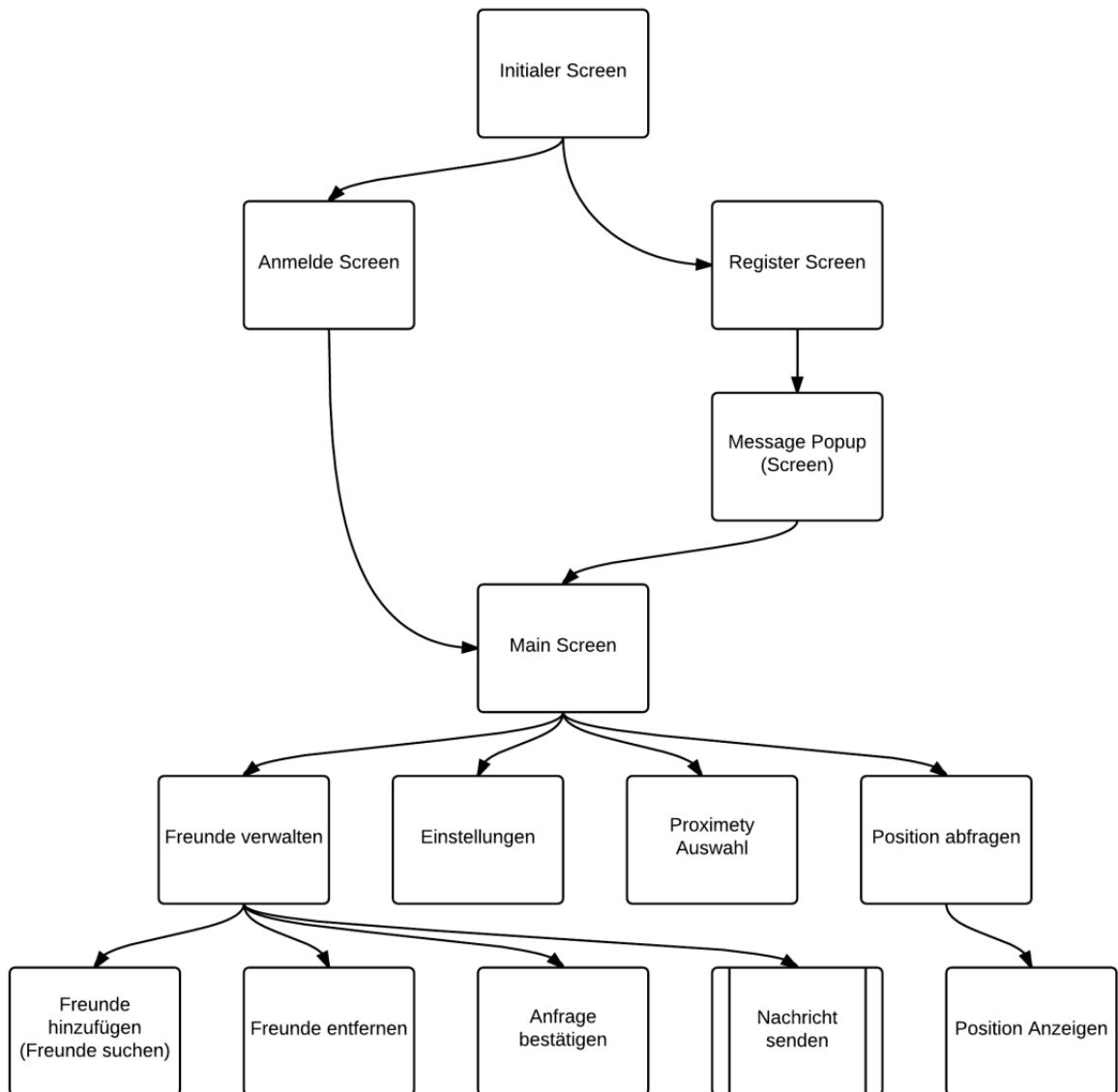


Abbildung 2: Design Flow Diagramm



(a) Initiales Fenster

(b) Registrierungsfenster

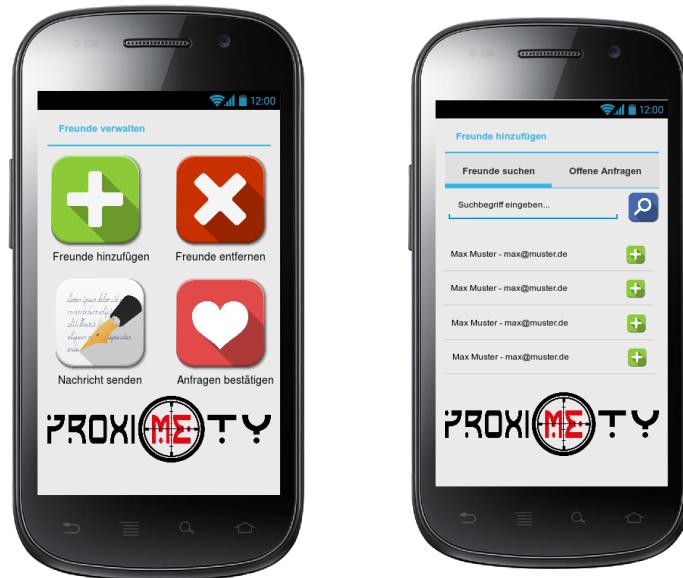
Abbildung 3: Initialer und Registrierungsfenster



(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.

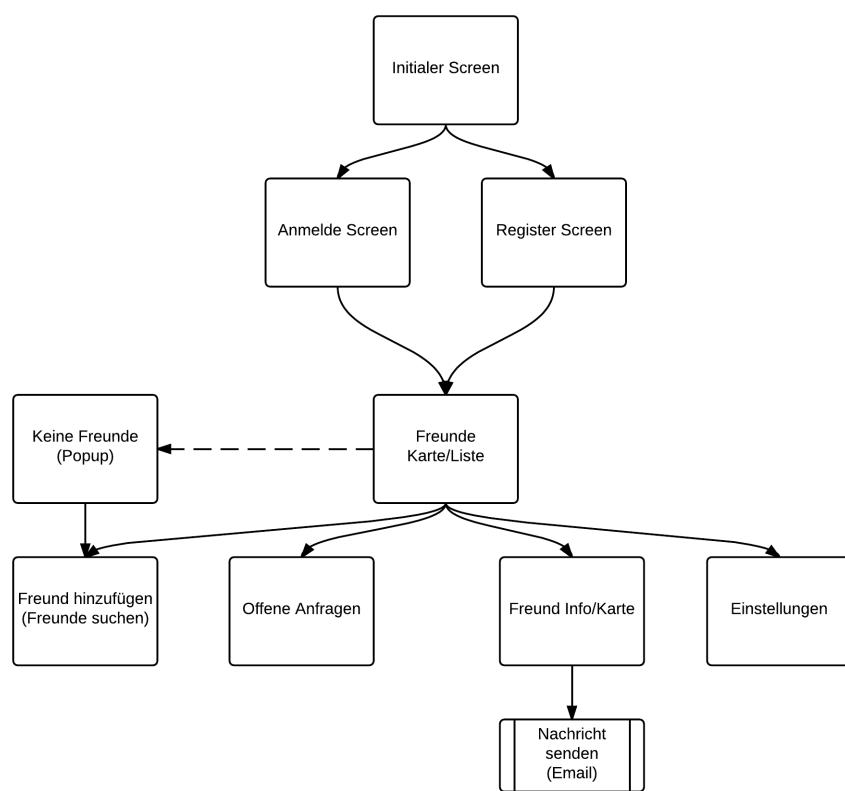
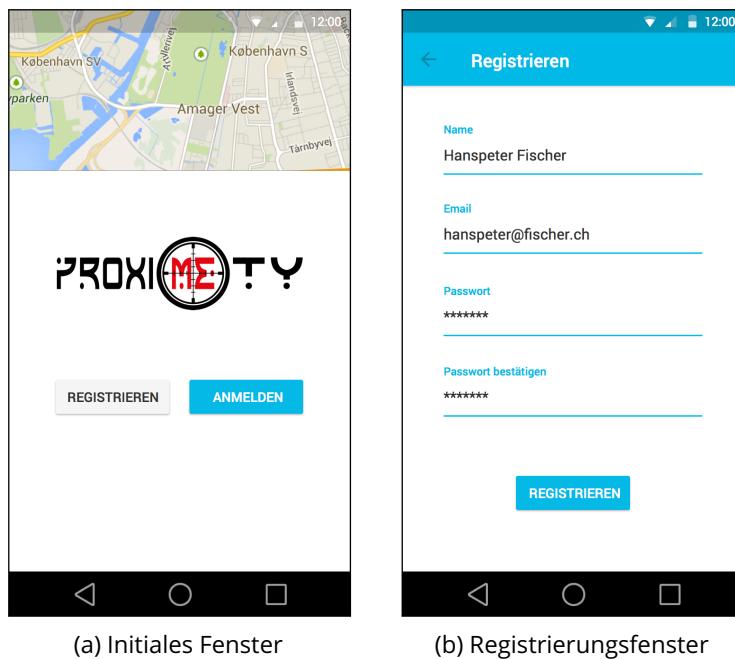


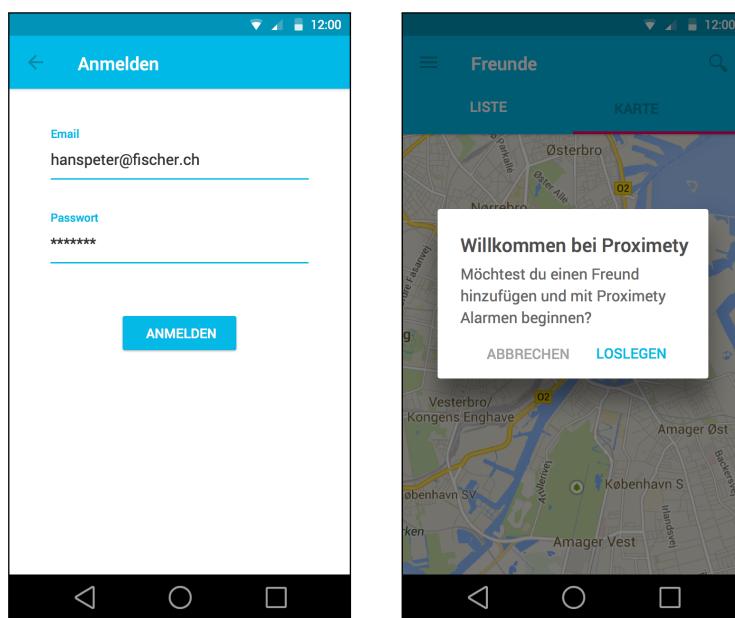
Abbildung 11: Design Flow Diagramm



(a) Initiales Fenster

(b) Registrierungsfenster

Abbildung 12: Initialer und Registrierungsfenster



(a) Anmeldung

(b) Keine Kontakte

Abbildung 13: Anmeldungs- und Hauptfenster (Ohne Kontakte)

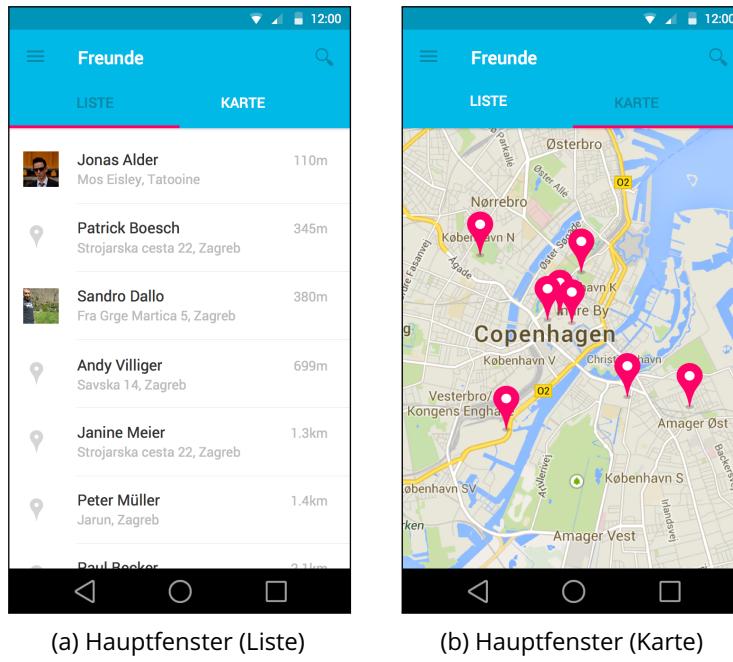


Abbildung 14: Hauptfenster in Listen- und Kartenansicht

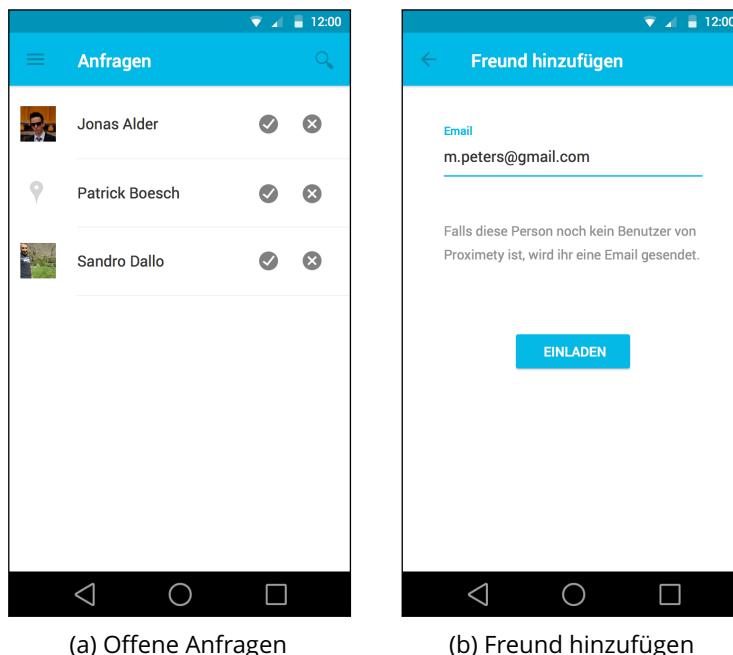


Abbildung 15: Freundschaftsanfragen stellen und beantworten

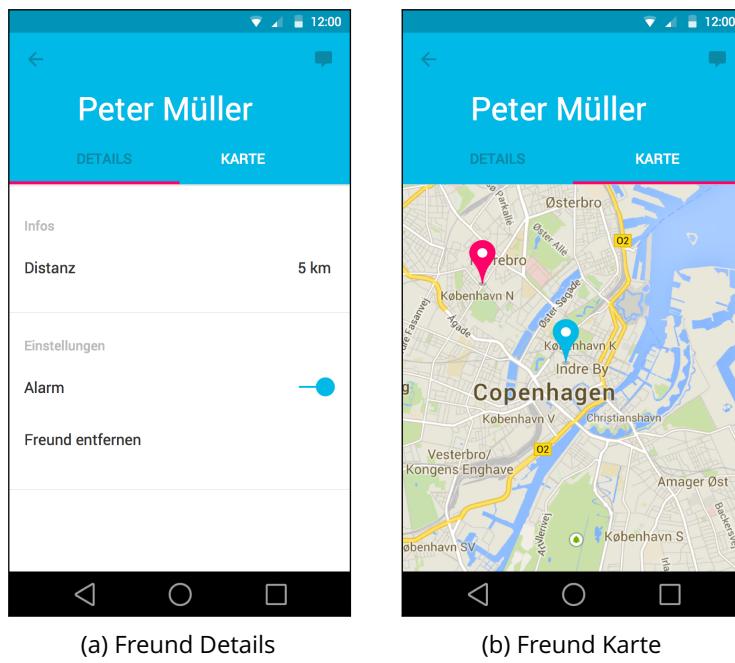


Abbildung 16: Detail- und Kartenansicht für Freund

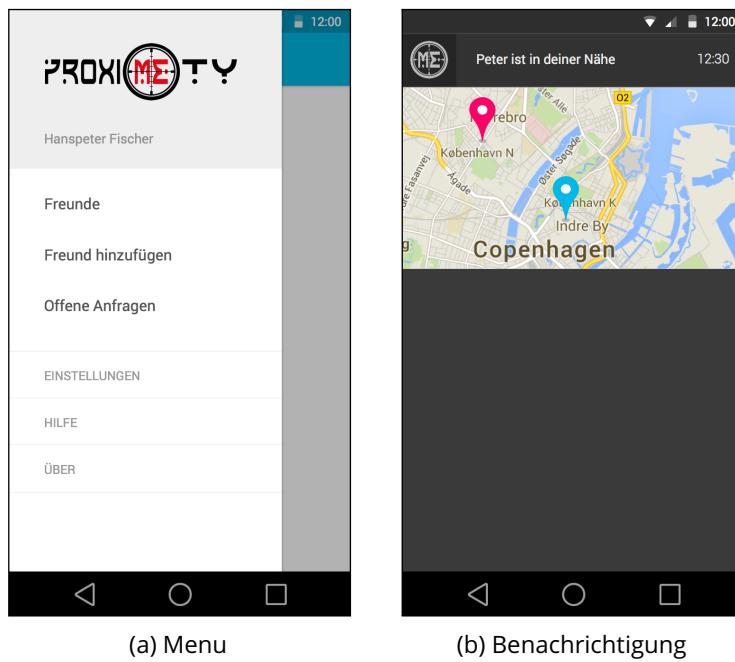


Abbildung 17: Menu und Benachrichtigung

## 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