

ESA - Proximety

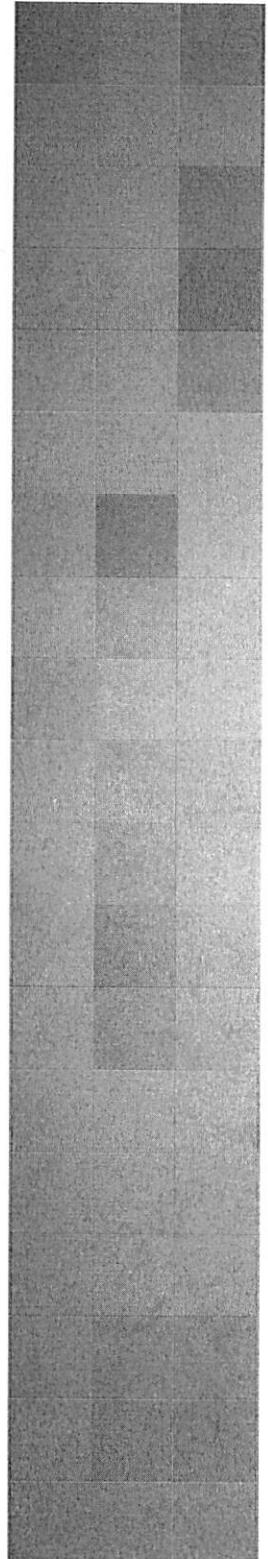
Projektdokumentation



Fernfachhochschule Schweiz
Zürich | Basel | Bern | Brig

Mitglied der SUPSI

Jonas Alder, Patrick Bösch
Sandro Dallo, Andy Villiger
FFHS Bsc Inf 2011
Version 0.2
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

Use-Case-Diagramme gut und übersichtlich
Zusammenhang zwischen Use-Case-Diagramm
und Use-Case-Beschreibung weitgehend konsi-
stent ✓

Use-Case-Beschreibungen sind klar und gut
nachvollziehbar

Kapitel 5.5

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 UCo1 - Account registrieren	4
2.1.2.2 UCo2 - Freundschaftsanfrage versenden	5
2.1.2.3 UCo3 - Freundschaft auflösen	6
2.1.2.4 UCo4 - Aktueller Standort eines Freundes abfragen	7
2.1.2.5 UCo5 - Proximity Alarm	7
2.1.2.6 UCo6 - Freund kontaktieren	8
2.1.2.7 UCo7 - Freund-Spezifische Einstellungen anpassen	9
2.1.2.8 UCo8 - Allgemeine App Einstellungen vornehmen	9
2.2 Nicht Funktional	10
A Abbildungsverzeichnis	11
B Tabellenverzeichnis	11
Quellenverzeichnis	12

1 Allgemeines

1.1 Einleitung

Dieses Dokument beinhaltet die Beschreibung und Analyse des Projektes Proximity, 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.

Proximity 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

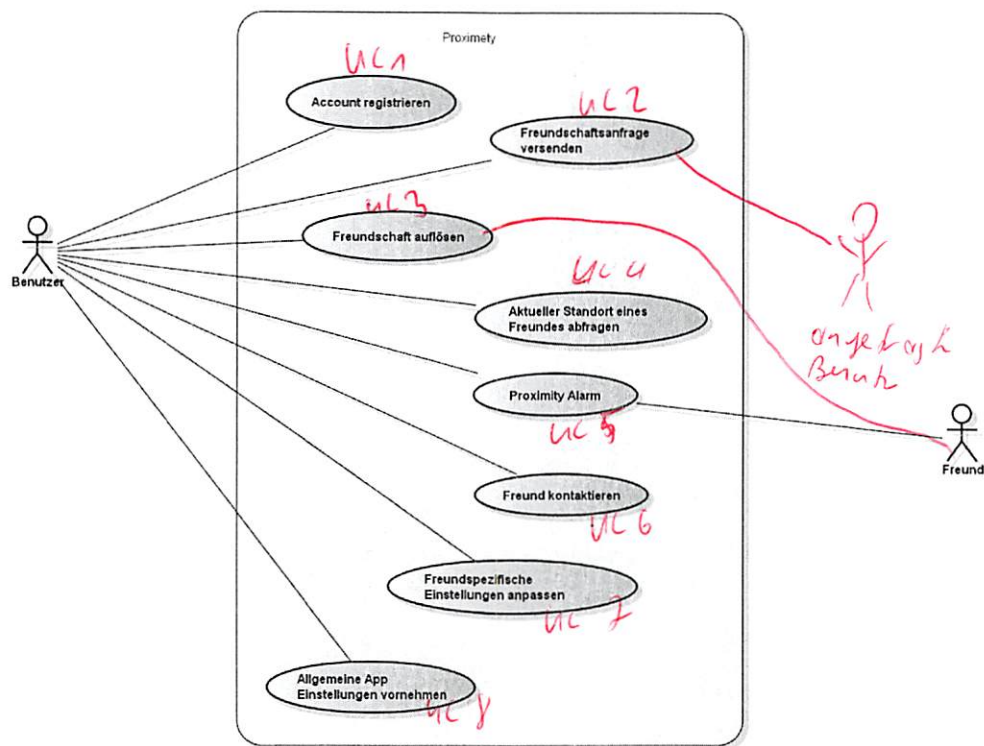


Abbildung 1: Basis UseCase Diagramm

2.1.2 Use Case Beschreibungen

2.1.2.1 UC01 - 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	App auf dem Gerät des Benutzers installiert kann vor das gesch werden, wie beschreibt je die App
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. <p>Es wird explizit darauf verzichtet eine aktivierungs E-Mail zu versenden.</p>
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 hinzu zufügen.
Akteur	Benutzer , angefragte 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	?

Wie liegt das?

• Time out
• Asynchron

2.1.2.3 UCo3 - Freundschaft auflösen

ID	UCo3
Name	Freundschaft auflösen
Kurzbeschreibung	Der Benutzer entfernt einen anderen Benutzer von seiner Freundesliste
Akteur	Benutzer, <i>Freund</i>
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 beide Benutzer <i>und Freund</i>
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 Freundes 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"> Akteur sucht einen Benutzer in seiner Freundesliste Akteur wählt Benutzer aus und wählt "Freund lokalisieren" Aktueller Standort wird von Freund abgerufen und auf einer Karte dargestellt.
Alternativer Ablauf	3a. Standort kann nicht abgerufen werden: Entsprechende 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	System 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	Beide Benutzer sind Freunde
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 Navigation "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	?

2.2 Nicht Funktional

A Abbildungsverzeichnis

1	Basis UseCase Diagramm	4
---	----------------------------------	---

B Tabellenverzeichnis

Quellenverzeichnis