

UE Software Engineering 050052
WS 2016/17

Gruppe: 6
LV-Leiter: Vasko

Projektname: OldFace

Projektteam:

Nachname	Vorname	Matrikelnummer	E-Mail-Adresse
Balaz	Melanie	1507236	a1507236@unet. univie.ac.at
Berg	Stefan	1425065	a1425065@unet. univie.ac.at
Haag	Valentin	1425622	a1425622@unet. univie.ac.at
Kramml	Hannes	1503268	a1503268@unet. univie.ac.at

CEWebS-Teamseite: <https://cewebs.cs.univie.ac.at/SWE/...>

Oldface-Website: <http://oldface.omaha17.at/>

Datum: November 2016

1. Funktionale Anforderungen

Wir suchten nach einer Art von sozialem Netzwerk dass sich inhaltlich von bereits Existierenden abhob.

Mit Hilfe von Brainstorming, Befragung von Mitstudenten und vergleichen von schon existierenden Sozialen Netzwerken kamen wir auf die Idee von OldFace, ein Social Network für Senioren.

Beschreiben Sie kurz, wie die funktionalen Anforderungen erhoben wurden.

- *Brainstorming*
- *Befragung von Endbenutzen (Interviews)*
- *Analogien (Erfahrungen aus gleichartigen Systemen)*
- *usw.*

1.1. Funktionalität

OldFace ist speziell an die Zielgruppe Senioren gerichtet, insofern sollen die Funktionalitäten in ihrem Aufbau und das User Interface daran angepasst werden. Grundsätzliche Funktionalitäten werden jedoch auch zur Verfügung gestellt:

- Es wird ein Login Formular zur Verfügung gestellt, bei dem sich neue User mit Passwort und Email Adresse, sowie einem selbst gewählten Benutzernamen, registrieren können. Bereits registrierte User können sich über diesen Bereich einloggen.
- Es wird die Möglichkeit geben einen User Account, Forscher Account oder Admin-Account anzulegen.
- Es gibt natürlich auch einen Logout Button zum ausloggen.
- Jeder User hat eine persönliche Pinnwand, auf dem er seine Beiträge verfassen und posten kann. Diese Pinnwand ist die Startseite nach dem Login.
- User können nach anderen Usern mittels Namen suchen und ihnen folgen.
- Neue Beiträge von Usern denen man folgt, werden auf einem Dashboard dargestellt. Die Beiträge sind zeitlich sortiert, der neueste Beitrag steht ganz oben.
- Jeder Beitrag kann von einem selber, oder von anderen Usern kommentiert werden.
- Es gibt eine Like Funktion für jeden Beitrag und jedes Kommentar
- Die Funktion der Admins besteht darin die Website zu moderieren. Er kann Beiträge löschen und User sperren.

- Der User kann sowohl seine Pinnwand, als auch die von Anderen aufrufen und ihre Informationen anzeigen bzw. auf deren Pinnwand Beiträge verfassen.
- Der User kann bei seinem eigener Pinnwand Informationen hinzufügen und bearbeiten.
- Auf der Pinnwand eines Users, gibt es einen Button Freunde, über den man sich dessen Freundesliste anschauen kann.
- Dem Forscher steht nur die Funktion sich Statistiken über die Website anzusehen zur Verfügung.

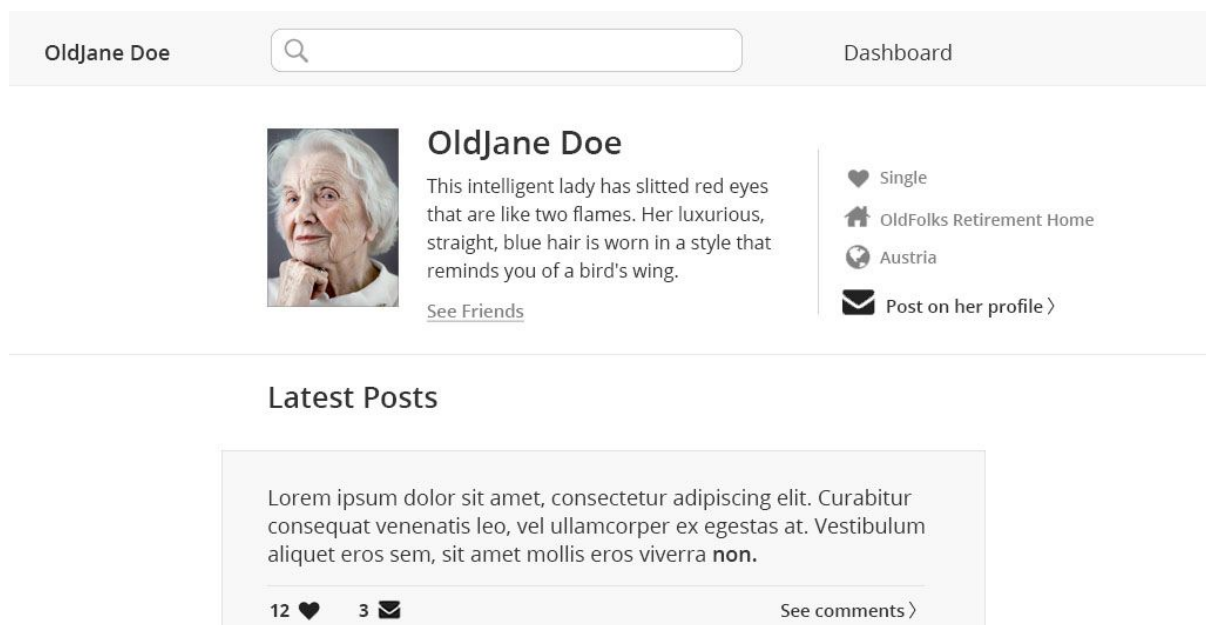
Beschreiben Sie die grundlegende Funktionalität des zu erstellenden Softwaresystems auf 1-2 Seiten.

Beantworten Sie die Frage, was Ihre Anwendung leisten soll (externe/Benutzersicht), jedoch nicht, wie diese Funktionalität erreicht werden soll (interne/Entwicklersicht).

Siehe auch Beispiele in M. Hitz, G. Kappel, UML@Work, dpunkt Verlag 2003, (Abschnitt 1.5, S.13 ab "Das CALENDARIUM ist ... " bis S.15 "Diverse Darstellungsaspekte ...").

1.2. Bedienoberfläche

Skizzieren Sie die Bedienoberfläche (User Interface), insbesondere die Navigationsmöglichkeiten.



Design Vorschlag eines User-Profils für OldFace

2. Use-Case-Modell

Erstellen sie für das zu entwickelnde System ein Use-Case Modell bestehend aus einem Use-Case-Diagramm und den Beschreibungen der einzelnen Use-Cases. Siehe dazu „Konstruktive Schritte zur Erstellung von Use Case Diagrammen“ in den Vorlesungsunterlagen.

Unterteilen Sie Ihre Use Cases in primäre und sekundäre Use Cases:

- Primärer Use Cases beschreiben die zentrale Funktionalität eines Systems (z.B. Buchen, etc.) und sollen in der ersten Phase des Projekts realisiert werden.

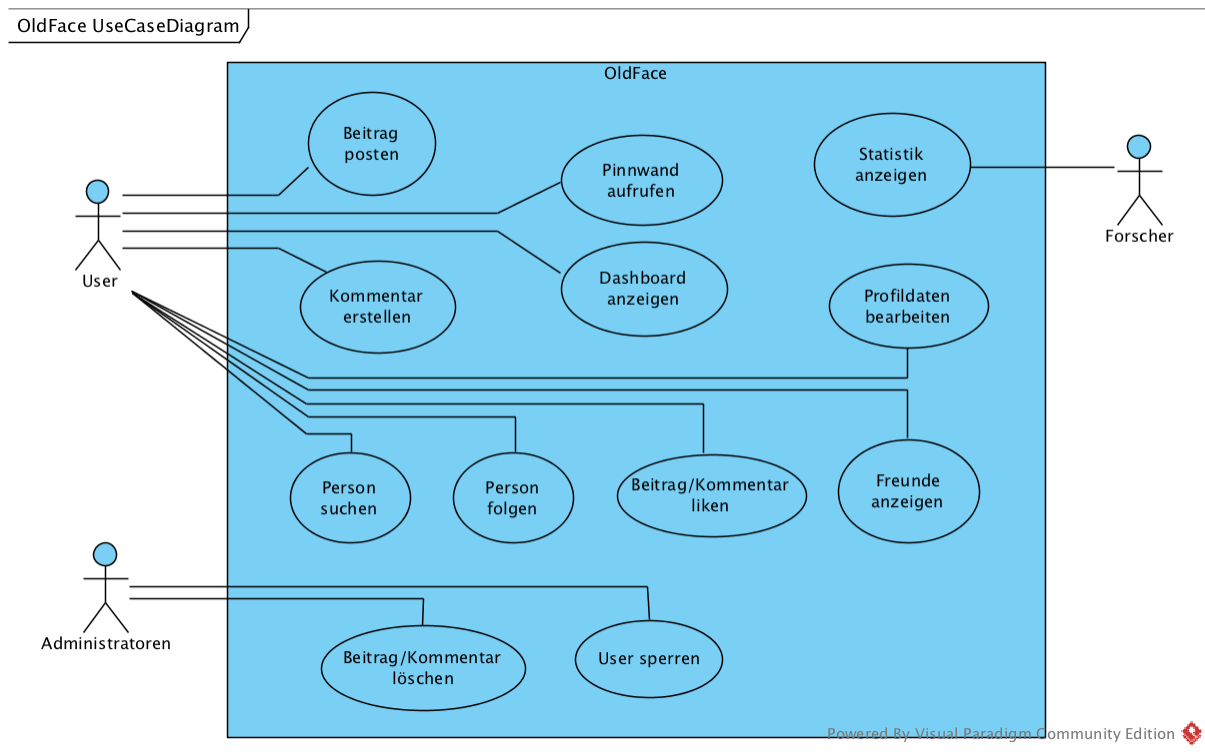
- Sekundäres Use Case beschreiben die zusätzliche Funktionalität des Systems (Suchfunktionen, etc..),

die in der zweiten Phase des Projekts realisiert werden soll.

- Beachten Sie: Login/Registrierung sind KEINE Use Cases!

2.1. Use-Case-Diagramm

Was soll das Software-System aus Sicht der Benutzer leisten? Stellen Sie das zu entwickelnde System, die Akteure, die Verwendung des Systems (Use-Cases) und Beziehungen zwischen Use-Cases und Akteuren dar. (siehe Vorlesungsunterlagen).



2.2. Use-Case Beschreibungen

Beschreiben Sie die einzelnen Use Cases detailliert und erstellen Sie für jeden Use Case eine entsprechende User Interface Skizze (schematisch, keine screen shots).

Benutzen Sie folgendes Schema, siehe Vorlesungsfolien UseCaseModeling.pdf, Folien 14 bis 16, und M. Hitz, G. Kappel, UML@Work, dpunkt Verlag 2003: Abschnitt 4.1, insbes. Tabelle 4-2, ohne Aktivitätsdiagramm (S.192 bis S.204):

Zumindest müssen Use Case Beschreibungen folgende Aspekte umfassen:

Überschrift: **Use-Case n: Bezeichnung**

1. Ziel
2. Kurzbeschreibung
3. Vorbedingung
4. Nachbedingung, bei Erfolg
5. Fehlersituationen
6. Nachzustand im Fehlerfall
7. Akteure
8. Trigger: Auslösendes Ereignis (außer Benutzereingaben)
9. Basisablauf (Standardablauf) als Folge von Aktionen: 1., 2., 3., ...
10. Alternativabläufe #1, #2, ..., als Folgen von Aktionen

Geben Sie für etwaige Beziehung zwischen Use-Cases (include, extend, Generalisierung) eine Begründung an.

Use-Case 1: Pinnwand aufrufen

<u>Ziel:</u> Dem User wird die gewünschte Pinnwand vollständig angezeigt Phase: Primär	
<u>Kurzbeschreibung:</u> Die Pinnwand dient als persönliches Profil, in dem alle selbst erstellten Beiträge gelistet werden und fungiert als Startseite nach dem Login. In einer späteren Projektphase sollen hier auch Nutzerinformationen (Kurzbiographie, Profilbild, etc.) angezeigt bzw. bearbeitet werden können. Die eigene Pinnwand kann durch einen Menüpunkt in der Navigationsleiste von jeder beliebigen Seite aus aufgerufen werden. Weiters kann jede Pinnwand der anderen User im Sozialen Netzwerk durch unterschiedliche Aktionen wie z.B. im Zuge der Personensuche aufgerufen werden.	
<u>Vorbedingung:</u> User ist eingeloggt	<u>Nachbedingung:</u> Pinnwand wird vollständig dargestellt
<u>Fehlersituationen:</u> User versucht auf ein Profil von einem anderen User zuzugreifen, dass inzwischen gelöscht wurde	<u>Nachbedingung im Fehlerfall:</u> Fehlerseite darstellen
<u>Akteur(e):</u> User	<u>Trigger:</u> direkt nach einem erfolgreichen Login oder durch Benutzer ausgelöst

Basisablauf:

1. a) User wird erfolgreich authentifiziert oder
b) User klickt auf den Menüpunkt für seine persönliche Pinnwand
2. Die persönliche Pinnwand mit Userdaten und selbst erstellten Beiträgen wird angezeigt

Alternativablauf:

1. a) User klickt auf den Autor eines beliebigen Beitrages im Dashboard oder
b) klickt auf eine Person im Zuge der Suchfunktion oder
c) ...evtl. später... auf eine Person in der Freundesliste
2. Die Pinnwand der gewünschten Person wird dargestellt

Use-Case 2: Dashboard anzeigen

Ziel: Das userbezogene Dashboard wird vollständig dargestellt

Phase: Primär

Kurzbeschreibung:

Im Dashboard werden die neuesten Beiträge von Nutzern, denen man folgt, angezeigt. Wird später womöglich noch mit zusätzlichen Funktionen ausgestattet.

Vorbedingung: User ist eingeloggt

Nachbedingung: Dashboard wird vollständig angezeigt

Fehlersituationen:

Nachzustand im Fehlerfall:

Akteur(e): User

Trigger: durch Benutzer ausgelöst

Basisablauf:

1. User klickt auf Dashboard Menüpunkt
2. Persönlich zugeschnittenes Dashboard mit neuen Beiträgen befreundeter Personen wird dargestellt.

Alternativabläufe:

keine

Use-Case 3: Beitrag posten

Ziel: Userbeitrag wurde erfolgreich auf der Pinnwand gepostet

Phase: Primär

Kurzbeschreibung:

Einer der wichtigsten Funktionalitäten des Sozialen Netzwerkes ist das Verfassen von Beiträgen. Diese werden entweder in der persönlichen Pinnwand oder in der Pinnwand eines anderen - in der Regel befreundeten - Users erstellt und veröffentlicht.

<u>Vorbedingung:</u> User ist eingeloggt; User hat eine beliebige Pinnwand geöffnet	<u>Nachbedingung:</u> Veröffentlichter Beitrag wird auf der richtigen Pinnwand zusätzlich zu etwaigen anderen Beiträgen angezeigt
<u>Fehlersituationen:</u>	<u>Nachzustand im Fehlerfall:</u>
<u>Akteur(e):</u> User	<u>Trigger:</u> durch Benutzer ausgelöst
<u>Basisablauf:</u> <ol style="list-style-type: none"> 1. User navigiert zur Beitragseingabe in seiner persönlichen Pinnwand 2. User verfasst die gewünschte Nachricht im dargestellten Textfeld 3. User klickt auf einen Button, um den Beitrag in der persönlichen Pinnwand zu veröffentlichen 	
<u>Alternativabläufe:</u> <ol style="list-style-type: none"> 1. User navigiert zur Beitragseingabe in der Pinnwand eines anderen Users 2. User verfasst die gewünschte Nachricht im dargestellten Textfeld 3. User klickt auf einen Button, um den Beitrag in der fremden Pinnwand zu veröffentlichen 	

Use-Case 4:Kommentar erstellen

<u>Ziel:</u> Userkommentar wurde erfolgreich zu einem Beitrag oder Bild gepostet Phase: Primär	
<u>Kurzbeschreibung:</u> Wichtig in einem sozialen Netzwerk ist die Möglichkeit eine eigene Meinung preisgeben zu können. Diese wird in Form von Kommentaren auf Beiträge anderer Personen realisiert.	
<u>Vorbedingung:</u> User ist eingeloggt; User ist insofern aktiv, dass er andere zu kommentierende Beiträge sieht;	<u>Nachbedingung:</u> Veröffentlichter Kommentar wird zu dem richtigen Beitrag angezeigt
<u>Fehlersituationen:</u>	<u>Nachzustand im Fehlerfall:</u>
<u>Akteur(e):</u> User	<u>Trigger:</u> durch Benutzer ausgelöst
<u>Basisablauf:</u> <ol style="list-style-type: none"> 1. User navigiert zu Beitrag einer anderen Person über dessen Profil 2. User verfasst den gewünschten Kommentar im dargestellten Textfeld 3. User klickt auf einen Button, um den Kommentar zu dem Beitrag zu veröffentlichen 	
<u>Alternativabläufe:</u>	

1. User navigiert per Dashboard zu gewünschten Beitrag
2. User verfasst die gewünschten Kommentar im dargestellten Textfeld
3. User klickt auf einen Button, um den Kommentar zu veröffentlichen

Use-Case 5: Person suchen

Ziel: Person wurde erfolgreich gefunden

Phase: Sekundär

Kurzbeschreibung:

Wichtig in einem sozialen Netzwerk ist es andere Person zu suchen und zu finden um mit ihnen Kontakt aufnehmen zu können.

Vorbedingung: User ist eingeloggt;

Nachbedingung: gesuchte Person wird angezeigt;

Fehlersituationen:

Nachzustand im Fehlerfall:

Akteur(e): User

Trigger: durch Benutzer ausgelöst

Basisablauf:

1. User navigiert zu Suchfunktion
2. User schreibt den gewünschten Namen im dargestellten Textfeld
3. User klickt auf einen Button, um die Person zu finden und um auf dessen Profil zu kommen

Alternativabläufe:

1. User navigiert über befreundete Kontakte und dessen Freunden zu noch nicht bekannten Personen
2. User klickt auf eine Person um auf dessen Profil zu gelangen

Use-Case 6: Person folgen

Ziel: Person wird gefolgt und damit kann man all dessen Beiträge sehen und kommentieren

Phase: Sekundär

Kurzbeschreibung:

Da man in einem sozialen Netzwerk mehr als eine Person braucht damit es funktioniert, gibt es die Möglichkeit anderen Menschen zu folgen. Je mehr Personen man folgt desto mehr Beiträge sieht man und kann man auch kommentieren und liken usw.

Vorbedingung: User ist eingeloggt;

Nachbedingung: Person A folgt Person B;
Person A kann nun die Beiträge, Pinnwand, Freunde etc. von Person B;

<u>Fehlersituationen:</u>	<u>Nachzustand im Fehlerfall:</u>
<u>Akteur(e):</u> User	<u>Trigger:</u> durch Benutzer ausgelöst
<u>Basisablauf:</u> <ol style="list-style-type: none"> 1. User navigiert zu Suchfunktion 2. User schreibt den gewünschten Namen im dargestellten Textfeld 3. User klickt auf einen Button, um die Person zu finden und um auf dessen Profil zu kommen 4. User klickt auf einen Button, um der Person zu folgen 	
<u>Alternativabläufe:</u> <ol style="list-style-type: none"> 1. User navigiert per über befreundete Kontakte und dessen Freunden zu noch nicht bekannten Personen 2. User klickt auf eine Person um auf dessen Profil zu gelangen 3. User klickt auf einen Button, um der Person zu folgen 	

Use-Case 7: Beitrag/Kommentar liken

<u>Ziel:</u> User hat angegeben ob ihm ein bestimmter Beitrag gefällt(Like) Phase: Sekundär	
<u>Kurzbeschreibung:</u> Um seine eigene Meinung nicht nur durch Kommentare preisgeben zu können, gibt es die Möglichkeit Beiträge anderer zu Liken.	
<u>Vorbedingung:</u> User ist eingeloggt;	<u>Nachbedingung:</u> Verfasser des Beitrags kann sehen welchen Personen ihr Beitrag gefällt und welchen nicht
<u>Fehlersituationen:</u>	<u>Nachzustand im Fehlerfall:</u>
<u>Akteur(e):</u> User	<u>Trigger:</u> durch Benutzer ausgelöst
<u>Basisablauf:</u> <ol style="list-style-type: none"> 1. User navigiert zu Beitrag 2. User klickt auf einen Button, um Beitrag zu Liken 	
<u>Alternativabläufe:</u> /	

Use-Case 8: Statistik anzeigen

<u>Ziel:</u> Forscher lässt sich eine Statistik zu bestimmten vorgegeben Parameter anzeigen Phase: Sekundär	
<u>Kurzbeschreibung:</u> Ein Forscher ist eine externe Person. Er steht mit keinem User in direktem Kontakt und muss auch kein User- Konto bei dem sozialen Netzwerk haben. Er hat einen eigenen "Forschierzugang" mit dem er sich Statistiken zu OldFace auswerten und ansehen kann.	
<u>Vorbedingung:</u> Forscher ist eingeloggt;	<u>Nachbedingung:</u> gesuchte Person wird angezeigt;
<u>Fehlersituationen:</u>	<u>Nachzustand im Fehlerfall:</u>
<u>Akteur(e):</u> Forscher;	<u>Trigger:</u> durch Forscher ausgelöst
<u>Basisablauf:</u> <ol style="list-style-type: none">1. Forscher navigiert zu Statistikfunktion2. Forscher gibt gewünschten Parameter für Statistik an3. Forscher klickt auf einen Button, um die Statistik auszuwerten und um sie sich anzeigen zu lassen	
<u>Alternativabläufe:</u> <ol style="list-style-type: none">1. Forscher navigiert zu Statistikfunktion2. Forscher klickt ohne Auswahl von Parameter auf einen Button, um die Statistik auszuwerten und um sie sich anzeigen zu lassen3. Forscher bekommt die drei vom System standardisiert vorgegebenen Statistiken angezeigt	

Use-Case 9: Beitrag/Kommentar löschen

<u>Ziel:</u> Unangebrachte Kommentare oder Beiträge von User sind von OldFace entfernt Phase: Sekundär	
<u>Kurzbeschreibung:</u> Sollten User andere User verbal attackieren oder unangebrachte Kommentare oder Beiträge auf OldFace stellen ist es dem Administrator vorbehalten solche zu löschen.	
<u>Vorbedingung:</u> Administrator ist eingeloggt	<u>Nachbedingung:</u> Beitrag oder Kommentar wurde erfolgreich gelöscht
<u>Fehlersituationen:</u>	<u>Nachbedingung im Fehlerfall:</u>
<u>Akteur(e):</u> Administrator	<u>Trigger:</u> durch Administrator ausgelöst

Basisablauf:

1. Administrator findet selber unangebrachten Kommentar
2. Administrator löscht nach Überprüfung den Beitrag oder Kommentar von OldFace

Alternativablauf:

/

Use-Case 10: User sperren

Ziel: User die unangebrachte Kommentare oder Beiträge auf OldFace teilen entfernt der Administrator

Phase: Sekundär

Kurzbeschreibung:

Wenn ein User nicht in das Bild des sozialen Netzwerkes passt ist es dem Administrator vorbehalten das Userkonto zu sperren.

Vorbedingung: Administrator ist eingeloggt

Nachbedingung: User wurde erfolgreich gelöscht

Fehlersituationen:

Nachbedingung im Fehlerfall:

Akteur(e): Administrator

Trigger: durch Administrator ausgelöst

Basisablauf:

1. Administrator findet User den er entfernen möchte
2. Administrator sperrt User von OldFace durch Sperr-button

Alternativablauf:

/

Use-Case 11: Freunde anzeigen

Ziel: User kann von jedem Freundesliste einsehen

Phase : Sekundär

Kurzbeschreibung:

Innerhalb des sozialen Netzwerkes ist es wichtig andere aktive Personen zu sehen. Am besten findet man als User diese über die Freundesliste von Freunden. Auch seine eigenen Freunde kann man sich als eine Liste anzeigen lassen.

Vorbedingung: User ist eingeloggt;

Nachbedingung: Freundesliste wird vollständig angezeigt

<u>Fehlersituationen:</u>	<u>Nachzustand im Fehlerfall:</u> Fehlerseite darstellen
<u>Akteur(e):</u> User	<u>Trigger:</u> durch Benutzer ausgelöst
<u>Basisablauf:</u> <ol style="list-style-type: none"> 1. User navigiert zu Freundesliste 2. Freundesliste wird dargestellt 	
<u>Alternativabläufe:</u> <ol style="list-style-type: none"> 1. User navigiert zu Suchfunktion 2. User schreibt den gewünschten Namen im dargestellten Textfeld 3. User klickt auf einen Button, um die Person zu finden und um auf dessen Profil zu kommen 4. User navigiert zu Freundesliste der Person 5. Freundesliste der Person wird angezeigt 	

Use-Case 12: Profildaten bearbeiten

<u>Ziel:</u> Profildaten können vom User bearbeitet werden Phase: Sekundär	
<u>Kurzbeschreibung:</u> Die Pinnwand dient als persönliches Profil, in dem alle selbst erstellten Beiträge gelistet werden und fungiert als Startseite nach dem Login. Sollten sich persönliche Daten ändern kann man diese auch nach dem erstellen ändern.	
<u>Vorbedingung:</u> User ist eingeloggt	<u>Nachbedingung:</u> Profildaten wurden erfolgreich geändert
<u>Fehlersituationen:</u>	<u>Nachbedingung im Fehlerfall:</u> Fehlerseite darstellen
<u>Akteur(e):</u> User	<u>Trigger:</u> durch Benutzer ausgelöst
<u>Basisablauf:</u> <ol style="list-style-type: none"> 1. a) User wird erfolgreich authentifiziert oder b) User klickt auf den Menüpunkt für seine persönliche Pinnwand 2. Die persönliche Pinnwand mit Userdaten und selbst erstellten Beiträgen wird angezeigt 3. User klickt auf Button um Profildaten zu verändern 4. User ändert gewünschte Profildaten 5. User bestätigt Änderung durch klick auf Button 	

Alternativablauf:

/

3. Nichtfunktionale Anforderungen

Beschreiben Sie, unter welchen Bedingungen die funktionalen Anforderungen erfüllt werden sollen.

1.1. Qualitätsanforderungen

(Effizienz, Benutzerfreundlichkeit, Zuverlässigkeit, usw.)

Aufgrund der Zielgruppe Senioren, steht natürlich die Benutzerfreundlichkeit ganz besonders im Fokus unseres Projektes. Auch mit minimalen bis keinen Vorkenntnissen was Social Networks und das generelle Surfen im Internet angeht, soll das Bedienen unserer Website möglich sein. Das Design soll simpel und intuitiv verständlich sein. Auch ein Tutorial mit einfacher Erklärung der Funktionen ist vorgesehen.

1.2. Technische Anforderungen

(Hardware, Netzwerk, Betriebssystem, benötigte Softwareversionen, usw.)

Aufgrund dessen, dass OldFace eine browser- basierte Anwendung ist, sind die Anforderungen an Hardware, Netzwerk und Betriebssystem marginal. Es ist Cross- Browser kompatibel und soll problemlos auf den wichtigsten Browsern wie Google Chrome, Internet Explorer und Mozilla Firefox laufen.

1.3. Realisierungsanforderungen

(Installation, Dokumentation, usw.)

Der User registriert sich auf der Website von OldFace und führte alle Aktivitäten online im Browser aus. Nach dem Registrieren ist zum Aufruf der Website eine beim Registrieren festgelegte Kombination aus Username und persönlichem Passwort notwendig.

Eine Installation ist nicht notwendig. Am Gerät des Users können Username und Passwort, wenn gewünscht, als Cookies gespeichert werden.

1.4. Diverses

(Annahmen, Risiken, usw.)

Es ist anzunehmen, dass die Zielgruppe zu größten Teilen technisch nicht sehr versiert ist, und noch wenig bis gar keine direkte Erfahrung mit Social Networks besitzt. Dies könnte sich entweder als Risiko oder als Chance herausstellen. Einerseits kann ein guter Ersteindruck von Social Networks gewonnen werden und es besteht eine gute Chance, dass der User auf keinen anderen Social Networks tätig ist. Andererseits müssen der Zielgruppe auch so schnell wie möglich die Sinnhaftigkeit und die Vorteile eines Social Networks näher gebracht werden, um ihr Interesse zu wecken.