

UE Software Engineering 050052 – Gruppe 10
WS 2015/16

LV-Leiter: Hans Moritsch

Anforderungsanalyse II

Projektname: Blue Couch - Das soziale Netzwerk

Projektteam:

Nachname	Vorname	Matrikelnummer	E-Mail-Adresse
Gazar	Mohamed	a0928951	a0928951@unet.univie.ac.at
Kolhaupt	Raphael	a1407523	a1407523@unet.univie.ac.at
Misurec	Patrik	a1325267	a1325267@unet.univie.ac.at
Pfneisl	Christian	a9525708	a9525708@unet.univie.ac.at

CEWebS-Teamseite: [Team 6: SN2 - Blue Couch](#)

Datum: 30.Dez.2015

1. Funktionale Anforderungen

Die funktionalen Anforderungen wurden aufgrund der Aufgabenstellen der LV sowie durch Analogie zu gleichartigen Systemen am Markt erhoben.

1.1. Funktionalität

BlueCouch ist eine Online-Plattform für ein **soziales Netzwerk**, welches für drei verschiedene Benutzergruppen unterschiedliche Funktionalitäten zur Verfügung stellt.

Die **primäre Funktion** eines sozialen Netzwerkes ist es mit anderen **Usern** in **Kontakt** zu treten / zu bleiben und mit diesen **Informationen** (hauptsächlich aus dem Bereich persönlicher Interessen) **auszutauschen**. Die Offline-Analogie sind real bestehende sozialen Netzwerke (z.B. Freundes-, Familien- und Bekanntenkreise), in welchen der Informationsaustausch über unterschiedlichste Kommunikationswege (persönliche Treffen, Telefonat, E-Mail, Apps, etc.) erfolgt.

Zusätzlich ist es **Forschern** bei BlueCouch möglich **statistische Daten** über die User und ihr Verhalten für Forschungszwecke zu erheben.

User können ihre persönlichen Profilinformationen verwalten, mit anderen Usern Freundschaftsbeziehungen eingehen, Beiträge auf ihrer Pinnwand posten. Zusätzlich können User Freunde zu verschiedenen Freundesgruppen zusammenfassen, sowie die Sichtbarkeit ihrer Profilinformation (auf Basis der Freundesgruppen) für jede Art seiner Information (z.B. Persönliche Interessen, Kontaktadressen, etc.) separat festlegen. Das Benutzerbild sowie der Vor- und Nachname eines Users sind jedoch jederzeit von allen anderen Usern einsehbar.

User haben weiters die Möglichkeit sich die eigene Seite (Freunde, Profilinformation, Pinnwand) und die Seiten anderer User anzuschauen, sofern sie mit diesen in einer Freundschaftsbeziehung stehen und ihnen die notwendigen Berechtigungen seitens des jeweiligen Eigentümers der Information gegeben wurde. Weiters können sie mittels Stichwortsuche nach nach anderen Usern suchen (um bspw. Diesen eine Freundschaftsanfrage zu schicken).

Administratoren können unangemessene Beiträge sperren und User (befristet oder unbefristet) sperren.

Sämtliche Anwender (User, Forscher, Administratoren) müssen sich einmalig registrieren bevor Ihnen ihre Funktionalitäten zur Verfügung stehen. Sie haben weiters selbstredend die Möglichkeit zum Login bzw. zum Logout.

1.2. Beschreibung der Funktionalitäten

Profilinformation ändern:

User haben die Möglichkeit ihre Profilinformation zu verwalten und die Sichtbarkeit (auf Basis der Freundesgruppen) jedes Informationsdetail (z.B. Interessen, etc.) separat festlegen. Vor- und Nachname sind jedoch jedem User uneingeschränkt einsehbar!

Freunde verwalten:

User können andere User einladen und mit Ihnen befreundet zu sein. Der eingeladene User kann die Einladung „ignorieren“ (= sie nicht beachten), annehmen oder ablehnen. Freunde können gruppiert werden (Freundesgruppen). Jeder User kann Freunde aus seinem Freundeskreis löschen. Freundesgruppen sind die Basis für die Festlegung der Sichtbarkeit der Profilinformation.

Beitrag hinzufügen:

User können Beiträge auf der eigenen Pinnwand hinzufügen.

Administratoren können unangemessene Beiträge eines User sperren, sofern diese von einem anderen User zuvor gemeldet wurden.

Userseite eines Freundes ansehen:

User und Administratoren haben die Möglichkeit sich die eigene Seite (Freunde, Profilinformation, Pinnwand) und die Seiten anderer User anzuschauen. Administratoren haben dieses Recht uneingeschränkt; User nur insofern sie für die entsprechende Information die Berechtigung durch den anderen User haben.

User Suchen:

User und Administratoren können mittels Stichwortsuche nach Beiträgen auf Pinnwänden sowie nach anderen Personen suchen.

Beitrag melden (User) und prüfen (Admin):

User haben die Möglichkeit einen unangemessenen Beitrag zu melden (einfach mittels Button und ohne Begründung). Dieser wird vom Administrator geprüft und gegebenenfalls wird der Beitrag von der Pinnwand gelöscht und/oder der User befristet oder unbefristet gesperrt.

User sperren:

Administratoren können User (befristet und unbefristet) sperren.

Beitrag von Pinnwand löschen:

Administratoren können unangemessene Beiträge eines User von der Pinnwand löschen, sofern diese von einem anderen User zuvor gemeldet wurden.

Statistik auswerten:

Forscher können auf aggregierter Ebene diverse Statistiken (Anzahl der geschlossenen Freundschaften aller Benutzer in gegebener Zeitspanne, Anzahl der durchschnittlichen Freunde aller User, Anzahl der gesamten Freundschaftsbeziehungen aller User, Anzahl der Beiträge aller User in gegebener Zeitspanne, Anzahl der durchschnittlichen Beiträge aller User, Anzahl der aktuell gesperrten User) abrufen.

1.3. Bedienoberfläche

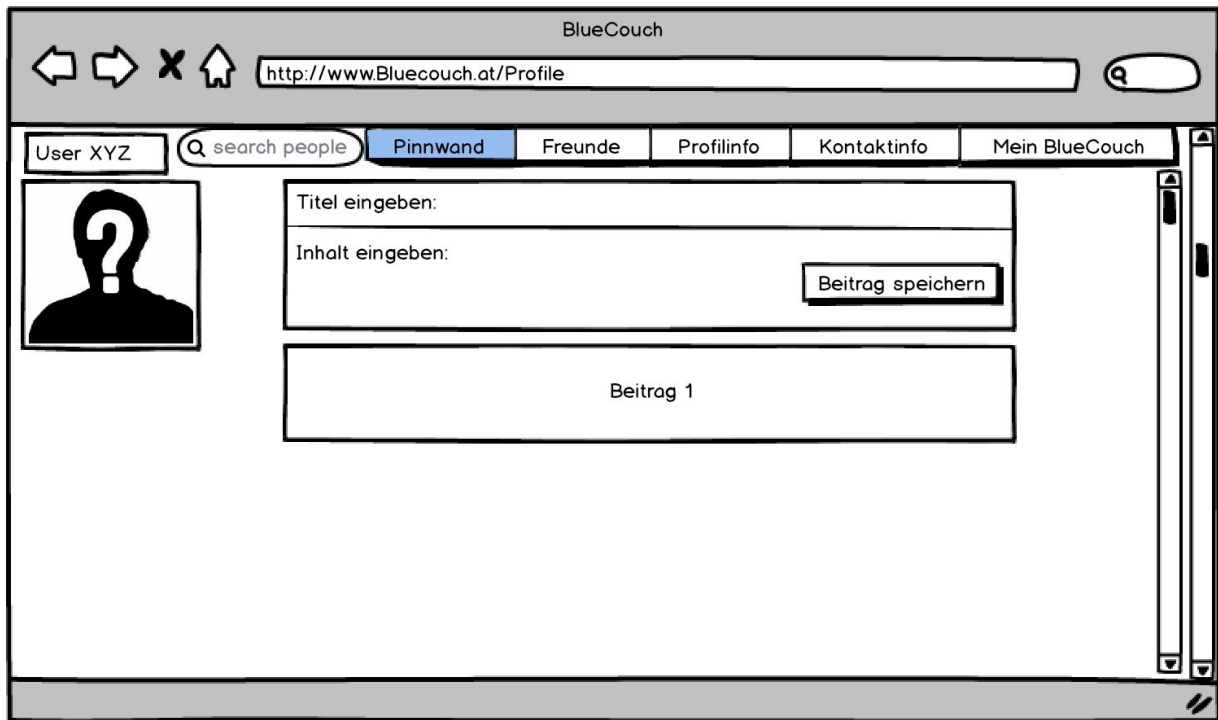


Abbildung 1 Profile – Persönliche Pinnwand

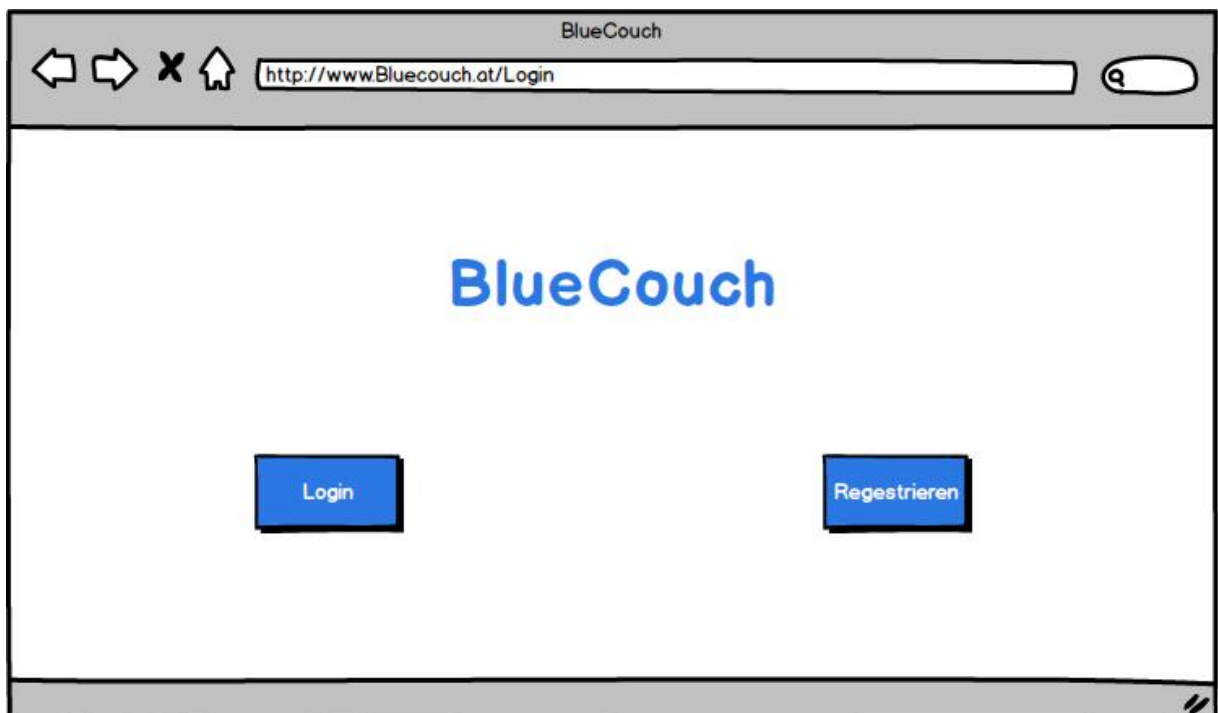


Abbildung 2 Startseite (wenn nicht eingeloggt)

BlueCouch

http://www.Bluecouch.at/Home/Login

Home
BlueCouch

Email:

Passwort:

☐ User
☐ Admin
☐ Forscher

Register

Abb. 1: Registrierungsseite

BlueCouch

http://www.Bluecouch.at/Home/Login

Home
BlueCouch

Email:

Passwort:

Login

Abb. 2 Login Seite

2. Use-Case-Modell

2.1. Use-Case-Diagramm

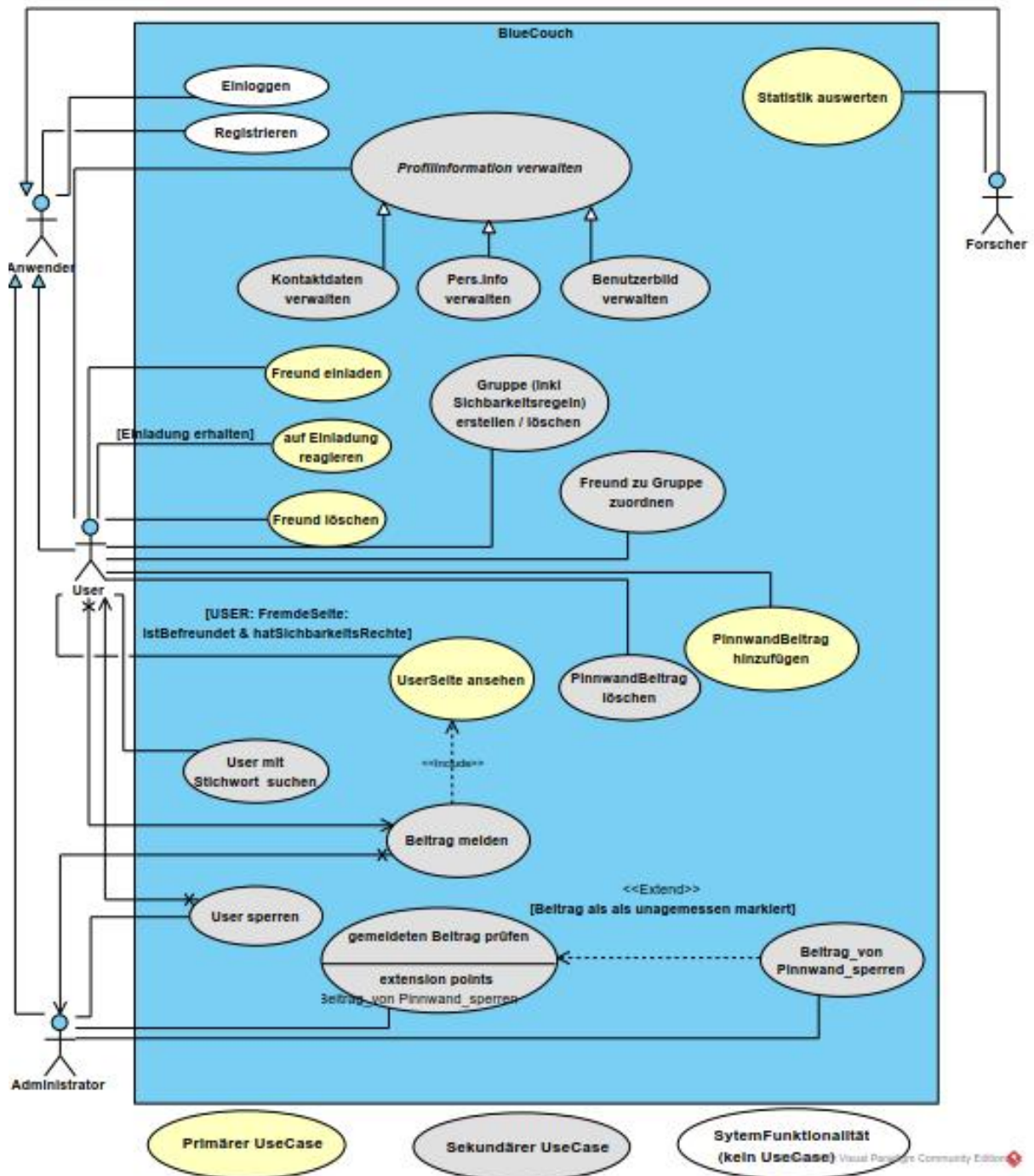


Abb. 3: Use-Case Übersicht

2.2. Use-Case Beschreibungen

Generelle Festlegung Nutzerechte / Einloggen: Man kann darüber diskutieren ob das Vorhandensein von Rechten (Akteur hat Rolle XY) bzw. der Zustand des eingeloggt sein Vorbedingungen darstellen und somit bei Nichterfüllen zu Fehlersituation führen. Es wurde festgelegt, dies *nicht als Vorbedingung bzw. Fehlersituation im Use-Case zu modellieren*, sondern als Systemvoraussetzung zu sehen (welche natürlich in der Implementierung überprüft werden muss).

2.2.1. Use-Case 1: Kontaktinformationen ändern

Ziel:	Kontaktinformationen im System aktualisieren/ändern
Kurzbeschreibung:	Der Benutzer hat die Möglichkeit seine Kontaktinformationen zu ändern. Diese beinhalten Wohnort, Adresse, E-Mail-Adresse, Telefonnummer, Homepage
Kategorie:	sekundär
Vorbedingung:	keine
Nachbedingung bei Erfolg:	Kontaktinformationen gespeichert
Fehlersituation:	Keine
Nachbedingung bei Fehler:	Keine
Akteure:	Benutzer
Auslösendes Ereignis:	Keines
Basisablauf:	<ol style="list-style-type: none">1. User gibt neue jeweilige Kontaktinformation ein2. User wählt "Änderungen speichern" aus3. Änderungen werden gespeichert
Alternativabläufe:	<ol style="list-style-type: none">2a User wählt "Abbruch" aus oder verlässt die Seite ohne zu speichern3a Änderungen werden nicht gespeichert

BlueCouch

http://www.Bluecouch.at/Forscherxy

search people

PinnwandFreundeProfilinfoMein BlueCouch

Kontaktinformationen ändern

Vorname

Adresse

Nachname

Wohnort

Telefonnummer

Email

Speichern

Abbrechen

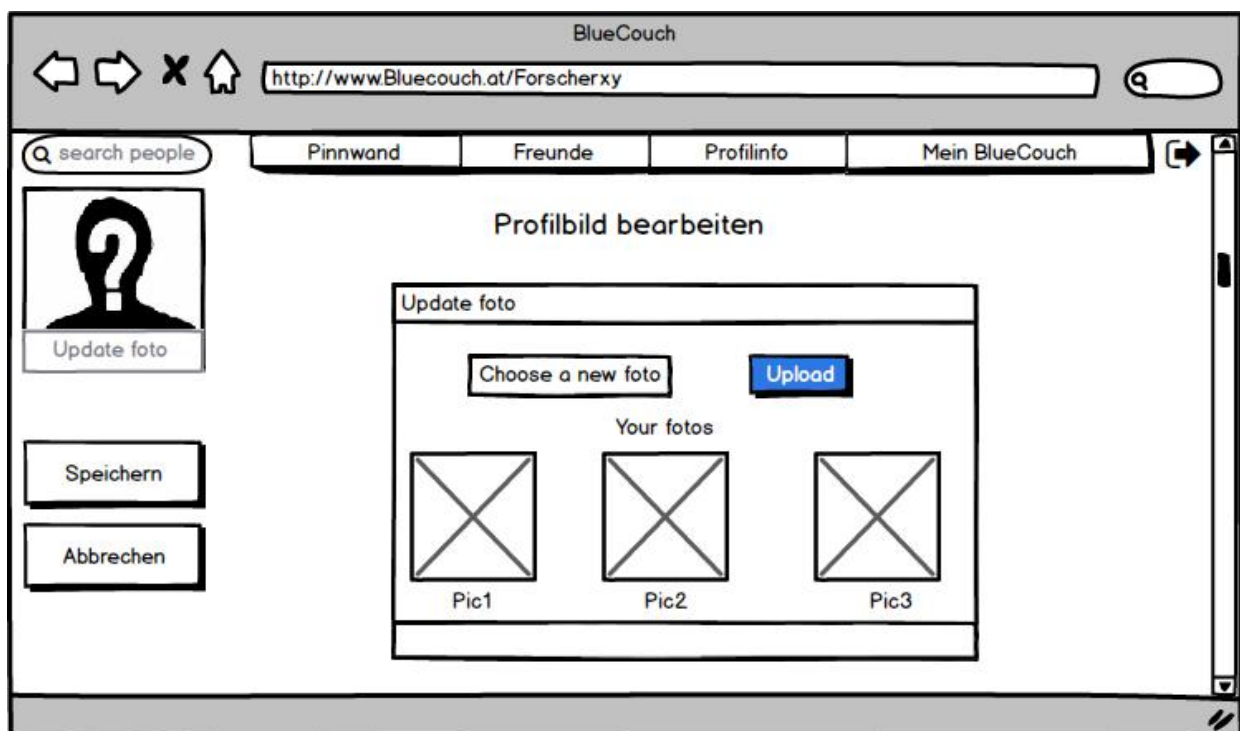
2.2.2. Use-Case 2: Persönliche Informationen ändern

- Ziel:** Persönliche Informationen im System aktualisiert
- Kurzbeschreibung:** Benutzer hat die Möglichkeit seine persönlichen Informationen zu verändern. Dies beinhaltet seinen Vor- und Nachnamen, sein Geburtsdatum, seine Hobbies, seine Interessen, sein Geschlecht, seinen Familienstatus, seine sexuelle Orientierung, seine Religion, seine politische Einstellung und ein frei zur Verfügung stehendes Textfeld.
- Kategorie:** sekundär
- Vorbedingung:** Keine
- Nachbedingung bei Erfolg:** Profilinformationen gespeichert
- Fehlersituation:** Keine
- Nachbedingung bei Fehler:** Keine
- Akteure:** Benutzer
- Auslösendes Ereignis:** Keine
- Basisablauf:**
1. User gibt neue jeweilige persönliche Information ein
 2. User wählt "Änderungen speichern" aus
 3. Änderungen werden gespeichert
- Alternativabläufe:**
- 2a User wählt "Abbruch" aus oder verlässt die Seite ohne zu speichern
 - 3a Änderungen werden nicht gespeichert

The screenshot shows a web browser window titled 'BlueCouch' with the address bar displaying 'http://www.Bluecouch.at/Forscherxy'. The browser's navigation bar includes back, forward, and home buttons. Below the address bar is a search bar labeled 'search people' and a navigation menu with tabs: 'Pinnwand', 'Freunde', 'Profilinfo' (selected), 'Kontaktinfo', and 'Mein BlueCouch'. The main content area is titled 'Persönliche Informationen bearbeiten' and features a profile picture placeholder with a question mark. The form contains several input fields and dropdown menus for personal data: 'Vorname', 'Nachname', 'Geburtsdatum', 'Hobbies', 'Interessen', 'Geschlecht' (with a dropdown menu showing 'Male'), 'Familienstatus' (with a dropdown menu showing 'Ledig'), 'Sexuelleorientierung', 'Religion', and 'Politische Einstellung'. On the left side of the form, there are two buttons: 'Speichern' (Save) and 'Abbrechen' (Cancel).

2.2.3. Use-Case 3: Benutzerbild ändern

Ziel:	Benutzerbild im System aktualisiert
Kurzbeschreibung:	Benutzer hat die Möglichkeit ein Bild seiner Wahl auf der Plattform als eigenes Benutzerbild anzeigen zu lassen.
Kategorie:	sekundär
Vorbedingung:	Keine
Nachbedingung bei Erfolg:	Benutzerbild gespeichert
Fehlersituation:	Keine
Nachbedingung bei Fehler:	Keine
Akteure:	Benutzer
Auslösendes Ereignis:	Keines
Basisablauf:	<ol style="list-style-type: none">1. Benutzer wählt „Neues Benutzerbild auswählen“2. Benutzer wählt ein neues Benutzerbild aus.3. Benutzer wählt „Bild speichern“ aus.4. Änderungen werden gespeichert
Alternativabläufe:	<ol style="list-style-type: none">3a User wählt „Abbruch“ aus oder verlässt die Seite ohne zu speichern4a Änderungen werden nicht gespeichert



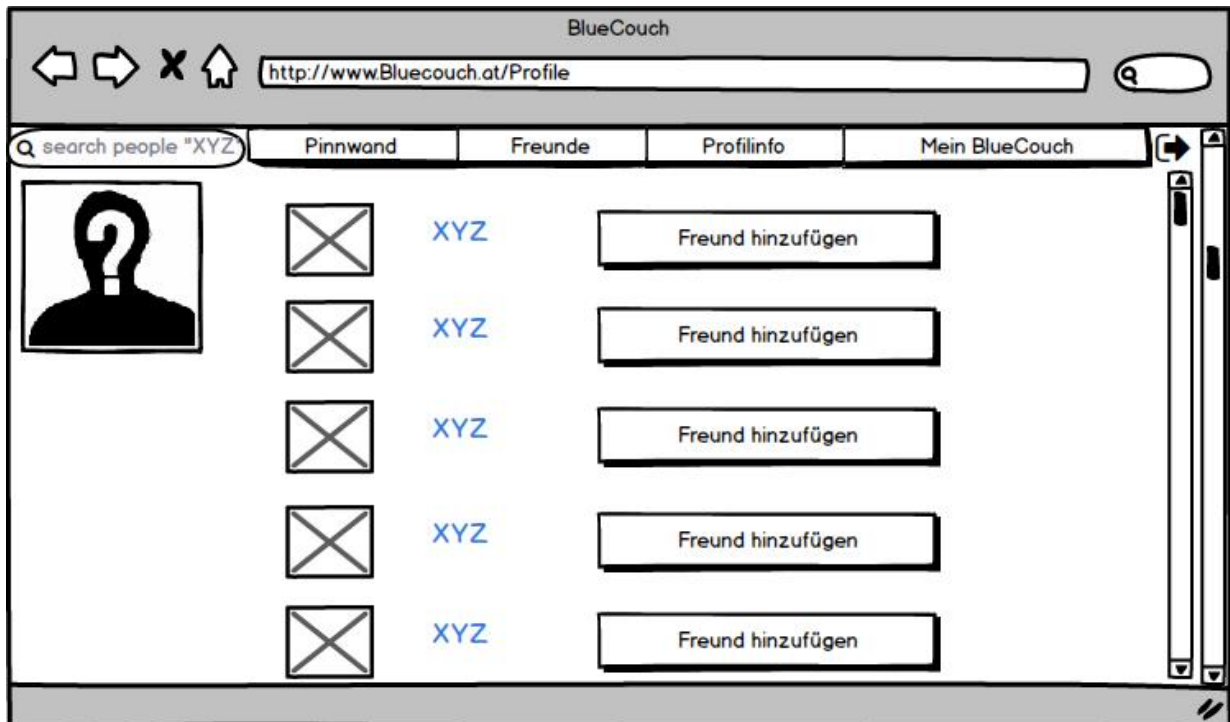
2.2.4. Use-Case 4: Gruppe samt Sichtbarkeitsrechte erstellen

Ziel:	Gruppe erstellen und festlegen über welche Sichtbarkeitsrechte ihre Mitglieder verfügen.
Kurzbeschreibung:	Benutzer erstellen Gruppen, mittels einem Gruppennamen, wobei Sie festlegen können über welche Sichtbarkeitsrechte die Mitglieder dieser Gruppe verfügen. Dies entscheidet was den jeweiligen Freunden beim betrachten der Seite des Users angezeigt bekommen.
Kategorie:	Sekundär
Vorbedingung:	keine
Nachbedingung bei Erfolg:	Gruppe erstellt und Sichtbarkeitsrechte f. Gruppe festgelegt
Fehlersituation:	Keine
Nachbedingung bei Fehler:	Keine
Akteure:	Benutzer
Auslösendes Ereignis:	Keine
Basisablauf:	<ol style="list-style-type: none">1. User wählt Button „Gruppe“ unter dem Register „Freunde“2. Namen f. Gruppe einfügen2. Checkboxen f. jeweilige Sichtbarkeiten festlegen3. „Gruppe erstellen“ drücken4. Gruppe wird gespeichert
Alternativabläufe:	<ol style="list-style-type: none">3a User verlässt die Seite ohne zu speichern4a Änderungen werden nicht gespeichert, Grundzustand

The screenshot shows the BlueCouch web application interface. At the top, there is a browser window with the address bar showing "http://www.Bluecouch.at/Profile". Below the browser window, there is a navigation bar with tabs: "Pinnwand", "Freunde", "Profilinfo", "Kontaktinfo", and "Mein BlueCouch". The "Freunde" tab is currently selected. On the left side, there is a sidebar with a search bar labeled "search people" and a button labeled "Gruppen". The main content area displays a form to create a new group. The form has a field for "Gruppenname:" and four checkboxes for visibility settings: "Freundliste", "Pinnwand", "Kontaktinformationen", and "Profilinfo". Below these checkboxes is a button labeled "Gruppe erstellen". Below this form, there is a section for "UniGruppe" with the same four checkboxes and a button labeled "Gruppe löschen".

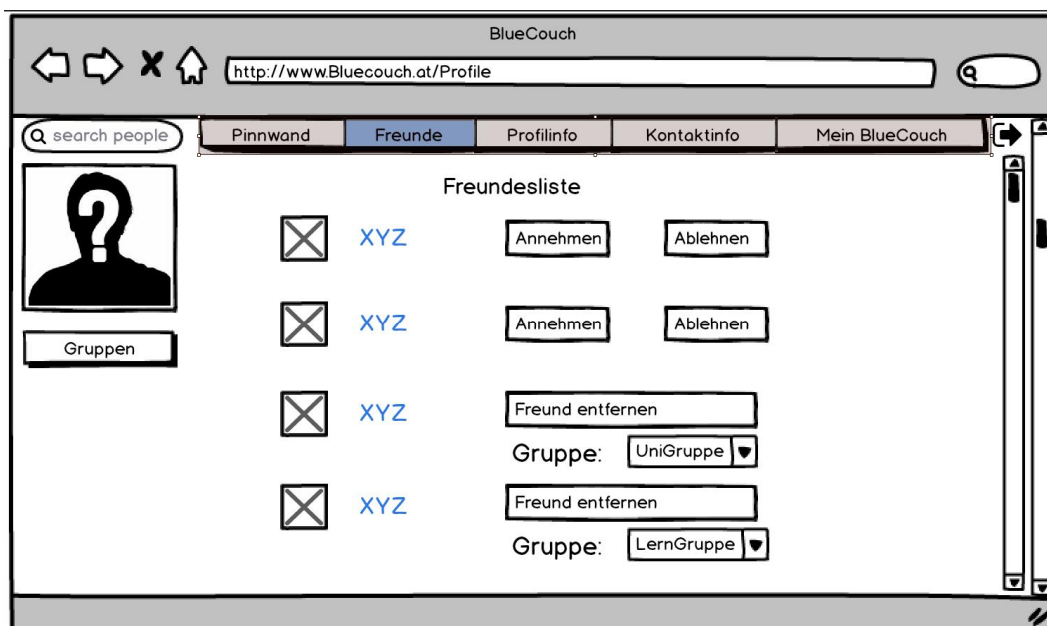
2.2.5. Use-Case 5: Freund einladen

Ziel:	Freundschaftsanfrage verschicken
Kurzbeschreibung:	Ein Benutzer hat die Möglichkeit einem anderen Benutzer eine Freundschaftseinladung zu senden, so er mit diesem Benutzer eine Freundschaft haben möchte.
Kategorie:	Primär
Vorbedingung:	keine
Nachbedingung bei Erfolg:	Freundschaftsanfrage an zu befreundenden Benutzer versandt (= Anzeige der Anfrage im Profil des an zu befreundenden Benutzer)
Fehlersituation:	Keine
Nachbedingung bei Fehler:	Keine
Akteure:	Benutzer
Auslösendes Ereignis:	Keines
Basisablauf:	<ol style="list-style-type: none">1. Benutzer sucht anderen Benutzer2. Benutzer besucht Profilseite des zu befreundenden Benutzers.3. Benutzer sieht hierbei lediglich leere Seite, mit individuellem Button „Freundschaftseinladung versenden“4. Benutzer betätigt Button.5. Freundschaftseinladung wird an anderen Benutzer versandt
Alternativabläufe:	Keine



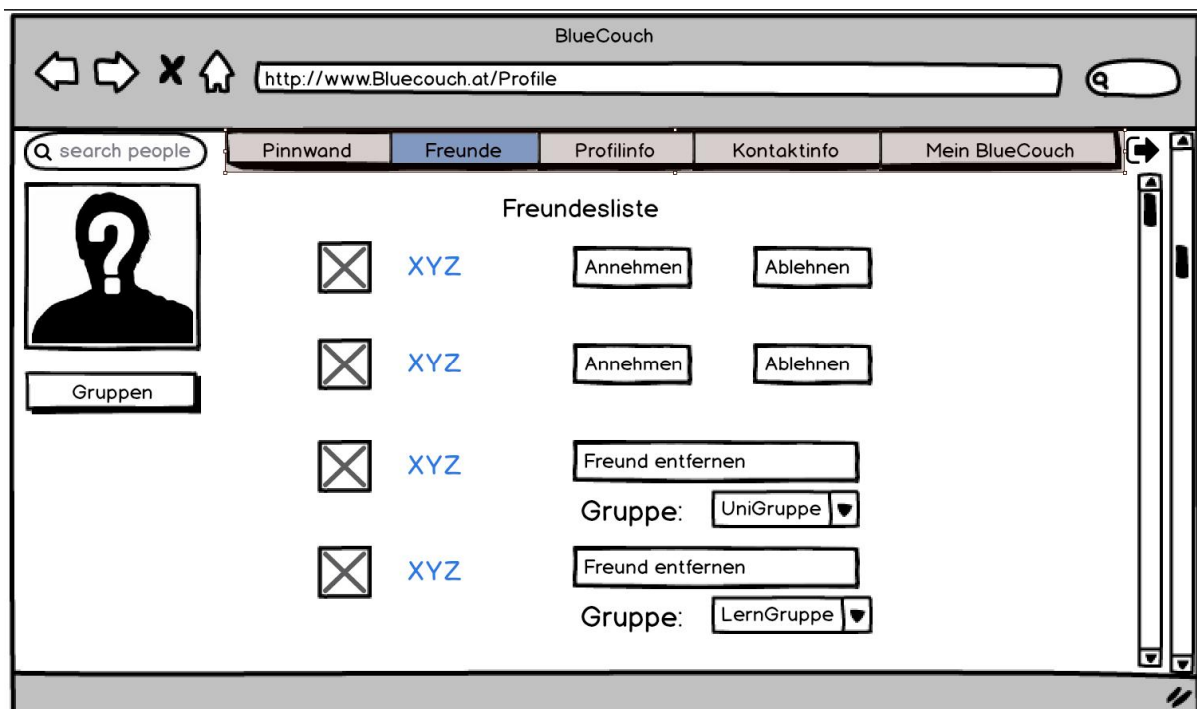
2.2.6. Use-Case 6: auf Freundschaftseinladung reagieren

Ziel:	Auf Freundschaftseinladung reagieren
Kurzbeschreibung:	Wenn der Benutzer eine Freundschaftseinladung eines anderen Benutzers erhält, kann er auf diese mittels 2 Buttons reagieren und so entweder die Einladung annehmen oder Ablehnen.
Kategorie:	primär
Vorbedingung:	Benutzer erhält eine Freundschaftseinladung eines anderen Benutzers.
Nachbedingung bei Erfolg:	Möglichkeit 1: Benutzer mit anderen Benutzer befreundet Möglichkeit 2: Benutzer mit anderen Benutzer nicht befreundet
Fehlersituation:	Keine
Nachbedingung bei Fehler:	Keine
Akteure:	Benutzer
Auslösendes Ereignis:	Freundschaftseinladung erhalten
Basisablauf:	1. Benutzer erhält Freundschaftseinladung. 2. Benutzer sieht an erster Stelle in seine Freundesliste die Einladung 3. Benutzer kann über Button „Annehmen“ o. „Ablehnen“ auszuführende Aktion bestimmen.
Alternativabläufe:	Keine
Sonstiges:	Neu befreundete Benutzer werden in Gruppe „Allgemein“ eingeteilt → Keine Sichtbarkeitsrechte



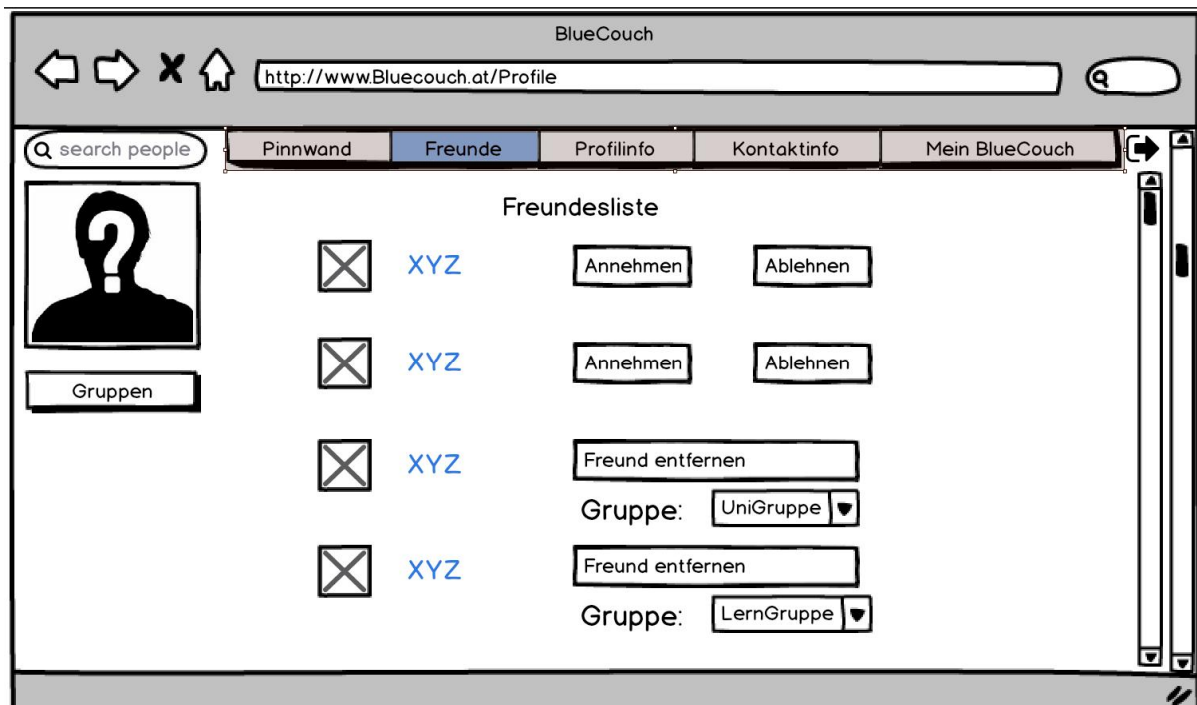
2.2.7. Use-Case 7: Freund aus Freundesliste löschen

Ziel:	Freund entfernen
Kurzbeschreibung:	Benutzer soll Freunde aus seiner Freundesliste entfernen können, wodurch alle Sichtbarkeitsrechte beiderseits selbstverständlich entzogen werden.
Kategorie:	Sekundär
Vorbedingung:	Freund vorhanden
Nachbedingung bei Erfolg:	Freund aus Freundesliste entfernt
Fehlersituation:	Kein Freund vorhanden
Nachbedingung bei Fehler:	Keine
Akteure:	Benutzer
Auslösendes Ereignis:	Keine
Basisablauf:	1.Freund auswählen. 2.„Freund entfernen“ klicken.
Alternativabläufe:	Keine



2.2.8. Use-Case 8: Freunde zu Gruppe zuordnen

Ziel:	Freunde in verschiedene Gruppen Einteilen
Kurzbeschreibung:	Der Benutzer soll seine Freunde in Gruppen zwecks Sichtbarkeit seiner Informationen einteilen können.
Kategorie:	sekundär
Vorbedingung:	mind. 1 Freund vorhanden
Nachbedingung bei Erfolg:	Freund zu Gruppe hinzugefügt
Fehlersituation:	Kein Freund vorhanden
Nachbedingung bei Fehler:	Keine
Akteure:	Benutzer
Auslösendes Ereignis:	Keines
Basisablauf:	<ol style="list-style-type: none">1. Freund auswählen.2. Freund zu Gruppe hinzufügen3. Button „Gruppen“ zum speichern drücken
Alternativabläufe:	keine
Sonstiges:	Freunde in maximal eine Gruppe einteilbar



2.2.9. Use-Case 9: Beitrag erstellen

Ziel:	Beitrag zu Pinnwand hinzufügen
Kurzbeschreibung:	Benutzer soll Nachricht eingeben, und diese an seiner Pinnwand anzeigen lassen
Kategorie:	primär
Vorbedingung:	Keine
Nachbedingung bei Erfolg:	Nachricht wird an Pinnwand angezeigt
Fehlersituation:	Kein Titel eingegeben / Kein Inhalt eingegeben
Nachbedingung bei Fehler:	Fehlermeldung erscheint f. Nutzer
Akteure:	Benutzer
Auslösendes Ereignis:	Keines
Basisablauf:	<ol style="list-style-type: none">1. Benutzer tippt Titel der Nachricht ein2. Benutzer tippt eine Nachricht ein3. Benutzer wählt „Beitrag speichern“
Alternativabläufe:	Keine

BlueCouch

http://www.Bluecouch.at/Profile

User XYZ search people Pinnwand Freunde Profilinfo Kontaktinfo Mein BlueCouch

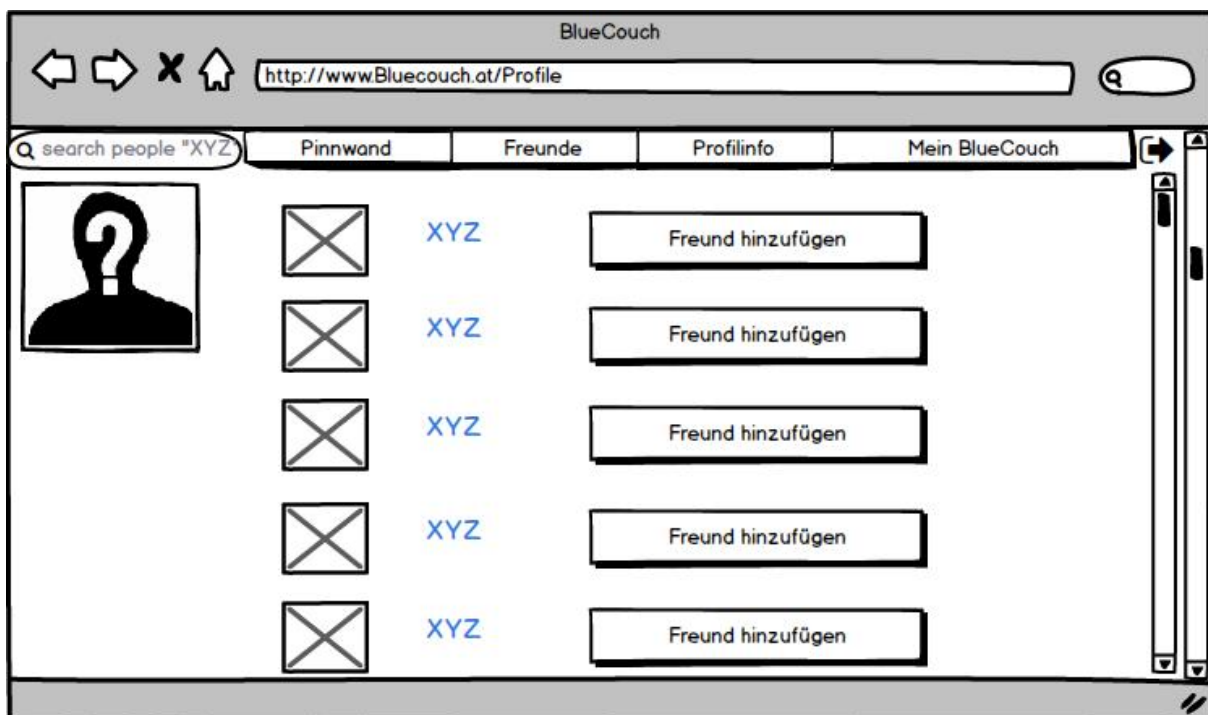
Titel eingeben:

Inhalt eingeben: Beitrag speichern

Beitrag 1

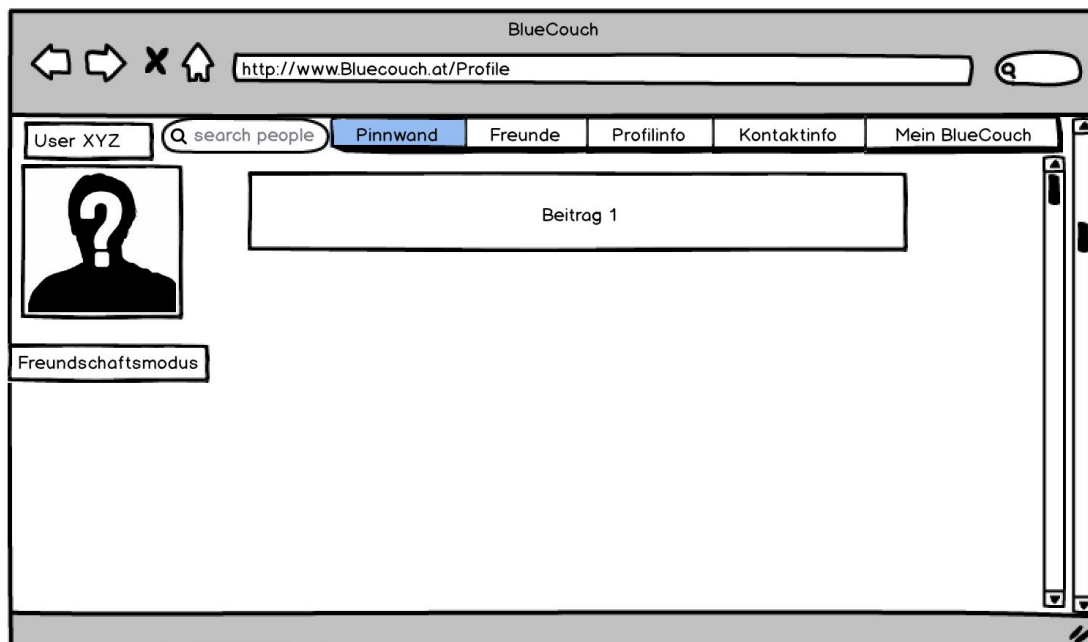
2.2.10. Use-Case 10: User suchen

Ziel:	User erhält die Information der Stichwortsuche am Bildschirm angezeigt
Kurzbeschreibung:	Der User kann einen anderen User mittels Stichwort(wörtern) suchen.
Vorbedingung:	keine
Nebenbedingung bei Erfolg:	Treffer sind am Bildschirm ausgegeben
Fehlersituationen:	keine
Nachzustand im Fehlerfall:	keiner
Akteure:	User
Auslösendes Ereignis:	keines
Basisablauf:	1.User sucht eine bestimmte Person durch eine Stichwort 2.Alle Personen die das gleiche Stichwort bzw. eine ausreichende Ähnlichkeit haben werden gezeigt



2.2.11. Use-Case 11: UserSeite eines Freundes ansehen

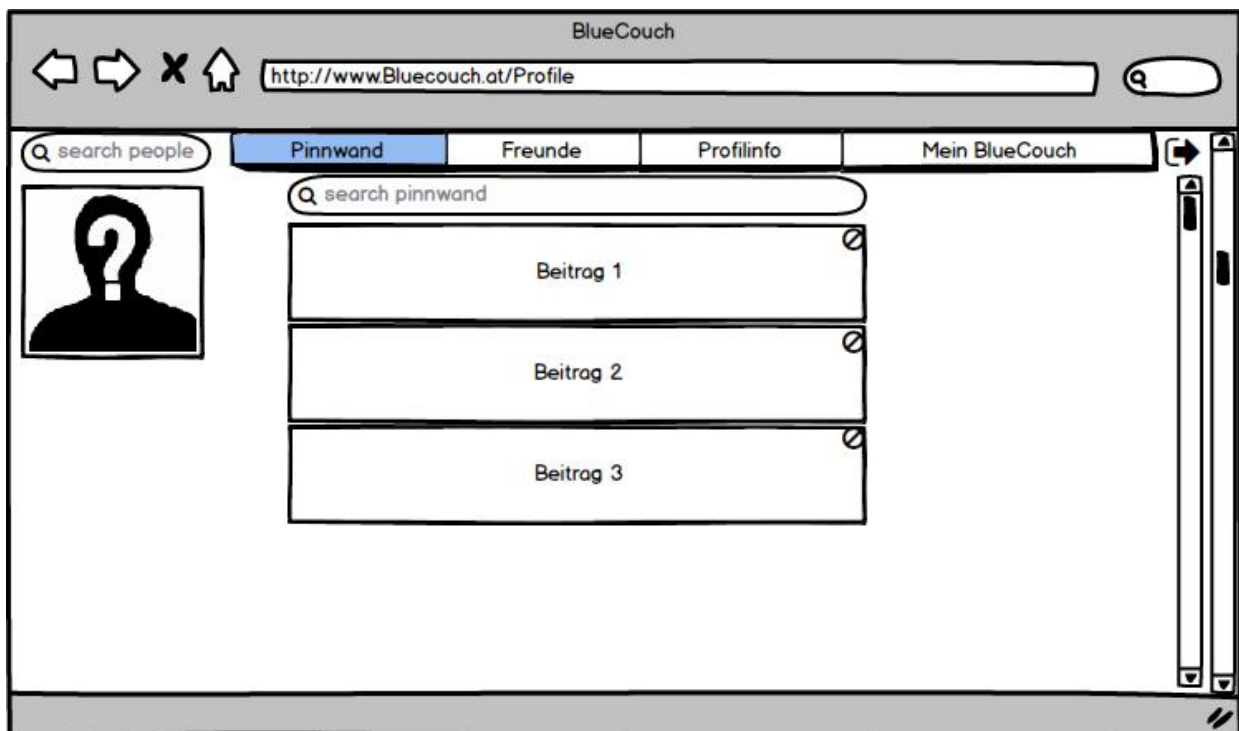
Ziel:	Pinnwand samt Beiträge und Information eines Benutzers anzeigen
Kurzbeschreibung:	Der Benutzer bzw. ein Admin soll die Informationen sowie die Pinnwandbeiträge eines anderen Benutzers sehen können.
Kategorie:	primär
Vorbedingung:	Anwender ist Benutzer und befreundet mit anderen Benutzer oder Anwender ist Admin
Nachbedingung bei Erfolg:	UserSite wird angezeigt
Fehlersituation:	Anwender ist Benutzer jedoch nicht befreundet
Nachbedingung bei Fehler:	Leere Benutzerseite wird angezeigt
Akteure:	Admin oder Benutzer
Auslösendes Ereignis:	Keine
Basisablauf:	1.Person auswählen (über Suchfeld oder Freundesliste). 2.Seite anzeigen lassen.
Alternativabläufe:	Keine
Sonstiges:	Nutzer können desweiteren unangemessene Beiträge melden (mittels Extend realisiert)



2.2.12.

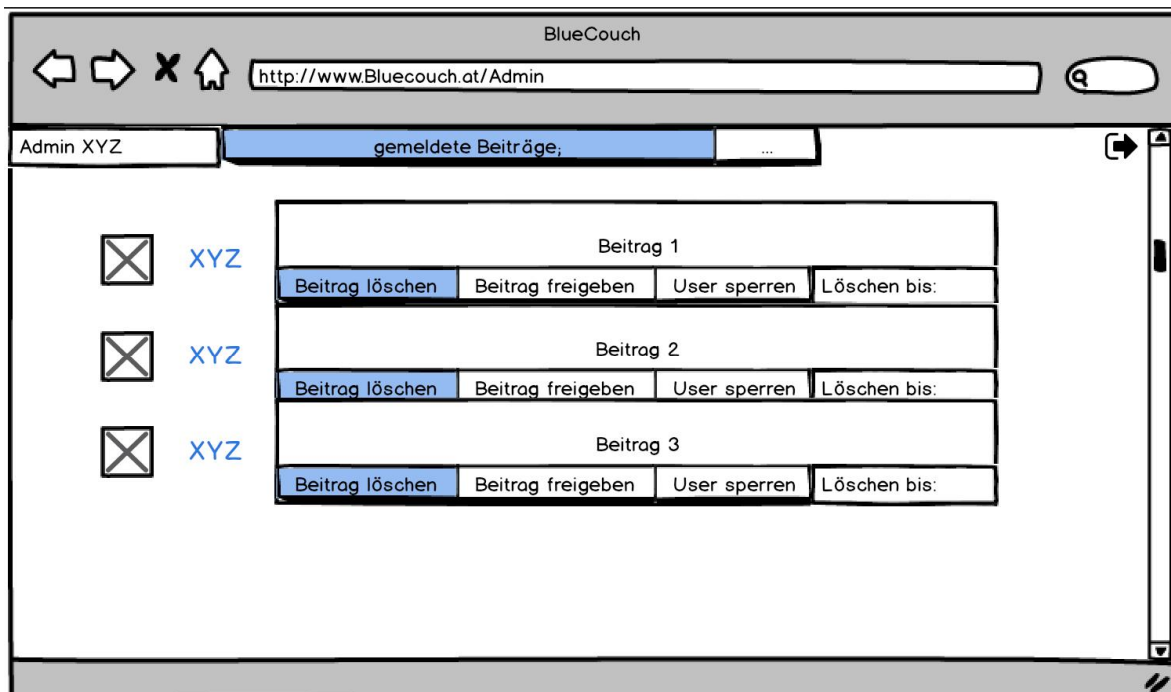
Use-Case 12: Beitrag eines Freundes melden

Ziel:	User meldet ungemessenen Beiträge
Kurzbeschreibung:	Findet ein User einen Beitrag unangemessen kann er diesen melden (einfach Button bei jedem Beitrag). Es erfolgt eine automatische Benachrichtigung des Administrators.
Kategorie:	sekundär
Vorbedingung:	keine
Nachbedingung bei Erfolg:	Meldung ist erfolgt. Information scheint bei Administrator auf
Fehlersituation:	keine
Nachbedingung bei Fehler:	keine
Akteure:	User, Administrator
Auslösendes Ereignis:	keines
Basisablauf:	1. User klickt Symbol für „Beitrag melden“
Alternativabläufe:	keine



2.2.13. Use-Case 13: gemeldeten Beitrag prüfen

Ziel:	gemeldeten Pinnwandbeiträge eines Benutzers prüfen
Kurzbeschreibung:	Administratoren prüfen gemeldete
Kategorie:	sekundär
Vorbedingung:	Beitrag erscheint in der Liste „gemeldeten Beiträge“ auf
Nachbedingung bei Erfolg:	Beitrag bleibt auf Pinnwand oder ist als ungemessen markiert
Fehlersituation:	Keine
Nachbedingung bei Fehler:	Keine
Akteure:	Admin
Auslösendes Ereignis:	Keines
Basisablauf:	<ol style="list-style-type: none">1.Admin lässt Liste „gemeldeten Beiträge“ anzeigen.2.Admin wählt entsprechenden Beitrag aus.3.Admin entscheidet ob der Beitrag auf der Pinnwand bleibt oder als unangemessen markiert wird
Alternativabläufe:	<ol style="list-style-type: none">4a Admin wählt „Beitrag löschen“5a Beitrag wurde gelöscht4b Admin wählt „Beitrag freigeben“5b Beitrag wurde freigegeben
Sonstiges:	Alternativ kann der Admin den User dem der Beitrag gehört auch Sperren (siehe UseCase „User sperren“)



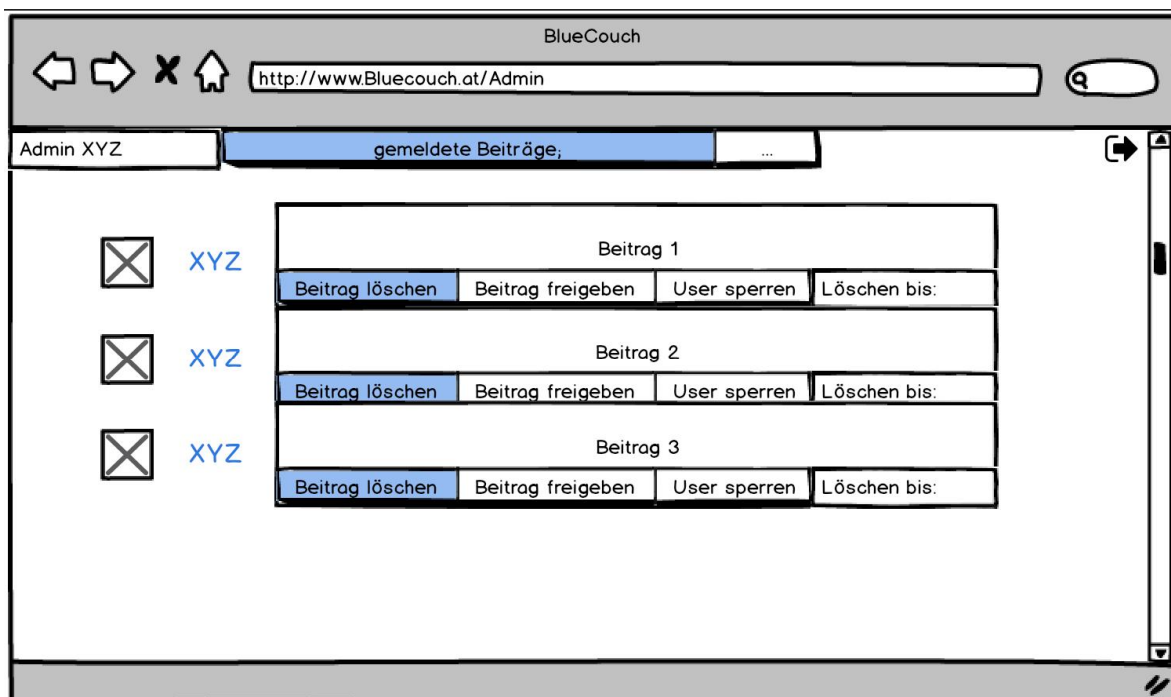
2.2.14. Use-Case 14: Beitrag von Pinnwand löschen

Ziel:	Beitrag löschen
Kurzbeschreibung:	Benutzer können Ihre erstellen Beiträge löschen
Vorbedingung:	keine
Kategorie:	sekundär
Nachbedingung bei Erfolg:	Beitrag ist von Pinnwand gelöscht
Fehlersituation:	keine
Nachbedingung bei Fehler:	keine
Akteure:	User
Auslösendes Ereignis:	keines
Basisablauf:	1. Benutzer wählt „Beitrag löschen“
Alternativabläufe:	keine
Sonstiges:	/

The screenshot shows a web browser window titled "BlueCouch". The address bar displays "http://www.Bluecouch.at/Profile". The browser's navigation bar includes back, forward, and home buttons. Below the address bar, there is a navigation menu with tabs: "User XYZ", "search people", "Pinnwand" (highlighted), "Freunde", "Profilinfo", "Kontaktinfo", and "Mein BlueCouch". On the left side of the page, there is a placeholder for a user profile picture, represented by a silhouette with a question mark. The main content area contains a form for creating a post with two input fields: "Titel eingeben:" and "Inhalt eingeben:". A "Beitrag speichern" button is located to the right of the "Inhalt eingeben:" field. Below the form, there is a box labeled "Beitrag 1". The right side of the page features a vertical scrollbar.

2.2.15. Use-Case 15: User sperren

Ziel:	User werden vom Administrator (befristet oder unbefristet) gesperrt
Kurzbeschreibung:	Der Administrator sperrt User eine bestimmte Zeitlang. Bei groben Verstößen wird der User unbefristet gesperrt.
Kategorie:	sekundär
Vorbedingung:	keine
Nachbedingung bei Erfolg:	User (befristet oder unbefristet) gesperrt; User erhält statt seiner Profilseite eine Seite mit der Information, dass er gesperrt ist
Fehlersituation:	keine
Nachbedingung bei Fehler:	keine
Akteure:	Administrator, User
Auslösendes Ereignis:	keines
Basisablauf:	1. Admin betätigt Button „User sperren“
Alternativabläufe:	keine



2.2.16.

Use Case 16: Statistik auswerten

Ziel:	Forschern hat statistische Auswertung auf dem Bildschirm zur Verfügung
Kurzbeschreibung:	Der Forscher kann aus einer vorgegeben Liste möglicher Auswertungen wählen und erhält das Resultat für die jeweilige Auswertung am Bildschirm angezeigt. Es ist ihm auch möglich alle Statistiken auf einmal anzeigen zu lassen
Kategorie:	primär
Vorbedingung:	keine
Nachbedingung bei Erfolg:	Auswertungsergebnisse angezeigt
Fehlersituation:	keine
Nachbedingung bei Fehler:	keine
Akteure:	Forscher
Auslösendes Ereignis:	keines
Basisablauf:	<ol style="list-style-type: none">1. Forscher wählt gewünschte statistischen Auswertung2. Forscher wählt Auswertungsparameter (wo erforderlich) aus3. Anzeige des Ergebnisses
Alternativabläufe:	1a. Forscher wählt die Berechnung aller Auswertungen aus
Sonstiges:	keine

The screenshot shows a web browser window titled "BlueCouch" with the address bar displaying "http://www.Bluecouch.at/Forscherxy". The page content includes a header with "Forscher XYZ" and a "Statistiken" button. Below this, there is a list of six statistical categories, each with a text input field and a checkbox:

- Anzahl der Freunde / User
- Anzahl neuer Freunde / User (letztes Monat)
- Gesamte Anzahl von Freundschaftsbeziehungen
- Anzahl Beiträge / User
- Anzahl neuer Beiträge / User (letztes Monat)
- Anzahl aktuell gesperrter User

At the bottom of the form, there are three buttons: "Auswahl berechnen", "Alle berechnen", and "Auswahl aufheben".

3. Nichtfunktionale Anforderungen

3.1. Qualitätsanforderungen

Benutzerfreundlichkeit: das System soll für alle User von der Sprache her leicht verständlich sein. Die einzelnen Funktionen (wie man einen Beitrag auf der Pinnwand postet, einen Freund hinzufügt, usw. – sämtliche Funktionalitäten) sollten so gestaltet werden, dass diese für den Benutzer schnell und einfach zu erlernen sind. Ebenso sollte das System eine einfache und übersichtliche Bedienbarkeit für den Benutzer gewährleisten.

Auch Administratoren und Personen welche für die statistische Datenerfassung verantwortlich sind, sollen einen möglichst einfachen Einstieg ermöglicht bekommen – Ziel ist nicht eine hochgradig komplexe Anwendung zu erstellen, sondern die Benutzerfreundlichkeit, auf ein Maximum zu erhöhen.

Zuverlässigkeit: Fehler sollen abgefangen und behandelt werden. Eine entsprechende Fehlermeldung wird zurückgegeben, damit die User entsprechend dieser weiter reagieren können. Sollte der Fehler nicht systemseitig behebbar sein, so soll die Fehlermeldung möglichst eindeutig und klar definiert sein, um dem Benutzer eine einfache Möglichkeit aufzuzeigen, wie dieser zu seinem gewünschten Ergebnis kommt.

Effizienz: User soll auf die Antwort seiner Anfrage nicht lange warten müssen. Abfragen sollten eine möglichst positives Laufzeitverhalten aufweisen, um ein zügiges arbeiten zu ermöglichen. Dies trägt maßgeblich zum Wohlbefinden der Benutzer bei.

Sicherheit: Nutzerdaten dürfen, abgesehen der allgemeinen Statistiken, auf keinen Fall an Dritte – aus welchem Zweck auch immer – weitergegeben werden. Auch darf kein Benutzer unerlaubten Zugriff auf Daten eines anderen Benutzers haben, ebenso sind Funktion, welche nicht für den Endanwender bestimmt sind, vor diesem zu schützen (e.g. Statistiken und Administratorfunktionen)

3.2. Technische Anforderungen

Der Benutzer benötigt ein Internetfähiges Gerät, mit beliebigem Betriebssystem sowie einen Html-fähigen Browser. Desweiteren benötigt der Benutzer eine Möglichkeit zur Eingabe (e.g. Tastatur & Maus) und Ausgabe (e.g. Monitor).

3.3. Realisierungsanforderungen

Es ist keine Installation notwendig, da die Anwendung auf einem Webserver läuft, mit dem per Browser interagiert wird.

Erforderliche Daten der Nutzer werden mittels ArrayList gespeichert. Zusätzlich werden anhand dieser Daten anonyme Statistiken erhoben.

Die Dokumentation erfolgt mittels JavaDoc Html Pages.

3.4. Diverses

Risiko: Datenverlust, durch schlechte Verarbeitung der Daten, sowie die generelle Stabilität des Webserver (Wobei die Teilnehmer der LVA hierauf keine Auswirkung haben, die Verantwortung liegt hierbei beim Projektauftraggeber).

Annahme: Nutzer weiß die grundsätzlichen Abläufe um auf einer Website navigieren zu können. Außerdem ist er der deutschen Sprache mächtig, Übersetzungen werden keine angeboten!