

Der grosse Dalmuti - Kartenspiel

Autoren: Stefan Waldenmaier, Aurelia Erhardt, Moritz Leimgrübler, Yves Notter, Daniel Herzog

Klasse: Wirtschaftsinformatik Vollzeit 2.51

Dozierende: Bradley Richards, Lukas Frey

Ort, Datum: Olten, 16.10.2013

FHNW Olten

[1 Projektbeschreibung 4](#_Toc368830135)

[1.1 Situation 4](#_Toc368830136)

[1.2 Angestrebte Lösung 4](#_Toc368830137)

[2 Wichtige Rahmenbedingungen 5](#_Toc368830138)

[3 Abgrenzung 5](#_Toc368830139)

[4 Anforderungen an die Software 6](#_Toc368830140)

[5 Use Cases 7](#_Toc368830141)

[5.1 Übersicht 7](#_Toc368830142)

[5.2 Use Case-Gruppen 8](#_Toc368830143)

[5.2.1 Use Case-Gruppe 100: Einträge verwalten 8](#_Toc368830144)

[5.2.1.1 Use Case 110: kreieren 9](#_Toc368830145)

[5.2.1.2 Use Case 120: ändern 10](#_Toc368830146)

[5.2.1.3 Use Case 130: löschen 11](#_Toc368830147)

[5.2.1.4 Use Case 140: Anhänge hinzufügen 12](#_Toc368830148)

[5.2.1.5 Use Case 150: Kommentare verfassen 13](#_Toc368830149)

[5.2.2 Use Case-Gruppe 200: Kommentare 14](#_Toc368830150)

[5.2.2.1 Use Case 210: Kommentar verfassen 15](#_Toc368830151)

[5.2.2.2 Use Case 211: Captcha-Kontrolle 16](#_Toc368830152)

[5.2.2.3 Use Case 212: Meldung an Sabine Krogemann 17](#_Toc368830153)

[5.2.2.4 Use Case 220: Bewilligung des Kommentars 18](#_Toc368830154)

[5.2.2.5 Use Case 221: Kommentar wird kreiert 19](#_Toc368830155)

[5.2.2.6 Use Case 222: Benachrichtigung an registrierten Kunden 20](#_Toc368830156)

[5.2.3 Use Case-Gruppe 300: Kundenverwaltung 21](#_Toc368830157)

[5.2.3.1 Use Case 310: Kunden sperren 22](#_Toc368830158)

[5.2.3.2 Use Case 311: Meldung an den Kunden 23](#_Toc368830159)

[5.2.3.3 Use Case 320: Kundenprofil ansehen 24](#_Toc368830160)

[5.2.4 Use Case-Gruppe 400: Anmeldung 25](#_Toc368830161)

[5.2.4.1 Use Case 410: Registrierung der Kundendaten 26](#_Toc368830162)

[5.2.4.2 Use Case 411: Captcha-Kontrolle 27](#_Toc368830163)

[5.2.4.3 Use Case 412: Login wird angelegt 28](#_Toc368830164)

[5.2.4.4 Use Case 420: Login 29](#_Toc368830165)

[5.2.4.5 Use Case 430: Passwort vergessen 30](#_Toc368830166)

[5.2.5 Use Case-Gruppe 500: Kundenkontoverwaltung 31](#_Toc368830167)

[5.2.5.1 Use Case 510: Profildaten ändern 32](#_Toc368830168)

[5.2.5.2 Use Case 520: Notifikation aktivieren / deaktivieren 33](#_Toc368830169)

[6 Systemarchitektur 34](#_Toc368830170)

[6.1 Analyse der nicht-funktionalen Anforderungen 34](#_Toc368830171)

[6.1.1 Kundenfreundliche Benutzeroberfläche 35](#_Toc368830172)

[6.1.2 Einfache Handhabung der Webauftritte durch Sabine Krogemann 35](#_Toc368830173)

[6.1.3 Sicherheit beim Kundenkontakt 35](#_Toc368830174)

[6.1.4 Moderne Technik und Design 35](#_Toc368830175)

[6.1.5 Rund um die Uhr verfügbar 35](#_Toc368830176)

[7 Aktivitätsdiagramme 36](#_Toc368830177)

[7.1 Use Case-Gruppe 100: Einträge verwalten 36](#_Toc368830178)

[7.1.1 Aktivitätsdiagramm zu Use Case 110: kreieren 36](#_Toc368830179)

[7.1.2 Aktivitätsdiagramm zu Use Case 120: ändern 37](#_Toc368830180)

[7.1.3 Aktivitätsdiagramm zu Use Case 130: löschen 38](#_Toc368830181)

[7.1.4 Aktivitätsdiagramm zu Use Case 140: Anhänge hinzufügen 39](#_Toc368830182)

[7.1.5 Aktivitätsdiagramm zu Use Case 150: Kommentare verfassen 40](#_Toc368830183)

[7.2 Use Case-Gruppe 200: Kommentare 41](#_Toc368830184)

[7.2.1 Aktivitätsdiagramm zu Use Case 210: Kommentare verfassen 41](#_Toc368830185)

[7.2.2 Aktivitätsdiagramm zu Use Case 211: Captcha-Kontrolle 42](#_Toc368830186)

[7.2.3 Aktivitätsdiagramm zu den Use Cases 212-222 43](#_Toc368830187)

[7.3 Use Case-Gruppe 300: Kundenverwaltung 44](#_Toc368830188)

[7.3.1 Aktivitätsdiagramm zu Use Case 310: Kunden sperren 44](#_Toc368830189)

[7.3.2 Aktivitätsdiagramm zu Use Case 320: Kundenprofil ansehen 45](#_Toc368830190)

[7.4 Use Case-Gruppe 400: Anmeldung 46](#_Toc368830191)

[7.4.1 Aktivitätsdiagramm zu Use Case 410: Registrierung der Kundendaten 46](#_Toc368830192)

[7.4.2 Aktivitätsdiagramm zu den Use Cases 411/412 47](#_Toc368830193)

[7.4.3 Aktivitätsdiagramm zu Use Case 430: Passwort vergessen 48](#_Toc368830194)

[7.5 Use Case-Gruppe 500: Kundenkontoverwaltung 49](#_Toc368830195)

[7.5.1 Aktivitätsdiagramm zu Use Case 510: Profildaten ändern 49](#_Toc368830196)

[7.5.2 Aktivitätsdiagramm zu Use Case 520: Notifikationen 50](#_Toc368830197)

# Projektbeschreibung

## Situation

Als Vorlage für das diesjährige Informatikprojekt dient das traditionelle Kartenspiel “Der grosse Dalmuti “. Dieses Spiel soll in einer Client-Server Anwendung zu einem Online Spiel umfunktioniert werden mit Hilfe von Java-Programmierung. Als Grundlage für dieses Software-Engineering Projekt dienen die Vorkenntnisse, die wir uns in den Modulen Java 1, Java 2, und Software-Engineering 1 angeeignet haben.

## Angestrebte Lösung

Ziel ist es, eine durchdachte und lauffähige Spielversion unter Einhaltung der Spielregeln zu gestalten und zu programmieren. Das Spiel soll eine begrenzte Anzahl von 4 Spielern zulassen welche jeweils einen Client ausführen und über einen Server koordiniert werden. Zuerst wird nach dem Motto “Keep it Simple“ vor allem darauf geachtet, dass die Basisanforderungen eingehalten werden und das Spiel läuft. Sollte man zum Schluss noch überschüssige Zeit haben, kann man noch Leistungs-und Begeisterungsanforderungen realisieren.

# Wichtige Rahmenbedingungen

**Technisch:** Java 7 als Programmiersprache. Keine J2EE oder andere schwergewichtige Frameworks. Als Mockup Designertool dient Balsamic und für die Gestaltung der UML-und anderer Diagramme wird Visio verwendet.

**Architektur:** Das Spiel basiert auf einer Client-Server Architektur und eine Datenbank ist für dieses Projekt nicht notwendig.

**Gestaltung:** Dem GUI sind keine Grenzen gesetzt, es sollte allerdings relativ selbsterklärend für die Spieler sein und ein ansprechendes Design haben und sinngemäss zum Spiel passen.

**Vorkenntnisse:** Das Projekt und insbesondere die Programmierung basiert auf den Vorkenntnissen der Module Java 1, Java 2, und Software-Engineering 1.

# Abgrenzung

Dieses Projekt befasst sich hauptsächlich mit der Erreichung einer lauffähigen Applikation und es sind keine Vorgaben gemäss GUI gesetzt. Die Spielregeln “Aufstand und Revolution“ und “Abrechnung und Verwaltung der Punkte“ müssen nicht implementiert werden, der Fantasie für zusätzliche Anforderungen sind aber keine Grenzen gesetzt. Die Dokumentation ist wichtig aber im Vordergrund steht das Ziel eine funktionierende Software zu entwickeln.

# Anforderungen an die Software

**Standard-/ Basisanforderungen**- Die Spielregeln werden komplett implementiert.  
- Das Spiel läuft einwandfrei als Client-Server Anwendung.  
- Die Installation und die Bedienung ist erklärt.  
- Das GUI ist übersichtlich, ansprechend und enthält die notwendigen Spielinfos.

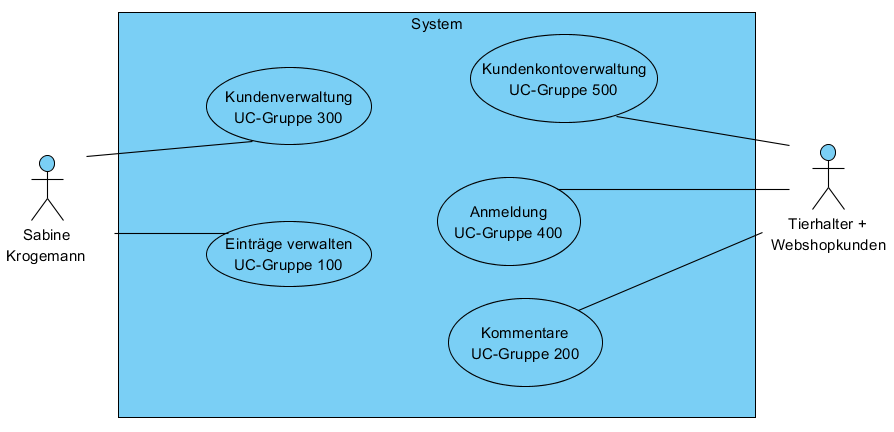
**Leistungsanforderungen**- Jeder Spieler hat einen Benutzernamen mit dem er sich an-und abmelden kann.  
- Die Ränge der Spieler werden automatisch vergeben.  
- Es können mehrere Runden nacheinander gespielt werden.  
- Der Server startet das Spiel automatisch, wenn genügend Spieler vorhanden sind.

**Begeisterungsanforderungen**- Flexible Anzahl Spieler.  
- Lobby und mehrere Spielräume.  
- Chat Funktion.

# Use Cases

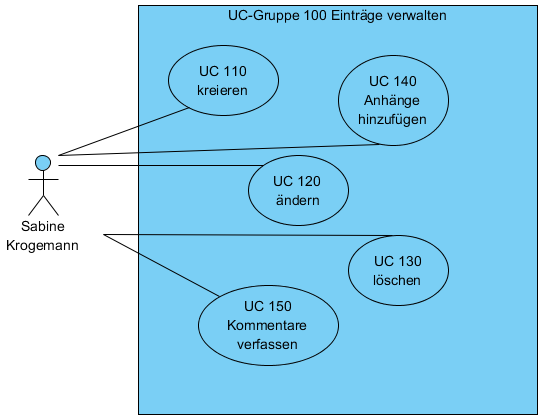
## Übersicht

Diese Grafik stellt die Use Case-Gruppen des zu implementierenden Blogs dar.



## Use Case-Gruppen

### Use Case-Gruppe 100: Einträge verwalten



**Use Case Gruppe 100: Einträge verwalten**

|  |
| --- |
| Beschreibung: Das System zeigt die Verwaltungsmöglichkeiten von Sabine Krogemann |
| Beteiligte: Sabine Krogemann |
| Ablauf: Sabine Krogemann kann Einträge kreieren, ändern oder löschen und Kommentare verfassen. Des Weiteren kann sie Anhänge hinzufügen. |
| Beinhaltende Use Cases: UC110: kreieren  UC120: ändern  UC130: löschen  UC140: Anhänge hinzufügen  UC150: Kommentare verfassen |

#### Use Case 110: kreieren

|  |
| --- |
| Use Case: kreieren Use Case Nummer: 110 Ziel: Sabine Krogemann kann Einträge kreieren Hauptakteur: Sabine Krogemann Nebenakteure: keine Vorbedingung: keine  Auslöser: Sabine Krogemann hat eine Idee für einen Eintrag und möchte diesen im Blog veröffentlichen.  Eingabedaten: Text Ausgabedaten: Texteintrag  Nachbedingung: Eintrag wird freigeschaltet Garantie falls erfolgreich: Eintrag wurde hochgeladen Garantie falls Fehlschlag: Eintrag wurde nicht hochgeladen |

**Hauptszenario:**   
  
1. Benutzer hat Ideen für einen neuen Eintrag  
2. Benutzer meldet sich im Onlineblog an  
3. System zeigt Benutzeroberfläche an  
4. Benutzer wählt „neuen Eintrag kreieren“  
5. System zeigt leeren Eintrag, welcher direkt bearbeitet werden kann  
6. Benutzer verfasst Text  
7. Benutzer bestätigt das Erstellen des Eintrags  
8. System prüft, ob der Eintrag korrekt verfasst wurde  
9. System zeigt Vorschau  
10. Benutzer bestätigt Vorschau  
11. System prüft, ob der Eintrag hochgeladen werden kann  
12. System lädt den Eintrag hoch  
13. System zeigt den Eintrag im Blog an  
14. Szenario Ende

**Erweiterungen:**

8. System prüft, ob der Eintrag korrekt verfasst wurde  
 1. System erkennt keine Fehler im Eintrag  
 2. Fortfahren mit „Punkt 9 System zeigt Vorschau"  
  
 1. System zeigt eine Fehlermeldung, dass der Eintrag Fehler beinhaltet wie z.B. zu viel Text oder fehlender Titel  
 2. Benutzer prüft Fehlermeldung und verbessert den Eintrag  
 3. Fortfahren mit „Punkt 8 System prüft, ob der Eintrag korrekt verfasst wurde“

11. System prüft, ob der Eintrag hochgeladen werden kann  
 1. System erkennt keine Fehler im Eintrag  
 2. Fortfahren mit „Punkt 12 System lädt den Eintrag hoch“  
  
 1. System zeigt eine Fehlermeldung, dass das Hochladen fehlschlug, weil z.B. die angehängten Dateien zu gross sind oder ein inkompatibles Format haben  
 2. Benutzer prüft Fehlermeldung und verbessert den Fehler  
 3. Fortfahren mit „Punkt 11 System prüft, ob der Eintrag hochgeladen werden kann“

#### Use Case 120: ändern

|  |
| --- |
| Use Case: ändern Use Case-Nummer: 120 Ziel: Sabine Krogemann kann Einträge ändern Hauptakteur: Sabine Krogemann Nebenakteure: keine Vorbedingung: Einträge sind vorhanden  Auslöser: Sabine Krogemann findet es notwendig einen Eintrag zu ändern  Eingabedaten: Text hinzufügen oder löschen Ausgabedaten: Eintrag mit Text Garantie falls erfolgreich: Eintrag wurde verändert Garantie falls Fehlschlag: Eintrag wurde nicht verändert |

**Hauptszenario:**  
  
1. Benutzer findet es für notwendig einen Eintrag zu ändern  
2. Benutzer meldet sich im Onlineblog an  
3. System zeigt Benutzeroberfläche an  
4. Benutzer wählt den Eintrag, den er ändern möchte  
5. System zeigt Bearbeitungsmaske des gewählten Eintrages  
6. Benutzer ändert den Eintrag  
7. Benutzer bestätigt Änderung des Eintrags  
8. System prüft, ob der Eintrag korrekt verfasst wurde  
9. System zeigt Vorschau  
10. Benutzer bestätigt Vorschau  
11. System aktualisiert den Eintrag  
12. System zeigt den Eintrag im Blog an  
13. Szenario Ende

**Erweiterungen:**

8. System prüft, ob der Eintrag korrekt verfasst wurde  
 1. System erkennt keine Fehler im Eintrag  
 2. Fortfahren mit „Punkt 9 System zeigt Vorschau"  
  
 1. System zeigt eine Fehlermeldung, dass der Eintrag Fehler beinhaltet wie z.B. zu viel Text oder fehlender Titel  
 2. Benutzer prüft Fehlermeldung und verbessert den Eintrag  
 3. Fortfahren mit „Punkt 8 System prüft, ob der Eintrag korrekt verfasst wurde“

#### Use Case 130: löschen

|  |
| --- |
| Use Case: löschen Use Case-Nummer: 130 Ziel: Sabine Krogemann kann Einträge löschen Hauptakteur: Sabine Krogemann Nebenakteure: keine Vorbedingung: Einträge sind vorhanden Auslöser: Sabine Krogemann findet es notwendig einen Einträg zu löschen  Eingabedaten: keine  Ausgabedaten: keine  Nachbedingung: zu löschender Eintrag ist nicht mehr vorhanden Garantie falls erfolgreich: Eintrag wurde gelöscht Garantie falls Fehlschlag: Eintrag wurde nicht gelöscht |

**Hauptszenario:**  
  
1. Benutzer findet es notwendig einen Eintrag zu löschen  
2. Benutzer meldet sich im Onlineblog an  
3. System zeigt Benutzeroberfläche an  
4. Benutzer wählt den Eintrag, den er löschen möchte  
5. System zeigt Bearbeitungsmaske des gewählten Eintrags an  
6. Benutzer wählt Funktion ‚löschen‘  
7. System zeigt Bestätigungsmeldung (Wollen Sie wirklich den gewählten Eintrag löschen: Ja/nein)   
8. Benutzer bestätigt  
9. System bestätigt  
10. Szenario Ende

**Erweiterungen:**  
8. Benutzer bestätigt  
 1. Benutzer bestätigt mit Nein  
 2. System zeigt Bearbeitungsmaske des gewählten Eintrags  
 3. Fortfahren mit „Punkt 9 System bestätigt“

1. Benutzer bestätigt mit Ja  
 2. System löscht den Eintrag  
 3. Fortfahren mit „Punkt 9 System bestätigt“

#### Use Case 140: Anhänge hinzufügen

|  |
| --- |
| Use Case: Anhänge hinzufügen Use Case-Nummer: 140 Ziel: Sabine Krogemann kann Medienanhänge oder Dokumente anhängen Hauptakteur: Sabine Krogemann Nebenakteure: keine Vorbedingung: keine Auslöser: Sabine Krogemann möchte einem Eintrag eine Datei anhängen  Eingabedaten: Anhang  Ausgabedaten: Eintrag mit Anhang  Nachbedingung: Eintrag mit Anhang wird freigeschaltet Garantie falls erfolgreich: Anhang ist im Eintrag abrufbar Garantie falls Fehlschlag: Anhang ist im Eintrag nicht abrufbar |

**Hauptszenario:**  
  
1. Benutzer möchte einem Eintrag eine Datei anhängen  
2. Benutzer meldet sich im Onlineblog an  
3. System zeigt Benutzeroberfläche  
4. Benutzer wählt den Eintrag, welchem er eine Datei anhängen möchte  
5. System zeigt gewählten Eintrag  
6. Benutzer wählt Datei und bestätigt Hochladen  
7. System prüft, ob der Anhang hochgeladen werden kann  
8. System lädt den Anhang hoch  
9. System prüft, ob der Eintrag korrekt verfasst wurde  
10.. System zeigt Vorschau  
11. Benutzer bestätigt Vorschau  
12. System aktualisiert den Eintrag  
13. System zeigt den Eintrag im Blog an  
14. Szenario Ende  
  
**Erweiterungen:**  
  
7. System prüft, ob der Anhang hochgeladen werden kann  
 1. System erkennt keine Fehler im Anhang  
 2. Fortfahren mit „Punkt 8 System lädt den Anhang hoch“  
  
 1. System zeigt eine Fehlermeldung, dass das Hochladen fehlschlug, weil z.B. Dateien zu gross sind oder ein nicht kompatibles Format haben  
 2. Benutzer prüft Fehlermeldung und verbessert den Fehler  
 3. Fortfahren mit „Punkt 7 System prüft, ob der Anhang hochgeladen werden kann“

9. System prüft, ob der Eintrag korrekt verfasst wurde  
 1. System erkennt keine Fehler im Eintrag  
 2. Fortfahren mit „Punkt 10 System zeigt Vorschau“  
  
 1. System zeigt eine Fehlermeldung, dass der Eintrag Fehler beinhaltet wie z.B. zu viel Text oder fehlender Titel  
 2. Benutzer prüft Fehlermeldung und verbessert den Eintrag  
 3. Fortfahren mit „Punkt 9 System prüft, ob der Eintrag korrekt verfasst wurde“

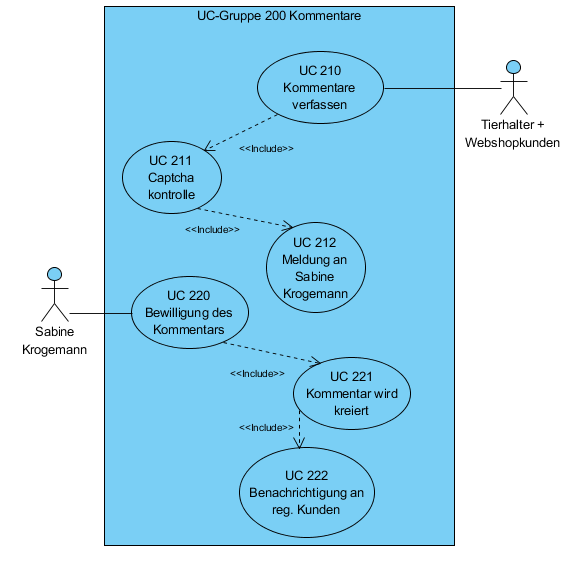
#### Use Case 150: Kommentare verfassen

|  |
| --- |
| Use Case: Kommentare verfassen Use Case-Nummer: 150 Ziel: Sabine Krogemann kann einen Kommentar zu einem Eintrag verfassen Hauptakteur: Sabine Krogemann Nebenakteure: keine Vorbedingung: Ein zu kommentierender Eintrag muss vorhanden sein Auslöser: Sabine Krogemann möchte einen Kommentar zu einem Eintrag verfassen  Eingabedaten: Text  Ausgabedaten: Eintrag mit Textkommentar  Nachbedingung: Textkommentar zu Eintrag wird freigeschaltet Garantie falls erfolgreich: Kommentar wird angezeigt Garantie falls Fehlschlag: Kommentar wird nicht angezeigt |

**Hauptszenario:**  
  
1. Benutzer möchte einen Kommentar zu einem Eintrag verfassen  
2. Benutzer meldet sich im Onlineblog an  
3. System zeigt Benutzeroberfläche an  
4. Benutzer wählt den zu kommentierenden Eintrag  
5. System zeigt Kommentarfeld an  
6. Benutzer verfasst Kommentar  
7. Benutzer bestätigt das Verfassen des Kommentars  
8. System prüft, ob der Kommentar korrekt verfasst wurde  
9. System zeigt Vorschau  
10. Benutzer bestätigt Vorschau  
11. System aktualisiert den Eintrag  
12. System zeigt den Kommentar im Eintrag an  
13. Szenario Ende  
  
**Erweiterungen:**

8. System prüft, ob der Kommentar korrekt verfasst wurde  
 1. System entdeckt keine Fehler im Kommentar  
 2. Fortfahren mit „Punkt 9 System zeigt Vorschau“  
  
 1. System zeigt eine Fehlermeldung, dass der Kommentar Fehler beinhaltet wie z.B. zu viel Text  
 2. Benutzer prüft Fehlermeldung und verbessert den Kommentar  
 3. Fortfahren mit „Punkt 8 System prüft, ob der Kommentar korrekt verfasst wurde“

### Use Case-Gruppe 200: Kommentare



**Use Case Gruppe 200: Kommentare**

|  |
| --- |
| Beschreibung: Das System zeigt zwei Funktionen. Einerseits das Verfassen der Kommentar aus der Sicht der Kunden und andererseits das Bewilligen der gemeldeten Kommentare aus der Sicht von Sabine Krogemann. |
| Beteiligte: Sabine Krogemann, Kunden |
| Ablauf: Die Kunden können Kommentare verfassen unter Eingabe eines Captchas, was eine Meldung an Sabine Krogemann zur Folge hat. Sabine Krogemann kann Kommentare bewilligen oder nicht. Bei Bewilligung wird der Kommentar erstellt und eine kurze Bestätigung an den registrierten Benutzer gesendet. |
| Beinhaltende Use Cases: UC210: Kommentar verfassen  UC 211: Captcha-Kontrolle  UC 212: Meldung an Sabine Krogemann  UC220: Bewilligung des Kommentars  UC221: Kommentar wird erstellt  UC222: Benachrichtigung an registrierten Benutzer |

#### Use Case 210: Kommentar verfassen

|  |
| --- |
| Use Case: Kommentar verfassen Use Case-Nummer: 210 Ziel: Die Kunden können einen Kommentar verfassen Hauptakteur: Kunde Nebenakteure: Sabine Krogemann Vorbedingung: Ein zu kommentierender Eintrag muss vorhanden sein Auslöser: Die Kunden möchten einen Kommentar verfassen  Eingabedaten: Text  Ausgabedaten: Eintrag mit Textkommentar  Nachbedingung: Meldung an Sabine Krogemann wird generiert Garantie falls erfolgreich: Meldung an Sabine Krogemann Garantie falls Fehlschlag: Fehlermeldung |

**Hauptszenario:**  
  
1. Benutzer möchte einen Kommentar zu einem Eintrag verfassen  
2. Benutzer meldet sich im Onlineblog an  
3. System zeigt Benutzeroberfläche an  
4. Benutzer wählt den zu kommentierenden Eintrag  
5. System zeigt Kommentarfeld an  
6. Benutzer verfasst Kommentar  
7. Benutzer bestätigt das Verfassen des Kommentars  
8. System zeigt Vorschau  
9. Benutzer bestätigt Vorschau  
10. System meldet, dass der Kommentar noch bewilligt werden muss  
11. Szenario Ende  
  
**Erweiterungen:**  
  
7. Benutzer bestätigt das Verfassen des Kommentars  
 1. aus Use Case 211: Captcha-Kontrolle, siehe Kapitel 7.2.2.2  
 2. Fortfahren mit „Punkt 11 Szenario Ende“  
  
10. System meldet, dass der Kommentar noch bewilligt werden muss  
 1. aus Use Case 212: Meldung an Sabine Krogemann, siehe Kapitel 7.2.2.3  
 2. Fortfahren mit „Punkt 11 Szenario Ende“

#### Use Case 211: Captcha-Kontrolle

|  |
| --- |
| Use Case: Captcha-Kontrolle Use Case-Nummer: 211 Ziel: Zulassung von Kommentaren von ausschliesslich Menschen Hauptakteur: System Nebenakteure: Kunden Vorbedingung: Ein Kommentar muss verfasst werden Auslöser: Ein Kunde verfasst einen Kommentar  Eingabedaten: Text gemäss Captcha  Ausgabedaten: keine  Nachbedingung: Meldung an Sabine Krogemann wird generiert Garantie falls erfolgreich: Meldung an Sabine Krogemann Garantie falls Fehlschlag: Fehlermeldung von System an Benutzer |

**Hauptszenario:**  
  
1. System zeigt dem Benutzer ein Fenster für Captcha-Eingabe  
2. Benutzer gibt Captcha ein  
3. System prüft Captcha  
4. System sendet Meldung an Sabine Krogemann  
5. Szenario Ende  
  
**Erweiterungen:**  
  
3. System prüft Captcha  
 1. System erkennt, dass die Eingabe mit dem Captcha übereinstimmt  
 2. Fortfahren mit „Punkt 4 System sendet Meldung an Sabine Krogemann“  
  
 1. System erkennt, dass die Eingabe nicht mit dem Captcha übereinstimmt  
 2. System zeigt dem Benutzer eine Fehlermeldung   
 3. Fortfahren mit „Punkt 1 System zeigt dem Benutzer ein Fenster für Captcha- Eingabe“

#### Use Case 212: Meldung an Sabine Krogemann

|  |
| --- |
| Use Case: Meldung an Sabine Krogemann Use Case-Nummer: 212 Ziel: Sabine Krogemann kann unerwünschte Kommentare blockieren Hauptakteur: System Nebenakteure: Kunden, Sabine Krogemann Vorbedingung: Captcha-Kontrolle muss erfolgreich sein Auslöser: Erfolgreiche Captcha-Kontrolle  Eingabedaten: keine  Ausgabedaten: keine  Nachbedingung: Sabine Krogemann erhält generierte Meldung Garantie falls erfolgreich: Sabine Krogemann erhält eine Meldung Garantie falls Fehlschlag: Sabine Krogemann erhält keine Meldung |

**Hauptszenario:**  
1. System erhält Bestätigung, dass Captcha erfolgreich war  
2. System generiert Meldung von Kommentarabgabe  
3. System sendet Meldung an Sabine Krogemann  
4. Benutzer beantwortet die Meldung  
5. Szenario Ende  
  
**Erweiterungen:**  
  
4. Benutzer beantwortet die Meldung  
 1. System lässt den Kommentar zu  
 2. System fügt den Kommentar zum Eintrag hinzu  
 3. Fortfahren mit „Punkt 5 Szenario Ende“

1. System lässt den Kommentar nicht zu  
 2. System sendet Meldung an registrierten Kunden  
 3. Fortfahren mit „Punkt 5 Szenario Ende“

#### Use Case 220: Bewilligung des Kommentars

|  |
| --- |
| Use Case: Bewilligung des Kommentars Use Case-Nummer: 220 Ziel: Sabine Krogemann kann unerwünschte Kommentare blockieren Hauptakteur: Sabine Krogemann Nebenakteure: Kunden Vorbedingung: Ein Kommentar muss verfasst sein Auslöser: Ein Kommentar wurde eingetragen und Captcha war erfolgreich  Eingabedaten: keine  Ausgabedaten: keine  Nachbedingung: Textkommentar wird zum Freischalten an das System übergeben Garantie falls erfolgreich: Kommentar wird zum Eintrag hinzugefügt Garantie falls Fehlschlag: Kommentar wird nicht zum Eintrag hinzugefügt |

**Hauptszenario:**  
  
1. System zeigt den Kommentar an  
2. Benutzer liest den Kommentar  
3. Szenario Ende  
  
**Erweiterungen:**  
  
2. Benutzer liest den Kommentar  
 1. Benutzer bewilligt den Kommentar  
 2. aus Use Case 221: Kommentar wird kreiert, siehe Kapitel 7.2.2.5  
 3. Fortfahren mit „Punkt 3 Szenario Ende“  
  
 1. Benutzer bewilligt den Kommentar nicht  
 2. Fortfahren mit „Punkt 3 Szenario Ende“

#### Use Case 221: Kommentar wird kreiert

|  |
| --- |
| Use Case: Kommentar wird kreiert Use Case-Nummer: 221 Ziel: Der Kommentar wird zum Eintrag hinzugefügt Hauptakteur: System Nebenakteure: keine Vorbedingung: Sabine Krogemann muss einen Kommentar bewilligen Auslöser: Sabine Krogemann hat einen Kommentar bewilligt  Eingabedaten: Bestätigung von Sabine Krogemann  Ausgabedaten: keine  Nachbedingung: Textkommentar zu Eintrag wird freigeschaltet Garantie falls erfolgreich: Kommentar wird zum Eintrag hinzugefügt Garantie falls Fehlschlag: Kommentar wird nicht zum Eintrag hinzugefügt |

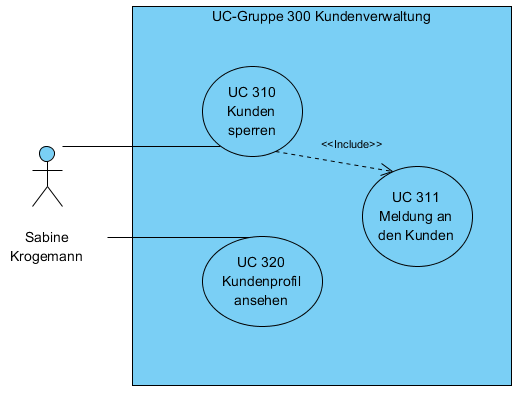
**Hauptszenario:**  
  
1. System erhält Bestätigung, dass ein Kommentar von Sabine Krogemann bewillgt wurde  
2. System fügt den Kommentar zum Eintrag hinzu  
3. Szenario Ende  
  
**Erweiterungen:**  
  
2. System fügt den Kommentar zum Eintrag hinzu  
 1. aus Use Case 222: Benachrichtigung an registrierten Kunden, siehe Kapitel 7.2.2.6  
 2. Fortfahren mit „Punkt 3 Szenario Ende“

#### Use Case 222: Benachrichtigung an registrierten Kunden

|  |
| --- |
| Use Case: Benachrichtigung an registrierten Benutzer Use Case-Nummer: 222 Ziel: Registrierter Kunde wird darüber informiert, dass sein Kommentar zugelassen wurde Hauptakteur: System Nebenakteure: Keine Vorbedingung: Kommentar wurde zu einem Eintrag hinzugefügt Auslöser: Kommentar wurde zu einem Eintrag hinzugefügt  Eingabedaten: keine  Ausgabedaten: Textmeldung  Nachbedingung: Textmeldung wird an registrierten Kunden gesendet Garantie falls erfolgreich: Registrierter Kunde wird darüber informiert, dass sein Kommentar zugelassen wurde Garantie falls Fehlschlag: Registrierter Kunde wird nicht darüber informiert, dass sein Kommentar zugelassen wurde |

**Hauptszenario:**  
  
1. System sendet dem registrierten Kunden eine Benachrichtigung, dass sein Kommentar zugelassen wurde  
2. Szenario Ende

### Use Case-Gruppe 300: Kundenverwaltung



**Use Case Gruppe 300: Kundenverwaltung**

|  |
| --- |
| Beschreibung: Dieses System zeigt, wie Sabine Krogemann in der Kundenverwaltung Profile ansehen und Kunden sperren kann, welche eine entsprechende Meldung erhalten. |
| Beteiligte: Sabine Krogemann |
| Ablauf: Sabine Krogemann sieht alle Details der Kundenprofile. Sie kann auch Kunden sperren und diese erhalten eine Meldung. |
| Beinhaltende Use Cases: UC310: Kunden sperren  UC311: Meldung an den Kunden  UC320: Kundenprofil ansehen |

#### Use Case 310: Kunden sperren

|  |
| --- |
| Use Case: Kunden sperren Use Case-Nummer: 310 Ziel: Sabine Krogemann kann unerwünschte Kunden sperren Hauptakteur: Sabine Krogemann Nebenakteure: keine Vorbedingung: Kunde fällt negativ auf Auslöser: Sabine möchte einen Kunden nicht mehr auf ihrem Blog haben  Eingabedaten: keine  Ausgabedaten: keine  Nachbedingung: Kunde wird für das Verfassen von Kommentaren gesperrt Garantie falls erfolgreich: Kunde ist gesperrt und kann somit keine Kommentare mehr verfassen Garantie falls Fehlschlag: Kunde ist nicht gesperrt und kann weiterhin Einträge kommentieren |

**Hauptszenario:**  
  
1. Benutzer entdeckt einen Kunden, der negativ auffält  
2. Benutzer meldet sich im Onlineblog an  
3. System zeigt Benutzeroberfläche an  
4. Benutzer wählt Kundenverwaltung  
5. System zeigt Kundenverwaltung an  
6. Benutzer wählt den zu sperrenden Kunden  
7. System zeigt zu sperrenden Kunden  
8. Benutzer wählt „Profil sperren“  
9. System fragt nach, ob das Profil wirklich gesperrt werden soll  
10. System sperrt Kunden  
11. Szenario Ende  
  
**Erweiterungen:**  
  
9. System fragt nach, ob das Profil wirklich gesperrt werden soll  
 1. Benutzer wählt ‚Ja‘  
 2. Fortfahren mit „Punkt 10 System sperrt Kunden“

1. Benutzer wählt ‚Nein‘  
 2. Fortfahren mit „Punkt 7 System zeigt zu sperrenden Kunden“

10. System sperrt Kunden  
 1. aus Use Case 311: Meldung an den Kunden, siehe Kapitel 7.2.3.2  
 2. Fortfahren mit „Punkt 11 Szenario Ende“

#### Use Case 311: Meldung an den Kunden

|  |
| --- |
| Use Case: Meldung an den Kunden Use Case-Nummer: 311 Ziel: Kunde erhält Meldung über seine Profilsperre Hauptakteur: System Nebenakteure: keine Vorbedingung: Sabine Krogemann hat einen Kunden gesperrt Auslöser: Sperrung eines Kundenkontos  Eingabedaten: keine  Ausgabedaten: Textmeldung  Nachbedingung: Kunde erhält Textmeldung Garantie falls erfolgreich: Kunde erhält Meldung über seine Profilsperre Garantie falls Fehlschlag: Kunde erhält keine Meldung über seine Profilsperre |

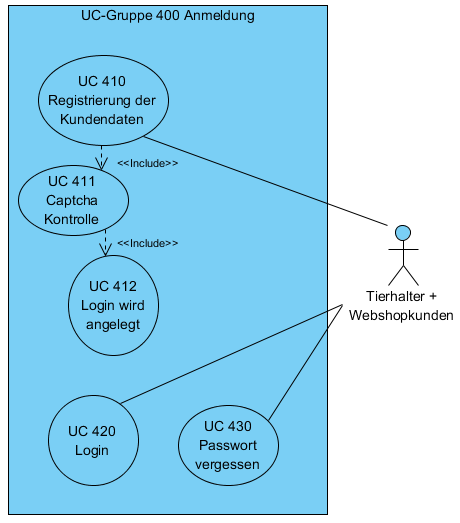
**Hauptszenario:**  
  
1. System erhält Bestätigung, dass Sabine Krogemann einen Kunden gesperrt hat  
2. System sendet dem Kunden eine Meldung über seine Profilsperre  
3. Szenario Ende

#### Use Case 320: Kundenprofil ansehen

|  |
| --- |
| Use Case: Kundenprofil ansehen Use Case-Nummer: 320 Ziel: Sabine Krogemann kann die Profile ihrer Kunden einsehen Hauptakteur: Sabine Krogemann Nebenakteure: keine Vorbedingung: Kunden müssen Profile erstellt haben Auslöser: Sabine Krogemann möchte ein Profil einsehen  Eingabedaten: keine  Ausgabedaten: keine  Nachbedingung: Kundenprofil wird von System angezeigt Garantie falls erfolgreich: System zeigt Übersicht über die Profile Garantie falls Fehlschlag: System kann nichts anzeigen |

**Hauptszenario:**  
  
1. Benutzer möchte ein Profil einsehen  
2. Benutzer meldet sich im Onlineblog an  
3. System zeigt Benutzeroberfläche an  
4. Benutzer wählt das anzuschauen Profil  
5. System zeigt das Profil an  
6. Szenario Ende

### Use Case-Gruppe 400: Anmeldung



**Use Case Gruppe 400: Kommentare**

|  |
| --- |
| Beschreibung: Dieses System zeigt, wie sich die Kunden im Blog registrieren und einloggen können. |
| Beteiligte: Kunden |
| Ablauf: Die Kunden registrieren sich im Blog mittels Captcha-Kontrolle oder sie haben bereits einen Login und können sich einloggen. |
| Beinhaltende Use Cases: UC 410: Registrierung der Kundendaten  UC 411: Captcha-Kontrolle  UC 412: Login wird angelegt  UC 420: Login |

#### Use Case 410: Registrierung der Kundendaten

|  |
| --- |
| Use Case: Registrierung der Kundendaten Use Case-Nummer: 410 Ziel: Der Kunde kann sich für den Blog registrieren Hauptakteur: Kunde Nebenakteure: keine Vorbedingung: Registrierungsmöglichkeit muss bestehen Auslöser: Kunde möchte sich für den Blog registrieren, um z.B. Kommentare zu Verfassen oder Fragen per Kommentar zu stellen.  Eingabedaten: Registrationsdaten in Text  Ausgabedaten: keine  Nachbedingung: Registrationsdaten werden an Captcha weitergegeben Garantie falls erfolgreich: Kunde hat einen Login Garantie falls Fehlschlag: Kunde kann sich nicht einloggen |

**Hauptszenario:**  
  
1. Benutzer möchte sich für den Blog registrieren  
2. Benutzer öffnet den Onlineblog  
3. System zeigt den Onlineblog an  
4. Benutzer wählt ‚Jetzt registrieren‘  
5. System zeigt Registrationsformular  
6. Benutzer gibt seine Daten ein und wählt registrieren  
7. System prüft die eingegeben Daten  
8. System legt das Kundenprofil an  
9. Szenario Ende  
  
**Erweiterungen:**  
  
6. Benutzer gibt seine Daten ein und wählt registrieren  
 1. aus Use Case 411: Captcha-Kontrolle, siehe Kapitel 7.2.4.2  
 2. Fortfahren mit „Punkt 7 System prüft die eingegebenen Daten“  
  
7. System prüft die eingegebenen Daten  
 1. System prüft, ob alle Daten angegeben wurden  
 2. Fortfahren mit „Punkt 8 System legt das Kundenprofil an“  
  
8. System legt das Kundenprofil an  
 1. aus Use Case 412: Login wird angelegt, siehe Kapitel 7.2.4.3  
 2. Fortfahren mit „Punkt 9 Szenario Ende“

#### Use Case 411: Captcha-Kontrolle

|  |
| --- |
| Use Case: Captcha-Kontrolle Use Case-Nummer: 411 Ziel: Ausschliesslich Menschen sollen sich für den Blog registrieren können Hauptakteur: System Nebenakteure: Kunden Vorbedingung: Ein Registration muss vorgenommen werden Auslöser: Ein Kunde füllt das Registrationsformular aus  Eingabedaten: Text gemäss Captcha  Ausgabedaten: keine  Nachbedingung: Registrationsdaten werden an System weitergegeben Garantie falls erfolgreich: Login wird angelegt Garantie falls Fehlschlag: Login wird nicht angelegt |

**Hauptszenario:**  
  
1. System zeigt dem Benutzer ein Fenster für Captcha-Eingabe  
2. Benutzer gibt Captcha ein  
3. System prüft Captcha  
4. System legt Login an  
5. Szenario Ende  
  
**Erweiterungen:**  
  
3. System prüft Captcha  
 1. System erkennt, dass die Eingabe mit dem Captcha übereinstimmt  
 2. Fortfahren mit „Punkt 4 System legt Login an“  
  
 1. System erkennt, dass die Eingabe nicht mit dem Captcha übereinstimmt  
 2. System zeigt dem Benutzer eine Fehlermeldung  
 3. Fortfahren mit „Punkt 1 System zeigt dem Kunden ein Fenster für Captcha- Eingabe“

#### Use Case 412: Login wird angelegt

|  |
| --- |
| Use Case: Login wird angelegt Use Case-Nummer: 412 Ziel: Die Kunden können sich im Blog einloggen Hauptakteur: System Nebenakteure: keine Vorbedingung: Captcha-Kontrolle muss durchgeführt werden Auslöser: Captcha-Kontrolle war erfolgreich  Eingabedaten: keine  Ausgabedaten: keine  Nachbedingung: Kundenprofil wird freigeschaltet Garantie falls erfolgreich: Kunde kann sich einloggen Garantie falls Fehlschlag: Kunde kann sich nicht einloggen |

**Hauptszenario:**  
  
1. System erhält Bestätigung, dass die Captcha-Kontrolle erfolgreich war  
2. System erstellt ein Kundenprofil  
3. System schaltet das Profil auf  
4. System zeigt Fenster „Danke für Ihre Registrierung“  
5. Szenario Ende

#### Use Case 420: Login

|  |
| --- |
| Use Case: Login Use Case-Nummer: 420 Ziel: Der Kunde kann Kommentare verfassen und sein Kundenprofil einsehen und anpassen Hauptakteur: Kunde Nebenakteure: keine Vorbedingung: Kunde hat sich bereits registriert Auslöser: Kunde möchte sich einloggen  Eingabedaten: Logindaten  Ausgabedaten: keine  Nachbedingung: Kunde wird eingeloggt Garantie falls erfolgreich: Kunde hat sich erfolgreich eingeloggt Garantie falls Fehlschlag: Kunde konnte sich nicht einloggen |

**Hauptszenario:**  
  
1. Benutzer möchte sich einloggen  
2. Benutzer öffnet den Onlineblog  
3. System zeigt den Onlineblog an  
4. Benutzer wählt ‚zum Login‘  
5. System zeigt Loginmaske an  
6. Benutzer gibt Benutzername und Passwort ein und wählt ‚Login‘  
7. System prüft die Eingabe  
8. System lässt den Benutzer zu  
9. Szenario Ende  
  
**Erweiterungen:**  
  
7. System prüft die Eingabe  
 1. System erkennt, dass die Eingabe korrekt ist  
 2. Fortfahren mit „Punkt 8 System lässt den Benutzer zu“  
  
 1. System erkennt, dass die Eingabe nicht korrekt ist  
 2. System zeigt Fehlermeldung, dass die Daten nicht korrekt eingegeben wurden  
 3. Fortfahren mit „Punkt 5 System zeigt Loginmaske an“

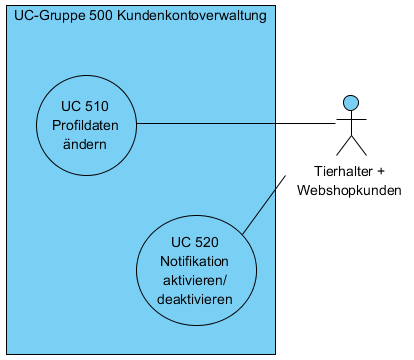
#### Use Case 430: Passwort vergessen

|  |
| --- |
| Use Case: Passwort vergessen Use Case-Nummer: 430 Ziel: Kunde kann sein Passwort einholen Hauptakteur: System Nebenakteure: Kunden Vorbedingung: Kunde hat sein Passwort nicht zur Hand Auslöser: Kunde wählt ‚Passwort vergessen‘  Eingabedaten: Emailadresse in Text  Ausgabedaten: keine  Nachbedingung: System sendet Logindaten Garantie falls erfolgreich: Kunde hat sein Passwort wieder Garantie falls Fehlschlag: Kunde kann sich nicht mehr einloggen |

**Hauptszenario:**  
  
1. Benutzer hat sein Passwort vergessen  
2. Benutzer öffnet den Onlineblog  
3. System zeigt den Onlineblog an  
4. Benutzer wählt ‚zum Login‘  
5. System zeigt Loginmaske an  
6. Benutzer wählt ‚Passwort vergessen‘  
7. System fragt nach Emailadresse  
8. Benutzer gibt Emailadresse ein  
9. System prüft Emailadresse  
10. System sendet das Passwort an die eingegebene Emailadresse  
11. Szenario Ende  
  
**Erweiterungen**  
  
9. System prüft Emailadresse  
 1. System findet Logindaten  
 2. Fortfahren mit „Punkt 10 System sendet das Passwort an die eingegebene Emailadresse“

1. System findet Logindaten nicht  
 2. System zeigt Fehlermeldung ‚Emailadresse wurde nicht gefunden“  
 3. Fortfahren mit „Punkt 7 System fragt nach Emailadresse“

### Use Case-Gruppe 500: Kundenkontoverwaltung



**Use Case Gruppe 500: Kommentare**

|  |
| --- |
| Beschreibung: Das System zeigt die Verwaltung des Kundenkontos durch den Kunden |
| Beteiligte: Kunden |
| Ablauf: Der Kunde kann seine Profildaten ändern und die Notifikation aktivieren oder deaktivieren |
| Beinhaltende Use Cases: UC510: Profildaten ändern  UC520: Notifikation aktivieren / deaktivieren |

#### Use Case 510: Profildaten ändern

|  |
| --- |
| Use Case: Profildaten ändern Use Case-Nummer: 510 Ziel: Der Kunde kann seine Profildaten ändern Hauptakteur: Kunde Nebenakteure: keine Vorbedingung: Der Kunde ist bereits registriert Auslöser: Der Kunde möchte seine Profildaten ändern  Eingabedaten: neue Profildaten in Text  Ausgabedaten: keine  Nachbedingung: neue Profildaten werden freigeschaltet Garantie falls erfolgreich: Profildaten wurden geändert Garantie falls Fehlschlag: Profildaten wurden nicht geändert |

**Hauptszenario:**  
  
1. Benutzer möchte seine Profildaten ändern  
2. Benutzer meldet sich im Onlineblog an  
3. System zeigt Benutzeroberfläche an  
4. Benutzer wählt ‚Profildaten ändern‘  
5. System zeigt Profildatenmaske an  
6. Benutzer ändert seine Daten  
7. Benutzer wählt ‚bestätigen‘  
8. System überprüft die Daten  
9. System ändert die Daten  
10. Szenario Ende  
  
**Erweiterungen:**  
  
8. System überprüft die Daten  
 1. System erkennt, dass alle Daten korrekt eingegeben wurden  
 2. Fortfahren mit „Punkt 9 System ändert die Daten“  
  
 1. System erkennt, dass die eingegebenen Daten Fehler beinhalten, wie z.B. zu viel Text, unvollständige Daten oder falsche Zeichen  
 2. System zeigt entsprechende Fehlermeldung an  
 3. Fortfahren mit „Punkt 5 System zeigt Profildatenmaske an“

#### Use Case 520: Notifikation aktivieren / deaktivieren

|  |
| --- |
| Use Case: Notifikation aktivieren / deaktivieren Use Case-Nummer: 520 Ziel: Kunde kann auswählen, ob er für Antworten auf seine Kommentare eine Notifikation erhält Hauptakteur: Kunde Nebenakteure: keine Vorbedingung: Kunde ist registriert Auslöser: Kunde möchte ändern, ob er für Antworten auf seine Kommentare eine Notifikation erhält  Eingabedaten: keine  Ausgabedaten: keine  Nachbedingung: Notifikationseinstellungen werden geändert Garantie falls erfolgreich: Kunde erhält die Notifikation ( / nicht) Garantie falls Fehlschlag: Kunden erhält die Notifikation immer noch ( / nicht) |

**Hauptszenario:**  
  
1. Benutzer möchte auswählen, ob für Antworten auf seine Kommentare eine Notifikation erhält  
2. Benutzer meldet sich im Onlineblog an  
3. System zeigt die Benutzeroberfläche an  
4. Benutzer wählt ‚Notifikationen‘  
5. System zeigt Notifikationen-Maske  
6. Benutzer wählt aus  
7. System ändert Notifikation  
8. Szenario Ende  
  
**Erweiterungen:**  
  
6. Benutzer wählt aus  
 1. Benutzer wählt ‚Ja‘  
 2. Fortfahren mit „Punkt System ändert die Notifikation“  
  
 1. Benutzer wählt ‚Nein‘  
 2. Fortfahren mit „Punkt System ändert die Notifikation“

# Systemarchitektur

## Analyse der nicht-funktionalen Anforderungen

Da die Architektur der zu implementierenden Software schon durch viele Anbieter auch kostenfrei angeboten wird, haben wir uns dafür entschiedene keine eigene Software zu entwickeln, sondern einen fertigen Blog zu implementieren.

Darum haben wir die nicht-funktionalen Anforderungen nur hinsichtlich ihrer Relevanz gewichtet. Dabei haben wir eine Skala von 1 -10 verwendet: 1 ist am wenigsten relevant und 10 am relevantesten

|  |  |
| --- | --- |
| Anforderungen | Gewichtung |
| Kundenfreundliche Benutzeroberfläche | 8 |
| Einfache Handhabung der Webauftritte durch Sabine Krogemann | 10 |
| Sicherheit beim Kundenkontakt | 5 |
| Moderne Technik und Design | 7 |
| Rund um die Uhr verfügbar | 3 |

### Kundenfreundliche Benutzeroberfläche

Die kundenfreundliche Benutzeroberfläche wird eher stark gewichtet, weil sie ein wichtiger Teil des Auftretens ist. Wir gehen davon aus, dass die Benutzer des Blogs eher weniger versiert sind im Umgang mit Computern und Internet. Daher muss der Blog für sie einfach und übersichtlich dargestellt sein. Die Oberfläche soll sie also ansprechen.

### Einfache Handhabung der Webauftritte durch Sabine Krogemann

Die einfache Handhabung der Webauftritte durch Sabine Krogemann ist zentral, da es ein Firmenziel ist und Sabine Krogemann einen sehr grossen Wert auf dieses Ziel legt.

### Sicherheit beim Kundenkontakt

Die Sicherheit beim Kundenkontakt wird mittel gewichtet, da die Kundendaten dieses Blogs eher unbedeutend sind, nicht wie z.B. bei einer Bank.

### Moderne Technik und Design

Moderne Technik und Design werden heutzutage vorausgesetzt und erhalten daher eine relativ starke Gewichtung. Auch weil es von Sabine Krogemann gefordert wird. Glücklicherweise sind die angebotenen Blogsysteme auf dem neusten Stand.

### Rund um die Uhr verfügbar

Praktisch alle angebotenen Blogsysteme garantieren eine 24-stündige Verfügbarkeit. Sollte es trotzdem vorkommen, dass das System nicht funktioniert, ist es nicht so tragisch, wie z.B. bei einem Börsencrash. Ausfälle bei solchen Systemen nehmen nicht sehr viel Zeit in Anspruch. Daher erhält diese Anforderung nicht sehr viel Gewicht.

# Aktivitätsdiagramme

## Use Case-Gruppe 100: Einträge verwalten

### Aktivitätsdiagramm zu Use Case 110: kreieren



### Aktivitätsdiagramm zu Use Case 120: ändern



### Aktivitätsdiagramm zu Use Case 130: löschen



### Aktivitätsdiagramm zu Use Case 140: Anhänge hinzufügen



### Aktivitätsdiagramm zu Use Case 150: Kommentare verfassen



## Use Case-Gruppe 200: Kommentare

### Aktivitätsdiagramm zu Use Case 210: Kommentare verfassen



### Aktivitätsdiagramm zu Use Case 211: Captcha-Kontrolle



### Aktivitätsdiagramm zu den Use Cases 212-222



## Use Case-Gruppe 300: Kundenverwaltung

### Aktivitätsdiagramm zu Use Case 310: Kunden sperren



### Aktivitätsdiagramm zu Use Case 320: Kundenprofil ansehen



## Use Case-Gruppe 400: Anmeldung

### Aktivitätsdiagramm zu Use Case 410: Registrierung der Kundendaten



### Aktivitätsdiagramm zu den Use Cases 411/412



### Aktivitätsdiagramm zu Use Case 430: Passwort vergessen



## Use Case-Gruppe 500: Kundenkontoverwaltung

### Aktivitätsdiagramm zu Use Case 510: Profildaten ändern



### Aktivitätsdiagramm zu Use Case 520: Notifikationen

