|  |  |
| --- | --- |
| **Softwareentwicklungsplan Chatmail** | |
| Gruppenmitglieder: | Jeremia Haackmann  Jonas Kaufmann |

# Anforderungsanalyse

### Use Case Analyse

<*Grafische Darstellung der Anforderungen in Form von Use Cases>*

### 

### Use Case Beschreibung

*<Detaillierte Beschreibung der Use Cases nach einer vorgegebenen Struktur>*

|  |  |
| --- | --- |
| USE CASE: Empfänger auswählen | |
| Primärer Aktor: | Benutzer |
| Vorbedingung: | Benutzer vorhanden, Datenbank Verbindung aktiv |
| Wichtigstes Erfolgsszenario: | 1. Dropdown öffnen 2. Empfänger suchen 3. Empfänger aus Liste anklicken |
| Wichtige Varianten: | 1. Empfänger nicht in Liste |
| Auswirkungen: | Empfänger ausgewählt |
| Anmerkungen: |  |
| Offene Fragen: | Was soll passieren, wenn ein User nicht in der Liste ist? |

|  |  |
| --- | --- |
| USE CASE: Nachricht schreiben | |
| Primärer Aktor | Benutzer |
| Vorbedingung | keine |
| Wichtigstes Erfolgsszenario: | 1. Nachricht eingegeben |
| Wichtige Varianten: |  |
| Auswirkungen: | Textfeld gefüllt |
| Anmerkungen: |  |
| Offene Fragen: |  |

|  |  |
| --- | --- |
| USE CASE: Nachricht senden | |
| Primärer Aktor | Benutzer |
| Vorbedingung | Empfänger ausgewählt, Textfeld enthält mindestens ein Zeichen, das kein/e Leerzeichen/Leerzeile ist, Datenbankverbindung aktiv |
| Wichtigstes Erfolgsszenario: | 1. Button geklickt 2. Benachrichtigung im Nachrichtenfeld: Nachricht versendet |
| Wichtige Varianten: | Fehlermeldung Pop-Up: Nachricht konnte aus folgendem Grund nicht versendet werden:  2.a) Kein Empfänger ausgewählt!  2.b) Keine gültige Nachricht im Textfeld |
| Auswirkungen: | Nachricht in Datenbank |
| Anmerkungen: |  |
| Offene Fragen: |  |

|  |  |
| --- | --- |
| USE CASE: Nachricht laden | |
| Primärer Aktor | Benutzer |
| Vorbedingung | Es befindet sich eine Nachricht für den Nutzer in der Datenbank |
| Wichtigstes Erfolgsszenario: | 1. Text erscheint: „RadioButton Datenbankverbindung leuchtet“ 2. Nachrichten wurden geladen |
| Wichtige Varianten: | Keine Nachricht erhalten:  1.a) Popup erscheint „Keine Datenbankverbindung“  2.a) Dem Nutzer wurde keine Nachricht gesendet -> Textfeld leer |
| Auswirkungen: | Nachricht kann gelesen werden |
| Anmerkungen: |  |
| Offene Fragen: | Sollen immer alle ehemaligen Nachrichten geladen werden? |

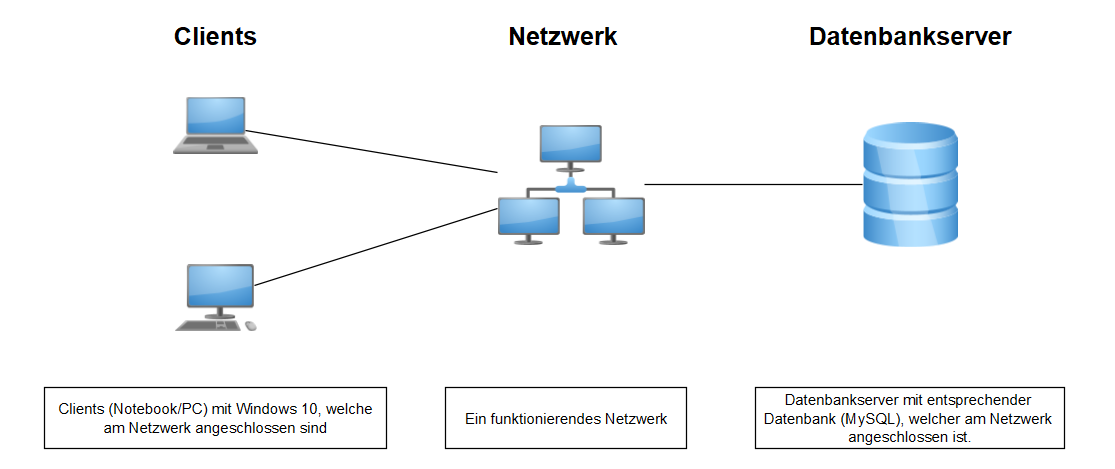
### Abgrenzungskriterien

*<Funktionalität die das System nicht besitzt>*

* Benutzer hinzufügen
* Chatverlauf kann nicht vom Nutzer gelöscht werden
* Kein 1 zu 1 / FaceToFace Chatverlauf (man sieht die Nachrichten von allen)
* Wenn neue Benutzer in die Datenbank kommen, muss das Programm neugestartet werden

# Systemarchitektur

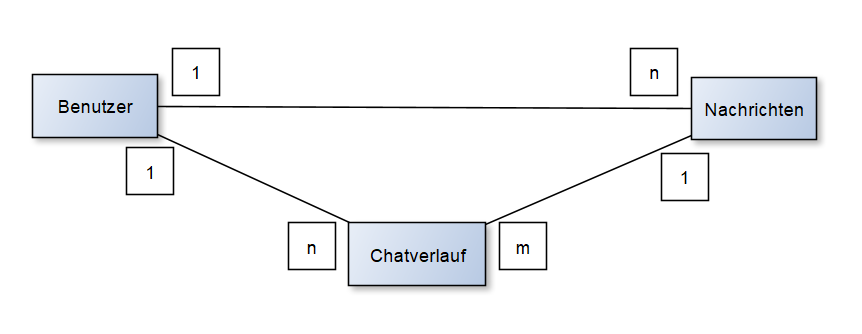
*<Beschreibung der verwendeten Infrastrukturkomponenten (Datenbanken, Server, Technologien) und grafische Darstellung der Gesamtarchitektur>*



# Implementierung

### Statische Perspektive

*<ER-Modell>*



<Relationenmodell>

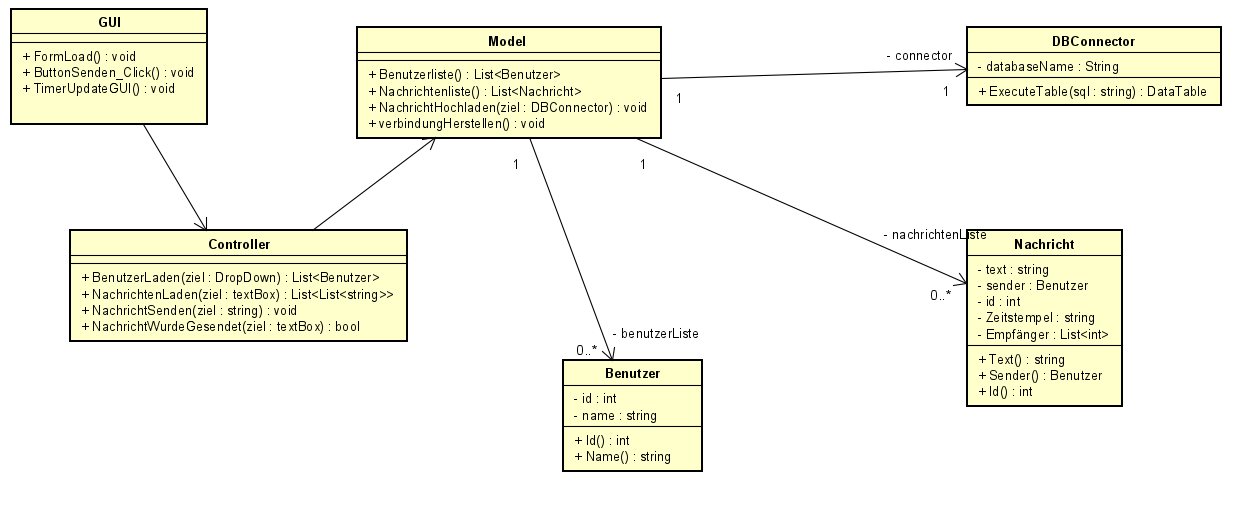
Benutzer(BenutzerID, Namen)

Nachricht(NachrichtenID, TimeStamp, Text, SenderID)

Chatverlauf(EmpfängerID, NachrichtenID)

Primärschlüssel, Fremdschlüssel

*<UML-Klassendiagramm der wesentlichen Klassen>*

**

# Test

*<Beschreibung der wesentlichen Testfälle und deren Durchführung>*

|  |  |  |  |
| --- | --- | --- | --- |
| Testid | Bezeichnung | Testbeschreibung mit erwartetem Ergebnis | Status |
| 1 | Doppelname | Bei dem Sendeevent wird an einem Halbgeviertstrich gesplittet. Ein Nutzer könnte einen Doppelnamen haben, welcher mit einem Minus geteilt wird. Erkennt C# den Unterschied zwischen einem Bindestrich und einem Halbgeviertstrich?  Erwartetes Ergebnis: C# erkennt es | Bestanden |
| 2 | Kein Benutzer ausgewählt bei senden | Erwartetes Ergebnis: Programm läuft weiter + Fehlermeldung | Nicht bestanden  Edit: Bestanden |
| 3 |  |  |  |