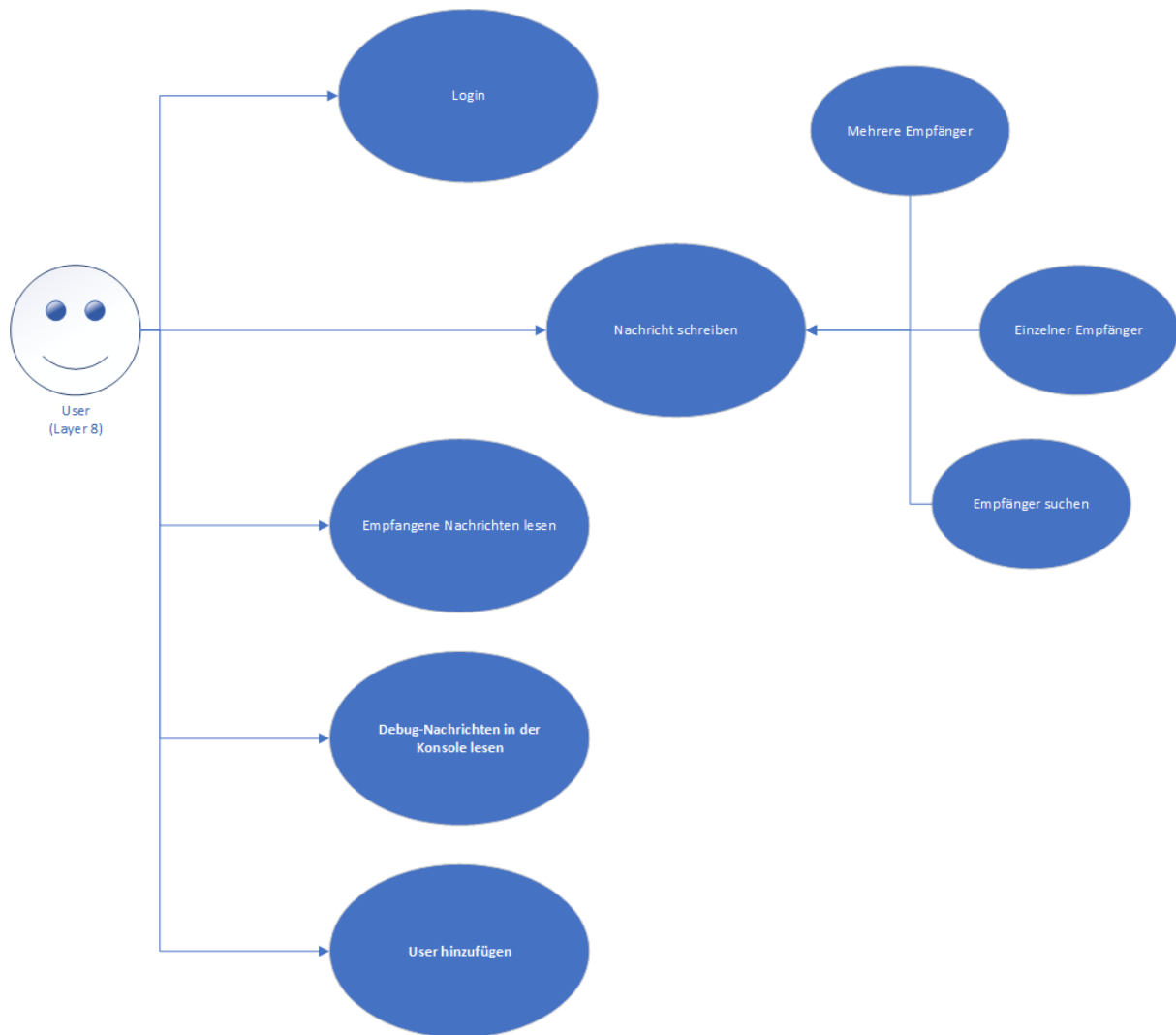


Softwareentwicklungsplan Chatmail	
Gruppenmitglieder:	Björn Biester Yannick Groß

Anforderungsanalyse

Use Case Analyse



Use Case Beschreibung

USE CASE: Login	
Primärer Akteur:	User
Vorbedingung:	User nicht eingeloggt
Wichtigstes Erfolgsszenario:	User wählt seine UserId aus User wird eingeloggt
Wichtige Varianten:	-
Auswirkungen:	User ist eingeloggt und kann Nachrichten senden/empfangen
Anmerkungen:	-
Offene Fragen:	-

USE CASE: Nachricht schreiben	
Primärer Akteur	User
Vorbedingung	User eingeloggt
Wichtigstes Erfolgsszenario:	User wählt ein oder mehrere Empfänger aus User gibt Nachrichtentext ein User drückt "Senden" Nachricht wird gesendet
Wichtige Varianten:	<ul style="list-style-type: none">• An einzelnen Empfänger• An mehrere Empfänger
Auswirkungen:	Schülerliste wird aktualisiert
Anmerkungen:	-
Offene Fragen:	-

USE CASE: Empfangene Nachrichten lesen	
Primärer Akteur	User
Vorbedingung	User eingeloggt
Wichtigstes Erfolgsszenario:	Alle 5 Sekunden wird geprüft ob neue Nachrichten an den User existieren Ungelesene Nachrichten werden dem User angezeigt
Wichtige Varianten:	-
Auswirkungen:	User empfängt ungelesene Nachrichten
Anmerkungen:	-
Offene Fragen:	-

USE CASE: Debug-Nachrichten in der Konsole lesen	
Primärer Akteur	User
Vorbedingung	-
Wichtigstes Erfolgsszenario:	User öffnet über Menü Konsole und kann Debug-Nachrichten die während des Programmablaufs ausgegeben wurden lesen
Wichtige Varianten:	-
Auswirkungen:	User liest Debug-Log
Anmerkungen:	-
Offene Fragen:	-

USE CASE: User hinzufügen	
Primärer Akteur	User
Vorbedingung	-
Wichtigstes Erfolgsszenario:	User öffnet über Menü Admin-Bereich und erhält eine Auflistung aller User User fügt über Textfelder einen neuen User mit Name, Vorname und Displayname hinzu
Wichtige Varianten:	-
Auswirkungen:	User fügt neuen User hinzu
Anmerkungen:	-
Offene Fragen:	-

Abgrenzungskriterien

Dieses System ist nicht für Kommunikation in festen Gruppen gedacht. Ein User kann zwar nach dem BCC-Prinzip eine Nachricht an mehrere Empfänger senden, eine Antwort ist aber nur an den direkten Absender möglich.

Da dieses System nur eine Form des Kurznachrichtendienstes darstellt, ist das Versenden von Anhängen nicht vorgesehen.

Systemarchitektur

Verwendete Technologien:

- GUI und Controller geschrieben in C# und .NET-Framework
- Datenbank-Server mit mariaDB und XAMPP als Development-Tool

Statische Perspektive

ER-Modell



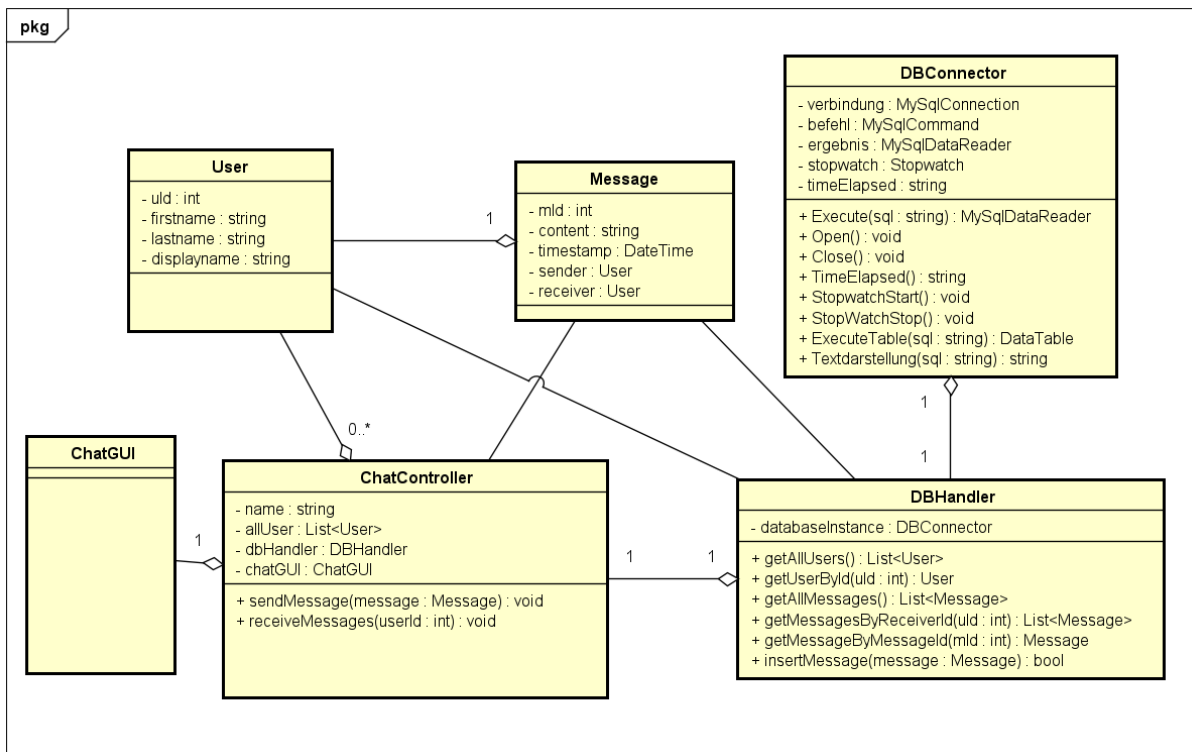
Relationen-Modell

User(uID, vorname, nachname, displayname),

Message(mId, content, timestamp, senderId)

MessageReceiver(Id, mId, rId)

UML-Klassendiagramm



Tests / QM

Testid	Bezeichnung	Testbeschreibung
1	Git	Git ist ein verteiltes Versionskontrollsystem zur Verfolgung von Änderungen im Quellcode während der Softwareentwicklung. Es dient zur Koordinierung der Arbeit zwischen Programmierern, kann jedoch verwendet werden, um Änderungen in einem beliebigen Satz von Dateien zu verfolgen. Zu seinen Zielen gehören Geschwindigkeit, Datenintegrität und Unterstützung für verteilte, nichtlineare Workflows. Aufgrund der "Branching" Technologie von Git können wir mehrere Versionen unseres Projektes "parallel" speichern. Dies ermöglicht uns eine "Deployment" Version (master) bereitzustellen, die nur gut getestete und fertige Features beinhaltet.
2	GitHub	GitHub stellt eine moderne Schnittstelle für git-basierte Versionierung bereit und bietet darüber hinaus Tools wie PullRequest-Kontrollen und automatisierte Softwaretests.
3	dotnet build	Hiermit testen wir alle eingehenden PullRequests (Änderungen) auf Kompilierungsfehler. Dies testen wir mit GitHub Actions; GitHub Actions ist ein VM-basiertes Tool, welches automatisch Scripte u.Ä. ausführt, Ergebnisse kontrolliert und entsprechend die Änderungen als fehlerhaft oder gültig markiert.
4	Unittests	Desweiteren verwenden wir Unittests, die die korrekte Funktion des Datenbankkonnektors, Frontend, u.Ä. überprüft.
5	Integrationtest	Ein Integrationstest spiegelt die Kombination mehrere Unittests gleichzeitig wider. Hierbei wird also die korrekte Funktion der Einheiten (Units) untereinander überprüft.