

# Programmeertechnieken opdracht 3

Robert Arntzenius, Daniël Lin, Jelle Kientz

May 2018

## 1 Introductie

In deze opdracht hebben we programma's geschreven zodat je met elkaar kan chatten. De programma's bestaan uit meerdere clients en een server, er is ook nog een database gemaakt om informatie op te slaan. De client leest wat de gebruiker wilt doen en stuurt dit door naar de server. De server kijkt vervolgens wat het met deze info moet doen, zoekt zo nodig dingen op uit de database en stuurt de bijbehorende info dan weer terug naar de client. Zo kan de gebruiker inloggen, chatten en wisselen met wie hij of zij wilt chatten. Er wordt niets op de client opgeslagen, dit zorgt er voor dat je op elk apparaat je chatgeschiedenis kan terugvinden. In de console van de server kan je ook nog duidelijk zien wat er allemaal gebeurt, dus of er iemand succesvol inlogt of dat er een bericht wordt gestuurd.

## 2 Opzetten programma

x Om het programma op te zetten start men met het opstarten van de applicatie Monodevelop. Vervolgens moet je de volgende packages installeren

1. System.data.sqlite
2. Microsoft.sqlserver.SQLmanagementObjects

Je hoeft niets te doen om de database te verbinden, die wordt aangemaakt op het moment dat de server aan gaat. Wat er nog wel moet gebeuren is de configuratie. Wat je moet doen is als volgt:

1. geef packages mee aan
2. geef app.config mee aan

Het enige wat je nu nog hoeft te doen is als eerste de server aan zetten, en vervolgens de clients. Je kan inloggen met een bestaande gebruikersnaam met het bijbehorende wachtwoord, of je kan een nieuwe gebruiker aanmaken. Een nieuwe gebruikersnaam aanmaken doe je simpelweg door een nog niet bestaande gebruikersnaam in te vullen met daarbij het wachtwoord wat je er voor wilt gebruiken.

## 3 Taakverdeling

Robert heeft aan de database gewerkt. Dit bevat het maken van de database, het schrijven van queries om informatie uit de database te halen en om nieuwe informatie toe te voegen aan de database. In dit deel van de code wordt ook gecontroleerd of de gebruiker is ingelogd en het inloggen vindt hier ook plaats.

Daniël heeft zich vooral bezig gehouden met de communicatie tussen de front- en backand. aangezien hij al eerder in c# heeft gewerkt kon hij ook ondersteunend bij de andere onderdelen helpen.

Jelle is bezig geweest met de GUI. Dit gaat over hoe het programma duidelijk en overzichtelijk voor de gebruiker moet zijn. Ook het aanroepen van wat er wanneer aan de server gevraagd moet worden is hier een onderdeel van.

## 4 Ontwerpdocumenten

Jelle

Front end

- 1 Login screen
  - send user name
  - send password
- 2 send message
  - clear message box
- 3 online users
  - show online users
  - scroll through online users
  - if click on user, send user name
- 4 chat history
  - show last 15 rows
  - scroll trough text

Daniel

API

- 1 establish connection between server and client
- 2 format everything between front end and back end

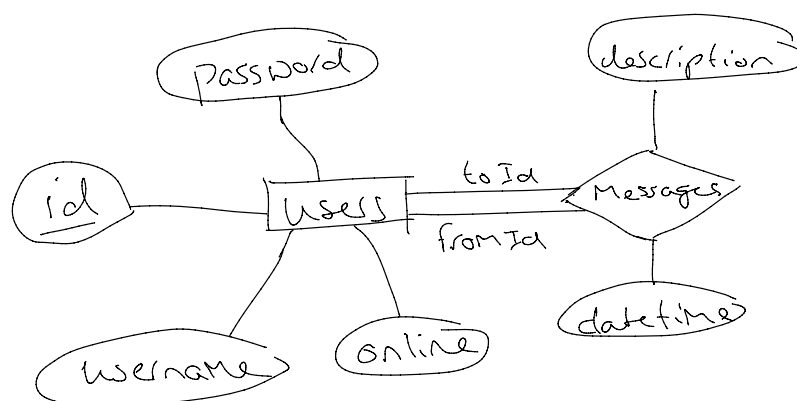
Robert

Back end

- 1 queries
  - 1) give chat history
  - 2) add to chat history
  - 3) check if user + password is correct
  - 4) if user does not exist. Create new user
- 2 send users online
- 3 ping user if messaged
- 4 create user Id's

### ER-Diagram Database

01 June 2018 01:13



## 5 Documentatie API

Voor de Login case op de server heb je de format "username.password" nodig. Dit wordt verzonden naar de server en daar worden de username en password uit elkaar gehaald en gekeken of deze bij elkaar horen. Er wordt naar de client teruggestuurd of het inloggen is voltooid of niet.

Om een "Message" case te sturen heb je de format "username,recipient.message" nodig. Dit wordt ook verzonden naar de server en daar worden username recipient en message uit elkaar gehaald en naar een database query gestuurd die het vervolgens in de database zet

Om alle "GetMessage" tussen de jezelf en een gebruiker op te halen gebruik je "username,recipient". Er wordt dan na het verkrijgen van de berichten uit de database een string teruggestuurd van de zeventien gevraagde berichten aan elkaar gehecht met het karakter: "`.

En om alle online users te ontvangen krijg je zonder format een list van alle online users met de functie "GetUsers". Hieruit kan de GUI vervolgens zelf bepalen welke er op het scherm worden via de scrollbar.