# **Chat-Program**

The follow points, which are listed below, show or describe my initial requirements. With the aid of the comments (*which have a cursive font style*) I am going to explain what points I actually accomplished. Additionally, I marked the accomplished ones with a green font colour.

# Initial Requirements VS. accomplished Requirements

* desktop application / fat client

*WPF-Desktop client*

* change colour theme

*actually, I implemented a function to change the background of the application, by uploading and image*

User

* username
* profile picture
* status (shown to other users)
* login
* logout

*when a user registers himself, he can add a username, status as well as a profile picture; Also the login and logout was implemented.*

Messages

* text
* pictures
* files
* emoji
* information if message is transmitted
* offline storage / re-delivery

currently my chatmessage model only can contain a message, send date, sender id and the chat id to which the message belongs

contact list

* add other contacts/users
  + ask/send an invitation for adding
* remove contact/user
* create groups with added contacts
* blocking of contacts

in my chat program the contacts of the user are shown but unfortunately not the things mention above.

Back-up

* back-up all messages
* if chat is loaded return a chat history (maybe last week)

all messages are backed up in the database, and when the user choses a chat -> all messages will be loaded

# Significant hacks & things that took the most of effort.

* Using the MVVM pattern

During my implementation phase I made sure to use the MVVM pattern correctly. Nearly every View has a viewmodel as datacontext and only the viewmodel knows the model. Furthermore, I used a lot the databinding-function. Additionally, I implemented converters to convert images dynamically when they are needed.

* Design

From the beginning I focused one the design because I personally prefer a at least a bit good-looking and pleasing application. Unfortunately, I was focussing a lot on the design and the layout the real-time messaging was moved into the background.

* inherited Get-Method

On my WPF-Client, the restclient classes take over the communication with the webserver’s api. To save some lines of code, I implemented in the BaseRestClient class a get-method which the subclasses can use.

* BaseUser class

When I load all contacts of a specific user, I send back a collection of baseusers. That means, I do not send all information of the user to another but only the name, statusmessage and usericon. So, the user only gets a trimmed version of the normal user objects.

* Login by pressing the Enter key

After the user entered his/her credentials, the login process can be started by pressing the enter key. This is a very small and easy-to-implement feature but in my eyes, it is very convenient.