

## Mezzago – Software Engineering Mini Project

This documentation is about the second mini-project from the course Software Engineering conducted by Bradley Richards. The project idea is to create a messaging application based on the documentation provided by the lecturer.

We decided to create an application called “Mezzago”. “Mezzago” provides all of the features provided by the server. Different connections are called channels (chat room, direct message).

### Functionalities

Here are some of the features listed:

The user has the possibility to *create a login* and *login* (if an account exists). The username is stored locally.

The user can *create* and *join chat rooms* (also called group chats). If the user creates a chat room, the user automatically is joined.

The chatroom can be *deleted* or *left*. Leave is always successful.

The user can *connect* to another user (*direct message*). Therefore, the other user needs to be *online*, otherwise the connection is denied. If a connection is established once, the connection stays until the user *removes* the connection.

In addition to the basic functions of the server, the user can choose from a variety of settings and functionalities:

The user can change the *password* and *delete* the account.

The user can change the *audio* settings. Change, dis- and enable notification sounds.

A channel can be *blocked*. If a channel is blocked, no message is added to the according channel. If a single direct message is blocked, messages of this channel are ignored in chatrooms (only locally supported).

For *lazy-user-functions*, there is the functionality to create chat-rooms and direct message connection via the main text-field. To use this functionality, the user has to indicate it via '#', then choose the type of connection ('room'; 'dm'). If it is a chat-room it is to be chosen if public or private ('hidden').

Specifications must be separated by ':'.

*Example – hidden chat-room: '#room:hubbabubba:hidden'*

- A hidden (private) chat-room is created with the name '*hubbabubba*'

*Example – public chat-room: '#room:alabama'*

- Creates a public chatroom called '*alabama*'

*Example – direct message connection: '#dm:hansli'*

- Creates a new dm-connection with the user hansli, if the user is online

In the banner of every chat-room, there is a dropdown which contains a list with each member of this room. The user can select an item of this list, which opens the window to connect a new direct-message connection.

We decided to implement some functionalities via the context menu.

- Direct-message channels:
  - o Block channel
  - o Delete connection
- Group-chats only:
  - o Leave room
  - o If admin:
    - Add user (only if the corresponding user is online)
    - Remove user
    - Delete room (delete also on server)

### Settings and Styling

For every user that logs into the application, a separate directory is created. This directory is used to store the channel objects, user-specific settings, and blocked users list.

If a user doesn't interact for 5 days with a certain channel, the channel is deleted automatically after the user successfully logged.

The application is mainly styled with the usage of CSS.

For multilingualism we decided to go with English as standard and German as a secondary language. The application language can be changed in the settings.

### Group members and further information

The group consists of:

- Antonio Briatico – antonio.briatico@students.fhnw.ch
- Mathis Hermann – mathis.herman@students.fhnw.ch
- Lars Stegmann – lars.stegmann@students.fhnw.ch

Further Documentation and source code:

- <https://gitlab.fhnw.ch/mathis.hermann/le-mezzago>
- [mezzago.ch](https://mezzago.ch)