

FEATURES

Login - Ein bereits registrierter User kann sich mit seinem Account Namen und Passwort einloggen. Es wird eine Session erstellt.

Register - Es kann ein neuer Account mittels Account Name, Passwort und einer Wiederholung des Passworts erstellt werden.

Logout - Ein eingeloggter User beendet die derzeitige Session.

Delete Account - Ein User kann nach dem Login seinen Account permanent löschen

Join Chat - Ein User mit aktiver Session tritt einem Chatraum bei.

Leave Chat - Ein User mit aktiver Session verlässt einen Chatraum.

Create Chat - Ein User mit aktiver Session erstellt einen neuen Chatraum.

Delete Chat - Ein User mit aktiver Session löscht einen Chatraum.

Show Chats - Ein User mit aktiver Session kriegt alle Chaträume angezeigt.

Show Chatroom Users - Ein User mit aktiver Session sieht von einem Chatraum alle teilnehmenden User.

Send message - Ein User mit aktiver Session, der sich in einem Chatraum befindet kann eine Nachricht tippen und mit "Enter" absenden.

Delete message - Ein User mit aktiver Session, kann Nachrichten die er versendet hat löschen.

Refresh Chat - Nach dem das Event eintrifft, dass eine neue Nachricht gesendet wurde. Wird der Chat für alle Teilnehmer aktualisiert.

Password forgotten - Ein Fenster erscheint um eine Email-Adresse einzugeben. Nach der Bestätigung der Eingabe wird an diese ein neues Passwort gesendet.

Change Password - Ein User mit aktiver Session kann nach Eingabe seines alten Passwortes ein neues Passwort festlegen.



Interface

```
interface Chat_Server {
    public void join_chat(sid, chat_id);
    public void show_chats(sid);
    public void create_chat(sid, chat_name);
    public void delete_chat(sid, chat_name);
    public void leave_chat(sid, chat_name);
    public void refresh_chat(sid, chat_name);
    public void show_chatroom_users(sid, chat_name);
    public void send_message(sid, chat_id, message);
    public void delete_message(sid, chat_id, message_id);
    public void forgot_password(sid, user_id);
    public void change_password(sid, user_id);
    public void login(sid, user_name, password);
    public void register(sid);
    public void delete_account(sid);
    public void logout(sid, user_id);
}
```

Pseudo-Funktionen

```
login(acc_name, pw)
    if not acc_name exists:
        return AccountError
    if not acc_name["pw"] is pw:
        return PasswordError
    sid = create_session()
    return sid

register(acc_name, pw, pw_repeat)
    if acc_name exists:
        return AccountError
    if not pw is pw_repeat:
        return PasswordError
    user = create_user(acc_name, pw)
    return login(user)

logout(sid)
    (leave_all_chats(sid))
    destroy_session(sid)

join_chat(sid, chat_id)
    if not chat_id exists:
        return ChatroomError
    chat_id.add(SID)

show_chats(sid) -> array(chat_names)
    return Chat.chat_rooms

create_chat(sid, chat_name) -> chat_id
    if chat_name in Chat.chat_rooms:
        return ChatroomError
    Chat.chat_rooms.add(chat_name)

delete_chat(sid, chat_name)
    if not chat_name in Chat.chat_rooms:
        return ChatroomError
    chat_name.remove(SID)

leave_chat(sid, chat_name)
    if not chat_name in Chat.chat_rooms:
        return ChatroomError

send_message(sid, chat_id, message)
    if not chat_id exists:
        return ChatroomError
    chat_id.send(sid, message)

delete_message(sid, chat_id, message_id)
    if not chat_id exists:
        return ChatroomError
    if not message_id exists:
        return MessageError
    chat_id.delete(sid, message_id)

show_chatroom_users(sid, chat_name) -> Array(Users)
    if not chat_name in Chat.chat_rooms:
        return ChatroomError
    return chat_name.get_id().users

refresh_chat(chat_id)
    while chat_id
        wait Event.new_message
        for user in chat_id.users
            user.refresh()

change_password(sid, old_pw, new_pw, repeat_pw)
    if not old_pw is user["PW"]
        return PasswordError
    if not new_pw is repeat_pw
        return PasswordError
    if old_pw is new_pw
        return PasswordError
    user["PW"] = new_pw

delete_account(sid, user_name)
    if not sid in users[]
        return userError
    user_name.remove(sid)

forgot_password(sid, email):
    new_password = generate_password()
    send_email(email, new_password)
```