**Documentation of**

**Assignment 3**

**Web sockets and security**

Name: Onea Vlad

Group: 30442

1. **Conceptual architecture of the distributed system**

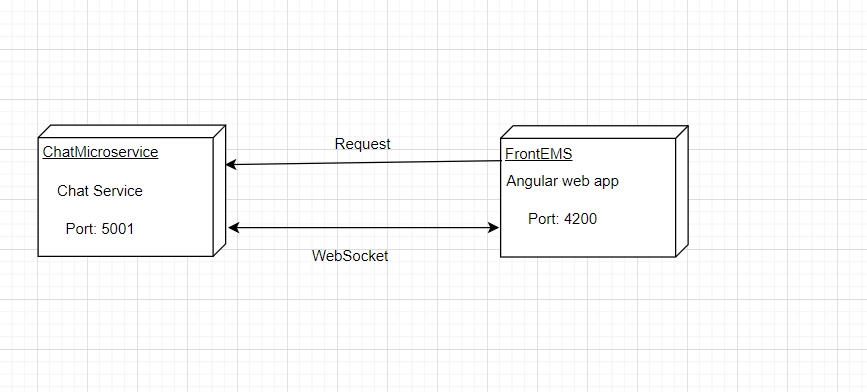
**Backend:** For the backend I created a microservice called ChatMicroservice for which I used .Net. I structured it in the following way: I have a chat controller called Chat Controller which handles HTTP requests. Here I used a method called SendMessage which takes message requests and forwards them to the ChatHub for broadcasting. On the Hub folder I have ChatHub which is SignalR Hub and which manages WebSocket connections. Here I have 2 methods: OnConnectedAsync() which manages new connections and maps users to their connection IDs and SendMessage which sends messages to a specific users based on connection mapping. Another Class which I used here is ConnectionMappings which maintains a map of usernames to connection IDs to track active users and their WebSocket connections. I also have a package for DTOs **(Data Transfer Objects):** DTOs are objects used to transfer data between layers of an application. They’re used transferring data between different parts of the application, like from the controller to the hub. I secured this microservice by adding JWT authentication which ensures secure communication and verifies user identities. Used in the authentication process when establishing WebSocket connections. Another security feature is that is that I added CORS policy which ensure that requests are only accepted from trusted origins.

As a flow of ChatMicroservice I can say that users connect to the ChatMicroservice via a WebSocket connection established through SignalR (at the /chat-hub endpoint). When a user sends a message, the request is sent to the ChatController. The ChatController processes the request and sends the message to ChatHub. ChatHub then broadcasts this message to the appropriate recipients based on the connection mappings.

**Frontend:** For the frontend part which is new is that I added a chat service which manages the SignalR hub connection and message communication. Here I added some functions: startConnection which initializes the connection to the SignalR hub, addMessageListener which listens for incoming messages from the SignalR hub, and sendMessageToAdmin and sendMessageToClient which send messages to either the admin or a specific client. Then on the user dashboard I added a button which will redirect the user or the admin to ClientChatComponent or to AdminChatComponent. AdminChatComponent represents the admin side of the chat application and allows the admin to view and interact with multiple chats from different users. Here I have 2 functions: sendMessage which sends a message to the selected client and selectChat where a chat is selected for interaction. ClientChatComponent represents the client side of the chat application and is responsible for displaying messages between the client and the admin. Here I used sendMessage functions which send a message to the admin.

1. **UML Deployment Diagram**

I used Docker for the deployment of the application. Web app and chat microservice are deployed on containers. Web app is using port 4200 and chat microservice is using port 5001.

****