Assignment 1

Bogdan Cupsa

Group 3

**Conceptual Architecture of the Distributed System**

Microservices:

User:

The User microservice manages user operations for administrators and clients, including CRUD operations on client profiles and login functionality. It is implemented using the Django framework and uses SQLite for the database. It's accessible on port 8000.

Device:

The Device microservice handles CRUD operations for devices. It is also implemented using Django and uses SQLite as its database. This service is accessible on port 8012.

Chat:

The Chat microservice facilitates real-time messaging between administrators and clients. It is built with Django Channels and uses WebSocket protocol for bi-directional communication. It interacts with the frontend for real-time data exchange and is accessible on port 8024.

Frontend:

The frontend is developed in Angular and includes multiple pages such as login, admin operations, client operations, and a client page. The frontend interacts with the User and Device microservices via RESTful APIs and with the Chat microservice using WebSockets.

Docker:

The environment is containerized using Docker, with three primary services defined in the Docker Compose file:

user: Exposed on port 8000, using the default Python image.

device: Exposed on port 8012, also using the default Python image.

frontend: Exposed on port 80, built using the default Node image.

chat: Exposed on port 8024, using a Python image and integrated with a Redis server for channel layer support in Django Channels.

Communication:

The microservices communicate with the frontend through RESTful APIs for synchronous operations. The Chat microservice communicates asynchronously with the frontend via WebSockets to enable real-time messaging. Microservices also have the capability to interact with each other as needed, forming a cohesive backend system.

Security:

Security is enforced through token-based authentication, protecting API endpoints and ensuring that only authorized users can access sensitive data and operations.

