***Distributed Systems***

***Assignment 3***

***Remote Procedure Call (RPC)***

***Chat System for Client Support***

Student : Scutea Emilia

Group: 30442

Laboratory Assistant: Arcas Gabriel

Teacher: Pop Cristina



Technical University of Cluj Napoca

2022

# **Conceptual architecture**

The main purpose of this project is to design a platform on which the energy of users’ devices is stored.

Diagram

Description automatically generated

The implementation of the project has 3 parts: the message producer, the backend part and the frontend part.

The message producer contains the following classes:

1. CSVReader -> used to read data from csv and send them to the RabbitMQ AMQP Cloud
2. HourlyConsumption -> used to create a model which is sent to the queue
3. Main -> used to run the application based on a specified user device

Graphical user interface, text, application, chat or text message

Description automatically generated

The backend part contains the following packages and layers:

1. Entities -> represented by the classes in which the entities are declared: User, Role, Device, DeviceEnergyConsumption
2. DTOs -> used to send the data to the frontend
3. Mappers -> used to map data from an entity to its corresponding dto class
4. Repositories -> used to retrieve or send data from/to the database
5. Servicies -> used to implement methods required for the implementation of the functionalities of the project: CRUD methods on users, CRUD methods on devices, the mapping between device and user
6. Controllers -> rest controllers used to create http methods for sending/receiving data to/from the api
7. Security -> layer used for the security of the application, login implementation with jwt token and spring security
8. RabbitMQ -> contains the message consumer i.e. all services used to read data from the queue and store it in the database
9. Websocket -> contains all services used to send notifications to the clients when the maximum energy consumption is exceeded
10. Grpc -> contains the greet.proto file and the grpc files
11. Grpc\_Impl -> contains the service and server implementation for RPC

Graphical user interface

Description automatically generated with low confidence

The frontend part contains several structures:

1. auth -> auth interceptor
2. alert -> used to display alert when the maximum energy consumption is exceeded
3. common -> class declarations for models: Consumption, Device, Energy, User, Chart
4. components -> UI components for displaying the data on admin/user page, login form
5. services -> for authentication of the users&admins, for CRUD operations on devices and for CRUD operations on users, for user->device mappings
6. websocket -> contains the services used to connect to the websocket and receive notifications
7. grpc -> contains the greet.proto file
8. output -> contains the generated files for RPC

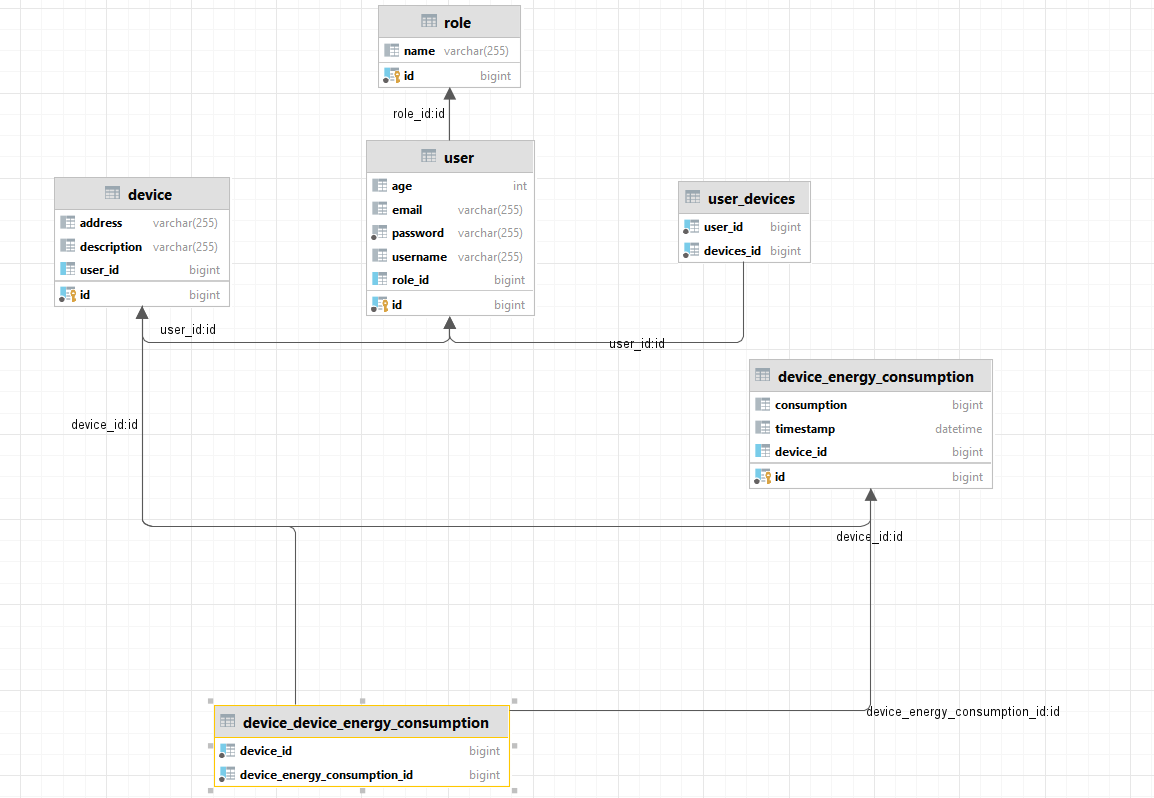
Graphical user interface, text

Description automatically generated with medium confidence

# **Database design**

The database design was made with MySql database. The database contains the following tables:

1. User -> used for storing user data: id, age, email, username, password, role\_id
2. Role -> used for storing the existing roles of users of the application : id, role (USER/ADMIN)
3. Device -> used to store data about the devices: id, address, description, user\_id
4. DeviceEnergyConsumption -> this table is used to keep all the data related to the energy consumption of all devices and has the following columns: id, consumption, timestamp, device\_id
5. Device\_DeviceEnergyConsumption -> used for the relationship between foreign keys between tables device and deviceenergyconsumption
6. User\_Devices -> used for the relationship between foreign keys of tables user and device



# **UML Deployment Diagram**



Diagram

Description automatically generated

