DISTRIBUTED SYSTEMS

Assignment 1

Request-Reply Communication Paradigm

Online Energy Utility Platform

a) Conceptual architecture of the online platform.

The main purpose to build this application to :

Calculate the energy consumption against any user's device according to the respective date.

Architecture:

Roles: Two role is this application:

1) admin

2) client

Responsibilty:

Admin: **- to create,read,delete,update the user;**

- to create,read,delete,update the devices;

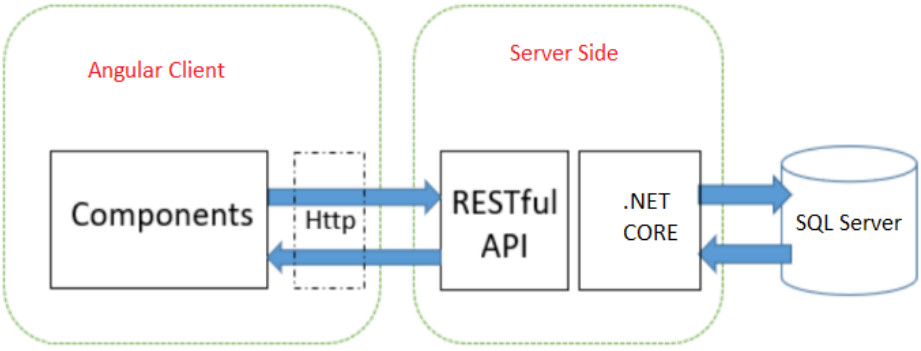
- assign the devices to particullar user;

- add daily energy consumptions against every user devices;

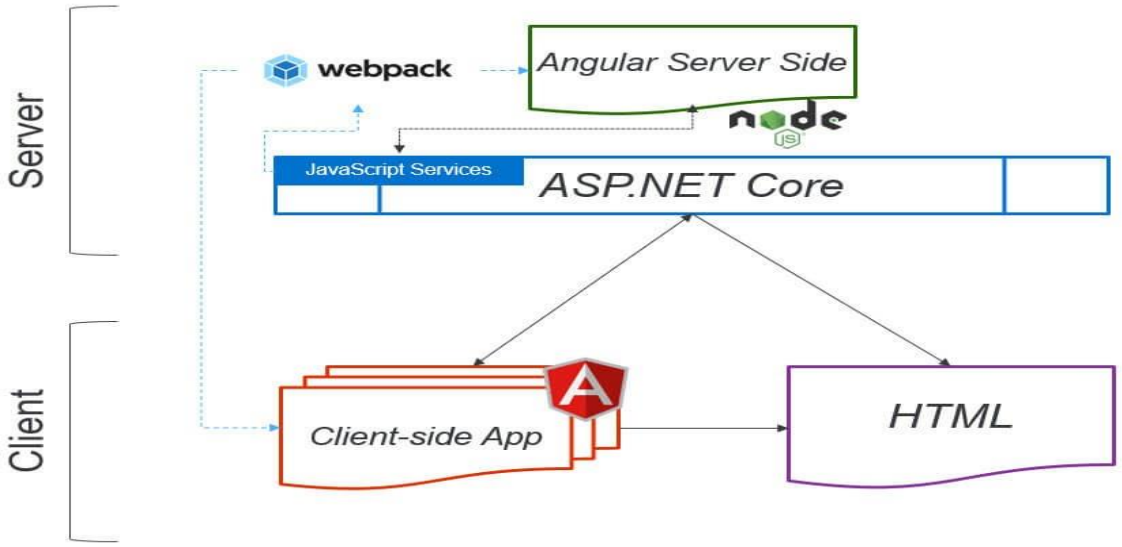
- assign the role to particular user;

Client: **- can view their devices which admin assigned;**

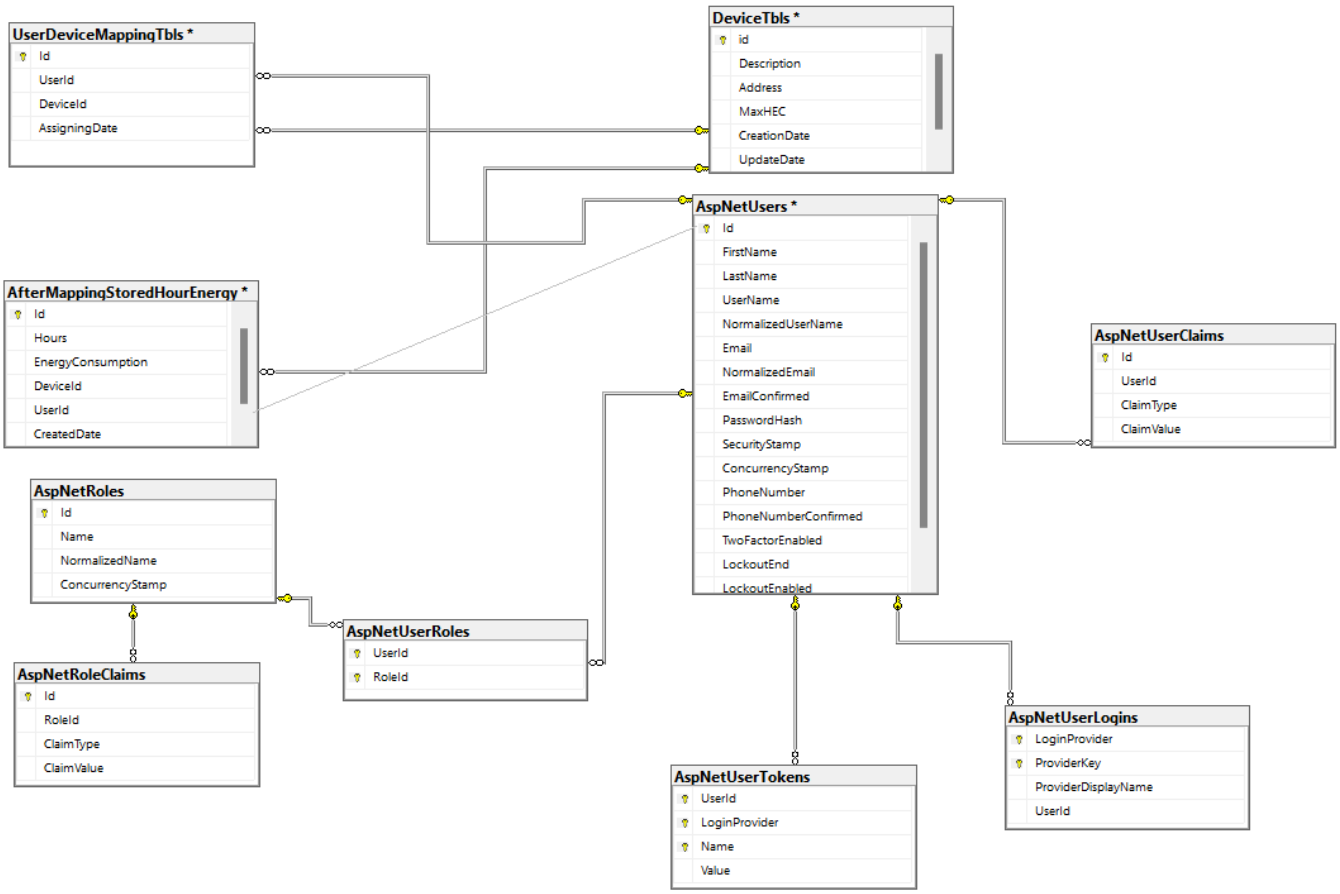
- check daily consumption in barchart by selecting device and date.



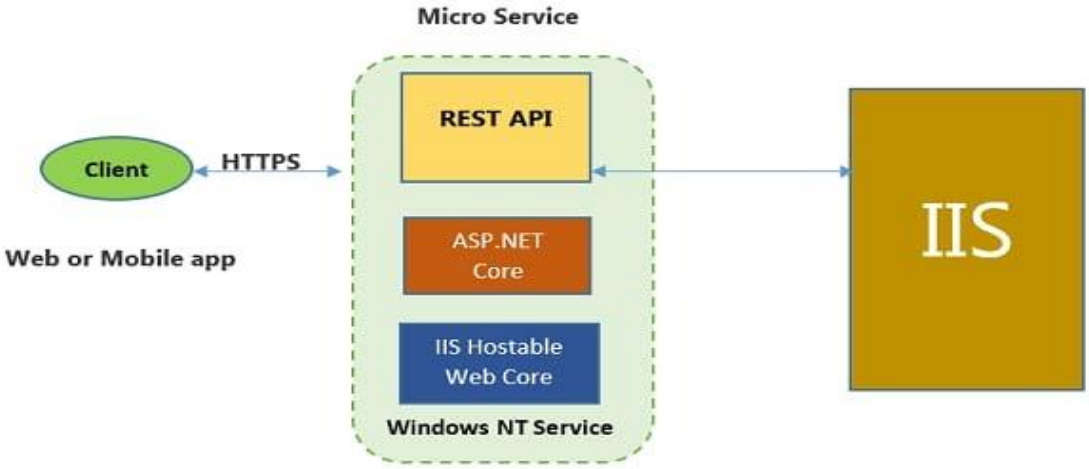
Figură 1 Diagrama conceptuală a arhitecturii aplicației dezvoltate

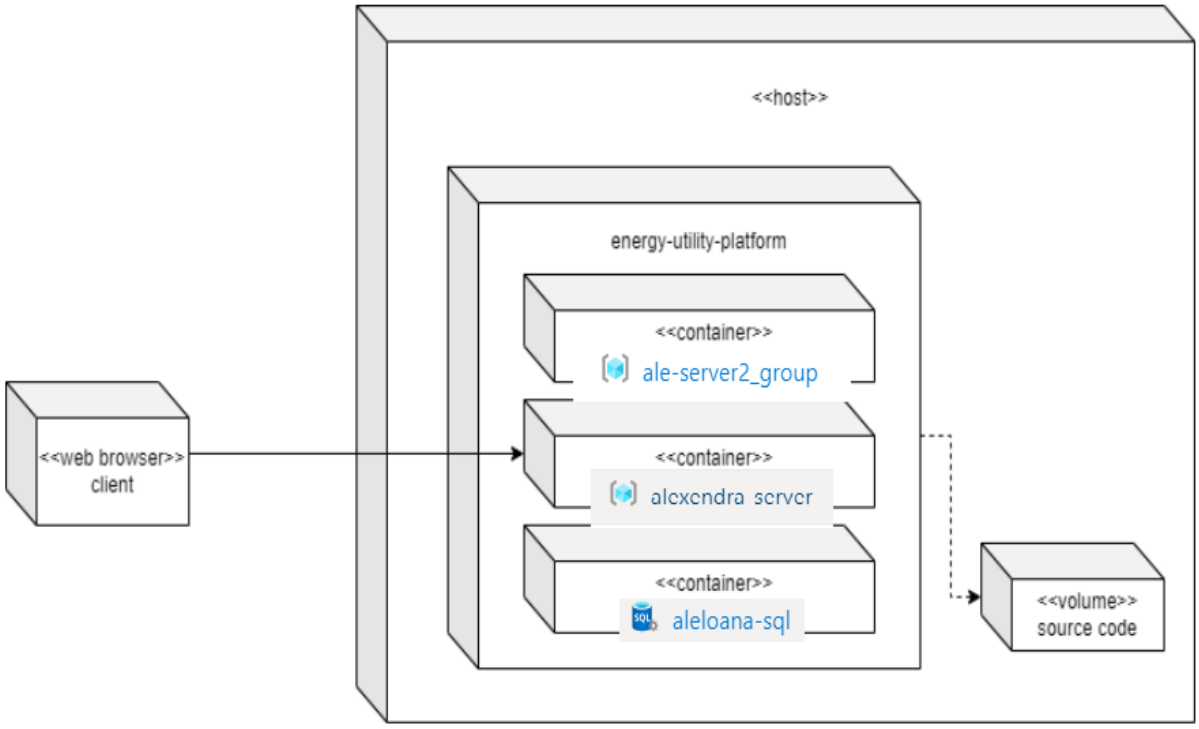

Server
webpack
Angular Server Side
node
JavaScript Services
ASP.NET Core
Client
Client-side App
HTML

b) DB design.


UserDeviceMappingTbls *
Id
UserId
DeviceId
AssigningDate
DeviceTbls *
Id
Description
Address
MaxHEC
CreationDate
UpdateDate
AspNetUsers *
Id
FirstName
LastName
UserName
NormalizedUserName
Email
NormalizedEmail
EmailConfirmed
PasswordHash
SecurityStamp
ConcurrencyStamp
PhoneNumber
PhoneNumberConfirmed
TwoFactorEnabled
LockoutEnd
LockoutEnabled
AfterMappingStoredHourEnergy *
Id
Hours
EnergyConsumption
DeviceId
UserId
CreatedDate
AspNetUserClaims
Id
UserId
ClaimType
ClaimValue
AspNetRoles
Id
Name
NormalizedName
ConcurrencyStamp
AspNetUserRoles
UserId
RoleId
AspNetRoleClaims
Id
RoleId
ClaimType
ClaimValue
AspNetUserTokens
UserId
LoginProvider
Name
Value
AspNetUserLogins
LoginProvider
ProviderKey
ProviderDisplayName
UserId

c) UML Deployment diagram.





Figură 5 Diagrama de deployment local al aplicației

d) Readme file containing build and execution considerations.

How to Run the Project:

Our Application Consist of two parts

1) Frontend [Developed in Angular]

2) Backend [Developed in .NET Core Web APIs]

Frontend:

1) Clone the Application By Coping the Url

2) Go to 'Frontend' Directory and open it in VS CODE

3) Write Command 'npm i --force' in command prompt (it will install all require packages)

4) Run 'ng s' in command prompt

Backend:

1) Open the 'Backend' Project in Visual studio

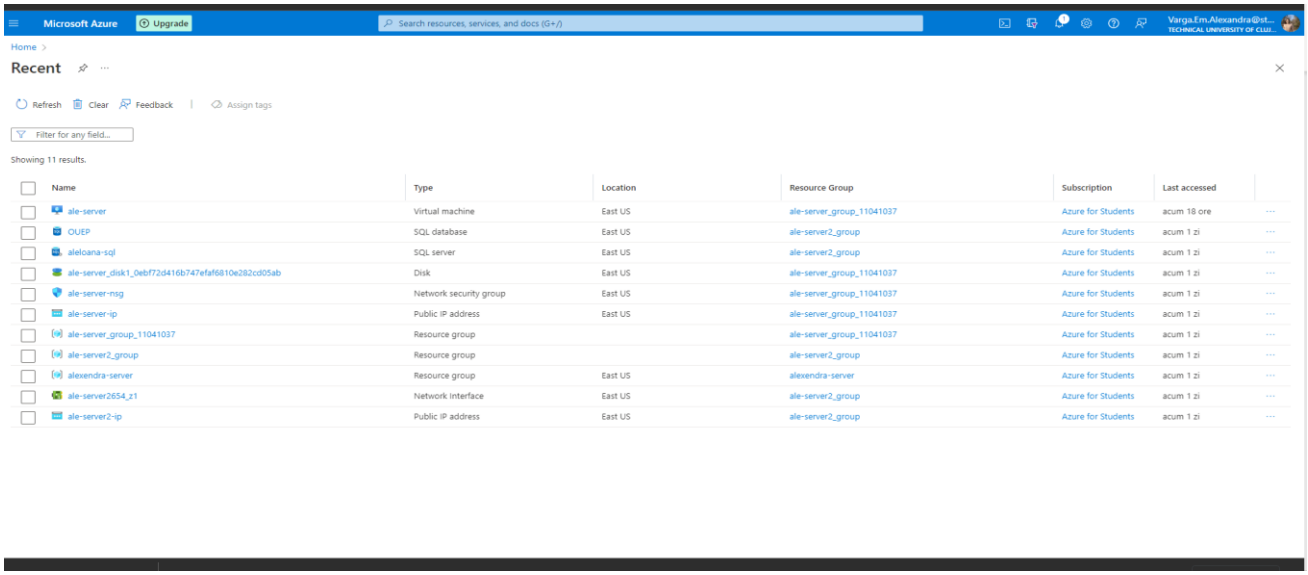
2) Goto 'Appsettings.json' and change the Data Source and Database name as you created in your Local SQL Server

3) Go Sql Server and open the datavase then go to ASPNETROLE Table and Edit it after that add new Role Admin ,Client

4) Then Run the App and put admin username and admin password then you will redirect to your Account of admin

5) After that you can create users and device and perform every task as define in portal

Azure created VMs:



See my deployed application here: http://20.127.152.252/login , credentials: user: Admin, password: Admin