Facultatea de automatică

Și Calculatoare

Sisteme distribuite

Documentație Assigment 1

Online Energy Utility Platform

2022 – 2023

Nume și prenume Student:

Tămaș Ionuț-Valentin

Grupa 30241

1. Arhitectura conceptuală a aplicației

Fiind o aplicație web, aceasta respectă arhitectura client server, unde clientul este reprezentat de browser, iar serverul este implementarea in framework-ul ASP.Net Core a funcționalităților cerute.

Graphical user interface, application

Description automatically generated

Am gândit arhitectura drept una layered cu 4 layere. Primul layer de persistence este reprezentat de baza de date aceasta fiind SQL Server deoarece toate fiind implementate de Microsoft, având suport mai bun între ele. Urmatorul layer, cel de repository, acesta foloseste framework-ul EntityFramework pentru a salva modelele din cod, în baza de date. Urmatorul layer, cel de business se ocupă de logica și interacțiunea dintre diferitele modele, asigurând o desfășurare lină a aplicației. Ultimul layer este cel de presentation; aici am ales sa combin 2 tehnologii: Razor pages de la ASP.Net și frameworkul Vue.JS pe javascript pentru a putea implementa diferite animații și pentru a putea oferi dinamicitatea uzuală a site-urilor moderne.

Diagram

Description automatically generated

1. Diagrama modelului din baza de date

Pentru implementarea modelului, am folosit metoda code-first. Pe frameworkul ASP.Net am folosit Entity Framework pentru crearea migrării modelelor.

Diagram

Description automatically generated

1. Diagrama de deployment

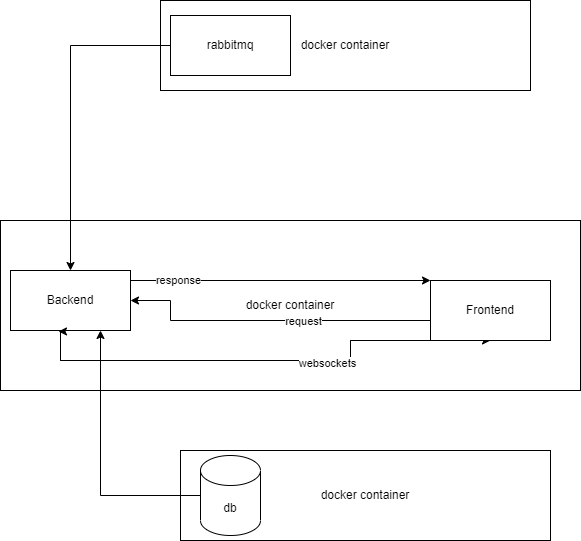
Deploymentul l-am făcut local cu docker. Am folosit imaginea existentă pentru baza de date, iar pentru backend am creat o imagine. Frontendul este integrat pe paginile Razor si este compilat cu WebPack.

Pentru partea de cloud pregătesc să folosesc Azure de la Microsoft.

Diagram

Description automatically generated

Pentru tema 2 am folosit coada de mesaje rabbitmq pentru transmiterea datelor de la simulatorul device-ului. In coada se adauga un string sub forma json in care se stocheaza timestampul, consumul inregistrat si id-ul device-ului. Coada este deploy-ata local pe docker pe porturile 15672 si 5672. Pentru transmiterea mesajelor de eroare in cazul in care consumul depaseste consumul maxim admis de device, am folosit websockets.

Diagrama de deploy actualizata este aceasta:  


Pentru tema 3 am folosit implementarea SignalR pentru transmiterea mesajelor de chat intre client si admin. SignalR foloseste websockets si are support petru Remote Procedure Calls. Am mai creat o pagina in care userii pot sa vada chatul si sa converseze. Userilor le apare cand cineva tasteaza si de asemenea un mesaj cu cine a dat seen.

**Bibliografie**

1. https://dsrl.eu/courses/sd/

2. Lab Book: M. Antal, C. Pop, D. Moldovan, T. Petrican, C. Stan, I. Salomie, T. Cioara, I.

Anghel, Distributed Systems – Laboratory Guide, Editura UTPRESS Cluj-Napoca, 2018

ISBN 978-606-737-329-5, 2018,

https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/329-5.pdf

3. Lab Book: I. Salomie, T. Cioara, I. Anghel, T.Salomie, Distributed Computing and

Systems: A practical approach, Albastra, Publish House, 2008, ISBN 978-973-650-234-7

1. <https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-7.0>
2. <https://vuejs.org/guide/introduction.html>
3. <https://docs.docker.com/get-started/overview/>
4. <https://stackoverflow.com/>
5. <https://learn.microsoft.com/en-us/aspnet/core/signalr/introduction?view=aspnetcore-7.0>
6. <https://www.rabbitmq.com/#getstarted>