

## DOCUMENTATIA TEMEI 2

# Sensor Monitoring System and Real-Time Notification

**Timofte Maximilian**  
**30244**



## Contents

1.	Cerinte Functionale .....	3
2.	Obiective .....	3
2.1.	Obiectiv Principal: .....	3
2.2.	Obiective Secundare: .....	3
3.	Analiza Problemei .....	3
4.	Proiectare .....	3
4.1.	Structuri de date .....	3
4.2.	Diagrama de clase .....	3
4.3.	Algoritmi .....	3
5.	Implementare .....	4
6.	Testare .....	4
7.	Concluzii si Dezvoltari Ulterioare .....	4
8.	Bibliografie .....	4

## 1. Cerințe Functionale

- Implementarea unei aplicații cu 3 module: Producator de mesaje, message broker, message consumer
- Afișarea mesajelor extrase din coada de mesaje din rabbitmq (cel puțin în consola de comandă)

## 2. Obiective

### 2.1. Obiectiv Principal:

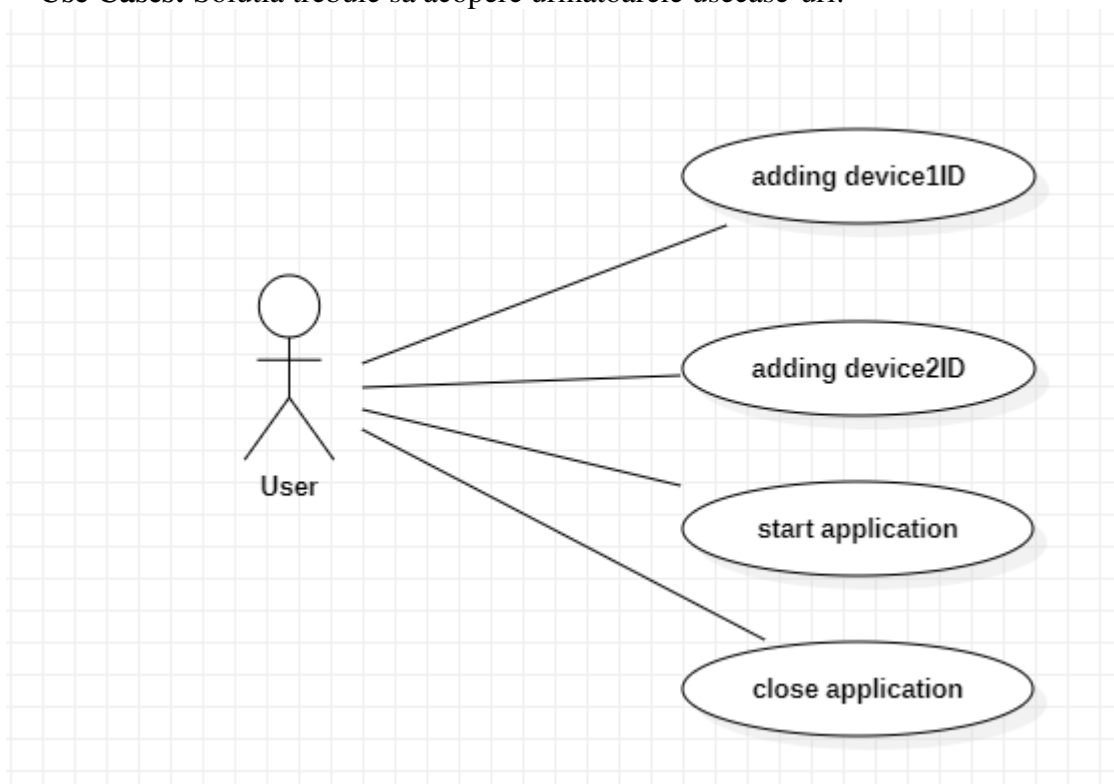
Propuneți, proiectați și implementați un sistem de procesare a senzorilor și de monitorizare a senzorilor prin comunicare asincronă.

### 2.2. Obiective Secundare:

Obiectiv Secundar	Descriere	Capitol
Dezvoltarea de use case-uri și scenarii		3
Alegerea structurilor de date		4
Diagramele		5
Implementarea soluției		6
Testare		7

## 3. Analiza Problemei

**Use Cases:** Soluția trebuie să acopere următoarele usecase-uri:



## 4. Proiectare

### 4.1. Arhitectura solutiei

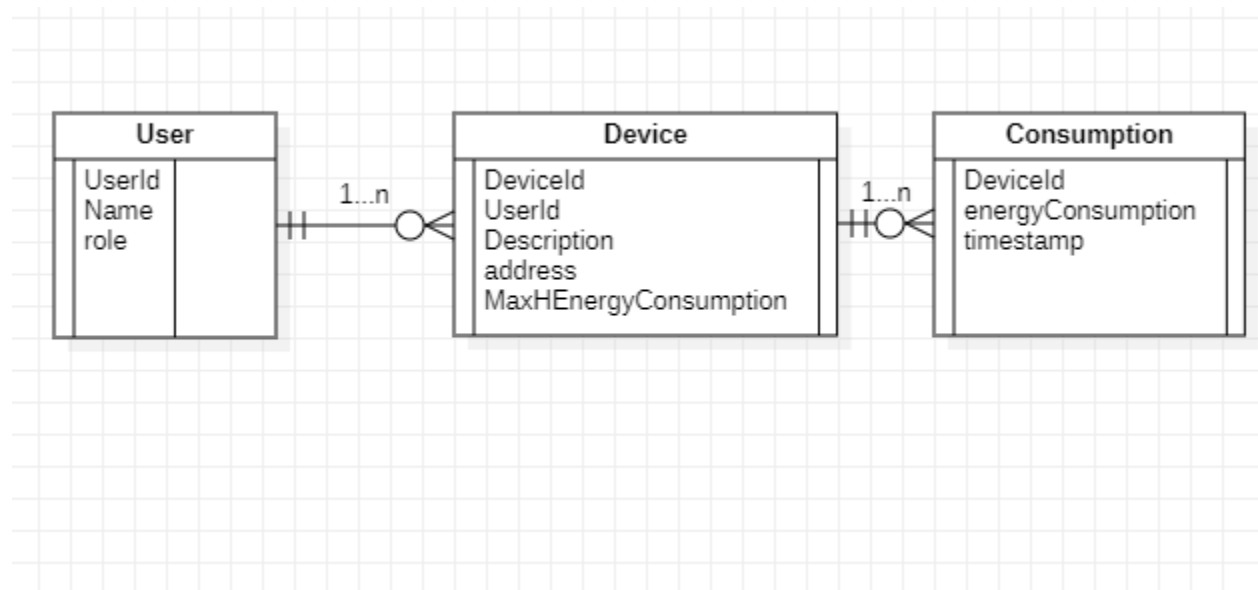
Solutia este formata din 3 componente principale: producers, message broker si consumer.

Cei doi producatori de mesaje reprezinta 2 aplicatii create in framework-ul .net core care citesc valorile dintr-un fisier excel si transmit mesaje cozii rabbitmq.

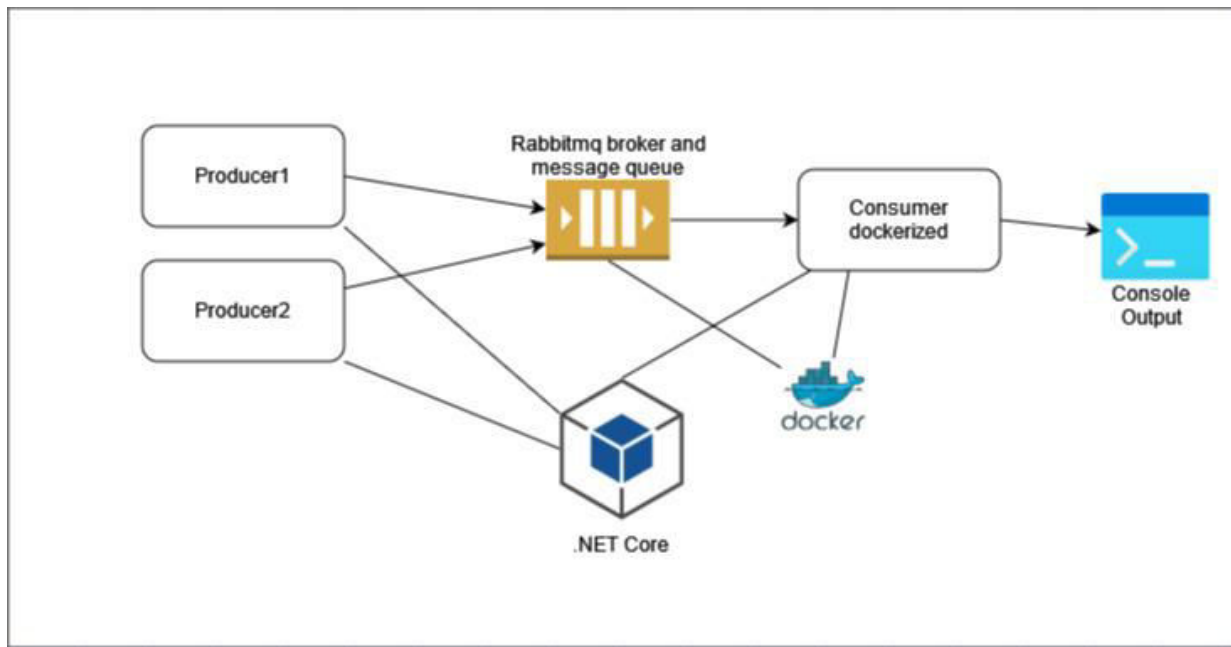
Message broker-ul reprezinta instanta de rabbitmq dockerizata.

Consumer-ul reprezinta aplicatia creata cu framework-ul .net core care citeste respons-urile transmise de message broker si le afiseaza in consola.

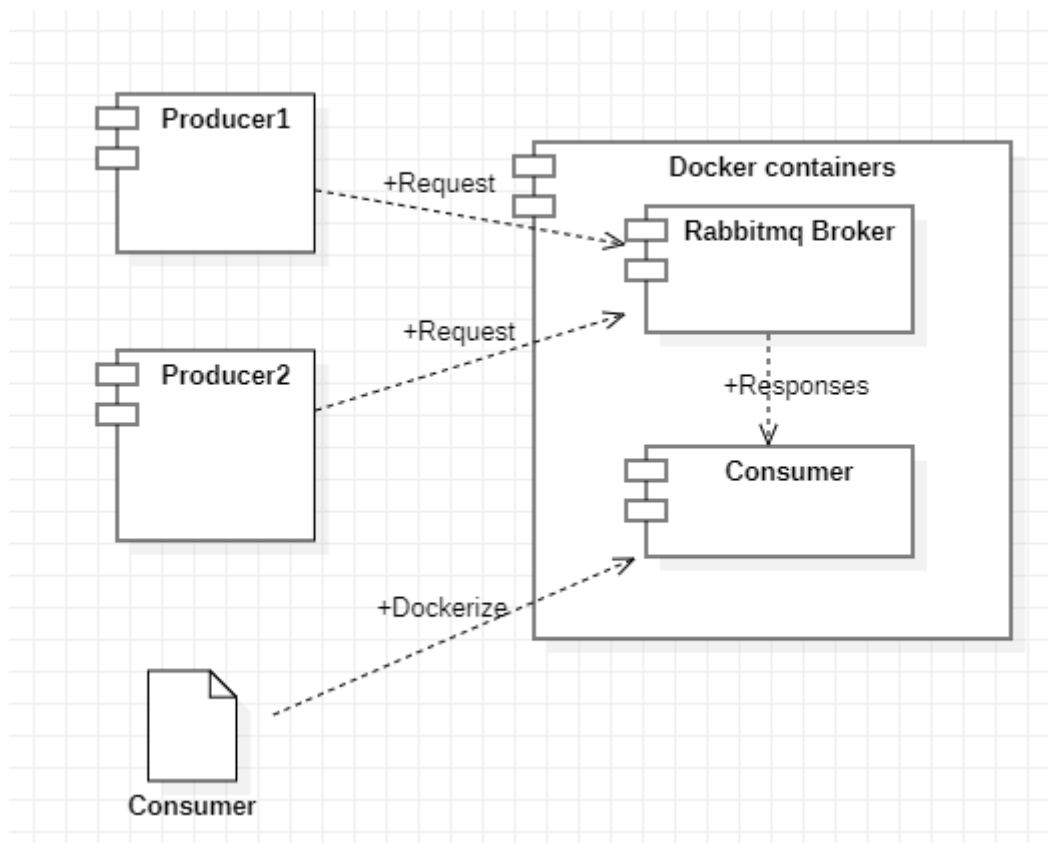
## 5. Implementare



Aceasta este diagrama ER a bazei de date MS SQL.



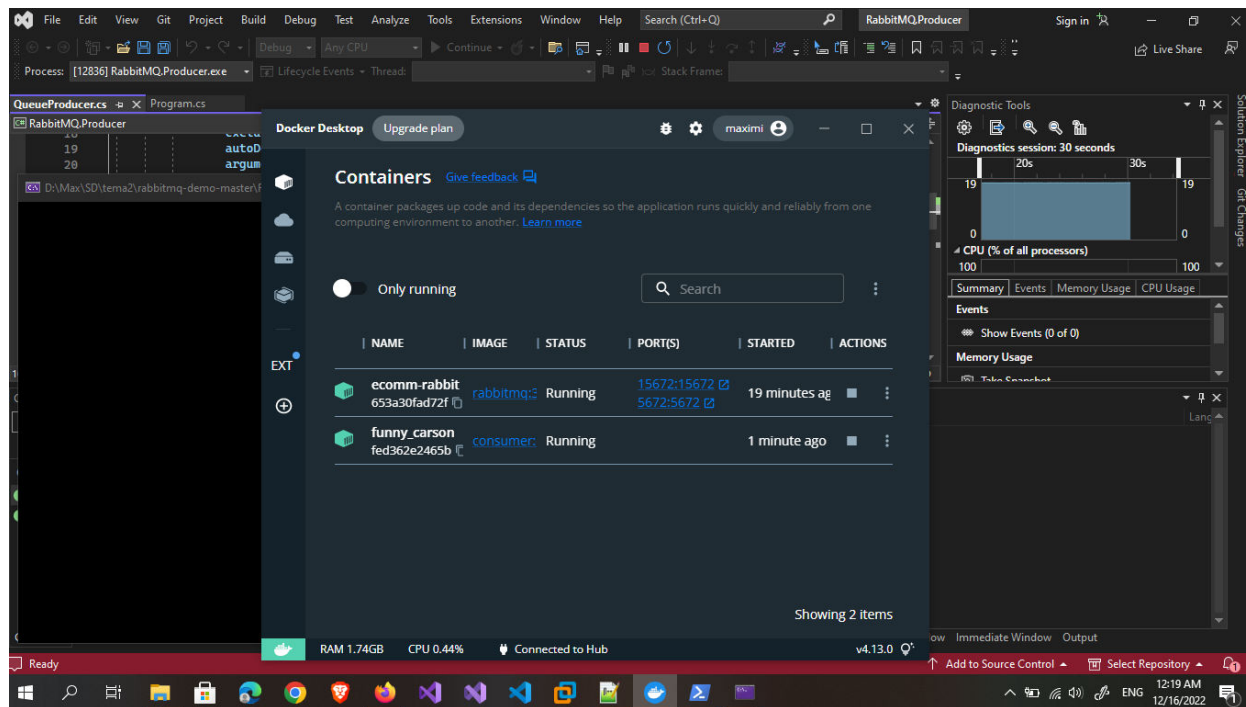
Aceasta este diagrama de arhitectura a intregului proiect.



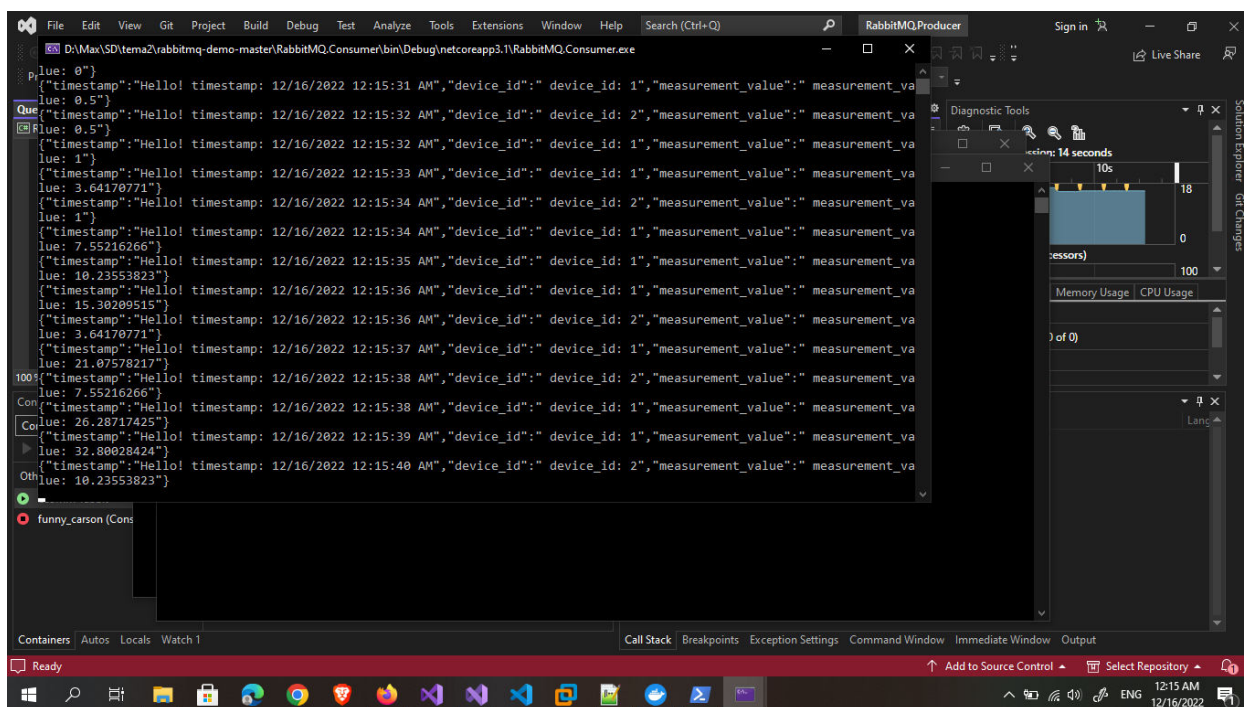


Aceasta este diagrama de deployment a proiectului sau mai bine spus a soluției software oferite.

## 6. Testare



Se poate vedea deploy-ul consumer-ului și a message broker-ului în Docker



functionarea buna a celor doi Producers.

## 7. Concluzii si Dezvoltari Ulterioare

Am dezvoltat o aplicatie capabila de comunicare asincrona pentru monitorizarea senzorilor si a consumului inregistrat de toti acei senzori.

Ceea ce poate fi dezvoltat este incorporare a monitorizarii in angular intr-un format mai user-friendly si crearea de webservice-uri pentru accesarea monitorizarii din localhost sau de pe internet.

## 8. Bibliografie

Google.com

Youtube.com

<https://stackoverflow.com/>

[https://www.w3schools.com/java/java\\_arraylist.asp](https://www.w3schools.com/java/java_arraylist.asp)

<https://www.geeksforgeeks.org>

<https://docs.oracle.com/javase/8/docs/api/java/util>

<https://www.javatpoint.com/>

<https://beginnersbook.com/2013/12>

<https://courses.cs.washington.edu/courses/cse341/98au/java/jdk1.2beta4/docs/api/java/util>

