SISTEM ROBOTIZAT SEMI-AUTOMAT PENTRU TRANSPORTUL MEDICAMENTELOR

Colectivul de elaborare:

Kalapis Eduard-Petru

Lazăr Corina-Mădălina

Mateș Șerban

Miha Andrei-Cristian

Mihai Florentin Constantin

Mocrii Andreea-Alexandra

Muntean Laura-Lavinia

Nichici Miriana

Oltean Emanuel-Liciniu

Cuprins PROIECTARE

[1. Arhitectura programului 3](#_Toc69837853)

[2. Descrierea componentelor (modulelor) 6](#_Toc69837854)

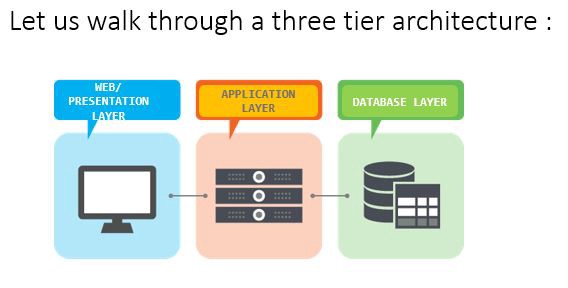
[3. Descrierea comunicării între module 8](#_Toc69837855)

[4. Structuri de baze de date şi fişiere 13](#_Toc69837856)

# Arhitectura programului

**Aplicația Web**:

- Model arhitectural: three tier



Mediu de dezvoltare: Pentru partea frontend vom folosi mediul de dezvoltare Visual Studio Code. Pentru partea de backend vom folosi Visual Studio.

O sa folosim React Typescript, CSS, HTML pentru front-end, iar pentru back-end o sa folosim C#.

C# a apărut ca o versiune îmbunătăţită a limbajele de programare Java, C, şi C++. Cel mai mare avantaj al programării C#, în comparație cu limbajele din care a derivat, este utilizarea mult mai uşoară. Ca să scrieţi un program C# nu aveți nevoie decât de un editor de text simplu ca Notepad şi programul .NET Framework instalat pe calculatorul vostru. Acest program este o mare bibliotecă de instrumente pe care o puteţi folosi cu ajutorul a peste 20 de limbaje de programare.

Creatorii C# au intenţionat să înzestreze limbajul cu mai multe facilităţi. Succesul de care se bucură în prezent, confirmă calităţile sale:

• Este un limbaj de programare simplu, modern, de utilitate generală, cu productivitate mare în programare.

• Este un limbaj orientat pe obiecte.

• Permite dezvoltarea de aplicaţii industriale robuste, durabile.

• Oferă suport complet pentru dezvoltarea de componente software, foarte necesare de pildă în medi distribuite. De altfel, se poate caracteriza C# ca fiind nu numai orientat obiect, ci şi orientat spre componente.

React este o biblioteca JavaScript cu scopul utilizarii pe frontend, folosita pentru construirea interfetelor de utilizator sau componentelor UI.

TypeScript este un limbaj compilat, orientat pe obiecte si foarte restrictiv cand vine vorba de tipuri de date. TypeScript este atat un limbaj de programare cat si un set de instrumente. TypeScript este o versiune de JavaScript cu restrictii, care este compilat in JavaScript. Cu alte cuvinte TypeScript este JavaScript plus alte cateva caracteristici, deci orice cod scris in JavaScript functioneaza fara probleme daca este rulat folosind TypeScript.

CSS sau Cascading Style Sheets este un standard pentru formatarea elementelor unui document Typescript. Stilurile se pot atașa elementelor Typescript prin intermediul unor fișiere externe sau în cadrul documentului, prin element.

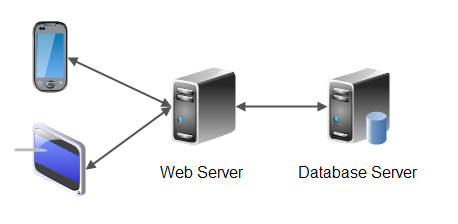
HTML sau hypertext markup language este standard limbajul de markup pentru documentele destinate sa fie afișate într-un browser web. Browserele web primesc documentele HTML de la un server web sau din stocarea locală și redau documentele în pagini web multimedia.

Visual Studio Code este un editor de cod sursă freeware realizat de Microsoft pentru Windows , Linux și macOS. Funcțiile includ suport pentru depanare , evidențierea sintaxei , completarea inteligentă a codului , fragmente , refactorizarea codului și Git încorporat . Utilizatorii pot modifica tema , comenzile rapide de la tastatură , preferințele și pot instala extensii care adaugă funcționalități suplimentare.

Visual Studio include un set complet de instrumente de dezvoltare pentru generarea de aplicații ASP.NET, Servicii Web XML, aplicații desktop și aplicații mobile. Visual Basic, Visual C++, Visual C# și Visual J# toate folosesc același mediu de dezvoltare integrat (IDE) care le permite partajarea instrumentelor și facilitează crearea de soluții folosind mai multe limbaje de programare

**Aplicația mobilă:**

- Model arhitectural: three tier



Arhitectura N-tier oferă un model după care dezvoltatorii pot crea aplicații flexibile și reutilizabile. Doar segmentând aplicațiile în niveluri, dezvoltatorii pot obține opțiunea de a modifica sau de a adauga un anumit strat, în loc de a reface întreaga aplicație de la început.

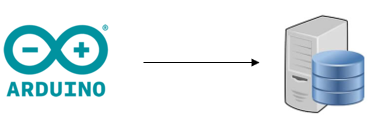
O arhitectură pe 3 niveluri este în general compusă dintr-un nivel de prezentare, un nivel pentru domeniul logic și un nivel pentru stocarea datelor.

Mediul de dezvoltare folosit va fi Xamarin.

**Xamarin** este o tehnologie open source care permite compilarea unei baze de cod C# comune în aplicații native iOS, Android sau Windows. Avantajele utilizării aceluiași mediu de dezvoltare (IDE - Integrated Development Environment), aceluiași limbaj de programare (C#) și aceluiași set de API-uri (Application Programming Interfaces) pentru toate platformele, atrag un număr din ce în ce mai mare de dezvoltatori. Prin metodele de interfațare cu sistemele de operare native pe care le asigură, Xamarin permite dezvoltatorilor să obțină cu C# practic aceleași rezultate pe care le-ar obține separat cu Objective-C, Swift sau Java. Ceea ce este important de subliniat în acest context este caracterul nativ al aplicațiilor rezultate, un element important de diferențiere al Xamarin față de alte tehnologii multiplatformă (cum ar fi de exemplu cele bazate pe HTML5 și JavaScript).

**Modul comandă robot:**

- Model arhitectural: client-server

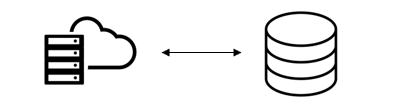


Mediul de dezvoltare utilizat va fi Arduino IDE.

Arduino este un instrument prin care se pot realiza sisteme informatice capabile să “perceapă” şi să “controleze” mediul înconjurător. Acest instrument este open-source şi este compus dintr-un mediu de dezvoltare ( Arduino IDE) şi o placă de dezvoltare cu microcontroller. Arduino poate fi folosit pentru dezvoltarea de obiecte interactive. Informaţia este preluată de la o gamă variată de senzori şi comutatoare, se procesează în interiorul microcontrollerului , şi este transmisă către o gamă la fel de variată de lumini, motoare, actuatoare etc.

C este un limbaj de programare standardizat. Este implementat pe majoritatea platformelor de calcul existente azi, și este cel mai popular limbaj de programare pentru scrierea de software de sistem. Este apreciat pentru eficiența codului obiect generat de compilatoarele C, și pentru portabilitatea sa.

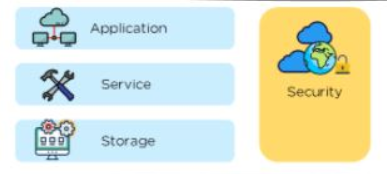
**Modulul Cloud:**



-Mediul de dezvoltare folosit va fi Microsoft Azure.

Azure este o platformă profesională de servicii în cloud, oferită de Microsoft. Datorită funcționalităților avansate, poți porni mașini virtuale, baze de date SQL, poți crea copii de rezervă pentru resursele existente și poți face multe altele, fără a te teme de defecțiuni sau de învechirea echipamentelor și a programelor informatice.

-Arhitectura programului



• Componenta aplicație: Comunicarea cu aplicația web și aplicația mobilă pentru a primi datele de autentificare ale medicilor și asistentelor, dar și datele despre pacienți și pentru a trimite erorile și notificările ajunse de la robot în cloud.

• Componenta service: Menținerea în permanență a datelor în online și asigurarea integrității acestora.

• Componenta stocare: Memorarea tuturor datelor de autentificare ale medicilor, asistentelor, dar și datele despre pacienți

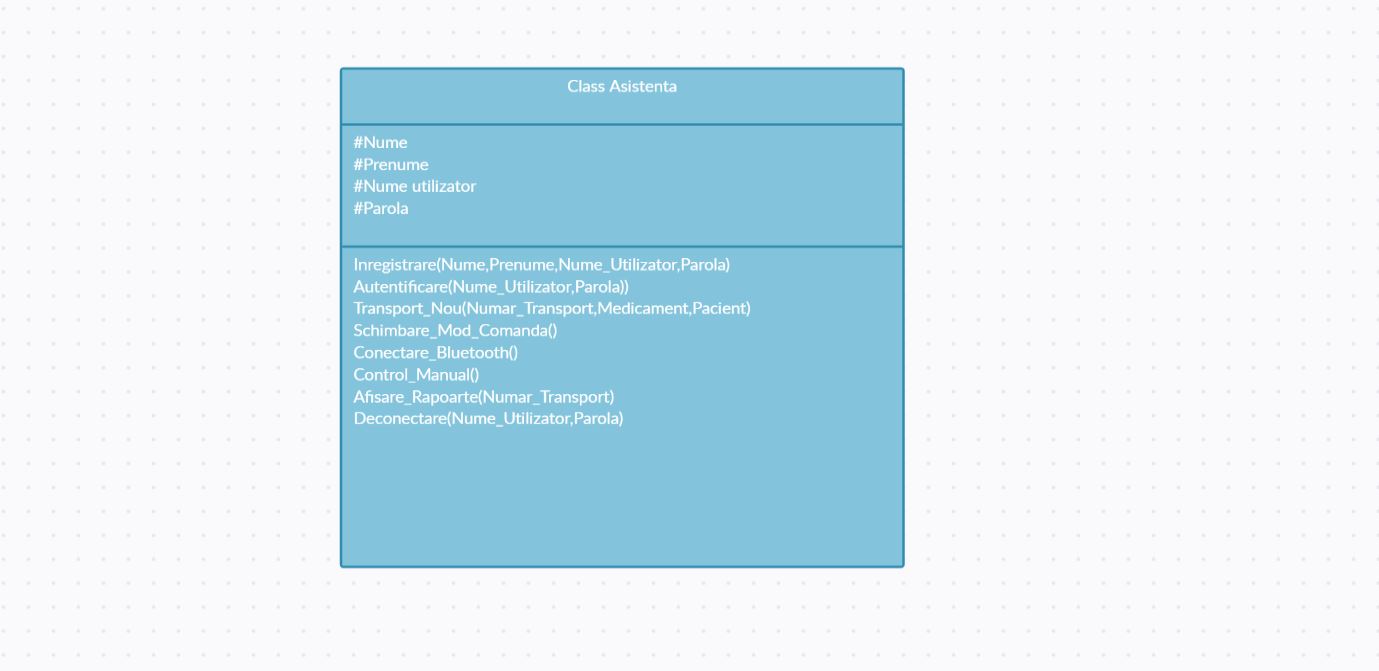
• Componenta securitate: Menținerea în siguranță a datelor medicilor, asistentelor, dar și a parolelor de utilizatori ale acestora.

# Descrierea componentelor (modulelor)

Table

Description automatically generated with medium confidence

**Aplicația mobilă:**

****

Principalele funcționalități :

1. ***Înregistrare***

* Intrări: -Nume

-Prenume

-Nume utilizator

-Parolă

-Buton pentru creare cont

* Prelucrări: -Înregistrare în baza de date a datelor utilizatorului nou

-Afișare mesaj eroare în cazul în care numele de utilizator există deja

-Afișare mesaj eroare în cazul în care parola este prea slabă

* Ieșiri:-Contul nou creat și accesibil pentru autentificare

1. ***Autentificare***

* Intrări:-Nume utilizator

-Parolă

-Buton pentru autentificare

* Prelucrări: -Validarea datelor de autentificare

-Mesaj de eroare în care numele de utilizator sau parola sunt greșite

* Ieșiri:Se permite accesul la meniul principal pentru asistenta care sa autentificat precum si accesul la functionalitătile aplicației

1. ***Inițiere transport***

* Intrări-Buton specific pentru comandă

-Număr transport

-Medicamente

-Pacient

-Buton pentru efectuare transport

* Prelucrări: -Validare date introduse de asistenta
* Ieșiri:-Se va efectua transportul

1. ***Schimbare mod comandă***

* Intrări:-Buton specific pentru comandă

-Casete de selecție pentru control automat sau manual

-Buton pentru validare

* Prelucrări : Validarea casetei de selecție
* Ieșiri:Se va selecta modul de control și va ramane neschimbat pana la urmatoarea accesarea a acestei funcționalități

1. ***Conectare Bluetooth***

Intrări:

Prelucrări:

Ieșiri:Se va conecta robotul la telefon prin interemediul Bluetooth-ului

1. ***Control manual***

* Intrări:-Buton specific pentru comandă

-4 săgeți pentru control

* Prelucrări : -Se va valida comanda fiecarei sageți
* Ieșiri:Robotul se va mișca în direcția indicată de săgeată

1. ***Afișare rapoarte***

* Intrări:-Buton specific pentu comandă

-Număr transport

Buton pentru afișare detalii transport

* Prelucrări:Validarea numărului de transport
* Ieșiri:Se vor afișa detaliile transportului, acestea fiind:

-Număr transport

-Pacient

-Medicament

-Asistentă

-Avarii apărute

**8) Deconectare**

* Intrări:-Buton specific pentru comandă

-Nume utilizator

-Parolă

* Prelucrări:Validarea numelui de utilizator si parolei
* Ieșiri:Se va ieși din meniul principal și se va reveni la fereastra de autentificare

**9) get\_date**

* Intrari:-Numele pacientului
* Ieșiri:Se vor afișa detaliile fiecărui Pacient, acestea fiind:
  + CNP
  + Nume
  + Prenume
  + Varsta
  + Sex
  + Telefon
  + Profesie
  + Pat
  + Diagnostic
  + Medicamente

***10)Autentificare***

* Intrări:-Nume

-Parolă

-Parafa

* Prelucrări: -Validarea datelor de autentificare

-Mesaj de eroare în care numele de utilizator sau parola sunt greșite

* Ieșiri:Se permite accesul la meniul principal

***11)Inregistrare***

* Intrări:-Nume

-Prenume

-Nume\_utilizator

-parola

-cod\_parafa

* Prelucrări: -Înregistrare în baza de date a datelor utilizatorului nou

-Afișare mesaj eroare în cazul în care numele de utilizator există deja

-Afișare mesaj eroare în cazul în care parola este prea slabă

* Ieșiri:-Contul nou creat și accesibil pentru autentificare

***12)Adaugare\_pacient***

* Intrări:-CNP

-Prenume

-Nume\_pacient

-parola

-cod\_parafa

-varsta

-sex

-telefon

-profesie

-pat

-diagnostic

-medicamente

* Prelucrări: -Înregistrare în baza de date a datelor pacientului nou

-Afișare mesaj eroare în cazul în care numele de pacient există deja

-Afișare mesaj eroare în cazul în care parola este prea slabă

* Ieșiri:-Datele pacientului nou adaugat

***12)Cautare\_pacient***

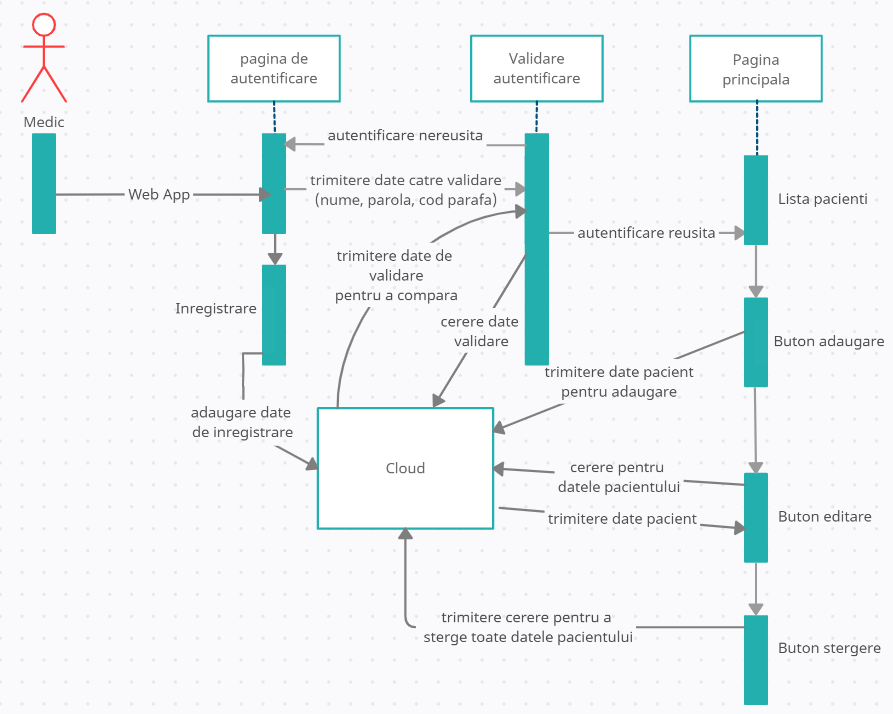
* Intrări: -Nume\_pacient
* Prelucrări: -Afișare mesaj eroare în cazul în care numele de pacient nu există

Ieșiri:-Datele pacientului cautat

# Descrierea comunicării între module

**Aplicația Web**:

**Diagrama de comunicare Web-Cloud**



Medicul, utilizatorul aplicației web, va interacționa prima data cu **pagina de autentificare**. Unde există doar două cazuri posibile: medicul are cont înregistrat în baza de date sau nu are cont creat.

În cazul în care medicul nu are un cont, va apăsa pe butonul cu numele **Înregistrare**. Prin urmare, acesta va introduce datele: nume de utilizator, codul parafa si parola care vor fi transmise online către serverul create. Dacă pacientul introduce un nume de utilizator existent în baza de date, acesta va fi avertizat și îndemnat să introducă un alt utilizator. Astfel, se face trecerea la cel de-al doilea caz.

În cel de-al doilea caz, când medicul dorește să se autentifice, acesta va introduce numele de utilizator,codul de parafa și parola în câmpurile corespunzătoare. Dacă se efectuează asocierea între datele introduse de utilizator și datele existente în Cloud, atunci autentificarea se realizează cu succes.

În cazul în care medical s-a autentificat cu succes, acesta va ajunge la **pagina principală** unde vede lista cu pacienții. În acest moment el are trei opțiuni: să adauge, să editeze sau să șteargă un pacient.

Butonul **Adaugare pacient** permite medicului sa insereze datele unui pacient(nume , prenume, CUI, CNP).

Butonul **Stergere pacient** permite medicului sa elimine un pacient din listă.

Butonul **Editare pacient** permite medicului sa modifice datele unui pacient(nume , prenume, CUI, CNP).

**Aplicația mobilă:**

1. ***Înregistrare***

Dacă există o asistentă nouă aceasta își va crea un cont nou și își va introduce numele, prenumele, numele de utilizator și parola pe care o va folosi. Aceste date urmează să fie stocate în **baza de date**.

1. ***Autentificare***

În cazul în care asistenta are deja cont, ea își va adăuga numele de utilizator și parola și va exista o **funcție care verifică în baza de date** dacă utilizatorul există și dacă parola introdusă este corectă. Dacă este totul corect, se va intra în meniul principal, altfel se va semnala eroare.

1. ***Inițiere transport***

Asistenta va putea să creeze un transport nou prin care se va folosi numărul transportului, ce medicament este necesar, dar și pacientul. Aceste date vor fi deasemenea stocate în **baza de date.**

1. ***Conectare Bluetooth***

Aplicația mobilă se conectează la bluetooth, asftel, realizându-se conexiunea între telefon și ***robot.***

1. ***Schimbare mod comandă***

Asistenta va alege ori modul de comandă manual, ori modul de comandă automat( se deplasează singur urmărind harta, ocolește obstacole), iar **robotul** va funcționa după cum a selectat asistenta.

1. ***Control manual***

***Robotul*** va fi controlat de asistentă prin aplicația mobilă să execute comenzi de înainte, înapoi, stânga, dreapta, deasemnea având funcția de a detecta singur obstacolele.

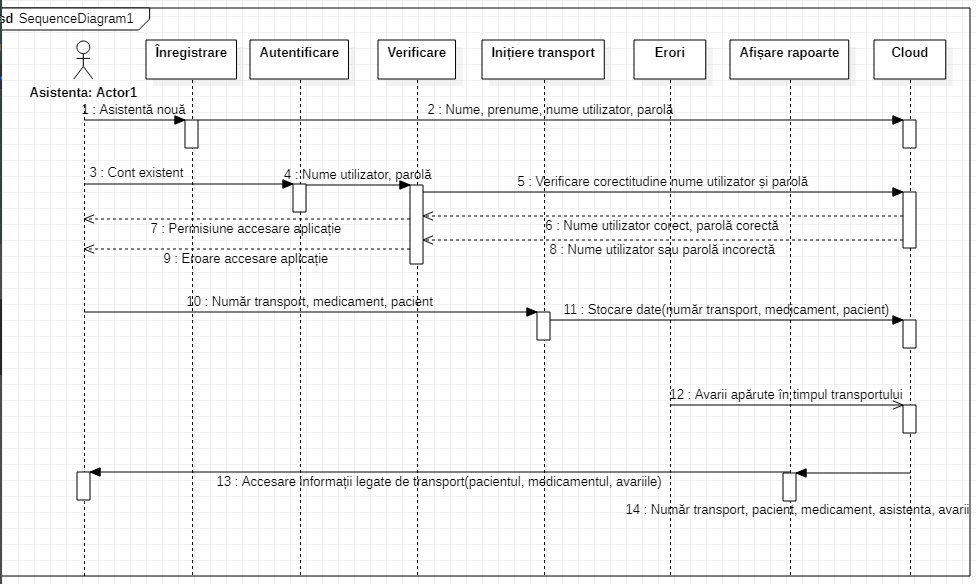
1. ***Erori***

Dacă în timpul transportului apar erori( nu găsește patul, pierde medicamentul) sau avarii(se termina bateria, se pierde conexiunea), **robotul** le va transmite aplicației mobile, apoi acestea vor fi stocate în **baza de date**.

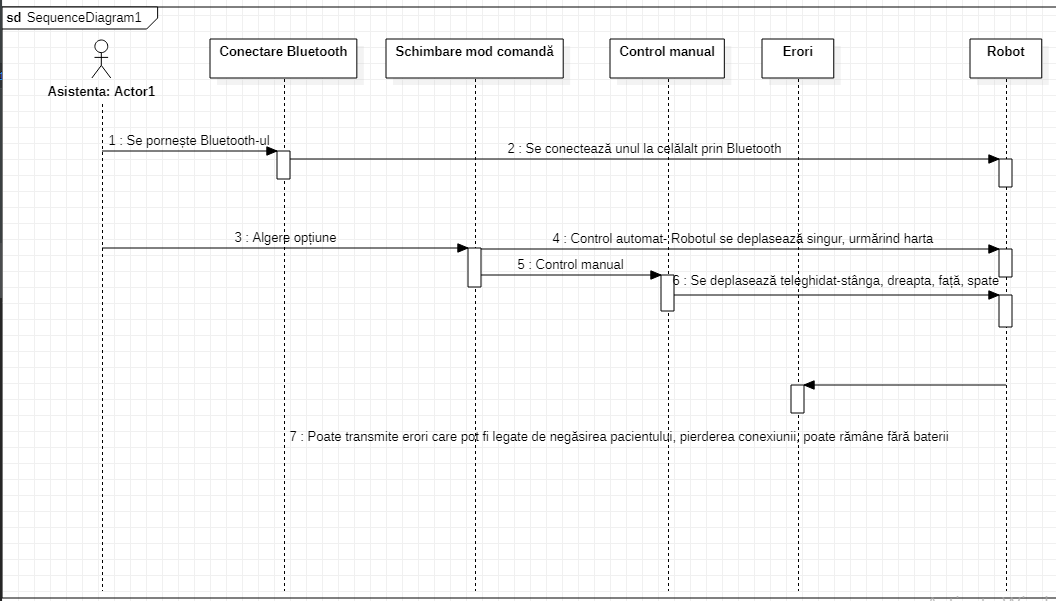
1. ***Afișare rapoarte***

Aceste rapoarte sunt accesate prin selectarea unui transport, iar datele pe care asistenta le poate vedea sunt: numărul transportului, pacientul, medicamentul, asistenta sau doctorul care a dat inițiat acel transport, iar deasemenea daca a aparut avarii sau nu și descrierea lor. Toate aceste rapoarte sunt preluate din **baza de date**.

**Diagrama de secvență pentru comunicarea Aplicație mobilă- Cloud**



**Diagrama de secvență pentru comunicarea Aplicație mobilă- Robot**



**Modul comandă robot:**

1) Conectare Bluetooth

Activarea funcției Bluetooth a robotului permite realizarea conexiunii între acesta și aplicația mobilă.

2) Control manual

Robotul va fi controlat manual de cadrul medical cu ajutorul aplicației mobile. Acesta se va putea deplasa înainte, înapoi, la stânga, la dreapta în funcție de controalele apăsate sau se va putea opri, la comandă sau când va întâlni un obstacol.

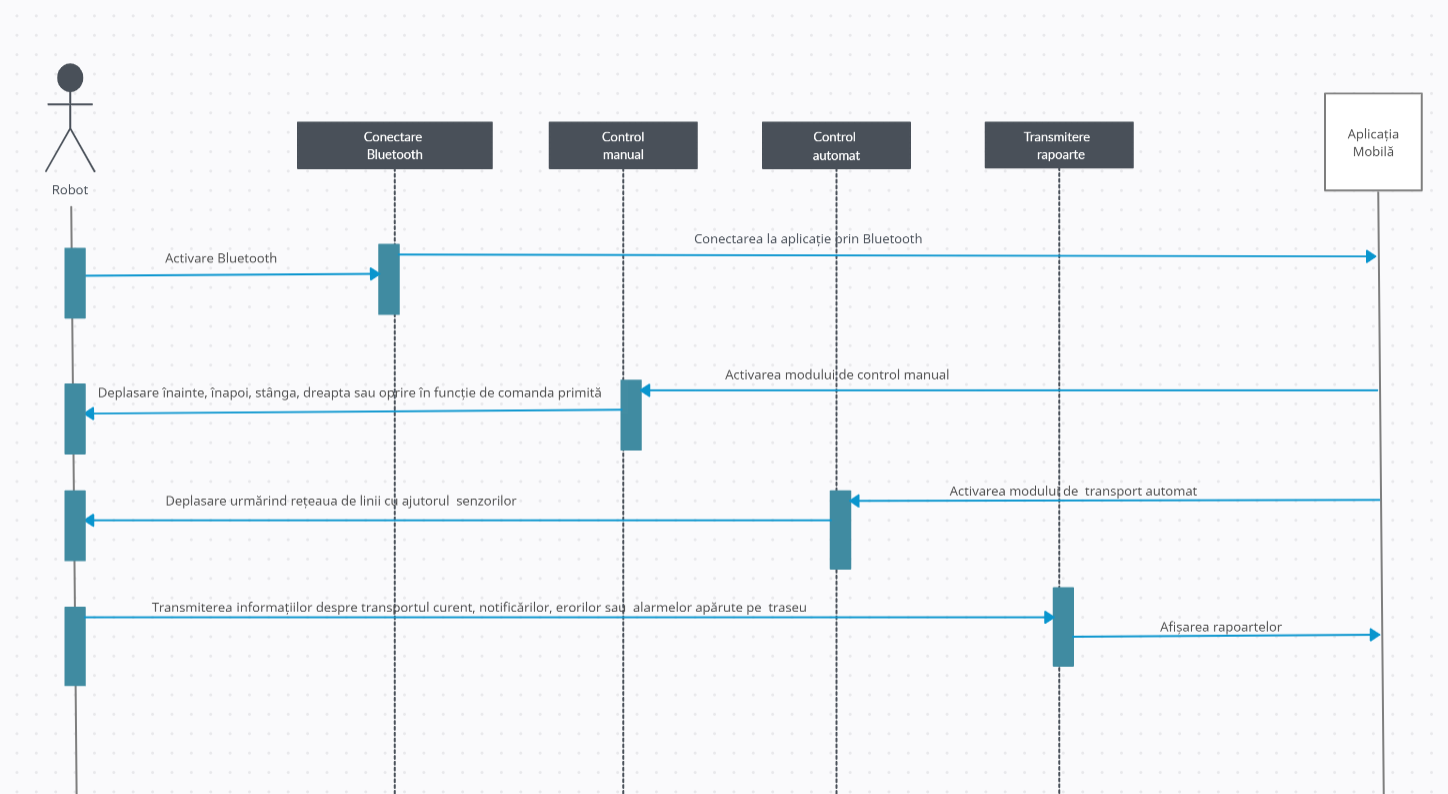
3) Rapoarte

Robotul poate transmite aplicației mobile rapoarte referitoare la transportul curent, care să conțină numele pacientului, patul, medicația sau statusul comenzii.

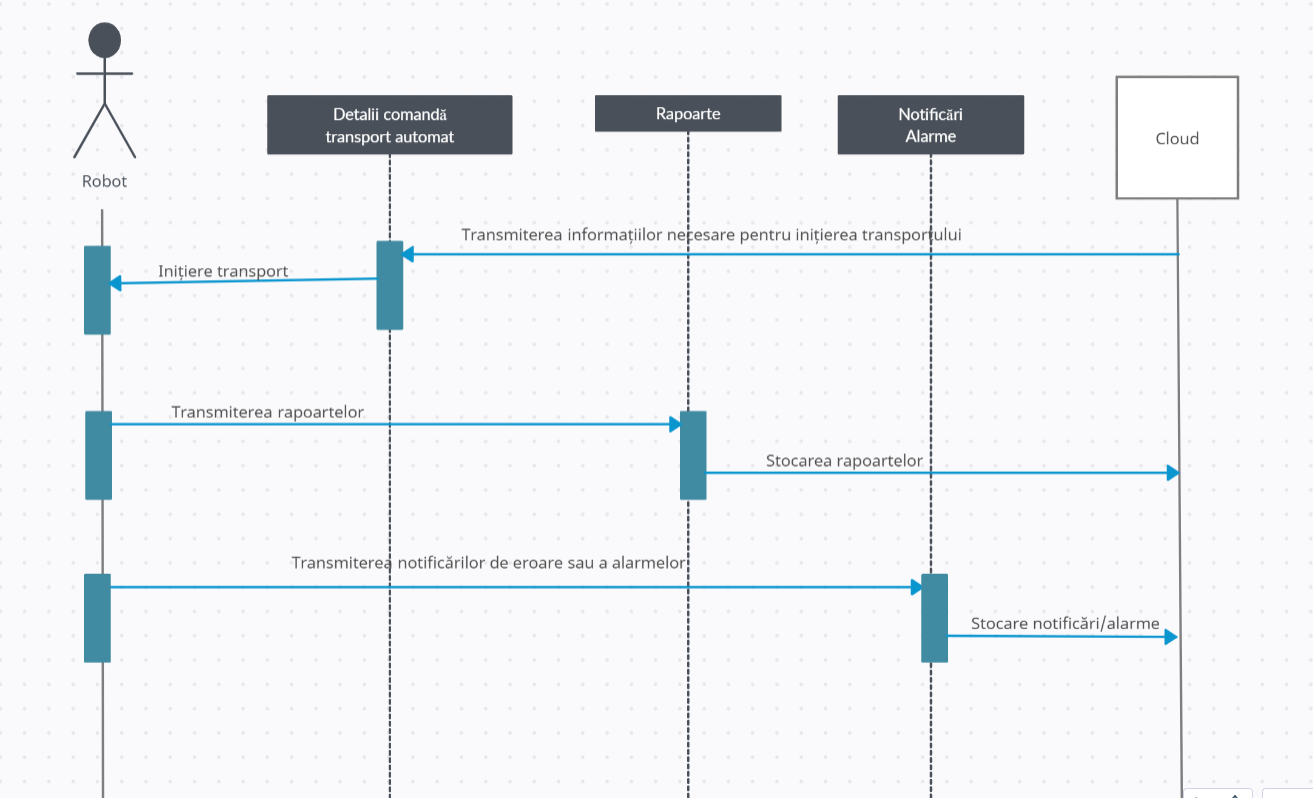
4) Notificări de eroare/Alarme

Robotul poate transmite către cloud mesaje de eroare (ex. nefinalizarea transportului) sau alarme apărute pe parcurs (traseu blocat, pierdere conexiunii)

Comunicarea Robot-Aplicație mobilă



Comunicarea Robot-Cloud



**Modulul Cloud:**

Chart, box and whisker chart

Description automatically generated

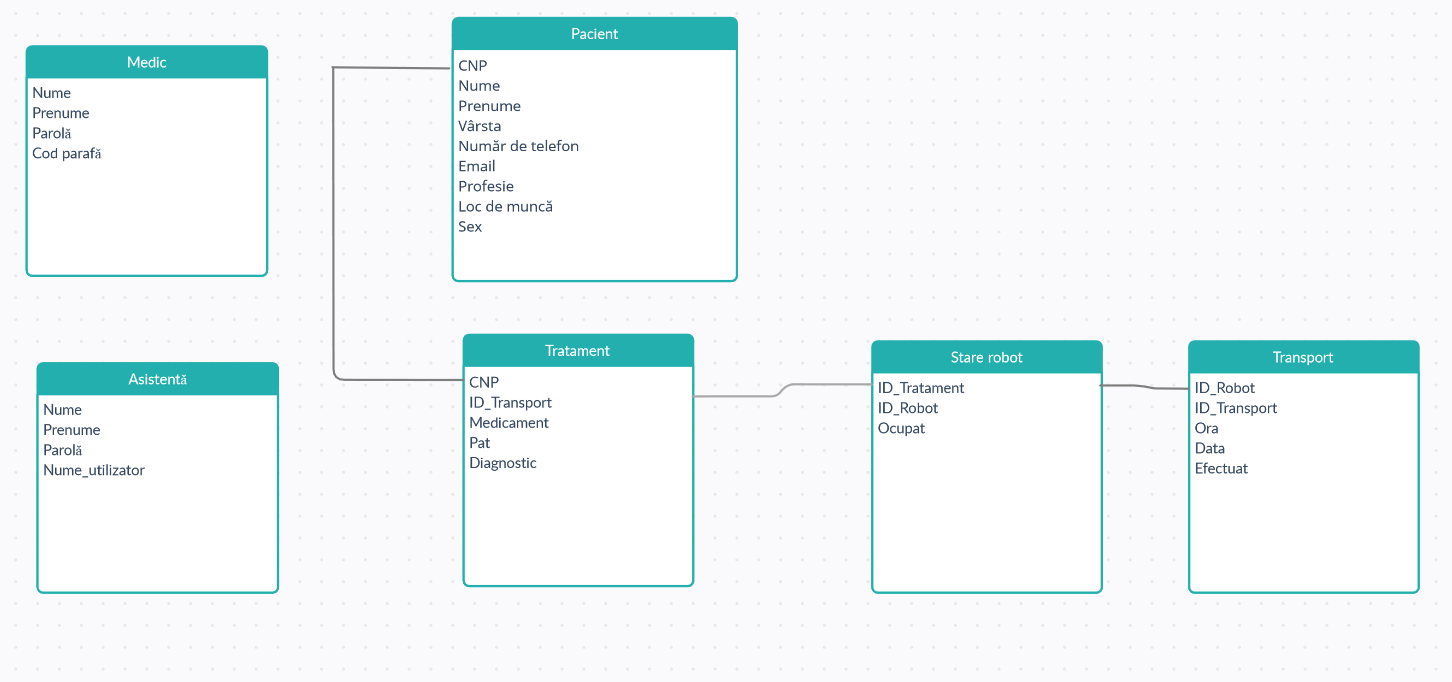
Chart, box and whisker chart

Description automatically generated

Chart, waterfall chart, box and whisker chart

Description automatically generated

# Structuri de baze de date şi fişiere



Baza de date va conține următoarele tabele:

* Tabela Medic cu următoarele câmpuri:
  + Nume (numele medicului): Varchar (50) NOT NULL
  + Prenume (prenumele medicului): Varchar (50) NOT NULL
  + Parolă (parola de autentificare): Varchar (50) NOT NULL
  + Cod parafă (parafa medicului): Varchar (50) PRIMARY KEY
* Tabela Asistentă cu următoarele câmpuri:
  + Nume (numele asistentei): Varchar (50) NOT NULL
  + Prenume (prenumele asistentei): Varchar (50) NOT NULL
  + Parolă (parola de autentificare): Varchar (50) NOT NULL
  + Nume\_utilizator (numele de utilizator cu care se autentifică): Varchar (50) PRIMARY KEY
* Tabela Pacient cu următoarele câmpuri:
  + CNP (codul numeri personal al pacientului): Varchar (13) PRIMARY KEY
  + Nume (numele pacientului): Varchar (50) NOT NULL
  + Prenume (prenumele pacientului): Varchar (50) NOT NULL
  + Vârsta (vârsta pacientului): INT NOT NULL
  + Număr de telefon (numarul de telefon al pacientului): Varchar (50) NOT NULL
  + Email (email-ul pacientului): Varchar (50)
  + Profesie (cu ce se ocupă pacientul): Varchar (50)
  + Loc de muncă (unde lucrează pacientul): Varchar (50)
  + Sex (sexul pacientului): Varchar (50) NOT NULL
* Tabela Tratament cu următoarele câmpuri:
  + CNP: Varchar (13) REFERENCES Pacient(CNP) ON DELETE CASCADE
  + ID\_Tratament: INT PRIMARY KEY
  + Medicament (medicamentul care trebuie livrat): Varchar (50) NOT NULL
  + Pat (patul pacientului unde se livrează): Varchar (50) NOT NULL
  + Diagnostic (diagnosticul pacientului): Varchar (50) NOT NULL
* Tabela Stare robot cu următoarele câmpuri:
  + ID\_Robot (id-ul pentru starea robotului): INT PRIMARY KEY
  + ID\_Tratament INT REFERENCES Tratament(ID\_Tratament) ON DELETE CASCADE
  + Ocupat (starea robotului într-un anumit timp care poate fi 1-ocupat sau 0-liber): BIT NOT NULL
* Tabela Transport cu următoarele câmpuri:
  + ID\_Transport (id-ul fiecărui transport) INT PRIMARY KEY
  + ID\_Robot INT REFERENCES Stare robot(ID\_Robot) ON DELETE CASCADE
  + Ora (ora la care se face transportul) TIME NOT NULL
  + Data (data în care se face transportul) DATE NOT NULL
  + Efectuat (starea transportului care poate fi 1-transport efectuat și 0-neefectuat) BIT NOT NULL