

# Documentație tehnică

## Monitorizarea traficului - Proiect propus de Continental

Enia Vlad Ieftimie

Facultatea de Informatică - Universitatea Alexandru Ioan Cuza” din Iași

**Abstract.** Acest document va întocmi raportul tehnic al proiectului **Monitorizare trafic**, propus de compania **Continental** în cadrul materiei **Rețele de Calculatoare** a Facultății de Informatică. Conținutul acestui document include o introducere a proiectului și a funcționalităților pe care mi-am propus să le implementez, tehnologiile utilizate și scopul acestora, arhitectura aplicației, împreună cu diverse detalii de implementare și scenarii de utilizare, dar și o concluzie care va conține, printre altele și câteva idei de îmbunătățire pe viitor a acestui serviciu care să concluzioneze raportul tehnic.

**Keywords:** Monitorizare trafic · Alerte trafic · Client/Server · TCP · Baze de date · Fork() · Threaded

## 1 Introducere

În această secțiune voi cuprinde o descriere generală a proiectului, împreună cu funcționalitățile și serviciile oferite de acesta, dar și modul de utilizare și alte aspecte generale.

### 1.1 Actori

- **Utilizatori:** ei vor reprezenta clienții din modelul arhitectural Client/Server, folosit în implementarea acestui proiect; ei vor primi primi la un anumit interval de timp diverse informații legate de starea traficului dar și a infrastructurii, pe baza informațiilor trimise de ei către Server;
- **Sistem Server:** acesta așteaptă mai multe conexiuni din partea utilizatorilor și interpretează în timp real datele furnizate de aceștia pentru ca și el, la rândul lui, să trimită trimită înapoi clienților informații relevante legate de trafic și infrastructură.

### 1.2 Servicii

Pentru ca **utilizatorii** să poată beneficia de serviciile oferite de **server**, ei mai întâi trebuie să realizeze o conexiune cu acesta. În urma realizării cu succes a conexiunii, ei vor furniza, în mod automat, la un interval de un minut, date legate de locația acestora și viteza cu care circulă.

Totodată, clienții vor preciza, atunci când inițiază cererea de conexiune, dacă optează pentru a primi informații suplimentare precum starea vremii, prețul combustibilului la stațiile de alimentare sau evenimente sportive din acea zi.

Aceste date vor fi interpretate în timp real de către server iar pe baza acestora el va putea oferi diverse servicii clienților, cum ar fi:

- **Viteza recomandată** pentru acea porțiune de drum, încadrată în limitele legale; aceasta poate varia în funcție de diverși factori precum:
  - **Localitate** - dacă șoseaua pe care clientul circulă conține porțiuni în afara localității și/sau în localitate, acesta va fi notificat în legătură cu limitele legale impuse: 50km/h în localități, 90km/h drumuri naționale, 100km/h drumuri național-europene, 130km/h autostradă;
  - **Instituții de învățământ** - dacă utilizatorul se află în localitate, circulând pe o stradă pe care se află o instituție de învățământ, viteza legală este 30km/h pe acea porțiune;
  - **Drumuri neasfaltate** - 30km/h;
  - **Drumuri offroad** - 20km/h;
  - **Zone rezidențiale** - 30km/h.
- **Alerte** raportate de alți utilizatori; fiecare client va putea raporta un eveniment ce a avut loc pe șoseaua pe care se află; acest raport va fi trimis de către server tuturor utilizatorilor sub forma unei alerte prestabilite, precum:
  1. **Accident;**
  2. **Polizie;**
  3. **Cameră de viteză;**
  4. **Obstacol mic** - gropi, denivelări, canale, animale sau orice alt obstacol de dimensiuni mici ce poate fi evitat;
  5. **Obstacol mare** - orice obstacol de dimensiuni mari, imobil, ce nu poate fi evitat, fapt ce rezultă în blocarea traficului pe acea porțiune de drum;
  6. **Lucrări;**
  7. **Ambuteiaj.**
- **Notificări** - dacă aceștia optează de la început, utilizatorii vor fiificați în legătură cu diverse aspecte, precum:
  1. **Starea vremii;**
  2. **Prețurile carburantului;**
  3. **Evenimente sportive.**

## 2 Tehnologii utilizate

### 2.1 Protocol de transport: TCP

În implementarea acestui proiect am ales să utilizez TCP drept protocol de transport din mai multe motive.

În primul rând, fiecare utilizator va stabili la început o conexiune cu serverul, autentificându-se folosind un user unic, pentru ca ceilalți utilizatori să poată vede alerta de atenționare raportată pe un anumit drum drum.

Mai mult decât atât, serverul va menține constantă conexiunea cu fiecare dintre clienți, deoarece aceștia vor trimite, în mod automat, cu o frecvență de 1 minut, locația acestora împreună cu viteza cu care circulă. Serverul, la rândul lui, va răspunde fiecărui utilizator cu informații în funcție de datele furnizate de aceștia.

Așadar, deoarece protocolul TCP ne oferă o comunicare mai sigură și fiabilă decât UDP, de exemplu, am considerat că pentru un utilizator al unui astfel de serviciu este mai important ca informațiile și alertele de atenționare să ajungă corect, decât să existe lipsuri sau erori.

## 2.2 Metodă de stocare a informațiilor: Baze de date MySQL

Serviciul de monitorizare a traficului al acestui proiect va fi implementat zona județului Iași. Deoarece localizarea se realizează la nivel de stradă, șosea, drum, bulevard etc., va fi necesară stocarea numelor acestora, împreună cu alte informații relevante care pot interveni în recomandarea de către server a unei viteze pentru fiecare utilizator în parte.

Așadar, pentru stocarea drumurilor, va fi utilizat un tabel din cadrul unei baze de date MySQL cu următoarele atribute:

- **ID** int - câmp de tip cheie primară, utilizat pentru indexarea tuturor drumurilor introduse în tabel;
- **Nume** varchar - numele străzii, șoselei, bulevardului, drumului etc.;
- **Loc\_in** bit - 1 dacă vreo porțiune a drumului respectiv se află în localitate, 0 dacă nu conține porțiuni care să fie în localitate;
- **Loc\_out** bit - analog ca la atributul Loc\_In, doar că pentru afara localității; a se observa că un drum poate conține porțiuni atât în localitate, cât și în afară
- **Sc** varchar - dacă acest câmp este NULL, atunci drumul nu conține nici o instituție de învățământ; în caz contrar, va conține numele acesteia;
- **Asfalt** bit - 1 dacă drumul e asfaltat, 0 altfel;
- **Offroad** bit - 1 dacă drumul e offroad, 0 altfel;
- **Accident\*** number  $\geq 0$  -  $>0$  dacă pe drumul respectiv a fost raportat vreun accident, 0 altfel; cu fiecare raport de prezență incrementăm cu 1, iar cu fiecare raport de absență decrementăm cu 1;
- **Politie\*** number  $\geq 0$  -  $>0$  dacă pe drumul respectiv a fost raportată prezența vreunui echipaj de poliție, 0 altfel; cu fiecare raport de prezență incrementăm cu 1, iar cu fiecare raport de absență decrementăm cu 1;
- **Cameră\_vit\*** number  $\geq 0$  - analog;
- **Obst\_mic\*** number  $\geq 0$  - analog;
- **Obst\_mare\*** number  $\geq 0$  - analog;
- **Ambuteiaj\*** number  $\geq 0$  - analog;

**Observație:** toate atributele cu '\*' vor suferi modificări în mod dinamic în funcție de rapoartele oferite de utilizatori. Astfel, una sau mai multe alerte de atenționare vor rămâne valabile pe un drum din momentul în care un utilizator

care circulă pe acel drum emite un raport. Dacă vor mai emite și alți utilizatori același raport în aceeași, atunci numărul acestora va fi incrementat cu 1. Pentru un raport de absență, numărul rapoartelor se va decrementa cu 1.

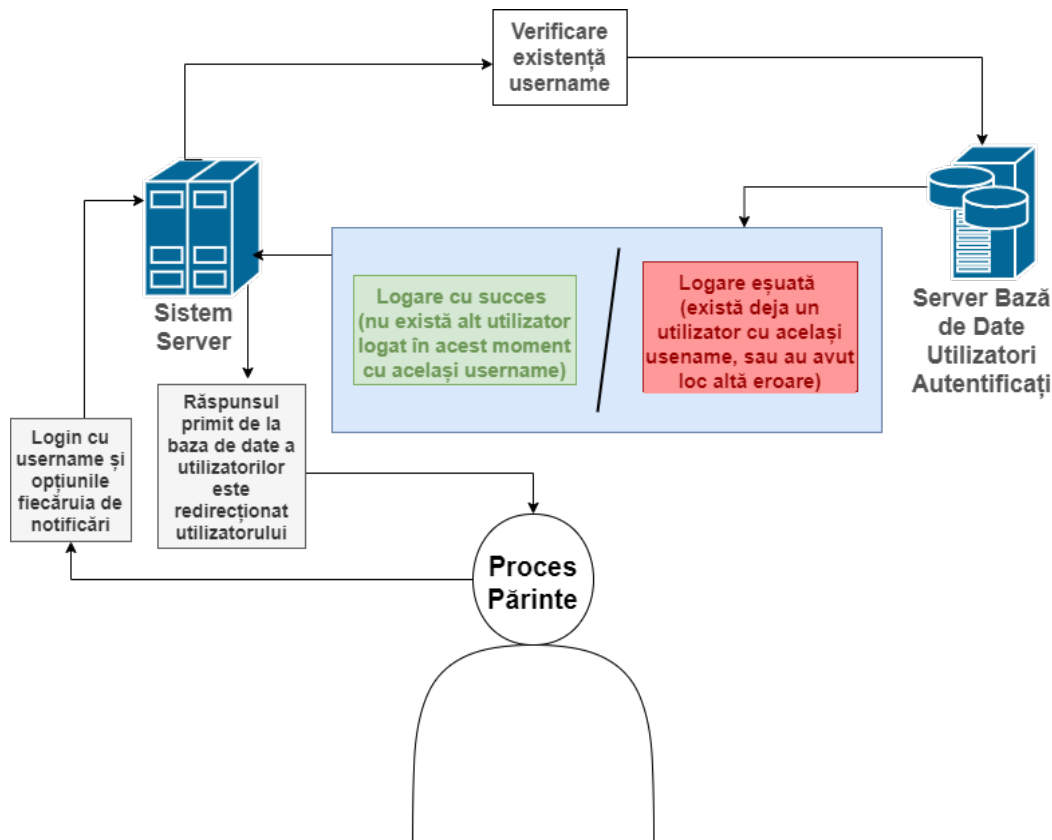
Totodată, pentru stocarea utilizatorilor conectați, se va folosi un alt tabel, care va stoca următoarele informații:

- **User\_id** varchar - numele de utilizator; câmp de tip cheie primară (not null și unic);
- **Vreme** bit - 1 dacă utilizatorul a optat pentru notificări legate de vreme, 0 altfel;
- **Pret\_carb** bit - 1 dacă utilizatorul a optat pentru notificări prețul carburantului la stațiile de alimentare, 0 altfel;
- **Sport** bit - 1 dacă utilizatorul a optat pentru notificări legate de evenimente sportive, 0 altfel;

### 3 Arhitectura aplicației

**Observație:** Am ales să divizez diagrama arhitecturii aplicației în 3 părți, pentru a fi mai facil de urmărit schema de funcționare. Astfel, această secțiune va fi împărțită în mai multe subsecțiuni.

#### 3.1 Logarea utilizatorilor

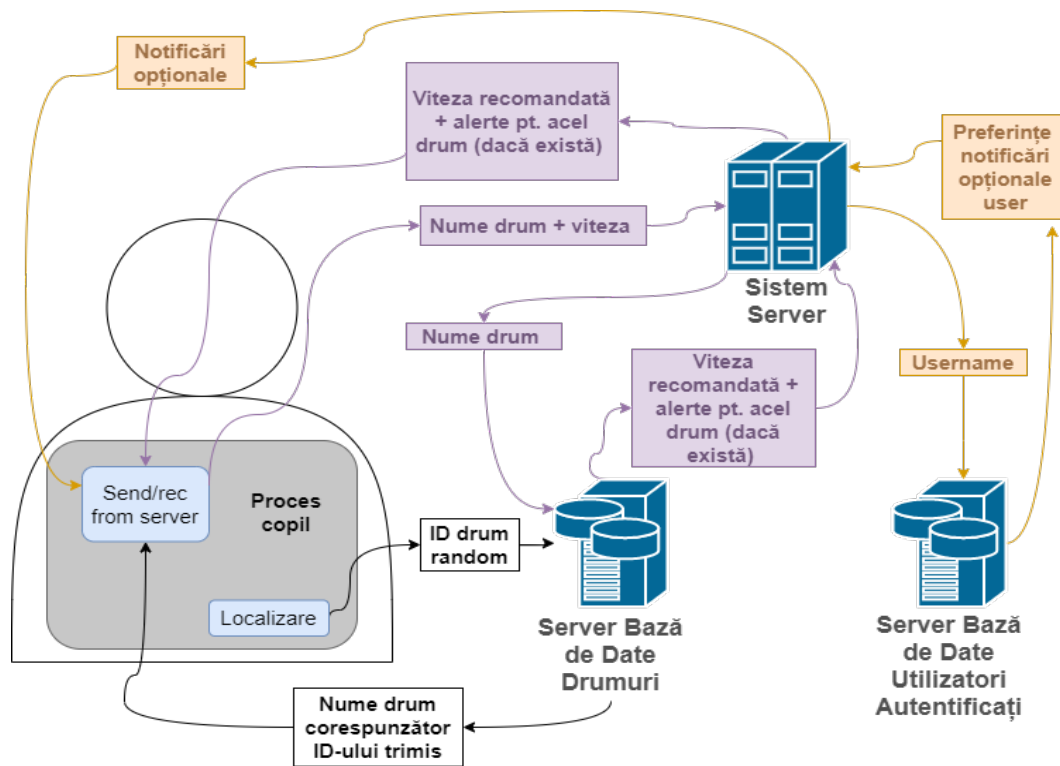


Pentru a putea beneficia de serviciile oferite de aplicația Monitorizare trafic, fiecare utilizator trebuie mai întâi să stabilească o conexiune cu serverul, prin inițierea unui **request de login**, care conține un **username** și **opțiunile utilizatorului** în legătură cu notificări legate de vreme, prețul carburantului din stațiile de alimentare și evenimente sportive.

Pentru ca operația de login să se realizeze cu succes, este necesar ca **username-ul** să conțină cel puțin un caracter de tip literă și să nu coincidă cu vreun alt username al altui utilizator logat în acel moment la server. Așadar, orice username trebuie să respecte condițiile de **cheie primară** din **baza de date a utilizatorilor**, adică să fie **unic** și **nenul**. Bineînțeles, în urma unei logări cu

succes, în baza de date a utilizatorilor va apărea o nouă intrare, iar la deconectarea unui utilizator, intrarea sa din baza de date va fi ștearsă.

### 3.2 Serviciul de recomandare viteză și notificări opționale



După cum se poate observa, diagrama de mai sus conține 3 “trasee”, fiecare colorate diferit și reprezentând cate un serviciu oferit de aplicație:

1. **Localizare** - așa cum se vede în diagramă, serviciul de localizare nu este unul în adevăratul sens al cuvântului, deoarece realizarea unuia cu geolocație reală nu ar fi fezabil și ar depăși scopul acestui proiect; tocmai de aceea, am ales să realizez localizarea tot cu ajutorul comunicării, accesând baza de date ce conține informațiile legate de drumuri; astfel, clientul generează un număr natural random, reprezentând un ID, apoi interoghează baza de date, primind în schimb numele drumului aferent ID-ului generat;
2. **Viteză recomandată** - pentru a primi o recomandare în legătură cu viteza cu care utilizatorul ar trebui să circule pe acel drum, acesta îi trimite serverului numele drumului pe care se află (primit cu ajutorul serviciului de **Localizare**), împreună cu o viteză generată tot în manieră aleatorie; odată intrat în posesia acestor date, serverul interoghează baza de date a drumurilor, obținând de la aceasta viteza recomandată pentru acel drum, împreună cu

alertele aferente, dacă acestea există, cum ar fi: drum neasfaltat, drum în localitate și/sau în afară, obstacole, poliție, accident etc., informații pe care acesta le trimite înapoi clientului;

3. **Notificări opționale** - la început, când clientul se înregistrează, acesta poate opta pentru a primi diverse notificări suplimentare, legate de vreme, prețul carburantului la stațiile de alimentare sau evenimente sportive; așadar, serverul interoghează baza de date a utilizatorilor pentru a verifica ce opțiuni de notificări suplimentare au fost alese de acel client; odată intrat în posesia răspunsului de la baza de date, serverul îi trimite informațiile relevante utilizatorului;

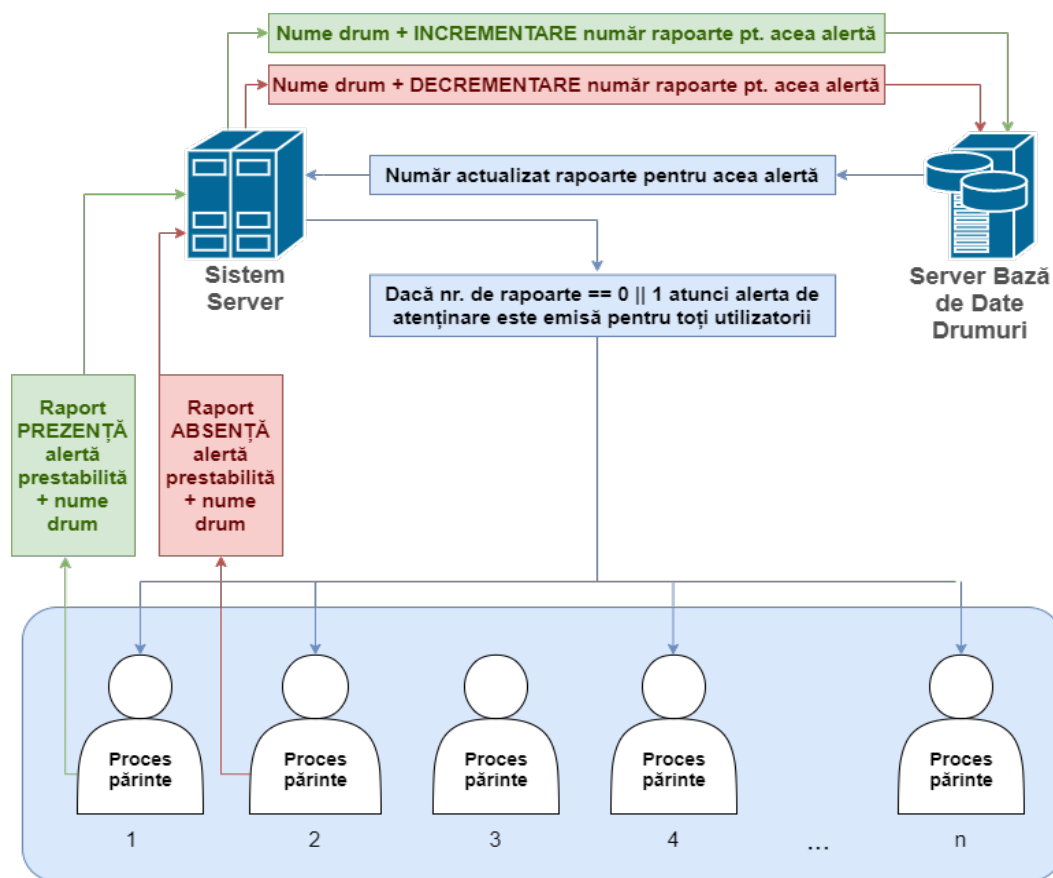
**Observație:** Operațiile de **Localizare** și **Viteză recomandată** au loc automat cu o frecvență de 1min, în timp operația de **Notificări opționale** se realizează automat cu o frecvență de 5min;

**Implementarea concurentă - server.** Pentru ca serverul să trateze clienții într-o manieră concurentă, se vor folosi mai multe fire de execuție, astfel încât pentru fiecare client conectat va exista câte un thread alocat acestuia.

**Implementare concurentă - client** În ceea ce privește clientul, acesta este și el implementat concurent, în cazul acesta cu primitiva **fork()**.

**Procesul copil** este acela care se ocupă de comunicarea automatizată cu serverul, adică asigură serviciul de **viteză recomandată** și **notificări opționale**. **Procesul părinte** este acela care asigură serviciul de **login** și de **raportare evenimente**. Acesta așteaptă input de la tastatură, trimite serverului mesajul și primește imediat răspuns, în timp ce procesul copil, în paralel, recomandă viteza.

### 3.3 Serviciul de raportare evenimente



Fiecare client va putea raporta un eveniment ce a avut loc pe șoseaua pe care se află; acest raport va fi trimis de către server tuturor utilizatorilor sub forma unei alerte prestabilite, precum:

1. **Accident;**
2. **Poliție;**
3. **Cameră de viteză;**
4. **Obstacol mic** - gropi, denivelări, canale, animale sau orice alt obstacol de dimensiuni mici ce poate fi evitat;
5. **Obstacol mare** - orice obstacol de dimensiuni mari, imobil, ce nu poate fi evitat, fapt ce rezultă în blocarea traficului pe acea porțiune de drum;
6. **Lucrări;**
7. **Ambuteiaj.**

Acest lucru se realizează cu ajutorul câmpurilor aferente ale acestor alerte din baza de date a drumurilor.



Astfel, odată cu un primirea de către server a unui raport de prezență a unei alerte de la un utilizator, serverul execută un update în baza de date, pe rândul drumului cu numele primit tot de la client, incrementând cu 1 în câmpul corespunzător alertei.

În schimb, dacă serverul primește un raport de absență a unei alerte, serverul face update pe baza de date, decrementând cu 1, pe rândul drumului cu numele primit, câmpul aferent alertei.

Apoi, serverul execută o interogare pe aceeași bază de date, pentru a primi în schimb numărul actualizat de rapoarte pentru acea alertă. În cazul în care este 0 sau 1, adică evenimentul a dispărut sau a apărut, serverul emite apoi un mesaj de atenționare corespunzător, pe care îl trimite apoi tuturor utilizatorilor.

**Observație:** Valorile din câmpurile aferente alertelor de atenționare sunt numere naturale pozitive. Astfel, dacă au raportat **n** utilizatori prezența unui eveniment, atunci va fi nevoie de **n** rapoarte de absență pentru ca acesta să nu mai fie trimis ca alertă de atenționare utilizatorilor.

**Implementare concurentă - server.** Principalul motiv pentru care am ales să folosesc o implementare cu **thread-uri** și nu cu procese copil, este faptul că firele de execuție ale aceluiași proces rulează împărțind aceeași memorie, iar acest lucru este relevant tocmai în această situație, când un utilizator trimite serverului un raport predefinit al unei alerte de atenționare, iar acesta, la rândul lui, trebuie să trimită mai departe alerta tuturor utilizatorilor.

## 4 Detalii de implementare

**Observație:** Orice schimb de informații între client și server se vor realiza urmând urmatorul protocol: înainte de a trimite orice mesaj, expeditorul va trimite mai întâi dimensiunea acestuia și apoi mesajul în sine. Astfel, destinatarul va ști exact cât trebuie să citească.

### 4.1 Login - protocoale de comunicare și use case-uri

Pentru a putea beneficia de serviciile oferite de aplicația **Monitorizare Trafic**, fiecare utilizator trebuie mai întâi să se înregistreze cu un username **unic și nenul** (adică să conțină cel puțin un **caracter literă** și să nu existe deja un username identic în baza de date). Totuși, username-ul nu ar trebui să depășească **50 de caractere**, iar acesta poate conține orice caracter simbol în afară de **‘:’**. Astfel, la nivelul aplicației client, prin intermediul procesului părinte, se vor face verificările de format, înainte ca username-ul să fie trimis serverului, iar la nivelul serverului se va verifica, cu ajutorul bazei de date a utilizatorilor, dacă username-ul este folosit deja de alt client.

Pentru ca un utilizator să se poată autentifica cu succes, comanda de login trebuie să arate în felul următor:

**login username**

Unde **username** reprezintă numele de utilizator cu care clientul dorește să fie înregistrat în baza de date și care trebuie să respecte condițiile menționate mai sus.

Dacă username-ul nu respectă condițiile menționate, utilizatorul va fi notificat iar acesta va putea introduce alt username.

Dacă username-ul este corect, utilizatorul va fi întrebat dacă dorește să opteze pentru a primi notificări suplimentare precum:

1. **Starea vremii;**
2. **Prețurile carburantului;**
3. **Evenimente sportive.**

Utilizatorul nu trebuie decât să introducă orice combinație de cifre corespunzătoare opțiunilor din lista de mai sus, însă nu mai lungă de 3 caractere.

Exemple:

- **1** - utilizatorul va primi doar notificări despre starea vremii;
- **2** - utilizatorul va primi doar notificări despre prețuri la carburant;
- **23** - utilizatorul va primi doar notificări despre prețuri la carburant și evenimente sportive;
- **213** - utilizatorul va primi toate tipurile de notificări suplimentare;
- **11** - utilizatorul va primi doar notificări despre starea vremii;
- **333** - utilizatorul va primi doar notificări despre evenimente sportive;
- **1231** - invalid, utilizatorului i se va cere să reintroducă o secvență validă de opțiuni;

La sfârșit, aplicația client va pregăti următorul mesaj text, care urmează să fie trimis către server:

**login username:X**

Unde X reprezintă o combinație validă de opțiuni de notificări suplimentare. Se poate observa că simbolul ‘:’ este folosit ca delimitator în comanda de login. Tocmai din acest motiv am impus restricția ca username-ul poate conține orice simbol în afară de ‘:’.

Odată ajuns mesajul la server, acesta identifică comanda și separă username-ul pentru a-l verifica în baza de date dacă există deja.

Dacă da, atunci serverul va returna un mesaj clientului, informându-l că username-ul este deja luat, iar acesta poate încerca să se autentifice cu un alt username.

Dacă nu, atunci serverul va înregistra utilizatorul în baza de date, împreună cu opțiunile sale.

#### **4.2 Servicul de recomandare viteză și notificări opționale - protocol de comunicare și use case-uri**

Fiecare utilizator, prin intermediul procesului copil, va trimite în mod automat, cu o frecvență de 1min, un mesaj text serverului cu următorul format:

**auto nume\_drum:viteza**

**nume\_drum** este obținut cu ajutorul serviciului de localizare descris mai sus în **subsecțiunea 3.2** iar **viteza** va fi și ea generată tot random.

Serverul, odată intrat în posesia acestui mesaj, recunoaște comanda și separă numele drumului și viteza. Acesta apoi efectuează o interogare în baza de date a drumurilor, unde verifică atributele drumului cu numele primit de la client. Server-ul apoi preia valorile din baza de date, compară apoi viteza cu cea adecvată pentru acea porțiune de drum și îi pregătește clientului un mesaj de atenționare adecvat.

**Exemplu:** Să presupunem că clientul circulă pe Bv. Independenței, cu o viteză de 70km/h. El îi va trimite serverului următorul mesaj:

**auto Bv Independenței:70**

Serverul caută apoi în baza de date înregistrarea care pe câmpul **Nume** conține "Bv Independenței" și vede că este o șosea asfaltată din localitatea Iași. Deoarece viteza utilizatorului este mai mare decât cea legală pentru acel drum, serverul îi va trimite următorul mesaj:

**Viteză recomandată: 50km/h - localitate**

Dacă, în plus, în baza de date există informații cum că pe Independenței ar fi avut loc un accident, mesajul va arăta astfel:

**Atenție! ACCIDENT raportat pe Bv Independenței**

**Viteză recomandată: 50km/h - localitate**

Dacă există și alte astfel de evenimente raportate pe bulevard, atunci în mesajul de atenționare vor fi incluse toate.

În cazul notificărilor opționale, ele vor veni automat de la server, la fiecare 5min, fără a fi nevoie de mesaj de la client. Serverul interoghează baza de date pentru a verifica ce opțiuni a ales clientul, iar dacă acestea există atunci îi va trimite notificările corespunzătoare.

În final, există posibilitatea ca, dintr-un motiv sau altul, clientul să nu mai trimită nimic. În acest caz, dacă de la client nu a primit nimic timp de câteva minute, atunci serverul va înceta conexiunea cu acesta și îl va îndepărta din baza de date pentru a economisi resurse pentru utilizatorii activi.

#### **4.3 Serviciul de raportare evenimente - protocol comunicare și use case-uri**

În orice moment, orice utilizator poate raporta, pe drumul pe care circulă, unul sau mai multe dintre următoarele evenimente prestabilite:

1. **Accident;**
2. **Poliție;**
3. **Cameră de viteză;**
4. **Obstacol mic** - gropi, denivelări, canale, animale sau orice alt obstacol de dimensiuni mici ce poate fi evitat;
5. **Obstacol mare** - orice obstacol de dimensiuni mari, imobil, ce nu poate fi evitat, fapt ce rezultă în blocarea traficului pe acea porțiune de drum;

#### 6. Lucrări;

#### 7. Ambuteiaj.

Atunci când utilizatorul dorește să raporteze ceva, aplicația client, prin intermediul procesului părinte, îi va arăta lista de mai sus, iar utilizatorul nu trebuie decât să tasteze o secvență de cifre și litere, nu mai lungă de 14 caractere, corespunzătoare alertelor din listă, fiecare cifră fiind urmată de litera ‘p’ pentru prezența unui eveniment sau litera ‘a’ pentru absența unui eveniment.

**Exemplu:** Utilizatorul dorește să raporteze că a avut loc un accident, care a cauzat un ambuteiaj pe Bv. Independenței. El trebuie doar să tasteze secvența de caractere “1p7p”. Serverul va primi următorul mesaj de la client:

**raport Bv Independenței:1p7p**

Apoi acesta va face un update la baza de date, incrementând cu 1 câmpurile **Accident** și **Ambuteiaj** aferente înregistrării pentru Bv. Independenței.

În mod similar, dacă un utilizator raportează absența vreunui eveniment pe un anumit drum, atunci în baza de date vor avea loc decrementări cu 1 pe câmpurile corespunzătoare.

După fiecare operație de update pe baza de date, serverul o interoghează pentru a obține nr. de rapoarte actualizat, iar dacă acesta este 0 sau 1 pentru un anumit eveniment, adică statusul acestuia s-a schimbat din existent în inexistent sau viceversa, atunci o alertă de atenționare va fi trimisă tuturor utilizatorilor.

**Exemplu:** A raportat pentru prima dată un accident și un ambuteiaj pe Bv. Independenței. Atunci toți utilizatorii vor primi imediat următorul mesaj:

#### **Atenție! ACCIDENT și AMBUTEIAJ raportate pe Bv Independenței**

Totodată, cât timp în baza de date există o valoare strict pozitivă pe câmpurile pentru evenimente ce pot fi raportate, atunci această alertă de atenționare va fi trimisă tuturor utilizatorilor atunci când solicită viteza recomandată, așa cum am ilustrat în exemplul de mai sus.

#### 4.4 Comanda quit

Tot în cadrul procesului părinte, utilizatorul va putea tasta comanda **quit**. Aceasta este trimisă mai departe la server, care ca întrerupe conexiunea cu clientul. Iar clientul mai întâi oprește execuția copilului și apoi se oprește de tot.

#### 4.5 Rezumatul comenzilor

1. **login username:opțiuni;**
2. **auto locație:viteză** - realizată automat de către procesul copil;
3. **raport locație:alerte;**
4. **quit.**

## 5 Concluzii

Voi menționa aici câteva idei care să concluzioneze acest raport tehnic al aplicației **Monitorizare Trafic**.

Ca îmbunătățiri, consider că implementarea unei funcții de localizare reală și precisă s-ar număra printre cele mai importante lucruri de adăugat la acest proiect. Bineînțeles, odată cu asta am putea calcula și viteza reală cu care utilizatorul circulă. Totodată, odată cu adăugarea unei funcții de locație reală, felul în care sunt distribuite alertele ar fi mult mai relevant.

O altă îmbunătățire considerabilă ar fi o interfață grafică, care să îl scutească pe utilizator de a tasta preferințele atunci când se loghează, sau atunci când vrea să raporteze un eveniment de pe drum.

## 6 Bibliografie

1. <https://profs.info.uaic.ro/~computernetworks/cursullaboratorul.php#s2>
2. <https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines>
3. <https://dev.mysql.com/>
4. <https://www.overleaf.com/learn/latex>
5. <https://stackoverflow.com/questions/>
6. <https://app.diagrams.net/>
7. <https://www.latex-tutorial.com/tutorials/>