UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ

LUCRARE DE LICENȚĂ

Serviciul de ambulanțe

propusă de

Neculai-Vasile Anghel

Sesiunea: *Februarie, 2019*

Coordonator științific

Lect. Dr. Alex Moruz

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ

Serviciul de ambulanțe

Neculai-Vasile Anghel

Sesiunea: *februarie, 2019*

Coordonator științific

Lect. Dr. Alex Moruz

Avizat,

Îndrumător Lucrare de Licenţă

Titlul, Numele şi prenumele

_____

Data _____ Semnătura

_____

DECLARAȚIE privind originalitatea conținutului lucrării de licență

Subsemntatul(a) ……

domiciliul în ……..

născut(ă) la data de ……., identificat prin CNP ……., absolvent(a) al(a) Universităţii „Alexandru Ioan Cuza" din Iaşi, Facultatea de …….. specializarea ……., promoţia ……., declar pe propria răspundere, cunoscând consecinţele falsului în declaraţii în sensul art. 326 din Noul Cod Penal şi dispoziţiile Legii Educaţiei Naţionale nr. 1/2011 art.143 al. 4 si 5 referitoare la plagiat, că lucrarea de licenţă cu titlul: _____

_____

_____elaborată sub îndrumarea dl. / d-na _____, pe care urmează să o susţină în faţa comisiei este originală, îmi aparţine şi îmi asum conţinutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licenţă să fie verificată prin orice modalitate legală pentru confirmarea originalităţii, consimţind inclusiv la introducerea conţinutului său într-o bază de date în acest scop.

Am luat la cunoştinţă despre faptul că este interzisă comercializarea de lucrări ştiinţifice in vederea facilitării fasificării de către cumpărător a calităţii de autor al unei lucrări de licenţă, de diploma sau de disertaţie şi în acest sens, declar pe proprie răspundere că lucrarea de faţă nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Dată azi, …… Semnătură student ……

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „*Titlul complet al lucrării*”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, *Februarie 2018*

Absolvent *Neculai-Vasile Anghel*

_____

(semnătura în original)

Cuprins

INTRODUCERE

Într-o lume în care tehnologia a preluat controlul vieții de zi cu zi, gadget-urile și aplicațiile au facilitat sau chiar au înlocuit servicii costisitoare în timp și efort precum corespondența prin poștă, plățile facturilor, interacțiunea vizuală și auditivă în timp real cu persoane de peste mări și tări, serviciile de urgență încă sunt predispuse la amenințări care pot însemna o înclinare defavorabilă a balanței pentru un pacient aflat în stare de pericol. Greșeala umană, preluarea unor date neclare sau neînțelese de către dispecer și, desigur, starea instabilă a pacientului sunt doar trei dintre factorii care pot pune în dificultate un serviciu deja supus la încercări consumatoare de resurse importante : SERVICIUL DE AMBULANȚE.

Serviciu de ambulanțe, este alcătuit din unități sanitare publice aflate în coordonarea departamentului de specialitate din Ministerul Sănătății și a Autorităților de sănătate publică județene, având în structura lor un compartiment pentru asistență medicală de urgență și transport medical asistat, cu echipaje medicale de urgență, cu sau fără medic, și un compartiment pentru consultații medicale de urgență la domiciliu și transport sanitar neasistat. Compartimentul pentru asistență medicală de urgență funcționează în regim de lucru continuu, în așteptarea solicitărilor de asistență medicală de urgență.

Ocupând una din primele poziții pe scara importanței serviciilor publice, Serviciul de Ambulanțe trebuie să evolueze și să devină din ce în ce mai rapid și eficient pe măsură ce puterea și răspândirea tehnologiei creste. Nu este posibil și permis ca într-o lume a automatizării, să mai existe încă posibilitatea greșelii umane. Ține de interesul întregului sistem medical ca procedurile de acționare în cazul urgențelor să se bazeze tot mai mult pe algoritmi de automatizare și mai puțin pe preluarea și manipularea datelor de către o persoana fizică.

Prin lucrarea de licență ,,Serviciul de Ambulanțe”, pune la dispoziție o alternativă mai rapidă și mai sigură a procedurii de preluare a urgențelor adresate serviciului de ambulanțe județene, actual integrat în apelul 112.

Prin implementarea acestui sistem de aplicații, ne dorim să diminuam cât mai mult timpul de preluare pentru fiecare urgență, să automatizăm cât mai mult transmiterea de informații pentru a evita o posibilă distorsionare a mesajelor verbale și să punem la dispoziție o monitorizare transparentă atât a urgențelor cât și a disponibilității resurselor.

Lucrarea se axează pe prezentarea punctelor cheie care stau la baza unui sistem de aplicații menit să ducă serviciul actual de ambulanțe la un nivel de încredere demn de cele mai înalte standarde de eficiență și rapiditate:

Problema → Soluție → Arhitectura aplicației/ Metoda de implementare a soluției →
→ Securitatea sistemului implementat → Metode de îmbunatățire a soluției

CONTRIBUȚII

Aruncând o privire obiectivă asupra serviciilor publice din România în mâna cărora ne încredințăm de cele mai multe ori viața , nu e nevoie de mai mult de câteva minute să ne dăm seama că avem proleme uriașe cu procedurile de acționare, securitatea împotriva oricărui tip de atac și de ce nu, probleme cu a te încrede într-o unitate pătată cu un număr de eșecuri peste limita acceptării.

În urma conștientizării punctelor vulnerabile din sistemul actual de sănătate, am decis să încep prin găsirea soluției optime pentru *problema timpului și metodei de preluare a urgențelor medicale*.

Implementarea sistemului de aplicații ,,Serviciul de Ambulanțe" este un rezultat al cunoștințelor acumulate în timpul celor trei ani de studiu bazându-mă în principal pe informații din Programarea Orientată Obiect , Tehnologii Java , Programare Android, Rețelistica, dar și pe gândirea algoritmică plus diferite motode de abordare a subiectului aflat în discuție.

Esența lucrării stă în alegerea celor mai potrivite și eficiente funcționalități spre a fi implementate și folosite cu ușurință atât de utilizatorul aflat în stare de urgență cât și de utilizatorul ce urmează să preia apelul S.O.S. semnalat ulterior.

1. Serviciul de ambulanțe – S.D.A.

3.3.1 Descrierea problemei

Cea mai mare problemă cu performanța serviciilor de ambulanțe din România este timpul de răspundere și modul de preluare a urgențelor medicale semnalate prin apelul 112. Impactul acestei perfomanțe scăzute îl găsim în numărul enorm de cazuri care necesitau asistență medicala de urgență dar în urma întârzierilor exagerate s-au finalizat prin agravarea stării pacientului pâna la punctul de deces.

Principalii factori consumatori de timp în preluarea unei urgențe :4

a. *Integrarea în serviciul de urgentă 112* : Cea mai rapidă metodă de cerere a serviciilor de asistentță medicală de urgență existentă în momentul acesta este apelul de urgență 112. Acesta nu este un serviciu dedicat urgențelor medicale ci întregului serviciu public de siguranță și sănătate : poliție , ambulanță, pompieri , SMURD. Integrarea într-un serviciu general de activități este consumatoare de timp deoarece intervine nevoia de a transfera urgența către aria de interes.
b. *Preluarea apelurilor de către operatori* : Un al doilea factor de importanță majoră este faptul că apelurile sunt preluate manual de operatori. Asta însemnă că urgența nu este procesată instant ci este pusă într-o coadă și o să fie preluată doar în momentul în care un operator este disponibil. Chiar dacă o unitate are în funcție un număr mare de operatori, pe lângă faptul că această practică e consumatoare uriașă de resurse , în cazul unei catastrofe naturale, produse de om sau orice alt dezastru semnificativ, intervine preluarea urgențelor prin practica cozii sau a gravității cazului
c. *Transmiterea verbală a informațiilor între pacient și dispecer :* în urma intervenirii unor factori distorsionanți precum : sunetul de calitate inferioară sau alterat de alte sunete din jur, starea instabilă a pacientului care a făcut apelul, necunoașterea locației din care s-a semnalat urgența atât de către pacient care poate fi străin acolo cât și de personalul de

dispecer pentru că i se comunică o denumire ambiguă sau locală a reperului geografic.

Toți acești factori sunt cât se poate de prezenți și dăunători atât pentru instituția publică deoarece metoda actuală e consumatoare de resurse , dar și pentru pacienți deoarece un minut adăugat la timpul de așteptare pentru o ambulanță este o șansă în minus în scăparea cu viață din impasul medical în care se află.

3.3.2 Analiza situației actuale

Conform SERVICIULUI DE TELECOMUNICAȚII SPECIALE, în 2018 peste 50% din apelurile de urgență la 112 au fost făcute pentru Serviciul de Ambulanțe. Situația procentajului de apeluri pe unitate de intervenție de la sfârșitul lunii decembrie 2018 (**Figura 1.**) arată în felul următor :

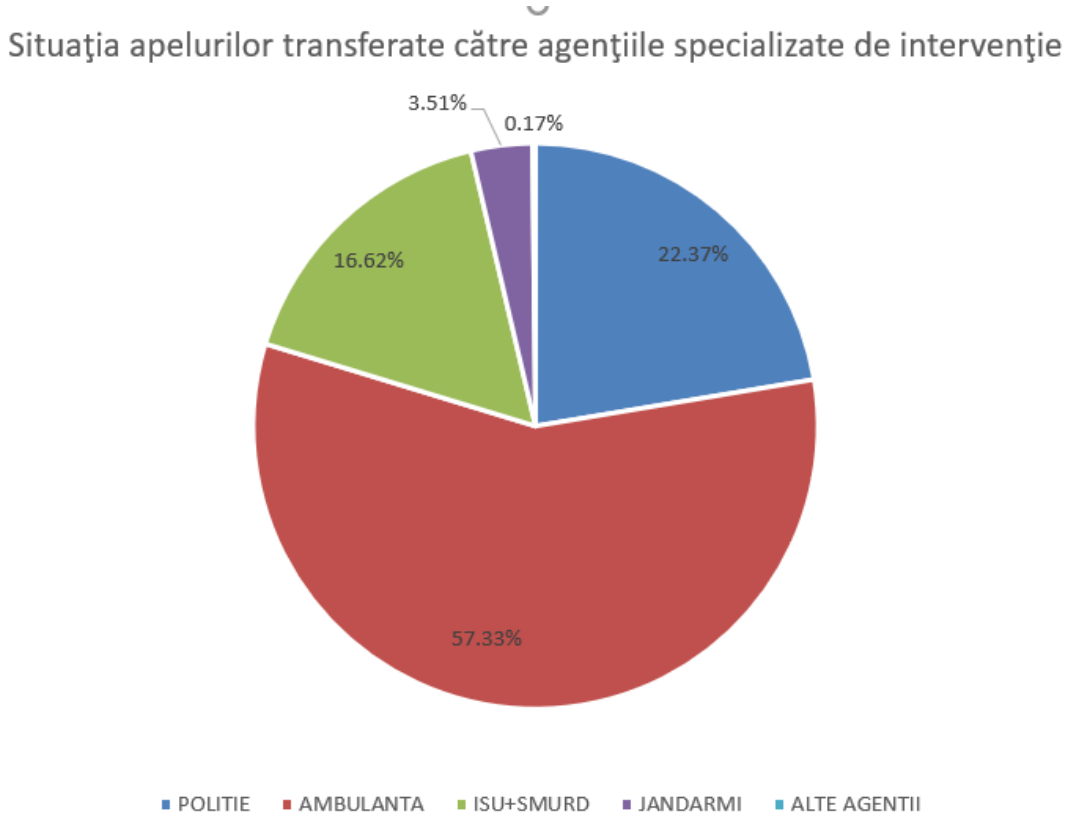

Figura 1. Statistica apeluri decembrie 2018

Această situație nu se modifică cu mai mult de 5-10 procente de la lună la lună sau 10- 15 procente de la an la an. Conform statisticilor de mai sus Serviciul de Ambulanțe este cel mai solicitat din rândul serviciilor de urgență. Cu cât încărcarea liniei de urgență 112 cu cerințe de asistență medicală este mai mare cu atât funcționarea precedurii de preluare în standarde acceptabile este mai greu de satisfăcut .

3.3.3 Descrierea Soluției

Sistemul de aplicații propus prin această lucrare de licență oferă o cale cât mai sigură, mai rapidă și mai clară pentru solicitarea asistenței medicale de urgență.

Apelul S.O.S este semnalat printr-o simplă apăsare de buton care preia inițial datele de identitate și locația pacientului și apoi, în timpul deplasării ambulanței către locul urgenței medicale, pacientul poate furniza informații ajutătoare despre cazul medical, pentru a ușura stabilirea unui diagnostic de către echipajul ambulanței înca dinaintea sosirii la locația furnizată de GPS-ul aplicației PACIENT.

Sistemul de aplicații S.D.A. pune la dispoziție un serviciu destinat numai pentru preluarea și manipularea urgențelor medicale folosindu-se de un proces mult mai optimizat din punct de vedere al resurselor și metodelor de prelucrare a informației schimbate în timpul unei urgențe.

Automatizarea procesului actual prin introducerea tehnologiilor de rulare în paralel a mai multor activități scoate din calclul preluarea și manipularea apelurilor de urgență de către operatorii umani. Prin această practică se evita procesarea urgenței după practica cozii, asignarea ambulanței făcându-se în maxim 6 secunde. Fiecare urgență este asignată la ambulanța care are cel mai scut durm de parcurs până la pacientul aflat în pericol.

2. Arhitectura sistemului S.D.A.

2.1 Aplicații componente și rolul lor în S.D.A.

În spatele acestui serviciu avem un sistem Server - Client alcătuit din patru aplicații interactive (Pacient, Ambulanță, Medic, Dispecer – **Figura 2.**) și o aplicație server care asigură căi de comunicare între clienți și securitatea întregului serviciu.

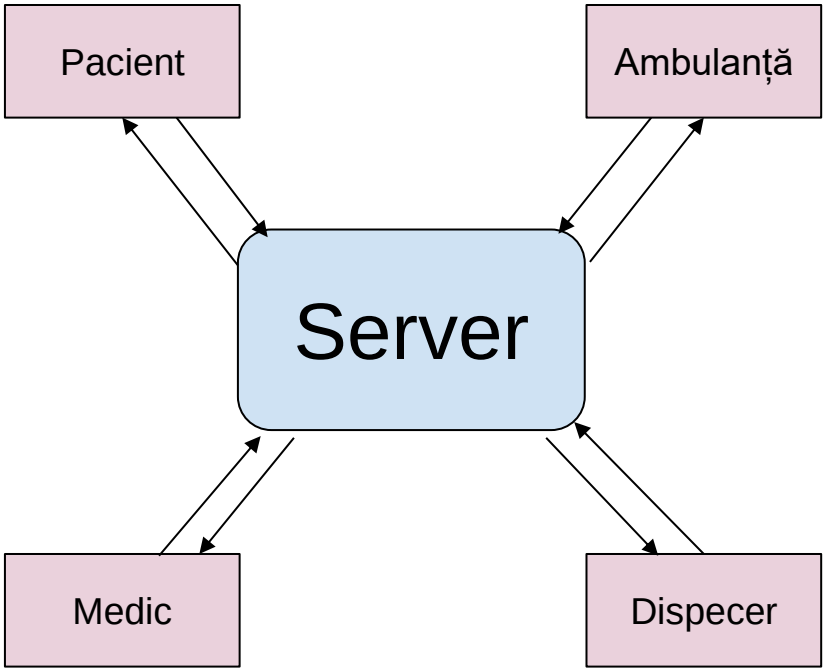

Figura 2. Structura Sistemului S.D.A.

Instanțele Serviciului S.D.A. corespund cu părțile participante în realitate la procesul de inițierea apelului S.O.S. de către PACIENT și preluarea urgenței de către AMBULANȚĂ.

DISPECERUL monitorizează situația urgențelor și statusul individual pentru fiecare dintre ele. MEDICUL preia informații prin panoul de monitorizare urgențe pentru a putea anticipa din timp ce acțiuni medicale necesită fiecare caz în parte.

2.2.1 ***Dispecer -*** Aplicație Desktop (implementată în JAVA) :

Această entitate are rol principal de a monitoriza urgențele medicale inițiate prin aplicația Pacient. Pe langă funcția principală , dispecerul (**Figura 3.**) pune la dispoziție un set secundar de funcționalități pentru a facilita întregul proces de preluare urgențe.

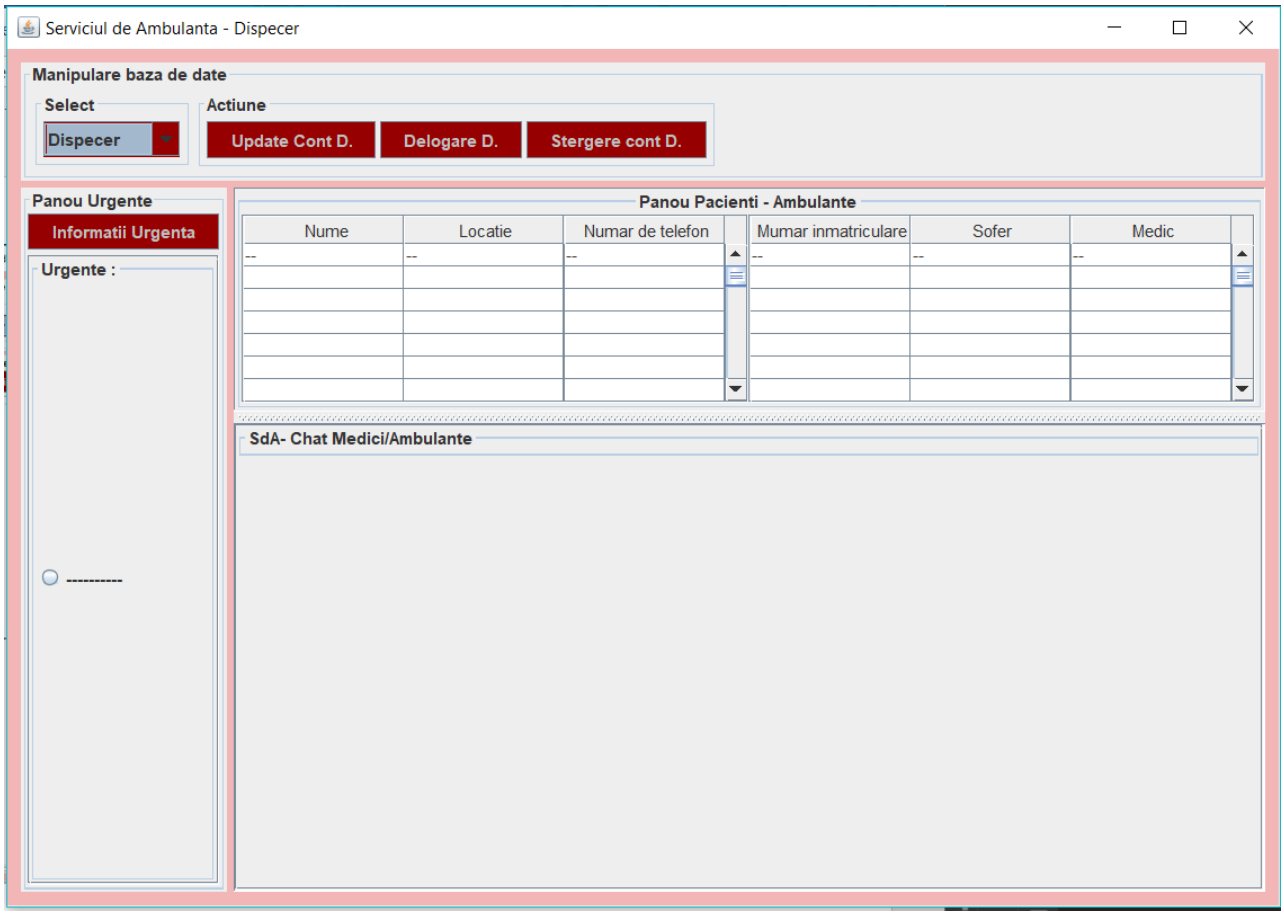

Figura 3. Dispecer

Funcționalități implementate:

- Autentificarea aplicației

  Autentificare aplicației este primul pas în procesul de obținerea accesului la serviciile puse la dispoziție de Sistemul S.D.A. Scopul acestei autentificări este împiedicarea aplicațiilor corupte - „hacker” care au ca țintă destabilizarea întregului serviciu.

- Logare

  Logarea este funcția prin care utilizatorul se identifică cu profilul deja înregistrat în sistem și pe baza căruia poate să se foloseascp de funcționalitățile aplicației.

- Înregistrare user Dispecer

  Funcția de înregistrare are ca scop crearea unui profil de utiliazator nou , unic prin datele personale înscrise , cu ajutorul căruia se poate realiza logarea în Sistemul S.D.A.

- Modificare cont Dispecer

  Oferă utilizatorului opțiunea de a-și modifica datele asociate contului personal

- Ștergere cont Dispecer

  În cazul unui operator de dispecer care trebuie scos din sistem din diferite motive, funcția de ștergere cont dispecer permite această acțiune

- Vizualizare panou urgențe

  Panoul de urgențe, aflat printre cele mai importante elemente ale aplicației dispecer, monitorizează în timp real urgențele active și informațiile trimise atât de pacient cât și cele trimise de personalul care a preluat urgența asignată.

- Vizualizare ambulanțe disponibile

  Panoul ambulanțelor disponibile, monitorizează în timp real situația ambulanțelor active.

- Vizualizare pacienți activi

  Panoul pacienților activi, are ca scop monitorizarea în timp real a pacienților logați în sistem.

- Citirea datelor trimise de Pacient

  Dispecerul, pe lângă monitorizarea ambulanțelor, pacienților și urgențelor active , poate vizualiza informații trimise de pe aplicația Pacient sau preluate din baza de date în momentul inițierii unui nou apel S.O.S.

- Citirea datelor trimise de Ambulanță

  Funcția de citire a datelor trimise de aplicația ambulantă are ca scop preluarea informațiilor sesizate de personalul ambulanței despre situația pacientului aflat în stare de urgență.

- Avertizare pacient

  În cazul în care un apel S.O.S. inițiat prin aplicația android pacient, se dovedește a fi fals dar nu au fost prejudicii medicale în cazul unei alte urgențe la care nu s-a putut ajunge din cauza apelului curent sau prejudicii materiale pentru serviciul de ambulanțe precum deplasarea la locația urgenței false , dispecerul poate acorda un singur avertisment de încalcare a regulilor de folosire către pacientul inițiator.

- Banare pacient (alarme false care se pedepsesc penal)

  În momentul în care un apel S.O.S. fals care provoacă prejudicii medicale sau materiale de orice fel, dispecerul poate bana profilul pacientului, având ca și consecință imposibilitatea de a mai folosi Serviciul S.D.A. Deoarece înregistrarea pacientului se face cu date reale (CNP,număr telefon etc.) persoana banată nu se mai poate înregistra cu identitatea sa reală iar dacă face un apel de pe un cont cu date false, acțiunea este semnalată la poliție și se creează un dosar penal conform legilor împotriva apelurilor de urgență false înscrise în Ordonanța de urgență *nr. 34/2008 privind organizarea și funcționarea Sistemului național unic pentru apeluri de urgență.*

- Trimitere raport lunar de infracțiuni către poliție

  Funcția este destinată raportării apelurilor false cu prejudicii care urmează să fie penalizate

- Creare user Medic

  Crearea profilului unui medic este realizat doar prin intermediul dispecerului.

- Modificare cont Medic

  Ofera dispecerului opțiunea de a modifica datele asociate unui cont creat pentru aplicația MEDIC

- Ștergere user Medic

  Ștergerea unui cont de medic este funcțe asignată doar aplicației dispecer

- Creare user Ambulanță

  Funcția de creare user ambulanță are ca scop introducere în sistem a unei noi resurse de tip ambulanță

- Modificare cont Ambulanță

  Oferă dispecerului opțiunea de a modifica datele asociate unui cont creat pentru aplicația AMBULANȚĂ

- Ștergere user Ambulanță

  Funcția de ștergere user ambulanță are ca scop scoaterea din sistem a unei resurse de tip ambulanță

- Vizualizarea cerințelor speciale

  Panoul de vizualizare a cerințelor speciale are ca scop primirea în timp real a cerințelor sau mesajelor trimise prin intermediul aplicației ambulanță sau medic

- Deconectare

  Permite utilizatorului de a se deconecta de la contul curent

2.2.2 Pacient - Aplicație android :

Punctul din care pleacă punerea în acțioune a funcționalității principale oferite de Serviciul S.D.A. este aplicația PACIENT care inițiază apelul S.O.S.

Aplicația Pacient pune la dispoziție cea mai rapidă metodă de a solicita ambulanța pentru o situație de urgență medicală.

Functionalități implementate:

- Autentificarea aplicației

  La fel ca și la aplicația Dispecer, autentificarea are rolul de a se asigura că clientul care s-a conectat la Serviciul S.D.A. nu o face printr-o aplicație malițioasă ci prin aplicația PACIENT certificată pentru acest serviciu.

- Logarea userului pacient

  Are la bază același rol ca și în cazul dispecerului și anume identificarea utilizatorului cu un profil înregistrat în Serviciul S.D.A

- Trimiterea semnalului de urgență

  Este acțiunea care se identifică cu apelul S.O.S. prin care pacientul solicită asistentă medicală de urgență

- Trimitere date pacient:

  Are rolul de a oferi pacientului sau persoanei care a inițiat un apel de urgență S.O.S., un formular prin care să introducă informații adiționale despre identitatea și starea pacientului dar și despre condișiile în care s-a produs urgența.

  - Stare/simptome :
    - conștient : da/nu
    - stabil /instabil
    - temperatura : valoare=?
    - puls : valoare=?
    - tensiune : valoare=?
    - Imobilizat: da/nu
    - Altele : ……..

  - Date personale:
    - Pacientul : eu/alta persoana
    - Sex: bărbat/femeie
    - Varstă: valoare=?
    - Nume: valoare=?
    - Altele: …….

- Locație (se trimite automat de la GPS cand e trimis și apelul de urgență)

- Apel 112

  Permite utilizatorului PACIENT inițierea unui apel către serviciul 112 în cazul în care aplicația nu găsește o soluție optimă în asignarea unei ambulanțe la urgența semnalată

- Înregistrare pacient

  Funcția de înregistrare pacient pune la dispoziție crearea unui nou profil de utilizator cu ajutorul căruia pacientul se identifică în momentul logării

- Setări(update user profile, logout)

  Meniul de setări are ca rol oferirea unui set de opțiuni adiționale prin care pacientul poate seta anumite preferințe sau date asociate profilului

- Blocare cont (telefon pierdut)

  Permite unui utilizator care nu mai are acces la profilul său, din diferite motive, să blocheze de pe contul altui utilizator profilul pierdut cu ajutorul datelor de logare plus CNP-ul

- Deconectare

  Permite utilizatorului de a se deconecta de la contul curent

2.2.3 Ambulanța - Aplicație android :

Facilitatea pusa la dispoziție de către aplicația AMBULANȚĂ este cea de a putea prelua instant o urgență medicală fără a aștepta detaliile de locație de la dispecer iar în timpul deplasării asistate de navigator, aceasta poate primi toate datele legate de pacient și simptomele acestuia.

Funcționalitati implementate :

- Autentificare aplicație

  Funcționalitatea este identică ca și la celelalte aplicații prezentate anterior.

- Logare user

  Logarea utilizatorului se face pe baza aceleiași funcționalități ca la aplicațiile prezentate anterior

- Setare disponibilitare

  Opțiunea pusă la dispoziție prin funcția de disponibilitate, oferă utilizatorului posibilitatea de a schimba statusul ambulanței în funcție de activitatea personalului în acel monent

- Preluare urgenței pacient

  Activarea acestei funcționalități are rolul de a pune aplicația în ascultare dupa urgente noi aparute și nepreluate

- Citire date pacient

  Oferă personalului posibilitatea de a afla date despre urgența spre care se îndreaptă înainte să ajungă, datele fiind trimise de catre persoana care a inițiat apelul S.O.S. dacă aceasta este în stare de conștiență.

- Navigare catre locația primită de la pacient

  E funcția prin care aplicația AMBULANȚĂ pornește serviciul de navigație către poziția pacientului

- Trimiterea datelor observate de echipajul ambulantei:

  Funcționalitatea este dată de completarea unui formular simplu cu datele despre urgența preluată în momentul deplasării către spital

  - Stare pacient : stabil/instabil/critic/decedat
  - Date persoanle : bărbat/femeie, vârstă=?,Altele=?
  - Simptome: tensiune=?,temperatura=?,Puls=?, Altele=?

- Anulare urgență: pacient stabil/ alarmă falsă

  Opțiunea de anulare urgență are ca rol semnalarea unui apel fals sau închiderea urgenței prin stabilizarea pacientului care nu mai are nevoie de asistentă medicală într-un spital

- Cerințe speciale către dispecer

  Implementează un serviciu de transmitere de mesaje sub forma de cerințe către aplicația DISPECER

- Deconectare

  Permite utilizatorului de a se deconecta de la contul curent

2.2.4 ***Medic*** - Aplicație android :

Într-un serviciu automat prin care toate informațiile se transmit digital și transparent, cea mai bună metodă de pregatire a unui parcurs în spital potrivit pentru o urgentă medicală este monitorizarea urgențelor active și prin intermediul unei aplicații MEDIC .

Aceasta pune la dispoziție posibilitatea de vizualizare a panoului de urgențe și a informațiilor adunate în cursul procedurii de preluare.

Funcționalități implementate :

- Autentificare aplicație

  Funcția de autentificare este asigurată de aceeași funcționalitate implementată și în aplicațiile descrise anterior.

- Logare

  Serviciul de logare asignează utilizatorului profilul care se identifică cu datele introduse prin aceeași procedură ca în aplicațiile de mai sus.

- Vizualizare panou urgențe medicale

  Este principala funcționalitate a aplicației Medic cu ajutorul căreia utilizatorul poate vedea în timp real urgențele inițiate și informațiile adunate în procedura de preluare.

- Cerințe speciale către dispecer

  Ca și în cazul aplicației AMBULANȚĂ , implementează un serviciu de transmitere de mesaje sub forma de cerințe către aplicația DISPECER

- Deconectare

  Permite utilizatorului de a se deconecta de la contul curent

2.2.5 ***Server -*** Aplicație desktop fără interfață grafică :

Serverul asigură comunicarea între cele patru aplicații interactive prin căi de comunicare sigure, oferă servicii de backend în paralel pentru toți clienții conectați pe cele patru tipuri de aplicații și pune la dispoziție stocare de informații într-o bază de date cu ajutorul serviciului MySQL.

2.2 Principalele tehnologii folosite

2.2.1 Java

Implementarea aplicației client - Dispecer și Serverul au fost implementate cu ajutorul limbajului de programare Java.

Java este o tehnologie inovatoare lansată de compania Sun Microsystems în 1995, care a avut un impact remarcabil asupra întregii comunități a dezvoltatorilor de software, impunându-se prin calități deosebite cum ar fi simplitate, robustețe și nu în ultimul rând portabilitate. Denumită inițial OAK, tehnologia Java este formată dintr-un limbaj de programare de nivel înalt pe baza căruia sunt construite o serie de platforme destinate implementării de aplicații pentru toate segmentele industriei software.

Limbajul de programare Java a fost folosit la dezvoltarea unor tehnologii dedicate rezolvării unor probleme din cele mai diverse domenii. Aceste tehnologii au fost grupate în așa numitele platforme de lucru, ce reprezintă seturi de librării scrise în limbajul Java, precum și diverse programe utilitare, folosite pentru dezvoltarea de aplicații sau componente destinate unei anume categorii de utilizatori

Pentru implementarea celor doua entități din cadrul Serviciului S.D.A s-a folosit platforma de lucru **J2SE**. Aceasta este platforma standard de lucru ce ofera suport pentru crearea de aplicații independente și appleturi.

2.2.2 Android

Atat aplicația PACIENT care instantiază apelul S.O.S. cât și aplicația AMBULANȚĂ care este responsabilă cu preluarea apelului S.O.S., sunt aplicații create pentru utilizatorii dispozitivelor ce folosesc Android.

Android-ul este un sistem de operare pentru mobile dezvoltat de Google, bazat pe Linux kernel și proiectat inițial pentru mobilele cu touchscreen precum telefoanele inteligente și tablete. Este un produs open-source (putând fi dezvoltat de producătorii de dispozitive mobile cu extensii proprietare pentru a-și particulariza platforma), conceput pe ideea transformării

dispozitivelor mobile în adevărate mașini de calcul. In **Figura 4.** putem observa ciclul de viață al unei activități android.

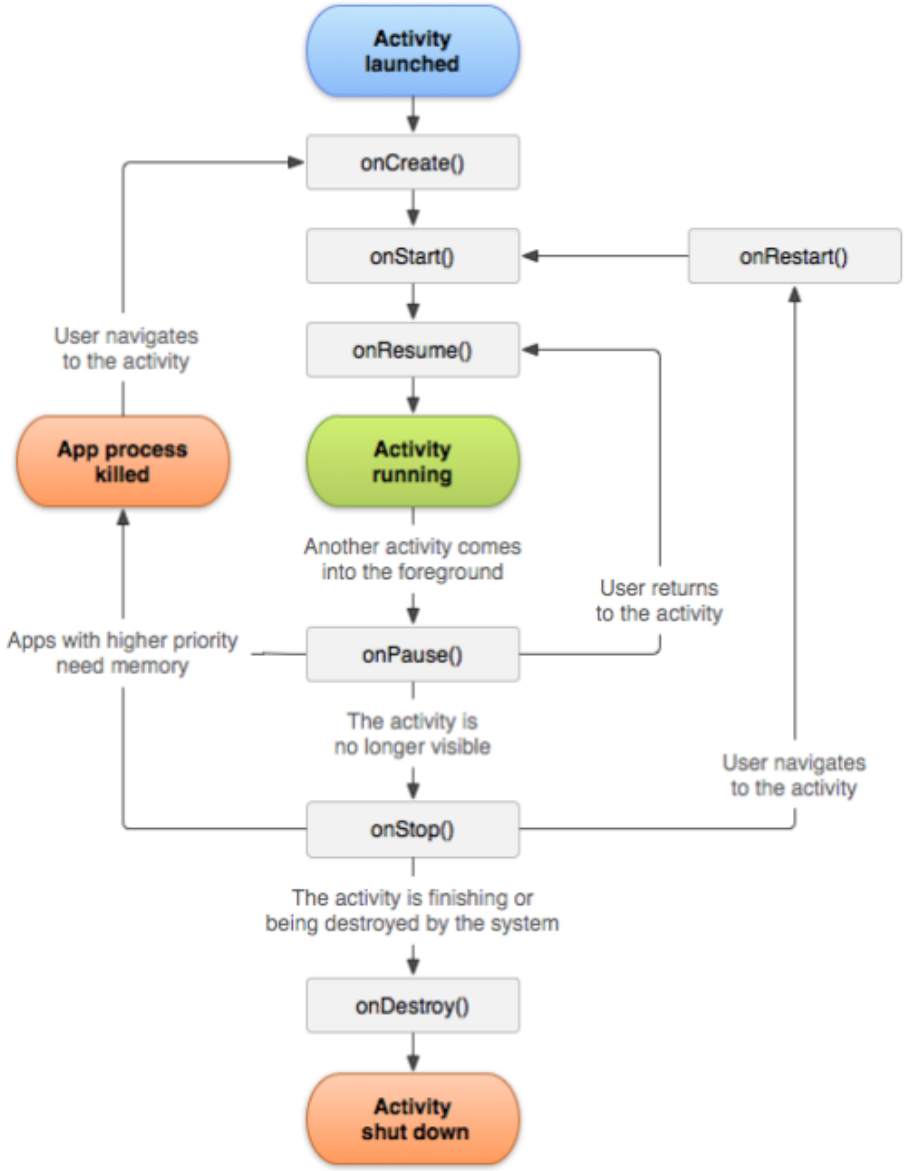

Figura 4. Ciclu de viață al unei activități

2.2.3 MySQL

Stocarea și gestionarea informațiilor este realizată cu ajutorul sistemului de gestiune a bazelor de date relaționale MySQL. Operațiile asupra bazelor de date, tabelelor, valorilor înregistrărilor etc., se specifică într-un limbaj declarativ: SQL (Structured Query Language.

Cu o arhitectura open source, MySQL este un sistem de gestiune a bazelor de date foarte rapid, optimizat pentru acces la date fiind larg folosit în cadrul soluțiilor de comerț electronic.

Sistemul S.D.A. folosește stocarea în baza de date, pusă la dispoziție prin intermediul MySQL, doar pentru date de autentificare și logare plus setările de profil care pot fi manipulate în funcție de accesul fiecărui utilizator. Deoarece MySQL pune la dispoziție un serviciu optimizat capabil să manipuleze pachete uriașe de date, serviciul S.D.A nu necesită optimizări adiționale celor deja existente.

Creare conexiune cu baza de date :

```
Private static final String driver = "com.mysql.jdbc.Driver";
private static final String url =
"jdbc:mysql://localhost:5555/serviciuldeambulante?autoReconnect=true&useSSL=false";
...
private Properties getProperties() {
    if (properties == null) {
        properties = new Properties();
        properties.setProperty("user", getUser());
        properties.setProperty("password", getPass());

    }
    return properties;
}
...
public Connection connect() {
    if (DBconnection == null) {
           try {
                Class.forName(driver);
                DBconnection=DriverManager.getConnection(url, getProperties());
             } catch (SQLException e) {
                   // TODO Auto-generated catch block
                   e.printStackTrace();
             }catch (ClassNotFoundException e1) {
                   // TODO Auto-generated catch block
               System.out.println("Database Connection error!");
                   e1.printStackTrace();
             }
       }
        return DBconnection;
   }
```

2.2.4 Programre socketi

Comunicarea între cei trei clienți și server este realizată cu ajutorul socketilor. Ca și abstractizare, un socket este interfața pe care sistemul de operare o pune la dispoziția aplicației pentru ca aceasta să poată comunica prin intermediul retelei cu o altă aplicație de pe un alt sistem. Un socket identifică în mod unic un capat (endpoint) dintr-o conexiune. Socketul client și socketul server formează o conexiune.

Instanțiere socket server :

```
private void openServerSocket() {
  try {
     this.serverSocket = new ServerSocket(this.Port);
  } catch (IOException e) {
     throw new RuntimeException("Cannot open port " + this.Port + ": ", e);
  }
}
```

Instanțiere socket client :

```
socket = new Socket(getHostName(), getPortNumber());
```

Creare conexiune server – client :

```
clientSocket = this.serverSocket.accept();
```

Scrierea și citirea de date pe canalul furnizat de socketi este realizată cu ajutorul obiectelor ObjectInputStream și ObjectOutputStream:

```
oin = new ObjectInputStream(socket.getInputStream());
oop = new ObjectOutputStream(socket.getOutputStream());
...
String authKey_flag_str = (String) oin.readObject(); // citire
oop.writeObject(cpryptedAuthkey); // scriere
```

2.3 Funcționarea Sistemului S.D.A.

2.3.1 Funcționalitatea principală – S.O.S

Prima și cea ma prioritară funcționalitate , după ce userul este logat în aplicație, este trimiterea semanlului de urgentă S.O.S.

Automat de activitate PACIENT – S.O.S ():

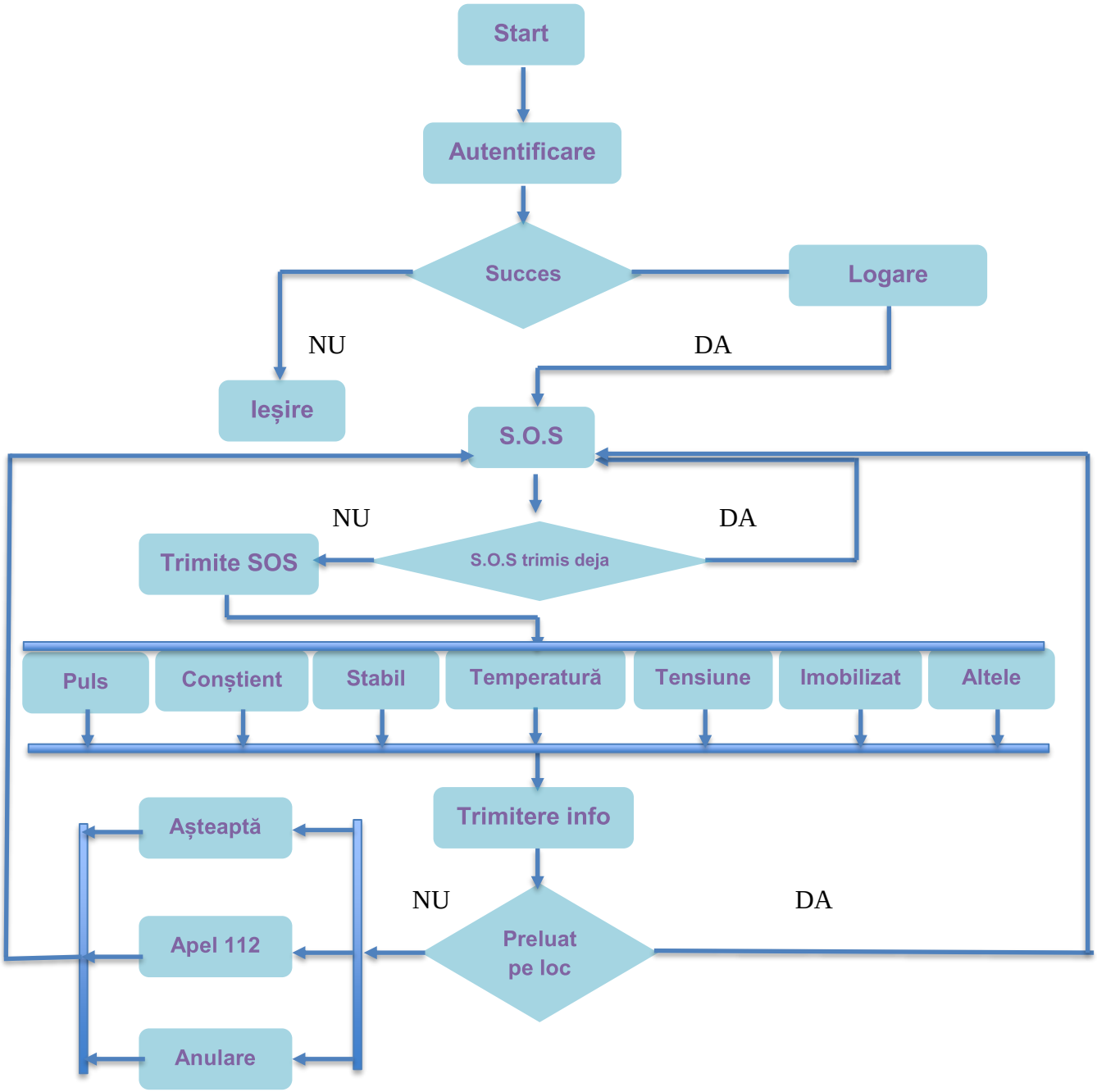

Figura 5. Automat de activitate Pacient SOS

Figura 6. Pacient

Pentru a trimite semnalul de urgență S.O.S. clientul trebuie să țină apăsat timp de 5 secunde , până telefonul încetează să vibreze , vibratie care pornește în momentul atingerii butonului S.O.S. (**Figura 6.**). Timpul de apăsare de 5 secunde a fost introdus ca o măsură de siguranță pentru atingerea din greșeală a butonului de semnal. În cazul în care pacientul nu ține apăsat timp de 5 secunde , aplicația o să arate un mesaj de atenționare „Țineti apăsat cel putin 5 secunde pentru a trimite semnalul SOS!”.

După cele 5 secunde de apăsare al butonului S.O.S. , aplicația trimite serverului semnalul de urgență împreună cu locația geografică. În acest moment serverul alege din lista ambulanțelor active cea mai bună variantă pentru locația pacientului. Dacă ambulanța nu poate ajunge în mai puțin de 40 de minute , pacientul este atenționat și i se sugerează să inițieze un apel la serviciul de urgentă 112.

In cazul în care ambulanța poate ajunge într-un timp acceptabil, aplicația pacient trece la pasul de introducere a informațiilor despre situația urgenței .

Pacientul poate introduce date pentru fiecare dintre cele șapte elemente (**Figura 7.**) : puls, conștient, stabil, temperatură, tensiune, imobilizat ți alte detalii sau poate să nu acceseze niciun element și să trimită pachetul de informații gol. Deși este recomandat să se completeze pachetul de informații despre starea pacientului, acesta poate fi ignorat în cazul în care apelul este făcut de un pacient instabil care nu e abil fizc sau coerent în gandire. În acest caz e deajuns apelul S.O.S. care trimite locația pacientului iar restul de informații despre cine a inițiat apelul se vor extrage din baza de date cu ajutorul datelor asignate profilului respectiv.

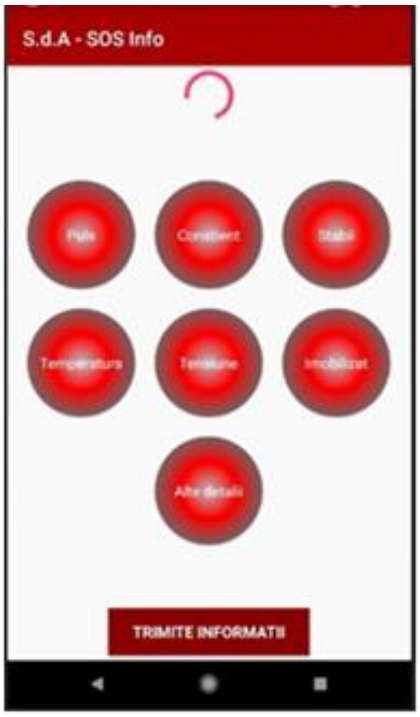

Figura 7. Informații urgență

Dupa trimiterea informațiilor aplicația pacient revine la activitatea principala S.O.S.

Automat de activitate Ambulanță – S.O.S (Figura 8.) :

Cea de-a doua aplicație ca și importanță este AMBULANȚA responsabilă cu preluarea urgențelor asignate automat de server.

Figura 8. Automat de activitate Ambulanță – S.O.S :

Când ambulanța este disponibila pentru a primi urgențe de preluat, se actionează butonul „Mod preluare urgențe” printr-o singură apăsare moment în care ambulanța trimite la server stausul de „SOS” care arată că poate fi luată în considerare în cazul unei urgențe noi.

Figura 9. Navigație

Atunci când serverul primește un apel de urgență S.O.S. și găsește ambulanța respectivă ca cea mai bună variantă pentru preluarea acelui pacient, se trimite către aplicația ambulanței datele de localizare primite de la pacient cu ajutorul cărora se pornește automat sistemul de navigație.

Folosirea navagației instalate local fie că este Waze sau cea oferită de Google Maps este o alegere mult mai bună decât implementarea propriului serviciu care la rândul lui are nevoie de API-uri precum cel de la Google sau Bing. Pe lângă asta, implementarea unui serviciu de navigație personal aduce un plus mare de costuri.

Serviciul local de navigație este pornit automat de aplicația AMBULANȚĂ. Tot ce e necesar pentru plecare la durm este acționarea butonului START în momentul în care navigația este instanțiată cu ruta către pacient.

Pe drumul către pacient, echipajul ambulanței poate verfica cu ajutorul butonului „Informații pacient” dacă acesta a trimis date adiționale despre starea în care se află.

Urmatorul pas în procedura de preluare urgențe, apare cand ambulanța ajunge la destinație și acordă primul ajutor persoanei sau persoanelor care necesită asistența medicală. Dupa ce situația urgenței a fost evaluată, echipajul ambulanței poate anula urgența din diferite motive sau în cazul în care pacientul necesită asistență medicală specializată în cadrul unui spital,

Figura 10. Situație urgență

se completează formularul cu situația pacientului în momentul de față.

Aplicația revine la activitatea principală dupa ce trimite datele sau dupa ce anulează urgența curentă.

Automat de activitate Dispecer – S.O.S (Figura 11.) :

În contextul procedurii S.O.S. Dispecerul are rol doar de monitorizare a urgenței și informațiilor asociate acesteia precum: cine este pacientul, ce ambulanță și personal se ocupă cu preluarea pacientului, datele trimise de către pacient despre starea sa sau datele trimise de ambulanță despre starea pacientului.

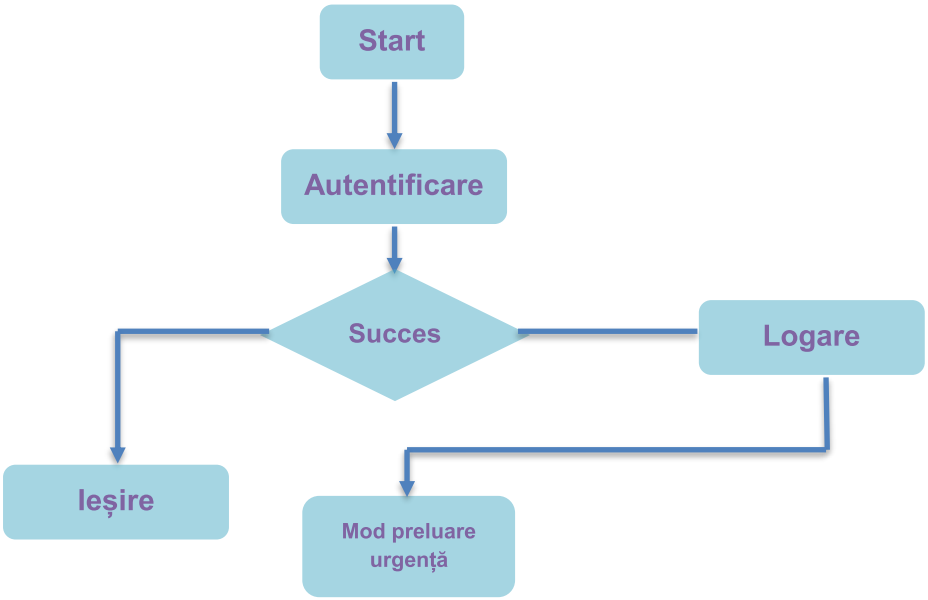

Figura 11. Automat de activitate Dispecer – S.O.S

Activitate Server – S.O.S :

În procedura S.O.S. serverul este entitatea care joacă rolul operatorului din cadrul call center-ului 112. Acesta gestioneaza fluxul de date și urgențe, comunicarea între ambulanță pacient și dispecer, dar în același timp asigură securitatea căilor de comunicare, sincronizarea thread-urilor care tratează fiecare client în parte și pune la dispozitie metode de calcul pentru cele mai bune alegeri de gestionare a urgențelor precum calculul distanței de parcurs între pacient și fiecare ambulanță disponibilă. Pe baza acestui criteriu se atribuie ambulanța cu cea mai mică distantă de parcurs.

Serverul creează câte un thread pentru fiecare client, și tratează separat și în paralel pe fiecare dintre aceștia. În momentul în care un pacient trimite semnalul S.O.S. serverul

calculează distanța pentru fiecare ambulanță disponibilă, atribuie ambulanța pacientului și trimite către aceasta datele primite de la pacient.

Dupa instanțierea procedurii S.O.S. serverul asigură schimbul de informații între pacient și ambulantă dar în același timp, trimite și către dispecer și medic informațiile necesare pentru monitorizarea urgențelor.

2.3.2 Funcționalități secundare

2.3.2.1 Pacient:

- Inregistrare pacient

  Procedura de înregistrare este implemntată în același mod pentru fiecare din cei patru clienți. Pentru a avea acces la formularul de înregistrare aplicația trebuie mai intâi să treacă de autentificare. În cazul în care nu se trece de autentificare, ceea ce înseamna că avem un client atacator care încearca să corupă serverul, accesul la formularul de înregistrare este oprit.

  Creare contului nou este realizat doar dacă pacientul introduce un CNP valid și care nu a mai fost utilizat și dacă username-ul nu se gaseste deja în baza de date.

- Schimbare parola

  Schimbarea parolei pentru aplicația client – PACEINT se face pe baza datelor de logare : user și parolă plus CNP-ul asignat profilului .

- Schimbare reședință

  În cazul în care pacientul își schimbă reședința principală, este obligat să facă update și la datele de adresă asociate profilului cu care se identifică în Serviciul S.D.A.

- Update stare de sănătate

  Funcția de actualizare a stării de sănătate permite utilizatorului să modifice câmpul „boli” din baza de date pentru userul său. Aceasta informație este importantă pentru stabilirea mai rapidă a unui posibil diagnostic și procedeului de acționare în cazul apelului S.O.S.

2.3.2.2 Ambulanță:

- Stare disponibilitate

  Schimbarea statusului unei ambulanțe în OCUPAT blochează posibilitatea de a pune ambulanța în „Mod preluare urgențe”

2.3.2.3 Alte funcționalități

- Stocare informațiilor în baza de date MySQL (**Figura 12 .**)

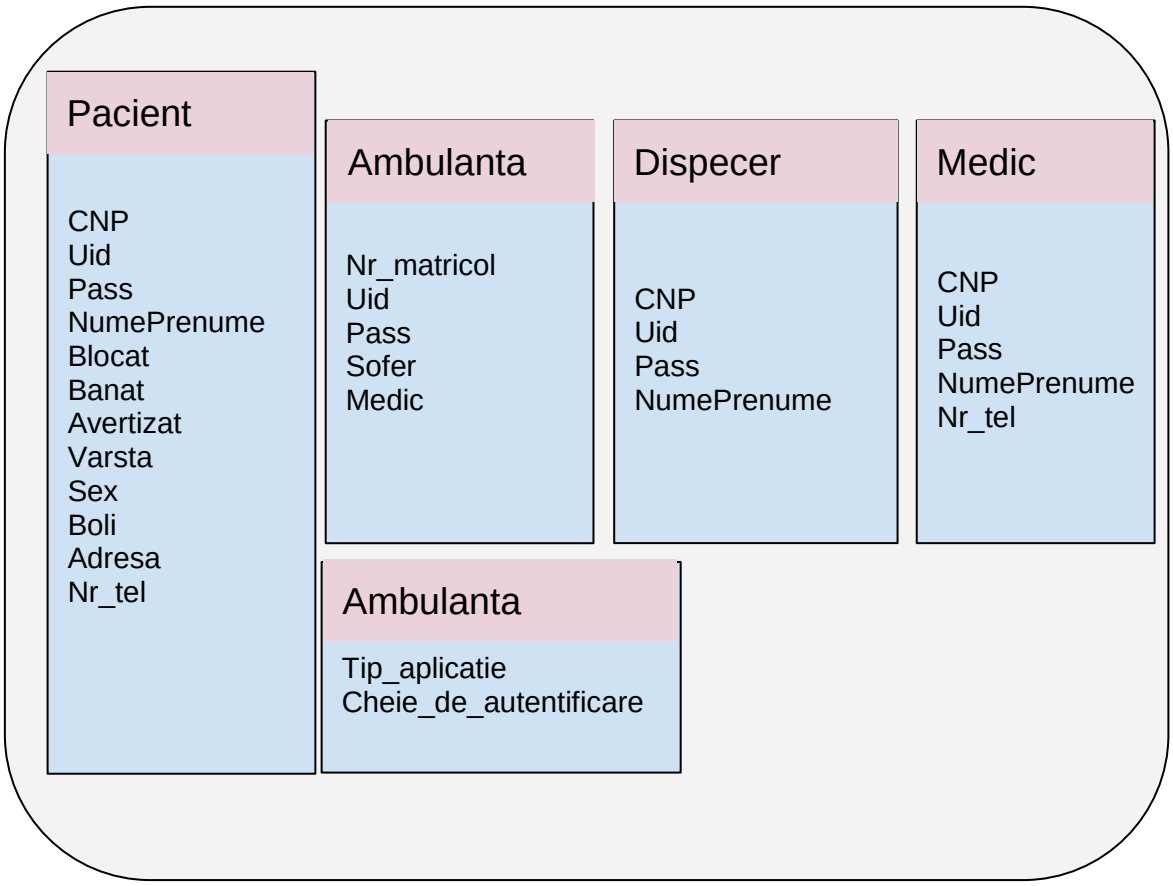

Figura 12 . Structura bazei de date

3. Securitatea sistemului Serviciul de ambulanțe

3.1 Riscurile implementării unui astfel de sistem

Fiind un serviciu public, responsabil cu asisitența medicală în cazurile de urgență, orice breșă de securitate poate avea consecințe uriașe care pot duce și la pierderea de vieți omenești.

Principalele riscuri de punerea imediată și fără testare a aplicației în lucru sunt :

-Prejudicii materiale pentru serviciul de ambulanțe

Acest risc este produs de posibilitatea unor apeluri false care să ducă până la deplasarea ambulanței la o locație falsă.

-Suprasolicitarea echipajelor de medicale

Apelurile false într-un număr mare și sincronizat au, deasemenea, ca și conseciță suprasolicitarea personalului responasbil cu preluarea urgențelor medicale.

-Pierderi de vieți omenești în cazul unei securități slabe și atacuri puternice

Cel mai important risc care trebuie redus la 0 pana în momentul punerii în funcțiune a Serviciului S.D.A. este pierderea de vieti omenești venită ca și consecita a întarzierii din cauza unor apeluri false realizate în același timp cu urgența reală.

3.2 Atacuri posibile

1) Atacuri asupra serverului

Cele mai cunoscute și folosite atacuri asupra serverelor sunt cele prin care atacatorul încearca să reducă performanța serverului sau chiar să blocheze serviciile oferite de acesta: **DOS** sau **DDOS**.

Un atac cibernetic de tip DoS (Denial of Service) sau DDoS (Distributed Denial of Service) este o încercare frauduloasă de a indisponibiliza sau bloca resursele unui calculator.

Deși mijloacele și obiectivele de a efectua acest atac sunt foarte diverse, în general acest atac este efectul eforturilor intense ale unei (sau a mai multor) persoane de a împiedica un site web, un server sau servicii din Internet de a funcționa eficient, temporar sau nelimitat.

Un atac de tip DoS - Denial of Service, se definește ca un atac ce are ca țintă restricționarea accesului la un serviciu disponibil în rețea pentru un potențial client. Termenul DDoS - Distributed Denial of Service - reprezintă o generalizare în sensul că atacul DoS este lansat în mod coordonat din mai multe puncte cu scopul principal de a crește sansele de succes. Punctele respective nu reprezintă neaparat stații aflate din start sub controlul direct al atacatorului, constituindu-se deseori în agenți corupți pentru lansarea atacului prin exploatarea diverselor vulnerabilități de sistem, infectarea cu troieni, etc.

Ca modalități efective de exploatare a victimei finale, atacurile DDoS sunt clasificate uneori în ***atacuri semantice*** care reprezintă categoria de atacuri ce se bazează pe vulnerabilitățile victimei la nivel de stivă de protocol sau implementare pentru a cauza întreruperea serviciului și ***de forță brută*** care sunt cele ce în mod normal nu au șanse de reusită doar pe baza exploatării unei vulnerabilități ci necesită în mod obligatoriu un volum crescut de trafic pentru a impiedica oferirea unui serviciu.

Deși atacurile de tip Denial of Service sunt considerate încălcări ale politicii de utilizare corectă a internetului elaborate de Internet Architecture Board (IAB) si deseori încălcări ale legislației din țara respectivă, numarul si gradul lor de complexitate crește de la zi la zi.

2) Atacuri asupra rețelei și a informațiilor ce circulă pe rețea

Atacurile din aceasta arie au ca țintă interceptarea informațiilor schimbate între server și client și folosirea lor în scop malițios sau chiar ilegal.

Pe langă aflarea unor informații senzitive schimbate pe rețeaua aplicației, atacatorii mai au ca țintă manipularea serverului sau a clientului cu ajutorul injectării de informații false în rețea acestea din urmă fiind folosite pentru requesturi false de date confidențiale.

Sniffing-ul și **Spoofing**-ul sunt doua dintre atacurile care ar putea pune în pericol securitatea Serviciului S.D.A.

3) Atacuri asupra bazei de date

SQL Injection este o tehnică de injectare de cod care exploatează o vulnerabilitate de securitate ce apare în stratul "baza de date" al unei aplicații. Vulnerabilitatea este prezentă atunci cand datele introduse de utilizator sunt incorect filtrate sau greșite, aceste lucruri ducând la o execuție neașteptată. Este o instanță dintr-o clasă mai generală de vulnerabilități care pot apărea ori de câte ori un limbaj de programare sau de scripting este încorporat în interiorul altuia.

Forme ale vulnerabilității :

- **Șiruri de caractere de control filtrate incorect :** Această formă de insertie SQL se produce atunci cand datele introduse de utilizator nu sunt filtrate pentru a scăpa de secventele de control și se trece direct într-o instructiune SQL. Acest lucru duce la manipularea potentialului de declarații efectuate pe baza de date de către utilizatorul final al aplicației.

- **Manipularea incorectă a tipurilor :** Această formă de isertie SQL apare atunci cand un câmp utilizator nu este puternic declarat (ca tip de dată) sau nu este verificat pentru constrangeri de tip. Acest lucru ar putea avea loc atunci când un câmp numeric urmează să fie utilizat într-o declarație SQL, dar programatorul nu face verificări pentru a valida ca datele introduse de utilizator sunt numerice.

- **Vulnerabilitățile în interiorul server de baze de date :** Uneori vulnerabilitățle pot exista în cadrul serverului de baze de date în sine, cum a fost cazul cu funcția MySQL mysql_real_escape_string (). Acest lucru permite unui atacator să efectueze cu succes un atac prin inserție SQL pe bază de caractere Unicode nocive, chiar dacă de intrarea utilizatorului este controlată. Acest bug a fost remediat o dată cu lansarea versiunii 5.0.22 (în data de 24.05.2006)

- **Insertia SQL oarba (Blind SQL Injection) :** Blind SQL Injection este utilizată atunci când o aplicație web este vulnerabilă la o inserție SQL, dar rezultatele de inserției nu sunt vizibile pentru atacator. Pagina cu vulnerabilitatea s-ar putea să nu fie una care afișează date, dar afișajul va fi diferit în funcție de rezultatele declarației logice inserate. Acest tip de atac poate deveni intensiv în timp deoarece o noua declarație trebuie concepută pentru fiecare bit recuperat.

- **Erori condiționale :** Acest tip de insertie SQL oarba cauzează o eroare SQL prin forțarea bazei de date să evalueze o declarație care cauzează eroare în cazul în care declarația WHERE este adevarată.

- **Întârzieri** : Întârzierile sunt un tip de insertie SQL oarba care provoacă motorul SQL să execute o interogare de lungă durată sau o intirziere de timp, în funcție de logica inserată. Atacatorul poate masura apoi timpul necesar pentru încarcarea paginii pentru a determina dacă declarația inserată este adevarată.

3.3 Metode de securitate implementate

Serviciul de ambulanțe pune la dispozitie o implementare riguroasă a celor mai eficiente metode de securitate folosite în aplicațiile java/android.

3.3.1 Securizarea căilor de cumunicare prin criptare RSA

Pentru a combate inteceptarea mesajelor comunicate între server și clienți, canalele de comunicare sunt securizate prin criptarea mesajelor în clar cu ajutorul algoritmului de criptare asimetrică RSA. În cazul unei interceptări , atacatorul are acces doar la mesajul criptat.

În criptografie, **RSA** este un algoritm criptografic cu chei publice, primul algoritm utilizat atât pentru criptare, cât și pentru semnatură electronică. Algoritmul a fost dezvoltat în 1977 și publicat în 1978 de Ron Rivest, Adi Shamir și Leonard Adleman la MIT și își trage numele de la inițialele numelor celor trei autori.

Puterea sa criptografică se bazează pe dificultatea problemei factorizării numerelor întregi, problema la care se reduce criptanaliza RSA și pentru care toți algoritmii de rezolvare cunoscuți au complexitate exponentială.

3.3.2 Autentificare

Aplicația dispune de doi pași de autentificare pentru a primi accesul seviciilor implementate:

1) Autentificarea aplicației client

Fiecare aplicație client primește o cheie de autentificare în momentul instalării care v-a fi stocată local. Când aplicația este pornită , primul procedeu de autentificare constă în trimiterea cheiei de autentificare către server, acesta o verifică cu cheia de autentificare din baza de date asociată fiecărui tip de aplicație iar dacă cheile se potrivesc, utilizatorul este autorizat pentru operația de logare

Măsura de prevenire a atacului DOS : Doar 3 încercări de autentificare în timp de 3 minute sunt acceptate.

2) Autentificarea utilizatorului

Utilizatorul criptează RSA cu cheia publică a serverului username-ul și parola, și le trimite prin retea. Serverul decriptează cu cheia privată datele primite, face interogare la baza de date și verifică dacă valorile primite corespund cu cele extrase din baza de date.
În caz afirmativ se oferă acces la serviciile aplicației client.

3.3.3 Combaterea apelurilor false

Pe lângă securitatea tehnică implementată, Serviciul de Ambulanțe este acoperit și de o serie de legi împotriva apelurilor de urgență false înscrise în Ordonanța de urgență *nr. 34/2008 privind organizarea și funcționarea Sistemului național unic pentru apeluri de urgență.*

La sfârșitul fiecărei luni se trimite un raport către sediul de poliție cu lista apelurilor false și eventualele prejudicii aduse în urma acestora.

Aceste reguli pot fi trecute în secțiunea “Termeni și condiții” prezentă la începutul instalării aplicației pe dispozitivul utilizatorului.

3.3.4 Declarațiile parametrizate împotriva atacului SQL Injection

Conectarea la baza de date MySQL se face controlat doar prin intermediul aplicației server , niciodată direct din aplicațiile client. Totuși în diferite funcționalități ale aplicației, clientii introduc date care ulterior vor fi folosite în interogarea bazei de date de către server.

Combaterea SQL Injection-ului în acest caz se realizează prin parametrizarea comenzilor sql.

O dată cu dezvoltarea de platforme, declarațiile parametrizate pot fi folosite deoarece funcționează cu parametri în locul includerii datelor introduse de utilizator în declarație. În multe cazuri, declarația SQL este fixa, și fiecare parametru este un scalar, nu un tabel. Intrarea unui utilizator este apoi alocată unui parametru.

3.3.5 Funcția de validare CNP

Pentru a evita înregistrarea unui utilizator cu mai multe conturi, am folosit o funcție de validare a CNP-ului.

Codul numeric personal sau **C.N.P.** este un cod numeric de 13 cifre, unic fiecărei persoane născute în România. Acesta este atribuit la nașterea fiecărui copil și este înregistrat pe certificatul de naștere. CNP figurează atât în actele de identitate (buletin de identitate sau carte de identitate) cât și în permisul de conducere auto.

Structura C.N.P.-ului (**Figura 13.**):

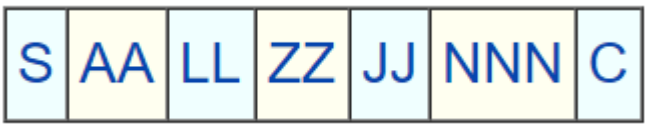

Figura 13. Structura C.N.P.-ului

- **S :** reprezintă sexul și secolul în care s-a născut persoana care posedă acel C.N.P. Persoanelor de sex masculin le sunt atribuite numerele impare iar persoanelor de sex feminin, numerele pare.

Prima cifră a C.N.P.-ului este: (sex bărbătesc / sex femeiesc)

- 1 / 2 - născuți între 1 ianuarie 1900 și 31 decembrie 1999
- 3 / 4 - născuți între 1 ianuarie 1800 și 31 decembrie 1899
- 5 / 6 - născuți între 1 ianuarie 2000 și 31 decembrie 2099
- 7 / 8 - pentru persoanele străine rezidente în RomâniaS reprezintă sexul și secolul în care s-a născut persoana care posedă acel C.N.P.[1] Persoanelor de sex masculin le sunt atribuite numerele impare iar persoanelor de sex feminin, numerele pare.

- **AA :** este un număr format din 2 cifre și reprezintă ultimele 2 cifre din anul nașterii. O persoană născută în anul 1970 va avea la AA 70.

- **LL :** este un număr format din 2 cifre și reprezintă luna nașterii persoanei.

- **ZZ :** reprezintă ziua nașterii în format de 2 cifre. Pentru zilele de la 1 la 9 se adaugă 0 înaintea datei. Spre exemplificare, o persoană născută în prima zi a lunii va avea codul 01.

- **JJ :** este un număr format din două cifre și este reprezentat de codul județului sau sectorului (în cazul municipiului București) (**Figura 14.**)în care s-a născut persoana ori în care avea domiciliul sau reședința în momentul acordării C.N.P.-ului.

Cod	Județ	Cod	Județ	Cod	Județ
1	Alba	19	Harghita	37	Vaslui
2	Arad	20	Hunedoara	38	Vâlcea
3	Argeș	21	Ialomița	39	Vrancea
4	Bacău	22	Iași	40	București
5	Bihor	23	Ilfov	41	București - Sector 1
6	Bistrița-Năsăud	24	Maramureș	42	București - Sector 2
7	Botoșani	25	Mehedinți	43	București - Sector 3
8	Brașov	26	Mureș	44	București - Sector 4
9	Brăila	27	Neamț	45	București - Sector 5
10	Buzău	28	Olt	46	București - Sector 6
11	Caraș-Severin	29	Prahova	51	Călărași
12	Cluj	30	Satu Mare	52	Giurgiu
13	Constanța	31	Sălaj		
14	Covasna	32	Sibiu		
15	Dâmbovița	33	Suceava		
16	Dolj	34	Teleorman		
17	Galați	35	Timiș		
18	Gorj	36	Tulcea		

Figura 14. Tabel Cod - Județ

- **NNN :** este un număr format din 3 cifre din intervalul 001 - 999. Numerele din acest interval se împart pe județe, birourilor de Evidență a Populației, astfel încât un anumit număr din acel interval să fie alocat unei singure persoane într-o anumită zi.

- **C :** este cifră de control (un cod autodetector) aflată în relație cu toate celelate 12 cifre ale C.N.P.-ului. Cifra de control este calculată după cum urmează: fiecare cifră din C.N.P. este înmulțită cu cifra de pe aceeași poziție din numărul 279146358279; rezultatele

sunt însumate, iar rezultatul final este împărțit cu rest la 11. Dacă restul este 10, atunci cifra de control este 1, altfel cifra de control este egală cu restul.

CONCLUZIILE LUCRĂRII

În concluzie, folosirea Sistemului S.D.A. ca o alternativă și de ce nu ca o înlocuire permanentă odată cu evoluția tehnologiei și îmbunatațirea funcționalităților puse la dispoziție în acest moment, este cea mai potrivită schimbare pe termen imediat și lung necesară în metodologia actuală de preluare a urgențelor .

Scopul punerii în funcțiune a S.D.A. este atins prin eliminarea unui seviciu mult prea instabil și ruginit care este înconjurat de factori distorsionanți precum transmitere verbală a mesajelor și integrarea în serviciul general de urgente 112 ceea ce îl face și mai dificil de manevrat,

Sistemul de aplicații S.D.A este doar un punct de plecare în automatizarea serviciilor publice de urgența punând la dispoziție o arie vastă de îmbunatatiri ce pot fi aduse nu numai în procedurile de acționare ci și în instrumentele folosite pe fiecare arie de activitate .

În opinia mea, implementarea unui sistem automat nu doar în serviciul de ambulanță cât și în serviciul de poliție , serviciu de pompieri sau SMURD, schimbă total dinamica ți parcursul societății spre o era tehnologică care poate fi controlată mult mai usor obținand cele mai bune rezultate în timpi cât mai optimi.

Idei de îmbunătățire a Serviciului de Ambulanțe :

1. REțELE NEURONALE

Implementarea unei rețele neuronale care să învețe timp de 2 luni comportamentul și starea de sanătate a unui pacient cu ajutorul unu dispozitiv smartwatch care poate înregistra informațiile urmatoare :

- **Puls** : Bătaie ritmică (pulsaţie) a arterelor datorată trecerii sângelui propulsat la fiecare contracţie cardiacă.
- **Tensiune :** Tensiunea arterială reprezintă produsul dintre debitul cardiac și rezistența vasculară periferică totală.
- **Glicemie :** Glicemia reprezintă nivelul zahărului din sânge. Glicemia poate fi determinată în momente diferite ale zilei (dimineaţa, prânz, seara sau în cursul

nopţii) precum şi în raport diferit faţă de masă – înainte sau după consumarea mancarii.

- **Temperatura corporala :** este menţinută constantă (homeotermic) printr-o reglare fiziologică.Temperatura corpului uman are o valoare medie de 37°C. Ea variază în mod normal de la 36,5°C (către ora 3 dimineaţa) la 37,2°C (către ora 6 scara).

Pe baza datelor învatate în cele 2 luni de antrenare, se poate construi un algoritm care detetectează când comportamentul iese din limitele sablonului stabilit în perioada de învățare.

În acel moment aplicația poate atenționa pacientul printr-o vibrație sau sunet sau chiar poate trimite un semnal S.O.S. automat în cazul în care comportamentul încalcă alarmant limitele patternului normal.

Ca datele să ramână consistente, perioada de antrenare trebuie repetată o data la 6 luni.

2. PRELUARE DATE AUTOMATE.

Odată cu evoluția tehnologiei și apariția gadget-urilor care pot prelua mai multe informații despre starea de sănătate a utilizatorului, formularul de trimitere a informațiilor despre starea pacientului se poate înlocui cu citiriea valorilor de la senzorii gadget-ului.

3. MONITORIZARE ÎN TIMP REAL – PACIENT

Datele preluate de la un dispozitiv mai ofertant tehnologic decat cele disponibile pe piață în momentul acesta se pot trimite și la dispecer sau la medicul de garda în mod recursiv de la începutul urgenței medicale pana la încheierea ei . În aces mod se poate genera un raport care să ofere deja un set de date medicale ca input incă de la internarea în spital.

4. LOCALIZARE VIZUALĂ PE HARTĂ A AMBULANȚELOR

O vizualizare în timp real a locației în care se află ambulanța asignată unei urgente medicale usurează metoda de monitorizare prin DISPECER dar poate fi și un stimulent vizual pentru pacient oferindu-i acestuia o stare de siguranță.

Bibliografie

1. „Curs practic de Java”, Cristian Frasinaru
2. „Android programming cookbook”, Chryssa Aliferi
3. „Rețele locale”, Răzvan Rughiniș , Răzvan Deaconescu, Andrei Ciorba, Bogdan Doinea, Editura Printech
4. https://profs.info.uaic.ro/~acf/java
5. https://profs.info.uaic.ro/~busaco
6. https://profs.info.uaic.ro/~eonica/
7. https://developer.android.com/guide/
8. https://developer.android.com/guide/components/activities/activity-lifecycle
9. https://www.csid.ro/dictionar-medical/
10. https://www.stsnet.ro/ro/statistici-lunare
11. https://ocw.cs.pub.ro
12. http://www.rasfoiesc.com/educatie/informatica/baze-de-date/
13. http://www.scritub.com/stiinta/informatica/baze-de-date/
14. http://www.securitatea-informatiilor.ro/solutii-de-securitate-it/
15. https://ro.wikipedia.org/wiki/