FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE
CATEDRA DE CALCULATOARE

APLICAŢIE PENTRU MANAGEMENTUL ŞI MONITORIZAREA UNUI CENTRU DE DATE

PROIECT DE DIPLOMĂ

Autor: **Daniel GROZA**

Coordonator : **şef lucrări ing. Cosmina Ivan**

2010

FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE
CATEDRA DE CALCULATORE

FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE
CATEDRA DE CALCULATOARE

SINTEZA

proiectului de diplomă cu titlul:

APLICAŢIE PENTRU MANAGEMENTUL ŞI MONITORIZAREA UNUI CENTRU DE DATE

Autor: **Daniel GROZA**
Coordonator: **şef lucrări Ing. Cosmina Ivan**

1. Cerinţele temei:
Proiectarea si implementarea unei aplicații de management si monitorizare a unui centru de date

2. Soluţii alese:
Pentru implementarea aplicației s-a ales platforma Java și framework-ul JSF datele fiind stocate intr-o bază de date MySQL. Comunicarea cu echipamentele de rețea se face folosind protocolul SNMP și Toolkit-ul oferit de proiectul open-source OpenDMK. Datele primite de la echipamente sunt stocate în arhive de tip Round Robin iar pe baza acestora se întocmesc grafice folosind unealta RRDTool. Aplicația trimite alerte pe email în cazul apariției anumitor evenimente în cadrul rețelei.

3. Rezultate obţinute:
Aplicația a fost implementată si instalată, rulează pe un server din centrul de date și monitorizează o parte din echipamente. Administratorul poate adăuga echipamente noi, șterge sau modifica echipamente existente în baza de date în scopul monitorizării acestora. Utilizatorii pot vizualiza graficele cu traficul realizat de echipamentele proprii.

4. Testări şi verificări:
Aplicația a fost testată, verificată și îmbunătățită pe parcursul testarii. Testele realizate și modificările aduse în urma testării sunt prezentate în capitolul 7 „Testare și rezultate experimentale”

5. Contribuţii personale:
Autorul a proiectat și implementat modulul de logică și control a aplicației, modulele ce interactionează cu celelalte sisteme și interfața grafică a aplicației.

6. Surse de documentare:
Biblioteca UTCN, Internet, cărți procurate online, manualele de utilizare a uneltelor și documentația api-urilor și a librăriilor folosite în cadrul aplicației.

Data:_____ Semnătura autorului_____
Semnătura coordonatorului_____

Cuprins

1. Introducere....9

1.1 Context.... 9
1.2 Motivație.... 9
1.3 Obiective.... 9
1.4Prezentarea lucrării.... 10

2. Studiu bibliografic....12

2.1 Conținut bibliografic.... 12
2.2 Evoluția centrelor de date.... 13
2.3 Alte aplicații de monitorizare.... 13
2.3.1 Cacti.... 14
2.3.2 Nagios.... 15
2.3.3 OpManager.... 16
2.3.4 Analiza comparativă.... 17

3. Fundamentare teoretica....19

3.1 Gestionarea centrelor de date.... 19
3.2 Monitorizarea echipamentelor.... 19
3.3 Tehnologii, standarde, protocoale și unelte utilizate în centre de date.... 20
3.3.1 Standardul IEEE 802.3 20
3.3.2 Standardul TIA-942.... 21
3.3.3 Protocolul SNMP.... 21
3.3.4 Java și JSF 22
3.3.5 MySQL 23
3.3.6 RRDTool 23
3.4 Arhitecturi software multinivel.... 24

4. Specificațiile și arhitectura aplicației....25

4.1 Gestionarea și monitorizarea centrelor de date.... 25
4.2 Schema bloc a sistemului.... 25
4.3 Cerințele aplicației....26
4.3.1 Cerințe funcționale.... 26
4.3.1 Cerințe non funcționale.... 26
4.4 Scenarii de utilizare....28
4.5 Arhitectura aplicației....34

5. Proiectarea de detaliu....36

5.1 Structura aplicației.... 36
5.2 Interfața cu utilizatorul.... 41

5.3 Structura bazei de date..... 41
5.4 Comunicarea cu alte sisteme..... 42
5.4.1 Comunicarea cu serverul MySQL..... 42
5.4.2 Comunicarea cu serverul de email..... 43
5.4.3 Preluarea datelor folosind protocolul SNMP 44
5.4.4 Comunicarea cu RRDTool..... 45

6. Utilizarea aplicației.....47

6.1 Echipamente necesare..... 47
6.2 Software necesar..... 47
6.3 Utilizare..... 47
6.3.1 Administrare 48
6.3.2 Utilizare client 53

7. Testare și rezultate experimentale.....55

7.1 Tehnologia folosită..... 55
7.2 Probleme întâmpinate și modul de rezolvare..... 55
7.3 Rezultate experimentale..... 56

8. Concluzii 57

8.1 Realizări..... 57
8.2 Avantaje aduse de soluție..... 57
8.3 Direcții de dezvoltare..... 59

Bibliografie.....60
Acronime.....62

Lista Figurilor

Figura 1.1 Imagine centru de date..... 10
Figura 2.1 Exemplu de grafice oferite de Cacti..... 14
Figura 2.2 Exemplu de interfață oferita de Nagios..... 15
Figura 2.3 Exemplu de interfață oferita de OpManager..... 16
Figura 3.1 Exemplu de etichetă server..... 19
Figura 3.2 Funcționare SNMP..... 22
Figura 3.3 Exemplu de bază de date de tip round robin..... 24
Figura 3.2 Schema bloc a sistemului..... 26
Figura 4.1 Diagrama Use Case pentru Administrator..... 28
Figura 4.2 Diagrama Use Case pentru Utilizator..... 29
Figura 4.3 Modelul arhitectural Three-tier..... 34
Figura 5.1 Diagrama sistemului..... 36
Figura 5.2 Diagrama de clase a aplicației..... 39
Figura 6.1 Pagina Login.jsf..... 48
Figura 6.2 Pagina Admin.jsf..... 48
Figura 6.3 Pagina Addswitch.jsf..... 49
Figura 6.4 Pagina Adduser.jsf..... 49
Figura 6.5 Pagina Addcompany.jsf..... 50
Figura 6.6 Pagina Adddevice.jsf..... 51
Figura 6.7 Pagina Settings.jsf..... 51
Figura 6.8 Pagina Graph.jsf..... 52
Figura 6.9 Exemplu de grafic..... 53
Figura 6.10 Pagina User.jsf..... 54
Figura 8.1 Graficul unui server cu probleme..... 58

Lista Tabelelor

Tabel 2.1 Analiza comparativa a soluțiilor existente descrise..... 17
Tabel 7.1 Rezultatul unui test de benchmark..... 56
Tabel 8.1 Avantajele aduse de soluție..... 58

1. INTRODUCERE

1.1 Context

Internetul este în continuă expansiune serviciile web fiind folosite din ce în ce mai mult. Pentru găzduirea echipamentelor pe care rulează site-urile, aplicaţiile şi alte servicii online sunt necesare reţele special amenajate care oferă conexiuni redundante la internet, surse neîntreruptibile de curent atât pentru servere cât şi pentru echipamentele de reţea precum şi alimentare de la reţeaua electrică din mai multe puncte sau de la un generator propriu. Spaţiul în care se află trebuie răcit şi bine ventilat pentru a asigura condiţii optime de funcţionare tuturor echipamentelor. Acest spaţiu, împreună cu echipamentele, poartă numele de „data center" sau centru de date şi impune existenta unor măsuri de securitate suplimentare pentru protecţia datelor.

Centrele de date, în prima parte a existenţei lor, nu constau decât în nişte încăperi mari conţinând calculatoarele primilor ani ai erei informatice. Odată cu explozia industriei microprocesoarelor şi a dezvoltării microcalculaoarelor aceste calculatoare au devenit tot mai simple, nemaifiind nevoie de echipe complexe de specialişti pentru a le opera. Apariţia pe piaţă a echipamentelor de reţea tot mai ieftine şi a standardelor pentru cablarea reţelelor a făcut posibil ca unele companii să îşi poată ţine serverele în spaţiul propriu. Pentru multe alte companii soluţia nu a fost practică sau eficientă, astfel a apărut migrarea către centre de date private[11].

Figura 1.1 prezintă o imagine a unui centru de date. Se poate observa așezarea și organizarea rack-urilor în rânduri și coloane. Echipamentele din rack-uri au carcase speciale, „rack-mountable", fiind clasificate în funcție de dimensiuni. Dimensiunea celor mai înguste servere este 1U, o unitate de aproximativ 5 centimetri. Pentru a se pute fixa în rack, dimensiunile serverelor mai voluminoase sunt multiple ale acestora (2U, 3U, 4U…). Serviciile oferite clienților sunt colocarea echipamentelor proprii sau închirierea acestora de la centrul de date[8]. Pentru clienții care au un număr mai mare de echipamente, inclusiv propriile echipamente de rețea, se poate oferi un rack întreg sau un grup de rack-uri. Majoritatea companiilor ce optează pentru aceste soluții sunt fie companii de web hosting, fie companii care au dezvoltat aplicații web distribuite.

1.2 Motivaţie

Pe măsură ce se extinde, ca urmare a solicitărilor de servicii din partea unui număr crescut de clienti, un centru de date trebuie foarte bine organizat şi monitorizat. O bună organizare ajută atât extinderea cât şi localizarea serverelor şi a problemelor hardware. Monitorizarea are un rol important, buna funcţionare şi menţinerea online a serverelor şi serviciilor depinzând de intervenţii cât mai rapide asupra problemelor de orice natură ce pot apare.

Având în vedere faptul că soluţiile tradiţionale pentru managementul şi monitorizarea unui centru de date folosesc mai multe aplicaţii independente iar soluţiile Enterprise sunt de multe ori prea costisitoare, a fost propusă proiectarea şi implementarea unei aplicaţii care să conţină atât partea de management cât şi partea de monitorizare a echipamentelor din centru de date.

1.3 Obiective

Organizarea unui centru de date trebuie realizată pe mai multe nivele:

- La nivelul fizic, serverele, cablurile şi porturile din switch-uri şi routere trebuie etichetate, iar cablurile de reţea şi de alimentare aranjate în paturi de cablu speciale.

Serverele sunt puse în rack-uri iar rack-urile sunt numerotate pentru localizarea cât mai rapidă. De obicei rack-urile sunt aşezate în rânduri uşurând astfel accesul la echipamente[18].

- La nivelul logic, există o diagramă a centrului de date cu locaţia rack-urilor, există o bază de date care conţine pentru fiecare server cel puţin locaţia sa exactă(rândul şi rack-ul), adresele IP alocate, datele de access (parola de root sau administrator), numele şi datele de contact ale utilizatorului sau a proprietarului, după caz.

Figura 1.1 Imagine centru de date

Lucrarea îşi propune proiectarea unei aplicaţii personalizate de management şi monitorizare al centrului de date, soluţiile actuale existente pe piaţă fiind fie incomplete, un sistem complet ar însemna instalarea mai multor aplicaţii independente, fie cu mai multe facilităţi decât ar fi nevoie dar foarte costisitoare.

Partea de management va facilita localizarea rapidă a serverelor şi va dezvălui administratorului toate detalile relevante legate de un anumit server şi utilizatorul sau proprietarul său. Se vor monitoriza atât lăţimea benzii folosite de servere la un moment dat cât şi traficul total făcut de acestea într-o anumită perioadă de timp.

Monitorizarea constă în întocmirea de grafice pentru toate porturile switch-urilor iar în cazul unor defecţiuni, funcţionări anormale sau alte evenimente nedorite detectate se vor timite alerte sub formă de email. Sistemul va fi scalabil permitând adăugarea de noi echipamente, modificarea sau ştergerea celor existente fără a aduce modificări codului aplicaţiei, toate datele ţinându-se într-o bază de date.

1.4 Prezentarea lucrării

Lucrarea conţine specificaţiile, detaliile de proiectare şi utilizare a sistemului de management şi monitorizare al centrului de date. Aceasta este împărţita în 8 capitole după cum urmează:

Introducerea este capitolul în care sunt prezentate contextul şi motivaţia care au determinat dezvoltarea acestui sistem precum şi obiectivele acestuia.

Capitolul 2, Studiul bibliografic prezintă o scurtă descriere a referinţelor bibliografice şi evoluţia centrelor de date care a determinat apariţia acestor sisteme. Tot în acest capitol sunt descrise câteva din actualele sisteme care îndeplinesc total sau parţial cerintele, împreună cu avantajele şi dezavantajele acestora.

În capitolul 3, Fundamentare teoretică sunt prezentate aspecte despre gestionarea centrelor de date şi monitorizarea echipamentelor subliniindu-se importanţa acestora. Aici sunt prezentate şi tehnologiile folosite în implementarea sistemului, standardele principale utilizate într-un centru de date, protocoalele şi aplicaţiile pe care se bazează sistemul.

Specificaţiile şi arhitectura sistemului este capitolul în care se prezintă arhitectura generală şi o schemă bloc a sistemului. Tot aici sunt descrise cerinţele detaliate atât în ceea ce priveşte gestionarea cât şi monitorizarea echipamentelor împreună cu motivaţia acestor cerinţe.

Proiectarea de detaliu prezintă detaliile de implementare ale aplicaţiei. Sunt descrise structura sistemului şi a bazei de date împreună cu diagramele modulelor şi modelele arhitecturale folosite. Se descrie deasemenea realizarea interfeţei cu utilizatorul şi alte detalii de implementare.

În capitolul 6, Utilizarea sistemului sunt prezentate cerinţele hardware/software ale echipamentelor monitorizate şi ale serverului pe care va rula aplicaţia împreună cu condiţiile de funcţionare. Se prezintă şi un scurt manual de utilizare a aplicaţiei.

Testare şi rezultate experimentale este un capitol destinat testării aplicaţiei şi a modulelor din care este compusă. Tot în acest capitol sunt descrise cateva din probleme întâlnite pe parcurs şi soluţionarea acestora.

Capitolul 8, Concluzii şi dezvoltare, descrie realizările şi avantajele aduse de sistem, direcţii de dezvoltare şi posibile imbunătăţiri.

2. STUDIU BIBLIOGRAFIC

Pentru documentarea necesară aspectelor teoretice specifice domeniului abordat au fost consultate o serie de surse bibliografice pe care le menționez in cele ce urmează. Acestea s-au constituit ca punct de plecare absolut necesar fundamentării teoretice a proiectului, unele reprezentând surse reale de inspirație pentru abordarea propusă.

2.1 Conținut bibliografic

Cartea "*core JavaServer Faces*"[1] prezintă avantajele tehnologiei JSF, în mod special rapiditatea cu care se face dezvoltarea interfetei cu utilizatorul. În carte sunt prezentate toate tag-urile JSF si raspunsuri la foarte multe întrebări.

În lucrarea „*Essential SNMP*"[2] este descris protocolul SNMP și utilizarea sa în managementul și monitorizarea rețelelor. Sunt prezentate arhitecturile sistemelor de monitorizare și o parte din aplicațiile de monitorizare existente.

Lucrarea „*Retele Locale de Calculatoare de la cablare la interconectare*"[3] tratează primele trei nivele arhitecturale ale modelului de referință OSI și abordează toate problemele teoretice și practice legate de rețelele de calculatoare.

În cartea „*Cascading Style Sheets, 2nd Edition*"[4] se prezintă în mod detaliat tehnicile pentru integrarea CSS în aplicatiile web, alcătuind un mic ghid ce acoperă standardele CSS1 și CSS2.

Lucrările „*Windows Admin Scripting Little Black Book, Second Edition*"[5] și „*Linux Shell Scripting with Bash*"[6] prezintă metode de creare a scripturilor si comenzile disponibile pentru crearea lor pentru sistemele de operare Windows și Linux.

Metodele prin care aplicațiile Java accesează date de pe un server MySQL folosind standardul JDBC sunt prezentate in cartea „*MySQL and Java Developer's Guide*"[7].

„*Data Center Fundamentals*"[8] descrie conceptele care stau la baza organizării infrastructurii unui centru de date.

În cărțile „*Designing Enterprise Applications with the JavaTM 2 Platform, Enterprise Edition*"[9] și „*J2EE Design Patterns*„[10] sunt descrise amănunțit tehnologiile, modelele arhitecturale și componentele arhitecturilor utilizate în proiectarea aplicațiilor Enterprise pe platforma Java.

Articolul „*History of the data center*„[11] prezintă istoria centrelor de date, de la primele forme de existență până în prezent.

Pagina web „http://httpd.apache.org/docs/2.0/programs/ab.html"[12] descrie utilitarul de benchmark ab si manualul său de utilizare.

Documentul „*MySQL 5.0 Reference Manual*"[13] este manualul de referință al utilizării serverului de baze de date MySQL.

Sursele librăriilor proiectului OpenDMK și documentația sa completă sunt disponibile la adresa web https://opendmk.dev.java.net[14]

Aplicația OpManager este descrisă amănunțit împreună cu toate feature-urile la adresa web [15] unde există și un demo al acesteia.

Documentația completă și sursele proiectului RRDJTool se gasesc pe pagina web „http://www.dnseurope.net/opensource/rrdjtool"[16].

La adresa web „http://oss.oetiker.ch/rrdtool/doc"[17] se găsește documentația utilitarului RRDTool, descrierea detaliată a fiecărei funcții și exemple demonstrative pentru fiecare din acestea.

Descrierea standardului TIA-942 este rezumată in documentul „*TIA-942 Data Center Standards Overview*"[18].

Pagina web „http://jcp.org/en/home/index"[19] prezintă descrierea mecanismelor pentru dezvoltarea specificațiilor tehnice standard în tehnologiile Java.

Articolul „http://www.eweek.com/c/a/Application-Development/OpenSource-Framework-Means-Happy-Trails-for-Java-Developers”[20] descrie framework-ul Trails şi avantajele acestuia.

2.2 Evolutia centrelor de date

Deşi centrele de date existente sub forma cunoacută acum pe piața serviciilor de profil au fost mediatizate şi perfecţionate în perioada exploziei dot com, spre sfârşitul anilor 90, ideea unzor astfel de soluții îşi are rădăcinile la începutul erei calculatoarelor. Primele calculatoare erau imense, maşini ce aveau dimensiunile unor camere, aveau nevoie de foarte mult spaţiu şi de un mediu controlat. Consumând multă energie se încălzeau iar pentru a putea menţine temperatura în parametri de funcţionare a fost nevoie să fie mutate în încăperi dedicate lor.

Aceste sisteme erau foarte scumpe şi multe dintre ele erau folosite în scopuri militare sau proiecte civile cu o importanţă majoră[11], securitatea lor devenind foarte importatntă. Camerele speciale în care se aflau facilitau siguranţa sistemelor şi permiteau un control strict al accesului la maşini.

Complexitatea calculatoarelor vechi necesita interconectarea multor componente folosind cablurice trebuiau să fie foarte bine organizate. Acest lucru a dus la apariţia standardelor care se folosesc şi în zilele noastre, evident îmbunătăţite şi adaptate.

În anii 1980, odată cu apariţia microcalculatoarelor, acestea se instalau în diverse locaţii, comapniile şi utilizatorii nu ţineau cont de cerinţele de mediu şi condiţiile de funcţionare, astfel în scurt timp s-a ajuns la pierderi de date iar organizarea informaţiei nu mai era posibilă. În ciuda apariţiei echipelor de specialişti care instalau, configurau şi menţineau aceste calculatoare, pentru industrie era nevoie de o soluţie.

Acest context a dus la apariţia centrelor de date private, cu atât mai mult cu cât în anii 1990 complexitatea sistemelor informationale a crescut iar aplicaţiile de tip client-server au devenit tot mai utilizate[11]. Serverele au început să ocupe locul vechilor calculatoare iar apariţia echipamentelor de reţea tot mai ieftine au dus la apariţia standardelor de cablare.

Pe măsură ce internetul devenea din ce în ce mai folosit, companiile au inţeles necesitatea prezenţei lor în internet, prezenţă ce presupunea conexiuni sigure şi rapide şi capabilitatea de a opera 24 de ore pe zi atât pentru mentenanţă cât şi pentru activarea sistemelor noi. Aceste noi cerinţe au dus la apariţia altor centre de date care la rândul lor au revoluţionat tehnologiile şi practicile de operare în industria IT.

Nu toate companiile şi-au permis să investească în asemenea centre de date fie din cauza costurilor fie datorită faptului că necesităţile lor nu se ridicau la acest nivel şi astfel au apărut centrele de date private. Aceste centre de date au adus soluţii noi, pe care majoritatea companiilor şi le pot permite.

2.3 Alte aplicatii de monitorizare

Datorită faptului că monitorizarea serverelor a fost necesară încă de la apariţia primelor calculatoare folosite în acest scop, au fost create mai multe aplicaţii care îndeplineau cerinţele administratorilor de sistem. În primă fază se monitorizau caracteristicile mediului ambiant şi ale sistemului pentru ca încăperea să îndeplinească toate condiţiile de funcţionare, apoi s-a început monitorizarea traficului, a pachetolr şi a altor caracteristici de reţea şi sistem, urmând ca odată cu apariţia centrelor de date de dimensiuni mari şi diversificării echipamentelor să fie nevoie şi de soluţii de management.

Pentru familiarizarea cu domeniul temei abordate au fost identificate o serie de instrumente și aplicații similare existente pe piață și au fost studiate. Au fost instalate și analizate sub aspectul funcționalității oferite cât și a raportului preț performanță, criteriu necesar pentru centrele de date de dimensiuni medii. Cele mai reprzentative solutii dintre cele identificate vor fi descrise succint in continuare.

2.3.1 Cacti

Cacti reprezintă un frontend pentru RRDtool, prima versiune a fost lansată în anul 2001, urmând ca pe parcursul anilor să fie îmbunătăţit, adus la zi cu noutăţile şi să se adauge capabilităţi noi. Este scris în PHP, cu ajutorul lui se pot crea, menţine şi actualiza grafice şi baze de date de tip RRA(Round Robin Archive).

Se pot seta căi către scripturi externe pentru colectarea datelor, sau se pot folosi scripturi interne. Se pot crea grafice fără a fi nevoie să se cunoască sintaxa comenzilor pentru RRDTool acestea putând fi afişate în mai multe moduri. Deasemenea se pot crea template-uri pentru sursele de date şi grafice, acestea putând fi aplicate pentru maşinile nou adăugate în sistem.

Avantajele sistemului sunt interfaţa intuitivă, uşor de folosit, posibilitatea aplicării de template-uri şi metode diferite de afişare a graficelor. Echipamentele sunt aşezate structurat şi există posibilitatea de căutare după nume.

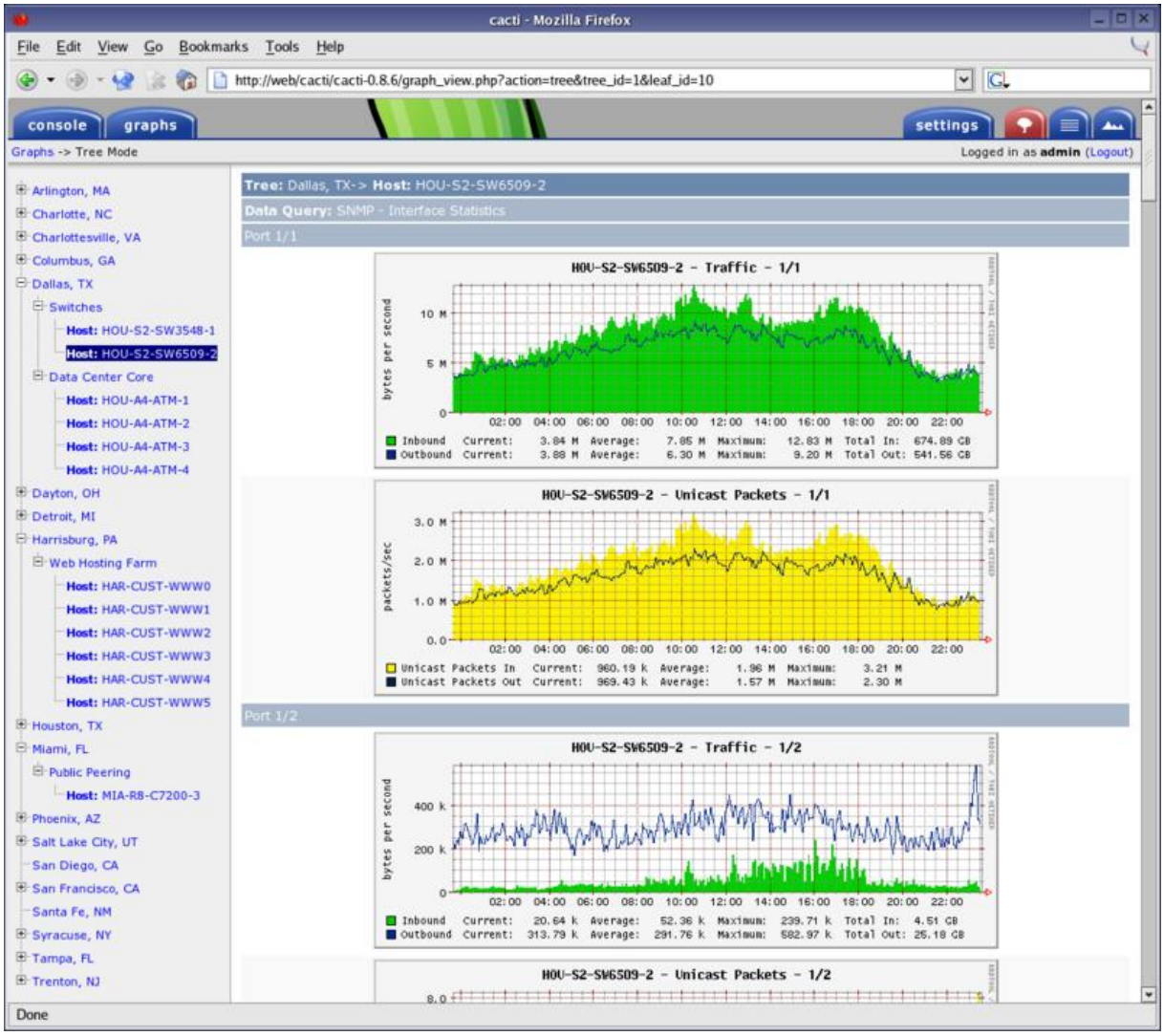

Figura 2.1 Exemplu de grafice oferite de Cacti

Dezavantajele sunt evidente: nu este un sistem complet de monitorizare, ci doar o componentă a unui sistem clasic alcătuit din mai multe aplicaţii idividuale; nu este capabil să trimită alerte în cazul unui incident şi îi lipseşte partea de management a dispozitivelor şi a clienţilor.

2.3.2 Nagios

Nagios este o aplicaţie open source pentru monitorizarea serverelor şi a serviciilor, capabil să trimită alerte în cazul apariţiei unei probleme. A fost iniţiat de către Ethan Galstad împreună cu o echipă de programatori şi lansat în luna martie a anului 1999.

Pe lângă monitorizarea serviciilor online (SMTP, POP3, HTTP, FTP, SSH şi altele) mai monitorizează şi starea şi resursele serverelor (încărcarea procesorului, consumul de memorie, utilizarea discurilor) şi log-urile de sistem.

Există o muţime de plugin-uri pentru monitorizarea temperaturii sau a alarmelor externe, monitorizarea de la distanţă prin SSH sau tuneluri encriptate, crearea de grafice pe baza datelor existente, se pot crea handlere ce intervin pentru rezolvarea proactivă a problemelor apărute şi se poate implementa un sistem redundant de monitorizare. Alertele pot fi trimise prin email, pager, SMS sau alte metode, în funcţie de plugin-ul ales.

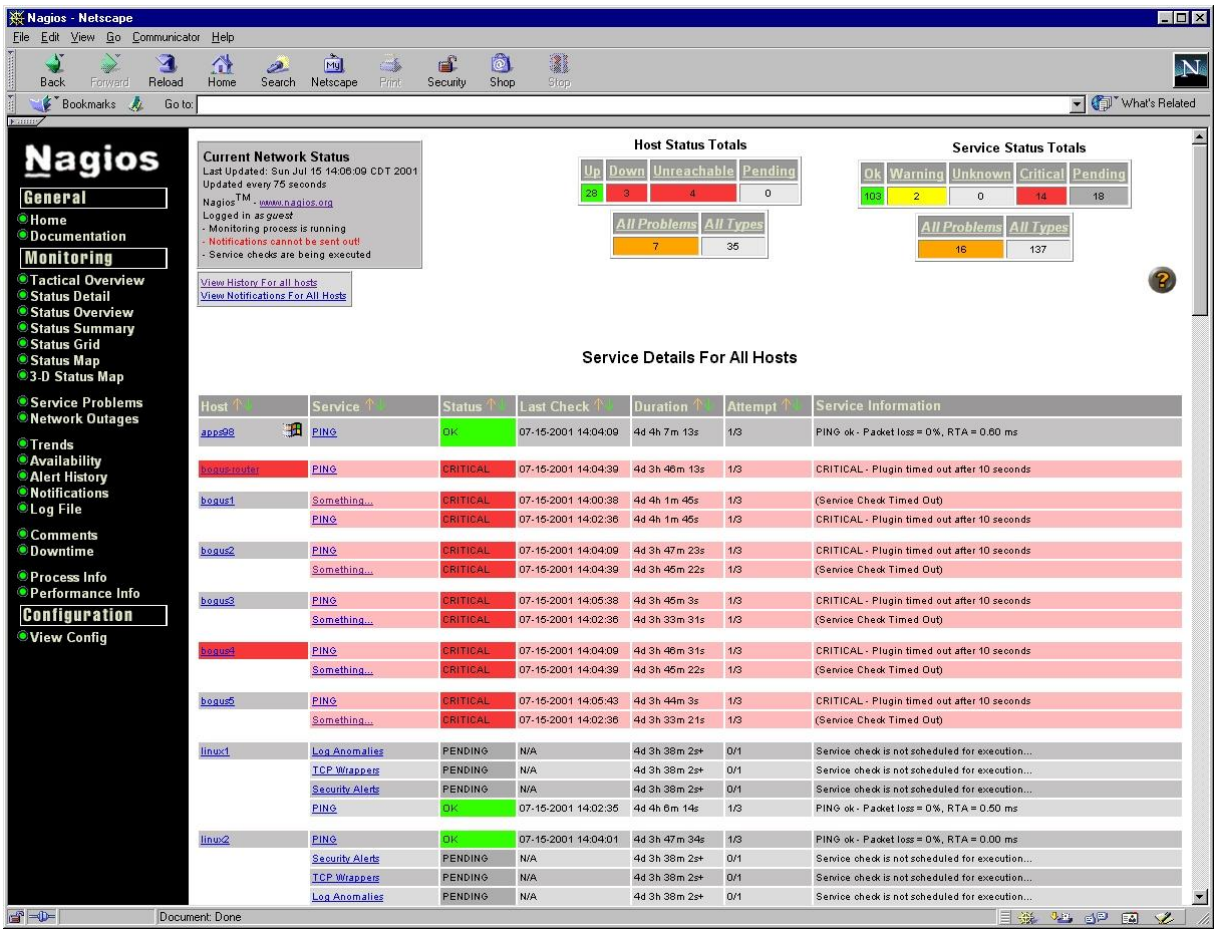

Figura 2.2 Exemplu de interfata oferita de Nagios

Dezavantajele solutiei sunt multitudinea de pluginuri ce trebuiesc instalate, configurate, testate si analizate pentru a le alege pe cele care corespund cerintelor, (fara a avea

experienta in utilizarea sa urmarea acestor pasi consumand mult timp) precum si lipsa partii de management sau pretul in cazul variantelor enterprise.

2.3.3 OpManager

OpManager este o aplicaţie complexă pentru managementul şi monitorizarea reţelelor, serverelor şi aplicaţiilor dezvoltată de compania ManageEngine. Printre funcţiile oferite de partea de monitorizare a reţelelor se numără: monitorizarea traficului, utilizării la un momnet dat şi a uptime-ului, descoperirea şi maparea automată a reţelei, precum şi monitorizarea stării echipamentelor de rețea. Deasemenea poate monitoriza serverele virtuale bazate pe Wmware şi starea serverelor fizice(utilizarea procesorului, a memoriei, a discurilor precum şi starea componentelor hardware)

Pentru monitorizarea aplicaţiilor ce rulează pe servere, aplicaţia poate monitoriza servere SQL, Active Directory, Servere de email bazate pe Exchange, servicii şi procese definite de administrator precum şi log-uri de sistem, website-uri, anumite fişiere sau directoare.[15]

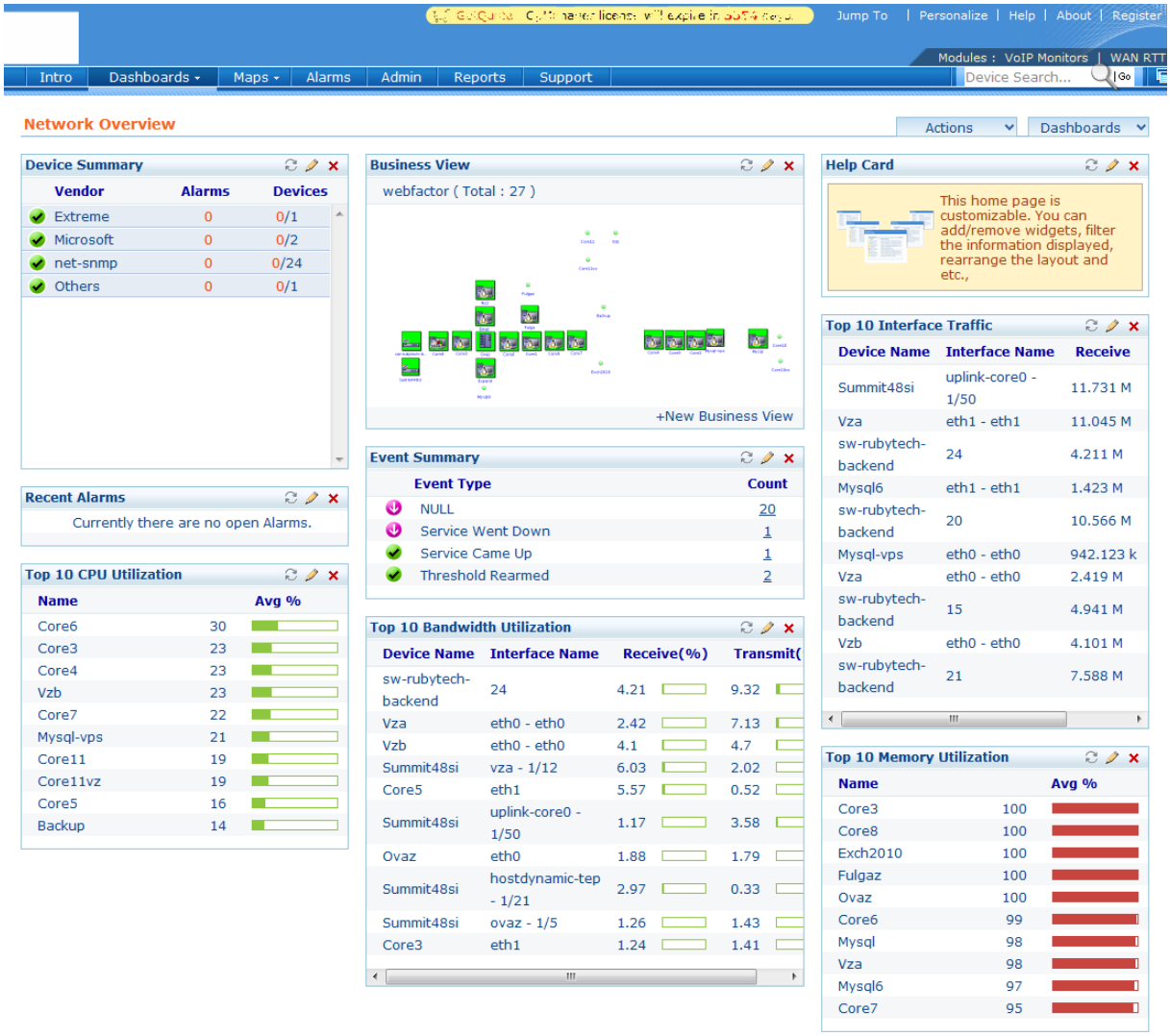

Figura 2.3 Exemplu de interfata oferita de OpManager

Alertele se pot trimite prin SMS sau email. Se pot seta valori minime sau maxime ale diferitelor variabile care atunci când sunt atinse sau depăşite se generează alertele[15]. Odată cu trimiterea alertelor, se pot lansa automat scripturi pentru remedierea problemei semnalate(dacă există această posibilitate).

Soluţia este cea mai completă, partea de management a clienţilor este oferită ca un plugin, singura problemă o reprezintă preţul pachetului care în funcţie de numărul serverelor monitorizate poate ajunge la zeci de mii de dolari americani.

2.3.4 Analiza comparativa

În tabelul de mai jos sunt prezentate prin comparaţie principalele caracterisitci ale celor trei aplicaţii. Au fost stabilite un set de caracteristici necesare realizării comparației și anume: monitorizarea echipamentelor(a traficului realizat, a stării serverelor sau a serviciilor ce rulează pe servere) și crearea de grafice, trimiterea de alerte sub orice formă (email, sms, internet messenger), managementul echipamentelor și al bazei de date cu clienților. Comparația este generică întru identificarea unor aspecte importante și necesare a fi incluse intr-un produs de monitorizare si management util unui centru de date.

Caracteristici	Cacti	Nagios	OpManager
Creare grafice	DA	DA	DA
Monitorizare echipamente	NU	DA	DA
Trimitere Alerte	NU	DA	DA
Management echipamente	NU	Partial	DA
Management clienti	NU	NU	DA
Pret	gratuit	gratuit/mediu	mare

Tabel 2.1 Analiza comparativa a solutiilor existente descrise

Cacti este doar o interfaţă pentru rrdtool, dar este gratuită şi foarte uşor de instalat, configurat şi folosit – este o soluție parțială fiind utilizat doar în combinaţie cu alte aplicaţii. Nagios, deşi are mai multe facilităţi decât Cacti, este destul de greu de configurat, dar poate fi o soluţie pentru un administrator experimentat. OpManager este de departe cea mai completă aplicaţie, dar preţul este foarte mare şi în licenţă sunt incluse unele facilităţi care nu sunt întotdeauna necesare.

În urma analizei aplicaţiilor prezentate anterior, a facilităţilor puse la dispoziţie de acestea şi a funcţionalităţilor folosite de un administrator în majoritatea intervenţiilor sale, am putut identifica un set minimal de elemente necesare unei aplicaţii de management şi

monitorizare a unui centru de date de dimensiuni medii. Setul cuprinde în mod obligatoriu crearea de grafice pentru traficul realizat de echipamente, monitorizarea cel puțin a existenței conexiunii pe porturile echipamentelor, trimiterea de alerte în cazul apariției anumitor evenimente precum și managementul echipamentelor – pentru identificarea și localizarea rapidă a acestora și managementul bazei de date cu clienți.

3. FUNDAMENTARE TEORETICĂ

3.1 Gestionarea centrelor de date

Gestionarea centrelor de date este o componentă foarte importantă pentru buna funcţionare a sa. Aceasta trebuie bine indeplinită atât la nivel fizic cât şi virtual. Odată cu creşterea numărului echipamentelor, ele trebuie bine organizate şi etichetate, deasemenea trebuie ţinută o evidenţă clară a tuturor componentelor rezervate schimburilor(în cazul defectării) sau îmbunătăţirilor sistemelor.

La nivel fizic aceasta se realizează în prima fază prin aşezarea echipamentelor în rack-uri, organizate pe rânduri şi numerotarea sau atribuirea de nume pentru fiecare rând şi coloană. Echipamentele se etichetează cât mai sugestiv pentru a ajuta la localizarea şi identificarea cât mai rapidă a acestora. Deasemenea cablurile de reţea sunt organizate şi aşezate în paturi speciale, având la rândul lor etichete cu numărul porturilor în care sunt introduse celelalte capete. De exemplu, o etichetă a unui server poate conţine următoarele elemente:

- ID – număr sau expresie unică pentru fiecare echipament
- Hostname – numele efectiv al ecchipamentului
- Client – date de identificare a clientului

Figura 3.1 Exemplu de eticheta server

Imaginea reprezintă un exemplu de etichetă pentru un server. ID-ul este compus dintr-o literă şi un număr, în cazul de faţă „D” înseamnă că serverul este dedicat, adică este în proprietatea centrului de date. În cazul echipamentelor colocate, aduse de clienţi, se foloseşte litera „C” iar „I” se foloseşte în ID-ul serverelor interne, cum ar fi un server de DNS, de email sau un nod de servere virtuale. Hostname-ul este indicat să se rezolve la adresa IP a serverului, acest lucru implică crearea unei înregistrări de tip A în zona de DNS a domeniului respectiv care să pointeze către adresa IP principală. Pentru identificarea clientului se foloseşte numele companiei care îl deţine.

Pe de altă parte, este nevoie de o aplicaţie care să conţină toate aceste date şi să ofere suport pentru adăugarea, modificarea, ştergerea şi extragerea lor. În momentul când sistemul de monitorizare sau clientul semnalează o problemă la unul din echipamente, identificarea locaţiei acestuia trebuie să se facă cât mai repede pentru a se interveni.

3.2 Monitorizarea echipamentelor

Monitorizarea este importantă din mai multe considerente. În primul rând, pentru buna funcţionare a echipamentelor, atât individual, prin detectarea problemelor legate de componentele hardware şi software cât şi colectiv deoarece printr-un comportament neadecvat un echipament influenţează funcţionarea celorlalte.

Problemele hardware intervin datorită uzurii şi a defectelor de fabricaţie a componentelor, fiind nevoie să se intervină pentru înlocuirea acestora. Un HDD defect poate duce la pierderi de date, pe când alte componente precum memoria sau plăcile de reţea pot provoca perioade de downtime neprogramat. Problemele software, apărute de multe ori în urma configurării defectuoase sau a update-urilor automate pot provoca la rândul lor întreruperi ale serviciilor, sau, mai rar pierderi de date. O problemă detectată la timp şi rezolvată cât mai eficient minimizează pagubele ce pot fi cauzate clienţilor.

Serverele compromise sunt deasemenea un risc deoarece de cele mai multe ori de pe acestea se iniţiază activităţi ilegale de tip DOS, flood, port scaning sau spam, care pot afecta buna funcţionare a celorlalte servere sau pot incomoda utilizatorii acestora.

În al doilea rând, monitorizarea traficului este importantă deoarece lăţimea de bandă este limitată şi taxată. Astfel, pentru acoperirea cheltuielilor centrului de date, există pachete oferite spre vânzare clienţilor, pachete ce au cantităţi de trafic exacte, incluse în abonament. Deasemenea, în cazul maşinilor virtuale, care de cele mai multe ori împart o singură placă de reţea, monitorizarea şi limitarea traficului duce la buna funcţionare a tuturor serverelor de pe un nod.

3.3 Tehnologii, standarde, protocoale si unelte utilizate in centre de date

Au fost identificate mai multe standarde, tehnologii şi protocoale utilizate într-un centru de date de dimensiuni mici şi medii, printre acestea amintim standardul IEEE 802.3, standard ce defineşte Ethernet-ul, TIA-942, satandard care descrie infrastructura unui centru de date şi protocolul SNMP, utilizat în comunicarea cu echipamentele de reţea în scopul monitorizării acestora. Pentru dezvoltarea aplicaţiei am folosit un set de tehnologii şi tool-uri care cuprinde Java Enterprise Edition împreună cu framework-ul JSF, utilizate în dezvoltarea aplicaţiilor web, MySQL, un sistem de gestiune a bazelor de date foarte cunoscut şi des utilizat şi RRDTool, o unealtă folosită pentru a stoca date şi a genera grafice folosind aceste date stocate.

3.3.1 Standardul IEEE 802.3

IEEE(Institute of Electrical and Electronics Engineers) este o organizaţie internaţională, non-profit înfiinţată în 1963 orein reuniunea IRE(Institute of Radio Engineers) şi a AIEE(American Institute of Electrical Engineers). Organizaţia, împreună cu IEEE Standards Association dezvoltă standarde legate de tehnologia bazată pe electricitate. Printre domeniile care utilizează aceste standarde se numară: energia electrică, tehnologia informaţiei, telecomunicaţiile, transportul, nanotehnologia şi multe altele.

IEEE 802 este standardul ce acoperă reţelele locale(LAN) şi metropolitane(MAN) acestea fiind compuse din Familia Ethernet, Token ring, Wireless LAN/MAN şi altele. Ethernet-ul este o familie de tehnologii ce se aplică în reţelele locale(LAN) [3].

IEEE 802.3 este colecţia de standarde ce definesc Ethernet-ul, mai precis standarde pentru cablajul şi semnalele nivelului fizic al Modelului de Referintă OSI şi pentru MAC(Media Access Control) şi formatul de adresare al nivelului 2(nivelul legăturilor de date) al stivei. Acest standard stă la baza arhitecturii reţelei în centrul de date, reţea compusă dintr-o colecţie de VLAN-uri.

3.3.2 Standardul TIA-942

TIA(Telecommunications Industry Association) este o asociaţie organizaţie fondată în anul 1988 ce are domeniul de activitate format din oportunităţi de afaceri, evoluţia pieţelor, certificări şi dezvoltarea de standarde.

TIA-942 este standardul ce defineşte *infrastructura unui centru de date*, în mod special din privinţa sistemului de cablare şi al design-ului reţelei, dar acoperă şi locaţia, răcirea, alimentarea cu energie electrică şi amenajarea sa precum şi considerente legate de mediu.

Standardul împarte centrele de date în 4 categorii:[18]

- **Tier1 –** Disponibilitatea 99.671%
  - Pasibil de întreruperi datorate activităţilor atât plănuite cât şi neplănuite
  - Circuit unic pentru răcire şi alimentare cu energie electrică
  - Fără componente redundante
  - Downtime anual 28.8 ore
- **Tier2 –** Disponibilitatea 99.741%
  - Mai puţin pasibil de întreruperi datorate activităţilor plănuite şi neplănuite
  - Circuit unic pentru răcire şi alimentare cu energie electrică
  - Include componente redundante N+1
  - Include podea înălţată, UPS şi generator
  - Downtime anual 22 ore
- **Tier3 –** Disponibilitatea 99.982%
  - Activităţile plănuite nu provoacă întreruperi, dar cele plănuite vor provoca
  - Mai multe circuite pentru răcire, unul singur activ
  - Include componente redundante N+1
  - Include podea înălţată, UPS şi generator
  - Downtime anual 1.6 ore
- **Tier4 –** Disponibilitatea
  - Activităţile plănuite nu provoacă întreruperi, iar centru de date poate susţine cel puţin un eveniment neplănuit grav fără apărea întreruperi
  - Mai multe circuite active pentru răcire şi alimentare cu energie electrică
  - Include componente redundante 2(N+1)
  - Include podea înălţată, UPS şi generator
  - Downtime anual 0.4 ore

3.3.3 Protocolul SNMP

Simple Network Management Protocol este o parte a suitei de protocoale TCP/IP şi este folosit de sistemele de management ale reţelelor pentru a monitoriza dispozitivele active şi a aduce în atenţia administratorilor anumite condiţii ale echipamentelor sau evenimente apărute în reţea. Acest protocol este o dezvoltare a Simple Gateway Monitoring Protocol(SGMP). Variabilele accesibile prin SNMP sunt organizate ierarhic într-o structură numită Management Information Base(MIB) şi se numesc Object Identifiers(OID). Fiecare OID reprezintă o variabilă a dispozitivului care poate fi citită sau după caz scrisă[2].

Exemple de OID specificat în RFC 1213:

- 1.3.6.1.2.1.1.3 – sysUpTime – reprezintă perioada de timp de la ultima restartare a dispozitivului.
- 1.3.6.1.2.1.2.1 – ifNumber – este un întreg ce reprezintă numărul tututor intrfeţelor dispozitivului.

- 1.3.6.1.2.1.2.2.1.14 – ifInErrors – este un întreg păstrat într-un numărător pe 32 de biţi ce contorizează erorile de intrare pe interfețe.

Versiunea 1 a protocolului a fost definită în RFC 1155 şi RFC 1157, urmând ca în 1992 să apara versiunea 2c care prezintă o securitate primitivă şi performanţe crescute. V2c este definită în RFC 1901-1908, 1909 şi 1910. Ultima versiune, v3, conţine opţiuni avansate de securitate oferite de un mecanism de control al accesului numit View-based Access Control Model(VACM). Protocolul SNMP este construit peste UDP, care este un protocol de transport fără conexiune.[2] Formatul datelor (PDU) pentru SNMP este definit în acord cu sintaxa ASN.1

Cinci tipuri de PDU-uri SNMP sunt definite astfel: GetRequest, GetNextRequest, SetRequest, Response şi Trap. GetRequest şi GetNextRequest sunt folosite pentru a aduce informaţie de administrat de la agent, SetRequest este folosită pentru a stoca sau modifica informaţia de administrat.

Comanda GetRequest este folosită pentru a citi date, variabila solicitată trebuie să fie specificată prin OID-ul ei. GetNextRequest se foloseşte pentru a citi valori consecutive aflate într-un tabel. SetRequest este comanda cu ajutorul căreia se pot seta variabile ale obiectului gestionat, comanda se poate utiliza doar dacă cel ce o lansează are drepturi de scriere. Dacă toate cele trei comenzi de mai sus sunt folosite doar de managerul SNMP, comanda GetResponse este folosită de agenţi, fiind comanda care returnează valoarea solicitată sau un mesaj de eroare. Trap-urile sunt folosite tot de către agent, acestea fiind trimise spre manager fără a fi solicitate, dând agentului abilitatea de a notifica staţia de gestiune la apariţia evenimentelor semnificative.

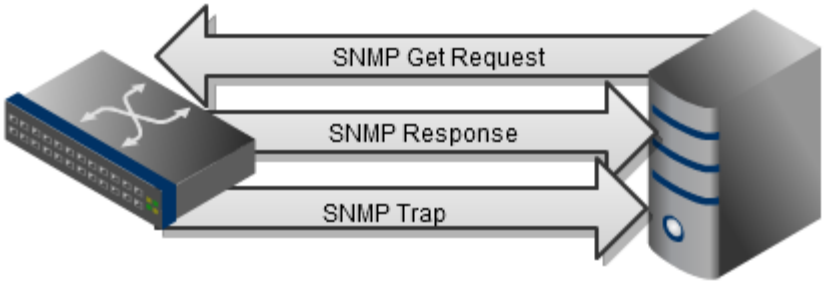

Figura 3.2 Functionare SNMP

Protocolul SNMP a fost standardizat pentru managemenul echipamentelor din rețelele bazate pe IP. Tot mai multe echipamente prezente în aceste rețele oferă suport pentru comunicarea pe SNMP, printre acestea se numără echipamentele specifice precum routere, switch-uri sau servere, dar și dispozitive ca imprimante, modem rack-uri sau surse neintreruptibile de curent(UPS)[2].

3.3.4 Java si JSF

Tehnologia Java a devenit un sistem software complet ce reprezintă diferite valori pentru diverse tipuri de utilizatori oferind dezvoltatorilor alegerea platformei în funcţie de necesităţi: pentru dispozitive mici şi mobile (Java Micro Edition), pentru aplicaţii destinate calculatoarelor personale (Java Standard Edition) sau pentru aplicaţii Enterprise şi WEB (Java Enterprise Edition).

Când majoritatea aplicaţiilor au trecut pe arhitecturi de tipul client-server, programatorii au început să caute metode pentru a simplifica dezvoltarea de proiecte ce folosesc tehnologii similare şi au structuri asemănătoare, punându-se astfel bazele framework-

urilor software moderne. Unele organizaţii au construit propriile framework-uri care satisfac cerinţele specifice domeniului în care lucrează, dar au apărut şi framework-uri open source precum Struts, Spring, JSF, Tapestry, Makumba, Seam sau Trails, care a fost numit de către fondatorul său, Chris Nelson, un metaframework sau framework de framework-uri (utilizează Hibernate pentru persistenţa, Tapestry pentru componente web, Spring pentru rezolvarea dependinţelor şi Acegi pentru securitate)[20].

JSF este un framework dezvoltat folosind Java Community Process, un mecanism de dezvoltare a specificaţiilor standard tehnice pentru Java.[19] Pentru a introduce o specificaţie nouă se folosesc documente formale care descriu specificaţiile şi tehnologiile respective. Aceste documente se numesc Java Specification Requests (JRSs). De exemplu JSF versiunile 1.0 şi 1.1 sunt descrise în JSR 127, versiunea 1.2 în JSR 252 iar versiunea 2.0 în JSR 314.

Printre avantajele oferite de JSF se număra: uşurinţa în a crea interfaţa cu utilizatorul, separarea clară între nivelul de logică şi cel de prezentare al aplicaţiei, arhitectura extensibilă(componentele şi funcţionalităţile se pot customiza), cicluri de dezvoltare mai scurte datorită separării între logică şi prezentare, suport pentru limbaje internationale şi existenţa aplicaţiilor care oferă unelte pentru a crea interfaţa cu utilizatorul[1]. Aceste avantaje au condus la alegerea tehnologiei pentru implementarea aplicației.

3.3.5 MySQL

MySQL este un sistem de gestiune a bazelor de date(SGBD) relaţional distribuit sub GNU General Public License. În prezent este cel mai popular SGBD open-source, fiind o componentă a stivei Linux, Apache, MySQL, Php(LAMP). Programul rulează ca un server oferind acces la bazele de date mai multor utilizatori simultan.

Deşi este folosit foarte des împreună cu limbajul de programare PHP, cu MySQL se pot construi aplicaţii în aproape orice limbaj. Există multe scheme API disponibile pentru MySQL ce permit scrierea aplicaţiilor în numeroase limbaje de programare pentru accesarea bazelor de date din MySQL, cum are fi: C, C++, C#, Borland Delphi, Java, Perl, Python, FreeBasic, Lua, etc. De foarte multe ori, aplicaţii relativ simple nu au nevoie decât de comenzile standard SQL (Select, Insert, Update, Delete, Create şi Drop) pentru a opera cu bazele de date.

S-a optat pentru folosirea acestui SGBD datorită avantajelor sale. Câteva din principalele avantaje sunt: MySQL este un sistem open-source și gratuit, portabil – rulează pe majoritatea sistemelor de operare, rapid – algoritmii și tehnicile de indexare sunt foarte eficiente, flexibil – se pot alege tipurile de structuri de date folosite, ușor de folosit și configurat – sintaxa este simplă și bine documentată și nu în cele din urmă este accesibil din alte limbaje și sisteme prin intermediul API-urilor[7].

3.3.6 RRDtool

RRDtool a fost scris de către Tobi Oetiker ca un înlocuitor pentru Multi Router Traffic Grapher (MRTG – o aplicaţie monitorizare a traficului scrisă în perl), software-ul este liber şi disponibil în termenii licenţei GNU General Public License. Numele este un acronim pentru Round Robin Database Tool.

Round robin este o tehnică ce lucrează cu cantităţi fixe de date şi un pointer le elementul curent. Pentru ilustrare se poate considera că datele sunt aranjate într-un cerc iar o săgeată îndreptată dinspre centru spre margine este pointerul. Când o dată este citită sau scrisă se trece la următorul element. Datele fiind aşezate sub forma unui cerc nu există început sau sfârşit, astfel pointerul se poate deplasa către următorul element la infinit, refolosind locaţia

cea mai veche. În acest fel dimensiunea bazei de date va rămâne constantă evident în detrimentul pierderii datelor vechi. Într-o baza de date round robin se pot stoca date de orice fel, cu condiţia ca acestea să fie numerice şi să fie măsurabile în timp, adică să reprezinte serii de timp[17].

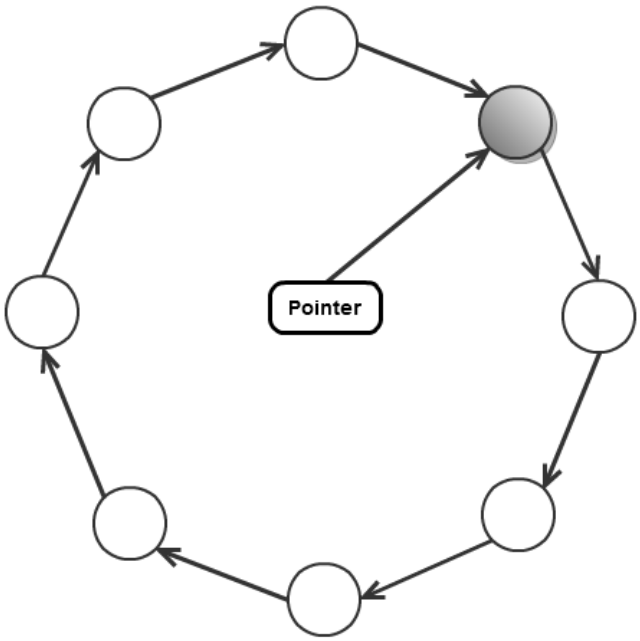

Figura 3.3 Exemplu de bază de date de tip round robin

RRDtool are funcţii cu ajutorul cărora se pot crea, modifica, actualiza, exporta baze de date de tip round robin, funcţii de preluare a anumitor informaţii legate de acestea sau de datele stocate. Programul mai are si o altă facilitate prin care recuperează datele stocate şi le afişează sub formă de grafice.[17]

Cele mai importante funcţii sunt:

- create – crează o nouă bază de date de tip round robin
- update – reţine o nouă valoare în baza de date, aceasta fiind scrisă în locaţia valorii celei mai vechi
- graph – creează un grafic pornind de la datele uneia sau a mai multor baze de date transmise ca parametri
- dump – exportă datele din baza de date în format ASCII.
- restore – reface o bază de date dintr-un fişier în format XML, împreună cu dump, ajută la migrarea unor baze de date de pe o arhitectură pe alta.
- fetch – returnează datele dintr-o anumită perioadă de timp

Funcţiile utilizate în cadrul aplicaţiei sunt create, update şi graph

3.4 Arhitecturi software multinivel

Există trei modele de dezvoltare a unei aplicaţii de tip client-server:

Arhitectura pe un nivel (one tier architecture) apare când serverul şi clientul sunt una şi aceeaşi aplicaţie. Acest model descrie aplicaţii relativ simple dezvoltate în întregime într-un singur mediu de programare. Datele, formularele folosite pentru introducerea lor (interfaţa grafică) şi regulile care guvernează controlul corectitudinii datelor sunt scrise toate într-un singur loc.

Arhitectura pe două nivele (two layer architecture) apare când datele se găsesc într-un alt mediu şi sunt accesate prin intermediul primului nivel. Această arhitectură descrie unele aplicaţiile client-server. Datele sunt stocate într-un server de baze de date primul nivel fiind responsabil pentru interfaţa utilizator şi logica programului.

Arhitectura pe trei nivele (three tier architecture) regulile de validare sunt stocate într-un mediu propriu (de multe ori pe echipamente dedicate), astfel încât aplicaţiile client le pot folosi independent una de alta şi independent de datele existente. În acest fel, aplicaţia client oferă interfaţa, serverul de baze de date oferă datele, iar al doilea nivel se ocupă de validarea datelor[9]. Acest lucru înseamnă că există două comunicaţii client-server: una între aplicaţiile client şi nivelul al doilea, iar cealaltă intre nivelul al doilea şi serverul de baze de date. Aplicaţiile client nu comunică niciodată direct cu serverul de baze de date.

Aplicaţia prezentată în lucrare este dezvoltată pe trei nivele astfel: nivelul întâi reprezentat de interfaţa aplicaţiei web accesată prin intermediul unui browser, nivelul doi compus din agentul snmp, logica de control şi validare a datelor şi nivelul trei reprezentat de serverul de baze de date(MySQL) şi bazele de date de tip round robin.

4. SPECIFICAŢIILE ŞI ARHITECTURA APLICAȚIEI

4.1 Gestionarea şi monitorizarea centrelor de date

Un sistem de gestionare şi monitorizare a unui centru de date are rolul de a uşura munca administratorilor, punând la dispoziţia acestora toate informaţiile necesare pentru intervenţii asupra echipamentelor. Datele din sistem ajută la identificarea rapidă a locaţiei unui echipament, pentru mentenanţa, sş a proprietarului acestuia, pentru notificarea sa. Deasemenea un astfel de sistem este necesar pentru observarea stării reţelei şi echipamentelor şi pentru contorizarea traficului făcut de către server.

4.2 Schema bloc a sistemului

Sistemul este alcătuit din trei componente:

- Utilizatorii
- Aplicaţia
- Centrul de date

Utilizatorii sau administratorii (utilizatori cu drepturi depline) se pot afla în orice locaţie, condiţia pentru a putea utiliza aplicaţia este ca aceştia să dispună de un calculator conectat la internet.

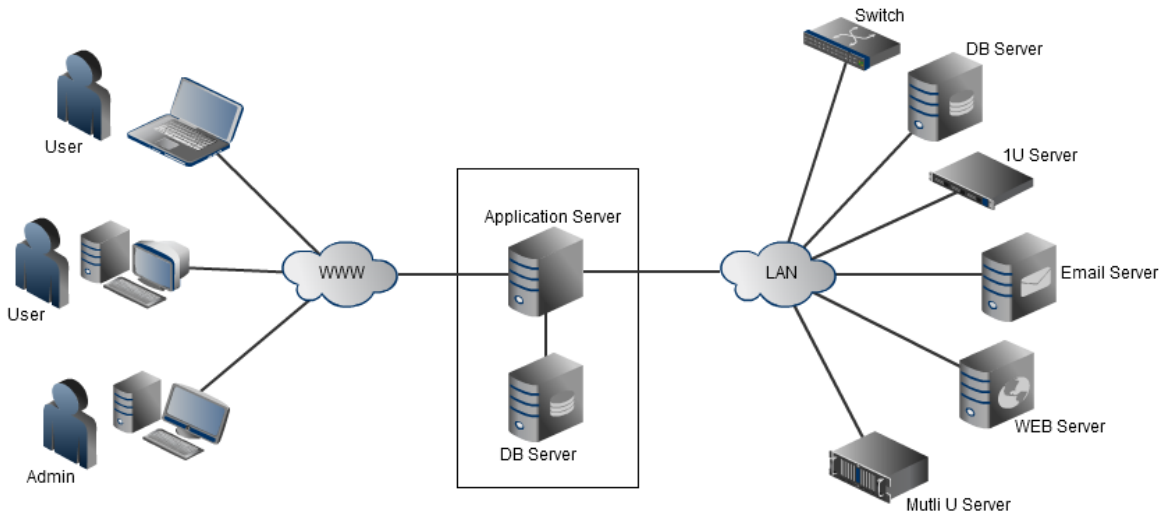

Figura 3.2 Schema bloc a sistemului

Aplicaţia compusă din serverul web împreună cu logica de control şi serverul de baze de date se poate afla în orice locaţie, dar pentru fiabilitate şi din considerente de securitate este indicat să se afle cât mai aproape, chiar în acelaşi LAN cu cea de a treia componentă, centrul de date. Din punct de vedere al sistemului centrul de date reprezintă colecţia de echipamente ce sunt monitorizate.

4.3 Cerințele aplicației

Sistemul a fost proiectat să răspundă urmatoarelor cerinţe:

4.3.1 Cerinţe funcţionale

- **Managementul echipamentelor**

Prin intermediul aplicaţiei administratorul va putea adăuga echipamente noi în sistem. Acesta va putea şterge un echipament sau actualiza datele echipamentelor existente dacă intervin modificări. Utilizatorul va putea la rândul său să modifice datele echipamentelor care ii aparţin.

- **Managementul utilizatorilor**

Tot administratorul va adăuga în sitem companiile sau persoanele care deţin aceste echipamente, împreună cu un nume de utilizator şi o parolă. Acesta va face şi maparea între proprietar şi echipament, un proprietar va putea deţine mai multe echipamente. Datele utilizatorilor pot fi actualizate dacă intervin modificări.

- **Preluarea şi stocarea datelor**

Sistemul preia la intervale regulate date pe care le stochează în baza de date şi arhivele de tip RRA. Intervalul stabilit este de 1 minut, timp suficient de mare pentru ca preluările să nu se suprapună şi suficient de mic pentru ca rezultatele să aibă o rezoluţie cât mai mare indiferent de perioadă.

- **Trimiterea de alerte**

În cazul apariţiei unor evenimente cum ar fi indisponibilitatea unui serviciu monitorizat, defectarea unei componente sau pierderea conectivităţii unui echipament, sistemul trimite alerte pe email specificând problema şi echipamentul pentrua a se putea verifica şi interveni cât mai repede.

- **Crearea şi afişarea graficelor**

Pe baza datelor stocate în RRA-uri se creează grafice cu traficul total făcut de servere în anumite perioade (zilnic, săptămânal, lunar sau anual). Utilizatorul sau administratorul va putea alege perioada pe care să fie generat graficul. Pe aceste grafice se va observa şi lăţimea de bandă folosită în medie în ultimul minut de funcţionare(cu o marjă de eroare de maximum un minut).

- **Vizualizarea datelor**

Utilizatorii sau administratorul vor putea vizualiza în orice moment datele şi graficele. Administratorul va avea acces la toate datele iar utilizatorii doar la datele echipamentelor care le deţin.

4.3.2 Cerinţe non funcţionale

- **Accesibilitate**

Sistemul va fi disponibil atât din punct de vedere al locaţiei utilizatorilor, fiind o aplicaţie web aceştia vor avea nevoie doar de conectivitate la internet şi un browser, cât şi din puctul de vedere al cunoştinţelor necesare, interfaţa fiind intuitivă, prietenoasă şi uşor de utilizat.

- **Scalabilitate**

Administratorul va putea adăuga noi echipamente fără a se face vreo modificare asupra aplicaţiei, acestea, odată introduse în baza de date vor fi monitorizate automat de către sistem. Scalabilitatea este necesară deoarece centrul de date este într-o continuă dezvoltare şi extindere.

- **Disponibilitate**

Sistemul va fi disponibil 24 de ore pe zi, 7 zile pe săptămână pentru ca utilizatorii să poată accesa datele în orice moment. Acest lucru este uşor realizabil, locaţia sa fiind chiar în centrul de date, având la dispoziţie conexiuni redundante la internet, UPS-uri de capacitate mare şi generator de tensiune propriu.

4.4 Scenarii de utilizare

Utilizatorii sunt împărtiţi în două categorii, administratorul, cu drepturi depline, şi utilizatori obişnuiţi(clienţii). Atât administratorul cât şi restul utilizatorilor vor trebui să se autentifice folosind nume de utilizator şi parola pentru a putea accesa aplicaţia. După autentificare, fiecare rol are anumite drepturi, acces la diferite zone ale aplicaţiei şi funcţionalităţi specifice.

Funcţionalitatile corespunzătoare rolurilor sunt:

- Administrator:
  - Adăugarea, editarea şi ştergerea utilizatorilor
  - Adăugarea, editarea şi ştergerea switch-urilor
  - Adăugarea, editarea şi ştergerea companiilor
  - Adăugarea, editarea şi ştergerea echipamentelor
  - Atribuirea echipamentelor către utilizatorii ce le deţin
  - Vizualizarea graficelor
  - Editarea setărilor şi variabilelor aplicaţiei

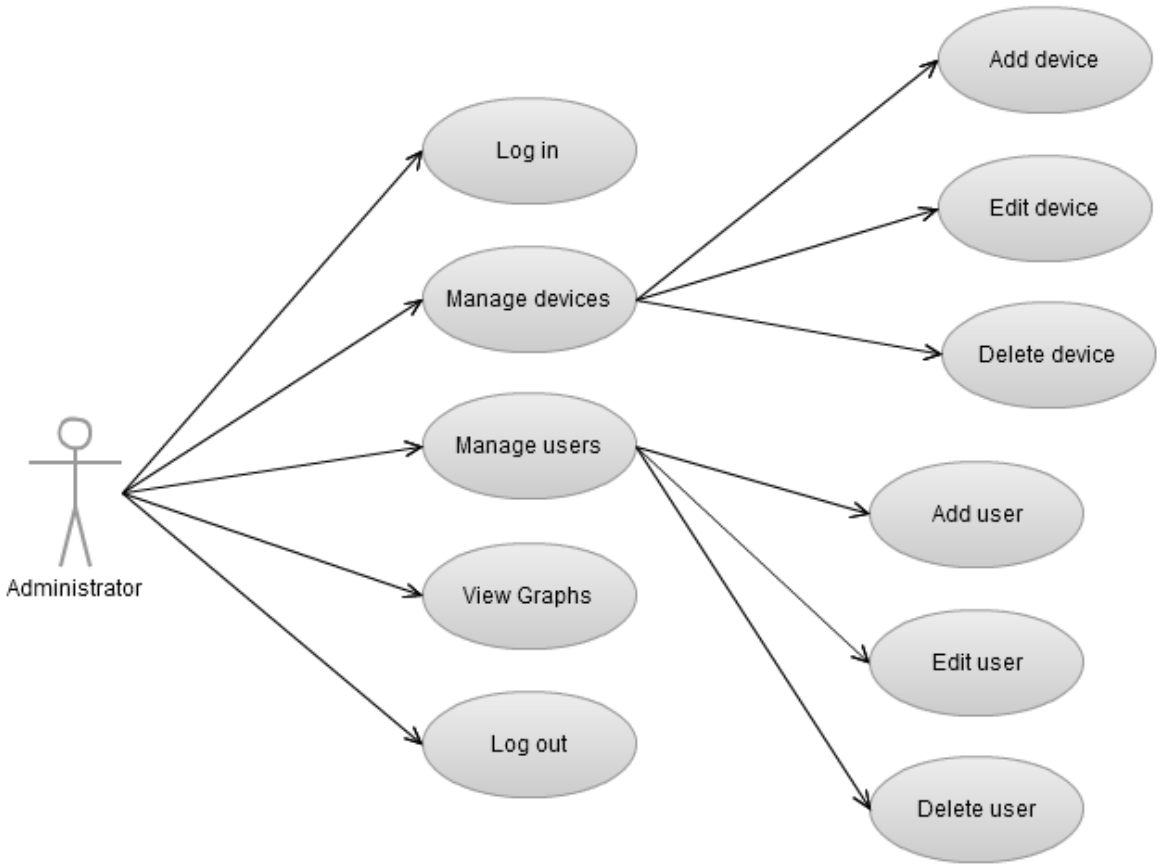

Figura 4.1 Diagrama Use Case pentru Administrator

- Utilizator:
  - Editarea datelor echipamentelor proprii
  - Vizualizarea graficelor echipamentelor proprii

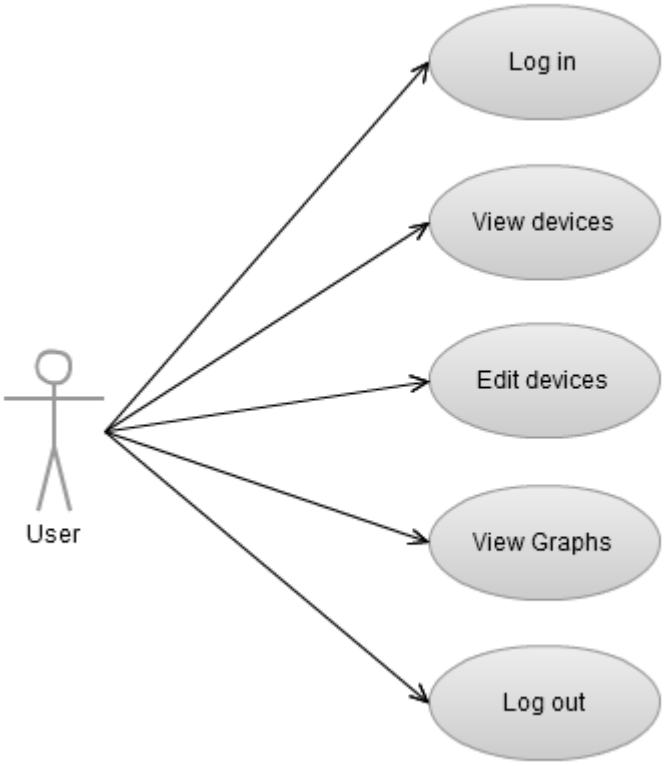

Figura 4.2 Diagrama Use Case pentru Utilizator

În paginile următoare sunt prezentate câteva dintre cele mai importante sau cele mai utilizate cazuri de utilizare.

Caz de utilizare nr 1: Log in

Descriere: Autentificarea utilizatorului şi redirectarea sa în zona aplicaţiei specifică rolului indeplinit

Actori: Administrator, Utilizator

Precondiţie: Aplicaţia trebuie să ruleze, utilizatorul are nevoie de un browser web şi conexiune la internet. În browser trebuie introdusă adresa web a aplicaţiei.

Postcondiţie: Utilizatorul este autentificat şi în funcţie de rolul sau are acces la funcţionalităţile specifice.

Scenariu de succes:

1. La accesarea adresei se afişează pagina de login în care se cere introducerea numelui de utilizator şi a parolei.
2. Sistemul verifică datele furnizate şi redirecţionează utilizatorul spre zona aplicaţiei ce corespunde rolului acestuia.

Extensie:

*. Din motive tehnice(conectivitate la internet, eroare de DNS) pagina nu poate fi afişată: apare pagina default a browserului cu mesajul corespunzător(în funcţie de browser şi eroarea întâlnită)

2. Datele de autentificare sunt incorecte: apare un mesaj de eroare şi se revine la punctul 1.

Caz de utilizare nr 2: Adăugarea unui utilizator în sistem

Descriere: Adăugarea unui utilizator în baza de date a sistemului, acestuia i se vor atribui un nume de utilizator şi o parolă pentru log in.

Actori: Administrator

Precondiţie: Aplicaţia trebuie să ruleze şi utilizatorul trebuie să fie logat în sistem cu drepturi de administrator

Postcondiţie: Un nou utilizator a fost adăugat în sistem, împreună cu datele sale sau ale companiei

Scenariu de succes:

1. Administratorul navighează la pagina de adăugare a unui utilizator.
2. Administratorul introduce datele untilizatorului în câmpurile din pagină şi apasă butonul „Submit”.
3. Noul utilizator este adăugat în sistem.
4. Se deschide pagina iniţială a administratorului.

Extensie:

*. Din motive tehnice(conectivitate la internet, eroare de DNS) pagina nu poate fi afişată: apare pagina default a browserului cu mesajul corespunzator(în funcţie de browser şi eroarea întâlnită)

2a. Administratorul nu a completat toate câmpurile obligatorii de pe pagină: apare un mesaj care cere completarea tuturor câmpurilor obligatorii

2b. Numele de utilizator există deja în sistem: apare un mesaj care cere introducerea unui al nume de utilizator

Caz de utilizare nr 3: Adăugarea unui switch în sistem

Descriere: Adăugarea unui switch în baza de date a sistemului, acesta va putea fi folosit de echipamente noi

Actori: Administrator

Precondiţie: Aplicaţia trebuie să ruleze şi utilizatorul trebuie să fie logat în sistem cu drepturi de administrator

Postcondiţie: În sistem a fost introdus un nou switch

Scenariu de succes:

1. Administratorul navighează la pagina de adăugare a unui switch
2. Administratorul introduce datele switch-ului în câmpurile din pagina şi apasă butonul „Submit”.
3. Noul switch este adăugat în sistem.
4. Se deschide pagina inițiala a administratorului.

Extensie:

*. Din motive tehnice(conectivitate la internet, eroare de DNS) pagina nu poate fi afişată: apare pagina default a browserului cu mesajul corespunzător(în funcţie de browser şi eroarea întâlnită)

2. Administratorul nu a completat toate câmpurile obligatorii de pe pagină: apare un mesaj care cere completarea tuturor câmpurilor obligatorii

Caz de utilizare nr 4: Adăugarea unui echipament în sistem

Descriere: Adăugarea unui echipament în baza de date a sistemului

Actori: Administrator

Precondiţie: Aplicaţia trebuie să ruleze şi utilizatorul trebuie să fie logat în sistem cu drepturi de administrator.
Postcondiţie: În sistem a fost introdus un nou echipament care va fi monitorizat şi pentru care se vor crea grafice.
Scenariu de succes:

1. Administratorul navighează la pagina de adăugare a unui echipament
2. Administratorul introduce datele echipamentului în câmpurile din pagina şi apasă butonul „Submit”.
3. Noul echipament este adăugat în sistemul de monitorizare.
4. Se deschide pagina iniţială a administratorului.

Extensie:

*. Din motive tehnice(conectivitate la internet, eroare de DNS) pagina nu poate fi afişată: apare pagina default a browserului cu mesajul corespunzător(în funcţie de browser şi eroarea întâlnită)

3. Administratorul nu a completat toate câmpurile obligatorii de pe pagină: apare un mesaj care cere completarea tuturor câmpurilor obligatorii

Caz de utilizare nr 5: Căutarea

Descriere: Căutarea după un cuvânt cheie în baza de date a sistemului
Actori: Administrator, Utilizator
Precondiţie: Aplicaţia trebuie să ruleze şi utilizatorul trebuie să fie logat în sistem.
Postcondiţie: Se afişează lista cu rezultatele căutării.
Scenariu de succes:

1. Utilizatorul introduce cuvântul cheie în rubrica destinată şi apasă butonul „Search”.
2. Se afişează un tabel cu rezultatele căutării, cuvântul cheie este comparat cu numele, prenumele şi adresa de email a utilizatorului, hostname-ul, adresa ip şi câmpul cu detalii suplimentare ale echipamentului

Extensie:

*. Din motive tehnice(conectivitate la internet, eroare de DNS) pagina nu poate fi afişată: apare pagina default a browserului cu mesajul corespunzător(în funcţie de browser şi eroarea întâlnită)

1. Câmpul destinat cuvantului cheie este gol: apare un mesaj care informează utilizatorul despre acest lucru
2. În urma căutării nu s-a găsit nici un rezultat valid: apare un mesaj care informează utilizatorul despre acest lucru

Caz de utilizare nr 6: Vizualizarea graficelor

Descriere: Vizualizarea graficelor de trafic ale unui echipament din baza de date a sistemului
Actori: Administrator, Utilizator
Precondiţie: Aplicaţia trebuie să ruleze şi utilizatorul trebuie să fie logat în sistem
Postcondiţie: Utilizatorul vizuaizeaza graficele create de aplicaţie
Scenariu de succes:

1. Utilizatorul caută echipamentul al cărui grafic doreşte să îl vadă şi apasă link-ul „View Graph”
2. Se deschide pagina ce conţine graficele echipamentului respectiv

Extensie:

*. Din motive tehnice(conectivitate la internet, eroare de DNS) pagina nu

poate fi afişată: apare pagina default a browserului cu mesajul corespunzător(în funcţie de browser şi eroarea intâlnită)

Caz de utilizare nr 7: Editarea datelor unui utilizator

Descriere: Modificarea numelui, companiei sau a altor detalii ale unui utilizator din baza de date a sistemului

Actori: Administrator

Precondiţie: Aplicaţia trebuie să ruleze şi utilizatorul trebuie să fie logat în sistem cu drepturi de administrator.

Postcondiţie: Datele utilizatorului au fost modificate.

Scenariu de succes:

1. Administratorul caută utilizatorul ale cărui date doreşte să le modifice.
2. Administratorul modifică datele untilizatorului în câmpurile din pagină şi apasă butonul „Update”.
3. Datele utilizatorului sunt modificate.
4. Se deschide pagina iniţială a administratorului.

Extensie:

*. Din motive tehnice(conectivitate la internet, eroare de DNS) pagina nu poate fi afişată: apare pagina default a browserului cu mesajul corespunzător(în funcţie de browser şi eroarea întâlnită)

2a. Administratorul nu a completat toate câmpurile obligatorii de pe pagină: apare un mesaj care cere completarea tuturor câmpurilor obligatorii

2b. Numele de utilizator există deja în sistem: apare un mesaj care cere introducerea unui al nume de utilizator

Caz de utilizare nr 8: Editarea datelor unui echipament

Descriere: Modificarea datelor unui echipament din baza de date a sistemului

Actori: Administrator

Precondiţie: Aplicaţia trebuie să ruleze şi utilizatorul trebuie să fie logat în sistem cu drepturi de administrator

Postcondiţie: Datele echipamentului au fost modificate

Scenariu de succes:

1. Administratorul caută echipamentul ale cărui date doreşte să le modifice.
2. Administratorul modifică datele echipamentului în câmpurile din pagina şi apasă butonul „Update”.
3. Datele utilizatorului sunt modificate.
4. Se deschide pagina iniţială a administratorului.

Extensie:

*. Din motive tehnice(conectivitate la internet, eroare de DNS) pagina nu poate fi afişată: apare pagina default a browserului cu mesajul corespunzător(în funcţie de browser şi eroarea întâlnită)

2. Administratorul nu a completat toate câmpurile obligatorii de pe pagină: apare un mesaj care cere completarea tuturor câmpurilor obligatorii

Caz de utilizare nr 9: Reamintirea parolei

Descriere: Reamintirea parolei în cazul în care aceasta a fost uitată

Actori: Utilizator

Precondiţie: Aplicaţia trebuie să ruleze iar utilizatorul nu se poate loga în sistem

Postcondiţie: Utilizatorul primeşte un email ce conţine numele de utilizator şi parola pentru a se putea loga în sistem.

Scenariu de succes:

1. Utilizatorul urmează link-ul „Forgot password” de pe pagina de login a aplicaţiei
2. Utilizatorul introduce numele de utilizator şi adresa de email în câmpurile de pe pagina de reamintire a parolei şi apasă butonul „Submit”
3. Email-ul cu datele de logare este trimis la adresa utilizatorului
4. Se deschide pagina de login

Extensie:

*. Din motive tehnice(conectivitate la internet, eroare de DNS) pagina nu poate fi afişată: apare pagina default a browserului cu mesajul corespunzător(în funcţie de browser şi eroarea întâlnită)

2. Adresa de email sau numele de utilizator nu există în sistem: se afişează un mesaj de eroare şi se revine la punctul 2.

Caz de utilizare nr 10: Ştergerea unui echipament

Descriere: Ştergerea unui echipament din baza de date a sistemului

Actori: Administrator

Precondiţie: Aplicaţia trebuie să ruleze şi utilizatorul trebuie să fie logat în sistem cu drepturi de administrator

Postcondiţie: Echipamentul a fost şters din baza de date acesta nu mai este monitorizat şi nu se mai creează grafice

Scenariu de succes:

1. Administratorul caută echipamentul care doreşte să fie sters.
2. Administratorul apasă butonul „Remove device” din dreptul echipamentului care trebuie şters.
3. Echipamentul împreună cu graficele acestuia sunt şterse din sistem
4. Se deschide pagina iniţiala a administratorului.

Extensie:

*. Din motive tehnice(conectivitate la internet, eroare de DNS) pagina nu poate fi afişată: apare pagina default a browserului cu mesajul corespunzător(în funcţie de browser şi eroarea întâlnită)

Caz de utilizare nr 11: Ştergerea unui utilizator

Descriere: Ştergerea unui utilizator din baza de date a sistemului

Actori: Administrator

Precondiţie: Aplicaţia trebuie să ruleze şi utilizatorul trebuie să fie logat în sistem cu drepturi de administrator

Postcondiţie: Utilizatorul a fost şters din baza de date împreună cu toate echipamentele ce le deţine

Scenariu de succes:

1. Administratorul caută utilizatorul care doreşte să fie şters.
2. Administratorul apasă butonul „Remove user” din dreptul utilizatorului care trebuie şters.
3. Utilizatorul împreună cu tote echipamentele pe care acesta le deţine sunt şterse din sistem
4. Se deschide pagina iniţiala a administratorului.

Extensie:

*. Din motive tehnice(conectivitate la internet, eroare de DNS) pagina nu poate fi afişată: apare pagina default a browserului cu mesajul corespunzător(în funcţie de browser şi eroarea întâlnită)

4.5 Arhitectura aplicației

Arhitectura aplicației este una de tipul three-tier: nivelul superior fiind reprezentat de interfaţa cu utilizatorul, nivelul de mijloc de către managerul SNMP şi celelalte componente de logică şi control, iar nivelul cel mai de jos fiind alcătuit din bazele de date MySQL şi RRDtool.

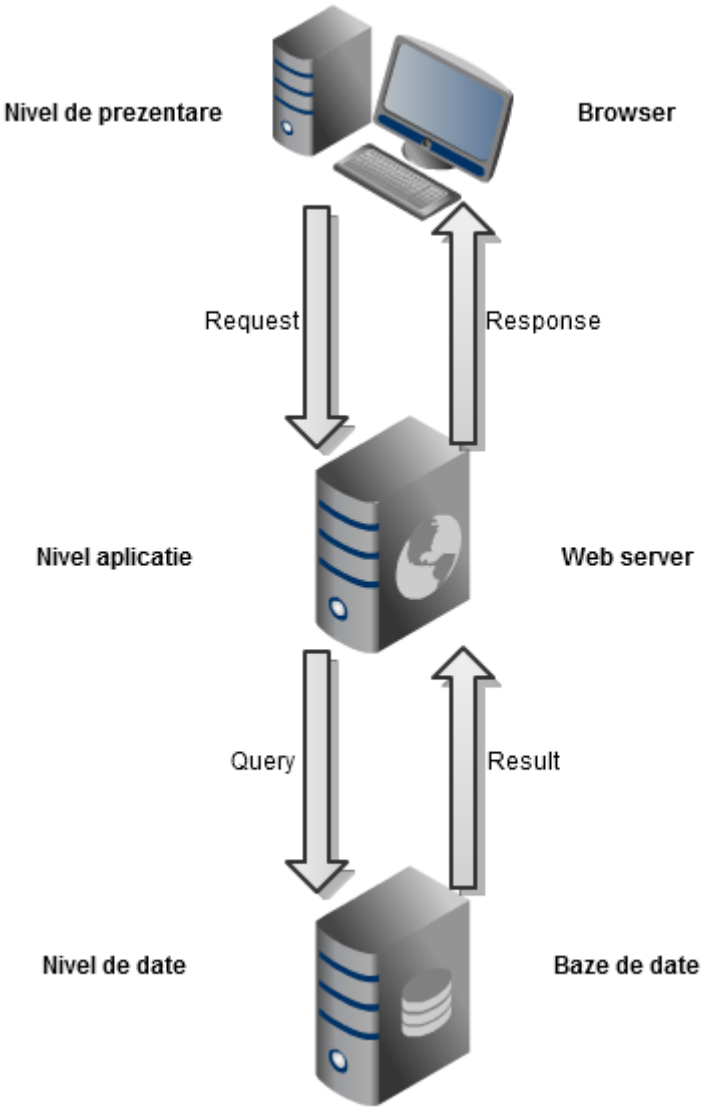

Figura 4.3 Modelul arhitectural Three-tier

Printre avantajele arhitecturii pe trei nivele se numără[10]:

- Flexibilitate – se pot modifica sau înlocui oricare dintre cele trei nivele fără a le afecta pe celelalte.

- “Load Balancing” – separarea logicii şi controlului de bazele de date ajută la distribuirea încărcării serverelor, o încărcare mai mare a serverului web, datorată unui număr mare de requesturi nu va îngreuna întreaga aplicaţie.

- Securitate – nivelul de prezentare neavând acces direct la nivelul de date, toate operaţiunile efectuate asupra datelor sunt sigure şi verificate, deasemenea se pot lua măsuri suplimentare de siguranţă pentru date în cazul în care nivelul de mijloc este compromis.

- Scalabilitate – nivelele 2 şi 3 pot fi găzduite pe acelaşi server, pe servere diferite sau pe sisteme tip cluster, în funcţie de necesităţi

Am optat pentru utilizarea acestei arhitecturi în cadrul aplicaţiei pentru că dezvoltarea ulterioară a centrului de date şi eventualele modificări minimale să nu necesite modificarea întregii aplicaţii. Deasemenea arhitectura folosită simplifică şi dezvoltarea ulterioară a aplicaţiei, adăugarea de noi funcţionalităţi putându-se face pe etape, nivel cu nivel, începând de la nivleul bazelor de date, continuând cu nivelul de logică şi control, urmând ca în final să se creeze şi partea de interfaţă pentru a accesa noile feature-uri.