



GRUIA LAVINIA-ANDREEA

---

# Proiectarea si dezvoltarea unui sistem de e-learning pentru invatarea limbajului C++

---

*Coordonator:* PROF. DR. VIOREL NEGRU

ASIST. DRD. OVIDIU ARITONI

*Available at:* <http://code.google.com/p/elearning-system>

Universitatea de Vest din Timișoara  
2011

## Cuprins

<b>1</b>	<b>Studiul sistemelor de e-learning</b>	<b>4</b>
1.1	State-of-art . . . . .	4
1.2	Studiu platforme software similare . . . . .	9
<b>2</b>	<b>Elemente</b>	<b>11</b>
2.1	Student Record . . . . .	11
2.2	Curba de învățare . . . . .	14
2.3	Arhitecturi ale sistemelor de E-Learning . . . . .	22
2.3.1	Arhitectură cu agent dublu . . . . .	24
2.3.2	Arhitectură Web Service bazată pe mesaje . . . . .	28
2.3.3	O arhitectură pentru sisteme e-learning cu laboratoare on-line . . . . .	38
2.3.4	Comunicarea utilizând SOAP . . . . .	40
2.3.5	Serviciul de mesaje bazat pe XML . . . . .	40
2.3.6	<i>Dispozitiv pentru Web</i> . . . . .	44
2.3.7	Arhitectura unui sistem de e-learning utilizat în insti- tuțiile de învățământ superior din România . . . . .	44
<b>3</b>	<b>Proiectarea și dezvoltarea sistemului de E-learning</b>	<b>51</b>
3.1	Raport de modelare . . . . .	51
3.1.1	Modelarea funcțională și dinamică . . . . .	51
3.1.2	Diagrama de context static . . . . .	51
3.1.3	Diagrama cazurilor de utilizare pentru studenți . . . . .	51
3.1.4	Diagrama cazurilor de utilizare pentru profesor . . . . .	57
3.1.5	Diagrama cazurilor de utilizare pentru administrator . . . . .	66
3.1.6	Diagrama de clasă . . . . .	68
3.1.7	Structura în package-uri a aplicației . . . . .	68
3.1.8	Modelul utilizat pentru baza de date . . . . .	75
3.1.9	Șabloane de proiectare utilizate . . . . .	75
3.1.10	Modelul arhitectural al aplicației . . . . .	75
3.1.11	Designul interfeței utilizator . . . . .	77
3.1.12	Designul bazei de date . . . . .	77

## Listă de figuri

1	Arhitectura generală cu agent dublu . . . . .	25
2	Arhitectură cu mai mult de doi agenți . . . . .	26
3	Implementarea arhitecturii cu mai mulți agenți . . . . .	27
4	Browser SVG derivat din MVC bazat pe mesaj . . . . .	31
5	Schema Public/Subscrie bazat pe mesaj condus de eveniment . . . . .	32
6	Portul de intrare pentru SVG colaborativ . . . . .	34
7	Colaborarea monolitică . . . . .	35
8	Web Service colaborativ de tip SIMD . . . . .	36
9	. . . . .	37
10	Arhitectura pentru sistemul e-learning cu laboratoare on-line . . . . .	39
11	Diagrama mesajului SOAP . . . . .	40
12	Mediul de execuție App . . . . .	42
13	Arhitectura standard a sistemului GESCO . . . . .	46
14	Servicii și aplicații Web în arhitectura GESCO . . . . .	49
15	Diagrama de context static . . . . .	52
16	Diagrama cazurilor de utilizare student . . . . .	53
17	Diagrama de secvență pentru logare . . . . .	55
18	Diagrama de activitate pentru logare . . . . .	56
19	Diagrama de secvență pentru alegerea lecției . . . . .	58
20	Diagrama de activitate pentru lecției . . . . .	59
21	Diagrama cazurilor de utilizare . . . . .	60
22	Diagrama de secvență pentru logare . . . . .	62
23	Diagrama de activitate pentru logare . . . . .	63
24	Diagrama de secvență pentru introducerea testului . . . . .	64
25	Diagrama de activitate pentru introducerea testului . . . . .	65
26	Diagrama cazurilor de utilizare . . . . .	67
27	Diagrama de secvență pentru logare . . . . .	69
28	Diagrama de activitate pentru logare . . . . .	70
29	Diagrama de secvență pentru introducerea userilor . . . . .	71
30	Diagrama de activitate pentru introducerea userilor . . . . .	72
31	Diagrama de clasă . . . . .	73
32	Designul bazei de date . . . . .	78

## Listă de tabele

1	Compararea diferitelor Sisteme de E-learning . . . . .	10
2	Compararea Sistemelor pentru realizarea sistemelor de E-learning	10
3	Tabela : Student . . . . .	77
4	Tabela : Profesor . . . . .	77
5	Tabela : Administrator . . . . .	77
6	Tabela Test . . . . .	79
7	Tabela Chestionare . . . . .	79
8	Situație Școlară . . . . .	79
9	Tabela Verificare . . . . .	79

# 1 Studiul sistemelor de e-learning

## 1.1 State-of-art

Definiții ale educației online și ale e-learning-ului Educația online: Sunt mulți termeni pentru educația online. Unii sunt: pentru educația virtuală, educația Internet-based, educația web-based și educația prin meditare pe calculator. Definiția educației online este dezvoltată din definiția lui Keegan(1996): Educația la distanță este o formă de educație caracterizată prin:

- separarea cvasi-permanentă a profesorului și studentului pe toată durata procesului de învățare (aceasta îl diferențiază de educația convențională față-în-față);
- influența educațională organizată, atât în planificarea și pregătirea materialelor de învățare cât și în furnizarea de servicii pentru sprijinire a studenților (aceasta o diferențiază de studiu privat și programe de învățat singur);
- utilizarea mijloacelor tehnice - printare, audio, video sau calculator - să unească profesor, cursant și transportarea conținutului cursului;
- furnizarea de comunicare în ambele sensuri, astfel încât studentul poate beneficia sau chiar iniția un dialog (aceasta se diferențiază de alte utilizări ale tehnologiei în educație); și
- absența cvasi-permanent al grupului de învățat pe tot parcursul duratei procesului de învățare, astfel încât oamenii sunt de obicei învățați ca indivizi, mai degrabă decât în grupuri, cu posibilitatea de întâlniri ocazionale, fie față-în-față sau prin mijloace electronice, atât pentru scopuri didactice cât și sociale."

Dacă acceptăm că educația on-line reprezintă un subset de învățământ la distanță putem defini educația online prin acceptarea definiției Keegan și schimbarea punctelor 3-4 la:

- utilizarea computerelor și a rețelelor de calculatoare să se unească profesori, cursanții și conținutul cursului;
- furnizarea de comunicare în ambele sensuri prin intermediul rețelelor informatice, astfel încât studentul poate beneficia de inițierea unui dialog (aceasta se diferențiază de alte utilizări ale tehnologiei în educație);

Cei mai mulți susținătorii ai educației on-line ar exclude punctul cinci al lui Keegan, ca de învățare colaborativă, în cazul în care elevii pot comunica pe toată durata procesului de învățare este văzut ca unul dintre cele mai importante avantaje de învățare online față de "generațiile" precedente de învățământ la distanță (McConnell 2000). Pe de altă parte, există motive întemeiate să sublinieze faptul că majoritatea studenților adulți necesită să organizeze studiile lor în conformitate cu cerințele de muncă, viața socială și responsabilitățile familiale. Aceste nevoi trebuie să fie echilibrate față de un ideal didactic pentru învățare co-operativă. Astfel, flexibilitatea instituției de adaptare a cerințelor, astfel încât elevii pot organiza învățarea independentă de un grup de studiu este important pentru mulți elevi, on-line (Rekkedal 1999). Aceasta nu exclude, toate metodele de învățare exploatând avantajele de a fi parte a unei comunități sau grup de învățare.

"Educația la distanță" și învățarea la distanță", astfel cum sunt definite de Keegan (1996) sunt concepte bine stabilite. "Studentul la distanță" este o persoană care, pentru anumite motive, nu vor sau nu pot participa la programe educaționale care necesită prezența în anumite momente sau locuri. Termeni precum "e-learning" și, de asemenea, "m-learning" au intrat pe scenă. Pentru noi, învățarea este o activitate sau un proces care demonstrează o schimbare a percepției unei persoane, atitudini sau abilități cognitive sau fizice. Termeni e-learning și d-learning merită să fie analizați. De exemplu, termenul, e-learning, este adesea folosit pentru a convinge utilizatorii că unele lucruri supranaturale se întâmplă cu creierul tău când te afli în fața unui ecran de computer. Cu toate acestea, în lumea reală acest miracol este foarte puțin probabil să se întâmple, așa cum este învățarea, o muncă grea. Cele mai multe exemple de programe de e-learning par a fi extrem de costisitoare pentru a fi dezvoltate și mai acoperă adesea un nivel scăzut de cunoștințe și fapte bazate pe o viziune simplistă a ceea ce este învățarea. (A se vedea de exemplu, Dichanz 2001 "E-learning, o analiză lingvistică, psihologică și pedagogică a unui termen înșelător").

Cu toate acestea, termenul pare să fi devenit parte din terminologia acceptată, este imperativ pentru cercetători în domeniul educației și pentru a defini și a atribui în sensul că este în conformitate cu opiniile noastre cu privire la predare și învățarea. Văzut dintr-o perspectivă universitară, Dichanz, care este profesor de educație și German FernUniversität încheie analiza critică a termenului, e-learning, cu următoarea definiție:

E-learning este colectarea și predarea - a unui pachet de informații - în continuarea educației care este disponibil la orice moment și în orice loc și sunt livrate la cursanții electronic. Acestea conțin unități de informații, baterii de auto-testare și teste, care permit o auto-evaluare rapidă pentru o plasarea rapidă. E-learning ofera un nivel mai scăzut al scopului. Obiective

de ordin superior cum ar fi înțelegerea, raționamentul și (morale) judecarea sunt mai greu de realizat. Ele necesită un discurs interactiv și individualizat și cu greu pot fi planificate." (Dichantz 2001)

Chiar dacă nu suntem de acord în totalitate cu Dichantz că obiectivele de nivel superior de învățare nu pot fi planificate, suntem de acord că aceste obiective sunt mult mai dificil de planificat, și că cele mai multe așa-numite programe de e-learning nu demonstrează atenția asupra obiectivelor de nivel superior de învățare.

Pentru scopurile noastre aici e-learning este definit ca învățare interactivă în care conținutul de învățare este disponibil online și oferă feedback-ul automat la activități de învățare al elevului. Comunicare online cu oameni reali poate sau nu poate fi inclusă, dar focalizarea de e-learning este de obicei mai mult pe conținutul de învățare decât pe comunicarea între cursanți și tutori.

Din păcate, termenul de e-learning este adesea folosit ca un termen mai generic și ca un sinonim pentru educație on-line. Kaplan-Leiserson a dezvoltat un glosar online e-learning, care prevede această definiție:

E-learning acoperă o gamă largă de aplicații și procese, cum ar fi învățarea bazată pe Web, de învățare pe calculator, săli de clasă virtuale, colaborare digitală. Acesta include livrarea de conținut prin Internet, intranet / extranet (LAN / WAN), audio-și casete video, difuzat prin satelit, TV interactiv, și CD-ROM.

În glosarul de [elearningeuropa.info](http://elearningeuropa.info), e-Learning este definit ca: utilizarea de noi tehnologii multimedia și Internet pentru a îmbunătăți calitatea de învățare prin facilitarea accesului la resurse și servicii, precum schimburile și colaborarea la distanță.

Termenul de e-learning este, după cum se poate vedea, nu foarte precis, și ar trebui să se sublinieze învățarea este doar un element de educație. Deci, termenul de educație on-line ar trebui să acopere o gamă mult mai largă de servicii decât termenul de e-learning. Se poate pretinde, de asemenea, că companiile de e-learning se concentrează adesea pe conținutul cursului, în timp ce instituțiile de învățământ on-line acopere întreaga gamă de servicii educaționale de sprijinire a studenților care de cele mai multe ori este dat un accent major.

În ultimii 10 ani, un mare număr de instituții din întreaga lume sunt în curs de dezvoltare și oferă educație on-line la distanță. Instituții cu un fundal istoric pentru educație tradițională on-campus par adesea pentru a transfera predarea învățarea filozofiei, teoriei, conceptelor și metaformelor din acest mediu. Keegan (2000) susține:

Încă educația bazată pe web este cel mai bine privit ca un subset de învățământ la distanță și că aptitudinile, literatura și decizii practice de man-

agement care au fost dezvoltate în formă de dispoziții educațională cunoscută ca învățământ la distanță", că va fi aplicabil de mutatis mutandis educația web-based. Rezultă de asemenea că literatura de specialitate de cercetare educațională cunoscută ca învățământul la distanță, este de valoare pentru cei care se îmbarcă pe instruirea web. "

### **Conceptualizarea procesul de învățare**

Nu vine ca o surpriză faptul că educația devine din ce în ce mai importantă în societatea bazată pe cunoaștere, care rezultă în idei noi în zona de învățare și de predare (cf., de exemplu, Bleimann 2004). În plus, în general evoluțiile în învățământul superior, care rezultă din cerințele societății, precum și o nevoie crescută ca studenții să devină autonomi, au crescut nevoia de cadre universitare care să înțeleagă procesul de învățare (Webster și Sudweeks 2006)

Țin să spun, dar mai multe abordări pentru e-Learning au fost sugerate de-a lungul anilor, dar cei mai mulți oameni au tendința să fie de acord că învățarea este un proces prin care elevii își ating obiectivele de învățare prin realizarea unui număr de activități de învățare și participarea la interacțiuni pentru a reflecta înțelegerea lor (cf., de exemplu, Sun et al. 2004). Astfel, de învățare pare să rezulte dintr-o schimbare în percepția studenților a realității legate de problema zona de sub studiu după cum s-a discutat, de exemplu, (Rekkedal și Dye 2007). Învățarea este apoi în cauza cu modul în care oamenii dobândesc noi cunoștințe, competențe și modul în care cunoștințele și abilitățile existente sunt modificate pentru a rezolva probleme (Shuell 1986). Aceasta nu implică un fel de transfer de cunoștințe obscure de la o sursă la alta, ci mai degrabă este format din rolul activ jucat de elev pentru a procesa informațiile și pentru a le utiliza (Barnard 2006).

În acest sens, Shuell și Lee (1976) definește trei criterii de învățare care sunt:

- o schimbare într-un comportament individual sau capacitatea de a face ceva;
- o prevedere că această modificare trebuie să rezulte dintr-un fel de practică sau de experiență, și
- o prevedere că schimbarea este una de durată;

Ultimele două exclude schimbări de comportament, cum ar fi maturizarea și schimbarea temporară din cauza drogurilor, etc.

În plus, a devenit tot mai evident că valoarea cunoștințelor pe care elevii le posedă are un impact substanțial asupra proceselor lor de învățare (Chi et al 1982.) și stilurilor de învățare. Elevii învață în diferite moduri și modul în care sunt prezentate informațiile care le afectează capacitatea lor de a învăța.



Prin urmare, stilurile de învățare trebuie să fie diferențiate. În acest sens, Sun et al. (2003) identifică trei stiluri de învățare pentru a sprijini studenții în procesul de învățare:

- Visual elevii învață cel mai bine prin intermediul vizualului, cum ar fi imagini, demonstrații, expresii faciale, și ale corpului instructorului pentru a înțelege pe deplin conținutul lecției;
- Auditiv elevii învață cel mai bine prin ascultarea lucrurilor prin prelegeri verbale și, discuții, vorbind lucruri și prin ascultare la ceea ce alții au de spus;
- Tactile / kinestetică elevii învață cel mai bine prin experimentare, reflectare, interacționare, și prin intermediu de a face lucruri. Acești elevii preferă să exploreze în mod activ lumea fizică din jurul lor și beneficiază de manipularea obiectelor reale și / sau care acționează asupra lor într-un mediu simulat.

Cu toate acestea, elevii trebuie să utilizeze stiluri diferite de învățare interschimbabile în timpul procesului de învățare pentru ca acestea să fi o experiență de învățare eficientă.

### **Teorii ale învățării**

Teoriile de învățare sunt implicate în procesul efectiv de învățare, nu cu valoarea a ceea ce se învață. Ideologia centrală teoriei de învățare este faptul că învățarea are loc în interiorul unei persoane (Siemens 2004). Există trei perspective principale în procesul teorie de învățare care oferă o înțelegere inerentă a procesului de învățare prin care cursanții pot construi cunoștințe într-un anumit mediu.

Constructivismul (Honebein et al 1993; Fosnot 1996) învață ghiduri și instructori în realizarea, gestionarea și încurajarea activităților de învățare personalizate prin învățare colaborativă. Acest tip de învățare facilitează gândirea critică și rezolvarea de probleme. Elevul este foarte activ construiește idei noi folosind cunoștințele anterioare și experiența atinsă. În timpul procesului de învățare, profesorul ia rolul unui facilitator concentrându-se pe a face corecturi, promovarea înțelegeri noi, și crearea socializării. Cursanți își asumă responsabilitatea de învățare prin participarea activă la activități de învățare puse la centrul procesului de învățare. Această teorie de învățare a ghidat mulți educatorii la încurajarea elevilor (Gregson 1997). În acest scop, Koohang și Harman (2005) afirmă că, într-un mediu constructivist, situațiile de învățare reprezintă complexitatea normală a lumii reale. Ca un rezultat, perspective multiple și reprezentări care promovează învățarea cooperativă și colaborativă sunt încurajate.

Teoriile cognitive de învățare descriu implicarea realizării de structuri cognitive, prin care ființele umane procesează și stochează informații (Brophy și Good 1990). Ele demonstrează cum un student percepe, procesează, interpretează, înmagazinează, precum și preluarea informațiilor sunt, în principal cu schimbări în înțelegerea unui student care rezultă din învățare. Studentul este implicat în procesul de învățare, astfel încât profesorii trebuie să prezinte informații organizate într-un mod la care studentul poate face referire. Shuell (1986) subliniază că o abordare cognitivă subliniază învățare ca un activ constructiv, orientat spre scop și proces care este dependentă de activități mentale ale elevului.

Comportamentul definește învățarea ca o schimbare în comportament observabilă. În consecință, își asumă comportamentele care sunt observabile și pot fi corelate cu alte evenimente observabile. Învățarea se concentrează pe stimuli, de exemplu, ceea ce are un impact, și un răspuns, de exemplu, modul în care comportamentul unui organism se modifică. Belkin și Gray (1977) subliniază importanța condiționată în teoria comportamentului. Acesta indică faptul că învățarea apare ca un rezultat pozitiv de întărire care să conducă la abandonarea tiparelor vechi, ca urmare a armării negative. Activități de învățare desfășurate în timpul de predare sunt aranjate neprevăzut de întărirea în care elevii își construiesc cunoștințele.

Astfel, teoriile învățării explică procesul de învățare prin care elevii sunt capabili de a dobândi cunoștințe, dar nu există nici o teorie unică de învățare care poate explica pe deplin toate tipurile de învățare. În consecință, mai multe teorii coexistă și se completează reciproc în timpul unui proces de învățare. Ar trebui să fie păstrate în minte, deși, realizarea de concepte de învățare variază de la un elev la altul și metodele de învățare dictate de nivelul de cunoștințe care trebuie atins.

## 1.2 Studiu platforme software similare

Această etapă tratează o problemă destul de importantă aceasta este găsirea pe piață a altor platforme similare cu proiectul de Sisteme de E-learning. Având în vedere că E-learning înseamnă: colectarea și predarea - a unui pachet de informații - în continuarea educației care este disponibil la orice moment și în orice loc și sunt livrate la cursanții electronic. Acestea conțin unități de informații, baterii de auto-testare și teste, care permit o auto-evaluare rapidă pentru o plasare rapidă. S-au găsit următoarele platforme de sisteme bazate pe tehnologie web: TuringScraft, Codepad și System C++ Web based compiler acestea sunt compilatoare web care ajută cursantul să învețe un limbaj de programare. În tabelul următor aveți o comparație a celor trei produse.

Criterii de comparare	TuringScraft
Descriere	Eu un sistem bazat pe tehnologie web pentru programare c++
Site	www.turingscraft.com
Portabilitate Linux/vista/7	DA
Salvarea codului scris	NU
Mesaje de ajutor	DA
Lectii pentru ajutor	DA
Free source	NU

Tabela 1: Compararea diferitelor Sisteme de E-learning

Criterii de comparare	
Site	
Portabilitate Linux/vista/7	
Ușor de folosit	
Conținut multimedia bogat	
Conținutul de e-learning creat poate fi exportat	
Conținutul de e-learning creat este portabil pe alte sisteme de operare	
Limitare în crearea conținutului de e-learning	Nu deoarece are in
Necesită cunoașterea unui limbaj de programare pentru a putea fii utilizat	Nu necesit
Free source	

Tabela 2: Compararea Sistemelor pentru realizarea sistemelor de E-learning

În continuare vom prezenta câteva platforme software care realizează produse de E-learning și sunt larg folosite de către profesori universitari sau de către firme pentru angajați acestora la diferite cursuri pentru o mai bună înțelegere a acestora. Platformele software sunt următoarele: Adobe eLearning Suite care este o suită Adobe care conține printre alte programe și softul Adobe Captive CS5 un utilitar puternic pentru realizarea conținutului de e-learning ușor de utilizat, programul Cognitive Tutor Authoring Tools (CTAT) este un program destul de puternic pentru realizarea conținutului de e-learning folosit de către profesori în diferite universități din lume, programul CourseLab care este un program ușor de utilizat și are încorporat opțiuni de realizare a tutorialelor destul de complexe având în vedere că este un program gratis. În tabelul următor aveți o comparație a celor trei produse.

## 2 Elemente

### 2.1 Student Record

În sensul Drepturilor Educationale de Familie și a Actului de Confidențialitate (FERPA), un registru de student este definit ca o înregistrare de educație, care este compusă din acele înregistrări care sunt direct legate de un student și întreținute de către universitate sau un agent care acționează pentru universitate. Există un număr de înregistrări care sunt luate în considerare la determinarea înregistrării educaționale. Acestea pot fi găsite în actul Drepturilor de Familie și Actul de Confidențialitate din 1974.

#### **Ce este un sistem de înregistrare?**

Un sistem de înregistrare a studenților (SIS) este un program pentru calculator realizat pentru instituții de învățământ pentru a gestiona datele studenților. Sistemele de înregistrare oferă posibilitatea de a introduce note de la teste și alte scoruri de evaluare într-un catalog electronic, precum și urmărirea prezenței la cursuri, și gestionând multe alte date legate de un student/elev într-o facultate sau o școală.

SIS este echivalent cu un sistem de Enterprise Resource Planning sau ERP pentru un client corporate. Ca atare, multe dintre probleme cu ERP Metodologia de Selecție de sistem, implementarea și operarea unui sistem ERP se aplică în școli și sistemelor lor SIS.

#### **Funcțiile unui sistem de înregistrare**

Aceste sisteme variază în dimensiune, domeniul de aplicare și de capacitate, de la pachetele care sunt puse în aplicare în cadrul organizațiilor relativ mici pentru a acoperi înregistrările studenților, la soluții de nivel de întreprindere, care urmăresc să acopere cele mai multe aspecte de funcționare a organizațiilor mari de multi-campusuri cu responsabilitate locală semnificativă. Multe sisteme pot fi scalate la diferite niveluri de funcționalitate prin achiziționarea add-onurilor "module" și poate fi configurat de obicei de către instituțiile lor de origine pentru a satisface nevoile locale.

Până de curând, funcțiile de bază ale unui sistem de înregistrări a datelor de studenți sunt acelea care sprijină menținerea de informații cu caracter personal și de studiu cu privire la:

- Anchete de manipulare de la viitorii studenți
- Manipularea procesului de admitere
- Înregistrarea studenților noi și stocarea obținuturilor
- Crearea automată a programului de clasă și al profesorului

- Manipularea înregistrărilor de examinare, notificări, note, și a dezvoltării academice
- Menținerea de înregistrări ale absețelor și prezențelor
- Comunicații înregistrate cu elevii
- Menținerea înregistrării disciplinei
- Asigurarea rapoartelor statistice
- Întreținerea detaliilor de pensiune
- Comunicarea informațiilor despre studenți la părinți prin intermediul unui portal web
- Educație Specială/Plan Educațional Individual
- Servicii de resurse umane
- Servicii de contabilitate și bugetare
- Dosarele medicale ale studenților/elevilor

### **Sisteme integrate/Hosted Service**

În ultimi ani, mai multe forțe au condus la o evoluție a programelor pentru registrul studenților, ca rezultat, multe instituții au fost nevoite să le înlocuiască sistemele vechi cu cele noi.

Aceste forțe sunt:

- Cererea de acces 24/7 prin intermediul internetului la informații de către studenți, instructori, și părinți.
- Creșterea cererii în cantitate și frecvență de raportare a datelor pentru contabilitate și alte scopuri
- Importanță de integrare a sistemelor de informații pentru studenți cu alte instrumente, în special cu privire la instruire, cursuri și de învățare
- SIF

### **Sistemul registrului de studenți migranți**

#### **Ce este și cine îl folosește?**

Când Programul Federal de Educație al Studenților Migranți a fost creat în 1966 pentru a oferi servicii suplimentare educaționale și de susținere pentru copii migranți, a inclus cerința ca înregistrările copiilor migranți să fie

transmise de la școală la școală. Această sarcină a fost imposibilă fără un sistem eficient de comunicații pentru urmărirea acestor elevi. Prin urmare, în 1969, Migrant Student Record Transfer System (MSRTS) a fost creat pentru a oferi școlilor o metodă de a menține în permanență evidențe cu privire la studenții migranți.

### **Ce sunt studenții migranți?**

Studentul migrant este un copil al cărui părinte sau tutore își schimbă reședința regulat în interes de serviciu în cadrul unui an școlar, schimbând astfel și locul de școlarizare al studentului.

### **Ce este sistemul de registru al studenților migranți?**

MSRTS este o rețea națională de informații computerizate care menține și transferă datele educaționale și de sănătate de la școală la școală dacă copii migranți se mută cu familiile lor. Sistemul funcționează în cadrul unui contract între Departamentul de Educație al Statelor Unite și Departamentul de Educație Arkansas.

### **De ce a fost creat?**

Înainte de 1968, a existat o lipsă de continuitate în programul educațional pentru copii migranți. În acel an, a avut loc o conferință între reprezentanți din 37 de state pentru a discuta aceasta lipsă de continuitate și de alte probleme, inclusiv transferul de informații educative despre copii migranți de la un stat la altul și de la un profesor la altul.

Comitetul registrului de transfer, instituit de această conferință, a fost însărcinat cu două responsabilități:

- Pentru a concepe un registru pentru studenți adecvat
- Proiectarea unui sistem de transfer

Încă de la începuturile sale, sistemul a coordonat eforturile sale cu programe locale și naționale de migranți, facilități medicale, și nenumărate alte entități în mod direct sau indirect, responsabil pentru deservirea populației migrante.

### **Cine îl folosește?**

MSRTS este utilizat de aproximativ 30\% din școlile din Statele Unite, Districtul Columbia și Puerto Rico, precum și de nenumărate facilități de sănătate care sunt subcontractate de a presta servicii de sănătate pentru studenții migranți. Multe state solicită și primesc rapoarte de management create și produse de MSRTS.

### **Erorile care pot apărea în timpul actualizării**

Spre deosebire de o actualizare la un browser web sau un procesor de text, modificările și actualizările la aceste sisteme tind să aibă un impact semnificativ asupra operațiunilor de zi cu zi a fiecărui angajat din școală.

Aceste sisteme ating de obicei fiecare aspect al operațiilor din școală chiar și atunci când doar modulele de bază sunt utilizate.

Din aceste motive, trebuie să se ia în considerare impactul asupra:

- Fluxului de lucru: Din moment ce aceste programe sunt strâns legate de fluxul de afaceri dintr-o școală și a proceselor acesteia, o modificare la un sistem SIS poate să cauzeze modificări la fluxul de lucru. Acest lucru poate avea un impact semnificativ asupra operațiunilor de zi cu zi, dacă nu se consideră cu atenție înainte de punerea în aplicare
- De conversie a datelor: Conversia datelor de date istorice (transcrieri, prezență, registrele de sănătate, etc), atât pentru studenți actuali cât și pentru foști studenți poate fi de asemenea o problemă importantă pentru tranziția către un nou SIS. Deoarece majoritatea școlilor au obligația de a păstra datele istorice despre foști studenți, ar trebui să ne gândim ce informație ar trebui convertită și care ar trebui băgate în arhive.
- Rapoarte personalizate: Deoarece există puțină standardizare în modul în care informațiile despre studenți sunt stocate, majoritatea școlilor au propriile lor procese și proceduri. Deoarece cele mai multe programe SIS nu sunt compatibile cu programele mai vechi SIS, actualizarea lor poate fi un proces lung și plictisitor.
- Instruire: Unele programe noi SIS au tendința de a include anumite caracteristici inutile, în principal ca să fie folosite de utilizatori experimentați, astfel instruirea noilor angajați va fi cel mai probabil un proces costisitor și de lungă durată.

## 2.2 Curba de învățare

Tipuri fundamentale ale activităților de evaluare educațională

Astăzi se discută din ce în ce mai mult despre necesitatea unui „cultură evaluative” care trebuie formată și promovată în rândul cadrelor didactice, dar și în rândul elevilor și părinților. În teoria pedagogică modernă, evaluarea nu mai este văzută ca o acțiune de control-verificare a cunoștințelor însușite, ci mai degrabă, ca o apreciere a rezultatelor învățării, pe deoparte și a proceselor de învățare, pe de altă parte, ceea ce „semnifică trecerea de la o pedagogie a transmiterii cunoștințelor la o pedagogie a însușirii cunoștințelor și a științei de a deveni”. În cele din urmă vom evalua principalele tipuri de evaluare prezentate în practica educațională curentă din sistemul

nostru de învățământ, focalizându-se pe aspecte de mare actualitate ale pedagogiei moderne potrivit căreia evaluare normativă și ceea sumativă pierd tot mai mult teren și tind să fie înlocuite cu evaluarea criterială, cu evaluarea formativă sau mai recent cu evaluarea formatoare.

În prezentarea evaluării inițiale, normative, criteriale, formative, formatoare și sumative, tipurile de evaluare educațională ce urmează să fie etalate mai jos, vom avea în vedere următoarele esențe:

- definirea conceptelor;
- exemplificare practici de evaluare;
- evidențierea scopului și necesității educaționale;
- descrierea principalelor caracteristici;
- esența fiecărui tip de evaluare în parte ;

Evaluarea inițială Evaluarea inițială reprezintă activitatea de evaluare desfășurată la începutul procesului de învățământ cu scop constatativ, diagnostic și prognostic, și care produce un set de rezultate ce vor sta la baza desingnului și desfășurării ulterioare a procesului instructiv-educativ.

Evaluarea inițială îndeplinește două funcții importante:

- funcția diagnostică: determinarea măsurii în care elevii stăpânesc cunoștințele și dețin abilitățile necesare implicării în procesul de învățare care va urma; în anumite cazuri, „ evaluarea inițială poate demonstra necesitatea conceperii unei secvențe de recuperare a ceea ce elevii nu stăpânesc satisfăcător, pentru reamintirea unor conținuturi sau chiar învățarea acestora”;
- funcția prognostică: anticiparea condițiilor optime de proiectare și desfășurare a activităților instructiv –educative, stabilirea obiectivelor pragului de învățare viitor și a strategiilor didactice care vor fi utilizate pentru a asigura condițiile de reușită ale activității.

Ca atare, prin realizare evaluări inițiale, cadrul didactic poate identifica lacunele existente în pregătirea elevilor și principalele dificultăți de învățare ale clasei, fixându-și astfel, în consecință de cauză, obiectivele educaționale și eventual, introducerea în programul de predare-învățare a unei secvențe de recuperare a cunoștințelor pe care elevii nu le stăpânesc suficient. Deși evaluarea inițială nu este o noutate în practica școlară aceasta nu este în prezent foarte frecvent utilizată din cauza presiunii volumului conținutului educațional, a timpului școlar și a mentalității anumitor cadre didactice conform cărora evaluarea inițială constituie o risipă de timp.

Considerăm că, dat fiind specificul învățării elevilor și dinamica schimbărilor educaționale actuale, evaluarea inițială este o etapă absolut necesară realizarea unui demers de instruire eficient, care să aibe efectiv șanse de



reușită, pornind și bazându-se în mod real pe capacitățile, nevoile și cerințele educabililor concreți cu care se lucrează. Pentru a putea realiza o planificare realistă și o proiectare focalizată pe nevoile și pe specificul elevilor, trebuie să abordăm evaluarea inițială ca o primă etapă a demersului educațional în urma căreia putem reuși să obținem informații utile pentru desfășurarea ulterioară a activității instructiv-educative. Așadar, este limpede că atunci când cadrele didactice ignoră acest tip de evaluare pot apărea probleme în proiectarea activităților instructiv-educative, în sensul neadaptării obiectivelor educaționale și strategiilor didactice la realitățile concrete din clasă, ceea ce va determina un proces de predare insuficient adaptat nevoilor clasei, implicit o învățare sincopată și consecutiv o evaluare cu rezultate viciate.

### **Evaluarea normativă**

Evaluarea normativă verifică achizițiile și performanțele elevilor sub formă de rezultate ale învățării, recurgând la un referențial specific reprezentat de o normă, o medie statistică a rezultatelor înregistrate de un număr suficient de mare și de omogen de elevi. Evaluarea normativă încă cel mai frecvent tip de evaluare, deși, în timp, ponderea ei s-a diminuat substanțial și continuă să se diminueze.

În cadrul evaluării normative, achizițiile tuturor elevilor (cunoștințe sau competențe) sunt raportate la o normă, stabilită statistic în funcție de rezultatele obținute la evaluări de către elevi aparținând generației sau generațiilor anterioare, ceea ce înseamnă că, în evaluarea normativă se pornește totuși de la premisă, discutabilă de altfel, că în orice grup școlar mai mic reproduce simptomatic și repetitiv, din punct de vedere performanțial, post-învățarea grupul mare, grupul generației, ceea ce legitimează raportarea tuturor elevilor oricărei clase de același nivel la norma statistică aferentă, motiv pentru care notele acordate elevilor nu măsoară performanțele individuale, nu au valoare propriu zisă, ci compară ceea ce știu elevii cu norma statistică. Așadar, referențiali de gen normă pentru evaluarea normativă pot fi de exemplu, indicatori probelor sportive de rezistență, viteză, forță etc. Desfășurate în școală la orele de Educație fizică și sport sau al examenul de Bacalaureat.

Evaluarea normativă se bazează pe distribuția statistică a performanțelor elevilor reprezentată grafic prin curba în formă de clopot a lui Gauss, conform căreia, în urma raportării elevilor, și a rezultatelor acestora la criteriile de notare și instrumente de evaluare a performanțelor școlare, se realizează clasificarea elevilor și încadrarea acestora în intervalele de pe curba distribuției normale. Demersul normativ de evaluare presupune definirea în termeni relativi sau procentali a performanțelor standard conform modelului oferit de curba lui Gauss și elaborarea în termen de conținut a standardelor evaluative. Astfel unii elevi au acces sau ating numai nivelul standardelor inferioare, iar alții înaintează către standardele superioare și o parte din ei le și ating.

Conform curbei lui Gauss, în orice clasă există: elevi foarte buni(2%), elevi buni(13%), elevi de nivel mediu(70%), elevi slabi(13%) și elevi foarte slabi(2%), rezultând o distribuție grafică în formă de clopot, precum cea de mai jos.

În urma analizei rezultatelor evaluării, elevi sunt comparați, clasati, ierarhizați în cadrul clasei în cadrul școlii din care fac parte. De cele mai multe ori, elevii ajung să conștientizeze aparența lor la categoria elevilor buni sau slabi, foarte buni sau foarte slabi. Din păcate, foarte frecvent la această etichetare contribuie, pe lângă profesori, chiar părinții elevilor, colegi etc. În așa fel încât școlarii se identifică cu una sau alta din categoriile rezultate din utilizarea Curbei lui Gauss.

În opinia anumitor cadre didactice, evaluarea normativă este necesară tocmai pentru că reușește să ierarhizeze elevii, considerându-se că, astfel, aceștia reușesc să își cunoască mai bine propria valoare și să devină mai competitivi. Principalele dezavantaje ale evaluării normative se referă la faptul că aceasta ignoră procesul de învățare și calitatea sa, are drept scop ierarhizarea, clasificarea și selecția elevilor, este imprecisă și irelevantă pentru că reușește să surprindă doar dimensiunea cantitativă a procesului de predare-învățare, dar și faptul că este o evaluare constatativă, insuficient diagnostică, stresantă, constrângătoare și frustrantă. Tocmai aceste neajunsuri a evaluării normative au determinat apariția și dezvoltarea altor tipuri de evaluare, dintre care cele mai importante le consideră a fi evaluarea criterială și evaluarea formativă pe care le prezentăm în cele ce urmează.

### **Evaluarea criterială**

Evaluarea criterială este procesul evaluativ ce urmărește verificarea și aprecierea cunoștințelor elevilor prin raportarea acestora la obiectivele educaționale stabilite și nu la o normă statistică, fiind denumită și evaluare prin obiective. Prin urmare, evaluarea criterială constă în stabilirea cu mai multă rigoare și finețe a ceea ce se numește în literatura de specialitate standardul minim acceptat sau performanța minimă acceptată, care exprimă pragul de reușită al unui elev într-o anumită situație educațională.

Principalele caracteristici ale evaluării criteriale sunt:

- raportează achizițiile elevilor la obiectivele educaționale(în special operaționale) stabilite care devin criterii de evaluare, nu la o normă statistică impersonală;
- criteriile de evaluare pot fi definite ca parametrii acceptabili, adaptați și flexibili la care se raportează performanțele elevilor;
- permite nuanțarea și personalizarea sarcinilor și timpului instructiv-educativ;

- are drept scop furnizarea de informații funcționale, permițând elevilor să își conștientizeze achizițiile în raport cu obiectivele stabilite la începutul programului de învățare;
- facilitează identificarea și punerea în practică a măsurilor ameliorative, acolo unde este cazul, pentru a ușura învățarea elevului;
- umanizează și personalizează procesul de evaluare, elevul fiind abordat ca o personalitate, nu ca un elev mediu statistic.

Evaluarea criterială respinge distribuția statistică aleatoare a elevilor și promovează „lupta împotriva curbei lui Gauss”, tocmai din cauză că aceasta duce la scăderea motivației elevilor, pe deoparte a cadrelor didactice, pe de altă parte, soldându-se cu o atitudine de resemnare în ambele tabere. Curba în formă de „J” sau curba succesului, specifică evaluării criteriale propune o abordare mult mai optimistă a evaluării educaționale, potrivit căreia cel puțin 80% dintre elevii unei clase pot învăța (la un nivel satisfăcător ceea ce li se predă) și se pot atinge obiectivele educaționale propuse. O astfel de curbă a succesului poate fi reprezentată grafic conform figurii de mai jos.

#### **Evaluarea formativă/continuă**

Evaluarea formativă/continuă poate fi definită ca ansamblul activităților de evaluare, desfășurat de-a lungul procesului instructiv-educativ, urmărind identificarea punctuală a nivelului de pregătire al educabilului, nivel considerat mediat în designul și desfășurarea efectivă, momentană și ulterioară a procesului educațional.

Denumită și evaluarea pe parcurs sau de progres, evaluarea formativă își merită cu adevărat numele în măsura în care datele produse de procesul evaluativ sunt reintroduse în activitatea didactică facilitând realizarea acesteia într-o manieră centrată efectiv pe dificultățile, nevoile și dezideratele educabililor concreți cu care se lucrează.

Principalele caracteristici ale evaluării formative sunt următoarele:

- evaluarea devine o componentă firească a activităților educaționale;
- intervine în timpul fiecărei sarcini de învățare;
- informează elevul și profesorul asupra gradului de stăpânire a obiectivelor;
- permite profesorului și elevului să determine dacă acesta din urmă posedă achizițiile necesare pentru a aborda viitoarele sarcini de învățare;
- permite adaptarea activităților de învățare la necesitățile elevilor;

- are ca scop facilitarea depășirii dificultăților de învățare ale elevilor;
- este o evaluare internă continuă, analitică și centrată mai mult pe cel ce învață decât pe produsul finit;
- acest tip de evaluare devine util atât pentru elev cât și pentru profesor.

În raport cu celelalte tipuri de evaluare, evaluarea formativă prezintă avantaje incontestabile pentru că:

- identifică și corectează prompt și eficient problemele apărute în cadrul procesului de învățare foarte aproape de momentul în care acestea apar, fără a le lăsa să se perpetueze;
- elevul nu este judecat și clasificat;
- abordează aproape toate conținuturile de învățare și aproape toți elevii;
- menține motivația și interesul elevilor și profesorilor la un nivel ridicat;
- facilitează relaționarea și comunicarea dintre profesor și elev;
- profesorul beneficiază de un feed-back imediat din partea elevilor;
- oferă posibilitatea tratării diferențiate a elevilor;
- asigură un grad ridicat de activism în procesul instructiv;
- educativ și crează o anumită obișnuință a elevului cu evaluarea;
- consumă mai puțin timp în raport cu celelalte tipuri de evaluări.

În evaluarea formativă nereușitele elevului nu sunt considerate eșecuri sau manifestări ale interesului/dezinteresului acestuia față de respectiva disciplină, ci momente cruciale în dezvoltarea problemelor apărute în învățarea elevului. Pentru a putea pune în practică cu succes demersurile de evaluare formativă, cadrul didactic trebuie să îl determine pe elev să realizeze importanța autoevaluării, și să participe la evaluarea propriilor sale rezultate sau realizări, să devină conștient de eventualele greșeli care au apărut în munca lui și de necesitatea îndreptării propriilor greșeli. În lipsa practicilor de evaluare formativă corectă, pot apare dificultăți majore în procesul de învățare a elevilor, care vor produce consecutiv dificultăți în predare și evaluare.

#### **Evaluarea formatoare/conștientizată**

Actualmente, teoriile și practicile referitoare la evaluarea „conștientizată” sau „formatoare” se află în prim proces evolutiv, având la bază argumente

recente din domeniul pedagogiei și psihologiei cognitive, privind integrarea evaluării în procesul de învățare. Evaluarea formativă/conștientizată definește demersul de implicare activă a educabilului într-o activitate asistată de gen autoevaluare cu scop formativ, care să îi permită acestuia conștientizarea propriului nivel de pregătire și, pe cale de consecință, deciderea următorilor pași ai procesului învățării.

Acest tip de evaluare educațională se constituie ca o nouă megaconcepție corespunzătoare unui demers pedagogic care favorizează participarea activă și autonomia elevului, furnizându-i repere explicite în scopul de a lua în mâini propria transformare fiind conștient de propriile dificultăți și lacune.

Pentru a înțelege mai bine conceptul de evaluare formativă și modalitățile prin care aceasta poate fi aplicată în situații educaționale reale este necesar să cunoaștem principalele repere care jalonează acest tip de evaluare:

- evaluarea formativă devine eficace când se transformă în autoevaluare și reglează procesul învățării;
- evaluarea formativă se focalizează nu doar pe procesul de învățare ci și pe competențele și produsele rezultate;
- deși valorizează relația predare-învățare, evaluarea formativă se axează mai mult pe învățare;
- evaluarea formativă pune accentul mai mult pe rolul elevului și pe înțelegerea de către acesta a obiectivelor educaționale vizate și a procesului de învățare;
- cel mai indicat „reglator” al activității de învățare este însuși elevul, acesta fiind cel mai în măsură să își corecteze propria activitate de învățare.

Prin intermediul evaluării formative, elevii își însușesc obiectivele și criteriile de evaluare, reușind astfel să își aprecieze propriile rezultate în cunoștință de cauză. Evaluarea formativă îl ajută pe elev să își conștientizeze nivelul la care se plasează, îl ajută să învețe pentru că îl implică în procesul de învățământ și din perspectiva de evaluator, nu doar din cea de subiect al evaluării. Lipsa evaluării formative se reflectă în lipsa unei imagini adecvate a elevilor despre propriul lor nivel intelectual și educațional.

### **Evaluarea sumativă/finală**

Evaluarea sumativă/finală/cumulativă reprezintă demersul de evaluare ce se realizează la finalul unui program de învățare, urmărind identificarea și valorizarea efectelor acțiunii educaționale, concretizate în rezultate brute ale

învățării. Acesta este tipul de evaluare pe care îl întâlnim cel mai frecvent în sistemele de învățământ ierarhizate, normative și bogate în conținut de învățare, cum este de altfel și cazul sistemului românesc de învățământ.

Evaluarea finală are drept scop măsurarea gradului în care subiecții evaluații, elevi, au atins obiectivele educaționale vizate și au dobândit competențele cerute. Evaluarea sumativă sau cumulativă se practică la finalul unor intervale suficient de mari de timp care permit acumularea unei importante cantități de cunoștințe, a unui cumul de achiziții.

Evaluarea cumulativă deține următoarele caracteristici:

- Este o evaluare de bilanț realizată preponderent la final, dar poate include și verificări pe parcursul programului;
- Este focalizată pe rezultatele globale ale învățării, pe produsele acestora și nu pe procesul de învățare;
- Are ca finalitate atribuirea unor note sau calificative, certificate sau diplome și, uneori, sancționarea pozitivă sau negativă a elevilor și selectarea sau ierarhizarea acestora;
- De cele mai multe ori, evaluarea finală este o evaluare externă;
- Este o evaluare predominant retrospectivă pentru că, intervenind la final nu mai poate influența procesul de învățare și rezultatele elevilor;
- Favorizează competiția între elevi, dar aduce un plus de stres în activitatea educațională și a cadrelor didactice, destructurând relația de comunicare formată între profesori și elevi;
- Reprezintă un indicator indirect al eficacității prestației cadrelor didactice și a calității procesului de instruire;
- Are o fidelitate redusă raportată la nivelul general de pregătire a elevilor și implicit o credibilitate scăzută.

Deși evaluarea sumativă constituie un tip de evaluare complex, ar fi absurd să se creadă, însă, că la evaluarea finală elevul trebuie să știe tot. El trebuie să știe lucrurile esențiale din materia parcursă și, în plus, să știe să se folosească de ele în diverse situații, deci să știe să utilizeze cunoștințele învățate în practică.

## 2.3 Arhitecturi ale sistemelor de E-Learning

Serviciile Web reprezintă un standard al comunicării între diferite aplicații software, funcționând pe tipuri de platforme diferite și pe framework-uri diferite, reprezentând un standard de referință în promovarea interoperabilității, extensibilității între aplicații, precum și combinării pentru realizarea unor operații complexe. Există numeroase arhitecturi de e-learning care:

- propun un framework pentru realizarea și dezvoltarea de sisteme educaționale on-line bazate pe agenți care integrează agenții software și tehnologii specifice obiectelor educaționale;
- se adresează problemei interoperabilității (drept consecință a proliferării sistemelor educaționale on-line) și alege CORBA ca infrastructură tehnologică;
- dezvoltă o arhitectură funcțională deschisă, bazată pe analiza proceselor implicate în managementul și pe furnizarea de conținut educațional, arătând cum rezolvă tehnologia agent problemele de planificare a învățării colaborative și care este prezentarea optimă a conținutului didactic;
- dezvoltă un model pentru aplicații colaborative orientate pe Web.

Arhitectura funcțională a unui sistem de e-learning definește componentele acestuia (de exemplu standardul SCORM definește modelul general funcțional al unui sistem de management al învățării). Standardele unui sistem de e-learning se referă în principal la:

- Metadata: pentru conținutul învățării și catalogare este necesară o etichetare consistentă care va permite indexarea, stocarea și regăsirea de obiecte ale învățării de către diferite utilitare (LOM).
- Pachete de conținut: standardele și specificațiile pachetelor de conținut permit transferul cursurilor de la o platforma la alta (de exemplu IMS Content Packaging, IMS Simple Sequencing, ADL SCORM).
- Profilul cursantului: datele personale, planurile învățării, istoricul învățării, cerințele de accesibilitate, certificările și diplomele, gradul de cunoaștere și participare la învățare (Learner Information Profile - LIP).
- înregistrarea cursantului: informațiile furnizate la înregistrare permit componentelor de administrare și furnizare de conținut să identifice și să

regăsească elementele necesare acestora (IMS Enterprise Specification, Schools Interoperability Framework).

- Comunicarea conținutului: se furnizează datele cursantului și informațiile despre activitățile precedente (ADL SCORM).

Sistemul de management al învățării definit de SCORM poate fi împărțit în sistem de management al conținuturilor învățării (LCMS) și sistem de management al învățării (LMS). Un sistem de management al conținutului învățării este un mediu multi-utilizator, unde dezvoltătorii de conținut pot crea, stoca, reutiliza, realiza management și furniza conținut digital de învățare. LMS realizează managementul din jurul învățării, iar LCMS realizează crearea și furnizarea conținutului de învățare. LCMS permite utilizatorilor să creeze și să refolosească mici unități de conținut digital educațional. Utilizarea standardelor structurilor de metadate ale învățării, precum și a standardelor formatelor de import și export permit obiectelor învățării să fie create, folosite în comun de utilitare multiple și depozite de conținuturi de învățare. Pentru realizarea interoperabilității între sisteme, LCMS este realizat utilizând specificațiile standard pentru metadatele de conținut, pachetele de conținut și comunicarea conținutului. El interschimbă profilul utilizatorului și informațiile furnizate la înregistrare cu alte sisteme, localizează cursurile în cadrul LCMS și furnizează acțiunile cursantului.

Serviciile Web sunt perfect fezabile pentru interoperabilitatea sistemelor de e-learning din următoarele motive:

- Informația interschimbată între sistemele de e-learning respectă standardul XML.
- Arhitectura serviciilor Web este independentă de limbaj și platformă, promovând interoperabilitatea și extensibilitatea între diferite aplicații, platforme și framework-uri existente pe piața reală.
- Serviciile Web furnizează un model de programare unificat pentru dezvoltarea și folosirea serviciilor intranet și Internet.

Arhitectura unui sistem de e-learning definește modul diferit de transfer de mesaje în interacțiunea cu serviciile agenților Web pentru fiecare sistem. Furnizorul de servicii este platforma care găzduiește accesul la servicii și este un mediu de execuție pentru servicii sau un container pentru serviciu (rolul său în schimbul de mesaje este cel al unui server). Serviciul de cerere caută sau inițializează interacțiunea cu serviciul. Agenția de descoperire a serviciilor este un set de servicii unde furnizorii publică descrierea serviciului lor, care poate fi centralizat sau distribuit. Informația educațională prezentată



în format XML și ținând cont de specificațiile SOAP (Simple Object Access Protocol) este schimbată între serviciul de cerere și furnizorul de servicii. Furnizorul de servicii publică un fișier WDSL conținând descrierea mesajului și informația de sfârșit, ceea ce permite serviciului de cerere să genereze un mesaj SOAP și să-l trimită spre destinația corectă.

### 2.3.1 Arhitectură cu agent dublu

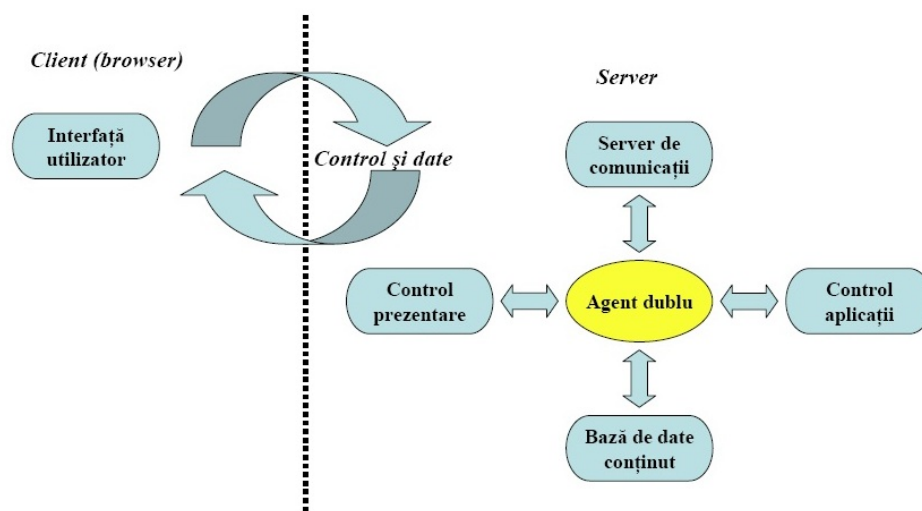
La realizarea unei astfel de arhitecturi trebuie să se țină seama de un set de întrebări legate de interactivitatea conținutului: cum se realizează acest lucru?; care este tehnologia care ar trebui folosită?; de care aspecte ar trebui să se țină seama la implementarea unei aplicații?; ce se întâmplă dacă sunt concomitent mai mulți utilizatori și mai multe aplicații? Unul din aspectele fundamentale ale aplicațiilor de e-learning este controlul acțiunilor utilizator, cu alte cuvinte adaptarea preferințelor utilizatorului și a performanțelor. Cerințele educaționale fac necesară pe lângă adaptarea preferințelor utilizatorului și adaptarea performanței cursantului. Astfel, agentul dublu nu reține numai istoricul acțiunilor utilizatorului ci și ghidează utilizatorul. Cerințele de care ar trebui să se țină seama la realizarea unei astfel de arhitecturi sunt :

- Controlul sesiunii: capacitatea de identificare a utilizatorului și a activităților acestuia în cadrul sesiunii;
- Controlul interactivității, în care se includ: -independența de platforme și aplicații; -suportul pentru utilizatori simultani; -suportul pentru aplicații diferite; -compatibilitatea cu tehnologiile actuale; -posibilitatea de a integra noi tehnologii.

Arhitectura constă într-o interfață utilizator, un agent dublu și conținut (figura 1).

Ideea centrală este că orice cerere a utilizatorului este verificată și procesată mai întâi de un agent. Din punctul de vedere al utilizatorului, acestuia îi este vizibil numai agentul, funcționalitățile fiind încapsulate. Agentul este componenta responsabilă de feedback-ul furnizat utilizatorului, fiind de asemenea cel care controlează ceea ce vede utilizatorul. Natura duală îi dă numele acestuia (agent dublu) și simulează procesul educațional dintre student și profesor. El trebuie să identifice, să verifice și să evalueze cererile ținând cont și de cele precedente, ceea ce reprezintă controlul sesiunii, care se realizează prin autentificare, mecanismul de identificare a cererilor, un istoric și un jurnal al acestora. De asemenea trebuie ținut cont de preferințele utilizatorului și de mediul de lucru al clientului.

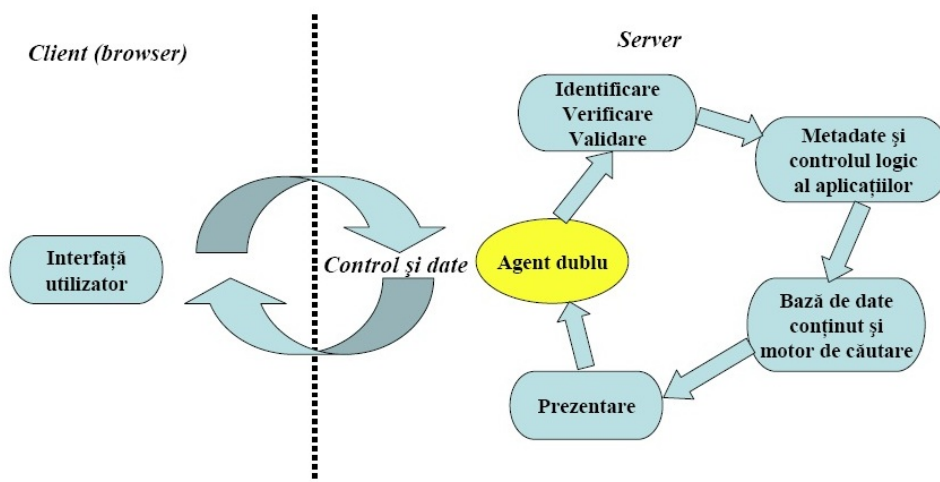
Figura 1: Arhitectura generală cu agent dublu



Blocul de control al aplicației este cel care se ocupă de găsirea și crearea unui răspuns adecvat pentru cererea realizată de client. Conținutul poate fi stocat într-o bază de date sau realizat din componente preexistente în momentul cererii. Indiferent de modul de stocare, răspunsul privind conținutul adecvat este realizat în momentul cererii, dar cel puțin o parte din acesta este stocat într-o bază de date. Blocul de control al aplicației conține cheile de indexare a conținutului în baza de date sau regulile de creare a acestora. Blocul de control al prezentării adaptează conținutul la preferințele utilizatorului și la posibilitățile de lucru ale browser-ului acestuia. Din punct de vedere funcțional, agentul dublu este definit ca o structură care conține toate blocurile funcționale descrise sau consideră în aplicație că blocul de control și cel de prezentare sunt separate. Separarea este utilă, dar din punctul de vedere al utilizatorului de întreaga funcționare se ocupa agentul. Un alt mod de a descrie funcționarea sistemului este de a considera sistemul având mai mult de doi agenți. De exemplu se separă agentul funcțional, într-un agent utilizator și un agent aplicație, dând astfel un alt sens agentului dublu (figura 2). Agentul primește cererea de la client, o identifică, o verifică și o transmite la blocul de control al aplicației. Pe baza informației transmise de agent, conținutul necesar prezentării se regăsește în baza de date.

Există trei cazuri speciale de cereri: login, logout și eroare. Pentru a realiza cererea de login sunt necesare mecanisme de realizare a autentificării și pregătirea sesiunii de control, la care se adaugă crearea unui nou utilizator, schimbarea parolei, inactivarea unui utilizator. Pentru logout sunt necesare mecanisme de terminare a sesiunii și salvarea stării curente a utilizatorului

Figura 2: Arhitectură cu mai mult de doi agenți



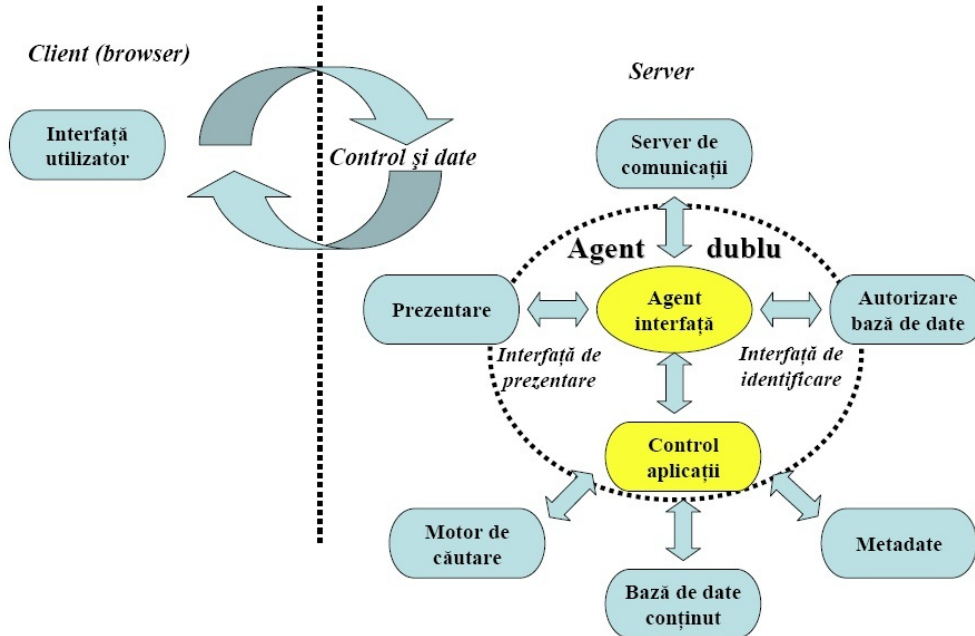
și, de asemenea, trebuie avut în vedere un criteriu de timp pentru a afla dacă utilizatorul mai folosește aplicația sau nu. Erorile pot fi de două tipuri: la nivel de sistem și la nivel de conținut. Erorile la nivel de sistem pot fi legate de controlul sesiunii, atacuri și probleme la server, iar erorile la nivel de conținut se referă la parametri de intrare eronați. Această problemă se rezolvă în principiu la nivelul blocului de control al aplicației.

Interactivitatea poate fi realizată la nivelul clientului, al serverului sau la ambele. Cea mai răspândită soluție este de a folosi tehnologiile orientate-server, limitând operațiile pe client la introducerea de date în formulare HTML (această abordare are avantajul că furnizorul aplicației are control complet asupra procesului de interacțiune). Un alt beneficiu este independența de software-ul client, interacțiunea constând într-un simplu transfer de parametri. Interactivitatea la nivelul serverului implică patru pași :

1. crearea conținutului interactiv;
2. modificarea conținutului interactiv astfel încât să răspundă cerințelor prezentării și ale sesiunii de lucru;
3. validarea parametrilor de intrare;
4. evaluarea conținutului actual transmis de utilizator;

Interactivitatea la nivelul clientului este utilă în momentul când în conținut există animații, simulări, jocuri educative. În acest caz sunt necesare mai multe operații la nivelul arhitecturii, deoarece nu toți clienții furnizează

Figura 3: Implementarea arhitecturii cu mai mulți agenți



funcționalitățile și elementele necesare comunicării cu agentul. O problemă care poate apărea în acest caz este legată de transferul și furnizarea intrărilor utilizatorului.

Pentru serverul de aplicație este bine de folosit un server securizat (SSL) care oferă în afară de securitate și actualizarea conținutului (în cazul existenței unui server proxy, datele nu sunt cele furnizate din cache). Activitate în rețea crește prin folosirea acestei soluții, iar clienții necesită suport SSL.

Autentificarea utilizatorului se face prin identificarea acestuia, după care informațiile sunt ascunse prin intermediul sesiunii de control. În practică, parolele și conturile sunt reținute într-o bază de date împreună cu alte informații utile. Înregistrările din baza de date au cont de identificare unic, acesta fiind inclus în ID-ul de control al sesiunii. Scopul sesiunii de control este de a identifica cererile și a crea legăturile dintre blocurile existente, formând o sesiune unitară care ține cont de limitările temporale. Metoda standard de implementare este de a transmite un ID de sesiune la fiecare răspuns URL și de a accesa datele specifice utilizator prin intermediul ID-ului. De asemenea, cookie-urile pot fi folosite pentru a asigura nemodificarea software-ului client sau a calculatorului acestuia. Sesiunea de control ar trebui să aibă implementată posibilitatea de realizare a unui checksum astfel încât ID-ul sesiunii să nu poată fi alterat. Fiind o arhitectură cerere-răspuns, Web-ul nu realizează un control al timpului. Tehnicile prezente în controlul sesiunii trec peste

această limitare prin realizarea unui istoric al sesiunii (în care se face și conținutul temporală), ceea ce permite agentului să măsoare și să folosească informațiile temporale pentru aplicația de control. Dacă paginile Web sunt statice, blocului de control al aplicației îi sunt necesare operații de regăsire a listei de fișiere care trebuie furnizate din baza de date. Pentru a găsi următoarea pagină se folosește istoricul sesiunii, bazându-se pe cea precedentă. O altă abordare pentru regăsirea paginii următoare este folosirea metadatelor, a căror definire este însă dependentă de aplicație. Odată cu adăugarea de elemente interactive, trebuie să existe metode pentru manipularea datelor de intrare. Logica blocului de control al aplicației trebuie să cuprindă metode pentru identificarea elementelor client, precum și metode de validare a datelor de intrare. Pentru descrierea și modelarea logicii blocului de control al aplicației se poate folosi UML, iar pentru blocul de prezentare ar trebui folosit un format XML/XSL (pentru reprezentarea internă a conținutului) și limbajul XSLT (pentru convertirea formatului în prezentări).

### 2.3.2 Arhitectură Web Service bazată pe mesaje

Internetul oferă o infrastructură distribuită pentru partajarea informațiilor la nivel global. Piața foarte mare de utilizatori devine o motivație mare pentru noile tehnologii, constituind punctul de plecare în realizarea următoarei generații de aplicații Web. Această evoluție aduce schimbări fundamentale în societate, în comunicații și pattern-uri de achiziție de cunoștințe oricând, oriunde, oamenii nu trebuie să se mai întâlnească față în față pentru a comunica, toată informația fiind furnizată clientului prin interfață on-line sau la cerere. Noua tendință cuprinde caracteristici tehnologice inovative:

- oferă o platformă care facilitează accesul omniprezent la calculatorul, PDA-ul și telefonul mobil (Windows, MacOS, UNIX, Linux, PalmOS) ale clienților;
- oferă o interfață de servicii, care oferă disponibilitate ușoară la resursele globale, cum ar fi date, text, grafice 2D și 3D, fluxuri audio/video, muzică MP3;
- promovează un mecanism sincron interoperabil care capturează interacțiunile dintre participanți profesor și student, instructor și ucenici pentru experiențe în timp real. Instrumentele de colaborare au revoluționat industria de instruire. În special soluțiile elearning bazate pe Web adoptă medii de colaborare care intensifică accesul la o gamă completă de resurse educaționale ce suportă interacțiuni cu părțile participante într-un mod sincron sau asincron. Educația la distanță devine

un real succes prin folosirea unor instrumente cum ar fi conferințe audio/video și accesul la programe prin vizualizarea lor sau arhitecturi de evenimente.

Arhitectura pentru e-learning Web Service bazată pe mesaje oferă o abordare unificată pentru aplicațiile și aplicațiile de tip Web Service care încorporează flexibilitatea mesajelor și componentelor distribuite. O arhitectură generală bazată pe un model de Web service de tip SIMD și MIMD, combină atât învățarea coordonată de instructor cât și învățarea participativă. Această abordare derivă din arhitectura ModelViewController a aplicațiilor Web, cuprinde o schemă de publicare/subscriere bazată pe evenimente și oferă o colaborare efectivă peste medii de rețea eterogene cu o interactivitate mare a conținutului Web pentru clienți diverși. Premisa acestei arhitecturi de colaborare este construirea unei arhitecturi pentru aplicații Web care să ofere scalabilitate, reutilizare, interoperabilitate, accesibilitate universală și colaborare automată.

#### *Caracteristici tehnice*

Istoria Internetului și tehnologiei Web a arătat o evoluție a aplicațiilor Web de la arhitecturi dominate de sisteme client-server centralizate, cu conexiuni tradiționale punct la punct (unicast) la sisteme descentralizate de tipul P2P (peer-to-peer) și mai apoi la o rețea cu mecanisme multicast la nivel de aplicație și un model RPC (CORBA) derivat dintr-un sistem bazat pe metode, cerut de un singur sistem CPU (de exemplu aplicațiile desktop), dar cu apeluri de proceduri de la distanță pentru a suporta obiecte distribuite. Modelele client-server și P2P sunt potrivite pentru rezolvarea problemelor cu caracteristici aplicabile pattern-urilor lor, dar problemele reale pot fi complicate. Exemple pot fi văzute în aplicațiile paralele cu decompoziție pe mai multe dimensiuni. Pe de altă parte, RPC ca model funcționează bine cu obiecte distribuite sau componente reutilizabile, dar nu este prea scalabil. Modelul Web Service bazat pe mesaje oferă o abordare unificată, care încorporează flexibilitatea mesajelor cu distribuția componentelor. El se adaptează la natura scalabilă și diferită a Internetului și de asemenea promovează dezvoltarea aplicațiilor Web cu Web Services pentru reutilizare, interoperabilitate și scalabilitate.

Abordarea bazată pe mesaje descompune un sistem Web pe trei niveluri:

- rețelele fizice sau Internetul,
- infrastructura de mesaje și
- aplicația Web (figura 6).

Această separare poate îmbunătăți mult portabilitatea aplicațiilor (prin reducerea dependențelor datorate configurațiilor de rețea sau platformelor de bază) și, de asemenea, reduce timpul de publicare a aplicațiilor Web. Pe de altă parte, necesită o infrastructură de mesaje solidă în rețeaua TCP/IP, oferind o diversitate de servicii de comunicație care reconciliază diferențele dintre configurațiile de rețele de conexiuni și dezvoltarea aplicațiilor de nivel înalt.

Dezvoltarea aplicațiilor Web prezintă direcții diferite, dar și caracteristici comune cum ar fi interfețe utilizator și servicii pentru partajarea informațiilor și resurselor peste infrastructura Internet. Partajarea poate fi realizată sincron și asincron la fiecare etapă posibilă aflată de-a lungul liniei dezvoltării. Obiectele care trebuie sincronizate pot varia de la conținuturi Web (video, audio și fluxuri de rânduri de date), interacțiuni utilizator (operații de editare pe documente), programe distribuite (componente de simulare pe scară largă distribuite) la participanții unei echipe care sunt implicați în dezvoltare sau management. Partajarea poate fi organizată în stil unicast sau multicast al comunicației în grup. Prin urmare, în sensul cel mai general, este problema și serviciul principal al aplicațiilor Web a partajării, deși oamenii în mod obișnuit referă terminologia colaborare la aplicații Web sincrone în timp real cu constrângeri de timp și condiții. Obiectivul cheie al acestei abordări îl reprezintă proiectarea și implementarea unei arhitecturi uniforme pentru aplicații Web cu capabilități de colaborare automată.

În continuare este prezentată o paradigmă MVC explicit bazată pe mesaje (MMVC) ca arhitectură generală pentru aplicații Web. Este construită în jurul utilizării sistematice a serviciilor Web și prezintă o separare între model și view în patternul MVC. Au fost realizate experiențe inițiale de colaborare ca Web Service pentru a testa viabilitatea arhitecturii propuse în susținerea aplicațiilor interoperabile cu conținut grafic bogat și constrângeri de timp, prin exemple de scenarii profesor-student și jocuri on-line cu mai mulți jucători. Ca o extensie a scopului cercetării, autorii au convertit aplicațiile desktop în sisteme distribuite la nivel arhitectural. Aceasta s-a realizat prin înlocuirea modelului MVC bazat pe metode convențional cu modelul MVC bazat pe metode în schema Publică/Subscrie pentru o reutilizare maximă a produselor software existente.

Ca o provocare cheie a unei abordări noi cu utilizare sistematică a Web Service-ului pentru crearea de aplicații bazate pe mesaje, mulți factori subtili pot să nu fie adresați de considerația arhitecturală generală. Prin urmare, autorii au construit un prototip cu o arhitectură previzionată și au realizat experimente sistematice pentru a explora și identifica principii generale și caracteristici de implementare cheie asociate cu această abordare. Multimedia cu un conținut bogat în grafice și combinații de fluxuri media formează o

Figura 4: Browser SVG derivat din MVC bazat pe mesaj

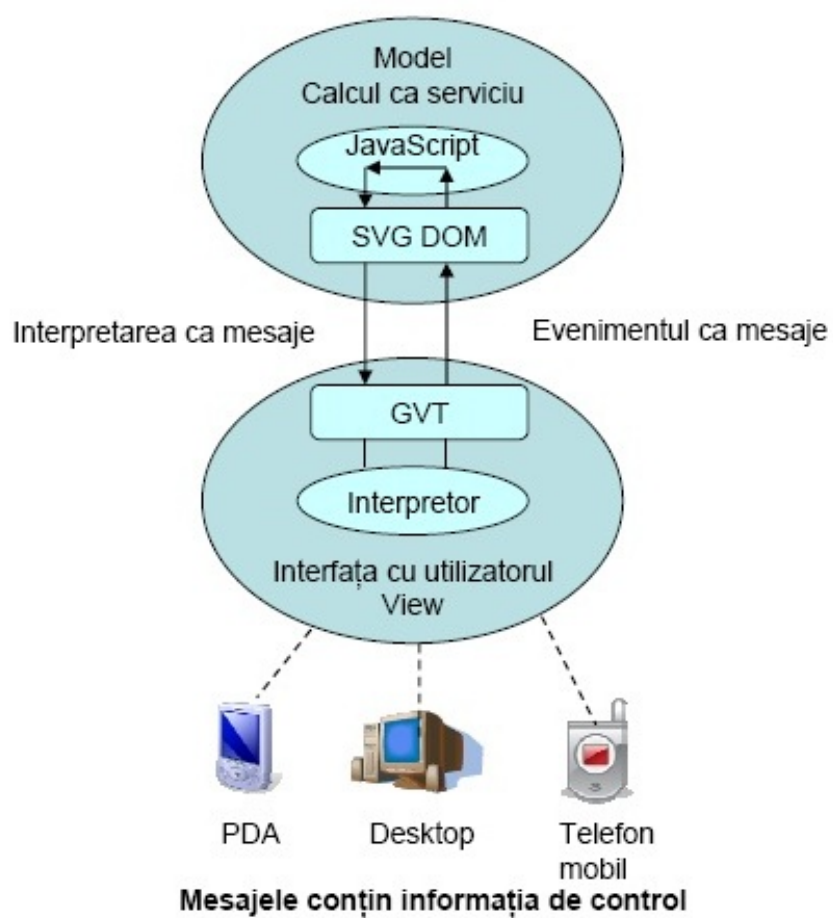
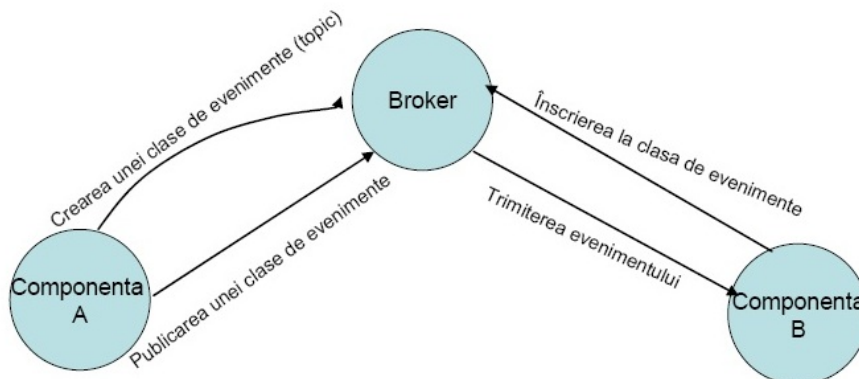




Figura 5: Schema Public/Subscrie bazat pe mesaj condus de eveniment



caracteristică importantă a aplicațiilor Web noi. A fost selectat un stil sugestiv de aplicație desktop cu două conținuturi SVG (Scalable Vector Graphics) dimensionale browser SVG Batik de la Apache pentru partea de testare, pentru experimente și evaluări viitoare. Autorii au restructurat software-ul Batic (open source) într-un model MVC bazat pe mesaje explicite (figura 5). Documentul SVG este analizat ca un arbore DOM (Document Object Model) cu noduri reprezentate de fragmente de document și obiecte grafice. Arborele GVT (Graphical Vector Toolkit) reflectă structura DOM și este folosit pentru interpretare avantajoasă. În această abordare, NaradaBrokering furnizează toate serviciile de mesaje (interacțiunea dintre clienți, evenimente și interpretare la client) cu arhitectura de Publică/Subscrie. Crearea de instrumente de colaborare pe SVG prezintă unele caracteristici generale importante:

- SVG este un standard open source pentru grafice vector 2D al W3C (World Wide Web Consortium). Este o tehnologie importantă pentru vizualizare, deoarece formatul conținutului XML și caracteristicile graficelor vector scalabile fac din ea o alegere ideală pentru transportul intermediar și redarea la o rezoluție înaltă. A fost aplicată la diferite aplicații incluzând servicii de conectare în Sistemul Informațional Geologic (GIS - Geological Information System) și instrumente de autorizare (viewer plug-in SVG de la Adobe și Corel). Caracteristica următoare este în special importantă pentru accesul universal de la dispozitivele wireless mici.
- Browser-ul SVG Batik are suport complet de specificații SVG 1.0. Claritatea standardului cât și a implementării ne oferă o experiență de valoare, oferindu-ne analize complete ale structurii sistemului și interacțiunilor dintre componente, care nu sunt disponibile la unele instru-

mente comerciale similare (exemplu: Microsoft PowerPoint, Macromedia Flash, Adobe Photoshop și Illustrator, și Corel Draw) cu implementare și format de date brevetate.

- SVG este construit pe W3C DOM, care este descrierea naturală pentru toate documentele desktop folosite în aplicații Office de acest tip realizate de Microsoft, Macromedia și OpenOffice. Astfel se poate generaliza o nouă arhitectură de Web service pentru următoarea generație de aplicații desktop, inclusiv acelea folosite pentru autorizarea programelor (figura 8). Specificațiile evenimentelor DOM oferă un model de eveniment generic care propagă schimbări în nodurile (obiectele) structurii arbore. Aceasta permite acestui tip de aplicație să aibă un serviciu Web comun care conduce o diversitate de interfețe client ce pot fi ori standalone ori colaborative.

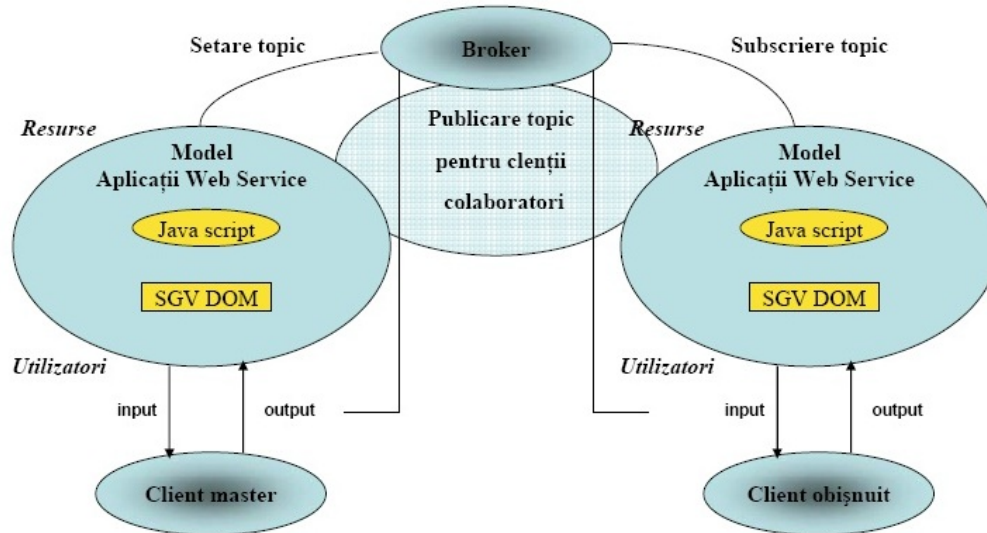
*Mediul de colaborare* Programarea bazată pe evenimente a devenit un stil de programare folosit pe scară largă care suportă mecanisme de manipulare întreruperi pentru interacțiuni cu utilizatorul la nivelul dispozitivelor de intrare. Comparat cu procedurile de procesare a întreruperilor anterioare, aceasta promovează modularitatea la nivel de sistem și răspuns asincron. Cele mai multe dintre sistemele desktop moderne, incluzând Microsoft Windows și aplicațiile lor folosesc paradigma MVC.

Sistemul este descompus în trei elemente - Model, View și Controller - și este comparabil cu evaluarea centrală (inclusiv structura de date), componentele vizuale și comunicarea dintre cele două clase separate sau moștenite. În MVC-ul bazat pe metode conduse de evenimente convenționale, serviciul de transmitere de mesaje este ascuns la nivel de sistem. Abordarea evenimentului bazat pe metode este de asemenea utilizată extensiv în sisteme distribuite, inclusiv Java AWT, Swing și alte aplicații. Ca un mecanism obișnuit de programare bazată pe evenimente, componentele de ascultare a evenimentelor subscriu la componenta producătoare a evenimentului și preia notificarea producerii evenimentului. În cazurile MVC ale browser-ului Swing și Batik SVG, componentele vizuale (view) alcătuiesc observatorii structurii de date (model) cu redarea modificării corespunzător cu schimbările modelului.

Abordarea de colaborare bazată pe mesaje conduse de evenimente cu schemă de publicare/subscriere are următoarele implicații:

- Un eveniment definește schimbarea incrementală a stării sistemului. Sistemul colaborativ bazat pe evenimente funcționează în mod sincron cu evenimentele de actualizare a informațiilor comunicate între părțile participante. În plus, evenimentele pot fi așezate la coadă și stocate ca înregistrări pentru recuperare și repunere și avem aceste servicii în

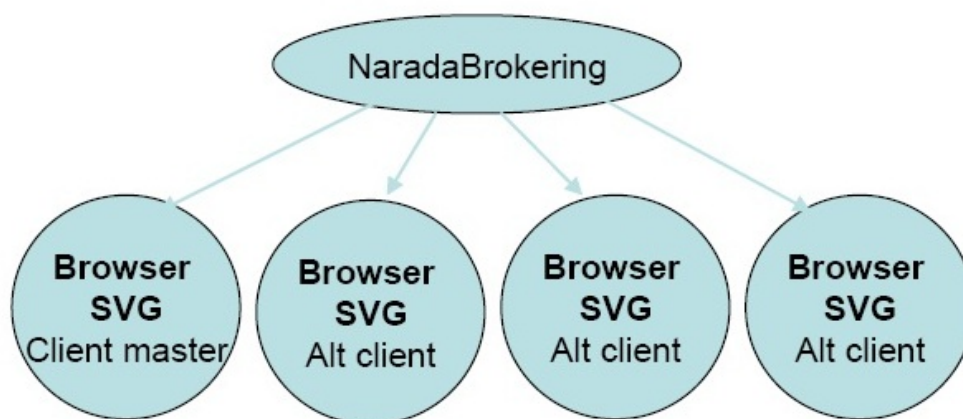
Figura 6: Portul de intrare pentru SVG colaborativ



infrastructura de mesaje, pentru a suporta fiabilitatea, calitatea serviciului și funcționalitatea. Workflow-ul de evenimente al prezentării aplicației poate fi ilustrat prin propagarea să dealungul unui canal cu etape constând din obiecte (componente de sistem). După cum se poate vedea în figura 1, U-turn drumul pentru browser-ul Batik SVG începe de la interacțiunea utilizatorului care produce un eveniment de mouse și se termină la împlinirea redării modificărilor în buffer-ul imagine. Fiecare etapă formează baza sincronizării normale pentru colaborare. într-un model de SVG Web Service portul de intrare și portul de ieșire se referă la interfețele dintre view și portul de protecție al utilizatorului de Web Service la capătul de intrare și capătul de ieșire/redare al canalului.

- Colaborarea bazată pe evenimente poate fi implementată într-un mod bazat pe metode, cum ar fi acelea construite deasupra RPC ca un sistem (CORBA). Oricum, mai repede se adoptă o altă abordare a modelului de Web service bazat pe mesaje conduse de evenimente cu detalii de platforme de bază ascunse în implementarea nivelului infrastructurii de mesaje. Comunicarea între componentele distribuite este condusă indirect prin intermediul brokerilor de mesaje.
- Schemele Publică/Subscrie prezintă capacitatea de a manipula topologii complexe cu multiple subiecte de discuție și clienți mulți. Infrastructura de mesaje oferă un serviciu de management de subiecte de discuție și

Figura 7: Colaborarea monolitică



**Programe identice primesc evenimente identice**

serviciul de înregistrare (pentru Publică/Subscrie) astfel încât sistemul colaborativ poate găzdui activități ale comunităților virtuale într-un stil dinamic și paralel.

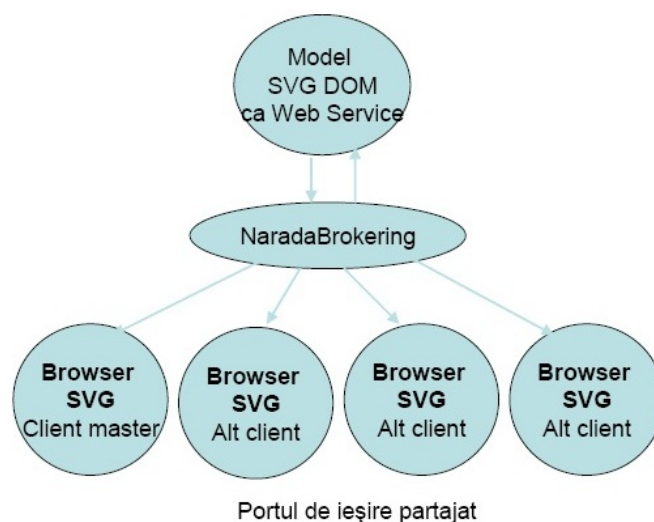
- Realizat deasupra mediului colaborativ, oricine poate dezvolta aplicații SVG ale modelelor de programare cu instructori SIMD și participanți (MIMD) cu Java și JavaScript. Se așteaptă ca această abordare să poată fi aplicată asupra altor tipuri de aplicații și limbaje de programare și există în laborator alte inițiative pe OpenOffice și PowerPoint.

Există două moduri de construire a unui sistem colaborativ bazat pe evenimente: monolitic și Web service. Colaborarea monolitică (figura 9) se obține când toate componentele participante sunt formate ca replici ale unei aplicații existente fără a fi explicit finalizate într-un model separat sau componentă view așa cum cere o arhitectură Web service.

Această abordare funcționează prin interceptarea evenimentelor pe aplicația master și permite broker-ului de mesaje să le transmită la grupul de clienți din modelul de colaborare. Este o strategie obișnuită pentru sistemele colaborative realizate deasupra API-urilor vânzătorului cu eveniment expus fie cu implementări brevetate fie cu implementări open source. A fost demonstrat acest lucru în laborator cu PowerPoint și OpenOffice.

S-au prezentat deja rezultate preliminare pe un browser SVG colaborativ și un joc de șah multi-jucători JavaScript ca un prototip pentru a explora o abordare generală a serviciilor Web colaborative. Au separat apoi SVG într-un model și componente view și au convertit aplicația SVG desktop într-un

Figura 8: Web Service colaborativ de tip SIMD



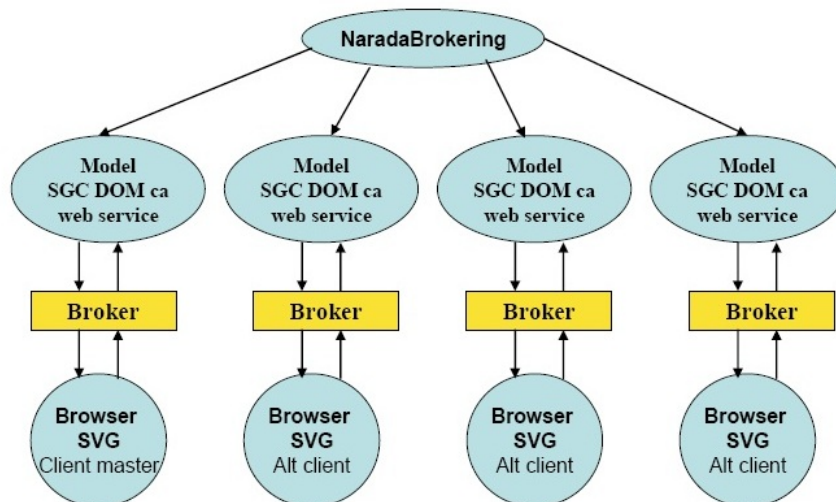
sistem distribuit. Web service-ul colaborativ de tip SIMD este prezentat în figura 8.

Autorii au explicat cum se pot realiza aplicații de rețea bazate pe mesaje colaborative în două moduri: portul de intrare partajat și portul de ieșire partajat. În fiecare caz cineva preia mesajele și le transmite grupului fie acelea care ajung la portul de intrare sau acelea produse de portul de ieșire. Trebuie observat că în fiecare caz un client asignat cu master a preluat rolul de master. Cererile pentru schimbări de roluri (de exemplu master versus non-master, jucător versus observator) pot fi realizate în mod dinamic. Web service-ul colaborativ de tip MIMD este prezentat în figura 9.

Utilizarea serviciilor de mesaje oferite de NaradaBrokering cuprinde două cazuri: unul este înregistrarea pentru serviciul Publică/Subscrie în monolitic, SIMD și MIMD; celălalt este comunicarea între model și view separate în fiecare componentă de aplicație. În cel din urmă caz, Broker-ii din figura 6 rulează în modul punct la punct, NaradaBrokering furnizând servicii de transport cum ar fi: securitatea, firewall and NAT, alegeri de protocol și compresie.

Arhitectura colaborativă unificată care suportă o varietate de aplicații SVG se poate aplica la o tablă albă care suportă SVG și interoperabilitatea applet-uri Java cu o arhitectură de Web service comună. Ea este integrată cu instrumentele NaradaBrokering existente (vizualizare partajată și conferințe audio/video) pentru a oferi un sistem educațional interactiv. Whiteboard ilustrează abilitatea de a realiza aplicații în arhitectura pusă la dispoziție de

Figura 9:



NaradaBrokering. Cele două view-uri (Java și JavaScript) cu un Web service reprezintă suportul pentru clienți.

Modelul de colaborare SIMD poate fi folosit pentru cursuri în educația la distanță, iar modelul de colaborare MIMD poate suporta învățare participativă. Educația necesită diferite moduri de interacțiune plecând de la lecturare pasivă la interactivitate și colaborare de nivel înalt în învățarea participativă (cum ar fi proiecte cuplate). Whiteboard-ul reprezintă un exemplu bun pentru învățarea bazată pe proiecte interactive, pentru că permite mai multor persoane să participe împreună interactiv. Proiectele de modelare cuplate au aceeași structură ca și whiteboard-ul deși folosește instrumente detaliate diferit.

Este propus un model modular universal cu un model de servicii legat prin mesaje care unifică avantajele aplicațiilor desktop, aplicațiilor Web și colaborării Internet. Această abordare permite o maximă reutilizare a componentelor existente; utilizarea unei scheme de mesaje flexibile cu scalabilitate ridicată; colaborare automată și eficientă cu interactivitate de conținut bogat Web pentru clienți diverși peste medii de rețea eterogene; o interfață uniformă pentru următoarea generație de client Web cu accesibilitate omniprezentă. Aplicată educației, această arhitectură permite noi instrumente educaționale participative și un mediu educațional la distanță bogat.

### 2.3.3 O arhitectură pentru sisteme e-learning cu laboratoare on-line

Laboratorul bazat pe Internet (laborator on-line) reprezintă un domeniu de cercetare în continuă creștere în universități. În unele cazuri mai apare și sub denumirea de laborator virtual, laborator la distanță, laborator Internet sau laborator Web. În sistemele de e-learning existente echipamentele hardware nu sunt suportate de infrastructura e-learning, deși IMS Learning Design Best Practice și Ghidul de Implementare (IMS, ver 1.0, 2003) prezintă câteva exemple de laboratoare virtuale. Aceste exemple au fost alese să valideze modelul conceptual al sistemului de învățare. Oricum, nu oferă detalii de proiectare și implementare pentru platforme de învățare bazate pe laboratoare. Obiectivul principal al unei asemenea arhitecturi este de a combina cercetările în domeniul laboratoarelor on-line și infrastructura e-learning pentru a îndeplini obiectivul de creștere a oportunităților și experiențelor de învățare pentru studenți.

*Arhitectura propusă pentru sistemul e-learning cu laboratoare on-line*  
Arhitectura propusă (figura 23) este bazată pe specificația SCORM pentru un sistem elearning cu module suplimentare (Apparatus LMS) și are la bază funcționalitățile pentru sistemele de învățare bazate pe hardware. Specificația SCORM de la ADL se ocupă cu lansarea, comunicarea și urmărirea traseului conținutului dintre resursele de învățare și sistemul de management de învățare. Ea oferă mijloace prin care resursele de învățare pot fi reutilizate și interoperabile între sisteme LMS/LCMS multiple. Specificația constă din următoarele trei componente:

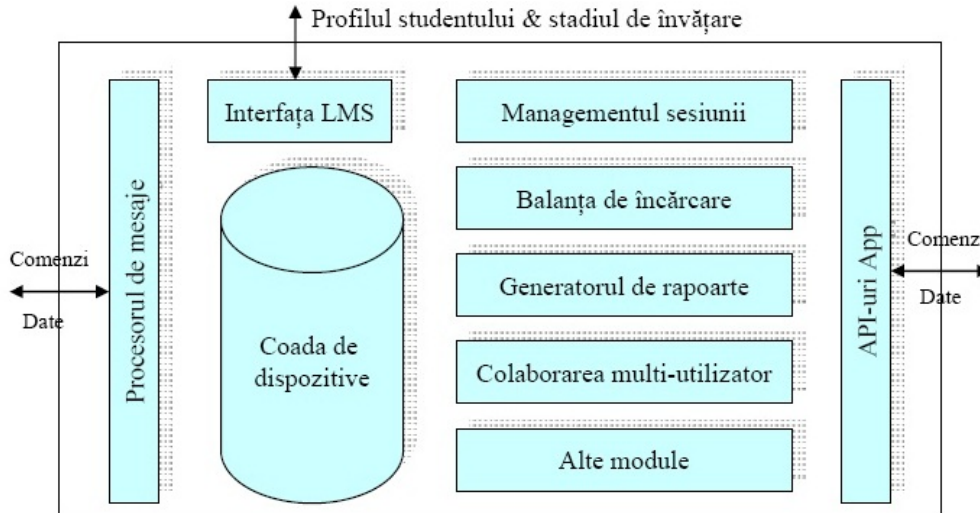
- Resurse de învățare. Reprezintă produse (pagina Web, JavaScript, document XML, obiect Flash, figură etc.) și obiecte de conținut partajabil (o colecție de unul sau mai multe produse).
- LMSAPIs - mecanism de comunicație între LMS și SCO (Sharable Content Object). Acestea sunt folosite pentru colectarea datelor legate de procesul de învățare.
- Mediul de execuție SCROM - un sistem de management de învățare care realizează un management al studenților și a evenimentelor e-learning pentru a colaționa date despre progresul cursantului.

#### *Module Apparatus LMS*

Pentru a extinde LMS la cursuri bazate pe laborator, dispozitivul LMS propus adaugă următoarele module:

- Interfața utilizator virtuală - dispozitiv. App-VUI pe scurt, reprezintă un tablou de control de la distanță pentru student pentru a controla

Figura 10: Arhitectura pentru sistemul e-learning cu laboratoare on-line



dispozitivul real și observa starea experimentului pe parcursul sesiunii de laborator.

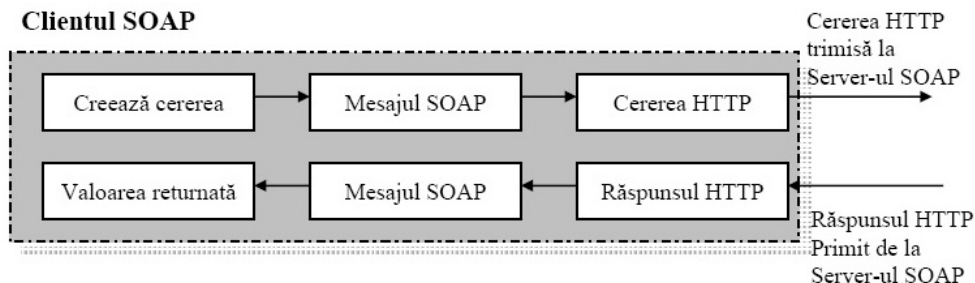
- Mediul de execuție - dispozitiv. App Run Time pe scurt, reprezintă standardul și un mediu LMS de dispozitiv uniform, care oferă servicii de laborator pentru studenți
- APIs dispozitiv. AppAPI pe scurt, reprezintă mecanismul de comunicare pentru Mediul de execuție App și dispozitivul de schimb de date. Ele oferă toate funcțiile necesare pentru inițierea unei conexiuni la dispozitivul actual și pentru livrarea datelor și comenzilor de control.
- Dispozitivul pentru Web. Echipament hardware real, care constă atât din hardware cât și din software-ul care controlează hardware-ul.

Cel mai important element al arhitecturii este mediul de execuție App. Acesta oferă multe funcții standard, cum ar fi managementul sesiunii de laborator, echilibrarea încărcării, generatorul de rapoarte și subsistemul de colaborare multi-utilizator. Aceste funcții sunt reutilizabile și partajate de toate obiectele educaționale, care sunt experimente în scenariul elearning bazat pe laborator.

Pentru a realiza un experiment de pe internet, studenții trebuie să lucreze într-un mediu interactiv. App-VUI (Apparatus Virtual User Interface) oferă un mediu în timp real cu mulți studenți care observă și controlează experimentul. App-VUI permite studentului să emită comenzi de control prin



Figura 11: Diagrama mesajului SOAP



intermediul interfeței utilizator. În același timp, feedback-urile experimentului cum ar fi răspunsul și starea dispozitivului sunt toate afișate în aria de vizualizare App-VUI. Pentru a porni o sesiune de laborator, App-VUI creează o conexiune HTTP cu mediul de execuție App folosind protocolul SOAP (Simple Object Access protocol) (W3C SOAP). În momentul în care conexiunea este autorizată, studentul începe să realizeze experimentul.

#### 2.3.4 Comunicarea utilizând SOAP

SOAP este un protocol simplu pentru schimbul de informație într-un mediu descentralizat. El definește un protocol de comunicare de bază, oferind posibilitatea clienților să schimbe mesaje XML cu serverul. Figura 11 arată diagrama serviciului de mesaje SOAP legat cu HTTP.

În sistemul propus, după ce conexiunea este stabilită, studentul emite comenzi de control, cum ar fi: schimbarea parametrilor, citirea stării dispozitivului și descărcarea datelor de experiment. În cele mai multe cazuri, interacțiunea dintre student și LMS este un schimb de mesaje în două etape (cerere-răspuns). Studentul emite un mesaj de cerere și Mediul de execuție App îi trimite un mesaj de răspuns. Mesajul de răspuns întoarce starea întrebării inițiale (succes sau eșec).

Informațiile schimbate între utilizator și mediul de execuție App este bazat pe mesaje XML. Aceasta va intensifica lizibilitatea atât pentru om cât și pentru mașină. În același timp, este extensibil pe deplin pentru experiențe diferite în LMS diferite.

#### 2.3.5 Serviciul de mesaje bazat pe XML

Utilizând serviciul de mesaje bazat pe XML, mediul de execuție App va fi interoperabil cu orice echipament de laborator bazat pe XML. Astfel, experimentul de laborator poate fi integrat în orice LMS într-un mod plug-and-play

ușor. Pentru a fi siguri că serverul LMS a primit mesajul, va fi trimisă o confirmare înapoi la client. Într-un astfel de caz, sunt propuse o serie de coduri de stări de confirmare.

#### *Mediul de execuție App*

Mediul de execuție App este cheia obținerii obiectivului e-learning care are la bază cursuri bazate pe laborator. Figura 25 descrie modelul propus. Există trei tipuri de interfețe pentru mediul de execuție App în procesul de comunicare cu alte module:

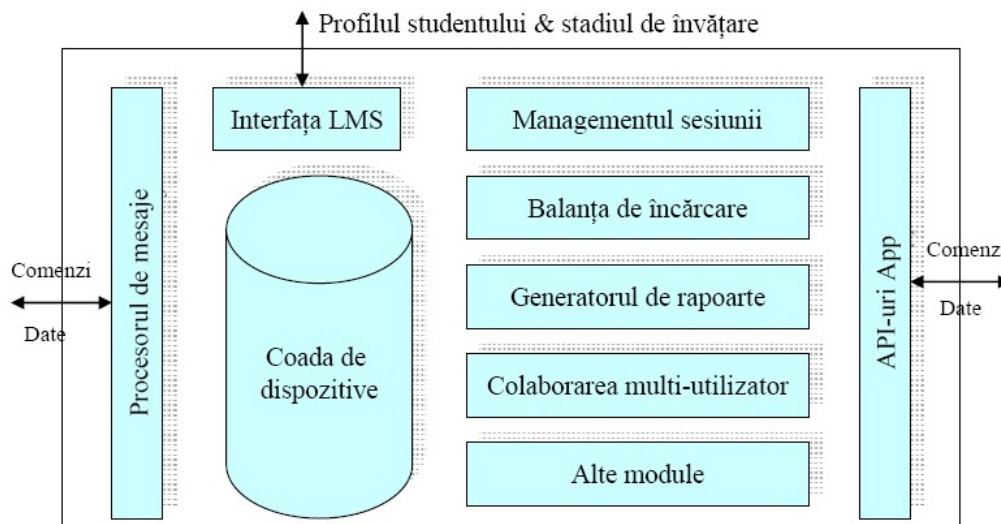
- Procesorul de mesaje responsabil pentru interacțiunea cu studentul. Funcțiile sale includ primirea comenzilor de control și oferirea unui feedback asupra datelor și stării experimentului, la student.
- AppAPIs oferă un canal de comunicare pentru mediul de execuție App și dispozitiv pentru schimbul de date. Aceste API-uri oferă toate funcțiile necesare pentru a iniția o conexiune la dispozitivul actual și livrarea datelor și comenzilor.
- Interfața LMS folosită la schimbarea informației despre student și progresul realizat de el

Mediul de execuție App poate fi considerat un container (sau mediu) pentru LLO (Laboratory Learning Objects). Fiecare LLO care este integrat în mediu poate partaja resursele oferite de mediu. AppAPIs definit în mediul de execuție ADL SCORM, reprezintă un mecanism obișnuit pentru resursele educaționale. Oferă comunicare cu un LMS și un limbaj predefinit sau o formă de vocabular bazat pe un scenariu de comunicare, care este Cursant <-> LMS. Oricum, cursurile bazate pe laborator au încă un tip de comunicare: LMS <-> Dispozitiv. Este o comunicare duală în acest caz.

Pentru a proiecta un experiment bazat pe laborator interoperabil și reutilizabil de-a lungul a mai multe LMS, trebuie să existe o cale comună de a porni, opri și controla dispozitivele. În mod obișnuit, LMS oferă interfața Web necesară unui student în selectarea experimentului online. Apoi, mediul se va conecta la dispozitivul pe care studentul l-a selectat prin intermediul AppAPI. Odată ce mediul de execuție App s-a conectat la dispozitiv, studentul poate emite comenzi de control și primi datele de experiment prin intermediul socket-ului de comunicație stabilit. Aceste AppAPI-uri îndeplinesc cerințele de reutilizare și interoperabilitate. Ele oferă o metodă standardizată pentru un LMS pentru a comunica cu experimente de laborator reale.

Coadă de dispozitive. ținând cont de managementul unui laborator real, toți studenții care sunt înscriși la curs au acces la resursele laboratorului pe o perioadă stabilită de timp. În acest scop sunt redefinite următoarele două principii importante:

Figura 12: Mediul de execuție App



- pentru o sesiune de laborator reală, doar unui utilizator îi este permis să realizeze un experiment particular la un moment dat.
- orice utilizator poate conduce un experiment particular într-o perioadă specifică de timp. Pe baza acestor două principii, s-a propus un sistem de coadă de așteptare la dispozitive de tipul primul venit primul servit, cu timp maxim stabilit de așteptare pentru a asigura lucrul în mediul de execuție App.

Orice utilizator care dorește să realizeze un experiment particular trebuie să se înscrie la coada de așteptare la dispozitive. Pentru identificarea sistemului, va fi generat un unic UID pentru fiecare utilizator. După ce primul utilizator care conduce experimentul a terminat sau a depășit timpul alocat, următorul utilizator din coadă poate porni experimentul. Proiectarea garantează că numai un utilizator poate accesa dispozitivul la un anumit timp. În același timp, un utilizator neautorizat nu poate avea acces la dispozitiv în absența unui UID.

**Managementul sesiunii.** În cele mai multe cazuri, tehnicianul laboratorului este responsabil pentru îndrumarea studentului în realizarea unui experiment și în colectarea rezultatelor experimentului. În mediul de execuție App managerul sesiunii realizează același rol. Uneori, el oferă asistență sincronizată (de mesaje chat instantane sau telefoane on-line) pentru student atunci când un experiment important este în desfășurare și când are o problemă. Pe de altă parte, asistența asincronă (prin email sau BBS) este inclusă

în unele situații neimportante. Pentru a asigura o disponibilitate mărită la cursurile bazate pe laborator la cât mai mulți studenți, modulul de management al sesiunii este responsabil pentru monitorizarea sesiunilor de laborator și realizează următoarele:

- terminarea experimentului odată cu expirarea timpului;
- resetarea dispozitivului pentru un nou utilizator;
- informarea managerului de laborator în cazul eșecului unui echipament hardware.

Echilibrarea încărcării. În lumea reală, resursele de laborator cum ar fi programul de laborator, orele deschiderii, echipamentul, asistenții laboratorului sunt întotdeauna limitate. Resursele laboratorului pot fi accesate acum 24 de ore pe zi, 7 zile pe săptămână prin infrastructura e-learning, dar echipamentul de laborator este încă limitat. Prin urmare este necesară o formă de echilibru. Scopul acesteia este creșterea utilizării și intensificarea disponibilității dispozitivului. Pe de altă parte, o disponibilitate mai mare poate fi definită ca redundantă. În acest caz, dispozitive de laborator care au aceeași funcții și I/O trebuie să fie dezvoltate și să devină disponibile mediului de execuție App. Astfel, când mai multe cereri vin către un același experiment, sistemul poate redirecționa cererea către un alt dispozitiv conform unei reguli prestabilite.

Generatorul de rapoarte. Odată ce studentul finalizează experimentul, el trebuie să descarce rezultatele experimentului, datele referitoare la comenzile de control pentru analize sau rapoarte ulterioare. În general, aceste date sunt toate bazate pe ASCII. Pentru a intensifica lizabilitatea pentru om și mașină, generatorul de rapoarte propus procesează aceste date conform cu schema XML.

Colaborarea multi-utilizator. Când conduc experimente on-line, studenții se pot simți izolați de profesor și colegii de clasă, în special când aceștia au întrebări legate de experiment sau au probleme în continuarea experimentului. Astfel, mediul de execuție App trebuie să conțină un mecanism pentru colaborare la distanță care să permită studenților să conducă experimente în colaborare. Beneficiile obținute de studenți atunci când lucrează în colaborare într-un mediu la distanță este dezvoltarea unei echipe de lucru și intensificarea experienței dobândite. Mediul de colaborare este obținut prin includerea de facilități de conferință, forum-uri și servicii de chat în sistem. În momentul în care instructorul inițiază o sesiune de laborator, mai mulți studenți se pot alătura sesiunii. Ei pot monitoriza experimentul de laborator prin primirea unui flux video de la o cameră care filmează dispozitivul actual.

La sfârșitul experimentului toți studenții din aceeași sesiune de laborator pot descărca rezultatele experimentului.

### 2.3.6 *Dispozitiv pentru Web*

Abordările curente legate de proiectarea cursurilor de laboratoare on-line sunt împărțite în două mari grupuri: folosirea simulatorilor educaționali și accesul de la distanță la echipamentul unui laborator real. În primul caz, simulările software sunt folosite pentru a simula comportamentul dispozitivului hardware real. În cel de al doilea caz dispozitivul hardware real este folosit pentru a permite studentului să interacționeze cu echipamentul. Mediul e-learning propus poate lucra cu ambele grupe. Experimentele de laborator sunt tratate ca o cutie neagră în ambele cazuri. Aceste cutii negre doar oferă interfața necesară comunicării cu sistemul elearning. Toate datele (starea dispozitivului, feedback-ul experimentului) sunt transferate prin intermediul interfeței.

Un dispozitiv care oferă Web este un hardware real cu un controller (dispozitiv electric, echipament medical, experiment optic) care poate fi conectat la rețea prin protocolul TCP/IP. Controller-ul poate fi orice formă de dispozitiv bazat pe PC cu un modul de comunicare care permite stabilirea unei conexiuni la rețea. Pentru partea de software, el execută controlul local al hardware-ului și colectează datele experimentului. Privit ca un întreg, controller-ul trebuie să aibă interfață fizică, cum ar fi convertorul DA/AD sau card-ul de achiziție date pentru a realiza legătura cu echipamentul actual.

### 2.3.7 *Arhitectura unui sistem de e-learning utilizat în instituțiile de învățământ superior din România*

În contextul cerințelor deosebite impuse de beneficiari, două dintre deciziile majore în procesul de proiectare au fost legate de alegerea sistemului de operare pentru nucleul GESCO precum și a motorului de gestiune a bazelor de date. Experiența specialiștilor GeniSoft dublată de calitățile incontestabile în ceea ce privește securitatea, stabilitatea și flexibilitatea, au făcut din sistemului de operare Windows 2000 Server și sistemul de gestiune a bazelor de date SQL Server 2000 alegeri naturale în cadrul acestor decizii. În plus, unul din factorii decisivi în alegerea SQL Server 2000 a fost capacitatea deosebită a acestui sistem de a se acomoda eficient la volume de date cu dimensiuni variabile. Având în vedere specificul mediului în care GESCO urma să opereze, era absolută nevoie de capacitate de adaptare și performanță atât pentru volumele de date generate de facultăți cu sute de studenți cât și pentru cele generate de universități cu zeci de mii de studenți.

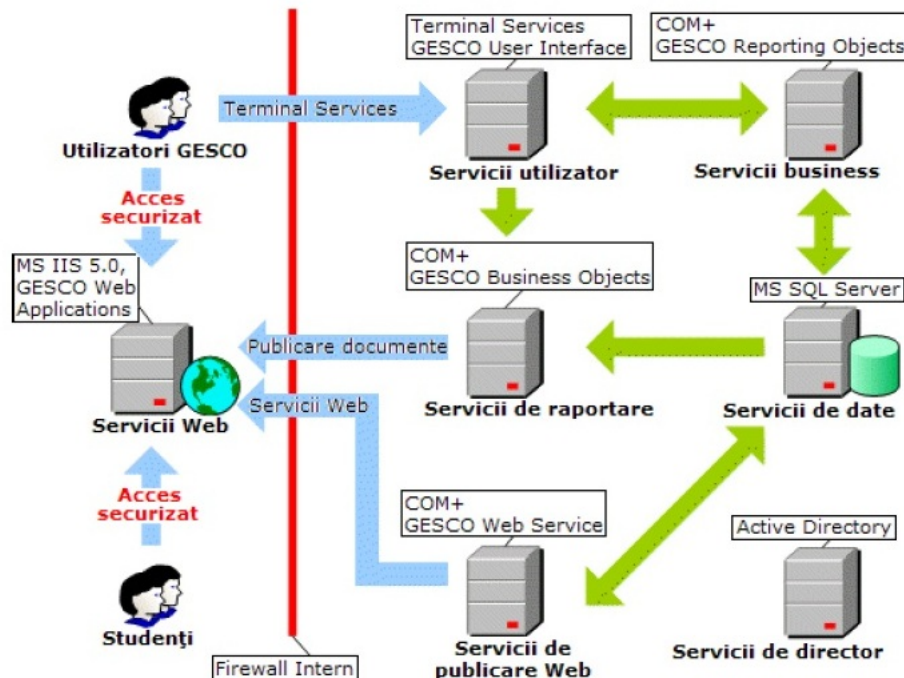
Accesul la sistem trebuia asigurat pentru două categorii majore de utilizatori: persoane implicate în gestiunea școlarității (rectori, prorectori, decani, secretari șefi, alte persoane cu capacitate de decizie, secretare etc.) și studenți. ținând cont de infrastructura TIC neomogenă a beneficiarilor și de calitatea pe alocuri mai slabă a rețelelor, accesul prin Terminal Services s-a dovedit a fi soluția ideală pentru prima categorie de utilizatori. în ceea ce privește accesul studenților, necesitatea de consultare a datelor personale în orice moment și din orice loc, a făcut dintr-un portal Web singura soluție posibilă.

Dintre tehnologiile disponibile pe platforma Windows 2000 Server, COM+ a fost aleasă ca soluție strategică pentru dezvoltarea nivelului de business logic. Această opțiune permite în prezent scalarea sistemului GESCO într-un mod simplu și eficient, în funcție de factori cum sunt încărcarea pe servere, volumul de date procesat, numărul de utilizatori concurenți și altele. în ceea ce privește mediul de dezvoltare, s-a optat inițial pentru Visual Studio 6.0 Enterprise Edition. Lansarea în anul 2002 a platformei .NET și a mediului Visual Studio .NET a marcat un punct de cotitură în procesul de dezvoltare a sistemului GESCO. Excelentul suport pentru aplicații și servicii Web precum și integrarea foarte bună a componentelor EnterpriseServices cu tehnologia COM+ au stat la baza deciziei de a furniza versiunea 2.0 pe platforma .NET. în prezent, modulele de publicare și acces Web sunt în totalitate bazate pe ASP.NET, servicii Web și XML, iar o parte semnificativă a nucleului sistemului operează pe platforma .NET. Este de asemenea de remarcat faptul că flexibilitatea deosebită a limbajului C a permis utilizarea acestuia ca unic limbaj de programare pentru toate modulele sistemului. *Arhitectura sistemului* Varianta standard a nucleului GESCO are la bază o arhitectură care furnizează șase servicii fundamentale:

- servicii de director pentru autentificare, autorizare și securizarea resurselor;
- servicii de stocare și gestiune a datelor;
- servicii de acces la aplicație și de interfață utilizator;
- servicii de business logic pentru manipularea datelor;
- servicii de generare de rapoarte;
- servicii de publicare a datelor pe Web.

Nucleul sistemului este protejat de un firewall intern care îl separă de un server Web extern care asigură suportul pentru accesarea datelor publicate de nucleu. Prin intermediul firewall-ului, accesul la nucleu este restrâns la

Figura 13: Arhitectura standard a sistemului GESCO



un număr foarte mic de porturi IP, ceea ce permite securizarea și monitorizarea eficientă a accesului. În general, cei mai mulți beneficiari ai sistemului GESCO optează pentru un al doilea firewall extern, care oferă protecție suplimentară pentru serverul Web (figura 29). Derivată direct din flexibilitatea tehnologiilor și platformelor utilizate (în special COM+ și .NET Framework), flexibilitatea GESCO asigură capacitatea de adaptare la mediul universitar din România. În funcție de specificul beneficiarului, serviciile nucleului GESCO pot fi grupate fie pe un singur server (în cazul unui volum restrâns de date), fie pe oricâte servere este necesar. Scalarea nucleului se face exclusiv pe cale administrativă fără a fi necesară intervenția în cod. Acest avantaj permite nucleului GESCO să se adapteze rapid la orice fel de schimbări survenite în mediul în care operează.

Serviciile de director sunt asigurate de Active Directory. Gestiunea utilizatorilor, autentificarea și autorizarea acestora, precum și securizarea accesului la resursele nucleului se realizează cu ajutorul Active Directory, într-un mod simplu și eficient.

Serviciile de stocare și gestiune a datelor au la bază SQL Server 2000. Arhitectura ADO.NET asigură suport pentru accesarea rapidă și într-o manieră simplă a datelor stocate, contribuind în același timp și la creșterea nivelului de securitate. Mai mult, serviciile oferite de SQL Server 2000 permit

adoptarea unor soluții ingenioase pentru rezolvarea problemelor legate de performanță. Un exemplu concludent este cel de la Facultatea de Economie și Administrare a Afacerilor din cadrul Universității A. I. Cuza Iași, unde se utilizează un sistem suplimentar SQL Server pentru generarea rapoartelor. Acest server suplimentar realizează o replicare a datelor specifice facultății de pe serverul principal, asigurând suportul de date pentru generarea rapoartelor complexe ce implică procesare intensivă.

Serviciile de acces la aplicație asigură accesul utilizatorilor la interfața utilizator a nucleului GESCO. Tehnologia Terminal Services face ca accesul să nu fie influențat în mod negativ de calitatea liniilor de comunicație. Aceeași tehnologie permite echipei de specialiști de la GeniSoft să intervină rapid direct pe serverele beneficiarului pentru rezolvarea problemelor ce apar în exploatarea sistemului. În acest mod, GESCO beneficiază de un nivel de suport care nu putea fi atins în alte condiții.

Serviciile de business logic beneficiază la rândul lor din plin de facilitățile oferite de mediul de execuție COM+. Dintre aceste facilități, cele mai importante sunt suportul pentru tranzacții, mecanismul de object pooling și nu în ultimul rând flexibilitatea deosebită în ceea ce privește configurarea arhitecturii de aplicații COM+. Cu un efort administrativ minim, o parte din componentele de business logic pot fi oricând mutate pe noi servere, sau chiar regrupate pentru a asigura funcționalitatea sistemului în parametri ceruți în cazul creșterilor semnificative ale volumului de date sau ale numărului de utilizatori concurenți. Integrarea tehnologiei EnterpriseServices de pe platforma .NET cu COM+ a adus un plus de eficiență și o simplificare semnificativă a procesului de proiectare, dezvoltare și implementare a componentelor de business logic. Ceea ce însă a contribuit decisiv la un salt semnificativ al calității componentelor de business logic a fost tehnologia ADO.NET. Noul model de acces la date bazat pe arhitecturi deconectate și XML a adus cu sine eficientizarea modului în care GESCO accesează și procesează datele.

Serviciile de generare de rapoarte au la bază un motor propriu dezvoltat de GeniSoft care permite transformarea datelor în format XML în documente în format Excel și PDF. Acest motor a fost dezvoltat în C cu scopul de a permite adaptarea din mers la cerințele complexe de raportare existente. Una din caracteristicile mediului universitar românesc este existența unei cereri constante de rapoarte, cu variații frecvente în ceea ce privește forma și conținutul. În acest context, având la bază capacitatea SQL Server 2000 de a extrage date direct în format XML precum și orientarea fundamentală spre XML a tehnologiei ADO.NET, s-a luat decizia de a utiliza XML ca suport principal pentru conținutul rapoartelor și Excel respectiv PDF pentru prezentarea rapoartelor. Exportarea datelor inclusiv în format Excel s-a dovedit utilă în special în contextul prelucrărilor suplimentare efectuate la



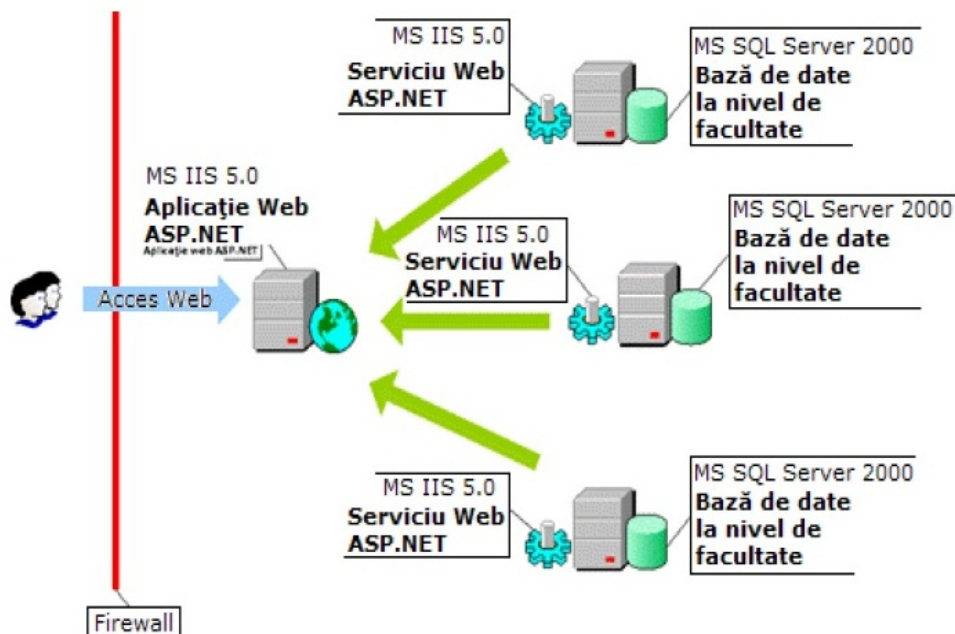
nivel de secretariate de către utilizatori familiarizați cu Microsoft Office.

Serviciile de publicare a datelor se bazează pe tehnologia ASP.NET (în special pe suportul pentru servicii Web) și pe platforma oferită de Internet Information Services 5.0. Publicarea datelor se realizează prin intermediul unui serviciu Web consumat la rândul său de unul sau mai multe site-uri Web. Pe lângă securitatea sporită, un avantaj important este cel dat de flexibilitatea cu care se pot combina diversele modele de implementare GESCO. Concludentă în acest sens este comparația dintre arhitecturile de la Universitatea A. I. Cuza Iași (UAIC) și Universitatea din București (UB). La UAIC, nucleul GESCO este concentrat într-un singur site, pe un număr de aproximativ 10 servere. Portalul Web consumă un unic serviciu Web care asigură publicarea datelor pentru toate facultățile universității. În contrast cu această abordare, la UB există 16 nuclee GESCO (câte unul pentru fiecare facultate a universității). Portalul Web central consumă în acest caz 16 servicii Web, în funcție de facultatea de care aparține utilizatorul care îl accesează. Exemplul ilustrează perfect o parte din avantajele de care beneficiază aplicațiile Web dezvoltate pe platforma .NET. Figura 30 prezintă arhitectura implementată la Universitatea din București. *Beneficiile arhitecturii Securitate*. Una din obligațiile principale asumate de GeniSoft a fost realizarea unui sistem care să pună securitatea datelor în prim plan. Caracterul sensibil al datelor gestionate (note, taxe etc.) face din orice sistem de gestiune a școlarității o potențială țintă pentru atacuri. Îmbinarea diverselor tehnologii și platforme (comunicații criptate prin Terminal Services, servicii de director furnizate de Active Directory, forms authentication la aplicațiile ASP.NET, securitate Windows la SQL Server) au permis obținerea unei soluții care dovedește în prezent satisfacerea tuturor cerințelor complexe legate de securitate.

*Flexibilitate*. Succesul de care se bucură în prezent implementările GESCO la unele din cele mai mari universități din România dovedește caracterul flexibil al acestui sistem. Un argument în plus la această afirmație este cel dat de caracterul eterogen al mediilor TIC în învățământul superior românesc. Prin implementarea GESCO, universitățile au obținut un sistem informatizat de gestiune a școlarității care este capabil nu numai să răspundă la cerințele actuale, ci și să se adapteze la cerințele viitoare. Este evident faptul că o asemenea flexibilitate nu se putea obține decât utilizând tehnologii și platforme flexibile în procesul de dezvoltare

*Extensibilitate*. Ușurința și eficiența în dezvoltarea aplicațiilor pe platforma .NET stau în prezent la baza caracterului extensibil al sistemului GESCO. O altă caracteristică importantă a mediului universitar românesc este schimbarea permanentă. Element pozitiv și foarte important din punct de vedere academic și didactic, această schimbare pune probleme foarte mari celor care își propun să modeleze această dinamică în sisteme informatizate.

Figura 14: Servicii și aplicații Web în arhitectura GESCO



Adoptarea platformei .NET a fost pentru GeniSoft decizia strategică ce a permis creșterea semnificativă a capacității de răspuns la cererile frecvente de extindere a sistemului GESCO.

**Scalabilitate.** Exemplele anterioare în ceea ce privește posibilitățile de acomodare la mediu și la cerințele acestuia definesc caracterul scalabil al sistemului GESCO. La fel ca în cazul flexibilității, caracterul de scalabilitate nu putea fi obținut decât prin utilizarea unor platforme și tehnologii scalabile. Deși scalabilitatea este o trăsătură de bază pentru toate tehnologiile Microsoft, dintre cele utilizate de sistemul GESCO se remarcă în special SQL Server 2000, COM+ și .NET Enterprise Services.

**Performanță și stabilitate.** În prezent, GESCO este implementat la instituții de învățământ superior cu număr de studenți de ordinul zecilor de mii (pentru informații suplimentare se poate consulta site-ul [www.gesco.ro](http://www.gesco.ro)). Serviciile furnizate de SQL Server și ADO.NET au permis îmbunătățirea semnificativă a performanțelor într-un mediu stabil. Lucru remarcabil pentru un mediu tranzacțional, timpii de răspuns la nivelul business logic (și implicit la nivelul interfeței utilizator) rămân reduși chiar și în cazul creșterilor semnificative ale numărului de utilizatori concurenți. GESCO a dovedit acest lucru în vara anului 2002 când Universitatea de Vest Timișoara a decis derularea procesului de înscriere la admitere pe acest sistem. Numărul mare

de candidați la facultățile universității a atras după sine o creștere de peste 4 ori a numărului de utilizatori concurenți, situație în care sistemul nu a înregistrat nici o modificare semnificativă a timpilor de răspuns. Creșterea nivelului de acuratețe a datelor. Introducerea GESCO a permis obținerea unui control eficient asupra tuturor categoriilor de informații specifice activității de gestiune a școlarității: date personale ale studenților; planuri de învățământ; examene, înscrieri la examene, rezultate la examene; burse; taxe, obligații de plată, încasări; cazări; medii semestriale, anuale, traiectorii universitare. Generarea unei palete largi de rapoarte începând cu procese verbale și cataloage și mergând până la documente de studii (situații școlare, foi matricole, diplome) fac din GESCO o soluție completă pentru informatizarea activităților de gestiune a școlarității.

Accesul studenților la datele personale. GESCO este primul și la ora actuală singurul sistem integrat de gestiune a școlarității din România care permite studenților unei universități accesul la datele personale și înscrierea la examene prin intermediul portalului Web MyGESCO. Prin acest mecanism, studenții pot verifica rezultatele la examene, notele și mediile proprii, situația plății taxelor, datele personale. În plus, studenții își pot exprima opțiunile în ceea ce privește înscrierea la examene și participarea la examen la diverse date prestabilite. O altă premieră introdusă de portalul GESCO este posibilitatea studentului de a transmite cereri în format electronic către secretariat, fapt ce ușurează semnificativ munca personalului din cadrul acestuia. Informații pentru managementul universitar. Pe lângă suportul pentru informatizarea activității de gestiune a școlarității, sistemul GESCO pune la dispoziția persoanelor implicate în managementul universitar (rectori, prorectori, decani, prodecani, cadre didactice) o paletă largă de informații consolidate deosebit de utile în luarea deciziilor de management universitar. Eficientizarea activității de gestiune a școlarității. Prin implementarea sistemului GESCO, Universitatea din București beneficiază de avantaje cum ar fi:

- Existența unei baze de date consistente care cuprinde datele complete a aproximativ 18000 de studenți.
- Scurtarea semnificativă a timpilor necesari pentru efectuarea operațiilor de actualizare a informațiilor de școlaritate.
- Scurtarea semnificativă a timpilor necesari pentru generarea documentelor de bază (foi matricole, situații școlare, centralizatoare). De exemplu, măsurătorile efectuate pe sistemul GESCO au demonstrat posibilitatea generării centralizatorului de an la nivelul unei facultăți într-un timp de ordinul zecilor de secunde. Comparat cu timpul necesar completării manuale a documentului respectiv, sporul de eficiență devine evident.

- Scurtarea semnificativă a timpilor necesari pentru calcularea/recalcularea mediilor la nivel de an și semestru.
- Obținerea rapidă de informații statistice din baza de date.
- Îmbunătățirea modului de interacțiune cu studenții la nivel de secretariate, dând posibilitatea studenților de a transmite cereri în format electronic permite o scădere a numărului de vizite la secretariatul facultății, permițând personalului din secretariat să se concentreze pe rezolvarea problemelor de școlaritate.

## 3 Proiectarea și dezvoltarea sistemului de E-learning

### 3.1 Raport de modelare

Un asemenea sistem de e-learning este capabil să urmărească evoluția studentului în procesul de învățare fiind folositor atât profesorilor cât și studenților pentru a își vedea nivelul de cunoștințe. Testele din acest tutorial ajută studentul să înțeleagă mai bine limbajul de programare C++ deoarece se folosește de mesaje sugestive astfel încât studentul să înțeleagă ce a greșit și cum poate corecta în caz că nu reușește tutorialul îl va redirectiona la cursul respectiv sau la un exemplu asemănător de exercițiu.

#### 3.1.1 Modelarea funcțională și dinamică

1. Actori:
  - studentul
  - profesorul
  - administratorul

#### 3.1.2 Diagrama de context static

#### 3.1.3 Diagrama cazurilor de utilizare pentru studenți

Diagrama cazurilor de utilizare prezintă modul de folosire a sistemului de către studenți.

##### Descriere cazuri de utilizare

Titlu: Logare.

Caz de utilizare: Logare

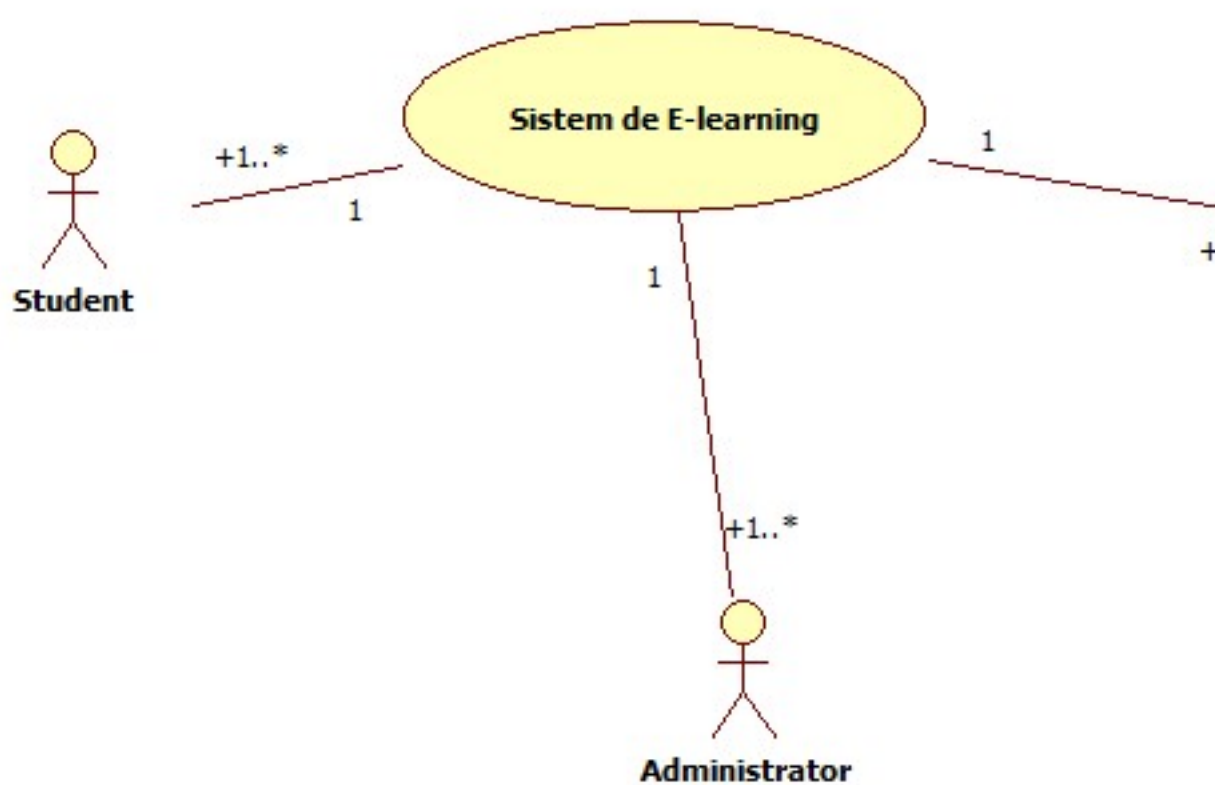
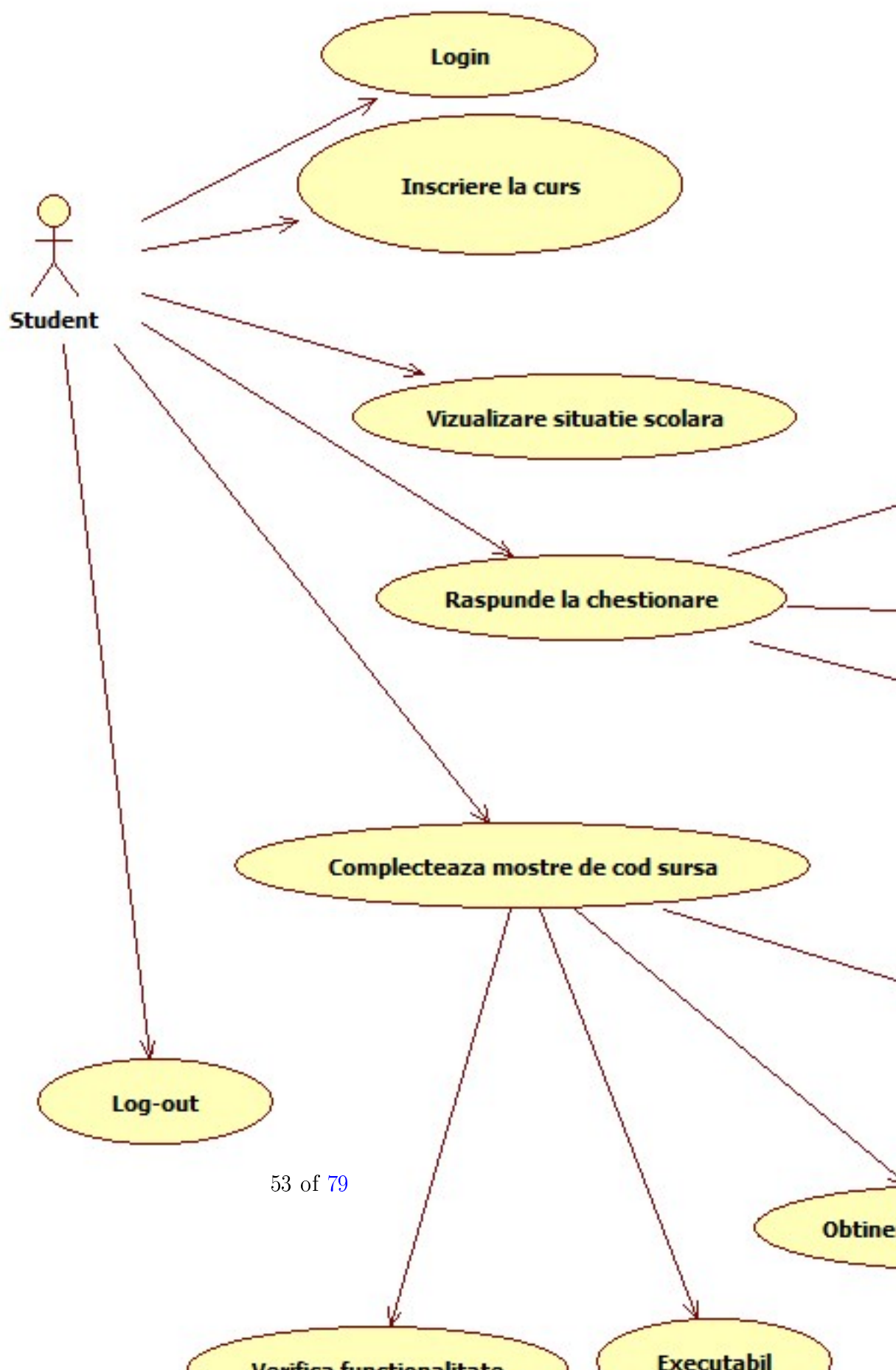


Figura 15: Diagrama de context static



Rezumat: Persoana introduce numele de utilizator și parola iar sistemul verifică existența numelui de utilizator și corectitudinea parolei pentru a deschide contul persoanei.

Actori: studentul, sistemul

**Descrierea scenariului:**

Precondiții:

- persoana se afla în program
- programul trebuie să fie disponibil

Scenariu nominal:

1.Studentul se logează.

Postcondiții:

- Sistemul solicită numele și parola.
- Persoana introduce numele și parola.
- Persoana cere logare.
- Sistemul verifică numele.
- Dacă numele de utilizator este eronat va afișa un mesaj de eroare.
- Sistemul verifică parola.
- Dacă parola este eroantă va afișa un mesaj de eroare și cere reintroducerea parolei.
- Dacă numele și parola sunt valide se deschide contul utilizatorului.

Scenariu alternativ:

A1 Verificarea numelui de utilizator: Dacă numele de utilizator este eronat va afișa un mesaj de eroare și cere reintroducerea numelui de utilizator.

A2: Verificarea parolei: Dacă parola este eronată va afișa un mesaj de eroare și cere reintroducerea parolei.

Scenariu de Eroare:

Persoana nu se poate loga deoarece sistemul nu se poate conecta la baza de date pentru a vedea existența ei în baza de date.

Postcondiții:

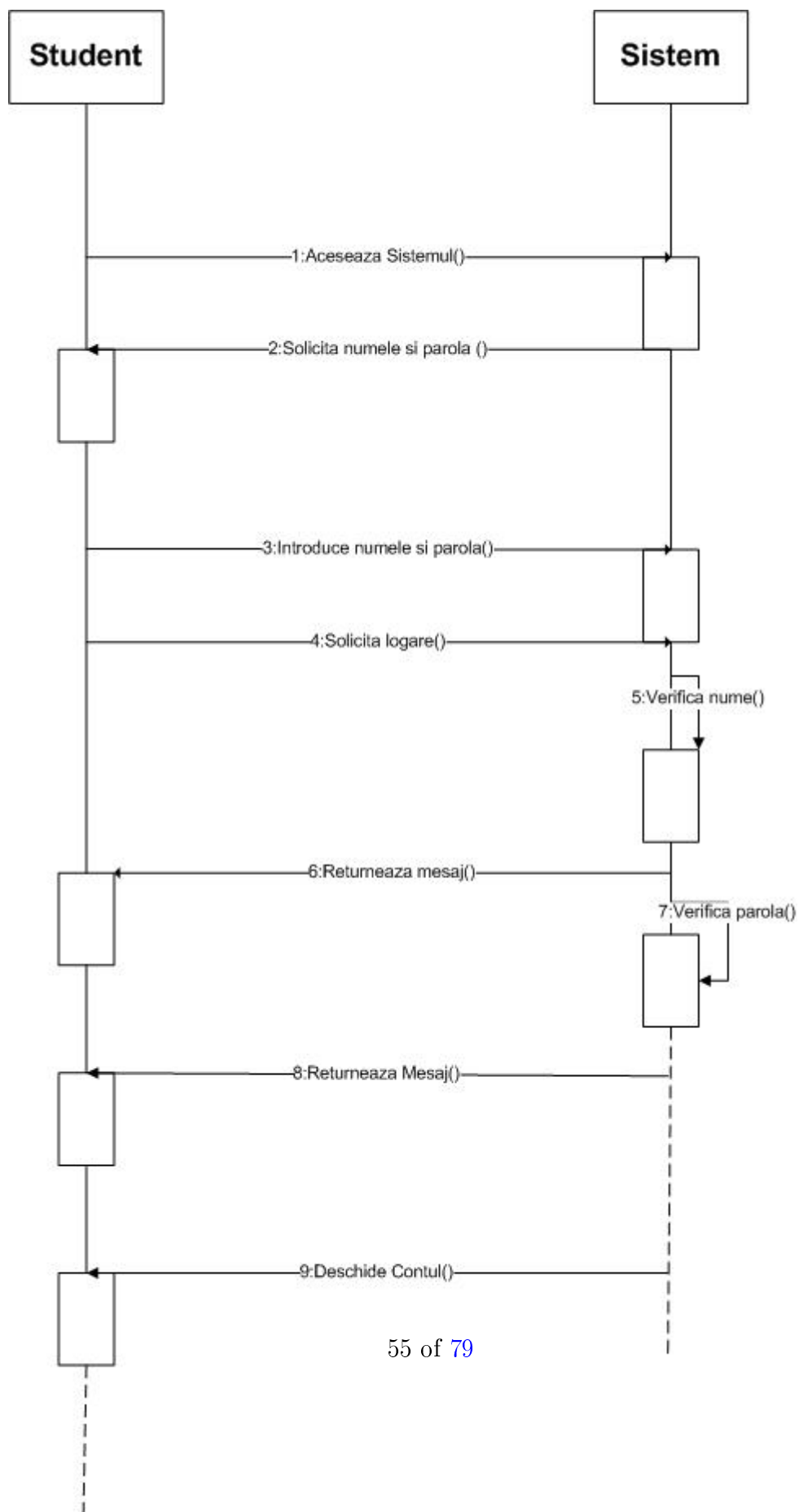
Persoana este conectată la sistem și poate accesa opțiunile valabile pentru tipul său de utilizator.

Caz de utilizare: Alege testul

Titlu: Alege testul.

Rezumat: Studentul alege testul pe care dorește să îl parcurgă

Actori: studentul.





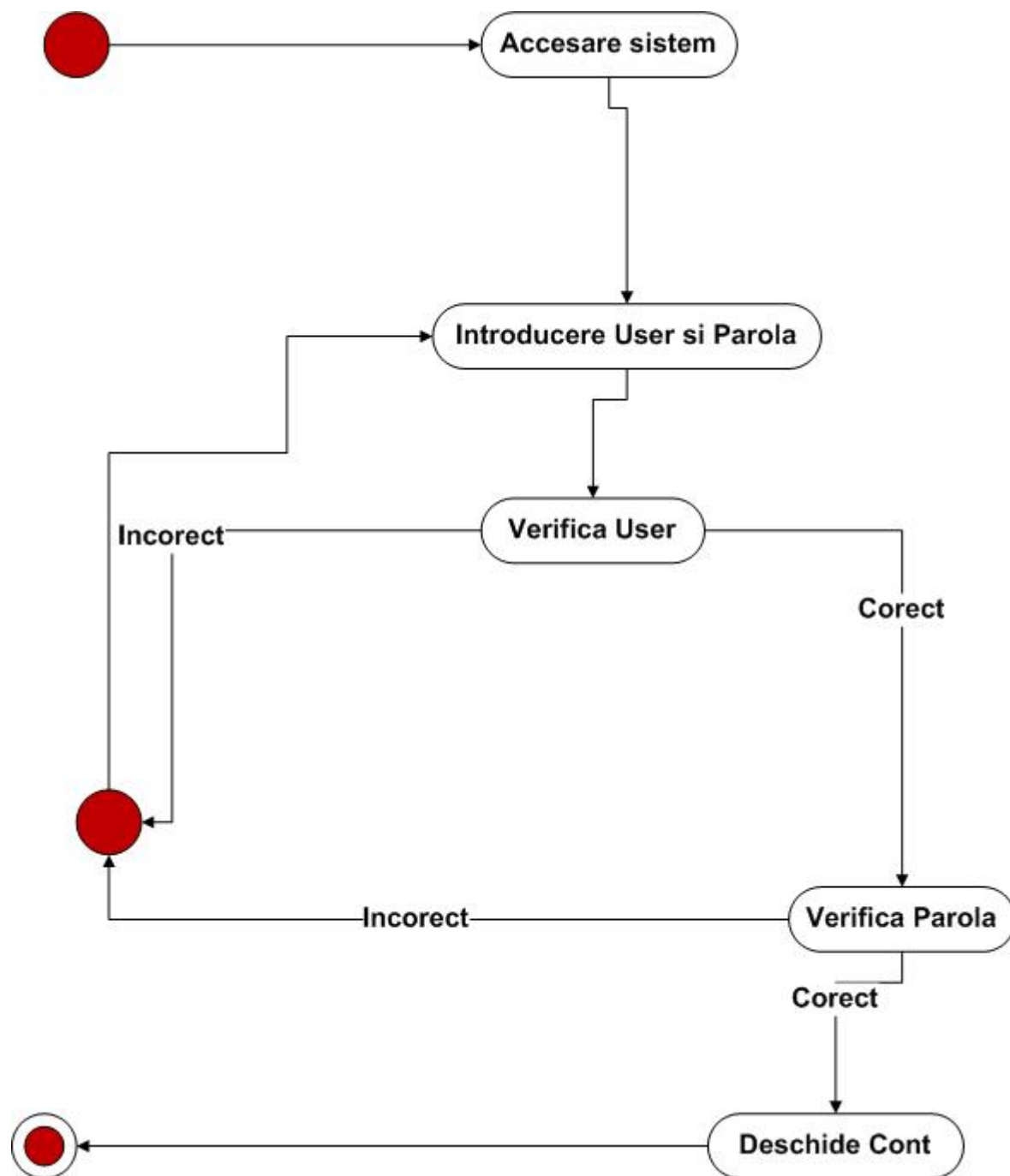


Figura 18: Diagrama de activitate pentru logare

Descrierea scenariului:

Precondiții: - programul trebuie să fie disponibil

Scenariu nominal:

- Studentul se logează.
- Studentul alege testul, și introduce răspunsurile.
- Studentul completează mostre cod sursă.

Scenariu alternativ:

A1. Studentul poate alege vizualizarea situației școlare.

Scenariu de eroare :

Programul nu păstrează informațiile de notare a studentului.

Postcondiții:

Sistemul afișează testul selectat

### 3.1.4 Diagrama cazurilor de utilizare pentru profesor

Diagrama cazurilor de utilizare prezintă modul de folosire a sistemului de către profesor.

#### **Descriere cazului de utilizare pentru profesor**

Titlu: Logare.

Caz de utilizare: Logare

Rezumat: Persoana introduce numele de utilizator și parola iar sistemul verifică existența numelui de utilizator și corectitudinea parolei pentru a deschide contul persoanei.

Actori: profesor, sistemul

Descrierea scenariului:

Precondiții:

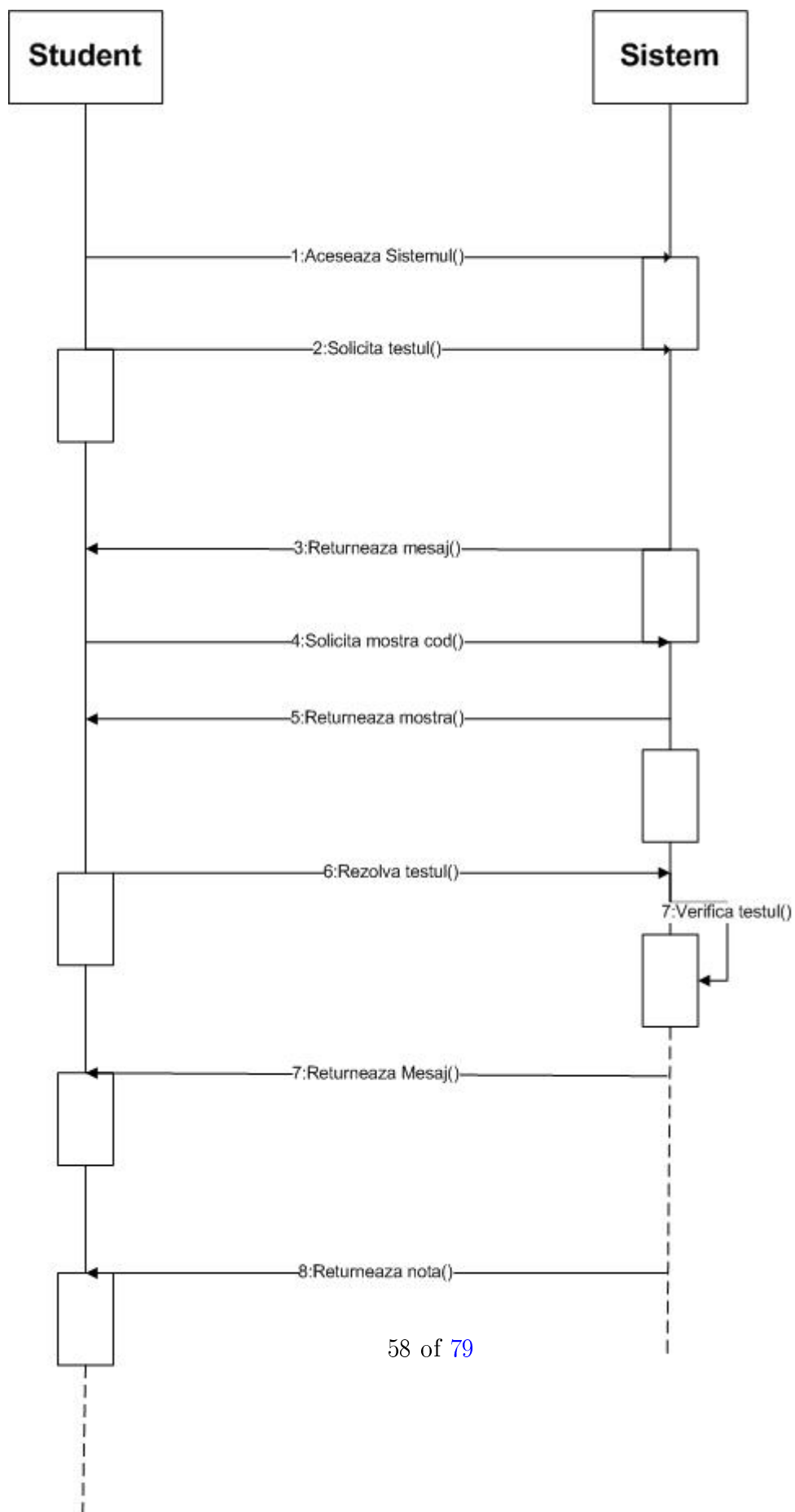
- persoana se află în program
- programul trebuie să fie disponibil

Scenariu nominal:

1. Profesorul se logează.

Postcondiții:

- Sistemul solicită numele și parola.
- Persoana introduce numele și parola.
- Persoana cere logare.
- Sistemul verifică numele.



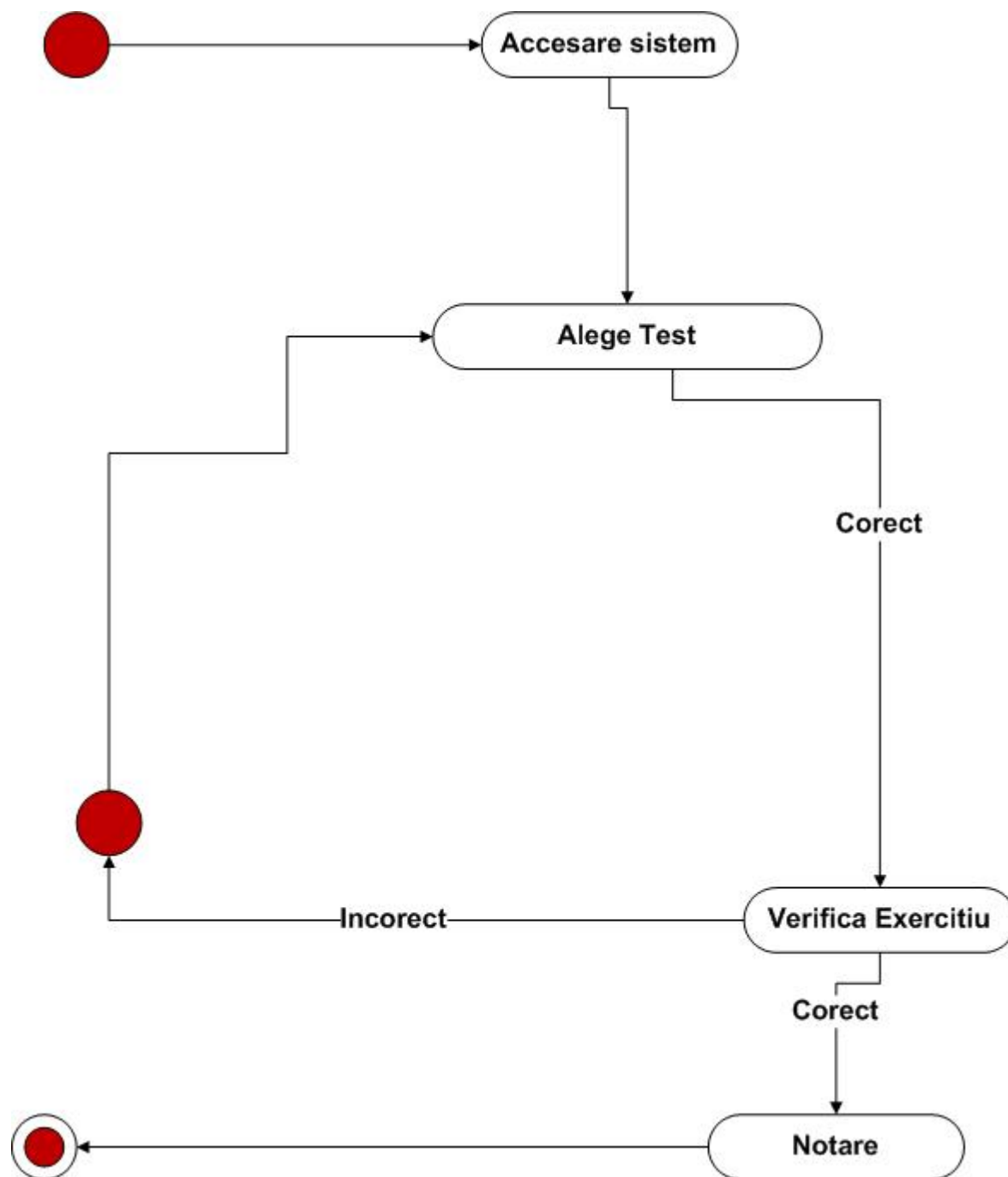


Figura 20: Diagrama de activitate pentru lecție

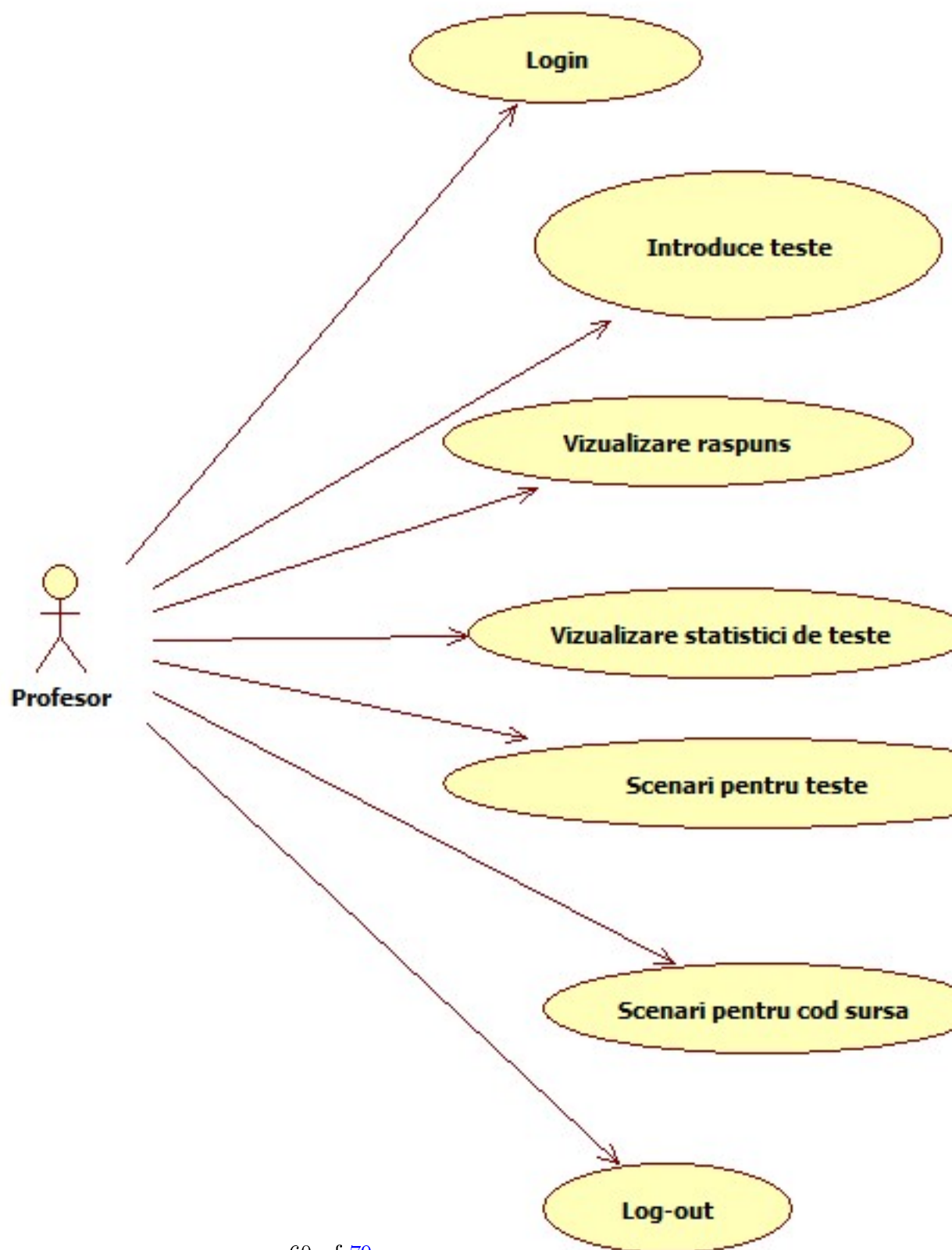


Figura 21: Diagrama cazurilor de utilizare

- Dacă numele de utilizator este eronat va afișa un mesaj de eroare.
- Sistemul verifică parola.
- Dacă parola este eronată va afișa un mesaj de eroare și cere reintroducerea parolei.
- Dacă numele și parola sunt valide se deschide contul utilizatorului.

Scenariu alternativ:

A1 Verificarea numelui de utilizator: Dacă numele de utilizator este eronat va afișa un mesaj de eroare și cere reintroducerea numelui de utilizator.

A2: Verificarea parolei: Dacă parola este eronată va afișa un mesaj de eroare și cere reintroducerea parolei.

Scenariu de Eroare:

Persoana nu se poate loga deoarece sistemul nu se poate conecta la baza de date pentru a vedea existența ei în baza de date.

Postconditii:

Persoana este conectată la sistem și poate accesa opțiunile valabile pentru tipul său de utilizator.

Caz de utilizare: Introduce teste

Titlu: Introduce teste.

Rezumat: Profesorul introduce teste pe care dorește ca studentul să le parcurgă

Actori: profesorul.

Descrierea scenariului:

Precondiții:

- programul trebuie să fie disponibil

Scenariu nominal:

- Profesorul se logează.
- Profesorul verifică statisticiile pentru teste
- Profesorul va verifica scenariile de teste și pentru cod sursă
- Profesorul va vizualiza testele

Scenariu alternativ:

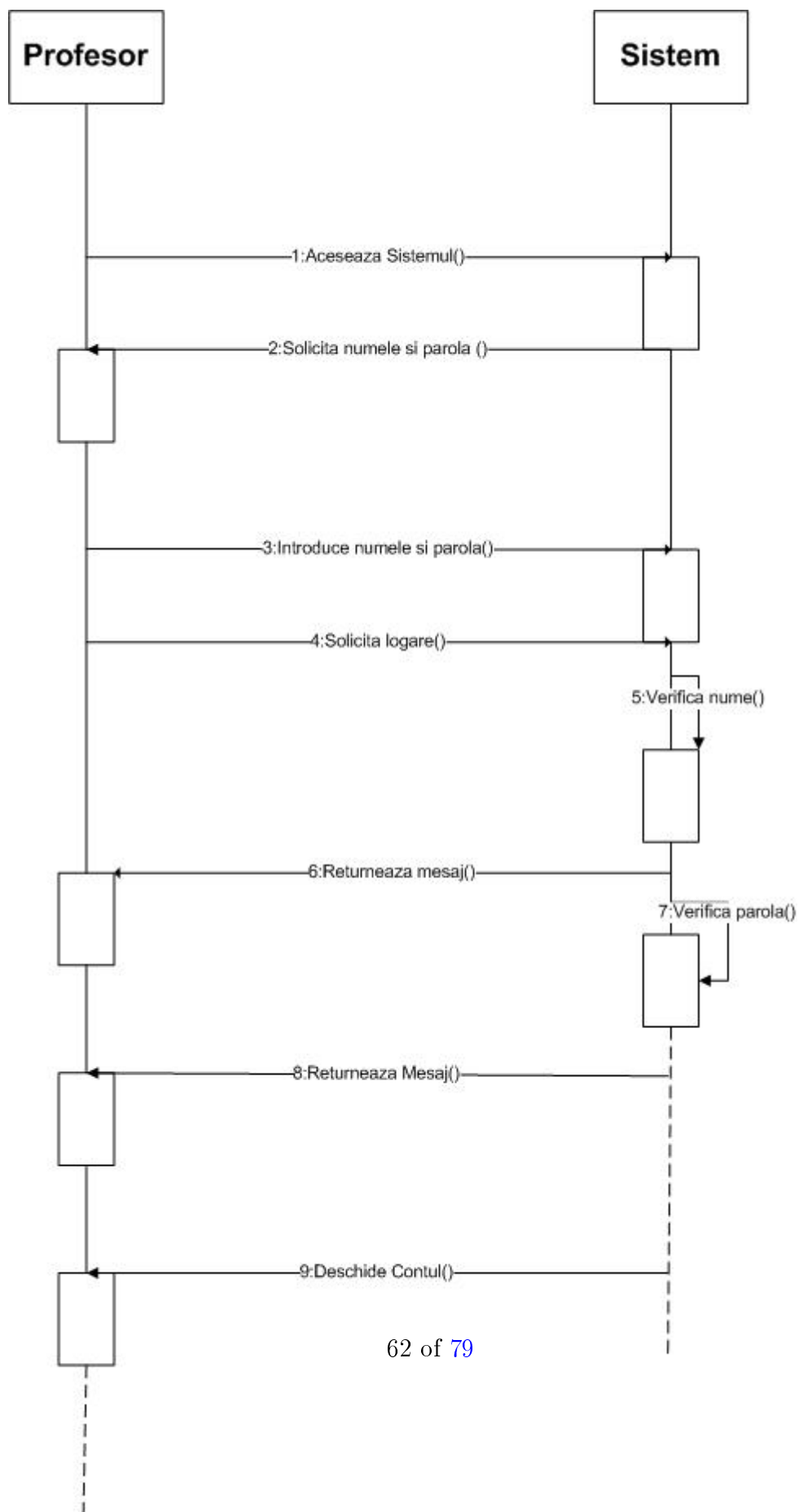
A1.Profesorul poate alege vizualizarea răspunsurilor la teste

Scenariu de eroare :

Programul nu păstrează informațiile de notare a studentului.

Postconditii:

Sistemul afișează testul introdus



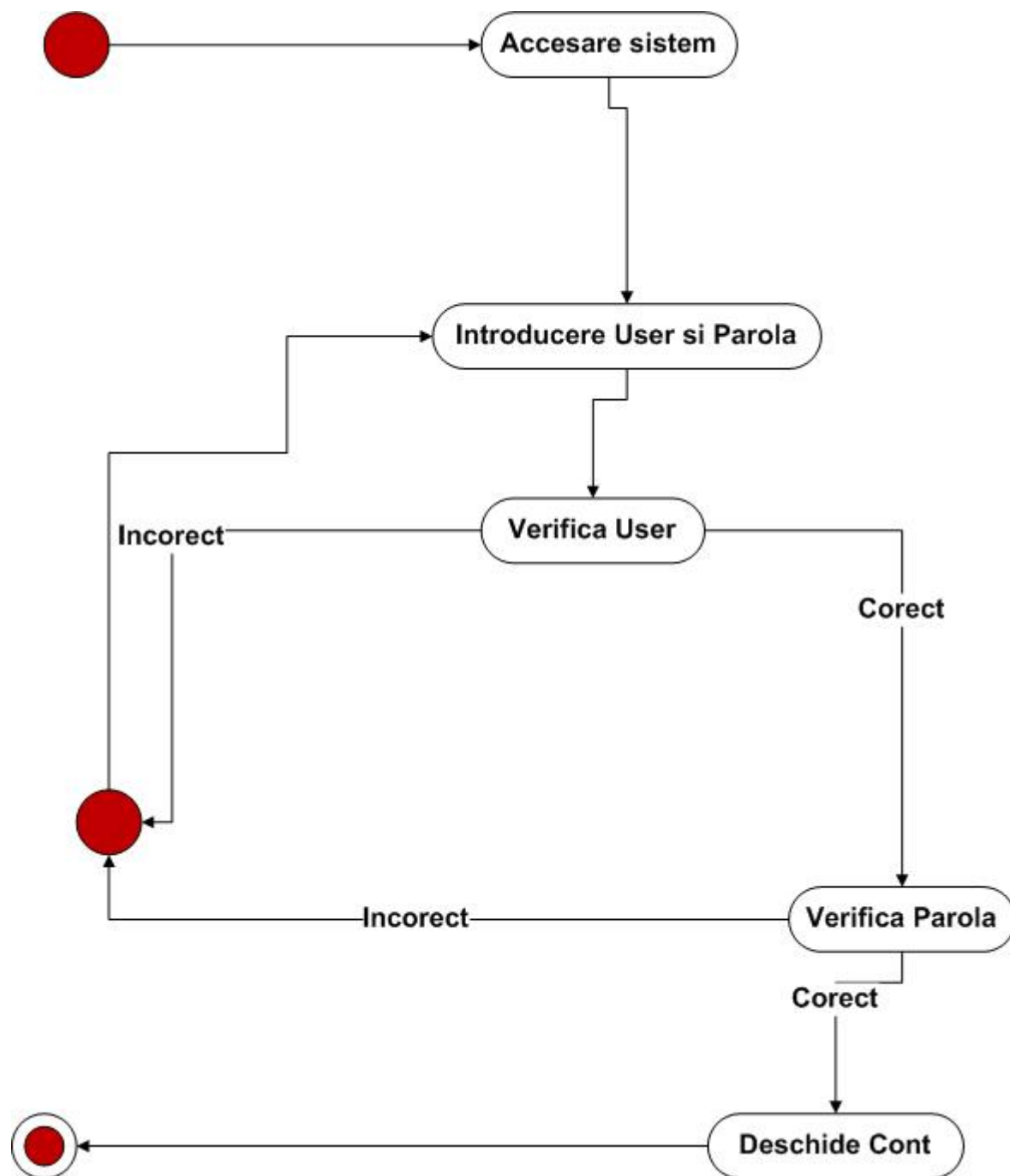


Figura 23: Diagrama de activitate pentru logare



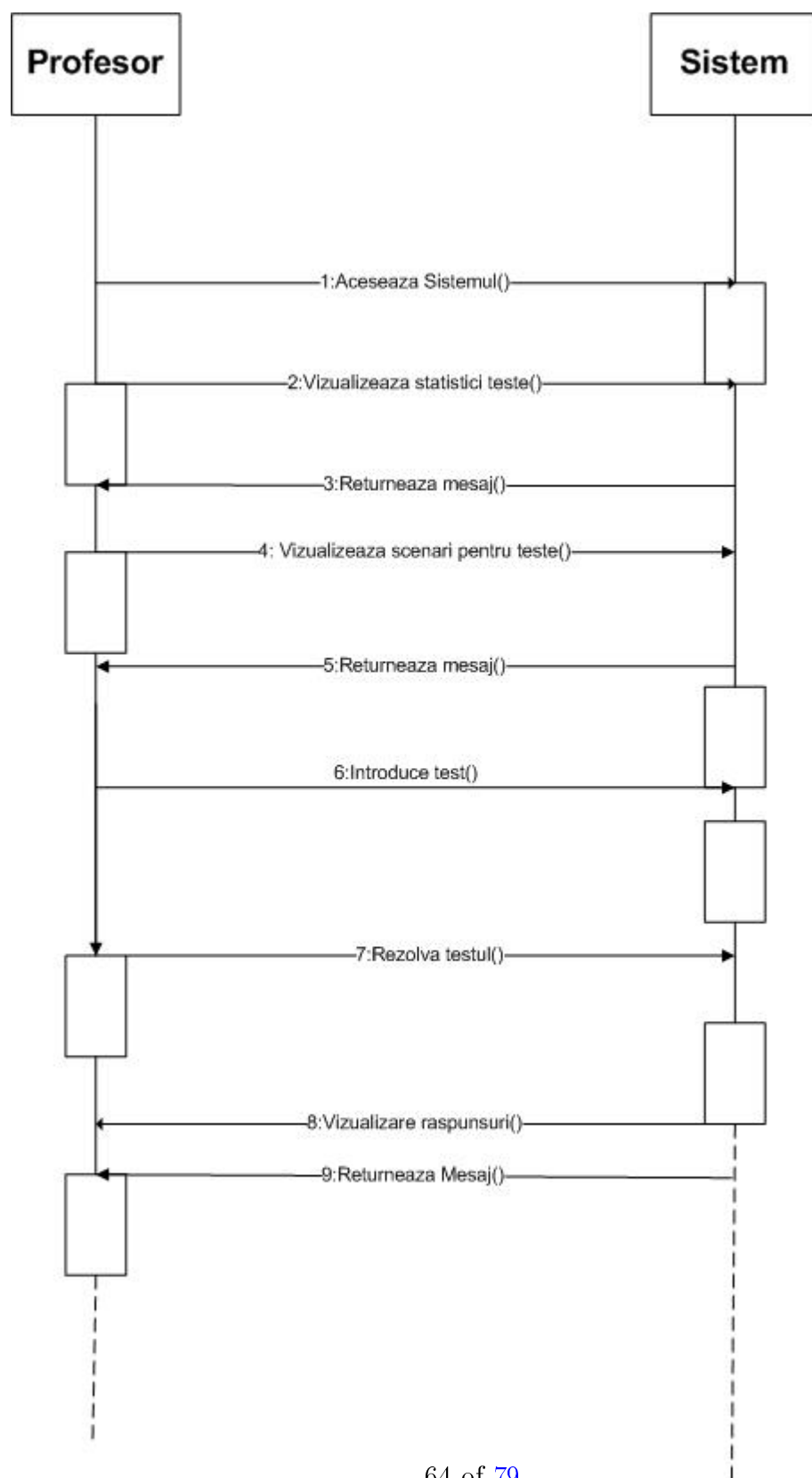


Figura 24: Diagrama de secvență pentru introducerea testului

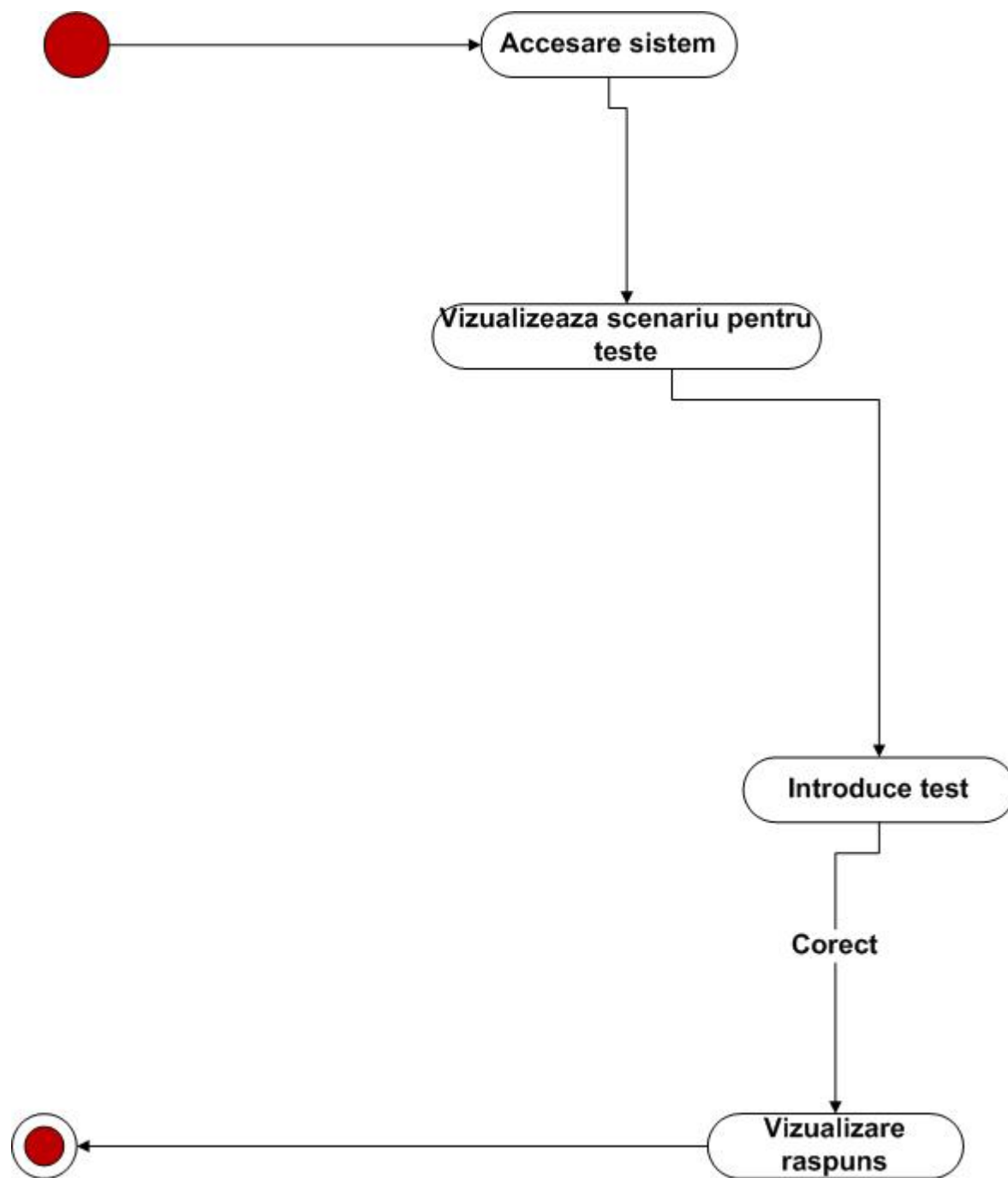


Figura 25: Diagrama de activitate pentru introducerea testului

### 3.1.5 Diagrama cazurilor de utilizare pentru administrator

Diagrama cazurilor de utilizare prezintă modul de folosire a sistemului de către administrator.

#### Descriere cazului de utilizare pentru administrator

Titlu: Logare.

Caz de utilizare: Logare

Rezumat:

Persoana introduce numele de utilizator și parola iar sistemul verifică existența numelui de utilizator și corectitudinea parolei pentru a deschide contul persoanei.

Actori: administrator, sistemul

Descrierea scenariului:

Preconditii:

- persoana se află în program
- programul trebuie să fie disponibil

Scenariu nominal:

1.Administratorul se logează.

Postconditii:

- Sistemul solicită numele și parola.
- Persoana introduce numele și parola.
- Persoana cere logare.
- Sistemul verifică numele.
- Dacă numele de utilizator este eronat va afișa un mesaj de eroare.
- Sistemul verifică parola.
- Dacă parola este eronată va afișa un mesaj de eroare și cere reintroducerea parolei.
- Dacă numele și parola sunt valide se deschide contul utilizatorului.

Scenariu alternativ:

A1 Verificarea numelui de utilizator: Dacă numele de utilizator este eronat va afișa un mesaj de eroare și cere reintroducerea numelui de utilizator.

A2: Verificarea parolei: Dacă parola este eronată va afișa un mesaj de eroare și cere reintroducerea parolei.

Scenariu de Eroare:

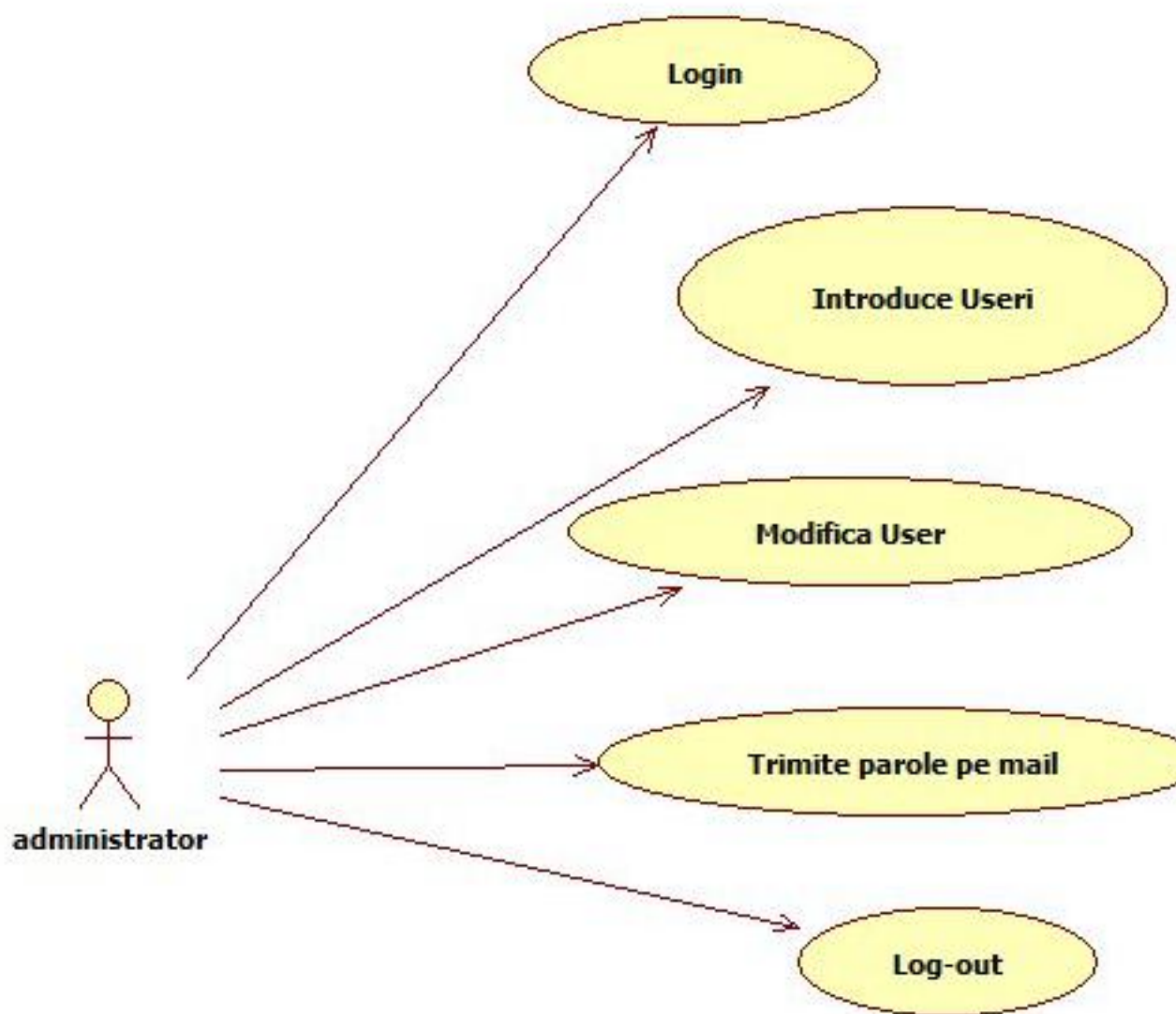


Figura 26: Diagrama cazurilor de utilizare

Persoana nu se poate loga deoarece sistemul nu se poate conecta la baza de date pentru a vedea existența ei în baza de date.

Postcondiții:

Persoana este conectată la sistem și poate accesa opțiunile valabile pentru tipul său de utilizator.

Caz de utilizare: Introduce USER

Titlu: Introduce USER.

Rezumat: Administratorul introduce datele studentului sau ale profesorului

Actori: administratorul.

Descrierea scenariului:

Precondiții:

- programul trebuie să fie disponibil

Scenariu nominal:

- Administratorul se logează.
- Administratorul verifică useri existenți.
- Administratorul va introduce noi useri
- Administratorul va trimite parola pe mail userilor

Scenariu alternativ:

A1. Administratorul poate modifica useri

Scenariu de eroare :

Programul nu păstrează informațiile userilor.

Postcondiții:

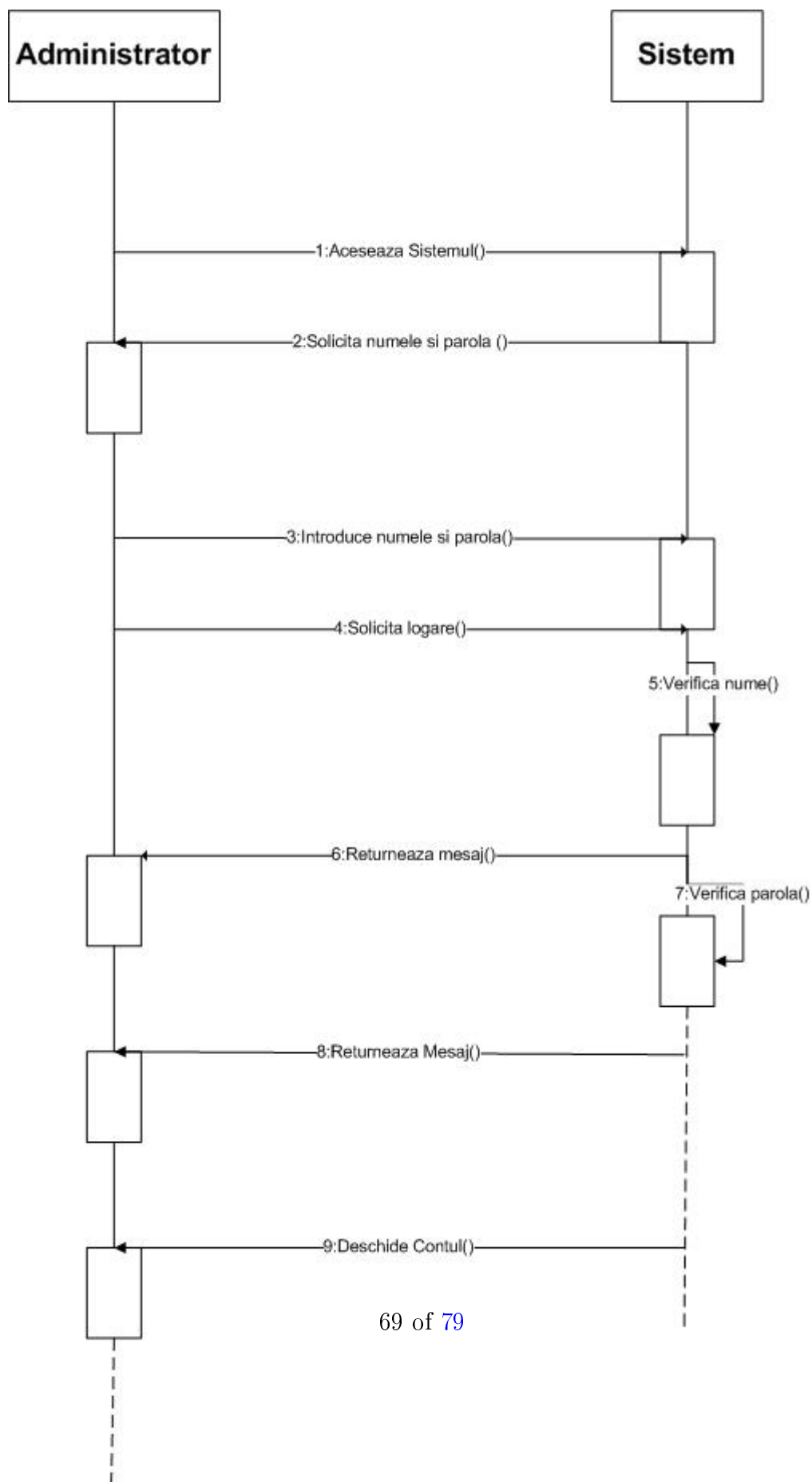
Sistemul afișează useri introduși

### 3.1.6 Diagrama de clasă

### 3.1.7 Structura în package-uri a aplicației

Proiectul conține următoarele packageuri:

1. Student:
  2. Profesor:
  3. Administrator:
  4. Situație școlară
  5. Chestionare:
- Teste  
Context: Student  
Nume[]



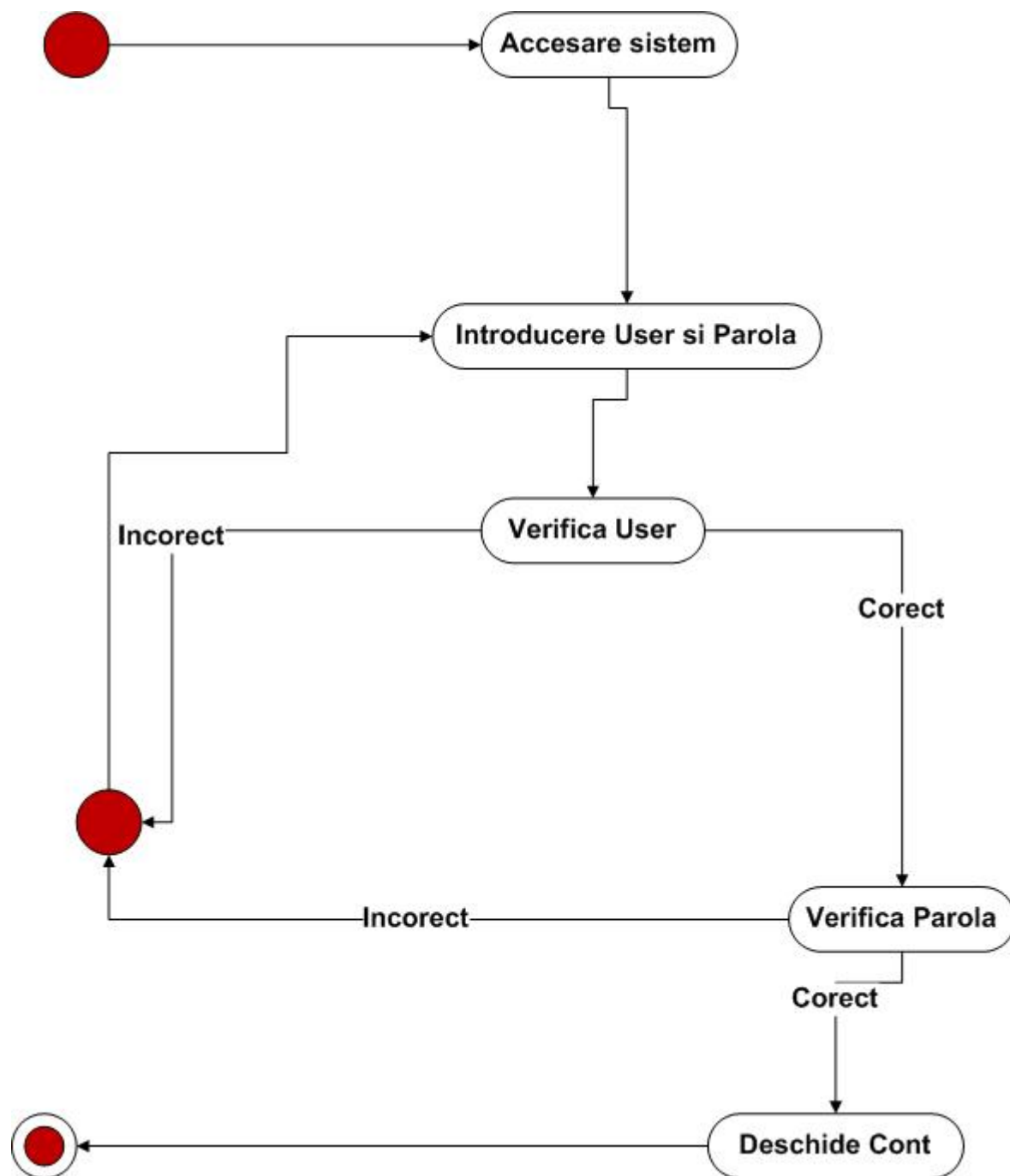


Figura 28: Diagrama de activitate pentru logare

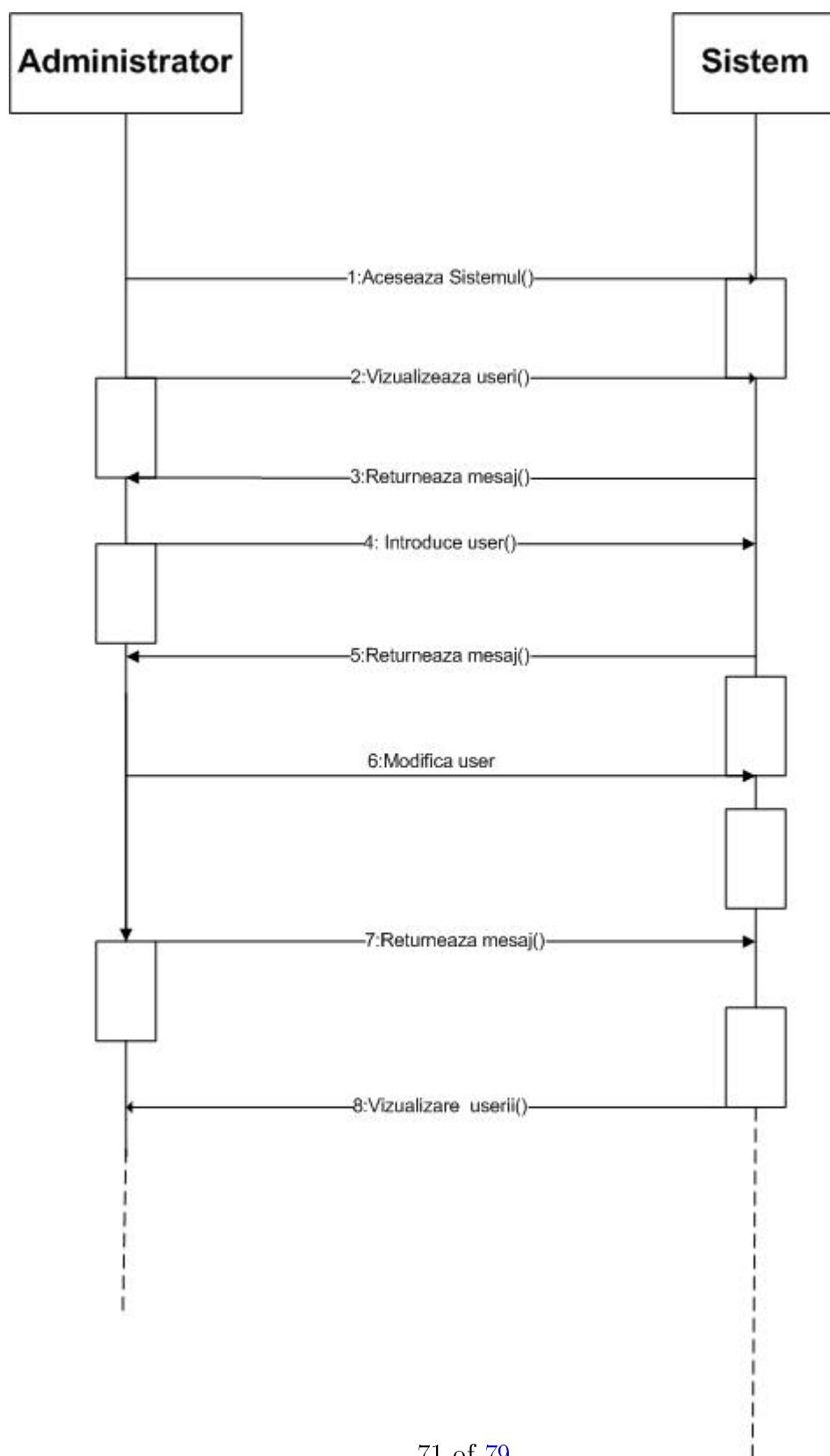


Figura 29: Diagrama de secvență pentru introducerea userilor



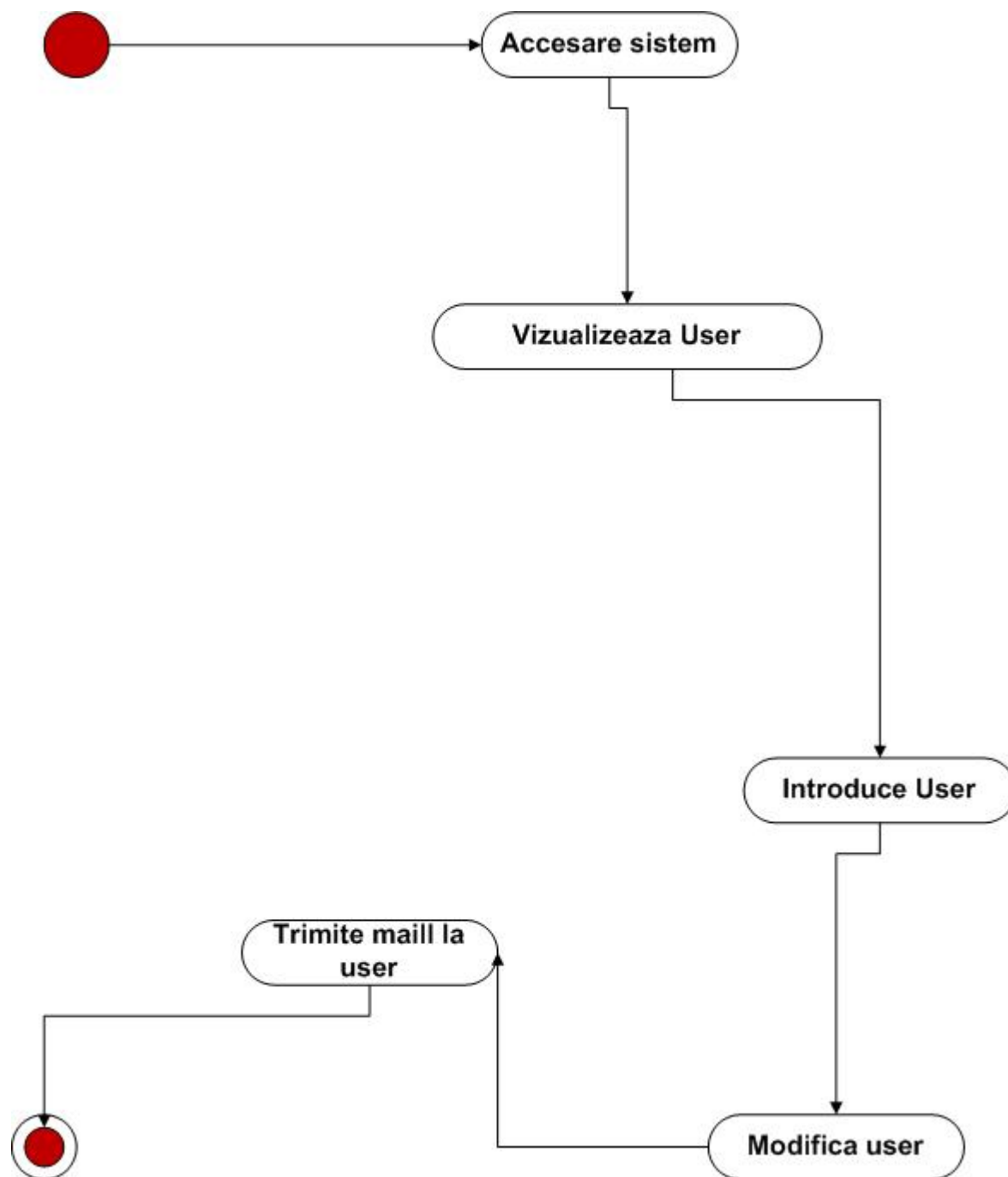


Figura 30: Diagrama de activitate pentru introducerea userilor

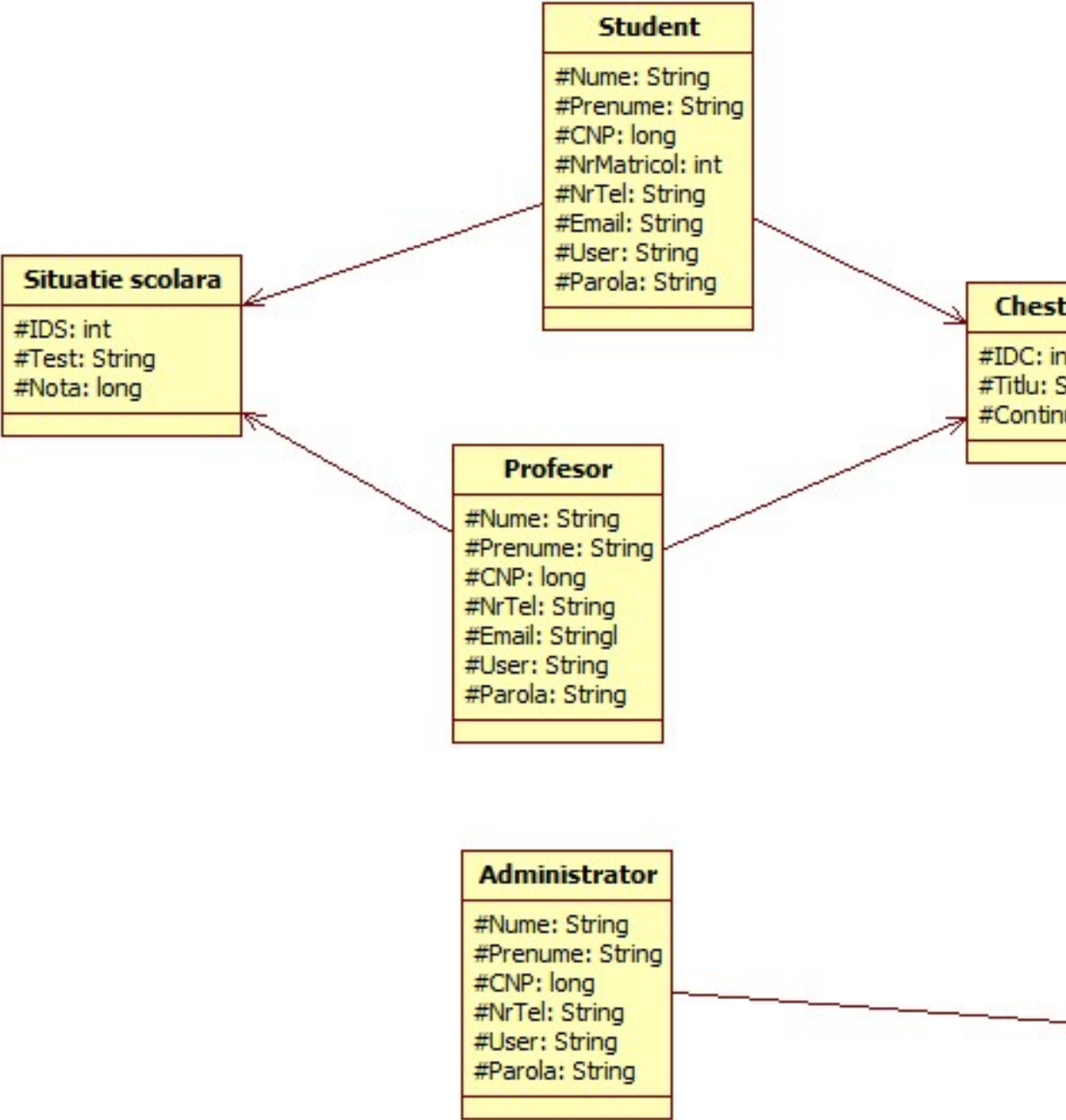


Figura 31: Diagrama de clasă

Prenume[]  
CNP-unic  
NrMatricol-unic  
NrTel[]  
User- unic  
Parola-unic  
Parola = {"a".. "z"} {"A".. "Z"} {"0".. "9"} {orice caracter special}  
Parola.length > 5  
Context: Profesor  
Nume[]  
Prenume[]  
CNP-unic  
NrTel[]  
User- unic  
Parola-unic  
Parola = {"a".. "z"} {"A".. "Z"} {"0".. "9"} {orice caracter special}  
Parola.length > 5  
Context: Administrator  
Nume[]  
Prenume[]  
CNP-unic  
NrTel[]  
User- unic  
Parola-unic  
Parola = {"a".. "z"} {"A".. "Z"} {"0".. "9"} {orice caracter special}  
Parola.length > 5  
Context: Situație școlară  
IDC-unic  
Titlu[]  
Continut[]  
Context: Chestionar  
IDC-unic  
Titlu[]  
Continut[]  
Context: Test  
IDT-unic  
Titlu[]  
Conținut[]

### 3.1.8 Modelul utilizat pentru baza de date

Pentru sistemul de E-learning am folosit o bază de date ierarhică.

O bază de date ierarhică arborescent este considerat un caz particular al modelului rețea, în care diagrama asociată este o pădure (mulțime de arbori) și în care toate legăturile sunt pe direcția drumului, de la rădăcină la nodul fiu din relație, toate relațiile fiind de tipul 1:N.

Există posibilitatea interpretării diagramelor entitate-relație sub forma modelului ierarhic. Pentru evitarea redundanțelor în modelul ierarhic, se folosește noțiunea de element virtual, care înlocuiește dublura unui element prin adresa elementului respectiv, fiecare element apărând în baza de date reală o singură dată.

Operațiile din bazele de date de tip ierarhic se traduc în procese de parcurgere a arborilor. Elementele virtuale permit legarea informațiilor din aceeași entitate sau din entități diferite.

Implementarea la nivel logic pentru modelul ierarhic poate fi cea utilizată pentru modelul rețea sau prin înregistrări de lungime variabilă.

Datele sunt stocate pe mediul extern în ordinea dată de parcurgerea în preordine a arborilor, ceea ce ușurează determinarea informațiilor pentru cererile care se referă la descendenții unor noduri printr-un număr mic de accese la mediul extern.

### 3.1.9 Șabloane de proiectare utilizate

Șabloane de proiectare folosite sunt cele creaționale și anume:

- Factory Method
- Abstract Factory
- Builder
- Prototype
- Singleton

### 3.1.10 Modelul arhitectural al aplicației

În principiu arhitectura aplicațiilor este pe următoarele trei nivele:

1. Baza de date  
Serverul SQL.

Acest nivel include logica aplicației (răspunde la interogările aplicației, face prelucrări masive, etc.)

## 2. Serverul de aplicație.

Asigură legătura între nivelul 3 (clientul) și nivelul 1 (serverul SQL) și realizează diverse prelucrări de date.

## 3. Interfața cu utilizatorul

Graphical User Interface - este clientul aplicației. Fiecare din aceste nivele ale aplicației pot rula pe sisteme hard separate. Cea mai comună configurație este ca baza de date și serverul de aplicație să fie pe același server.

1. Baza de date folosită (phpmyadmin) este un produs care asigură o capacitate de stocare extensibilă, securitatea datelor, instrumente de analiză și management a datelor, multiple posibilități de accesare / import / export a datelor, în diferite formate, respectiv integrare cu alte surse de date.

2. Serverul de aplicație este nivelul la care se realizează majoritatea prelucrărilor de date.

3. Programul client este interfața prin care utilizatorul interacționează cu aplicația. La acest nivel se fac prelucrări de date. Interfața este perfect integrată cu mediul Windows, Ubuntu bogată în opțiuni, permițând flexibilitate în utilizare.

Avantajul arhitecturii pe trei nivele, față de o arhitectură client server, este că majoritatea procesărilor se fac pe serverul de aplicație și pe baza de date, nu pe calculatorul client și pe baza de date. Aceasta permite o stabilitate mult mai bună a aplicației în condițiile unui volum de tranzacții în creștere. Este necesară doar adăugarea de servere suplimentare pentru creșterea capacității de procesare.

Utilizarea phpmyadmin ca și server de baze de date asigură:

- prelucrări rapide pe volum mare de date
- securitate sporită a datelor \item instrumente de analiză și suport decizional (OLAP, Data Mining)
- preluarea datelor din aplicațiile existente și exportul datelor în diferite formate (Excel, PDF, XML, text, etc.)

Limbajele PHP JavaScript permit:

- crearea unui sistem modern, cu interfață ergonomică și intuitivă
- dezvoltarea de module flexibile, ușor de adaptat la cerințele particulare ale studenților și la modificările din materialele de studiu
- extinderea funcționalităților aplicației
- înglobarea automată a noilor versiuni ale aplicației

Nr curent	Denumire camp	Tip(lungime)	Descriere
1	Nume	CHAR(20)	Numele Studentului
2	Prenume	CHAR(20)	Prenumele studentului
3	CNP	INTEGER	CNP student
4	NrMatricol	INTEGER	Numarul matricol al studentului
5	Email	CHAR(10)	Adresa de mail a studentului
6	User	CHAR(20)	Numele pentru logare
7	Parola	CHAR(20)	Parola pentru logare

Tabela 3: Tabela : Student

Nr curent	Denumire camp	Tip(lungime)	Descriere
1	Nume	CHAR(20)	Numele Profesorului
2	Prenume	CHAR(20)	Prenumele Profesorului
3	CNP	INTEGER	CNP Profesorului
4	Email	CHAR(10)	Adresa de mail a Profesor
5	User	CHAR(20)	Numele pentru Profesor
6	Parola	CHAR(20)	Parola pentru Profesor

Tabela 4: Tabela : Profesor

**3.1.11 Designul interfetei utilizator****3.1.12 Designul bazei de date**

Nr curent	Denumire camp	Tip(lungime)	Descriere
1	Nume	CHAR(20) &	Numele Administrator
2	Prenume	CHAR(20)	Prenumele Administrator
3	CNP	INTEGER	CNP Administrator
4	Email	CHAR(10)	Adresa de mail a Administrator
5	User	CHAR(20)	Numele pentru Administrator
6	Parola	CHAR(20)	Parola pentru Administrator

Tabela 5: Tabela : Administrator

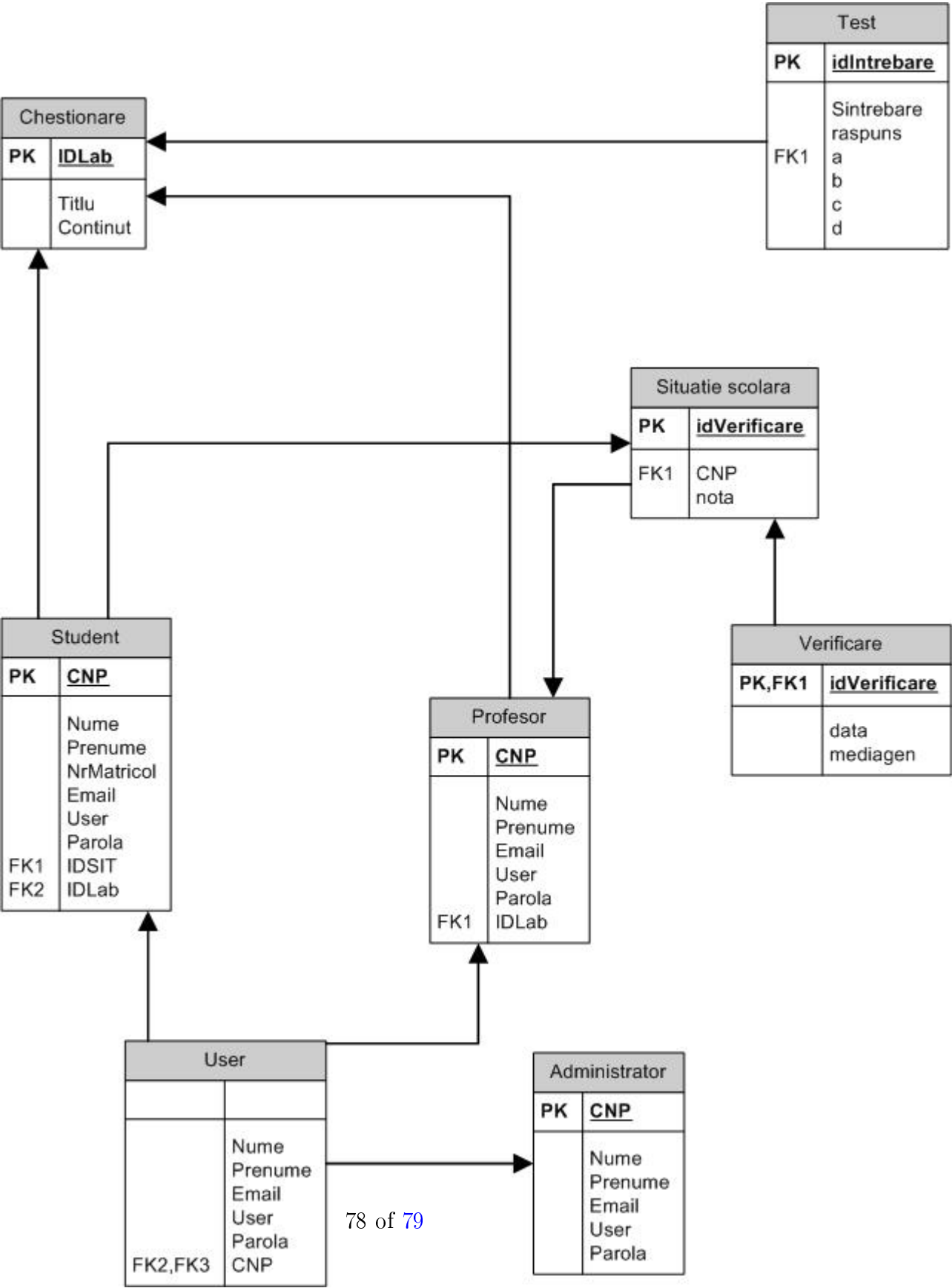


Figura 32: Designul bazei de date

Nr curent	Denumire camp	Tip(lungime)	Descriere
1	idIntrebare	CHAR(20) &	ID întrebare
2	Sintrebare	STRING(50)	Întrebarea
3	raspuns	TEXT(30)	Răspunsul corect
4	a	TEXT(10)	Răspunsul a
5	b	TEXT(10)	Răspunsul b
6	c	TEXT(10)	Răspunsul c
7	d	TEXT(10)	Răspunsul d

Tabela 6: Tabela Test

Nr curent	Denumire camp	Tip(lungime)	Descriere
1	IDLab	CHAR(20)	ID Laborator
2	titlu	STRING(20)	Titlu Lecție
3	continut	TEXT(50)	Conținut lecție

Tabela 7: Tabela Chestionare

Nr curent	Denumire camp	Tip(lungime)	Descriere
1	idVerificare	CHAR(20)	ID Verificare
2	CNP	CHAR(20)	CNP Student
3	nota	CHAR(20)	Nota Studentului

Tabela 8: Situație Școlară

Nr curent	Denumire camp	Tip(lungime)	Descriere
1	idVerificare	CHAR(20)	ID Verificare
2	data	CHAR(20)	Data testului
3	mediagen	CHAR(20)	Media generală

Tabela 9: Tabela Verificare