

Aplicații Integrate pentru Întreprinderi

Tema 3

Proiectarea unei aplicații publisher-subscriber folosind socket-uri TCP

Lansare	Termen de predare	Pondere
22.11.2011, 23:55	13.12.2011, 23:55	10 / 70

Obiective

Scopul laboratorului constă în realizarea unui sistem informatic performant pentru gestiunea resurselor unei întreprinderi de dimensiuni mari. În scop didactic dar și pentru a vă ajuta în procesul de analiză a sistemului ERP, instituția pentru care se va dezvolta aplicația nu face parte din sectorul comercial ci din spațiul academic – cu care se presupune că sunteți mai familiari.

Aplicația va fi implementată gradual prin intermediul unor teme de casă, pe parcursul a patru etape.

În cadrul acestei etape, componentele sistemului informatic vor comunica după modelul *publisher-subscriber* prin intermediul unor mesaje care pot fi sincrone sau asincrone.

După rezolvarea temei de casă, studentul va fi capabil să:

- proiecteze o aplicație având o structură distribuită pe componente putând rula pe mașini diferite;
- modeleze o comunicație (asincronă) între componentele distribuite ale aplicației folosind mecanisme de nivel scăzut.

Cunoștințele necesare¹ pentru rezolvarea temei de casă sunt:

- programarea în limbajul Java;
- manipularea bazelor de date folosind MySQL;
- folosirea API-ului Java DataBase Connectivity;
- folosirea de socket-uri TCP pentru comunicația între componente.

Enunț

Arhitectura sistemului informatic pentru facilitarea activităților didactice din cadrul Universității „Politehnica” București va implementa o componentă **Jurnal** având rolul de a înregistra solicitările venite din partea studenților, acestea urmând a fi distribuite corespunzător către secretari și cadre didactice care le vor analiza, transmițând rezultatul la solicitare. Astfel, comunicația între studenți și secretari / cadre didactice va fi asincronă întrucât nu se poate estima când va fi rezolvată o cerere. Comunicația între utilizatori și componentele **Arhiva**, respectiv **RegistruActivitățiDidactice** va fi sincronă (acestea păstrându-și funcționalitatea²), operațiile putând fi realizate în mod automat prin accesarea informațiilor corespunzătoare din cadrul bazei de date.

Comunicația între componente va fi implementată folosind mecanisme de nivel scăzut, și anume **socket-uri TCP**.

¹ Se poate folosi orice limbaj de programare și orice sistem de gestiune pentru baze de date, însă este obligatorie folosirea de socket-uri TCP pentru comunicația între componente.

² Accesul la baza de date este rezervat componentelor Arhiva și RegistruActivitățiDidactice, operând asupra tabelelor discipline / catalog, respectiv utilizatori / orar / programare examene.

Funcționalitatea unei componente nu va mai fi accesată prin invocarea unor metode, ci prin intermediul unor mesaje, a căror structură este cunoscută (atât de sursă cât și de destinație), astfel încât din analiza lor se poate identifica tipul de operație care trebuie realizată și datele asupra cărora aceasta se aplică.

Pentru un mesaj care conține m parametri având forma { atribut / valoare } și n parametri având forma { atribut₁ / valoare₁ ... atribut_k / valoare_k }³, structura ar putea fi următoarea:

mesaj				
parametru ₁ ... parametru _m	dimensiune (n)	atribut ₁₁ ... atribut _{1k}		
		...		
		atribut _{n1} ... atribut _{nk}		

Parametrii vor fi delimitați prin caracterul „*”, în timp ce atributele vor fi delimitate prin caracterul „#”⁴.

Exemplu:

```
mesaj*parametru1*...*parametrum*n*atribut11#...#atribut1k
...
*atributn1#...#atributnk5
```

1. Un student

a. poate obține notele pentru una sau mai multe discipline prin intermediul componentei **Arhiva**:

solicitare_note		
CNP_student	numar_discipline (n)	cod_disciplina ₁
		cod_disciplina ₂
		...
		cod_disciplina _n

Exemplu:

```
solicitare_note*1900101001002*2*UPB03L1001-01*UPB03L2A02-02
```

Răspunsul va fi transmis instantaneu de componenta **Arhiva** sub forma unui mesaj conținând toate notele obținute de student la disciplinele respective:

raspuns_solicitare_note	
numar_note (n)	nota_disciplina ₁
	nota_disciplina ₂
	...
	nota_disciplina _n

Exemplu:

```
raspuns_solicitare_note*2*10*10
```

³ Astfel de parametri corespund unor obiecte – instanțe ale unor clase.

⁴ În situația în care totuși acest caracter se regăsește în informațiile reținute în baza de date, puteți folosi alt caracter pentru a delimita elementele din structura mesajului.

⁵ Mesajul nu va conține caractere de tip linie nouă (' \n '), în exemplu s-a procedat la împărțirea pe mai multe linii pentru a scoate în evidență structura sa.

Trebuie avute în vedere situațiile în care nota nu este trecută în catalog (valoarea întoarsă pentru disciplina respectivă este -1), disciplina pentru care studentul face solicitarea nu se găsește în contractul său de studii (se întoarce -2) respectiv a fost specificat un CNP care nu este valid⁶.

b. poate obține situația școlară pentru anul universitar curent prin intermediul componentei **Arhiva**:

solicitare_situatie_scolara
CNP_student

Exemplu:

solicitare_situatie_scolara*1900101001002

Răspunsul va fi transmis instantaneu de componenta **Arhiva** sub forma unui mesaj conținând informații cu privire la situația școlară:

raspuns_solicitare_situatie_scolara
medie_aritmetica medie_ponderata puncte_credit examene_nepromovate

Exemple:

raspuns_solicitare_situatie_scolara*10.00*10.00*60*0
raspuns_solicitare_situatie_scolara*0.00*0.00*40*4

Se vor calcula media aritmetică și media ponderată doar în situația în care studentul a promovat toate examenele prevăzute în contractul de studii.

Se va trata și în acest caz situația unui CNP care nu este valid.

c. poate obține orarul pentru semestrul în curs prin intermediul componentei **RegistruActivitățiDidactice**:

solicitare_orar
CNP_student

Exemplu:

solicitare_orar*1900101001002

Răspunsul va fi transmis instantaneu de componenta **RegistruActivitățiDidactice** sub forma unui mesaj conținând informații referitoare la orar:

raspuns_solicitare_orar							
numar_ore (n)	zi ₁	interval_timp ₁	sala ₁	disciplina ₁	cadru_didactic ₁	tip_activitate ₁	frecvența ₁
	zi ₂	interval_timp ₂	sala ₂	disciplina ₂	cadru_didactic ₂	tip_activitate ₂	frecvența ₂

	zi _n	interval_timp _n	sala _n	disciplina _n	cadru_didactic _n	tip_activitate _n	frecvența _n

Exemplu:

raspuns_solicitare_orar*4
*marti#14-17#EC101#Inteligența Artificială#Adina Florea#curs#saptamanal
*marti#17-20#AN034#Aplicații Integrate pentru Întreprinderi#Mihai Ionescu#curs#saptamanal
*miercuri#14-17#EC105#Sisteme de Prelucrare Grafică#Florica Moldoveanu#curs#saptamanal
*miercuri#17-20#EC105#Interfața Om Calculator#Stefan Trausan Matu#curs#saptamanal

⁶ Mesajul care va fi transmis în această situație va fi raspuns_solicitare_note*-1.

Se vor avea în vedere situațiile în care orarul nu a fost completat, respectiv CNP nu este valid.

d. poate obține calendarul examenelor pentru sesiunea în curs prin intermediul componentei **RegistruActivitățiDidactice**:

solicitare_calendar_examene
CNP_student

Exemplu:

solicitare_calendar_examene*1900101001002

Răspunsul va fi transmis instantaneu de componenta **RegistruActivitățiDidactice** sub forma unui mesaj conținând informații referitoare la calendarul de examene:

raspuns_solicitare_calendar_examene					
numar_examene (n)	data ₁	ora ₁	sala ₁	disciplina ₁	cadru_didactic ₁
	data ₂	ora ₂	sala ₂	disciplina ₂	cadru_didactic ₂

	data _n	ora _n	sala _n	disciplina _n	cadru_didactic _n

Exemplu:

raspuns_calendar_examene*4
*17-01-2012#08.00#EC001#Inteligența Artificială#Adina Florea
*20-01-2012#11.00#EC002#Aplicații Integrate pentru Întreprinderi#Mihai Ionescu
*24-01-2012#14.00#EC003#Sisteme de Prelucrare Grafică#Florica Moldoveanu
*27-01-2012#17.00#EC004#Interfața Om Calculator#Stefan Trausan Matu

Se vor avea în vedere situațiile în care calendarul de examene nu a fost completat, respectiv CNP nu este valid.

e. poate solicita unui secretar schimbarea grupei din care face parte folosindu-se de componenta **Jurnal**:

schimbare_grupa
CNP_student grupa_noua

Exemplu:

schimbare_grupa*1900101001002*341C4

Componenta **Jurnal** va stoca această solicitare (precum și starea ei⁷), transmițând înapoi studentului un răspuns conținând un identificator având semnificația unui număr de înregistrare:

raspuns_schimbare_grupa
numar_inregistrare CNP_student grupa_noua

Exemplu:

raspuns_schimbare_grupa*111111*1900101001002*341C4

În situația în care mai există o solicitare de schimbare a grupei din partea aceluiași student, se va genera un mesaj de eroare.

⁷ Valorile pe care le poate lua starea unei solicitări sunt: neanalizată (0), respectiv acceptată (1) sau respinsă (-1).

f. poate transmite unui cadru didactic o contestație cu privire la nota acordată pentru o disciplină folosindu-se de componenta **Jurnal**:

contestare_nota		
CNP_student	CNP_cadru_didactic	cod_disciplina

Exemplu:

contestare_nota*1900101001002*1700202001002*UPB03L1001-01

Componenta **Jurnal** va stoca această solicitare (precum și starea ei), transmițând înapoi studentului un răspuns conținând un identificator având semnificația unui număr de înregistrare:

raspuns_contestare_nota			
numar_inregistrare	CNP_student	CNP_cadru_didactic	cod_disciplina

Exemplu:

raspuns_contestare_nota*222222*1900101001002*1700202001002*UPB03L1001-01

În situația în care mai există o solicitare de contestare a notei pentru aceeași disciplină din partea aceluiași student, se va genera un mesaj de eroare.

Oricând, studentul poate verifica starea solicitărilor sale printr-un mesaj către componenta **Jurnal**:

rezultat_solicitari	
CNP_student	

Exemplu:

rezultat_solicitari*1900101001002

Componenta **Jurnal** va întoarce starea tuturor solicitărilor, urmând ca toate solicitările care au fost rezolvate (acceptate sau respinse) să fie eliminate ulterior comunicării rezultatelor.

raspuns_rezultat_solicitari		
numar_solicitari (n)	numar_inregistrare ₁	stare_solicitare ₁
	numar_inregistrare ₂	stare_solicitare ₂

	numar_inregistrare _n	stare_solicitare _n

Exemplu:

raspuns_rezultat_solicitari*2*111111#-1*222222#1

Pentru exemplul de mai sus, solicitările cu numerele de înregistrare 111111 și 222222 sunt șterse din componenta Jurnal după transmiterea lor către student.

O solicitare poate fi respinsă și ca urmare a furnizării unor date care nu sunt valide.

2. Un șef de catedră

a. poate stabili planul de învățământ, transmițând un mesaj către componenta **Arhiva**

stabilire_plan_de_invatamant									
numar_discipline (n)	operatie ₁	cod ₁	denumire ₁	tip ₁	numar_ore ₁	puncte_credit ₁	forma_examinare ₁	an ₁	semestru ₁
	operatie ₂	cod ₂	denumire ₂	tip ₂	numar_ore ₂	puncte_credit ₂	forma_examinare ₂	an ₂	semestru ₂

	operatie _n	cod _n	denumire _n	tip _n	numar_ore _n	puncte_credit _n	forma_examinare _n	an _n	semestru _n

Parametrul operație poate avea valorile adăugare, editare sau ștergere. Pentru operațiile de adăugare și de editare este obligatorie specificarea tuturor parametrilor.

Operația de adăugare se va realiza cu succes dacă nu există o disciplină având același cod în tabela disciplina.

Operația de editare se va realiza cu succes dacă există o disciplină cu același cod în tabela disciplina, atributele fiind actualizate corespunzător cu valorile din mesaj.

Pentru operația de ștergere este suficientă precizarea parametrului cod.

Exemplu:

```
stabilire_plan_de_invatamant*3
*adaugare#UPB03L1001-01#Inteligenta Artificiala#obligatorie#5#6#examen#4#1
*editare#UPB03L1001-02#AplicatiiIntegrate pentru Intreprinderi#obligatorie#5#6#examen#4#1
*stergere#UPB03L1001-03
```

b. poate realiza repartizarea unui cadru didactic pentru un tip de activitate corespunzător unei discipline de învățământ, transmițând un mesaj către componenta **RegistruActivitățiDidactice**

repartizare_cadre_didactice			
numar_repartizari (n)	CNP_cadru_didactic ₁	cod_disciplina ₁	tip_activitate_didactica ₁
	CNP_cadru_didactic ₂	cod_disciplina ₂	tip_activitate_didactica ₂

	CNP_cadru_didactic _n	cod_disciplina _n	tip_activitate_didactica _n

Exemplu:

```
repartizare_cadre_didactice*3
*1700101001002#UPB03L1001-01#curs
*1700202001002#UPB03L1001-02#seminar
*1700303001002#UPB03L1001-03#laborator
```

Răspunsurile la aceste operații vor fi transmise instantaneu de componentele **Arhiva / RegistruActivitățiDidactice** sub forma unor mesaje conținând rezultatele operațiilor respective:

raspuns_stabilire_plan_de_invatamant raspuns_repartizare_cadre_didactice	
numar_operatii (n)	rezultat_operatie ₁
	rezultat_operatie ₂
	...
	rezultat_operatie _n

Exemple⁸:

```
raspuns_stabilire_plan_de_invatamant*3*1*1*-1
raspuns_repartizare_cadre_didactice*3*-1*1*1
```

⁸ Se poate folosi convenția ca valorile pozitive să indice realizarea cu succes a operației respective, în timp ce valorile negative specifică o eroare.

De asemenea, șeful de catedră va implementa și funcționalitatea descrisă pentru un cadru didactic.

3. Un cadru didactic

a. poate specifica note obținute de către studenți la examene prin intermediul componentei **Arhiva**

stabilire_nota					
CNP_cadru_didactic	numar_nota (n)	cod_disciplina ₁	CNP_student ₁	nota ₁	data ₁
		cod_disciplina ₂	CNP_student ₂	nota ₂	data ₂
	
		cod_disciplina _n	CNP_student _n	nota _n	data _n

Exemplu:

```
stabilire_nota*1700101001002*3
*UPB03L1001-01#1900101001002#10.00#17-01-2012
*UPB03L1001-02#1900202001002#9.00#19-01-2012
*UPB03L1001-03#1900303001002#8.00#21-01-2012
```

Răspunsul va fi transmis instantaneu de componentele **Arhiva** sub forma unui mesaj conținând rezultatele operațiilor respective:

raspuns_stabilire_nota	
numar_operatii (n)	rezultat_operatie ₁
	rezultat_operatie ₂
	...
	rezultat_operatie _n

Exemplu:

```
raspuns_stabilire_nota*3*1*1*-1
```

b. poate vizualiza solicitările venite din partea studenților prin intermediul componentei **Jurnal**:

cereri_în_așteptare_cadru_didactic
CNP_cadru_didactic

Exemplu:

```
cereri_in_asteptare_cadru_didactic*1700101001002
```

Componenta **Jurnal** va transmite din solicitările înregistrate doar pe cele adresate respectivului cadru didactic:

raspuns_cereri_în_așteptare_cadru_didactic			
numar_solicitări (n)	numar_inregistrare ₁	CNP_student ₁	cod_disciplina ₁
	numar_inregistrare ₂	CNP_student ₂	cod_disciplina ₂

	numar_inregistrare _n	CNP_student _n	cod_disciplina _n

Exemplu:

```
raspuns_cereri_in_asteptare_cadru_didactic*3
*111111#1900101001002#UPB03L1001-01
*222222#1900202001002#UPB03L1001-02
*333333#1900303001002#UPB03L1001-03
```

Aceste solicitări vor fi analizate pentru a se verifica dacă în catalog există note ale studenților puse de cadrul didactic în cauză la disciplina respectivă. Dacă există o astfel de notă, este cerută o decizie din partea cadrului didactic (acceptarea sau respingerea contestației cu modificarea corespunzătoare a notei) respectiv specificarea unei erori care va fi înregistrată de componenta **Jurnal**.

c. poate specifica rezolvări pentru solicitările la care nu au conținut erori⁹, transmise către componenta **Jurnal**¹⁰

rezolvare_cereri_cadru_didactic			
numar_solicitări (n)		numar_inregistrare ₁	rezultat_solicitare ₁
		numar_inregistrare ₂	rezultat_solicitare ₂
	
		numar_inregistrare _n	rezultat_solicitare _n

Exemplu:

rezolvare_cereri_cadru_didactic*3*111111#1*222222#-1*333333#1

Starea solicitărilor va fi marcată corespunzător de componenta **Jurnal**.

4. Un secretar

a. poate stabili grupa din care fac parte mai mulți studenți (formație de studiu) prin intermediul componentei **RegistruActivitățiDidactice**:

stabilire_formatie_de_studiu		
grupa	numar_solicitări (n)	CNP_student ₁
		CNP_student ₂
		...
		CNP_student _n

Exemplu:

stabilire_formatie_de_studiu*341C4*3*1900101001002*1900202001002*1900303001002

b. poate edita unele atribute din orar

stabilire_orar							
numar_ore (n)	zi ₁	interval_timp ₁	sala ₁	disciplina ₁	cadru_didactic ₁	tip_activitate ₁	frecvența ₁
	zi ₂	interval_timp ₂	sala ₂	disciplina ₂	cadru_didactic ₂	tip_activitate ₂	frecvența ₂

	zi _n	interval_timp _n	sala _n	disciplina _n	cadru_didactic _n	tip_activitate _n	frecvența _n

Exemplu:

stabilire_orar*4
 *marti#14-17#EC101#Inteligența Artificială#Adina Florea#curs#saptamanal
 *marti#17-20#AN034#Aplicații Integrate pentru Întreprinderi#Mihai Ionescu#curs#saptamanal
 *miercuri#14-17#EC105#Sisteme de Prelucrare Grafică#Florica Moldoveanu#curs#saptamanal
 *miercuri#17-20#EC105#Interfața Om Calculator#Stefan Trausan Matu#curs#saptamanal

⁹ Solicitățile care au conținut erori (în urma analizei mesajului raspuns_cereri_in_asteptare) sunt respinse în mod automat, tot printr-un mesaj de acest tip.

¹⁰ În cazul în care solicitarea este rezolvată cu succes, va trebui modificată și nota în catalog, printr-un mesaj stabilire_nota.

Pentru ca operația să poată fi realizată cu succes, orarul trebuie să conțină date despre { disciplina, cadru_didactic, tip_activitate }¹¹, introduse în prealabil de șeful de catedră.

c. poate stabili calendarul de examene prin intermediul componentei

RegistruActivitățiDidactice:

stabilire_calendar_examene						
numar_examene (n)	operație ₁	data ₁	ora ₁	sala ₁	disciplina ₁	grupa ₁
	operație ₂	data ₂	ora ₂	sala ₂	disciplina ₂	grupa ₂

	operație _n	data _n	ora _n	sala _n	disciplina _n	grupa _n

Exemplu:

```
stabilire_calendar_examene*3
*adaugare#17-01-2012#08.00#EC001#Inteligența Artificială#341C4
*editare#20-01-2012#11.00#EC002#Aplicații Integrate pentru Întreprinderi#342C4
*stergere#24-01-2012#14.00#EC003#Sisteme de Prelucrare Grafică#343C4
```

Parametrul operație poate avea valorile adăugare, editare sau ștergere. Pentru operațiile de adăugare și de editare este obligatorie specificarea tuturor parametrilor.

Operația de adăugare se va realiza cu succes dacă nu există programat un examen pentru o anumită disciplină aferent unei grupe.

Operația de editare se va realiza cu succes dacă există un examen programat pentru o anumită disciplină aferent unei grupe, atributele fiind actualizate corespunzător cu valorile din mesaj.

Pentru operația de ștergere este suficientă precizarea unor atribute precum disciplina și grupă¹².

Răspunsul va fi transmis instantaneu de componenta

RegistruActivitățiDidactice sub forma unor mesaje conținând rezultatele operațiilor respective:

raspuns_stabilire_formatie_de_studiu	
raspuns_stabilire_orar	
raspuns_stabilire_calendar_examene	
numar_operatii (n)	rezultat_operatie ₁
	rezultat_operatie ₂
	...
	rezultat_operatie _n

Exemple:

```
raspuns_stabilire_formatie_de_studiu*3*-1*1*1
raspuns_stabilire_orar*3*1*-1*1
raspuns_stabilire_calendar_examene*3*1*1*-1
```

¹¹ Toate aceste atribute fac parte din cheia primară a tabelului orar.

¹² Se presupune că un examen pentru o disciplină este programat o singură dată pentru o grupă într-o sesiune. În cazul în care au fost folosite mai multe atribute pentru cheia primară, vor fi folosite toate câmpurile care definesc în mod unic o înregistrare.

d. poate vizualiza solicitările venite din partea studenților prin intermediul componentei **Jurnal**:

cereri_în_asteptare_secretar
CNP_secretar

Exemplu:

```
cereri_in_asteptare_secretar*1700101001002
```

Componenta **Jurnal** va transmite din solicitările înregistrate doar pe cele adresate respectivului secretar:

raspuns_cereri_în_asteptare_secretar			
numar_solicitări (n)	numar_inregistrare ₁	CNP_student ₁	grupa_noua ₁
	numar_inregistrare ₂	CNP_student ₂	grupa_noua ₂

	numar_inregistrare _n	CNP_student _n	grupa_noua _n

Exemplu:

```
raspuns cereri in asteptare secretar*3
*111111#1900101001002#341C4
*222222#1900202001002#342C4
*333333#1900303001002#343C4
```

Aceste solicitări vor fi analizate pentru a se verifica dacă CNP-ul specificat de student este valid și dacă grupa în care solicită să fie înscris nu aparține altui an de studiu sau dacă nu face parte deja din această formație de studiu.

Dacă sunt întrunite toate aceste condiții, se cere o decizie a secretarului (acceptarea sau respingerea solicitării de schimbare a grupei) respectiv are loc specificarea unei erori care va fi înregistrată de componenta **Jurnal**.

e. poate specifica rezolvări pentru solicitările la care nu au conținut erori¹³, transmise către componenta **Jurnal**¹⁴

rezolvare_cereri_secretar		
numar_solicitări (n)	numar_inregistrare ₁	rezultat_solicitare ₁
	numar_inregistrare ₂	rezultat_solicitare ₂

	numar_inregistrare _n	rezultat_solicitare _n

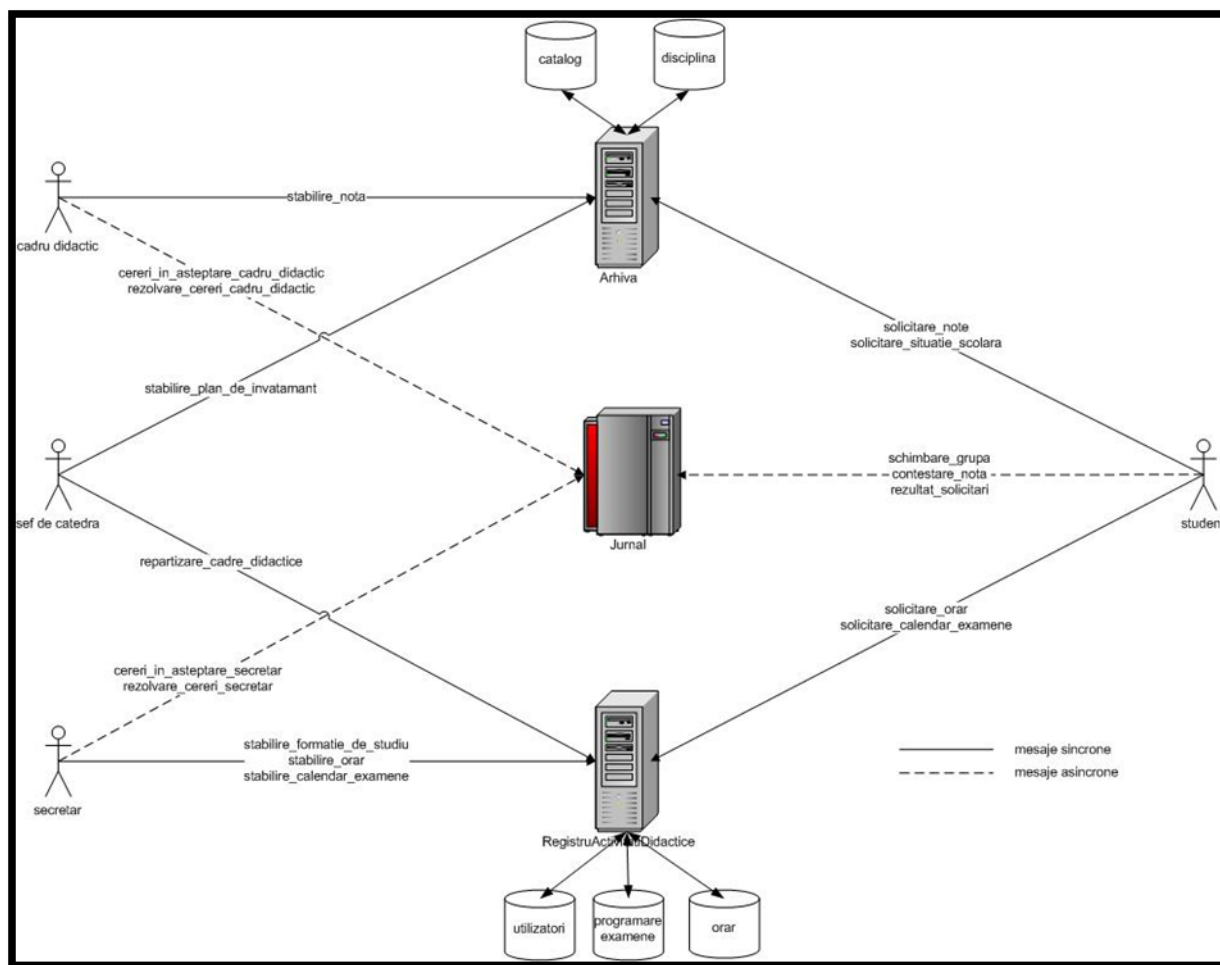
Exemplu:

```
rezolvare_cereri_secretar*3*111111#1*222222#-1*333333#1
```

Starea solicitărilor va fi marcată corespunzător de componenta **Jurnal**.

¹³ Solicitățile care au conținut erori (în urma analizei mesajului raspuns_cereri_in_asteptare) sunt respinse în mod automat, tot printr-un mesaj de acest tip.

¹⁴ În cazul în care solicitarea este rezolvată cu succes, va trebui modificată și grupa în utilizatori, printr-un mesaj stabilire_formatie_de_studiu.



**Arhitectura distribuită pe componente a aplicației
și mesaje transmise între componente**

Mesajele schimbate între componentele de tip server și componentele de tip client¹⁵

Nr. Crt.	Denumire Mesaj	Sursă	Destinație	Tip Mesaj
1.	solicitare_nota	student	Arhiva	sincron
2.	solicitare_situatie_scolara	student	Arhiva	sincron
3.	solicitare_orar	student	registruActivitatiDidactice	sincron
4.	solicitare_calendar_examene	student	registruActivitatiDidactice	sincron
5.	schimbare_grupa	student	Jurnal	asincron
6.	contestare_nota	student	Jurnal	asincron
7.	rezultat_solicitari	student	Jurnal	sincron
8.	stabilire_plan_de_invatamant	șef de catedră	Arhiva	sincron
9.	repartizare_cadre_didactice	șef de catedră	registruActivitatiDidactice	sincron
10.	stabilire_nota	cadru didactic	Arhiva	sincron
11.	stabilire_formatie_de_studiu	secretar	registruActivitatiDidactice	sincron
12.	stabilire_orar	secretar	registruActivitatiDidactice	sincron
13.	stabilire_calendar_examene	secretar	registruActivitatiDidactice	sincron
14.	cereri_in_asteptare_cadru_didactic	cadru didactic	Jurnal	sincron
15.	rezolvare_cereri_cadru_didactic	cadru didactic	Jurnal	asincron
16.	cereri_in_asteptare_secretar	secretar	Jurnal	sincron
17.	rezolvare_cereri_secretar	secretar	Jurnal	asincron

¹⁵ Au fost incluse mesajele din partea componentelor care le transmit inițial, nu și răspunsurile pentru aceste mesaje.

Precizări suplimentare

Mesajele descrise mai sus nu trebuie să aibă neapărat structura specificată însă trebuie implementată funcționalitatea descrisă la nivelul comunicației dintre componente. După cum veți observa pe parcursul dezvoltării temei, arhitectura nu conține toate mesajele necesare transferului de informații, considerând restricțiile de acces la tabelele din baza de date. De aceea, pot fi stabilite și alte tipuri de mesaje, considerându-se decizie de implementare.

Obiectele `ServerSocket` vor fi create în cadrul componentelor de tip server **Arhiva**, **RegistruActivitățiDidactice** și **Jurnal**, iar comunicația dintre componentele de tip client student, șef de catedră, cadru didactic, secretar și acestea va fi încheiată printr-un mesaj de tipul `terminare_comunicatie`.

În cadrul arhitecturii descrise comportamentul componentelor de tip server **Arhiva**, **RegistruActivitățiDidactice** și **Jurnal** este de tip *publisher*, iar comportamentul componentelor student, șef de catedră, cadru didactic și secretar este de tip *subscriber*.

Barem de notare

Punctaj	Criterii de acordare
2 p	proiectarea interacțiunii dintre componentele student, respectiv Arhiva / RegistruActivitățiDidactice <ul style="list-style-type: none">• mesajele student ↔ Arhiva• mesajele student ↔ RegistruActivitățiDidactice• mesajele student ↔ Jurnal
1 p	proiectarea componentei șef de catedră <ul style="list-style-type: none">• comunicația șef de catedră ↔ Arhiva• comunicația șef de catedră ↔ RegistruActivitățiDidactice
2,5 p	proiectarea componentei cadru didactic <ul style="list-style-type: none">• comunicația cadru didactic ↔ Arhiva• comunicația cadru didactic ↔ Jurnal
2,5 p	proiectarea componentei secretar <ul style="list-style-type: none">• comunicația cadru didactic ↔ RegistruActivitățiDidactice• comunicația cadru didactic ↔ Jurnal
1 p	proiectarea componentei Jurnal <ul style="list-style-type: none">• proiectarea unor structuri de date pentru gestiunea solicitărilor transmise de către student• implementarea unor mesaje necesare accesului la informațiile din tabelele bazei de date pentru stabilirea unor erori
1 p	modularizare <ul style="list-style-type: none">• structura aplicației (separarea funcționalităților în pachete și clase distincte)• lizibilitatea codului, nume sugestive pentru obiecte și metode• comentarii, README

BONUS. Se pot obține punctaje suplimentare (care se vor cumula cu punctajele obținute la celelalte teme), astfel:

- 0,50 p – implementarea unui protocol bazat pe limbajul XML pentru comunicația între componente;
- 0,50 p – sincronizarea accesului la resurse pentru componentele server (obiecte care ar putea fi accesate simultan din mai multe fire de execuție corespunzătoare comunicației cu anumite componente client).

Condiții de realizare și predare

Tema va fi realizată individual. Dacă nu ați implementat tema precedentă, puteți folosi cod sursă dezvoltat de un coleg (cu permisiunea acestuia), precizând acest lucru în README. În această situație, prezentarea temei e obligatorie. Trebuie să puteți explica și funcționalitatea codului sursă folosit de la coleg.

La fiecare laborator puteți realiza o anumită parte a temei:

- Laborator 08 – proiectarea componentei Jurnal;
- Laborator 09 – realizarea comunicației dintre componentele student și Arhiva, RegistruActivitățiDidactice și Jurnal folosind socketi TCP;
- Laborator 10 – realizarea comunicației dintre componentele de tip client șef de catedră, cadru didactic și secretar și componentele de tip server Arhiva, RegistruActivitățiDidactice și Jurnal folosind socketi TCP;

După prezentarea temei, va trebui să încărcați pe site o arhivă de tip .zip (cu denumirea Grupa34XCX_NumePrenume_Tema3.zip) care să conțină scriptul cu operațiile asupra baza de date, sursele însoțite de exemple de utilizare (diferite lansări în execuție ale componentelor de tip server și client) precum și un README în care să explicați deciziile de implementare și tehnologiile folosite

Temele vor fi comparate prin diferite aplicații de verificare a plagiatului spre a se depista fraudele. În această situație, punctajul pe parcursul semestrului va fi anulat, urmând să fiți obligați să repetați disciplina anul universitar următor.