

Aplicații Integrate pentru Întreprinderi

Tema 2

Implementarea modulelor de aprovizionare și de verificare a prescripțiilor medicale pentru un sistem informatic din cadrul unei unități farmaceutice

Lansare	Termen de predare	Pondere
22.11.2012, 23:55	23.12.2011, 23:55	10 / 50

Obiective

Scopul laboratorului constă în realizarea unui sistem informatic performant pentru gestiunea resurselor unei întreprinderi de dimensiuni mari. În acest an universitar, organizația pentru care se va dezvolta aplicația ERP este o unitate farmaceutică, dată fiind prevalența acestui tip de instituție în cadrul economiei din România.

Se va urmări implementarea graduală a acestui sistem informatic prin intermediul unor teme de casă, pe parcursul a trei etape.

În cadrul acestei etape, sistemul informatic va fi capabil să interogheze alte aplicații care rulează în alte mașini virtuale și ale căror servicii le cunoaște, având acces la acestea prin intermediul unei rețele de calculatoare.

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 comunicația între componentele distribuite ale aplicației.

Cunoștințele necesare pentru rezolvarea temei de casă sunt:

- programarea în limbajul Java;
- manipularea bazelor de date folosind MySQL;
- folosirea API-ului Java DataBase Connectivity;
- utilizarea tehnologiei RMI¹ pentru apelarea unor metode aparținând unor obiecte existente în altă mașină virtuală.

Enunț

Aplicația pentru gestiunea activității dintr-o unitate farmaceutică dezvoltată va fi completată spre a comunica cu alte sisteme informatice care oferă – prin intermediul rețelei Internet – anumite servicii al căror mod de accesare este cunoscut.

Un **diriginte** poate interoga bazele de date ale unor furnizori pentru vizualizarea produselor pe care aceștia le au disponibile, prețul la care acestea sunt comercializate precum și stocul disponibil în momentul respectiv.

```
public ArrayList<DrugInformation> getDrugInformation  
(String drugName, String drugActiveSubstance, String drugConcentration,  
String drugPresentation);
```

Astfel, lista cu informații despre medicamente poate fi filtrată după unul sau mai multe criterii².

¹ Alternativ se pot folosi tehnologiile CORBA/JAX-WS, definițiile metodelor descrise în cadrul acestui enunț trebuind adaptate corespunzător.

² Dacă se dorește să fie afișată lista cu toate medicamentele disponibile în stocul furnizorului respectiv, toți parametrii metodei `getDrugInformation` vor fi null.

Pentru medicamentele care nu sunt disponibile în stocul furnizorului, dirigintele va realiza o cerere³ referitoare la data calendaristică la care ar putea fi procurat un anumit stoc, precum și prețul solicitat pentru lotul respectiv.

```
public int setDrugRequest(String drugName,  
String drugActiveSubstance, String drugConcentration,  
String drugPresentation, int quantity);
```

Medicamentul poate fi precizat prin mai mulți parametri, fiind obligatorii cel puțin drugName sau drugActiveSubstance precum și drugConcentration.

Metoda va întoarce un număr de ordine⁴ pentru cererea pe care dirigintele a realizat-o, statutul acesteia putând fi interogată mai târziu.

```
public RequestStatus getDrugRequestStatus(int requestId);
```

În clasa RequestStatus se va reține (cel puțin) data calendaristică la care medicamentul poate fi disponibil și prețul unitar pentru produsul solicitat.

Dirigintele va putea transmite o cerere de aprovizionare către un furnizor conținând o listă de medicamente pentru care se precizează termenul de livrare (exprimat ca dată calendaristică) și cantitățile corespunzătoare fiecărui produs în parte.

```
public int setSupplyOrder(ArrayList<Drug> drugs,  
ArrayList<Integer> quantities, GregorianCalendar deliveryDate);
```

Metoda va întoarce un număr de ordine asociat cererii de aprovizionare care va fi folosit ulterior pentru verificarea statutului acesteia.

Statutul unei cereri de aprovizionare poate fi interogată în orice moment, aplicația de la nivelul furnizorului informând dirigintele dacă medicamentele sunt disponibile sau nu.

```
public int getSupplyOrderStatus(int supplyOrderId);
```

Răspunsul indică numărul de medicamente (din lista pe care o conține comanda de aprovizionare) care pot fi livrate la momentul respectiv.

Pentru o comandă al cărei conținut este disponibil, dirigintele poate realiza recepția, medicamentele fiind transferate în stocul unității farmaceutice, statutul comenzii de aprovizionare modificându-se corespunzător (onorată).

```
public boolean setSupplyOrderStatus(int supplyOrderId);
```

Metoda va întoarce un rezultat care va specifica dacă operația a fost realizată cu succes sau nu⁵.

Un **angajat**⁶ al farmaciei va trebui să verifice prescripțiile medicale compensate la casa de asigurări de sănătate pentru care au fost emise, astfel ca acestea să accepte decontările corespunzătoare medicamentelor prescrise, după ce acestea au fost eliberate către pacient.

³ Cererea se poate referi și la mai multe medicamente, astfel că metoda respectivă poate avea și următoarea semnătură:

```
public int setDrugRequest (ArrayList<String> drugName,  
ArrayList<String> drugActiveSubstance, ArrayList<String> drugConcentration,  
ArrayList<String> drugPresentation, ArrayList<Integer> quantity);
```

iar verificarea statutului cererii se va face prin metoda:

```
public ArrayList<RequestStatus> getDrugRequestStatus (int requestId);
```

⁴ Pentru a asigura unicitatea numărului de ordine între diferiți furnizori (astfel încât la nivelul unității farmaceutice un număr de ordine să nu fie asociat la două cereri adresate unor distribuitori de medicamente distincți), acesta se va obține prin concatenarea unui identificator (unic pentru producătorul respectiv) la numărul de ordine propriu-zis.

⁵ Operația va eșua în momentul în care dirigintele va încerca modificarea statutului comenzii de aprovizionare atunci când există medicamente care nu pot fi livrate.

⁶ Astfel de verificări pot fi realizate de către orice angajat (atât de diriginte, cât și de farmacist, respectiv asistent de farmacie).

```
public int checkPrescription (String medicalClinicId, String physicianCode, String patientCode, ArrayList<DrugPrescription> drugPrescription);
```

La nivelul casei de asigurări de sănătate va exista o bază de date conținând toate unitățile medicale, toți medicii și toți pacienții ce au un contract încheiat cu aceasta. Metoda va verifica dacă unitatea medicală care a emis prescripția respectivă este înscrisă în lista furnizorilor de servicii medicale acreditați de casa de asigurări de sănătate respectivă, dacă medicul și-a plătit contribuția la casa de asigurări de sănătate și dacă pacientul este asigurat.

De asemenea, se va verifica dacă medicamentul face parte din lista la care a fost înscris (corespunzătoare unui grad de compensare), precum și adecvarea sa (compatibilitatea sa cu diagnosticul pentru care a fost prescris).

Clasa `DrugPrescription` va conține denumirea substanței active și concentrația, modul de administrare, lista de compensare din care face parte medicamentul respectiv și codul diagnosticului pentru care a fost prescris.

În funcție de rezultatul verificărilor realizat la casa de asigurări, metoda va întoarce următoarele valori:

0	prescripția medicală poate fi decontată
-10	unitatea medicală nu este înscrisă în lista furnizorilor de servicii medicale acreditați de casa de asigurări de sănătate respectivă
-20	medicul nu și-a plătit contribuția la casa de asigurări de sănătate
-30	pacientul nu este asigurat
-n	medicamentul aflat pe poziția n în lista de medicamente conținută de n=1..5
prescripția medicală nu poate fi compensat	

Pentru celelalte prescripții medicale (ce nu presupun vreun regim special de compensare), se va verifica doar faptul că medicul care a eliberat-o are drept de liberă practică, printr-un serviciu pus la dispoziție de Colegiul Medicilor⁷.

```
public boolean checkPhysician (String physicianCode);
```

Metoda va întoarce un rezultat indicând dacă medicul respectiv are sau nu dreptul de a elibera prescripții medicale.

Precizări suplimentare

La nivelul aplicației corespunzătoare furnizorului de medicamente, informațiile cu privire la cererile realizate de diriginte precum și la diferitele comenzi de aprovizionare vor fi reținute într-o bază de date (locală).

Se presupune că fiecare furnizor și fiecare casă de sănătate folosesc aceleași interfețe pentru interogările referitoare la aprovizionarea cu medicamente, respectiv pentru verificarea de prescripții medicale.

Când se lansează în execuție o aplicație de tip furnizor, respectiv de tip casă de asigurări de sănătate, se va specifica identificatorul acesteia. Aplicația unității farmaceutice va gestiona mai multe instanțe ale sistemelor informatice corespunzătoare furnizorilor, respectiv caselor de asigurări de sănătate.

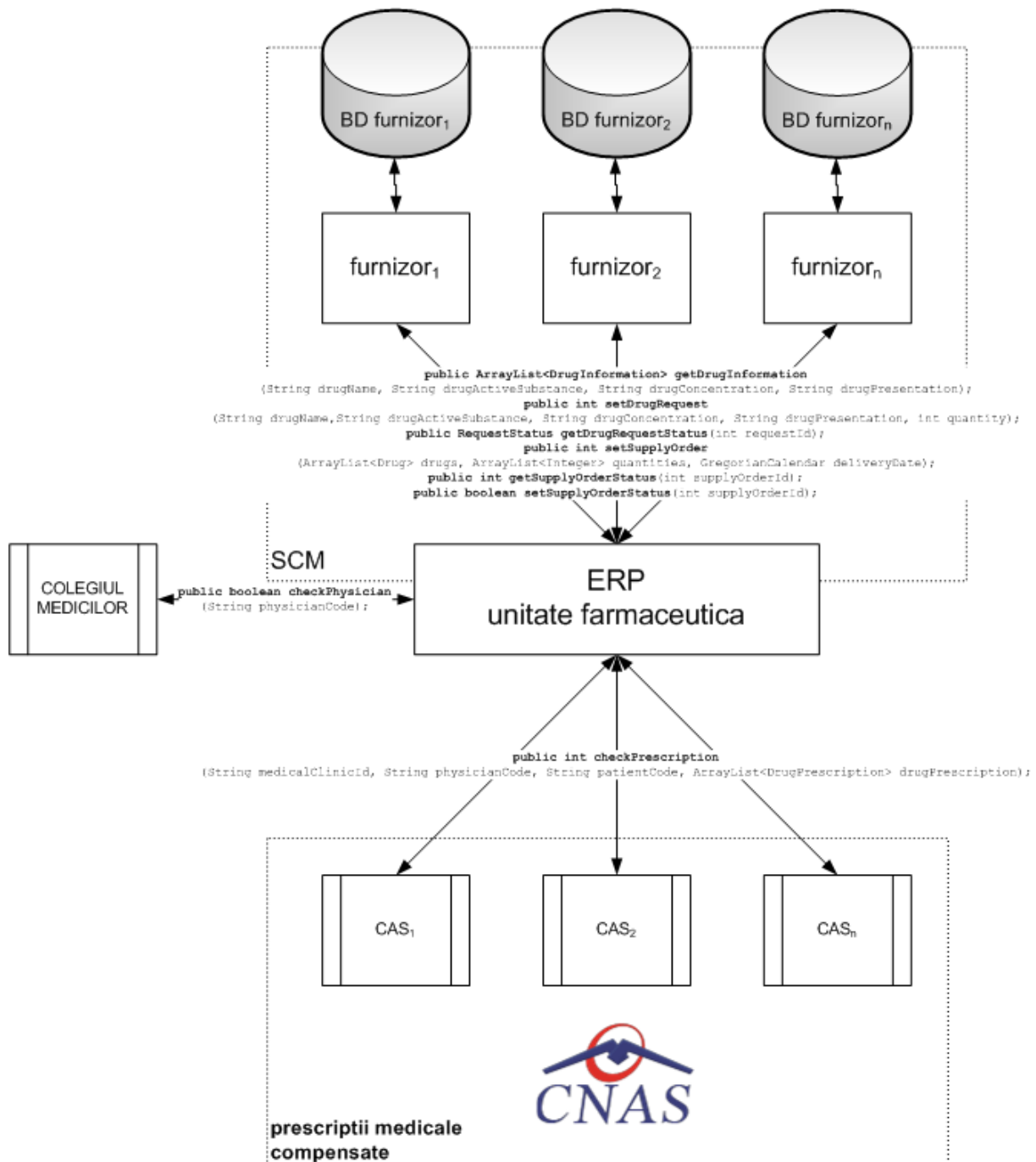
Modificarea statutului unei cereri de medicamente, respectiv a unei comenzi de aprovizionare se va realiza în aplicațiile furnizor de către utilizator, fără a fi necesară însă dezvoltarea unei interfețe grafice în acest caz.

⁷ La nivelul aplicației dezvoltate de către Colegiul Medicilor există implementată o bază de date care conține informații despre toți medicii care au aviz de liberă practică (nume și prenume medic, cod parafă, gradul profesional (medic, medic rezident, medic specialist, medic primar), specializarea (medicina internă, cardiologie, gastroenterologie, neurologie, psihiatrie, chirurgie, ortopedie-traumatologie), titlul universitar (N/A, asistent, sef de lucrari, conferentiar, profesor)).

În cazul în care un medicament este identificat prin substanța activă, respectiv concentrația, metoda `getDrugInformation` va furniza informații referitoare la toate medicamentele care satisfac criteriile respective.

În momentul în care se face recepția unei comenzi de aprovizionare, stocurile medicamentelor respective vor fi modificate corespunzător la nivelul bazei de date a furnizorului respectiv.

Orice specificație care nu este menționată mai sus se constituie în decizie de implementare. Puteți considera orice simplificare în condițiile în care enunțul nu precizează altfel.



Arhitectura de nivel înalt a sistemului informatic

Barem de corectare și notare

Punctaj	Criterii de acordare
1 p	proiectarea interfețelor corespunzătoare componentelor de tip server <ul style="list-style-type: none">• componenta Furnizor (70%)• componenta Casa de Asigurări de Sănătate (15%)• componenta Colegiul Medicilor (15%)
5 p	implementarea metodelor corespunzătoare componentei Furnizor <ul style="list-style-type: none">• getDrugInformation (20%)• setDrugRequest (10%)• getDrugRequestStatus (5%)• setSupplyOrder (20%)• getSupplyOrderStatus (5%)• setSupplyOrderStatus (10%)• proiectarea bazei de date de la nivelul modulului respectiv (30%)
1 p	implementarea metodelor corespunzătoare componentei Casa de Asigurări de Sănătate <ul style="list-style-type: none">• checkPrescription (70%)• proiectarea bazei de date de la nivelul modulului respectiv (30%)
1 p	implementarea metodelor corespunzătoare componentei Colegiul Medicilor <ul style="list-style-type: none">• checkPhysician (70%)• proiectarea bazei de date de la nivelul modulului respectiv (30%)
1 p	apelarea metodelor din componentele de tip server din componentele de tip client
1 p	modularizare <ul style="list-style-type: none">• structura aplicației (50%)• lizibilitatea codului (20%)• comentarii, README (30%)

BONUS. Se pot obține punctaje suplimentare, astfel:

- 0,25 p – predarea și prezentarea temei până la 16.12.2012, ora 23:55;
- 0,50 p – să se modifice comportamentul metodei `getDrugInformation` ca aceasta să verifice informațiile de la nivelul tuturor furnizorilor identificând producătorul ce poate oferi medicamente la prețul cel mai bun respectiv cea mai mare cantitate de medicamente⁸; astfel, pentru fiecare medicament din lista întoarsă ca rezultat al metodei se va preciza și furnizorul care îl comercializează;
- 0,25 p – în cazul în care prescripțiile medicale compensate nu au putut fi validate, să se indice pacientului intervalul de prețuri corespunzător achiziției medicamentelor respective cu preț întreg⁹.

Condiții de realizare și predare

Tema va fi realizată individual și va fi prezentată în cadrul laboratorului, până la sfârșitul semestrului. Neprezentarea temei de casă atrage după sine nepunctarea acesteia. În cadrul prezentării veți specifica succint funcționalitățile pe care le-ați dezvoltat și veți răspunde la întrebări cu privire la diferite soluții adoptate în rezolvarea problemelor întâlnite.

⁸ Criteriul în funcție de care va fi selectat furnizorul de medicamente poate fi indicat ca parametru al funcției respective.

⁹ Costul cel mai mic va fi calculat ca sumă a prețurilor celor mai ieftine medicamente conținând substanța activă respectivă, iar costul cel mai mare ca sumă a celor mai scumpe medicamente având substanța activă în concentrația indicată.

După prezentarea temei, va trebui să încărcați pe site o arhivă de tip .zip (cu numele Grupa34XCX_NumePrenume_Tema2.zip) care să conțină script-ul pentru crearea tabelor din baza de date (numele bazei de date trebuie să aibă forma Grupa34XCX_NumePrenume), sursele aplicației cu exemple de utilizare (lansări în execuție a componentelor de tip server și a componentelor de tip client cu diferite operații, specificând proprietățile Java necesare¹⁰ și fișierele conținând politica de securitate), precum și un README în care să explicați soluția aleasă și tehnologiile folosite. Prezentarea se face după ce tema a fost încărcată pe site.

Încărcarea pe site nu este redundantă, temele vor fi comparate prin aplicații specializate pentru a se depista eventualele fraude. În această situație, întreg punctajul pe parcursul semestrului va fi anulat, fiind obligați să repetați disciplina – cu taxă – în anul universitar următor.

¹⁰ Este vorba despre `java.rmi.server.codebase`, `java.rmi.server.hostname`, `java.security.policy`.