Aplicații Integrate pentru Întreprinderi

Tema de Recuperare
Implementarea modulelor de aprovizionare, financiar şi de vânzare din cadrul unei aplicații integrate pentru întreprinderi

Lansare	Termen de Predare	Pondere
27.02.2012, 23:55	23.04.2012, 23:55	20 / 70

Obiective

Scopul laboratorului constă în realizarea unui sistem informatic performant pentru gestiunea resurselor unei întreprinderi de dimensiuni mari.

În cadrul temei de recuperare veți implementa modulele responsabile de aprovizionare şi de relația cu clienții din cadrul unui sistem ERP[1].

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

- proiecteze o bază de date cu mai multe tabele normalizate, între care există relații de dependență, gestionând informațiile reținute prin intermediul unei aplicații;
- proiecteze o aplicație având o structură distribuită pe componente putând rula pe maşini diferite;
- realizeze o interfață grafică pentru a facilita interacțiunea utilizatorului cu aplicația;
- realizeze o aplicație web (formată din mai multe pagini, între care există legături) care implementează o funcționalitate complexă, având rolul de interfață între utilizator şi baza de date.

Cunoştințele necesare pentru rezolvarea temei de casă sunt:

- programarea în limbajul Java;
- manipularea bazelor de date folosind MySQL;
- folosirea API-ului Java DataBase Connectivity;
- dezvoltarea de interfețe grafice utilizând Swing;
- utilizarea tehnologiei RMI pentru apelarea unor metode aparținând unor obiecte „la distanță”;
- folosirea tehnologiilor Java Servlets / JSP (Java Server Pages) pentru dezvoltarea de aplicații web.

Enunț

Se cere să se dezvolte trei aplicații, una responsabilă de aprovizionarea dintr-o întreprindere, sub forma unui sistem informatic cu interfață grafică (Swing) care va apela nişte metode la distanță, una responsabila de contabilitatea organizației (sistem informatic cu interfață grafică) şi una responsabilă de procesul de vânzare sub forma unei aplicații web care poate fi accesată dintr-un client universal, browser-ul.

1. **Departamentul de aprovizionare** din cadrul organizației analizează prin intermediul unei aplicații cu interfață grafică comenzile venite din partea clienților. Acestea conțin o listă formată din mai multe produse precum şi cantitatea din fiecare produs în parte.

[1] După unele teorii mai noi, modulele de tipul Supply Chain Management (SCM) respectiv Customer Relationship Management (CRM) nu fac parte din sistemul ERP propriu-zis, ci interacționează cu aceasta, fiind aplicații de sine stătătoare.

Se verifică dacă aceste produse sunt disponibile în stoc şi în acesta situație este emisă o factură (ce va avea proprietatea „neachitată"), cantitățile respective fiind actualizate corespunzător în stoc. Se va emite o factură care va trebui plătită de client în decurs de 14 zile. În cazul în care acest lucru nu se întâmplă, reprezentantul departamentului de vânzare poate trimite o notificare clientului sau poate anula factura[2].

În cazul în care unele produse dintre cele comandate nu sunt disponibile, reprezentantul departamentului de aprovizionare va utilizeaza un serviciu, care îi oferă alternative cu privire la procesul de achiziție a produselor necesare.

Serviciul de Aprovizionare este o componentă de tip server, conectată la mai multe baze de date[3], toate având aceeaşi structură, corespunzătoare diferiților parteneri cu care lucrează organizația în cauză. Acesta oferă mai multe metode[4] referitoare la produsele pe care le comercializează:

a. informații despre disponibilitatea unor produse:

```
public ArrayList<Oferta> cautareProdus (int codProdus);
```

Se consideră ipoteza simplificatoare că fiecare furnizor de produse foloseşte aceeaşi convenție pentru codificarea produselor (un produs este identificat prin `codProdus` la nivelul tuturor acestor organizații).
Serviciul de Aprovizionare va realiza interogări pentru fiecare bază de date corespunzătoare unui furnizor şi va oferi lista organizațiilor care pot oferi acest produs.
O ofertă este caracterizată prin codOrganizație, prețUnitar, stocExistent, locațieDepozitAprovizionare[5], timpLivrare[6].

b. achiziționarea unor produse:

```
public double cumparareProdus (ArrayList<Cerere> comanda);
```

O cerere este definita de codOrganizație, codProdus, cantitateSolicitată.
Există posibilitatea ca acelaşi produs să fie achiziționat de la
Serviciul de Aprovizionare va actualiza corespunzător stocurile furnizorilor
Există posibilitatea ca nu toate produsele să fie disponibile în cantitatea care este cerută[7]. În acest caz se va onora toată cantitatea existentă în stoc la momentul respectiv.
Metoda va întoarce prețul total pe care organizația trebuie să îl plătească pentru produsele care vor fi livrate. Acest preț va fi specificat în bilanțul organizației la capitolul cheltuieli. Stocurile se vor actualiza corespunzator

[2] În acest caz, factura va avea proprietatea „anulată", iar produsele vor fi restituite în stoc.
[3] Se pot reține într-un fişier de tip text informațiile despre bazele de date la care se conectează serviciul de aprovizionare. Mai simplu, se poate preciza doar şirul de conectare pentru fiecare bază de date (de forma `protocol:subprotocol:<nume_bazadedate>[lista_de_proprietati]`, vezi Laborator 3, p. 6).
[4] Informații cu privire la implementarea metodelor „la distanță" folosind tehnologia RMI găsiți în Laboratorul 5, pp. 7-13.
[5] Locația depozitului de aprovizionare este un şir de caractere care specifică adresa la care sunt disponibile produsele din punct de vedere fizic. Astfel de informații pot fi necesare pentru a stabili modul de transport relativ la timpul de livrare cu relevanță pentru starea în care produsul ajunge la destinație (în cadrul produselor perisabile sau a produselor fragile, spre exemplu).
[6] Timpul de livrare va fi exprimat în zile calendaristice.
[7] De la momentul în care s-au solicitat informații despre produse până la realizarea comenzii, şi alte organizații pot solicita furnizarea de produse.

Servicul de Aprovizionare va putea fi apelat şi de componente de tip client care pot adăuga, edita sau şterge înregistrări din baza de date corespunzătoare unei organizații. Metodele oferite în acest sens vor fi:

```
public int adaugareProduse (ArrayList<Produs> produse);
public int editareProdus (ArrayList<Produs> produse);
public int stergereProdus (ArrayList<Produs> produse);
```

Un produs este caracterizat prin codOrganizatie (desemnând baza de date unde vor fi realizate operațiile), codProdus, pretProdus, cantitateProdus, locatieDepozitAprovizionare, timpLivrare[8].
Metodele vor întoarce numărul de operații care au fost realizate cu succes.

În cazul în care s-a reuşit procurarea tuturor produselor, factura va fi emisă cu data la care toate produsele vor fi disponibile în stoc (pe baza informațiilor referitoare la disponibilitate[9]).

Dacă unele produse sunt indisponibile temporar şi la furnizori, reprezentantul departamentului de vânzare va informa printr-un mesaj clientul, existând posibilitatea de a elimina din comandă produsele care nu sunt disponibile sau de a anula comanda.

În momentul în care factura este achitată de client, suma respectivă este specificat în bilanțul organizației la capitolul venituri.

2. **Departamentul financiar** din cadrul organizației va putea vizualiza, în orice moment, profitul ca diferență dintre venituri şi cheltuieli:

$$profit = \sum_{i \text{ in facturi_achitate}} facturi_i - \sum_{j \text{ in comenziaprovizionare_achitate}} comenzi_aprovizionare_j$$

Acesta va avea forma unui tabel de forma următoare:

VENITURI	CHELTUIELI	Observații
10.000		factura nr. 1234 / 27.02.2012
	10.000	comanda de aprovizionare nr. 5678 / 23.04.2012

Se consideră ipoteza simplificatoare că organizația nu are alte activități care să genereze venituri sau cheltuieli.

3. **Departamentul de vânzare** va pune la dispoziție clienților organizației o aplicație web (dezvoltată cu Java Servlets sau Java Server Pages[10]) prin care aceştia pot cumpăra produsele pe care aceasta le comercializează.

Clienții se vor autentifica în urma unui proces de înregistrare în care specifică nume, prenume, adresa la care doresc să se realizeze livrarea, telefonul la care pot fi contactați şi opțional, un email.

După autentificare, clienții vor vizualiza produsele împărțite pe categorii. Clientul va avea posibilitatea de a ordona produsele în funcție de preț.

Pentru fiecare produs în parte se va afişa denumirea, descrierea împreună cu o imagine reprezentativă[11], prețul şi disponibilitatea[12].

[8] Pentru operația de adăugare trebuie specificate toate aceste informații. În cazul în care exista deja un produs cu acelaşi `codProdus` pentru organizația respectivă, operația nu se va realiza. Pentru operația de editare sunt obligatorii câmpurile `codOrganizație` / `codProdus` şi cel puțin unul din celelalte câmpuri (necompletarea celorlalte atribute implică menținerea lor neschimbată). Pentru operația de ştergere sunt obligatorii doar câmpurile `codOrganizație` / `codProdus`.
[9] Se va lua în considerare produsul care ajunge cel mai târziu în stoc.
[10] Informații cu privire la dezvoltarea aplicațiilor web folosind tehnologiile Java Servlets sau JSP (Java Server Pages) găsiți în laboratoarele 9-10.

Totodată, aplicația va oferi sugestii de cumpărare clienților astfel:

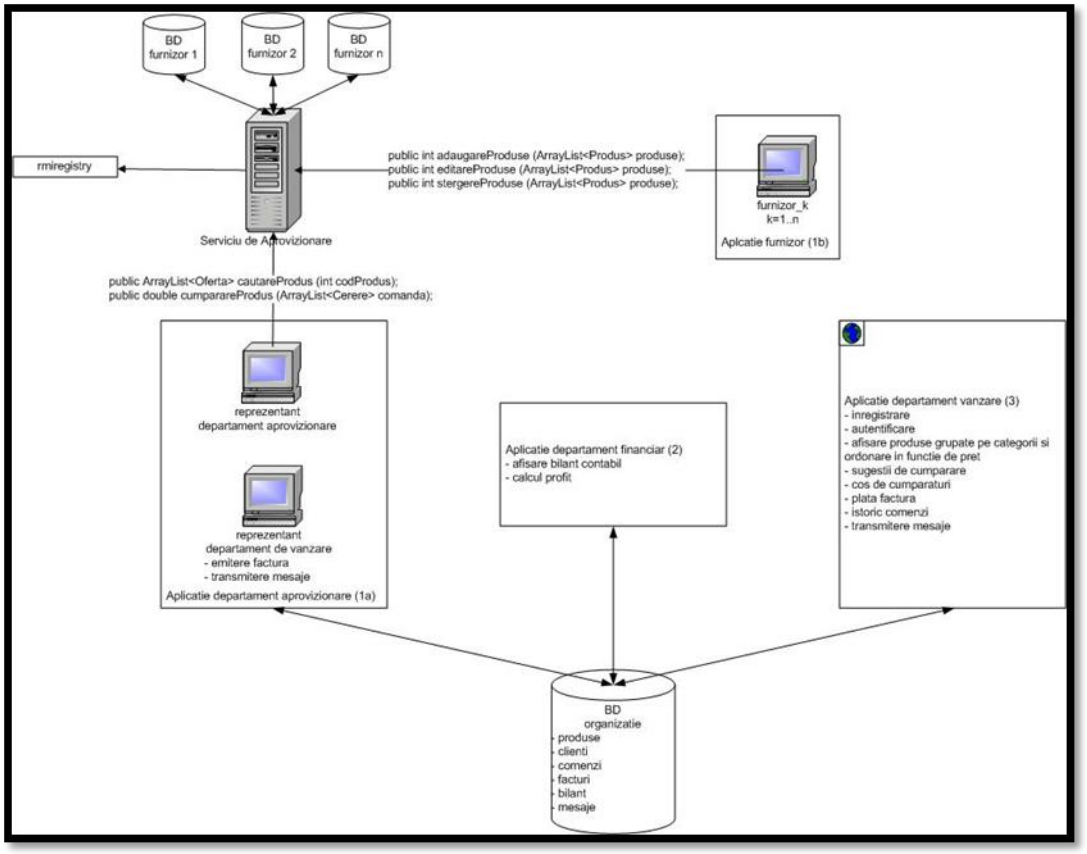
- pentru clienții noi (care nu au mai făcut cumpărături) se vor indica produsele cele mai populare ale organizației (care au înregistrat numărul cel mai mare de vânzări);
- pentru clienții care au mai realizat achiziții, se vor indica acele produse prezente cu cea mai mare frecvență în coşurile de cumpărături ale altor clienți care au achiziționat şi produsele cumpărate de clientul respectiv.

Pentru fiecare produs pe care îl achiziționează, clienții vor specifica şi cantitatea dorită. Acesta va fi trecut în coşul de cumpărături.

În momentul în care au fost specificate toate produsele care se doresc a fi achiziționate, comanda se va putea finaliza, afişându-se prețul total.

Clientul poate vizualiza starea comenzii care poate fi neanalizată (nu a fost vizualizata de reprezentantul departamentului de vânzări) sau analizată – afişându-se data de la care produsele vor fi disponibile. În acest moment, clientul poate achita factura[13].

Totodată, clientul are la dispoziție şi istoricul comenzilor în care vizualizeaza produsele pe care le-a cumpărat, valoarea facturii şi data la care aceasta a fost emisă.



Arhitectura de nivel înalt a aplicației

[11] În acest caz, în baza de date se va reține locația la care este stocata imaginea corespunzătoare produsului respectiv.

[12] Pentru produsele în care stocul este mai mare de 5 se va specifica „în stoc" , pentru produsele cu mai puțin de 5 produse în stoc se va indica „stoc limitat", iar pentru produsele care nu există (temporar) în stoc se va indica „indisponibil momentan".

[13] Operația de achitare a facurii implică modificarea stării ei din „neachitat" în „achitat" precum şi marcarea sumei respective ca venituri în bilanțul organizației.

Precizări suplimentare

NU se permite folosirea altor tehnologii în afara celor specificate în enunț. Ca server web veți folosi Apache Tomcat.

Nu este necesar să implementați un mecanism de autentificare pentru aplicația de aprovizionare Astfel interfața pentru reprezentanții departamentelor de aprovizionare şi de vânzare poate fi comună.

Componenta Serviciu Aprovizionare NU specifică în politica de securitate contextul din care sunt apelate metodele pe care le pune la dispoziție.

Serviciul de Aprovizionare trebuie să comunice cu cel puțin 2 furnizori.

Comunicarea între reprezentanții departamentului de vânzare şi clienți este realizată prin intermediul unui sistem de mesagerie implementat în cadrul aplicațiilor corespunzătoare.

Pentru un produs, prețul de vânzare (cel pe care îl vor plăti clienții) este diferit de prețul de cumpărare (platit către furnizori), calculându-se prin adăugarea TVA-ului (24%) precum şi a unui comision (10%).

Barem de corectare

Punctaj	Criterii de acordare
3 p	proiectarea bazelor de date - baza de date furnizori (10%) - baza de date organizație (40%) - definire chei primare (10%) - relații între tabelele bazei de date (10%) - populare tabele (20%) - conformitatea cu forma normală 3 (10%)
1 p	proiectarea interfetelor de comunicare - definirea interfetelor Remote pentru aplicatie departament aprovizionare, respectiv furnizori (50%) - definirea claselor Serializable (Oferta, Cerere, Produs)
4 p	aplicația departamentului de aprovizionare - interfata grafica (35%) - analiză comenzi (10%) - investigare produse disponibile în Serviciu de Aprovizionare (10%) - cumpărare produse prin Serviciu de Aprovizionare (20%) - emitere factură (15%) - comunicare cu clienții (10%)
2 p	aplicația furnizor - interfața grafică (40%) - adăugare produse (20%) - editare produse (20%) - ştergere produse (20%)
1,5 p	aplicația departamentului financiar - interfața grafică (50%) - afişarea bilanțului contabil (35%) - calculul profitului (15%)
7 p	aplicația departamentului de vânzare - înregistrare client (10%) - autentificare (10%) - afişare produse pe categorii şi ordonare în funcție de preț (20%) - sugestii de cumpărare (25%) - coş de cumpărături (20%) - plata factură (5%) - istoric comenzi (5%) - transmitere mesaje (5%)

0,5 p	validarea datelor - verificarea că operațiile nu influențează consistența informațiilor reținute în baza de date (50%) - mesaje de eroare (50%)
1 p	modularizare - separarea logicii aplicației de modulul de prezentare, împărțirea corespunzătoare pe pachete (60%) - lizibilitatea codului – nume corespunzătoare pentru metode şi obiecte (15%) - respectarea convențiilor de nume (10%) - fişier README: funcționalități implementate, descrierea soluției (15%)

Temele cu probleme de compilare sau nefuncționale nu se punctează.

Pot trimite această temă studenții care au obținut minim 15 puncte la activitatea pe parcursul semestrului.

Punctajul se va calcula după următoarea formulă:

PT = MAX (AL + TG, 20) + MAX (TC + TR, 50)

unde

PT = punctaj total

AL = nota la activitatea de laborator (~ 12 p)

TG = nota la testul grilă (maxim 12 p)

TC = nota la temele de casă 1-4 (~ 50 p)

TR = nota la tema de recuperare (maxim 20 p)

Studenții care după corectarea temei de recuperare vor avea un punctaj mai mic de 35 de puncte vor trebui să refacă laboratorul aferent disciplinei *Aplicații Integrate pentru Întreprinderi* în anul universitar următor. Totodată, punctajul obținut în cadrul acestui an universitar nu va fi echivalat.

Condiții de realizare şi predare

Tema va fi realizată individual şi va trebui prezentată. Nu se acceptă trimiterea temei după termenul de predare.

De asemenea, va trebui să încărcați pe site o arhivă tip .zip (cu numele Grupa34XCX_NumePrenume_TemaRecuperare.zip) care să conțină script-ul pentru crearea şi popularea tabelelor din baza de date (numele bazelor de date trebuie să aibă forma Grupa34XCX_NumePrenume_Furnizor_k, unde k=1..n, respectiv Grupa34XCX_NumePrenume_Organizatie), sursele aplicației cu exemple de utilizare (lansarea în execuție a componentei Serviciu Aprovizionare şi a componentei de tip client cu diferite operații, specificând proprietățile Java necesare[14] şi fişierele conținând politica de securitate) şi un fişier README unde să explicați soluția aleasă.

[14] Este vorba despre `java.rmi.server.codebase`, `java.rmi.server.hostname`, `java.security.policy`.