

## Tema de laborator, restanta

Pentru cei care nu au situatia de laborator la materia "Arhitectura sistemelor software Enterprise. Platforma .NET" incheiata, in sesiunea de restante din toamna 2021 se va prezenta tema de mai jos.

Sa se creeze o aplicatie in care se permite licitarea pentru produse vandute de catre utilizatori. Produsele sunt caracterizate prin apartenenta la categorii, acestea putand face parte din alte categorii etc. (ex: electronice <- aparatura foto <- aparat foto concret). **Se va modela situatia in care o categorie poate sa aiba mai multi parinti – conditie de luare in considerare a temei.**

Se cere implementare care sa trateze urmatoarele:

1. Pentru un produs se specifica data de incepere si data de finalizare a licitatiei; se va valida ca data inceperii < data finalizarii, **dar diferite sa fie de maxim 7 luni de zile**, sau ca data inceperii nu poate sa fie in trecut (relativ la momentul prezent); impuneti alte validari ale valorilor utilizate.

2. Pentru fiecare produs pentru care se incepe licitatie se specifica un pret de pornire; se vor rejecta preturi de pornire mai mici decat un prag, predefinit ca o constanta a aplicatiei; la inceputul licitatiei se va specifica o moneda de tranzactie si toate celelalte licitati pentru obiectul curent trebuie sa se faca in aceasta moneda (**validati sau modelati corespunzator**). Pentru o licitare, pretul oferit la momentul  $n+1$  trebuie sa fie strict mai mare decat pretul de la momentul  $n$ , **dar cel mult cu 10% mai mult fata de pretul de la momentul  $n$** . Impuneti alte validari asupra datelor, demonstrati prin testare automata implementarea.

3. Finalizarea unui proces de licitatie se face

- fie prin atingerea datei declarate pentru finalizare
- fie prin incheierea la dorinta de catre cel care a depus obiectul la licitare (o persoana nu poate sa incheie o licitatie pe care nu a demarat-o);

Pentru o licitatie incheiata nu se mai poate modifica nimic. Impuneti validari asupra fluxului de operatii si acoperiti cu teste.

4. O persoana care depune obiecte spre a fi licitate nu poate sa aiba la un moment dat mai mult de  $k$  licitatii incepute si nefinalizate (finalizare = incheierea perioadei de finalizare sau terminarea explicita de catre ofertantul obiectului). Valoarea  $k$  este constanta a aplicatiei ce se va citi din baza de date sau din fisier de configurare. Modificarea ei nu trebuie sa duca la recompilarea si re-deploy de aplicatie.

5. Nu se permite ca o persoana sa aiba la un moment dat mai mult de  $m$  licitatii demarate si nefinalizate in cadrul unei categorii. Se permite deschiderea unei alte licitatii in acea categorie doar momentul in care numarul celor active (deschise de utilizator) este sub pragul  $m$ .

6. Pentru fiecare persoana care foloseste platforma se asigneaza un scor, reprezentand seriozitatea; scorul este cu o valoare initiala  $S$ , pana la prima asignare de scor facuta de alt utilizator, moment dupa care se folosesc doar scorurile date de utilizatori; pentru cel putin un scor asociat unui utilizator, scorul total este media ultimelor cel mult  $n$  scoruri ( $n$  citit din fisier de configurare sau baza de

date,  $n \geq 1$ ). Dacă scorul unei persoane scade sub un prag prestabilit  $p$ , atunci persoana va fi împiedicată să deschidă noi licitații timp de  $z$  zile ( $p$  și  $z$  preluate din fișier de configurare sau baza de date).

Roluri (neexcluzive mutual, se preiau la crearea contului sau ulterior):

1. ofertant de obiect

2. licitant

Cerinte:

1. Unit testing folosind NUnit, xUnit sau Visual Studio 2019+; minim 210 de metode de testare (**conditie de luare in considerare a temei**); se vor testa dacă valorile proprietatilor au valori adecvate, restricțiile de mai sus și altele de validare a corectitudinii rezultatelor și fluxului de lucru.

2. Folosire de blocuri din suita Microsoft Enterprise Library sau echivalente: logging (se poate folosi alternativ log4net), validation application block (sau se poate scrie cod de validare manual, sau se poate folosi o bibliotecă de tipul <https://github.com/JeremySkinner/FluentValidation>), optional: exception handling application block, optional: security application block. Nu se face logging în unit testing (se permite în straturile aplicației, însă). **Validarea stării obiectelor prin teste este condiție obligatorie pentru luarea în considerare a temei. Logging-ul este obligatoriu pentru luarea în considerare a temei.**

3. Dezvoltare pe straturi (layers): Domain layer și Data Layer (**conditie de luare in considerare a temei**); nu se ia în considerare/nu se cere user interface. Se pot folosi alte pattern-uri de design, dacă se considera necesar (DTO, remote facade etc.).

4. Baza de date relatională.

5. Domain Layer implementat prin Domain Model; Data Layer implementat prin Data Mapper (**conditiile de la acest punct sunt obligatorii**).

8. Code coverage de minim 90% pentru codul de Domain Model și minim 90% pentru Service Layer (**conditie de luare in considerare a temei**).

9. Mocking (**conditie de luare in considerare a temei**).

10. Comentarii, minim pentru metode, constructori, proprietati, indexatori, destructori, tipuri imbricate. Puteti folosi GhostDoc.

11. Se va face analiza codului sursă folosind StyleCop (plugin de Visual Studio; versiune recomandată 4.7.x). Toate regulile (tab-ul Rules) trebuie bifate; în tab-ul Company Information la secțiunea „Company Name” să figureze „Transilvania University of Brasov”, iar la Copyright numele studentului. Maxim 30 de avertismente sunt permise (**conditiile de la acest punct sunt obligatorii**).

Puteti folosi pluginuri auxiliare de forma CodeItRight.

Notarea aplicației se va face prin rularea unitatilor de test scrise. Unitatile de test vor include validarea stării obiectelor (poate apărea validare selectivă, în anumite contexte doar anumite proprietati trebuie să fie satisfăcute) și verificarea coerenței la nivel de service layer (dacă parametrii trimisi unor metode

de serviciu nu sunt legali, e de asteptat ca metoda sa arunce exceptie sau sa returneze un rezultat din care se deduce faptul ca nu se poate indeplini cererea); validati fluxul de mesaje, starea curenta si contextual; urmariti cerintele si precizarile din enuntul temei.

Studentii se pot consulta in legatura cu tema, dar munca va fi individuala. Implementarile pentru care exista suspiciune de fraudă vor fi notate cu 1.