

Laborator .NET - Evaluarea finala.

Enunt problema

Proiectati o baza de date care sa modeleze aplicatii de tip "Teste online" si aplicatiile necesare utilizarii acestei baze de date.

Aplicatiile care se vor realiza, folosind aceasta baza de date, vor trebui sa ofere urmatoarele functionalitati:

1. Introducerea disciplinelor pentru care se doreste evaluarea. Fiecare disciplina va fi impartita in capitole pentru a realiza teste specifice. Exemplu: Disciplina "Topici speciale .NET" va avea capitolele: Entity Framework, WCF, WPF, ASP.NET, ASP.NET cu MVC, etc. In aceste conditii pot sa aleg sa sustin un test la "Topici speciale ..." , capitolul WCF. In sustinerea unui test sa pot selecta mai multe capitole. Un test este compus din mai multe intrebari.
2. Gestionarea cursantilor unei anumite discipline. Fiecare cursant va avea un cont propriu si o parola pentru autentificare.
3. Lectorul cursului sa aiba posibilitatea crearii unei grupe pentru un anumit test. De exemplu azi vor da testul cursantii X, Y si Z. Ceilalti cursanti neavand acces la modulul de sustinere teste.
4. Un test va avea o perioada de timp specificata si care va fi monitorizata de aplicatie. La terminarea timpului aplicatia poate afisa doar rezultatele testului. Salvarea rezultatelor unui test in baza de date cade in sarcina cursantului folosind elemente din interfata aplicatiei (butoane de tip Save, etc.).
5. Baza de date va contine teste pentru fiecare disciplina si capitol. Fiecare intrebare va contine si raspunsurile corecte precum si motivarea acestor raspunsuri.
6. Testele vor fi de tipul: intrebari cu mai multe raspunsuri precum si motivarea raspunsurilor selectate si neselectate.
7. Testele vor fi generate aleator din baza de date pentru fiecare cursant in parte si vor fi memorate pentru a realiza un istoric al testelor. Cursantul va trebui sa aleaga unul din testele predefinite de catre lector. Un test va avea o descriere, de exemplu "Test WCF si EF" si va contine capitolele preselectate de catre lector; va avea un identificator unic sub care se vor gasi toate testele sustinute de cursanti. Cursantii vor selecta acest tip de test si nu vor putea face modificari. Se vor genera (selecta) intrebari din capitolele selectate. Numarul intrebilor din fiecare capitol trebuie sa fie acelasi sau plus/minus o intrebare. Un algoritm simplu de stabilire al numarului de intrebari ar fi sa impartim numarul total de intrebari al testului la numarul de capitole si sa ajustam in plus sau minus atunci cand impartirea nu se face exact. **Numele testelor, capitolelor si numarul de intrebari al unui test vor fi stabilite de catre lector printr-o functionalitate oferita de aplicatie.**
8. Trebuie sa existe posibilitatea de a fixa numarul intrebilor continute in fiecare test. Acest lucru nu trebuie sa fie facut de cursant.
9. Vor exista urmatoarele niveluri de acces la aplicatie:
 - a. Utilizator conectat pentru a realiza teste demonstrative, in sensul de a se obisnui cu aplicatia si cu intrebarile din cadrul testului. Toate testele demonstrative sustinute vor fi pastrate in baza de date.

- b. Utilizator pentru a realiza un test real. Mare atentie la modul de autentificare al utilizatorului.
 - c. Utilizator cu drepturi depline ce are posibilitatea de a crea discipline, capitole, cursanti, intrebari, de a evalua testele reale sustinute de cursanti. Posibilitatea modificarii intrebarilor si a raspunsurilor. In momentul crearii unei intrebari sa existe posibilitatea de seta numarul raspunsurilor. Fiecare intrebare va avea minim trei raspunsuri si maximum cinci. Motivarea raspunsurilor va fi continuta intr-un singur text in care se va specifica ceva de genul: R1 nu este posibil deoarece..., R3 este corect deoarece ..., etc.
10. La testele demo trebuie sa existe posibilitatea de afisare a raspunsului cursantului precum si a raspunsului inregistrat in baza de date, raspuns care se considera ca e cel corect. Imaginati-va un test pentru dobandirea carnetului de conducator auto: raspund la intrebari si apoi verific cu raspunsurile oficiale. In situatia noastra doresc ca raspunsurile cursantilor si cele oficiale sa fie date "in oglinda", adica am raspuns cursant si alaturi sau mai jos raspunsul oficial.
 11. Baza de date va fi SQL Server (versiune 2012, 2014, ?).
 12. Vetii folosi Entity Framework.
 13. Vetii folosi servicii WCF. Indiferent de tipul aplicatiei, accesul la baza de date va fi furnizat de un serviciu WCF.
 14. Aceasta baza de date va fi folosita de aplicatii de tip WEB si WPF si va ingloba tehnologia WCF. Aplicatia de tip WEB trebuie sa ruleze in IIS. Aplicatia de tip WPF trebuie sa se comporte ca o aplicatie client-server, mai multi clienti un singur server. Atentie la configurari. Vetii folosi fisiere de configurare acolo unde este posibil. Trebuie sa evitam configurari in cod (hard coded).
 15. O intrebare din cadrul unui test (real sau demo) nu va fi salvata in baza de date daca nu se completeaza motivarea raspunsurilor. Mesaj de atentionare.
 16. Dupa sustinerea unui test real, lectorul sa aiba posibilitatea de a evalua fiecare test al fiecarui cursant si de a valida sau nu raspunsurile la intrebari precum si de a acorda un punctaj altul decat cel stabilit pentru fiecare intrebare. De exemplu la o intrebare se prezinta trei raspunsuri posibile: R1, R2 si R3. Raspunsul corect este R1. Cursantul alege raspunsul corect dar motivarea pentru R1 este eronata in timp ce pentru R2 si R3 este corecta. In acest caz lectorul ii va acorda un punctaj pentru R2 si R3, punctajul depinzand de modul de evaluare anuntat la inceputul testului, sau prezentat intr-o fereastra (pagina) a aplicatiei. Ideea este de a putea face si evaluari parțiale si nu de tipul "totul sau nimic".
 17. Pentru fiecare capitol: EF, WCF, ASP.NET, ASP.NET si MVC, WPF vetii realiza minimum 15 intrebari intrebari ce vor contine si motivatiile corespunzatoare. In motivatie nu scriu numai de ce am ales un anumit raspuns / anumite raspunsuri, scriu si de ce nu sunt corecte celelalte raspunsuri pe care nu le-am selectat.
 18. Cursantul ce sustine un test trebuie sa aiba posibilitatea de a reveni la raspunsurile unei intrebari anterioare si a o modifica, in baza de date salvandu-se ultima modificare nu si incercarile anterioare.
 19. Aplicatiile sa ofere posibilitatea editarii unei situatii a cursantilor (cu punctajele obtinute) pentru fiecare tip de test.
 20. Aplicatiile sa raporteze cursantii ce nu au sustinut testele obligatorii.

21. Foarte important este raportarea datei si orei de sustinere a testului precum si a timpului consumat.
22. Aveti in vedere urmatoarea ierarhie ce trebuie respectata:
23. Test "WCF si EF", data: ..., ora:..., timp acordat:...
 - a. Test cursant X – acest test este "legat" de "WCF si EF" ...
 - i. Intrebări si raspunsuri test cursant X (urmeaza toate intrebarile pentru X).
 - b. Test cursant Y
 - i. Intrebări si raspunsuri test cursant Y.
24. Posibilitatea de a trimite prin email cursantului rezultatele obtinute.
25. *E posibil sa mai adaug cerinte pe parcursul desfasurarii proiectelor. Atentie la modul de sistematizare al proiectelor pentru ca mentenanta sa nu devina o problema majora.*

Aplicatiile trebuie sa ofere posibilitatea editarii unor rapoarte specifice problemei abordate, transfer informatii in fisiere Excel ,XML si rapoarte in format PDF.

Proiectele finale (creare si exemplificare servicii WCF, aplicatie de tip WEB cu ASP.NET si MVC si aplicatie client-server cu WPF, ambele folosind servicii WCF) vor fi aplicatiile ce modeleaza aceasta problema.

Va voi anunta din timp cand veti prezenta proiectele.

La laborator veti demonstra implementarea diferitelor functionalitati folosind aplicatii tip CUI sau GUI (cu Windows Forms), definirea bazei de date si a modelului (EF). Acest lucru va constitui evaluarea obtinuta in cadrul laboratorului. In cadrul laboratoarelor ce urmeaza vom face exemple cu WCF, ASP.NET MVC, WPF pentru a va usura munca la proiect.

Evaluarea va fi pe etape.

Proiect 1.

Prima etapa a evaluarii va consta in construirea modelului. Functionalitatea modelului va fi demonstrata cu ajutorul unei aplicatii GUI construita de Dv. In aceasta etapa nu folosim servicii WCF pentru acces la date. Ma intereseaza aici modul cum folositi modelul construit si cat de bine raspunde la cerinte, cat de usor poate fi intretinut (adica aduagare de noi cerinte, modificarea unor cerinte existente, eliminarea unor cerinte, structurarea proiectului, etc.). Baza de date va contine minimum 15 intrebari.

Data prezentare: 18-19 Martie 2016

Proiect 2.

A doua etapa a evaluarii va consta din construirea serviciilor WCF si testarea functionarii corecte a acestora. In aceasta etapa se poate modifica aplicatia de la Proiect 1 pentru a demonstra cerintele.

Baza de date va contine minimum 30 intrebari (EF + WCF).

Data prezentare: 15-16 Aprilie 2016

Proiect 3.

Aplicatia WEB, ASP.NET MVC. Voi urmari cum folositi modelul de la proiect 1 si serviciile de la proiect 2. *UI trebuie realizata cat mai profesionist cu putinta.*

Baza de date va contine minimum 45 de intrebari (EF + WCF + ASP.NET).

Data prezentare: 13-14 Mai 2016

Proiect 4.

Aplicatia WPF.

Baza de date va contine minimum 60 de intrebari.

Data prezentare: 27-28 Mai 2016.

In final intrebarile vor trebui sa fie din capitolele pe care le studiem: Entity Framework, WCF, ASP.NET MVC, ASP.NET, WPF.

Neconstruirea bazei de date cu intrebari si raspunsurile aferente se penalizeaza cu cate un punct (la nivel de proiect – adica medie finala) pentru fiecare capitol pe care nu l-ati tratat complet, adica 15 intrebari.