

PROIECT

Termen predare:

21-24.05.2024

Obiectiv:

Obiectivul acestui proiect este familiarizarea cu șablonul architectural *Client/Server*, cu șabloanele arhitecturale orientate pe servicii (SOA) și cu șabloanele de proiectare.

Pentru persistența informației se va utiliza o bază de date relațională (SQL Server, MySQL, etc.).

Cerințe:

Transformați aplicația implementată la **tema 3** într-o aplicație **client/server** astfel încât să utilizați minim **5** șabloane de proiectare (minim un șablon de proiectare creațional, un șablon de proiectare comportamental și un șablon de proiectare structural) și o arhitectură orientată pe servicii **SOA** pentru comunicare între aplicația server și aplicația client.

❖ În *faza de analiză* se va realiza **diagrama cazurilor de utilizare** și **diagramele de activități** pentru fiecare caz de utilizare (*Observație*: numărul diagramelor de activități trebuie să fie **egal** cu numărul de cazuri de utilizare din diagrama cazurilor de utilizare).

❖ În *faza de proiectare* se vor realiza:

- **2 diagrame de clase** corespunzătoare aplicației soft **server** și aplicației soft **client** respectând principiile **DDD** și folosind o arhitectură orientată pe servicii (**SOA**) și minim **5** șabloane de proiectare;
- **diagrama entitate-relație** corespunzătoare bazei de date;
- **diagrame de secvență** corespunzătoare tuturor cazurilor de utilizare (*Observație*: numărul diagramelor de secvență trebuie să fie cel puțin **egal** cu numărul de cazuri de utilizare din diagrama cazurilor de utilizare).

❖ În *faza de implementare* se va scrie cod pentru îndeplinirea tuturor funcționalităților precizate de diagrama cazurilor de utilizare utilizând:

- proiectarea dată de diagramele de clase și diagramele de secvență;
- unul dintre următoarele limbaje de programare: C#, C++, Java, Python.

❖ Finalizarea temei va consta în predarea unui director ce va cuprinde:

- Un fișier cu diagramele UML realizate;
- Baza de date;
- Aplicația soft;
- Documentația (minim 20 pagini) - un fișier care cuprinde:
 - numele studentului, grupa;
 - enunțul problemei;
 - instrumente utilizate;
 - justificarea limbajului de programare ales;
 - descrierea diagramelor UML (inclusiv figuri cu diagramele UML realizate);
 - descrierea aplicației (inclusiv figuri reprezentând interfețele grafice ale aplicației client).

❖ Arhiva se va trimite la adresa de email: anca.iordan@cs.utcluj.ro. Denumirea arhivei va respecta următoarea structură: **Proiect_NumePrenumeStudent**.

Problema 24

Dezvoltați o *aplicație client/server* care poate fi utilizată pentru **organizarea de conferințe științifice**. Aplicația va avea 3 tipuri de utilizatori: participant conferință, organizator conferință și administrator.

Utilizatorii de tip **participant** pot efectua următoarele operații fără autentificare:

- ❖ Înscrierea la conferință (introducere date personale, inclusiv CV-ul și o fotografie);
- ❖ Vizualizarea programului conferinței pe secțiuni (la fiecare secțiune, pentru fiecare participant să fie vizibilă și o fotografie a acestuia, dar și accesarea CV-ului).

Utilizatorii de tip **participant** pot efectua următoarele operații după autentificare:

- ❖ Accesarea volumului conferinței.

Utilizatorii de tip **organizator conferință** pot efectua următoarele operații după autentificare:

- ❖ Acceptarea sau respingerea unui participant (notificare prin email);
- ❖ Operații CRUD în ceea ce privește persistența participanților și a lucrărilor prezentate;
- ❖ Generarea programului conferinței pe secțiuni;
- ❖ Filtrarea listei de participanți după secțiunea la care participă;
- ❖ Salvare liste filtrate cu informații despre lucrările prezentate în mai multe formate: csv, json, xml, doc;
- ❖ Vizualizarea unor statistici utilizând grafice (structură radială, structură inelară, de tip coloană, etc.).

Utilizatorii de tip **administrator** pot efectua următoarele operații după autentificare:

- ❖ Operații CRUD pentru informațiile legate de utilizatorii aplicației care necesită autentificare;
- ❖ Vizualizarea listei tuturor utilizatorilor care necesită autentificare;
- ❖ Filtrarea listei utilizatorilor care necesită autentificare după tip;
- ❖ Notificarea fiecărui utilizator care necesită autentificare prin cel puțin 2 variante (email, SMS, WhatsApp, Skype, etc.) la orice modificare a informațiilor de autentificare aferente acelui utilizator.

Interfața grafică a aplicației client va fi disponibilă în cel puțin 3 limbi de circulație internațională.