**UNIVERSITATEA „ARTIFEX” DIN BUCUREŞTI**

**FACULTATEA DE FINANȚE ŞI CONTABILITATE**

**CONTABILITATE ŞI INFORMATICĂ DE GESTIUNE**

**LUCRARE DE LICENȚĂ**

**Coordonator științific:**

Prof. Univ. Dr. Alexandru MANOLE

**Student**

Silviu Pompiliu SERIGEANU

**BUCUREŞTI, 2018**

**UNIVERSITATEA „ARTIFEX” DIN BUCUREŞTI**

**FACULTATEA DE FINANȚE ŞI CONTABILITATE**

**CONTABILITATE ŞI INFORMATICĂ DE GESTIUNE**

**REALIZAREA UNUI CATALOG ELECTRONIC FOLOSIND O BAZĂ DE DATE SQL**

**Coordonator științific:**

Prof. Univ. Dr. Alexandru MANOLE

**Student**

Silviu Pompiliu SERIGEANU

**BUCUREŞTI, 2018**

CUPRINS

1. Introducere 4
2. Proiectarea aplicației 5
   1. Prezentarea tehnologiilor folosite 5
   2. Proiectarea bazei de date 5
   3. Prezentarea bazei de date 8
   4. Diagrame UML 8
3. Realizarea aplicației 9
   1. Conectarea la baza de date 10
   2. Algoritmi folosiți 11
   3. Realizarea interfeței grafice 12
4. Manual de utilizare 13
5. Concluzii și dezvoltări ulterioare 14
6. Bibliografie 15

Capitolul 1

INTRODUCERE

Lucrarea de fata prezinta modul de realizare și funcționare al unei aplicații Windows care își propune organizarea unui catalog electronic, folosind o colecție de baze de date.

Aplicația a fost realizată conectând o interfață grafica de tip Windows, la o bază de date SQL ce conține informații despre entitățile bazei de date. Aceasta combinație a fost aleasă deoarece legătura dintre interfață si baza de date se poate face cu ușurință folosind același set de instrumente ce se regăsește în Visual Studio. Versiunea folosita pentru realizarea aplicației este Microsoft Visual Studio Community 2017 15.7.2.

Aceasta aplicație poate înlocui cu ușurință modelul clasic al cataloagelor folosite în sistemul de învățământ și prezinta mai multe avantaje:

* Organizează datele într-o structura logica, ușor de înțeles
* Toate datele pot fi găsite într-un singur loc
* Asigura persistenta datelor înregistrate
* Economisește timp și spațiu pentru utilizatori
* Prezinta un grad ridicat de flexibilitate în modificare datelor existente

Funcționalități dorite:

* Înregistrarea cu ușurință a datelor specifice unui catalog (adăugare/modificare/ștergere)
* Organizarea tuturor listelor de entități folosite (studenți, profesori, materii etc.)
* Afișarea datelor într-o singura fereastra
* Exportarea datelor într-un format ușor de folosit

Capitolul 2

PROIECTAREA APLICATIEI

2.1 Prezentarea tehnologiilor folosite

Pentru gestionarea bazei de date s-a folosit limbajul **SQL** (Structured Query Language). Acces limbaj a apărut la începutul anilor ’70 fiind dezvoltat la IBM de către Donald D. Chamberlain și Raymond F. Boyce după studierea modelului relațional conceput de către Ted Codd în 1968. SQL are la baza tabele de date legate prin relații, iar din interogarea acestor tabele rezulta șiruri de date ce îndeplinesc cerințele interogării.

Pentru realizarea interfeței grafice s-a folosit sistemul grafic **WPF** (Windows Presentation Foundation) ca parte integranta a .NET Framework 3.0, iar pentru definirea elementelor grafice s-a folosit limbajul **XAML** (Extensible Application Markup Language). S-a ales aceasta varianta deoarece aplicațiile realizate în sistemul WPF sunt compatibile cu orice versiune de Windows, iar limbajul XAML permite un grad mare de personalizare a interfeței grafice.

Pentru legaturile dintre elementele grafice și baza de date SQL s-a folosit limbajul de programare **C#,** dezvoltat de Microsoft. Fiind un limbaj orientat pe obiect, permite definirea cu ușurință a claselor de obiecte folosite de către aplicație precum și construirea algoritmilor necesari rezolvării problemelor legate de interogarea bazelor de date.

2.2 Proiectarea bazei de date

**Modelul relațional al unei baze de date**, formulat de Ted Codd, presupune existenta a trei elemente componente:

1. **Structuri de date**. Acestea sunt alcătuite din relații de date bidimensionale si se refera la:

1. **Atribute si domenii**. Plecând de la unul din avantajele fundamentale ale modelului relațional, omogenitatea, care presupune așezarea datelor în tabele cu linii si coloane care au același format, **atributul** reprezintă o caracteristica a unei entități din baza de date si este reprezentat de valoarea ce se regăsește la intersecția dintre o linie si o coloana, iar **domeniul** unui atribut reprezintă mulțimea finita de valori de același tip pe care le poate primi un atribut.
2. **Tuplul** reprezintă mulțimea de valori aflata pe o linie dintr-o relație tabelara, adică mulțimea valorilor atributelor atașate unei entități din baza de date.
3. **Relația** este o mulțime finita de tupluri si de regula ea reprezintă numele tabelei din baza de date. **Schema** relației reprezintă suma proprietăților comune ale tuplurilor conținute in relație.

2. **Restricțiile de integritate** reprezintă un set de cerințe ce trebuie îndeplinite pentru a asigura corectitudinea si coerenta datelor din baza de date. **Cheia** unui tuplu reprezintă un identificator unic, reprezentat de valoarea unui atribut. In acest caz ea se mai numește si **cheie simplă**. In cazul in care identificatorul este format din mai multe atribute, atunci **cheia** este **compusă**. **Cheia externa** reprezintă un atribut dintr-o schema definit pe același domeniu ca si cheie primară a altei scheme.

3. **Operații asupra bazei de date**. Principalele modificări operate asupra unei baze de date presupun:

- **adăugarea** unui tuplu nou într-o relație a bazei de date

- **modificarea** valorilor atributelor unui tuplu existent într-o relație

- **ștergerea** unui tuplu dintr-o relație

Pentru proiectarea bazei de date a aplicației de față, se va lucra cu următoarele entități:

* Studenți
* Profesori
* Materii
* Specializări
* Grupe

,precum si cu înregistrări privind examenele (notă, dată, etc.) ce se vor regasi sub schema **Catalog**.

**Lista de studenți** va stoca informațiile relevante despre studenți. Fiecare student va avea un număr matricol care va fi un identificator (cheie) unic, nume, prenume si posibilitatea de a fi asociat unei grupe de studenți. Pot fi adăugați studenți in aceasta tabela, fără a fi asociați cu o grupa, presupunând ca sunt înscriși de curând si nu au fost încă alocați unei grupe sau nu s-a creat încă grupa unde vor fi alocați. Un student poate fi alocat unei singure grupe, dar o grupa poate conține mai mulți studenți (asociere de tipul 1:M).

**Lista de profesori** va conține informațiile relevante despre profesori. Fiecare profesor va avea un număr de identificare (cheie) unic, nume si prenume.

**Tabela de materii** va conține o lista de nume de materii si un identificator unic pentru fiecare materie.

Informații similare for fi stocate si in **lista de specializări**:nume specializare si identificator unic.

Atribute:

* **NumarMatricol**
  + Tip de data *int*
  + Este cheie primara
  + Nu accepta valori nule
  + Valoare inițială = 1
  + Valoare este crescuta cu 1 automat, la crearea unui tuplu nou
  + Valoare nu poate fi modificata de utilizator
* **NumeStudent**
  + Tip de data *nvarchar*
  + Lungime maxima 50
  + Nu accepta valori nule
* **PrenumeStudent**
  + Tip de data *nvarchar*
  + Lungime maxima 50
  + Nu accepta valori nule
* **IdGrupa**
  + Tip de data *int*
  + Este cheie străină din tabela **ListaGrupe**
  + Permite valori nule
  + Face legătura intre un student si o grupă