

# **Biblioteca Virtuala**

**12.01.2021**

**Îndrumător:**

**dr. ing. Daniel Morariu**

**Student:**

**Soaita Alin-Mihai**

**223/2**

**Istoric Versiuni**

<b>Data</b>	<b>Versiune</b>	<b>Descriere</b>	<b>Autor</b>
21.11.2020	1.0	Realizarea interfetei grafice, cu toate form-urile necesare	Soaita Alin-Mihai
10.12.2020	1.1	Conectarea la o baza de date, si crearea unei clase pentru gestiunea ei	Soaita Alin-Mihai
20.12.2020	2.0	Adaugate functionalitatile de Login si Register, cat si clasele pentru utilizatori si tipuri de utilizator	Soaita Alin-Mihai
21.12.2020	2.1	Adaugata o lista cu utilizatorii conectati care se actualizeaza in timp real folosind baza de date si o animatie pentru afisarea ei	Soaita Alin-Mihai
22.12.2020	2.2	Adaugat profil pentru fiecare utilizator, in care se pot vedea majoritatea informatiilor despre un utilizator	Soaita Alin-Mihai
02.01.2021	3.0	Adaugate clasele pentru carti si categorii de carti, cat si o lista pentru afisarea acestora	Soaita Alin-Mihai
03.01.2021	3.1	Adaugata posibilitatea de a cauta o carte dupa nume sau un autor si sortarea lor dupa o anumita categorie de carti	Soaita Alin-Mihai
04.01.2021	3.2	Adaugate functionalitatile de adaugare, editare, stergere si vizualizare a unei carti	Soaita Alin-Mihai

## Cuprins

<b>ISTORIC VERSIUNI .....</b>	<b>2</b>
<b>CUPRINS .....</b>	<b>3</b>
<b>1 SPECIFICAREA CERINTELOR SOFTWARE .....</b>	<b>4</b>
<b>1.1 Introducere .....</b>	<b>4</b>
1.1.1 Obiective .....	4
1.1.2 Definiții, Acronime și Abrevieri .....	4
1.1.3 Tehnologiile utilizate .....	5
<b>1.2 Cerințe specifice.....</b>	<b>5</b>
<b>2 FUNCTIA DE SORTARE A CARTILOR IN FUNCTIE DE CATEGORIE SI CAUTAREA ACESTORA DUPA CUVINTE CHEIE.....</b>	<b>6</b>
<b>2.1 Descriere.....</b>	<b>6</b>
<b>2.2 Fluxul de evenimente .....</b>	<b>6</b>
2.2.1 Fluxul de bază .....	6
2.2.2 Pre-condiții.....	7
2.2.3 Post-condiții .....	7
<b>3 AFISAREA UTILIZATORILOR CONECTATI INTR-O LISTA CU ACTUALIZARE 8</b>	<b>8</b>
<b>3.1 Descriere.....</b>	<b>8</b>
<b>3.2 Fluxul de evenimente .....</b>	<b>8</b>
3.2.1 Fluxul de bază .....	8
3.2.2 Pre-condiții.....	8
3.2.3 Post-condiții .....	8
<b>4 IMPLEMENTARE .....</b>	<b>9</b>
<b>4.1 Diagrama de clase.....</b>	<b>9</b>
<b>5 BIBLIOGRAFIE .....</b>	<b>10</b>

# 1 Specificarea cerințelor software

## 1.1 Introducere

Aplicatia permite gestionarea unei biblioteci.

### 1.1.1 Obiective

Obiectivul proiectului este acela de a putea gestiona o biblioteca prin intermediul unei aplicatii care sa permita administratorilor sa gestioneze biblioteca, iar utilizatorilor sa le permita folosirea ei.

Obiectivele atinse:

- Crearea mai multor tipuri de utilizatori (utilizator normal, administrator, student) si crearea unei profil al fiecaruia
- Adaugarea unei liste cu utlizatorii conectati pe aplicatie folosind baza de date, lista se actualizeaza o data la 5 secunde
- Crearea unei metode in plus de securitate pentru utilizatorii de tip Administrator, cu un PIN separat de parola
- Posibilitatea de a vizualiza, edita, adauga si sterge carti din baza de date
- Posibilitatea de a sorta cartile pe anumite categorii, cat si cautarea acestora dupa nume sau autor

Obiective nerealizate:

- Posibilitatea de a modifica date despre un utilizator direct din aplicatie
- Posibilitatea de a previzualiza sau vizualiza o carte
- Posibilitatea de a rezerva o carte pentru ridicare sau descarcare in format pdf
- Actualizarea listei de carti pentru toti utilizatorii care folosesc aplicatia in acelasi moment, la adaugarea/modificarea/stergera unei carti
- Categoria „Studenti” trebuie sa fie mereu ultima adaugata in baza de date pentru a functiona corect
- Id-urile categoriilor in baza de date trebuie sa fie numere consecutive incepand de la 1 pentru a functiona corect
- Posibilitatea de a adauga categorii noi direct din aplicatie

### 1.1.2 Definiții, Acronime și Abrevieri

- Clasa cu numele „DB” (Database) face conexiunea cu baza de date si contine niste functii care lucreaza pe baza de date cum ar fi „LoginUser”
- Clasa cu numele „Main” contine niste variabile generale (cum ar fi lista de utilizatori conectati sau lista de carti) si niste functii generale (cum ar fi updatarea listei de utilizatori conectati sau listei de carti), de asemenea se foloseste baza de date si in aceasta clasa
- Unde se gaseste denumirea „Qty” se refera de fapt la „Quantity”
- Codul este scris in engleza, adica toate denumirile sunt scrise in engleza, dar interfata aplicatiei este in romana
- Pentru numirea variabilelor si metodelor (in afara de cele generate automat) s-a incercat sa se respecte o anumita regula; la variabile se incepe cu litera mica, iar daca variabila este formata din 2 cuvinte separate, al doilea cuvint se incepe cu litera mare (de ex: „bookName”) ; metodele incep cu litera mare, iar fiecare cuvint nou incepe si el cu litera mare (de ex: „UpdateUsersList”)

### 1.1.3 Tehnologiile utilizate

- Visual Studio Community 2019 -> pentru realizarea proiectului
  - S-a folosit pachetul „Costura.Fordy” pentru a include toate referintele in executabilul final (pentru a se putea rula executabilul final pe orice calculator, chiar daca nu are instalat MySqlConnection)
  - MySqlConnection, trebuie instalat pentru ca proiectul sa poata fi compilat deoarece se foloseste o baza de date MySQL (<https://dev.mysql.com/downloads/file/?id=498572>)
- Microsoft Office Word -> pentru redactarea documentatiei

## 1.2 Cerințe specifice

Functionalitati:

- Logare si inregistrare utilizator
- Afisarea unor informatii pe scurt ale utilizatorilor conectati intr-o lista cu actualizare
  - Folosind un timer, la fiecare 5 secunde se verifica in baza de date ce utilizatori sunt conectati si ii adauga sau ii scoate din lista de utilizatori conectati
- Afisarea profilului oricarui utilizator care contine informatii mai in detaliu
- Functia de verificare a unui PIN pentru utilizatorii de tip Administrator
  - Fiecare utilizator cu grad de Administrator o sa aiba un PIN separat de parola, care va fi cerut la logare
- Vizualizarea, adaugarea, editarea si stergerea unei carti din baza de date
- Functia de sortare a cartilor in functie de categorie si cautarea acestora dupa cuvinte cheie

## 2 Functia de sortare a cartilor in functie de categorie si cautarea acestora dupa cuvinte cheie

### 2.1 Descriere

Aceasta functionalitate permite organizarea cartilor pe diferite categorii, este importanta pentru ca experienta utilizatorului sa fie mai placuta si aplicatia sa fie mai usor de folosit.

### 2.2 Fluxul de evenimente

#### 2.2.1 Fluxul de bază

In form-ul principal („fMain”) a fost creat un element de tip **comboBox** numit „comboBox1” care contine in el toate categoriile incarcate in aplicatie din baza de date.

Ne vom folosi de o lista numita „bookList” de tip **ListItemView** pentru a tine minte toate elementele ce trebuie adaugate in lista de carti.

Metoda care actualizeaza lista de carti in functie de categoria aleasa se regaseste in clasa „Main” cu numele „UpdateBookList”, care primeste ca si parametri de tipul urmator: **comboBox**, **listView**, **bool**, **int**.

Parametrii de tip **comboBox** si **listView** sunt transmisi pentru ca acestia sa poata fi modificati direct din functie, fara a fi necesara modificarea lor in form-ul principal.

Parametrul de tip **bool** „isStudent” verifica daca utilizatorul curent este de tip Student (utilizatorii de tip Administrator sunt considerati si ei Student in acest caz) pentru a determina daca categoria destinata studentilor ii va fi afisata.

Parametrul de tip **int** „category” transmite id-ul categoriei care trebuie sa ii fie afisata utilizatorului respectiv (daca categoria e = 0, se vor afisa toate cartile).

Primul lucru pe care il verifica metoda este daca s-au adaugat deja in **comboBox** categoriile incarcate din baza de date (se incarca cu metoda „LoadCategories” din clasa „DB”). Aceasta functie adauga intr-o lista de tip „Category” din clasa „Main” toate categoriile gasite.

Daca categoriile nu au fost adaugate se apeleaza o functie separata „UpdateCategoriesList” din aceeasi clasa „Main”. Aceasta functie adauga in **comboBox** fiecare cateogrie in parte, iar la inceput adauga optiunea „Toate categoriile” pentru vizualizarea cartilor.

Dupa adaugarea categoriilor, lista „bookList” va fi resetata si urmeaza un foreach care parcurge toate categoriile si se cauta categoria care are id-ul = parametrul „category”, iar daca il gaseste, un alt foreach va parcurge fiecare carte din categoria respectiva si se va crea un **ListViewItem** cu datele cartii respective (metoda „BookListInfo” din clasa „Book” va fi apelata pentru a utiliza datele despre o carte in aceasta lista). Dupa ce ListViewItem-ul a fost creat, acesta va fi adaugat in „bookList”.

In final va fi apelata metoda „DisplayItems” care parcurge toate **ListViewItem**-urile din lista „bookList” si le adauga in **listView**-ul transmis ca parametru in metoda „UpdateBookList”.

In form-ul principal exista o variabila numita „selectedCategory” care va tine minte mereu categoria care trebuie afisata utilizatorului (a fost creata pentru ca alte evenimente care folosesc „UpdateBookList” sa nu interfereze cu categoria care trebuie afisata)

În form-ul principal, metoda „UpdateBookList” este apelată în interiorul event-ului „comboBox1\_SelectedIndexChanged”. Variabila „selectedCategory” va lua valoarea index-ului selectat în **comboBox1**, iar metoda se va apela cu următorii parametri:

- *comboBox1* (**comboBox**-ul folosit pentru selectarea categoriei)
- *listView2* (lista cu cărți)
- *meObject is Student || meObject is Administrator* (această verificare returnează „true” dacă obiectul care reține date despre utilizatorul care folosește aplicația în acel moment „meObject” este de tip Student sau Administrator)
- *selectedCategory* (categoria selectată de către utilizator)

Pentru căutarea cărților după cuvinte cheie se va folosi evenimentul „textBox1\_TextChanged”. Pe lângă asta, mai există o variabilă numită „sortIndex” care ține minte tipul de căutare pe care trebuie să îl facă (după nume sau după autor), acest „sortIndex” se modifică la evenimentele de click pentru radio button-urile „rbName” și „rbAuthor”.

Se folosește un `foreach` care trece prin toate **ListViewItem**-urile din lista „bookList” creată în clasa „Main” și folosindu-se de metoda „Where” aplică un filtru listei în felul următor: alege sub-elementele din lista care conțin text din „textBox1” pe index-ul („sortIndex”) selectat (compară coloana de nume sau coloana de autor).

### 2.2.2 Pre-condiții

Pentru căutarea unei cărți după nume sau autor, se va completa casuta de sub „Caută o carte:”, iar rezultatele vor apărea în timp real. Pentru a schimba modul de căutare, se va face click pe „După nume” (în mod normal această opțiune e deja aleasă) sau „După autor”.

Pentru sortarea pe categorii, se va folosi lista de lângă casuta de căutare a cărților.

### 2.2.3 Post-condiții

După schimbarea categoriei, lista de cărți va fi actualizată cu cărți doar din acea categorie.

### 3 Afisarea utilizatorilor conectati intr-o lista cu actualizare

#### 3.1 Descriere

.Aceasta functionalitate permite afisarea tuturor utilizatorilor conectati pe aplicatie, folosind baza de date.

#### 3.2 Fluxul de evenimente

##### 3.2.1 Fluxul de bază

Functionalitatea este realizata cu ajutorul unui timer „timerUpdateList” definit in form-ul principal „fMain” si cu ajutorul a doua metode: „UpdateConnectedList” din clasa „DB” si „UpdateUsersList” din clasa „Main”.

Metoda „UpdateConnectedList” are ca parametru un obiect de tip „Main”, iar acesta va fi transmis din form-ul principal cu numele „mainObject”.

Aceasta functie verifica in baza de date toti utilizatorii care au campul „Connected” = 1 (la inchiderea aplicatiei de catre un utilizator se va executa un query care seteaza „Connected” = 0).

Apoi cu un „while” se parcurg toate randurile citite din baza de date. La fiecare rand parcurs, se verifica daca utilizatorul este deja adaugat in lista „connectedUsers” din clasa „Main”, adica daca apare deja in lista cu utilizatori conectati. Daca acesta nu exista, se adauga un obiect nou in lista „connectedUsers” cu ajutorul constructorilor de la clasele de utilizatori, respectiv „User”, „Administrator” si „Student”.

De asemenea, metoda verifica si toti utilizatorii care au campul „Connected” = 0, iar dupa cu un foreach verifica daca mai apar pe lista de utilizatori conectati si ii elimina daca este cazul.

Prin variabila „modified” metoda returneaza „true” doar daca s-au produs schimbari asupra listei „connectedUsers” pentru a nu face operatii inutile.

Daca metoda „UpdateConnectedList” returneaza valoarea „true”, atunci se apeleaza si metoda „UpdateUsersList” din clasa „Main” care are ca parametru un ListView, din form-ul principal se va trimite „listView1”.

Aceasta metoda actualizeaza lista trimisa ca parametru in felul urmator:

- se va sterge tot ce e deja in lista
- se parcurg toate obiectele de tip User din lista „connectedUsers” si folosind metoda „OnlineInfo” din clasa „User” se vor obtine detalii despre utilizatorul respectiv
- se creeaza un element al listei, cu informatiile utilizatorului ca sub-elemente

##### 3.2.2 Pre-condiții

Trebuie apasat butonul „Utilizatori conectati >> ” pentru a afisa lista.

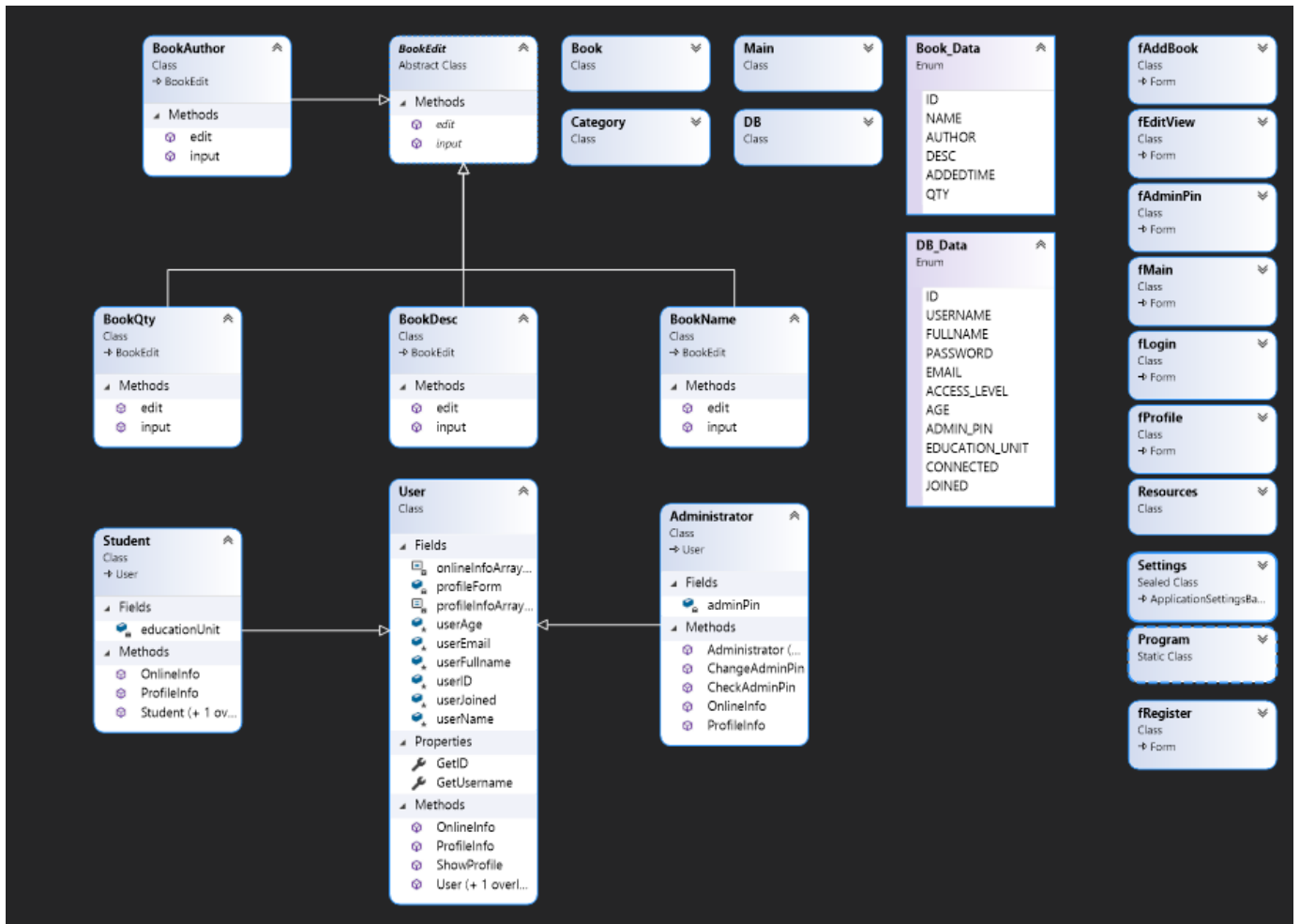
##### 3.2.3 Post-condiții

Lista se va actualiza automat, iar selectand un utilizator si apasand click dreapta, se va putea deschide profilul utilizatorului respectiv.



## 4 Implementare

#### 4.1 Diagrama de clase



## 5 Bibliografie

Principii OOP in C#: <https://stackify.com/oop-concepts-c-sharp/>

Polimorfism in C#: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/polymorphism>

Conectarea la baza de date: <https://ourcodeworld.com/articles/read/218/how-to-connect-to-mysql-with-c-sharp-winforms-and-xampp>

Utilizarea unui ListView: <https://www.youtube.com/watch?v=Q8XQg8PoB9U>

Filtrarea rezultatelor intr-un ListView : <https://stackoverflow.com/questions/51885778/filter-items-in-a-listview-in-real-time/51886069>