
Software Requirements Specification

for

“Bibliotecă Virtuală”

Realizat de:
Cojocaru Georgiana-Vasilica, 1309A
Lupu Andra-Maria, 1309A
Martin Maria, 1309A
Petrescu-Miron Ștefănel, 1309A

28 Mai 2023

Cuprins

Cuprins.....	2,3
1. Introducere.....	4
1.1 Scop.....	4
1.2 Convențiile Documentului.....	4
1.3 Publicul Vizat și Sugestii de Lectură.....	4
1.4 Domeniul Produsului.....	5
2. Descriere Generală.....	5
2.1 Perspectiva Produsului.....	5
2.2 Funcțiile Produsului.....	5
2.3 Clasele și Caracteristicile Utilizatorilor.....	6
2.4 Mediul de Operare.....	6
2.5 Constrângeri de Proiectare și Implementare.....	6
2.6 Documentația Utilizatorului.....	6
2.7 Ipoteze și Dependențe.....	7
3. Cerințe de Interfață Externă.....	7
3.1 Interfețe cu Utilizatorul.....	7
3.2 Interfețe Hardware.....	8
3.3 Interfețe Software.....	8
3.4 Interfețe de Comunicații.....	8
4. Alte Cerințe Nefuncționale.....	9
4.1 Cerinte de Performanță.....	9
4.2 Cerințe de Siguranță.....	9
4.3 Cerințe de Securitate.....	9
4.4 Atribute de Calitate Software.....	9

4.5 Reguli de Afaceri.....	9
5. Alte Cerințe.....	10
Anexa A: Glosar.....	10
Anexa B: Modele de Analiză.....	10, 11, 12, 13, 14, 15
Anexa C: Lista "Urmează să fie determinat".....	15
Anexa D: Tasks.....	16
Anexa E: Părți semnificative din codul sursă.....	17, 18, 19, 20

1. Introducere



1.1 Scop

“Bibliotecă virtuală” este o aplicație de tip bibliotecă în care sunt adunate titluri de cărți pe care un utilizator le poate împrumuta. Scopul este crearea unui program care să vină în ajutorul utilizatorilor, punând la dispoziție titluri de cărți din diverse domenii. Poate fi considerată drept o bibliotecă tradițională, dar care furnizează informații într-un format digital. Aplicația dispune de un meniu utilizator, în care se vor loga utilizatorii și vor putea împrumuta cărțile care sunt disponibile în bibliotecă sau returna cărțile pe care le-au împrumutat, și un meniu administrator, în care se va loga administratorul bibliotecii și va putea adăuga noi cărți în bibliotecă sau șterge cărți și va putea adăuga noi utilizatori sau îi va putea șterge.

1.2 Convențiile Documentului

Acest document urmează formatarea standard IEEE pentru dezvoltarea de software. Standardul definește o formatare obișnuită pe care o urmează acest document, inclusiv scrierea care trebuie făcută la persoana a treia, precum și text lizibil și corect din punct de vedere gramatical. Pentru redactarea documentului se va folosi font Times New Roman, de dimensiune 12.

1.3 Publicul Vizat și Sugestii de Lectură

Acest document nu este destinat utilizatorului, deoarece oferă detalii despre modul în care este implementată aplicația. Deoarece un utilizator are nevoie de informații despre cum să folosească aplicația realizată, în loc de cum să o creeze, acest document este destinat persoanelor care vor să facă teste sau să creeze la rândul lor o bibliotecă virtuală.

1.4 Domeniul Produsului

“Bibliotecă virtuală” este o aplicație de tip bibliotecă dezvoltată în limbajul C#, folosind Windows Forms App (.NET Framework), în Visual Studio. Are o interfață grafică unde se poate selecta dacă se dorește accesare meniului utilizator sau meniului administrator. Obiectivul este de a împrumuta cărți și a le returna înainte de termenul limită, altfel se va primi un email de avertizare trimis de către administrator. Aplicația este concepută pentru a fi ușor de utilizat. Este destinată să ruleze pe orice PC care are instalat framework-ul .NET.

2. Descriere Generală

2.1 Perspectiva Produsului

“Biblioteca Virtuală” este o aplicație ce prezintă o colecție de cărți, în format digital, pe care un utilizator dorește să le citească. Pentru realizarea acesteia a fost folosit limbajul C#.

Tehnologia este într-o continuă dezvoltare și are un impact semnificativ în viețile oamenilor. O bibliotecă virtuală poate oferi cititorilor avantajul de a accesa oricând și oriunde titluri de cărți din orice domeniu, astfel, utilizatorii pot citi cărți și informații relevante fără a fi nevoiți să se deplaseze la o bibliotecă tradițională, poate include o gamă diversificată de cărți, reviste, articole, oferind utilizatorilor acces la informații din diferite domenii de interes, aplicația putând satisface nevoile și preferințele diverse ale utilizatorilor.

Totodată, utilizarea unei biblioteci virtuale elimină necesitatea de a depozita cărți fizice, ceea ce duce la economisirea spațiului fizic și a resurselor necesare pentru întreținerea și gestionarea unei biblioteci tradiționale, implică mai puțină hârtie, ceea ce duce la reducerea tăierii arborilor, și contribuie la eficientizarea utilizării resurselor și la protejarea mediului. Pentru a citi o carte, un utilizator se poate conecta și o poate accesa cu ușurință de pe un computer pentru a fi responsabil față de mediu.

Un alt beneficiu pe care o bibliotecă virtuală îl prezintă este promovarea și stimularea creativității unei persoane, accesul la informație fiind mult mai facil.

2.2 Funcțiile Produsului

Principalele funcții pe care biblioteca virtuală le poate oferi utilizatorului sunt următoarele:

- Colecție extinsă de cărți digitale: o bibliotecă virtuală oferă o gamă diversificată de cărți și materiale digitale. Aceasta oferă utilizatorilor acces la o mare varietate de conținut.
- Accesibilitate: utilizatorii pot accesa biblioteca virtuală de pe un computer sau laptop și pot citi și să acceseze conținutul în orice moment și de oriunde.
- Actualizări și conținut nou: biblioteca virtuală poate oferi actualizări regulate de conținut, inclusiv noi lansări de cărți și informații actualizate în domeniile de interes ale utilizatorilor.

2.3 Clasele și Caracteristicile Utilizatorilor

Biblioteca virtuală este proiectată să fie utilizată cu ușurință de către un utilizator. Pentru a avea acces la cont, o persoană care dorește să folosească aplicația va trebui să anunțe administratorul bibliotecii, printr-un email la virtual.library.ip@gmail.com, acesta îl va înregistra în aplicație și îi va trimite username-ul și parola cu care se poate conecta. Odată introduse datele de autentificare, utilizatorul poate selecta cărțile pe care dorește să le împrumute și, înainte de termenul limită, le poate returna. Dacă a depășit acel termen, administratorul îi va trimite o avertizare pe mail. În lista cărților pe care un utilizator le poate împrumuta vor apărea doar cărțile ce sunt disponibile, cele deja împrumutate de alții utilizatori nu vor fi listate.

2.4 Mediul de Operare

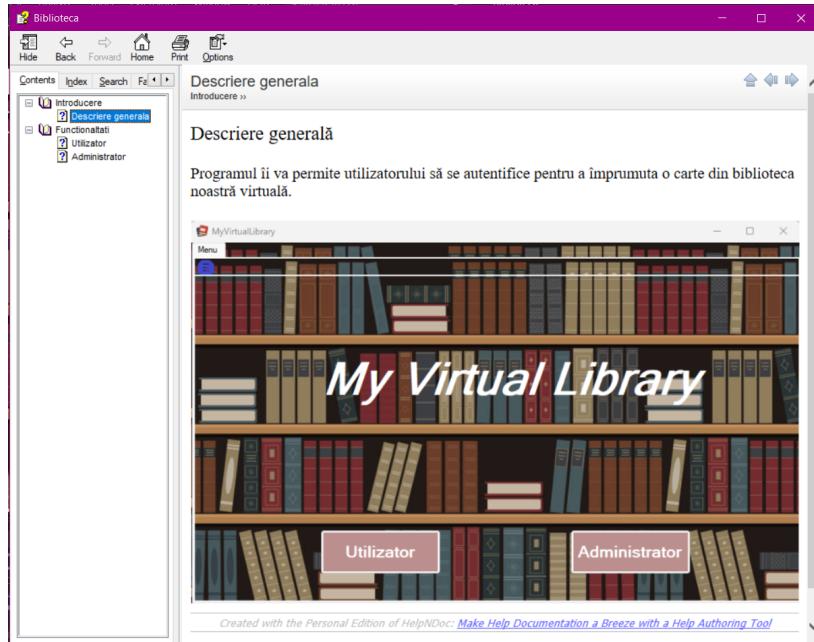
“Bibliotecă virtuală” este o aplicație de tip bibliotecă dezvoltată în limbajul C#, folosind Windows Forms App (.NET Framework), în Visual Studio. Aplicația este compatibilă cu sistemul de operare Windows și necesită .NET Framework 4.5 sau o versiune ulterioară. Pentru eventualele avertizări pe care un utilizator le va primi, este necesar ca utilizatorul să dețină un cont de gmail valid (username@gmail.com)

2.5 Constraineri de Proiectare și Implementare

Deoarece “Bibliotecă virtuală” este proiectată și implementată într-un singur semestru ca proiect pentru Universitatea Tehnică Gheorghe Asachi, disciplina Ingineria Programării, este posibil ca timpul să fie cel mai limitativ factor în dezvoltarea aplicației. Deoarece aplicația conține doar titluri de cărți și autorii acesteia, va fi necesară o bază de date pentru a gestiona informațiile și relațiile dintre titluri și autori. Un alt aspect ce trebuie îmbunătățit este interfața utilizatorului, în care vor fi adăugate noi funcționalități, precum căutarea unei cărți sau filtrarea ei după categorie, autor.

2.6 Documentația Utilizatorului

Meniul principal al aplicației prezintă un mic meniu în partea stânga sus în care sunt butoanele: About, care conține numele echipei ce a realizat aplicația, Exit, care închide aplicația, și Help, care deschide documentația utilizatorului, o documentație API, realizată folosind Doxygen. Aceasta conține informații despre cum funcționează aplicația și capturi de ecran ajutătoare.



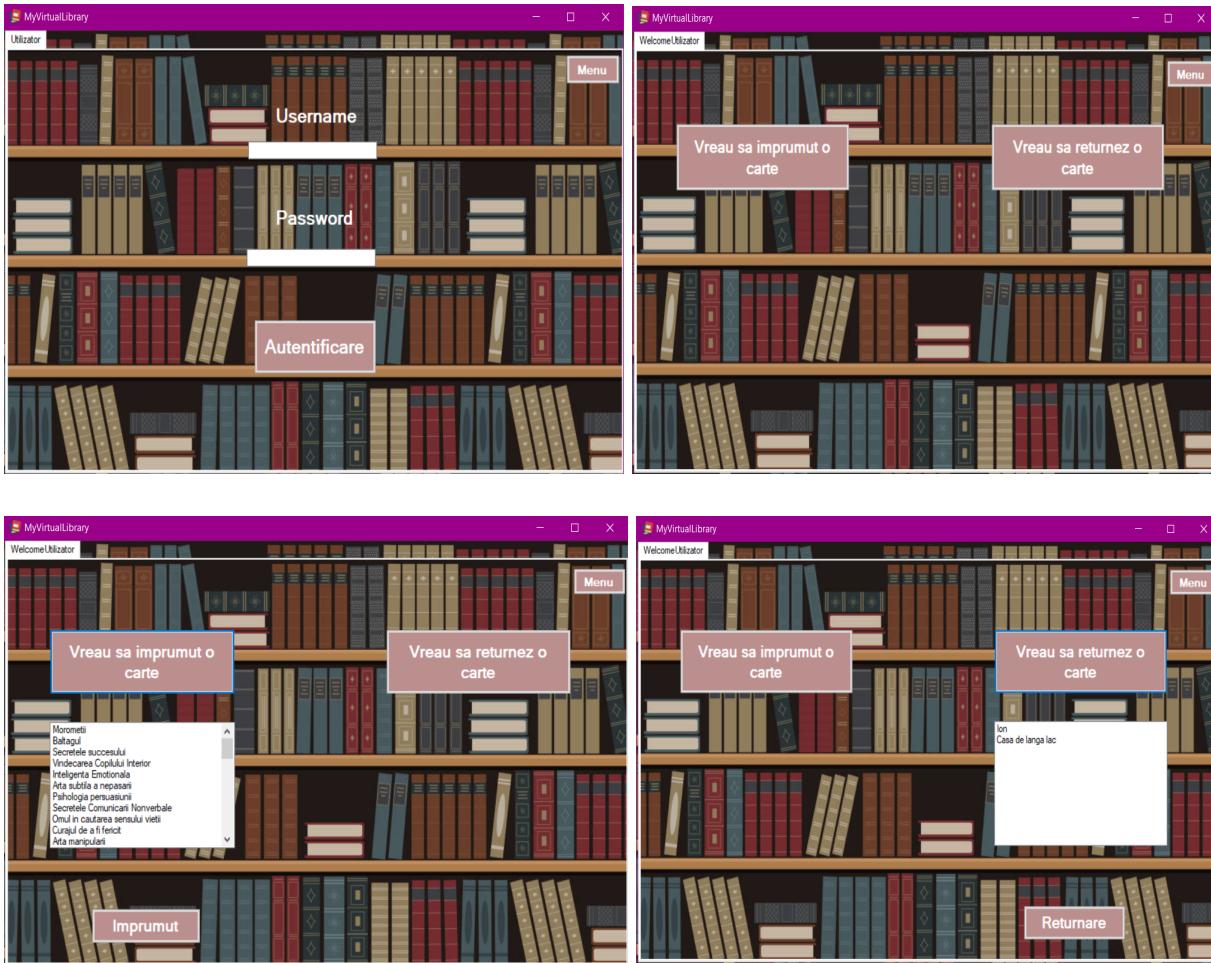
2.7 Ipoteze și Dependențe

Aplicația depinde de existența și disponibilitatea fișierului text care conține informațiile despre cărțile bibliotecii virtuale și de fișierul de utilizatori, care face posibilă autentificarea unui utilizator în aplicație. Informațiile despre cărți din fișierele text vor fi valide și complete, astfel încât aplicația să poată prelua și utiliza aceste date în mod corespunzător. Utilizatorii vor avea acces la internet și vor putea utiliza aplicația de pe un dispozitiv compatibil (computer sau laptop). Accesul la internet este necesar pentru accesarea contului de gmail valid pe care îl dețin, cont obligatoriu pentru a crea un cont în aplicație și pentru a primi eventuale avertizări.

3. Cerințe de Interfață Externă

3.1 Interfețe cu Utilizatorul

Interfața cu utilizatorul este foarte ușor de folosit. În interfața cu utilizatorul din biblioteca virtuală va fi afișat meniul de autentificare a utilizatorului. Dacă datele introduse sunt eronate, se va deschide o fereastră în care i se va preciza acest lucru. Odată ce s-a autentificat cu succes, vor apărea două butoane: un buton care deschide opțiunea pentru împrumutarea cărților (“Vreau să împrumut o carte”), iar la apăsarea acestuia va fi prezentată lista cărților disponibile în bibliotecă și prin selectarea unei cărți și apăsarea butonului “Împrumut” se va realiza împrumutul cărții, și un buton care deschide opțiunea pentru returnarea cărților (“Vreau să returnez o carte”) și prin apăsarea acestuia vor fi afișate titlurile cărților pe care un utilizator le-a împrumutat și trebuie să le returneze. Pentru returnarea cărților se procedează asemănător cu împrumutul lor: se va selecta cartea pe care utilizatorul dorește să o returneze, iar prin apăsarea butonului “Returnare” se va realiza returnarea cărții.



3.2 Interfețe Hardware

Interfețele hardware vor include un mouse, tastatura și monitorul de afișare. Nu este nevoie de mult hardware pentru a rula aplicația, doar un computer/laptop cu sistem de operare Windows. Click stânga al mouse-ului va permite utilizatorului să interacționeze cu obiectele din interfața aplicației. Tastatura va fi necesară pentru introducerea datelor de autentificare a utilizatorului.

3.3 Interfețe Software

Pentru a putea folosi aplicația, utilizatorul va trebui să aibă sistem de operare Windows (cel puțin Windows 7). Aplicația fiind dezvoltată în limbajul C# și utilizând platforma .NET Framework, utilizatorul trebuie să aibă instalat .NET Framework 4.5 sau o versiune ulterioară și de Visual Studio Runtime, un pachet de componente necesare pentru a executa aplicații dezvoltate în Visual Studio, pentru a rula aplicația corect.

3.4 Interfețe de Comunicații

Aplicația bibliotecii virtuale utilizează interfețe de comunicare adaptate la stocarea datelor în fișiere text. Interacțiunea utilizatorului cu aplicația se realizează prin intermediul unei interfețe grafice intuitive, unde utilizatorii pot accesa și utiliza funcționalitățile bibliotecii virtuale. Pentru

autentificare, utilizatorii introduc datele de autentificare, care sunt verificate în fișierul de utilizatori. Datele despre cărți și utilizatori sunt stocate în fișiere text, iar aplicația efectuează operațiile de citire și scriere pentru a accesa și actualiza aceste informații. În plus, aplicația permite trimitera de emailuri pentru a oferi utilizatorilor avertizări cu privire la întârzierea returnării unei cărți, necesitând un cont de email valid.

4. Alte Cerințe Nefuncționale

4.1 Cerinte de Performanță

Sistemul de operare minim necesar este Windows 7 sau o versiune ulterioară. Este recomandat să aibă un procesor dual-core sau mai performant și cel puțin 4 GB de memorie RAM pentru a asigura o funcționare fluentă. De asemenea, trebuie să existe suficient spațiu de stocare disponibil pentru gestionarea colecției de cărți.

4.2 Cerințe de Siguranță

Cerințele de siguranță ale bibliotecii virtuale includ protejarea confidențialității datelor utilizatorilor și a informațiilor personale. Aceasta implică implementarea măsurilor de securitate adecvate pentru protejarea conturilor utilizatorilor.

4.3 Cerințe de Securitate

Cerințele de securitate ale aplicației cuprind implementarea unei autentificări securizate pentru utilizatori, protejarea datelor personale și criptarea informațiilor sensibile.

4.4 Atribute de Calitate Software

Acest software trebuie să fie cât mai lipsit de erori posibil pentru a se asigura că utilizatorii au o experiență pozitivă. Fiabilitatea este o prioritate, asigurând că aplicația funcționează în mod stabil și fără erori. Performanța optimă este de asemenea un obiectiv, oferind utilizatorilor o experiență fluidă și un timp de răspuns rapid. Ușurința în utilizare este realizată prin intermediul unei interfețe intuitive și ușor de navigat, facilitând accesul la funcționalități și interacțiunea fără dificultăți.

4.5 Reguli de Afaceri

Politica echipei de dezvoltare este să respecte toate codurile de conduită stabilite de Universitate.

5. Alte Cerinte

Anexa A: Glosar

- Bibliotecă virtuală: Aplicația dezvoltată pentru gestionarea și accesarea cărților în format digital.
- Autentificare: Procesul de confirmare a identității utilizatorului pentru a avea acces la funcționalitățile aplicației.
- Împrumut de carte: Acțiunea de a solicita și obține temporar o carte din biblioteca virtuală.
- Returnare de carte: Acțiunea de a restitu o carte împrumutată înapoi în biblioteca virtuală.
- Cont de utilizator: Un profil individual creat de către administrator pentru a accesa aplicația
- Avertizare: Alertă trimisă utilizatorului pentru a furniza informații cu privire la întârzierea returnării unei cărți
- Administrator: Un utilizator care are acces la funcționalități și acțiuni suplimentare în aplicație, cum ar fi gestionarea cărților și utilizatorilor.

Anexa B: Modele de Analiză

Diagrame UML

Actori multipli

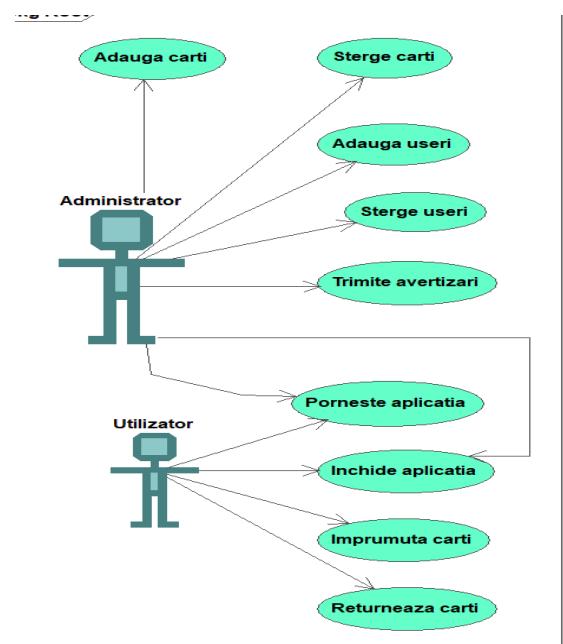


Diagrama de activități cu decizii

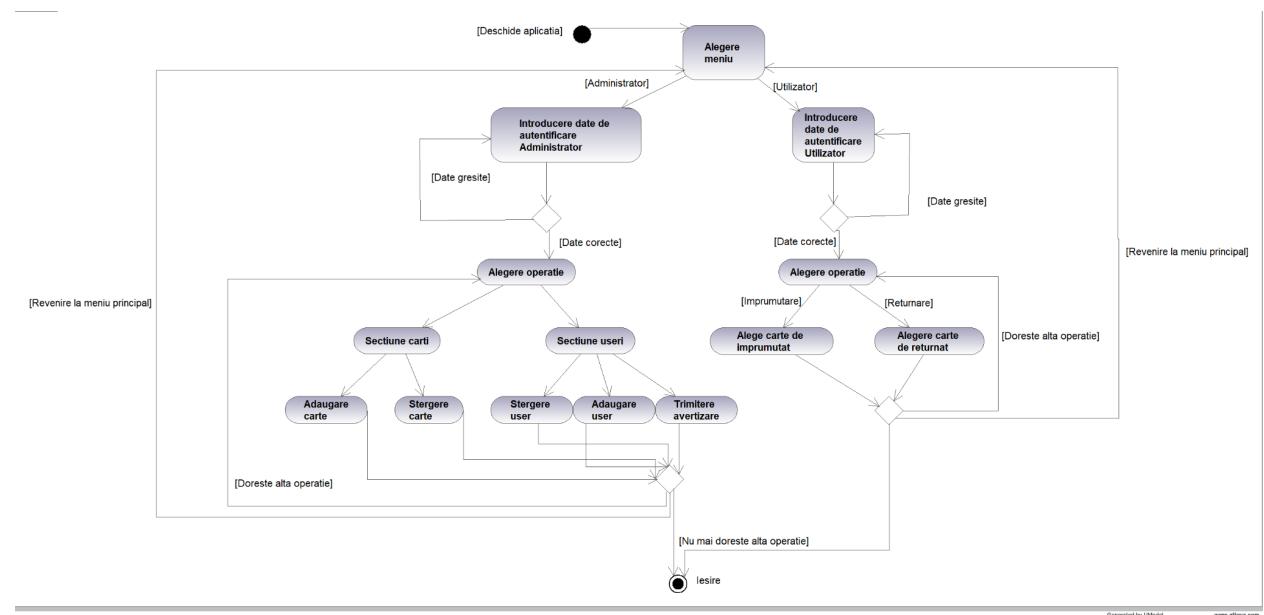
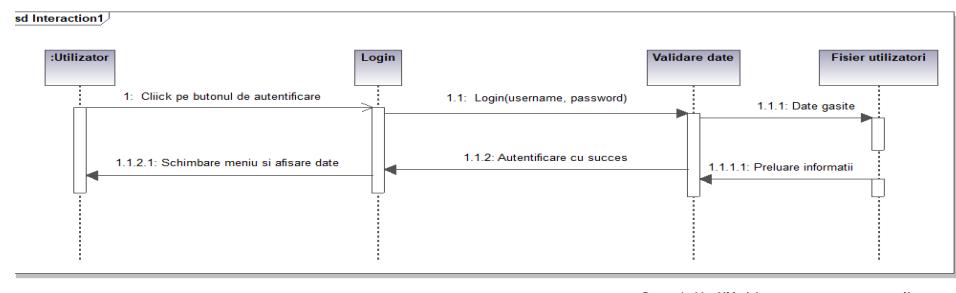
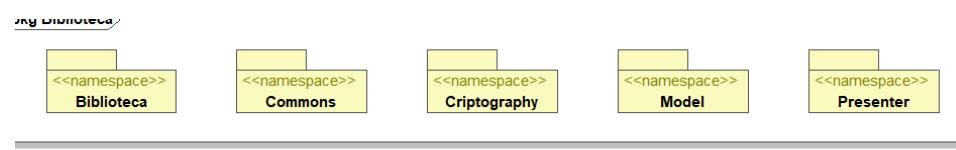


Diagrama de secvențe



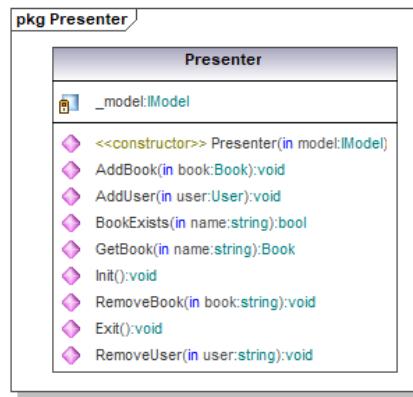
Conținutul Bibliotecii (packages)



Continut Commons



Continut Presenter



Generated by UModel

www.altova.com

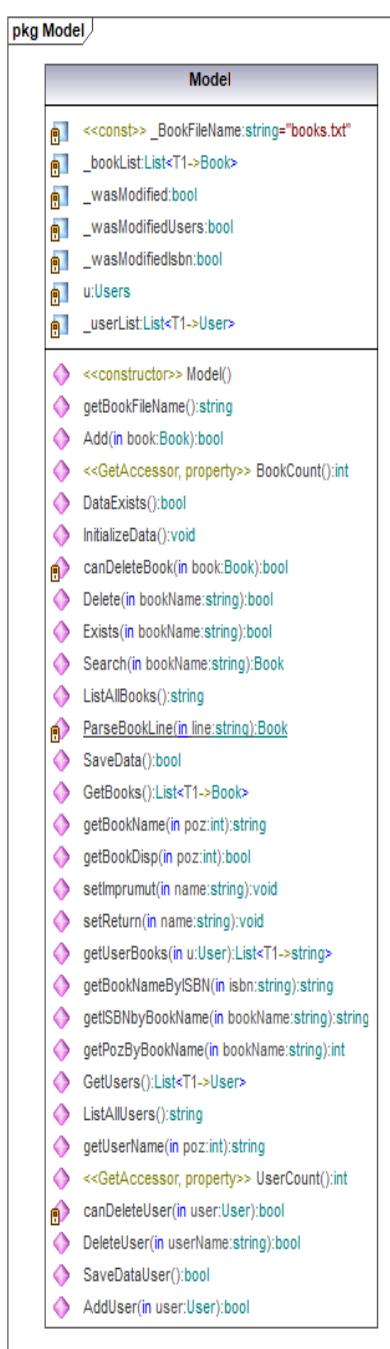
Continut Criptography



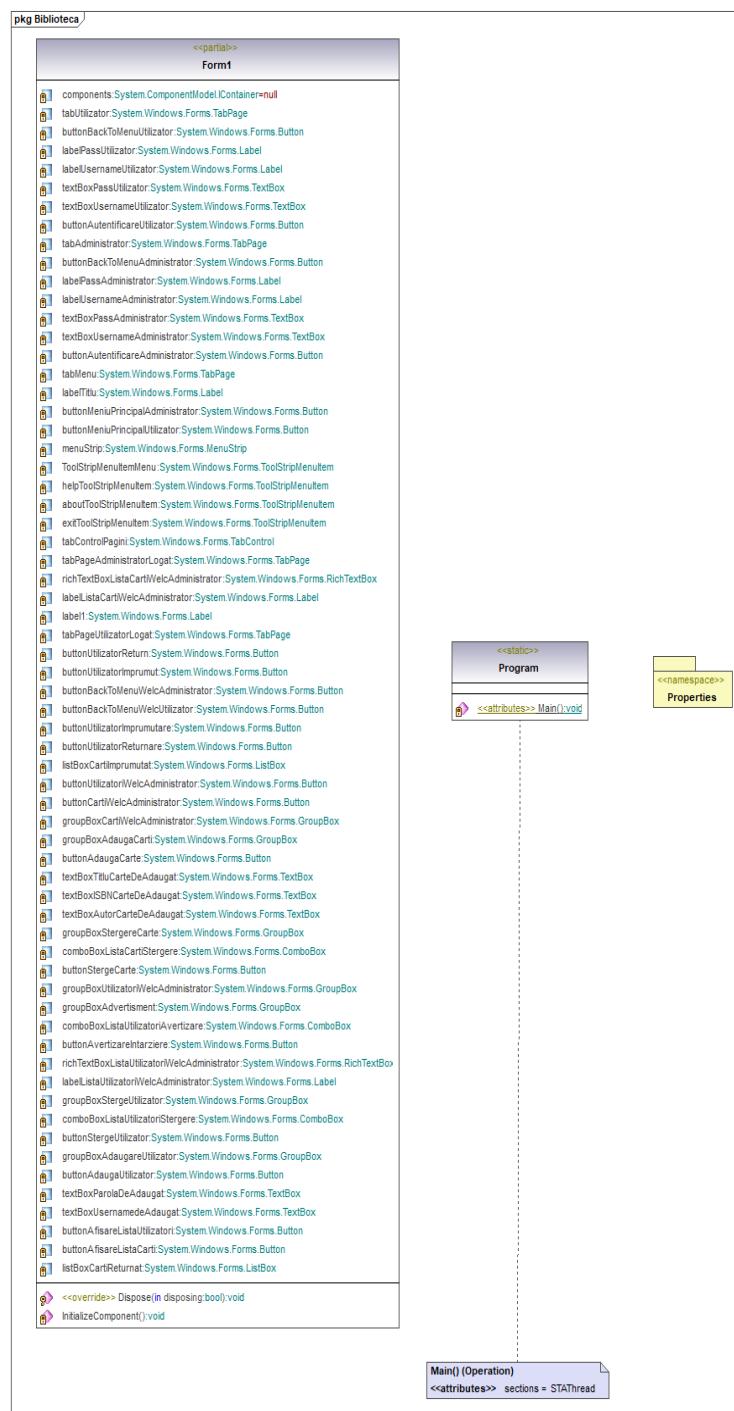
Generated by UModel

www.altova.com

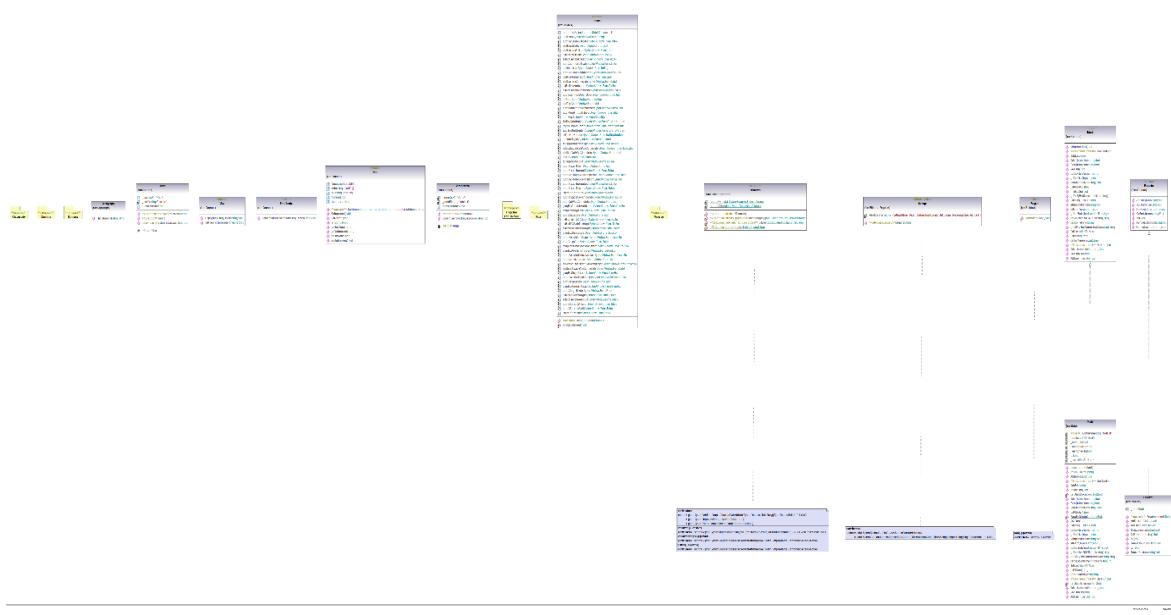
Conținut Model



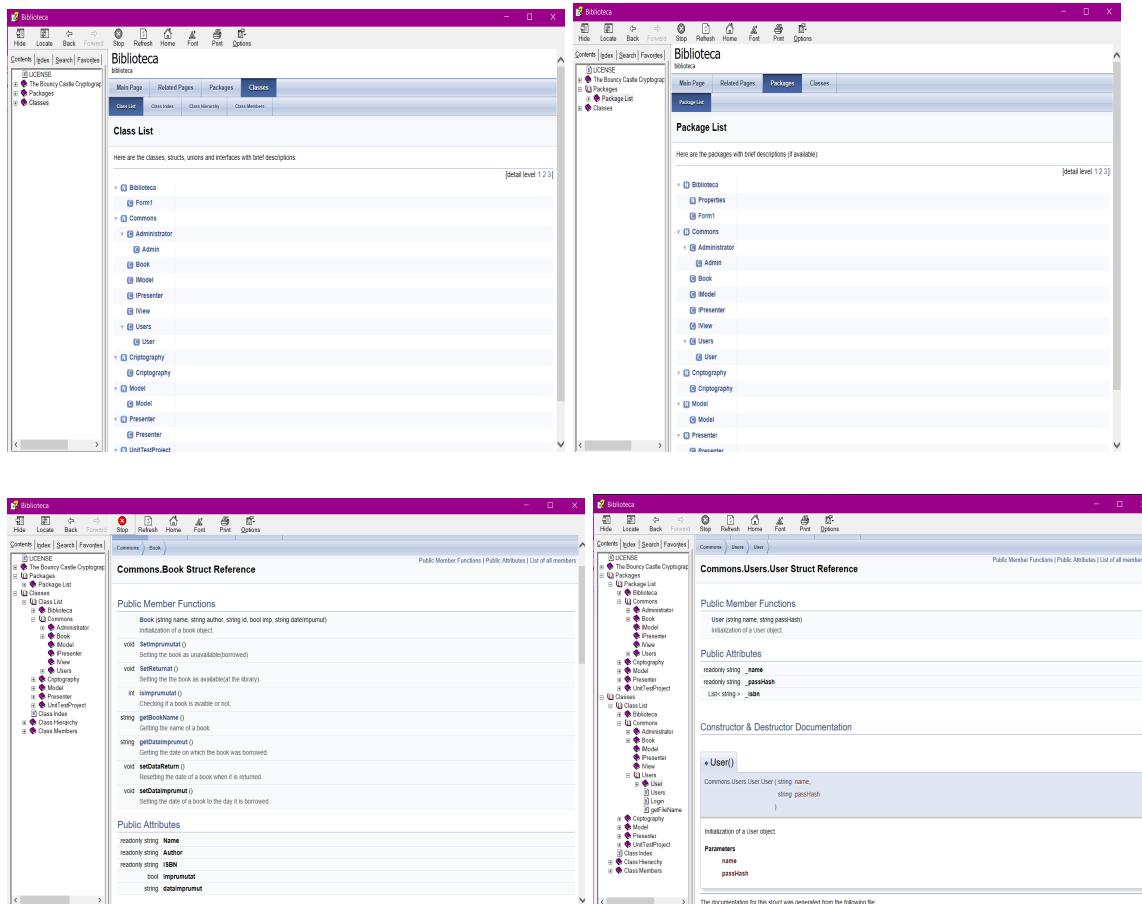
Conținut namespace Biblioteca



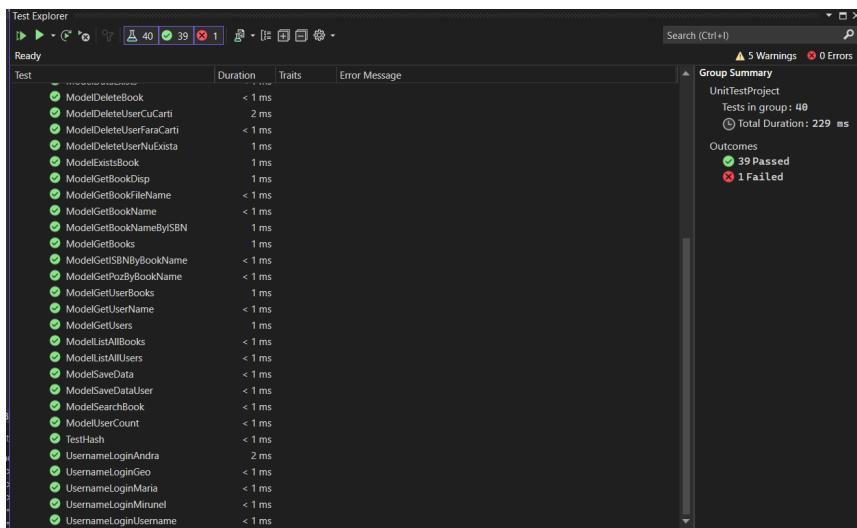
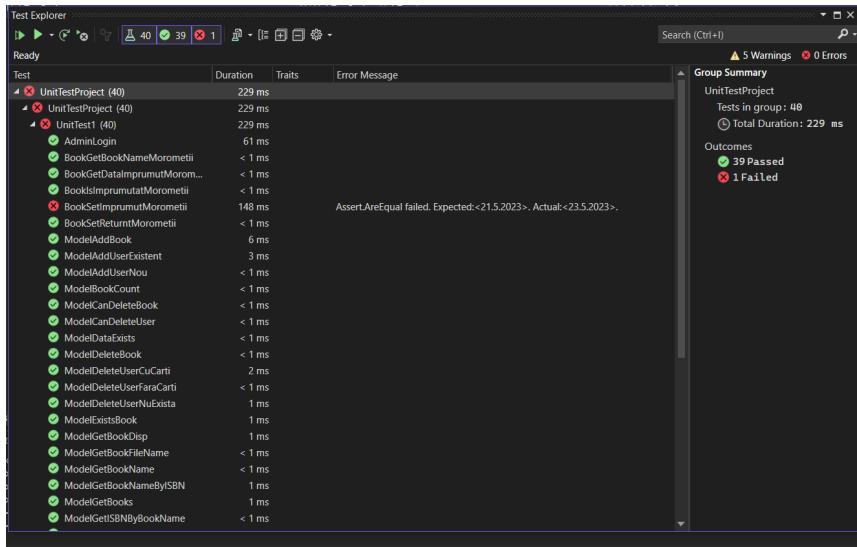
Conținut Bibliotecă



Documentatie Doxygen



Teste



Anexa C: Lista "Urmează să fie determinat"

TBD1: Crearea unei baze de date pentru a gestiona informațiile și relațiile dintre titluri și autori.

TBD2: Adăugarea de noi funcționalități în interfața utilizatorului, precum căutarea unei cărți sau filtrarea ei după categorie sau autor.

TBD3: Adaugarea conținutului cărților, nu doar a titlurilor.

Anexa D: Task-uri

Cojocaru Georgiana-Vasilica	Lupu Andra-Maria	Martin Maria	Petrescu-Miron Ştefanel
Interfața cu utilizatorul: meniu pentru logarea administratorului	Interfața cu utilizatorul: meniu pentru logarea utilizatorului și WelcomeUtilizator	Interfața cu utilizatorul: meniu principal și WelcomeAdministrator	Documentație Doxygen
Interfețe: IModel, IPresenter,	Clasa: User, Criptography	Interfața IView	Help utilizator
Clasa: Books, Presenter	Din clasa Model: 1. Pentru user: aflarea cărților unui user 2. Pentru cărți: aflarea titlului unei cărți după ISBN, aflarea ISBN-ului unei cărți după titlu	Clasa: EmailSender, Admin	Din clasa Form1: Implementat help în Strip Menu
Din clasa Model: 1. Partea de useri: adăugare, ștergere, salvare în fișier 2. Partea de cărți: disponibilitatea la afișare	Din clasa Form1: 1. Parte de user: autentificare, întoarcere la meniu principal, deschiderea meniului de împrumut al cărților, deschiderea meniului de returnare a cărților.	Din clasa Model: 1. Partea de cărți: verific dacă o pot șterge, ștergere, adăugare, aflarea poziției după nume	
Din clasa Form1: 1. Partea de împrumut și returnare	Comentarii cod	Din clasa Form1: 1. Toate butoanele care duc înapoi la meniu principal 2. Implementare Strip Menu (mai puțin help) 3. Implementarea meniului de cărți (adăugare, ștergere, afișare) și actualizarea lor 4. Implementare buton pentru avertizare	
Comentarii cod	Antete program	Comentarii cod	
Diagrame UML	Teste (clasa UnitTest1)	Documentație	

Anexa E: Părți semnificative din codul sursă

Partea de returnare carte

```
/// <summary>
    /// For when an user returns a book. We get rid of it in the users
file
    /// </summary>
    /// <param name="isbn"></param>
private void RetBookSaveIsbnList(string isbn)
{
    StreamWriter sw = new StreamWriter(_user.getFileName);
    for (int i = 0; i < _user._users.Count; i++)
    {
        Users.User b = _user._users[i];
        sw.Write(b._name + "\t" + b._passHash);
        //if its not the current user, we dont modify his line
        if (b._name != _user.currentUsername._name)
            for (int j = 0; j < b._isbn.Count; j++)
            {
                sw.Write("\t" + b._isbn[j]);
            }
        else
        {
            for (int j = 0; j < _user.currentUsername._isbn.Count;
j++)
            {
                // we skip writing the returned book
                if (_user.currentUsername._isbn[j] != isbn)
                    sw.Write("\t" + _user.currentUsername._isbn[j]);
            }
            sw.WriteLine();
        }
        sw.Close();
    }
}
```

Partea de imprumutare carte

```
/// <summary>
    /// For when an user borrows a book. We write it in the users file
    /// </summary>
    /// <param name="isbn"></param>
private void BorBookSaveIsbnList(string isbn)
{
    StreamWriter sw = new StreamWriter(_user.getFileName);
    for (int i = 0; i < _user._users.Count; i++)
    {
        Users.User b = _user._users[i];
        sw.Write(b._name + "\t" + b._passHash);
        //if its not the current user, we dont modify his line
        if (b._name != _user.currentUsername._name)
            for (int j = 0; j < b._isbn.Count; j++)
            {
                sw.Write("\t" + b._isbn[j]);
            }
        else
        {
            // if we find the current user, we add the new book in
            the file
            for (int j = 0; j < _user.currentUsername._isbn.Count;
j++)
            {
                sw.Write("\t" + _user.currentUsername._isbn[j]);
            }
        }
        sw.WriteLine();
    }
    sw.Close();
}
```

Partea de trimitere email

```
/// <summary>
    /// Setting up email sending details and sending email over the secure
connection using MailKit
    /// </summary>
    /// <param name="adresaDestinatar"></param>
    /// <param name="mesaj"></param>
public static void SendEmail(string adresaDestinatar, string mesaj)
{
    // Configurarea detaliilor de trimitere a emailului
    var message = new MimeMessage();
    message.From.Add(new MailboxAddress("Virtual Library",
"virtual.library.ip@gmail.com"));
    message.To.Add(new MailboxAddress("", adresaDestinatar));
    message.Subject = "Avertisment intarziere";
    message.Body = new TextPart("plain")
    {
        Text = mesaj
    };

    // Trimiterea emailului prin conexiunea securizata folosind
MailKit
    using (var client = new MailKit.Net.Smtp.SmtpClient())
    {
        client.Connect("smtp.gmail.com", 587,
SecureSocketOptions.StartTls);
        client.Authenticate("virtual.library.ip@gmail.com",
"geimdttwnfumywcb");
        client.Send(message);
        client.Disconnect(true);
    }
}
```

Partea de autentificare a unui utilizator

```
/// <summary>
/// User Login
/// </summary>
/// <param name="username"></param>
/// <param name="password"></param>
/// <returns>Return true or false depending if the login was
successful</returns>
public bool Login(string username, string password)
{
    foreach (User user in _users)
    {
        if (user._name == username && user._passHash ==
Criptography.Criptography.HashString(password))
        {

            currentUsername = new User(username,
Criptography.Criptography.HashString(password));
            if (user._isbn.Count > 0)
            {
                for (int i = 0; i < user._isbn.Count; i++)
                {
                    currentUsername._isbn.Add(user._isbn[i]);
                }
            }

            return true;
        }
    }
    return false;
}
```