

Management Biblioteca

06.01.2024



Îndrumător:

Dr.ing. Daniel Morariu

Student:

Urluescu Alexandre Joao

Istoric Versiuni

Data	Versiuni	Descriere	Autor
20.11.2023	1.0	Realizarea interfetei grafice (toate interfetele care alcatuiesc aplicatia)	Urluescu Alexandre Joao
16.12.2023	1.5	Realizarea Flow-urilor pentru fiecare interfata (clasa care au metode care ne folosesc in aplicatie)	Urluescu Alexandre Joao
20.12.2023	1.8	Conectare aplicatiei la o baza de date locala	Urluescu Alexandre Joao
05.01.2024	2.0	Implementarea interfetei grafice cu flow-urile realizate	Urluescu Alexandre Joao

Cuprins

1. Specificarea cerintelor software

- 1.1 Tehnologii folosite
- 1.2 Aplicatia de tip Form
- 1.2 Cerinte specifice

2. Crearea interfetei grafice

- 2.1. Descriere
 - 2.2. Principalele aspecte ale creării unei interfețe în C#
 - 2.3. Interfetele aplicatiei

3. Fluxul de evenimente

- 3.1 Fluxul de baza
- 3.2 Pre-conditii
- 3.3. Post-conditii

4. Crearea Flow-urilor din aplicatie

- 4.1 Descriere
- 4.2 Scopul
- 4.3 Separarea UX/UI de business logic

5. Conectarea la baza de date locala

- 5.1. Microsoft SQL Server Management Studio (SSMS)
- 5.2.Folosirea unei baze de date in cadrul aplicatie

5.3. Conectarea la aplicatie

6. Implementare

6.1. Diagrame de clase

6.2 Descriere detaliata

7. Bibliografie

1 Specificarea cerintelor software

1.1 Tehnologii folosite

Proiectul Management Biblioteca a fost dezvoltat in limbajul de programare C# in mediu de dezvoltare Visual Studio.

- **Structură și Sintaxă:** C# a fost proiectat pentru a combina eficient funcționalitățile limbajelor de programare C și C++ cu avantajele tehnologiilor moderne. Sintaxa sa asemănătoare cu C++ facilitează înțelegerea pentru programatorii familiarizați cu limbajele C.
- **Platformă și Integrare:** Este un limbaj orientat pe platformă, fiind utilizat în principal pentru dezvoltarea de aplicații pe platforma Microsoft .NET. C# este integrat strâns cu Framework-ul .NET, permițând dezvoltatorilor să creeze aplicații variate, inclusiv aplicații desktop, web și mobile.
- **Programare Obiect-Orientată:** C# este un limbaj puternic orientat pe obiect, permițând dezvoltatorilor să creeze aplicații bazate pe concepte precum clase, obiecte, moștenire și polimorfism. Această abordare facilitează dezvoltarea și întreținerea codului.

1.2 Aplicatia de tip form

Aplicatia de tip form pe care limbajul de programare C# il ofera ne pune la dispozitie urmatoarele

- **Interfață Grafică:** Formularele în C# oferă o modalitate vizuală de proiectare a interfeței utilizator (UI). Dezvoltatorii pot adăuga controale precum butoane, casete de text, etichete și altele pe formulare pentru a crea o interfață interactivă și intuitivă.
- **Evenimente și Acțiuni:** Formularele permit asocierea de evenimente la diferite acțiuni. De exemplu, un buton de pe formular poate declanșa un eveniment de clic, iar acest eveniment poate fi tratat în codul C# pentru a executa o anumită acțiune.
- **Colectare de Date:** Prin intermediul formularelor, aplicațiile pot colecta date de la utilizatori utilizând controale precum casete de text, meniuri dropdown sau casete de selectare. Aceste date pot fi apoi procesate și utilizate în logica aplicației.
- **Navigare și Control de Flux:** Formularele pot fi utilizate pentru a gestiona fluxul de control în cadrul aplicației. De exemplu, prin intermediul butoanelor "Înapoi" și "Înainte", utilizatorii pot naviga între diferite ecrane sau funcționalități ale aplicației.

Aceasta aplicația are ca scop multi-user, având 2 tipuri de utilizatori user și admin. Adminul poate adăuga cărți noi în aplicație, iar userii au posibilitatea de a le vizualiza și rezerva până la o anumită dată.

2. Crearea interfeței grafice

2.1. Descriere

Crearea unei interfețe într-o aplicație C# de tip form presupune proiectarea și implementarea unui cadru interactiv care facilitează interacțiunea utilizatorului cu aplicația. Acest proces implică utilizarea elementelor grafice și a controalelor disponibile în C# pentru a realiza o experiență utilizator intuitivă și plăcută.

2.2 Principalele aspecte ale creării unei interfețe în C# includ:

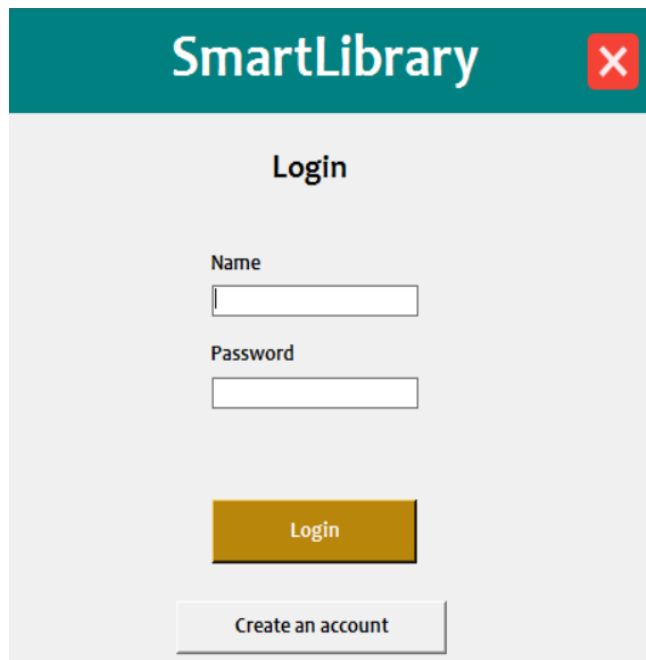
- **Design Vizual:** Proiectarea formularului înseamnă plasarea și configurarea elementelor grafice pe suprafața acestuia. Acestea pot include butoane, casete de text, etichete, meniuri și alte controale, aranjate astfel încât să ofere o structură coerentă și ușor de navigat.

- **Evenimente și Manipularea lor:** Controalele de pe formular sunt asociate cu evenimente specifice, cum ar fi clic, schimbare sau tastare. Dezvoltatorul definește cum se va comporta aplicația în răspuns la aceste evenimente, permițând execuția de acțiuni specifice sau modificarea interfeței în consecință.
- **Organizare și Flux de Navigare:** O interfață bine proiectată include o organizare logică a elementelor, facilitând navigarea utilizatorului. De exemplu, utilizarea de butoane de navigare sau tab-uri pentru a ghida utilizatorul între diferitele funcționalități ale aplicației.
- **Personalizare Estetică:** Posibilitatea de a personaliza aspectul formularului, inclusiv culorile, stilurile și fonturile, contribuie la crearea unei estetici care reflectă identitatea și scopul aplicației.
- **Manipularea Datelor:** Controale precum casetele de text sau combobox-urile permit introducerea și afișarea datelor. Dezvoltatorul implementează logica necesară pentru a gestiona aceste date, inclusiv validarea și stocarea lor în baza de date sau procesarea ulterioară.
- **Feedback și Validare:** O interfață eficientă oferă feedback clar utilizatorului, de exemplu, prin mesaje de eroare sau confirmare, în funcție de acțiunile întreprinse.
- **Adaptabilitate și Responsivitate:** Interfața trebuie să fie adaptabilă la diferite dimensiuni de ecrane și să ofere o experiență responsive pentru a asigura că aplicația poate fi utilizată corespunzător pe diverse dispozitive.

2.3. Interfetele aplicatiei

Login

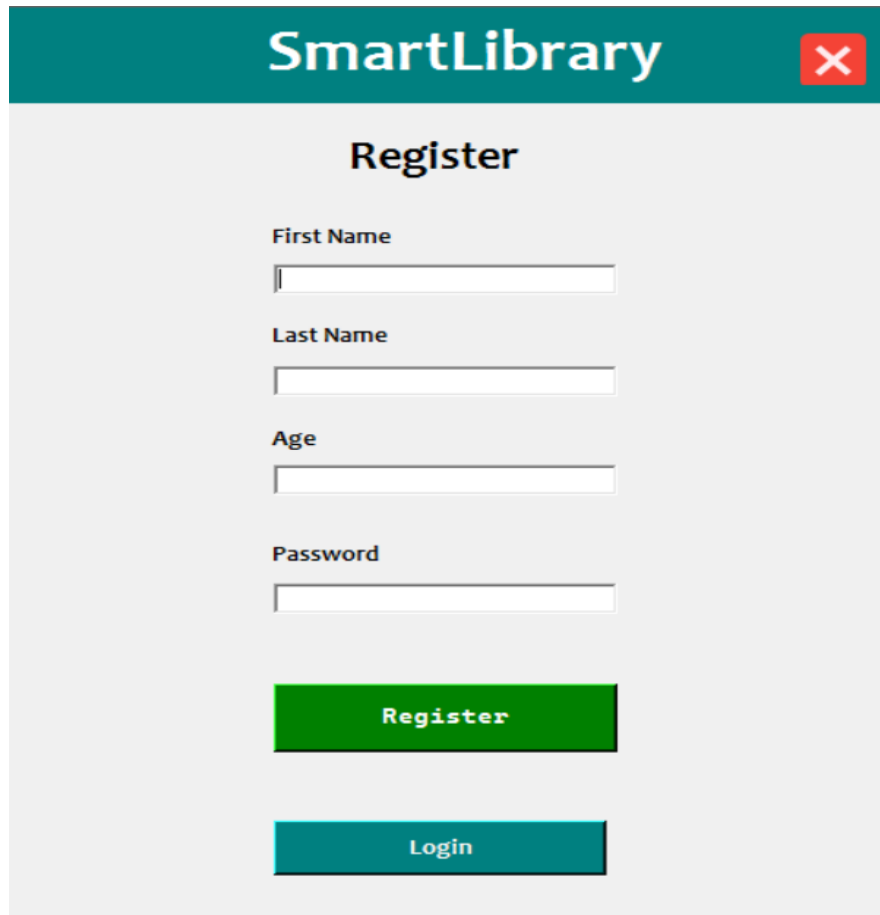
- pe aceasta interfata user-ul / admin-ul se pot autentifica in aplicatie
- In cazul in care autentificarea a decurs cu bine in functie de userName se va deschide o anumita parte din aplicatie.
- Pentru user - interfata de user, iar pentru admin - interfata de admin
- In cazul in care autentificarea nu a decurs bine, va aparea un pop-up sa avertizeze user-ul, iar el va trebui sa reincearca autentificare sau sa isi creeze un cont apasand CREATE AN ACCOUNT



The image shows a login form for 'SmartLibrary'. At the top, there is a teal header with the 'SmartLibrary' logo in white and a red square button with a white 'X' icon. Below the header, the word 'Login' is centered in bold. The form contains two input fields: 'Name' and 'Password', each with a white input box. Below these fields is a large orange 'Login' button. At the bottom of the form is a white button with a black border labeled 'Create an account'.

Register

- In fereastra Register user-ul isi poate crea un cont
- User-ul va completa formular-ul cu datele sale personale
- Daca totul a decurs corect va aparea un pop-up care descrie starea procesului. Iar el poate sa aleaga sa faca login cu contul creat, sau sa reincearca sa se registreze in caz ca au intervenit probleme in proces.



The image shows a web form titled "SmartLibrary" with a teal header bar. The form itself is titled "Register" and is set against a light gray background. It contains four input fields: "First Name", "Last Name", "Age", and "Password". Below these fields are two buttons: a green "Register" button and a teal "Login" button. A red close button (an 'X' in a red square) is located in the top right corner of the teal header bar.

SmartLibrary ✕

Register

First Name

Last Name

Age



Password

Register

Login

User Page

- UserPage-ul este fereastra dedicata entitatii de tip user in aplicatie.
- Primul tabel contine cartile available din aplicatie, mai exact cele care inca nu au fost rezervate de alti utilizatori
- Al doilea tabel contine toate cartile rezervate de user-ul care este autentificat in acel moment in aplicatie
- In momentul in care apasam un rand din tabelul Books Available, se vor auto-completa toate textbox-urile, iar user-ul poate sa aleaga cate zile vrea sa rezerve o anumita carte.
- Odata apasat butonul Rezerver va aparea, user-ul va primi un feedback despre starea actiunii.
- Daca totul a mers corect datele din tabele se vor actualiza.

 **SmartLibrary** 

Books Available

	id	bookName	bookAuthor	bookDate
▶	23	Ion	Liviu Rebreanu	
*				

My Books

	id	bookName	bookAuthor	bookUser	bookDate	book
▶	22	Moara	cccc	Alex	06-01-2024	11-
*						

Name

Author

Start Date

Id



Number Days

My Name

Reserve

Admin Page

- AdminPage-ul este fereastra dedicata entitatii de tip admin in aplicatie.
- Tabelul contine cartile care au fost adaugate in aplicatie, adica si cele care sunt inca nerezervate, dar si cele care sunt deja rezervate pana o anumita data de catre un anumit user.
- In momentul in care apasam un rand din tabelul Books Available, se vor auto-completa toate textbox-urile, iar adminul poate face un update la proprietatile Name si Author al unei carti sau sa dea delete. In acel moment va aparea un mesaj de feedback, iar daca totul a decurs corect se vor actualiza datele din tabel
- Adminul poate da create, completand casutele Name si Author urmat de click pe butonul CREATE. In acel moment va aparea un mesaj de feedback, iar daca totul a decurs corect se vor actualiza datele din tabel
- Butonul Clear are ca functionalitate de a curata toate textbox-urile

 **SmartLibrary** Admin 

Id

Name

Author

User Name

Start Date

End Date

	id	bookName	bookAuthor	bookUser	bookDate	bookEndDate
▶	22	Moara	cccc	Alex	06-01-2024	11-01-2024
	23	Ion	Liviu Rebreanu			
*						

Create

Update

Delete

Clear

3. Fluxul de evenimente

3.1. Fluxul de baza

Pentru a porni aplicatia este necesar deschidere fisierului *.sln, urmat de comanda run. Odata efectuata acesti pasi se va deschide prima fereastră a aplicatie, mai precis fereastră de Login. Daca user-ul este deja inregistrat in aplicatie, el se va putea loga cu user name-ul si parola, iar daca nu este inca inregistrat, va fi nevoit sa se inregistreze in fereastră Register.

Fereastră Register contine un formular, unde user-ul isi va adauga datele personale. Odata inregistrat in aplicatie, el va primi un feedback si va trebui sa se autentifice in fereastră de Login.

Aplicatia este multi-user, in sensul ca are 2 entitati de utilizatori admin si user. La autentificare in functie de userName, aplicatia va deschide o anumita interfata. In cazul in care utilizatorul este de tip user se va deschide fereastră de UserPage, iar in caz contrar se va deschide fereastră de AdminPage

3.2 Pre-conditii

Utilizatorul trebuie sa aiba un cont inregistrat in aplicatia, astfel incat sa beneficieze de toate facilitatile ei. Daca nu este inca inregistrat, va avea posibilitatea sa se inregistreze in fereastră Register si mai apoi sa se poate autentifica in pagina Login.

3.3 Post-conditii

Utilizatorul in functie de entitate va putea beneficia de facilitatile aplicatiei. Admin-ul poate adauga noi carti, iar user-ul isi poate rezerva cartile disponibile in aplicatie.

4. Crearea Flow-urilor din aplicatie

4.1 Descriere

Flow-urile sunt clasele create pentru a facilita interactiunea cu baza de date.

In aplicatie am 2 flow-uri principale:

- AppFlow care se ocupa de conexiunea la baza de date. Aceasta clasa are toate metodele pe care le folosesc pentru a interactiona cu baza de date
- BookFlow este o clasa care se ocupa in special de gestionarea datelor care vin din database de tip book

4.2 Scopul

- Separarea business logic de interfata grafica
- Reutilizarea codului in alte componente
- Faciliteaza debugging-ul

5. Conectarea la baza de date locala

5.1 Microsoft SQL Server Management Studio (SSMS)

Microsoft SQL Server Management Studio (SSMS) este un mediu avansat de administrare și dezvoltare pentru gestionarea bazelor de date Microsoft SQL Server. Cu o interfață grafică puternică și unelte avansate, SSMS facilitează administrarea și manipularea

eficientă a datelor într-un mediu SQL Server. Integrarea sa într-o aplicație C# se realizează prin intermediul conexiunilor și utilizării unor interogări SQL.

5.2 .Folosirea unei baze de date in cadrul aplicatie

Folosirea unei baze de date în cadrul unei aplicații aduce numeroase beneficii, contribuind semnificativ la eficiența, securitatea și scalabilitatea sistemului. Iată câteva motive pentru care este benefic să implementăm o bază de date într-o aplicație:

- Stocarea structurata a datelor
- Eficienta in manipularea datelor
- Consistenta si integritate
- Securitatea datelor
- Cautare si acces rapid
- Scalabilitate

5.3. Conectarea la aplicatie

Visual Studio ne ofera o multitudine de unelte astfel incat sa lucram cu baze de date.

In sectiunea Data Source putem alege sa ne conectam baza de date locala, apasand click dreapta pe Data Connections si mai apoi click pe add database.

Dupa toti acesti pasi mai trebuie sa adaugam numele serverului si numele bazei de date, iar in final putem vizualiza grafic tot ce contine baza noastra de date, de la tabele, pana la coloanele sale.

Visual Studio code ne ofera posibilitatea de a copia stringul de conectiune si al utiliza in aplicatia noastra. Acest string fiind adaugat in fisierul App.config care ne faciliteaza utilizarea string-ului in restul aplicatiei.

Integrarea interactivitatii intre baza de date si aplicatie se va face in flow-urile create, unde se gasesc metodele specifice care contine business logic-ul si comenzile sql. Aceste flow-uri fiind mai apoi legate la interfete grafice de tip Form, care ni le ofera C#.

6. Implementare

6.1. Diagrame de clase

Diagrama 1

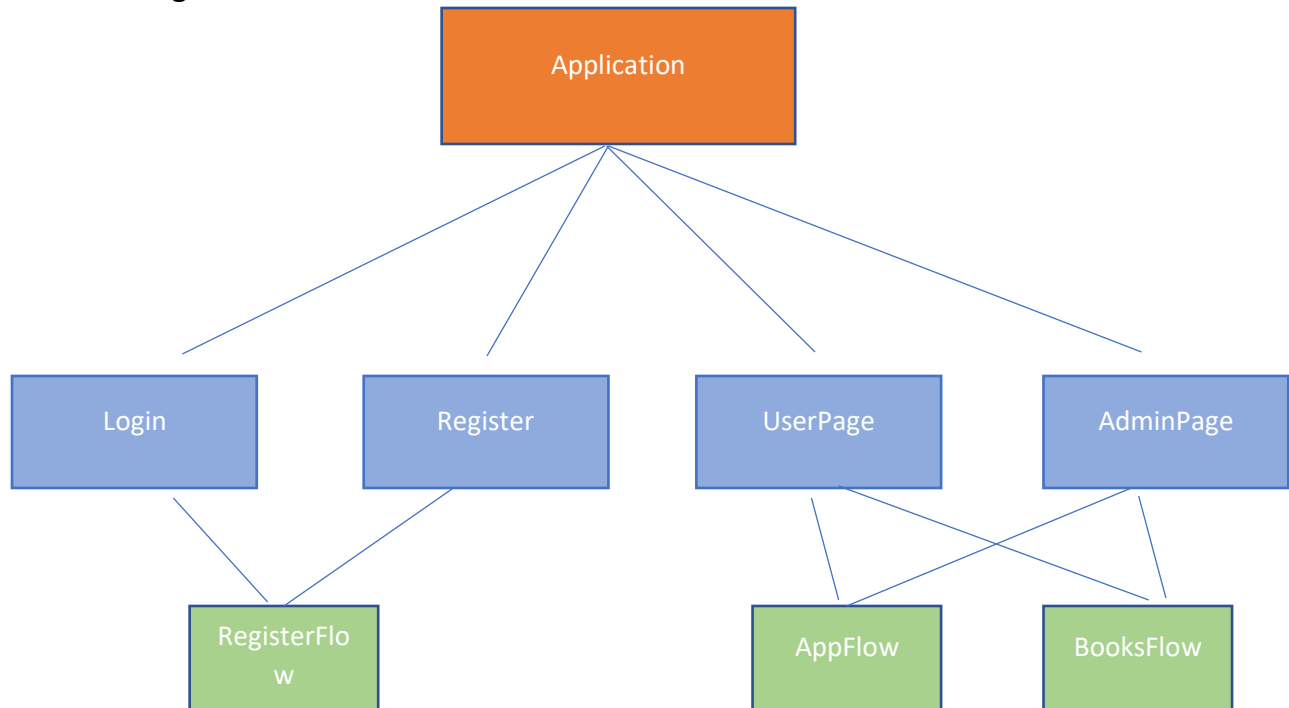
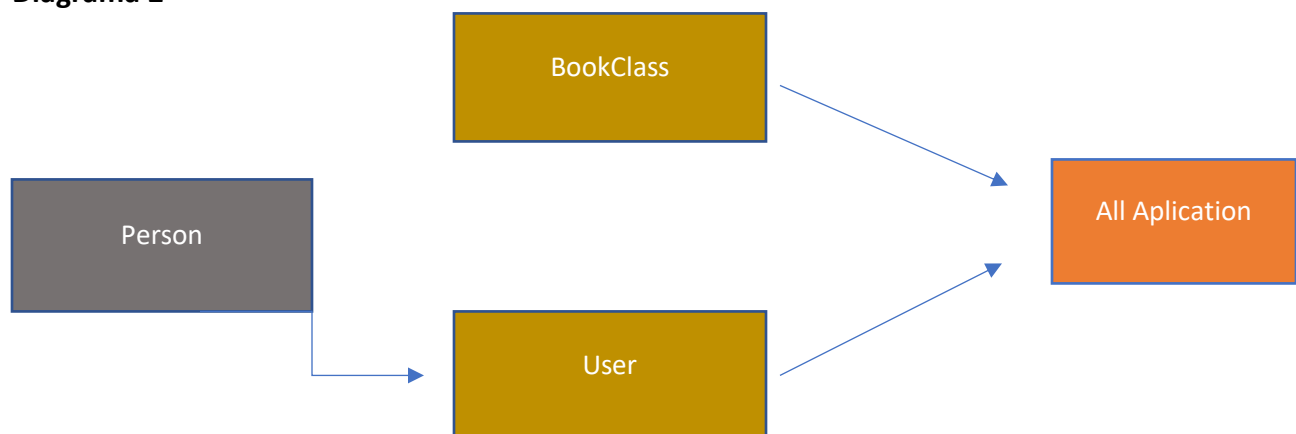


Diagrama 2



6.2 Descriere detaliata

Diagrama 1:

Intreaga aplicatie (culoare portocalie) este compusa de clase de tip Form (culoare albastru) create default de catre C#.

- Login
- Register
- UserPage
- AdminPage

In clasele acestea am dezvoltat eventurile care se produc asupra interfetei.

Pentru o separare cat mai buna a codului de business logic si codul de interfata am create alte clase (culoare verde) care se ocupa de metodele necesare pentru a interactiona cu database-ul.

- RegisterFlow
- AppFlow
- BookFlow

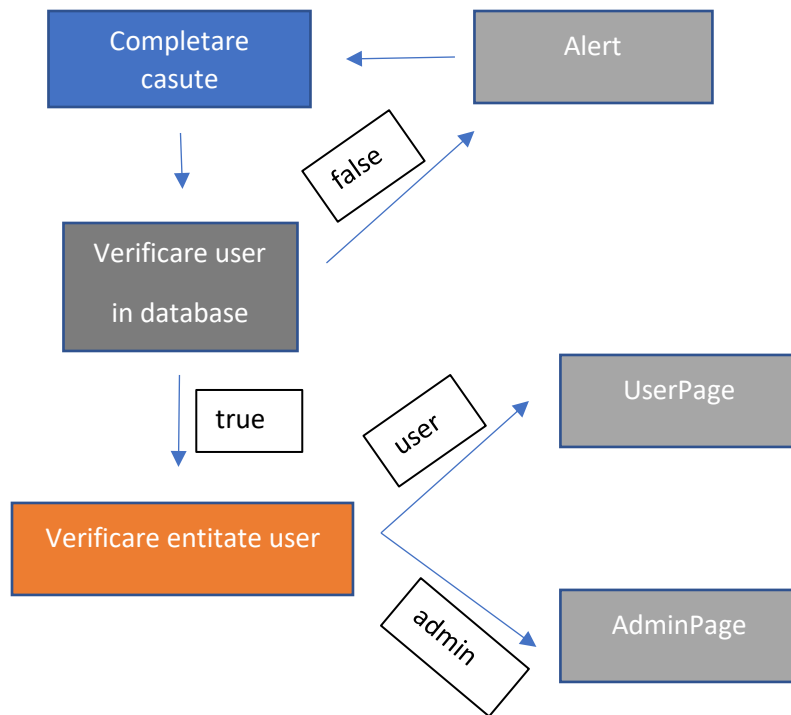
Diagrama 2:

In diagrama 2 putem vedea entitatile care alcituiesc aplicatia, mai exact:

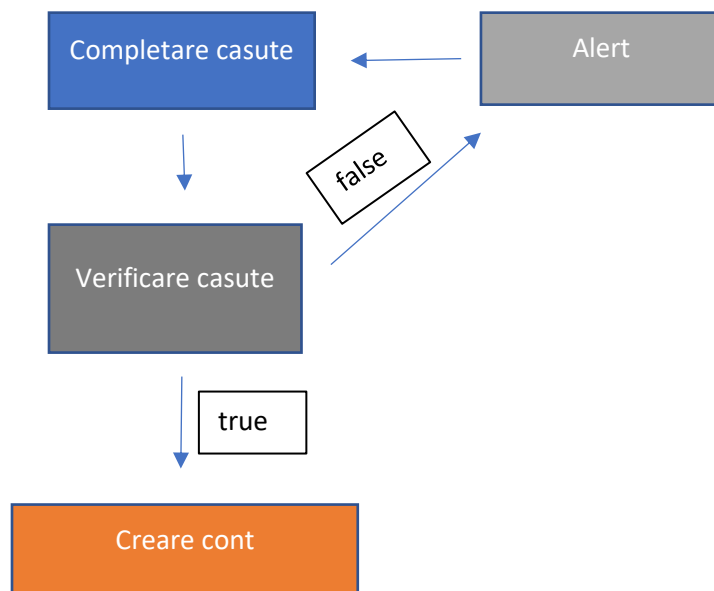
- Book
- Person
- User

Fiecare entitate are o clasa predefinita, iar entitatile acestea sunt folosite in toata aplicatia.

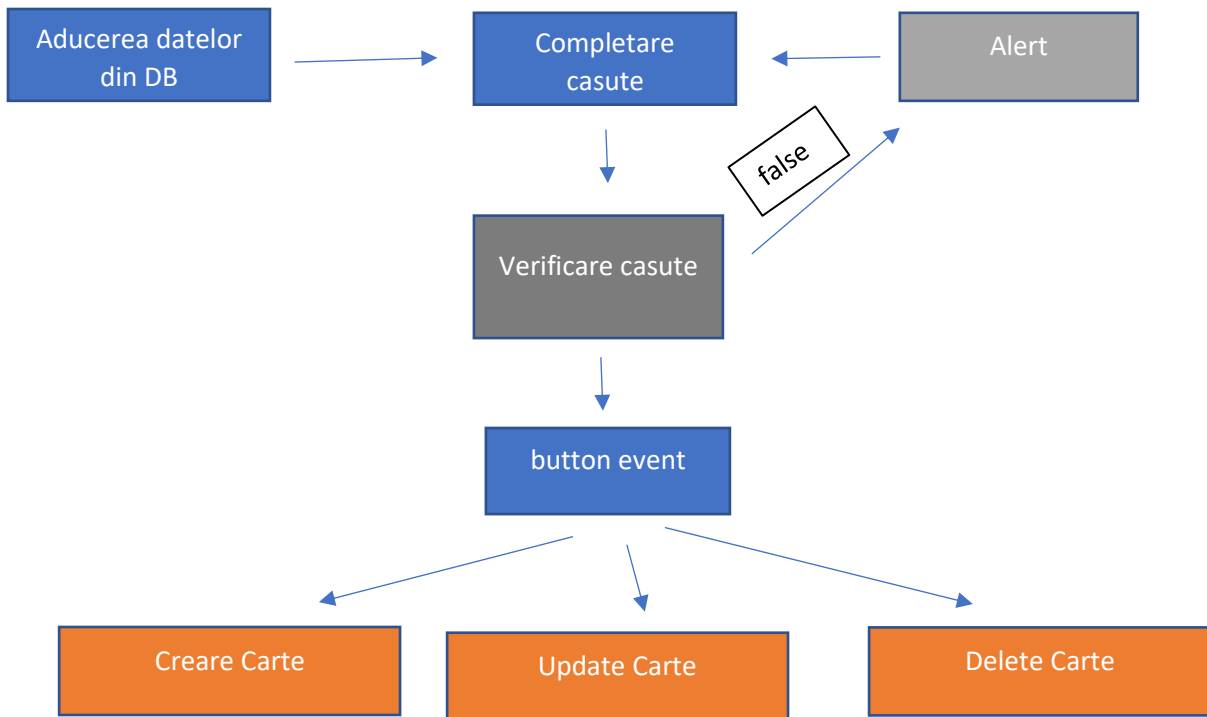
Login:



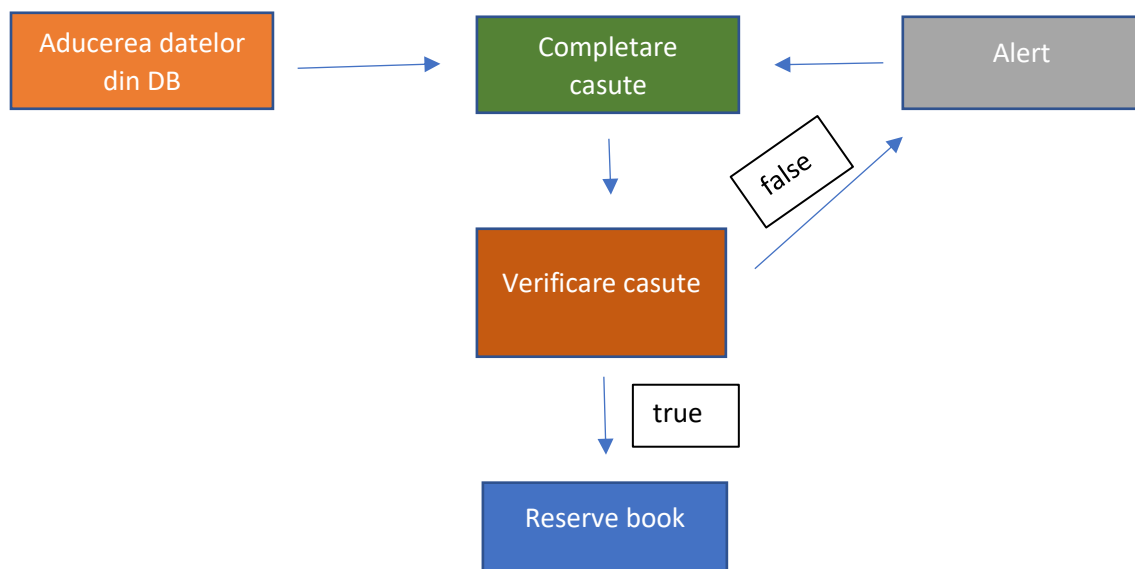
Register:



AdminPage:



UserPage:



7. Bibliografie

[Windows Forms for .NET 7 documentation | Microsoft Learn](#)

[SQL Server Management Studio \(SSMS\) - SQL Server Management Studio \(SSMS\) | Microsoft Learn](#)

[sql-docs/docs/sql-server/sql-server-offline-documentation.md at live · MicrosoftDocs/sql-docs · GitHub](#)

[Getting Started - Windows Forms .NET Framework | Microsoft Learn](#)