

Universitatea Tehnică „Gheorghe Asachi” din Iași

Facultatea de Automatică și Calculatoare

Domeniul: Calculatoare și Tehnologia Informației

Baze de date proiect

Sistem de gestiune al unei biblioteci

Coordonator:

asoc.ing.Baltariu Ionuț-Alexandru

Studentă:

Bîrleanu Andreea 1307B

Descrierea proiectului non-tehnică

Proiectul este o aplicație de gestionare a unei biblioteci. Aceasta permite organizarea cărților pe genuri, autori și edituri, gestionarea cititorilor și a împrumuturilor de cărți.

Fiecare carte este asociată unui singur autor și încadrată într-un singur gen literar, asigurând o organizare clară și evitând atribuiri multiple care ar putea duce la confuzii. De asemenea, fiecare carte provine de la o singură editură, iar informațiile suplimentare, cum ar fi anul publicării, limba sau descrierea care este în română indiferent de limba în care este scrisă cartea, sunt gestionate separat într-o secțiune dedicată pentru a evita supraîncărcarea.

Pentru împrumutul cărților, aplicația respectă reguli stricte: un cititor poate împrumuta o carte doar dacă există cel puțin două exemplare disponibile, astfel încât unul să rămână întotdeauna în bibliotecă. Dacă stocul este insuficient, împrumutul este refuzat până la reînnoirea acestuia. De asemenea, un cititor nu poate împrumuta din nou o carte pe care o deține deja, dar poate face mai multe împrumuturi pentru cărți diferite, fiecare înregistrare fiind gestionată printr-o fișă de împrumut separată.

Aplicația gestionează detaliile cititorilor, impunând unicitatea adreselor de email și a numerelor de telefon, pentru a preveni înregistrările duplicate. Mai mult, un cititor poate împrumuta o carte dacă este deja înscris în bibliotecă sau dacă se înscrie în aceeași zi în care dorește să efectueze împrumutul, dar returnarea cărții se poate face începând cu următoarea zi.

Astfel, aplicația vine în sprijinul bibliotecarilor, simplificând procesul de catalogare al cărților și de gestionare al împrumuturilor, precum și a relației cu cititorii.

Descrierea tehnică

(Baza de date)

1. Entitatea CARTE conține cărțile disponibile din bibliotecă:

- id_carte
 - obligatorie
 - NUMERIC(3)
 - Se folosește un număr întreg auto-incrementabil pentru a garanta unicitatea (autoincrementarea începe cu 100)
- titlu
 - obligatorie
 - VARCHAR(25)
 - Reprezintă titlul cărții înregistrate
- gen

- obligatorie
- VARCHAR(20)
- Reprezintă genul cărții înregistrate
- id_autor
 - cheie externă care face legătura între fiecare carte și autorul său din tabela autor
- id_editura
 - cheie externă care asigură legătura între carte și editura de unde provine

2. Entitatea DETALII_CARTI conține detalii suplimentare despre cărți:

- cantitate
 - obligatorie
 - NUMERIC(2)
 - Numărul de cărți disponibile în bibliotecă
- limba
 - obligatorie
 - VARCHAR(8)
 - Limba în care este scrisă cartea
- descriere
 - obligatorie
 - VARCHAR(255)
 - Prezintă o scurtă descriere a cărții în română indiferent de limbă
- an_publicare
 - neobligatorie pentru că anumite cărți mai vechi nu au un an de publicare clar definit
 - NUMERIC(4)
- id_carte
 - cheie externă pentru a lega fiecare detaliu al cărții de o anumită carte

3. Entitatea AUTOR conține autorii cărților disponibile în bibliotecă:

- id_autor
 - obligatorie
 - NUMERIC(3)
 - Se folosește un număr întreg auto-incrementabil pentru a garanta unicitatea (autoincrementarea începe cu 500)
- nume
 - obligatorie
 - VARCHAR(50)
 - Reprezintă numele întreg al autorului
- nationalitate

- neobligatorie pentru că există cazuri în care informațiile despre autori nu sunt complet disponibile
- VARCHAR(15)
- Țara de origine a autorului

4. Entitatea EDITURA conține editurile de la care provin cărțile:

- id_editura
 - obligatorie
 - NUMERIC(2)
 - Se folosește un număr întreg auto-incrementabil pentru a garanta unicitatea (autoincrementarea începe cu 1)
- nume
 - obligatorie
 - VARCHAR(35)
- adresa
 - obligatorie
 - VARCHAR(50)
 - Reprezintă adresa la care se găsește editura, incluzând strada și numărul
- telefon
 - obligatorie pentru contactarea editurii în cazul unei comenzi
 - VARCHAR(10)
- email
 - neobligatorie, nu e necesar să fie furnizat de către editură
 - VARCHAR(25)

5. Entitatea CITITOR stochează informațiile despre cititori:

- id_cititor
 - obligatorie
 - NUMERIC(4)
 - Se folosește un număr întreg auto-incrementabil pentru a garanta unicitatea (autoincrementarea începe cu 1000)
- nume
 - obligatorie
 - VARCHAR(15)
- prenume
 - obligatorie
 - VARCHAR(40)
 - Permite nume compuse

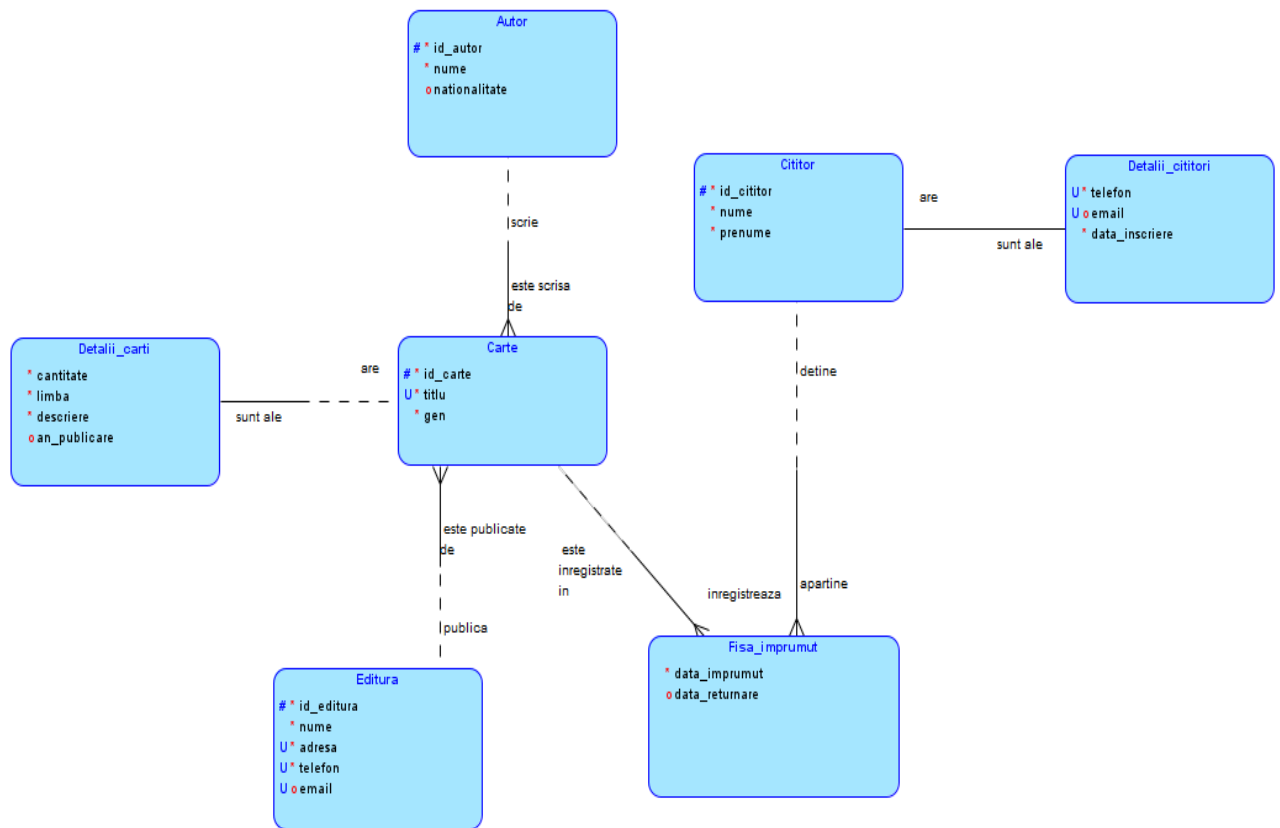
6. Entitatea DETALII_CITITORI conține informații suplimentare despre cititori:

- telefon
 - obligatorie pentru contactarea cititorului atunci când o carte este întârziată
 - VARCHAR(10)
- email
 - neobligatorie pentru că nu toți cititorii au o adresă de email sau nu-s dornici să o furnizeze
 - VARCHAR(25)
- data_inscriere
 - obligatorie
 - Default value: SYSDATE
 - DATE
- id_cititor
 - care face legătura între fiecare set de detalii suplimentare și cititorul corespunzător

7. Entitatea FISA_IMPRUMUT ține evidența tuturor împrumuturilor făcute în bibliotecă:

- data_imprumut
 - obligatorie
 - DATE
 - Default value: SYSDATE
- data_returnare
 - DATE
 - Se completează după aducerea cărții, neobligatorie
- id_cititor
 - cheie externă care asigură legătura între fiecare împrumut și cititorul corespunzător din tabela cititor
- id_carte
 - cheie externă care asociază fiecare înregistrare de împrumut cu cartea corespunzătoare din tabela carte

Structura si inter-relationarea tabelelor



Modelul logic

Normalizarea

Tabela Carte

1. Prima Formă Normală (1NF)

- Este în 1NF deoarece fiecare atribut conține valori atomice din domeniul său (și nu grupuri de astfel de valori): id_carte are o singură valoare pe rând, titlu este reprezentat de un singur șir de caractere, fiecare carte are un singur gen asociat. Cu ajutorul cheilor externe id_autor și id_editura nu se vor stoca aceleași informații de mai multe ori.

2. A Doua Formă Normală (2NF)

- id_carte este cheie primară, iar titlul și genul sunt atribute non-cheie. Nu există dependențe parțiale, deoarece ultimele două atribute depind în totalitate de cheia primară.

3. A Treia Formă Normală (3NF)

- Nu există nicio dependență tranzitivă între attributele non-cheie (titlul nu depinde de gen și nici invers), ambele depind direct de cheia primară.

id_carte	titlu	gen	id_autor	id_editura
100	Maitreyi	Literatura clasica	504	1
101	Arta subtila a nepasarii	Dezvoltare personala	502	2

Tabela Detalii_carti

1. Prima Formă Normală (1NF)

- Toate attributele sunt valori atomice: cantitate este un număr, limba un șir de caractere, descriere un text și an_publicare un număr. Nu există grupuri repetate sau seturi de valori multiple într-o singură coloană.

2. A Doua Formă Normală (2NF)

- id_carte este cheia externă, fiecare carte având o singură înregistrare în această tabelă. Această cheie este unică și simplă, deci nu există riscul de dependențe parțiale, toate attributele non-cheie (cantitate, limbă, descriere, an_publicare) depind în totalitate de aceasta.

3. A Treia Formă Normală (3NF)

- Attributele non-cheie depind direct de id_carte și nu depind între ele, astfel nu există dependențe tranzitive. Atributul descriere ar putea depinde de limbă, dar s-a impus la crearea acestei baze de date că descrierea va fi făcută în română indiferent de limbă.

cantitate	limba	descriere	an_publicare	id_carte
2	romana	dragoste	2015	100
5	romana	psihologic	2017	101

Tabela Autor

1. Prima Formă Normală (1NF)

- Este în 1NF deoarece fiecare atribut conține valori atomice: id_autor este cheia primară, iar nume și naționalitate reprezentate de un singur șir de caractere.

2. A Doua Formă Normală (2NF)

- Cheia primară este id_autor. Atributele non-cheie, nume și naționalitate, depind complet de cheia primară, neexistând dependențe parțiale.

3. A Treia Formă Normală (3NF)

- Atributele non-cheie nu depind unul de celălalt, deci nu există dependențe tranzitive.

id_autor	nume	nationalitate
500	Nicolae Iorga	Romania
501	Jane Austen	Marea Britanie

Tabela Editura

1. Prima Formă Normală (1NF)

- Este în 1NF deoarece fiecare atribut conține valori atomice: id_editura este cheia primară, iar adresă, nume, telefon și email reprezentate de un singur șir de caractere.

2. A Doua Formă Normală (2NF)

- Cheia primară este id_editura. Atributele non-cheie, adresă, nume, telefon și email, depind complet de cheia primară, neexistând dependențe parțiale.

3. A Treia Formă Normală (3NF)

- Atributele non-cheie nu depind unul de celălalt, deci nu există dependențe tranzitive.

id_editura	nume	adresa	telefon	email
1	Litera	Strada Moeciu 7A	0713196393	litera_contact @gmail.com
2	Trei	Strada Calea Victoriei 101A	0728049211	editura_trei@e mail.com

Tabela Cititor

1. Prima Formă Normală (1NF)

- Este în 1NF deoarece fiecare atribut conține valori atomice: id_cititor este cheia primară, iar nume și prenume reprezentate de un singur șir de caractere.

2. A Doua Formă Normală (2NF)

- Cheia primară este id_cititor. Atributele non-cheie, nume și prenume, depind complet de cheia primară, neexistând dependențe parțiale.

3. A Treia Formă Normală (3NF)

- Atributele non-cheie nu depind unul de celălalt, deci nu există dependențe tranzitive.

id_cititor	nume	prenume
1000	Birleanu	Andreea
1001	Coca	Mihai

Tabele Detalii_cititori

1. Prima Formă Normală (1NF)

- Este în 1NF deoarece fiecare atribut conține valori atomice: telefon și email reprezentate de un singur șir de caractere, data_inscriere formată dintr-o singură dată.

2. A Doua Formă Normală (2NF)

- id_cititor este cheie externă, fiecare cititor având o singură înregistrare în această tabelă. Această cheie este unică și simplă, deci nu există riscul de dependențe parțiale, toate atributele non-cheie (telefon, email, data_inscriere) depind în totalitate de aceasta.

3. A Treia Formă Normală (3NF)

- Atributele non-cheie depind direct de id_cititor și nu depind între ele, astfel nu există dependențe tranzitive.

telefon	email	data_inscriere	id_cititor
0730005530	a_birleanu14@gmail.com	03-DEC-24	1000
0734568832	cocamihai@gmail.com	18-DEC-24	1001

Tabela Fisa_imprumut

1. Prima Formă Normală (1NF)

- Este în 1NF deoarece atributele data_imprumut și data_returnare conțin valori atomice (și nu grupuri de astfel de valori). Nu există grupuri repetate sau seturi de valori multiple în acestea.

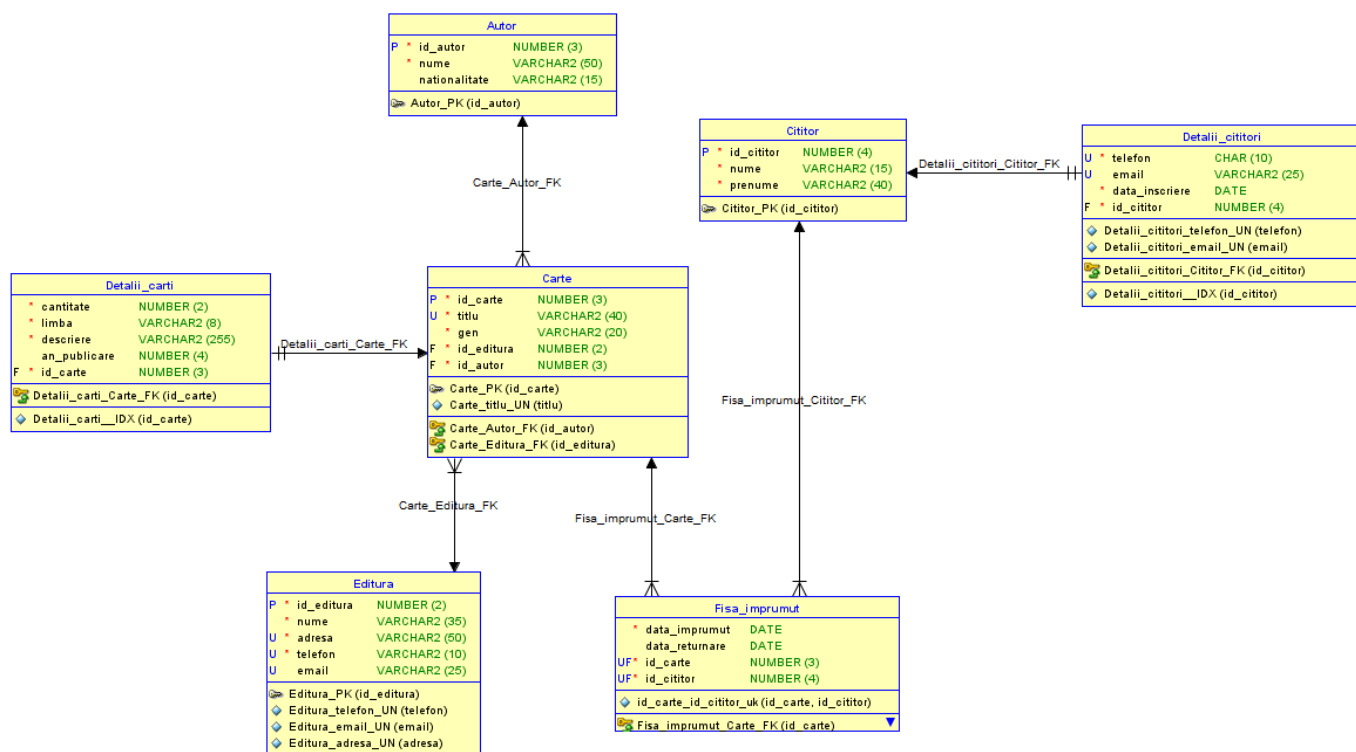
2. A Doua Formă Normală (2NF)

- Fiecare atribut non-cheie (data_imprumut și data_returnare) depinde complet de cheia primară compusă (id_cititor, id_carte). Nu există dependențe parțiale.

3. A Treia Formă Normală (3NF)

- Nu există dependențe tranzitive. Atributele non-cheie depind direct doar de cheia primară compusă (id_cititor, id_carte).
- Relația Carte și Cititor (M:N)

Fiecare carte poate fi împrumutată de mai mulți cititori, iar fiecare cititor poate împrumuta mai multe cărți. Această relație este modelată prin tabela intermediară Fisa_imprumut care conține FK: id_carte și id_cititor și adaugă informații suplimentare, precum data_imprumut și data_returnare.



Modelul relațional

Relațiile dintre tabele:

- Autor și Carte (1:N)

Fiecare autor poate avea scrise mai multe cărți în bibliotecă, dar o carte este scrisă de un singur autor. Această relație este reprezentată prin FK id_autor în tabela Carte.

Exemplu: Un autor cu `id_autor` = 505 are două cărți, fiecare identificată prin `id_carte` diferit (102, 103).

- Editură și Carte (1:N)

Fiecare editură poate avea publicate mai multe cărți, dar o carte aparține unei singure edituri. Această relație este reflectată prin FK `id_editura` în tabela `carte`.

Exemplu: O editură cu `id_editura` = 1 are trei cărți, fiecare identificată prin `id_carte` diferit (100, 105, 106).

- Carte și Detalii_carti (1:1)

Fiecare carte poate avea un set de detalii suplimentare, iar detaliile aparțin unei singure cărți. Tabela `Detalii_carti` include FK `id_carte` pentru această relație.

Exemplu: Cartea cu `id_carte` 100 are un set de detalii care include cantitate, limba, descriere și `an_publicare`.

- Carte și Fisa_imprumut (1:N)

Fiecare carte poate fi împrumutată de mai mulți cititori (mai multe înregistrări în tabela `Fisa_imprumut` pentru același `id_carte`), dar fiecare înregistrare din `Fisa_imprumut` se referă la o singură carte.

Exemplu: Cartea cu `id_carte` = 101 a fost împrumutată de doi cititori cu id-uri diferite (1002, 1003).

- Cititor și Fisa_imprumut (1:N)

Fiecare cititor poate avea mai multe împrumuturi (adică mai multe înregistrări în tabela `Fisa_imprumut`), dar fiecare înregistrare din `Fisa_imprumut` se referă la un singur cititor.

Exemplu: Cititorul cu `id_cititor` = 1000 a împrumutat cărțile cu id-uri diferite (100, 104).

Constrângeri

1. PRIMARY KEY

Garantează unicitatea fiecărei înregistrări într-o tabelă. Acest lucru este esențial pentru gestionarea eficientă a datelor și prevenirea duplicatelor. Acest tip de constrângere este aplicat tabelelor: `Carte` (`id_carte`), `Cititor` (`id_cititor`), `Editura` (`id_editura`), `Autor` (`id_autor`).

2. FOREIGN KEY

Constrângerile de tip FK asigură referențialitatea între tabele. De exemplu, în tabela Fisa_imprumut, coloanele id_cititor și id_carte sunt chei străine care garantează că aceste valori există deja în tabelele corespunzătoare. Aceeași situație se aplică și în tabelele Carte (cu FK id_autor și id_editura), Detalii_carti (cu FK id_carte) și Detalii_cititori (cu FK id_cititor). În absența acestora, ar fi foarte greu să ne asigurăm că toate relațiile între tabele sunt respectate și că nu există date invalide (de exemplu, împrumuturi care nu sunt asociate unui cititor valid).

3. UNIQUE KEY

Constrângerile UNIQUE sunt folosite pentru a preveni duplicarea valorilor într-o coloană. De exemplu, în tabela Detalii_cititori telefonul și emil-ul sunt unice pentru a evita existența mai multor cititori cu aceleași date de contact. La fel și în cazul tablei Editura, adresa, telefonul și email-ul sunt unice din același motiv. De asemenea, titlul în tabela Carte este unic, evitându-se înregistrarea cărților cu același titlu. Există o constrângere de acest tip și pe perechea (id_carte, id_cititor) în tabela Fisa_imprumut, astfel un cititor nu poate împrumuta o carte pe care o deține deja.

4. NOT NULL

Constrângerea NOT NULL asigură că o coloană nu poate conține valori NULL, adică nu poate fi lăsată necompletată atunci când o înregistrare este inserată. Această constrângere este folosită pentru a preveni crearea de înregistrări incomplete, care ar putea să ducă la erori sau la interpretări greșite ale datelor. Atributele care au acest tip de constrângere sunt precizate mai sus, în secțiunea Descrierea tehnică prin cuvintele “obligatorie” și “neobligatorie”.

5. CHECK

Constrângerea CHECK se folosește pentru a asigura că valorile introduse într-o coloană respectă o anumită condiție sau logică.

- Constrângerea pe numele editurii, autorului, naționalitatea acestuia și pe titlul cărții permite introducerea de grupuri de cuvinte, primul fiind cu literă mare urmat de litere mici, iar următoarele cuvinte permit secvențe de litere mari sau mici.
- Constrângerea pe atributul telefon din Detalii_cititori și Editura permite numere de telefon de zece cifre și primele două trebuie să fie “07”.
- Constrângerea pe cantitate asigură că valorile introduse se încadrează între 1 și 20.

- Constrângerea pe email verifică respectarea unui format cu o parte locală validă, urmată de simbolul “@”, un domeniu și un sufix de domeniu cu 2-4 caractere.
- Constrângerea pe limba în care este scrisă cartea permite cărți în: engleză, română, franceză, germană, rusă, italiană, spaniolă și franceză.
- Constrângerea pe an_publicare permite intervalul 1900-2024.
- Constrângerea pe gen asigură că acest atribut respectă o listă predefinită: Aventură, Dezvoltare personală, Istorie, Literatură clasică, Mister/Polițist și Roman de dragoste.
- Constrângerea pe adresa editurii permite numele străzii și numărul acesteia.
- Constrângerea pe numele cititorului un singur cuvânt care începe cu literă mare, iar prenumele permite mai multe cuvinte.

Triggere

- Trigger-ul pentru data_inscriere a unui cititor validează că aceasta nu poate fi în viitor.
- Trigger-ul pentru data_imprumut a unei cărți se activează când aceasta este din viitor sau este mai mică decât data_inscriere.
- Pentru a asigura actualizarea atributului cantitate, am implementat două triggere: unul se ocupă cu scăderea din cantitate cu o unitate la fiecare împrumut de o anumită carte, iar celălalt asigură adunarea la cantitate când cartea este returnată.

Use case-uri

1. Împrumutul de cărți

Acest proces implică verificarea existenței unui cititor pe baza telefonului. Dacă cititorul nu există, acesta este adăugat în baza de date împreună cu detaliile sale de contact. Ulterior, cititorul poate împrumuta o carte dintr-un anumit gen, iar împrumutul este înregistrat. După aceea, sunt afișate informațiile referitoare la împrumutul efectuat.

2. Returnarea de cărți împrumutate

Acest use case permite returnarea unei cărți împrumutate. Se verifică dacă cititorul are un împrumut activ pentru cartea pe care dorește să o returneze. Dacă da, se actualizează data returnării în baza de date. Apoi sunt afișate informațiile despre împrumutul returnat.

3. Actualizarea informațiilor unui cititor

În cazul în care un cititor își actualizează informațiile de contact (telefon, email), aceste informații sunt modificate în tabela `Detalii_cititori`. Se verifică dacă cititorul există, iar apoi se actualizează datele sale. La final, se confirmă actualizarea și se afișează informațiile noi.

4. Raport lunar: Cele mai împrumutate cărți

Acest use case generează un raport cu cele mai împrumutate cărți din ultima lună. Se numără împrumuturile pentru fiecare titlu de carte și se sortează cărțile în funcție de numărul de împrumuturi, pentru a obține o listă a celor mai populare titluri într-o perioadă de o lună.