

Nr. Înregistrare [REDACTED]

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII

[REDACTED]

[REDACTED]

[REDACTED]

**LUCRARE PRACTICĂ DE TESTARE A COMPETENȚELOR  
PROFESIONALE ÎN SPECIALITATEA INFORMATCĂ**

**Titlul lucrării**

**Aplicație pentru gestionarea unei biblioteci**

**Candidat,**

**David Alex Robert**

**Profesor coordonator,**

[REDACTED]

**2024**

# Cuprins

<b>1</b>	<b>Introducere</b>	<b>2</b>
1.1	Motivarea temei . . . . .	2
1.2	Tehnologii folosite . . . . .	2
1.2.1	Tauri . . . . .	2
1.2.2	Rust . . . . .	2
1.2.3	Diesel . . . . .	3
1.2.4	React . . . . .	3
1.2.5	SQLite . . . . .	3
1.2.6	Canva . . . . .	3
<b>2</b>	<b>Mediul de programare</b>	<b>3</b>
2.1	WebStorm . . . . .	3
2.2	RustRover . . . . .	4
<b>3</b>	<b>Prezentarea aplicației</b>	<b>4</b>
3.1	Diagrama ERD . . . . .	4
3.2	Funcționalități generale . . . . .	5
3.2.1	Operațiunile CRUD asupra clienților . . . . .	5
3.2.2	Operațiunile CRUD asupra cărților . . . . .	6
3.2.3	Modificarea setărilor de către un administrator . . . . .	7
3.2.4	Căutarea de cărți în baza de date OpenLibrary . . . . .	7
3.3	Funcționalități specifice . . . . .	8
3.3.1	Scanarea codurilor de bare . . . . .	8
3.3.2	Gestionarea ID-urilor clienților . . . . .	8
3.3.3	Generarea de coduri de bare. Legitimății . . . . .	9
<b>4</b>	<b>Concluzii</b>	<b>10</b>
	<b>Bibliografie</b>	<b>11</b>

# 1 Introducere

## 1.1 Motivarea temei

În lumea modernă, cu tehnologia avansată care ne înconjoară, biblioteca tradițională și-a găsit un partener neașteptat în programare. Dezvoltarea unui program de gestionare a unei biblioteci s-a dovedit a fi un pas esențial în evoluția și adaptarea acestei instituții venerabile la noile cerințe și nevoi ale comunității.

La prima vedere, motivul pentru care am decis să creez acest program pare simplu - îmbunătățirea eficienței. Și totuși, în spatele acestei alegeri se află o dorință profundă de a conserva și promova cultura și educația într-un mod accesibil pentru toată lumea.

Această aplicație își propune, în primul rând, să asigure practicalitatea și ușurința utilizării pentru orice bibliotecar. Cu o interfață modernă, programul conceput profită de faptul că fiecare carte cu *ISBN* are garantat un cod de bare asociat. Deși pare simplu în principiu, înlocuirea tastării de cifre cu o simplă scanare economisește timp și previne erori de transcriere.

Al doilea scop al aplicației a fost organizarea eficientă a datelor. Cu posibil mii de cărți și sute de clienți, era necesară adoptarea unui model de date scalabil pentru cerințele din viitor. Astfel, resursele pot fi gestionate mai eficient, iar cărțile pot fi localizate rapid și ușor. În acest fel, biblioteca devine un loc mai ordonat și mai plăcut atât pentru cititori cât și pentru personal.

Dar poate cel mai important beneficiu al acestui program este gestionarea împrumuturilor de cărți. Cu funcții precum prelungirea termenului și notificări de întârziere, aplicația se asigură că biblioteca nu mai este doar o instituție care se ocupă cu împrumutul cărților, ci devine o parte integrantă a vieții de zi cu zi a comunității, oferind servicii și resurse care să satisfacă nevoile și interesele individuale ale fiecăruia.

În final, alegerea de a realiza un program de gestionare a bibliotecii nu a fost doar o decizie practică, ci și una inspirată de dorința de a aduce biblioteca în secolul XXI și de a asigura că rămâne un loc viu și relevant pentru toți cei care îi trec pragul. Este o călătorie în lumea digitală, în care tradiția și tehnologia se întâlnesc pentru a crea ceva cu adevărat special și valoros pentru întreaga comunitate.

## 1.2 Tehnologii folosite

### 1.2.1 Tauri

*Tauri* este un kit de dezvoltare pentru aplicații desktop, ce permite utilizarea aceluiași cod sursă pentru a crea aplicații destinate tuturor sistemelor de operare<sup>[1]</sup>. Spre deosebire de alți competitori de pe piață, precum *Electron*, acest kit se folosește de limbajul *Rust* pentru interacțiunea cu sistemul de operare, iar interfața este construită cu ajutorul tehnologiilor *web*. Acest duo are un avantaj evident: aplicațiile sunt performante și cu un consum redus de memorie și spațiu de stocare.

### 1.2.2 Rust

*Rust* este un limbaj de programare conceput în 2006 cu prima versiune stabilă apărută în 2015<sup>[2]</sup>. Acest limbaj pune accentul pe performanță și integritate a memoriei, fiind astfel un competitor pentru familia *C/C++* în dezvoltarea sistemelor de operare și integrate.

Un lucru important despre *Rust* este modul în care se asigură că referințele în memorie sunt valide, fără a folosi un concept al limbajelor de nivel înalt, numit *garbage collector*. Un *garbage collector* are scopul de a verifica și elibera referințele în memorie odată ce nu mai sunt utilizate pe parcursul

execuției<sup>[3]</sup>. *Rust* are o filozofie diferită, folosind un *borrow-checker* pentru a verifica încă din etapa de compilare a programului că referințele și durata lor de viață sunt valide<sup>[2]</sup>. Acest lucru este posibil datorită regulilor impuse în ceea ce privește adresarea la memorie, duratele de viață ale referințelor și a altor aspecte relevante pentru a asigura integritatea memoriei.

### 1.2.3 Diesel

*Diesel* este un *Object-Relational Mapper*, o librărie concepută pentru limbajul *Rust* cu scopul de a facilita scrierea de interogări *SQL* fără manipularea manuală de șiruri de caractere<sup>[4]</sup>. *Diesel* poate defini tipuri de date care reflectă tabelele bazei de date, unde câmpurile corespund coloanelor, iar toate interogările se construiesc prin apeluri de funcții, eliminând complet concatenarea sau interpolarea de valori în șiruri de caractere. Această librărie este avantajoasă în lucrul cu relații complexe, eliminând o posibilă sursă de erori în timpul dezvoltării aplicației.

### 1.2.4 React

*React* este o librărie pentru limbajul *JavaScript*, dezvoltată de *Meta*, pentru realizarea de aplicații *web* dinamice<sup>[5]</sup>. Această librărie folosește *JSX*, un dialect asemănător cu *HTML* pentru construirea aplicației, dar care permite crearea de componente reutilizabile<sup>[6]</sup>. Deși construirea unui website static în mod convențional, folosind multiple fișiere *HTML* și scripturi de *JavaScript* era o variantă de realizare a interfeței, am optat pentru *React* deoarece facilitează un ecosistem de alte librării și oferă programatorului unelte necesare pentru a construi o aplicație dinamică cu ușurință.

### 1.2.5 SQLite

*SQLite* este o bază de date relațională, concepută pentru a fi integrată în diferite aplicații<sup>[7]</sup>. Spre deosebire de alte dialecte *SQL*, *SQLite* asigură o viteză superioară de executare a interogărilor și încarcă din memorie doar informațiile necesare operației. Din aceste motive, *SQLite* este alegerea potrivită pentru o bază de date relațională în contextul aplicației dezvoltate, fiind ușor de folosit și integrat.

### 1.2.6 Canva

*Canva* este un editor *web* folosit în domeniul designului, punând la dispoziție o multitudine de unelte, imagini și iconițe pentru realizarea de materiale publicitare, precum logo-uri, coperti sau afișe<sup>[8]</sup>. Spre deosebire de un software profesional de editare, precum *Photoshop*, *Canva* pune la dispoziție șabloane pentru eficiența în crearea documentelor.



Figura 1: Logo-ul aplicației realizat în *Canva*

## 2 Mediul de programare

### 2.1 WebStorm

*WebStorm* este un mediu de programare integrat dezvoltat de compania *JetBrains* exclusiv pentru tehnologii *web*<sup>[9]</sup>. Facilitează crearea de aplicații *web*, website-uri, servere și integrează cu ușurință folosirea bazelor de date. Printre particularitățile generale ale mediilor de programare de la *JetBrains* se numără abilitatea de căutare avansată prin proiecte, unelte pentru redenumirea,

mutarea și copierea claselor, variabilelor și a altor identificatori, dar și analiza avansată a sintaxei și unelte de depanare.

## 2.2 RustRover

*RustRover* este un mediu de programare integrat dezvoltat tot de *JetBrains* pentru limbajul *Rust*<sup>[9]</sup>. Acest mediu este încă în versiunea alfa, dar oferă aceleași unelte precum *WebStorm*. Fiind dedicat limbajului *Rust*, acesta avantajează analizarea codului și a sintaxei, oferind mai multe facilități dedicate comparativ cu un plugin.

## 3 Prezentarea aplicației

### 3.1 Diagrama ERD

Este esențial pentru aplicație ca baza de date să stocheze eficient datele în tabele și pentru a le accesa facil. Clienții unei biblioteci pot împrumuta mai multe cărți, iar cărțile pot fi împrumutate de mai mulți clienți în limita exemplarelor, astfel se creează o relație de tip „mai mulți-la-mai mulți” care trebuie eliminată printr-o tabelă de intersecție, care, pe lângă cheile primare ale cărții împrumutate și a clientului aferent, stochează și data împrumutului și cea a returnării. Deoarece aceeași carte poate fi împrumutată de mai multe ori de același client, perechea de chei străine nu poate forma cheia primară a împrumutului, deci am convenit la folosirea unei chei artificiale, astfel tabela are și rol de a stoca istoricul de împrumuturi a unui client sau carte.

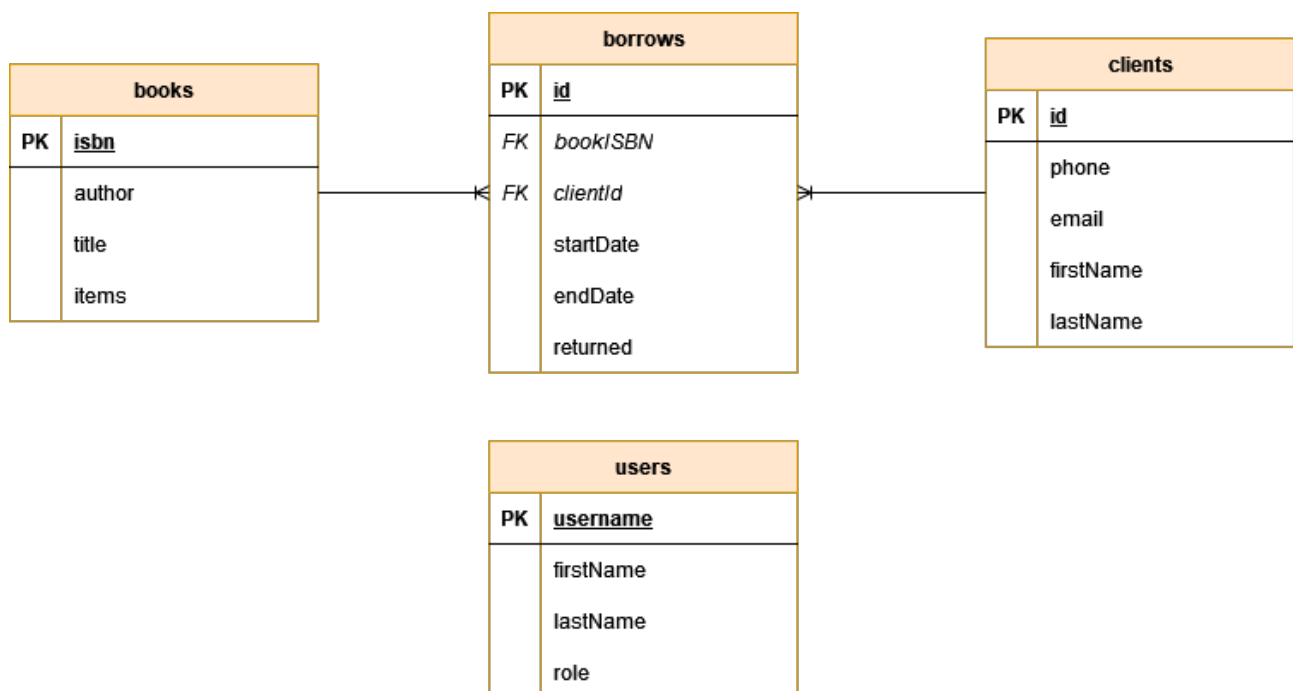
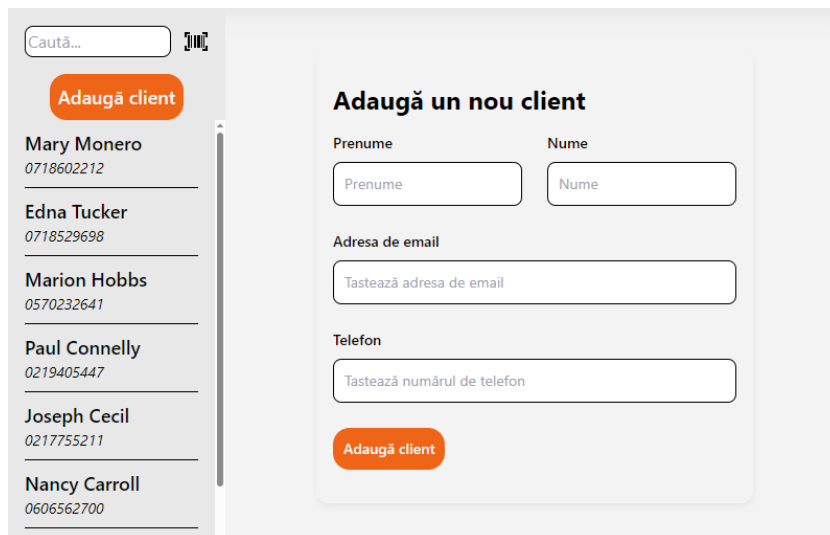


Figura 2: Tabelele utilizate

## 3.2 Funcționalități generale

### 3.2.1 Operațiunile CRUD asupra clienților

Pe pagina clienților, coloana din stânga ne permite filtrarea și accesarea datelor acestora, dar și adăugarea unui nou client.



The screenshot displays a web interface for managing clients. On the left, there is a sidebar with a search bar labeled 'Caută...' and a list of clients: Mary Monero (0718602212), Edna Tucker (0718529698), Marion Hobbs (0570232641), Paul Connelly (0219405447), Joseph Cecil (0217755211), and Nancy Carroll (0606562700). An orange button labeled 'Adaugă client' is positioned above the list. On the right, a form titled 'Adaugă un nou client' contains input fields for 'Prenume' and 'Nume', an 'Adresa de email' field, and a 'Telefon' field. An orange button labeled 'Adaugă client' is at the bottom of the form.

Figura 3: Formular pentru adăugat clienți

Formularul verifică corectitudinea datelor: adresa de email să fie validă, iar numărul de telefon să fie format din 10 cifre și să înceapă cu 0. La apăsarea butonului se va verifica în baza de date ca adresa de mail și numărul de telefon să fie unice.

Pe pagina fiecărui client putem accesa o multitudine de informații precum istoricul de împrumuturi și împrumuturile active, pe lângă datele personale adăugate prin formular.

Putem apăsa butonul „Editează” pentru a fi aduși pe o pagină cu un formular asemănător paginii de adăugare.

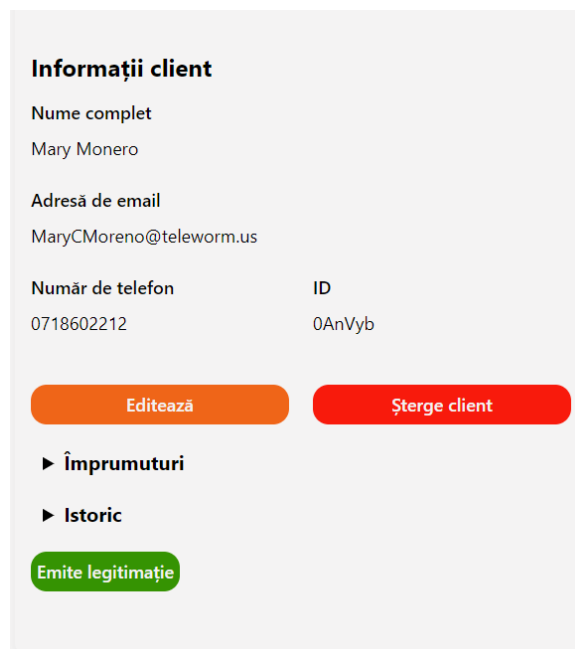
Butonul „Șterge client” va șterge recursiv și toate împrumuturile și istoricul pe care le-a avut, pe lângă datele acestuia.

Butonul „Emite legitimație” generează o imagine care conține codul de bare aferent clientului, ce poate fi scanat pentru identificare.

În rubrica „Istoric” vor apărea toate cărțile ce au fost împrumutate și returnate, ordonate descrescător după data de returnare (cele mai recent returnate cărți vor apărea primele).

În rubrica „Împrumuturi” apar cărțile împrumutate la acel moment, ordonate crescător după presupusa dată de returnare (cele în întârziere vor apărea primele).

Pentru fiecare împrumut există 3 acțiuni pe care bibliotecarul le poate efectua:



The screenshot shows the 'Informații client' page for Mary Monero. It displays her 'Nume complet' (Mary Monero), 'Adresă de email' (MaryCMoreno@teleworm.us), 'Număr de telefon' (0718602212), and 'ID' (0AnVyb). There are two buttons: 'Editează' (orange) and 'Șterge client' (red). Below these are sections for 'Împrumuturi' and 'Istoric', each with a right-pointing arrow. At the bottom, there is a green button labeled 'Emite legitimație'.

Figura 4: Informații client

1. Marchează cartea ca fiind returnată în momentul apăsării pe buton
2. Prelungește termenul cu 7 zile
3. Șterge împrumutul din baza de date (în cazul adăugării accidentale)

Titlu	Autor
Pride and Prejudice	Jane Austen
Împrumutat pe	Data retur
2024-05-21	2024-06-04
<b>Prelungește</b>	<b>Revocă</b>
<b>Returnează</b>	

Figura 5: Gestionare împrumut

### 3.2.2 Operațiile CRUD asupra cărților

În mod asemănător paginii clienților, coloana din stânga ne permite filtrarea și accesarea datelor cărților și avem butonul de adăugare care ne deschide un formular.

Adaugă carte

Poesii

Mihai Eminescu

Gallant

V. E. Schwab

Solenoid

Mircea Cartarescu

And Then There Were None

Agatha Christie

Pride and Prejudice

Jane Austen

Morometii

Marin Preda

Adaugă o nouă carte

Titlu

Tastează titlul cărții

Autor(i)

Tastează autorul(ii) cărții

ISBN

ISBN

Exemplare

1

Adaugă carte

Figura 6: Formular pentru adăugat cărți

Formularul se asigură că *ISBN-ul* are lungimea de 13 caractere și este format doar din cifre, iar la apăsarea butonului va fi verificat în baza de date ca acesta să fie unic.

Pe pagina fiecărei cărți putem accesa o multitudine de informații precum istoricul de împrumuturi și împrumuturile active, pe lângă datele adăugate prin formular.

Putem apăsa butonul „Editează” pentru a fi aduși pe o pagină cu un formular asemănător paginii de adăugare.

Butonul „Șterge carte” va șterge recursiv și toate împrumuturile și istoricul pe care le-a avut, pe lângă datele acesteia.

Informații carte

Titlu

Pride and Prejudice

Autor(i)

Jane Austen

ISBN

9781509896370

Exemplare

1

Editează

Șterge carte

► Împrumuturi

► Istoric

Emite cod de bare

Figura 7: Informații carte

Butonul „Emite cod de bare” generează o imagine care conține codul de bare aferent cărții, ce poate fi folosit pentru găsirea ei rapidă la căutare. De menționat este că nu toate cărțile au această facilitare, existând cărți care au fost publicate cu *ISBN-uri* invalide.

În rubrica „Istoric” vor apărea toți clienții ce au împrumutat și returnat cartea respectivă, ordonați descrescător după data de returnare (cei mai recenti clienți vor apărea primii).

În rubrica „Împrumuturi” apar clienții ce împrumută cartea la acel moment, ordonați crescător după presupusa dată de returnare (cei în întârziere vor apărea primii).

### 3.2.3 Modificarea setărilor de către un administrator

Există două tipuri de conturi pentru bibliotecari: de administrare și utilizatori normali.

Administratorii pot gestiona utilizatorii (prin adăugarea sau ștergerea lor) și pot modifica două setări generale ale aplicației: numele bibliotecii care va fi imprimat pe legitimații și camera folosită pentru scanarea codurilor de bare. Acest meniu este accesibil după logarea utilizatorului în cont din secțiunea „Login”.

Primul cont din baza de date are numele „admin” și rol de administrator și servește la efectuarea primelor setări, ulterior putând fi șters de un nou administrator adăugat.

Paginile setărilor pot fi accesate din panoul din dreapta, unde vom avea un formular de modificat setări sau de adăugat utilizatori, plus o listă a utilizatorilor și posibilitatea de a îi șterge.



Figura 8: Pagina de login a bibliotecarilor

### 3.2.4 Căutarea de cărți în baza de date OpenLibrary

Folosind o bază de date publică, numită *OpenLibrary*, bibliotecarul poate căuta cărți după *ISBN*, și această pagină va oferi informațiile relevante precum titlul și autorul și va căuta în baza de date a bibliotecii existența cărții căutate.



Figura 9: Bara de căutare după *ISBN*



După căutare va expune informațiile relevante imediat sub bara de căutare.



Figura 10: Carte găsită în baza de date publică

### 3.3 Funcționalități specifice

#### 3.3.1 Scanarea codurilor de bare

Cărțile publicate după anul 1970 au un *ISBN*<sup>[10]</sup> de 10 cifre, iar cele de după anul 2007 - de 13 cifre. *ISBN-urile* au fost concepute a putea genera coduri de bare *EAN-13*<sup>[11]</sup> valide, astfel că a fost necesară adăugarea unei funcționalități de scanare a acestora pentru filtrarea cărților. Mai mult, într-o manieră asemănătoare am optat și pentru crearea unor coduri de bare pentru conturile clienților, care urma să se dezvolte în legitimații ce conțin și alte informații despre client.

Scanarea se realizează din interfața *web*, cu ajutorul librăriei *ZXing*. Aceasta a fost scrisă în *Java* și ulterior portată spre multe alte limbaje de programare<sup>[12]</sup>.

#### 3.3.2 Gestionarea ID-urilor clienților

Având în vedere contextul în care era necesară generarea de coduri de bare pentru clienți, folosirea unui număr întreg drept ID nu era o idee bună din prisma lipsei de securitate: aceste ID-uri sunt predictibile și nu au lungime constantă.

O primă idee a fost folosirea de *UUID-uri*<sup>[13]</sup>, un standard pentru ID-uri, având o probabilitate de coliziune extrem de redusă și o lungime constantă. În schimb a apărut o nouă problemă: *UUID-urile* sunt reprezentate pe 128 de biți și generează coduri de bare mult prea dense și late. De asemenea, acest tip de ID utilizează literele mici din alfabetul latin și cifrele de la 0 la 9, impunând astfel necesitatea unui cod de bare care suportă cel puțin aceste caractere.

Prima iterație a aplicației folosește o metodă de compresie a *UUID-urilor*, folosind o librărie *web*<sup>[14]</sup>, care presupunea convertirea codului într-o bază destul de mare pentru a reduce semnificativ numărul de caractere. Baza aleasă folosea litere mici și mari, cifre și alte caractere din codul *ASCII*, însumând 90 de caractere și scurtând astfel lungimea la 22 de caractere. În urma analizei asupra diferitelor moduri de reprezentare sub forma unui cod de bare utilizabil, singurul care poate codifica tot alfabetul *ASCII* este *Code128*<sup>[15]</sup>.



be0a5c6a-8ff8-4b53-91aa-b15b3e8bbdc4

(a) Cod de bare din *UUID*



il!}%gSYp+9m0HSC@cLK

(b) Cod de bare din *UUID* compresat

Figura 11: Diferența între codurile de bare

Deși codurile de bare compresate erau acceptabile, am găsit o soluție cu rezultate mult mai bune: o librărie de generare de ID-uri folosind alfabetul furnizat, care asigură șanse de coliziune mici<sup>[16]</sup>. Alfabetul a fost redus la litere mici și mari și cifre, iar lungimea fixată la 6 caractere.



il!}%gSYp+9m0HSC@cLK

(a) Cod de bare din *UUID* compresat



97Pp3g

(b) Cod de bare din ID unic

Figura 12: Diferența între cele două abordări

### 3.3.3 Generarea de coduri de bare. Legitimații

Odată stabilit tipul de cod de bare, trebuiau generate din aplicație deoarece nu toate cărțile își imprimă codul de bare aferent *ISBN-ului*. Tot odată, trebuie ca legitimațiile să poată fi generate și să conțină codul de bare al ID-ului clientului, numele complet, data generării și denumirea bibliotecii. Cum trebuie salvate pe unitatea de stocare a calculatorului, a fost necesară utilizarea a două librării de procesare de imagini<sup>[17,18]</sup>, a uneia pentru generarea imaginilor ale codurilor de bare<sup>[19]</sup> și nu în ultimul rând, o librărie pentru utilizarea fonturilor<sup>[20]</sup>.



9780063230255

(a) Cod de bare *ISBN-13*



(b) Legitimație de client

Figura 13: Imaginile generate în *Rust*

## 4 Concluzii

Proiectarea acestei aplicații a presupus cercetări semnificative într-o multitudine de domenii precum design, modelarea și alegerea unei baze de date potrivite și protocoale standard pentru coduri de bare.

Am realizat că la orice pas m-am aflat a trebuit să fac un compromis între performanță, timp de implementare și cercetare și aspect, scopul fiind găsirea unui numitor comun între aceste trei fațete importante ale fiecărei funcționalități pe care aplicația o are în final. După această experiență, pot spune că pe departe cel mai complicat lucru a fost crearea de la 0 a unor coduri de bare unice care să fie pe atât de greu de reprodus pe cât de ușor de scanat în aplicație. Sunt convins că există soluții mai bune, dar sunt mulțumit de produsul final și de eleganța soluțiilor găsite.

În același timp, este important de menționat că aplicația nu este pe deplin finalizată și mai sunt funcționalități ce ar putea fi adăugate sau îmbunătățite, precum:

- Suport pentru *ISBN-10*.
- Suport pentru cărți apărute înainte de standardizarea *ISBN*
- Poze de copertă în baza de date
- Poze de profil pentru clienți
- Suport cloud pentru utilizarea aplicației de pe mai multe calculatoare
- Portal și conturi pentru clienți pentru operațiuni online

În acest sens, am pus la dispoziție codul sursă al aplicației pe *GitHub*, unde are deja 2 versiuni oficiale<sup>[21]</sup>.

# Bibliografie

- [1] Tauri. What is Tauri? URL <https://tauri.app/about/intro>.
- [2] Wikipedia. Rust (programming language), . URL [https://en.wikipedia.org/wiki/Rust\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Rust_(programming_language)).
- [3] Wikipedia. Garbage collection (computer science), . URL [https://en.wikipedia.org/wiki/Garbage\\_collection\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)).
- [4] Diesel. Diesel. URL <https://diesel.rs>.
- [5] Meta. React. URL <https://react.dev>.
- [6] Wikipedia. JSX (JavaScript), . URL [https://en.wikipedia.org/wiki/JSX\\_\(JavaScript\)](https://en.wikipedia.org/wiki/JSX_(JavaScript)).
- [7] Wikipedia. SQLite, . URL <https://en.wikipedia.org/wiki/SQLite>.
- [8] Wikipedia. Canva, . URL <https://en.wikipedia.org/wiki/Canva>.
- [9] Wikipedia. JetBrains, . URL <https://en.wikipedia.org/wiki/JetBrains>.
- [10] Wikipedia. ISBN, . URL <https://en.wikipedia.org/wiki/ISBN>.
- [11] Wikipedia. EAN-13, . URL [https://en.wikipedia.org/wiki/International\\_Article\\_Number](https://en.wikipedia.org/wiki/International_Article_Number).
- [12] GitHub. ZXing, . URL <https://github.com/zxing/zxing>.
- [13] K. Davis, B. Peabody, and P. Leach. Universally Unique IDentifiers (UUIDs). URL <https://datatracker.ietf.org/doc/html/rfc9562>.
- [14] NPM. short-uuid, . URL <https://www.npmjs.com/package/short-uuid>.
- [15] Wikipedia. Code 128, . URL [https://en.wikipedia.org/wiki/Code\\_128](https://en.wikipedia.org/wiki/Code_128).
- [16] NPM. Short Unique ID (UUID) Generating Library, . URL <https://www.npmjs.com/package/short-unique-id>.
- [17] crates. image, . URL <https://crates.io/crates/image>.
- [18] crates. imageproc, . URL <https://crates.io/crates/imageproc>.
- [19] crates. barcoders, . URL <https://crates.io/crates/barcoders>.
- [20] crates. ab\_glyph, . URL [https://crates.io/crates/ab\\_glyph](https://crates.io/crates/ab_glyph).
- [21] GitHub. LibraManager, . URL <https://github.com/AlexTutzZu/LibraManager>.