



Departamentul Automatică și Informatică Industrială
Facultatea Automatică și Calculatoare
Universitatea Națională de Știință și Tehnologie
POLITEHNICA București



LUCRARE DE DIPLOMĂ

Platformă colaborativă pentru gestionarea și recomandarea cărților

Coordonator

Prof. Dr. Ing. Olteanu Adriana

Absolvent

Ciuculan Ștefan

2025

Cuprins

1. Introducere.....	4
1.1. Justificarea alegerii temei.....	4
1.2. Prezentarea succintă a capitolelor lucrării.....	4
2. Domeniul din care face parte lucrarea.....	5
2.1. Platformele de împrumut de cărți electronice (eBook).....	5
2.2. Biblioteconomie.....	8
2.3. Aspecte legale.....	9
2.4. Aplicații populare în domeniu.....	10
2.5. Studiu despre utilizare și acceptanța aplicațiilor de citit cărți între studenții de la universitate.....	11
2.5.1. Introducere a studiului.....	11
2.5.2. Metoda de a investiga situația actuală a utilizării și acceptării aplicațiilor de cărți.....	11
2.5.3. Colectarea datelor.....	12
2.5.4. Informațiile colectate.....	12
3. Descrierea problemei abordate și a metodei de rezolvare propuse.....	17
3.1. Definirea problemei și soluția propusă.....	17
3.2. Conceptul aplicației.....	18
3.3. Structura bazei de date.....	19
3.4. Popularea bazei de date și etapa de preprocesare a datelor.....	21
3.5. Design-ul aplicației.....	22
4. Documentație tehnică.....	25
4.1. Tehnologiile utilizate.....	25
4.1.1. Inițializarea proiectului cu Vite.....	26
4.1.2. React.....	26
4.1.3. Flask.....	27
4.1.4 SQLite.....	27
4.2. Structura aplicației.....	27
4.2.1. Tabelele bazei de date.....	27
4.2.2. Modulul de inițializare.....	29

4.2.3. Stocarea datelor în tabela books.....	29
4.2.4. Rutele.....	29
4.2.5. Sistemul de recomandări.....	31
4.2.5.1. TF-IDF.....	31
4.2.5.2. Similaritate cosinus.....	32
4.2.5.3. Implementarea sistemului.....	32
4.2.6. Migrările din cadrul bazei de date.....	33
4.2.7. Paginile de autentificare și înregistrare.....	33
4.2.8. Pagina principală.....	34
4.2.9. Pagina de descriere a cărții.....	34
4.2.10. Explorarea lecturilor după genuri.....	35
4.2.11. Funcția de căutare.....	35
4.2.12. Biblioteca personală a utilizatorului.....	36
4.2.13. Gestionarea împrumuturilor.....	36
4.2.14. Mesageria.....	37
4.2.15. Structura generală a frontend-ului.....	37
4.3. Fluxul de funcționare al aplicației.....	37
4.4 Rezultate obținute.....	41
5. Concluzii și dezvoltări ulterioare	42
6. Bibliografie.....	43

1. Introducere

1.1. Justificarea alegerii temei

Astăzi ne confruntăm cu tehnologizarea extraordinar de rapidă a tuturor proceselor ce ne influențează viața de zi cu zi. Progresiv, mediile sociale, economice, politice globale sunt influențate și, totodată, schimbate definitiv de evoluția tehnologică. În consecință, mulți oameni au impresia că schimbarea nu va avea loc doar în bine.

De exemplu, social media joacă un rol important în cultura tinerilor de astăzi [1]. Cu toate acestea, are un impact negativ asupra lor în ceea ce privește sănătatea mintală, relaționarea în lumea reală și capacitatea de concentrare [2]. Adolescentii sunt mai nefericiti ca oricând. Trăim secolul însingurării [3], deși promisiunea cu care rețelele sociale au intrat în viețile noastre a fost aceea că ne vor aduce mai aproape unii de ceilalți. Tehnologia ne poate îmbunătăți viețile, însă numai dacă știm să controlăm această unealtă care a devenit atât de puternică. Ea este pusă acum în mâinile oamenilor de toate vârstele și din toate categoriile sociale și aproape nimeni nu este instruit cum să o folosească.

Pe lângă aceasta, o îngrijorare mare o reprezintă înlocuirea forței de muncă umane prin automatizarea diverselor procese tehnologice și evoluția inteligenței artificiale [4]. Îngrijorarea este una reală și fondată, dar adevărul este că oamenii nu vor ieși niciodată cu totul din afaceri. Dacă stăm să ne gândim la 2 resurse antice, de bază în societatea noastră, care nu vor fi niciodată demodate, acestea sunt oamenii și hârtia. Oricât s-ar încerca înlocuirea lor, ele nu vor putea dispărea din viețile noastre. Conform Statista [5], din 2010 producția globală de hârtie a rămas relativ stabilă, în jurul cantității de 400 de milioane de tone metrice de hârtie pe an. În anul 2023 fiind produse 401 milioane tone metrice de hârtie. Așadar, nu ne aşteptăm prea curând ca aceste două resurse prea importante să fie scoase din uz.

Cărțile vor rămâne deci, "la modă", ca formă de informare și evadare din cotidian [6]. În ciuda faptului că majoritatea preferă astăzi să se informeze din surse online, unde informația este mult mai accesibilă, cărțile rămân la mare căutare. Chiar și în privința divertismentului, în defavoarea filmelor și a serialelor, odată cu pandemia [6], oamenii au reînceput să aprecieze din nou valoarea unei cărți bune.

Având toate acestea în minte, ideea lucrării mele de licență vine din dorința viscerală de a promova relațiile reale dintre oameni și cititul ca formă de terapie și culturalizare. Pe ambele încerc să le aduc în actualitatea lumii noastre atât de influențată de mișcarea transumanistă. Astfel, consider că o aplicație online care încurajează lectura, cât și activitățile sociale este necesară în societatea noastră.

1.2. Prezentarea succintă a capitolelor lucrării

După primul capitol în care este prezentată o foarte scurtă introducere și motivul alegerii acestei teme de proiect, în al doilea capitol al lucrării se puntează în mod destul de amplu domeniul și multe aspecte constitutive ale acestuia. Cu ajutorul a două articole de specialitate, prezint o scurta istorie a eBook-urilor și a împrumutului de cărți, precum și cum au evoluat, percepția, opțiunile utilizatorilor și atât software-ul cât și hardware-ul în această arie de dezvoltare. În același timp, am găsit interesantă teoria biblioteconomiei, de la originea conceptului până la rezultatele în descoperirea intereselor utilizatorilor în ceea ce

privește o bibliotecă online. Tot în acest capitol precizez și câteva aspecte legale necesare pentru o bună desfășurare ulterioară pe piață a aplicației. Nu am omis nici studiul pieței în ceea ce privește cărțile de gestiune a cărților virtuale, dorind să observ ce există și ce se poate prezenta nou pentru a servi cititorilor, ce le lipsește lor și ce au deja. Ultima parte a capitolului a constat în prezentarea unui studiu desfășurat în mediul online pe un grup de studenți de la o universitate din Beijing în 2015. Datele rezultate în urma întrebărilor și punctelor atinse sunt foarte grăitoare, atât pentru un mediu societal cum este cel din China, cât și pentru alte țări din afara continentului asiatic.

Cel de-al treilea capitol al documentului conține idei despre problemele pe care le-am identificat în cea mai populară aplicație din domeniu și propunerea de schimbare pe care aplicația mea o aduce. Totodată, punctez conceptul aplicației, de unde a pornit ideea de început, cum am formulat și am conturat totul și ce surse de inspirație am avut. Structura bazei de date, faza de preprocesare a datelor cât și ideea inițială a design-ului platformei după care m-am ghidat de-a lungul dezvoltării aplicației sunt descrise pe larg în acest capitol.

Documentația tehnică, incluzând atât tehnologiile utilizate cât și funcționalitățile implementate și algoritmii au fost punctate în **capitolul al patrulea**. Separat, backend-ul și frontend-ul au fost descrise pentru a avea o vedere amplă asupra metodologiei de lucru. Am detaliat în cadrul acestei secțiuni aspecte despre fiecare pagină web în parte, navigarea, interacțiunea și experiența utilizatorului pe care mi-am dorit a o obține. Se prezintă cu ajutorul unor diagrame de activitate UML fluxul logic al fiecărei pagini în această secțiune. Totodată, am considerat necesară prezentarea generală a tuturor realizărilor la care am ajuns prin intermediul acestui proiect.

În al cincilea capitol am venit cu anumite concluzii și note de încheiere. În plus, mi s-a părut că aplicația permite multe dezvoltări ulterioare pentru a face experiența utilizatorilor mai plăcută, așa că am introdus câteva idei ce merită luate în considerare pe viitor pentru un proiect mai amplu ce ar trebui adus pe piață.

2. Domeniul din care face parte lucrarea

2.1. Platformele de împrumut de cărți electronice (eBook)

Definiția unei platforme de împrumut de cărți electronice (eBook) este aceea a unei platforme care include eBook-uri de la mai mulți editori, pe care utilizatorii le pot parcurge și împrumuta. Deși aceste platforme oferă eBook-uri care permit accesul simultan pentru mai mulți utilizatori, în general, ele funcționează după modelul „un singur exemplar – un singur utilizator”.

Exemple de platforme de împrumut de cărți eBook sunt Axis 360, 3M Cloud Library, Freading, LexisNexis Digital Library, și Overdrive. Următoarele caracteristici sunt comune pentru majoritatea acestor platforme. În primul rând, utilizatorii au opțiunea de a citi eBook-uri pe diverse platforme, fie în cloud prin intermediul browserului preferat, pe un dispozitiv dedicat de tip e-reader (sau e-ink reader, cum ar fi Kindle, Nook, Sony Reader etc.), sau printr-o aplicație pe o tabletă. Fiecare platformă oferă o versiune offline a eBook-ului, pe care utilizatorii o pot descărca și accesa.

Pentru a preveni pirateria în domeniul cărților electronice și pentru a aplica regulile și limitele impuse de perioadele de împrumut, aceste eBook-uri sunt protejate prin un software de gestionare a drepturilor digitale (Digital Rights Management - DRM). DRM-ul pentru aceste eBook-uri este de obicei gestionat prin Adobe Digital Editions și necesită un ID Adobe creat de utilizator.

În acest model de tip „o carte, un utilizator”, bibliotecile pot stabili perioade de împrumut pentru eBook-uri, care în general durează între una și trei săptămâni. De asemenea, bibliotecile pot limita numărul de împrumuturi per utilizator și pot crea o listă de așteptare pentru ca utilizatorii să primească eBook-urile odată ce sunt returnate. Pentru a satisface cererea, este obișnuit ca bibliotecile să achiziționeze mai multe copii ale unui eBook pe aceste platforme.

Platformele de împrumut de eBook-uri sunt populare și ușor de utilizat. În primul rând, nu există taxe pentru întârziere, iar aplicațiile disponibile permit utilizatorilor să citească eBook-urile pe diverse dispozitive e-reader sau mobile. De asemenea, utilizatorii pot descărca eBook-uri de la bibliotecă oricând și de oriunde există acces la internet, cu condiția ca eBook-ul să nu fie deja împrumutat.

Platformele de împrumut de eBook-uri oferă, de asemenea, suport pentru anumite tehnologii-asistent de tip screen-reader, care permit utilizatorilor cu dificultăți de citire sau deficiențe de vedere să acceseze eBook-urile. De exemplu, anul trecut (2012), Axis 360 a fost premiat de Federația Națională a Nevăzătorilor pentru eforturile sale de a face eBook-urile mai accesibile, oferind posibilitatea de a fi citite cu voce tare prin aplicația Blio.

Comparând platformele de împrumut de eBook-uri cu cele de tip agregator, precum EBL, ebrary sau EBSCOhost, se pot observa similarități. Toate lucrează cu mai mulți editori, iar unii dintre aggregatori, ca ebrary, oferă chiar și un reader pentru dispozitive mobile, pe care utilizatorii îl pot accesa ca să vizualizeze conținut de tip eBook. În plus, diferențele dintre aggregatori și platformele de împrumut se estompează. De exemplu, ebrary începe să exploreze sisteme de distribuție bazate pe împrumuturi, unde utilizatorii bibliotecilor pot împrumuta un eBook. Până când utilizatorul returnează cartea electronică, aceasta devine indisponibilă pentru ceilalți utilizatori, precum în cazul platformelor de împrumut. Totodată, pe ambele platforme eBook-urile au DRM asociat.

O mare diferență între aggregatori și platformele de împrumut este conținutul de masă valabil în principal în cea de-a doua. Conținutul de la cele 6 mari publicații (Hachette, HarperCollins, Macmillan, Penguin, Random House și Simon & Schuster) este regăsit în principal pe platformele de împrumut și unii dintre editori dețin propriile lor platforme, sau chiar relații exclusive cu acestea. De exemplu, cărțile de la HarperCollins au o limită de 26 de împrumuturi pe an la fiecare dintre cărțile lor, ceea ce înseamnă că la finalul unui an platformele trebuie să cumpere alt eBook. „A cumpără” este totuși un termen inexact, căci bibliotecile nu cumpără per-se un eBook din acest model. În schimb, ele le închiriază. Editorii nu numai că determină termenii închirierii, dar și prețurile. De exemplu, la timpul la care a fost scris acest articol (2013), Random House le cerea un preț de trei ori mai mare decât cel al versiunii tipărite bibliotecilor care doreau licențierea unui eBook [7].

Este important să precizăm că lucrurile s-au schimbat parțial de acum 12 ani, când a fost publicat articolul lui John Novak. Astăzi, utilizatorii bibliotecilor online pot obține, descărca sau citi o carte digitală cu doar câțiva pași simpli. În primii ani ai împrumutului de

eBook-uri, procesul prin care utilizatorii trebuia sa treacă era cunoscut pentru complexitatea sa, implicând descărcarea unui software specializat și deschiderea mai multor conturi cu diferite parole.

Termenii editorilor pentru cărțile electronice încă specifică prevederile din politica de împrumut pentru biblioteci și durata licenței. Unele licențe încă nu permit decât un singur împrumut simultan pentru un titlu pentru fiecare exemplar subscris pentru o perioadă determinată. Alte licențe astăzi sunt mai permisive totuși, dar tot există un plafon pentru numărul total de împrumuturi permise.

Bibliotecile publice se confruntă cu provocări în împrumutul digital. Interesele de lectură ale comunităților lor nu pot fi satisfăcute numai prin titluri gratuite de pe domeniul public, ci necesită acces la titluri actuale care sunt protejate prin drepturi de autor și sunt aproape întotdeauna achiziționate prin intermediul editorilor comerciali. Împrumutul de cărți tipărite în bibliotecă se încadrează în doctrina primei vânzări din legea drepturilor de autor din SUA, permîțând utilizarea nerestricționată, inclusiv capacitatea de a vinde, a da sau a împrumuta.

Această doctrină nu se aplică copiilor electronice ale cărților, care implică un set complex de probleme de afaceri și componente tehnologice. Practica împrumuturilor digitale controlate își propune să confere practici pentru materiale tipărite cărților digitalizate deținute de o bibliotecă, deși acest concept este contestat de editorii care au inițiat un proces împotriva Internet Archive în iunie 2020 [8].

În afara de împrumutul digital controlat, capacitatea bibliotecilor de a împrumuta cărți electronice comerciale depinde de licențele care specifică prețurile și condițiile de împrumut permise de editor. A existat o tensiune constantă între marile edituri și serviciile de împrumut ale bibliotecilor publice. În timp ce aproape toți editorii își oferă astăzi cărțile electronice bibliotecilor pentru împrumut, prețurile și condițiile continuă să prezinte provocări. Editorii se văd nevoiți să restrângă împrumutul pentru biblioteci pentru a-și proteja vânzările digitale. În timp ce bibliotecile continuă să facă presiuni pentru condiții mai favorabile, participarea aproape universală a editorilor la serviciile de împrumut de cărți electronice pentru biblioteci reprezintă un progres important în raport cu primii ani ai acestor servicii [8].

În general, editorii comerciali nu oferă bibliotecilor opțiunea de a cumpăra cărți electronice în mod permanent. Douglas County Libraries din Colorado a condus o inițiativă, începând din 2012, prin care bibliotecile au achiziționat cărți electronice de la editori și au oferit acces controlat prin propriile platforme digitale, folosind Adobe Content Server pentru gestionarea drepturilor digitale. În timp ce unii editori mai mici au susținut inițial acest model, acesta nu a reușit să câștige acceptarea în rândul industriei editoriale mai largi, iar inițiativa a dispărut prin 2014. Licențele bazate pe tranzacții și pe termen limitat au devenit modelul de afaceri stabilit pentru împrumutul bibliotecilor [8].

În ciuda acestor provocări, aproape toate bibliotecile publice din Statele Unite oferă un anumit nivel de împrumut de cărți electronice. Bibliotecile se abonează la servicii de împrumut de cărți electronice direct sau prin consorții. Ofertele bibliotecilor individuale pot fi completate de programe regionale sau la nivel de stat [8].

Bibliotecile publice fac investiții substanțiale în cărți electronice și conținut digital divers. Conform statisticilor agregate raportate către IMLS, bibliotecile publice din Statele

Unite au cheltuit 449.240.581 USD pe conținut electronic și 761.564.869 USD pe conținut tipărit în 2019. Cheltuielile cu resursele digitale au crescut constant în ultimul deceniu, în timp ce cheltuielile cu imprimarea s-au stabilizat (Fig. 1) [8].

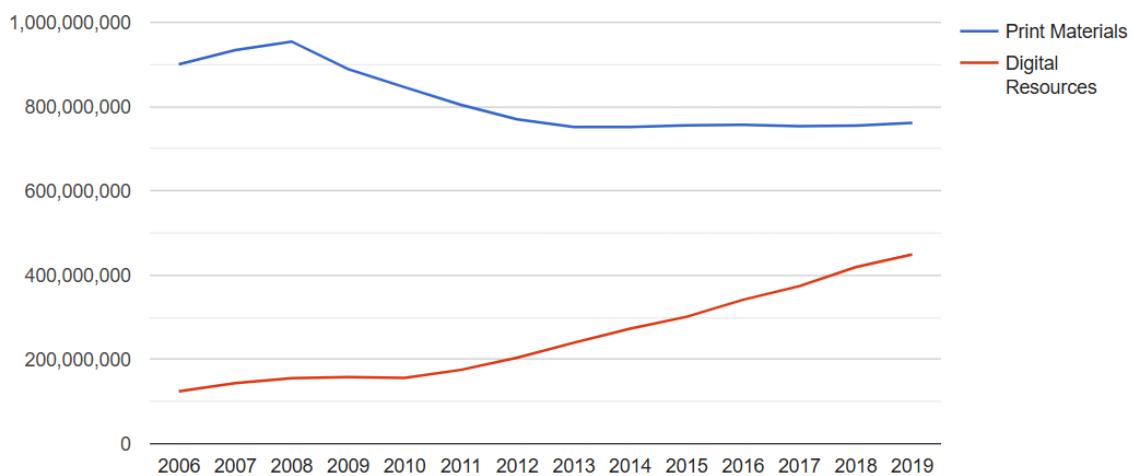


Fig. 1. Cheltuielile tuturor bibliotecilor publice din SUA (2006-2019)

2.2. Biblioteconomia

Biblioteconomia și știința informației este, precum implică numele, o combinație a două domenii: știința bibliotecilor și știința informației. Examineză atât aspectele practice cât și cele teoretice ale managementului sistemelor de informații, bibliotecilor și comunicării scolare. Include subdomenii precum organizarea cunoștințelor, bibliometrie, recuperarea informațiilor, documentarea și studiile pe utilizatori și interacționează cu discipline precum știința computerelor (Computer Science), epistemologie socială, filosofie și sociologia științelor [9].

O definiție pentru știința bibliotecilor ar fi cea oferită de Joan Reitz în dicționarul său online: cunoștințele profesionale și abilitățile cu care informațiile înregistrate sunt selectate, dobândite, organizate, stocate, menținute, recuperate și diseminate ca să împlinească nevoile unei clientele specifice, de obicei învățăte într-o școală profesională de biblioteci calificată pentru a garanta diploma de postliceală pentru M.L.S sau M.L.I.S.

Tehnica bibliotecilor este un termen comun pentru metodele folosite în organizarea, catalogarea, uzul și administrarea bibliotecilor. În particular, în Germania, termenul "știința bibliotecilor" este deseori utilizat atât pentru tehnica bibliotecilor, cât și pentru alte discipline (istoria cărților, bibliografie, istoria bibliotecilor și statistică bibliotecilor); totuși acest termen nu este corect ales, deoarece managementul bibliotecilor nu este o știință, chiar dacă cunoștințe științifice și calificări sunt necesare [9].

În ceea ce privește cititorii, aceștia sunt obiectele de serviciu ale bibliotecilor și obiectele de informare ale sistemului de management al informației. În ultima perioadă de explozie la nivel informațional, bibliotecile inteligente ajută cititorii să găsească informația de care chiar au nevoie din masa de informație existentă și le oferă servicii personalizate și eficiente. Cititorii au următoarele cerințe când vine vorba de sistemele bibliotecilor inteligente:

1) Căutarea și împrumutul de documente: fie că vorbim de biblioteci tradiționale sau inteligente, nevoile de bază ale cititorilor sunt împrumutul și căutarea documentelor. Cu toate că ne aflăm într-o perioadă a modernizării continue, bibliotecile inteligente ar trebui să se bazeze pe fundamentele bibliotecilor tradiționale și să ofere cititorilor servicii inteligente de căutare și împrumut de documente [10].

2) Setări de vizualizare și informații personalizate: majoritatea bibliotecilor de obicei folosesc opțiunea “My Library” pentru a le permite cititorilor să-și gestioneze singuri informațiile personale, să vizualizeze istoricul de căutări și de împrumuturi, dar lipsește componenta de setare a informațiilor oferită de noile tehnologii, precum terminalele mobile sau bibliotecile inteligente [10].

3) Cererea de învățare independentă: astăzi, funcționalitățile de învățare online, precum bibliotecile mobile și cursurile MOOC au devenit obișnuite în acest context, iar conștientizarea oamenilor cu privire la aspectul învățării autonomă s-a îmbunătățit. Sistemul de management trebuie să gestioneze și să clasifice conținutul de învățare online pentru ca acesta să fie fructificat la maximum [10].

4) Recomandările personalizate: Serviciile personalizate și mai bine direcționate reprezintă direcția principală a dezvoltării bibliotecilor pe viitor, iar personalizarea este partea centrală a nevoilor clientilor. Principalul motiv al introducerii tehnologiei inteligente în biblioteci este de a oferi servicii mai personalizate. Odată cu creșterea nivelului de educație literară a oamenilor, cerințele pentru servicii personalizate devin tot mai ridicate. Principala formă prin care biblioteca utilizează tehnologia pentru a oferi servicii potrivite contextual pentru fiecare cititor este recomandarea personalizată, ceea ce face ca personalizarea să ocupe o poziție tot mai importantă în nevoile acestora [10].

5) Schimbul de informație și nevoile de partajare: în 2019, doar o mică parte din bibliotecile chineze au implementat forumuri (BBS) sau panouri de mesaje pentru ca cititorii să comunice între ei. De fapt, creând o atmosferă propice transmiterii de informație contribuie la sporirea entuziasmului cititorilor pentru a se folosi de biblioteci în învățare și citit [10].

6) Colectarea opinii și a feedback-ului: bibliotecile ar trebui în primul rând să inițieze identificarea nevoilor cititorilor. Apoi, cititorii pot ei însăși să-și ofere sprijinul împărtășind opinii atât în mediul online, cât și offline, făcându-și cunoscute nevoile în calitate de consumatori [10].

2.3. Aspecte legale

În domeniul drepturilor de autor, bibliotecile digitale ridică o serie de probleme juridice din cauza împrumutului de cărți fizice între utilizatori. Aceste platforme, precum și aplicația mea, permit posesorilor de cărți tipărite, achiziționate legal, să le împrumute temporar altor utilizatori. Deși la prima vedere nu pare că împrumutul unor copii fizice ar trebui să încalce dreptul de autor, trebuie să analizăm în ce măsură astfel de practici pot fi considerate legale.

Dreptul de autor protejează reproducerea și comunicarea publică a operei, dar nu și simplul transfer fizic al unui exemplar deja tipărit. “Epuizarea dreptului de distribuire” sau “principiul primei vânzări” este un concept legal potrivit căruia, ca urmare a vânzării unui exemplar al operei cu acordul titularului, acesta își pierde dreptul de a controla exemplarul

[11]. Astfel, cumpărătorul unui exemplar fizic are dreptul de a revinde sau a se debarasa în orice mod, fără a necesita permisiunea autorului.

Cu toate acestea, există și un drept exclusiv de împrumut al autorului separat de dreptul de distribuire. La nivel de Uniune Europeană, dreptul de închiriere și împrumut prevede că autorii au posibilitatea de a autoriza sau interzice împrumutul originalelor și copiilor lor [12]. Acest drept de împrumut este definit ca “punerea la dispoziție pentru utilizare, pe o perioadă limitată, fără avantaj economic sau comercial direct sau indirect, atunci când este realizat de instituții accesibile publicului” [12]. Atât legislația din România cât și cea de la nivel european fac distincție între împrumut și închiriere [13]. Împrumutul este un termen folosit pentru a defini utilizarea temporară a operei fără a beneficia în vreun fel economic de pe urma acesteia, iar de obicei este realizat prin intermediul unei instituții și nu necesită autorizarea autorului, însă poate fi nevoie de a se oferi o remunerație autorului [14]. Pe când închirierea este opusul, aceasta presupunând un avantaj economic sau comercial, necesitând acordul autorului, iar acesta are dreptul la o remunerație oferită prin intermediul unui contract. Astfel că, dacă utilizatorii își împrumută exemplare în mod gratuit între ei, iar nimeni nu obține un beneficiu financiar de pe urma acestui schimb, activitatea se poate identifica cu procesul legal de împrumut.

Cadrul legal prezent permite împrumutul cărților fizice, dar impune condiții strice dacă acesta capătă caracter public sau comercial. O platformă ca cea pe care am conceptualizat-o, ce facilitează împrumutul între persoane fizice, se situează “într-o zonă gri”, deoarece nu vorbim nici de o biblioteca publică instituționalizată, dar nici de un act prietenesc, ci ceva intermediar. Totuși, acest împrumut este legal atât timp cât nu are scop comercial și nu implică realizarea de copii.

2.4. Aplicații populare în domeniu

1. Fable

Fable a devenit, fără niciun dubiu, una dintre cele mai promițătoare aplicații din lumea literară. Dincolo de urmărire și evaluarea cărților pe care le citești, Fable permite utilizatorului să socializeze cu alți cititori în cluburi de carte și forumuri de discuții. În plus, dacă îți place televizorul la fel de mult precum cărțile, poți adăuga și conținut din ceea ce urmărești și vizionezi. Fortune afirmă că Fable este cu adevărat “o aplicație de socializare mai puțin toxică” [15].

2. StoryGraph

StoryGraph folosește învățarea automată și inteligența artificială ca să analizeze cărțile și să ofere cele mai bune recomandări. Poți căuta cărți în funcție de stare, gen și altele ca să fii sigur că iezi cea mai bună decizie. Ca și alte aplicații populare, StoryGraph oferă posibilitatea să vezi impresiile prietenilor tăi în legătură cu ce lecturile lor curente [15].

3. Bookly

Pentru persoanele neinteresate de partea socială a multor aplicații, Bookly este o alternativă bună. Această aplicație subapreciată se străduiește să vă mențină aventurile de lectură organizate într-un singur loc compact. Puteți să vă revizuiți statisticile din ultimii câțiva ani, să vă stabiliți obiective de lectură și să vă urmăriți cărțile într-un jurnal. Pentru o motivație suplimentară, Bookly oferă repere de fiecare dată când citiți [15].

4. LibraryThing

Oricine este interesat de arta biblioteconomiei are nevoie de LibraryThing. Această aplicație este una dintre singurele care oferă date de catalogare aprofundate despre cărțile dvs., inclusiv sistemul Dewey Decimal, formate media și multe altele. Este o platformă grozav de folosit împreună cu o altă aplicație—LibraryThing ajută la urmărirea a ceea ce deține utilizatorul, în timp ce o altă aplicație îl ajută în organizarea gândurilor în legătură cu ce vrea să citească [15].

5. Goodreads

Goodreads a fost (și continuă să fie) aplicația de căptâi pentru iubitorii de cărți timp de decenii, și pentru un motiv întemeiat. Acest site are una dintre cele mai plăcute interfețe — ușor de navigat și placută la ochi. Cu vasta comunitate pe care aplicația o are, ești împins să întâlnești oameni noi și să te conectezi cu cei dragi. Aplicația are și cel mai bun sistem de recomandări și oferă știri despre cele mai noi cărți [15].

2.5. Studiu despre utilizare și acceptanța aplicațiilor de citit cărți între studenții de la universitate

2.5.1. Introducere a studiului

Chiar dacă tema nu este exact aceeași pe care proiectul meu o abordează, am considerat rezultatele relevante pentru domeniul din care face parte lucrarea. Astfel că, acest studiu reprezintă un chestionar online și investighează cum citesc studenții de la Universitatea Peking din Beijing, China eBook-uri pe aplicații, precum și ce cred despre aceste aplicații. Rezultatele arată că studenții de la facultate citesc cărți cu aplicații dintr-o gamă largă, frecvent, dar fragmentat, ușor și pe îndelete.

Odată cu dezvoltarea internetului și a internetului wireless, oamenii au început să petreacă din ce în ce mai mult timp citind pe dispozitive mobile, cum ar fi smart phone-uri, tablete, PC-uri etc. Cititul pe dispozitive mobile devine o parte importantă a lecturii de zi cu zi. Anumite fenomene care au loc în viețile noastre de zi cu zi ar putea să amplifice aceste rezultate. În trecut, oamenii pierdeau timpul citind ziarul sau cărți în metrou, fapt care nu se mai observă acum. În momentul acesta, oamenii înlocuiesc ziarul cu telefonul. Pentru majoritatea oamenilor, lectura pe telefon este o modalitate de a-și omori timpul, citind știrile sau literatură de pe internet.

În 2012, website-ul 1448.biz, urmărind aplicațiile care sunt descărcate de pe App Store, a anunțat că pe App Store se aflau peste 6000 de aplicații de citit cărți, care ocupau aproximativ 11% din tot cumulul de aplicații. Având în considerare interfața prietenoasă și accesibilitatea de care dădeau dovadă, aplicațiile de acest tip au devenit din ce în ce mai populare.

2.5.2. Metoda de a investiga situația actuală a utilizării și acceptării aplicațiilor de cărți

Pentru a investiga opiniile studenților despre aplicațiile de cărți și cum le folosesc pe acestea, a fost folosit un chestionar online, alcătuit din două părți. Prima parte, aflată la începutul chestionarului, constă în colectarea informațiilor individuale, precum genul și fondul educațional. A doua parte include 15 întrebări. Aceste întrebări se referă la conținutul cărților citite, strategiile de lectură, timpul de lectură etc. În plus, avantajele și dezavantajele aplicațiilor de citit sunt investigate pentru a se compara lectura pe hârtie cu cea de pe ecrane.

2.5.3. Colectarea datelor

Chestionarul a fost distribuit în mod aleatoriu pe internet printre studenții din universitate. În final, 426 de formulare au fost reținute, în timp ce 418 dintre ele au fost considerate valide, ceea ce reprezintă un procentaj de 98.12% din total.

2.5.4. Informațiile colectate

Precum se poate observa în Fig. 2, printre studenții care au răspuns s-au aflat 245 de bărbați, ceea ce reprezintă 58.61% din totalul participanților la studiu. Fapt care face ca restul de 41.39% dintre persoane să fie femei. Astfel că, deși au participat puțin mai mulți bărbați la studiu, vorbim totuși de un oarecare echilibru de gen [16].

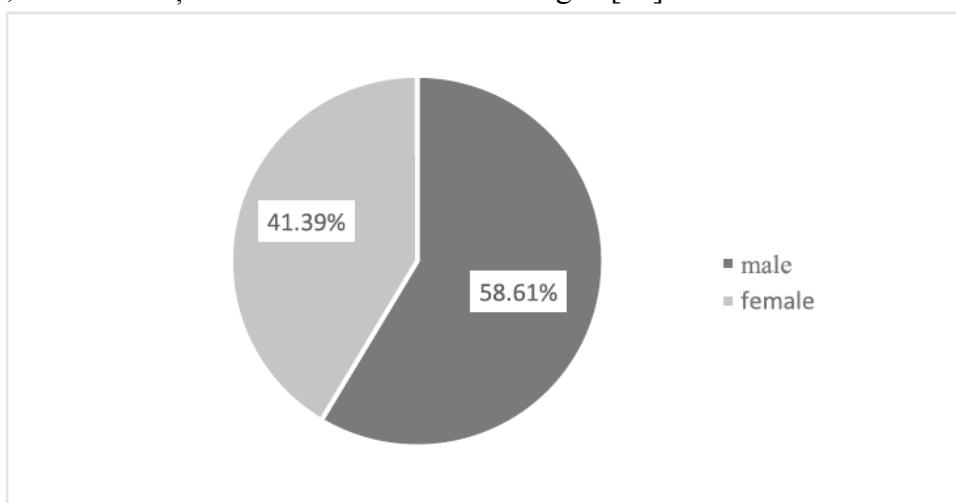


Fig. 2. Raportul de gen

Având loc în cadrul unei universități, studiul prezintă rezultate diverse în ceea ce privește nivelul educațional al persoanelor participante, incluzând atât studenți de licență (38.52%), cât și masteranzi (39.71%) și doctoranzi (21.77%) (Fig. 3). Numărul doctoranzilor este mai mic, fapt normal, căci este obișnuit ca în toate universitățile doctoranzii să fie mai puțini.

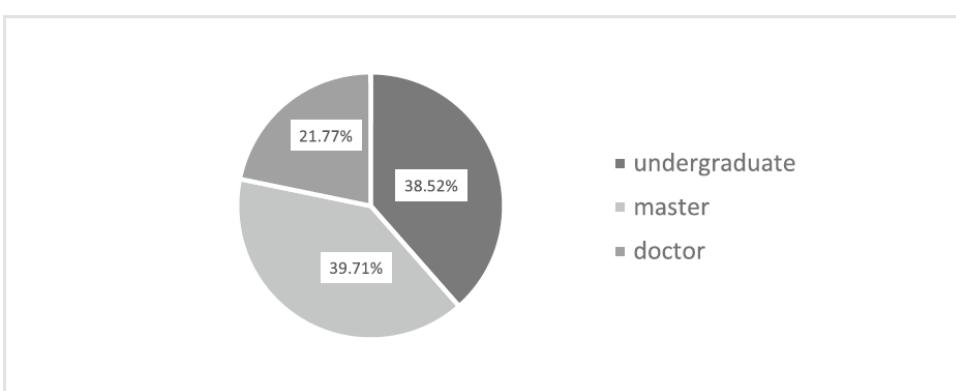


Fig. 3. Nivelul educațional al participanților la studiu

Când vorbim despre domeniile pe care studenții le lecturează, observăm o plajă largă și varietate semnificativă. Domeniile cărților citite pe aplicațiile de cărți de către participanții

la studiu sunt legate de literatură, limbă, umanism, management, economie, viață și modă, biografii și aşa mai departe (Fig. 4) [16].

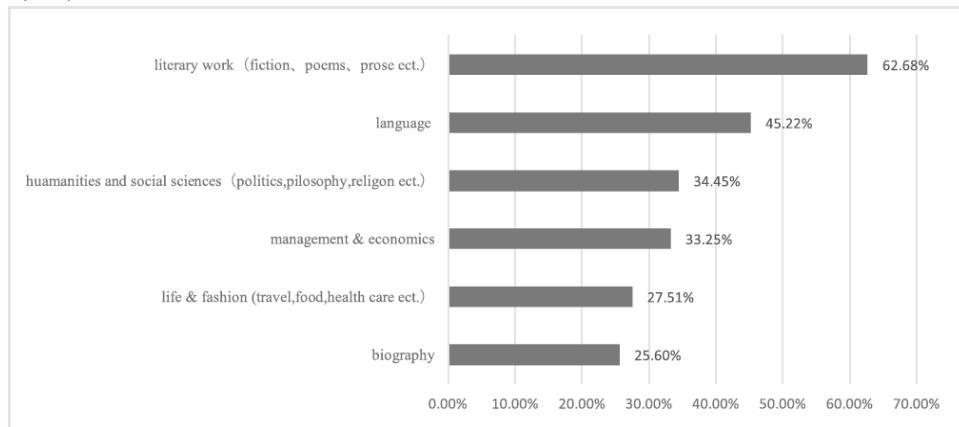


Fig. 4. Conținutul lecturii

Din Fig. 5, concluzionăm că studenții de la universitate folosesc aplicațiile fragmentat și, în mare parte, pentru relaxare, în timpul liber. Observăm că cel mai des aceste aplicații sunt folosite de participanții la studiu înainte de a merge la culcare și la toaletă. Plus de aceasta, unii studenți folosesc aplicațiile când așteaptă la cozi sau se află între cursuri sau seminare. Deducem astfel, că aplicațiile de citit sunt utilizate fragmentat deoarece timpul în care sunt folosite este foarte fragmentat. Totodată, această concluzie este întărită de Fig. 6. Aceasta arată că numai 15.55% dintre studenți folosesc aplicațiile acestea mai mult de 2 ore pe zi, iar cei mai mulți dintre ei le utilizează mai puțin de 2 ore pe zi, chiar mai puțin de 30 de minute. Astfel că timpul petrecut în aplicațiile de citit este scurt [16].

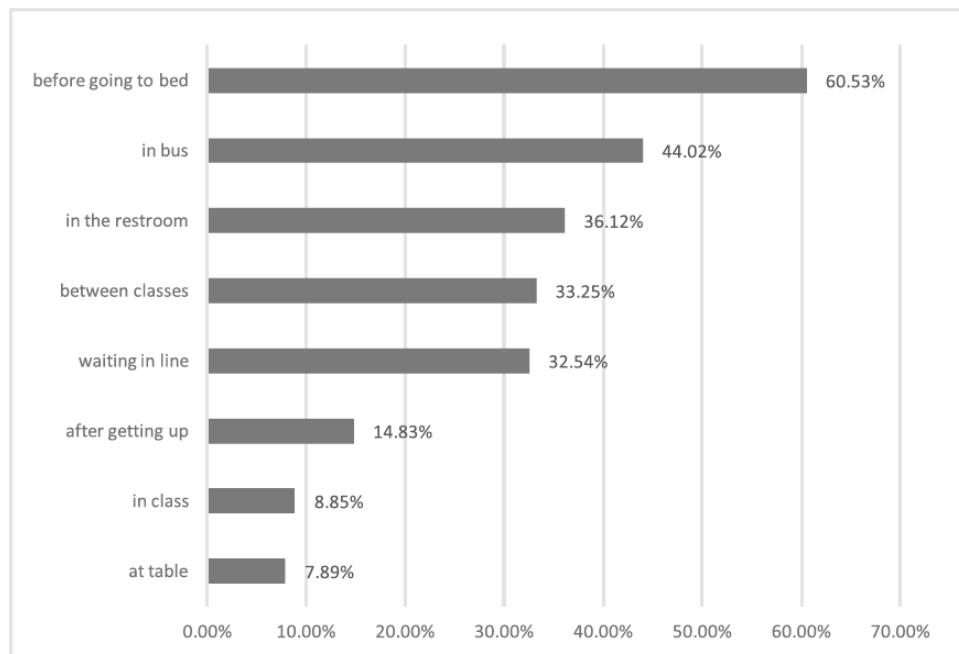


Fig. 5. Când și unde citesc studenții cărți

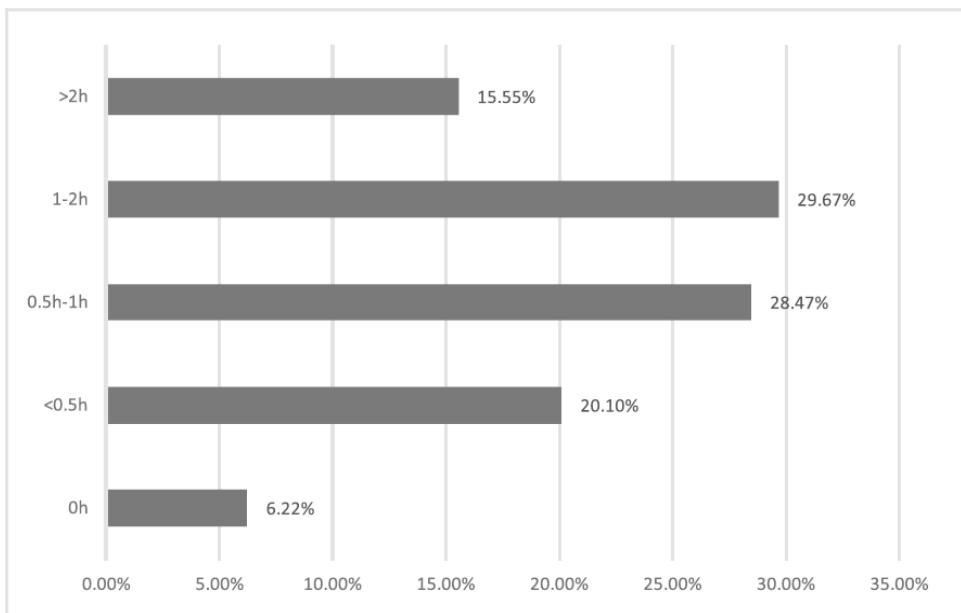


Fig. 6. Timpul petrecut în aplicațiile de citit cărți

Când folosesc aplicații pentru citit, utilizatorii obișnuiesc să citească repede. Foaia și ecranul reprezintă două moduri diferite de a transmite informație, fapt care rezultă în moduri diferite de a citi. Datele din Fig. 7 arată că aproape două treimi dintre participanții la studiu citesc la repezelă, în timp ce aproximativ o treime dintre ei citesc cu atenție cărțile de pe ecrane. Timpul alocat pentru utilizarea aplicațiilor acestora poate fi relevant pentru că, precum se observă în Fig. 5, respectiv Fig. 6, studenții citesc când sunt în autobuz, la toaletă sau la masă și petrec mai puțin de două ore citind. Lectura atentă necesită mai mult timp și un mediu specific, în timp ce o lectură superficială poate fi desfășurată în perioade de timp fragmentate [16].

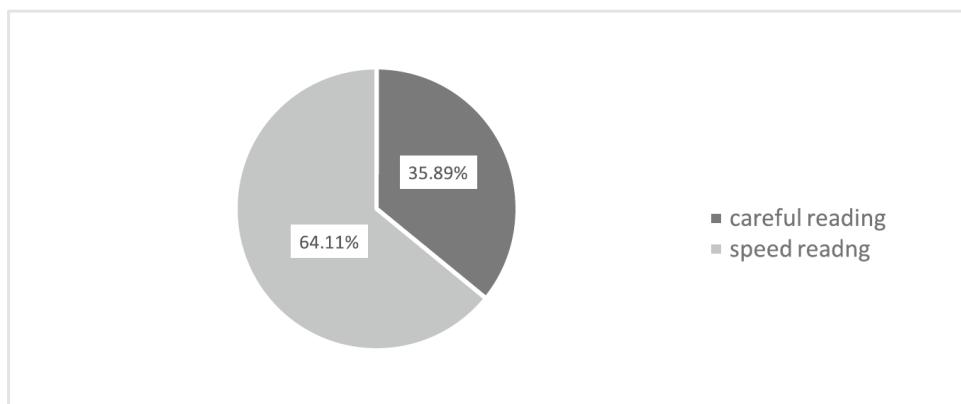


Fig. 7. Moduri de a citi pentru utilizatorii aplicațiilor mobile

Studenții aleg să folosească aplicații de cărți fiind influențați din mai multe domenii, dar mai ales primind recomandări de la prietenii. Din Fig. 8, aflăm că aproximativ jumătate dintre participanții la studiu au aflat de o aplicație de cărți de la un prieten, în timp ce 39.95% dintre ei au primit informația din clasamentele de aplicații de pe App Store sau Play Store. Alții au aflat despre o aplicație prin intermediul reclamelor sau anunțurilor încorporate.

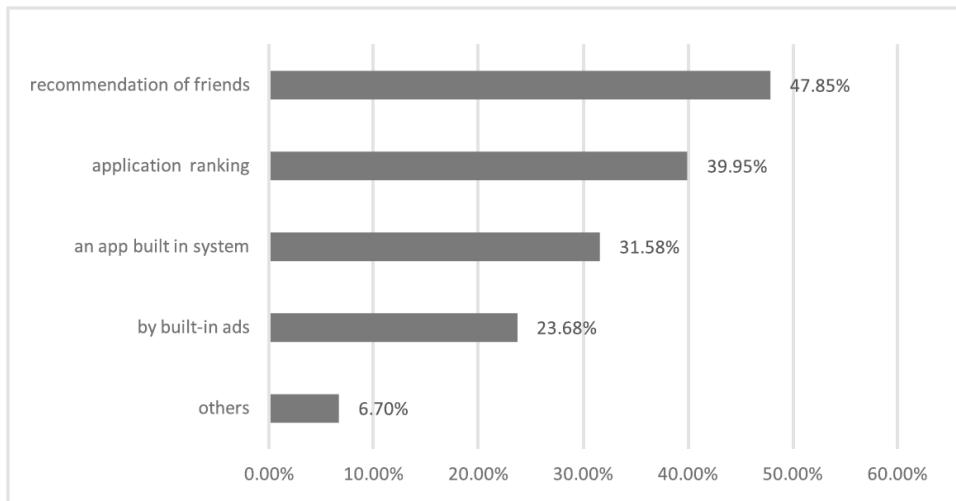


Fig. 8. Surse de informare cu privire la aplicații

Cei mai importanți factori ce-i influențează pe studenți să aleagă o anumită aplicație sunt gratuitatea și larga varietate de cărți ce sunt incluse în aceasta. Din Fig. 9, deducem că, pentru studenții care nu au o sursă de venit, gratuitatea aplicației, cât și varietatea cărților sunt factorii cei mai însemnați, urmați desigur de buna organizare a interfeței și comoditatea cu care se adaptează la operațiile din aplicație. Cât despre buna funcționare a algoritmului de recomandare a cărților bazat pe preferințele din aplicație, participanții la studiu nu o consideră foarte importantă. Este deci destul de clar că studenții nu au cerințe avansate în ceea ce privește funcționarea software-ului aplicațiilor [16].

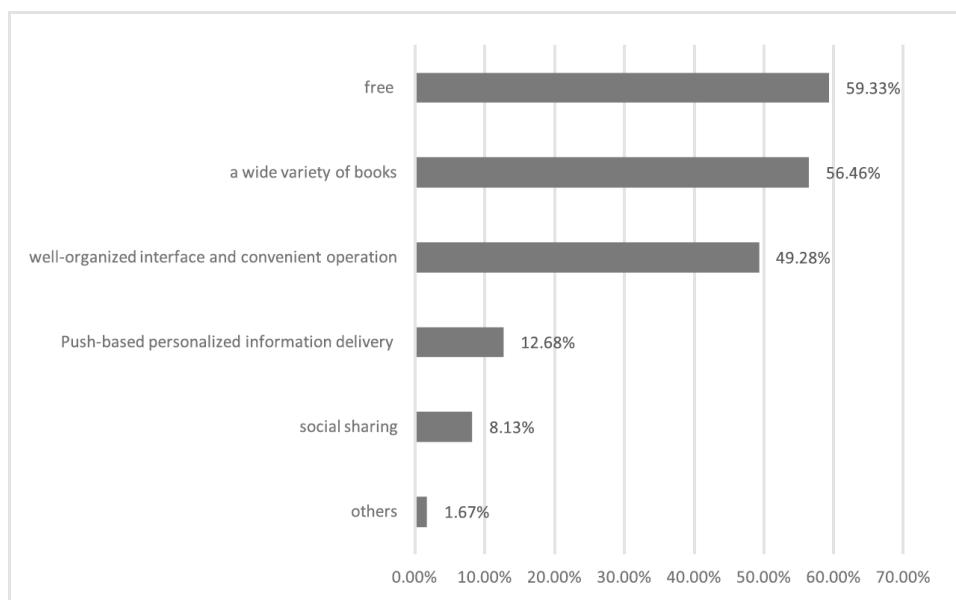


Fig. 9. Factori ce influențează alegerea unei aplicații de către studenți

Ca să revenim și la informațiile expuse anterior în secțiunea 2.1., studiul concluzionează, printre altele, căci, cu toate că aplicațiile de cărți au multiple avantaje în defavoarea cărților printate, lectura pe hârtie rămâne favorita studenților. Astfel, sunt puse

față în față avantajele și dezavantajele cărților digitale, respectiv ale celor printate. De exemplu, cărțile online sunt mai ieftine, iar uneori chiar sunt gratis. Aplicațiile de cărți permit o accesibilitate rapidă, deoarece nu este necesară deplasarea la o bibliotecă pentru a obține o carte. În plus, cărțile digitale pot fi citite oriunde și oricând, iar acestea sunt mult mai portabile decât cele printate (Fig. 10).

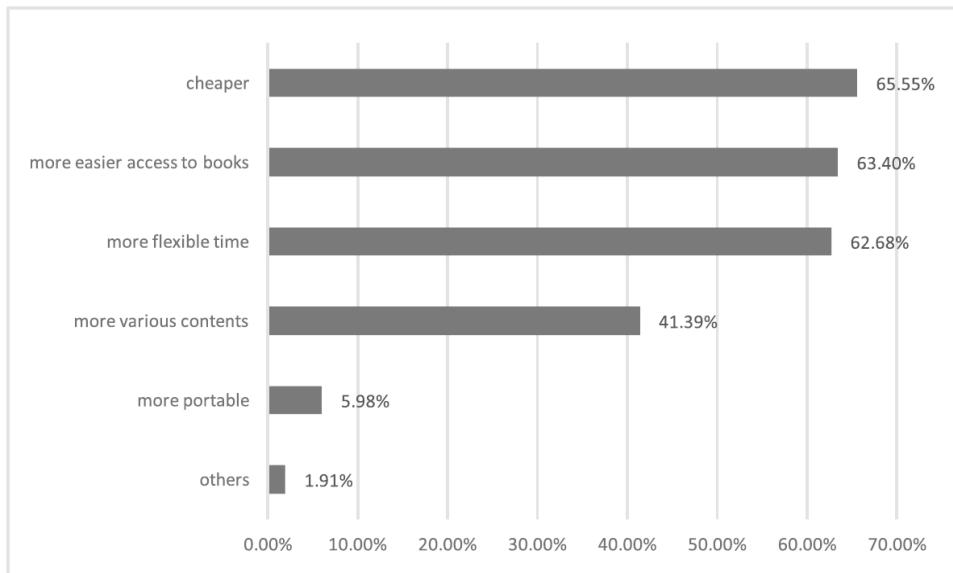


Fig. 10. Avantajele cărților digitale în față celor printate

Totuși există și dezavantaje în cazul aplicațiilor de cărți, precum sunt semnalate de datele colectate în cadrul studiului. 80% dintre studenți reclamă faptul că lectura de pe dispozitive digitale provoacă dureri de ochi mai des decât în cazul cărților printate. În timp ce câțiva dintre participanții la studiu spun că rețin mult mai puțină informație citind de pe aplicații de cărți, puțini dintre aceștia mai relatează o răbdare scăzută când citesc cărți digitale (Fig. 11) [16].

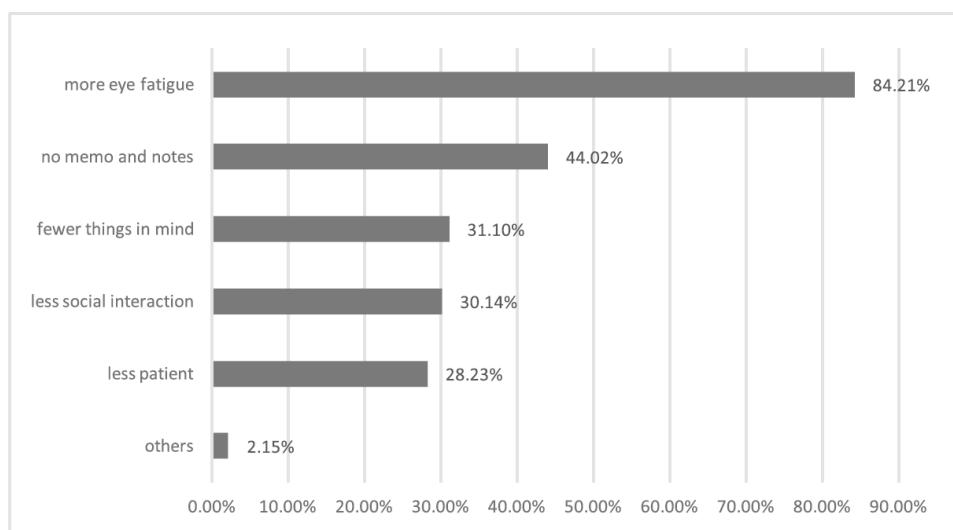


Fig. 11. Dezavantajele cărților digitale

Din Fig. 12 observăm ce preferă studenții dintre cele două variante: cărți digitale sau cărți printate. În cazul în care studenții beneficiază atât de o aplicație de cărți, cât și de cărți printate, 63.63% dintre ei le preferă pe cele printate, iar aproximativ o treime le preferă pe cele online. Probabil preferința aceasta provine și din faptul că primul contact pe care participanții la studiu l-au avut cu lectura a fost pe cărți printate, iar un obicei deja format este greu de schimbat. Probabil rămâne de studiat în viitor ce preferințe vor avea cei născuți după 2010 [16].

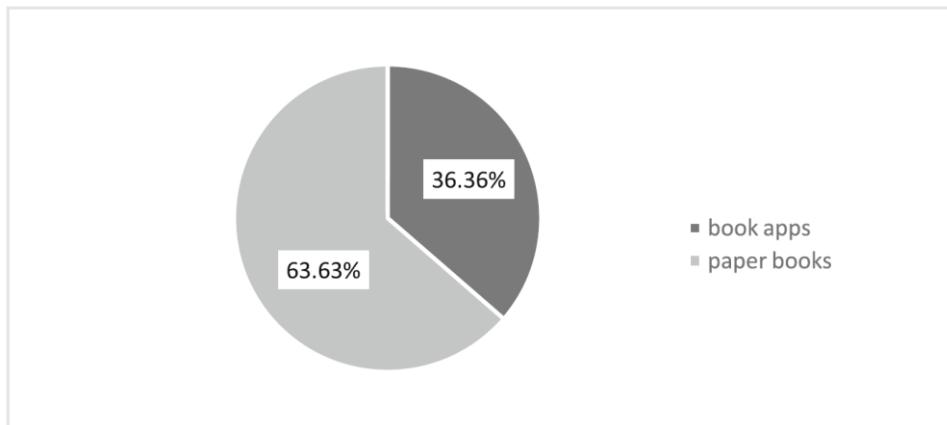


Fig. 12. Alegerea dintre cărțile printate și cele digitale

3. Descrierea problemei abordate și a metodei de rezolvare propuse

3.1. Definirea problemei și soluția propusă

În lumea cititorilor, a “șoareciilor de bibliotecă” și a celor cărora cărțile încă le sunt cea mai plăcută sursă de informare, de distracție și culturalizare mi s-a părut că lipsește un element important, în care ingineria și noile tehnologii pot avea un cuvânt de spus pentru a umple acest gol. Cititorii obișnuiți, cei de modă veche, preferă să nu se complice cu aplicații menite să-i ajute în a-și organiza și a avea o evidență a cărților citite în mediul virtual. Totuși asta poate să se schimbe dacă li se oferă un mediu plăcut, ușor de utilizat și performant. Există astfel de aplicații, însă, din dorința dezvoltatorilor de a aduce cât mai multe îmbunătățiri programelor lor, aplicațiile ajung să fie obosităre pentru un cititor ce dorește lucruri mici. Dezvoltatorii își doresc, înainte de toate, ca aplicația lor să fie distractivă, să atragă și să țină utilizatorul cât mai mult timp în aplicație. Or, asta pentru un cititor înrăit nu este plăcut. „Șoareci de bibliotecă” nu vor medalii virtuale pentru cât au citit, nu vor concursuri de citit și cu atât mai mult nu doresc să-și piardă timpul în care ar fi putut lectura o carte bună pe o aplicație online.

Soluția la această lipsă, la această problemă, este o aplicație care să nu aibă un interes de la cititor. Vorbim de o aplicație cu design simplu, cu funcționalități care să reprezinte strictul necesar și să nu distra de la adevăratul scop: lectura.

În plus, interacțiunea umană, cum am prezentat și în secțiunile anterioare, este redusă în ultimul timp. În special pentru oamenii avizi de lectură, componenta socială poate reprezenta o mare lipsă. Soluțiile cu care diverse aplicații au venit sunt fie concentrate în zona online – și aici vorbim de forumuri, secțiuni de recenzii cu posibilitatea de replică la comentarii etc – fie fac aplicația, precum am precizat mai sus, să devină exhaustivă și neplăcută pentru un simplu cititor – secțiuni de evenimente în aplicație, cluburi de lectură etc. Cum am considerat aceste soluții ineficiente, am considerat că este necesară o oarecare inovație, ceva neobișnuit: împrumuturile de cărți fizice între utilizatori. S-a demonstrat, în urma diverselor studii și cercetări, că împrumutul, implicând un act de „posesiune împărtășită” (shared possession), ține ambele părți implicate deoarece bunul este în posesia destinatarului, iar realul deținător al bunului așteaptă să-l primească înapoi [17]. Acest fapt face ca cei doi indivizi să țină legătura pentru a-l asigura pe proprietarul de drept de siguranță bunului său. Astfel că, nu doar că se sudează relații noi între oameni până atunci necunoscuți, ci se realizează un schimb față în față, în afara locuinței, fapt care îl face pe cititor să iasă din confortul locuinței sale.

3.2. Conceptul aplicației

Am conceput aplicația având ca inspirație două aplicații deja existente pe piață: Goodreads și Stremio.

Goodreads este cea mai populară platformă socială pentru cititori, este deținută de Amazon și are pentru utilizatori o mulțime de funcționalități. Le permite acestora să creeze liste de cărți, să evalueze cărțile prin recenzii scrise și un sistem de rating de la 1-5, să primească recomandări personalizate, le pune la dispoziție forumuri și pot urmări activitatea prietenilor lor. În 2023, Goodreads avea în jur de 150 de milioane de utilizatori în toată lumea [18], un catalog extins, funcționalitate de căutare avansată, un concurs anual pentru încurajarea lecturii și recomandări bazate pe preferințele utilizatorului și ale comunității. Astăzi, deși aplicația este încă populară, există totuși critici la adresa Amazon și cum gestionează dezvoltarea aplicației. Interfața pare neschimbată din 2007, funcția de căutare nu ar funcționa foarte bine, nu poți replica oamenilor în comentarii și altele [19]. Pe lângă toate acestea, deși nu am folosit foarte mult aplicația, mi s-a părut obosită prin multiplele opțiuni și meniu existente în aplicație. Eu ca mulți alții, mi-aș fi dorit o aplicație simplă de gestiune a bibliotecii virtuale. Totuși, Goodreads m-a inspirat în design-ul aplicației și în ideea de bază a acesteia.

Cât despre **Stremio**, este o aplicație lansată în 2015, ce permite utilizatorilor să vizualizeze seriale și filme online din diverse surse printr-un sistem de plugin-uri. Stremio m-a inspirat cel mai mult pe partea de UI/UX. Este o aplicație ușor de folosit odată configurață, foarte intuitivă și gratis. Și acesta include un sistem de recomandare bazat pe vizionări anterioare, permite organizarea filmelor într-o listă de dorințe, oferă notificări pentru episoade noi apărute în serialele urmărite și centralizează conținutul a mai multor platforme de streaming într-o singură.

Din cauza problemelor legale, legate de drepturile de autor nu am considerat potrivit să implementez o aplicație similară cu aceasta pentru cărți. Proprietatea intelectuală a cărților nu trebuie încălcată și trebuie să respectăm legile în ceea ce privește multiplicarea edițiilor.

3.3. Structura bazei de date

Întâi de toate, am asigurat o bună structură a bazei de date, ca fiecare tabelă să aibă atributele necesare, iar relațiile dintre acestea să fie logice (Fig. 13). Astfel, am considerat necesare următoarele:

1) Tabela **Users**, care include:

- user_id – cheie primară, identificator unic pentru fiecare utilizator în parte, autoincremental;
- username – numele utilizatorului ales de acesta în momentul înregistrării necesită să fie unic;
- email – înscris tot de utilizator la înregistrare, de asemenea, acesta trebuie să fie unic;
- password_hash – parola utilizatorului hashuită pentru a fi privată și securizată;
- is_admin – câmp boolean (0/1) pentru determinarea dacă un utilizator este administrator al platformei sau nu;
- date_joined – marchează data la care s-a înregistrat utilizatorul.

2) Tabela **Books**, include câmpurile:

- book_id – cheie primară a tablei, identificatorul unic pentru fiecare carte, autoincremental;
- title – titlul cărții nu poate fi nul;
- author – autorul cărții, precum titlul, nu poate fi nul;
- genre – coloana pentru genul sau domeniul din care face parte cartea;
- cover_image_url – link către poza copertei cărții;
- description – descrierea cărții;
- rating – nota oferită de utilizatorii Goodreads cărții;
- date_added – data la care a fost adăugată cartea în baza de date;

3) **Reviews**, tabelă care conține următoarele coloane:

- review_id – cheie primară, identificator unic pentru fiecare recenzie;
- user_id – cheie străină către utilizatorul care a scris recenzia;
- book_id – cheie străină către cartea pentru care a fost scrisă recenzia;
- rating – notă acordată (între 1 și 5);
- review_text – comentariu/opinie a utilizatorului despre carte;
- review_date – data publicării recenziei.

4) Tabela **Loans** include coloanele:

- loan_id – cheie primară, identificator unic pentru fiecare împrumut;
- book_id – cheie străină către cartea împrumutată;
- lender_id – cheie străină către proprietarul cărții, care oferă cartea spre împrumut;
- borrower_id – cheie străină către utilizatorul care primește cartea spre împrumut;
- loan_start_date – data începerii împrumutului;
- loan_end_date – data estimată a returnării cărții;
- loan_status – statusul împrumutului.

5) **Wishlist**, tabelă ce prezintă urmatoarele câmpuri:

- user_id – cheie străină către utilizatorul care dorește carte;
- book_id – cheie străină către carte;

- `date_added` – data adăugării cărții în lista de dorințe;

Cheia primară a tabelei **Wishlist** este compusă din `user_id` și `book_id`, astfel asigurând unicitatea fiecărei combinații utilizator-carte.

6) **ReadBooks**, tabelă similară cu **Wishlist**, dar utilizată pentru cărțile deja citite de utilizator:

- `user_id` – cheie străină către utilizatorul conectat;
- `book_id` – cheie străină către cartea citită;
- `date_read` – câmp ce indică data la care a fost marcată cartea ca citită.

Similar cu tabela **Wishlist**, cheia primară este compusă din aceleasi prime două câmpuri.

7) **ChatMessage**, stochează mesajele din fiecare conversație a fiecărui împrumut și include următoarele câmpuri:

- `message_id` – cheie primară, identificator unic pentru fiecare mesaj existent;
- `loan_id` – cheie străină de tip Integer ce face referire către împrumutul din conversația căruia face parte mesajul;
- `sender_id` – cheie străină către tabela **Users**, pentru că este necesar să știm doar expeditorul mesajului, iar destinatarul este implicit din tabela **Loans**;
- `text` – textul mesajului efectiv;
- `ts` – momentul trimiterii mesajului înregistrat ca `DateTime`.

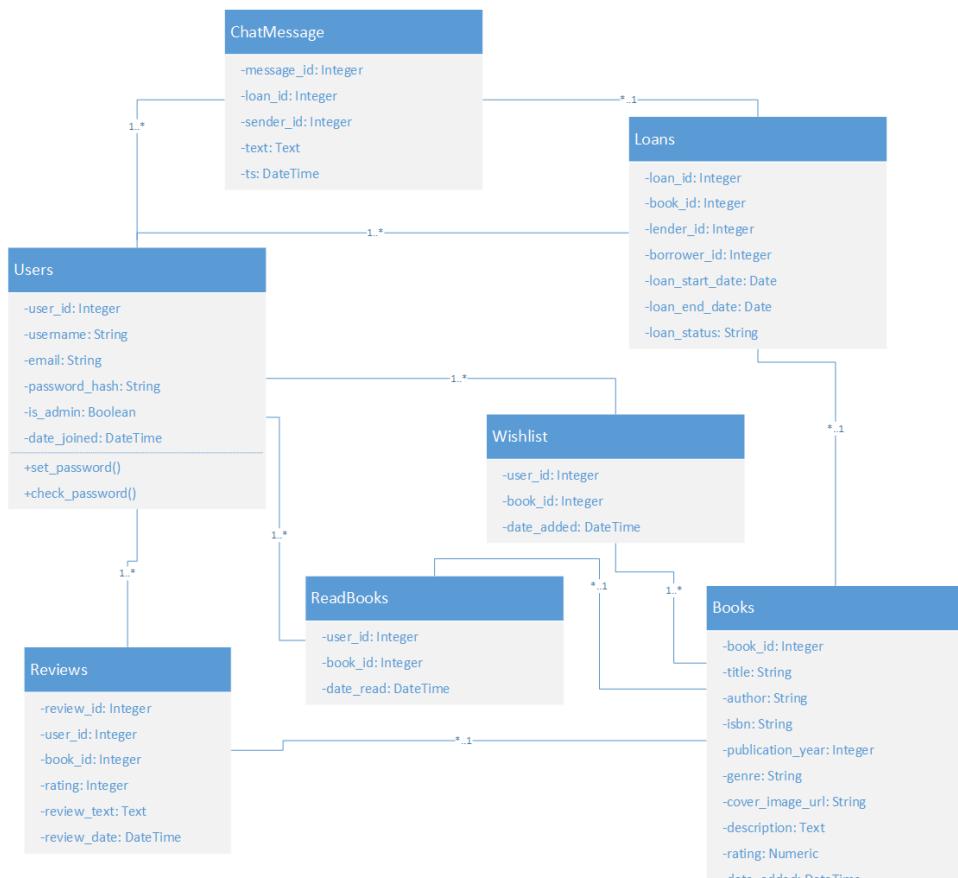


Fig. 13. Diagrama bazei de date

Toate relațiile dintre tabele sunt de tip one-to-many, iar relațiile de tip many-to-many au fost mediate prin tabele de joncțiune, cum sunt **Wishlist** și **ReadBooks**. După cum se poate observa și în Fig. 13, relațiile sunt următoarele:

- Users – Reviews: relația este de tip one-to-many, deoarece fiecare utilizator poate adăuga mai multe recenzii.
- Users – ChatMessage: relația este de tip one-to-many, deoarece fiecare utilizator poate trimite mai multe mesaje.
- Users – Loans: relație de tip one-to-many, deoarece fiecare utilizator poate lua parte la mai multe împrumuturi.
- Books – Reviews: relație de tip one-to-many, datorată faptului că fiecare carte poate avea scrise mai multe recenzii de la mai mulți utilizatori.
- Books – Loans: relație de tip one-to-many, deoarece fiecare carte poate fi subiect a mai multor împrumuturi.
- Loans – ChatMessage: relație de tip one-to-many, datorată faptului că fiecare împrumut poate avea în conversația aferentă mai multe mesaje.
- Users – ReadBooks, Books – ReadBooks: prin intermediul tabelei ReadBooks, se realizează relația de tip many-to-many între tabelele Users și Books, fiecare utilizator având posibilitatea să aibă mai multe cărți citite, iar fiecare carte poate fi citită de mai mulți utilizatori.
- Users – Wishlist, Books – Wishlist: tabela Wishlist este o tabelă de joncțiune care realizează relația many-to-many între tabelele Users și Books, relație necesară prin prisma faptului că fiecare utilizator poate adăuga mai multe cărți în lista de dorințe personale, iar fiecare carte poate fi adăugată de mai mulți utilizatori în această listă.

De menționat este că și tabela Reviews se comportă ca o tabelă de joncțiune, însă aceasta are și date adăugate (i.e. review_id, rating, review_text, review_date), fapt care m-a făcut să o consider o tabelă de sine stătătoare. Totuși, relația realizată este de aceasta între tabelele Users și Books este de tip many-to-many deoarece fiecare utilizator poate adăuga o recenzie la mai multe cărți, iar fiecare carte poate primi mai multe recenzii de la mai mulți utilizatori. De asemenea, tabela Reviews este independentă ca scop de celelalte tabele, având propria specializare – recenziile.

3.4. Popularea bazei de date și etapa de preprocesare a datelor

Pentru ca aplicația să aibă o funcționalitate practică, am considerat necesar un set de date cu informații relevante pentru proiectul nostru. Deși necesita modificări printr-un activitate de preprocesare a datelor, acest set de date de pe Kaggle [20] mi s-a părut cel mai potrivit, conținând următoarele informații: isbn13, isbn10, titlu, subtitlu, autori, genuri (categorii), coperta cărții sub formă de link URL, descriere, anul publicării, nota medie primită de carte pe aplicația Goodreads, numărul de recenzii primite, cât și numărul de pagini ale prezentei ediții. Dintre aceste coloane, au fost păstrate doar câteva care s-au pretat mai bine pe structura bazei de date deja concepute. Astfel au fost eliminate coloanele: isbn13, subtitlu, numărul recenziilor și numărul de pagini.

Totodată, 5% dintre înregistrări aveau câmpuri nule pe coloana cu link-ul pentru coperta cărții, iar pentru un proces mai rapid, am considerat mai bine ca acestea să fie șterse complet. Pentru cele aproximativ 1% care nu aveau autor, am căutat autorul (autorii) în

funcție de ISBN. La fel am procedat atât pentru cele care nu aveau titlul înregistrat sau prezenta erori din cauza caracterelor ce nu puteau fi procesate de către editor (de exemplu titluri ca “??????”), cât și pentru cele câteva care nu aveau un gen asignat. În plus, la cărțile la care am considerat că sirul de autori este prea lung, am eliminat din aceștia, considerând contribuția lor irelevantă pentru exercițiul nostru. Ultimele modificări în ceea ce privește înregistrările din baza de date au fost făcute în urma observațiilor din timpul testării aplicației. Am realizat deci, că unele link-uri către copertă nu sunt valide. Procesul de verificare a validității link-urilor fiind unul exhaustiv și neputând să-l automatizez în niciun fel, am modificat doar câmpurile observate în timpul testării.

Demersul de preprocesare a datelor a fost desfășurat în Excel, dar și în linia de comandă prin interogări (SELECT) și comenzi de modificare (UPDATE) din limbajul SQL.

3.5. Design-ul aplicației

Design-ul aplicației a fost realizat de la 0 cu ajutorul aplicației Figma. Paleta de culori și fonturile din cadrul aplicației au fost alese astfel încât să se încadreze cât mai bine în tema de bibliotecă vintage. Logo-ul este de asemenea un concept original, realizat cu ajutorul Adobe Illustrator, din care poate fi dedus și numele aplicației – OurBooks (Fig. 14). Prin acest nume vreau să subliniem aspectul de comuniune și socializare între cititori, cât și aspectul colaborativ al aplicației.



Fig. 14. Logo-ul aplicației

Paginile de autentificare și înregistrare în platformă au fost primele realizate. Acestea includ o imagine de fundal sugestivă, cu rafuri de cărți și logo-ul platformei în stânga paginii. În dreapta se află cardul în care utilizatorul introduce datele necesare acțiunii sale. Din fiecare dintre cele două pagini, utilizatorul are posibilitatea de a naviga pe celălaltă, în funcție de preferință și situație. Dacă are un cont deja creat, îi este sugerat să se autentifice pe pagina de Login (Fig. 15), iar dacă nu are cont, se poate înregistra pe pagina de Register (Fig. 16). Toate elementele sunt centrate, distanțate și distribuite cu grijă pe pagină pentru a crea o interfață plăcută și accesibilă utilizatorului.

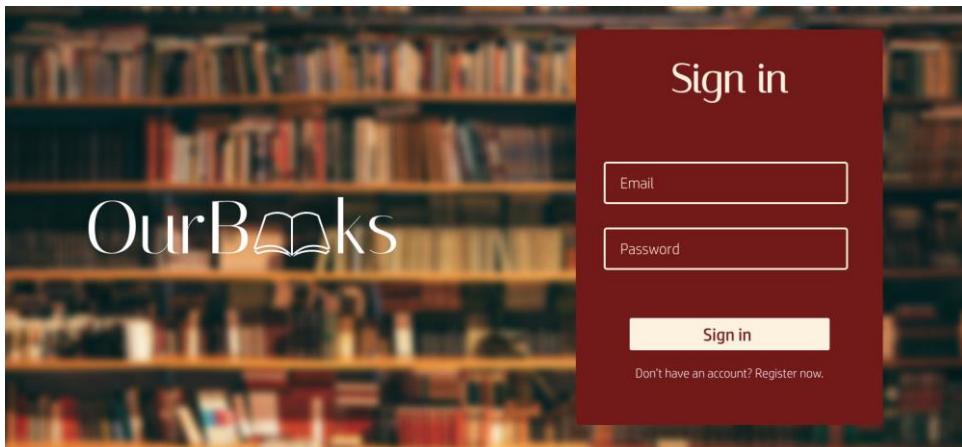


Fig. 15. Design-ul paginii de Login



Fig. 16. Design-ul paginii de Register

Desfășurând mai mult subiectul pe care l-am început în cadrul unei secțiuni anterioare – **3.2** – deși design-ul este unul original, iar elementele sunt construite de mine, inspirația în ceea ce privește experiența utilizatorului și interacțiunea acestuia cu platforma provin de la aplicația Stremio, despre care am discutat anterior. Odată accesată platforma prin contul utilizatorului, acesta are acces la mai multe rânduri de sugestii de cărți bazate pe preferințele sale, recenziile lăsate și lecturile personale. Conceptul inițial se poate observa în Fig. 17, unde există următoarele elemente:

- Bara de navigație cu logo-ul aplicației în partea stângă;
- Butoanele ce ne trimit la pagina de Home, Library (biblioteca personală) și My Loans (împrumuturile utilizatorului);
- Bara de căutare;
- Un buton cu trimitere către profilul utilizatorului;
- Secțiunea de recomandări diverse bazată pe preferințe (cu posibilitate de derulare);
- Secțiunea de recomandări pe baza autorului favorit al cititorului (cu posibilitate de derulare).

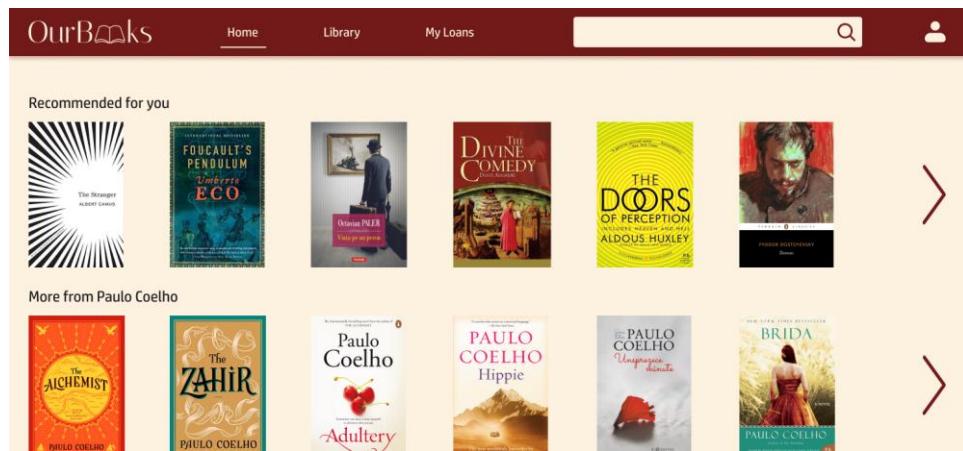


Fig. 17. Conceptul inițial al paginii de Home

În același timp, am considerat relevant, pentru fazele incipiente ale aplicației, un concept pentru pagina care va prezenta elementele importante ale unei cărți – Book Description (Fig. 18).



Fig. 18. Conceptul inițial al design-ului paginii de descriere a cărții

Astfel, componentele prezente în această pagină sunt:

- Bara de navigație, care cuprinde aceleași elemente precum și în pagina principală, Home;
- Imaginea de copertă a cărții;
- Titlul și autorul cărții;
- Media notelor primite de la utilizatori pe pagina Goodreads, însorită de 5 steluțe reprezentând acest quantum;
- Descrierea cărții;
- Genurile în care este încadrată cartea;
- Notele utilizatorilor aplicației, însorite de recenziile acestora și alăturate avatarului utilizatorului (neimplementat) și username-ului acestuia;

- Posibilitatea de a adăuga o recenzie, cât și de a adăuga cartea respectivă în lista de dorințe.

În plus, apăsarea genului cărții se vrea a naviga la o secțiune ce afișează toate cărțile încadrate la acea categorie (Fig. 19). Această pagină include o listă de cărți desfășurată sub titlul genului, fiecare carte prezentând poza de copertă, titlul, autorul, descrierea, nota medie a recenziilor și un buton de adăugare a cărții în lista de dorințe.

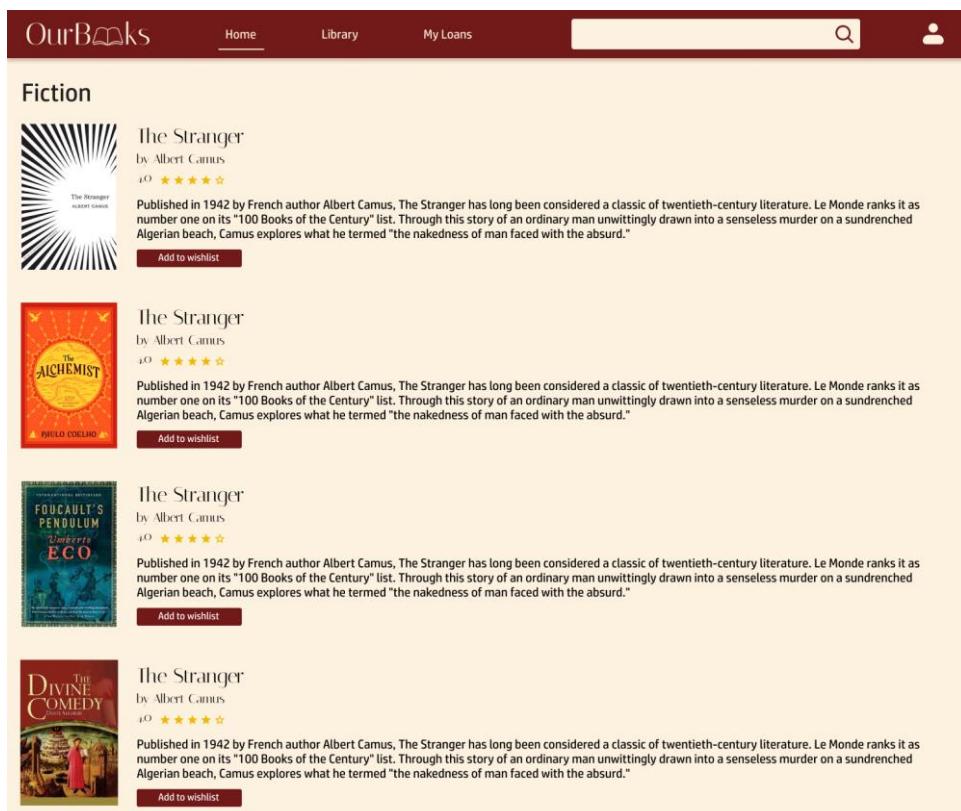


Fig. 19. Conceptul inițial al paginii de cărți din același gen (mock-up)

Toate acestea sunt ideile inițiale ale interfeței aplicației. Odată cu implementarea, aspectul paginilor a suferit mai mult sau mai puțin modificări, iar toate aceste aspecte vor fi surprinse în următoarele secțiuni, care vor descrie algoritmi, tehnicalități, echipamente utilizate și prezentarea mai în detaliu a aplicației.

4. Documentație tehnică

4.1. Tehnologiile utilizate

Pentru a începe lucrul în cadrul aplicației a fost întâi nevoie de a stabili tehnologiile pe care le voi utiliza în implementarea ideilor mele. Am decis să îmbin mai multe limbaje de programare de actualitate, ce sunt folosite în numeroase companii pe proiectele lor reale. Am considerat necesar lucrul cu aceste limbaje mai puțin utilizate până acum în cadrul cursurilor și proiectelor mele pentru a oferi utilizatorilor o interacțiune cât mai plăcută cu aplicația.

4.1.1. Inițializarea proiectului cu Vite

Conform paginii oficiale [21], Vite este un instrument ce își propune să ofere o experiență mai rapidă și mai lină pentru dezvoltatorii de aplicații și este constituit din două părți majore: un server de dezvoltare ce oferă îmbunătățiri bogate de funcții față de modulele ES native și o comandă de construire ce pune tot codul la un loc prin intermediul Rollup, preconfigurat să ofere resurse statice extrem de optimizate pentru producție.

Pe scurt, ca să rezumăm informațiile celor de la Vite de pe pagina lor, acest instrument m-a ajutat în următoarele instanțe:

- Hot Module Replacement (HMR) – datorită faptului că atunci când editezi un fișier Vite trebuie doar să invalideze precis lanțul de legătură dintre modulul editat și cea mai apropiată limită HMR, pagina editată (stilurile sau componenta React) se actualizează complet, automat, aproape instant, păstrând starea curentă și formularele completate.
- Server de dezvoltare rapid – datorită Vite, după modificări nu au fost necesare așteptări lungi, iar aplicația a fost pornită aproape instant prin comanda `npm run dev`.
- Integrare facilă a modulelor CSS – deoarece instrumentul acesta suportă variate posibilități în dezvoltarea aplicației, implementarea modulelor CSS nu a necesitat configurare complicată, iar importarea `react-icons`, `axios`, `react-router-dom`, a putut fi realizată fără probleme.
- Crearea rapidă a proiectului – proiectul a fost inițializat fără configurare manuală și a necesitat doar câteva comenzi, lăsând Vite să se ocupe de tot.

4.1.2. React

React permite dezvoltatorului să construiască interfețe din piese individuale numite componente, ca apoi să fie combinate în ecrane, pagini și aplicații. Fie că lucrezi pe cont propriu sau cu alții dezvoltatori împreună, React funcționează la fel. Este făcut pentru a combina ușor componente scrise de oameni diferiți, echipe și organizații. Componentele din React sunt funcții JavaScript. Ele primesc date și returnează ce ar trebui să apară pe ecran. În plus, se pot oferi noile date ca răspuns la o interacțiune, ca atunci când utilizatorul oferă niște date de intrare.

Totuși React este doar o bibliotecă. Permite punerea componentelor împreună, dar nu prescrie cum să faci rutare sau fetch de date. Pentru realizarea unei aplicații complete cu React se recomandă un framework precum Next.js sau Remix [22]. Însă se pot folosi și alte framework-uri, precum în aplicația mea, Flask.

JSX este o extensie de sintaxă pentru JavaScript care permite scrierea unui HTML într-un fișier JavaScript. Chiar dacă sunt alte moduri de a scrie componente, majoritatea dezvoltatorilor de React preferă JSX deoarece este concis și multe proiecte îl folosesc.

Web-ul a fost construit cu HTML, CSS și JavaScript. Timp de mulți ani, dezvoltatorii au păstrat conținutul din HTML, design-ul din CSS și logica din JavaScript – de obicei în fișiere separate. Dar deoarece internetul a devenit din ce în ce mai interactiv, logica a determinat conținutul. Ceea ce înseamnă că JavaScript se ocupa de HTML [23].

Mai precis, React a facilitat următoarele aspecte:

- Structurarea aplicației pe componente – interfața este compusă din pagini precum `Login.jsx`, `Register.jsx`, `BookDescription.jsx`. Componentele m-au ajutat în reutilizarea codului, fără să rescriu mark-up-ul pentru fiecare pagină.

- Actualizarea rapidă a interfeței – React actualizează automat paginile doar în locurile unde au fost făcute modificări.
- react-router-dom – m-a ajutat pentru a comuta cu ușurință între pagini fără a fi necesară reîncărcare, iar, după autentificare, se navighează automat la pagina de Home.
- Separarea între UI și logică – cu ajutorul React am reușit să separ stilurile (.module.css) și să structurez proiectul pe directoare.

4.1.3. Flask

Flask este un framework Python creat pentru a începe ușor și rapid proiecte și are abilitatea de a scala chiar și aplicații complexe. Flask este un WSGI (Web Server Gateway Interface), ce reprezintă un standard în Python pentru dezvoltarea de aplicații web. WSGI este specificația unei interfețe comune dintre serverele web și aplicații [24]. În cazul meu, acesta a fost esențial pentru susținerea părții de backend:

- Gestionarea rutelor și definirea API requests.
- Gestionarea răspunsurilor HTTP.
- Definirea modelelor și integrarea SQLAlchemy pentru gestionarea bazei de date.

4.1.4. SQLite

SQLite este o bibliotecă din C, care oferă o bază de date simplă și compactă, bazată pe stocare pe disc, ce nu necesită un proces pe server separat și permite accesarea bazei de date utilizând o variantă nestandardizată a limbajului de interogări din SQL [25]. Atât pentru a interoga baza de date, cât și pentru a modifica anumite câmpuri, am folosit sintaxa sistemul de gestiune a bazei de date oferit de SQLite. În cadrul directorului /instance se regăsește baza de date ce este accesată prin comanda sqlite3. În Fig. 20 se pot observa toate tabelele din cadrul bazei noastre de date, o interogare ce prezintă toți utilizatorii înregistrăți în aplicație, alături de toate detaliile stocate și denumirea ultimei versiuni a structurii bazei de date migrate.

```
PS C:\Users\stefal\OneDrive\Documente\faculta\an4sem2\licenta\cod> cd instance
PS C:\Users\stefal\OneDrive\Documente\faculta\an4sem2\licenta\cod\instance> sqlite3 .\books.db
SQLite version 3.44.2 2023-11-24 11:41:44 (UTF-16 console I/O)
Enter ".help" for usage hints.
sqlite> .tables
alembic_version  chat_messages  read_books  users
books           loans          reviews      wishlist
sqlite> select * from users;
1|testuser1|testuser1@example.com|$2b$12$pwXgeaia2uZtmC7ku7UXB.MJVTytcKxRM976BeXJkpGSGRjba6RQe|1|2025-03-25 13:31:02
2|testuser2|testuser2@example.com|$2b$12$ReW2xuawWebqEVVALiwhuQ.mC10G3cdIpKu0nk6sjqQ3bLY5oDy|0|2025-03-25 13:31:02
3|testuser3|testuser3@example.com|$2b$12$0ReB5k9WKGX/qcRZWuqc.9Mw2sBxUyc5.ZhBlIba4TDks2B8Tln2|0|2025-03-27 12:14:18.472928
4|StefanCiuculan05@gmail.com|$2b$12$DPG0DT.R/hRaNP5VH889uUGg/J4mvCZ/seoIQHUca3oo8kvPC35K|0|2025-04-03 10:42:17.479749
sqlite> select * from alembic_version;
0bd259bdd011
```

Fig. 20. Interacțiunea cu SQLite din terminal

4.2. Structura aplicației

Proiectul este structurat cu ajutorul unui arbore de directoare. Directorul /app include fișierele necesare pentru partea de backend, în /instance regăsim baza de date, în /migrations se află fișiere de configurare și migrări ale bazei de date, directorul /frontend include tot ce este necesar pentru interfața aplicației, iar restul reprezintă fișiere necesare pentru rulare și inițializare (Fig. 21).

4.2.1. Tabelele bazei de date

Printre primele module implementate este models.py, în care se stabilește structura bazei de date cu ajutorul SQLAlchemy, detaliată în secțiunea 3.3. Fiecare clasă reprezintă o

tabelă, iar fiecare atribut al clasei reprezintă un câmp al tabelei. Se definesc deci tipurile de date, constrângerile, metodele necesare pentru securizarea parolelor și verificarea lor prin intermediul bcrypt. Fiecare clasă (tabelă) este integrată în structura bazei de date prin relații și chei străine care fac legăturile între ele. Totodată, se folosesc parametri specifici pentru a garanta integritatea bazei de date în ceea ce privește ștergerea în cascadă a “orfanilor” în cazul eliminării anumitor înregistrări din tabelele principale.

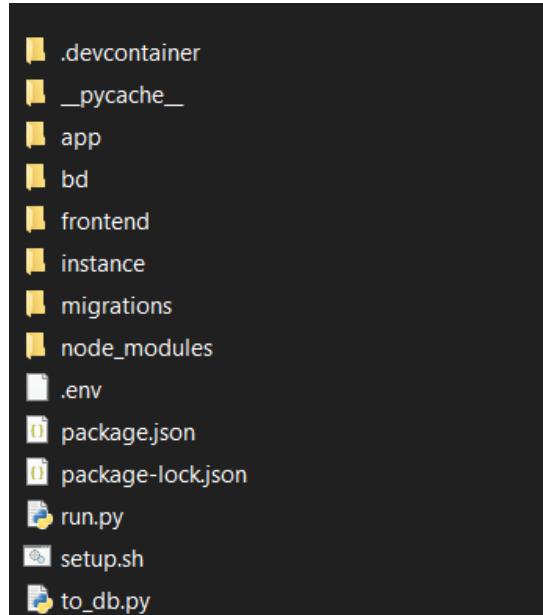


Fig. 21. Directoarele din cadrul proiectului

Bcrypt și stocarea unei parole în siguranță

MD5, SHA1, SHA2, SHA3 sunt funcții de hashing pentru utilizare generală, create să creeze uriașe cantități de date în cel mai scurt timp posibil. Asta înseamnă că sunt fantastice pentru a asigura integritatea datelor și foarte slabe în a stoca parole.

Un server modern poate calcula 330 MB a hash-ului MD5 într-o secundă. Dacă utilizatorii folosesc parole cu litere mici, alfanumerice, de 6 caractere, toate acele parole pot fi încercate în aproximativ 40 de secunde.

Este important de notat faptul că sistemul de „salt” este inutil în prevenirea atacurilor bazate pe dicționare (dictionary attacks) sau a celor prin forță brută (brute force attacks). Orice tip de „salt” am folosi, nu influențează în niciun fel timpul necesar unui atacator pentru a compromite parola unui utilizator, având la dispoziție hash-ul parolei și „salt”-ul din baza de date.

Bcrypt rezolvă problema prin faptul că este foarte încet. Folosește o variantă de keying a algoritmului de criptare Blowfish și introduce un factor de lucru (work factor), care permite să se observe cât de costisitoare va fi funcția hash. De exemplu, s-a observat că o parola pe care MD5 o criptează în mai puțin de o microsecundă, bcrypt reușește să o cripteze în 0.3 secunde. Asta rezultă în loc de a se compromite o parolă la fiecare 40 de secunde, aceasta se întâmplă la 12 ani [26].

Astfel, funcția `generate_password_hash()` generează o parolă hash folosind acest bcrypt. Primul parametru reprezintă parola necriptată, al doilea este parametrul `rounds`, care determină complexitatea “salt”-ului, valoarea implicită fiind 12 (care a fost folosită și în cazul aplicației noastre). Prefix, este al treilea parametru posibil care determină versiunea algoritmului folosit pentru crearea parolei criptate.

`check_password_hash()` compară o parolă hash cu o parolă-candidat. Parola-candidat este întâi transformată într-o parolă hash, iar cele două sunt comparate ulterior. Valoarea returnată va fi `true` sau `false` în funcție de rezultatul comparației dintre cele două parole [27].

4.2.2. Modulul de inițializare

În modulul de inițializare se definește funcția ce creează și configerează aplicația propriu-zisă, `create_app()`, precum și alte funcții globale ce vor fi folosite în cadrul lor module (`db()`, `migrate()`, `bcrypt()`, `jwt()`).

În funcția `create_app()` se creează instanța Flask, se setează locația bazei de date, cheia secretă pentru JWT, timpul în care un token expiră și câteva opțiuni necesare pentru conexiunea cu baza de date. Următorul pas pe care modulul îl urmează este inițializarea extensiilor globale. Astfel că se configerează SQLAlchemy, Flask Migrate, Bcrypt și JWTManager.

Urmează să se înregistreze prefixul `api` pentru toate rutele din `routes.py`, iar la final, prin funcția `shutdown_session()`, se asigură închiderea conexiunii la finalul fiecărei cereri. Ultima funcție, deși nu a fost apelată direct în backend-ul aplicației, aceasta este automat apelată de Flask la finalul fiecărei cereri HTTP. Acest fapt are loc datorită `@app.teardown_context`.

Rularea aplicației în backend a fost realizată cu un modul `run.py`, ce apelează inițializarea aplicației prin funcția `create_app()`, iar apoi, prin intermediul funcției `run()`, se rulează aplicația pe un server local de dezvoltare.

4.2.3. Stocarea datelor în tabela books

Pentru înregistrarea în tabela `books` a tuturor datelor preprocesate anterior, am considerat necesară o funcție care să automatizeze acest proces, care de altfel ar fi fost unul exhaustiv. Am utilizat deci, un fișier `.csv` în care au fost stocate toate datele obținute din setul de date curățat. Pentru citirea fișierului am folosit `encoding="ISO-8859-1"` din cauza unei erori provocate de imposibilitatea UTF-8 să citească anumite caractere speciale aflate în titlurile, numele autorilor sau descrierile cărților. În mod implicit, funcția `read_csv()` folosește pentru codificare UTF-8, care nu suportă caractere precum “ș, ţ, î”.

Cu o structură iterativă `for` am trecut pe fiecare linie a fișierului și am stocat în câmpul aferent fiecărei coloane, informația din tabel. La fiecare iterare se adaugă căte o carte în baza de date, iar la final se face un `commit()` pentru a da drumul la toate schimbările puse în așteptare și se încarcă toate tranzacțiile curente.

4.2.4. Rutele

Probabil unul dintre cele mai complexe module din cadrul proiectului este cel în care implementez rutele, cererile HTTP, iar acesta reprezintă punctul de legătură cel mai important între partea de backend și cea de frontend. Fiecare rută are un URL unic, care este precedat de prefixul `api` definit anterior cu ajutorul Blueprint. Totodată, fiecare rută are

una sau mai multe metode, ce determină comportamentul cererii HTTP. Cererile ce nu au o metodă specificată, folosesc, în mod implicit metoda GET.

Metoda GET este folosită pentru a returna date de la server [28]. Exemple de rute ce folosesc metoda GET în aplicație sunt: afișarea cărților aflate în lista de dorințe ale utilizatorului (`api/wishlist`) sau cea pentru returnarea recomandărilor bazate pe preferințe (`api/user/recommendations`).

POST trimite date la server spre a fi procesate. Metoda aceasta nu este considerată o operație sigură pentru că are puterea de a schimba starea serverului și poate cauza efecte adverse asupra stării serverului când este executată [28]. Această metodă am folosit-o în cereri precum cea de adăugare a unei recenzii (`api/reviews/add`) sau precum înregistrarea unui utilizator în baza de date (`api/register`).

DELETE este o metodă utilizată pentru a șterge o resursă de pe server. Precum POST, nu este considerată o metodă sigură, însă poate fi necesară în anumite cazuri. În cadrul aplicației, metoda DELETE a fost folosită pentru cererea de eliminare a unei cărți din lista de dorințe (`api/wishlist/remove`), dar și în combinație cu metoda POST în momentul în care se oferea o carte spre a fi împrumutată sau oferta era revocată (`api/loans/offer`).

Din cauză că uneori reprezentările obiectelor devin foarte mari, o cerere PUT poate fi costisitoare. Astfel că, metoda PATCH rezolvă această problemă, fiind mai eficientă și modificând doar parțial resursa de pe server [28]. Metoda PATCH a fost utilă în cazul modificării statusului unui împrumut (`/api/loans/<int:loan_id>/status`) pentru a nu fi necesară retrimiterea acelorași informații neschimbate precum titlul cărții, autorii, părțile implicate în împrumut etc.

Codurile de stare HTTP

În cadrul rutelor, pentru a trata fiecare caz de utilizare și erori posibile, am utilizat sistemul standard de coduri de stare. Am considerat o necesitate implementarea de acest fel pentru a cunoaște atât în timpul testării, cât și în timpul utilizării aplicației care sunt problemele și unde se mai poate lucra. Mai jos, în Tabel 1, se poate observa mai clar utilitatea acestor coduri.

Cod	Răspuns standard	Utilitate în cadrul aplicației
200	OK	Autentificare reușită, carte deja existentă în lista de dorințe, carte eliminată cu succes din lista de dorințe, afișare reușită a listei de dorințe, recomandare reușită etc.
201	Creat	Utilizator înregistrat pe platformă, carte adăugată cu succes în lista de dorințe, recenzie adăugată cu succes, carte marcată ca citită etc.
400	Cerere greșită	Câmpuri necompletate în cadrul formularului de înregistrare pe platformă, status al împrumutului neconform, mesaj fără conținut.
401	Neautorizat	Credențiale invalide la autentificare.
403	Interdicție	Utilizatorul care vrea să schimbe statusul unui împrumut nu este proprietarul de drept al cărții, utilizatorii care nu sunt implicați în cadrul unui schimb de cărți nu pot folosi chat-ul respectivului împrumut.

404	Resursă inexistentă	Cartea ce încearcă să fie adăugată în lista de dorințe nu a putut fi găsită, cartea ce încearcă să fie eliminată din lista de dorințe nu se află în listă, cartea la care se încearcă a se adăuga o recenzie nu a putut fi găsită etc.
409	Conflict	Doi utilizatori nu pot avea același mail și nici același nume de utilizator, iar un utilizator nu poate oferi mai multe recenții pentru o carte.

Tabel 1. Codurile de stare HTTP utilizate

4.2.5. Sistemul de recomandări

Reprezentând cea mai importantă componentă a aplicației, prin sistemul de recomandări mi-am propus să se ofere utilizatorilor cele mai calitative sugestii bazate pe preferințele lor. Pentru a ajuta algoritmul să le propună cele mai adecvate idei de lecturi noi, utilizatorii pot contribui în 3 moduri:

- Să-și adauge cărți în lista de dorințe;
- Să-și marcheze cărțile deja citite;
- Să adauge recenții cărților.

Algoritmul de recomandare este bazat pe conținutul textual al cărților pentru a asigura relevanța sugestiilor.

4.2.5.1. TF-IDF

TF-IDF (Term Frequency – Inverse Document Frequency) este o măsurătoare statistică folosită în procesarea de limbaj natural (NLP – Natural Language Processing) și în accesarea de informații pentru a evalua importanța unui cuvânt într-un document relativ la o colecție de documente (corpus). În mod deosebit față de simpli algoritmi pentru determinare frecvenței de cuvinte, TF-IDF balansează cuvintele comune și pe cele rare pentru a pune în evidență termenii cei mai comuni [29].

Frecvența termenilor (TF) măsoară cât de des apare un cuvânt într-un document. O frecvență mai mare sugerează o importanță pe măsură. Dacă un cuvânt apare des într-un document, cel mai probabil este relevant pentru conținutul respectiv. Cu toate acestea, TF nu ia în calcul importanța globală a cuvântului în întregul corpus, iar prepozițiile și cuvintele de legătură pot avea un scor mare conform TF, însă ele de fapt nu au o relevanță anume în cadrul documentelor [29].

$$TF(t, d) = \frac{\text{De câte ori apare termenul } t \text{ în documentul } d}{\text{Numărul total de termeni ce apar în documentul } d} \quad [29]$$

Frecvența inversă a documentelor (IDF) reduce importanța cuvintelor comune din conținutul a mai multor documente, în timp ce amplifică relevanța unor cuvinte mai rar întâlnite. Dacă un termen apare în mai multe documente, este cel mai probabil însemnat și specific [29].

$$IDF(t, D) = \log \frac{\text{Numărul total de documente din corpusul } D}{\text{Numărul de documente ce conțin termenul } t} \quad [29]$$

Logaritmul din relația de mai sus este utilizat pentru a atenua efectul valorilor foarte mari sau foarte mici, asigurând astfel un scor IDF corespunzător. Așadar neajunsurile scorului IDF se rezumă la faptul că nu ia în considerare cât de des apare un termen într-un anume document, iar un termen poate fi rar pentru tot corpusul, ceea ce ar rezulta într-un scor IDF mare, dar irelevant într-un document specific (scor TF mic) [29].

4.2.5.2. Similaritate cosinus

Implementarea metodei similarității cosinus este utilă în clasificarea datelor pe baza numărului de obiecte ce au anumite asemănări. Este deseori utilizată pentru calcularea unghiului cosinus dintre vectorii documentelor și pentru dezvoltarea algoritmilor de clustering ierarhic aplicați asupra documentelor, ca o metodă de a crește eficiența și performanța.

Similaritatea cosinus este o rată a asemănării dintre doi vectori comparați, calculată pe baza produsului unghiurilor cosinus ale celor doi vectori. Deoarece $\cos 0 = 1$ și orice alt unghi are cosinusul mai mic decât 1, atunci doi vectori sunt considerați similari în cazul în care valoarea similarității cosinus este 1 [30]. Formula pentru calculul similarității cosinus:

$$\cos \alpha = \frac{A \times B}{|A| \times |B|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \quad [30]$$

4.2.5.3. Implementarea sistemului

Primul atribut al clasei este `_vectorizer`, un obiect folosit pentru a realiza conversia descrierilor cărților în vectori numerici. `_matrix` reprezintă matricea TF-IDF, iar fiecare linie din cadrul acesteia este un vector numeric, corespunzător unei cărți. Ultimul atribut al clasei este `_book_ids`, lista identificatorilor unici ai cărților existente deja în baza de date. Fiecare identificator corespunde unei linii din matricea TF-IDF.

Pentru început, în funcția `_build()`, se preiau datele cărților din modelul Books, iar titlurile și descrierile fiecărei cărți sunt concatenate pentru fiecare în parte. Cu ajutorul clasei `TfidfVectorizer()` din `scikit-learn`, se transformă textele în matrice numerică TF-IDF.

Metoda `similar_books()` produce de fapt recomandările personalizate, iar parametrii necesari sunt următorii:

- `seed_ids` sunt identificatorii unici ai cărților la care deja utilizatorul a adăugat recenzii, le-a trecut în lista de dorințe s-au pe care le-a citit.
- `topn` reprezintă numărul de recomandări solicitate. Câte recomandări sunt cerute, atâtea vor fi și afișate pe pagină (în cazul nostru, am decis 24).
- `exclude_ids` este un parametru necesare pentru a evita recomandarea cărților de care autorul deja știe. Acestea sunt practic aceleași cu cele din `seed_ids`.
- `weight_map` este un dicționar în care se stochează cărțile cunoscute, împreună cu ponderile lor, acestea reflectând preferința utilizatorului în ceea ce privește respectivele cărți.

În cadrul acestei metode se începe cu construirea vectorului de profil al utilizatorului. În cazul în care parametrul `weight_map` nu este nul, vectorii numerici ai cărților sunt combinați ponderat în funcție de importanța lor pentru utilizator. Totuși dacă nu este transmis parametrul acesta, se calculează doar media vectorilor cărților preferate.

Pentru următorul pas se calculează similaritatea cosinus dintre vectorul de profil al utilizatorului și fiecare vector numeric de carte în parte. În final, rezultatele obținute sunt sortate descrescător în funcție de scorul de similaritate și se filtrează datele pentru ca toate cărțile deja cunoscute de către utilizator să fie eliminate.

În cadrul cererii ce se apelează pentru generarea recomandărilor (`api/user/recommendations`) se pun la socoteală atât cărțile adăugate de autor în lista de dorințe, cât și cele la care a scris recenzii și pe care le-a citit deja. Fiecare categorie are o pondere personalizată, în funcție de cum am considerat relevanța acesteia. Recenziile având atașate o notă, am considerat relevantă nota pentru recomandări. Astfel că, o carte la care a fost oferită o notă de 5, va avea o pondere de +2, una cu o notă de 4 va primi ponderea +1, cele cu nota 3 primesc pondere 0. Celor cu note sub 3 le-am asignat o pondere negativă (-1). Cărțile citite primesc pondere +1, iar cele adăugate în lista de dorințe +2. Acest sistem de scoruri ajută în a nu oferi recenziile eronate. Nu dorim un sistem de recomandări pasiv, ci unul atent la activitatea utilizatorului.

4.2.6. Migrările din cadrul bazei de date

Alembic este un instrument de migrare pentru bazele de date, utilizat împreună cu SQLAlchemy [31]. În urma inițializării `alembic` se creează mediul de migrare, un director de scripturi necesare pentru actualizările bazei de date. Doresc să punctez doar două aspecte importante generate automat odată cu inițializarea mediului.

- `env.py` este un script Python care rulează de fiecare dată când `alembic` este invocat. Conține instrucțiuni pentru configurarea și generarea motorului SQLAlchemy, procură o conexiune la motor împreună cu tranzacția, iar apoi invocă motorul de migrare, utilizând conexiunea ca sursă a conectivității bazei de date [32].
- Directorul `/versions` ține evidența scripturilor tuturor versiunilor [32]. În fiecare dintre aceste scripturi se pot observa modificările aduse structurii bazei de date în versiunea aferentă, acțiunile ce vor fi operate odată cu un upgrade al acestei versiuni sau cu un downgrade.

4.2.7. Paginile de autentificare și înregistrare

Secțiunile de autentificare și înregistrare gestionează accesul utilizatorilor la platformă. Funcționalitățile lor sunt implementate prin două componente, `Login.jsx` și `Register.jsx` cu modulele aferente pentru stiluri, `Login.module.css`, respectiv `Register.module.css`.

Componenta de autentificare permite utilizatorului să-și introducă credențialele (email și parolă) pentru a-și accesa contul. Funcționalitatea paginii de login constă în următoarele:

- Se validează local câmpurile complete.
- Se trimite o cerere de tip POST cu email-ul și parola introdusă de utilizator.
- Token-ul JWT se salvează în `localStorage`.
- Se salvează identificatorul unic al utilizatorului (`uid`) din payload-ul token-ului.
- În caz de succes, se afișează un mesaj și utilizatorul este redirecționat direct către pagina principală.
- În cazul în care credențialele nu corespund, se afișează un mesaj de eșec.

Înregistrarea are rolul de a salva datele utilizatorului în baza de date și a-i permite ulterior, prin autentificare, accesul la platformă. Sunt puse la dispoziție 4 câmpuri: numele de utilizator, email-ul, parola și confirmarea parolei. Următoarele funcționalități sunt implementate:

- Parolele sunt comparate printr-o validare locală. Acestea două trebuie să se potrivească.
- Către backend este trimisă o cerere de tip POST.
- În cazul în care înregistrarea a avut succes, se afișează un mesaj de confirmare și utilizatorul este redirecționat pe pagina de login.
- Dacă este întâmpinată o eroare, se afișează un mesaj personalizat, în funcție de situație.

4.2.8. Pagina principală

În cadrul paginii principale sunt prezentate recomandările personalizate către utilizatorul verificat prin pagina de autentificare. Aceasta este implementată în componenta `Home.jsx` și pagina este stilizată cu ajutorul modulului `Home.module.css`. Funcționalitatea paginii este următoarea:

- Prin cererea HTTP (`api/user/recommendations`) se afișează trei rânduri de cărți recomandate: `books` (recomandări generale, bazate pe recenziile utilizatorului, lista de dorințe și cărțile citite), `secondary` (recomandări de cărți scrise de autorul preferat), `genre_books` (recomandări de cărți din genul preferat).
- Bara de navigație ajută utilizatorul să poată accesa oricare dintre paginile aplicației. Prin cele trei butoane principale, el poate accesa lista de dorințe, cărțile împrumutate, lista cărților citite și pe cea cu cărți la care a scris recenzii.
- Tot în bara de navigație se află bara de căutare, prin intermediul căreia poate căuta cărți din baza de date pe baza unor cuvinte cheie introduse. Sirul de caractere va fi căutat întâi în titlul cărților, apoi în câmpul cu autorii cărților, iar apoi în descrierile acestora.
- În locul butonului de profil, care reprezinta ideea inițială în partea din dreapta a barei de navigație, se află un buton de ieșire din cont (`log out`).
- Cu ajutorul bibliotecii `framer-motion` am implementat un sistem de parcursere a cărților recomandate cu glisare pe orizontală
- În funcție de lățimea ferestrei browser-ului, este calculat numărul de cărți afișate într-un set de recomandări pe rând, pentru a asigura o experiență plăcută pentru toți utilizatorii.

4.2.9. Pagina de descriere a cărții

În componentă implementată prin `BookDescription.jsx` și modulul de stiluri `BookDescription.module.css` se afișează toate informațiile unei cărți selectate necesare utilizatorului. Odată selectată o carte printr-un click pe cardul ei, utilizatorului este redirecționat pe această pagină care ar următoarea funcționalitate:

- Se realizează mai multe cereri către backend pentru a se obține următoarele date: informațiile despre carte (`/api/books/:id`), statutul cărții în lista de dorințe (`/api/wishlist`), lista recenziilor (`/api/books/:id/reviews`), lista de cărți citite de către utilizator (`api/library/read`).

- Pe pagină sunt afișate deci titlul cărții și autorul, descrierea, nota cărții de pe platforma Goodreads însotită de steluțe sugestive, genul în care se încadrează scrierea, coperta cărții, lista de recenzii și meniul de acțiuni.
- Meniul de acțiuni cuprinde butoane pentru adăugarea/eliminarea unei cărți din lista de dorințe, marcarea unei cărți ca citită sau necitită, oferirea cărții spre împrumut sau retragerea ofertei și solicitarea unui împrumut de la un utilizator.

Componentele folosesc state hooks pentru încărcarea asincronă a conținutului, afișarea mesajului de „Loading...” în timpul încărcării și gestionarea stărilor din lista de dorințe, de cărți citite sau de cărți disponibile pentru împrumut. Totodată sunt tratate erori ce pot apărea, ca inexistența unei cărți sau lipsa de recenzii.

În aceeași pagină utilizatorii pot să-și exprime opinia despre o carte pe care au citit-o, adăugând o recenzie și dând o notă de la 1-5 cărții. Acest lucru ajută alți utilizatori să observe părerile cititorilor de pe platformă și să-și facă o idee despre carte înainte de a o citi. Astfel, am implementat un sistem cu accesibil în care utilizatorul are un chenar în care scrie textul recenziei, selectează nota apăsând pe numărul de steluțe corespunzătoare și apasă butonul de submit pentru a trimite recenzia. În momentul acționării butonului prin click, se trimit cererea /api/reviews/add de tip POST către backend, se reîncarcă automat pagina, iar recenzia este adăugată. Un utilizator nu poate adăuga mai multe recenzii la o carte, iar dacă încearcă să afișează un mesaj de eroare sugestiv cu cod de eroare 409 - conflict. Textul recenziei este optional, însă este obligatoriu ca utilizatorul să selecteze un rating.

4.2.10. Explorarea lecturilor după genuri

Deseori, ca cititor, te regăsești în situații în care îți dorești să citești un anumit tip de cărți. Pentru a facilita procesul de alegere, am considerat utilă implementarea unei pagini în care să fie afișate toate cărțile ce aparțin unui anumit gen. Componența `Genre.jsx` împreună cu modulul de stiluri `Genre.module.css` conțin implementarea experienței pe care o va avea utilizatorul cu acest tip de explorare. Funcționalitatea are următoarele caracteristici:

- Odată ajuns pe pagina de descriere a cărții, cititorul poate apăsa pe genul din descriere. Acest simplu click va acționa cererea de tip GET /api/books/genre.
- Rezultatele de pe pagină sunt afișate sub formă de carduri ce includ descrierea cărții (maxim 260 de caractere), titlul, autorul, coperta, rating-ul ca steluțe și un buton ce permite adăugarea cărții la lista de dorințe a utilizatorului.
 - Orice card apăsat trimite pe pagina de descriere a cărții.
 - Dacă cererea primește mai mult de 10 cărți ca rezultat, este implementată funcționalitatea de paginare, prin care utilizatorul poate naviga între paginile aplicației cu ajutorul săgeților din subsol.

4.2.11. Funcția de căutare

Pentru ca cititorii să aibă posibilitatea de a căuta o carte în mod direct după cuvinte cheie, am socotit necesară o funcție de căutare după cuvinte cheie. Implementarea este conținută în fișierul `Search.jsx`, care păstrează stilurile din modulul `Genre.module.css`. Astfel că aspectul este similar cu pagina prezentată anterior, dar cu mici retușuri:

- În loc ca titlul paginii să fie genul se afișează “Results for ...”
- Funcția de căutare este implementată astfel încât să caute siruri de caractere în titlurile cărților, în câmpul de autori, iar apoi în descrierile acestora.

Utilizatorul poate accesa funcția de căutare din orice pagină a aplicației, având bara de căutare în partea de sus a paginii permanent existentă. Prin pagina de căutare am pus accentul pe reutilizare de cod și m-am lipsit de redundanță unui alt modul de stiluri.

4.2.12. Biblioteca personală a utilizatorului

În cadrul paginii `Library.jsx` (cu modulul de stiluri `Library.module.css`) se centralizează cărțile la care utilizatorul a adăugat recenzii, cele pe care le-a citit și cele pe care le-a adăugat la lista de dorințe. Am proiectat pagina ca un panou de control asupra lecturilor personale.

- Trei tab-uri sunt disponibile pe această pagină: recenzii (`/api/library/reviews`), listă de dorințe (`/api/library/library`) și cărți citite (`/api/library/read`), iar utilizatorul poate naviga între aceste 3 pagini după bunul plac.
 - Datele sunt afișate sub formă de carduri, precum în `Genre.jsx` și `Search.jsx`
 - În fiecare dintre liste de cărți citite și cărți preferate, există posibilitatea eliminării cărții prin apăsarea unui buton de “remove”.
 - Se gestionează și cazurile în care sunt întâmpinate erori, iar în cazul în care tab-ul nu are înregistrări prezente, se afișează un mesaj specific.

4.2.13. Gestionarea împrumuturilor

În ceea ce privește împrumuturile, cea mai importantă componentă în privința nouării pe care aplicația o aduce pe piață, aceasta oferă utilizatorilor o interfață în care pot acționa în diferite moduri pentru a duce la bun sfârșit un împrumut. Fluxul de funcționare a sistemului de împrumut este următorul:

- În momentul în care, din meniul de acțiuni al descrierii cărții, un utilizator oferă cartea spre împrumut, se adresează către backend cererea `/api/loans/offer` de tip POST care marchează cartea ca disponibilă pentru împrumut.
- Orice alt utilizator de pe platformă, din același meniu poate cere cartea spre împrumut prin cererea `api/loans/request`.
 - În pagina My Loans, implementată în componenta `MyLoans.jsx`, fiecare utilizator, navigând prin tab-urile, similare cu cele din componenta `Library.jsx`, poate accesa cărțile pe care le-a dat spre împrumut, cele împrumutate de la alți utilizatori, cât și toate cărțile acestea puse la un loc într-o evidență.
 - Fiecare proprietar de carte are dreptul să accepte sau să respingă cererea altui utilizator pentru împrumutul cărții sale prin cererea HTTP de tip PATCH, `/api/loans/<int:loan_id>/status`.
 - Odată acceptat împrumutul, cele două părți pot conversa prin sistemul de chat, vizibil pe cardul cărții.
 - Proprietarul poate marca împrumutul ca finalizat, iar, din acel moment, chat-ul este închis, iar statusul împrumutului este actualizat ca “returned”.
 - Design-ul paginii respectă sistemul de listă implementat și la biblioteca personală a utilizatorului.

4.2.14. Mesageria

În componenta Chat.jsx am implementat un sistem de mesagerie pentru a facilita comunicarea între cele două părți ale împrumutului. Pagina se încarcă prin cererea GET /api/chat/<int:loan_id>, iar funcționalitățile sunt următoarele:

- Mesajele sunt afișate în ordine cronologică
- De-o parte și de cealaltă a imaginii sunt puse mesajele de la utilizatorul conectat și de la cel cu care se află în conversație.
- Pentru actualizarea conversației la fiecare 4 secunde se folosește setInterval.
- Prin apăsarea butonului de Enter sau pe butonul din stânga barei de text, mesajul se trimit și se apelează cererea POST /api/chat/<int:loan_id>.
- Pentru utilizatorul cu care ne aflăm în conversație se afișează numele și avatarul acestuia.
- În bara de navigație se află un buton de întoarcere pe pagina precedentă și butonul de log-out.
 - Orice mesaj gol este blocat înainte de trimitere, nefiind permis un mesaj de acest tip
 - Pentru a se glisa pagina automat la ultimul mesaj am folosit useRef, axios pentru trimiterea și preluarea mesajelor și React Router pentru obținerea identificatorului unic al împrumutului din URL.

4.2.15. Structura principală a frontend-ului

Deși două componente aparent simple, main.jsx și App.jsx reprezintă nucleul aplicației noastre, fără de care utilizatorul nu ar putea interacționa cu platforma. Prin intermediul lor, se definește și se configerează sistemul de rutare. Ele determină, în funcție de rutele active, ce componente îi sunt afișate utilizatorului.

App.jsx funcționează ca un șevalet pe care sunt întins pe rând fiecare dintre componente din main.jsx. Prin intermediul Outlet se afișează componenta copil (“child”) aferentă rutei curente.

main.jsx creează și configerează ruterul principal și randează aplicația în DOM. Sunt utilizate createBrowserRouter și RouterProvider din react-router-dom. Utilizatorul este redirecționat imediat de la rădăcină (root, '/') către pagina de autentificare (/login). Toate paginile expuse mai sus sunt importate și plasate în router pentru ca utilizatorul să le aibă puse la dispoziție când încearcă să acceseze paginile corespunzătoare lor. Interceptorul axios asigură că utilizatorul este scos din cont automat după expirarea token-ului JWT.

4.3. Fluxul de funcționare al aplicației

În momentul accesării aplicației, utilizatorul este redirecționat pe pagina de autentificare. În cazul în care deja are un cont creat, utilizatorul trebuie să-și introducă credențialele, sunt verificate, iar dacă sunt corecte, se accesează automat pagina principală. În caz contrar, utilizatorul poate reintroduce credențialele. Pentru cei care se află prima dată pe platformă, se poate urma procesul de înregistrare care constă în introducerea unor date cerute în cadrul unui formular. Dacă cerințele sunt îndeplinite, noul utilizator se poate autentifica pe platformă urmând procesul descris anterior (Fig. 22).

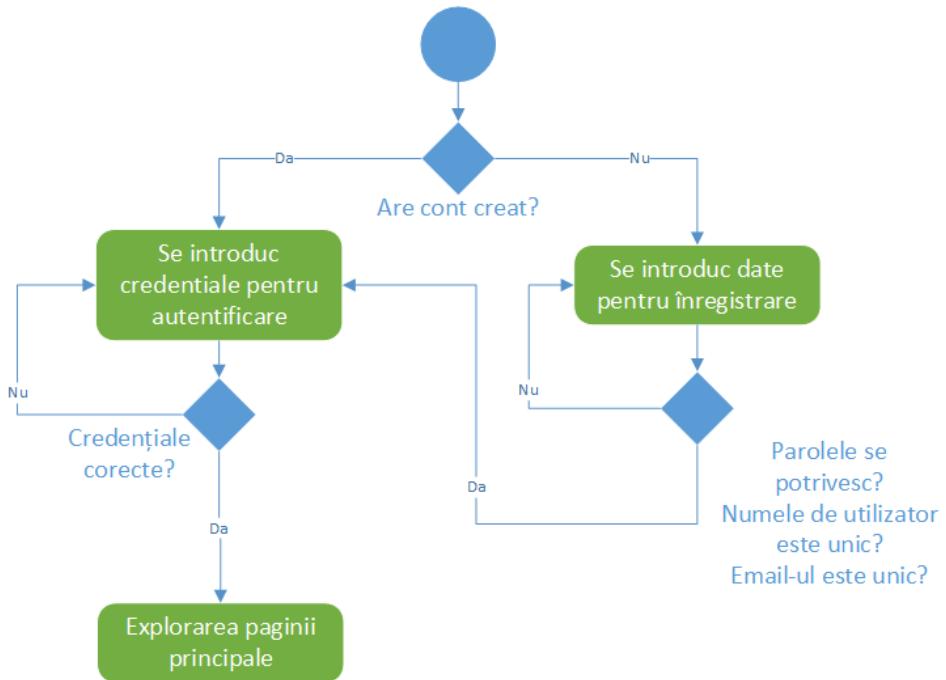


Fig. 22. Activitățile din procesul de autentificare / înregistrare

Pagina principală oferă multe opțiuni utilizatorului. Acesta poate alege dintre a-și verifica biblioteca personală, a selecta și cerceta o carte din recomandări, a căuta cărți în baza de date în funcție de anumite cuvinte cheie, a-și verifica împrumuturile și a ieși efectiv din cont. Pagina principală chiar dacă este complexă în aparență, este totuși o pagină de legătură și un punct de ramificație pentru toate activitățile posibile din flux, aşa că nu prezintă o succesiune de activități foarte amplă (Fig. 23).

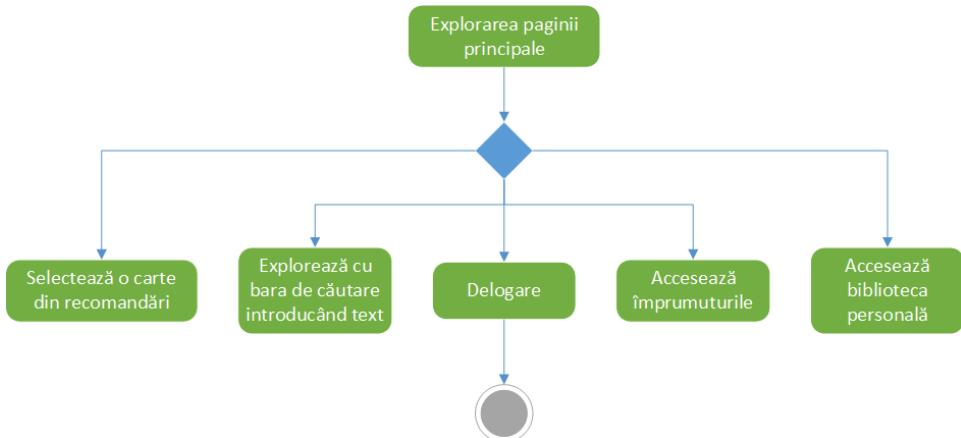


Fig. 23. Diagramă de activități pentru pagina principală

În pagina de descriere a cărții, utilizatorul poate opera multiple activități. Pe lângă a investiga toate detaliile cărții, acestuia i se oferă posibilitatea de a explora cărți din același gen doar apăsând pe genul cărții. Poate marca o carte ca citită sau necitită. Totodată, i se permite să adauge o recenzie unică fiecărei cărți, să-și adauge cartea în lista de dorințe sau să o eliminate, să ofere cărți spre împrumut și chiar să facă cerere de împrumut spre o carte disponibilă (Fig. 24).

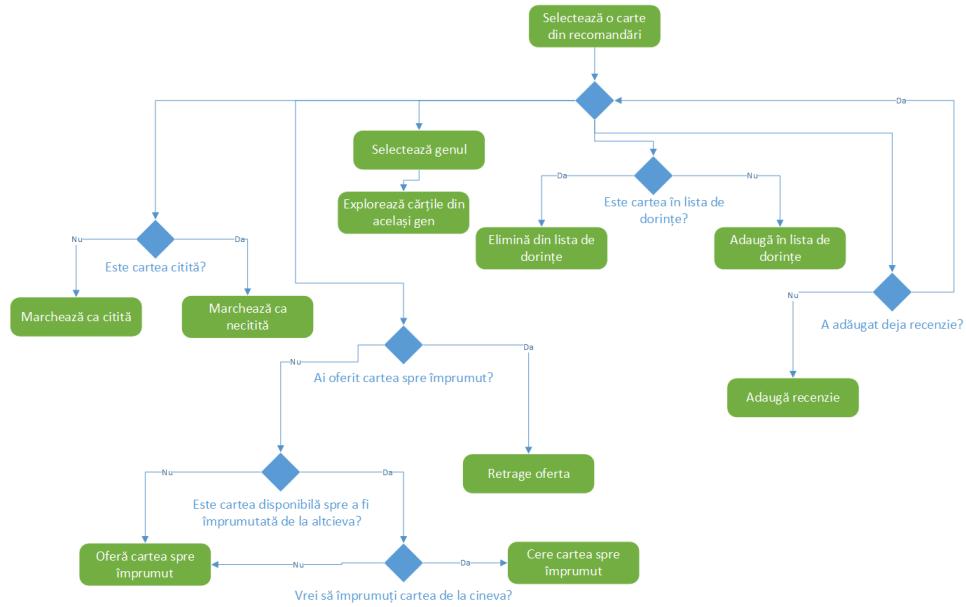


Fig. 24. Fluxul de activități din pagina de descriere a cărții

În biblioteca personală, utilizatorul poate naviga între trei tab-uri: recenzii, lista de dorințe și cărți citite. În lista de recenzii, în afară de a se accesa descrierile cărților, alte activități nu sunt posibile. Lista de dorințe poate fi editată după placul utilizatorului, astfel că acesta are posibilitatea de a șterge cărți din listă, precum și în lista de cărți citite. Printre aceste tab-uri se poate naviga în voie. Din fiecare tab, utilizatorul are acces la celelalte taburi.

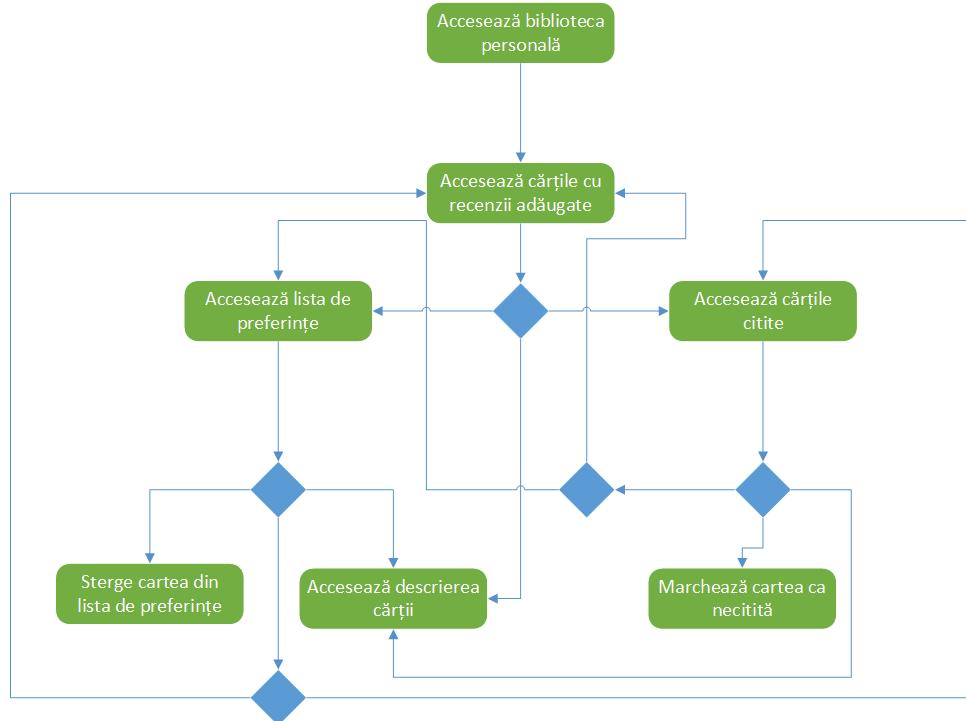


Fig. 25. Fluxul activităților din pagina de bibliotecă personală

Împrumuturile respectă același format ca pagina de bibliotecă personală, prezentând 3 tab-uri pentru navigare. Utilizatorul poate să aleagă să vizualizeze cărțile primite ca

împrumut, cărțile date ca împrumut și toate împrumuturile în care a fost și este angrenat. Dacă o carte i-a fost oferită spre împrumut, iar împrumutul a fost aprobat de proprietarul cărții, utilizatorul va avea posibilitatea să deschidă chat-ul prin care să converseze cu proprietarul (Fig. 26). Istoricul mesajelor este salvat, iar acestea sunt afișate sub formă de listă, cronologic, de o parte și de cealaltă a ecranului. Prin apăsarea săgeții din stânga sus a paginii, utilizatorul poate reveni la pagina anterioară. În cazul în care utilizatorul a oferit o carte spre împrumut, în momentul când statusul împrumutului este “în așteptare”, utilizatorul are dreptul, ca proprietar al cărții, să accepte sau să refuze cererea de împrumut a unui alt utilizator al platformei. Dacă decide să accepte, este deschis chat-ul pentru a comunica și poate transmite faptul că acea carte a fost returnată. Toată această logică este vizibilă și în diagrama din Fig. 26.

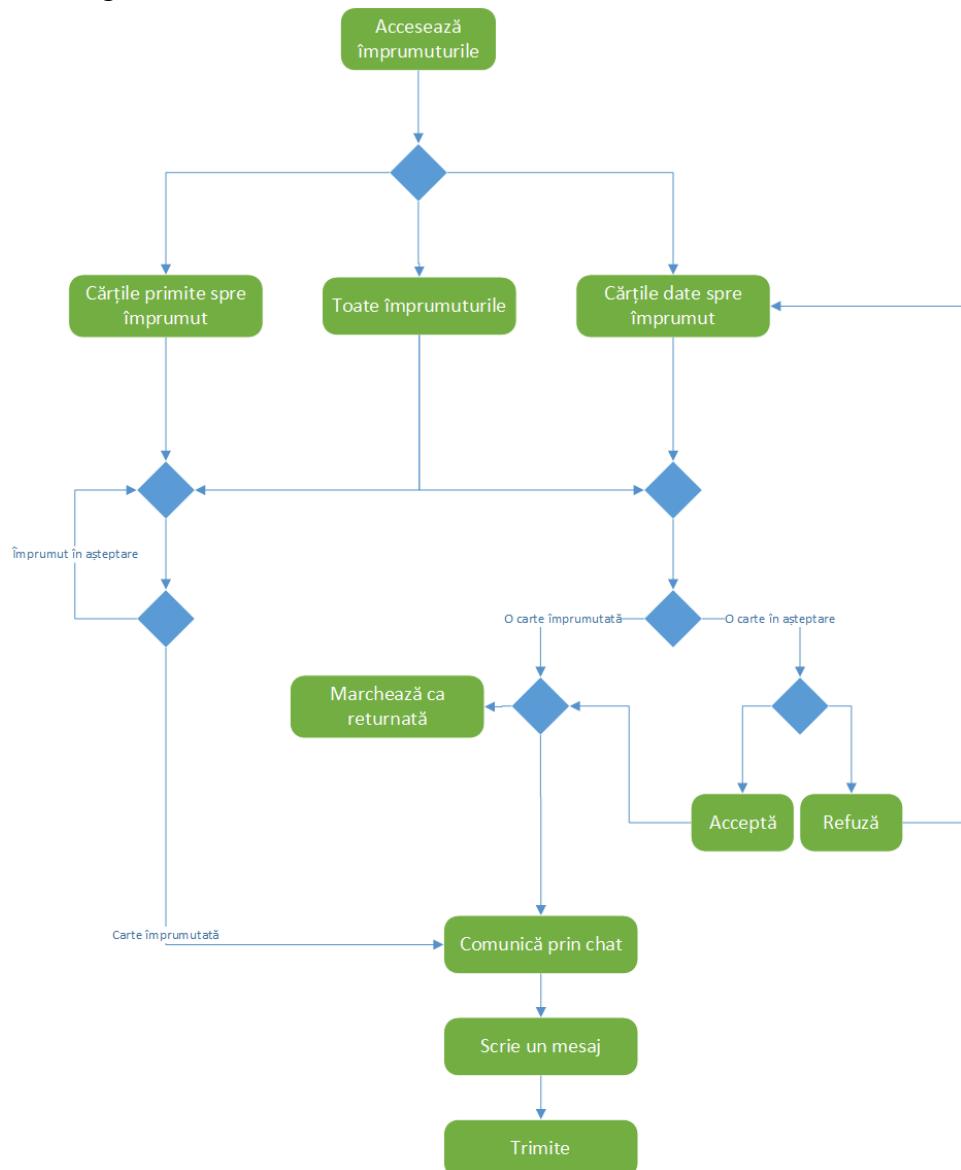


Fig. 26. Diagrama UML de activități din cadrul paginii de împrumuturi

4.4. Rezultate obținute

În cadrul lucrării am reușit să obțin rezultate pe mai multe planuri, de la documentarea și cercetarea domeniului, până la implementarea și testarea completă a aplicației. Realizările sunt semnificative, mai ales privind în perspectivă cu ce mi-am propus încă din început, odată cu ideea proiectului.

Cu ajutorul a mai multor articole din mediul online publicate de doctori, oameni de știință, profesori și specialiști, am reușit să conturez un context în care se află domeniul împrumutului de cărți. Am dezvoltat cu succes o analiză extinsă asupra comportamentului utilizatorilor în raport cu bibliotecile online și relevanța sistemelor de recomandări performante pentru aceștia. Am studiat funcționalitățile și popularitatea aplicațiilor mobile și web existente, cât și platformele digitale de împrumut și distribuție de eBook-uri. Am surprins aspectele biblioteconomiei și am examinat un studiu de specialitate desfășurat la o universitate de prestigiu din Beijing. Toate acestea au fost fundamentale pentru crearea unui punct de plecare pentru proiectul meu.

Investigând rețetele de succes de la alte aplicații, m-am inspirat și am conceptualizat de la zero toată experiența utilizatorului și interfața cu care acesta ar trebui să ia contact. Am construit o structură logică și am populat baza de date. Am trecut setul de date printr-o etapă de preprocesare pentru a pregăti datele brute și a le face utilizabile și relevante pentru algoritmul și aplicația finală. Totodată, am gândit fluxul de utilizare a aplicației, ce presupune navigarea între pagini și toate tipurile de interacțiuni.

Implementarea a fost o reușită semnificativă mai ales prin prisma sistemului de recomandări bazat pe conținut, utilizând metode de învățare automată. Am proiectat arhitectura aplicației și am făcut-o modulară atât pe backend cât și pe frontend. Am implementat un API REST, folosind Flask și am tratat toate cererile pentru a asigura funcționalitate totală a aplicației. Am utilizat JWT pentru autentificare și protejare de rute și Flask Migrate pentru a asigura cele mai recente versiuni în structura bazei de date. Frontend-ul l-am construit cu ajutorul React și cu module CSS pentru stiluri și un design consistent. Axios a fost utilizat pentru comunicarea cu backend-ul, iar pentru animații moderne, carusele și interacțiuni am folosit Framer Motion.

Aproape toate funcționalitățile propuse au fost implementate cu succes iar fluxul de funcționare este unul reușit. Utilizatorul are posibilitatea de a se autentifica sau înregistra pe platformă urmând apoi să aibă acces la toată baza de date de cărți. Se poate bucura de recomandările primite din aplicație sau poate căuta cărți specifice, chiar și în funcție de gen. I se pune la dispoziție posibilitatea de a-și adăuga cărți în lista de dorințe, le poate marca drept citite deja sau le poate oferi spre împrumut. Toate detaliile necesare despre cărți sunt puse în pagina de descriere, unde pot fi observate chiar și recenziile utilizatorilor cu note de la 1-5. Sistemul de împrumut este bine gândit beneficiind chiar și de un sistem de comunicare. Iar la toate cărțile preferate ale utilizatorului se ține evidența într-o bibliotecă personală accesibilă tot timpul din bara de navigație a platformei.

5. Concluzii și dezvoltări ulterioare

Originalitatea lucrării

Așadar, am creat o aplicație ce nu doar că își propune să promoveze lectura ca formă de terapie, ci are ca scop aducerea laolaltă a oamenilor și avansarea ideii că putem colabora chiar și în domeniile în care nu ne așteptam. Granița dintre oameni nu este aşa de mare atunci când au plăceri comune și interese asemenea. Cred cu tărie că dacă un cititor înrăit reușește să dea peste această aplicație, îi va schimba modul de a se raporta la lectură. Pe lângă un sistem de recomandări specializat, aspectul social din spatele aplicației este de fapt cel mai important.

Originalitatea aplicației constă în mai multe aspecte, dar cele principale sunt sistemul de recomandare baza pe conținutul textual utilizând algoritmul TF-IDF și similaritatea cosinus și funcționalitatea de împrumut dotată cu un sistem de chat.

Motorul de recomandare a cărților bazat pe analiza conținutului textual, utilizând algoritmi de text-mining (TF-IDF) și similaritate cosinus este o abordare ce diferă de cea a multor aplicații existente, deoarece acestea se bazează de obicei pe filtrare colaborativă sau preferințele explicate ale utilizatorilor și nu pe analiza semantică a textului descrierilor, titlurilor, genurilor cărților. Nici Goodreads și nici alternativele acestei aplicații nu par să folosească acest tip de algoritmi pentru a propune lecturi similare. Utilizarea TF-IDF împreună cu similaritatea cosinus rezultă în faptul că aplicația citește descrierile cărților și găsește asemănări cu restul cărților înainte ca utilizatorul să le fi asociat. Abordarea aceasta bazată pe conținutul efectiv umple un gol pe care mulți cititori îl reclamă: dorința de a primi sugestii pentru ce este o carte și nu pentru cine și câți alți oameni au mai citit cartea aceea.

Multe rețele de socializare cu direcție literară au pus accent pe recomandări și pe urmărirea lecturilor de-a lungul timpului, și nu pe facilitatea împrumutului de cărți fizice. Funcționalitatea de împrumut este un element de diferențiere clar față de alte aplicații, iar din acest motiv se poate observa că proiectul de față aduce o contribuție originală, integrând dimensiunea socială în cadrul unei aplicații web.

Dezvoltări ulterioare

Pe baza a ceea ce am implementat deja, aplicația are un potențial enorm de extindere, deși nu aş vrea să o umplu cu multe funcționalități inutile și obositoare pentru cititorul obișnuit sătul de atâtă tehnologie cu care este înconjurat. Așa că, pe parcursul procesului de creație al aplicației, am găsit multe idei ce pot ajuta aplicația să se dezvolte în viitorul apropiat.

O îmbunătățire necesară ar fi adăugarea funcționalității de localizare geografică, ce va permite utilizatorilor să introducă direct în aplicație orașul în care se află. Facilitatea aceasta va ajuta în principal pentru filtrarea ofertelor de împrumut în funcție de proximitate și optimizarea logisticii pentru livrarea cărților. Procesul de împrumut va deveni astfel mai practic și nu va rămâne la stadiul de concept.

În versiunea actuală a aplicației, utilizatorul nu are acces la niciun tip de profil personal, deși această opțiune este una necesară la aproape toate aplicațiile. Astfel că îmi propun ca pe viitor să implementez o pagină specială pentru profilul utilizatorului care să includă fotografie de profil și posibilitatea de a edita informațiile din cont, inclusiv locația geografică propusă anterior.

Pentru că dualitatea și greșelile umane sunt de înțeles, în aplicațiile sociale se solicită deseori posibilitatea de a putea edita recenziile. Această dezvoltare ar ajuta pe utilizatori să-și revizuiască recenziile și notele, schimbându-și impresiile între timp, și chiar să șteargă o anumită recenzie pe care poate nu o mai consideră relevantă.

Baza de date a platformei conține între 6000 și 7000 de cărți la acest moment și consider extinderea ei o necesitate pe viitor. Pentru ca platforma să fie cât mai atractivă pentru utilizatori oferta de lectură trebuie să fie diversă aşa că există posibilitatea integrării unui API extern cum ar fi Google Books sau Open Library. Totodată, deoarece unii dintre utilizatori au drepturi de administratori ai site-ului, acest atribut poate fi pus în valoare prin permiterea acestora să adauge cărți noi chiar din platformă.

În momentul lansării aplicației către publicul larg, nu în ultimul rând, va fi necesară formularea unor termene și condiții. Aceștia vor stabili, printre altele, regulile de utilizare ale aplicației, protecția datelor personale și respectarea drepturilor de autor.

Coroborând toate aspectele de mai sus, lucrarea de față a urmărit dezvoltarea unei aplicații web inovatoare de tip bibliotecă online colaborativă care are ca scop să faciliteze împrumutul de cărți fizice între cititori și oferă recomandări bazate pe conținutul textual al cărților. Am pornit de la o analiză a domeniului și a nevoilor reale ale cititorilor și am integrat componente de backend și de frontend folosind tehnologii actuale. Prin abordarea propusă, aplicația se diferențiază de aplicațiile prezente pe piață și demonstrează cât de util poate fi un model centrat pe socializare în comunitate, susținut de tehnologie accesibilă.

6. Bibliografie

1. Lima Rocha, A., Lima Correia, L., Aydin, D., & Bengü Göncü, H. (2025). From Childhood Experiences to Social Media Addiction: Unraveling the Impact on Adolescents. *Children 2025, Vol. 12, Page 1-17*
2. *How Social Media Affects Your Teen's Mental Health: A Parent's Guide* > News > Yale Medicine. (n.d.). Retrieved March 25, 2025, from <https://www.yalemedicine.org/news/social-media-teen-mental-health-a-parents-guide>
3. 'Epidemic' of loneliness and isolation plaguing 1 in 4 young Australians | news.com.au — Australia's leading news site. (n.d.). Retrieved March 25, 2025, from <https://www.news.com.au/lifestyle/epidemic-of-loneliness-and-isolation-plaguing-1-in-4-young-australians/news-story/05704cfe8119a672984746d905f46f8c>
4. Colombo, E., Mercorio, F., Mezzanzanica, M., & Serino, A. (2024). Towards the Terminator Economy: Assessing Job Exposure to AI through LLMs. <https://arxiv.org/abs/2407.19204v1>
5. Global paper and paperboard production 2023 | Statista. (n.d.). Retrieved March 22, 2025, from <https://www.statista.com/statistics/270314/global-paper-and-cardboard-production/>
6. Why Do We Still Read Books? | Psychology Today. (n.d.). Retrieved March 22, 2025, from <https://www.psychologytoday.com/us/blog/psychology-yesterday/202405/why-do-we-still-read-books>
7. Novak, J. (2013). eBook Lending Platforms. *Against the Grain*, 25(6), 22-26.

8. Breeding, Marshall. (2021). Smarter Libraries through Technology: The State of Digital Lending in Public Libraries. *Smart Libraries Newsletter* 41 (8), 1-2.
9. Hjorland, B. (2018). Library and information science (LIS), Part 1. In *Knowledge Organization* (Vol. 45, Issue 3, pp. 232–254). International Society for Knowledge Organization. <https://doi.org/10.5771/0943-7444-2018-3-232>
10. You, X. (2019). Architecture and Research of Smart Book Management System Based on Artificial Intelligence Technology. *International Journal of New Developments in Engineering and Society*, 4, 91–96. <https://doi.org/10.25236/IJNDES.040218>
11. *Tom Kabinet: CJEU rules resale of e-books requires permission of copyright holder - Lexology*. (n.d.). Retrieved June 13, 2025, from <https://www.lexology.com/library/detail.aspx?g=7420597a-d24a-461c-983a-c62a457f2763>
12. *CURIA - Documents*. (n.d.). Retrieved June 13, 2025, from <https://curia.europa.eu/juris/document/document.jsf?text=&docid=180332&doclang=EN>
13. *Cum a ajuns Bookster să câștige din împrumutul volumelor cât unele edituri care le tipăresc*. (n.d.). Retrieved June 13, 2025, from <https://romania.europalibera.org/a/scandal-bookster--edituri/33096283.html>
14. *WIPO Lex*. (n.d.). Retrieved June 13, 2025, from <https://www.wipo.int/wipolex/en/text/545968>
15. *The Best Apps to Track Your 2025 Reading Progress*. (n.d.). Retrieved April 2, 2025, from <https://www.sweetyhigh.com/read/the-best-books-to-track-your-2025-reading-journey-010725>.
16. Li, Y., Wang, S., & Guo, W. (2015). A study on use and acceptance of book apps among college students in a university. *Communications in Computer and Information Science*, 559, 152–160. https://doi.org/10.1007/978-3-662-48978-9_15/FIGURES/11
17. Angulo, A., Goldstein, N., & Norton, M. (2024). *Friendship fallout and bailout backlash: The psychology of borrowing and lending*. <https://doi.org/10.1002/jcpy.1410>
18. *How to Reach More Readers on Goodreads | News for Authors*. (n.d.). Retrieved May 27, 2025, from <https://authornews.penguinrandomhouse.com/how-to-reach-more-readers-on-goodreads/>
19. Price, S. (2021). *(11) Goodreads: A Case study in Outdated UX/UI | LinkedIn*. <https://www.linkedin.com/pulse/goodreads-case-study-outdated-uxui-sarah-price/>
20. *7k Books*. (n.d.). Retrieved June 5, 2025, from <https://www.kaggle.com/datasets/dylanjcastillo/7k-books-with-metadata>
21. *Getting Started | Vite*. (n.d.). Retrieved June 5, 2025, from <https://vite.dev/guide/>
22. *React*. (n.d.). Retrieved June 6, 2025, from <https://react.dev/>
23. *Writing Markup with JSX – React*. (n.d.). Retrieved June 6, 2025, from <https://react.dev/learn/writing-markup-with-jsx>
24. *What is Flask Python - Python Tutorial*. (n.d.). Retrieved June 6, 2025, from <https://pythonbasics.org/what-is-flask-python/>
25. *sqlite3 — DB-API 2.0 interface for SQLite databases — Python 3.13.4 documentation*. (n.d.). Retrieved June 9, 2025, from <https://docs.python.org/3/library/sqlite3.html>

26. *How To Safely Store A Password* | codahale.com. (n.d.). Retrieved June 6, 2025, from <https://codahale.com/how-to-safely-store-a-password/>
27. *Flask-Bcrypt — Flask-Bcrypt 1.0.1 documentation*. (n.d.). Retrieved June 6, 2025, from <https://flask-bcrypt.readthedocs.io/en/1.0.1/>
28. *HTTP request methods explained*. (n.d.). Retrieved June 7, 2025, from <https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/HTTP-methods>
29. *Understanding TF-IDF (Term Frequency-Inverse Document Frequency) - GeeksforGeeks*. (n.d.). Retrieved June 7, 2025, from <https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/>
30. Lahitani, A. R., Permanasari, A. E., & Setiawan, N. A. (2016). Cosine similarity to determine similarity measure: Study case in online essay assessment. *Proceedings of 2016 4th International Conference on Cyber and IT Service Management, CITSM 2016*. <https://doi.org/10.1109/CITSM.2016.7577578>
31. *Welcome to Alembic's documentation! — Alembic 1.16.1 documentation*. (n.d.). Retrieved June 9, 2025, from <https://alembic.sqlalchemy.org/en/latest/>
32. *Tutorial — Alembic 1.16.1 documentation*. (n.d.). Retrieved June 9, 2025, from <https://alembic.sqlalchemy.org/en/latest/tutorial.html#the-migration-environment>